HABITUS 2.1.0

HABITUS Başvuru Kitabı

Ekranlar, parametreler ve sonuçlara etkileri

Seda ÖRÜCÜ & Ömer K. ÖRÜCÜ

1. Giriş ve iş akışı
2. Veri I: kayıtlar, katmanlar ve çalışma sınırı (① Data)
3. Veri II: arka plan, yalancı yokluk, erişilebilir alan ve seyreltme (① Data)
4. Değişkenler (② Variables)
5. Modeller I: algoritmalar ve ayarları (③ Models → Algorithm Settings)
6. Modeller II: çapraz doğrulama, eşik, topluluk ve sonuçlar (③ Models)
7. Değerlendirme (④ Evaluation)
8. Gelecek senaryoları (⑤ Future Scenarios)
9. Alan değişimi (⑥ Range Change)
10. Bağımsız doğrulama (⑦ Validation)
11. Rapor, ODMAP ve çok türlü çalışma (⑧ Report)
12. Harita, Yardım ve çıktı dosyaları (🗺 Map, ⓘ Help)

Kaynak kod atıfları HABITUS 2.1.0 içindir · Derleme: 21.11.2026

İçindekiler

1. Giriş ve iş akışı 7
1.1 HABITUS nedir 7
1.2 Bu kitap nasıl kullanılır 7
1.3 Kurulum ve platformlar 8
1.4 Açılış 8
1.5 Ana pencerenin düzeni 9
1.6 Sekme şeridi ve kilit düzeni 11
1.7 Alt durum şeridi: ilerleme çubuğu ve günlük 14
1.8 İlerleme, iptal ve pencerenin kapanması 14
1.9 Çıktı klasörü ve otomatik analiz klasörü 15
1.10 Rastgele tohum ve tekrarlanabilirlik 16
1.11 Örnek veri 16
1.12 Uzman soruları 17
1.13 Sık hatalar ve tuzaklar 18
1.14 Kaynaklar 18

2. Veri I: kayıtlar, katmanlar ve çalışma sınırı (① Data) 19
2.1 Bu sekme ne yapar 19
2.2 Ekranın düzeni 20
2.3 Tür kayıtları: Occurrence Data (CSV) 21
2.4 Öngörücü katmanlar: Continuous Rasters ve Categorical Rasters 24
2.5 Çalışma sınırı: Study Boundary (mask, optional) 27
2.6 Tekrarlanabilirlik: Reproducibility 29
2.7 Çıktı klasörü: Output Directory 29
2.8 Load Data düğmesi: işlem sırası ve kayıt ayıklama 30
2.9 Çıktılar ve sonraki adıma geçenler 33
2.10 Uzman soruları 33
2.11 Sık hatalar ve tuzaklar 35
2.12 Kaynaklar 35

3. Veri II: arka plan, yalancı yokluk, erişilebilir alan ve seyreltme (① Data) 36
3.1 Bu bölüm ne anlatır 36
3.2 Ekranın düzeni 36
3.3 Yalancı yokluklar: ML Models 37
3.4 Kendi yokluk kayıtlarınız: Own Absences 41
3.5 MaxEnt arka planı: MaxEnt 43
3.6 Örnekleme yanlılığı: Sampling Bias 45
3.7 ENFA arka planı: ENFA / Mahalanobis 47
3.8 Erişilebilir alan: Study Area & Thinning (üst yarı) 48
3.9 Kayıtların seyreltilmesi: Study Area & Thinning (alt yarı) 51
3.10 Çıktılar ve sonraki adıma geçenler 52
3.11 Uzman soruları 52
3.12 Sık hatalar ve tuzaklar 54
3.13 Kaynaklar 54

4. Değişkenler (② Variables) 56
4.1 Bu sekme ne yapar 56
4.2 Ekranın düzeni 56
4.3 Hesaplar hangi örnek üzerinde yapılır 57

4.4 Analiz ayarları (Analysis Settings) 58
4.5 Yüksek korelasyonlu çiftler ve ısı haritası 60
4.6 Değişken listesi ve toplu seçim 61
4.7 Onay (Confirm) ve kayıt/öngörücü denetimi 64
4.8 Dışa aktarma 65
4.9 Gelişmiş analiz (Advanced Analysis) 66
4.10 Çıktılar ve sonraki adıma geçenler 71
4.11 Uzman soruları 71
4.12 Sık hatalar ve tuzaklar 73
4.13 Kaynaklar 73

5. Modeller I: algoritmalar ve ayarları (③ Models → Algorithm Settings) 75
5.1 Bu sekme ne yapar 75
5.2 Ekranın düzeni 76
5.3 Algoritma seçimi (Select Algorithms) 77
5.4 Bütün algoritmalar için ortak kurallar 77
5.5 Algoritma seçme rehberi 80
5.6 GLM: genelleştirilmiş doğrusal model 82
5.7 GBM: gradyan artırma (scikit-learn) 83
5.8 BRT: artırılmış regresyon ağaçları 85
5.9 RF: rastgele orman 86
5.10 SVM: destek vektör makinesi 88
5.11 ANN: yapay sinir ağı (çok katmanlı algılayıcı) 89
5.12 XGBoost: aşırı gradyan artırma 90
5.13 LightGBM: hafif gradyan artırma 91
5.14 CatBoost: kategorik artırma 93
5.15 GAM: genelleştirilmiş toplamsal model 94
5.16 MaxEnt: en yüksek entropi 95
5.17 ENFA: ekolojik niş faktör analizi 98
5.18 Mahalanobis uzaklığı 99
5.19 BIOCLIM: iklim zarfı 100
5.20 Domain: Gower uzaklığı 101
5.21 GLMNET: elastik ağ lojistik regresyon 102
5.22 CART: sınıflandırma ağacı 103
5.23 ESM: küçük modeller topluluğu 104
5.24 Çıktılar ve sonraki adıma geçenler 106
5.25 Uzman soruları 106
5.26 Sık hatalar ve tuzaklar 109
5.27 Kaynaklar 109

6. Modeller II: çapraz doğrulama, eşik, topluluk ve sonuçlar (③ Models) 112
6.1 Bu sekme ne yapar 112
6.2 Ekranın düzeni 113
6.3 Algoritma seçimi (Select Algorithms) 114
6.4 Eğitim ayarları (Training Settings) 115
6.5 Değerlendirme ayarları: eşik, omisyon referansı, Boyce 121
6.6 Topluluk ayarları (Combining Results) 124
6.7 Ayar araması (Tuning (optional)) 127
6.8 Değerlendirme sonuçları (Evaluation Results) 129
6.9 Bugünkü dağılım haritası (Current Distribution Map) 130
6.10 Çıktılar ve sonraki adıma geçenler 131
6.11 Uzman soruları 132

6.12 Sık hatalar ve tuzaklar 134
6.13 Kaynaklar 134

7. Değerlendirme (④ Evaluation) 136
7.1 Bu sekme ne yapar 136
7.2 Ekranın düzeni 137
7.3 Model skorları (Model Scores) 138
7.4 ROC eğrileri (ROC Curves) 140
7.5 Omisyon, kesinlik-duyarlılık ve kalibrasyon (Omission & PR) 141
7.6 Değişken önemi (Variable Importance) 142
7.7 Tepki eğrileri (Response Curves) 143
7.8 Boyce ayrıntısı (Boyce Detail) 145
7.9 Sınıflandırma eşikleri (Classification Thresholds) 146
7.10 Alt düğmeler: yenileme, kaydetme 147
7.11 Çıktılar ve sonraki adıma geçenler 148
7.12 Uzman soruları 148
7.13 Sık hatalar ve tuzaklar 150
7.14 Kaynaklar 150

8. Gelecek senaryoları (⑤ Future Scenarios) 151
8.1 Bu sekme ne yapar 151
8.2 Ekranın düzeni 152
8.3 Bilgi şeridi 153
8.4 Senaryo adı ve GCM (Projection Scenario) 153
8.5 Projekte edilecek algoritmalar (Algorithms to Project) 154
8.6 Raster listesi ve sıra kuralı (Environmental Rasters for This Scenario) 154
8.7 Çıktı ve görüntüleme ayarları (Output Settings) 156
8.8 Aktarım güvenliği seçenekleri: MESS, MESS maskesi, clamping 157
8.9 ExDet yenilik yüzeyleri (her zaman yazılır) 159
8.10 Izgara hizalama, çalışma sınırına kırpma ve şeritli projeksiyon 159
8.11 Çıktılar ve Completed Projections tablosu 161
8.12 Harita görüntüleyici (Projection Map Viewer) 162
8.13 Çıktılar ve sonraki adıma geçenler 163
8.14 Uzman soruları 163
8.15 Sık hatalar ve tuzaklar 165
8.16 Kaynaklar 165

9. Alan değişimi (⑥ Range Change) 166
9.1 Bu sekme ne yapar 166
9.2 Ekranın düzeni 166
9.3 İkili haritaların seçimi (Input Binary Projection Rasters) 167
9.4 Hızlı doldurma (Quick Fill from Completed Projections) 167
9.5 Analiz adı (Output · Analysis name) 168
9.6 Değişimin sayıldığı kapsam (Count change within) 168
9.7 Sonuç kodları ve haritanın yazılması 169
9.8 İstatistik tablosu (Range Change Statistics) 170
9.9 Harita 172
9.10 Hangi topluluk haritası seçilmeli: EMwmean_bin mi, EMca_bin mi? 172
9.11 Dağılım (dispersal) varsayımı 173
9.12 Çıktılar ve sonraki adıma geçenler 173
9.13 Uzman soruları 174
9.14 Sık hatalar ve tuzaklar 175

9.15 Kaynaklar 175

10. Bağımsız doğrulama (⑦ Validation) 176

10.1 Bu sekme ne yapar 176
10.2 Ekranın düzeni 177
10.3 Harita seçimi (1 Map Selection) 178
10.4 Karşılaştırma modu ve yeniden sınıflama (2 Reclassification) 179
10.5 Örnekleme tasarımı (3 Sampling Settings) 180
10.6 Çalıştırma (Run Validation) 182
10.7 Ölçütler ve sonuç metni 183
10.8 Dışa aktarma 185
10.9 Uzman soruları 185
10.10 Sık hatalar ve tuzaklar 187
10.11 Kaynaklar 187

11. Rapor, ODMAP ve çok türlü çalışma (⑧ Report) 188

11.1 Bu sekme ne yapar 188
11.2 Ekranın düzeni 188
11.3 Rapor seçenekleri (Report Sections) 189
11.4 Generate HTML Report ve Open in Browser 190
11.5 HTML raporun bölümleri 190
11.6 ODMAP dışa aktarımı (Export ODMAP protocol) 194
11.7 Yayın haritaları 195
11.8 Çok türlü toplu çalıştırma (Run Selected Species…) 196
11.9 Çıktılar 197
11.10 Uzman soruları 197
11.11 Sık hatalar ve tuzaklar 198
11.12 Kaynaklar 198

12. Harita, Yardım ve çıktı dosyaları (🗺 Map, ⓘ Help) 199

12.1 Bu bölüm ne anlatır 199
12.2 🗺 Map sekmesinin yeri 199
12.3 🗺 Map sekmesi 200
12.4 ⓘ Help sekmesi 203
12.5 Çıktı klasörünün dosya kataloğu 206
12.6 Koşu kaydı: habitus_run_config.json 209
12.7 Günlükler 211
12.8 Ortam değişkenleri 212
12.9 Sorun giderme 212
12.10 Sistem gereksinimleri 215
12.11 Uzman soruları 215
12.12 Sık hatalar ve tuzaklar 217
12.13 Kaynaklar 217

*Annemiz **Nevin Örücü**'nün anısına...*

14.09.2025

1. Giriş ve iş akışı

Kapsam: ana pencere · Kod: `main.py` (açılış), `main_dialog.py` (ana pencere, sekme kilitleri, günlük), `threads.py` (arka plan işleri), `run_config.py` (tohum ve koşu kaydı), `tabs/tab_data.py` (tohum ve çıktı klasörü alanları)

1.1 HABITUS nedir

HABITUS, tür dağılım modellemesi (species distribution modelling, SDM) için yazılmış, kod yazmayı gerektirmeyen bir masaüstü programıdır. Bir türün bulunduğu yerlerin koordinatlarını (sunum kayıtları, CSV) ve aynı ızgaradaki çevresel raster katmanlarını alır; bunlardan bir uygunluk (habitat suitability) modeli kurar, modeli çapraz doğrulamayla değerlendirir, bugünkü ve gelecekteki iklim için harita üretir, iki harita arasındaki alan değişimini hesaplar ve bütün adımları bir HTML raporda toplar (Elith ve Leathwick, 2009).

Hesaplar Python ile yapılır: scikit-learn, xgboost, lightgbm, catboost, pygam ve elapid kütüphaneleri ile NumPy. R kurulumu gerekmez ve R paketleri çağrılmaz. Kitapta R paketleri (biomod2, dismo, ENMeval, blockCV, maxnet, ecospat) yalnız karşılaştırma için anılır.

Programın dışında kalan iki iş vardır. Birincisi, sunum kayıtlarının toplanması ve temizlenmesi. İkincisi, çalışma sınırı verilmediğinde rasterların ortak bir ızgaraya getirilmesi: bütün sürekli ve kategorik rasterlar aynı boyut, hücre boyutu, orijin ve koordinat sistemine (CRS) sahip olmalıdır. HABITUS bunu yüklemede denetler ve farklı olan dosyayı adlandırır, ama yeniden örneklemez (Bölüm 2).

> **Not:** Arayüz İngilizcedir. Bu kitapta düğme, kutu ve sekme adları arayüzde yazdığı gibi `kod` biçiminde verilir; ilk geçişte yanına Türkçe karşılığı yazılır.

1.2 Bu kitap nasıl kullanılır

Kitap, programın sekmelerini soldan sağa izler. Her bölüm aynı düzendedir: sekmenin iş akışındaki yeri, ekranın düzeni, alan grupları ve parametre tabloları, sonucu etkileyen her parametre için dört başlıklı açıklama (**Ne işe yarar**, **Ne zaman, nasıl ayarlanır**, **Sonuca etkisi**, **Dikkat**), çıktılar, uzman soruları, sık hatalar ve kaynaklar.

- Parametre tablolarındaki varsayılan değerler ve aralıklar kaynak koddan okunmuştur; `Kod` sütunu `dosya.py:satır` biçiminde yeri gösterir.
- Ekran görüntüleri ve metinde verilen sayılar örnek bir koşudandır: *Abies nordmanniana* (626 kayıt), WorldClim 2.1 biyoiklim değişkenleri (2,5 dakika), gelecek senaryosu MPI-ESM1-2-HR SSP5-8.5 2081-2100, algoritmalar GLM, GAM, RF ve MAXENT. Bu sayılar arayüzü göstermek içindir; türün ekolojisi hakkında bir sonuç değildir.
- İlk kez kullanan okur bu bölümü, ardından Bölüm 2-6'yı sırayla okumalıdır. Bir ayarın anlamını arayan okur doğrudan ilgili bölümün parametre tablosuna gidebilir. Çıktı dosyalarının tam listesi, koşu kaydı ve sorun giderme Bölüm 12'dedir.

Bölümlerin sekmelere dağılımı:

Bölüm	Sekme	Konu
2-3	① `Data`	Kayıtlar, rasterlar, çalışma sınırı, sözde yokluk ve arka plan
4	② `Variables`	VIF, korelasyon, gelişmiş çoklu bağlantı analizi, değişken seçimi
5-6	③ `Models + Current Map`	Algoritmalar, çapraz doğrulama, eşik, topluluk, Tuning, güncel harita

Bölüm	Sekme	Konu
7	`④ Evaluation`	Skorlar, ROC, omisyon, değişken önemi, tepki eğrileri, Boyce, eşikler
8	`⑤ Future Scenarios`	Gelecek iklime projeksiyon, ekstrapolasyon tanısı
9	`⑥ Range Change`	Kayıp, kazanç ve kararlı alan
10	`⑦ Validation`	Bağımsız harita ya da noktalarla doğruluk
11	`⑧ Report`	HTML rapor, ODMAP, çok türlü dosyada toplu çalıştırma
12	`🗺 Map`, `ⓘ Help`	Kayıt haritası, yardım, çıktı dosyaları, koşu kaydı, sorun giderme

1.3 Kurulum ve platformlar

HABITUS üç masaüstü platformunda çalışır. Paketli sürüm kendi Python yorumlayıcısını ve bütün kütüphanelerini taşır; ayrıca Python kurmak gerekmez.

Platform	Paket	Kurulum
Windows	Kurulum programı (`HABITUS_Setup_v*.exe`) ya da taşınabilir ZIP	Kurulum programını çalıştırın; program Başlat menüsünden ve masaüstü kısayolundan açılır. SmartScreen uyarısında `More info` ve `Run anyway` seçilir
macOS	`.dmg` disk görüntüsü (Apple Silicon ve Intel)	`HABITUS.app` dosyasını `Applications` klasörüne sürükleyin; ilk açılışta sağ tık ve `Open`
Linux	`AppImage` ya da `tar.gz`	AppImage dosyasına çalıştırma izni verip çalıştırın; tar.gz arşivinde program `run_habitus.sh` ile başlatılır
Kaynak koddan	`main.py`	Bağımlılıkların kurulu olduğu bir Python ortamında `python main.py`

Kaynaktan çalıştırmak için gereken en düşük Python sürümü 3.10'dur (`pyproject.toml:11`). İsteğe bağlı algoritma kütüphaneleri (xgboost, lightgbm, catboost, pygam, elapid) kurulu değilse program yine açılır; ilgili algoritmalar `③ Models` sekmesinde kullanılamaz olarak işaretlenir (`main.py:365-367`). Ayrıntılı sistem gereksinimleri Bölüm 12'dedir.

Kendi kendine sınama. Paketli programa `--selftest` argümanı verilirse (`HABITUS.exe --selftest`, kaynaktan `python main.py --selftest`) pencere açılmaz; program koordinat veri tabanını, bir koordinat dönüşümünü, vektör okumayı, bir GeoTIFF yazıp okumayı, hangi algoritmaların çalıştırılabildiğini ve Qt'yi sınar ve bir rapor yazdırır (`main.py:195-197`, `selftest.py`). `--require-algorithms GLM,MAXENT` eklenirse adı geçen algoritmalardan biri yoksa sınama başarısız sayılır (`selftest.py:368-376`). Bir kurulumun çalışıp çalışmadığından emin olmanın en kısa yolu budur.

1.4 Açılış

Program açıldığında önce bir açılış ekranı (splash) görünür. Bu sırada bilimsel kütüphaneler tek tek yüklenir ve açılış ekranı yüklenen kütüphanenin adını gösterir: çizim (matplotlib), tablolar (pandas), SciPy, raster okuma (rasterio), vektör katmanları, makine öğrenmesi, XGBoost, LightGBM, CatBoost, MaxEnt (elapid) ve GAM (`main.py:223-235`). Soğuk bir açılışta bu birkaç saniye sürebilir; ekranın boş kalmaması için yükleme bilerek pencereden önce yapılır.

Açılışta program ayrıca şunları yapar:

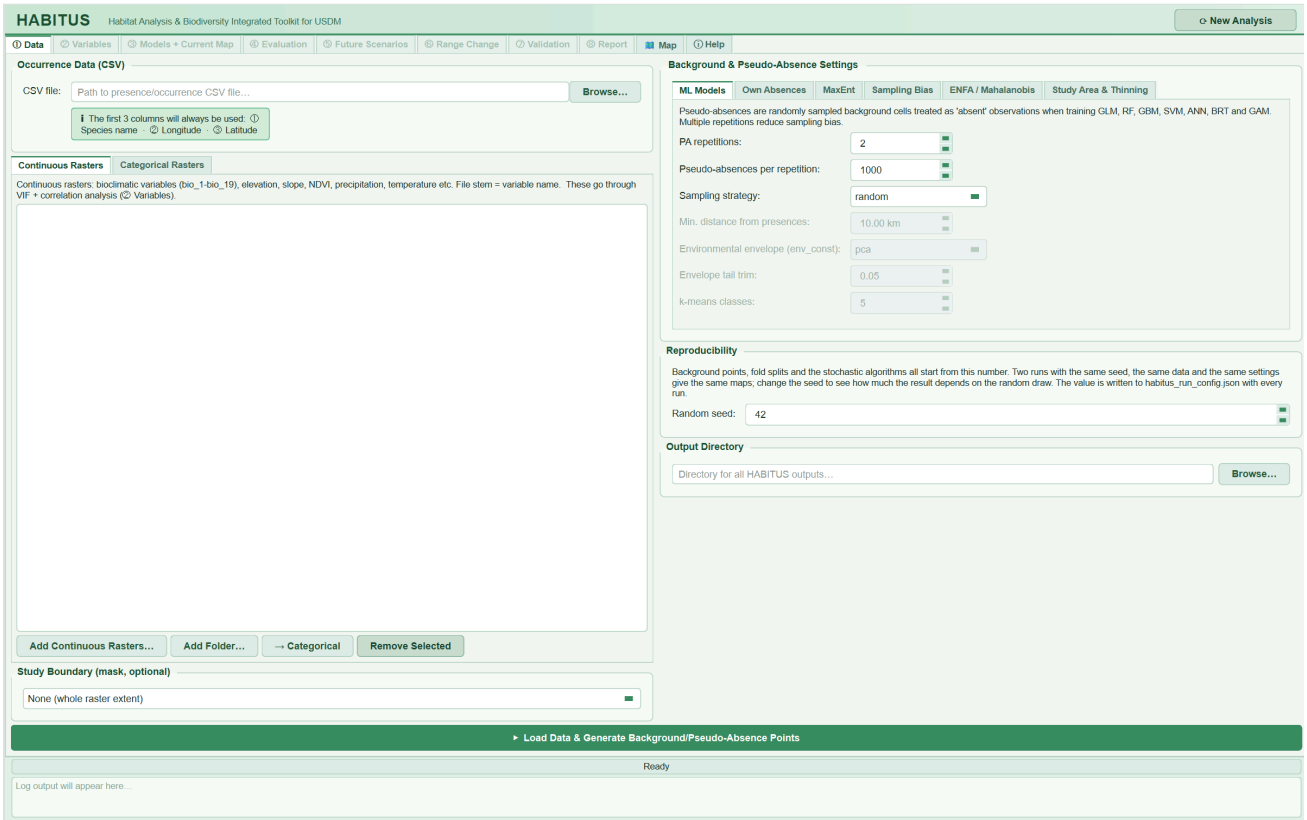
- Koordinat dönüşüm veri tabanını (`proj.db`) denetler. Ortamda başka bir kurulumdan (ör. PostGIS) kalma eski bir veri tabanı tanımlıysa programın kendi kopyasını seçer (`main.py:88-163`).
- Linux ve macOS'ta karakter ve sayı yerel ayarını yansız C yerel ayarına döndürür; böylece Türkçe yerel ayarda raster kütüphanesi büyük-küçük harf karşılaştırmalarında yanılmaz. Arayüz bundan etkilenmez; Windows'ta bu adım uygulanmaz (`main.py:293-315`).
- Çökme günlüğünü açar (Bölüm 12).
- Arka planda yeni sürüm denetimi başlatır; sonuç `ⓘ Help` > `About` sayfasında görünür (`main_dialog.py:130-148`).

Ana pencerenin ilk boyutu ekranın %95 genişliği ve %92 yüksekliğidir, en çok 1600 x 1000 mantıksal piksel (`main_dialog.py:113-121`). Pencere büyütülüp küçültülebilir; her sekme kaydırma alanı içindedir, dar ekranda içerik kaydırılır (`main_dialog.py:183-193`).

1.5 Ana pencerenin düzeni

Ana pencere dört bölgeden oluşur (`main_dialog.py:150-217`):

1. **Başlık şeridi.** Solda program adı ve alt başlık, sağda `⟳ New Analysis` (yeni analiz) düğmesi (§1.5.1).
2. **Sekme şeridi.** Sekiz numaralı iş akışı sekmesi ile `🗺 Map` ve `ⓘ Help` (§1.6).
3. **Sekme içeriği.** Seçili sekmenin ekranı; Bölüm 2-12'de anlatılır.
4. **Alt durum şeridi.** İlerleme çubuğu ve günlük paneli (§1.7).



Şekil 1.1. Program ilk açıldığında ana pencere: üstte başlık şeridi ve sağda New Analysis düğmesi, altında on sekmelik şerit (yalnız ① Data, 🗺 Map ve ⓘ Help etkin), ortada ① Data sekmesi, en altta Ready yazan ilerleme çubuğu ve boş günlük paneli.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
⟳ New Analysis	Onay sorar; bütün veri, model ve sonuçları bellekten siler, sekmeleri kilitler, ① Data'ya döner	—	düğme; onay penceresinde Yes / No (varsayılan No)	main_dialog.py:159-163, 241-249, 303-419
Sekme şeridi	İş akışı adımları; tamamlanmamış adımın sonraki sekmeleri soluk ve tıklanamaz	① Data, 🗺 Map, ⓘ Help açık	on sekme	main_dialog.py:186-199
İlerleme çubuğu	Çalışan işin adımını ve yüzdesini gösterir; iş bitince yeşil, hatada kırmızı	Ready, %0	%0-100	main_dialog.py:205-208, 659-703
Günlük paneli	Her iletiyi saatiyle yazar; aynı satırlar oturum günlüğü dosyasına da gider	boş (Log output will appear here…)	salt okunur	main_dialog.py:209-216, 790-801
Random seed (① Data > Reproducibility)	Bütün rastgele işlemlerin başlangıç sayısı	42	0-2147483647	tabs/tab_data.py:242-249
Output Directory (① Data)	Bütün çıktıların yazılacağı klasör	boş; CSV seçilince otomatik dolar	herhangi bir klasör	tabs/tab_data.py:253-259, 1125-1162
Output Directory (for saving map files) (③ Models)	Modellerin ve haritaların yazılacağı klasör; boşsa ① Data'daki klasörü kullanır	① Data'dan otomatik	herhangi bir klasör	tabs/tab_models.py:679-686, 1338-1348, 1470-1477

1.5.1 ⟳ New Analysis

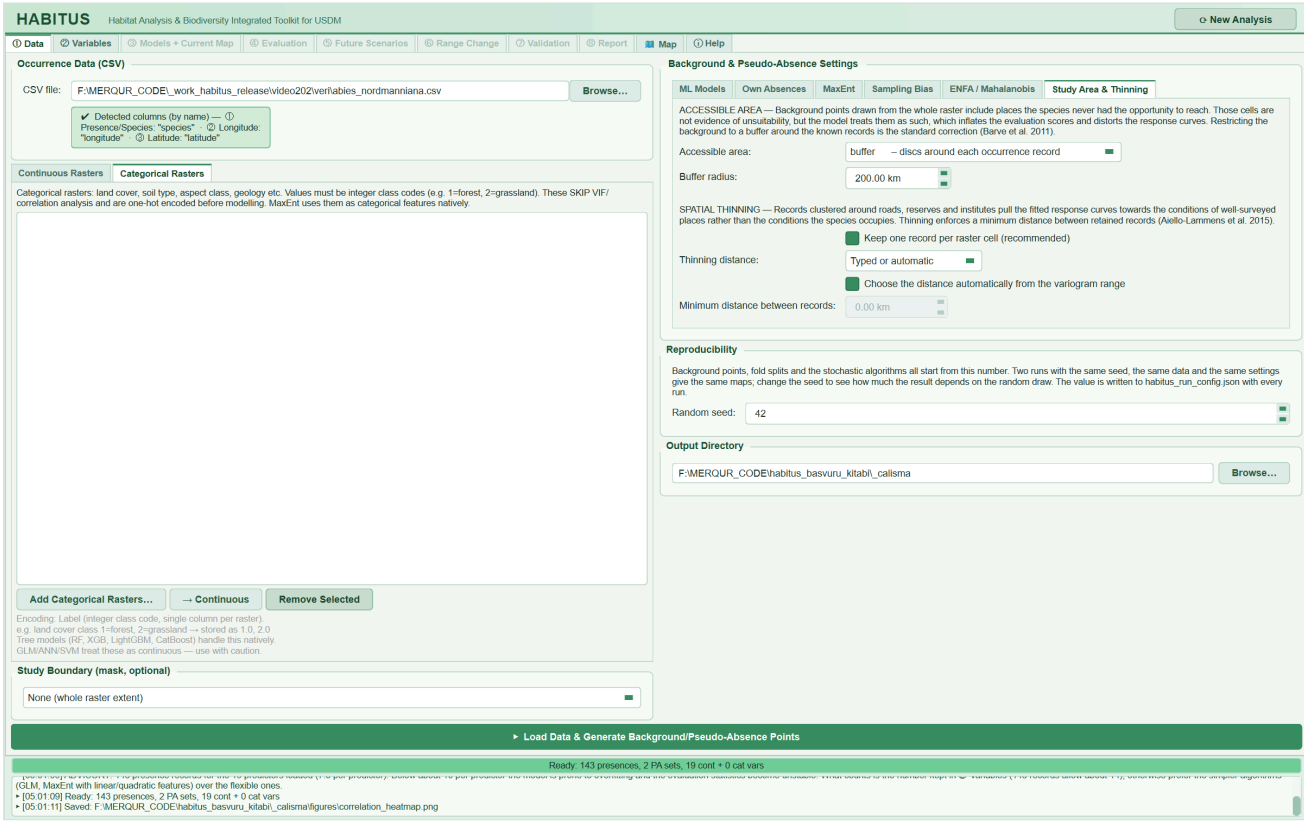
- **Ne işe yarar.** "This will clear all loaded data, models, and results. Proceed?" sorusuna Yes denirse program çalışan arka plan işlerini durdurur (en çok 1,5 saniye bekler), paylaşılan durumu sıfırlar, üç harita görüntüleyicisini temizler, ② Variables'tan ⑧ Report'a kadar bütün sekmeleri kilitler ve ① Data'ya döner. ① Data'da sunum CSV yolu, raster listeleri ve Output Directory alanı boşaltılır; oturum günlüğü kapatılır (main_dialog.py:303-419).
- **Ne zaman, nasıl ayarlanır.** Başka bir türe ya da tamamen farklı bir veri setine geçerken kullanılır. Aynı türde yalnız bir ayarı değiştirip yeniden çalıştırmak için gerekmez; ilgili adımı yeniden çalıştırmak yeter (§1.6.2).
- **Sonuca etkisi.** Bellekteki her şey silinir. Sekmelerdeki ayarlar (tohum, sözde yokluk sayısı, algoritma seçimleri vb.) ise **silinmez**; yeni analiz önceki ayarlarla başlar. Output Directory boşaltıldığı için yeni CSV seçildiğinde yeni bir analiz klasörü açılır (§1.9).
- **Dikkat.** Diskteki dosyalar silinmez. Önceki analizin klasörü olduğu gibi kalır.

1.6 Sekme şeridi ve kilit düzeni

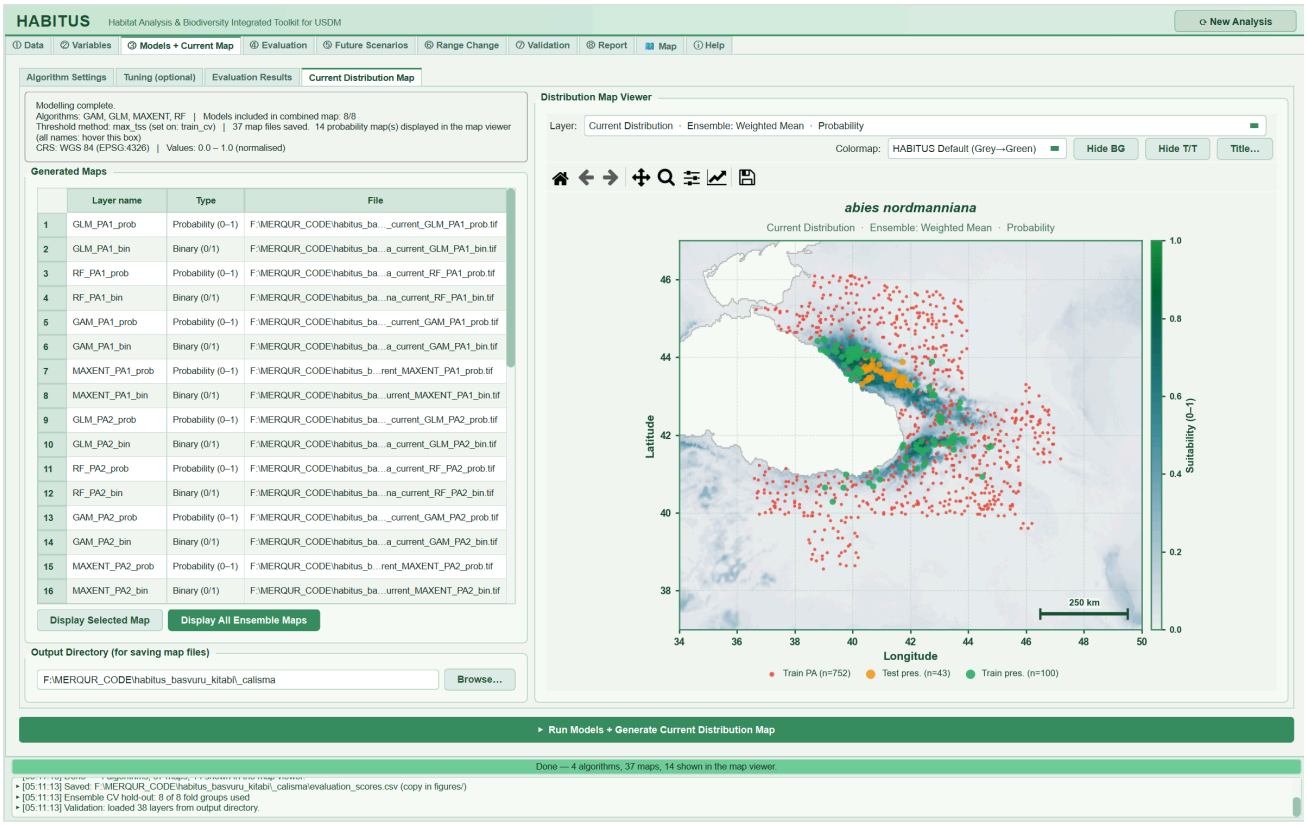
Sekmeler iş akışının sırasını izler. Program açıldığında yalnız `① Data`, `🗺 Map` ve `ⓘ Help` etkindir; `② Variables`'tan `⑧ Report`'a kadar yedi sekme kilitlidir (`main_dialog.py:198-199`). Bir adım başarıyla bittiğinde sonraki sekmenin kilidi açılır ve çoğu durumda program o sekmeye geçer.

Tamamlanan adım	Açılan sekmeler	Program nereye geçer	Kod
`① Data`: `Load Data & Generate Background/Pseudo-Absence Points`	`② Variables`	`② Variables`	`tabs/tab_data.py:1759-1760`
`② Variables`: değişken seçiminin onaylanması	`③ Models + Current Map`	`③ Models`, `Algorithm Settings` alt sekmesi	`tabs/tab_vif.py:1206-1208`
`③ Models`: `Run Models + Generate Current Distribution Map`	`④ Evaluation`, `⑤ Future Scenarios`, `⑥ Range Change`, `⑦ Validation`, `⑧ Report` (birlikte)	`③ Models` içinde `Current Distribution Map` alt sekmesi	`tabs/tab_models.py:1848-1852`
`⑤ Future Scenarios`: projeksiyon	(zaten açık olan) `⑥ Range Change`, `④ Evaluation`	sekme değişmez	`tabs/tab_projection.py:672-673`

`③ Models` bittikten sonra `④`-`⑧` sırası zorunlu değildir: örneğin `⑤ Future Scenarios` çalıştırılmadan `⑦ Validation` ya da `⑧ Report` kullanılabilir. `⑥ Range Change` iki ikili harita ister; gelecek haritası yoksa yalnız güncel haritalar arasında karşılaştırma yapılabilir (Bölüm 9). Çok türlü bir dosyanın öteki türleri `⑧ Report`'taki toplu çalıştırmayla işlendiğinde bütün sekmeler açılır (`batch_species.py:215-217`).



Şekil 1.2. ① Data adımı bittikten sonra ana pencere: ② Variables sekmesinin kilidi açılmış, öteki numaralı sekmeler hâlâ soluk; ilerleme çubuğu yeşil ve "Ready: 143 presences, 2 PA sets, 19 cont + 0 cat vars" yazıyor; günlük panelinde saatli iletiler görülüyor (örnek koşu).



Şekil 1.3. ③ Models adımı bittikten sonra ana pencere: ④ Evaluation'dan ⑧ Report'a kadar bütün sekmeler açılmış; ilerleme çubuğu "Done — 4 algorithms, 37 maps, 14 shown in the map viewer" yazıyor; Output Directory (for saving map files) alanı ① Data'daki klasörle otomatik doldurulmuş (örnek koşu).

1.6.1 Sekme şeridi özeti

İş akışının tamamı şu tablodadır. "Ana çıktı" sütunu çıktı klasörüne yazılan başlıca dosyaları verir; tam katalog Bölüm 12'dedir.

Sekme	Ne yapar	Ana çıktı	Bölüm
`① Data`	Kayıtları okur ve süzer (hücre başına bir kayıt, seyreltme), rasterları yükler ve ızgarayı denetler, erişilebilir alanı belirler, sözde yokluk ve MaxEnt arka plan noktalarını üretir	`accessible_area.tif`, isteğe bağlı `background_maxent.csv`; bellekte eğitim tabloları	2-3
`② Variables`	VIF ve korelasyonla değişkenleri sıralar; kullanıcı sürekli ve kategorik değişkenleri seçip onaylar; isteğe bağlı koşul sayısı, PCA, LASSO, Ridge	`figures/` içinde korelasyon ısı haritaları, VIF grafiği, `high_correlation_pairs.csv`, `variable_priority_ranking.csv`	4
`③ Models + Current Map`	Seçili algoritmaları çapraz doğrulamayla eğitir, eşik ve topluluk (ensemble) kurar, güncel iklim haritasını yazar; isteğe bağlı Tuning	`current/` klasöründe olasılık, ikili ve belirsizlik rasterları; `evaluation_scores.csv`; `occurrence_train.csv`, `occurrence_test.csv`; `habitus_run_config.json`	5-6
`④ Evaluation`	Skor, ROC, omisyon ve kalibrasyon, değişken önemi, tepki eğrileri, Boyce, eşik tabloları	`figures/evaluation_*.png`, `figures/classification_thresholds.csv`	7

Sekme	Ne yapar	Ana çıktı	Bölüm
⑤ Future Scenarios	Modelleri gelecek iklim rasterlarına uygular; ekstrapolasyon (ExDet, isteğe bağlı MESS) haritaları	<senaryo>/ klasöründe olasılık, ikili, NT1, NT2, MIC rasterları; projection_options.json	8
⑥ Range Change	İki ikili haritayı karşılaştırır: kayıp, kazanç, kararlı	<tür>_<analiz adı>_SRC.tif ve alan tablosu	9
⑦ Validation	Modeli bağımsız bir referans harita ya da noktalarla karşılaştırır	validation/ klasöründe yeniden sınıflanmış rasterlar; isteğe bağlı rapor ve nokta dosyası	10
⑧ Report	Bütün adımları bilimsel bir HTML raporda toplar; ODMAP protokolünü dışa aktarır; çok türlü dosyada öteki türleri çalıştırır	<tür>_HABITUS_report_<zaman>.html, odmap_protocol.md/.csv, batch_summary.csv	11
🗺 Map	Girdileri (kayıtlar, dışlananlar, kendi yokluklar, sınır) çevrimiçi altlık harita üzerinde gösterir; yayın için çalışma alanı şekli	<tür>_study_area.png (isteğe bağlı)	12
ⓘ Help	İngilizce kullanıcı rehberi ve güncelleme denetimi	—	12

1.6.2 Bir adımı yeniden çalıştırmak sonraki sonuçları siler

HABITUS, bir adımın girdisi değiştiğinde ondan sonra gelen sonuçları eski girdiyle bırakmaz. İki silme yolu vardır:

- ① Data **yeniden çalıştırılırsa** (daha önce veri yüklenmişse) program onay sormadan çalışan model, projeksiyon ve değişim işlerini durdurur, bellekteki değişken seçimini, modelleri, topluluğu, projeksiyonları, değişim sonuçlarını, Tuning sonuçlarını, değerlendirme grafiklerini ve doğrulama sonuçlarını siler, ② Variables'tan ⑧ Report'a kadar bütün sekmeleri kilitler. Günlüğe "Data inputs changed — all previous results cleared. Re-running data formatting…" yazılır (tabs/tab_data.py:1396-1397, main_dialog.py:421-526).
- ② Variables **'ta seçim yeniden onaylanırsa** ve modeller daha önce çalıştırılmışsa "Re-confirm Variable Selection" penceresi sorar: "Confirming now will discard all existing models, maps, projections, and evaluation results so everything is recomputed from scratch." Yes denirse modeller ve sonraki bütün sonuçlar silinir; ① Data, ② Variables ve ③ Models açık kalır, ④-⑧ kilitlenir. Günlüğe "Variable selection changed — downstream results cleared. Re-run Models to continue." yazılır (tabs/tab_vif.py:1170-1185, main_dialog.py:528-594).

③ Models yeniden çalıştırıldığında sekmeler kilitlenmez; yeni sonuçlar eskilerinin yerine geçer.

> **Dikkat:** Silme yalnız programın belleği içindir. Çıktı klasöründeki dosyalar silinmez; yeni koşu aynı adlı dosyaların üzerine yazar. İkinci koşuda seçilmeyen bir algoritmanın birinci koşudan kalan rasterları ve figures/ içindeki eski PNG'ler klasörde kalır. ⑧ Report figures/ içindeki bütün PNG'leri, ⑦ Validation da çıktı klasöründeki katmanları okuduğu için (tabs/tab_report.py:578-587), ayarı değişen her koşu için yeni bir çıktı klasörü kullanmak en güvenli yoldur.

1.7 Alt durum şeridi: ilerleme çubuğu ve günlük

İlerleme çubuğu

Uzun bir iş çalışırken çubuk o anki adımı ve yüzdeyi yazar (ör. "Reading raster 3 of 19: bio_11.tif 12%"). Çubuk iş boyunca yalnız ileri gider; %100'e yalnız iş gerçekten bittiğinde ulaşır. Bitişte çubuk yeşile döner ve sonucu özetler (ör. "Done — 4 algorithms, 37 maps, 14 shown in the map viewer"); hata olursa kırmızıya döner ve "Failed — see the message" yazar (`main_dialog.py:659-703`, `755-788`).

Bir iş çalışırken onu başlatan düğme devre dışı kalır ve metni değişir; ikinci bir tıklama ikinci bir iş başlatmaz. Düğme, iş başarıyla, hatayla ya da iptalle bittiğinde yeniden etkinleşir (`main_dialog.py:707-788`). `⑦ Validation` ve `⑧ Report` işleri arayüz iş parçacığında çalışır; bu sürede küçük bir "HABITUS" penceresi o anki adımı gösterir ve imleç bekleme imlecine döner (`main_dialog.py:712-753`).

Günlük paneli

Programın her iletisi günlük panelinde `▸ [SS:DD:ss] ileti` biçiminde görünür ve panel kendiliğinden son satıra kayar (`main_dialog.py:790-793`). Günlükte şunlar bulunur: her adımın ilerleme iletileri, uyarılar (`⚠`, `WARNING:`), yorumu etkileyen bilgi notları (`ADVISORY:`), kaydedilen her dosyanın yolu (`Saved: …`) ve hataların son satırı (`ERROR: …`).

Aynı satırlar, çıktı klasörü belli olduğu andan itibaren **oturum günlüğü** dosyasına da yazılır (`main_dialog.py:612-645`, `794-801`):

- Dosya adı `<tür>_<YYYYMMDD_HHMMSS>.log`'dur. Tür adı o anda bilinmiyorsa (ör. çıktı klasörü elle seçildiyse) ad `habitus_<YYYYMMDD_HHMMSS>.log` olur. Örnek koşunun günlük dosyası `habitus_20261121_050108.log` adını taşır.
- Dosyanın başında "HABITUS Analysis Log", tür adı ve başlangıç zamanı bulunur; `New Analysis` ya da pencerenin kapanması dosyayı "Closed" satırıyla kapatır.
- Çıktı klasörü değişirse (ör. çok türlü dosyada sonraki tür) eski günlük kapanır ve yenisi o klasörde açılır.
- Bir hata olduğunda hata iletisinin tamamı `--- ERROR ---` başlığıyla günlüğe eklenir; hata penceresi de günlük dosyasının yolunu gösterir (`main_dialog.py:806-827`).

Hata penceresi ("HABITUS Error") hatanın son satırını başlık olarak, tanınan hatalar için de sade bir öneri gösterir. `Show details` tam metni açar, `Copy full message` tamamını panoya kopyalar (`error_dialog.py:30-56`, `80-182`). Bir sorunu bildirirken bu metin ve oturum günlüğü birlikte gönderilmelidir.

1.8 İlerleme, iptal ve pencerenin kapanması

Veri hazırlama, modelleme, projeksiyon, alan değişimi, Tuning, raster klasörü taraması ve güncelleme denetimi arka plan iş parçacığında çalışır; pencere bu sırada donmaz, başka sekmeler gezilebilir (`threads.py:1-23`).

İptal işbirliğine dayanır: durdurma istenince işin bir bayrağı kurulur ve iş bir sonraki ilerleme bildiriminde durur; iptal edilen iş sonuç ya da hata bildirmez (`main_dialog.py:37-74`, `threads.py:34-63`). `③ Models`, `① Data` ve `⑤ Future Scenarios` işleri için ayrı bir "durdur" düğmesi yoktur; çalışan bir iş şu yollarla durdurulur:

- `⟳ New Analysis` (en çok 1,5 saniye bekler),
- `① Data`'nın yeniden çalıştırılması ya da değişken seçiminin yeniden onaylanması (aşağı akıştaki işleri durdurur),
- pencerenin kapatılması.

`Tuning (optional)` alt sekmesinin kendi `Cancel` düğmesi vardır (Bölüm 6); `④ Evaluation`'daki toplu grafik kaydı da ilerleme penceresinden iptal edilebilir (Bölüm 7).

Bir iş çalışırken pencere kapatılmak istenirse program sorar: "Data formatting, Models still running — close anyway? The unfinished work is discarded." (çalışan işlerin adlarıyla; varsayılan `No`). `Yes` denirse işler iptal edilir ve en çok birkaç saniye beklenir (`main_dialog.py:284-301`, `596-608`).

> **Not:** İptal bir ilerleme bildiriminde gerçekleşir. Tek parça ve uzun bir hesap (ör. büyük bir veri setinde tek bir modelin eğitimi) bitene kadar arka planda sürer; sonucu pencereye ulaşmaz ama işlemciyi o süre boyunca kullanır (`main_dialog.py:266-282`).

1.9 Çıktı klasörü ve otomatik analiz klasörü

Bütün çıktılar tek bir klasöre yazılır: `① Data` > `Output Directory`.

- **Otomatik klasör.** `Output Directory` boşken bir sunum CSV'si seçilirse program kullanıcının belgeler klasöründe `Documents/HABITUS/<tür>_<YYYYMMDD_HHMMSS>` adlı bir klasör açar ve alana yazar; günlüğe "Analysis folder auto-created: <yol>" düşer. Tür adı CSV'nin tür sütunundaki en sık değerdir (GBIF dosyasında `species` sütunu adından tanınır); çok türlü dosyada klasör CSV dosyasının adını alır. Belgeler klasörü yoksa ev klasörü kullanılır; adda harf, rakam, nokta ve tire dışındaki karakterler alt çizgiye çevrilir (`tabs/tab_data.py:1125-1162`, `main_dialog.py:831-853`).
- **Elle seçilen klasör.** `Browse…` ile seçilen ya da elle yazılan klasör olduğu gibi kullanılır; otomatik klasör açılmaz.
- **Çok türlü dosya.** Her tür, çıktı klasörünün altında kendi alt klasörüne yazılır (`tabs/tab_data.py:1115-1123`; Bölüm 2).
- **`③ Models` klasörü.** `③ Models`'taki `Output Directory (for saving map files)` alanı ① Data'daki klasörle kendiliğinden dolar. Kullanıcı buraya başka bir klasör yazarsa modelleme çıktıları ve bundan sonraki bütün adımlar (ve oturum günlüğü) o klasöre geçer (`tabs/tab_models.py:1338-1348`, `1470-1477`).

1.9.1 `Output Directory`

- **Ne işe yarar.** Raster haritaların, tabloların, şekillerin, koşu kaydının, oturum günlüğünün ve raporun yazılacağı yeri belirler. `Load Data & Generate Background/Pseudo-Absence Points` boşsa "Please specify an output directory." iletisiyle çalışmaz (`tabs/tab_data.py:1370-1371`).
- **Ne zaman, nasıl ayarlanır.** Her analiz için ayrı bir klasör önerilir; otomatik klasör bunu zaman damgasıyla kendiliğinden sağlar. Aynı türde farklı ayarlarla karşılaştırmalı koşular yapılacaksa her koşuya yeni klasör verilmelidir.
- **Sonuca etkisi.** Hesaplanan sayıları değiştirmez. Ama aynı klasöre ikinci kez yazmak `habitus_run_config.json`, `evaluation_scores.csv` ve rasterların üzerine yazar ve önceki koşudan kalan dosyalar raporda görünebilir (§1.6.2).
- **Dikkat.** `evaluation_scores.csv` elle düzenlenmişse üzerine yazılmadan önce `evaluation_scores.<zaman>.bak.csv` adıyla yedeklenir (`sdm_core.py:6764-6776`); öteki dosyalar yedeklenmez. Çıktı klasöründeki bir dosya başka bir programda (ör. bir CBS programı ya da hesap tablosu) açıksa yazma "Permission denied" ile başarısız olabilir.

1.10 Rastgele tohum ve tekrarlanabilirlik

`① Data` > `Reproducibility` grubundaki `Random seed` (rastgele tohum) bütün analizin tek başlangıç sayısıdır. Arayüzdeki açıklama: "Background points, fold splits and the stochastic algorithms all start from this number. Two runs with the same seed, the same data and the same settings give the same maps; change the seed to see how much the result depends on the random draw." (`tabs/tab_data.py:234-249`).

1.10.1 `Random seed`

- **Ne işe yarar.** `① Data` çalıştırıldığında, rastgele hiçbir şey yapılmadan önce tohum uygulanır (`tabs/tab_data.py:1505-1510`, `sdm_core.py:115-119`). NumPy'ın genel rastgele sayı üreticisi ve Python'un `random` modülü bu sayıyla başlatılır; `PYTHONHASHSEED` ortam değişkeni tanımlı değilse bu sayıya ayarlanır (`run_config.py:31-50`). Arka plan ve sözde yokluk örneklemesi, çapraz doğrulama katlarının atanması ve bütün stokastik algoritmalar (RF, GBM, BRT, MaxEnt vb.) aynı değerden türetilir. Tohum, modelleme bittiğinde `habitus_run_config.json` dosyasının `random_seed` alanına yazılır (`tabs/tab_models.py:2071`, `run_config.py:141`).
- **Ne zaman, nasıl ayarlanır.** Yayın için varsayılan 42 ya da herhangi bir sabit değer yeterlidir; önemli olan değerin raporlanmasıdır. Sonucun rastgele çekilişe ne kadar bağlı olduğunu görmek için aynı analiz birkaç farklı tohumla (ör. 1, 2, 3) tekrarlanıp skorlar ve haritalar karşılaştırılabilir.
- **Sonuca etkisi.** Aynı veri, aynı ayarlar ve aynı tohumla iki koşu aynı sözde yokluk noktalarını, aynı katları ve aynı haritaları verir. Tohum değişirse noktalar ve katlar değişir; skorlar ve haritalar bu değişkenlik kadar oynar. Az kayıtlı türlerde bu oynama belirgin olabilir.
- **Dikkat.** Tohum `① Data` çalıştırılırken okunur. Tohumu değiştirdikten sonra yalnız `③ Models`'i yeniden çalıştırmak yeni tohumu uygulamaz: `① Data` yeniden çalıştırılmalıdır (bu da sonraki sonuçları siler, §1.6.2). Eşitlik bit düzeyinde garanti edilmez; çok iş parçacıklı hesaplarda (ör. RF) toplama sırası son basamakta küçük farklar yaratabilir.

1.11 Örnek veri

Programı denemek için hazır bir örnek veri paketi (`habitus_sample.zip`) programın sürüm sayfasından indirilir. Paket bir türün sunum kayıtlarını, 19 biyoiklim değişkenini, yükselti ve NDVI gibi ek katmanları, kategorik katmanları (bakı, eğim), birkaç iklim senaryosunu ve dönemini ve doğrulama için bir referans dağılım haritasını içerir (`README.md`, "Sample Data"). Kaynak ağacındaki `ornek_veri` klasöründe ayrıca *Pinus brutia* kayıtları ile `Own Absences` alt sekmesini denemek için kurallarla üretilmiş iki **sentetik** yokluk dosyası ve bir kalibrasyon alanı poligonu bulunur; sentetik dosyalar saha verisi değildir ve yayına giden bir analizde kullanılmamalıdır (`ornek_veri/OKUBENI.md`).

Bu kitabın ekran görüntüleri ise *Abies nordmanniana* verisiyle alınmıştır (§1.2).

İlk deneme için en kısa yol. (1) `① Data`'da CSV'yi `Browse…` ile seçin; çıktı klasörü kendiliğinden açılır. (2) `Add Folder…` ile biyoiklim klasörünü ekleyin. (3) Öteki ayarları varsayılanda bırakıp `Load Data & Generate Background/Pseudo-Absence Points`'e basın. (4) `② Variables`'ta önerilen değişkenleri onaylayın. (5) `③ Models`'te birkaç algoritma seçip `Run Models + Generate Current Distribution Map`'e basın. (6) `⑧ Report`'ta raporu üretin.

1.12 Uzman soruları

S1. Aynı veri ve aynı ayarlarla iki koşu gerçekten aynı haritayı verir mi? Cevap: Tasarım olarak evet. Tek tohum ① Data başında NumPy'ın genel üreticisine, Python random modülüne ve (tanımlı değilse) PYTHONHASHSEED'e uygulanır (run_config.py:31-50, tabs/tab_data.py:1510); algoritmalar aynı değeri kendi random_state parametresinden alır (run_config.py:38-40). Bit düzeyinde eşitlik garanti edilmez: çok iş parçacıklı hesaplarda toplama sırası son basamakta fark yaratabilir. Bu kitap için iki koşunun bit bit karşılaştırması yapılmamıştır. Pratikte, aynı sürüm ve aynı kütüphane sürümleriyle elde edilen sonuç aynı olmalıdır; kütüphane sürümleri habitus_run_config.json > environment alanına yazılır (run_config.py:61-74).

S2. Tohumu değiştirip yalnız modelleri yeniden çalıştırırsam ne olur? Cevap: Yeni tohum uygulanmaz. Tohum yalnız ① Data çalıştırılırken okunur (tabs/tab_data.py:1505-1510); ③ Models formatter nesnesinde saklanan değeri kullanır ve koşu kaydına da onu yazar (tabs/tab_models.py:1535, 2071). Tohum duyarlılığı denemesi için her tohumda ① Data yeniden çalıştırılmalıdır.

S3. Program R paketlerinin yerine geçer mi; biomod2 ile aynı sonucu verir mi? Cevap: HABITUS R çağırmaz; algoritmalar Python kütüphaneleriyle yazılmıştır (main.py:223-235). Yöntem tanımları biomod2, dismo, ENMeval, maxnet ve blockCV'deki karşılıklarına yakın tutulmuştur, ama varsayılanlar ve uygulamalar birebir aynı değildir; algoritma düzeyindeki farklar Bölüm 5-7'de anlatılır. "Aynı sonuç" beklemek yerine yöntemlerin aynı tanımlanıp tanımlanmadığı denetlenmelidir. Algoritma seçiminin sonucu büyük ölçüde değiştirebildiği bilinmektedir (Qiao vd., 2015); bu yüzden birden çok algoritma ve topluluk önerilir.

S4. Bir adımı yeniden çalıştırmak hangi sonuçları geçersiz kılar? Cevap: ① Data'nın yeniden çalıştırılması değişken seçimi dahil bütün aşağı akışı siler ve ② - ⑧'i kilitler (main_dialog.py:421-526); değişken seçiminin yeniden onaylanması modelleri ve sonrasını siler, ④-⑧'i kilitler (main_dialog.py:528-594). Bu, eski girdiyle üretilmiş bir sonucun yeni girdiyle karıştırılmasını önler. Silme bellektedir; diskteki dosyalar kalır (§1.6.2).

S5. Neden ③ Models bitince beş sekme birden açılıyor; değerlendirme yapmadan projeksiyon yapılabilir mi? Cevap: Evet. ③ Models bitişinde ④-⑧ birlikte açılır (tabs/tab_models.py:1848-1852). Program değerlendirmeyi zorunlu bir ara adım saymaz, çünkü değerlendirme skorları modelleme sırasında çapraz doğrulamayla zaten hesaplanır ve evaluation_scores.csv'ye yazılır (tabs/tab_models.py:1841-1846). ④ Evaluation bu skorların grafiklerini ve ek tanıları üretir. Bilimsel olarak projeksiyondan önce modelin değerlendirilmesi elbette gereklidir; program bunu zorlamaz, kullanıcıya bırakır.

S6. Bir hesap uzun sürerse nasıl durdurulur; durdurulan iş yarım dosya bırakır mı? Cevap: Model, veri ve projeksiyon işleri için ayrı bir durdurma düğmesi yoktur; New Analysis, aşağı akışı silen yeniden çalıştırmalar ya da pencereyi kapatmak işi iptal eder (main_dialog.py:266-282). İptal bir sonraki ilerleme bildiriminde gerçekleşir ve iş sonuç bildirmez (main_dialog.py:54-74). O ana kadar diske yazılmış dosyalar silinmez; yarıda kesilmiş bir projeksiyonun klasöründe bazı rasterlar bulunabilir. Bu yüzden iptal edilen bir koşunun klasörü yeni koşuda kullanılmamalıdır.

S7. Çalışmanın hangi kayıtları bir hakeme sunulabilir? Cevap: Üç dosya birlikte: habitus_run_config.json (ayarlar, varsayılan bırakılanlar, tohum, kütüphane sürümleri; run_config.py:137-152), oturum günlüğü (<tür>_<zaman>.log; adım adım iletiler, uyarılar ve hatalar; main_dialog.py:612-645) ve ⑧ Report'un ürettiği ODMAP protokolü (Zurell vd., 2020; Bölüm 11). Ayrıntılar Bölüm 12'dedir.

S8. Günlük panelindeki ADVISORY: satırları hata mıdır? Cevap: Hayır. Bunlar sonucun yorumunu etkileyen bilgi notlarıdır; örneğin örnekleme yanlılığının düzeltilmediği, hücre boyutunun yerel habitat için kaba olduğu ya da kayıt başına düşen öngörücü sayısının az olduğu. ① Data bitişinde günlüğe yazılırlar, varsayılan

örnekleme yanlılığı notundan başka bir not varsa ayrıca bir uyarı penceresi açılır (`tabs/tab_data.py:1743-1755`). Notlar `habitus_run_config.json` > `notes` alanına da yazılır; örnek koşuda dört not vardır.

S9. Program büyük rasterlarla bellek yetersizliğine düşerse ne olur? Cevap: Hata penceresi "The analysis ran out of memory. Reduce the raster extent or resolution, or lower the number of background points." önerisini gösterir (`error_dialog.py:39-41`). Büyük projeksiyonlar şeritler hâlinde yapılır (`sdm_core.py:7512-7526`; Bölüm 8 ve 12). Belleği en çok azaltan önlem rasterları çalışma bölgesine kırpmaktır (`① Data` > `Study Boundary`, Bölüm 2).

1.13 Sık hatalar ve tuzaklar

- **Tohumu değiştirip yalnız modelleri yeniden çalıştırmak.** Tohum `① Data`'da uygulanır; yeni tohum için `① Data` yeniden çalıştırılmalıdır.
- **Farklı ayarlarla koşuları aynı klasöre yazmak.** Dosyalar üzerine yazılır, eski koşudan kalan PNG ve rasterlar rapora ve doğrulama listelerine girebilir. Her koşu için yeni klasör kullanın; `New Analysis` sonrası CSV seçmek yeni bir otomatik klasör açar.
- **`New Analysis`'in ayarları sıfırladığını sanmak.** Yalnız veri, sonuçlar, dosya yolları ve klasör alanı temizlenir; tohum, sözde yokluk ve algoritma ayarları kalır.
- **`③ Models`'teki klasör alanına başka bir yol yazmak.** Bundan sonraki bütün çıktılar ve oturum günlüğü o klasöre gider; `① Data` çıktıları (`accessible_area.tif`, korelasyon şekilleri) ilk klasörde kalır.
- **Pencerenin donduğunu sanmak.** Uzun işler arka planda çalışır; ilerleme çubuğu ve günlük güncelleniyorsa iş sürmektedir.
- **Hata penceresini kapatıp metni kaybetmek.** `Copy full message` tam metni kopyalar; metin ayrıca oturum günlüğündedir.
- **Kaynaktan çalıştırırken sistem Python'unu kullanmak.** Gerekli kütüphaneler yoksa program "No module named 'rasterio'" türünden bir hatayla durur; bağımlılıkların kurulu olduğu ortam kullanılmalıdır.

1.14 Kaynaklar

Elith, J., Leathwick, J.R., 2009. Species distribution models: ecological explanation and prediction across space and time. Annual Review of Ecology, Evolution, and Systematics 40, 677-697. https://doi.org/10.1146/annurev.ecolsys.110308.120159

Örücü, Ö. K., & Örücü, S. (2026). HABITUS: Habitat Analysis and Biodiversity Integrated Toolkit for Unified Species Distribution Modelling. Ecological Perspective, [Technical Report]. https://doi.org/10.53463/ecopers.20260435

Qiao, H., Soberón, J., Peterson, A.T., 2015. No silver bullets in correlative ecological niche modelling: insights from testing among many potential algorithms for niche estimation. Methods in Ecology and Evolution 6, 1126-1136. https://doi.org/10.1111/2041-210X.12397

Zurell, D., Franklin, J., König, C., Bouchet, P.J., Dormann, C.F., Elith, J., vd., 2020. A standard protocol for reporting species distribution models. Ecography 43, 1261-1277. https://doi.org/10.1111/ecog.04960

2. Veri I: kayıtlar, katmanlar ve çalışma sınırı (① Data)

Sekme: ① Data · Kod: `tabs/tab_data.py` (arayüz), `sdm_inputs.py` (CSV okuma), `sdm_mask.py` (klasör tarama, sınırla kırpma), `sdm_core.py` `DataFormatter` (yükleme ve kayıt ayıklama)

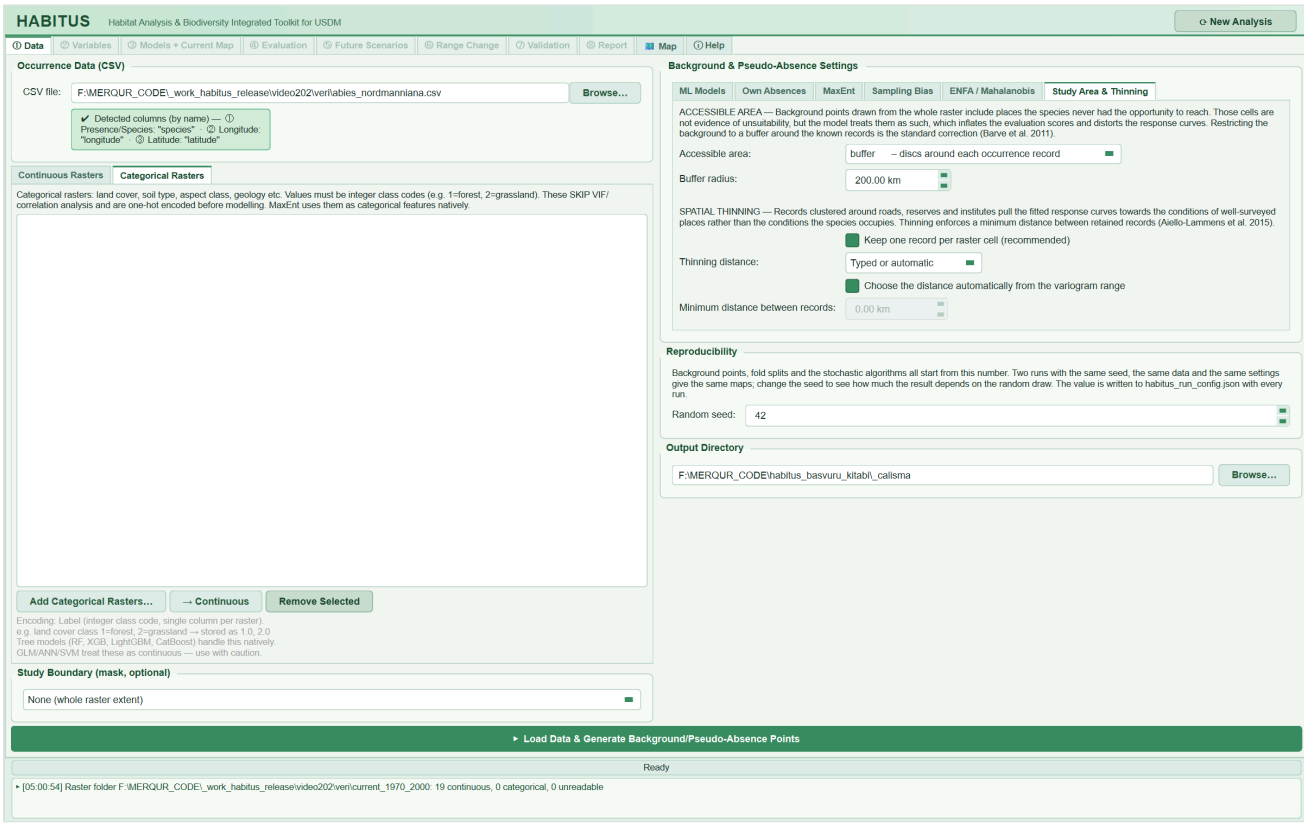
2.1 Bu sekme ne yapar

① Data, iş akışının ilk ve zorunlu adımıdır. Tür kayıtlarını (varlık noktalarını), öngörücü raster katmanlarını ve isteğe bağlı bir çalışma sınırını okur; kayıtları ayıklar; ardından makine öğrenmesi modelleri için yalancı yokluk (pseudo-absence) kümelerini, MaxEnt için arka plan noktalarını ve ENFA için ayrı bir arka plan örneğini üretir. Bu bölüm sekmenin "veri" yarısını anlatır: kayıt dosyası, katmanlar, sınır, tohum, çıktı klasörü ve `Load Data` düğmesinin kayıtlar üzerinde yaptığı her işlem. Arka plan, yalancı yokluk, erişilebilir alan ve seyreltme ayarları Bölüm 3'tedir.

- **Girdi:** bir kayıt CSV'si (tür adı, boylam, enlem), en az bir sürekli raster, isteğe bağlı kategorik rasterler ve bir sınır katmanı.
- **Çıktı:** bellekteki `DataFormatter` nesnesi (öngörücü dizileri, ayıklanmış varlıklar, yalancı yokluk kümeleri, arka planlar), günlüğe ve çalıştırma kaydına (`habitus_run_config.json`) yazılan sayılar; sınır verildiyse kırpılmış katmanlar (`masked_inputs/current/`), erişilebilir alan tanımlıysa `accessible_area.tif`.
- **Kilit düzeni:** adım başarıyla bitince ② Variables sekmesi açılır ve program o sekmeye geçer (`tabs/tab_data.py:1759-1760`). ① yeniden çalıştırılırsa sonraki sekmelerin bütün sonuçları sıfırlanır (`tabs/tab_data.py:1396-1397`).

2.2 Ekranın düzeni

Sekme iki sütundan oluşur (`tabs/tab_data.py:78-82`). Sol sütunda kayıt dosyası, raster listeleri ve çalışma sınırı; sağ sütunda arka plan/yalancı yokluk ayarları, tekrarlanabilirlik ve çıktı klasörü bulunur. En altta bütün adımı başlatan yeşil düğme, onun altında ilerleme şeridi ve oturum günlüğü yer alır.

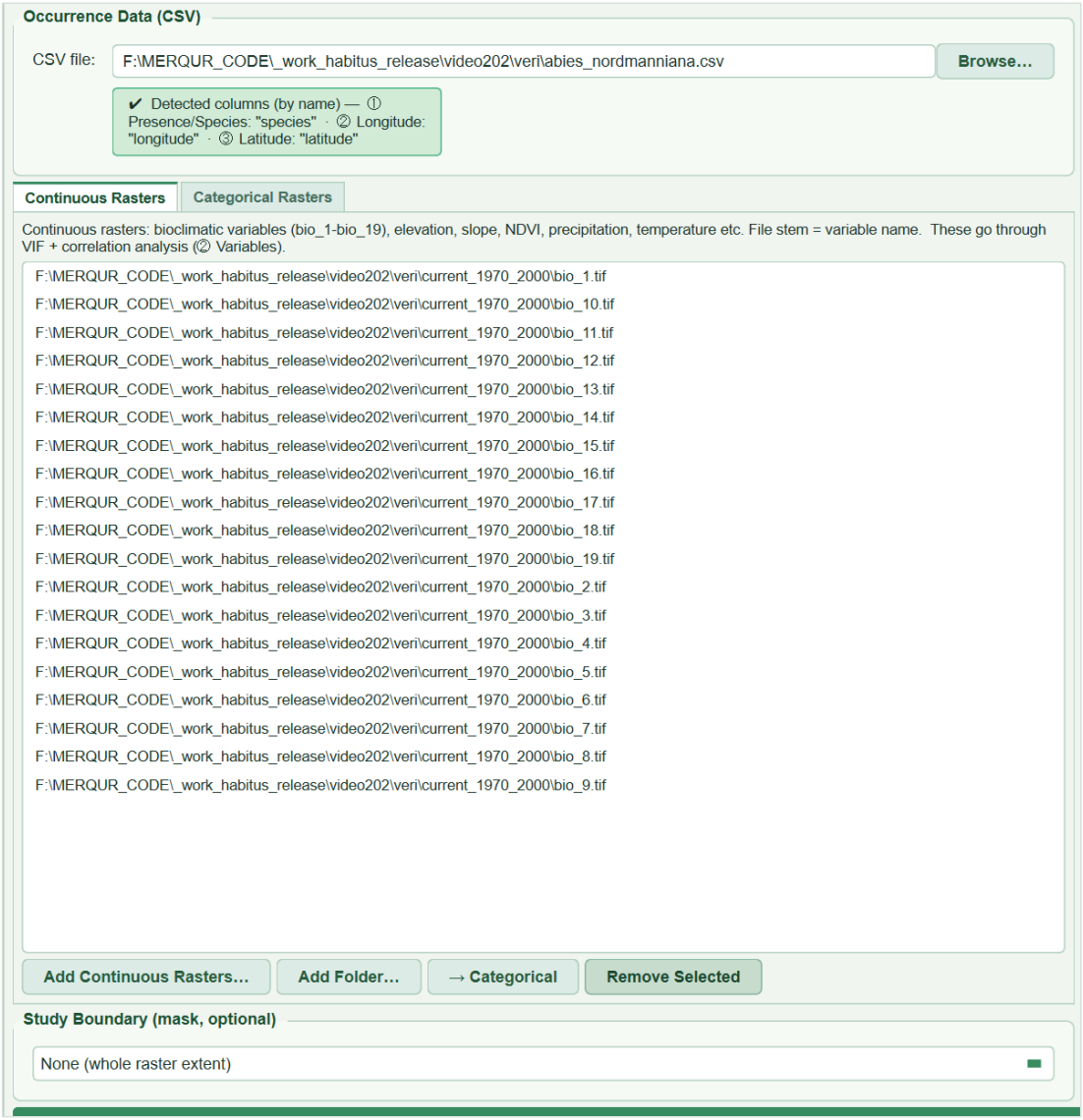


Şekil 2.1. ① Data sekmesi, örnek veri yüklenmiş ve düğmeye basılmadan önce. Solda Occurrence Data (CSV), Continuous/Categorical Rasters alt sekmeleri ve Study Boundary; sağda Background & Pseudo-Absence Settings (burada Study Area & Thinning alt sekmesi açık), Reproducibility ve Output Directory; altta Load Data düğmesi ve günlük.

Bölge	Arayüzdeki ad	Bu kitapta
Sol üst	`Occurrence Data (CSV)`	§2.3
Sol orta	`Continuous Rasters`, `Categorical Rasters`	§2.4
Sol alt	`Study Boundary (mask, optional)`	§2.5
Sağ üst	`Background & Pseudo-Absence Settings` (6 alt sekme)	Bölüm 3
Sağ orta	`Reproducibility`	§2.6
Sağ alt	`Output Directory`	§2.7
Alt şerit	`Load Data & Generate Background/Pseudo-Absence Points`	§2.8, §2.9

2.3 Tür kayıtları: Occurrence Data (CSV)

Bu grup, türün gözlendiği yerlerin koordinatlarını taşıyan tabloyu okur. Dosya seçilince program ilk satırlara bakar, sütunları bulur, yeşil bilgi kutusunda hangi sütunun ne olarak kullanılacağını yazar ve dosyada birden çok tür varsa bir tür listesi açar.



Şekil 2.2. Sol sütun: seçilen kayıt dosyası ve "Detected columns (by name)" bilgi kutusu (species, longitude, latitude adla bulundu), altında Continuous Rasters listesindeki 19 WorldClim biyoiklim katmanı ve Add Continuous Rasters..., Add Folder..., → Categorical, Remove Selected düğmeleri.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`CSV file:` + `Browse…`	Kayıt tablosunu seçer; seçince sütunları ve türleri tarar, çıktı klasörünü önerir	boş	`.csv`, `.tsv`, `.txt`	`tabs/tab_data.py:92-96`, `972-979`
Bilgi kutusu	Bulunan tür, boylam, enlem sütunlarını ve yok sayılacak sütunları gösterir	—	—	`tabs/tab_data.py:99-108`, `1008-1023`
`Species:` listesi	Çok türlü dosyada her tür için bir onay kutusu (kayıt sayısıyla)	hepsi işaretli	`All` / `None`	`tabs/tab_data.py:112-149`, `1045-1072`

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Analyse now:`	①–⑧ iş akışının şimdi hangi tür için çalışacağını seçer	ilk işaretli tür	işaretli türler	`tabs/tab_data.py:132-140`, `1085-1095`

2.3.1 Dosya biçimi, ayırıcı ve ondalık işaret

- **Ne işe yarar.** Program her CSV'yi aynı yordamla okur (`sdm_inputs.py:20-65`). Dosyanın ilk 64 KB'sinden boş olmayan ilk 50 satır alınır; virgül, noktalı virgül ve sekme için her satırdaki sayım çıkarılır ve satırların çoğunda aynı sayıda geçen işaret ayırıcı kabul edilir. Ayırıcı noktalı virgül ya da sekme ise ve gövdede `36,52` biçiminde sayılar varken noktalı ondalık hiç yoksa ondalık işaret virgül sayılır. Dosya UTF-8 olarak okunur; baştaki bayt sırası işareti (BOM) atılır.
- **Ne zaman, nasıl ayarlanır.** Ayar yoktur; Türkçe, Almanca ya da İspanyolca bölge ayarlı bir hesap tablosunun kaydettiği `;` ayırıcılı, virgül ondalıklı dosya olduğu gibi okunur. En güvenli biçim yine virgül ayırıcı ve nokta ondalıktır.
- **Sonuca etkisi.** Yanlış sezilen ayırıcı bütün tabloyu tek sütuna indirir; o durumda program "CSV has only 1 column(s)" uyarısı verir ve hiçbir koordinat okunmaz.
- **Dikkat.** Virgül ayırıcılı bir dosyada koordinatlar `"36,52"` gibi tırnak içinde virgüllü yazılmışsa ondalık virgül sezilmez (sezme yalnız `;` ve sekme için yapılır) ve koordinat sütunu metin olarak kalır. Sayıya çevrilemeyen her koordinat hücresi çalıştırmayı satır numarasıyla durdurur: "column '…' holds non-numeric coordinates in N row(s) (row 5: '36°31′…)" (`sdm_inputs.py:100-123`). Boş koordinat hücreleri ise durdurmaz; o kayıtlar "missing coordinate" gerekçesiyle atılır ve bir uyarı yazılır (`sdm_core.py:2944-2953`). UTF-8 olmayan bir dosya "Re-save it as 'CSV UTF-8'" iletisiyle reddedilir (`sdm_inputs.py:90-93`).

2.3.2 Sütunların bulunması (adla, konumla, GBIF)

- **Ne işe yarar.** `occurrence_columns` önce adlara bakar (`sdm_inputs.py:128-166`). Büyük-küçük harf gözetmeden boylam için `longitude, decimallongitude, lon, long, lng, x`, enlem için `latitude, decimallatitude, lat, y` aranır. İkisi de bulunursa tür sütunu `species, scientificname, acceptedscientificname, verbatimscientificname, taxon` adlarından ilki olur; hiçbiri yoksa koordinat dışı ilk sütun alınır. Koordinat adları bulunamazsa ilk üç sütun konumla okunur: ① tür, ② boylam, ③ enlem. Bilgi kutusundaki "(by name)" eki adla bulunduğunu gösterir. Dosyada üçten fazla sütun varsa fazlası yok sayılır ve bir bilgi penceresi bunları tek tek sayar ("Extra Columns Detected", `tabs/tab_data.py:1030-1041`).
- **Ne zaman, nasıl ayarlanır.** GBIF'ten indirilen Darwin Core tablosu (`gbifID, …, species, …, decimalLongitude, decimalLatitude`) düzenlenmeden kullanılabilir. Kendi tablonuzda sütunlara `species, longitude, latitude` adlarını vermek en açık yoldur; arayüzde sütun seçen ayrı bir alan yoktur.
- **Sonuca etkisi.** Boylam ile enlemin yer değiştirmesi kayıtların çoğunu raster dışına atar. Program bu durumu sezmeye çalışır (§2.8.6).
- **Dikkat.** Ad bulunamadığında sıra kesin olarak "tür, boylam, enlem"dir; "tür, enlem, boylam" sırasındaki bir dosya ters okunur. Yalnız `x` ve `y` adlı sütunlar varsa onlar da koordinat sayılır. Arayüzdeki ilk bilgi metni ("The first 3 columns will always be used") dosya seçilmeden önce görünür; dosya seçilince gerçekte kullanılan sütunlar yazılır, esas olan budur.

2.3.3 Çok türlü dosya ve `Analyse now`

- **Ne işe yarar.** Tür sütununda birden fazla ad varsa `Species:` satırı açılır; her tür kayıt sayısıyla bir onay kutusu olarak listelenir (`tabs/tab_data.py:1045-1072`). Sütundaki bütün değerler sayıysa (ör. 0/1 varlık kodu) bunlar tür adı sayılmaz ve liste açılmaz. `Analyse now:` kutusu işaretli türlerden birini seçer. `Load Data`'ya basıldığında yalnız o türün kayıtları alınır, çıktı klasörünün içinde türün adını taşıyan alt klasöre `occurrences_used.csv` olarak yazılır ve iş akışı bu dosyayla sürer (`tabs/tab_data.py:1500-1506`).
- **Ne zaman, nasıl ayarlanır.** Birçok türü aynı ayarlarla modellemek için: önce bir tür için ①–⑧ tamamlanır, sonra ⑧ Report'taki `Run Selected Species` aynı ayarları öteki işaretli türlere uygular (Bölüm 11).
- **Sonuca etkisi.** Her tür kendi kayıtlarıyla, ayrı ayrı modellenir; türler arasında bilgi paylaşılmaz. Alt klasör adları dosya sistemine uygun hâle getirilir ve büyük-küçük harf farkı gözetilmeden tekilleştirilir ("Quercus robur" ile "quercus robur" ayrı klasöre gider; `tabs/tab_data.py:34-60`).
- **Dikkat.** Hiç tür işaretlenmezse çalıştırma "The occurrence file holds several species. Tick at least one and choose it under 'Analyse now'." iletisiyle durur (`tabs/tab_data.py:1373-1378`). Tür adlarındaki baş ve sondaki boşluklar kırpılır; "Abies nordmanniana" ile "Abies nordmanniana" (iki boşluk) yine iki ayrı tür sayılır.

2.3.4 Tür adı ve önerilen çıktı klasörü

Raporlarda ve dosya adlarında kullanılan tür adı şöyle belirlenir (`sdm_core.py:1566-1593`): ilk sütun koordinat ya da tür sütunu değilse ve sayı değilse oradaki en sık metin; değilse tür sütunundaki en sık metin; o da yoksa CSV dosyasının adı (alt çizgi ve tireler boşluğa çevrilir). Örnek koşuda ad "abies nordmanniana"dır. `Output Directory` boşsa dosya seçildiği anda `Belgeler/HABITUS/<tür>_YYYYMMDD_HHMMSS` klasörü oluşturulur ve alana yazılır (`tabs/tab_data.py:1125-1160`, `main_dialog.py:832-853`); çok türlü dosyada klasör adı CSV'nin adından alınır.

2.4 Öngörücü katmanlar: Continuous Rasters ve Categorical Rasters

Öngörücüler iki ayrı listede tutulur. **Sürekli** (continuous) katmanlar ölçülen büyüklüklerdir: sıcaklık, yağış, yükseklik, eğim, NDVI. **Kategorik** (categorical) katmanlar ise sınıf kodlarıdır: arazi örtüsü, toprak tipi, jeoloji. Bu ayrım sonraki her adımı etkiler: sürekli katmanlar ② Variables'ta VIF ve korelasyon analizine girer, kategorik katmanlar bu analizin dışında kalır ve modellere sınıf olarak verilir.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Add Continuous Rasters…`	Seçilen dosyaları sürekli listeye ekler (aynı yol iki kez eklenmez)	—	`.tif .tiff .asc .img .nc .vrt`	`tabs/tab_data.py:911-912`, `1183-1191`
`Add Folder…`	Klasördeki bütün rasterleri okur, her birini sürekli/kategorik diye sınıflar	—	klasör	`tabs/tab_data.py:913-919`, `1205-1290`
`→ Categorical` / `→ Continuous`	Seçili dosyaları öteki listeye taşır	—	—	`tabs/tab_data.py:920-922`, `948-950`, `1197-1203`
`Remove Selected`	Seçili dosyaları listeden çıkarır	—	—	`tabs/tab_data.py:923-924`, `1193-1195`
`Add Categorical Rasters…`	Seçilen dosyaları kategorik listeye ekler	—	aynı uzantılar	`tabs/tab_data.py:946-947`
Encoding notu	Kategorik kodlamanın etiket (tamsayı sınıf kodu) olduğunu bildirir	label	yalnız label	`tabs/tab_data.py:959-968`, `sdm_core.py:1428-1440`

2.4.1 Tek tek ekleme ve değişken adları

- **Ne işe yarar.** Her dosyanın uzantısız adı değişken adı olur. Ad, boşluk, parantez, tire ve nokta alt çizgiye çevrilerek temizlenir: `bio_ (1)` → `bio_1`, `tr-elev` → `tr_elev` (`sdm_core.py:166-186`). Katmanlar listeye hangi sırayla eklenmiş olursa olsun program değişkenleri "doğal" sıraya dizer (`bio_2 < bio_3 < bio_12`; `sdm_core.py:189-200`, `1617-1620`); bu sayede sonuçlar ekleme sırasına bağlı değildir.
- **Ne zaman, nasıl ayarlanır.** Gelecek senaryolarında (⑤) katmanlar adla eşleştirildiği için güncel ve gelecek dosyalarının adları aynı olmalıdır (`bio_1.tif` ↔ `bio_1.tif`).
- **Sonuca etkisi.** Her katmanın yalnız **ilk bandı** okunur (`sdm_core.py:1608`); çok bantlı bir GeoTIFF ya da NetCDF'in öteki bantları kullanılmaz. Sürekli katmanlar bellekte, kaynak türü kayıpsız taşıyabiliyorsa float32, taşıyamıyorsa float64 tutulur (`sdm_core.py:1367-1379`); değerler bit bit korunur, bütün hesaplar float64'te yapılır.
- **Dikkat.** Katmanın NoData değeri dosyada tanımlı olmalıdır; tanımlı değilse deniz ya da boş alan hücreleri (ör. -3,4e38) gerçek değer sanılır. Temizlendikten sonra aynı adı veren iki dosya (`bio 1.tif` ve `bio_1.tif`) sınır kullanılmadığında denetlenmez; sonra okunan öncekinin yerine geçer. Sınır kullanıldığında aynı ada sahip iki dosya hata verir: "Two predictor rasters have the same name" (`sdm_mask.py:337-345`).

2.4.2 `Add Folder…`: sürekli/kategorik sınıflama ve gerekçesi

- **Ne işe yarar.** `scan_raster_folder` klasördeki `.tif .tiff .asc .img .nc .vrt` dosyalarını ada göre sıralar ve her birini `classify_raster` ile sınıflar (`sdm_mask.py:54-141`). Kurallar sırasıyla şunlardır:
  1. Adı biyoiklim katmanı gibi olan (`bio1`, `bio_12`, `wc2.1_30s_bio_5`) ve renk tablosu taşımayan dosya, değerleri okunmadan **sürekli** sayılır (30 açı saniyelik bir dünya `.asc`'sinin gigabaytlarca metni böylece okunmaz).
  2. Öteki dosyalar seyreltilerek okunur: uzun kenarı en çok yaklaşık 1000 hücre olacak biçimde en yakın komşuyla küçültülür.
  3. Geçerli hücre yoksa sürekli bırakılır ve şüpheli işaretlenir.
  4. Renk tablosu (palet) varsa **kategorik**.
  5. Ondalıklı değer varsa **sürekli**.
  6. Bütün değerler tamsayı ve farklı değer sayısı en çok 30 (`MAX_CATEGORICAL_CLASSES`) ise **kategorik**; en az 12 farklı değer varken adda sınıf çağrıştıran bir sözcük yoksa ya da ad sürekli bir değişken çağrıştırıyorsa (`elev`, `slope`, `ndvi`, `temp`, `prec` …) şüpheli işaretlenir.
  7. 30'dan fazla farklı tamsayı varsa **sürekli**; adda `landcover`, `soil`, `geolog`, `class`, `zone` … geçiyorsa şüpheli.
- **Ne zaman, nasıl ayarlanır.** Biyoiklim ve öteki katmanları tek klasörde tutan kullanıcı için en hızlı yoldur. Sonuç penceresi ("Rasters added") her dosyayı `N` (sürekli) ya da `C` (kategorik) harfiyle ve gerekçesiyle listeler; ⚠ işaretli satırlar elle denetlenmeli, yanlış sınıflanan dosya `→ Categorical` / `→ Continuous` ile taşınmalıdır.
- **Sonuca etkisi.** Örnek veride 19 dosyanın hepsi addan sürekli sayılmıştır (günlükte "19 continuous, 0 categorical, 0 unreadable"). Yanlış listede kalan bir katman modelde yanlış temsil edilir: kategorik listede kalan bir sıcaklık katmanı VIF analizine girmez ve sınıf gibi işlenir.
- **Dikkat.** 1–5 arası verimlilik puanı gibi sıralı tamsayı ölçekler değerlerinden sınıf kodundan ayrılamaz ve işaretlenmeden kategorik sayılır (`sdm_mask.py:48-51`). Açılamayan dosyalar "✕ … could not be opened" satırıyla raporlanır ve listeye eklenmez.

2.4.3 Kategorik katmanların modele girişi

- **Ne işe yarar.** Kategorik katmanlar tamsayıya (int32) çevrilerek okunur, NoData hücreleri `-9999` ile işaretlenir ve her katmanın sınıf listesi çıkarılır (`sdm_core.py:1653-1677`). Örneklenen noktalarda her katman tek bir sütun olarak, sınıf kodunun kendisiyle yer alır (etiket kodlaması). Kod sayı gibi kullanılmaz: GLM, GLMNET, SVM ve ANN boru hattının ilk adımı her sınıfı ayrı bir 0/1 sütuna açar (`_CatPrep`, `sdm_core.py:3364-3384`); GAM faktör terimi, MaxEnt kategorik özellik sınıfı kullanır; ağaç yöntemleri (RF, GBM, BRT, XGBoost, LightGBM, CatBoost, CART) kod üzerinden böler; BIOCLIM, Domain, ENFA ve Mahalanobis kategorik katmanları dışarıda bırakır (`sdm_core.py:1781-1787`).
- **Ne zaman, nasıl ayarlanır.** Arazi örtüsü gibi gerçek sınıf katmanları için. Az kayıtlı bir türde çok sınıflı bir katman, eğitim verisinde hiç ya da çok az kaydı olan sınıflar doğurur; program bunları "sparse classes" uyarısıyla bildirir (`sdm_core.py:3232-3246`) ve ② Variables'ta katmanı dışarıda bırakma olanağı verir.
- **Sonuca etkisi.** Kategorik bir katmanın NoData olduğu hücreler bütün örneklemeden çıkar (`_valid_cell_mask`, `sdm_core.py:2309-2330`); kapsaması sürekli katmanlardan dar olan bir arazi örtüsü katmanı çalışma alanını daraltır. Yalnız bu nedenle kaybedilen kayıt ve hücre sayıları ② Variables'a bildirilir (`sdm_core.py:3073-3078`).
- **Dikkat.** Ondalıklı değerler tamsayıya çevrilirken kesilir (2,7 → 2). Kategorik listedeki dosya da sürekli katmanlarla aynı ızgarada olmalıdır (§2.4.5). En az bir sürekli katman zorunludur; yalnız kategorik katmanla çalıştırma "Please add at least one environmental raster." iletisiyle durur (`tabs/tab_data.py:1368-1369`).

2.4.4 Büyük ASCII ızgaralarının GeoTIFF'e çevrilmesi

- **Ne işe yarar.** `Add Folder…` ile eklenen `.asc` dosyalarının toplam boyutu 200 MB'yi (`_ASC_CONVERT_MB`) aşarsa program bunları bir kez sıkıştırılmış, döşemeli (tiled, DEFLATE) GeoTIFF kopyalarına çevirmeyi önerir ("Convert ASCII grids to GeoTIFF?"; `tabs/tab_data.py:1292-1330`, `sdm_mask.py:144-178`). Kopyalar klasörün içindeki `geotiff/` alt klasörüne aynı adla yazılır ve listedeki yollar bunlarla değiştirilir; `.asc` dosyalarına dokunulmaz. Kopya, kaynaktan yeniyse yeniden üretilmez.
- **Ne zaman, nasıl ayarlanır.** Dünya ölçekli `.asc` katmanlarında "Yes" önerilir: ASCII metin her okumada baştan ayrıştırıldığı için kırpma, örnekleme ve izdüşümün her biri dakikalar sürer; GeoTIFF pencere pencere okunur.
- **Sonuca etkisi.** Değerler ve koordinat sistemi aynı kalır; yalnız süre kısalır.
- **Dikkat.** Öneri yalnız klasörle eklemede çıkar; tek tek eklenen `.asc` dosyaları için sorulmaz.

2.4.5 Izgara uyumu ve koordinat sistemi

- **Ne işe yarar.** Program bütün katmanları hücre hücre tek bir tabloya okur; bu yüzden sınır kullanılmadığında her katman ilk katmanla (doğal sıradaki ilk ad) aynı **satır-sütun sayısına, hücre boyuna, başlangıç noktasına ve koordinat sistemine** sahip olmalıdır. Hücre boyu ve başlangıçta 1e-6 birimlik fark hoş görülür (`sdm_core.py:1683-1744`). Uymayan her katman adıyla ve farkıyla listelenir: "Environmental rasters are not on the same grid. • bio_12: cell size …".
- **Ne zaman, nasıl ayarlanır.** Farklı kaynaklardan gelen katmanlar önceden hizalanmalıdır (QGIS: Raster > Align Rasters). Alternatif olarak bir çalışma sınırı verilirse program katmanları kendisi ortak ızgaraya getirir (§2.5.2).
- **Sonuca etkisi.** Koordinat sistemi coğrafi (derece) ya da projeksiyonlu (metre) olabilir. Mesafeye dayanan her karar (seyreltme, tampon, en az mesafe, blok boyu) büyük daire (haversine) mesafesiyle kilometre cinsinden verilir; projeksiyonlu ızgarada koordinatlar önce boylam/enleme çevrilir (`sdm_spatial.py:113-141`). Projeksiyonlu bir sistemin uzunluk birimi okunamıyorsa çalıştırma durur ("The length unit of the raster coordinate system could not be determined"; `sdm_spatial.py:93-110`).
- **Dikkat. Kayıt CSV'sinin koordinatları rasterlerle aynı koordinat sisteminde olmalıdır.** Program CSV'yi yeniden projekte etmez; koordinatları doğrudan rasterin dönüşüm matrisiyle hücreye çevirir (`sdm_core.py:1763-1765`). UTM'deki rasterle derece cinsinden CSV kullanılırsa kayıtların hepsi raster dışında kalır. Rasterler coğrafi olduğunda enlemi ±90'ı, boylamı ±180'i aşan koordinatlar için uyarı verilir (`sdm_inputs.py:179-196`). Hücre boyu ekvatorda yaklaşık 4 km ya da daha büyükse bir çözünürlük uyarısı yazılır (`sdm_core.py:1751-1761`); örnek koşuda (2,5′, yaklaşık 4,6 km) bu uyarı çıkmıştır.

2.5 Çalışma sınırı: Study Boundary (mask, optional)

Bu grup bütün rasterleri bir çokgenle ya da derece cinsinden bir dikdörtgenle kırpar. Böylece dünya ölçekli katmanlar kesilmeden kullanılabilir ve sınırın dışındaki her hücre NoData olur.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
Kip kutusu	Sınırın türünü seçer	`None (whole raster extent)`	`None (whole raster extent)` / `Polygon layer (SHP, GPKG, KML, GeoJSON)` / `Corner coordinates (degrees)`	`tabs/tab_data.py:173-181`
Dosya alanı + `Browse…` / `Clear`	Çokgen katmanı (yalnız ikinci kipte görünür); dosya seçilince kip kendiliğinden çokgene geçer	boş	`.shp .gpkg .geojson .json .kml .gml`	`tabs/tab_data.py:161-172`, `1332-1339`
`West` / `South` / `East` / `North`	Dikdörtgenin köşeleri, WGS84 derece (yalnız üçüncü kipte)	25 / 35 / 45 / 43	boylam −180…180, enlem −90…90; 4 ondalık	`tabs/tab_data.py:186-197`

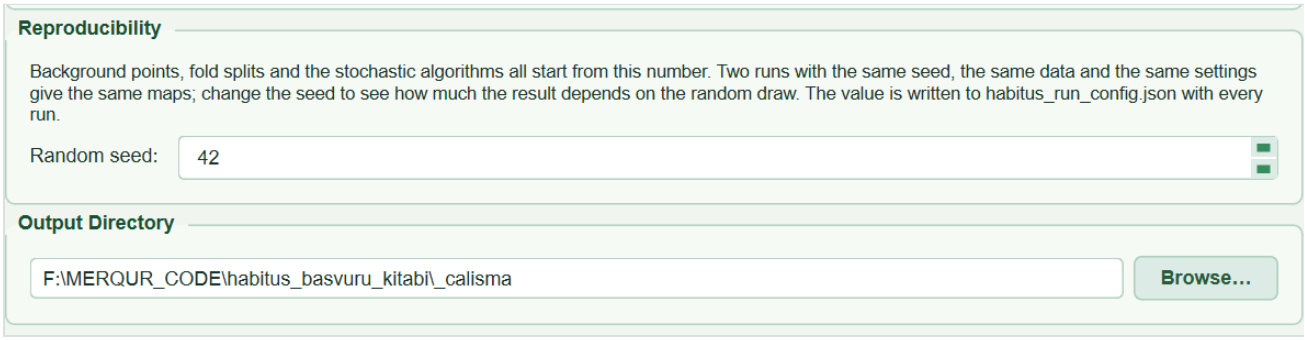
2.5.1 Kipler ve kırpmanın kapsamı

- **Ne işe yarar.** Sınır verildiğinde `Load Data` her şeyden önce bütün sürekli ve kategorik katmanları kırpar (`tabs/tab_data.py:1511-1524`). Bundan sonraki her adım (kayıt ayıklama, yalancı yokluklar, arka plan, çapraz doğrulama katları, güncel harita, ⑤'teki gelecek katmanları ve ⑥ alan değişimi) yalnız sınırın içindeki hücreleri görür; çünkü bir katmanda NoData olan hücre her yerde dışlanır. Köşe koordinatları EPSG:4326 dikdörtgeni olarak alınır ve gerekirse rasterin sistemine çevrilir (`sdm_mask.py:193-201`).
- **Ne zaman, nasıl ayarlanır.** Model ve haritaların bir ülke, havza ya da bölgeyle sınırlanması gerektiğinde. Sınır, Bölüm 3'te anlatılan **erişilebilir alan**dan farklıdır: sınır bütün katmanları ve haritaları kırpar; erişilebilir alan yalnız arka plan ve yalancı yoklukların çekildiği alanı daraltır, harita yine sınırın tamamı için üretilir.
- **Sonuca etkisi.** Sınırın dışına düşen kayıtlar "outside raster extent" ya da "falls on NoData cell" gerekçesiyle atılır. Arka plan yalnız sınırın içinden çekildiği için sınırı daraltmak, erişilebilir alanı daraltmaya benzer biçimde değerlendirme skorlarını etkiler (Bölüm 3).
- **Dikkat.** "West must be smaller than East and South smaller than North" denetimi yapılır (`tabs/tab_data.py:1488-1491`). Çokgen kipinde dosya yoksa "The study boundary layer was not found" iletisiyle durulur. Katmanda çokgen yoksa ("has no polygon features"), sınır rasterlerle hiç örtüşmüyorsa ya da bir katmanın sınır içinde hiç değeri yoksa çalıştırma adıyla birlikte durur (`sdm_mask.py:213-216`, `369-383`, `399-403`). Sınır katmanı ya da rasterler koordinat sistemi bildirmiyorsa ikisinin aynı olduğu varsayılır ve bu durum bir uyarıyla rapora yazılır (`sdm_mask.py:228-236`).

2.5.2 Kırpma nasıl yapılır: tek pencere, ortak ızgara, `masked_inputs`

- **Ne işe yarar.** `mask_rasters` (`sdm_mask.py:304-418`) listenin ilk katmanı üzerinde sınırın sınırlayıcı kutusuna karşılık gelen **tek bir pencere** hesaplar; bütün katmanlar bu pencerenin ızgarasına yazılır. Hücre boyu ve hizası aynı olan katman pencereyle okunur (değerler değişmez); farklı çözünürlükte, kaymış başlangıçlı ya da başka koordinat sistemindeki katman **en yakın komşu** yöntemiyle bu ızgaraya yeniden örneklenir (`sdm_mask.py:273-301`). Ardından çokgenin dışındaki hücreler NoData yapılır. Kırpılmış kopyalar özgün adlarıyla `<çıktı klasörü>/masked_inputs/current/` altına GeoTIFF olarak yazılır; ilerleme şeridinde "Rasters clipped to the study boundary: R x C cells." görülür.
- **Ne zaman, nasıl ayarlanır.** Farklı kapsamda kesilmiş ya da bir hücre geniş indirilmiş katmanları (GCM setlerinde sık görülür) birlikte kullanmak için sınır kipi pratik bir hizalama yoludur.
- **Sonuca etkisi.** En yakın komşu yöntemi yeni değer üretmez ve sınıf kodlarını korur; fakat ince çözünürlüklü sürekli bir katman kaba ızgaraya indirildiğinde ortalama değil tek bir hücrenin değeri alınır. ⑤'teki gelecek katmanları aynı ızgaraya kırpılır, böylece güncel ve gelecek haritaları hücre hücre örtüşür.
- **Dikkat.** Izgarayı belirleyen, listenin ilk katmanıdır (önce sürekli, sonra kategorik liste, eklenme sırasıyla). En ince çözünürlüklü katmanı ilk sıraya koymak, bilginin kaybolmasını önler. `masked_inputs` klasörü çok türlü dosyada da ana çıktı klasöründedir; türler aynı kırpılmış kopyaları paylaşır.

2.6 Tekrarlanabilirlik: Reproducibility



Şekil 2.3. Reproducibility grubu (Random seed 42) ve Output Directory alanı; açıklama metni tohumun nelere uygulandığını ve habitus_run_config.json dosyasına yazıldığını belirtir.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Random seed:`	Bütün rastgele işlemlerin başlangıç sayısı	42	0 – 2147483647	`tabs/tab_data.py:242-249`, `run_config.py:28`

2.6.1 Random seed

- **Ne işe yarar.** Veri adımı başlarken `set_run_seed` NumPy'nin genel üretecini, Python'un `random` modülünü ve `PYTHONHASHSEED` ortam değişkenini bu sayıyla tohumlar (`tabs/tab_data.py:1510`, `run_config.py:31-50`). `DataFormatter` ayrıca kendi üretecini (`numpy.random.default_rng(seed)`) bu sayıdan kurar; yalancı yokluk, arka plan çekilişi, seyreltme sırası, variogram alt örneği ve k-ortalamalar bundan türer (`sdm_core.py:1503-1505`, `2301-2302`, `2536`). Modellerin `random_state` değeri de aynı sayıdan verilir (Bölüm 5).
- **Ne zaman, nasıl ayarlanır.** Yayımlanacak bir çalışmada değer raporlanır; `habitus_run_config.json` dosyasının `random_seed` alanına her çalıştırmada yazılır. Sonucun rastgele çekilişe ne kadar bağlı olduğunu görmek için aynı ayarlarla birkaç farklı tohum denenir.
- **Sonuca etkisi.** Aynı veri, aynı ayar ve aynı tohumla aynı haritalar beklenir. Tohum değişince yalancı yokluk ve arka plan noktaları, katlar ve stokastik algoritmalar değişir; skorlardaki ve haritalardaki oynama, verinin bu çekilişlere duyarlılığını gösterir.
- **Dikkat.** `PYTHONHASHSEED` yalnız tanımlı değilse atanır (`setdefault`) ve Python yorumlayıcısı başladıktan sonra atanması karma sıralamasını etkilemez. Çok iş parçacıklı kütüphanelerin (ör. bazı gradyan artırma uygulamaları) farklı makinelerde bit bit aynı sonucu verip vermediği kodda güvenceye alınmaz ve sınanmamıştır; aynı makinede aynı kurulumla tekrarlanabilirlik amaçlanmıştır.

2.7 Çıktı klasörü: Output Directory

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
Klasör alanı + `Browse...`	Bütün çıktıların yazılacağı klasör	CSV seçilince `Belgeler/HABITUS/<tür>_<zaman>`	herhangi bir klasör	`tabs/tab_data.py:253-260`, `1125-1160`

Sonucu etkilemez; boş bırakılırsa "Please specify an output directory." iletisiyle durulur. Çok türlü dosyada her tür `<çıktı klasörü>/<tür>` alt klasörüne yazılır (`tabs/tab_data.py:1115-1123`); oturum günlüğü de

çalıştırma başında bu klasöre taşınır, böylece toplu çalıştırmada her türün günlüğü kendi klasöründe olur (`tabs/tab_data.py:1398-1408`).

2.8 Load Data düğmesi: işlem sırası ve kayıt ayıklama

`▶ Load Data & Generate Background/Pseudo-Absence Points` düğmesi bütün adımı arka planda, arayüzü dondurmadan çalıştırır (`tabs/tab_data.py:1363-1574`). İşlem sırası, hangi kaydın neden atıldığını anlamak için önemlidir:

1. **Ön denetim:** CSV var mı, en az bir sürekli raster var mı, çıktı klasörü dolu mu, hedef grup CSV'si ve yanlılık rasteri (verildiyse) var mı, sınır geçerli mi.
2. Önceki bir veri adımı varsa sonraki sekmelerin bütün sonuçları sıfırlanır.
3. Çok türlü dosyada yalnız seçili türün kayıtları `occurrences_used.csv`'ye ayrılır.
4. Tohum atanır (§2.6).
5. Sınır verildiyse bütün katmanlar kırpılır (§2.5).
6. `load_data`: CSV ve katmanlar okunur, tür adı belirlenir, ızgara uyumu denetlenir (`sdm_core.py:1549-1681`).
7. `generate_pa_datasets`: koordinatlar sayıya çevrilir, kayıtlar ayıklanır (bu bölüm), seyreltme ve erişilebilir alan uygulanır, yalancı yokluk kümeleri ve arka planlar çekilir (Bölüm 3), çoklu doğrusallık ön hesabı yapılır (`sdm_core.py:2911-3249`).
8. Erişilebilir alan `accessible_area.tif` olarak yazılır; istenmişse MaxEnt arka planı CSV'ye kaydedilir.

2.8.1 Kayıt ayıklamanın adımları

Varlık kayıtları aşağıdaki sırayla süzülür (`sdm_core.py:2931-2968`, `1824-2037`). Her atılan kayıt gerekçesiyle `dropped_points` listesine girer; liste uyarı penceresinde, günlükte, raporda ve `habitus_run_config.json`'da (`points_dropped_by_reason`) görünür.

Sıra	Gerekçe (arayüzdeki metin)	Kural	Kapatılabilir mi	Kod
1	`missing coordinate`	Boylam ya da enlem hücresi boş	hayır	`sdm_core.py:2944-2953`
2	`duplicate coordinate`	Koordinatlar 8 ondalığa yuvarlandığında daha önceki bir kayıtla aynı	hayır	`sdm_core.py:1844-1856`
3	`outside raster extent`	Hücre satırı/sütunu ızgaranın dışında	hayır	`sdm_core.py:1864-1876`
4	`falls on NoData cell`	Herhangi bir sürekli katmanda NaN ya da herhangi bir kategorik katmanda NoData	hayır	`sdm_core.py:1878-1905`
5	`same raster cell as another record`	Aynı öngörücü hücresinde dosya sırasında daha önce gelen bir kayıt var	evet: `Keep one record per raster cell`	`sdm_core.py:1913-1936`
6	`spatially thinned`	Seyreltme mesafesi içinde tutulan bir kayıt var	evet (Bölüm 3)	`sdm_core.py:2960-2968`

2.8.2 Yinelenen koordinat, raster dışı ve NoData

- **Ne işe yarar.** Birebir aynı koordinat (yaklaşık 1 mm'lik yuvarlama farkıyla) yalnız ilk kez geçtiği yerde tutulur. Kalanlar arasından ızgaranın dışına düşenler ve bir katmanda değer taşımayan hücreye düşenler atılır. Kategorik bir katman yüzünden atılan kayıtlar ayrıca sayılır (`cat_nodata_loss_presence`) ve ② Variables'a bildirilir.
- **Ne zaman, nasıl ayarlanır.** Bu üç süzgeç her zaman çalışır; ayar yoktur.
- **Sonuca etkisi.** Kıyıdaki kayıtların bir kısmı kaba çözünürlükte deniz hücresine düşerek NoData'ya takılabilir. Atılan kayıtların payı, kullanılan varlık sayısını ve dolayısıyla Bölüm 3'teki prevalansı belirler.
- **Dikkat.** Yinelenmeyen kayıtların %20'sinden fazlası (`LOSS_WARN_FRACTION`) raster dışında ya da NoData'da kalırsa program boylam ile enlemi yer değiştirip yeniden dener; değiştirilmiş hâlde kayıtların en az yarısı içeride kalıyorsa "The lon/lat columns are probably swapped" uyarısını, değilse koordinat sistemini ve sütunları denetlemenizi isteyen uyarıyı verir (`sdm_inputs.py:199-221`, `sdm_core.py:1969-1990`). Bütün kayıtlar düştüğünde ve bir katman içerideki her kayıtta NoData ise suçlu koordinatlar değil o katmandır; ileti katmanı adıyla gösterir (`sdm_core.py:2014-2029`).

2.8.3 Hücre başına bir kayıt

- **Ne işe yarar.** Aynı öngörücü hücresine düşen kayıtlar aynı öngörücü değerlerini taşır; her fazla kayıt o hücreye bilgi eklemeden ağırlık ekler. Seçenek açıkken her hücrede dosya sırasındaki ilk kayıt tutulur, ötekiler "same raster cell as another record" gerekçesiyle atılır (`sdm_core.py:1913-1936`). Seçenek kapalıyken de bu kayıtlar sayılır ve bir uyarı yazılır (`sdm_core.py:1962-1967`). Kutunun kendisi Study Area & Thinning alt sekmesindedir (Bölüm 3, `tabs/tab_data.py:722-731`); varsayılanı açıktır.
- **Ne zaman, nasıl ayarlanır.** Neredeyse her zaman açık kalmalıdır. Yalnız çok az kaydı olan ve kayıtları birkaç hücrede toplanan türlerde kapatmak gündeme gelir; o durumda da kayıtlar aynı hücre değerlerini tekrarlar.
- **Sonuca etkisi.** Örnek koşuda 626 kaydın 405'i daha önce kaydı olan bir hücreye düştüğü için atılmış, 221 kayıt kalmıştır (oturum günlüğü). Kapalı tutulsaydı yoğun gözlenen birkaç alan (ör. aynı vadide yüzlerce kayıt) hem model eğrilerini hem de çapraz doğrulama katlarını domine ederdi.
- **Dikkat.** Kural seyreltmeden önce uygulanır. Hangi kaydın tutulacağı dosya sırasına bağlıdır; aynı hücredeki kayıtların koordinatları farklı olsa da öngörücü değerleri aynı olduğu için model açısından fark yoktur, ancak seyreltme ve mekânsal bloklar kaydın tam konumunu kullanır. Dış test kümesi (⑦) bu kuraldan muaftır.

2.8.4 En az beş varlık

- **Ne işe yarar.** Süzgeçlerden sonra en az 5 kayıt (`MIN_PRESENCES`, `sdm_core.py:1269`) kalmazsa çalıştırma durur. İleti nedene göre değişir (`sdm_core.py:1997-2035`): kayıp yalnız hücre başına bir kural yüzündense "The N presence record(s) fall in only K predictor cell(s)…", hiç kayıp yoksa "Only K usable presence record(s). At least 5 are required…", kayıplar raster dışı/NoData kaynaklıysa koordinat sistemi uyarısı.
- **Ne zaman, nasıl ayarlanır.** Değiştirilemez. Eşik, SDM yöntemlerinin karşılaştırıldığı yayımlanmış çalışmalardaki en küçük örneklemlere dayanır (Hernandez vd. 2006; Pearson vd. 2007).
- **Sonuca etkisi.** 5–24 kayıtta çalıştırma sürer ama uyarılar çıkar (§2.8.5).
- **Dikkat.** Bu sınır seyreltmeden **önce** denetlenir. Seyreltme sonrası 10'dan az kayıt kalırsa yalnız uyarı verilir ("Thinning at … km leaves only N records"; `sdm_core.py:2547-2552`), çalıştırma durmaz.

2.8.5 Kayıt sayısı ve kayıt/öngörücü oranı uyarıları

- **Ne işe yarar.** `_check_sample_size` (`sdm_core.py:2843-2885`) seyreltme sonrası kalan varlık sayısını (n) yüklenen öngörücü sayısıyla (k: sürekli + kategorik) karşılaştırır ve üç uyarı üretebilir:
  - $n < 10$: "Only N usable presence record(s). No species distribution method produces a trustworthy model at this sample size…"
  - $n < 25$ (`JACKKNIFE_RECOMMENDED_MAX`, `sdm_spatial.py:594`): ③ Models'ta jackknife (her seferinde bir varlığı dışarıda bırakan) doğrulamanın seçilmesi önerilir (Pearson vd. 2007).
  - $n \geq 10$ ve $n/k < 10$: "N presence records for the K predictors loaded (R per predictor)…". Metin, ② Variables'ta tutulacak öngörücü sayısının asıl önemli sayı olduğunu ve n kaydın yaklaşık $\lfloor n/10 \rfloor$ öngörücüyü taşıyabileceğini söyler.
- **Ne zaman, nasıl ayarlanır.** Değişken başına 10 kayıt kuralı lojistik türü modeller için yaygın yönergedir; esnek öğreniciler (RF, artırma, menteşe özellikli MaxEnt) daha fazlasını ister (Wisz vd. 2008).
- **Sonuca etkisi.** Örnek koşuda 143 kayıt ve 19 katman için oran 7,5'tir; ileti "143 records allow about 14" der. ② Variables'ta 8 değişken tutulunca (oran 17,9) aynı denetim yeniden yapılır ve uyarı kalkar (`note_selected_predictors`, `sdm_core.py:2889-2909`; `tabs/tab_vif.py:1213`).
- **Dikkat.** Uyarılar sonucu değiştirmez; rapora "advisory" olarak yazılır. Bu bölümdeki k, ① Data'da yüklenen bütün katmanları sayar; seçimden sonraki sayı için ②'deki ileti esas alınır.

2.8.6 Veri adımı sonrası iletiler

Adım bitince sırasıyla şu pencereler çıkar (`tabs/tab_data.py:1576-1761`):

1. **"Invalid Occurrence Points Detected"**: en az bir kayıt atıldıysa; gerekçe başına bir başlık ve koordinat listesi (hücre ve seyreltme başlıklarında ilk 15 kayıt, kalanı "… and N more"), sonda "✓ N valid point(s) will be used for modelling." ve CRS hatırlatması. Aynı metin günlüğe yazılır.
2. **"Points worth knowing before you interpret this run"**: sonucu yorumlamayı etkileyen uyarılar (çözünürlük, kayıt/öngörücü oranı, erişilebilir alan, seyreltme, yokluk kayıtları…). Yalnız örnekleme yanlılığı hatırlatması varsa bu pencere açılmaz, uyarı yalnız günlüğe "ADVISORY:" önekiyle yazılır.
3. **"Data ready!"** özeti: kullanılan/yüklenen varlık sayısı, atılanların gerekçelere göre dökümü, yalancı yokluk ya da gerçek yokluk satırı, MaxEnt arka plan sayısı, erişilebilir alan ve kapsadığı pay, seyreltme mesafesi ve nasıl seçildiği, özilişki aralığı, prevalans, örnekleme yanlılığının düzeltilmediği notu, tohum, sürekli ve kategorik değişken adları.

Örnek koşunun özeti, oturum günlüğünden derlendiğinde şöyledir: 626 kayıt yüklendi; 405'i aynı hücre, 78'i seyreltme nedeniyle atıldı; 143 varlık kullanıldı; 2 küme × 1000 yalancı yokluk; 10 000 MaxEnt arka planı; erişilebilir alan tampon, 200 km, raster kapsamının %40'ı; seyreltme 4,7 km (otomatik); özilişki aralığı 151,5 km; prevalans 0,125.



Şekil 2.4. Veri adımı bittikten sonra alt şerit: ilerleme çubuğunda "Ready: 143 presences, 2 PA sets, 19 cont + 0 cat vars", günlükte kayıt/öngörücü uyarısının sonu ve ② Variables'a geçişle kaydedilen korelasyon ısı haritası satırı.

2.9 Çıktılar ve sonraki adıma geçenler

- **Bellekte (② ve sonrası için):** öngörücü dizileri, değişken adları (`var_names`, `cat_var_names`), yalancı yokluk kümeleri (`pa_datasets`), MaxEnt ve ENFA arka planları, ayıklanmış varlıklar, özilişki aralığı (③'te mekânsal blok boyu olarak kullanılır), çoklu doğrusallık ön sıralaması (`collinearity_df`, ② Variables'ın başlangıç tablosu).
- **Diskte:** `masked_inputs/current/*.tif` (sınır verildiyse), `occurrences_used.csv` (çok türlü dosyada), `accessible_area.tif` (erişilebilir alan tanımlıysa; 1 = içeride, 0 = dışarıda), `background_maxent.csv` (istenmişse), oturum günlüğü `habitus_<zaman>.log`.
- **Çalıştırma kaydı:** `habitus_run_config.json`'un `data` bölümü kullanılan CSV'yi, tür adını, kullanılan varlık sayısını, atılanların gerekçe dökümünü, raster listelerini, ızgara bilgisini (`raster_grid`: başvuru katmanı, satır-sütun, hücre boyu, ekvatorda km, CRS) ve prevalansı içerir; `study_boundary` bölümü sınırı, çokgen sayısını ve kırpılmış ızgarayı. Bu kayıt ③ Models çalıştırıldığında yazılır.

2.10 Uzman soruları

S1. Koordinatlar rasterin koordinat sistemine çevriliyor mu, yoksa aynı sistemde olmaları mı bekleniyor? Cevap: Aynı sistemde olmaları beklenir. Kayıtlar rasterin dönüşüm matrisiyle doğrudan satır-sütuna çevrilir (`sdm_core.py:1763-1765`, `1866`); CSV için bir koordinat sistemi alanı ya da yeniden projeksiyon yoktur. Yeniden projeksiyon yalnız sınır ve kalibrasyon çokgenleri için yapılır (`sdm_mask.py:219-227`, `sdm_spatial.py:1194-1225`). Projeksiyonlu rasterlerde CSV de metre cinsinden olmalıdır; coğrafi rasterlerde ±90/±180 dışındaki değerler uyarı üretir.

S2. Hücre başına bir kayıt kuralı R'deki hangi işleme denk düşer ve neden seyreltmeden önce uygulanıyor? Cevap: Öngörücü ızgarasında hücre kimliğine göre tekilleştirmedir; dismo'daki `gridSample` ile ızgara başına bir nokta seçmeye ya da hücre numarasına göre `duplicated` ile ayıklamaya benzer. HABITUS her hücrede dosya sırasındaki ilk kaydı tutar (`sdm_core.py:1919-1936`). Önce uygulanmasının nedeni, aynı hücredeki kayıtların öngörücü açısından birer kopya olması ve seyreltmenin bu kopyaları rastgele sırada eleyerek gereksiz zaman harcamasıdır; ayrıca mesafeye dayalı seyreltme hücre altı ölçekte anlamsızdır. Seyreltme mesafesi hücreden küçük girilirse program bunu bildirir (`sdm_core.py:2526-2534`).

S3. Yinelenen koordinat ne kadar hassas karşılaştırılıyor? Cevap: Boylam ve enlem 8 ondalığa yuvarlanıp karşılaştırılır (`sdm_core.py:1848`); derecede bu yaklaşık 1 mm'dir. Yani yalnız gerçekten aynı nokta yinelenen sayılır; birkaç metre arayla girilmiş kayıtlar bu süzgeçten geçer ve hücre başına bir kural ile seyreltmeye kalır.

S4. Boylam ve enlemin yer değiştirdiği nasıl anlaşılıyor? Cevap: Yinelenmeyen kayıtların %20'sinden fazlası raster dışı ya da NoData'ya düşerse kayıtlar enlem-boylam yer değiştirilmiş olarak bir kez daha ızgaraya yerleştirilir; bu hâlde içeride kalan pay en az 0,5 ve gerçek içeride kalma payından büyükse iletide sütunların büyük olasılıkla ters olduğu söylenir (`sdm_core.py:1972-1990`, `sdm_inputs.py:207-216`). Program sütunları kendiliğinden çevirmez; karar kullanıcıya bırakılır.

S5. En az kaç kayıt gerekiyor, bu sayı nereden geliyor ve az kayıtla ne öneriliyor? Cevap: Süzgeç sonrası en az 5 kayıt zorunludur (`sdm_core.py:1269`, `1993-2013`); gerekçe SDM yöntemlerinin küçük örneklemle sınandığı çalışmalardır (Hernandez vd. 2006; Pearson vd. 2007). 10'un altında güvenilir model üretilemeyeceği, 25'in altında jackknife doğrulamanın seçilmesi gerektiği uyarıları yazılır (`sdm_core.py:2864-2875`). Çok az kayıtlı türler için ③ Models'taki ESM (küçük modeller topluluğu) seçeneği ayrıca anlatılmaktadır (Bölüm 5).

S6. "10 kayıt / öngörücü" kuralı hangi öngörücü sayısıyla uygulanıyor? Cevap: ① Data'da yüklenen bütün katmanların sayısıyla (`sdm_core.py:2858`); bu sayı genellikle nihai modeldeki sayıdan büyüktür. ② Variables'ta seçim onaylanınca aynı denetim seçilen sayıyla tekrarlanır ve önceki ileti yerine yenisi konur

(`sdm_core.py:2889-2909`). Kural yönlendiricidir; çalıştırmayı durdurmaz. Kaynak: Wisz vd. (2008) ve lojistik regresyondaki olay/değişken oranı geleneği.

S7. Farklı çözünürlükte ya da kapsamda katmanlar birlikte kullanılabilir mi? Cevap: Sınır verilmezse hayır: satır-sütun sayısı, hücre boyu, başlangıç ve CRS aynı olmalıdır, aksi hâlde çalıştırma uymayan katmanı adıyla gösterip durur (`sdm_core.py:1698-1744`). Sınır verilirse evet: bütün katmanlar listenin ilk katmanının hücreleriyle tanımlanan ortak ızgaraya en yakın komşu yöntemiyle getirilir (`sdm_mask.py:283-301`). Program ortalama alarak toplulaştırma (aggregate) yapmaz; ince bir katmanı kaba ızgaraya indirmek için önceden bir GIS aracıyla toplulaştırmak daha doğru olur.

S8. Kategorik bir katman neden doğrusal modellerde sayı gibi kullanılmıyor; hangi algoritmalar onu dışarıda bırakıyor? Cevap: Katman tek sütunda sınıf koduyla taşınır, ancak GLM, GLMNET, SVM ve ANN boru hattının ilk adımı (`_CatPrep`) eğitimde görülen her sınıf için ayrı 0/1 sütun üretir; kod hiçbir zaman karesi alınan, ölçeklenen ya da çarpılan bir sayı olarak kullanılmaz (`sdm_core.py:3364-3384`). GAM faktör terimi, MaxEnt kategorik özellik kullanır; ağaç yöntemleri kod üzerinden böler; BIOCLIM, Domain, ENFA ve Mahalanobis katmanı dışarıda bırakır (`sdm_core.py:1781-1787`). Eğitimde görülmeyen bir sınıf izdüşüm alanında karşılaşılırsa o değişkenin bütün 0/1 sütunları sıfır olur.

S9. Aynı tohumla iki çalıştırma bit bit aynı sonucu verir mi? Cevap: Amaç budur: tohum veri adımının başında bütün üreteçlere uygulanır (`tabs/tab_data.py:1510`, `run_config.py:31-50`) ve arka plan çekilişi, seyreltme, variogram alt örneği, k-ortalamalar ve katlar kendi tohumlu üreteçlerinden gelir. Katman ekleme sırası da sonucu değiştirmez, çünkü değişkenler doğal sıraya dizilir (`sdm_core.py:1617-1620`). Farklı işletim sistemleri ya da kütüphane sürümleri arasında bit bit eşitlik kodda güvenceye alınmamış ve sınanmamıştır; bu yüzden çalıştırma kaydı kütüphane sürümlerini de yazar (`habitus_run_config.json`, `environment`).

S10. Çalışma sınırı ile erişilebilir alan arasındaki fark nedir; biri ötekinin yerine kullanılabilir mi? Cevap: Sınır, bütün katmanları kırpar: dışarıdaki hücreler NoData olur, harita da sınırla biter (`sdm_mask.py:304-418`). Erişilebilir alan yalnız arka plan ve yalancı yoklukların çekileceği hücreleri sınırlar; harita yine sınırın ya da raster kapsamının tamamı için üretilir (`sdm_core.py:2181-2203`). Barve vd. (2011) anlamındaki M alanını tanımlamak için erişilebilir alan kullanılmalıdır; sınırı M olarak kullanmak modelin izdüşüm alanını da M'ye indirir.

S11. Bellek gereksinimi nasıl hesaplanıyor? Cevap: Her sürekli katman tam olarak belleğe okunur; float32'ye kayıpsız sığan kaynaklar (float32, 16 bitlik tamsayılar) float32, ötekiler float64 tutulur (`sdm_core.py:1367-1379`). Kod açıklamasına göre 2,5′ çözünürlükte 19 dünya katmanı 2,6 GB yer tutar. Belleği azaltmanın yolu bir çalışma sınırı vermektir; kırpılmış katmanlar yalnız sınırın kutusunu kapsar.

2.11 Sık hatalar ve tuzaklar

- Rasterler UTM'de, CSV derece cinsinden: kayıtların hepsi "outside raster extent" olur; ileti koordinat sistemini denetlemeyi söyler. CSV'yi raster sistemine çevirin.
- Sütun adları yok ve sıra "tür, enlem, boylam": konumla okumada ters gelir; "probably swapped" uyarısı çıkar. Sütunlara `longitude` ve `latitude` adlarını verin.
- Koordinatlarda derece-dakika-saniye ya da tırnaklı ondalık virgül: "holds non-numeric coordinates" iletisiyle durur.
- Katmanlar farklı kapsamda kesilmiş: "Environmental rasters are not on the same grid." Hizalayın ya da bir çalışma sınırı verin.
- NoData değeri tanımlı olmayan katman: deniz hücreleri gerçek değer sanılır; kıyı kayıtları elenmez, arka plan denize düşer. Katmanın NoData değerini tanımlayın.
- Çok bantlı dosya: yalnız ilk bant okunur.
- Kategorik listede kalan bir sürekli katman (ör. tamsayı olarak saklanmış sıcaklık): ⚠ işaretine bakın ve → `Continuous` ile taşıyın.
- Sınırın ilk sıradaki katmanı kaba çözünürlükte: bütün katmanlar o ızgaraya indirilir.
- Çok türlü dosyada hiç tür işaretlenmemiş: çalıştırma durur.

2.12 Kaynaklar

Barve, N., Barve, V., Jiménez-Valverde, A., Lira-Noriega, A., Maher, S.P., Peterson, A.T., Soberón, J., Villalobos, F., 2011. The crucial role of the accessible area in ecological niche modeling and species distribution modeling. Ecological Modelling 222, 1810–1819. https://doi.org/10.1016/j.ecolmodel.2011.02.011

Hernandez, P.A., Graham, C.H., Master, L.L., Albert, D.L., 2006. The effect of sample size and species characteristics on performance of different species distribution modeling methods. Ecography 29, 773–785. https://doi.org/10.1111/j.0906-7590.2006.04700.x

Pearson, R.G., Raxworthy, C.J., Nakamura, M., Peterson, A.T., 2007. Predicting species distributions from small numbers of occurrence records: a test case using cryptic geckos in Madagascar. Journal of Biogeography 34, 102–117. https://doi.org/10.1111/j.1365-2699.2006.01594.x

Wisz, M.S., Hijmans, R.J., Li, J., Peterson, A.T., Graham, C.H., Guisan, A., NCEAS Predicting Species Distributions Working Group, 2008. Effects of sample size on the performance of species distribution models. Diversity and Distributions 14, 763–773. https://doi.org/10.1111/j.1472-4642.2008.00482.x

3. Veri II: arka plan, yalancı yokluk, erişilebilir alan ve seyreltme (① Data)

Sekme: ① Data › Background & Pseudo-Absence Settings · Kod: `tabs/tab_data.py` (arayüz), `sdm_core.py` `DataFormatter.generate_pa_datasets`, `_sample_background` (hesap), `sdm_sampling.py` (stratejiler, yanlılık yüzeyi), `sdm_spatial.py` (erişilebilir alan, seyreltme, variogram)

3.1 Bu bölüm ne anlatır

Çoğu tür kaydı yalnız varlık bildirir: türün nerede görüldüğü bilinir, nerede bulunmadığı bilinmez. Bir modelin varlığı ayırt edebilmesi için ona "karşılaştırma" noktaları gerekir. HABITUS bu noktaları üç ayrı amaçla, üç ayrı küme olarak üretir:

- **Yalancı yokluklar (pseudo-absences):** GLM, GAM, RF, GBM, BRT, SVM, ANN, XGBoost, LightGBM, CatBoost, GLMNET, CART gibi iki sınıflı (varlık/yokluk) algoritmalara "yok" etiketiyle verilen noktalar. Birden çok bağımsız küme (tekrar) çekilebilir.
- **MaxEnt arka planı (background):** MaxEnt'in "kullanılabilir çevre"yi tanımlamak için kullandığı örnek. Yokluk sayılmaz; bir kez çekilir ve bütün MaxEnt uydurmalarında kullanılır.
- **ENFA arka planı:** ENFA'nın küresel çevre ortalamasını ve kovaryansını hesaplamak için kullandığı ayrı örnek.

Bu üç kümenin **nereden** çekildiğini erişilebilir alan, **hangi ağırlıkla** çekildiğini örnekleme yanlılığı yüzeyi, **hangi kayıtlara göre** çekildiğini seyreltme belirler. Hepsi ① Data'nın sağ üstündeki `Background & Pseudo-Absence Settings` grubunun altı alt sekmesindedir ve `Load Data` düğmesiyle birlikte uygulanır (Bölüm 2.8).

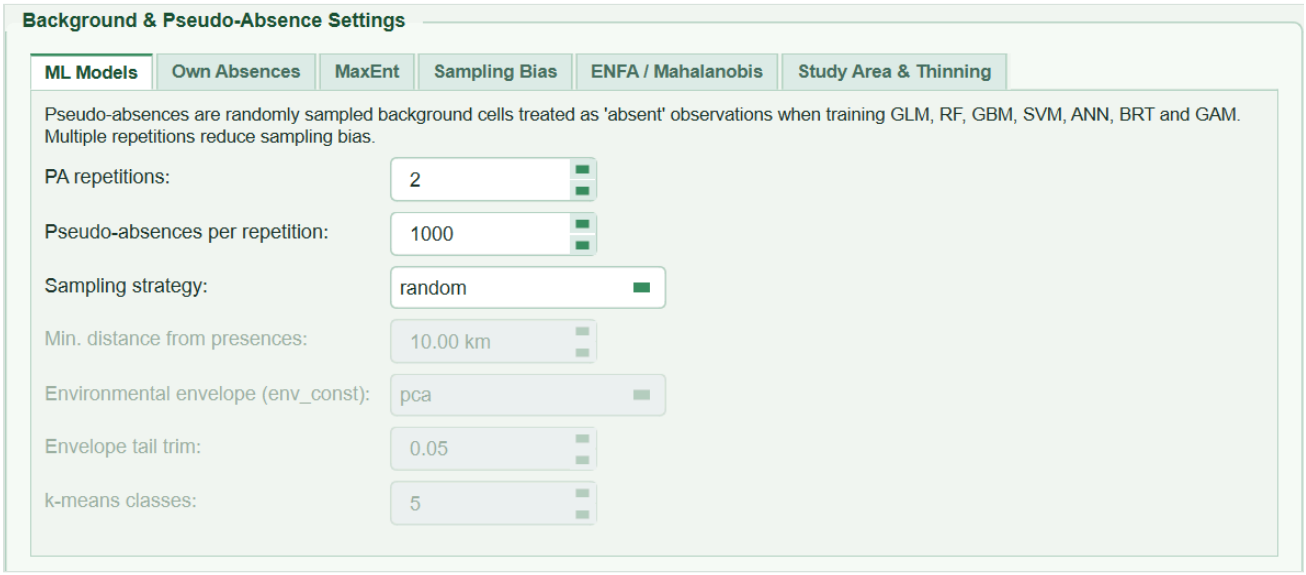
İşlem sırası (`sdm_core.py:2955-3207`): kayıt ayıklama → seyreltme → erişilebilir alan maskesi (seyreltilmiş kayıtlardan) → kullanıcının yokluk kayıtları → yanlılık yüzeyi → strateji kısıtları (zarf, k-ortalamalar) → yalancı yokluk kümeleri → MaxEnt arka planı → ENFA arka planı.

3.2 Ekranın düzeni

Alt sekme	İpucu metni	Hangi küme	Bu kitapta
`ML Models`	ML Models (Pseudo-Absences)	yalancı yokluk	§3.3
`Own Absences`	My Own Absence Data	yalancı yokluğun yerine/yanına gerçek yokluk	§3.4
`MaxEnt`	MaxEnt (Background Points)	MaxEnt arka planı	§3.5
`Sampling Bias`	Sampling Bias (Target Group)	üç kümenin ağırlıkları	§3.6
`ENFA / Mahalanobis`	ENFA / Mahalanobis (Presence-Only)	ENFA arka planı	§3.7
`Study Area & Thinning`	Study Area & Thinning	üç kümenin alanı; varlık kayıtları	§3.8, §3.9

Alt sekme ipuçları `tabs/tab_data.py:214-219`'da tanımlıdır. Bölümde kullanılan ekran görüntüleri örnek koşunun (*Abies nordmanniana*) ayarlarını gösterir.

3.3 Yalancı yokluklar: ML Models



Şekil 3.1. ML Models alt sekmesi: PA repetitions 2, Pseudo-absences per repetition 1000, Sampling strategy random. Min. distance from presences, Environmental envelope (env_const), Envelope tail trim ve k-means classes alanları random stratejisinde kullanılmadığı için gri.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`PA repetitions:`	Bağımsız yalancı yokluk kümesi sayısı	2	1 – 10	`tabs/tab_data.py:293-297`
`Pseudo-absences per repetition:`	Her kümedeki yalancı yokluk sayısı	1000	100 – 100 000 (adım 500)	`tabs/tab_data.py:299-304`
`Sampling strategy:`	Yalancı yoklukların hangi hücrelerden çekileceği	`random`	`random`, `disk`, `sre`, `kmeans`, `env_const`, `geo_const`, `geo_env_const`, `geo_env_km_const`	`tabs/tab_data.py:306-326`, `sdm_sampling.py:47-53`
`Min. distance from presences:`	Varlıklara en az uzaklık (km)	10 km	0 – 500 km	`tabs/tab_data.py:328-333`
`Environmental envelope (env_const):`	Zarfın kuruluşu	`pca`	`pca`, `bioclim`	`tabs/tab_data.py:335-342`
`Envelope tail trim:`	Zarfın her iki ucundan kırpılan pay	0.05	0.00 – 0.45	`tabs/tab_data.py:343-348`
`k-means classes:`	Çevre uzayındaki küme sayısı	5	2 – 30	`tabs/tab_data.py:349-353`

Alanlar yalnız onları kullanan stratejide etkindir (`tabs/tab_data.py:557-573`): en az mesafe `disk` ve `geo_*`; zarf ve uç kırpma `env_const`, `geo_env_const`, `geo_env_km_const`; k-ortalamalar `kmeans`, `geo_env_km_const`. `Own Absences`'ta `replace` seçiliyse bütün alanlar gri olur (§3.4).

3.3.1 PA repetitions (tekrar sayısı)

- **Ne işe yarar.** Aynı varlıklarla, her biri ayrı çekilmiş `n` yalancı yoklukla `PA repetitions` kadar eğitim kümesi kurulur (`sdm_core.py:3035-3082`). Kümelerin varlık tarafı aynıdır, yalnız yokluk tarafı değişir. ③ Models her küme için ayrı model kurar (günlükte `PA1`, `PA2`) ve topluluk her kümeden gelen modelleri birleştirir (Bölüm 5–6).
- **Ne zaman, nasıl ayarlanır.** Arayüz ipucu 2–5'i önerir. Az yalancı yoklukla çalışılıyorsa (ör. varlık sayısı kadar) tekrar sayısı artırılmalıdır; Barbet-Massin vd. (2012) sınıflandırma ağaçlarında varlık sayısı kadar yalancı yoklukla 10 tekrarın ortalamasını önerir.
- **Sonuca etkisi.** Tekrar, tek bir çekilişin şansına bağlı sonucu ortalamaya dönüştürür; kümeler arasındaki fark yalancı yokluk seçiminden doğan belirsizliği gösterir. Süre kabaca tekrar sayısıyla doğru orantılı artar.
- **Dikkat.** `Own Absences` `replace` kipinde tekrar sayısı 1'e indirilir, çünkü tekrarlanacak bir çekiliş kalmaz (`sdm_core.py:3031-3033`); rapor da 1 yazar.

3.3.2 Pseudo-absences per repetition (sayı) ve prevalans

- **Ne işe yarar.** Her kümede çekilecek yalancı yokluk sayısıdır. Hücreler yinelemesiz çekilir (`rng.choice(..., replace=False)`, `sdm_core.py:2301-2302`); aday hücre sayısı istenen sayıdan azsa hepsi alınır ve uyarı yazılır.
- **Ne zaman, nasıl ayarlanır.** İpucu 1000–10 000 aralığını ve varlık sayısından az olmamasını önerir. Barbet-Massin vd. (2012) regresyon yöntemleri (GLM, GAM) için büyük sayıyı (ör. 10 000) ve varlık ile yokluğun eşit ağırlıklandırılmasını, ağaç yöntemleri için varlık sayısı kadar yalancı yoklukla çok tekrarı önermiştir. HABITUS'ta sınıf ağırlıklandırması ③ Models'ta ayrıca ayarlanır (Bölüm 5).
- **Sonuca etkisi. Prevalans** (eğitim kümesinde varlıkların payı) bu sayıyla belirlenir: $p = n_{\text{varlık}}/(n_{\text{varlık}} + n_{\text{yokluk}})$ (`sdm_core.py:3087-3089`). Örnek koşuda $143/(143 + 1000) = 0{,}125$'tir. Prevalans ağırlıklandırılmamış modellerin ham çıktısının düzeyini, dolayısıyla eşik değerlerini doğrudan değiştirir: yalancı yokluk sayısını artırmak ham olasılıkları düşürür. AUC eşikten bağımsızdır ama yalancı yoklukların nereden geldiğine duyarlıdır (§3.3.3, §3.8).
- **Dikkat.** Yalancı yokluk sayısı erişilebilir alandaki geçerli hücre sayısını aşarsa alandaki bütün hücreler kullanılır ve "The accessible area holds N usable cells, fewer than the M points requested" uyarısı yazılır (`sdm_core.py:2190-2203`). Örnekleme sonrası bir öngörücüde NoData'ya düşen nokta atılır ve sayılır ("PA1: N sampled point(s) fell on NoData…", `sdm_core.py:2816-2841`); kategorik katmanların NoData hücreleri örneklemeden baştan çıkarıldığı için bu durum seyrektir.

3.3.3 Sampling strategy (sekiz strateji)

Stratejilerin hepsi önce aynı aday kümesinden başlar: bütün sürekli ve kategorik katmanlarda değeri olan hücreler (`_valid_cell_mask`, `sdm_core.py:2309-2330`), erişilebilir alan tanımlıysa onun içinde kalanlar (`sdm_core.py:2181-2203`). Ardından stratejiye özgü kısıt uygulanır (`sdm_core.py:2205-2297`), varsa yanlılık ve sınıf ağırlıklarıyla çekiliş yapılır.

Strateji	Ne yapar	Kullandığı alanlar	R'deki karşılığı	Kod
`random`	Aday hücrelerden eşit olasılıkla çekiliş	—	biomod2 `random`, flexsdm `random`	`sdm_core.py:2291-2305`
`disk`	Her varlığa `Min. distance` içindeki hücreleri dışlar, kalanlardan rastgele	en az mesafe	biomod2 `disk` (yalnız en az mesafe)	`sdm_core.py:2243-2258`
`sre`	Varlıkların biyoiklim zarfının **dışındaki** hücrelerden çeker; zarf her değişkenin varlıklardaki %5–%95 aralığı	(sabit 0,05)	biomod2 `sre`	`sdm_core.py:2079-2119`, `3005-3022`
`kmeans`	Ölçeklenmiş çevre uzayında k-ortalamalar; varlık içermeyen sınıflardan, her sınıfa eşit toplam ağırlıkla çeker	k	sdm `eRandom`, flexsdm `kmeans`	`sdm_sampling.py:245-288`
`env_const`	Varlıkların çevresel zarfının dışındaki hücrelerden çeker	zarf, uç kırpma	flexsdm `env_const`	`sdm_sampling.py:219-242`
`geo_const`	`disk` ile aynı kural	en az mesafe	flexsdm `geo_const`	`sdm_sampling.py:26-27`
`geo_env_const`	En az mesafe **ve** zarf dışı	en az mesafe, zarf, uç kırpma	flexsdm `geo_env_const`	`sdm_core.py:2997-3003`
`geo_env_km_const`	En az mesafe, zarf dışı ve varlıksız k-ortalama sınıfları	hepsi	flexsdm `geo_env_km_const`	`sdm_core.py:2399-2429`

- **Ne işe yarar.** Strateji, modelin varlıkları "neyle" karşılaştıracağını belirler. `random` erişilebilir alanın çevresel çeşitliliğini olduğu gibi temsil eder. Öteki stratejiler yalancı yoklukları varlıklardan coğrafi (`disk`, `geo_const`) ya da çevresel (`sre`, `env_const`, `kmeans`) olarak uzaklaştırır; amaç, aslında uygun olan ama kayıt bulunmayan hücrelerin "yok" etiketi almasını (yanlış yokluk) azaltmaktır.
- **Ne zaman, nasıl ayarlanır.** Barbet-Massin vd. (2012) regresyon yöntemleri için rastgele seçimi, sınıflandırma ağaçları ve diskriminant yöntemleri için coğrafi ve çevresel olarak uzaktan seçimi önermiştir. Varsayılan `random`, erişilebilir alanla birlikte kullanıldığında en az varsayım içeren seçenektir. Yoğun örneklenmiş bir türde kayıtların çevresinde yanlış yokluk riski yüksekse `disk` (ör. 10–50 km) makul bir ara yoldur.
- **Sonuca etkisi.** Yalancı yoklukları varlıklardan uzaklaştıran her kısıt ayırmayı kolaylaştırır: AUC ve TSS **yükselir**, ama bu yükselişin bir kısmı modelin değil örnekleme tasarımının ürünüdür; tepki eğrileri varlık aralığının dışında kesilir, haritada uygun alan genişler (VanDerWal vd. 2009; Barbet-Massin vd. 2012). Zarf dışı stratejiler (`sre`, `env_const`) bunu en güçlü biçimde yapar: model zarfın içini "uygun", dışını "uygun değil" diye öğrenmeye yönelir ve zarfı kendisi yeniden üretir.
- **Dikkat.** Zarf ve k-ortalama stratejileri, ① Data'da yüklenen **bütün** sürekli katmanlarla hesaplanır; ② Variables'taki seçim daha sonra yapıldığı için seçilmeyen katmanlar da yalancı yoklukların yerini etkiler. `sre`'de bir hücre zarfın içinde sayılmak için **bütün** değişkenlerde aralıkta olmalıdır; 19 katmanla

neredeyse her hücre en az bir değişkende dışarıda kalır ve kısıt zayıflar. Zarf dışında hiç hücre kalmazsa strateji rastgele çekilişe döner ve uyarı yazılır ("no cells outside the envelope found. Falling back to random sampling."; `sdm_core.py:2206-2212`). `random`, `disk` ve `sre` dışındaki stratejilerde aday hücre istenen sayıdan azsa daha az nokta çekilir ("leaves only N candidate cells"; `sdm_core.py:2294-2297`).

3.3.4 Min. distance from presences

- **Ne işe yarar.** `disk`, `geo_const`, `geo_env_const` ve `geo_env_km_const`'ta her aday hücrenin merkezinin her varlığa büyük daire (haversine) uzaklığı hesaplanır; uzaklığı bu değerden büyük olmayan hücreler dışlanır (`sdm_core.py:2243-2258`). Projeksiyonlu ızgarada koordinatlar önce boylam/enleme çevrilir.
- **Ne zaman, nasıl ayarlanır.** Kayıtların konum hatası ve türün yerel yayılımı kadar bir mesafe (birkaç km ile birkaç on km) tipik seçimdir. Barbet-Massin vd. (2012) sınıflandırma yöntemleri için yaklaşık 2° uzaklığı denemiştir; bu, küçük bir çalışma alanında adayların çoğunu siler.
- **Sonuca etkisi.** Mesafe büyüdükçe yalancı yokluklar varlıklardan ve onların çevresel koşullarından uzaklaşır; AUC/TSS yükselir, model daha "geniş" bir uygun alan çizer.
- **Dikkat.** Mesafe 0 iken `geo_*` stratejileri hiçbir şeyi dışlamaz ve bunu uyarı olarak bildirir (`sdm_core.py:2999-3003`). Çok büyük mesafe "No background cells remain after min_dist=… km. Reduce the minimum distance." hatasıyla durur. Hesap her varlık için bütün aday hücreleri tarar; dünya ölçekli ızgarada ve binlerce kayıtta uzun sürebilir.

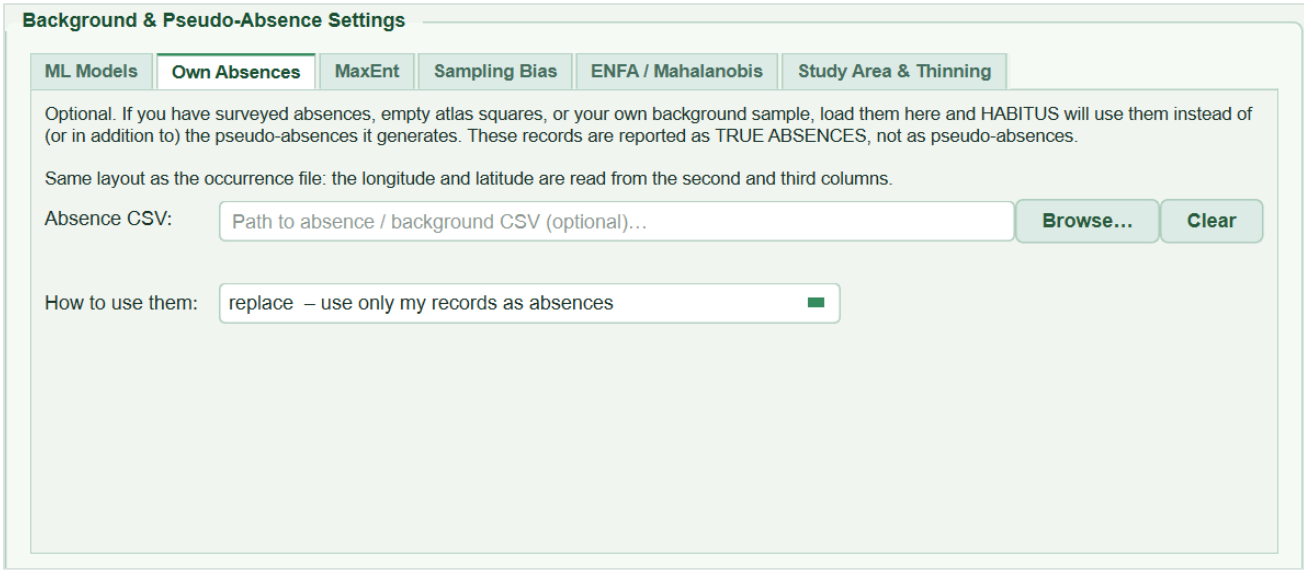
3.3.5 Environmental envelope ve Envelope tail trim

- **Ne işe yarar.** `env_const` ailesinde varlıkların çevresel zarfı iki yoldan biriyle kurulur (`sdm_core.py:2407-2420`):
  - `pca`: bütün geçerli hücrelerdeki standartlaştırılmış sürekli değişkenler üzerinde temel bileşenler (SVD) hesaplanır; ilk iki bileşende varlıkların $[q,\ 1-q]$ nicelik aralığı zarfı oluşturur. Bir hücre, iki bileşenden en az birinde bu aralığın dışındaysa zarf dışıdır (`sdm_sampling.py:219-242`).
  - `bioclim`: her değişken ayrı ayrı; hücre bütün değişkenlerde $[q,\ 1-q]$ aralığındaysa zarf içidir (`sre` ile aynı hesap, fakat q ayarlanabilir; `sdm_core.py:2410-2412`).
  - `Envelope tail trim` q'dur: 0 tam en küçük-en büyük aralığı, 0,05 ise %5–%95 aralığını verir.
- **Ne zaman, nasıl ayarlanır.** flexsdm'in `env_const` yöntemine en yakın olan `pca`'dır. Kayıtlarda konum hatası ya da aykırı değer şüphesi varsa q büyütülür (zarf daralır).
- **Sonuca etkisi.** q büyüdükçe zarf daralır, zarf dışı aday alan büyür ve yalancı yokluklar varlık koşullarına yaklaşır; q küçüldükçe zarf genişler ve yalancı yokluklar varlıklardan çevresel olarak uzaklaşır, ayırma kolaylaşır.
- **Dikkat.** Temel bileşenler erişilebilir alanla sınırlı değil, bütün geçerli hücrelerle kurulur; zarfın kendisi sonra erişilebilir alanla kesişir. Zarf dışı hücrelerin payı çalıştırma kaydına yazılır (`frac_cells_outside_envelope`). `sre` stratejisi bu iki alanı kullanmaz; payı her zaman 0,05'tir.

3.3.6 k-means classes

- **Ne işe yarar.** Geçerli hücreler standartlaştırılmış çevre uzayında `k` sınıfa ayrılır (scikit-learn `KMeans`, `n_init=3`; 50 000'den çok hücre varsa model 50 000 hücrelik rastgele alt örnekle kurulur, bütün hücreler sonra sınıflanır). Varlıklar en yakın sınıf merkezine atanır. Varlık içermeyen sınıflar yalancı yokluk kaynağıdır ve her biri aynı toplam ağırlığı alır (katmanlı çekiliş). Bütün sınıflarda varlık varsa sınıf ağırlığı $1/(1 + n_{\text{varlık}})$ olur ve uyarı yazılır (`sdm_sampling.py:245-288`).
- **Ne zaman, nasıl ayarlanır.** Varsayılan 5. Çok çeşitli bir çalışma alanında sınıf sayısını artırmak varlıksız sınıf bulmayı kolaylaştırır.
- **Sonuca etkisi.** Eşit sınıf ağırlığı, küçük ama çevresel olarak farklı sınıfların yalancı yokluklarda fazla temsil edilmesine yol açar; çevresel çeşitlilik dengelenir, ama coğrafi dağılım değişir.
- **Dikkat.** Sınıflar bütün geçerli hücrelerle kurulur; varlıksız sınıfların hepsi erişilebilir alanın dışında kalırsa çalıştırma "No candidate cell has a positive sampling weight" iletisiyle durur (`sdm_core.py:2281-2285`). k-ortalamaların tohumu çalıştırma tohumundan türetilir.

3.4 Kendi yokluk kayıtlarınız: Own Absences



Şekil 3.2. Own Absences alt sekmesi: açıklama metni, boş Absence CSV alanı (Browse… ve Clear) ve varsayılan kip "replace – use only my records as absences".

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Absence CSV:` + `Browse…` / `Clear`	Gözlenmiş yokluk (ya da kendi arka plan) kayıtları; seçilince kayıt sayısı ve okunacak sütunlar önizlenir	boş	CSV	`tabs/tab_data.py:478-491`, `1162-1181`
`How to use them:`	Kayıtların yalancı yoklukların yerine mi yanına mı konacağı	`replace`	`replace – use only my records as absences` / `augment – my records plus generated pseudo-absences`	`tabs/tab_data.py:493-507`

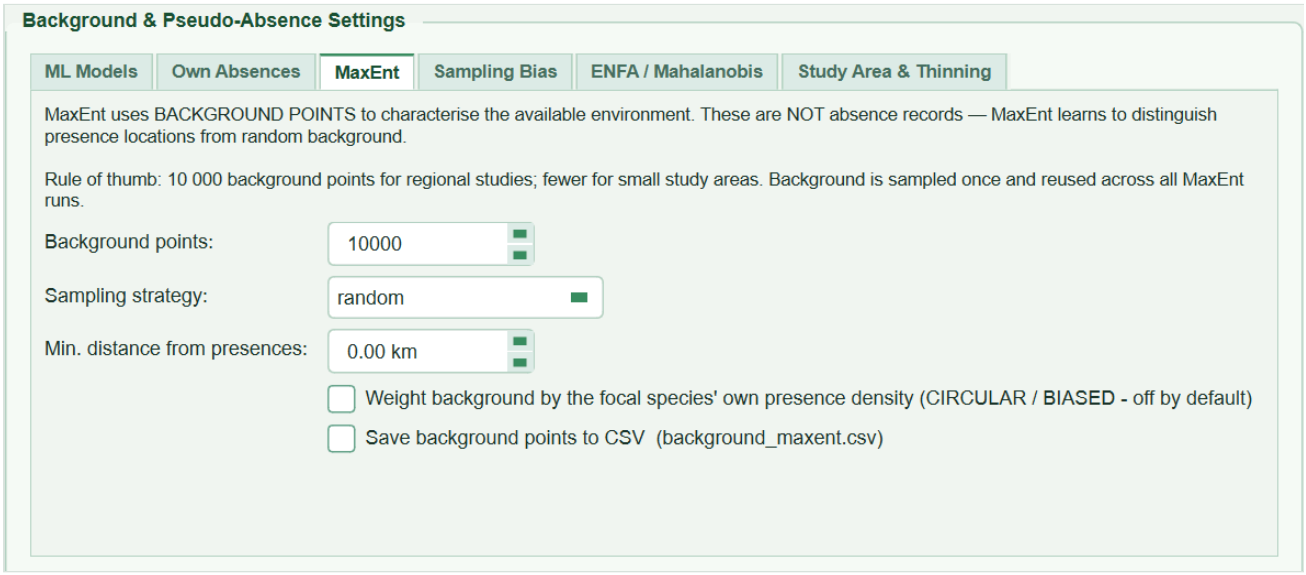
3.4.1 Absence CSV ve sütunlar

- **Ne işe yarar.** Dosya, kayıt dosyasıyla aynı okuyucuyla okunur (ayırıcı ve ondalık sezgisi dahil). Koordinat sütunları şu sırayla aranır (`sdm_core.py:2729-2754`): kayıt dosyasındaki boylam/enlem sütun adları bu dosyada da varsa onlar; yoksa adla bulunan `longitude/lat…` sütunları; yoksa 2. ve 3. sütun; yalnız iki sütun varsa 1. ve 2. sütun. Boş koordinatlı satırlar sayılarak atılır; raster dışı ya da NoData'ya düşenler "absence record outside extent or on NoData" gerekçesiyle atılır ve kayıp %20'yi aşarsa varlıklardaki gibi açıklayıcı uyarı verilir (`sdm_core.py:2770-2797`).
- **Ne zaman, nasıl ayarlanır.** Alan taramasında türün bulunmadığı noktalar, boş atlas kareleri ya da aynı yöntemle derlenmiş bir hedef grup arka planı için. Bu kayıtlar raporda ve özette **"TRUE absences"** diye, yalancı yokluktan ayrı yazılır.
- **Sonuca etkisi.** Gerçek yoklukla eğitilen model, "türün bulunmadığı yer" bilgisini tahminle değil gözlemle alır; değerlendirme istatistiklerinin anlamı da değişir (yalancı yokluğa göre değil gerçek yokluğa göre ayırma).
- **Dikkat.** Yokluk kayıtları hücre başına bire indirilmez, seyreltilmez ve bir varlıkla aynı hücreye düşüp düşmedikleri denetlenmez. Dosya verilmiş ama kullanılamıyorsa program sessizce yalancı yokluğa dönmez, durur ("All N user-supplied absence records fall outside the raster extent…", `sdm_core.py:2791-2797`).

3.4.2 replace ve augment kipleri

- **Ne işe yarar.**
  - `replace`: Yokluk tarafı yalnız sizin kayıtlarınızdır. Yalancı yokluk üretilmez; `ML Models`'taki bütün alanlar gri olur, tekrar sayısı 1'e iner, en az mesafe ve erişilebilir alan kayıtlarınıza uygulanmaz (`tabs/tab_data.py:535-546`, `sdm_core.py:3031-3040`).
  - `augment`: Her kümede kayıtlarınız olduğu gibi yer alır, üstüne `Pseudo-absences per repetition` kadar yalancı yokluk çekilir (`sdm_core.py:3060-3062`). Kümeler arasındaki fark yalnız üretilen kısımdan gelir.
- **Ne zaman, nasıl ayarlanır.** Kayıtlar gerçek ve kapsamlı bir yokluk taramasından geliyorsa `replace`; az sayıdaysa ya da tam bir tarama değil kısmi bir arka plansa `augment`.
- **Sonuca etkisi.** `replace`'te prevalans sizin varlık/yokluk oranınızdır. `augment`'te yokluk tarafı büyür, prevalans düşer.
- **Dikkat.** `replace` kipinde 20'den az yokluk kaydı varsa "Only N true absence records were supplied and pseudo-absences were switched off…" uyarısı verilir ve `augment` önerilir (`sdm_core.py:2807-2813`). Yokluk kayıtları yalnız iki sınıflı algoritmaların eğitim kümesine girer: MaxEnt kendi arka planını, ENFA kendi arka planını kullanmaya devam eder (§3.5, §3.7).

3.5 MaxEnt arka planı: MaxEnt



Şekil 3.3. MaxEnt alt sekmesi: Background points 10000, Sampling strategy random, Min. distance from presences 0.00 km; odak türün kendi kayıt yoğunluğuyla ağırlıklandırma (CIRCULAR / BIASED) ve arka planı CSV'ye kaydetme kutuları işaretsiz.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Background points:`	MaxEnt arka plan noktası sayısı	10000	100 – 200 000 (adım 1000)	`tabs/tab_data.py:789-794`
`Sampling strategy:`	Çekiliş kuralı	`random`	`random`, `disk`	`tabs/tab_data.py:796-801`
`Min. distance from presences:`	`disk`'te varlıklara en az uzaklık	0 km	0 – 500 km	`tabs/tab_data.py:803-806`
`Weight background by the focal species' own presence density (CIRCULAR / BIASED - off by default)`	Arka planı odak türün kayıt yoğunluğuyla ağırlıklandırır	kapalı	açık / kapalı	`tabs/tab_data.py:808-822`, `sdm_core.py:3102-3142`
`Save background points to CSV (background_maxent.csv)`	Arka plan koordinatlarını ve öngörücü değerlerini yazar	kapalı	açık / kapalı	`tabs/tab_data.py:824-830`, `1549-1563`

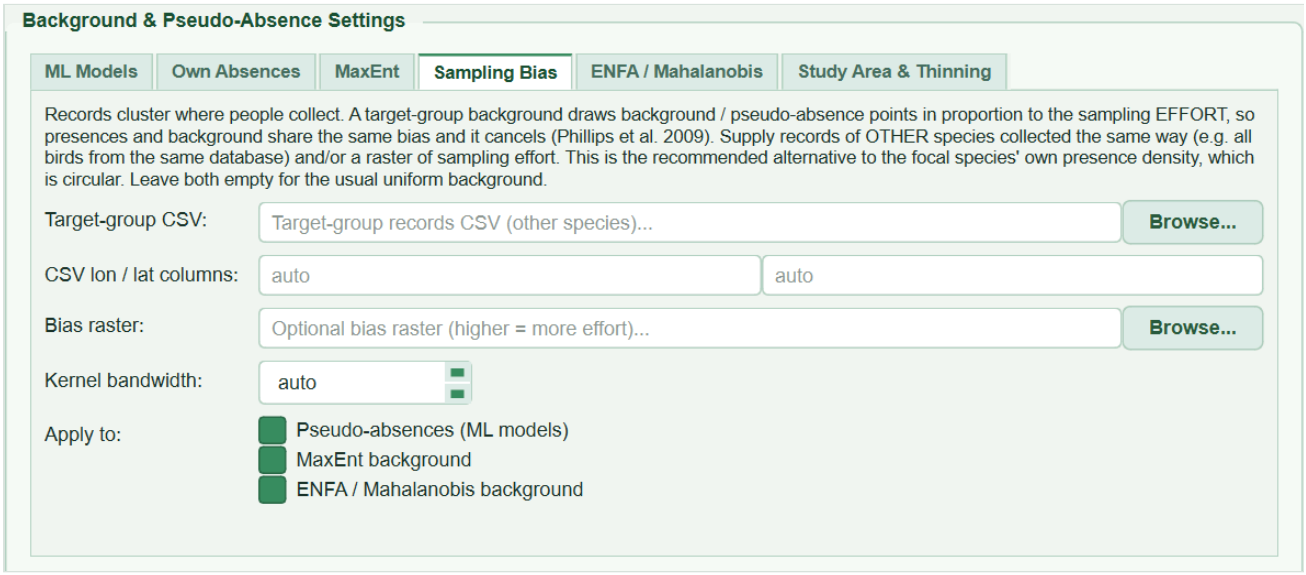
3.5.1 Background points ve strateji

- **Ne işe yarar.** Arka plan, yalancı yokluklardan ayrı ve bir kez çekilir (`sdm_core.py:3144-3166`); erişilebilir alan ve (seçildiyse) hedef grup yanlılığı uygulanır. MaxEnt her uydurmada, çapraz doğrulama katlarında da, bu arka planın tamamını kullanır (`sdm_core.py:4595-4603`). `disk` yalnız `Min. distance` 0'dan büyükse etkilidir; 0 iken `random` ile aynıdır.
- **Ne zaman, nasıl ayarlanır.** 10 000, Maxent'in yaygın varsayılanıdır (Phillips vd. 2006). Küçük bir erişilebilir alanda geçerli hücre sayısı bundan azsa bütün hücreler kullanılır; arka plan o zaman alanın tam bir sayımıdır (Phillips ve Dudík 2008).
- **Sonuca etkisi.** Arka planın büyüklüğü, alanın çevresel dağılımının ne kadar iyi temsil edildiğini belirler; 10 000'in üstünde kazanç genellikle küçüktür, süre artar. Arka planın **yeri** (erişilebilir alan) sayısından çok daha etkilidir (§3.8).
- **Dikkat.** Arka plan varlık hücrelerini dışlamaz; Maxent'in tanımındaki gibi varlıklar da kullanılabilir çevrenin bir parçasıdır. Özette "Background (MaxEnt): N points" satırı NoData nedeniyle atılanlar düşüldükten sonraki sayıyı verir.

3.5.2 Odak türün kendi yoğunluğuyla ağırlıklandırma (dairesel)

- **Ne işe yarar.** Açıkken her geçerli hücreye, bütün varlıklara olan uzaklıklarından hesaplanan Gauss çekirdek toplamı (bant genişliği 0,5 birim) ağırlık olarak verilir ve arka plan bu ağırlıkla çekilir (`sdm_core.py:3119-3136`). Arka plan böylece varlıkların yoğun olduğu yerlere yığılır.
- **Ne zaman, nasıl ayarlanır.** Önerilmez; kutunun adı ve ipucu bunu açıkça söyler. Odak türün kendi kayıtları hem yanlılığı hem türün gerçek dağılımını taşır; ağırlık bu ikisini ayıramaz ve model, en iyi örneklenmiş alanların çevresini yok saymayı öğrenir. Geçerli bir düzeltme bağımsız bir yanlılık yüzeyi ister (Phillips vd. 2009): `Sampling Bias` alt sekmesi.
- **Sonuca etkisi.** Varlık ile arka plan arasındaki karşıtlık azalır; AUC düşer, tepki eğrileri düzleşir.
- **Dikkat.** Açıldığında her çalıştırmada "MaxEnt background weighted by the focal species' own presence density: this is circular and biased…" uyarısı günlüğe yazılır. `Sampling Bias`'ta bir yüzey verilmiş ve MaxEnt'e uygulanıyorsa bu kutu yok sayılır (`sdm_core.py:3104-3107`). Bant genişliği raster koordinat birimindedir (coğrafi ızgarada 0,5°, yaklaşık 55 km).

3.6 Örnekleme yanlılığı: Sampling Bias



Şekil 3.4. Sampling Bias alt sekmesi: Target-group CSV ve Bias raster alanları boş, CSV lon / lat columns "auto", Kernel bandwidth "auto"; Apply to altında Pseudo-absences (ML models), MaxEnt background ve ENFA / Mahalanobis background işaretli.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Target-group CSV:`	Aynı yöntemle toplanmış **başka** türlerin kayıtları	boş (kapalı)	CSV	`tabs/tab_data.py:402-405`
`CSV lon / lat columns:`	Hedef grup CSV'sindeki koordinat sütunları	`auto`	sütun adı	`tabs/tab_data.py:407-409`, `sdm_sampling.py:77-93`
`Bias raster:`	Örnekleme çabası rasteri (büyük değer = çok çaba)	boş (kapalı)	`.tif .tiff .img .asc`	`tabs/tab_data.py:411-416`, `sdm_sampling.py:161-198`
`Kernel bandwidth:`	Hedef grup yoğunluğunun Gauss düzleştirmesi (sigma)	`auto` (0)	0 – 5000 km	`tabs/tab_data.py:418-423`, `sdm_sampling.py:103-158`
`Apply to:`	Yüzeyin uygulanacağı kümeler	üçü de işaretli	`Pseudo-absences (ML models)`, `MaxEnt background`, `ENFA / Mahalanobis background`	`tabs/tab_data.py:425-441`, `sdm_core.py:2991-2993`

3.6.1 Hedef grup yüzeyi ve yanlılık rasteri

- **Ne işe yarar.** Kayıtlar insanların topladığı yerlerde (yol, köy, istasyon çevresi) yoğunlaşır. Hedef grup yaklaşımı (Phillips vd. 2009), arka plan ve yalancı yoklukları aynı çabanın izini taşıyan başka türlerin kayıt yoğunluğuyla orantılı çeker; varlık ile arka plan aynı yanlılığı taşıdığı için yanlılık birbirini götürür. HABITUS iki kaynaktan bir ağırlık ızgarası kurar (`sdm_core.py:2332-2397`):
  - **Hedef grup CSV'si:** kayıtlar hücrelere sayılır ve Gauss filtresiyle düzleştirilir; ızgara en büyük değere bölünür, sıfır yoğunluklu hücrelere 10^{-6} taban ağırlık bırakılır (`sdm_sampling.py:103-158`).
  - **Yanlılık rasteri:** aynı ızgaradaysa doğrudan okunur, değilse çift doğrusal (bilinear) yöntemle yeniden örneklenir; NaN, NoData ve negatif değerler 0 olur, en büyük değere bölünür (`sdm_sampling.py:161-198`).
  - İkisi birlikte verilirse çarpımları kullanılır. Çekiliş olasılığı hücre ağırlığıyla orantılıdır; ağırlık erişilebilir alan ve strateji kısıtlarıyla birleşir (k-ortalama sınıf ağırlığıyla çarpılır; `sdm_core.py:2272-2289`).
- **Ne zaman, nasıl ayarlanır.** Örnekleme çabası açıkça dengesizse. Hedef grup, odak türle aynı veri tabanından ve aynı yöntemle toplanmış türler olmalıdır (ör. aynı herbaryumun bütün odunsu bitki kayıtları). Odak türün kendi kayıtları hedef gruba konmamalıdır; bu dairesel olur.
- **Sonuca etkisi.** Arka plan erişilebilir yerlere kayar; model "kolay ulaşılan" ile "uygun"u daha az karıştırır. Tipik olarak AUC düşer (ayrım zorlaşır), ama harita yollar ve yerleşimler yerine çevresel koşulları izler (Kramer-Schadt vd. 2013).
- **Dikkat.** Hedef grup koordinatları rasterle aynı koordinat sisteminde olmalıdır; hiçbiri ızgaraya düşmezse çalıştırma durur. 20'den az hedef grup kaydında yoğunluğun çok kaba olacağı, yüzeyin hücrelerin %0,1'inden azında toplandığında arka planın çoğunun oradan geleceği uyarısı yazılır (`sdm_core.py:2352-2393`). Hatalı dosya çalıştırmayı durdurur; program sessizce düzgün örneklemeye dönmez. Yanlılık rasteri koordinat sistemi bildirmiyorsa öngörücülerle aynı sistemde olduğu varsayılır ve uyarılır; ızgarası farklı ve CRS'si yoksa hizalanamaz ve hata verir. Hiçbir yüzey verilmediğinde rapora "Sampling bias: not corrected" notu düşülür (`sdm_core.py:1358-1363`, `2988-2990`).

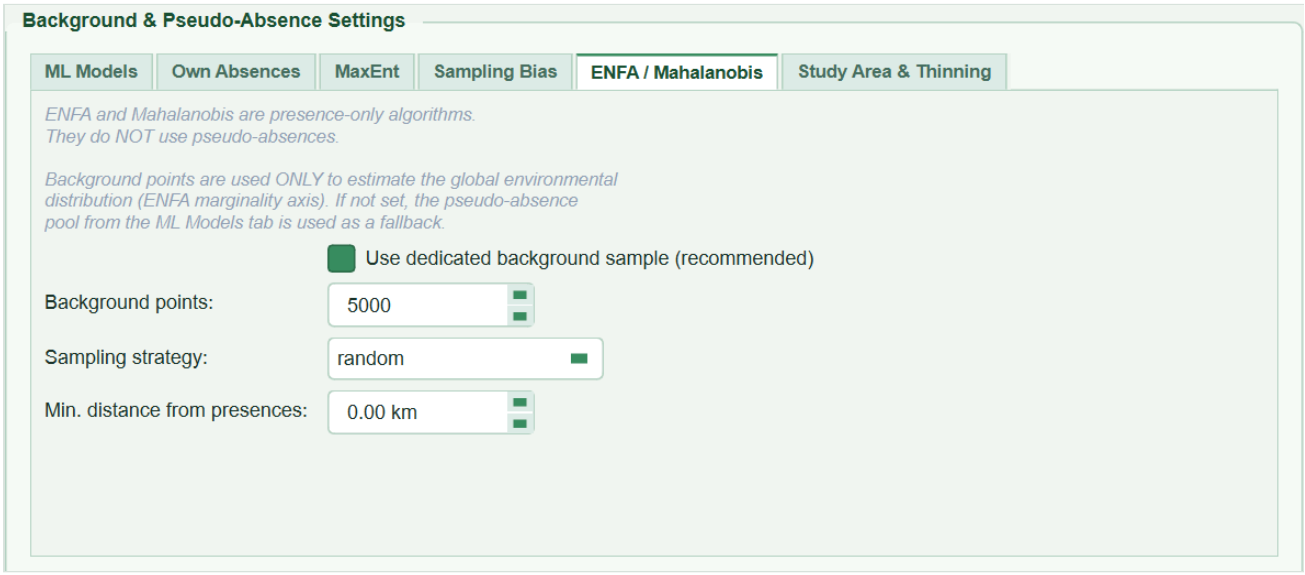
3.6.2 Kernel bandwidth

- **Ne işe yarar.** Gauss düzleştirmesinin standart sapmasıdır (km). Coğrafi ızgarada km dereceye çevrilir; boylam yönünde kayıtların ortalama enleminin kosinüsüyle düzeltilir. Projeksiyonlu ızgarada km, harita biriminin metre karşılığıyla çevrilir. `auto` (0) iken her eksen için normal başvuru kuralı kullanılır: $h = 1{,}06\ \min(s,\ \mathrm{IQR}/1{,}34)\ n^{-1/5}$ (MASS `bandwidth.nrd`/4, `kde2d`'deki gibi). Sigma hiçbir zaman bir hücreden küçük olmaz (`sdm_sampling.py:136-154`).
- **Ne zaman, nasıl ayarlanır.** Hedef grup kayıtları seyrekse genişletilir; çaba yolların hemen çevresinde toplanıyorsa birkaç hücre düzeyinde tutulur.
- **Sonuca etkisi.** Küçük bant genişliği yüzeyi kayıt noktalarına yapıştırır, arka plan neredeyse yalnız kayıtlı hücrelerden gelir; büyük bant genişliği yüzeyi düzleştirir ve düzgün örneklemeye yaklaştırır.
- **Dikkat.** Kullanılan bant genişliği çalıştırma kaydına yazılır (`auto (nrd)` ya da km).

3.6.3 Apply to

Yüzey yalnız işaretli kümelere uygulanır (`sdm_core.py:2991-2993`). Üçünü birden işaretlemek, bütün algoritmaların aynı yanlılık düzeltmesini görmesini sağlar ve algoritmalar arası karşılaştırmayı tutarlı kılar. Yalnız birini işaretlemek, düzeltmenin etkisini tek bir algoritma ailesinde sınamak için yararlıdır.

3.7 ENFA arka planı: ENFA / Mahalanobis

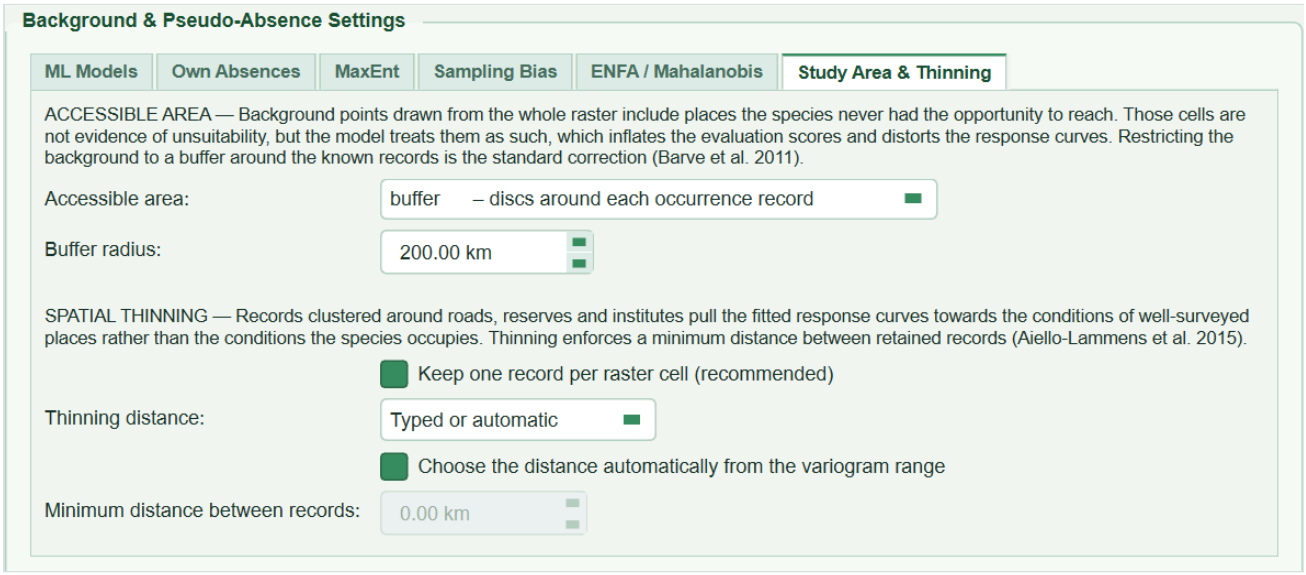


Şekil 3.5. ENFA / Mahalanobis alt sekmesi: Use dedicated background sample (recommended) işaretli, Background points 5000, Sampling strategy random, Min. distance from presences 0.00 km.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Use dedicated background sample (recommended)`	ENFA için ayrı arka plan çeker	açık	açık / kapalı	`tabs/tab_data.py:858-864`
`Background points:`	Arka plan nokta sayısı	5000	100 – 100 000 (adım 1000)	`tabs/tab_data.py:866-872`
`Sampling strategy:`	Çekiliş kuralı	`random`	`random`, `disk`	`tabs/tab_data.py:874-879`
`Min. distance from presences:`	`disk`'te en az uzaklık	0 km	0 – 500 km	`tabs/tab_data.py:881-885`

- **Ne işe yarar.** ENFA, türün koşullarını çalışma alanının "küresel" koşullarıyla karşılaştırır: küresel ortalama ve kovaryans bu arka plandan hesaplanır (`sdm_core.py:4079-4092`). Arka plan erişilebilir alan içinden ve (işaretliyse) yanlılık yüzeyiyle çekilir (`sdm_core.py:3176-3199`). Kutu kapalıysa ENFA küresel istatistikleri eğitim satırlarından (varlıklar + yalancı yokluklar) hesaplar.
- **Ne zaman, nasıl ayarlanır.** Açık bırakılması önerilir: küresel çevre, yalancı yokluk stratejisinin (ör. `disk`, `sre`) bozduğu bir örnekten değil, alanın düzgün bir örneğinden tahmin edilmelidir. İpucu 5000–10 000 noktayı yeterli bulur.
- **Sonuca etkisi.** ENFA'nın marjinallik (türün ortalamasının küresel ortalamadan sapması) ve özelleşme eksenleri bu örneğe göre ölçülür; erişilebilir alanı daraltmak marjinalliği küçültür.
- **Dikkat.** Kodda bu arka planı yalnız ENFA kullanır; Mahalanobis uzaklığı yöntemi yalnız varlıkların ortalamasını ve kovaryansını kullanır (`sdm_core.py:4207-4230`), bu sekmenin ayarları onu değiştirmez. Arka planın sütunları modeldeki değişkenlerle tam eşleşmezse ENFA eğitim satırlarına döner. Strateji kutusunda `disk` seçmek yalnız en az mesafeyi devreye sokar.

3.8 Erişilebilir alan: Study Area & Thinning (üst yarı)



Şekil 3.6. Study Area & Thinning alt sekmesi, örnek koşunun ayarlarıyla: Accessible area "buffer – discs around each occurrence record", Buffer radius 200.00 km; altta Keep one record per raster cell işaretli, Thinning distance "Typed or automatic", Choose the distance automatically from the variogram range işaretli (bu yüzden Minimum distance between records gri).

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Accessible area:`	Arka plan/yalancı yokluk alanının kuruluş yöntemi	`buffer`	`buffer – discs around each occurrence record`, `mcp_buffer – buffered minimum convex polygon`, `polygon – my own calibration area layer`, `none – sample the whole raster extent`	`tabs/tab_data.py:644-666`
`Buffer radius:`	Tampon yarıçapı; `polygon`'da çokgenlerin çevresine eklenen tampon	200 km	0 – 5000 km (adım 50)	`tabs/tab_data.py:668-674`, `609-618`
`Calibration layer:` + `Browse…`	Kalibrasyon çokgen katmanı (yalnız `polygon`'da görünür)	boş	`.shp .gpkg .geojson .json .kml .gml`	`tabs/tab_data.py:677-691`
`Keep only the polygons containing an occurrence record`	Yalnız kayıt içeren çokgenleri tutar (yalnız `polygon`'da)	açık	açık / kapalı	`tabs/tab_data.py:693-701`

3.8.1 Neden erişilebilir alan

Bir tür, uygun olduğu halde hiç ulaşamadığı yerlerde bulunmaz. Bu hücreler "uygun değil" kanıtı değildir; ama arka plan ya da yalancı yokluk olarak modele girdiklerinde model onları öyle öğrenir (Barve vd. 2011). Erişilebilir alan (BAM çerçevesindeki M), arka planın çekileceği alanı türün ulaşabileceği bölgeyle sınırlar. Kapsamı geniş tutmak, varlıklardan çok farklı koşullar içeren hücreleri arka plana katar; ayrım kolaylaşır ve AUC yapay olarak yükselir (Lobo vd. 2008; VanDerWal vd. 2009). Maske üç kümeye de (yalancı yokluk, MaxEnt, ENFA) uygulanır; kullanıcının yokluk kayıtlarına uygulanmaz. Harita yine bütün kapsam (ya da çalışma sınırı) için üretilir.

3.8.2 Accessible area yöntemleri

- **Ne işe yarar.**
  - `buffer`: Her varlığın hücresinden `Buffer radius` uzaklıktaki bütün hücreler (`sdm_spatial.py:943-1062`). Uzaklık, hücre boyunu km'ye çeviren anizotropik Öklid uzaklık dönüşümüyle (`distance_transform_edt`) hesaplanır; coğrafi ızgarada hücre genişliği kayıtların ortalama enleminin kosinüsüyle küçültülür.
  - `mcp_buffer`: Varlık hücrelerinin dışbükey zarfı (en küçük dışbükey çokgen) doldurulur, sonra aynı tampon eklenir. Kayıtlar bir doğru üzerindeyse (nehir, kıyı) zarf, en uzak iki kayıt arasındaki doğru parçasıdır.
  - `polygon`: Kullanıcının kalibrasyon katmanı (§3.8.4).
  - `none`: Kısıt yok; bütün geçerli hücreler.
- **Ne zaman, nasıl ayarlanır.** Rojas-Soto vd. (2024) yedi kalibrasyon alanı yöntemini 31 tür üzerinde karşılaştırmış ve biyocoğrafik birimleri (ekolojik bölge, havza, biyocoğrafik il) kullanan yöntemi önermiştir; HABITUS'taki karşılığı `polygon` + `Keep only the polygons containing an occurrence record`'dur. Tampon yöntemleri güvenli ve basittir ama engelleri (dağ sırası, deniz) temsil edemez. `mcp_buffer` kesintisiz yayılışlı türe uygundur; parçalı yayılışta zarf boşluğu da kapsar. `none` yalnız rasterler zaten ilgi alanına göre kesilmişse seçilmelidir.
- **Sonuca etkisi.** Alan daraldıkça arka plan varlık koşullarına benzer; AUC/TSS düşer ama daha dürüst olur, tepki eğrileri M içindeki gradyanı tanımlar. Örnek koşuda 200 km tampon raster kapsamının %40,0'ını tutmuştur.
- **Dikkat.** Tampon, seyreltilmiş kayıtlardan kurulur (`sdm_core.py:2971`). `buffer` ve `mcp_buffer`'da yarıçap 0 ise kısıt uygulanmaz. Tampon kapsamın %97'sinden fazlasını kapsıyorsa "The accessible-area buffer covers almost the whole raster extent, so it is not constraining anything." uyarısı verilir (`sdm_core.py:2694-2699`). Alanda istenen nokta sayısından az hücre varsa bütün hücreler kullanılır, uyarı yazılır ve çekiliş kapsamın tamamına genişletilmez (`sdm_core.py:2166-2203`). Maske kurulamazsa program bütün kapsama dönmez, durur.

3.8.3 Buffer radius

- **Ne işe yarar.** `buffer` ve `mcp_buffer`'da M'nin yarıçapı; `polygon`'da seçilen çokgenlerin çevresine eklenen tampon (`sdm_spatial.py:1260-1299`).
- **Ne zaman, nasıl ayarlanır.** Arayüz ipucunun dediği gibi türün yayılma yeteneğinden ve yayılışın yaşından seçilmelidir, raster boyutundan değil. Yayılımı sınırlı türlerde onlarca km, uzun mesafe yayılan türlerde yüzlerce km.
- **Sonuca etkisi.** Yarıçap büyüdükçe arka plan çevresel olarak çeşitlenir, AUC yükselir, prevalans değişmez (nokta sayısı aynıdır) ama yalancı yokluklar varlıklardan uzaklaşır.
- **Dikkat.** `polygon` kipine geçildiğinde alan varsayılan 200 km değerini korur; çokgenleri çizildiği gibi kullanmak için değeri **0'a** indirmek gerekir (ipucu: "Leave it at zero to use the polygon boundaries exactly as drawn"). Örnek koşu `buffer` 200 km ile çalıştırılmıştır.

3.8.4 Kalibrasyon katmanı ve "kayıt içerenler"

- **Ne işe yarar.** Katman pyogrio ile okunur; yalnız çokgenler kullanılır. Koordinat sistemi rasterlerden farklıysa yeniden projekte edilir; katman sistem bildirmiyorsa aynı olduğu varsayılır ve uyarılır (`sdm_spatial.py:1194-1225`). `Keep only the polygons containing an occurrence record` açıksa yalnız en az bir varlığın düştüğü çokgenler tutulur (biyocoğrafik birim yöntemi); kapalıysa katmandaki bütün çokgenler kullanılır. İsteğe bağlı tampon eklenir ve çokgenler ızgaraya `all_touched=True` ile (çokgene değen her hücre içeride) yakılır (`sdm_spatial.py:1234-1321`). Özette "Calibration area: K of N polygons… X% of the raster extent" yazılır.
- **Ne zaman, nasıl ayarlanır.** Bütün bir ekolojik bölge katmanını kesmeden yükleyip kutuyu açık bırakmak en pratik yoldur. Katman yalnız çalışma alanınızı içeriyorsa kutu kapatılır.
- **Sonuca etkisi.** M, biyocoğrafik sınırları izler; tampon yöntemlerinin aşamadığı engeller hesaba girer.
- **Dikkat.** Hiçbir çokgen kayıt içermiyorsa çalıştırma durur. Kutu kapalıyken hiçbir varlık çokgenlerin içinde değilse "Calibration polygon: NONE of the N presence records lies inside the polygon(s)…", varlıkların %80'inden azı içerdeyse "only K of N presence records… lie inside" uyarısı verilir (`sdm_core.py:2644-2672`). `polygon` seçilip katman verilmezse çalıştırma durur.

3.8.5 accessible_area.tif

Erişilebilir alan tanımlıysa maske türün çıktı klasörüne `accessible_area.tif` olarak yazılır (1 = içeride, 0 = dışarıda, sıkıştırılmış uint8; `tabs/tab_data.py:1536-1548`, `sdm_spatial.py:1065-1075`). ⑥ Range Change, değişimi bu alanın içinde ayrıca raporlamak için bu dosyayı kullanır (Bölüm 9). `none` seçiliyse dosya yazılmaz.

3.9 Kayıtların seyreltilmesi: Study Area & Thinning (alt yarı)

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Keep one record per raster cell (recommended)`	Her öngörücü hücresinde bir kayıt	açık	açık / kapalı	`tabs/tab_data.py:722-731` (Bölüm 2.8.3)
`Thinning distance:`	Seyreltme mesafesinin kaynağı	`Typed or automatic`	`Typed or automatic`, `One cell`, `10 km (Boria 2014)`	`tabs/tab_data.py:733-743`
`Choose the distance automatically from the variogram range`	Mesafeyi variogram aralığından seçer (yalnız `Typed or automatic`'te)	kapalı	açık / kapalı	`tabs/tab_data.py:745-751`
`Minimum distance between records:`	Elle girilen mesafe (yalnız `Typed or automatic` ve otomatik kapalıyken)	0 km (seyreltme yok)	0 – 500 km	`tabs/tab_data.py:753-765`

3.9.1 Seyreltmenin işleyişi

- **Ne işe yarar.** Yollar, koruma alanları ve araştırma kurumları çevresinde kümelenmiş kayıtlar, tepki eğrilerini türün yaşadığı koşullardan çok iyi örneklenmiş yerlerin koşullarına çeker (Aiello-Lammens vd. 2015). Seyreltme, tutulan iki kayıt arasında en az bir mesafe zorunlu kılar. Algoritma açgözlüdür: kayıtlar tohumlu rastgele bir sırayla gezilir, tutulmuş bir kayda mesafeden daha yakın olan atılır; üç farklı sırayla denenir ve en çok kayıt tutan sonuç seçilir (`sdm_spatial.py:888-937`, `sdm_core.py:2536`). Mesafeler büyük daire mesafesidir.
- **Ne zaman, nasıl ayarlanır.** Seyreltme, hücre başına bir kuraldan sonra uygulanır (Bölüm 2.8.1). Örnekleme açıkça kümeliyse açılır; kayıtlar zaten seyrek ve düzenliyse kapalı kalabilir.
- **Sonuca etkisi.** Kayıt sayısı düşer, prevalans düşer, kalan kayıtlar coğrafi olarak daha düzgün dağılır. Boria vd. (2014) seyreltmenin aşırı uyumu azaltıp bağımsız değerlendirmede performansı artırdığını göstermiştir. Atılan kayıtlar "spatially thinned" gerekçesiyle listelenir; kullanılan mesafe ve nasıl seçildiği günlüğe, özete, rapora ve çalıştırma kaydına (`thinning.how_chosen`) yazılır.
- **Dikkat.** spThin varsayılan olarak çok sayıda rastgele tekrardan en büyük kümeyi seçer; HABITUS'un 3 tekrarı bu sonuca yaklaşır ama en büyük kümeyi garanti etmez. Seyreltme sonrası 10'dan az kayıt kalırsa uyarı verilir. Elle girilen mesafe bir hücreden küçükse bunun yalnız komşu hücrelerdeki kayıtları ayırdığı bildirilir (`sdm_core.py:2526-2534`).

3.9.2 Ön ayarlar: One cell ve 10 km

- **Ne işe yarar.** `One cell`, mesafeyi bir öngörücü hücresinin kenarına eşitler: coğrafi ızgarada hücre yüksekliği × 110,57 km ile hücre genişliği × 111,32 km × cos(ortalama enlem) değerlerinin ortalaması (`sdm_core.py:2554-2569`). `10 km`, Boria vd. (2014)'ün kullandığı ve çok kopyalanan sabit mesafedir (`sdm_core.py:2478-2483`). Ön ayar seçilince otomatik kutu ve elle girilen mesafe gri olur.
- **Ne zaman, nasıl ayarlanır.** `One cell`, komşu hücrelerdeki kayıtları ayırmak için öngörücü çözünürlüğüne bağlı en küçük anlamlı mesafedir; 2,5′ verisinde yaklaşık 4 km'dir. `10 km` bir gelenektir, her tür ve çözünürlük için kural değildir.
- **Sonuca etkisi.** İkisi de genellikle otomatik mesafeden küçüktür; daha az kayıt atar.
- **Dikkat.** Ön ayarlar en az %50 kayıt koruma kuralına tabi değildir.

3.9.3 Otomatik mesafe: variogram aralığının yarısı ve azaltma kuralı

- **Ne işe yarar.** Program her çalıştırmada, seyreltme kapalı olsa bile, öngörücülerin mekânsal özilişki aralığını kayıt noktalarından tahmin eder (`sdm_spatial.py:166-281`): en çok 1500 kayıt alt örneklenir; her öngörücü için yönsüz deneysel yarı-variogram, en büyük çift uzaklığının yarısına kadar 15 aralıkta hesaplanır; eşik (sill) dış üçte bir aralıkların ortanca yarı-varyansı, aralık (range) bu eşiğin %95'ine ilk ulaşılan uzaklıktır; değişkenlerin ortancası alınır. 20'den az nokta ya da kullanılabilir variogram yoksa aralık, en büyük uzaklığın onda biri sayılır (`fallback_extent`). Otomatik seyreltme açıkken mesafe aralığın yarısıdır; bu mesafe kayıtların yarısından fazlasını (`THIN_MIN_RETAIN = 0.5`) atıyorsa mesafe en çok 8 kez yarıya indirilir, en az %50'yi koruyan ilk mesafe kullanılır (`sdm_core.py:2484-2522, 2571-2594`).
- **Ne zaman, nasıl ayarlanır.** Türe özgü bir mesafe gerekçesi yoksa savunulabilir bir başlangıçtır; aynı aralık ③ Models'ta mekânsal blok boyu olarak kullanıldığı için seyreltme ile blok boyu birbiriyle tutarlı olur (Roberts vd. 2017'nin ve blockCV'nin [Valavi vd. 2019] mantığı).
- **Sonuca etkisi.** Örnek koşuda aralık 151,5 km bulunmuş, yarısı 75,7 km kayıtların yarısından fazlasını atacağı için 37,9 → 18,9 → 9,5 → 4,7 km'ye inilmiş; 4,7 km'de 221 kaydın 143'ü (%65) korunmuştur (çalıştırma kaydı: "auto (half the variogram range, reduced to keep 65% of records)").
- **Dikkat.** Bölgesel bir eğilimin baskın olduğu iklim katmanlarında aralık çok büyük çıkar ve azaltma kuralı mesafeyi neredeyse hücre boyuna indirir; bu durumda mesafe artık özilişkiyi değil yalnız kümelenmeyi giderir. Her aday mesafe kayıtların yarısından fazlasını atıyorsa seyreltme yapılmaz ve uyarı verilir; aralık hiç hesaplanamazsa yine yapılmaz ve "Type a thinning distance in ① Data" denir (`sdm_core.py:2495-2522`). Elle girilen mesafe için %50 koruma sınırı yoktur.

3.10 Çıktılar ve sonraki adıma geçenler

- `pa_datasets`: her biri (öngörücü tablosu, 0/1 etiket, koordinatlar) olan `PA repetitions` kadar küme; ③ Models'a geçer.
- `background_X` ve koordinatları: MaxEnt arka planı; istenmişse `background_maxent.csv` (sütunlar `longitude, latitude, <değişkenler>`).
- `background_ponly`: ENFA arka planı.
- `accessible_area.tif` ve erişilebilir alan bilgisi; seyreltme bilgisi; özilişki aralığı (③'te blok boyu).
- Çalıştırma kaydının `background`, `study_area` ve `thinning` bölümleri: yokluk kaynağı (`pseudo`, `user`, `user+pseudo`), strateji ve ayrıntıları (aday hücre sayısı, zarf dışı pay, k-ortalama sınıf sayımları), arka plan sayıları, hedef grup bilgisi, erişilebilir alan yöntemi ve payı, kalibrasyon çokgenleri, seyreltme mesafesi ve nasıl seçildiği. Örnek koşuda varsayılan bırakılan ayarlar ayrıca `defaults_left_unchanged` altında listelenir.

3.11 Uzman soruları

S1. HABITUS'taki `random` yalancı yokluk çekilişi biomod2'deki `random` ile aynı mı? Cevap: Kural aynıdır: geçerli hücrelerden eşit olasılıklı, yinelemesiz çekiliş (`sdm_core.py:2301-2302`). İki fark vardır. Birincisi, HABITUS varsayılan olarak çekilişi 200 km'lik erişilebilir alanla sınırlar; biomod2'de böyle bir varsayılan yoktur, kapsamın tamamı kullanılır. İkincisi, yalancı yokluklar hücre merkezlerinden çekilir ve aynı küme içinde bir hücre bir kez seçilir. Varlık hücreleri `random`'da dışlanmaz; bir yalancı yokluk bir varlık hücresine düşebilir.

S2. Yalancı yoklukları varlıklardan uzaklaştıran stratejiler neden skorları şişirir? Cevap: AUC ve TSS, modelin varlık ile yokluğu ayırma becerisini ölçer. Yalancı yokluklar varlıklardan coğrafi (`disk`) ya da çevresel (`sre`, `env_const`) olarak uzaklaştırıldığında ayrım, verinin kuruluşu gereği kolaylaşır; ölçülen beceri

modelden çok örnekleme tasarımından gelir (VanDerWal vd. 2009; Barbet-Massin vd. 2012). Aynı veride stratejileri karşılaştırırken skorların değil, bağımsız bir doğrulama verisiyle (⑦ Validation) elde edilen performansın karşılaştırılması gerekir.

S3. Erişilebilir alan tamponu neden 200 km ve bu değer nereden geliyor? Cevap: 200 km bir başlangıç değeridir (`tabs/tab_data.py:669`, `sdm_core.py:1400`); kodda türe özgü bir gerekçesi yoktur. Arayüz ipucu yarıçapın türün yayılma yeteneğinden ve yayılışın yaşından seçilmesini ister. Barve vd. (2011) M'nin tür ve soru özelinde tanımlanması gerektiğini vurgular; Rojas-Soto vd. (2024) biyocoğrafik birimlere dayalı kalibrasyon alanını önerir (HABITUS'ta `polygon`). Değer her çalıştırmada kayda yazılır ve raporlanmalıdır.

S4. Tampon uzaklığı jeodezik mi hesaplanıyor? Cevap: Hayır, yaklaşık hesaplanır. `buffer` ve `mcp_buffer`, hücre yüksekliğini sabit, genişliğini kayıtların ortalama enleminin kosinüsüyle küçültülmüş kabul eden anizotropik Öklid uzaklık dönüşümü kullanır (`sdm_spatial.py:1022-1062`). Enlem açıklığı küçük alanlarda hata küçüktür; kuzey-güney yönünde binlerce km uzanan bir yayılışta tamponun doğu-batı genişliği uçlarda sapar. Seyreltme ve en az mesafe ise haversine ile hesaplanır.

S5. Kullanıcının yokluk kayıtları MaxEnt'e de giriyor mu? Cevap: Hayır. `Own Absences` kayıtları yalnız `pa_datasets`'in yokluk tarafına girer (`sdm_core.py:3035-3082`). MaxEnt kendi arka planını kullanır (`sdm_core.py:4595-4598`); arka plan yokluk sayılmaz. ENFA da kendi arka planını kullanır. Gerçek yoklukları MaxEnt'e "arka plan" olarak vermek isteyen kullanıcı için kodda bir yol yoktur.

S6. Hedef grup yüzeyinin bant genişliği nasıl seçiliyor; R'deki karşılığı nedir? Cevap: `auto`'da her eksen için $1{,}06\,\min(s,\ \mathrm{IQR}/1{,}34)\,n^{-1/5}$ kullanılır; bu, MASS `bandwidth.nrd` değerinin dörtte biridir ve `kde2d`'nin içte kullandığı sigma ile aynıdır (`sdm_sampling.py:143-151`). Yoğunluk hücre sayımlarının Gauss filtresiyle düzleştirilmesiyle elde edilir; sınırdan dışarı kütle kaybolur (`mode="constant"`). Yüzey en büyük değere ölçeklenir ve 10^{-6} ile tabanlanır, böylece hiç hedef kaydı olmayan hücreler çok düşük olasılıkla da olsa seçilebilir. Kramer-Schadt vd. (2013) ve Phillips vd. (2009) bu yaklaşımın gerekçesini verir.

S7. Otomatik seyreltme mesafesi neden variogram aralığının yarısı, ve neden azaltılıyor? Cevap: Aralık, öngörücülerdeki özilişkinin kaybolduğu uzaklıktır; yarısı kayıtlar arasındaki tekrar eden çevre bilgisini azaltmak için yaygın bir başlangıçtır ve aynı aralık blok boyunu belirlediği için iki karar tutarlı olur (`sdm_core.py:2433-2442`). Bölgesel eğilimi güçlü iklim katmanlarında aralık yüzlerce km çıkar ve yarısı kayıtların çoğunu atar; seyreltmenin amacı veri kümesini değil yinelemeyi gidermek olduğu için mesafe, kayıtların en az %50'si kalana kadar yarıya indirilir (`sdm_core.py:2571-2594`). Örnek koşuda 75,7 km'den 4,7 km'ye inilmiştir.

S8. HABITUS'un seyreltmesi spThin ile aynı sonucu verir mi? Cevap: Kural aynıdır (tutulan iki kayıt arasında en az mesafe, büyük daire uzaklığı), algoritma farklıdır. spThin `thin()` çok sayıda rastgele tekrar yapar ve en çok kayıt tutan kümeyi seçer (Aiello-Lammens vd. 2015); HABITUS 3 tekrarlı açgözlü seçim yapar (`sdm_spatial.py:888-937`), otomatik mesafe aramasında ise tek tekrar. Tutulan kayıt sayısı spThin'in en iyisinden biraz az olabilir; kesin eşitlik sınanmamıştır.

S9. `sre` ile `env_const` (bioclim) arasında ne fark var? Cevap: İkisi aynı hesabı (`_compute_sre_mask`) kullanır: her değişkende varlıkların $[q,\ 1-q]$ aralığı, bütün değişkenlerde aralıkta olan hücre zarf içidir (`sdm_core.py:2079-2119`). `sre`'de q sabit 0,05'tir (`sdm_core.py:3010-3011`); `env_const` + `bioclim`'de `Envelope tail trim` ile ayarlanır. 4'ten az varlıkta değeri olan değişkenler zarfa katılmaz. `env_const` + `pca` ise iki temel bileşen üzerinde çalışır ve değişken sayısı arttıkça zarfın anlamsızlaşmasını önler.

S10. Arka plan ve yalancı yoklukların hepsi aynı tohumdan mı geliyor; tekrarlar birbirinden bağımsız mı? Cevap: `DataFormatter` tek bir tohumlu üreteç kurar ve bütün çekilişler sırayla ondan yapılır (`sdm_core.py:1505`). Her yalancı yokluk kümesi üretecin sonraki durumundan çekildiği için kümeler

birbirinden farklıdır ve aynı tohumla aynı sırada yeniden üretilir. Tekrar sayısını değiştirmek, sonra gelen MaxEnt ve ENFA arka planlarını da değiştirir, çünkü üreteç durumu ilerlemiş olur.

S11. Erişilebilir alan küçük ve hücre sayısı istenen arka plan sayısından az olursa ne olur? Cevap: Alandaki bütün geçerli hücreler kullanılır, istenenden az nokta çekilir ve bir uyarı yazılır (`sdm_core.py:2190-2203`); hangi istekte kaç hücre bulunduğu çalıştırma kaydına (`cells_short_of_request`) eklenir. Arka plan bu durumda alanın tam sayımıdır (Phillips ve Dudík 2008). Çekiliş kapsamın tamamına genişletilmez; bu, kullanıcının tanımladığı M'yi korur.

S12. Bu sekmedeki ayarlardan sonucu en çok etkileyenler hangileri? Cevap: Örnek veriyle sistematik bir duyarlılık analizi yapılmamıştır. Kodun yapısından ve kaynaklardan şu sıra beklenir: (1) erişilebilir alan yöntemi ve yarıçapı (arka planın çevresel kapsamını belirler; Barve vd. 2011), (2) yalancı yokluk stratejisi (ayrımın zorluğunu belirler; Barbet-Massin vd. 2012), (3) yanlılık düzeltmesi (haritanın yanlılığı mı çevreyi mi izlediğini belirler; Phillips vd. 2009), (4) seyreltme ve hücre başına bir kural (kullanılan kayıt sayısını belirler; örnek koşuda 626'dan 143'e). Yalancı yokluk ve arka plan sayıları, makul aralıkta, daha az etkilidir.

3.12 Sık hatalar ve tuzaklar

- `polygon` kipinde `Buffer radius`'u 200 km'de bırakmak: çokgenler 200 km genişletilir. Çokgenleri olduğu gibi kullanmak için 0 girin.
- Rasterler zaten çalışma alanına kesilmişken tampon da kullanmak gereksiz değildir; ama tampon kapsamın %97'sinden fazlasını kaplıyorsa program bunu bildirir.
- `geo_const` ya da `disk` stratejisini 0 km ile kullanmak: hiçbir şey dışlanmaz.
- Odak türün kendi kayıtlarını `Target-group CSV` olarak vermek ya da MaxEnt'teki dairesel kutuyu açmak: yanlılık düzeltilmez, pekiştirilir.
- Hedef grup CSV'sini rasterlerden farklı koordinat sisteminde vermek: "None of the N target-group points falls inside the raster extent."
- Az sayıda gerçek yoklukla `replace`: eşiğe bağlı bütün istatistikler kararsızlaşır; `augment` kullanın.
- Stratejiler arasında AUC/TSS karşılaştırıp en yükseği seçmek: skor farkı tasarımdan gelir (S2).
- Otomatik seyreltmenin her türde özilişkiyi gidereceğini sanmak: azaltma kuralı mesafeyi çok küçültebilir; kullanılan mesafeyi özetten okuyun.
- ENFA / Mahalanobis sekmesinin Mahalanobis'i de değiştirdiğini sanmak: arka planı yalnız ENFA kullanır.

3.13 Kaynaklar

Aiello-Lammens, M.E., Boria, R.A., Radosavljevic, A., Vilela, B., Anderson, R.P., 2015. spThin: an R package for spatial thinning of species occurrence records for use in ecological niche models. Ecography 38, 541–545. https://doi.org/10.1111/ecog.01132

Barbet-Massin, M., Jiguet, F., Albert, C.H., Thuiller, W., 2012. Selecting pseudo-absences for species distribution models: how, where and how many? Methods in Ecology and Evolution 3, 327–338. https://doi.org/10.1111/j.2041-210X.2011.00172.x

Barve, N., Barve, V., Jiménez-Valverde, A., Lira-Noriega, A., Maher, S.P., Peterson, A.T., Soberón, J., Villalobos, F., 2011. The crucial role of the accessible area in ecological niche modeling and species distribution modeling. Ecological Modelling 222, 1810–1819. https://doi.org/10.1016/j.ecolmodel.2011.02.011

Boria, R.A., Olson, L.E., Goodman, S.M., Anderson, R.P., 2014. Spatial filtering to reduce sampling bias can improve the performance of ecological niche models. Ecological Modelling 275, 73–77.

https://doi.org/10.1016/j.ecolmodel.2013.12.012

Kramer-Schadt, S., Niedballa, J., Pilgrim, J.D., Schröder, B., Lindenborn, J., Reinfelder, V., Stillfried, M., Heckmann, I., Scharf, A.K., Augeri, D.M., Cheyne, S.M., Hearn, A.J., Ross, J., Macdonald, D.W., Mathai, J., Eaton, J., Marshall, A.J., Semiadi, G., Rustam, R., Bernard, H., Alfred, R., Samejima, H., Duckworth, J.W., Breitenmoser-Wuersten, C., Belant, J.L., Hofer, H., Wilting, A., 2013. The importance of correcting for sampling bias in MaxEnt species distribution models. Diversity and Distributions 19, 1366–1379. https://doi.org/10.1111/ddi.12096

Lobo, J.M., Jiménez-Valverde, A., Real, R., 2008. AUC: a misleading measure of the performance of predictive distribution models. Global Ecology and Biogeography 17, 145–151. https://doi.org/10.1111/j.1466-8238.2007.00358.x

Phillips, S.J., Anderson, R.P., Schapire, R.E., 2006. Maximum entropy modeling of species geographic distributions. Ecological Modelling 190, 231–259. https://doi.org/10.1016/j.ecolmodel.2005.03.026

Phillips, S.J., Dudík, M., 2008. Modeling of species distributions with Maxent: new extensions and a comprehensive evaluation. Ecography 31, 161–175. https://doi.org/10.1111/j.0906-7590.2008.5203.x

Phillips, S.J., Dudík, M., Elith, J., Graham, C.H., Lehmann, A., Leathwick, J., Ferrier, S., 2009. Sample selection bias and presence-only distribution models: implications for background and pseudo-absence data. Ecological Applications 19, 181–197. https://doi.org/10.1890/07-2153.1

Roberts, D.R., Bahn, V., Ciuti, S., Boyce, M.S., Elith, J., Guillera-Arroita, G., Hauenstein, S., Lahoz-Monfort, J.J., Schröder, B., Thuiller, W., Warton, D.I., Wintle, B.A., Hartig, F., Dormann, C.F., 2017. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. Ecography 40, 913–929. https://doi.org/10.1111/ecog.02881

Rojas-Soto, O., Forero-Rodríguez, J.S., Galindo-Cruz, A., Mota-Vargas, C., Parra-Henao, K.D., Peña-Peniche, A., Piña-Torres, J., Rojas-Herrera, K., Sánchez-Rodríguez, J.D., Toro-Cardona, F.A., Trinidad-Domínguez, C.D., 2024. Calibration areas in ecological niche and species distribution modelling: Unravelling approaches and concepts. Journal of Biogeography 51, 1416–1428. https://doi.org/10.1111/jbi.14834

Valavi, R., Elith, J., Lahoz-Monfort, J.J., Guillera-Arroita, G., 2019. blockCV: An R package for generating spatially or environmentally separated folds for k-fold cross-validation of species distribution models. Methods in Ecology and Evolution 10, 225–232. https://doi.org/10.1111/2041-210X.13107

VanDerWal, J., Shoo, L.P., Graham, C., Williams, S.E., 2009. Selecting pseudo-absence data for presence-only distribution modeling: How far should you stray from what you know? Ecological Modelling 220, 589–594. https://doi.org/10.1016/j.ecolmodel.2008.11.010

4. Değişkenler (② Variables)

Sekme: ② Variables · Kod: `tabs/tab_vif.py` (VIF + Correlation), `tabs/tab_vif_advanced.py` (Advanced Analysis), `sdm_core.py` `compute_vif`, `compute_correlation_matrix`, `variable_priority_ranking` (hesap)

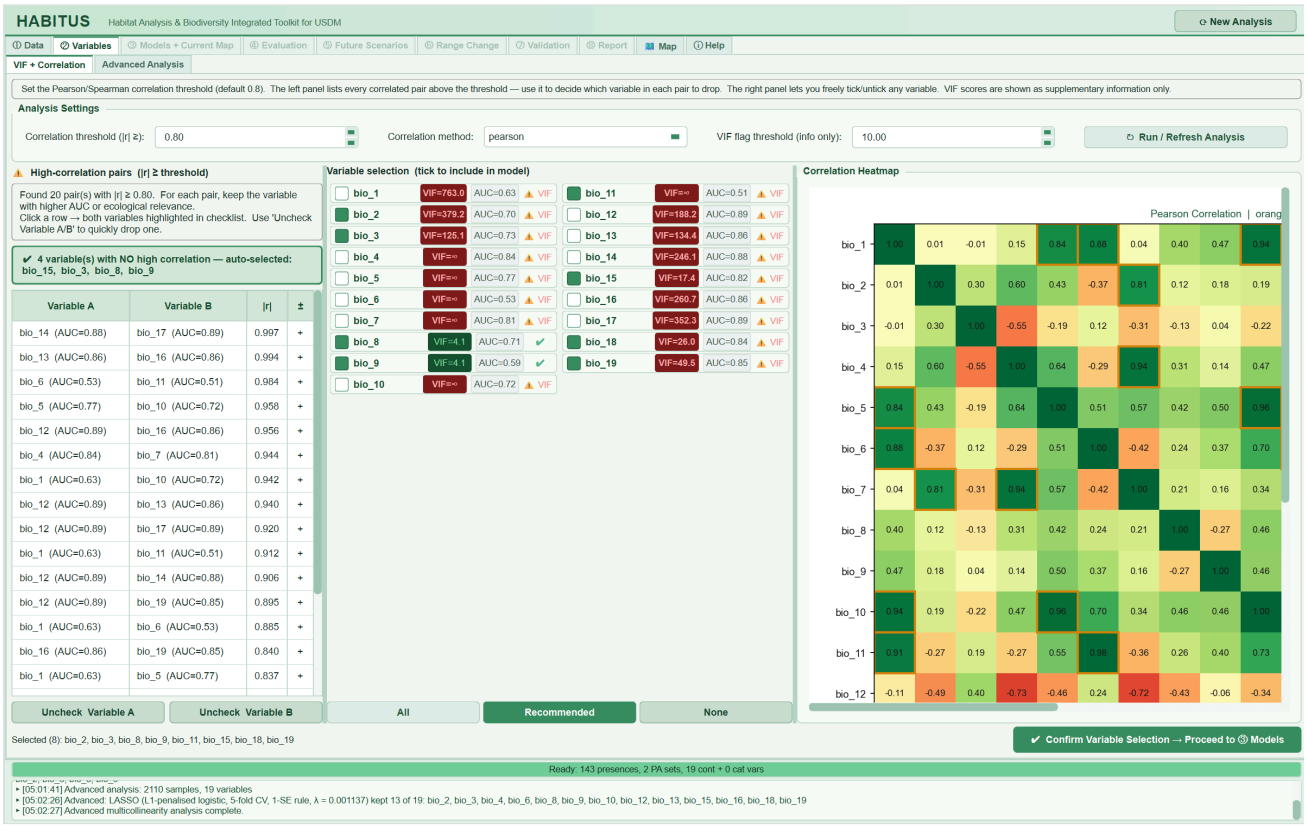
4.1 Bu sekme ne yapar

② Variables, ① Data adımında yüklenen sürekli öngörücüler arasındaki çoklu doğrusallığı (collinearity: bir öngörücünün öteki öngörücülerin doğrusal bir bileşimine yakın olması) gösterir ve modele girecek değişkenleri seçtirir. Seçimi kullanıcı yapar. Program yalnız sıralar, işaretler ve bir başlangıç önerisi (`Recommended`) sunar; onaylanmadan hiçbir değişken modelden çıkmaz.

- **Girdi:** ① Data'nın ürettiği sözde yokluk (pseudo-absence, PA) setleri (`state["pa_datasets"]`) ve sürekli değişken adları (`formatter.var_names`). Kategorik rasterlar ayrı bir bölümde listelenir; VIF'e ve korelasyona girmez.
- **Çıktı:** seçilen sürekli değişkenler (`state["var_names"]`), seçilen ve dışarıda bırakılan kategorik rasterlar (`state["cat_selected"]`, `state["cat_excluded"]`), `figures/` altına yazılan dört dosya (§4.8).
- **Kilit düzeni:** sekme ① Data tamamlanınca açılır. `Confirm Variable Selection → Proceed to ③ Models` düğmesi ③ Models sekmesini açar ve oraya geçer (`tabs/tab_vif.py:1206-1208`).

Sekmenin iki alt sekmesi vardır: `VIF + Correlation` (seçimin yapıldığı yer) ve `Advanced Analysis` (koşul sayısı, temel bileşenler, LASSO ve Ridge ile tamamlayıcı tanı). İkinci alt sekme seçim yapmaz; yalnız bilgi verir.

4.2 Ekranın düzeni



Şekil 4.1. VIF + Correlation alt sekmesi: üstte Analysis Settings şeridi, solda yüksek korelasyonlu çiftler, ortada değişken listesi (VIF ve AUC rozetleriyle), sağda korelasyon ısı haritası, altta seçim özeti ve Confirm düğmesi.

Şekil 4.1 örnek koşudan alınmıştır (*Abies nordmanniana*, 143 kullanılan varlık, 2 PA seti × 1000 nokta, 19 WorldClim biyoiklim değişkeni). Bölgeler:

Bölge	Ne içerir	Alt bölüm
Üst bilgi şeridi ve `Analysis Settings`	Korelasyon eşiği, korelasyon yöntemi, VIF eşiği, `Run / Refresh Analysis`	§4.4
Sol panel `High-correlation pairs (|r| ≥ threshold)`	Eşiği aşan çiftler tablosu, "güvenli" değişken kutusu, `Uncheck Variable A/B`	§4.5
Orta panel `Variable selection (tick to include in model)`	Her sürekli değişken için onay kutusu, VIF rozeti, AUC rozeti, durum simgesi; altında kategorik bölüm; `All`, `Recommended`, `None`	§4.6
Sağ panel `Correlation Heatmap`	Kare hücreli korelasyon matrisi; eşiği aşan hücreler turuncu çerçeveli	§4.5.3
Alt şerit	Seçim özeti ("Selected (n): ...") ve `Confirm Variable Selection → Proceed to ③ Models`	§4.7
`Advanced Analysis` alt sekmesi	`Condition Number`, `PCA`, `LASSO (L1)`, `Ridge (L2)`	§4.9

4.3 Hesaplar hangi örnek üzerinde yapılır

Bu sekmedeki bütün sayılar bir **örneklem** üzerinde hesaplanır, rasterın bütün hücreleri üzerinde değil. Örneklem, ① Data'nın ürettiği PA setlerinin alt alta eklenmesidir: her set, kullanılan varlıkların tamamını ve o setin sözde yokluklarını içerir (`tabs/tab_vif.py:349-353`, `sdm_core.py:3221-3226`). Örnek koşuda bu 2 × (143 + 1000) = 2286 satırdır. MaxEnt'in arka plan noktaları (10 000) bu havuza girmez.

İki alt sekme havuzu farklı hazırlar:

- `VIF + Correlation` havuzu olduğu gibi kullanır; yalnız eksik değer içeren satırları atar (`dropna`). Her varlık, PA seti sayısı kadar (örnekte iki kez) yer alır.
- `Advanced Analysis` yinelenen satırları atar: her PA setinde aynı olan varlıklar bir kez sayılır, iki sette rastlantıyla aynı hücreye düşen sözde yokluklar da tekilleşir (`tabs/tab_vif_advanced.py:356-363`). Örnek koşuda kalan satır sayısı 2110'dur (Şekil 4.2'deki durum satırı: "2110 samples × 19 variables").

> **Not:** VIF ve korelasyon, öngörücülerin kendi aralarındaki ilişkisidir; varlık/yokluk etiketine bakmaz. Ama hangi noktalarda ölçüldüğüne bağlıdır. Erişilebilir alan (① Data) geniş tutulursa sözde yokluklar daha geniş bir iklim aralığına yayılır ve korelasyonlar değişebilir. Aynı katmanların bütün çalışma alanındaki korelasyonu ile bu örneklemdeki korelasyon aynı olmak zorunda değildir.

Tek değişkenli AUC (aşağıda) ise etikete bakar: varlıklar 1, sözde yokluklar 0 alınır.

4.4 Analiz ayarları (Analysis Settings)

Bu şerit, çiftler tablosunun, ısı haritasının ve `Recommended` önerisinin hangi eşiklerle kurulacağını belirler.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Correlation threshold (|r| ≥):`	Bu değere eşit ya da büyük mutlak korelasyonlu çiftleri listeler, ısı haritasında çerçeveler; `Recommended`'ın ikinci adımında kullanılır	0.80	0.30–0.99, adım 0.05	`tabs/tab_vif.py:123-130`, `sdm_core.py:435-445`
`Correlation method:`	Pearson (doğrusal) ya da Spearman (sıra) korelasyonu	`pearson`	`pearson`, `spearman`	`tabs/tab_vif.py:132-136`, `sdm_core.py:300-302`
`VIF flag threshold (info only):`	Rozet renklendirmesi, `VIF_OK` bayrağı ve `Recommended`'ın adım adım VIF eşiği	10.00	2–50, adım 1	`tabs/tab_vif.py:138-143`, `sdm_core.py:405`, `sdm_core.py:427-434`
`↻ Run / Refresh Analysis`	Sıralamayı, çiftleri, listeyi ve ısı haritasını yukarıdaki üç değerle yeniden hesaplar	—	düğme	`tabs/tab_vif.py:339-376`

4.4.1 Korelasyon eşiği (`Correlation threshold`)

- **Ne işe yarar.** Korelasyon matrisinin üst üçgeninde $|r| \geq$ eşik olan her çifti sol tabloya yazar, $|r|$'ye göre büyükten küçüğe sıralar (`tabs/tab_vif.py:696-705`) ve ısı haritasında bu hücreleri turuncu çerçeveyle işaretler (`tabs/tab_vif.py:984-991`). Hiçbir çiftte yer almayan değişkenleri "güvenli" sayar (§4.5.2). `Run / Refresh Analysis` ile birlikte `Recommended`'ın korelasyon adımını da belirler (§4.6.2).
- **Ne zaman, nasıl ayarlanır.** İpucu metni üç yaygın değeri verir: 0.70 (temkinli), 0.80 (varsayılan), 0.90 (gevşek). Dormann vd. (2013) benzetim çalışmalarında $|r| > 0{,}7$'nin, çoklu doğrusallığın model kestirimini ve öngörüyü ciddi biçimde bozmaya başladığı yaklaşık sınır olduğunu bildirir. Model başka bir bölgeye ya da iklime aktarılacaksa (gelecek senaryosu) 0,70 daha güvenlidir; yalnız bugünkü dağılımın betimlenmesi isteniyorsa 0,80 çoğu zaman yeterlidir.
- **Sonuca etkisi.** Eşik düştükçe çift sayısı artar, güvenli değişken sayısı azalır, `Recommended` daha az değişken bırakır. Örnek koşuda 0,80 eşiğiyle 20 çift listelenmiştir (Şekil 4.1; en güçlü çift bio_14–bio_17, $|r| = 0{,}997$).
- **Dikkat.** Kutudaki değeri değiştirmek çift tablosunu ve ısı haritasını hemen yeniler (`tabs/tab_vif.py:880-884`), ama `Recommended` önerisini ve `Corr_OK` bayrağını yenilemez; bunlar için `Run / Refresh Analysis`'e basılmalıdır. Sekme ilk açıldığında gösterilen öneri, ① Data adımının otomatik ön hesabından gelir ve bu ön hesap 0,70 eşiğini, Pearson yöntemini ve VIF eşiği 10'u kullanır (`sdm_core.py:3225-3227`, `sdm_core.py:305-309`); kutuda ise 0.80 yazar. Arayüzdeki değerlerle hesaplanmış bir öneri için önce `Run / Refresh Analysis`'e basın.

4.4.2 Korelasyon yöntemi (`Correlation method`)

- **Ne işe yarar.** `pandas.DataFrame.corr(method=…)` ile Pearson ya da Spearman matrisini hesaplar (`sdm_core.py:300-302`). Çiftler tablosu, ısı haritası, ortalama mutlak korelasyon (`Mean_Corr`) ve `Recommended`'ın korelasyon adımı bu matrisi kullanır. VIF bu seçimden etkilenmez; VIF her zaman doğrusal regresyonla hesaplanır.
- **Ne zaman, nasıl ayarlanır.** İpucu: "pearson: linear correlation (assumes normality)", "spearman: rank-based, robust to non-normality and outliers". Yağış değişkenleri gibi çarpık dağılımlı ya da tekdüze ama doğrusal olmayan ilişkili öngörücülerde Spearman daha uygundur.
- **Sonuca etkisi.** Spearman, tekdüze doğrusal olmayan ilişkileri de yakaladığı için genellikle daha çok çift listeler. VIF adımı değişmediğinden `Recommended`'ın ilk aşaması aynı kalır.
- **Dikkat.** Yöntem kutusu değiştirilince hiçbir şey kendiliğinden yenilenmez (kutuya bağlı bir sinyal yoktur, `tabs/tab_vif.py:132-136`); `Run / Refresh Analysis`'e basılmalıdır.

4.4.3 VIF eşiği (`VIF flag threshold (info only)`)

- **Ne işe yarar.** Üç yerde kullanılır: (1) değişken listesindeki VIF rozeti eşikten büyükse kırmızı, değilse yeşil olur (`tabs/tab_vif.py:443-462`); (2) `VIF_OK` bayrağı $\mathrm{VIF} \le$ eşik ve $\mathrm{VIF} < 999$ olduğunda doğrudur (`sdm_core.py:405`); (3) `Run / Refresh Analysis`'ten sonra `Recommended`'ın adım adım VIF aşaması bu eşiğe kadar değişken çıkarır (`sdm_core.py:427-434`, `tabs/tab_vif.py:357`).
- **Ne zaman, nasıl ayarlanır.** 10 yaygın kullanılan sınırdır; usdm::vifstep'in varsayılanı da `th = 10`'dur. O'Brien (2007) bu tür eşiklerin kural olarak değil, örneklem büyüklüğü ve amaçla birlikte değerlendirilmesi gereken işaretler olarak okunması gerektiğini tartışır. Katsayıları yorumlanacak GLM ve GAM gibi modellerde 5 gibi daha sıkı bir eşik düşünülebilir.
- **Sonuca etkisi.** Eşik düştükçe `Recommended` daha az değişken bırakır. Değişkeni kendiliğinden çıkarmaz: listedeki işaretler değişmez, yalnız `Recommended` düğmesine basıldığında yeni öneri uygulanır.
- **Dikkat.** Arayüzdeki "(info only)" ifadesi ve ipucundaki "does NOT force deselection" cümlesi, eşiğin hiçbir değişkeni zorla çıkarmadığı anlamındadır. Eşik yine de `Recommended` önerisini belirler. Kutudaki değeri değiştirmek tek başına bir şey yenilemez; `Run / Refresh Analysis` gerekir.

4.4.4 VIF nasıl hesaplanır

Her sürekli değişken x_i, öteki bütün sürekli değişkenlere kesişimli doğrusal regresyonla bağlanır (`sklearn.linear_model.LinearRegression`) ve

$$\mathrm{VIF}_i = \frac{1}{1 - R_i^2}$$

hesaplanır (`sdm_core.py:221-297`). Ayrıntılar:

- Yalnız x_i'nin ve öteki değişkenlerin hepsinin sonlu olduğu satırlar kullanılır; 10'dan az böyle satır varsa ya da değişken sabitse sonuç `NaN`'dir ve rozet `VIF=–` gösterir (`sdm_core.py:269-282`).
- R^2, kayan nokta taşmasına karşı [0, 1] aralığına kırpılır. $R^2 > 0{,}9999$ ise (gerçek VIF 10 000'in üstünde) değer **999'a sabitlenir** ve rozet `VIF=∞` gösterir (`sdm_core.py:286-295`, `tabs/tab_vif.py:447-451`). Bu, WorldClim'de birbirinden türetilmiş değişkenlerde olağandır (ör. bio_7 = bio_5 − bio_6).
- En az iki değişken gerekir; tek değişkenle VIF tanımsızdır ve sıralama tablosu her değişkene VIF = 1 yazar (`sdm_core.py:356-368`).

4.5 Yüksek korelasyonlu çiftler ve ısı haritası

Sol panel ve sağdaki ısı haritası aynı korelasyon matrisinden kurulur. Amaç, birbirinin yerine geçen değişken çiftlerini göstermek ve her çiftten hangisinin bırakılacağına kullanıcının karar vermesini kolaylaştırmaktır.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
Çift tablosu (`Variable A`, `Variable B`, `|r|`, `±`)	Eşiği aşan her çift; ad yanında tek değişkenli AUC; mutlak r ve işaret	—	—	`tabs/tab_vif.py:682-801`
Çift satırına tıklama	İki değişkenin listedeki çerçevesini turuncu vurgular ve görünür alana kaydırır	—	—	`tabs/tab_vif.py:827-865`
`Uncheck Variable A` / `Uncheck Variable B`	Seçili çiftin ilk / ikinci değişkeninin işaretini kaldırır	—	düğme	`tabs/tab_vif.py:209-221`, `867-878`
Güvenli değişken kutusu	Hiçbir çiftte yer almayan değişkenler ("auto-selected")	—	—	`tabs/tab_vif.py:707-729`
`Correlation Heatmap`	Kare hücreli matris; RdYlGn renk ölçeği, −1…+1	—	—	`tabs/tab_vif.py:937-1009`

4.5.1 Çift tablosunun sütunları

- `Variable A`, `Variable B`: çiftin iki değişkeni, matristeki doğal sıraya göre (bio_1 önce, bio_10 sonra). Adın yanında parantez içinde o değişkenin tek değişkenli AUC'si yazar ve iki değişkenden AUC'si düşük olan kırmızı, yüksek olan yeşil gösterilir (`tabs/tab_vif.py:752-772`).
- `|r|`: mutlak korelasyon, üç ondalık. Renk: $\geq 0{,}95$ kırmızı, $\geq 0{,}85$ turuncu, altı nötr (`tabs/tab_vif.py:777-785`).
- `±`: korelasyonun işareti (`+` ya da `−`).

Panelin üstündeki bilgi metni çift sayısını yazar ("Found 20 pair(s) with |r| ≥ 0.80"). Hiç çift yoksa "No variable pairs exceed |r| ≥ … All variables can be safely used together." yazar.

> **Dikkat:** Çift tablosunun AUC renklendirmesi ve bilgi metnindeki "keep the variable with higher AUC or ecological relevance" önerisi, `Recommended`'ın ölçütüyle aynı değildir: `Recommended` bir çiftten VIF'i yüksek olanı çıkarır, AUC'ye bakmaz (§4.6.2). Tek değişkenli AUC yalnız bilgi içindir; bu örnekte iyi ayırt eden bir değişken, gerçek etkenle korelasyonu yüzünden iyi ayırt ediyor olabilir ve aktarımda başarısız olabilir (`sdm_core.py:323-339`). Bir çiftten hangisinin kalacağına öncelikle türün ekolojisi karar vermelidir.

4.5.2 Güvenli değişkenler

Eşiği aşan hiçbir çiftte yer almayan değişkenler yeşil bir kutuda "N variable(s) with NO high correlation — auto-selected" başlığıyla listelenir. Liste yeniden kurulduğunda bu değişkenler her zaman işaretli gelir (`tabs/tab_vif.py:427-431`). Örnek koşuda 0,80 eşiğiyle dört değişken güvenlidir: bio_15, bio_3, bio_8, bio_9 (Şekil 4.1).

> **Not:** "Güvenli" yalnız ikili korelasyon anlamındadır. Bir değişken hiçbir tek değişkenle güçlü ilişkili olmadan birkaçının bileşimine yakın olabilir; bunu VIF gösterir. Örnekte bio_3 hiçbir çiftte değildir ama tam kümede VIF'i 125,1'dir. `Recommended` güvenli değişkenleri otomatik eklemez; yalnız adım adım seçimin sonucunu uygular (`tabs/tab_vif.py:900-915`).

4.5.3 Korelasyon ısı haritası

Isı haritası seçilen yöntemle hesaplanan bütün sürekli değişkenlerin matrisini gösterir. Hücreler kare ve sabit boyutludur (25 değişkene kadar 0,5 inç, daha fazlasında 0,4 inç); matris büyükse sıkıştırılmaz, kaydırma çubuğuyla gezilir (`tabs/tab_vif.py:953-963`). 25 değişkene kadar hücre değerleri iki ondalıkla yazılır. Eşiği aşan köşegen dışı hücreler turuncu çerçevelidir; başlık "Pearson Correlation | orange border = |r| ≥ 0.80" biçimindedir. Her çizimde şekil `figures/correlation_heatmap.png` olarak 300 dpi kaydedilir (çıktı klasörü tanımlıysa; `tabs/tab_vif.py:1006-1009`).

4.6 Değişken listesi ve toplu seçim

Orta paneldeki liste, modele girecek sürekli değişkenlerin seçildiği yerdir. 12'den fazla değişken varsa iki sütuna bölünür (`tabs/tab_vif.py:522-528`).

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
Değişken onay kutusu	İşaretli değişken modele girer	güvenli değişkenler ve `Recommended` önerisi işaretli	açık / kapalı	`tabs/tab_vif.py:423-438`
VIF rozeti	Tam kümedeki VIF; yeşil ≤ eşik, kırmızı > eşik, `VIF=∞` sabitlenmiş, `VIF=—` hesaplanamadı	—	—	`tabs/tab_vif.py:441-469`
AUC rozeti	Tek değişkenli AUC; nötr renk, yalnız bilgi	—	0,5–1	`tabs/tab_vif.py:471-499`, `sdm_core.py:380-393`
Durum simgesi	`⚠ VIF` (VIF_OK yanlış), `⚠ Corr` (Corr_OK yanlış), ✓ (önerilen), `–`	—	—	`tabs/tab_vif.py:501-517`
`All`	Bütün sürekli değişkenleri işaretler	—	düğme	`tabs/tab_vif.py:890-893`
`Recommended`	Adım adım seçimin sonucunu uygular, çıkarmaları günlüğe yazar	—	düğme	`tabs/tab_vif.py:900-915`
`None`	Bütün sürekli değişkenlerin işaretini kaldırır	—	düğme	`tabs/tab_vif.py:895-898`

4.6.1 Öncelik tablosu ve sütunları

`variable_priority_ranking` her değişken için bir satır üretir; liste rozetleri bu tablodan okunur ve tablo onayla birlikte `figures/variable_priority_ranking.csv` olarak yazılır (`sdm_core.py:305-465`, `tabs/tab_vif.py:1083-1096`). Satırlar `Priority_Score`'a göre büyükten küçüğe sıralıdır.

Sütun	Anlamı
`Variable`	Değişken adı
`VIF`	Tam kümedeki VIF (iki ondalık; 999 = sabitlenmiş)
`Mean_Corr`	Öteki değişkenlerle ortalama mutlak korelasyon (köşegen 0 alınarak)
`Univariate_AUC`	Değişkenin tek başına varlık ile sözde yokluğu ayırma gücü; 0,5'in altındaysa $1 - \mathrm{AUC}$ alınır
`Priority_Score`	$0{,}5\,(1 - (\min(\mathrm{VIF}, 50) - 1)/49) + 0{,}5\,(1 - \mathrm{Mean_Corr})$, [0, 1]'e kırpılmış
`VIF_OK`	$\mathrm{VIF} \leq$ eşik ve $\mathrm{VIF} < 999$
`Corr_OK`	`Mean_Corr` < korelasyon eşiği
`Recommended`	Adım adım seçimde kaldı mı
`VIF_in_selection`	Önerilen küme içindeki VIF (önerilmeyenlerde boş)

`Priority_Score` yalnız çoklu doğrusallığa dayanır; tek değişkenli AUC tabloda bilgi olarak durur ama sıralamaya ağırlık olarak girmez (`sdm_core.py:395-401`). Gerekçe kodda açıktır: bu örneklemde iyi ayırt eden bir öngörücüyü öne çıkarmak, ekolojik anlamı ve başka bölgeye ya da iklime aktarılabilirliği gözetmeyen bir seçim olur.

> **Not:** `Corr_OK`, tek tek çiftlere değil değişkenin **ortalama** mutlak korelasyonuna bakar. Örnek koşuda 0,70 eşiğiyle 19 değişkenin hepsinde `Corr_OK` doğrudur (en yüksek `Mean_Corr` bio_12'de 0,533), oysa 20 çift 0,80'i aşar. Bu yüzden ⚠ `Corr` simgesi seyrek görülür; çiftleri sol tablo gösterir.

4.6.2 Önerilen seçim (`Recommended`): adım adım VIF ve korelasyon çifti

- **Ne işe yarar.** `Recommended` düğmesi, `variable_priority_ranking`'in `Recommended` sütununu listeye uygular; başka hiçbir değişkeni işaretli bırakmaz (`tabs/tab_vif.py:907-909`). Bu sütun iki aşamalı bir süreçle bulunur (`sdm_core.py:427-447`):

  1. **Adım adım VIF.** Bütün sürekli değişkenlerle başlanır. Kalan kümenin VIF'leri yeniden hesaplanır; en büyük VIF eşikten büyükse o değişken çıkarılır. En büyük VIF eşiğe eşit ya da altında kalınca ya da tek değişken kalınca durulur.
  2. **Korelasyon çifti.** Kalan değişkenler arasında $|r| \geq$ korelasyon eşiği olan çiftlerden en güçlüsü bulunur; çiftin iki üyesinden kalan kümedeki VIF'i büyük olan çıkarılır (eşitse ilki). Hiç böyle çift kalmayana kadar sürer.

  Çıkarılan her değişken, çıkarıldığı andaki VIF ya da $|r|$ değeriyle sırasıyla günlüğe yazılır.

- **Ne zaman, nasıl ayarlanır.** Bir başlangıç noktası olarak kullanılır. İpucu metni de bunu söyler: "A starting point: check it against what you know of the species." Önce eşikleri ayarlayın, `Run / Refresh Analysis`'e basın, sonra `Recommended`'a basın ve sonucu türün ekolojisine göre elle düzeltin (bir çiftten ekolojik olarak daha anlamlı olanı tutmak gibi).
- **Sonuca etkisi.** Örnek koşuda adım adım seçim 11 değişkeni şu sırayla çıkarmıştır: bio_4 (VIF 999), bio_5 (999), bio_1 (757,27), bio_7 (653,22), bio_17 (346,39), bio_16 (236,16), bio_6 (130,1), bio_12 (80,89), bio_10 (74,4), bio_14 (26,21), bio_13 (15,98). Kalan sekiz değişken bio_2, bio_3, bio_8, bio_9, bio_11, bio_15, bio_18, bio_19'dur; bu küme içindeki VIF'ler 1,65 (bio_15) ile 6,43 (bio_18) arasındadır ve aralarında korelasyon eşiğini aşan çift kalmadığından ikinci aşama bir şey çıkarmamıştır (`_calisma/figures/variable_priority_ranking.csv`, oturum günlüğü). Tam kümede yalnız iki değişkenin (bio_8 ve bio_9) VIF'i 10'un altındadır; tam küme VIF'ine göre süzmek bu yüzden çok az değişken bırakırdı. Adım adım yöntem bu sorunu giderir.
- **Dikkat.** Adım adım yöntem tam kümedeki VIF'e göre değil, her adımda yeniden hesaplanan VIF'e göre çalışır. Bu yüzden tam kümede `VIF=∞` gösteren bir değişken (örnekte bio_11) önerilen kümede kalabilir; listedeki rozet tam kümeyi, `VIF_in_selection` sütunu önerilen kümeyi anlatır. Sabitlenmiş değerler (999) eşit olduğunda değişkenlerin doğal sırasındaki ilki çıkarılır (`v_now.idxmax()`, `sdm_core.py:430`); bu durumda çıkarma sırası, VIF'i sabitlemeyen bir hesaptan farklı olabilir (§4.11, S2).

usdm::vifstep ile aynı sonuç

Adım adım VIF aşaması, R'deki usdm paketinin `vifstep` işlevinin (Naimi vd. 2014) yaptığı işi yapar. Bu kitap için ikisi aynı matriste karşılaştırılmıştır: çalışma sınırı (10° B–50° D, 30–46° K) içinden rastgele seçilmiş 5000 kara hücresinde 19 WorldClim 2.1 biyoiklim değişkeni; HABITUS'ta `variable_priority_ranking(..., vif_threshold=10)`, R'de `usdm::vifstep(d, th = 10)` (betik: `habitus_dogrulama\H3_yazilim_tanitim\betikler\Y10_vif_usdm.py`). İki program 11 değişkeni **aynı sırayla** çıkarmıştır (bio_5, bio_10, bio_11, bio_16, bio_7, bio_17, bio_1, bio_12, bio_3, bio_13, bio_18) ve kalan sekiz değişkenin VIF'leri on dördüncü ondalığa kadar aynıdır:

Değişken	usdm::vifstep VIF	HABITUS VIF
bio_2	2,375571	2,375571
bio_4	2,524650	2,524650
bio_6	6,864633	6,864633
bio_8	2,787451	2,787451
bio_9	6,130096	6,130096

Değişken	usdm::vifstep VIF	HABITUS VIF
bio_14	3,548789	3,548789
bio_15	3,131781	3,131781
bio_19	2,161437	2,161437

(Kaynak: `habitus_dogrulama\H3_yazilim_tanitim\yazim\dogrulama_vif_usdm.csv`; tablo altı ondalığa yuvarlanmıştır.) Karşılaştırmada korelasyon eşiği 0,99 tutulmuş, yani ikinci aşama devre dışı bırakılmıştır; usdm'de bu aşamanın karşılığı yoktur (usdm::vifcor korelasyonu önce ve farklı bir kuralla uygular).

4.6.3 Tek değişkenli AUC

Her değişken için `sklearn.metrics.roc_auc_score(y, x)` hesaplanır; 0,5'in altındaysa ilişki ters yönlü sayılıp $1 - \mathrm{AUC}$ yazılır (`sdm_core.py:380-393`). 10'dan az geçerli satır varsa ya da hesap başarısızsa 0,5 yazılır. Rozet bilerek nötr renklidir: düşük AUC bir değişkeni çıkarmak için gerekçe değildir (`tabs/tab_vif.py:471-499`). Örnek koşuda en yüksek değerler yağış değişkenlerindedir (bio_12 0,887; bio_17 0,886), en düşükler sıcaklık değişkenlerinde (bio_11 0,51; bio_6 0,533).

4.6.4 Kategorik değişken bölümü

Veride kategorik raster varsa listenin altında `Categorical variables (not part of VIF)` başlıklı sabit bir kutu açılır (`tabs/tab_vif.py:557-671`). Her kategorik raster bir satırdır:

Öğe	Ne yapar	Varsayılan	Kod
Onay kutusu	İşaretli raster modele girer	işaretli (önceki onay varsa o)	`tabs/tab_vif.py:602-616`
Sınıf rozeti (`N classes`)	Sınıf sayısı; ipucunda sınıf başına "varlık / kayıt" (ilk PA seti)	—	`tabs/tab_vif.py:618-639`
Sınıf sayımı metni	İlk dört sınıfın "sınıf:varlık/kayıt" değerleri	—	`tabs/tab_vif.py:641-651`
Seyrek sınıf uyarısı	3'ten az varlık içeren sınıf sayısı ("⚠ k classes <3 presences")	—	`tabs/tab_vif.py:653-664`, `sdm_core.py:1264`

Kategorik rasterlar VIF'e ve korelasyona girmez; `All`, `Recommended` ve `None` bu kutuya dokunmaz. İpucu metni her algoritmanın kategorik değişkeni nasıl kullandığını söyler: GLM, GLMNET, SVM ve ANN kukla (one-hot) sütunlarla; GAM faktör terimiyle; MaxEnt kategorik özellik sınıfıyla; ağaç tabanlı yöntemler tamsayı bölmeleriyle; BIOCLIM, Domain, ENFA, Mahalanobis ve ESM yalnız sürekli değişkenleri kullanır (`tabs/tab_vif.py:536-545`). Yalnız bir rasterın NoData'sı yüzünden ① Data'da kayıt düşmüşse ipucu bunu sayılarıyla yazar ve rasterı dışarıda bırakmanın örneklemeyi yeniden yapmadığını belirtir. Örnek koşuda kategorik raster yoktur, bu yüzden kutu görünmez.

4.7 Onay (Confirm) ve kayıt/öngörücü denetimi

Arayüzdeki ad	Ne yapar	Varsayılan	Kod
`✓ Confirm Variable Selection → Proceed to ③ Models`	Seçimi kaydeder, dosyaları yazar, ③ Models'i açar	—	`tabs/tab_vif.py:1160-1225`

4.7.1 Onay

- **Ne işe yarar.** İşaretli sürekli değişkenleri doğal sırayla `state["var_names"]`'e, kategorik seçimi `state["cat_selected"]`'e ve dışarıda bırakılanları `state["cat_excluded"]`'e yazar (`tabs/tab_vif.py:1187-1191`). Değişkenlerin işaretlenme sırası önemli değildir; seçim her zaman doğal ada göre sıralanır (`sdm_core.canonical_order`). Ardından §4.8'deki dosyaları yazar, ③ Models'i açar ve onun `Algorithm Settings` alt sekmesine geçer.
- **Ne zaman, nasıl ayarlanır.** Seçim bittiğinde bir kez basılır. Seçim sonradan değiştirilirse yeniden basılmalıdır; aksi hâlde ③ Models eski seçimi kullanır.
- **Sonuca etkisi.** Bundan sonraki bütün adımlar (modeller, gelecek projeksiyonları, yenilik haritaları) yalnız onaylanan değişkenleri kullanır.
- **Dikkat.** En az iki sürekli değişken gerekir: hiç seçim yoksa "Please select at least one variable.", tek değişken varsa "At least 2 variables are needed for modelling. Please select more variables." iletisi çıkar (`tabs/tab_vif.py:1162-1167`). Modeller daha önce çalıştırılmışsa "Re-confirm Variable Selection" penceresi, onaylamanın bütün modelleri, haritaları, projeksiyonları ve değerlendirme sonuçlarını sileceğini söyler; `Yes` denirse bunlar silinir (`tabs/tab_vif.py:1170-1185`, `main_dialog.py:528-542`).

4.7.2 Kayıt/öngörücü denetimi

Onaydan sonra program, kullanılan varlık sayısını seçilen öngörücü sayısına (sürekli + kategorik) böler (`sdm_core.py:2889-2909`). Oran 10'un altındaysa bilgi penceresinin başına ve günlüğe ("ADVISORY: ...") şu biçimde bir uyarı ekler: "... presence records for the k predictors selected (... per predictor). Below about 10 per predictor the model is prone to overfitting and the evaluation statistics become unstable; n records support about n/10 predictors." Bu, ① Data'daki aynı denetimin seçilmiş küme için yinelenmesidir; ① Data yalnız yüklenen katman sayısını görebilir. Örnek koşuda 143 varlık ve 19 katmanla ① Data uyarı vermişti (7,5 kayıt/öngörücü; "143 records allow about 14"); seçilen 8 değişkenle oran 17,9 olduğundan onayda uyarı çıkmamıştır. Yeni bir onay eski uyarının yerine geçer.

4.8 Dışa aktarma

`VIF + Correlation` alt sekmesi onayla birlikte çıktı klasörünün `figures/` alt klasörüne şunları yazar (`tabs/tab_vif.py:1193-1197`):

Dosya	İçerik
`high_correlation_pairs.csv`	`Variable_A`, `Variable_B`, `Correlation_r`, `AUC_A`, `AUC_B`, `Direction`, `Recommend_drop` (eşiği aşan çiftler)
`variable_priority_ranking.csv`	§4.6.1'deki tablo
`vif_barplot.png`	Tam küme VIF çubukları; eşik (kesikli) ve 10 (noktalı) çizgisi; çubuk 50'de kesilir, `∞` ve " (>50, capped)" etiketleri
`correlation_heatmap_selected.png`	Yalnız seçilen değişkenlerin korelasyon matrisi

Bunlara ek olarak `correlation_heatmap.png` her ısı haritası çiziminde yazılır. Çıktı klasörü tanımlı değilse hiçbir dosya yazılmaz.

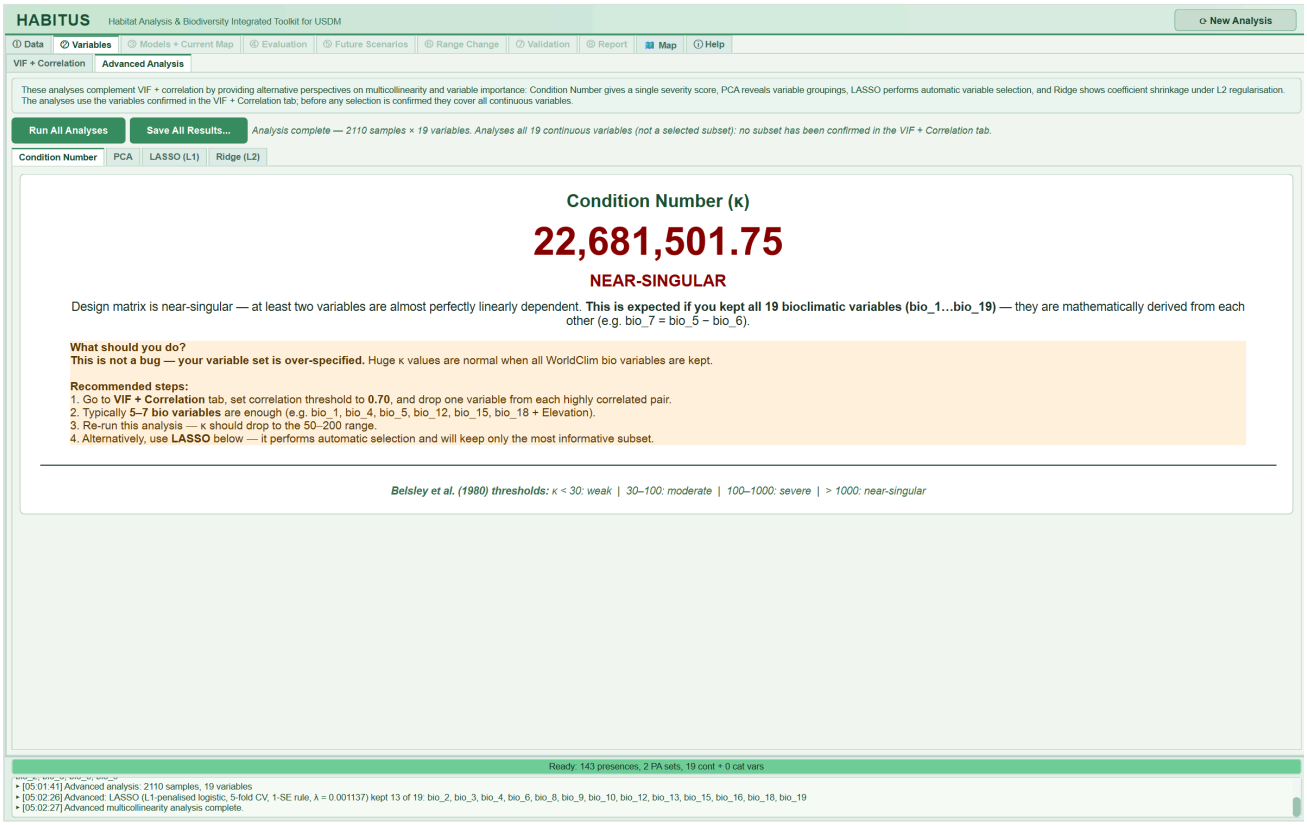
> **Dikkat:** `high_correlation_pairs.csv`'deki `Recommend_drop` sütunu, çiftten tek değişkenli AUC'si düşük olanı yazar (`tabs/tab_vif.py:1066-1070`). Bu, `Recommended`'ın kuralından (VIF'i yüksek olanı çıkar) farklıdır ve bir

> öneri değil, AUC karşılaştırmasıdır.

`Advanced Analysis`'in `Save All Results...` düğmesi ayrıca `advanced_analysis/` klasörüne yazar (§4.9.6). Raporun (⑧) değişken bölümü ve ODMAP dışa aktarımı, bu sekmedeki korelasyon eşiğini, yöntemini ve VIF eşiğini okur (`tabs/tab_report.py:528-530`, `odmap.py:487-492`).

4.9 Gelişmiş analiz (Advanced Analysis)

Bu alt sekme dört tamamlayıcı tanı sunar. Hiçbiri değişken listesindeki işaretleri değiştirmez; önerileri (PCA'daki `RECOMMENDED`, LASSO'daki `SELECTED`) yalnız bilgi içindir.



Şekil 4.2. Advanced Analysis > Condition Number: 19 sürekli değişkenin tamamıyla hesaplanan koşul sayısı (22 681 501,75, NEAR-SINGULAR), yapılacaklar kutusu ve Belsley eşikleri.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Run All Analyses`	Dört analizi sırayla çalıştırır	—	düğme	`tabs/tab_vif_advanced.py:384-407`
`Save All Results...`	Grafikleri ve tabloları `advanced_analysis/`'e yazar	ilk çalıştırmaya kadar devre dışı	düğme	`tabs/tab_vif_advanced.py:760-891`
`Penalty λ:` (LASSO)	Cezanın nasıl seçileceği	`1-SE rule (CV)`	`1-SE rule (CV)`, `Minimum CV deviance`, `Typed value`	`tabs/tab_vif_advanced.py:259-269`
λ değer kutusu (LASSO)	`Typed value` seçiliyken kullanılan λ	0.0100 (devre dışı)	0.0001–10, 4 ondalık	`tabs/tab_vif_advanced.py:270-281`

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`CV folds:`	LASSO ve Ridge için katmanlı çapraz doğrulama kat sayısı	5	3–10	`tabs/tab_vif_advanced.py:282-284`
`Recompute`	Yalnız LASSO'yu yeniden hesaplar	—	düğme	`tabs/tab_vif_advanced.py:285-287`, `409-415`

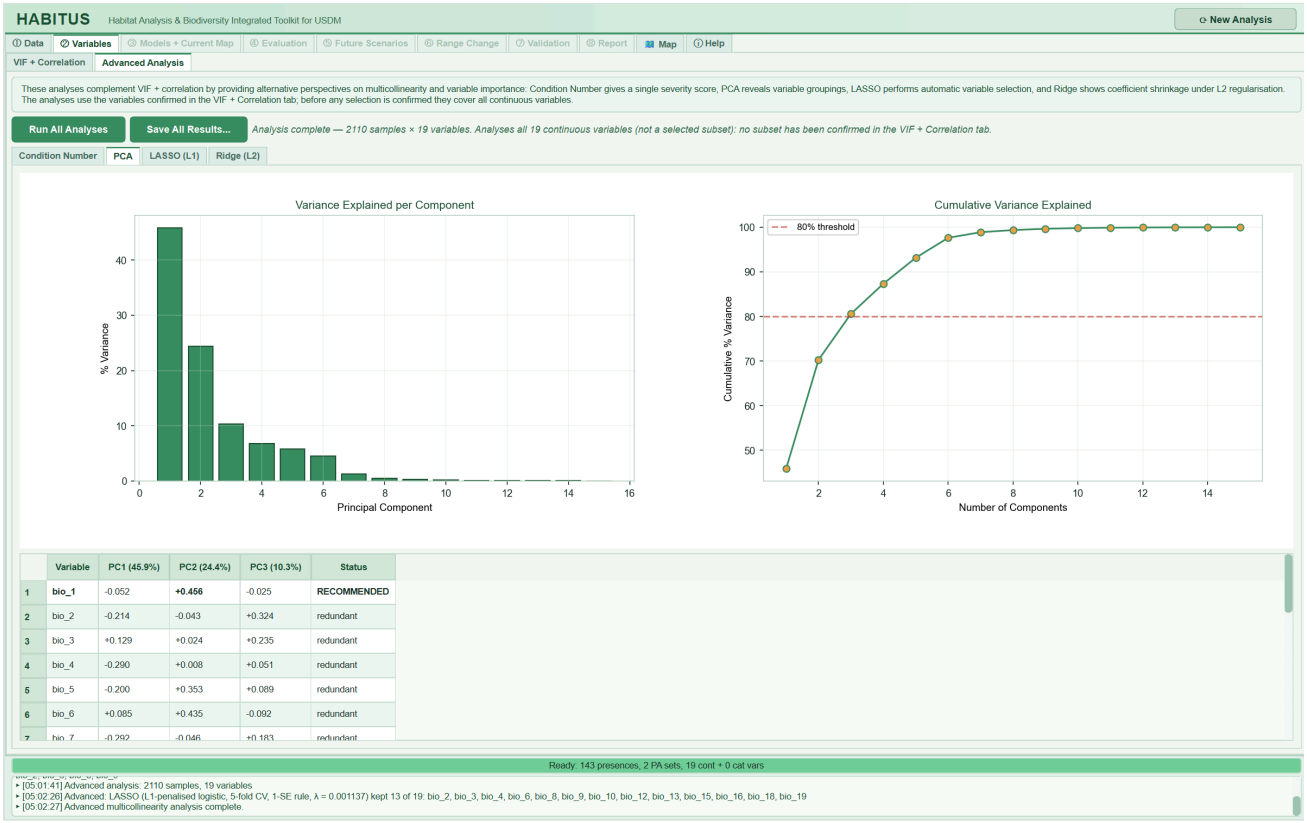
4.9.1 Kapsam ve örnek

Analiz, `VIF + Correlation`'da onaylanmış değişkenleri kullanır. Henüz onay yoksa ya da onaylanan küme bütün sürekli değişkenleri kapsıyorsa bütün sürekli değişkenler kullanılır ve durum satırı bunu yazar: "Analyses all 19 continuous variables (not a selected subset): no subset has been confirmed in the VIF + Correlation tab." (`tabs/tab_vif_advanced.py:337-345`, `375-380`). Örneklem, §4.3'teki tekilleştirilmiş PA havuzudur; sabit (varyansı 10^{-10}'dan küçük) sütunlar atılır ve günlüğe yazılır (`tabs/tab_vif_advanced.py:365-373`). Bu sırayla çalışmak önerilir: önce bütün değişkenlerle tanı, sonra onaydan sonra seçilen kümeyle yeniden `Run All Analyses`.

4.9.2 Koşul sayısı (`Condition Number`)

- **Ne işe yarar.** Değişkenler standartlaştırılır (ortalama 0, standart sapma 1), matrisin tekil değerleri hesaplanır ve $\kappa = s_{\max}/s_{\min}$ yazılır (`tabs/tab_vif_advanced.py:419-431`). 10^{-10}'dan küçük tekil değerler atılır; atılan varsa metne yakın doğrusal bağımlılık notu eklenir. Sınıflama: $\kappa < 30$ `LOW`, $30 \le \kappa < 100$ `MODERATE`, $100 \le \kappa < 1000$ `SEVERE`, $\kappa \ge 1000$ `NEAR-SINGULAR`; her sınıf için "What should you do?" kutusunda yapılacaklar yazar.
- **Ne zaman, nasıl ayarlanır.** Ayarı yoktur. Bütün küme için bir kez, seçimden sonra bir kez bakmak yararlıdır: seçim sonrasında κ'nın düşmesi beklenir.
- **Sonuca etkisi.** Yalnız tanıdır. Örnek koşuda 19 değişkenle κ = 22 681 501,75'tir (Şekil 4.2); bu, birbirinden türetilmiş biyoiklim değişkenlerinin hepsi tutulduğunda beklenen bir değerdir.
- **Dikkat.** Ekrandaki eşikler Belsley vd. (1980)'e atfedilir. Belsley'nin ölçütü her tekil değer için koşul indeksleri ve varyans ayrıştırma oranlarıdır; tek bir κ ve dört sınıflık bu ölçek pratik bir özetlemedir. NEAR-SINGULAR sınıfının yapılacaklar kutusundaki öneriler ("Typically 5–7 bio variables are enough …", "κ should drop to the 50–200 range") genel yol göstericidir, bu veriye özgü bir hesap değildir.

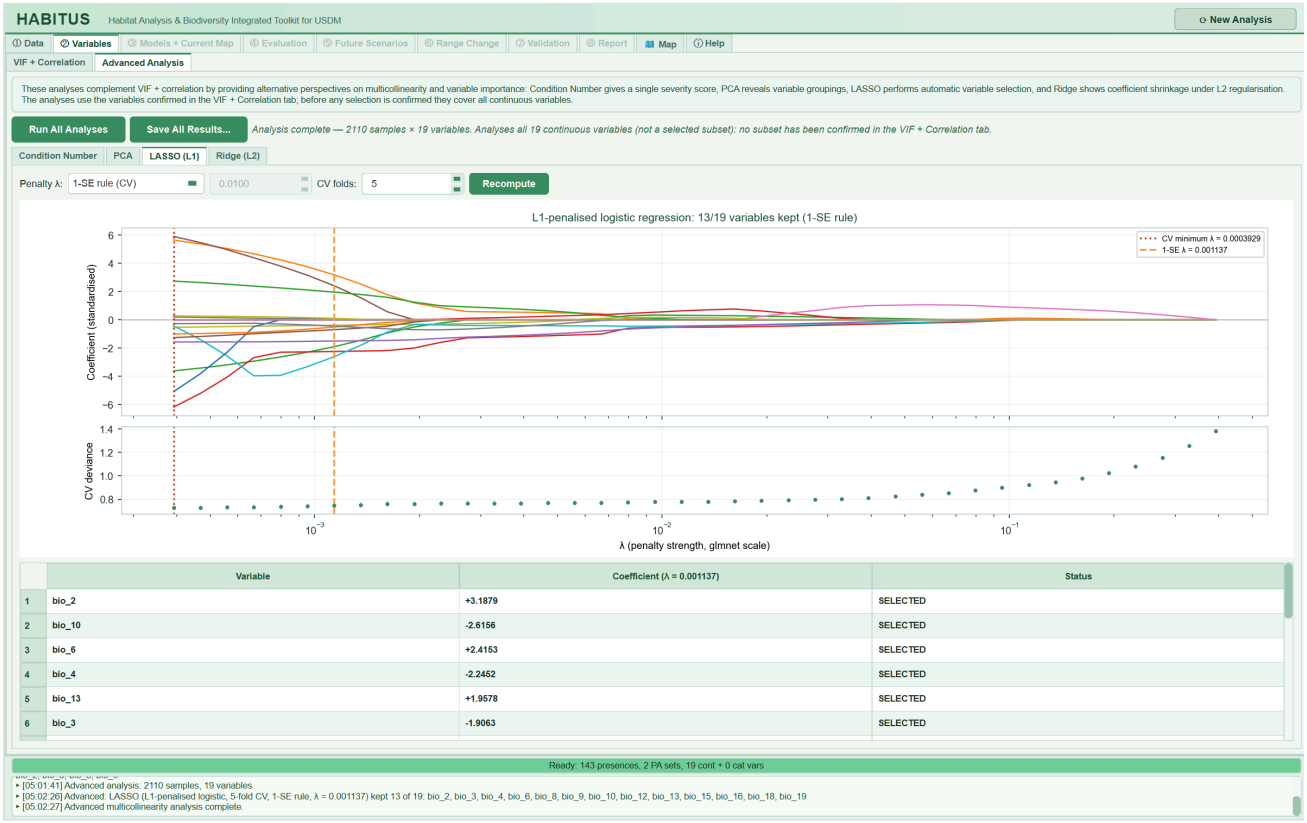
4.9.3 Temel bileşenler (`PCA`)



Şekil 4.3. Advanced Analysis > PCA: solda bileşen başına açıklanan varyans, sağda birikimli varyans ve yüzde 80 çizgisi; altta ilk üç bileşenin yükleri ve Status sütunu.

- **Ne işe yarar.** Standartlaştırılmış değişkenlere `sklearn.decomposition.PCA` uygulanır (`tabs/tab_vif_advanced.py:499-596`). Solda ilk 15 bileşenin açıkladığı varyans yüzdesi, sağda birikimli varyans ve %80 çizgisi çizilir. Tabloda birikimli varyansı %80'e ulaştıran k bileşenin yükleri (en az 2, en çok 8) gösterilir; sütun başlıkları bileşenin varyans payını yazar ("PC1 (45.9%)"). Her bileşende mutlak yükü en büyük değişken kalın ve yeşil vurgulanır ve `Status` sütununda `RECOMMENDED` alır; öteki değişkenler `redundant` yazar. $|\text{yük}| > 0{,}4$ olan öteki hücreler açık sarı boyanır.
- **Ne zaman, nasıl ayarlanır.** Ayarı yoktur. Değişkenlerin hangi gruplarda toplandığını (örneğin sıcaklık ve yağış eksenleri) görmek için kullanılır.
- **Sonuca etkisi.** Yalnız bilgi. Örnek koşuda ilk üç bileşen varyansın %45,9, %24,4 ve %10,3'ünü açıklar; birikimli %80 üçüncü bileşende aşıldığı için tabloda üç bileşen vardır ve en çok üç değişken `RECOMMENDED` alır (Şekil 4.3'te görülen satırlarda bio_1, PC2'de +0,456 yükle).
- **Dikkat.** Bileşen başına "en büyük yüklü değişken" kuralı bir sezgiseldir: iki bileşende aynı değişken en büyük yüke sahipse önerilen değişken sayısı k'dan az olur. %80 sınırı da gelenekseldir, koddan gelen sabit bir değerdir (`tabs/tab_vif_advanced.py:539-540`).

4.9.4 LASSO (L1) — cezalı lojistik regresyon

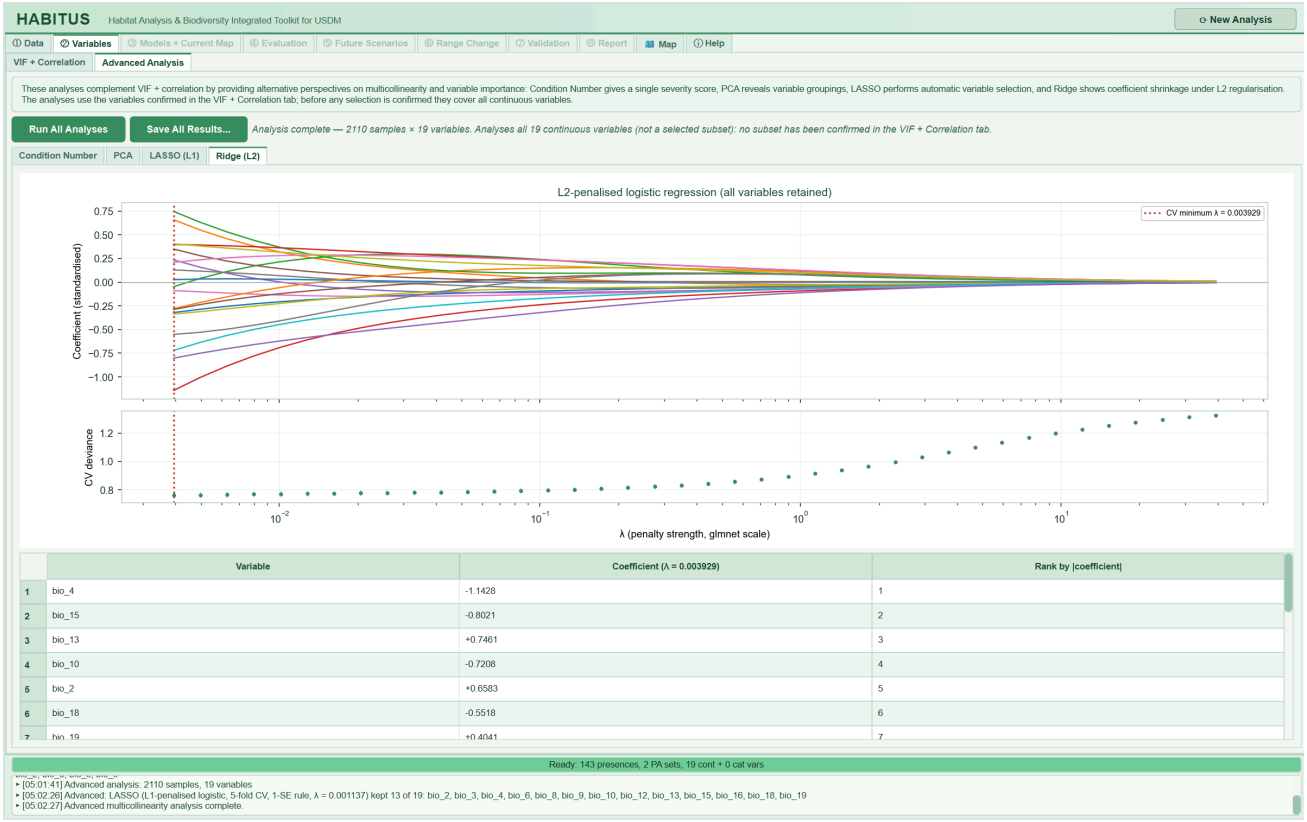


Şekil 4.4. Advanced Analysis > LASSO (L1): üstte standartlaştırılmış katsayıların λ'ya göre yolları, altta çapraz doğrulama sapması; kırmızı noktalı çizgi en küçük sapmanın, turuncu kesikli çizgi 1-SE kuralının λ'sı; altta katsayı tablosu ve Status sütunu.

- **Ne işe yarar.** Varlık (1) / sözde yokluk (0) yanıtına, standartlaştırılmış öngörücülerle L1 cezalı lojistik regresyon (Tibshirani 1996) uyarlanır (`tabs/tab_vif_advanced.py:43-128`, `638-711`). Ayrıntılar:
  - Varlıklar ve yokluklar **eşit toplam ağırlık** alır (`compute_sample_weight("balanced")`; Barbet-Massin vd. 2012), böylece çok sayıdaki sözde yokluk sonucu belirlemez.
  - λ, glmnet ölçeğindedir: ortalama ağırlıklı binom sapması + λ × ceza; scikit-learn'ün `C`'si $1/(n\lambda)$ alınır. 40 değerlik logaritmik ızgara, hiçbir değişkenin kalmadığı en küçük λ'dan ($\lambda_{\max}$) başlar ve $\lambda_{\max}/1000$ 'e iner.
  - Her λ için katmanlı k katlı çapraz doğrulamayla ağırlıklı binom sapması hesaplanır; her katta standartlaştırma yalnız eğitim kısmından yapılır. Kat sayısı `CV folds` ile, ama az sınıfın gözlem sayısından büyük olmayacak biçimde ve en az 2 olarak belirlenir.
  - `1-SE rule (CV)`: sapması en küçük sapmanın bir standart hatası içinde kalan en büyük λ (Hastie vd. 2009, §7.10). `Minimum CV deviance`: sapması en küçük λ. `Typed value`: kutuya yazılan λ ile bütün veriye tek bir uyum.
  - Mutlak katsayısı 10^{-8}'den büyük değişkenler `SELECTED`, ötekiler `dropped` yazar. Tablo mutlak katsayıya göre sıralıdır.
- **Ne zaman, nasıl ayarlanır.** Varsayılan `1-SE rule (CV)` değişken seçimi için yaygın tercihtir: daha az ve daha kararlı değişken bırakır. Tahmin gücüne bakılıyorsa `Minimum CV deviance` seçilebilir. Çok az varlık varsa kat sayısını düşürmek (3) katların kararlılığını artırabilir.
- **Sonuca etkisi.** Yalnız bilgi; model ayarlarına geçmez. Örnek koşuda (19 değişken, 2110 satır) 1-SE λ = 0,001137 ile 19 değişkenden 13'ü kalmıştır (bio_2, bio_3, bio_4, bio_6, bio_8, bio_9, bio_10, bio_12, bio_13, bio_15, bio_16, bio_18, bio_19); en küçük sapmanın λ'sı 0,0003929'dur (Şekil 4.4).

- **Dikkat.** LASSO, birbirine çok bağlı değişkenlerden birini neredeyse keyfî seçer; ilişkili değişkenler arasında seçim kararsızdır ve örneklemdeki küçük değişikliklerle değişebilir. Bu nedenle LASSO'nun tuttuğu küme ile `Recommended`'ın kümesi farklıdır (örnekte bio_4, bio_6, bio_10, bio_12, bio_13, bio_16 LASSO'da kalmış, adım adım VIF'te çıkarılmıştır) ve bu beklenen bir durumdur: LASSO ayırt etme gücüne, adım adım VIF yalnız çoklu doğrusallığa bakar. Katsayılar havuzlanmış PA setleri üzerindedir; son modelin katsayıları değildir. Örnek koşuda en küçük sapma ızgaranın en küçük λ'sına denk gelmiştir; bu, ızgaranın daha küçük λ'lara uzatılması gerekebileceğini gösterir (bkz. S8).

4.9.5 Ridge (L2) — katsayı büzülmesi



Şekil 4.5. Advanced Analysis > Ridge (L2): standartlaştırılmış katsayıların λ'ya göre yolları ve çapraz doğrulama sapması; kırmızı noktalı çizgi en küçük sapmanın λ'sı; altta mutlak katsayıya göre sıra.

- **Ne işe yarar.** LASSO ile aynı kurulum, L2 cezasıyla (`tabs/tab_vif_advanced.py:715-748`). Izgara 100 × $\lambda_{\max}$'tan başlar ve dört büyüklük mertebesi iner. λ her zaman en küçük çapraz doğrulama sapmasıyla seçilir; ayarı yoktur, yalnız `CV folds` LASSO ile ortaktır. Ridge hiçbir değişkeni sıfırlamaz; tablo değişkenleri mutlak standartlaştırılmış katsayıya göre sıralar ve `Rank by |coefficient|` sütununda sırayı yazar. Hiçbir değişken "önerilmez".
- **Ne zaman, nasıl ayarlanır.** İlişkili değişkenler bir arada tutulduğunda katsayıların nasıl paylaşıldığını görmek için. Ridge katsayıları LASSO'nunkinden daha kararlıdır.
- **Sonuca etkisi.** Yalnız bilgi. Örnek koşuda λ = 0,003929; en büyük mutlak katsayılar bio_4 (−1,1428), bio_15 (−0,8021), bio_13 (+0,7461), bio_10 (−0,7208), bio_2 (+0,6583) (Şekil 4.5).
- **Dikkat.** `Recompute` düğmesi yalnız LASSO'yu yeniden hesaplar; `CV folds` değiştirildikten sonra Ridge'in de yenilenmesi için `Run All Analyses`'e basılmalıdır. Örnek koşuda Ridge'in en küçük sapması da ızgaranın en küçük λ'sına denk gelmiştir.

4.9.6 Gelişmiş analizin dışa aktarımı

`Save All Results...` çıktı klasöründe `advanced_analysis/` alt klasörünü kurar ve şunları yazar (`tabs/tab_vif_advanced.py:760-891`): `condition_number.txt` (κ, değişken ve satır sayısı, ilk 10 tekil değer), `pca_variance.png`, `pca_loadings.csv` (ilk k bileşen), `pca_variance.csv` (bütün bileşenlerin varyans ve birikimli varyans yüzdesi), `lasso_coefficient_paths.png`, `lasso_coefficients.csv`, `lasso_path.csv` (λ ızgarası, CV sapması ve standart hatası, bütün katsayılar), `lasso_info.txt` (kayıt sayısı, kat sayısı, iki λ, kullanılan λ, seçilen değişkenler), `ridge_coefficient_paths.png`, `ridge_coefficients.csv`, `ridge_path.csv`. Çıktı klasörü tanımlı değilse "Set the output directory in the Data tab first." uyarısı çıkar.

4.10 Çıktılar ve sonraki adıma geçenler

- `state["var_names"]`: onaylanan sürekli değişkenler, doğal ad sırasıyla. ③ Models, ⑤ Future Scenarios (projeksiyon ve yenilik haritaları) ve ⑧ Report bunu kullanır.
- `state["cat_selected"]` / `state["cat_excluded"]`: kategorik seçim; kategorik raster yoksa `None` ve boş liste.
- `figures/` ve (istenirse) `advanced_analysis/` dosyaları (§4.8, §4.9.6).
- Sekmedeki eşikler ve yöntem, rapora ve ODMAP'in "Multicollinearity" öğesine yazılır.
- Kayıt/öngörücü uyarısı (varsa) `sample_warnings` listesine eklenir ve rapora geçer.

4.11 Uzman soruları

S1. Adım adım VIF seçimi usdm::vifstep ile aynı sonucu veriyor mu? Evet, aynı girdide aynı sonucu verir. İki yöntem de kalan kümenin VIF'ini her adımda yeniden hesaplar, en büyüğü eşikten büyükse onu çıkarır ve en büyük VIF eşiğe eşit ya da altında kalınca durur (`sdm_core.py:427-434`; usdm'de `th = 10`). Bu kitapta 5000 hücrelik bir WorldClim matrisinde karşılaştırılmıştır: çıkarma sırası 11 değişkende aynı, kalan sekiz değişkenin VIF'leri on dördüncü ondalığa kadar aynıdır (§4.6.2). VIF de usdm'deki gibi kesişimli doğrusal regresyonun R^2'sinden hesaplanır (`sdm_core.py:275-295`).

S2. VIF'in 999'a sabitlenmesi çıkarma sırasını değiştirebilir mi? Değiştirebilir. $R^2 > 0{,}9999$ olan her değişkenin VIF'i 999 olur (`sdm_core.py:292-293`); aynı adımda birden çok değişken bu durumdaysa `idxmax()` doğal sıradaki ilkini çıkarır (`sdm_core.py:430`). usdm gerçek VIF'i kullandığı için o adımda gerçekten en büyük olanı çıkarır. Örnek koşuda ilk iki adımda bio_4 ve bio_5 999 ile çıkarılmıştır. Doğrulama matrisinde bu durum sonucu değiştirmemiştir; ama sabitlenmiş eşitliklerin olduğu bir veride son küme usdm'den farklı olabilir. Fark, yalnız aşırı yüksek VIF'li değişkenlerin hangisinin önce gideceğiyle ilgilidir; kesin aynı sonuç gerekiyorsa en açık türetilmiş değişkenler (bio_7 gibi) önceden elle çıkarılabilir.

S3. VIF ve korelasyon neden raster hücrelerinde değil de varlık + sözde yokluk havuzunda hesaplanıyor? Çünkü model bu noktalarla eğitilir; çoklu doğrusallık modelin gördüğü veride önemlidir (`tabs/tab_vif.py:349-353`). Bunun sonucu olarak sonuçlar erişilebilir alanın genişliğine ve PA sayısına bağlıdır (§4.3). `VIF + Correlation` havuzunda her varlık PA seti sayısı kadar yinelenir, `Advanced Analysis` ise yinelenenleri atar (`tabs/tab_vif_advanced.py:356-363`); bu yüzden iki alt sekmedeki örneklem tam aynı değildir. Bütün alan üzerinde bir korelasyon isteyen kullanıcı, R'de usdm::vifstep'i doğrudan rasterlara uygulayabilir.

S4. `Recommended` neden tek değişkenli AUC'yi hesaba katmıyor? Kodun açıklaması: tek değişkenli ayırt etme gücü, bu örneklemde gerçek etkenle korelasyonu sayesinde iyi görünen bir değişkeni öne çıkarır; böyle bir değişken başka bölgeye ya da iklime aktarıldığında başarısız olabilir (`sdm_core.py:323-339`). Bu yüzden hem `Priority_Score` hem `Recommended` yalnız çoklu doğrusallığa dayanır ve AUC rozeti bilerek nötr renklidir

(`tabs/tab_vif.py:471-499`). Çiftten hangisinin bırakılacağına türün ekolojisi karar vermelidir (Dormann vd. 2013).

S5. Korelasyon adımı neden çiftten VIF'i yüksek olanı çıkarıyor; usdm::vifcor ile fark nedir? HABITUS önce VIF adımını uygular, sonra kalan kümede $|r| \geq$ eşik olan en güçlü çiftten, o anki kümede VIF'i büyük olanı çıkarır (`sdm_core.py:435-445`). usdm::vifcor ise önce korelasyon eşiğiyle çift çift çıkarır (çiftten öteki değişkenlerle daha çok ilişkili olanı), sonra VIF'e bakmaz. İki yöntemin son kümesi farklı olabilir. Varsayılan eşiklerle (VIF 10, $|r|$ 0,80) VIF adımından sonra yüksek korelasyonlu çift nadiren kalır; örnek koşuda hiç kalmamıştır.

S6. Eşikleri değiştirdim ama `Recommended` aynı öneriyi veriyor. Neden? Öneri, `variable_priority_ranking` en son çalıştığında hesaplanır. Sekme ilk açıldığında ① Data'nın ön hesabı gösterilir (korelasyon 0,70, Pearson, VIF 10; `sdm_core.py:3225-3227`). Korelasyon eşiğini değiştirmek yalnız çift tablosunu ve ısı haritasını yeniler (`tabs/tab_vif.py:880-884`); yöntem ve VIF eşiği hiçbir şeyi kendiliğinden yenilemez. Yeni bir öneri için `Run / Refresh Analysis`'e, sonra `Recommended`'a basılmalıdır.

S7. LASSO ve Ridge, R'deki glmnet ile aynı sonucu veriyor mu? Aynı tanımı kullanır (Friedman vd. 2010): binom aile, glmnet ölçeğinde λ, eşit sınıf ağırlıkları, standartlaştırılmış öngörücüler, katmanlı k katlı çapraz doğrulama, `lambda.min` ve `lambda.1se` (`tabs/tab_vif_advanced.py:63-128`). Programın test takımı, benzetilmiş bir veride iki ızgara noktasındaki katsayıları R glmnet ile karşılaştırır: Ridge için 2×10^{-4}, LASSO için 3×10^{-3} mutlak fark içinde ve LASSO'nun sıfırladığı değişkenler aynıdır (`tests/test_202_small_samples_lasso.py:81-98`). Çözücü liblinear'dır ve kesişim terimi çok küçük bir cezayla kestirilir (`intercept_scaling=100`); bu, glmnet'ten küçük farkların kaynağıdır. λ ızgarası glmnet'inkiyle (100 değer, `lambda.min.ratio`) aynı değildir: HABITUS 40 değer kullanır ve LASSO'da üç, Ridge'de dört büyüklük mertebesi iner.

S8. Çapraz doğrulama sapmasının en küçüğü ızgaranın ucuna düşerse ne olur? Program bunu ayrıca bildirmez; `lambda_min` ızgaranın en küçük değeri olur (`tabs/tab_vif_advanced.py:120`). Örnek koşuda LASSO'da ($\lambda_{\min} = 0{,}0003929 = \lambda_{\max}/1000$) ve Ridge'de (0,003929) böyle olmuştur (Şekil 4.4 ve 4.5'te kırmızı çizgi sol uçtadır). Bu durumda gerçek en küçük daha küçük bir λ'da olabilir; 1-SE kuralı ise sapma eğrisi düz olduğu için daha güvenilirdir. Bu analizler yalnız bilgi verdiği için model sonuçlarını etkilemez; ayrıntılı bir ceza seçimi gerekiyorsa `advanced_analysis/lasso_path.csv` incelenebilir ya da glmnet daha geniş bir ızgarayla çalıştırılabilir.

S9. Koşul sayısı neden bu kadar büyük, ve VIF ile ne ilişkisi var? κ, standartlaştırılmış matrisin en büyük ve en küçük tekil değerinin oranıdır (`tabs/tab_vif_advanced.py:421-431`). En küçük tekil değer sıfıra yaklaştıkça κ büyür; bu, en az bir değişkenin ötekilerin doğrusal bileşimine yaklaştığı anlamına gelir. VIF aynı durumu değişken başına gösterir. 19 biyoiklim değişkeninde bio_7 = bio_5 − bio_6 gibi tam türetmeler bulunduğundan κ milyonlar düzeyindedir (örnek koşuda 22 681 501,75). Seçilen kümeyle yeniden hesaplandığında κ'nın düşmesi seçimin işe yaradığını gösterir.

S10. Tek değişkenle model kurulabilir mi? Hayır. Tek değişkenle sıralama tablosu kurulur (VIF = 1; `sdm_core.py:356-368`) ama `Confirm` en az iki sürekli değişken ister (`tabs/tab_vif.py:1164-1167`); `Run / Refresh Analysis` de tek değişken yüklüyse ① Data'ya katman eklenmesini ister (`tabs/tab_vif.py:367-372`).

S11. Kategorik değişkenler çoklu doğrusallık denetimine neden girmiyor? VIF ve Pearson/Spearman korelasyonu sıralı sayısal değişken varsayar; kategorik kodların sayısal değeri anlamsızdır. Kategorik rasterlar bu yüzden ayrı listelenir ve seyrek sınıflar (3'ten az varlık) işaretlenir (`tabs/tab_vif.py:583-664`). Program kategorik ile sürekli değişkenler arasındaki ilişkiyi (örneğin bir arazi örtüsü sınıfının yükseltiyle örtüşmesi) ölçmez; bu kullanıcının denetlemesi gereken bir noktadır. `Advanced Analysis` kategorik rasterları kullanmaz; yalnız sürekli değişkenlerle çalışır (`tabs/tab_vif_advanced.py:343-354`).

S12. Bu sekmedeki seçim tekrarlanabilir mi? VIF, korelasyon, PCA ve koşul sayısı rastgelelik içermez; aynı PA setleri aynı sonucu verir. PA setleri ① Data'daki tohuma bağlıdır. LASSO ve Ridge'in çapraz doğrulama katları koşu tohumuyla (`formatter.random_seed`) karıştırılır (`tabs/tab_vif_advanced.py:600-604`, `108`). Seçilen değişkenler, eşikler ve yöntem rapora, ODMAP'e ve `figures/variable_priority_ranking.csv`'ye yazılır; `Recommended` ile yapılan çıkarmalar sırasıyla oturum günlüğündedir.

4.12 Sık hatalar ve tuzaklar

- **Eşiği değiştirip `Run / Refresh Analysis`'e basmamak.** Öneri eski eşiklerle kalır (§4.4.1). Önce eşikleri ayarlayın, sonra `Run / Refresh Analysis`, sonra `Recommended`.
- **`Recommended`'ı son karar saymak.** İpucu açıkça "A starting point" der. Önerilen kümeyi türün ekolojisiyle karşılaştırın; örneğin örnek koşuda önerilen kümede yıllık ortalama sıcaklık (bio_1) yoktur, onun yerine bio_11 (en soğuk çeyreğin ortalama sıcaklığı) kalmıştır.
- **Tam küme VIF rozetine bakıp önerilen bir değişkeni çıkarmak.** Rozet bütün değişkenlerin varlığındaki VIF'i gösterir; önerilen kümedeki VIF `variable_priority_ranking.csv`'nin `VIF_in_selection` sütunundadır (örnekte bio_11: tam kümede ∞, önerilen kümede 2,85).
- **Çift tablosundaki kırmızı AUC'yi "bunu at" diye okumak.** AUC renklendirmesi yalnız iki AUC'yi karşılaştırır; düşük tek değişkenli AUC bir değişkeni gereksiz yapmaz (§4.5.1).
- **Seçimi değiştirip onaylamamak.** ③ Models son onaylanan kümeyi kullanır. Modeller çalıştırıldıktan sonra yeniden onay her şeyi siler ("Re-confirm Variable Selection").
- **`Advanced Analysis`'in önerilerini listeye uygulandı sanmak.** PCA ve LASSO yalnız bilgi verir; değişken listesindeki işaretler değişmez.
- **Kayıt/öngörücü uyarısını görmezden gelmek.** "Below about 10 per predictor …" uyarısı çıkıyorsa daha az değişken seçin ya da ③ Models'te daha basit algoritmalar (GLM, doğrusal/karesel özellikli MaxEnt) kullanın.

4.13 Kaynaklar

Barbet-Massin, M., Jiguet, F., Albert, C.H., Thuiller, W., 2012. Selecting pseudo-absences for species distribution models: how, where and how many? Methods in Ecology and Evolution 3, 327–338. https://doi.org/10.1111/j.2041-210X.2011.00172.x

Belsley, D.A., Kuh, E., Welsch, R.E., 1980. Regression Diagnostics: Identifying Influential Data and Sources of Collinearity. Wiley, New York. https://doi.org/10.1002/0471725153

Dormann, C.F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., vd., 2013. Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. Ecography 36, 27–46. https://doi.org/10.1111/j.1600-0587.2012.07348.x

Friedman, J., Hastie, T., Tibshirani, R., 2010. Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software 33(1), 1–22. https://doi.org/10.18637/jss.v033.i01

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning, 2. baskı. Springer, New York. https://doi.org/10.1007/978-0-387-84858-7

Naimi, B., Hamm, N.A.S., Groen, T.A., Skidmore, A.K., Toxopeus, A.G., 2014. Where is positional uncertainty a problem for species distribution modelling? Ecography 37, 191–203. https://doi.org/10.1111/j.1600-0587.2013.00205.x

O'Brien, R.M., 2007. A caution regarding rules of thumb for variance inflation factors. Quality & Quantity 41, 673–690. https://doi.org/10.1007/s11135-006-9018-6

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B 58, 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

5. Modeller I: algoritmalar ve ayarları (③ Models → Algorithm Settings)

Sekme: ③ Models + Current Map → Algorithm Settings · Kod: `tabs/tab_models.py` (arayüz), `sdm_core.py` `build_model`, `SDMModeler` (hesap), `sdm_niche.py` (BIOCLIM, Domain, ESM), `sdm_tuning.py` (ayarlanabilir parametreler)

5.1 Bu sekme ne yapar

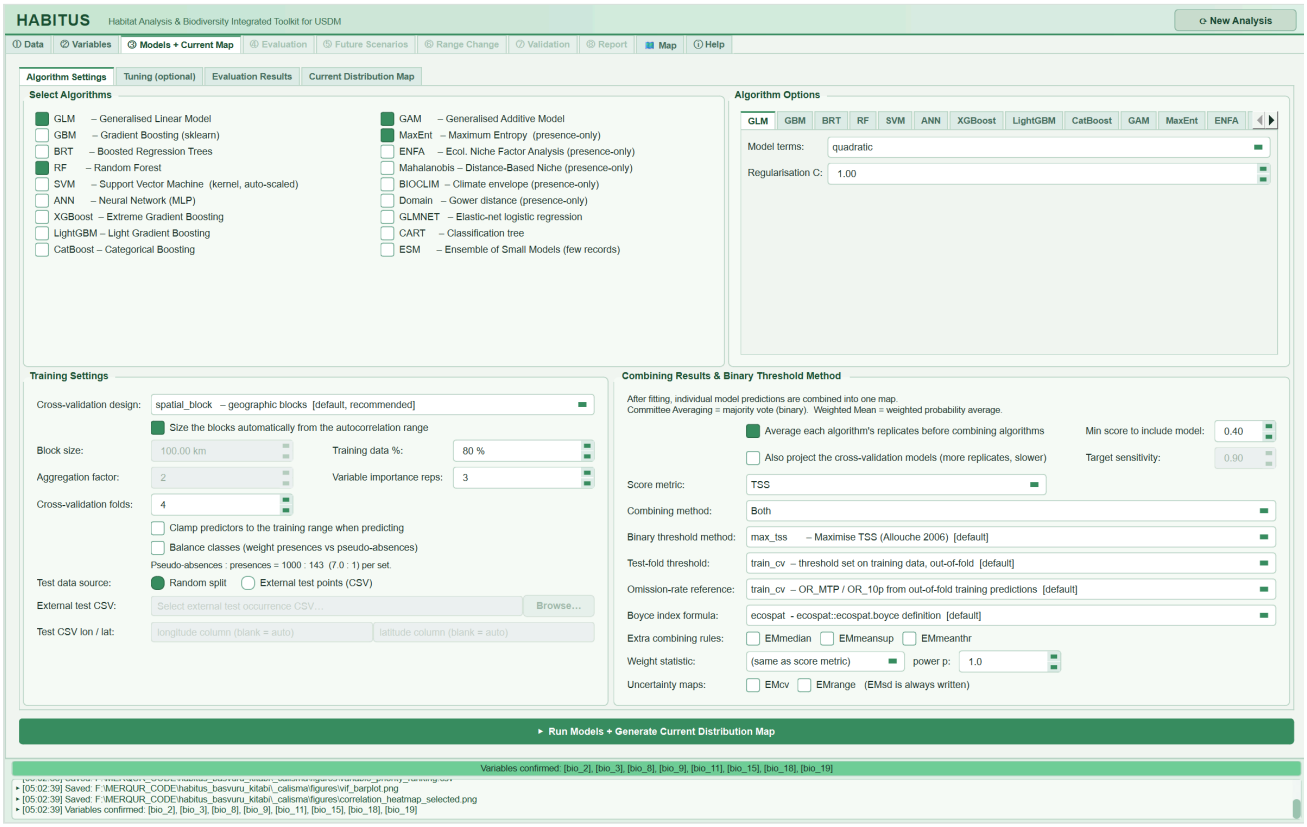
`Algorithm Settings` (algoritma ayarları) alt sekmesi, modellemenin "hangi yöntemle" sorusunu yanıtlar. Burada 18 algoritmadan hangilerinin çalıştırılacağı işaretlenir ve her birinin ayarları (hiperparametreleri) girilir. Hiperparametre, modelin veriden öğrenmediği, kullanıcının önceden belirlediği ayardır: ağaç sayısı, ceza katsayısı, katman sayısı gibi.

İş akışındaki yeri şöyledir:

- **Girdi:** ① Data sekmesinin ürettiği sözde yokluk veri kümeleri (her tekrar için varlık + sözde yokluk satırları), MaxEnt arka planı, isteğe bağlı yalnız-varlık arka planı ve ② Variables sekmesinde seçilen öngörücüler (sürekli ve kategorik). Bu iki adım tamamlanmadan çalıştırma düğmesi "Complete ① Data and ② Variables steps first." iletisini verir (`tabs/tab_models.py:1394-1395`).
- **İşlem:** ▶ `Run Models + Generate Current Distribution Map` düğmesine basıldığında işaretli her algoritma, her sözde yokluk kümesinde, çapraz doğrulama katlarında ve tam veriyle eğitilir (`sdm_core.py:5352` `SDMModeler.run`). Bu sayfadaki ayarlar `_collect_algo_opts` ile toplanıp `SDMModeler`'e `algo_options` olarak geçer (`tabs/tab_models.py:1142-1209`, `1569-1573`).
- **Çıktı:** eğitilmiş modeller, değerlendirme skorları, değişken önemleri ve güncel dağılım haritası. Bunlar `Evaluation Results` ve `Current Distribution Map` alt sekmelerinde ve ④ Evaluation sekmesinde görülür. Kullanılan bütün ayarlar `habitus_run_config.json` dosyasının `settings.models.algorithm_options` alanına yazılır.

Çapraz doğrulama tasarımı, eşik ve topluluk (ensemble) ayarları aynı sayfanın alt yarısındadır; onlar Bölüm 6'da anlatılır. Bu bölüm yalnız algoritma seçimini ve algoritma ayarlarını kapsar.

5.2 Ekranın düzeni



Şekil 5.1. Algorithm Settings alt sekmesi: solda Select Algorithms kutusu (örnek koşuda GLM, RF, GAM ve MaxEnt işaretli), sağda Algorithm Options paneli (GLM sayfası açık); altta Training Settings ve Combining Results & Binary Threshold Method grupları, en altta çalıştırma düğmesi.

Sayfa dört bölgeden oluşur:

1. **`Select Algorithms`** (algoritma seçimi; sol üst): 18 onay kutusu, iki sütunda dokuzar tane (`tabs/tab_models.py:125-150`). Bkz. §5.3.
2. **`Algorithm Options`** (algoritma seçenekleri; sağ üst): her algoritma için bir sekme. Sekme çubuğu 18 sekmeyi sığdıramadığı için sağdaki ok düğmeleriyle kaydırılır (`tabs/tab_models.py:173-194`). Altı ve daha fazla satırı olan sayfalar (BRT, XGBoost, LightGBM, CatBoost, MaxEnt, ESM) iki sütuna dizilir (`tabs/tab_models.py:197-201`). Her sekme §5.6–§5.23'te kendi alt bölümünde anlatılır.
3. **`Training Settings`** ve **`Combining Results & Binary Threshold Method`** (alt yarı): Bölüm 6.
4. **Çalıştırma düğmesi:** `▶ Run Models + Generate Current Distribution Map`.

Örnek koşunun verisi *Abies nordmanniana*'dır (626 kayıt; inceltme sonrasında 143 varlık; WorldClim 2.1 biyoiklim, 2,5′; ② Variables'ta seçilen 8 değişken: bio_2, bio_3, bio_8, bio_9, bio_11, bio_15, bio_18, bio_19; 2 sözde yokluk kümesi × 1000 nokta; MaxEnt arka planı 10 000 nokta; tohum 42). Bölümdeki sayılar bu koşunun `habitus_run_config.json`, `evaluation_scores.csv` ve oturum günlüğünden alınmıştır.

5.3 Algoritma seçimi (Select Algorithms)

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`GLM – Generalised Linear Model`	Lojistik regresyon	açık	açık / kapalı	`tabs/tab_models.py:129`
`RF – Random Forest`	Rastgele orman	açık	açık / kapalı	`tabs/tab_models.py:132`
Öteki 16 kutu (`GBM`, `BRT`, `SVM`, `ANN`, `XGBoost`, `LightGBM`, `CatBoost`, `GAM`, `MaxEnt`, `ENFA`, `Mahalanobis`, `BIOCLIM`, `Domain`, `GLMNET`, `CART`, `ESM`)	İlgili algoritmayı çalıştırır	kapalı	açık / kapalı	`tabs/tab_models.py:130-146`

Programın açılışında yalnız GLM ve RF işaretlidir. Gereken paketi kurulu olmayan algoritmanın kutusu kapatılır, devre dışı bırakılır, adının sonuna `[not installed]` eklenir ve ipucunda kurulum komutu gösterilir (ör. "Not installed — pip install pygam"; `tabs/tab_models.py:152-168`). Paket gerektirenler: XGBoost (`xgboost`), LightGBM (`lightgbm`), CatBoost (`catboost`), GAM (`pygam`), MaxEnt (`elapid`); öteki 13 algoritma scikit-learn ya da numpy ile çalışır (`sdm_core.py:127-146`). Hiçbir kutu işaretli değilse çalıştırma "Select at least one algorithm." iletisiyle durur (`tabs/tab_models.py:1397-1400`).

> **Not:** Algoritmaların hangi sırayla işaretlendiği ya da öngörücülerin hangi sırayla eklendiği sonucu değiştirmez. Sütunlar her koşuda önce sürekli, sonra kategorik ve her grup kendi içinde doğal ad sırasıyla dizilir (`sdm_core.py:3892-3899`).

5.4 Bütün algoritmalar için ortak kurallar

5.4.1 Hangi satırlarla eğitilir: varlık-yokluk ve yalnız-varlık

HABITUS'ta algoritmalar eğitimde kullandıkları satırlara göre üç gruba ayrılır:

- **Varlık–sözde yokluk (sınıflandırıcı) modeller:** GLM, GLMNET, GAM, GBM, BRT, RF, CART, SVM, ANN, XGBoost, LightGBM, CatBoost ve GLM/RF tabanlı ESM. Sözde yokluk kümesinin bütün satırlarını (varlık = 1, sözde yokluk = 0) kullanır (`sdm_core.py:4480-4498`).
- **Varlık + arka plan:** MaxEnt. Eğitim katındaki varlıkları ve ① Data'da ayrıca örneklenen MaxEnt arka planını kullanır; sözde yokluklar MaxEnt'in eğitimine girmez (`sdm_core.py:4595-4603`, `tabs/tab_models.py:1580`). ENFA da küresel ortam dağılımını kestirmek için, varsa yalnız-varlık arka planını kullanır (`sdm_core.py:4079-4092`).
- **Yalnız varlık:** Mahalanobis, BIOCLIM ve Domain yalnız varlık satırlarından uydurulur (`sdm_core.py:4227-4228`, `4281-4282`).

Değerlendirme ise grup ne olursa olsun aynıdır: test katındaki varlıklar ve sözde yokluklar üzerinden AUC, TSS, Boyce ve ötekiler hesaplanır. Yalnız-varlık yöntemlerinde test katında hiç varlık kalmazsa eğitim katındaki varlıkların yaklaşık beşte biri teste taşınır (`sdm_core.py:5455-5468`; tam model için `5686-5696`).

5.4.2 Girdinin ölçeklenmesi

Ölçek duyarlı yöntemlerde standartlaştırma (ortalama 0, standart sapma 1) programın içinde, yalnız eğitim satırlarıyla öğrenilerek yapılır; kullanıcının önceden ölçeklemesi gerekmez:

- GLM ve GLMNET: önce terimler (doğrusal, kare, etkileşim) üretilir, sonra bu terimler standartlaştırılır (`sdm_core.py:3488-3505`).
- SVM ve ANN: öngörücüler standartlaştırılır (`sdm_core.py:3562-3581`).
- Ağaç yöntemleri (GBM, BRT, RF, CART, XGBoost, LightGBM, CatBoost) ölçekten etkilenmez; ölçekleme yapılmaz.
- MaxEnt özellikleri elapid'in dönüştürücüsüyle üretilir. GAM, ENFA, Mahalanobis, BIOCLIM ve Domain ham değerlerle çalışır; ENFA ve Mahalanobis kovaryans matrisiyle, BIOCLIM yüzdeliklerle, Domain varlık aralığına bölünmüş farklarla ölçeği kendisi giderir.

5.4.3 Kategorik değişkenler

Kategorik rasterlar (arazi örtüsü, bakı sınıfı gibi) modele tamsayı sınıf kodu olarak gelir; her algoritma onları şöyle kullanır:

Algoritma	Kategorik değişkenin işlenmesi	Kod
GLM, GLMNET, SVM, ANN	Eğitimde görülen her sınıf için bir 0/1 sütunu (one-hot); kodlar karesi alınmaz, ölçeklenmez, çarpılmaz. Eğitimde görülmemiş sınıf o değişken için bütün sütunlarda 0 alır	`sdm_core.py:3364-3443`
GAM	Faktör terimi `f()`, sınıf başına bir katsayı; tek sınıflı değişken atlanır; görülmemiş sınıfın etkisi 0 sayılır	`sdm_core.py:3320-3361`, `4525-4549`
MaxEnt	elapid'in kategorik özellik sınıfı (one-hot, `beta_categorical` ile cezalı); görülmemiş sınıf 0	`sdm_core.py:4606-4612`, `4712-4718`, `4760-4770`
GBM, BRT, RF, CART, XGBoost, LightGBM, CatBoost	Tamsayı kodu sayısal bir sütun gibi bölünür	`sdm_core.py:3476-3479`
ENFA, Mahalanobis, BIOCLIM, Domain	Kategorik değişkenler dışarıda bırakılır; günlüğe "... is an envelope/distance model for continuous predictors: the categorical variable(s) ... are excluded from it." yazılır	`sdm_core.py:4010-4036`
ESM	Kategorik değişkenler çiftlere girmez	`sdm_core.py:4389-4393`

> **Dikkat:** Ağaç yöntemlerinde sınıf kodu sıralı bir sayı gibi bölünür: kod 3 ile 4 "yakın", 3 ile 12 "uzak" sayılır. Kodların sırası ekolojik bir anlam taşımıyorsa ağaç, sınıfları ancak birkaç bölünmeyle ayırabilir. CatBoost'un kendi kategorik işleme yeteneği HABITUS'ta kullanılmaz.

5.4.4 Tekrarlanabilirlik ve rastgele tohum

Rastgelelik içeren her öğrenici (RF, GBM, BRT, CART, SVM, ANN, GLMNET, XGBoost, LightGBM, CatBoost, MaxEnt, ESM katları) ① Data'daki `Random seed` (varsayılan 42) ile tohumlanır (`sdm_core.py:115-124`, `3474-3475`, `4643`). Aynı veri, aynı ayarlar ve aynı tohumla iki koşu aynı modelleri üretir. Permütasyon önemi de (PA kümesi, kat, algoritma) üçlüsüne özgü ama tohumdan türetilen bir üreteçle hesaplanır (`sdm_core.py:4847-4853`). Kullanılan tohum `habitus_run_config.json` içinde `random_seed` alanındadır.

5.4.5 Sınıf dengesi ve kırpma (Training Settings'ten gelen iki ayar)

`Balance classes (weight presences vs pseudo-absences)` (sınıfları dengele) ve `Clamp predictors to the training range when predicting` (tahminde öngörücüleri eğitim aralığına kırp) Training Settings grubundadır ve Bölüm 6'da anlatılır; ama algoritmaya göre farklı davrandıkları için burada özetlenir. Denge açıkken (varsayılan kapalı) her algoritma sınıf ağırlığını şöyle alır (`sdm_core.py:3670-3716`):

Algoritma	Dengeleme biçimi
GLM, RF, SVM, GLMNET, CART, LightGBM	`class_weight='balanced'`
CatBoost	`auto_class_weights='Balanced'`
GBM, BRT	dengeli örnek ağırlıkları (`sample_weight`)
XGBoost	`scale_pos_weight` = sözde yokluk sayısı / varlık sayısı
ANN	dengeli örnek ağırlıkları (kurulu scikit-learn bunu destekliyorsa)
ESM	yalnız GLM ya da RF tabanlı küçük modellerde
GAM, MaxEnt, ENFA, Mahalanobis, BIOCLIM, Domain	desteklenmez; ayar bu algoritmalarda etkisizdir

Kırpma açıkken her tahminden önce öngörücüler bütün sözde yokluk kümelerinin min–maks aralığına kırpılır; MaxEnt'e ayrıca elapid'in `clamp=True` ayarı açıkça verilir (`sdm_core.py:4650-4655`, `4929-4962`).

5.4.6 Ayar (tuning) alt sekmesiyle ilişki

`Tuning (optional)` alt sekmesi bazı algoritmaların parametrelerini çapraz doğrulamalı ızgara aramasıyla dener ve seçilen değeri `Apply selected setting` ile bu sayfaya yazar. Bu sayfaya yazılabilen parametreler (`tabs/tab_models.py:1212-1229`, `1231-1259`) ve ayar sekmesinin varsayılan ızgaraları (`sdm_tuning.py`) şunlardır:

Algoritma	Bu sayfaya yazılan parametreler	Varsayılan ızgara (kaynak)
MaxEnt	`Feature types`, `Regularisation multiplier (β / RM)`	rm 0,5–4,0 adım 0,5; fc L, LQ, H, LQH, LQHP (`sdm_tuning.py:502-552`)
GBM	`n_estimators`, `learning_rate`, `subsample`, `min_samples_leaf`, `max_depth`	flexsdm `tune_gbm`: ağaç 20–500, öğrenme hızı 0,01/0,1/0,3, yaprak 1–10, 20, 30 (`sdm_tuning.py:626-630`)
RF	`n_estimators`, `min_samples_leaf`, `max_features`	flexsdm `tune_raf`: mtry 2…öngörücü sayısı, ntree 200–1200 (`sdm_tuning.py:633-634`, `787-793`)
SVM	`C`, `gamma`	flexsdm `tune_svm`: C 1–16, sigma 0,001–0,2 → gamma aynı değer (`sdm_tuning.py:636-637`, `654-675`)
ANN	`alpha (L2)`, `Hidden layers`	flexsdm `tune_net`: tek katman 2–8 nöron, decay 0,01–10 → alpha = 2 × decay (`sdm_tuning.py:639-640`, `678-689`)
XGBoost	`n_estimators`, `max_depth`, `learning_rate`	HABITUS'un ızgarası: 100/300/500; 3/6/9; 0,01/0,05/0,1 (`sdm_tuning.py:779-781`)
LightGBM	`n_estimators`, `num_leaves`, `learning_rate`	HABITUS'un ızgarası: 100/300/500; 15/31/63; 0,01/0,05/0,1 (`sdm_tuning.py:782-784`)
GLMNET	`C (inverse penalty)`, `l1_ratio (alpha)`	C 0,01/0,1/1/10; l1_ratio 0–1 adım 0,25 (`sdm_tuning.py:837-838`)
CART	`max_depth`, `min_samples_leaf`, `ccp_alpha`	2/3/5/8; 1/5/10/20; 0/0,002/0,01 (`sdm_tuning.py:839-841`)
GAM	`n_splines`, `lam (smoothing)`	5/10/15/20; 0,1/0,6/2/10 (`sdm_tuning.py:842-843`)

GLM, BRT, CatBoost, ENFA, Mahalanobis, BIOCLIM, Domain ve ESM ayar sekmesinde yoktur; onların değerleri yalnız bu sayfadan girilir. Ayar aramasının kendisi Bölüm 6'da anlatılır.

5.5 Algoritma seçme rehberi

Aşağıdaki tablo, kodda görülen yapıdan ve örnek koşudan çıkarılmış bir başlangıç rehberidir; bir yöntemin her veri kümesinde ötekinden iyi olacağını söylemez. "Hız" sütunundaki sayılar örnek koşuda bir sözde yokluk kümesinin 4 çapraz doğrulama katının süresidir (iç katlarla birlikte; günlükten).

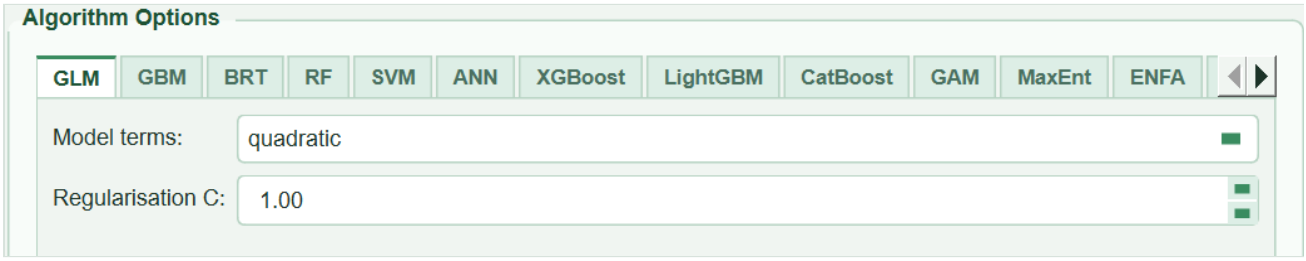
Algoritma	Veri türü	Hız	Yorumlanabilirlik	Aşırı uyum eğilimi (varsayılanla)	Ne zaman
GLM	varlık–sözde yokluk	çok hızlı (0,7 s)	yüksek (katsayılar)	düşük–orta (L2 cezalı, karesel)	her çalışmada temel model
GLMNET	varlık–sözde yokluk	hızlı	yüksek	düşük (elastik ağ)	çok öngörücü, az kayıt
GAM	varlık–sözde yokluk	orta (10 s)	yüksek (tepki eğrileri)	orta	doğrusal olmayan, düzgün tepkiler

Algoritma	Veri türü	Hız	Yorumlanabilirlik	Aşırı uyum eğilimi (varsayılanla)	Ne zaman
MaxEnt	varlık + arka plan	yavaş (165 s)	orta–yüksek	β ve özellik sınıflarına bağlı	yalnız-varlık verisi, karşılaştırılabilirlik
RF	varlık–sözde yokluk	orta (21 s)	düşük	yüksek eğitim uyumu (eğitim AUC 1,000)	çok kayıt, karmaşık etkileşimler
GBM	varlık–sözde yokluk	orta	düşük–orta	düşük (kütük ağaçlar)	toplamsal esnek model
BRT	varlık–sözde yokluk	orta–yavaş	düşük–orta	orta	Elith ve ark. (2008) çizgisinde boosting
XGBoost, LightGBM, CatBoost	varlık–sözde yokluk	hızlı–orta	düşük	yüksek (derin ağaçlar)	çok kayıt (yüzlerce), karşılaştırma
CART	varlık–sözde yokluk	çok hızlı	çok yüksek (tek ağaç)	kararsız, basamaklı yüzey	açıklama amaçlı
SVM	varlık–sözde yokluk	n büyüdükçe yavaşlar	düşük	C ve gamma'ya bağlı	karşılaştırma
ANN	varlık–sözde yokluk	orta	çok düşük	yüksek (100+50 nöron)	çok kayıt
ENFA	varlık + arka plan	çok hızlı	orta (eksenler)	düşük	yalnız-varlık, kaba niş
Mahalanobis	yalnız varlık	çok hızlı	orta	düşük	eliptik niş varsayımı
BIOCLIM	yalnız varlık	çok hızlı	çok yüksek	düşük; uç kayda duyarlı	iklim zarfı, karşılaştırma
Domain	yalnız varlık	hızlı (varlık sayısıyla artar)	yüksek	düşük; tek kayda duyarlı	benzerlik tabanlı karşılaştırma
ESM	varlık–sözde yokluk (çiftler)	çift sayısıyla artar	orta	düşük (iki değişkenli modeller)	çok az kayıt (yaklaşık < 30)

Örnek koşuda (143 varlık, 8 değişken) dört algoritmanın çapraz doğrulama ortalamaları: GLM AUC 0,889, TSS 0,687, Boyce 0,607; MaxEnt 0,895, 0,672, 0,605; RF 0,892, 0,566, 0,677; GAM 0,872, 0,611, 0,653. Eğitim ile test AUC'si arasındaki fark (dAUC) GLM'de 0,031, MaxEnt'te 0,055, GAM'de 0,087, RF'de 0,108'dir; eğitim verisine en sıkı uyan RF'dir.

> **İpucu:** Tek bir "en iyi" algoritma aramak yerine farklı ailelerden (bir regresyon, bir ağaç topluluğu, MaxEnt) birkaç algoritma çalıştırıp topluluk haritasıyla birleştirmek yaygın uygulamadır; birleştirme Bölüm 6'dadır.

5.6 GLM: genelleştirilmiş doğrusal model



Şekil 5.2. GLM sekmesi: Model terms (quadratic) ve Regularisation C (1.00).

Aile ve uygulama. İkili (lojistik) regresyon: varlık olasılığının logiti öngörücülerin doğrusal bir birleşimidir. scikit-learn `LogisticRegression` (çözücü `lbfgs`, en çok 2000 yineleme) ile, önce terim üretimi sonra standartlaştırma yapan bir boru hattı içinde uydurulur (`sdm_core.py:3481-3492`). Veri türü varlık–sözde yokluk. Ceza, standartlaştırılmış terimlerin katsayılarına uygulanan L2 (ridge) cezasıdır.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Model terms:`	Hangi terimlerin modele gireceği	`quadratic`	`linear`, `quadratic`, `interactions`	`tabs/tab_models.py:721-735`; `sdm_core.py:3300-3317`
`Regularisation C:`	Ters ceza gücü (L2)	1.00	0.001 – 1 000 000	`tabs/tab_models.py:736-745`; `sdm_core.py:3482`

5.6.1 `Model terms` (model terimleri)

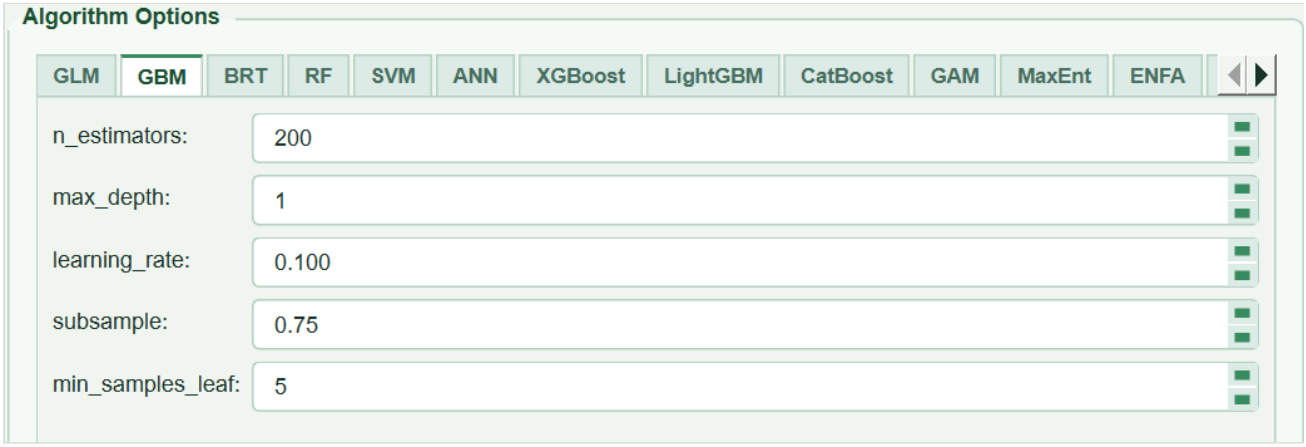
- **Ne işe yarar.** `linear` her öngörücü için yalnız x terimini, `quadratic` x ve x^2 terimlerini, `interactions` bunlara ek olarak bütün ikili çarpımları ($x_i x_j$) üretir (`sdm_core.py:3300-3317`). MaxEnt özellik sınıflarıyla karşılaştırılırsa sırasıyla L, LQ ve LQP'dir. Terim sayısı p öngörücüde p, $2p$ ve $p + p(p+1)/2$'dir; 8 öngörücüde 8, 16 ve 44.
- **Ne zaman, nasıl ayarlanır.** `quadratic` bir değişken boyunca tek tepeli (optimumlu) tepkiye izin verir; ekolojik nişin klasik varsayımıdır ve çoğu çalışma için uygundur. `linear` yalnız tekdüze artan ya da azalan tepki verir; çok az kayıtta ya da yorumu basit tutmak için seçilir. `interactions` en az birkaç yüz varlık kaydı gerektirir; ipucundaki on-olay-bir-değişken kuralına göre 8 öngörücüde yaklaşık 440 kayıt.
- **Sonuca etkisi.** Terim arttıkça eğitim uyumu artar, tepki eğrileri bükülür ve projeksiyonlarda kalibrasyon aralığı dışına taşma (ekstrapolasyon) riski büyür; kare terim aralık dışında hızla düşen ya da yükselen eğri verir.
- **Dikkat.** Kategorik değişkenlerin kodlarına terim uygulanmaz; yalnız sürekli değişkenlere uygulanır (`sdm_core.py:3364-3378`).

5.6.2 `Regularisation C` (düzenleme C'si)

- **Ne işe yarar.** scikit-learn'ün amaç işlevi $C \sum_i \ell_i + \frac{1}{2}\|\beta\|^2$'dir: C küçüldükçe ceza büyür ve katsayılar sıfıra doğru çekilir. Ceza standartlaştırılmış terimlere uygulandığından bütün terimlere eşit ölçüde etki eder.
- **Ne zaman, nasıl ayarlanır.** Terim sayısı kayıt sayısına göre büyükse (`interactions`, çok öngörücü) C küçültülür (0,1 ya da 0,01). R'deki `glm()` cezasızdır (biomod2, sdm ve flexsdm GLM'leri de); onların sonucunu yeniden üretmek için C = 1e6 girilir (ipucu, `tabs/tab_models.py:741-743`).
- **Sonuca etkisi.** Ceza bir kayıp toplamına göre tanımlandığından satır sayısı arttıkça göreli etkisi azalır: örnek koşuda bir eğitim katı yaklaşık 860 satırdır (1143 satırın dörtte üçü) ve C = 1 hafif bir cezadır. İpucu, *Pinus brutia* örneğinde varsayılan cezanın çapraz doğrulama AUC'sini 0,804'ten (cezasız) 0,824'e çıkardığını bildirir.
- **Dikkat.** GLM ayar sekmesinde yoktur; C için arama yapılacaksa aynı ceza ailesini taşıyan GLMNET (l1_ratio = 0) kullanılabilir.

R ile eşdeğerlik. `Model terms` = `linear` ve C = 1e6 ile model R'de `glm(y ~ x1 + x2 + …, family = binomial)` ile aynı amaç işlevini çözer; `quadratic` için `y ~ x + I(x^2)` biçimi karşılıktır. biomod2'nin GLM'i terimleri AIC'ye dayalı adımsal seçimle eler; HABITUS'ta adımsal seçim yoktur, onun yerine ceza vardır.

5.7 GBM: gradyan artırma (scikit-learn)



Şekil 5.3. GBM sekmesi: n_estimators 200, max_depth 1, learning_rate 0.100, subsample 0.75, min_samples_leaf 5.

Aile ve uygulama. Gradyan artırmalı karar ağaçları (boosting): her yeni ağaç, önceki ağaçların toplamının hatasını (log-kaybın gradyanını) düzeltir. scikit-learn `GradientBoostingClassifier` (`sdm_core.py:3516-3529`). Varlık–sözde yokluk. Ölçekleme gerekmez. GBM ile BRT aynı uygulamayı kullanır; yalnız varsayılanları ve aralıkları farklıdır.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`n_estimators:`	Ağaç sayısı	200	10 – 5000 (adım 100)	`tabs/tab_models.py:750`; `sdm_core.py:3524`
`max_depth:`	Ağaç derinliği (gbm: `interaction.depth`)	1	1 – 10	`tabs/tab_models.py:751-752`; `sdm_core.py:3525`
`learning_rate:`	Her ağacın katkısını küçülten katsayı (gbm: `shrinkage`)	0.100	0.001 – 1.0	`tabs/tab_models.py:753`; `sdm_core.py:3526`

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`subsample:`	Her ağaç için çekilen satır oranı (gbm: `bag.fraction`)	0.75	0.1 – 1.0 (adım 0.05)	`tabs/tab_models.py:756-758`; `sdm_core.py:3527`
`min_samples_leaf:`	Yapraktaki en az satır (gbm: `n.minobsinnode`)	5	1 – 200	`tabs/tab_models.py:754-755`; `sdm_core.py:3528`

5.7.1 `n_estimators` ve `learning_rate`

- **Ne işe yarar.** Model $F = \sum_{m=1}^{M} \nu\, h_m(x)$ biçimindedir; M ağaç sayısı, ν öğrenme hızıdır. İkisi birlikte modelin toplam "öğrenme miktarını" belirler.
- **Ne zaman, nasıl ayarlanır.** Öğrenme hızı küçültülürse aynı uyum için ağaç sayısı orantılı artırılmalıdır (0,1 → 0,01 ise yaklaşık 10 kat ağaç). Elith ve ark. (2008) en az 1000 ağaca ulaşacak bir öğrenme hızı önerir. En iyi çift `Tuning (optional)` sekmesinde flexsdm ızgarasıyla aranabilir.
- **Sonuca etkisi.** Çok ağaç + yüksek hız aşırı uyuma, az ağaç + düşük hız eksik uyuma götürür. Süre ağaç sayısıyla doğrusal artar.
- **Dikkat.** Ağaç sayısı çapraz doğrulamayla otomatik seçilmez (dismo `gbm.step`'teki gibi bir erken durdurma yoktur); girilen sayı aynen kullanılır.

5.7.2 `max_depth`

- **Ne işe yarar.** Her ağacın derinliği. 1, tek bölünmeli ağaç (kütük, "stump") demektir ve model öngörücülerin toplamsal bir işlevi olur, etkileşim içermez. 2 ve üstü değişken etkileşimlerine izin verir.
- **Ne zaman, nasıl ayarlanır.** Varsayılan 1, flexsdm `fit_gbm`'in `interaction.depth` = 1 varsayılanına yakındır (`sdm_core.py:3517-3522`). Etkileşimlerin önemli olduğu düşünülüyorsa ve kayıt sayısı yüksekse 2–3 denenir.
- **Sonuca etkisi.** İpucuna göre kütük ağaçlar yüzeyin hücrelerin çoğunda sıfıra çökmesini engeller ve aynı AUC'de daha yüksek Boyce verir (`tabs/tab_models.py:752`).
- **Dikkat.** gbm'in `interaction.depth`'i ile scikit-learn'ün `max_depth`'i 1'de aynı ağacı tanımlar; daha büyük değerlerde iki paketin ağaç büyütme biçimi farklı olduğundan sayılar birebir eşlenmez.

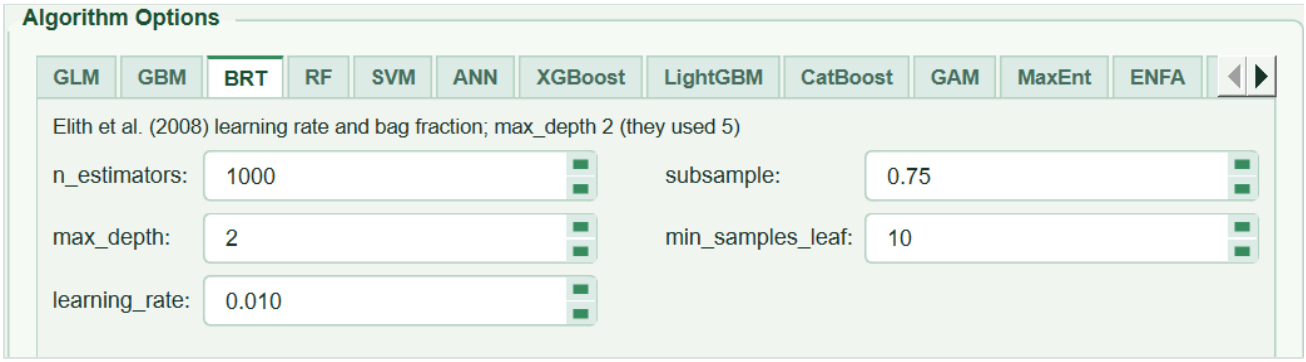
5.7.3 `subsample`

- **Ne işe yarar.** Her ağaç, eğitim satırlarının bu oranda rastgele bir alt kümesiyle kurulur (stokastik gradyan artırma).
- **Ne zaman, nasıl ayarlanır.** 0,5–0,75 aralığı yaygındır; 1,0 bütün satırları kullanır ve rastgeleliği kaldırır.
- **Sonuca etkisi.** 1'in altındaki değerler varyansı ve aşırı uyumu azaltır; sonuç tohuma bağlı hâle gelir (tohum sabit olduğu için tekrarlanabilir kalır).
- **Dikkat.** Çok küçük değerler az varlıklı veride bazı ağaçları neredeyse varlıksız bırakabilir.

5.7.4 `min_samples_leaf`

- **Ne işe yarar.** Bir yaprakta bulunabilecek en az satır sayısı; daha küçük yapraklar oluşturan bölünmeler yapılmaz.
- **Ne zaman, nasıl ayarlanır.** Varsayılan 5, gbm'in `n.minobsinnode` varsayılanına karşılık gelir. Gürültülü ya da az kayıtlı veride 10–20'ye çıkarılabilir.
- **Sonuca etkisi.** Büyüdükçe ağaçlar kabalaşır, yüzey düzgünleşir, aşırı uyum azalır.
- **Dikkat.** `max_depth` = 1'de her ağacın yalnız bir bölünmesi olduğundan etkisi sınırlıdır.

5.8 BRT: artırılmış regresyon ağaçları



Şekil 5.4. BRT sekmesi: Elith ve ark. (2008) notu; n_estimators 1000, max_depth 2, learning_rate 0.010, subsample 0.75, min_samples_leaf 10.

Aile ve uygulama. GBM ile aynı yöntem ve aynı scikit-learn sınıfı (`GradientBoostingClassifier`), ama Elith ve ark. (2008) rehberine göre seçilmiş varsayılanlarla: düşük öğrenme hızı, çok ağaç, %75 torbalama (`sdm_core.py:3530-3548`). Varlık–sözde yokluk.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`n_estimators:`	Ağaç sayısı	1000	100 – 10 000 (adım 100)	`tabs/tab_models.py:765`; `sdm_core.py:3543`
`max_depth:`	Ağaç derinliği (Elith ve ark.: `tree.complexity`)	2	1 – 10	`tabs/tab_models.py:766-767`; `sdm_core.py:3544`
`learning_rate:`	Öğrenme hızı	0.010	0.001 – 0.5	`tabs/tab_models.py:768`; `sdm_core.py:3545`
`subsample:`	Torbalama oranı (`bag.fraction`)	0.75	0.1 – 1.0	`tabs/tab_models.py:769`; `sdm_core.py:3546`
`min_samples_leaf:`	Yapraktaki en az satır	10	1 – 100	`tabs/tab_models.py:770`; `sdm_core.py:3547`

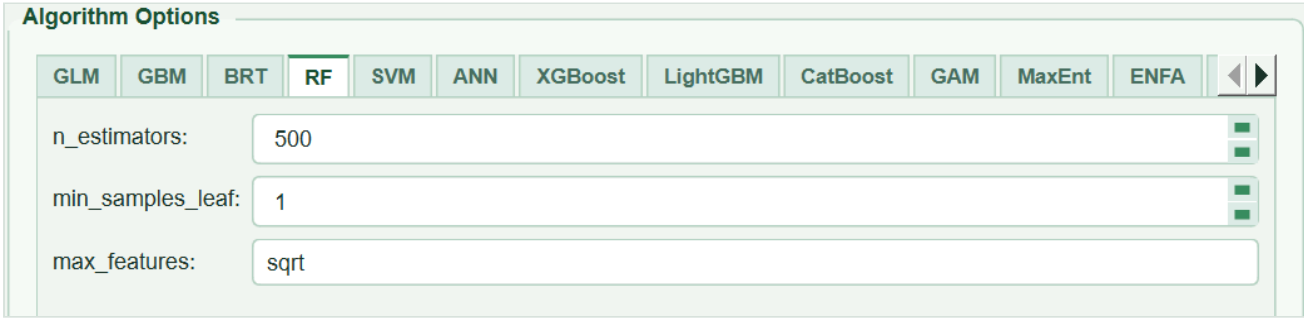
5.8.1 `max_depth`

- **Ne işe yarar.** Ağaç karmaşıklığı; etkileşim düzeyini belirler.
- **Ne zaman, nasıl ayarlanır.** Elith ve ark. (2008) örneklerinde `tree.complexity` = 5 kullanmıştır. HABITUS 2'yi varsayılan alır; kod açıklamasına göre varlık–sözde yokluk verisinde derin ağaçlar hücrelerin çoğunun uygunluğunu sıfıra iter ve Boyce indeksini yarıya düşürür (`sdm_core.py:3532-3537`). Özgün ayar için 5 girilir.
- **Sonuca etkisi.** Derinlik arttıkça etkileşim ve eğitim uyumu artar, uygunluk yüzeyi keskinleşir.
- **Dikkat.** Derinlik artınca aynı uyum için daha az ağaç gerekir; `n_estimators` ile birlikte düşünülmelidir.

5.8.2 `n_estimators`, `learning_rate`, `subsample`, `min_samples_leaf`

- **Ne işe yarar.** GBM'dekiyle aynıdır (§5.7).
- **Ne zaman, nasıl ayarlanır.** Elith ve ark. (2008) öğrenme hızını en az 1000 ağaç oluşacak biçimde seçmeyi önerir; varsayılan 0,01 ve 1000 ağaç bu kuralı izler. `subsample` 0,75 yazarların önerdiği torbalama oranıdır.
- **Sonuca etkisi.** Öğrenme hızı 0,01'den 0,005'e indirilip ağaç sayısı 2000'e çıkarılırsa model genellikle benzer ama daha düzgün olur; süre iki katına çıkar.
- **Dikkat.** dismo `gbm.step` ağaç sayısını 10 katlı çapraz doğrulamayla kendisi seçer; HABITUS'ta ağaç sayısı sabittir ve BRT ayar sekmesinde yoktur. Ağaç sayısını veriye göre seçmek isteyen kullanıcı GBM'i ayar sekmesinde arayıp aynı değerleri burada kullanabilir.

5.9 RF: rastgele orman



Şekil 5.5. RF sekmesi: n_estimators 500, min_samples_leaf 1, max_features sqrt.

Aile ve uygulama. Torbalanmış (bootstrap) sınıflandırma ağaçlarının topluluğu; her bölünmede öngörücülerin rastgele bir alt kümesi denenir (Breiman 2001). scikit-learn `RandomForestClassifier`; ağaç derinliği sınırsız, bütün işlemci çekirdekleri kullanılır (`sdm_core.py:3549-3561`). Uygunluk, ağaçların yaprak oranlarının ortalamasıdır. Varlık–sözde yokluk.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`n_estimators:`	Ağaç sayısı (randomForest: `ntree`)	500	50 – 2000 (adım 100)	`tabs/tab_models.py:778`; `sdm_core.py:3555`
`min_samples_leaf:`	Yapraktaki en az satır (randomForest: `nodesize`)	1	1 – 50	`tabs/tab_models.py:779`; `sdm_core.py:3557`
`max_features:`	Her bölünmede denenen öngörücü sayısı (randomForest: `mtry`)	`sqrt`	`sqrt`, `log2`, tamsayı, 0–1 arası oran	`tabs/tab_models.py:780-783`, `1160-1172`; `sdm_core.py:3558`

5.9.1 `n_estimators`

- **Ne işe yarar.** Ormandaki ağaç sayısı.
- **Ne zaman, nasıl ayarlanır.** Kod açıklaması Oshiro ve ark.'nın (2012) en az 128 ağaç önerisini ve 500'ün çoğu SDM verisinde kararlı olduğunu belirtir (`sdm_core.py:3550-3553`). 500 çoğu durumda yeterlidir; harita gürültülü görünüyorsa 1000'e çıkarılır.
- **Sonuca etkisi.** Ağaç sayısı aşırı uyum yaratmaz; yalnız tahminin rastgele dalgalanmasını azaltır. Süre ve bellek doğrusal artar (örnek koşuda 4 kat 21 s).
- **Dikkat.** Ağaç sayısını artırmak, RF'nin eğitim verisine sıkı uymasını (örnek koşuda eğitim AUC 1,000) değiştirmez; bunun için yaprak boyu kullanılır.

5.9.2 `min_samples_leaf`

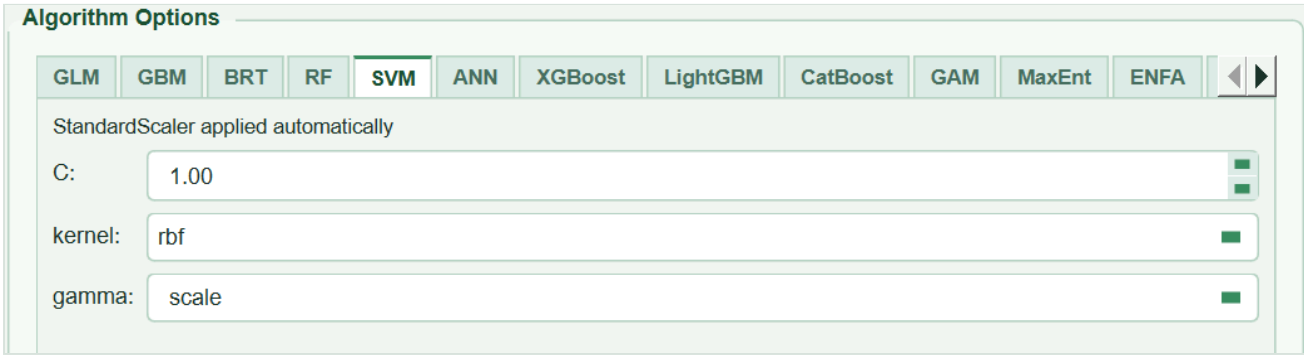
- **Ne işe yarar.** Yapraktaki en az satır; 1, ağaçların tam büyütülmesi demektir (randomForest sınıflandırma varsayılanı `nodesize` = 1).
- **Ne zaman, nasıl ayarlanır.** Eğitim–test AUC farkı büyükse ya da harita lekeli görünüyorsa 5–10'a çıkarılır.
- **Sonuca etkisi.** Büyüdükçe olasılık yüzeyi düzgünleşir, eğitim AUC'si düşer, aşırı uyum azalır.
- **Dikkat.** Çok büyük değer, az varlıklı veride varlıkları sözde yokluklarla aynı yaprağa zorlar ve ayırma gücünü düşürür.

5.9.3 `max_features`

- **Ne işe yarar.** Her bölünmede rastgele seçilen aday öngörücü sayısı. `sqrt` = $\lfloor\sqrt{p}\rfloor$ (8 öngörücüde 2), `log2` = $\lfloor\log_2 p\rfloor$, tamsayı = o kadar öngörücü, 0–1 arası sayı = öngörücülerin o oranı.
- **Ne zaman, nasıl ayarlanır.** `sqrt`, Breiman'ın sınıflandırma için önerdiği ve R randomForest'in varsayılan `mtry` değeridir. Ayar sekmesi flexsdm gibi 2'den öngörücü sayısına kadar dener.
- **Sonuca etkisi.** Küçük değer ağaçları birbirinden farklılaştırır (korelasyonu azaltır); büyük değer güçlü öngörücülerin her ağaçta öne çıkmasına yol açar.
- **Dikkat.** Okunamayan bir giriş çalıştırmayı "RF max_features: '…' is not sqrt, log2, a whole number of variables or a fraction between 0 and 1." iletisiyle durdurur; sıfır ya da negatif değer de reddedilir (`tabs/tab_models.py:1160-1172`). 1 girilirse bu "1 öngörücü" demektir, "%100" değil.

R ile eşdeğerlik. Varsayılanlar R `randomForest(…, ntree = 500, mtry = floor(sqrt(p)), nodesize = 1)` sınıflandırmasıyla aynıdır. Fark: randomForest olasılığı ağaç oylarının oranından, scikit-learn yaprak oranlarının ortalamasından hesaplar; yaprak boyu 1'de ikisi çok yakındır. Valavi ve ark.'nın (2021) önerdiği aşağı örnekli (down-sampled) RF HABITUS'ta yoktur; `Balance classes` ağırlıklandırma yapar, aşağı örnekleme yapmaz.

5.10 SVM: destek vektör makinesi



Şekil 5.6. SVM sekmesi: StandardScaler notu; C 1.00, kernel rbf, gamma scale.

Aile ve uygulama. Çekirdek (kernel) tabanlı sınır sınıflandırıcısı. scikit-learn `SVC`, `probability=True` ile: olasılıklar Platt ölçeklemesiyle, scikit-learn'ün kendi iç çapraz doğrulamasında elde edilir (`sdm_core.py:3562-3568`). Öngörücüler standartlaştırılır, kategorikler one-hot kodlanır. Varlık–sözde yokluk.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`C:`	Hata cezası; küçüldükçe daha düzgün sınır	1.00	0.001 – 1000	`tabs/tab_models.py:789`; `sdm_core.py:3565`
`kernel:`	Çekirdek işlevi	`rbf`	`rbf`, `linear`, `poly`, `sigmoid`	`tabs/tab_models.py:790`
`gamma:`	Çekirdek genişliği (RBF, poly, sigmoid)	`scale`	`scale`, `auto` ya da sayı	`tabs/tab_models.py:791-793`, `1156-1159`

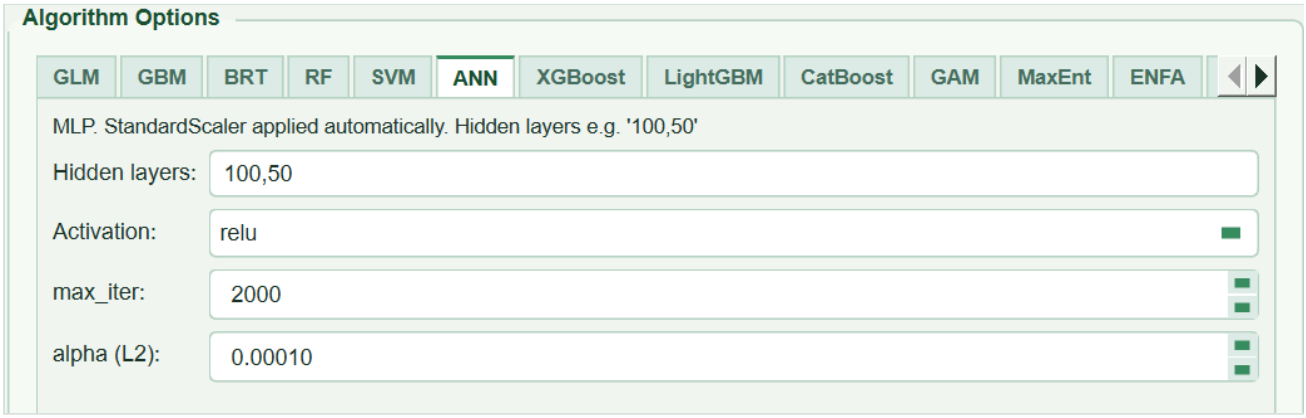
5.10.1 `C`

- **Ne işe yarar.** Yanlış sınıflanan eğitim noktalarının cezası. Büyük C eğitim verisine sıkı uyan karmaşık bir sınır, küçük C daha düzgün bir sınır verir.
- **Ne zaman, nasıl ayarlanır.** flexsdm `tune_svm` 1–16 aralığını dener; ayar sekmesi aynı ızgarayı kullanır.
- **Sonuca etkisi.** C büyüdükçe eğitim AUC'si yükselir, aşırı uyum artar.
- **Dikkat.** SVM'nin süresi satır sayısıyla kabaca kare-küp ölçeğinde artar; on binlerce sözde yoklukta yavaştır.

5.10.2 `kernel` ve `gamma`

- **Ne işe yarar.** RBF çekirdeği $k(x, x') = \exp(-\gamma \|x - x'\|^2)$'dir; γ bir eğitim noktasının etki yarıçapını belirler. `scale` = $1/(p \cdot \mathrm{Var}(X))$, `auto` = $1/p$. Öngörücüler standartlaştırıldığı için sürekli değişkenlerle ikisi aynı değeri verir; one-hot sütunlar varsa farklılaşır.
- **Ne zaman, nasıl ayarlanır.** RBF çoğu SDM için yeterlidir. Kutuya sayı da yazılabilir; ayar sekmesi buraya sayı yazar. flexsdm'in kernlab `sigma`'sı scikit-learn `gamma`'sıyla aynı tanımdır, dönüştürme gerekmez (`sdm_tuning.py:654-675`).
- **Sonuca etkisi.** Büyük γ dar etki alanı, lekeli ve aşırı uyumlu yüzey; küçük γ düzgün, doğrusala yakın yüzey verir. `linear` çekirdekte `gamma` etkisizdir.
- **Dikkat.** Platt ölçeklemesi kendi iç çapraz doğrulamasını kullandığından olasılık sıralaması `decision_function` sıralamasıyla birebir aynı olmayabilir; bu scikit-learn'ün belgelenmiş davranışıdır.

5.11 ANN: yapay sinir ağı (çok katmanlı algılayıcı)



Şekil 5.7. ANN sekmesi: Hidden layers 100,50; Activation relu; max_iter 2000; alpha (L2) 0.00010.

Aile ve uygulama. İleri beslemeli sinir ağı. scikit-learn `MLPClassifier` (Adam eniyileyicisi, scikit-learn varsayılanı), öngörücüler standartlaştırılır, kategorikler one-hot kodlanır (`sdm_core.py:3569-3581`). Varlık–sözde yokluk.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Hidden layers:`	Gizli katmanlardaki nöron sayıları, virgülle	`100,50`	pozitif tamsayılar listesi	`tabs/tab_models.py:799`, `1144-1154`
`Activation:`	Etkinleştirme işlevi	`relu`	`relu`, `tanh`, `logistic`	`tabs/tab_models.py:800`
`max_iter:`	En çok eğitim yinelemesi (dönem)	2000	100 – 5000	`tabs/tab_models.py:801`
`alpha (L2):`	Ağırlıklara L2 cezası	0.00010	0.0 – 100.0	`tabs/tab_models.py:802-804`; `sdm_core.py:3579`

5.11.1 `Hidden layers`

- **Ne işe yarar.** `100,50` iki gizli katman (100 ve 50 nöron) demektir; `5` tek katmanda 5 nöron.
- **Ne zaman, nasıl ayarlanır.** Varsayılan ağ binlerce ağırlık taşır; yüz–iki yüz varlıklı tipik SDM verisi için büyüktür. R nnet (biomod2, flexsdm) tek gizli katman ve birkaç nöron kullanır; flexsdm `tune_net` 2–8 nöron dener. Az kayıtta tek katman ve 4–8 nöron önerilir.
- **Sonuca etkisi.** Nöron arttıkça esneklik ve aşırı uyum artar, süre uzar.
- **Dikkat.** Okunamayan giriş "ANN hidden layers: '…' is not a list of positive whole numbers separated by commas (e.g. 100, 50)." iletisiyle çalıştırmayı durdurur (`tabs/tab_models.py:1144-1154`).

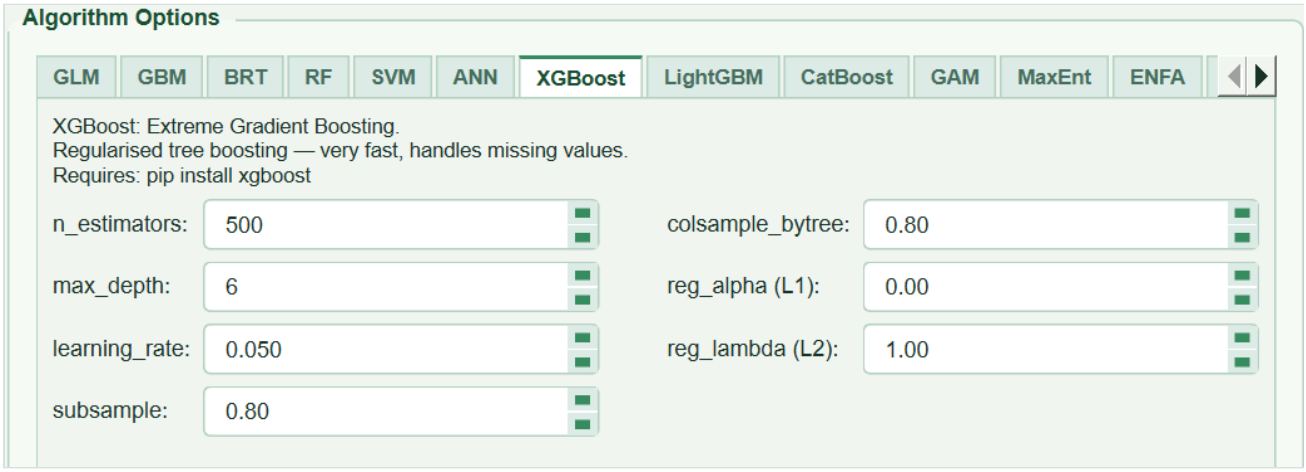
5.11.2 `alpha (L2)`

- **Ne işe yarar.** Ağırlıkların karelerine ceza (ağırlık sönümü).
- **Ne zaman, nasıl ayarlanır.** nnet'in `decay` parametresi $\alpha = 2\times$ `decay` ile karşılanır (ipucu ve `sdm_tuning.py:678-689`; bu bağıntı iki amaç işlevinden türetilmiştir, R nnet ile sayısal olarak sınanmamıştır). flexsdm ızgarasının decay 0,01–10 aralığı alpha 0,02–20'ye karşılık gelir; varsayılan 0,0001 bunlardan çok küçüktür.
- **Sonuca etkisi.** Büyüdükçe yüzey düzgünleşir, aşırı uyum azalır.
- **Dikkat.** Büyük ağ + küçük alpha birleşimi en kolay aşırı uyan ayardır.

5.11.3 `Activation` ve `max_iter`

- **Ne işe yarar.** `relu` parçalı doğrusal, `tanh` ve `logistic` düzgün S biçimli tepki üretir. `max_iter` eniyileyicinin en çok dönem sayısıdır.
- **Ne zaman, nasıl ayarlanır.** nnet `logistic` etkinleştirme kullanır; R ile karşılaştırmada `logistic` seçilebilir. Yakınsama uyarısı görülürse `max_iter` artırılır.
- **Sonuca etkisi.** `relu` tepki eğrilerinde köşeli geçişler, `logistic` / `tanh` daha düzgün eğriler verir.
- **Dikkat.** Erken durdurma kullanılmaz; ağ `max_iter`'e ya da scikit-learn'ün yakınsama ölçütüne kadar eğitilir.

5.12 XGBoost: aşırı gradyan artırma



Şekil 5.8. XGBoost sekmesi: açıklama notu; n_estimators 500, max_depth 6, learning_rate 0.050, subsample 0.80, colsample_bytree 0.80, reg_alpha (L1) 0.00, reg_lambda (L2) 1.00.

Aile ve uygulama. Düzenlemeli gradyan artırmalı ağaçlar (Chen ve Guestrin 2016), `xgboost.XGBClassifier`, kayıp işlevi log-kayıp (`sdm_core.py:3582-3608`). Varlık–sözde yokluk. Arayüzde olmayan iki ayar sabittir: `min_child_weight` = 1, `gamma` = 0.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`n_estimators:`	Artırma turu (ağaç) sayısı	500	50 – 5000	`tabs/tab_models.py:815`
`max_depth:`	Ağaç derinliği	6	1 – 12	`tabs/tab_models.py:816`
`learning_rate:`	Öğrenme hızı (eta)	0.050	0.001 – 0.5	`tabs/tab_models.py:817`
`subsample:`	Ağaç başına satır oranı	0.80	0.1 – 1.0	`tabs/tab_models.py:818`
`colsample_bytree:`	Ağaç başına öngörücü oranı	0.80	0.1 – 1.0	`tabs/tab_models.py:819`

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`reg_alpha (L1):`	Yaprak ağırlıklarına L1 cezası	0.00	0 – 10	`tabs/tab_models.py:820`
`reg_lambda (L2):`	Yaprak ağırlıklarına L2 cezası	1.00	0 – 10	`tabs/tab_models.py:821`

5.12.1 `n_estimators`, `learning_rate`, `max_depth`

- **Ne işe yarar.** GBM'deki gibi ağaç sayısı, öğrenme hızı ve derinlik.
- **Ne zaman, nasıl ayarlanır.** Varsayılan derinlik 6, GBM (1) ve BRT'ye (2) göre çok derindir ve yüksek etkileşim düzeyine izin verir; birkaç yüz varlıkta 2–3 daha güvenlidir. Ayar sekmesinin ızgarası derinlik 3/6/9, hız 0,01/0,05/0,1, tur 100/300/500'dür.
- **Sonuca etkisi.** Derinlik ve tur arttıkça eğitim uyumu ve aşırı uyum artar.
- **Dikkat.** Erken durdurma yoktur; tur sayısı sabittir.

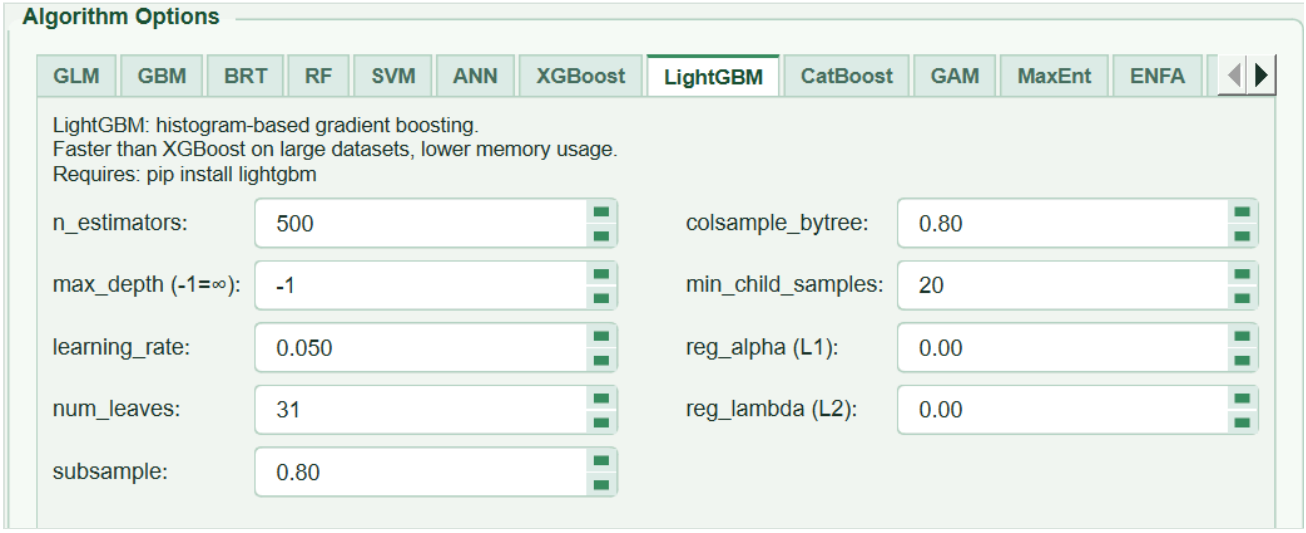
5.12.2 `subsample` ve `colsample_bytree`

- **Ne işe yarar.** Her ağaç satırların ve öngörücülerin bu oranlardaki rastgele alt kümesiyle kurulur.
- **Ne zaman, nasıl ayarlanır.** 0,5–0,9 aralığı yaygındır; az öngörücüde (ör. 4) `colsample_bytree` küçültülürse bazı ağaçlar önemli değişkeni hiç görmez.
- **Sonuca etkisi.** Küçük oranlar varyansı ve aşırı uyumu azaltır.
- **Dikkat.** Sonuç tohuma bağlıdır; tohum sabit olduğundan tekrarlanabilir.

5.12.3 `reg_alpha` ve `reg_lambda`

- **Ne işe yarar.** Yaprak değerlerine L1 ve L2 cezaları; yaprak tahminlerini küçültür.
- **Ne zaman, nasıl ayarlanır.** Varsayılanlar XGBoost'un kendi varsayılanlarıdır (0 ve 1). Aşırı uyum görülürse `reg_lambda` 5–10'a çıkarılabilir.
- **Sonuca etkisi.** Ceza arttıkça olasılık yüzeyi 0,5'e doğru sıkışır ve düzgünleşir.
- **Dikkat.** Arayüz notu XGBoost'un eksik değerleri işleyebildiğini söyler; HABITUS'ta eksik değerli satırlar veri hazırlığında zaten çıkarıldığından bu özellik kullanılmaz.

5.13 LightGBM: hafif gradyan artırma



Şekil 5.9. LightGBM sekmesi: açıklama notu; n_estimators 500, max_depth -1, learning_rate 0.050, num_leaves 31, subsample 0.80, colsample_bytree 0.80, min_child_samples 20, reg_alpha 0.00, reg_lambda 0.00.

Aile ve uygulama. Histogram tabanlı, yaprak öncelikli (leaf-wise) büyüyen gradyan artırma; `lightgbm.LGBMClassifier` (`sdm_core.py:3610-3628`). Ağaç büyüklüğü derinlikle değil yaprak sayısıyla denetlenir. Varlık–sözde yokluk.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`n_estimators:`	Tur sayısı	500	50 – 5000	`tabs/tab_models.py:837`
`max_depth (-1=∞):`	Derinlik sınırı	-1 (sınırsız)	-1 – 100	`tabs/tab_models.py:838`
`learning_rate:`	Öğrenme hızı	0.050	0.001 – 0.5	`tabs/tab_models.py:839`
`num_leaves:`	Ağaç başına en çok yaprak	31	2 – 255	`tabs/tab_models.py:840`
`subsample:`	Satır alt örnekleme oranı	0.80	0.1 – 1.0	`tabs/tab_models.py:841`
`colsample_bytree:`	Ağaç başına öngörücü oranı	0.80	0.1 – 1.0	`tabs/tab_models.py:842`
`min_child_samples:`	Yapraktaki en az satır	20	1 – 200	`tabs/tab_models.py:843`
`reg_alpha (L1):`	L1 cezası	0.00	0 – 10	`tabs/tab_models.py:844`
`reg_lambda (L2):`	L2 cezası	0.00	0 – 10	`tabs/tab_models.py:845`

5.13.1 `num_leaves` ve `max_depth`

- **Ne işe yarar.** `num_leaves` ağacın karmaşıklığını doğrudan sınırlar; `max_depth` = -1 derinliği sınırsız bırakır.
- **Ne zaman, nasıl ayarlanır.** 31 yaprak, derinliği 5 olan dengeli bir ağaçtan daha karmaşıktır; az kayıtta 7–15 yaprak ve `max_depth` 3–4 daha uygundur. Ayar sekmesi 15/31/63 dener.
- **Sonuca etkisi.** Yaprak arttıkça etkileşim ve aşırı uyum artar.
- **Dikkat.** Yaprak öncelikli büyüme az veride çok derin, dar dallar üretebilir; `max_depth` ile sınırlamak bunu önler.

5.13.2 `min_child_samples`

- **Ne işe yarar.** Bir yaprakta bulunması gereken en az satır.
- **Ne zaman, nasıl ayarlanır.** Varsayılan 20; 50'den az varlıkta varlıkları içeren yaprakların oluşmasını zorlaştırabilir, 5–10'a indirilebilir.
- **Sonuca etkisi.** Büyüdükçe model kabalaşır.
- **Dikkat.** Çok az varlıkta modelin hiç bölünme bulamaması ve neredeyse sabit tahmin vermesi mümkündür.

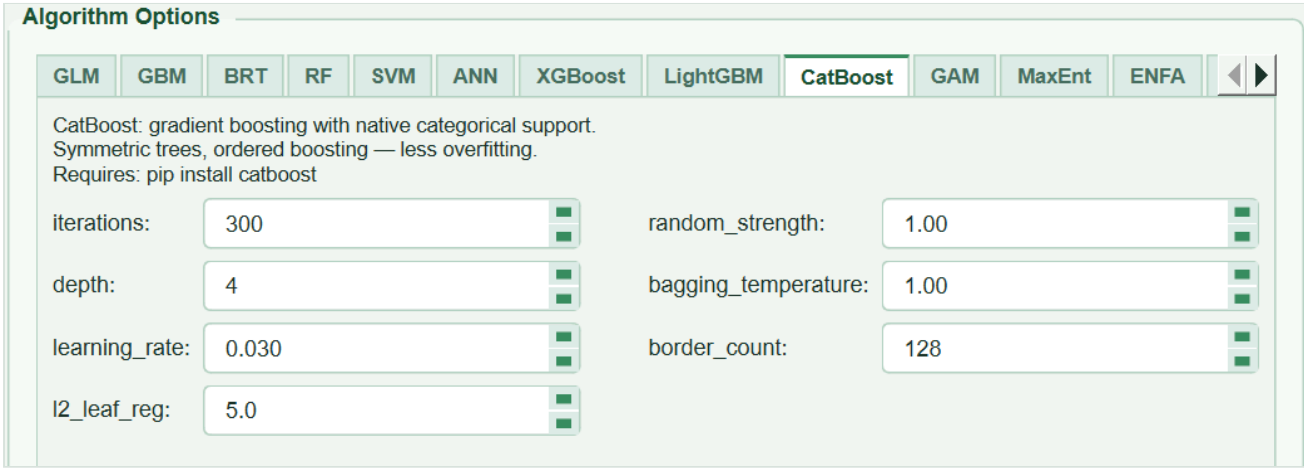
5.13.3 `subsample` ve `colsample_bytree`

- **Ne işe yarar.** Satır ve öngörücü alt örnekleme oranları.
- **Ne zaman, nasıl ayarlanır.** `colsample_bytree` her ağaçta uygulanır.
- **Sonuca etkisi.** `colsample_bytree` küçüldükçe ağaçlar çeşitlenir.
- **Dikkat.** LightGBM'de satır alt örneklemesi (`subsample`) yalnız `subsample_freq` sıfırdan büyükken etkindir; HABITUS bu ayarı vermediğinden (`sdm_core.py:3613-3628`) `subsample` değeri sonuca yansımaz.

5.13.4 `n_estimators`, `learning_rate`, `reg_alpha`, `reg_lambda`

- **Ne işe yarar.** XGBoost'taki anlamlarıyla aynıdır (§5.12).
- **Ne zaman, nasıl ayarlanır.** LightGBM'in kendi varsayılan cezaları sıfırdır; aşırı uyumda `reg_lambda` artırılır.
- **Sonuca etkisi.** Bkz. §5.12.
- **Dikkat.** Erken durdurma yoktur.

5.14 CatBoost: kategorik artırma



Şekil 5.10. CatBoost sekmesi: açıklama notu; iterations 300, depth 4, learning_rate 0.030, l2_leaf_reg 5.0, random_strength 1.00, bagging_temperature 1.00, border_count 128.

Aile ve uygulama. Simetrik (her düzeyde aynı bölünmeyi kullanan) ağaçlarla sıralı artırma; `catboost.CatBoostClassifier` (`sdm_core.py:3630-3647`). Varlık–sözde yokluk.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`iterations:`	Tur sayısı	300	50 – 5000	`tabs/tab_models.py:863`
`depth:`	Simetrik ağaç derinliği	4	1 – 16	`tabs/tab_models.py:864`
`learning_rate:`	Öğrenme hızı	0.030	0.001 – 0.5	`tabs/tab_models.py:865`
`l2_leaf_reg:`	Yaprak değerlerine L2 cezası	5.0	0 – 20	`tabs/tab_models.py:866`
`random_strength:`	Bölünme skorlarına eklenen gürültü	1.00	0.01 – 10	`tabs/tab_models.py:867`
`bagging_temperature:`	Bayesçi bootstrap ağırlıklarının gücü	1.00	0 – 10	`tabs/tab_models.py:868`
`border_count:`	Sürekli öngörücülerin bölünme sınırı sayısı	128	1 – 255	`tabs/tab_models.py:869`

5.14.1 `iterations`, `learning_rate`, `depth`

- **Ne işe yarar.** Tur sayısı, öğrenme hızı ve ağaç derinliği. Simetrik ağaçta derinlik d, tam olarak 2^d yaprak demektir (4 → 16 yaprak).
- **Ne zaman, nasıl ayarlanır.** Varsayılanlar (300 tur, 0,03, derinlik 4) CatBoost'un kütüphane varsayılanlarından (1000 tur, derinlik 6) daha küçük bir modeldir. CatBoost ayar sekmesinde yoktur; değerler elle denenir.
- **Sonuca etkisi.** Derinlik ve tur arttıkça eğitim uyumu artar.
- **Dikkat.** Hız ile tur sayısı birlikte düşünülmelidir (§5.7.1).

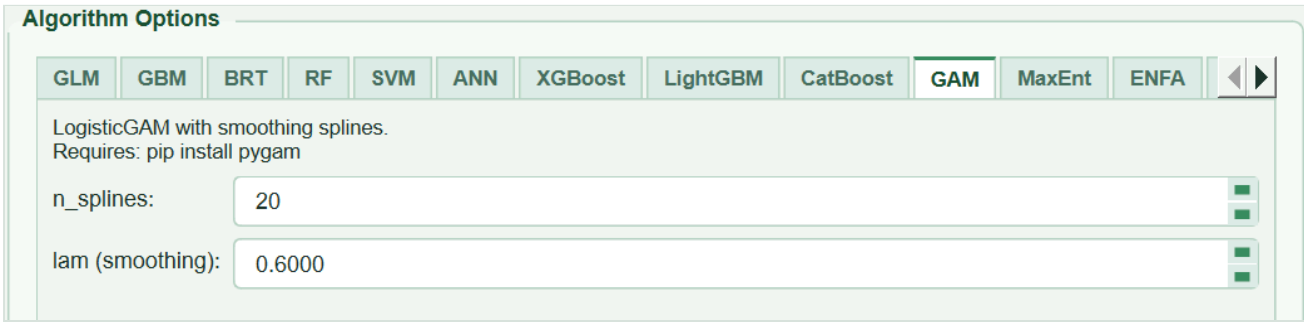
5.14.2 `l2_leaf_reg`, `random_strength`, `bagging_temperature`

- **Ne işe yarar.** `l2_leaf_reg` yaprak değerlerini küçültür; `random_strength` bölünme seçimine rastgelelik katar; `bagging_temperature` satır ağırlıklarının rastgeleliğini belirler (0 = bütün satırlar eşit ağırlıkta).
- **Ne zaman, nasıl ayarlanır.** Aşırı uyumda `l2_leaf_reg` artırılır (CatBoost'un kendi varsayılanı 3, HABITUS 5 kullanır).
- **Sonuca etkisi.** Üçü de büyüdükçe model düzgünleşir ve tohuma duyarlılığı artar.
- **Dikkat.** `bagging_temperature` = 0 rastgeleliği kaldırmaz; `random_strength` yine etkilidir.

5.14.3 `border_count`

- **Ne işe yarar.** Her sürekli öngörücünün kaç olası bölünme noktasına (kantil sınırına) ayrılacağı.
- **Ne zaman, nasıl ayarlanır.** 128 çoğu biyoiklim değişkeni için yeterli çözünürlüktür.
- **Sonuca etkisi.** Küçük değer tepki eğrilerini basamaklı yapar; büyük değer süreyi uzatır.
- **Dikkat.** Kategorik değişkenler HABITUS'ta CatBoost'a kategorik olarak bildirilmez; tamsayı kodu sürekli bir sütun gibi bu sınırlarla bölünür (§5.4.3). Arayüz notundaki "native categorical support" bu nedenle HABITUS'ta kullanılmaz.

5.15 GAM: genelleştirilmiş toplamsal model



Şekil 5.11. GAM sekmesi: LogisticGAM notu; n_splines 20, lam (smoothing) 0.6000.

Aile ve uygulama. Lojistik toplamsal model: logit, her öngörücünün düzgün (spline) bir işlevinin toplamıdır (Hastie ve Tibshirani 1986). pygam `LogisticGAM`; her sürekli öngörücüye cezalı bir B-spline terimi `s()`, her kategorik öngörücüye faktör terimi `f()` verilir; etkileşim terimi yoktur (`sdm_core.py:4500-4549`). Varlık–sözde yokluk. Ham değerlerle çalışır.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`n_splines:`	Öngörücü başına spline taban işlevi sayısı	20	4 – 100	`tabs/tab_models.py:881-886`; `sdm_core.py:4508-4512`
`lam (smoothing):`	Düzgünleştirme cezası	0.6000	0.0001 – 1000	`tabs/tab_models.py:887-892`; `sdm_core.py:4513-4514`

5.15.1 `n_splines`

- **Ne işe yarar.** Her tepki eğrisinin kurulduğu taban işlevi sayısı; eğrinin en yüksek esnekliğini belirler. Spline derecesi 3 olduğundan 3'ten büyük olmalıdır; aksi hâlde "GAM n_splines must exceed the spline order (3)." iletisi verilir.
- **Ne zaman, nasıl ayarlanır.** 20 pygam'ın varsayılanıdır. Az kayıtta 5–10 yeterlidir ve daha az parametre demektir; ayar sekmesi 5/10/15/20 dener.
- **Sonuca etkisi.** Arttıkça eğriler daha kıvrımlı olabilir (ceza izin verdiği ölçüde); süre artar.
- **Dikkat.** Bütün öngörücülere aynı değer uygulanır.

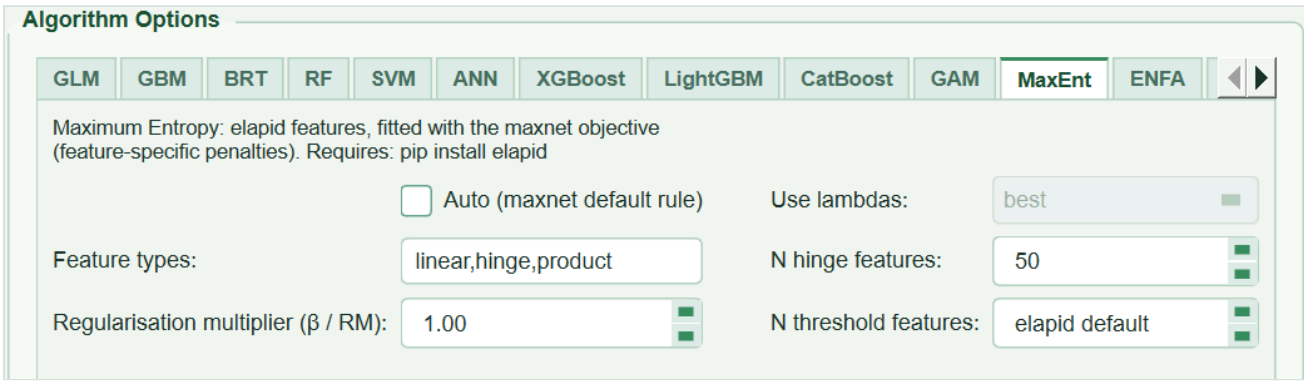
5.15.2 `lam` (düzgünleştirme cezası)

- **Ne işe yarar.** Eğrinin ikinci türevine (kıvrımına) uygulanan ceza katsayısı. Büyük `lam` eğriyi doğruya yaklaştırır.
- **Ne zaman, nasıl ayarlanır.** 0,6 pygam'ın varsayılanıdır. Tepki eğrileri dalgalı ve ekolojik olarak inandırıcı değilse 2–10'a çıkarılır; ayar sekmesi 0,1/0,6/2/10 dener.
- **Sonuca etkisi.** Örnek koşuda (varsayılanlarla) GAM'ın eğitim–test AUC farkı 0,087'dir; GLM'in (0,031) neredeyse üç katı. Bu, düzgünleştirmenin bu veride hafif kaldığına işaret eder.
- **Dikkat.** Ceza veriden otomatik seçilmez; girilen `lam` sabit kullanılır.

R ile eşdeğerlik. biomod2 ve flexsdm GAM'ları mgcv kullanır; mgcv düzgünleştirme katsayısını veriden (GCV/REML) kestirir ve varsayılan taban ince plaka regresyon spline'ıdır. HABITUS'taki pygam modeli sabit `lam` ile çalışır; bu nedenle iki programın tepki eğrileri aynı veride farklı düzgünlükte olabilir.

> **Dikkat:** GAM sınıf dengesini desteklemez; `Balance classes` GAM'ı etkilemez (`sdm_core.py:3681`).

5.16 MaxEnt: en yüksek entropi



Şekil 5.12. MaxEnt sekmesi: elapid ve maxnet amaç işlevi notu; Auto (maxnet default rule) kapalı, Feature types linear,hinge,product, Regularisation multiplier (β / RM) 1.00, Use lambdas best (devre dışı), N hinge features 50, N threshold features elapid default.

Aile ve uygulama. Varlık + arka plan yöntemi (Phillips ve ark. 2006, 2017). Öngörücülerden "özellikler" (doğrusal, kare, çarpım, menteşe, eşik) türetilir ve L1 cezalı bir modelle hangi özelliklerin kalacağı belirlenir. HABITUS özellikleri elapid kütüphanesiyle üretir (Anderson 2023), katsayıları ise R maxnet'in amaç işlevini çözerek kestirir (`sdm_core.py:3776-3836`, `4551-4757`). Çıktı elapid'in cloglog dönüşümüdür ve sonradan yeniden ölçeklenmez (`sdm_core.py:4772-4845`).

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Auto (maxnet default rule)`	Özellik sınıflarını varlık sayısından seçer	kapalı	açık / kapalı	`tabs/tab_models.py:1056-1063`; `sdm_core.py:3758-3773`
`Feature types:`	Özellik sınıfları, virgülle	`linear,hinge,product`	`linear`, `quadratic`, `hinge`, `product`, `threshold`	`tabs/tab_models.py:1048-1055`; `sdm_core.py:4616-4624`
`Regularisation multiplier (β / RM):`	Düzenleme çarpanı	1.00	0.1 – 20.0 (adım 0.5)	`tabs/tab_models.py:1064-1077`; `sdm_core.py:4625`
`Use lambdas:`	Kullanılmaz (devre dışı)	`best`	`best`, `last`	`tabs/tab_models.py:1078-1085`
`N hinge features:`	Öngörücü başına menteşe düğümü sayısı	50	10 – 200	`tabs/tab_models.py:1086`; `sdm_core.py:4626`
`N threshold features:`	Öngörücü başına eşik özelliği sayısı	`elapid default` (0 = elapid'in 10'u)	0 – 200	`tabs/tab_models.py:1087-1091`; `sdm_core.py:4645-4649`

5.16.1 maxnet amaç işlevi birebir uygulanır

HABITUS derlemelerinde R glmnet'in Python karşılığı bulunmadığından elapid kendi başına tek bir ceza katsayılı scikit-learn lojistik regresyonuna düşer. Bu yolda özelliğe özgü cezalar yoktur ve düzenleme çarpanı modeli neredeyse değiştirmez (`sdm_core.py:3780-3787`). HABITUS bu nedenle elapid'in kurduğu özellik dönüştürücüsünü kullanır, katsayıları ise `maxnet_refit` işleviyle yeniden kestirir (`sdm_core.py:4719-4746`). maxnet, glmnet'i `standardize = FALSE`, özellik başına ceza çarpanları ve $w_i = p_i + 100(1 - p_i)$ ağırlıklarıyla (varlık 1, arka plan 100) bir λ yolu boyunca çözer ve son λ'yı tutar. Bu son λ'da çözülen problem, sabit bir çarpan dışında şudur (`sdm_core.py:3789-3797`):

$$\min_{\beta_0,\beta} \sum_i w_i \, \ell_i(\beta_0, \beta) \; + \; n_{\text{varlık}} \sum_j r_j \, |\beta_j|$$

Burada ℓ_i lojistik kayıp, r_j özellik j'nin düzenleme değeridir. r_j elapid'in `compute_regularization` işleviyle özellik sınıfına, varlık sayısına, özelliğin varlıklardaki dağılımına ve β çarpanına göre hesaplanır (`sdm_core.py:3810-3816`). Kesişim cezasızdır. Problem liblinear ile, özellikler $n_{\text{varlık}} \cdot r_j$'ye bölünerek çözülür; ardından MaxEnt normalleştirme sabitleri (alpha, entropi) arka plan satırlarından yeniden hesaplanır (`sdm_core.py:3817-3835`). Koşunun kaydında bu çözücü "maxnet objective (feature-specific L1 penalties, last lambda)" olarak yazılır (`sdm_core.py:4743-4744`).

Varlıklar arka plana eklenir. maxnet'in `addsamplestobackground = TRUE` ve Maxent'in varsayılan davranışı gibi, öngörücü değerleri arka plan satırlarında bulunmayan her varlık arka plana da eklenir (`sdm_core.py:4698-4707`). Eklenen sayı `habitus_run_config.json` içinde `maxent_settings_used.presences_added_to_background` alanına yazılır; örnek koşunun tam modelinde 91'dir.

R maxnet ile otomatik test. `tests/test_maxent_maxnet.py`, Rutrough ve ark.'nın (2019) verisinin bir alt kümesinde (302 varlık, 2000 arka plan satırı; ppt, silt, slope) HABITUS'un cloglog tahminini R maxnet 0.1.4 çıktısıyla karşılaştırır. Üç ayar sınanır (LQH β=3, LQH β=1, LQPH β=3); her birinde Pearson ve Spearman korelasyonunun 0,995'ten, ortalama mutlak farkın 0,02'den küçük olması beklenir (`tests/test_maxent_maxnet.py:39-50`). İki ek test, β büyüdükçe sıfırdan farklı katsayı sayısının azaldığını (`53-59`) ve varlıkların arka plana eklendiğini (`62-64`) denetler. R çıktısı testle birlikte saklanan bir dosyadır; testin kendisi R çalıştırmaz. Kod açıklaması aynı veride R maxnet ile Schoener D = 0,998 uyum bildirir (`sdm_core.py:3799-3800`).

5.16.2 `Feature types` ve `Auto (maxnet default rule)`

- **Ne işe yarar.** Hangi özellik sınıflarının üretileceğini belirler: `linear` (doğrusal), `quadratic` (kare), `product` (ikili çarpım), `hinge` (menteşe; bir düğümden sonra doğrusal artan ya da azalan parçalı işlev), `threshold` (eşik; basamak). `Auto` işaretliyse sınıflar her uyumun eğitim varlık sayısından maxnet kuralıyla seçilir: 10'dan az varlık L; 10–14 LQ; 15–79 LQH; 80 ve üstü LQHP. Eşik sınıfı otomatik kuralda hiç kullanılmaz; kutu işaretliyken metin kutusu yok sayılır ve devre dışı kalır.
- **Ne zaman, nasıl ayarlanır.** Az kayıtta `linear` ya da `linear,quadratic` savunulabilir seçimdir (ipucu). Merow ve ark. (2013) karmaşıklığın soruya ve veriye göre seçilmesini önerir. ENMeval/Wallace'taki gibi bir arama için ayar sekmesi L, LQ, H, LQH, LQHP setlerini dener.
- **Sonuca etkisi.** Az sınıf düzgün ve aktarılabilir tepki, çok sınıf (özellikle menteşe ve eşik) kalibrasyon verisine sıkı uyan, basamaklı tepki verir. `Auto` açıkken çapraz doğrulama katları ve tam model, varlık sayılarına göre farklı sınıf setleri kullanabilir (`sdm_core.py:4619-4622`); kullanılan kural kayda yazılır.
- **Dikkat.** Varsayılan `linear,hinge,product`, maxnet'in kuralıyla aynı değildir (kare sınıfı yoktur). maxnet ile karşılaştırma yapılacaksa `Auto` işaretlenmelidir.

5.16.3 `Regularisation multiplier (β / RM)`

- **Ne işe yarar.** Bütün r_j değerlerini çarpan katsayı. Büyüdükçe L1 cezası artar, daha az özellik sıfırdan farklı kalır.
- **Ne zaman, nasıl ayarlanır.** maxnet ve Maxent varsayılanı 1'dir. Tek doğru değer yoktur; ENMeval (Muscarella ve ark. 2014) ve kuenm bu nedenle çarpanı arar. Ayar sekmesi rm 0,5–4,0'ı (adım 0,5) özellik setleriyle çapraz dener ve seçileni buraya yazar. En iyi değer kalibrasyon alanına bağlıdır; alan değişirse yeniden aranmalıdır (ipucu).
- **Sonuca etkisi.** β = 1'den 6'ya çıkarıldığında sıfırdan farklı katsayı sayısının azaldığı testle sınanır (`tests/test_maxent_maxnet.py:53-59`). Haritada büyük β daha geniş ve yumuşak, küçük β daha dar ve lekeli uygunluk verir.
- **Dikkat.** Değerin etkisi ancak maxnet amaç işleviyle anlamlıdır; HABITUS her uyumda bu yolu kullanır ve kullanıldığını kayda yazar.

5.16.4 `N hinge features` ve `N threshold features`

- **Ne işe yarar.** Öngörücü başına üretilecek menteşe düğümü ve eşik özelliği sayısı.
- **Ne zaman, nasıl ayarlanır.** 50 menteşe maxnet'in varsayılanıdır. Eşik sayısı yalnız `threshold` sınıfı seçiliyse kullanılır; 0 elapid'in kendi değerini (10) bırakır.
- **Sonuca etkisi.** Düğüm sayısı arttıkça tepki eğrisi daha ince ayrıntı yakalayabilir; ceza bunu sınırlar.
- **Dikkat.** Ayar sekmesinde bu iki değer bütün ızgara noktaları için sabit tutulur (`sdm_tuning.py:550-551`).

5.16.5 `Use lambdas`

Kutu devre dışıdır ve sonucu etkilemez: model maxnet gibi λ yolunun son değerinde uydurulur (`tabs/tab_models.py:1079-1085`). Değer yine de kayda (`use_lambdas`) yazılır.

> **Not:** MaxEnt çapraz doğrulamasında arka plan katlara bölünmez; her kat aynı arka planla eğitilir. Test skorları test katındaki varlık ve sözde yokluklarla hesaplanır. Arka planın büyüklüğü ve örnekleme biçimi ① Data'da ayarlanır (Bölüm 3). Kurulu elapid bir ayarı tanımıyorsa ayar sessizce atılmaz; "MaxEnt: the installed elapid (…) has no setting for …" uyarısı verilir (`sdm_core.py:4684-4691`).

5.17 ENFA: ekolojik niş faktör analizi



Şekil 5.13. ENFA sekmesi: Hirzel ve ark. (2002) notu; Specialisation axes (K) 3.

Aile ve uygulama. Türün kullandığı ortamı genel (küresel) ortamla karşılaştıran çok değişkenli niş yöntemi (Hirzel ve ark. 2002). HABITUS'ta numpy ile yazılmıştır (`sdm_core.py:4046-4204`):

1. Küresel ortalama ve kovaryans, varsa ① Data'daki yalnız-varlık arka planından, yoksa eğitim katının bütün satırlarından (varlık + sözde yokluk) kestirilir; kovaryans Ledoit–Wolf büzülmesiyle hesaplanır ve köşegenine 1e-6 eklenir.
2. Veri küresel kovaryansla beyazlatılır; marjinallik ekseni, tür ortalamasının küresel ortalamadan farkının yönüdür.
3. Marjinalliğe dik alt uzayda varlıkların kovaryansının özdeğer ayrışımı yapılır ve en büyük özdeğerli K eksen tutulur.
4. Ham puan $\exp\{-\frac{1}{2}[d_m^2 + \sum_k (z_k - \mu_k)^2/\lambda_k]\}$'dir; d_m marjinallik ekseni boyunca tür ortalamasına uzaklıktır.
5. Puan, uydurma sırasında eğitim satırlarının puan aralığıyla 0–1'e ölçeklenir; bu aralık modelle saklanır, böylece aynı hücre her tahminde aynı değeri alır (`sdm_core.py:3719-3743`, `4158-4162`).

Kategorik değişkenler dışarıda bırakılır. En az 2 varlık gerekir.

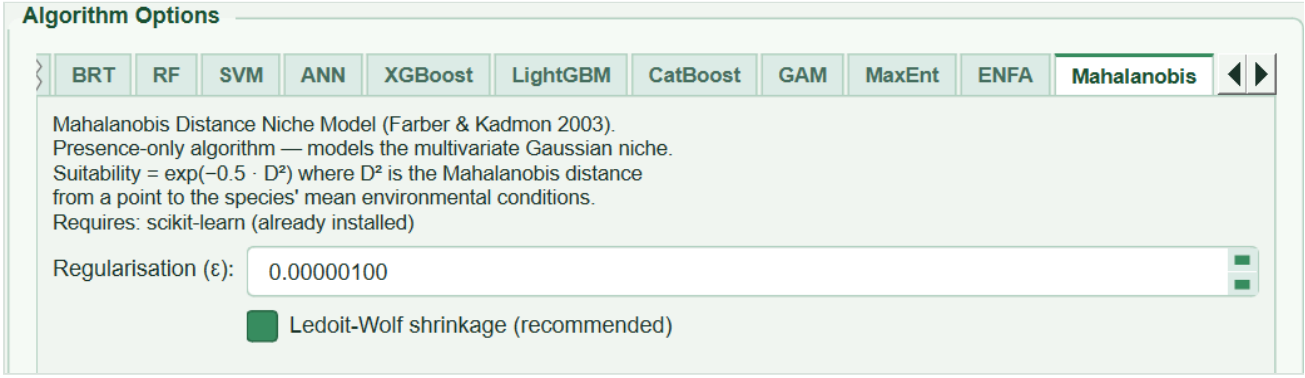
Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Specialisation axes (K):`	Tutulacak özelleşme ekseni sayısı	3	1 – 10	`tabs/tab_models.py:1102-1104`; `sdm_core.py:4074`, `4133`

5.17.1 `Specialisation axes (K)`

- **Ne işe yarar.** Puana marjinallik ekseninin yanında kaç ek eksenin gireceğini belirler. Öngörücü sayısının bir eksiğiyle sınırlanır (`sdm_core.py:4133`).
- **Ne zaman, nasıl ayarlanır.** 2–4 eksen çoğu durumda varyansın büyük kısmını taşır; az öngörücüde (ör. 4) K en çok 3 olabilir.
- **Sonuca etkisi.** K arttıkça puan daha çok boyutta uzaklık ölçer, uygun alan daralır.
- **Dikkat.** ENFA sınıf ağırlığı ve kırpma dışında başka ayar almaz; ölçek ve düzenleme sabitleri kodda sabittir.

R ile karşılaştırma. adehabitatHS `enfa` marjinallik ve özelleşme eksenlerini hesaplar; uygunluk haritası için Hirzel ve ark.'nın özgün çalışması ayrı bir hesap kullanır. HABITUS'un ENFA puanı yukarıdaki Gauss biçimli uzaklıktır; R uygulamalarıyla sayısal karşılaştırma testi yoktur.

5.18 Mahalanobis uzaklığı



Şekil 5.14. Mahalanobis sekmesi: Farber ve Kadmon (2003) notu; Regularisation (ε) 0.00000100; Ledoit-Wolf shrinkage (recommended) işaretli.

Aile ve uygulama. Yalnız-varlık uzaklık yöntemi: varlıkların ortalama ortamına Mahalanobis uzaklığı $D^2(x) = (x-\mu)^\top \Sigma^{-1}(x-\mu)$ hesaplanır ve ham puan $\exp(-D^2/2)$ olur (Farber ve Kadmon 2003). Kovaryans yalnız varlıklardan kestirilir (`sdm_core.py:4207-4273`). Puan, ENFA'daki gibi eğitim satırlarının (varlık + sözde yokluk) puan aralığıyla 0–1'e ölçeklenir ve bu aralık saklanır. Kategorik değişkenler dışarıda kalır; en az 2 varlık gerekir.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Regularisation (ε):`	Kovaryans köşegenine eklenen sabit	0.00000100	0.0 – 1.0	`tabs/tab_models.py:1116-1124`; `sdm_core.py:4237`, `4245`
`Ledoit-Wolf shrinkage (recommended)`	Kovaryansı Ledoit–Wolf büzülmesiyle kestirir	açık	açık / kapalı	`tabs/tab_models.py:1125-1129`; `sdm_core.py:4238-4244`

5.18.1 `Ledoit-Wolf shrinkage`

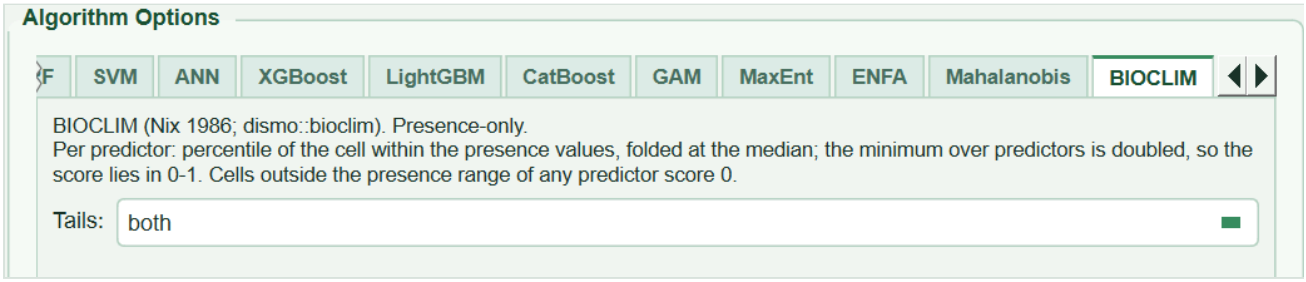
- **Ne işe yarar.** Örnek kovaryansını köşegen bir hedefe doğru en uygun miktarda büzer (Ledoit ve Wolf 2004); kapalıysa olağan örnek kovaryansı kullanılır.
- **Ne zaman, nasıl ayarlanır.** Varlık sayısı öngörücü sayısının birkaç katından azsa mutlaka açık tutulur.
- **Sonuca etkisi.** Büzülme uzaklıkları kararlı kılar; ilişkili öngörücülerde elips daha yuvarlak olur, uygun alan bir miktar genişler.
- **Dikkat.** Kapatılırsa ve öngörücüler güçlü ilişkiliyse matris neredeyse tekil olur; ters alma kararsızlaşır.

5.18.2 `Regularisation (ε)`

- **Ne işe yarar.** $\Sigma + \varepsilon I$ kullanılarak matrisin tersi alınabilir tutulur.
- **Ne zaman, nasıl ayarlanır.** `LinAlgError` alınırsa artırılır (ipucu); 1e-4–1e-2 aralığı denenebilir.
- **Sonuca etkisi.** Büyük ε, küçük varyanslı yönlerdeki uzaklıkları küçültür ve uygun alanı genişletir.
- **Dikkat.** Kutu 8 ondalık basamak gösterir; daha küçük değerler sıfıra yuvarlanır.

R ile karşılaştırma. dismo `mahal` ham uzaklığı farklı bir dönüşümle verir; HABITUS'un $\exp(-D^2/2)$ dönüşümü ve eğitim aralığıyla ölçeklemesi ondan farklıdır. Aynı ortalama ve kovaryans kullanıldığında sıralama (dolayısıyla AUC) büyük ölçüde aynıdır; yalnız HABITUS'un eğitim aralığına kırpması aralık dışındaki hücrelerde eşit değerler yaratabilir.

5.19 BIOCLIM: iklim zarfı



Şekil 5.15. BIOCLIM sekmesi: yöntem notu (Nix 1986; dismo::bioclim); Tails both.

Aile ve uygulama. Yalnız-varlık zarf (envelope) yöntemi (Nix 1986). numpy ile yazılmıştır (`sdm_niche.py:44-90`). Her öngörücü için hücre değerinin varlık değerleri içindeki yüzdeliği p orta-sıra kuralıyla hesaplanır; $p > 0{,}5$ ise $1 - p$ alınır (medyanda katlanır), öngörücüler üzerinden en küçük değer alınıp ikiyle çarpılır. Sonuç 0–1 arasındadır: 1 bütün öngörücülerin medyanında, 0 en az bir öngörücünün varlık aralığı dışında demektir. Puan yeniden ölçeklenmez. Kategorik değişkenler dışarıda kalır; en az 2 varlık gerekir.

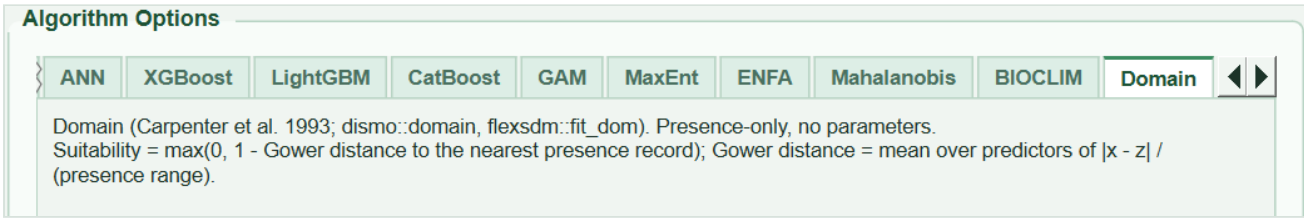
Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Tails:`	Hangi kuyruğun puanı düşüreceği	`both`	`both`, `low`, `high`	`tabs/tab_models.py:900-907`; `sdm_niche.py:79-88`

5.19.1 `Tails`

- **Ne işe yarar.** `both` klasik BIOCLIM'dir: iki kuyruk da puanı düşürür. `low` yalnız alt kuyruğu sayar (medyanın üstü 1, varlık aralığının üstü de 1); `high` yalnız üst kuyruğu sayar. dismo `predict(…, tails=)` ile aynı anlamdadır.
- **Ne zaman, nasıl ayarlanır.** Değişkenlerin hepsinin yalnız bir yönde sınırlayıcı olduğu düşünülüyorsa (ör. yalnız minimum sıcaklıklar) `low` ya da `high` seçilir. Seçim bütün öngörücülere uygulanır.
- **Sonuca etkisi.** `low` ya da `high`, `both`'a göre daha geniş bir uygun alan verir.
- **Dikkat.** Arayüzde öngörücü başına ayrı kuyruk seçimi yoktur.

R ile karşılaştırma. dismo `bioclim` ile aynı tanımdır. Yüzdelik orta sıra (mid-rank) ile hesaplanır; dismo'nun eşit değerleri işleme biçimi doğrulanamadığından değerler $1/n$ düzeyinde farklı olabilir (`sdm_niche.py:16-18`).

5.20 Domain: Gower uzaklığı



Şekil 5.16. Domain sekmesi: yöntem notu (Carpenter ve ark. 1993; dismo::domain, flexsdm::fit_dom); ayar yok.

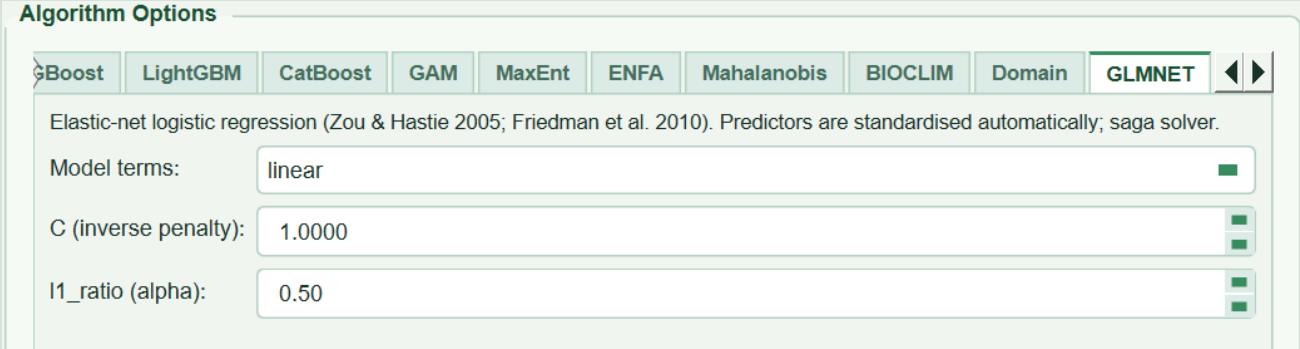
Aile ve uygulama. Yalnız-varlık benzerlik yöntemi (Carpenter ve ark. 1993). Bir hücre ile her varlık arasındaki Gower uzaklığı, öngörücüler üzerinden $|x_j - z_j|/\text{aralık}_j$ ortalamasıdır; aralık varlık değerlerinin aralığıdır. Uygunluk $\max(0,\, 1 - \min_z G)$'dir (`sdm_niche.py:95-127`). Aralığı sıfır olan öngörücü eşitse 0, değilse 1 katkı verir. Kategorik değişkenler dışarıda kalır; en az 2 varlık gerekir; yinelenen varlık satırları bir kez sayılır.

Ayarı yoktur; sayfada yalnız açıklama notu bulunur (`tabs/tab_models.py:917-925`).

> **Dikkat:** Domain, en yakın tek varlığa bakar. Tek bir hatalı ya da uç kayıt, çevresinde yüksek uygunluk yaratır; kayıt temizliği (Bölüm 2) bu yöntem için özellikle önemlidir. Hesap süresi varlık sayısıyla doğrusal artar; bellek, hücre bloklarıyla sınırlandırılır.

R ile karşılaştırma. dismo `domain` ve flexsdm `fit_dom` ile aynı tanımdır; HABITUS'ta karşılaştırmalı test yoktur.

5.21 GLMNET: elastik ağ lojistik regresyon



Şekil 5.17. GLMNET sekmesi: yöntem notu; Model terms linear, C (inverse penalty) 1.0000, l1_ratio (alpha) 0.50.

Aile ve uygulama. L1 ve L2 cezalarını birleştiren elastik ağ lojistik regresyon (Zou ve Hastie 2005; Friedman ve ark. 2010). scikit-learn `LogisticRegression`, `saga` çözücü, en çok 5000 yineleme; terimler GLM'deki gibi üretilip standartlaştırılır (`sdm_core.py:3446-3461`, `3493-3505`). Varlık–sözde yokluk.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Model terms:`	Terim yapısı	`linear`	`linear`, `quadratic`, `interactions`	`tabs/tab_models.py:929-934`
`C (inverse penalty):`	Ters ceza gücü	1.0000	0.0001 – 10 000	`tabs/tab_models.py:935-939`
`l1_ratio (alpha):`	L1 / L2 karışımı	0.50	0.0 – 1.0 (adım 0.1)	`tabs/tab_models.py:940-945`

5.21.1 `C (inverse penalty)`

- **Ne işe yarar.** scikit-learn amaç işlevini n ile bölerek glmnet biçimine getirirsek $\lambda \approx 1/(C\,n)$ olur; n eğitim satırı sayısıdır.
- **Ne zaman, nasıl ayarlanır.** Örnek koşuda bir eğitim katında yaklaşık 860 satır vardır; C = 1, $\lambda \approx 0{,}0012$ demektir ve hafif bir cezadır. Gerçek bir seçim için ayar sekmesi C = 0,01/0,1/1/10 dener.
- **Sonuca etkisi.** C küçüldükçe katsayılar küçülür; l1_ratio > 0 ise bazı öngörücüler tamamen düşer.
- **Dikkat.** R glmnet λ'yı genellikle `cv.glmnet` ile seçer; HABITUS'ta tek bir sabit C kullanılır.

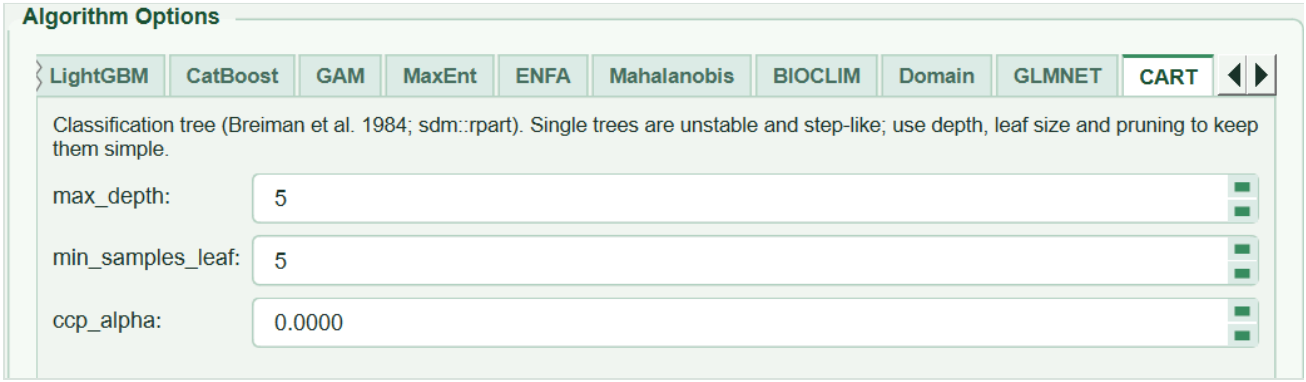
5.21.2 `l1_ratio (alpha)`

- **Ne işe yarar.** glmnet'in `alpha`'sıdır: 0 ridge (yalnız L2), 1 lasso (yalnız L1; değişken düşürür), arası elastik ağ.
- **Ne zaman, nasıl ayarlanır.** Güçlü ilişkili öngörücüler varsa 0,5 civarı, lasso'nun ilişkili değişkenlerden rastgele birini seçme eğilimini yumuşatır.
- **Sonuca etkisi.** 1'e yaklaştıkça model seyrekleşir ve yorumlanması kolaylaşır.
- **Dikkat.** `saga` yavaş yakınsayabilir; çok küçük C ve çok terimde yakınsama uyarısı görülebilir.

5.21.3 `Model terms`

- **Ne işe yarar.** GLM ile aynı (§5.6.1); varsayılan burada `linear`'dır (sdm paketinin glmnet'i gibi).
- **Ne zaman, nasıl ayarlanır.** Ceza aşırı uyumu yavaşlattığı için `quadratic` ya da `interactions` GLM'e göre daha güvenle denenebilir (ipucu).
- **Sonuca etkisi.** Terim arttıkça esneklik artar; ceza fazla terimleri sıfırlayabilir.
- **Dikkat.** Ayar sekmesi terim yapısını değiştirmez; buradaki değer sabit kalır.

5.22 CART: sınıflandırma ağacı



Şekil 5.18. CART sekmesi: yöntem notu (Breiman ve ark. 1984; sdm::rpart); max_depth 5, min_samples_leaf 5, ccp_alpha 0.0000.

Aile ve uygulama. Tek bir sınıflandırma ağacı (Breiman ve ark. 1984). scikit-learn `DecisionTreeClassifier`; uygunluk, hücrenin düştüğü yapraktaki varlık oranıdır (`sdm_core.py:3506-3515`). Varlık–sözde yokluk.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`max_depth:`	En çok derinlik	5	0 (`no limit`) – 30	`tabs/tab_models.py:957-960`; `sdm_core.py:3509-3511`
`min_samples_leaf:`	Yapraktaki en az satır	5	1 – 500	`tabs/tab_models.py:961-963`
`ccp_alpha:`	Maliyet-karmaşıklık budaması	0.0000	0.0 – 0.5	`tabs/tab_models.py:964-967`

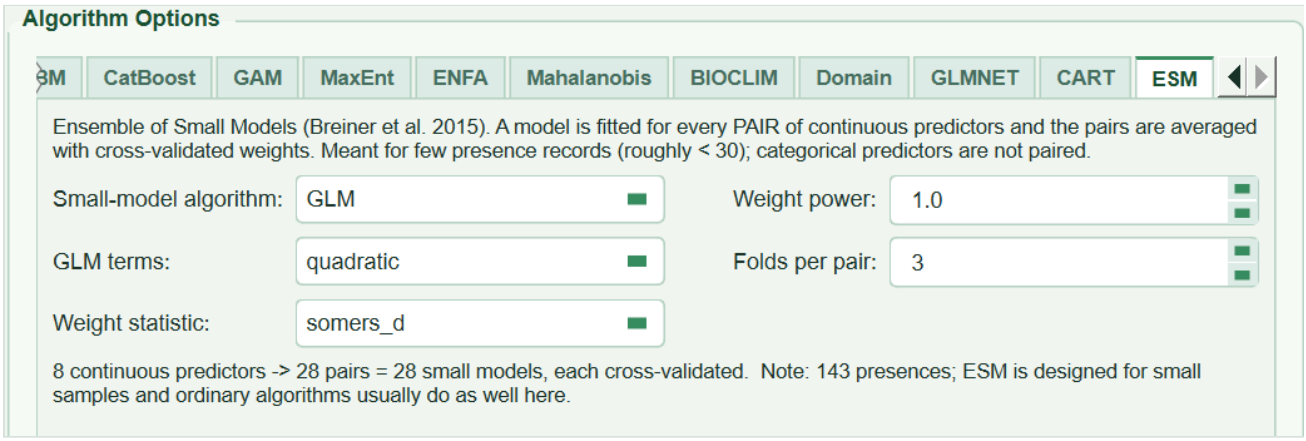
5.22.1 `max_depth` ve `min_samples_leaf`

- **Ne işe yarar.** Ağacın en çok derinliği (0 = sınırsız) ve yaprak büyüklüğü.
- **Ne zaman, nasıl ayarlanır.** Açıklama amaçlı kullanımda 3–5 derinlik okunabilir bir ağaç verir. Yaprak oranları olasılık olduğundan çok küçük yapraklar 0/1 yüzey üretir (ipucu).
- **Sonuca etkisi.** Derinlik arttıkça harita daha çok basamaklı bölgeye ayrılır ve eğitim uyumu artar.
- **Dikkat.** Tek ağaç kararsızdır: veri biraz değişince ağacın yapısı büyük ölçüde değişebilir.

5.22.2 `ccp_alpha`

- **Ne işe yarar.** Ağacı, eklenen her yaprağın karmaşıklık bedeline göre budar; rpart'ın `cp` parametresinin karşılığıdır, ama ölçeği aynı değildir.
- **Ne zaman, nasıl ayarlanır.** Ayar sekmesi 0/0,002/0,01 dener.
- **Sonuca etkisi.** Büyüdükçe ağaç küçülür.
- **Dikkat.** rpart varsayılan olarak `cp` = 0,01 ile budar; HABITUS'un varsayılanı 0'dır (budama yok), sınırlamayı derinlik ve yaprak boyu yapar.

5.23 ESM: küçük modeller topluluğu



Şekil 5.19. ESM sekmesi: yöntem notu; Small-model algorithm GLM, GLM terms quadratic, Weight statistic somers_d, Weight power 1.0, Folds per pair 3; altta örnek koşu için bilgi satırı (8 sürekli öngörücü, 28 çift; 143 varlık notu).

Aile ve uygulama. Çok az kayıtlı türler için önerilen yöntem (Lomba ve ark. 2010; Breiner ve ark. 2015). Her sürekli öngörücü çifti için iki değişkenli küçük bir model uydurulur; çiftler, kendi çapraz doğrulama başarılarıyla ağırlıklandırılıp ortalanır (`sdm_core.py:4370-4455`, `sdm_niche.py:130-265`). p öngörücüde $p(p-1)/2$ çift vardır; 8 öngörücüde 28. Kategorik değişkenler çiftlere girmez. 100'den fazla çift olursa uyarı verilir (`sdm_niche.py:39`).

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Small-model algorithm:`	Küçük modellerin algoritması	`GLM`	`GLM`, `RF`, `MAXENT`	`tabs/tab_models.py:979-983`; `sdm_core.py:4380-4385`, `4407-4414`
`GLM terms:`	GLM küçük modellerinin terimleri	`quadratic`	`quadratic`, `linear`	`tabs/tab_models.py:1004-1005`; `sdm_core.py:4409`
`Weight statistic:`	Çift ağırlığının ölçütü	`somers_d`	`somers_d`, `auc`, `boyce`	`tabs/tab_models.py:984-991`; `sdm_niche.py:142-157`
`Weight power:`	Ağırlığın üssü	1.0	0.0 – 10.0	`tabs/tab_models.py:992-996`
`Folds per pair:`	Her çifti puanlayan kat sayısı	3	2 – 10	`tabs/tab_models.py:997-1003`; `sdm_core.py:4296-4368`

Sayfanın altındaki bilgi satırı seçilen öngörücülerden çift sayısını hesaplar; 50 ve daha çok varlıkta ESM'nin küçük örneklemler için tasarlandığını hatırlatır (`tabs/tab_models.py:1022-1044`).

5.23.1 `Small-model algorithm` ve `GLM terms`

- **Ne işe yarar.** Her çiftin modeli. GLM, ana GLM ile aynı ayarlarla (C = 1) ve burada seçilen terimlerle; RF 200 ağaç ve yaprak boyu 2 ile (arayüzde değiştirilemez); MaxEnt ise MaxEnt sekmesindeki ayarlarla ve MaxEnt arka planının ilgili iki sütunuyla uydurulur (`sdm_core.py:4407-4445`).
- **Ne zaman, nasıl ayarlanır.** Doğrusal + karesel GLM olağan seçimdir (Breiner ve ark. 2015). RF daha yavaş, MaxEnt çok daha yavaştır. flexsdm'in `esm_glm` varsayılanı yalnız doğrusal terimdir; onunla karşılaştırma için `linear` seçilir.
- **Sonuca etkisi.** Karesel terimler her çiftte optimumlu tepkiye izin verir.
- **Dikkat.** `Balance classes` yalnız GLM ve RF tabanında küçük modellere uygulanır; MaxEnt tabanında yok sayılır ve kayda "not supported (ignored)" yazılır (`sdm_core.py:4402-4406`).

5.23.2 `Weight statistic` ve `Weight power`

- **Ne işe yarar.** Her çiftin ağırlığı, çiftin kat dışı (out-of-fold) tahminlerinden hesaplanan ölçütün `power` üssüdür. `somers_d` = $\max(2(\mathrm{AUC} - 0{,}5),\ 0)$; `auc` = ham AUC; `boyce` = $\max(\mathrm{Boyce},\ 0)$. Şanstan iyi olmayan çift `somers_d` ve `boyce` ile 0 ağırlık alır.
- **Ne zaman, nasıl ayarlanır.** Breiner ve ark. (2015) Somers' D kullanır. `auc` ağırlıkları neredeyse eşittir; ayırt edici olması için üs 1'den büyük seçilmelidir. Üs 0 eşit ağırlık demektir.
- **Sonuca etkisi.** Üs büyüdükçe topluluk en iyi birkaç çifte yoğunlaşır.
- **Dikkat.** Hiçbir çift şanstan iyi değilse eşit ağırlık kullanılır ve "No small model beat chance in cross-validation; the ensemble uses equal weights." uyarısı verilir (`sdm_niche.py:260-263`).

5.23.3 `Folds per pair`

- **Ne işe yarar.** Çiftlerin puanlandığı iç kat sayısı. Katlar ana çapraz doğrulama tasarımını (mekânsal blok, çevresel, bant, dama tahtası) ESM'in eğitildiği satırlara uygular; tasarım uygulanamazsa ya da tasarım `random` / `jackknife` ise rastgele tabakalı katlar kullanılır (`sdm_core.py:4296-4368`). Kat sayısı varlık sayısından büyük olamaz.
- **Ne zaman, nasıl ayarlanır.** Çok az varlıkta (10–20) 2–3 kat; daha çok varlıkta 4–5.
- **Sonuca etkisi.** Kat arttıkça ağırlıklar daha kararlı, süre daha uzun olur.
- **Dikkat.** Tasarım uygulanamadığında bir kez "ESM: the … design could not be applied to the folds that weight the small models (…); random stratified folds were used for them." uyarısı verilir.

R ile karşılaştırma. ecospat `ecospat.ESM.Modeling` aynı fikri biomod2 modelleriyle uygular. HABITUS'un uygulaması kendine özgüdür; ecospat ile sayısal karşılaştırma testi yoktur.

5.24 Çıktılar ve sonraki adıma geçenler

- **Kaydedilen ayarlar:** bu sayfadaki bütün değerler (işaretli olmayan algoritmalarınki de) `habitus_run_config.json` → `settings.models.algorithm_options` alanına; MaxEnt'e gerçekte verilen ayarlar (`maxent_settings_used`: elapid argümanları, tohum, arka plana eklenen varlık sayısı, çözücü) ayrıca yazılır. ESM çalıştırılmışsa çift sayısı, ağırlıklar ve en iyi on çift `esm` alanına yazılır (`sdm_core.py:4457-4478`). Sınıf dengesinin her algoritmaya nasıl uygulandığı `class_weight_applied` alanındadır.
- **Modeller:** her algoritma × PA kümesi × kat ve tam model; ③'ün `Evaluation Results` alt sekmesinde skorlar, ④ Evaluation'da tepki eğrileri, önem grafikleri ve ROC eğrileri.
- **Haritalar:** algoritma başına güncel uygunluk ve topluluk haritası (`Current Distribution Map`), ⑤ Future Scenarios'ta aynı modellerle projeksiyon.

5.25 Uzman soruları

S1. On sekiz algoritmadan hangilerini seçmeliyim; varsayılan neden yalnız GLM ve RF? Cevap: Program açılışta yalnız GLM ve RF'yi işaretler (`tabs/tab_models.py:129-132`): biri yorumlanabilir parametrik regresyon, öteki esnek bir ağaç topluluğu; ikisi de ek paket gerektirmez ve hızlıdır. Seçim için kesin bir kural kodda yoktur. Tipik bir çalışma farklı ailelerden birkaç yöntemi (GLM ya da GAM, RF ya da BRT, MaxEnt) çalıştırıp topluluk haritasıyla birleştirir. Çok az kayıtta (yaklaşık 30'un altı) ESM, GLMNET ya da doğrusal/karesel özellikli MaxEnt; yalnız varlık kaydı olan ve sözde yoklukla çalışmak istenmeyen durumlarda MaxEnt, ENFA, Mahalanobis, BIOCLIM ve Domain düşünülür. Valavi ve ark. (2022), yalnız-varlık verisinde karşılaştırmalı başarımını çok sayıda tür üzerinde inceler; sonuçları HABITUS'un uygulamalarıyla doğrudan aktarılamaz, yine de yöntem ailelerinin genel davranışı için başvurulabilir.

S2. Varsayılan değerler nereden geliyor? Cevap: Üç kaynaktan. (1) Kütüphane varsayılanları: RF (500 ağaç, `sqrt`, yaprak 1; randomForest ile aynı), SVM, ANN'in alpha'sı, GAM (20 spline, lam 0,6; pygam), XGBoost'un cezaları. (2) Yayın: BRT öğrenme hızı ve torbalama oranı Elith ve ark. (2008) (`sdm_core.py:3530-3548`), MaxEnt menteşe sayısı ve β = 1 maxnet, ESM Breiner ve ark. (2015). (3) HABITUS'un kendi eşli çapraz doğrulama karşılaştırmaları: GBM'in kütük ağaçları ve BRT'nin derinliği 2 (kod açıklamaları `sdm_core.py:3517-3522`, `3532-3537`), GLM'in C = 1 cezası (ipucu, `tabs/tab_models.py:741-745`). Bu karşılaştırmaların ayrıntısı kod açıklamalarında belirtilen iç notlardadır; kitabın konusu değildir. Bir yayında ayarlar "HABITUS varsayılanları" diye değil, `habitus_run_config.json`'daki sayılarla raporlanmalıdır.

S3. HABITUS'un MaxEnt'i R maxnet ile aynı sonucu verir mi? Cevap: Amaç işlevi aynıdır: özellik başına L1 cezası, varlık 1 / arka plan 100 ağırlığı, son λ, varlıkların arka plana eklenmesi (`sdm_core.py:3776-3836`, `4698-4707`). `tests/test_maxent_maxnet.py`, Rutrough ve ark. (2019) verisinde üç ayarda R maxnet 0.1.4 tahminleriyle korelasyonun 0,995'i aşmasını ve ortalama mutlak farkın 0,02'nin altında kalmasını sınar. Kalan farkların kaynağı çözücüdür (glmnet'in koordinat inişi yerine liblinear) ve kesişimin çok küçük ama sıfır olmayan cezası. Özellik üretimi elapid'e aittir; elapid ile maxnet'in menteşe düğümlerinin yerleştirilmesi bu test verisinde uyumludur, her veri için ayrıca sınanmamıştır. maxnet ile birebir karşılaştırma için `Auto (maxnet default rule)` işaretlenmelidir; HABITUS'un varsayılan `linear,hinge,product` seti maxnet'in kuralıyla aynı değildir.

S4. MaxEnt'te varlıkların arka plana eklenmesi neden önemli? Cevap: Maxent ve maxnet'in varsayılan davranışı budur; varlığın bulunduğu ortam, kullanılabilir ortamın da bir parçasıdır. Eklenmezse varlık hücrelerinin ortamı arka plan dağılımında eksik temsil edilir ve modelin ham çıktısı ile normalleştirme sabiti maxnet'ten farklılaşır. HABITUS yalnız öngörücü değerleri arka planda zaten bulunmayan varlıkları ekler (`sdm_core.py:4702-4706`); örnek koşunun tam modelinde 143 varlıktan 91'i eklenmiştir.

S5. Yalnız-varlık yöntemleri (MaxEnt, ENFA, Mahalanobis, BIOCLIM, Domain) neden sözde yokluklarla değerlendiriliyor? Cevap: HABITUS bütün algoritmaları aynı test katında, aynı varlık ve sözde yokluk satırlarıyla puanlar (`sdm_core.py:5530-5540`); böylece skorlar algoritmalar arasında karşılaştırılabilir ve topluluk ağırlıkları tutarlı olur. Bu durumda AUC "varlığı sözde yokluktan ayırma" gücüdür, gerçek yokluğa karşı başarı değildir; Boyce indeksi ise yalnız varlıklara dayandığından yalnız-varlık verisi için daha uygun bir ölçüttür. Test katında varlık kalmazsa eğitim varlıklarının beşte biri teste taşınır (`sdm_core.py:5455-5468`); bu durumda kat başka algoritmaların katından farklılaşır. Kullanıcı yalnız-varlık modellerini ayrıca değerlendirmek isterse Boyce ve OR_10p sütunlarına odaklanmalıdır.

S6. Sınıf dengesi (`Balance classes`) açılmalı mı; prevalans neyi değiştirir? Cevap: Varsayılan kapalıdır. Örnek koşuda sözde yokluk : varlık oranı 7,0 : 1'dir (`pa_to_presence_ratio` = 6,993). Denge açıldığında desteklenen algoritmalarda iki sınıfın toplam ağırlığı eşitlenir (§5.4.5). Bu, tahmin edilen değerin anlamını değiştirir: çıktı artık sözde yokluk oranını yansıtmaz, eşikler yeniden türetilir; sıralama ölçütleri (AUC) çoğunlukla az etkilenir (ipucu, `tabs/tab_models.py:346-348`). Sözde yokluk sayısı zaten keyfî olduğundan mutlak "olasılık" yorumlanmıyorsa denge şart değildir. Önemli olan, dengenin GAM, MaxEnt ve yalnız-varlık yöntemlerinde hiç uygulanmadığıdır; bir toplulukta dengeli ve dengesiz modeller karışabilir. Uygulamanın dökümü `class_weight_applied` alanındadır.

S7. Sonuçlar tekrarlanabilir mi; hangi kaynaklar rastgelelik katar? Cevap: Evet; tek tohumla. Bütün rastgele öğrenicilere ① Data'daki tohum verilir (`sdm_core.py:3474-3475`); MaxEnt'e de `random_state` geçer (`sdm_core.py:4643`), permütasyon önemi tohumdan türetilen üreteçle hesaplanır (`sdm_core.py:4847-4853`). Rastgelelik katan ayarlar RF'nin bootstrap'ı, GBM/BRT/XGBoost/LightGBM'in alt örneklemesi, CatBoost'un `random_strength`'i ve `bagging_temperature`'ı, ANN'in başlangıç ağırlıkları, SVM'nin Platt katları ve ESM'nin iç katlarıdır. Çok iş parçacıklı kütüphaneler (RF, XGBoost, LightGBM `n_jobs = -1`) farklı işlemci sayısında kayan nokta toplama sırası yüzünden son basamaklarda fark verebilir; bu kodda sınanmamıştır. Kütüphane sürümleri `habitus_run_config.json` → `environment` alanında kaydedilir; aynı sonuç için aynı sürümler gerekir.

S8. GLM neden cezalı; R'deki glm ile sonucumu nasıl karşılaştırırım? Cevap: Karesel ve etkileşim terimleri az kayıtta katsayıları şişirir; HABITUS standartlaştırılmış terimlere hafif bir L2 cezası uygular (`sdm_core.py:3481-3492`). R `glm()` cezasızdır. Karşılaştırma için `Regularisation C` = 1e6 girilir; terim yapısı R'de `y ~ x + I(x^2)` (quadratic) ya da `y ~ poly(x1, x2, degree = 2, raw = TRUE)` (interactions) ile kurulur. biomod2'nin GLM'i ayrıca AIC'ye göre adımsal seçim yapar; HABITUS'ta adımsal seçim yoktur, bu nedenle biomod2 ile katsayılar birebir aynı olmaz.

S9. GAM'ın sonuçları mgcv ile neden farklı olabilir? Cevap: HABITUS pygam `LogisticGAM`'ı sabit bir düzgünleştirme cezasıyla (`lam`, varsayılan 0,6) ve öngörücü başına 20 B-spline tabanıyla uydurur (`sdm_core.py:4500-4523`). mgcv ise cezayı veriden (GCV/REML) seçer ve farklı bir taban kullanır. Aynı veride mgcv daha düzgün ya da daha kıvrımlı eğri seçebilir. Yakınlaştırmak için ayar sekmesinde `lam` aranabilir; ama iki model yine özdeş olmaz. GAM'da etkileşim terimi ve sınıf dengesi yoktur.

S10. BRT ağaç sayısını dismo `gbm.step` gibi kendisi seçiyor mu? Cevap: Hayır. `n_estimators` sabittir (BRT 1000, GBM 200) ve erken durdurma kullanılmaz (`sdm_core.py:3523-3548`). dismo `gbm.step` ağaç sayısını 10 katlı çapraz doğrulama sapmasının en küçük olduğu yerde durdurur. HABITUS'ta yakın bir sonuç için GBM ayar sekmesinde `n_estimators` × `learning_rate` ızgarası aranır ve seçilen değer BRT'ye elle yazılır. Ayrıca Elith ve ark. `tree.complexity` = 5 kullanmışken HABITUS BRT'nin derinliği 2'dir.

S11. XGBoost, LightGBM ve CatBoost'un varsayılanları SDM için uygun mu? Cevap: Bu üçünün varsayılanları (XGBoost derinlik 6 ve 500 tur; LightGBM 31 yaprak, sınırsız derinlik ve 500 tur; CatBoost derinlik 4 ve 300 tur) GBM ve BRT'ye göre çok daha karmaşık modellerdir (`sdm_core.py:3586-3647`). Yardım sekmesindeki karşılaştırma tablosu bunlar için en az 100 kayıt ve aşırı uyum eğilimi not eder. Birkaç yüz

varlıktan az veride derinliği 2–3'e, yaprak sayısını 7–15'e indirmek ve sonucu eğitim–test AUC farkıyla (dAUC) denetlemek önerilir. LightGBM'de `subsample` ayarının etkisi olmadığına dikkat edilmelidir (§5.13.3).

S12. Kategorik bir öngörücü (ör. arazi örtüsü) eklediğimde her algoritma onu aynı biçimde mi kullanıyor? Cevap: Hayır (§5.4.3). Regresyon ve çekirdek yöntemleri sınıf başına 0/1 sütun, GAM faktör terimi, MaxEnt kategorik özellik kullanır; ağaç yöntemleri tamsayı kodunu sayısal bölünmeyle işler; zarf ve uzaklık yöntemleri değişkeni dışarıda bırakır ve günlüğe not düşer (`sdm_core.py:4010-4036`). Bu yüzden bir topluluktaki modellerin bir kısmı kategorik değişkeni hiç görmeyebilir. Eğitimde görülmemiş bir sınıf (yalnız projeksiyon alanında bulunan) regresyon yöntemlerinde, GAM'da ve MaxEnt'te "sınıf etkisi yok" olarak, ağaçlarda ise sayısal kodunun düştüğü dalda değerlendirilir.

S13. ENFA ve Mahalanobis puanları neden 0–1 aralığına yeniden ölçekleniyor; harita karşılaştırılabilir mi? Cevap: Ham puan $\exp(-D^2/2)$ çok öngörücüde hızla sıfıra yaklaşır. HABITUS, uydurma sırasında eğitim satırlarının puan aralığını saklar ve her tahminde aynı aralığı kullanır (`sdm_core.py:3719-3743`). Bu nedenle bir hücre, test katında da projeksiyonda da aynı değeri alır ve eşik iki ortamda aynı anlamdadır. Yine de puanın ölçeği algoritmaya özgüdür; ENFA'nın 0,6'sı ile GLM'in 0,6'sı aynı olasılık değildir. Topluluk ağırlıkları ve ikili eşikler bu yüzden algoritma başına ayrı belirlenir (Bölüm 6).

S14. Ayarlanan parametreleri bir yayında nasıl raporlamalıyım? Cevap: `habitus_run_config.json` içindeki `algorithm_options` (girilen değerler), `maxent_settings_used` (MaxEnt'e gerçekten verilenler ve çözücü), `class_weight`, `clamp`, `random_seed` ve `environment` (kütüphane sürümleri) birlikte raporlanmalıdır. Ayar sekmesi kullanıldıysa ızgara, seçim kuralı ve seçilen nokta da yazılmalıdır; ayar ile değerlendirme aynı katlarda yapıldığında skorun iyimser olabileceği Bölüm 6'da tartışılır. ODMAP raporu (Bölüm 11) bu bilgilerin çoğunu otomatik taşır.

5.26 Sık hatalar ve tuzaklar

- **Paket kurulu değil:** kutu gri ve `[not installed]` ekli. Örnek ipucu: "Not installed — pip install elapid". Kurulumdan sonra program yeniden başlatılmalıdır.
- **Hiç algoritma seçilmeden çalıştırma:** "Select at least one algorithm."
- **ANN katmanlarını yanlış yazmak** (ör. `100;50`, `100 50`, `0`): "ANN hidden layers: … is not a list of positive whole numbers separated by commas (e.g. 100, 50)."
- **RF `max_features`'a `auto` ya da `%50` yazmak:** "RF max_features: … is not sqrt, log2, a whole number of variables or a fraction between 0 and 1." Oran 0,5 olarak yazılmalıdır.
- **GAM `n_splines` ≤ 3:** arayüz alt sınırı 4'tür; betikle verilirse "GAM n_splines must exceed the spline order (3)."
- **MaxEnt özellik adını yanlış yazmak** (ör. `hinges`, `quad`): elapid ayarı reddeder ve MaxEnt "MaxEnt could not be fitted…" ya da "MaxEnt could not be configured with the settings you chose…" iletisiyle başarısız olur. Geçerli adlar: `linear`, `quadratic`, `hinge`, `product`, `threshold`.
- **MaxEnt'te `Auto` işaretliyken metin kutusunu değiştirmek:** kutu devre dışıdır ve yok sayılır.
- **`N threshold features`'ı değiştirip `threshold` sınıfını seçmemek:** değer kullanılmaz.
- **Yalnız-varlık yöntemlerine kategorik değişken vermek:** değişken sessizce değil, günlükte "NOTE: … is an envelope/distance model for continuous predictors…" iletisiyle dışarıda bırakılır; yalnız kategorik değişken varsa "… needs at least one continuous predictor (only categorical ones were given)." hatası verilir.
- **ESM'i çok öngörücüyle çalıştırmak:** 15 öngörücü 105 çift ve 105 ayrı çapraz doğrulanmış model demektir; bilgi satırı "WARNING: more than 100 pairs…" der.
- **ESM'i çok kayıtlı türde kullanmak:** 50 ve üstü varlıkta bilgi satırı ESM'nin küçük örneklemler için tasarlandığını hatırlatır; olağan algoritmalar genellikle o kadar iyidir.
- **Mahalanobis'te `LinAlgError`:** ε artırılır ya da Ledoit–Wolf açık tutulur; güçlü ilişkili öngörücüler ② Variables'ta elenir.
- **`Balance classes`'ın her algoritmayı etkilediğini sanmak:** GAM, MaxEnt, ENFA, Mahalanobis, BIOCLIM ve Domain'de etkisizdir.
- **LightGBM `subsample`'ı ayarlamak:** satır alt örneklemesi HABITUS'ta etkin olmadığından değer sonucu değiştirmez.
- **Varsayılan derin boosting modellerini az kayıtla kullanmak:** XGBoost/LightGBM'de eğitim AUC'si 1'e yaklaşırken test AUC'si düşer; dAUC sütunu izlenmelidir.

5.27 Kaynaklar

Anderson, C.B., 2023. elapid: Species distribution modeling tools for Python. Journal of Open Source Software 8, 4930. https://doi.org/10.21105/joss.04930

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32. https://doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees. Wadsworth, Belmont (yeni baskı: Routledge, 2017). https://doi.org/10.1201/9781315139470

Breiner, F.T., Guisan, A., Bergamini, A., Nobis, M.P., 2015. Overcoming limitations of modelling rare species by using ensembles of small models. Methods in Ecology and Evolution 6, 1210–1218. https://doi.org/10.1111/2041-210X.12403

Carpenter, G., Gillison, A.N., Winter, J., 1993. DOMAIN: a flexible modelling procedure for mapping potential distributions of plants and animals. Biodiversity and Conservation 2, 667–680. https://doi.org/10.1007/BF00051966

Chen, T., Guestrin, C., 2016. XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785

Elith, J., Leathwick, J.R., Hastie, T., 2008. A working guide to boosted regression trees. Journal of Animal Ecology 77, 802–813. https://doi.org/10.1111/j.1365-2656.2008.01390.x

Farber, O., Kadmon, R., 2003. Assessment of alternative approaches for bioclimatic modeling with special emphasis on the Mahalanobis distance. Ecological Modelling 160, 115–130. https://doi.org/10.1016/S0304-3800(02)00327-7

Friedman, J., Hastie, T., Tibshirani, R., 2010. Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software 33(1), 1–22. https://doi.org/10.18637/jss.v033.i01

Hastie, T., Tibshirani, R., 1986. Generalized additive models. Statistical Science 1, 297–310. https://doi.org/10.1214/ss/1177013604

Hirzel, A.H., Hausser, J., Chessel, D., Perrin, N., 2002. Ecological-niche factor analysis: how to compute habitat-suitability maps without absence data? Ecology 83, 2027–2036. https://doi.org/10.1890/0012-9658(2002)083[2027:ENFAHT]2.0.CO;2

Ledoit, O., Wolf, M., 2004. A well-conditioned estimator for large-dimensional covariance matrices. Journal of Multivariate Analysis 88, 365–411. https://doi.org/10.1016/S0047-259X(03)00096-4

Lomba, A., Pellissier, L., Randin, C., Vicente, J., Moreira, F., Honrado, J., Guisan, A., 2010. Overcoming the rare species modelling paradox: a novel hierarchical framework applied to an Iberian endemic plant. Biological Conservation 143, 2647–2657. https://doi.org/10.1016/j.biocon.2010.07.007

Merow, C., Smith, M.J., Silander, J.A., 2013. A practical guide to MaxEnt for modeling species' distributions: what it does, and why inputs and settings matter. Ecography 36, 1058–1069. https://doi.org/10.1111/j.1600-0587.2013.07872.x

Muscarella, R., Galante, P.J., Soley-Guardia, M., Boria, R.A., Kass, J.M., Uriarte, M., Anderson, R.P., 2014. ENMeval: An R package for conducting spatially independent evaluations and estimating optimal model complexity for Maxent ecological niche models. Methods in Ecology and Evolution 5, 1198–1205. https://doi.org/10.1111/2041-210X.12261

Nix, H.A., 1986. A biogeographic analysis of Australian elapid snakes. In: Longmore, R. (Ed.), Atlas of Elapid Snakes of Australia. Australian Flora and Fauna Series 7. Australian Government Publishing Service, Canberra, 4–15.

Oshiro, T.M., Perez, P.S., Baranauskas, J.A., 2012. How many trees in a random forest? In: Perner, P. (Ed.), Machine Learning and Data Mining in Pattern Recognition. Lecture Notes in Computer Science 7376. Springer, Berlin, 154–168. https://doi.org/10.1007/978-3-642-31537-4_13

Phillips, S.J., Anderson, R.P., Schapire, R.E., 2006. Maximum entropy modeling of species geographic distributions. Ecological Modelling 190, 231–259. https://doi.org/10.1016/j.ecolmodel.2005.03.026

Phillips, S.J., Anderson, R.P., Dudík, M., Schapire, R.E., Blair, M.E., 2017. Opening the black box: an open-source release of Maxent. Ecography 40, 887–893. https://doi.org/10.1111/ecog.03049

Rutrough, A., Widick, I.V., Bean, W.T., 2019. Reconstruction of the historical range alters niche estimates in an endangered rodent. Ecography 42, 1742–1751. https://doi.org/10.1111/ecog.04238

Valavi, R., Elith, J., Lahoz-Monfort, J.J., Guillera-Arroita, G., 2021. Modelling species presence-only data with random forests. Ecography 44, 1731–1742. https://doi.org/10.1111/ecog.05615

Valavi, R., Guillera-Arroita, G., Lahoz-Monfort, J.J., Elith, J., 2022. Predictive performance of presence-only species distribution models: a benchmark study with reproducible code. Ecological Monographs 92, e01486. https://doi.org/10.1002/ecm.1486

Velazco, S.J.E., Rose, M.B., de Andrade, A.F.A., Minoli, I., Franklin, J., 2022. flexsdm: An R package for supporting a comprehensive and flexible species distribution modelling workflow. Methods in Ecology and Evolution 13, 1661–1669. https://doi.org/10.1111/2041-210X.13874

Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society Series B 67, 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x

6. Modeller II: çapraz doğrulama, eşik, topluluk ve sonuçlar (③ Models)

Sekme: ③ Models + Current Map · Kod: `tabs/tab_models.py`, `tabs/tab_tuning.py` (arayüz); `sdm_core.py` `SDMModeler`, `EnsembleModeler`, `Projector`, `evaluate_all`; `sdm_spatial.py` (bloklar, variogram, katlar, SSB); `sdm_tuning.py` (ızgara araması)

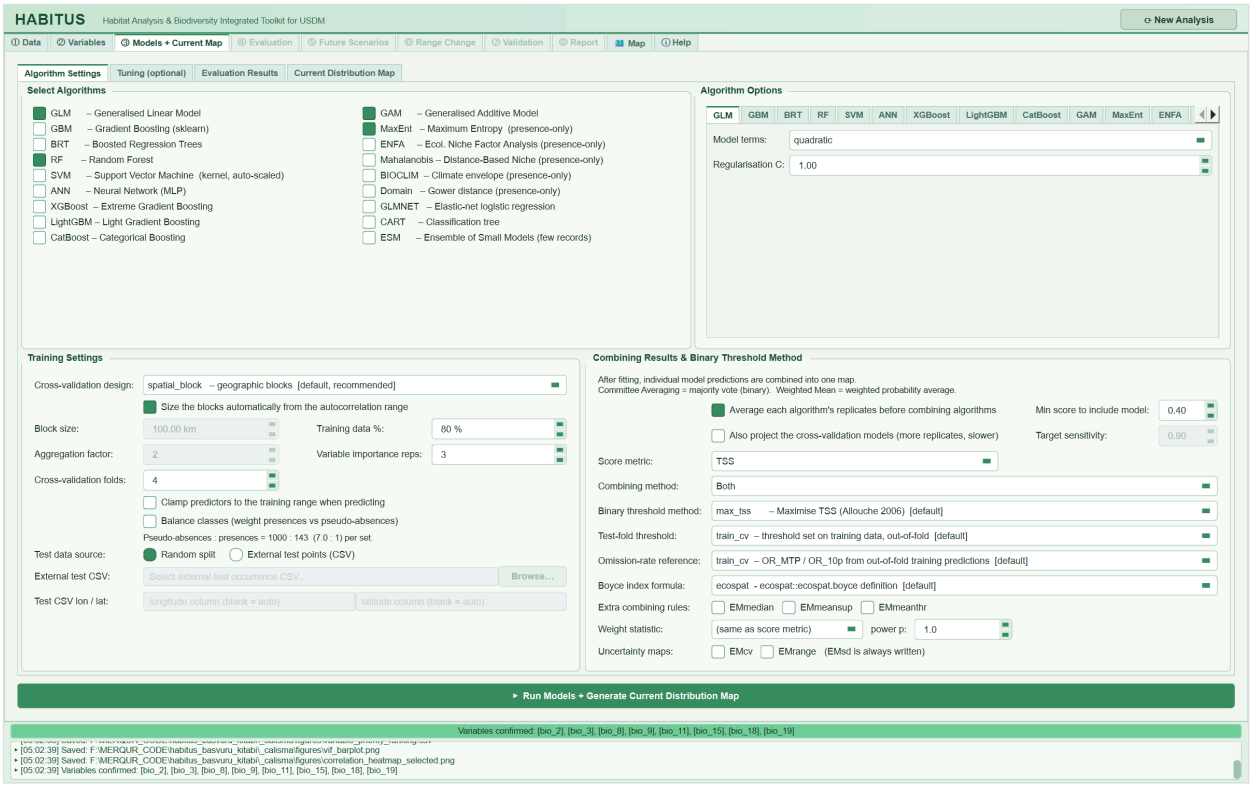
6.1 Bu sekme ne yapar

③ Models sekmesi, ② Variables adımında seçilen öngörücülerle seçilen algoritmaları kurar, her birini çapraz doğrulamayla (CV) sınar, tüm veriyle yeniden kurulan modelleri bugünkü iklim katmanlarına izdüşürür ve bunları bir topluluk (ensemble) haritasında birleştirir. Bölüm 5 algoritmaların kendi seçeneklerini (`Algorithm Options`) anlatır; bu bölüm aynı sekmedeki geri kalan her şeyi anlatır: hangi algoritmaların çalışacağı, verinin katlara nasıl bölündüğü, ikili (var/yok) eşiğin nasıl seçildiği, topluluğun nasıl kurulduğu, isteğe bağlı ayar araması ve sonuç alt sekmeleri.

- **Girdi:** ① Data'da üretilen yalancı yokluk (pseudo-absence, PA) setleri, ② Variables'ta onaylanan sürekli ve kategorik öngörücüler, eğitim raster'ları ve tohum (seed). Bunlar yoksa düğme `"Complete ① Data and ② Variables steps first."` iletisini verir (`tabs/tab_models.py:1394-1395`).
- **Çıktı:** her algoritma × PA seti × CV koşusu için değerlendirme skorları; tüm veriyle kurulan (`Full`) modellerin olasılık ve ikili haritaları; algoritma başına ortalama ve yayılım haritaları; `EM*` topluluk haritaları; `evaluation_scores.csv`, `habitus_run_config.json`, eğitim/test nokta dosyaları (§6.10).
- **Kilit düzeni:** Çalıştırma bitince ④ Evaluation, ⑤ Future Scenarios, ⑥ Range Change, ⑦ Validation ve ⑧ Report sekmeleri açılır (`tabs/tab_models.py:1848-1852`).

Tek düğme, `▶ Run Models + Generate Current Distribution Map`, dört işi sırayla yapar (`tabs/tab_models.py:1559-1659`): (1) `SDMModeler.run()` ile CV ve tam veri modelleri, (2) `EnsembleModeler` ile topluluk üyelerinin süzülmesi, (3) `Projector.project()` ile bugünkü iklime izdüşüm ve haritaların yazılması, (4) çalıştırma kaydının (`habitus_run_config.json`) yazılması.

6.2 Ekranın düzeni



Şekil 6.1. ③ Models + Current Map sekmesi çalıştırmadan önce: üstte dört alt sekme, solda Select Algorithms, sağda Algorithm Options, altta Training Settings ile Combining Results && Binary Threshold Method kutuları ve en altta çalıştırma düğmesi.

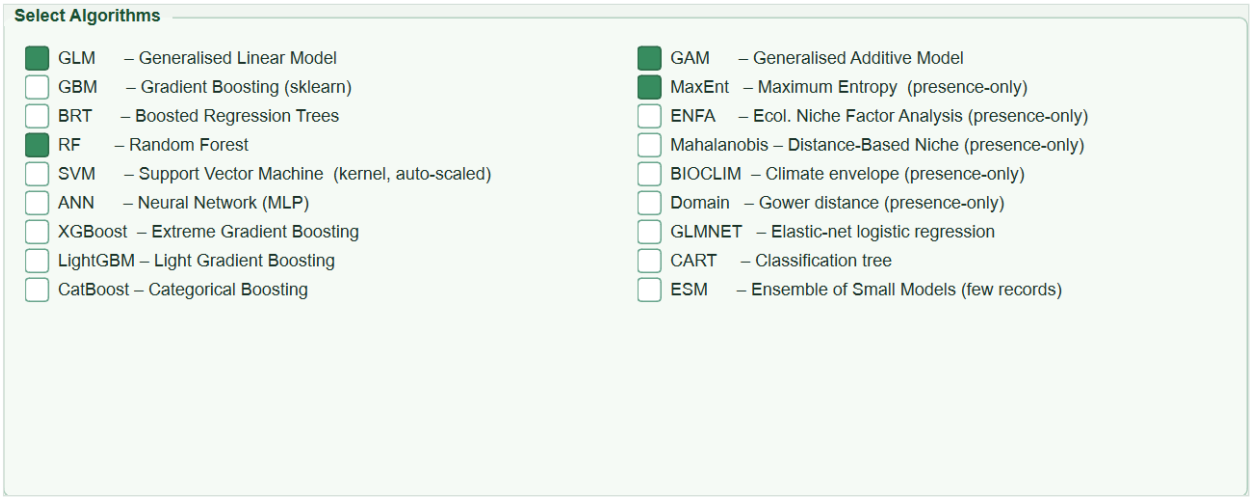
Şekil 6.1. ③ Models + Current Map sekmesi çalıştırmadan önce: üstte dört alt sekme, solda Select Algorithms, sağda Algorithm Options, altta Training Settings ile Combining Results && Binary Threshold Method kutuları ve en altta çalıştırma düğmesi.

Sekmede dört alt sekme ve tüm alt sekmelerin altında duran tek çalıştırma düğmesi vardır (`tabs/tab_models.py:91-110`):

Bölge	İçerik	Bu bölümde
`Algorithm Settings`	`Select Algorithms`, `Algorithm Options`, `Training Settings`, `Combining Results && Binary Threshold Method`	§6.3–6.6 (algoritma seçenekleri: Bölüm 5)
`Tuning (optional)`	Tek algoritma için ızgara araması	§6.7
`Evaluation Results`	Model başına ROC-AUC, TSS, Boyce tablosu	§6.8
`Current Distribution Map`	Durum metni, üretilen haritalar tablosu, çıktı klasörü, harita görüntüleyici	§6.9
Alt şerit	`▶ Run Models + Generate Current Distribution Map`	§6.1

Çalıştırma başlayınca program `Evaluation Results` alt sekmesine geçer, bitince `Current Distribution Map` alt sekmesini açar (`tabs/tab_models.py:1491`, `1835`).

6.3 Algoritma seçimi (Select Algorithms)



Şekil 6.2. Select Algorithms kutusu: on sekiz algoritma iki sütunda; örnek koşuda GLM, RF, GAM ve MaxEnt işaretli.

Şekil 6.2. Select Algorithms kutusu: on sekiz algoritma iki sütunda; örnek koşuda GLM, RF, GAM ve MaxEnt işaretli.

Kutuda on sekiz onay kutusu vardır; her biri bir algoritmayı çalıştırma listesine ekler (`tabs/tab_models.py:125-150`). İşaretli **ve** etkin olan her kutu çalışır (`tabs/tab_models.py:1397-1398`). Hiçbiri seçilmezse `"Select at least one algorithm."` iletisi çıkar.

Arayüzdeki ad (kısa)	Ne yapar	Varsayılan	Kod
`GLM`, `RF`	Genelleştirilmiş doğrusal model; rastgele orman	açık	`tabs/tab_models.py:129-132`
`GBM`, `BRT`, `SVM`, `ANN`	Gradyan artırma (sklearn), artırılmış regresyon ağaçları, destek vektör makinesi, çok katmanlı sinir ağı	kapalı	`tabs/tab_models.py:130-134`
`XGBoost`, `LightGBM`, `CatBoost`	Üç artırma kütüphanesi; kurulu değilse kutu soluk ve `[not installed]`	kapalı	`tabs/tab_models.py:135-137`, `152-168`
`GAM`	Genelleştirilmiş toplamsal model (pygam gerekir)	kapalı	`tabs/tab_models.py:138`
`MaxEnt`, `ENFA`, `Mahalanobis`, `BIOCLIM`, `Domain`	Yalnız varlık kullanan (presence-only) yöntemler	kapalı	`tabs/tab_models.py:139-143`
`GLMNET`, `CART`, `ESM`	Elastik ağ lojistik regresyon, sınıflandırma ağacı, küçük modeller topluluğu	kapalı	`tabs/tab_models.py:144-146`

Ne işe yarar. Seçilen her algoritma her PA setinde, her CV katında ve bir kez de tüm veriyle kurulur (`sdm_core.py:5409-5643`, `5654-5768`). Topluluk haritası yalnız seçilen algoritmalardan oluşur (`sdm_core.py:7844-7855`).

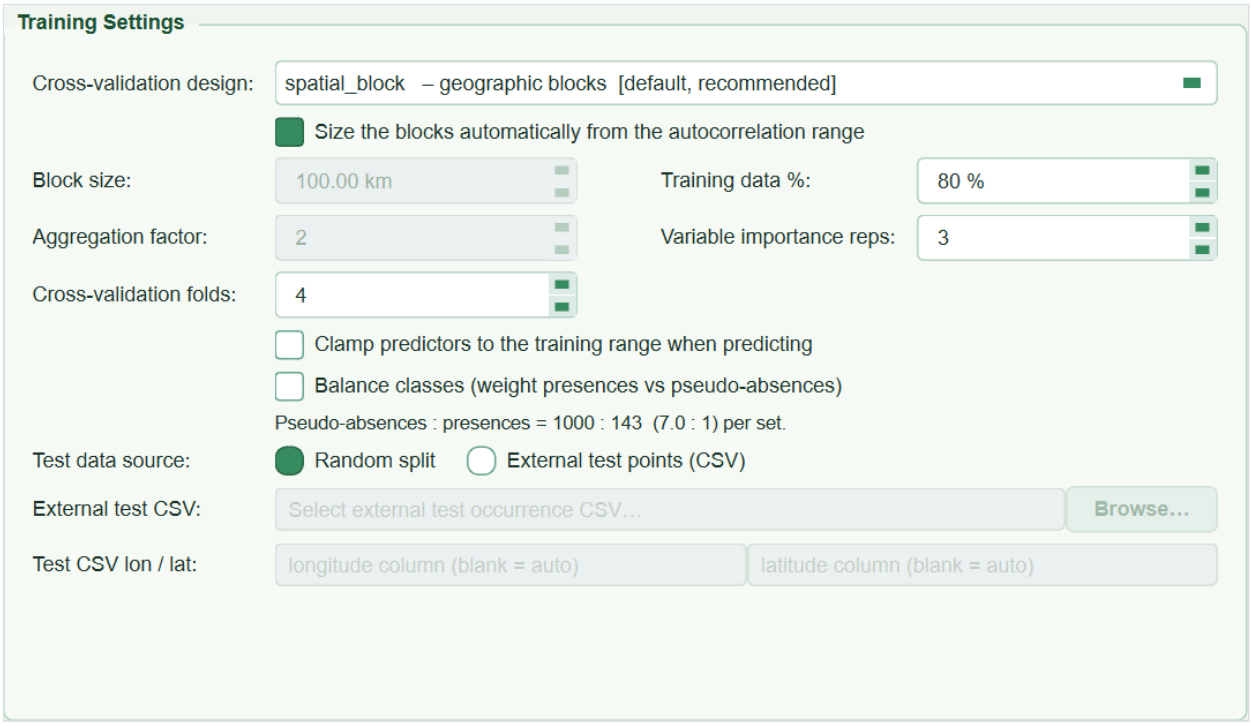
Ne zaman, nasıl ayarlanır. Topluluk yaklaşımı farklı yanlılıkları olan algoritmaların birleşimini öngörür (Araújo ve New 2007); birbirine çok benzeyen algoritmaları (ör. GBM, BRT, XGBoost, LightGBM hepsi birden) eklemek topluluğu tek bir aileye ağırlaştırır. İki katmanlı birleştirme (§6.6.1) bu etkiyi yineleme sayısı

bakımından giderir, aile sayısı bakımından gidermez. Az kayıtlı türlerde (yaklaşık 30'dan az varlık) ESM ve yalnız varlık kullanan yöntemler anlamlıdır (Bölüm 5).

Sonuca etkisi. Süre algoritma sayısıyla doğrusal artar: toplam kurulum sayısı yaklaşık algoritma × PA seti × (kat + 1) kadardır; `Test-fold threshold` = `train_cv` iken her CV kurulumuna üç iç kurulum daha eklenir (§6.5.2). Örnek koşuda MaxEnt tam modeli tek başına 25–28 s sürmüştür (oturum günlüğü).

Dikkat. Programın kendi yüklediği varsayılan yalnız GLM ve RF'dir; şekildeki dört algoritma örnek koşu için elle işaretlenmiştir.

6.4 Eğitim ayarları (Training Settings)



Şekil 6.3. Training Settings kutusu varsayılan değerlerle: spatial_block tasarımı, otomatik blok boyu (Block size ve Aggregation factor soluk), 4 kat, Random split ve kapalı Clamp/Balance kutuları.

Şekil 6.3. Training Settings kutusu varsayılan değerlerle: spatial_block tasarımı, otomatik blok boyu (Block size ve Aggregation factor soluk), 4 kat, Random split ve kapalı Clamp/Balance kutuları.

Bu kutu verinin eğitim ve test parçalarına nasıl bölüneceğini belirler. HABITUS skorlarını çapraz doğrulama katlarının tutulan (hold-out) parçasından üretir; tasarım seçimi bu nedenle raporlanan bütün skorları doğrudan etkiler.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Cross-validation design:`	Katlama tasarımı	`spatial_block`	8 tasarım (§6.4.1)	`tabs/tab_models.py:219-255`
`Size the blocks automatically from the autocorrelation range`	Blok boyunu variogram menzilinden alır	açık	açık / kapalı	`tabs/tab_models.py:265-271`

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Block size:`	Elle blok kenarı	100 km (yalnız otomatik kapalıyken)	1–5000 km	`tabs/tab_models.py:273-279`
`Aggregation factor:`	Dama tahtası hücre çarpanı	2	1–20 (yalnız `checkerboard*`)	`tabs/tab_models.py:256-263`
`Cross-validation folds:`	Kat sayısı	4	1–10 (`jackknife`'ta soluk)	`tabs/tab_models.py:211-216`
`Clamp predictors to the training range when predicting`	Öngörücüleri eğitim aralığına kırpar	kapalı	açık / kapalı	`tabs/tab_models.py:354-364`
`Balance classes (weight presences vs pseudo-absences)`	Sınıfları ters orantılı ağırlıklar	kapalı	açık / kapalı	`tabs/tab_models.py:333-352`
`Test data source:`	Test verisi kaynağı	`Random split`	`Random split` / `External test points (CSV)`	`tabs/tab_models.py:294-299`
`Training data %:`	Yedek değerlendirme bölmesinde eğitim payı	80 %	50–95 %	`tabs/tab_models.py:301-303`
`External test CSV:`	Dış test dosyası	boş	CSV dosyası	`tabs/tab_models.py:305-311`
`Test CSV lon / lat:`	Dış test dosyasında koordinat sütunları	boş (otomatik)	sütun adı	`tabs/tab_models.py:313-320`
`Variable importance reps:`	Permütasyon önemi tekrar sayısı	3	0–10	`tabs/tab_models.py:290-291`

Etkinlik kuralları `_toggle_block` işlevindedir (`tabs/tab_models.py:281-289`): blok kutuları yalnız `spatial_block`, `checkerboard` ve `checkerboard2` için etkindir; `Aggregation factor` yalnız dama tahtası tasarımlarında; `Cross-validation folds` `jackknife` dışında her tasarımda.

6.4.1 Çapraz doğrulama tasarımı (Cross-validation design)

Ne işe yarar. Her PA seti için her satıra bir kat numarası verir (`SDMModeler._build_folds`, `sdm_core.py:4997-5162`). Kat k sırayla test kümesi olur, geri kalan katlar eğitim kümesidir (`sdm_core.py:5429-5450`). Sekiz tasarım vardır:

Seçenek	Katlar nasıl oluşur	Kod
`spatial_block`	Çalışma alanına kare ızgara serilir; her blok bütün hâlinde bir kata verilir. Bloklar, her kattaki varlık ve yokluk sayıları dengeli olacak biçimde 50 tohumlu denemenin en iyisiyle dağıtılır	`sdm_spatial.py:385-511`, `305-357`
`environmental`	Öngörücüler standartlaştırılır, k-means ile `folds` kümeye ayrılır; her küme bir kat	`sdm_spatial.py:514-558`
`random`	Sınıflara göre tabakalı rastgele katlar	`sdm_spatial.py:561-580`
`checkerboard`	Hücre kenarı = blok boyu × `Aggregation factor` olan dama tahtası; iki renk iki kat	`sdm_spatial.py:623-706`
`checkerboard2`	İnce (blok boyu) ve kaba (blok boyu × çarpan) iki dama tahtası; kat = 2 × kaba renk + ince renk, dört kat	`sdm_spatial.py:666-672`
`jackknife`	Her varlık kendi katıdır; yokluklar katlara eşit ve rastgele dağıtılır	`sdm_spatial.py:709-755`
`lon_bands` / `lat_bands`	Boylam ya da enlem boyunca eşit varlık sayılı `folds` şerit	`sdm_spatial.py:758-790`

Ayrıca `Test data source:` = `External test points (CSV)` seçilirse katlar test için kullanılmaz (§6.4.5).

Ne zaman, nasıl ayarlanır. Kümelenmiş kayıtlarda rastgele bölme, test noktasının yanı başında neredeyse aynı öngörücü değerleri taşıyan bir eğitim noktası bırakır ve skorları şişirir (Roberts vd. 2017). Bu nedenle varsayılan `spatial_block`'tur. Gelecek iklim izdüşümü yapılacaksa `environmental`, modelin görmediği koşullara aktarımı daha doğrudan sınar. `checkerboard` ve `checkerboard2` ENMeval'deki aynı adlı bölmelerin karşılığıdır (Muscarella vd. 2014): test ve eğitim noktaları alanın her yerinde iç içe geçer. `jackknife` yaklaşık 25 varlığın altındaki türler içindir (`sdm_spatial.py:594`; Pearson vd. 2007). Şeritler bir enlem ya da boylam gradyanına aktarımı sınar. `random` yalnız eski bir sonucu yeniden üretmek ya da karşılaştırma için tutulmalıdır.

Sonuca etkisi. Tasarım, raporlanan her skoru (AUC, TSS, Boyce), topluluğa giren üyeleri (§6.6.4) ve üyelerin ağırlıklarını belirler; çünkü `Full` modellerin skoru CV katlarının ortalamasıdır (`sdm_core.py:5818-5853`). Bloklu tasarımlar rastgele bölmeden daha düşük, ama daha dürüst skor verir.

Dikkat.

- Tasarım uygulanamazsa (çok az blok, her iki sınıfı içermeyen katlar, koordinat yok) program rastgele katlara düşer ve bunu `"WARNING (design not applied): …"` notuyla günlüğe ve rapora yazar; özet satırda `[BLOCKING FAILED - scores are optimistic]` görünür (`sdm_core.py:5135-5140`, `5933-5934`).
- Varlık ya da yokluk içermeyen katlar başka katla birleştirilir; bu durumda istenenden az kat değerlendirilir ve `"WARNING: … requested folds held no presence or no absence and were merged into others …"` iletisi verilir (`sdm_core.py:5124-5134`; `sdm_spatial.py:360-382`).
- `random` tasarımında kat sayısı nadir sınıfın kayıt sayısını geçemez (`sdm_spatial.py:572-574`).
- Dama tahtası tasarımlarında kat sayısı sabittir (2 ya da 4); `Cross-validation folds:` bu tasarımlarda yalnız kaç katın koşulacağını sınırlar (`sdm_core.py:5400-5401`). `checkerboard2` ile 4'ten küçük bir değer

bazı katların hiç test edilmemesine yol açar; bu tasarımda kat sayısını 4'te bırakın.

6.4.2 Blok boyu: otomatik ya da elle

Ne işe yarar. `Size the blocks automatically from the autocorrelation range` açıkken blok kenarı, ① Data aşamasında varlık noktalarındaki öngörücü değerlerinden hesaplanan ampirik variogram menzilidir (`sdm_core.py:2433-2462`, `5096-5106`). Hesap şöyledir (`sdm_spatial.py:166-281`): en çok 1500 nokta alt örneklenir; en büyük çift uzaklığın yarısına kadar 15 uzaklık sınıfında yarı varyans $\gamma(h) = \frac{1}{2}\overline{(z_i - z_j)^2}$ hesaplanır; dış üçte bir sınıfın medyanı eşik (sill) sayılır; γ'nın eşiğin %95'ine ilk ulaştığı uzaklık o öngörücünün menzilidir; blok boyu bu menzillerin **öngörücüler üzerinden medyanıdır**. Nokta sayısı 20'nin altındaysa ya da variogram kurulamazsa çalışma alanının en büyük çift uzaklığının 1/10'u kullanılır (`method = "fallback_extent"`). Kutu kapalıyken `Block size:` değeri kullanılır.

Ne zaman, nasıl ayarlanır. Otomatik değer blockCV'deki mantığın sadeleştirilmiş bir karşılığıdır (Valavi vd. 2019). Menzil, makalede belirtilmiş bir ölçeğe (ör. bir dağ silsilesinin genişliği) ya da başka bir çalışmayla karşılaştırmaya göre sabitlenmek istenirse elle girilir.

Sonuca etkisi. Blok büyüdükçe katlar arası sızıntı azalır, skorlar düşer ve katlar arası fark büyür. Örnek koşuda variogram menzili 151,5 km, çalışma alanı genişliği 606 km'dir; 30 bloktan dört kat kurulmuş, katlardaki varlık sayıları 43/33/35/32 olmuştur (`habitus_run_config.json`).

Dikkat. Alan istenen kat sayısı kadar blok içermiyorsa blok altı kez yarıya indirilebilir; bu durumda not `"Block reduced from … km so that the study area could hold … folds …"` der (`sdm_spatial.py:454-464`, `503-507`). Bu küçültme hem otomatik hem elle girilen boyuta uygulanır. Izgara, ortalama enlemde boylam aralığı $\cos\varphi$ ile düzeltilerek derece cinsinden kurulur (`sdm_spatial.py:439-446`).

6.4.3 Toplama çarpanı (Aggregation factor)

Ne işe yarar. `checkerboard` için hücre kenarı = blok boyu × çarpan; `checkerboard2` için ince ızgara = blok boyu, kaba ızgara = blok boyu × çarpan (en az 2) (`sdm_spatial.py:651-672`). Blok boyu otomatikse variogram menzilidir.

Ne zaman, nasıl ayarlanır. ENMeval'deki `aggregation.factor` ile aynı rolü oynar (Muscarella vd. 2014). Varsayılan 2 çoğu durumda yeterlidir; büyütmek dama hücrelerini büyütür ve test noktalarını eğitim noktalarından uzaklaştırır.

Sonuca etkisi. Büyük çarpan daha kötümser (daha bağımsız) test verir; çok büyük çarpanda bazı renkler tek sınıf içerebilir ve hücre küçültülür.

Dikkat. Her kat her iki sınıfı içermezse hücre altı kez yarıya indirilir, yine olmazsa rastgele katlara düşülür (`sdm_spatial.py:685-698`).

6.4.4 Kat sayısı (Cross-validation folds)

Ne işe yarar. `spatial_block`, `environmental`, `random` ve şerit tasarımlarında oluşturulacak kat sayısıdır; tüm tasarımlarda koşulacak CV koşusu (`RUN1…RUNk`) sayısının üst sınırıdır (`sdm_core.py:5400-5407`).

Ne zaman, nasıl ayarlanır. 4–5 kat yaygın kullanımdır. Kat sayısı arttıkça her test kümesi küçülür ve kat skorlarının oynaklığı artar; küçük örneklemde 3–4 kat, geniş ve çok kayıtlı alanlarda 5 kat makuldür.

Sonuca etkisi. Süre kat sayısıyla doğrusal artar. Skorların ortalaması ve standart sapması kat sayısından etkilenir.

Dikkat. 1 seçilirse katlar yine iki parça olarak kurulur ama yalnız biri test edilir (`sdm_core.py:5018`, `5400-5401`); bu durumda kat skorlarının yayılımı hesaplanamaz ve topluluğun hold-out skoru üyelerin kendi kat

skorlarına dayanır (§6.6.9). `jackknife`'ta koşu sayısı varlık sayısına eşittir ve bu alan dikkate alınmaz (`sdm_core.py:5402-5406`).

6.4.5 Test verisi kaynağı ve dış test CSV'si

Ne işe yarar. `Random split` seçiliyken (varsayılan) testler CV katlarının tutulan parçasında yapılır. `External test points (CSV)` seçilince her `RUN` **tüm** PA verisiyle eğitilir ve dosyadaki noktalarda sınanır (`sdm_core.py:5417-5427`). Koordinat sütunları `Test CSV lon / lat:` alanlarından alınır; boşsa `longitude/long/lon/lng/decimallongitude/x` ve `latitude/lat/decimallatitude/y` adları aranır, bulunamazsa 2. ve 3. sütun kullanılır (`sdm_core.py:5165-5189`). Dosyada `presence`, `occurrence` ya da `pa` adlı bir sütun varsa 0 değerli satırlar gerçek yokluk sayılır; yoksa test yoklukları çalışma alanından yeni çekilen, eğitimden bağımsız arka plan noktalarıdır; bu da olmazsa eğitim yalancı yokluklarından tohumlu bir alt örnek alınır ve bağımsız olmadığı not edilir (`sdm_core.py:5191-5340`).

Ne zaman, nasıl ayarlanır. Bağımsız bir saha çalışmasından, başka bir dönemden ya da başka bir bölgeden gelen kayıtlar varsa kullanılır.

Sonuca etkisi. Skorlar bağımsız veriye dayanır; ancak tüm koşular aynı veriyle eğitilip aynı noktalarda sınandığı için koşular arası fark yalnız algoritmanın kendi rastlantısallığını gösterir. Program bunu rapora `"External test set: RUN1..RUNn all re-fit the same model on the SAME training data …"` notuyla yazar (`sdm_core.py:5341-5347`).

Dikkat. Raster dışına ya da NoData'ya düşen test kayıtları sayılarak bildirilir; 10'dan az kullanılabilir varlık kalırsa `"… the test statistics will be very unstable."` uyarısı çıkar (`sdm_core.py:5289-5301`). Bu modda `Training data %:` soluktur ve 80 kullanılır (`tabs/tab_models.py:1296-1301, 1578`). Dosya seçilmeden çalıştırılırsa `"External test points mode selected but no CSV provided."` hatası verilir.

6.4.6 Eğitim payı (Training data %)

Ne işe yarar. Tüm veriyle kurulan (`Full`) her model için ayrıca rastgele bir değerlendirme bölmesi yapılır: verinin bu yüzdesiyle bir model kurulur, kalanında skorlanır (`sdm_core.py:5675-5722`). Aynı algoritma ve PA seti için en az bir geçerli CV skoru varsa bu bölme skoru **CV katlarının ortalamasıyla değiştirilir** ve `Score_source = cv_mean` olur (`sdm_core.py:5818-5853`). Bölme skoru yalnız CV skoru hiç oluşmamışsa kalır (`Score_source = random_split`).

Ne zaman, nasıl ayarlanır. Normal kullanımda değiştirmeye gerek yoktur.

Sonuca etkisi. CV başarılı olduğunda raporlanan skorlara, eşiklere ve haritalara etkisi yoktur.

Dikkat. Alanın ipucu metni (`"% of occurrence data used for training; remainder used as test set"`) bu ayarın CV skorlarını belirlediği izlenimini verir; belirlemez.

6.4.7 Değişken önemi tekrarları (Variable importance reps)

Ne işe yarar. Her CV koşusunda, test katında her öngörücü karıştırılır ve AUC'deki düşüş önem sayılır. scikit-learn modellerinde `permutation_importance` (`n_repeats=3`, `scoring="roc_auc"`, sabit tohum) çağrılır; MaxEnt, GAM, ENFA, Mahalanobis, BIOCLIM, Domain ve ESM'de her öngörücü kendi tohumlu üreteciyle karıştırılıp AUC farkı alınır (`sdm_core.py:5562-5636`). Tekrarların ortalaması saklanır; 0 önemi kapatır.

Ne zaman, nasıl ayarlanır. Önem grafiği gerekmiyorsa 0 süreyi kısaltır. Varsayılan 3'tür.

Sonuca etkisi. Haritaları ve skorları değiştirmez; yalnız ④ Evaluation'daki önem grafiğini besler (Bölüm 7).

Dikkat. scikit-learn modellerinde döngü her tekrarda aynı tohumu kullandığından değerler tekrarlar arasında değişmez; bu algoritmalarda etkin tekrar sayısı `n_repeats` = 3'tür ve alan pratikte açma/kapama

işlevi görür. Tam veri (`Full`) modelleri için önem hesaplanmaz.

6.4.8 Kırpma ve sınıf dengeleme

Ne işe yarar. `Clamp predictors …` her tahminden önce öngörücüleri tüm PA setlerinin eğitim en küçük/en büyük değerlerine kırpar (`sdm_core.py:4929-4962`, `4979-4980`). `Balance classes …` destekleyen algoritmalarda varlık ve yalancı yokluk sınıflarını boyutlarıyla ters orantılı ağırlıklar; altındaki satır mevcut PA:varlık oranını gösterir (örnek koşuda `1000 : 143 (7.0 : 1)`) (`tabs/tab_models.py:1316-1335`). Algoritma bazındaki ayrıntı Bölüm 5'tedir.

Sonuca etkisi. Kırpma CV içinde neredeyse etkisizdir, gelecek izdüşümlerinde dışdeğerlemeyi sınırlar. Dengeleme olasılıkların ölçeğini değiştirir; eşikler yeniden türetilir, AUC sıralaması çoğunlukla değişmez (ipucu metni).

6.4.9 Mekânsal sıralama yanlılığı (SSB) raporu

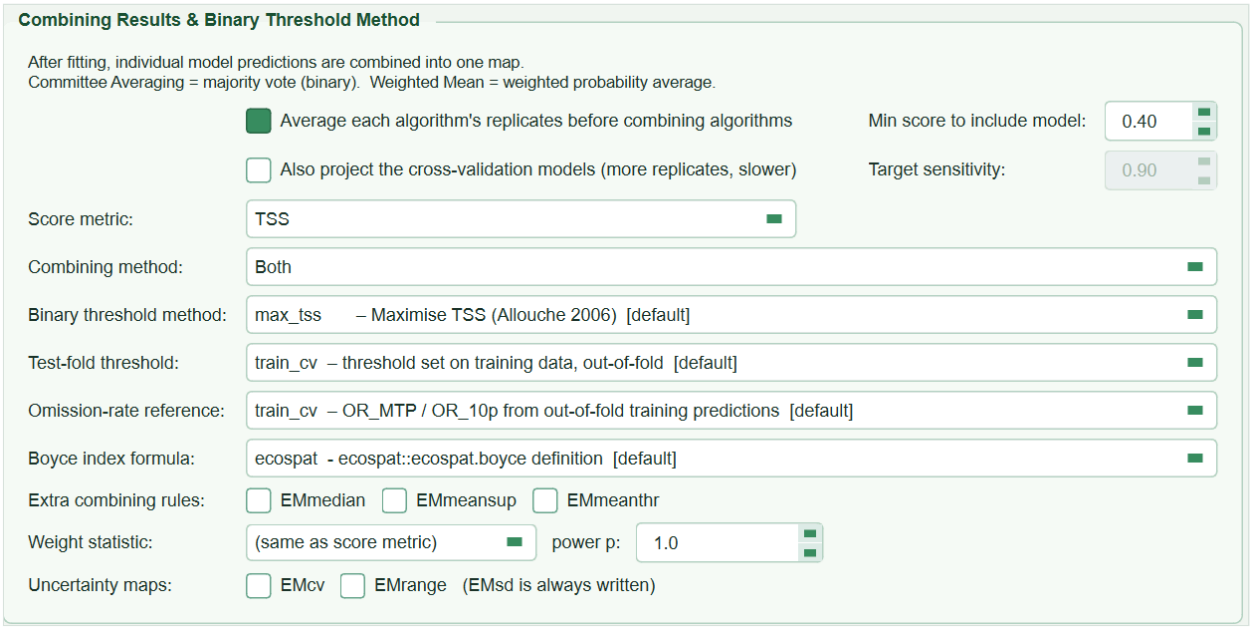
Ayar değildir, her çalıştırmada kendiliğinden hesaplanır. Kullanılan kat tasarımı için (Hijmans 2012):

$$\mathrm{SSB} = \frac{\overline{d}(\text{test varlık} \rightarrow \text{en yakın eğitim varlığı})}{\overline{d}(\text{test yokluk} \rightarrow \text{en yakın eğitim varlığı})}$$

Uzaklıklar büyük daire km'sidir, katlar üzerinden havuzlanır ve kat başına da verilir (`sdm_spatial.py:808-881`). Yorum eşikleri koddadır: 0,7'nin altı "spatial sorting bias … scores optimistic", 0,7–1,3 "little or no spatial sorting bias", 1,3'ün üstü dışdeğerlemeci (kötümser) bölme (`sdm_spatial.py:866-875`). Sonuç günlüğe `CV PA1: SSB = …` satırıyla, çalıştırma kaydına `per_pa_set.*.ssb` altında ve özet satıra `[SSB=…]` olarak yazılır (`sdm_core.py:5142-5153`, `5942-5944`).

> **Not:** Örnek koşuda mekânsal bloklara karşın SSB = 0,36 bulunmuştur (test varlıkları en yakın eğitim varlığına ortalama 45,5 km, test yoklukları 127,9 km). Yalancı yokluklar alanın tamamına yayıldığı, varlıklar ise dar bir kuşakta toplandığı için bu oran bloklamadan bağımsız olarak düşük çıkabilir; SSB'yi tasarımlar arasında karşılaştırmak için kullanın, tek başına bir geçme/kalma ölçütü olarak değil.

6.5 Değerlendirme ayarları: eşik, omisyon referansı, Boyce



Şekil 6.4. Combining Results && Binary Threshold Method kutusu varsayılan değerlerle: iki katmanlı birleştirme açık, TSS skor ölçütü ve 0.40 en düşük skor, max_tss eşik kuralı, train_cv eşik kaynağı ve omisyon referansı, ecospat Boyce formülü.

Şekil 6.4. Combining Results && Binary Threshold Method kutusu varsayılan değerlerle: iki katmanlı birleştirme açık, TSS skor ölçütü ve 0.40 en düşük skor, max_tss eşik kuralı, train_cv eşik kaynağı ve omisyon referansı, ecospat Boyce formülü.

Bu kutu iki iş görür: ikili haritanın ve eşiğe bağlı skorların (TSS, Kappa, MCC, duyarlılık, özgüllük, Jaccard, Sørensen, FPB) eşiğini, ve topluluk ayarlarını (§6.6). Önce eşik ve skor ayarları:

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Binary threshold method:`	Olasılığı var/yok'a çeviren kural	`max_tss`	11 kural (§6.5.1)	`tabs/tab_models.py:435-455`
`Target sensitivity:`	`sensitivity` kuralının hedefi	0.90	0.50–1.00	`tabs/tab_models.py:456-462`
`Test-fold threshold:`	Eşiğin seçildiği veri	`train_cv`	`train_cv` / `train` / `test`	`tabs/tab_models.py:465-490`
`Omission-rate reference:`	OR_MTP ve OR_10p eşiklerinin kaynağı	`train_cv`	`train_cv` / `train`	`tabs/tab_models.py:491-503`
`Boyce index formula:`	Sürekli Boyce indeksinin tanımı	`ecospat`	`ecospat` / `legacy`	`tabs/tab_models.py:504-514`

6.5.1 İkili eşik kuralı (Binary threshold method)

Ne işe yarar. Her model için bir eşik seçilir; olasılığı bu eşiğe eşit ya da büyük hücreler 1 olur (`sdm_core.py:7907-7915`). Kuralların hepsi `find_optimal_threshold` içindedir (`sdm_core.py:795-948`). Izgara taramalı kurallar 0,01–0,99 arasında 200 eşit aralıklı eşiği dener.

Kural	Seçilen eşik	Kaynak	Kod
`max_tss`	TSS = duyarlılık + özgüllük − 1'i en büyük yapan eşik	Allouche vd. 2006	`sdm_core.py:891-897`
`max_kappa`	Cohen kappa'sını en büyük yapan eşik	Fielding ve Bell 1997	`sdm_core.py:899-915`
`sens_spec_eq`	|duyarlılık − özgüllük| en küçük	Liu vd. 2005	`sdm_core.py:917-931`
`p10`	Varlık tahminlerinin 10. yüzdeliği	Pearson vd. 2007	`sdm_core.py:858-864`
`min_roi`	ROC eğrisinde (0,1) köşesine en yakın nokta: $\min \sqrt{\mathrm{FPR}^2 + (1-\mathrm{TPR})^2}$	—	`sdm_core.py:933-945`
`mtp`	En düşük varlık tahmini (hiç varlık dışarıda kalmaz; `lpt` ile aynı)	Pearson vd. 2007	`sdm_core.py:734-735`
`p5`	Varlık tahminlerinin 5. yüzdeliği	—	`sdm_core.py:736-738`
`sensitivity`	Varlıkların en az hedef payını tutan en yüksek eşik: sıralı varlık tahminlerinde $\lfloor (1-s)\,n \rfloor$. değer	Allouche vd. 2006	`sdm_core.py:866-876`
`max_jaccard`	$\mathrm{TP}/(\mathrm{TP}+\mathrm{FP}+\mathrm{FN})$ en büyük	—	`sdm_core.py:878-885`
`max_sorensen`	$2\mathrm{TP}/(2\mathrm{TP}+\mathrm{FP}+\mathrm{FN})$ en büyük	—	`sdm_core.py:878-885`
`max_fpb`	FPB = 2 × Jaccard en büyük (flexsdm'in varlık-arka plan F ölçüsü)	Li ve Guo 2013	`sdm_core.py:701, 878-885`

Ne zaman, nasıl ayarlanır. `max_tss` yalancı yokluklu verilerde yaygın ve yaygınlıktan (prevalence) görece bağımsızdır. Yanlış negatifin pahalı olduğu korunan alan çalışmalarında `p10`, `p5`, `mtp` ya da `sensitivity` (ör. 0,95) daha geniş alan verir. Omisyon ve komisyon hatası eşit önemliyse `sens_spec_eq` seçilir. Kappa ve Jaccard ailesi yaygınlıktan etkilenir; yalancı yokluk sayısı değiştikçe eşik kayar.

Sonuca etkisi. İkili haritalardaki alan, alan değişimi hesapları (⑥ Range Change) ve eşiğe bağlı tüm skorlar bu kurala bağlıdır. AUC, Boyce, Brier ve kalibrasyon eğimi eşikten bağımsızdır. Örnek koşuda `max_tss` ile seçilen eşikler 0,06–0,21 arasındadır (GLM_PA1 0,1294, RF_PA2 0,2058; oturum günlüğü): olasılık ölçeği düşük olduğundan 0,5 gibi sabit bir eşik bu verilerde çok dar bir harita verirdi.

Dikkat. Haritada kullanılan eşik, o modelin CV katlarında seçilen eşiklerin ortalamasıdır (`sdm_core.py:5852-5853`); tam veri modelinin kendi verisinde yeniden seçilmez. Eşik hesaplanamazsa 0,5 kullanılır ve `Threshold_source` sütununa `FAILED, 0.5 used (…)` yazılır (`sdm_core.py:1221-1227`). `Target sensitivity:` kutusu yalnız `sensitivity` seçiliyken etkindir; hedef anahtarın içinde taşınır (`sensitivity:0.90`) ve çalıştırma kaydına öyle yazılır (`tabs/tab_models.py:1504-1507`).

6.5.2 Eşiğin seçildiği veri (Test-fold threshold)

Ne işe yarar. Test katındaki eşiğe bağlı skorlar için eşiğin hangi tahminlerde seçileceğini belirler (`evaluate_all`, `sdm_core.py:1122-1257`):

- `train_cv` (varsayılan): eğitim satırları üç iç kata bölünür (`OOF_INNER_FOLDS = 3`, katlar kullanıcının CV tasarımıyla kurulur); her satır kendisini görmemiş bir modelle tahmin edilir; eşik bu kat dışı (out-of-fold) tahminlerde seçilip test katına uygulanır (`sdm_core.py:4876-4919`, `4296-4368`).
- `train`: eşik, modelin kendi eğitim satırlarındaki tahminlerde seçilir.
- `test`: eşik doğrudan test katında, o katın skorunu en büyük yapacak biçimde seçilir (kâhin eşik).

Ne zaman, nasıl ayarlanır. `train_cv` önerilen ve dürüst seçenektir. `train` ucuzdur ama eğitim verisini ezberleyen algoritmalarda (Domain, RF, GBM, XGBoost) eşik anlamsızlaşır. `test` yalnız eski sonuçlarla karşılaştırma içindir.

Sonuca etkisi. `test` seçeneği TSS'yi tipik olarak 0,1–0,2 yüksek verir (ipucu metni, `tabs/tab_models.py:480-482`). `train_cv` her CV kurulumuna üç iç kurulum ekler; süre kabaca iki katına çıkar. Haritadaki eşik CV katlarındaki eşiklerin ortalaması olduğundan (§6.5.1) bu seçim haritaları da etkiler.

Dikkat. `test` seçilince kutunun altında turuncu uyarı belirir: `"Warning: 'test' picks the threshold on the very data it is scored on …"` (`tabs/tab_models.py:1261-1272`). İç kurulum başarısız olursa kaynak `train (inner CV unavailable)` olarak etiketlenir (`sdm_core.py:1201-1207`). ESM iç kat tahmini üretmez (`sdm_core.py:4889-4890`).

6.5.3 Omisyon oranı referansı (Omission-rate reference)

Ne işe yarar. OR_MTP ve OR_10p, test varlıklarının sırasıyla en düşük ve 10. yüzdelik **eğitim** varlık tahmininin altında kalan payıdır (`sdm_core.py:1062-1115`). `train_cv` bu referans değerleri iç kat dışı tahminlerden, `train` modelin kendi eğitim tahminlerinden alır.

Ne zaman, nasıl ayarlanır. Varsayılan `train_cv` önerilir; ENMeval'in omisyon tanımıyla aynı mantıktadır (Muscarella vd. 2014), referansın ezberden etkilenmemesi farkıyla.

Sonuca etkisi. Yalnız OR_MTP ve OR_10p sütunlarını ve Tuning'deki ENMeval seçim kuralını (§6.7) etkiler; haritaları etkilemez.

Dikkat. `train` seçilince uyarı: `"Warning: with in-sample references OR_10p / OR_MTP are unreliable for over-fitting algorithms …"`. Eşik değeri değerlendirme kümesinin en düşük tahminine eşit ya da altındaysa omisyon yapı gereği 0 olur; HABITUS bunu `OR_10p_degenerate` sütunuyla işaretler ve katların en az yarısında olursa algoritma için not yazar (`sdm_core.py:1100-1114`, `5787-5816`).

6.5.4 Boyce indeksi formülü (Boyce index formula)

Ne işe yarar. `ecospat`, `ecospat::ecospat.boyce` işlevinin satır satır aktarımıdır (`sdm_core.py:475-543`): değerlendirilen tüm satırların tahminleri beklenen dağılımı (E), varlıkların tahminleri gözlenen dağılımı (P) verir; genişliği tahmin aralığının 1/10'u olan pencere 101 konumda kaydırılır, her pencerede $F = P_i/E_i$ hesaplanır, ardışık yinelenen F değerleri atılır ve indeks pencere alt sınırları ile F arasındaki Spearman korelasyonudur (Hirzel vd. 2006; Boyce vd. 2002). `legacy`, [0, 1] ekseninde sabit 0,1 genişlikli pencere ve 101 merkezle hesaplanan önceki HABITUS tanımıdır (`sdm_core.py:612-664`).

Ne zaman, nasıl ayarlanır. R'daki ecospat ile karşılaştırılabilir sonuç için `ecospat` (varsayılan). `legacy` yalnız önceki bir sonucu yeniden üretmek içindir.

Sonuca etkisi. Boyce skorlarını, Boyce'u skor ölçütü yapan topluluk seçimini ve ağırlıklarını değiştirir. Kod açıklaması aynı BIOCLIM tahminlerinde `legacy` için 0,007, `ecospat` için 0,334 verildiğini kaydeder (`sdm_core.py:560-562`).

Dikkat. Değerlendirme kümesinde varlık yoksa ya da 5'ten az satır varsa Boyce `NaN` olur (`sdm_core.py:605-609`).

6.6 Topluluk ayarları (Combining Results)

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Average each algorithm's replicates before combining algorithms`	İki katmanlı topluluk	açık	açık / kapalı	`tabs/tab_models.py:516-530`
`Also project the cross-validation models (more replicates, slower)`	CV modellerini de izdüşürür	kapalı	açık / kapalı	`tabs/tab_models.py:532-540`
`Combining method:`	Yazılan temel topluluk kuralları	`Both`	`Weighted Mean` / `Committee Averaging` / `Both`	`tabs/tab_models.py:403-417`
`Score metric:`	Üye süzme ölçütü	`TSS`	`TSS` / `ROC` / `Boyce`	`tabs/tab_models.py:418`
`Min score to include model:`	En düşük skor	ölçüte bağlı: TSS 0.40, ROC 0.70, Boyce 0.50	0–1	`tabs/tab_models.py:419-433`, `sdm_core.py:6093`
`Extra combining rules:`	Ek kurallar	hepsi kapalı	`EMmedian`, `EMmeansup`, `EMmeanthr`	`tabs/tab_models.py:543-558`
`Weight statistic:`	Ağırlık istatistiği	`(same as score metric)`	+ `AUC`, `TSS`, `Boyce`	`tabs/tab_models.py:559-566`
`power p:`	Ağırlık kuvveti	1.0	0.5–10.0	`tabs/tab_models.py:567-575`
`Uncertainty maps:`	Ek belirsizlik haritaları	kapalı (EMsd her zaman)	`EMcv`, `EMrange`	`tabs/tab_models.py:576-583`

6.6.1 İki katmanlı topluluk

Ne işe yarar. Açıkken (varsayılan) önce aynı algoritmanın yinelemeleri (her PA seti için bir tam veri modeli; §6.6.2 açıksa CV modelleri de) ortalanır; algoritma başına ortalama, yinelemeler arası standart sapma ve yinelemelerin oy payı hesaplanır (`EnsembleModeler.predict_by_algorithm`, `sdm_core.py:6234-6331`). Topluluk bu algoritma ortalamalarından kurulur (`sdm_core.py:6359-6376`). Kapalıyken süzgeçten geçen her model tek düz bir ağırlıklı ortalamaya girer (`sdm_core.py:6377-6407`).

Ne zaman, nasıl ayarlanır. Açık bırakın: aksi hâlde yinelemesi çok olan bir algoritma topluluğa daha fazla oy verir.

Sonuca etkisi. Ayrıca her algoritma için `<algo>_mean_prob`, `<algo>_mean_bin` ve birden çok yineleme varsa `<algo>_sd` haritaları yazılır (`sdm_core.py:7964-8021`). Ağırlıklar algoritma düzeyinde hesaplanır; örnek koşuda TSS ağırlıkları GLM 0,271, MaxEnt 0,265, GAM 0,241, RF 0,223'tür.

Dikkat. Kapalıyken `EMsd`, ek belirsizlik haritası seçilmedikçe yazılmaz (`sdm_core.py:6416-6419`); arayüzdeki `(EMsd is always written)` ifadesi yalnız iki katmanlı mod için geçerlidir.

6.6.2 CV modellerini de izdüşürme

Ne işe yarar. Varsayılan olarak yalnız tam veri modelleri izdüşürülür (algoritma başına PA seti kadar yineleme). Kutu açıkken CV katlarının modelleri de izdüşürülür ve her biri ayrı dosya (`<algo>_PA1_RUN1_prob.tif` …) olarak yazılır (`sdm_core.py:7850-7916`). Seçim ⑤ Future Scenarios'a da aktarılır (`tabs/tab_models.py:1538-1546`).

Ne zaman, nasıl ayarlanır. Algoritma içi kararsızlık (yinelemeler arası SD) için daha fazla yineleme istendiğinde. PA seti sayısı 1 iken algoritma başına tek yineleme kalacağı için `<algo>_sd` haritası ancak bu kutuyla üretilebilir.

Sonuca etkisi. İzdüşüm süresi ve disk kullanımı yaklaşık (kat + 1) katına çıkar. Topluluk ortalaması CV modellerini de içerir.

Dikkat. Topluluk yalnız izdüşürülen modellerden kurulur (`sdm_core.py:6343-6355`, `8028-8035`). Kutu kapalıyken en düşük skoru yalnız bazı CV modelleri geçip hiçbir tam veri modeli geçemezse topluluk yazılmaz ve program `"No full-data model reached the minimum … Lower the minimum score, or tick 'Also project the cross-validation models' …"` iletisini verir (`sdm_core.py:8040-8054`). Kutu açıkken her CV modeli kendi kat skoruyla ayrıca süzülür; zayıf bir katın modeli dışarıda kalabilir.

6.6.3 Birleştirme yöntemi ve ek kurallar

Ne işe yarar. `Combining method:` hangi temel kuralların hesaplanıp yazılacağını belirler: `Weighted Mean` → `EMwmean`, `Committee Averaging` → `EMca`, `Both` → ikisi (`tabs/tab_models.py:1670-1694`; `sdm_core.py:6173-6183`, `6409-6412`). Seçilmeyen kural hesaplanmaz, yazılmaz, skorlanmaz. Tanımlar (iki katmanlı modda, $\bar{p}_a$ algoritma ortalaması, w_a ağırlık, v_a algoritmanın yinelemelerinden kendi eşiğinde "var" diyenlerin payı):

$$\mathrm{EMwmean} = \sum_a w_a \, \bar{p}_a, \qquad \mathrm{EMca} = \frac{1}{A} \sum_a v_a$$

Ek kurallar (`sdm_core.py:6016-6040`): `EMmedian` hücre başına üye olasılıklarının medyanı; `EMmeansup` ağırlık istatistiği üyelerin ortalamasına eşit ya da üstünde olan "üstün" üyelerin basit ortalaması (flexsdm `meansup`); `EMmeanthr` hücrede kendi eşiğine ulaşan üyelerin olasılık ortalaması, hiçbiri ulaşmıyorsa 0 (flexsdm `meanthr`; Velazco vd. 2022).

Ne zaman, nasıl ayarlanır. Ağırlıklı ortalama ve komite ortalaması, Marmion vd. (2009) karşılaştırmasında iyi sonuç veren uzlaşı yöntemleridir. Medyan uç bir üyeye karşı dayanıklıdır; `meanthr` "en az bir üye var diyor" türünden geniş bir harita verir.

Sonuca etkisi. Her kural ayrı `_prob` ve `_bin` haritası üretir, ayrı hold-out skoru alır (§6.6.6). `EMca` bir oy payıdır, olasılık değildir.

Dikkat. Komite ortalaması algoritmalar arasında ağırlıksızdır; ağırlık istatistiği yalnız `EMwmean` ve `EMmeansup`'ı etkiler.

6.6.4 Skor ölçütü ve en düşük skor

Ne işe yarar. Bir model, `Score metric:` değeri `Min score to include model:` değerine eşit ya da büyükse topluluğa girer (`EnsembleModeler._filter_models`, `sdm_core.py:6185-6196`). Tam veri modellerinin skoru CV katlarının ortalamasıdır (§6.4.6). En düşük skor varsayılanı ölçüte göre değişir: ROC 0,70, TSS 0,40, Boyce 0,50 (`sdm_core.py:6086-6098`); kullanıcı bir değer yazana kadar ölçütü izler, yazdıktan sonra değiştirilmez (`tabs/tab_models.py:1696-1710`).

Ne zaman, nasıl ayarlanır. Mekânsal bloklu CV ile dürüst TSS'ler genellikle 0,3–0,6 aralığındadır; TSS için 0,40 bu nedenle seçilmiştir (kod açıklaması, `sdm_core.py:6086-6092`). Zayıf modelleri dışarıda bırakmak isteniyorsa yükseltilir; topluluğun boş kalmaması için en iyi modelin skoruna bakılarak ayarlanır.

Sonuca etkisi. Topluluk üye sayısı ve dolayısıyla harita değişir. Örnek koşuda sekiz tam veri modelinin sekizi de TSS ≥ 0,40 koşulunu sağlamıştır (`Models included in combined map: 8/8`).

Dikkat. Hiçbir model eşiği geçmezse `EM*` haritaları yazılmaz, algoritma haritaları yazılır; durum metni `WARNING - NO COMBINED MAP:` ile başlar ve ileti en iyi modelin skorunu söyler (`sdm_core.py:6198-6232`; `tabs/tab_models.py:1831-1833`).

6.6.5 Ağırlık istatistiği, ağırlık kuvveti ve belirsizlik haritaları

Ne işe yarar. Üye ağırlığı

$$w_i = \frac{\max(s_i, 0)^{\,p}}{\sum_j \max(s_j, 0)^{\,p}}$$

ile hesaplanır; s_i `Weight statistic:` ile seçilen istatistik (varsayılan: skor ölçütü), p `power p:` değeridir (`sdm_core.py:5988-6005`). Sonlu olmayan skorlar eşit pay alır; tüm ağırlıklar sıfırsa üyeler eşit ağırlıklanır. Süzme yine skor ölçütüyle yapılır. `EMsd` üyeler arası standart sapmadır (ddof = 1); `EMcv` = SD / ortalama (ortalama ≈ 0 olan hücrede 0); `EMrange` = en büyük − en küçük (`sdm_core.py:6068-6083`). Belirsizlik yüzeyleri normalleştirilmeden yazılır ve hiçbir zaman eşiklenmez (`sdm_core.py:8057-8068`).

Ne zaman, nasıl ayarlanır. $p = 1$ skorla orantılı ağırlıktır (sdm paketindeki `power` ile aynı anlamda); en iyi üyeleri öne çıkarmak için 2–3 kullanılabilir. Ağırlık istatistiğini skor ölçütünden farklı seçmek, ör. TSS ile süzüp Boyce ile ağırlıklamak mümkündür.

Sonuca etkisi. p büyüdükçe `EMwmean` en iyi algoritmaya yaklaşır. Belirsizlik haritaları topluluk haritasını değiştirmez.

Dikkat. Belirsizlik yüzeyleri en az iki üye gerektirir; tek algoritmalı bir çalıştırmada (iki katmanlı modda) `EMsd` oluşmaz (`sdm_core.py:6073-6074`).

6.6.6 Topluluğun ikili eşiği

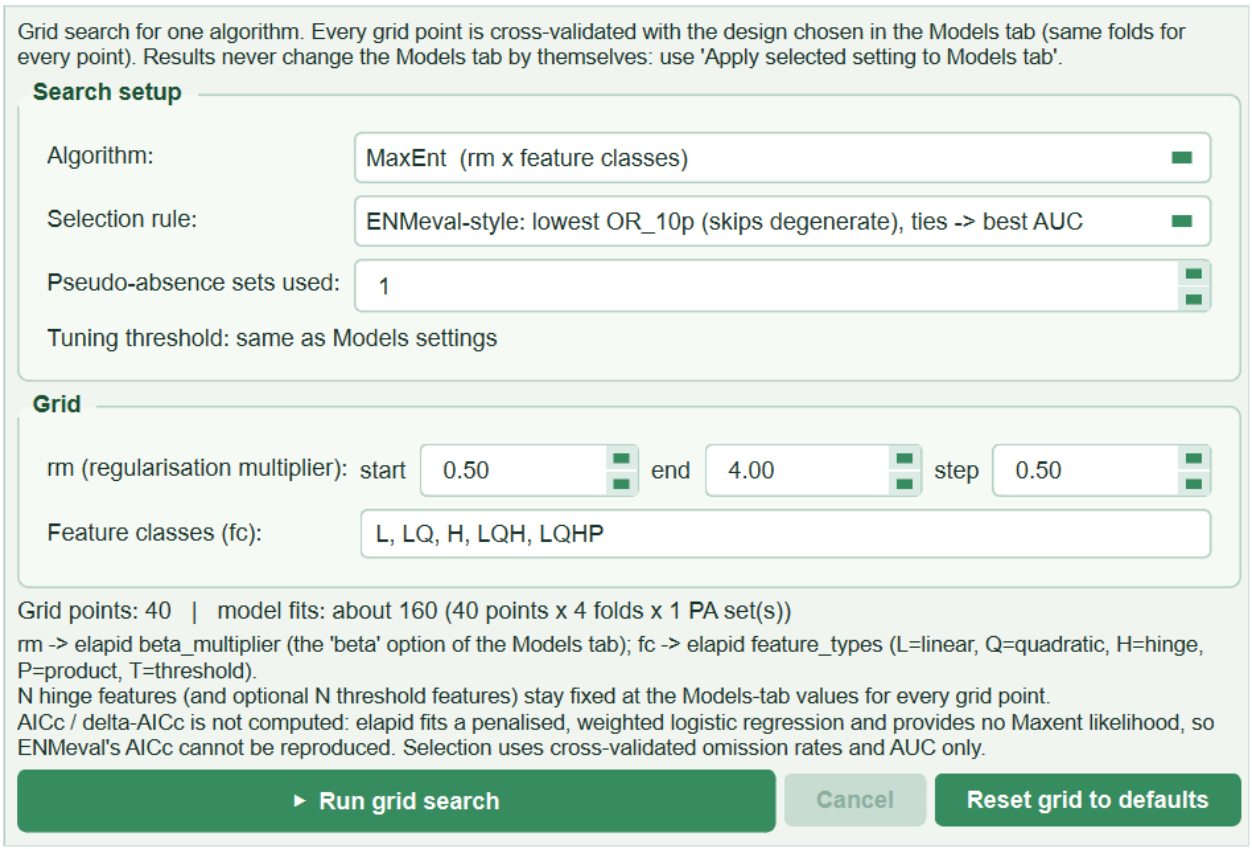
Program topluluk haritalarını şu eşiklerle ikiliye çevirir (`EnsembleModeler.binary_threshold`, `sdm_core.py:6423-6461`): `EMca` için 0,5 (çoğunluk oyu); `EMwmean` için üyelerin eşiklerinin, harita ağırlıklarıyla ağırlıklı ortalaması; `EMmedian` için üye eşiklerinin medyanı; `EMmeansup` için üstün üyelerin eşik ortalaması; `EMmeanthr` için en küçük üye eşiği (`sdm_core.py:6043-6065`). Örnek koşuda `EMwmean` eşiği 0,1166, `EMca` eşiği 0,5'tir. Topluluk haritaları kendi aralığına gerilmez; yalnız [0, 1]'e kırpılır (`sdm_core.py:8074-8081`).

6.6.7 Topluluğun hold-out skoru ve diğer katlardan üye seçimi

Topluluk haritası, tüm varlıkları görmüş modellerden kurulduğu için kendi haritasında skorlanmaz. Bunun yerine her PA seti × kat grubu için o katı görmemiş CV modelleri (algoritma başına bir tane) topluluğun kuralıyla birleştirilir ve yalnız o katın tutulan satırlarında skorlanır; raporlanan değer kat gruplarının ortalamasıdır (`EnsembleModeler.evaluate_cv`, `sdm_core.py:6516-6696`). Üyenin topluluğa girip girmediği ve ağırlığı, aynı algoritmanın **diğer** katlardaki skorlarının ortalamasından hesaplanır (bir katı dışarıda bırakma; aynı kat numarası diğer PA setlerinde de dışarıda kalır; `sdm_core.py:6463-6496`). Topluluğun eşiği, üyelerin iç kat dışı eğitim tahminlerinden kurulan eğitim yüzeyinde seçilir (`sdm_core.py:6619-6657`). Örnek koşuda sekiz kat grubunun sekizi kullanılmış; `EMwmean` için ROC 0,902, TSS 0,659, Boyce 0,726; `EMca` için ROC 0,878, TSS 0,660, Boyce 0,725 bulunmuştur. Bu satırlar `PA_set = Ensemble`, `CV_run = CV_holdout`, `Score_source = cv_holdout` ile `evaluation_scores.csv`'ye eklenir (`sdm_core.py:6698-6729`).

> **Not:** Hold-out skoru her katta düz (tek katmanlı) bir birleşimle hesaplanır, çünkü her grupta her algoritmadan tek model vardır. Tek katlı bir tasarımda "diğer kat" bulunmadığından üye kendi kat skoruyla ağırlıklanır ve bu durum `cv_note` metninde sayılır (`sdm_core.py:6511-6513`).

6.7 Ayar araması (Tuning (optional))



Şekil 6.5. Tuning (optional) alt sekmesinin ayar paneli (sağdaki sonuç tablosu ve grafik arama yapılmadığı için boş): MaxEnt için rm 0.50–4.00 adım 0.50 ve L, LQ, H, LQH, LQHP öznitelik sınıfları; 40 ızgara noktası, yaklaşık 160 model kurulumu.

Şekil 6.5. Tuning (optional) alt sekmesinin ayar paneli (sağdaki sonuç tablosu ve grafik arama yapılmadığı için boş): MaxEnt için rm 0.50–4.00 adım 0.50 ve L, LQ, H, LQH, LQHP öznitelik sınıfları; 40 ızgara noktası, yaklaşık 160 model kurulumu.

Bu alt sekme tek bir algoritmanın ayarlarını bir ızgara üzerinde arar. Her ızgara noktası, `Algorithm Settings`'teki CV tasarımı, eşik kuralı ve kaynağı, omisyon referansı, Boyce formülü ve kırpma ayarıyla, tüm noktalarda **aynı katlarla** sınanır (`tabs/tab_models.py:1274-1294`; `sdm_tuning.py:394-494`). Her nokta için yalnız CV koşulur (`run(cv_only=True)`); tam veri modeli ve değişken önemi hesaplanmaz.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Algorithm:`	Aranacak algoritma	`MaxEnt (rm x feature classes)`	MaxEnt, GBM, RF, SVM, ANN, XGBoost, LightGBM, GLMNET, CART, GAM (kurulu olanlar)	`tabs/tab_tuning.py:91-96`
`Selection rule:`	En iyi noktayı seçme kuralı	ENMeval-style	3 kural	`tabs/tab_tuning.py:97-110`, `sdm_tuning.py:84-88`
`Pseudo-absence sets used:`	Skorlamada kullanılan ilk n PA seti	1	1–20	`tabs/tab_tuning.py:111-114`
`rm (regularisation multiplier):`	MaxEnt düzenleme çarpanı dizisi	start 0.50, end 4.00, step 0.50	0.05–50	`tabs/tab_tuning.py:211-226`, `sdm_tuning.py:506`
`Feature classes (fc):`	MaxEnt öznitelik sınıfı setleri	`L, LQ, H, LQH, LQHP`	L, Q, H, P, T harfleri	`tabs/tab_tuning.py:227-233`, `sdm_tuning.py:505`
Diğer algoritmaların ızgara satırları	Virgülle değer listesi ya da `başla:bitir[:adım]`	algoritmaya göre (aşağıda)	—	`tabs/tab_tuning.py:234-242`
`▶ Run grid search`, `Cancel`, `Reset grid to defaults`	Başlat; noktalar arasında durdur; varsayılana dön	—	—	`tabs/tab_tuning.py:137-148`
`Apply selected setting to Models tab`, `Save table as CSV…`	Seçilen noktayı algoritma seçeneklerine yazar; tabloyu kaydeder	—	—	`tabs/tab_tuning.py:170-178`, `480-499`

Varsayılan ızgaralar (`sdm_tuning.py:626-643`, `776-793`, `837-843`): GBM `n_estimators` 20, 50, 100, 200, 500 × `learning_rate` 0.01, 0.1, 0.3 × `min_samples_leaf` 1–10, 20, 30 (flexsdm `tune_gbm`, 180 nokta); RF `max_features` 2'den öngörücü sayısına kadar × `n_estimators` 200–1200 (flexsdm `tune_raf`); SVM `C` 1, 2, 4, 8, 16 × `gamma` 0.001, 0.01, 0.1, 0.2 (kernlab `sigma` = sklearn `gamma`); ANN `hidden_layer_sizes` 2, 4, 6, 8 × `alpha` = 2 × nnet `decay`; XGBoost ve LightGBM için 3 × 3 × 3; GLMNET `C` 0.01–10 × `l1_ratio` 0–1; CART `max_depth` × `min_samples_leaf` × `ccp_alpha`; GAM `n_splines` 5–20 × `lam` 0.1–10.

6.7.1 Seçim kuralı (Selection rule)

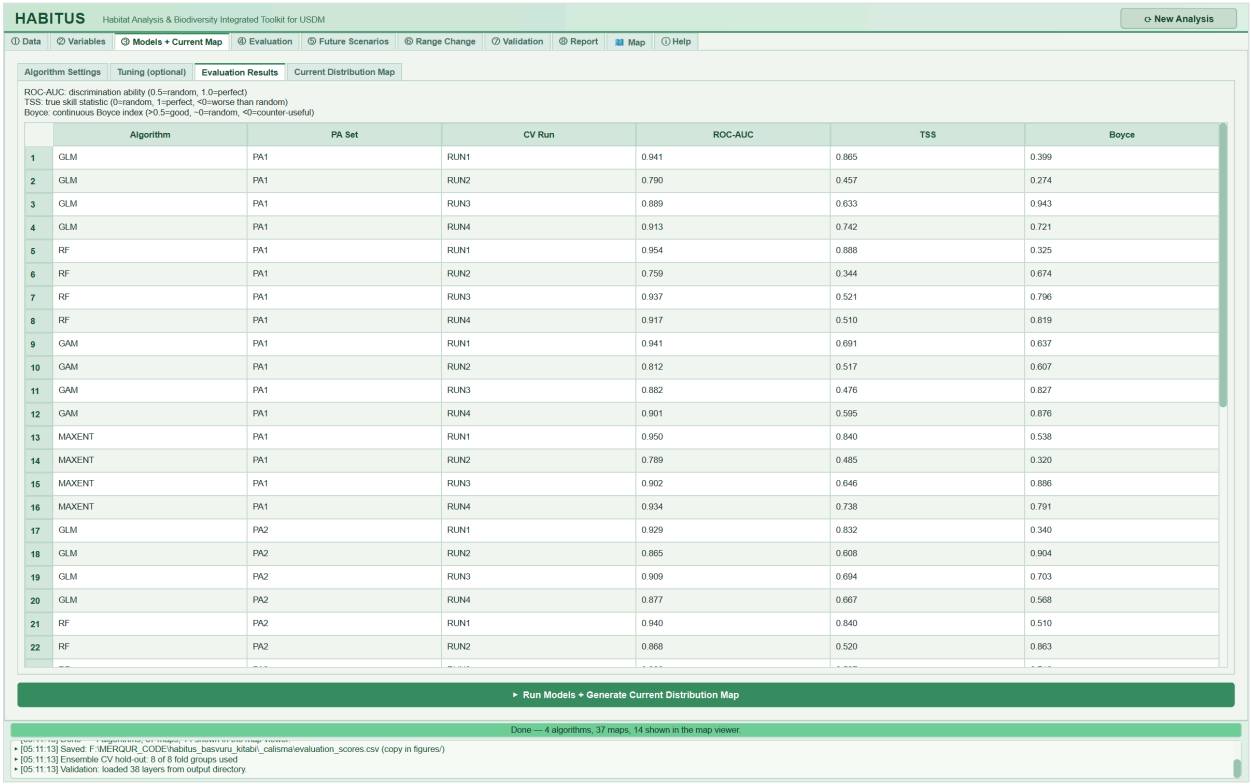
Ne işe yarar. `enmeval`: ortalama OR_10p'si en düşük nokta, eşitlikte en yüksek ortalama test AUC'si (Muscarella vd. 2014; Kass vd. 2021). `enmeval_guard`: önce en iyi AUC'nin bir kat-SD altından daha kötü noktaları atar, sonra aynı kural. `max_auc`: en yüksek ortalama test AUC'si. OR_10p'si yalnız alttaki eşit tahminler yüzünden 0 olan ("degenerate") noktalar ENMeval kurallarında seçilmez (`sdm_tuning.py:238-298`). Kural arama bittikten sonra yeniden kurulum yapmadan değiştirilebilir (`tabs/tab_tuning.py:415-418`).

Ne zaman, nasıl ayarlanır. MaxEnt için ENMeval-style yaygın kullanımla uyumludur; AUC'si belirgin düşük noktaların seçilmesini önlemek için `enmeval_guard`.

Sonuca etkisi. Seçilen nokta `Apply selected setting to Models tab` ile algoritma seçeneklerine yazılır; ana çalıştırma bunu kullanır. Arama sonucu, ızgara, kural ve seçilen noktanın çalıştırmada gerçekten kullanılıp kullanılmadığı `habitus_run_config.json`'a `tuning_<algo>` bölümü olarak yazılır (`tabs/tab_models.py:2195-2208`).

Dikkat. AICc hesaplanmaz; program bunu `"AICc / delta-AICc is not computed: …"` notuyla söyler (`sdm_tuning.py:62-66`). Seçim, raporlanan skorları üreten aynı katlarda yapıldığından ayarlanmış algoritmanın skorları iyimserdir; iç içe (nested) doğrulama yoktur (`sdm_tuning.py:68-78`). 60 noktayı ya da 600 kurulumu aşan aramalar onay ister (`tabs/tab_tuning.py:36-38`, `318-327`). Sonuç tablosu otomatik olarak `<çıktı>/tuning/tuning_<algo>.csv` dosyasına yazılır (`tabs/tab_tuning.py:397-413`).

6.8 Değerlendirme sonuçları (Evaluation Results)



	Algorithm	PA Set	CV Run	ROC-AUC	TSS	Boyce
1	GLM	PA1	RUN1	0.941	0.865	0.399
2	GLM	PA1	RUN2	0.790	0.457	0.274
3	GLM	PA1	RUN3	0.889	0.633	0.943
4	GLM	PA1	RUN4	0.913	0.742	0.721
5	RF	PA1	RUN1	0.954	0.888	0.325
6	RF	PA1	RUN2	0.759	0.344	0.674
7	RF	PA1	RUN3	0.937	0.521	0.796
8	RF	PA1	RUN4	0.917	0.510	0.819
9	GAM	PA1	RUN1	0.941	0.691	0.637
10	GAM	PA1	RUN2	0.812	0.517	0.607
11	GAM	PA1	RUN3	0.882	0.476	0.827
12	GAM	PA1	RUN4	0.901	0.595	0.876
13	MAXENT	PA1	RUN1	0.950	0.840	0.538
14	MAXENT	PA1	RUN2	0.789	0.485	0.320
15	MAXENT	PA1	RUN3	0.902	0.646	0.886
16	MAXENT	PA1	RUN4	0.934	0.738	0.791
17	GLM	PA2	RUN1	0.929	0.832	0.340
18	GLM	PA2	RUN2	0.865	0.608	0.904
19	GLM	PA2	RUN3	0.909	0.694	0.703
20	GLM	PA2	RUN4	0.877	0.667	0.568
21	RF	PA2	RUN1	0.940	0.840	0.510
22	RF	PA2	RUN2	0.868	0.520	0.863

Şekil 6.6. Evaluation Results alt sekmesi örnek koşudan sonra: her algoritma, PA seti ve CV koşusu için ROC-AUC, TSS ve Boyce değerleri.

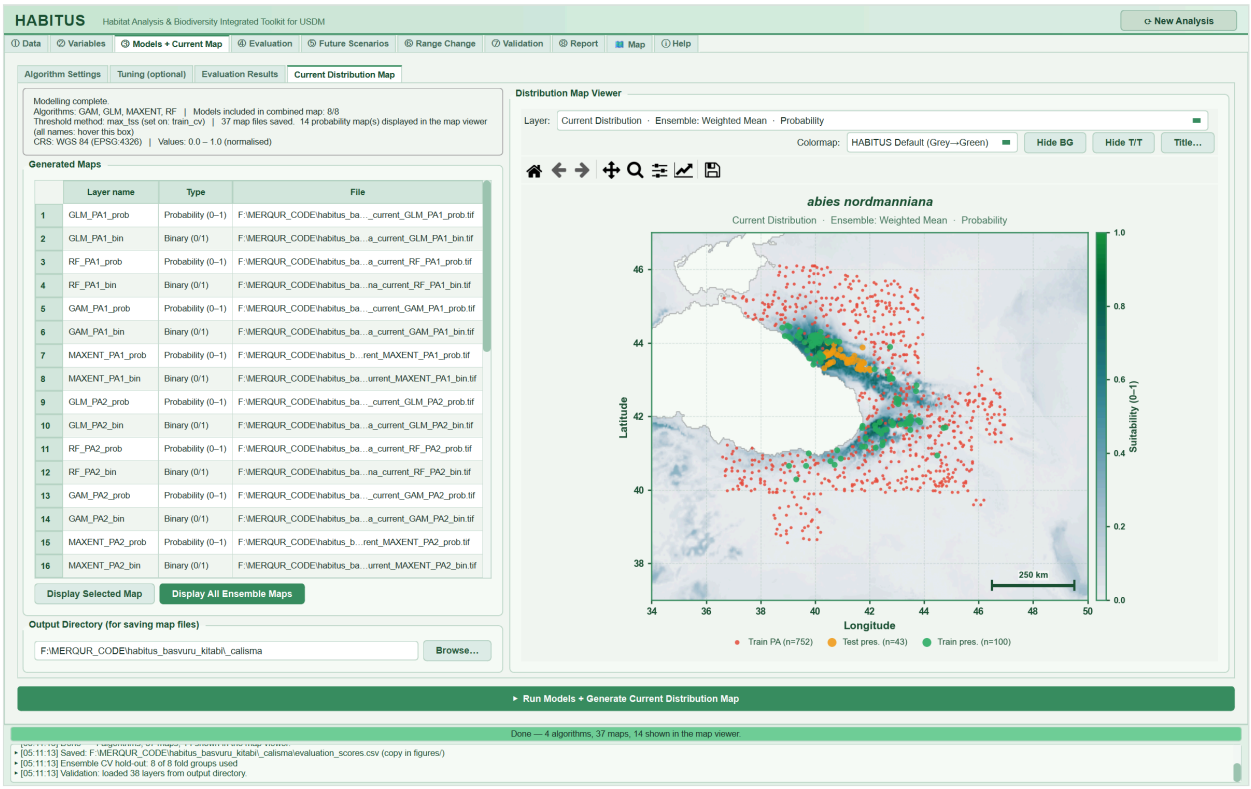
Şekil 6.6. Evaluation Results alt sekmesi örnek koşudan sonra: her algoritma, PA seti ve CV koşusu için ROC-AUC, TSS ve Boyce değerleri.

Tablo `evaluation_scores_table` çıktısının ilk altı sütununu gösterir: `Algorithm`, `PA Set`, `CV Run`, `ROC-AUC`, `TSS`, `Boyce` (`tabs/tab_models.py:618-634`, `1753-1782`). Satırlar önce CV koşuları (`RUN1…`), sonra tam veri modelleri (`Full`, CV ortalamasını taşır), en sonda topluluğun hold-out satırları (`EMwmean`, `EMca` …, `PA Set` = `Ensemble`) gelir. Değerler 0,8 ve üstü yeşil, 0,6–0,8 turuncu, 0,6 altı kırmızı yazılır; sayısal olmayan değerler `N/A` görünür. Topluluk satırlarının ipucu, skorun haritadan değil CV hold-out'larından geldiğini söyler (`tabs/tab_models.py:1769-1774`).

Tablonun tamamı ve diğer sütunlar (eşik, yaygınlık, Brier, kalibrasyon eğimi, Kappa, MCC, duyarlılık, özgüllük, omisyon oranları, `Threshold_source`, `Score_source` …) `evaluation_scores.csv` dosyasındadır; anlamları Bölüm 7'de anlatılır.

> **İpucu:** Renk eşikleri tüm ölçütler için aynıdır; Boyce'un 0,5'in üstü zaten iyi kabul edilir (sekme üstündeki açıklama). Renk bir yargı değil, tabloyu taramayı kolaylaştıran bir işarettir.

6.9 Bugünkü dağılım haritası (Current Distribution Map)



Şekil 6.7. Current Distribution Map alt sekmesi: solda durum metni, Generated Maps tablosu ve çıktı klasörü; sağda ağırlıklı ortalama topluluk haritası, üzerinde eğitim varlıkları (yeşil, n=100), test varlıkları (turuncu, n=43) ve eğitim yalancı yoklukları (kırmızı, n=752).

Şekil 6.7. Current Distribution Map alt sekmesi: solda durum metni, Generated Maps tablosu ve çıktı klasörü; sağda ağırlıklı ortalama topluluk haritası, üzerinde eğitim varlıkları (yeşil, n=100), test varlıkları (turuncu, n=43) ve eğitim yalancı yoklukları (kırmızı, n=752).

Arayüzdeki ad	Ne yapar	Varsayılan	Kod
Durum metni	Algoritmalar, topluluğa giren model sayısı, eşik kuralı ve kaynağı, harita sayısı, CRS; uzun katman listesi ipucunda	—	`tabs/tab_models.py:643-651`, `1817-1834`
`Generated Maps` tablosu	`Layer name`, `Type` (`Probability (0–1)` / `Binary (0/1)`), `File`	—	`tabs/tab_models.py:654-666`, `1798-1812`
`Display Selected Map`	Seçili satırların haritasını görüntüleyiciye yükler	—	`tabs/tab_models.py:1963-1977`
`Display All Ensemble Maps`	`EM` ile başlayan tüm haritaları yükler	—	`tabs/tab_models.py:1979-1984`
`Output Directory (for saving map files)`	Haritaların yazılacağı klasör; boşsa ① Data'daki çıktı klasörü	① Data'dan	`tabs/tab_models.py:679-686`, `1467-1477`
`Distribution Map Viewer`	Katman seçimi, renk ölçeği, `Hide BG`, `Hide T/T`, `Title…`	—	`tabs/tab_models.py:688-695`; `map_widget.py:346-401`

Çalıştırma bitince tüm olasılık haritaları görüntüleyiciye otomatik yüklenir (`tabs/tab_models.py:1808-1812`); katman adları `Current Distribution · <algoritma> · PA-<n> · Probability` ya da `… · Ensemble: Weighted Mean · Probability` biçimindedir (`tabs/tab_models.py:35-47`). Haritanın üzerine ilk PA setinin **ilk CV katındaki** bölme çizilir: eğitim varlıkları yeşil, test varlıkları turuncu, eğitim yalancı yoklukları kırmızı (`tabs/tab_models.py:1859-1907`; `sdm_core.py:5470-5513`). Aynı noktalar `occurrence_train.csv`, `occurrence_test.csv` ve `occurrence_all_splits.csv` olarak çıktı klasörüne yazılır (`tabs/tab_models.py:1914-1961`). `Hide BG` yalancı yokluk noktalarını, `Hide T/T` eğitim/test noktalarını gizler; görüntüleyicinin diğer araçları Bölüm 12'de anlatılır.

Dikkat. Örnek koşuda durum metni `37 map files saved. 14 probability map(s) displayed` der: 8 bireysel model × 2 (olasılık, ikili) + 4 algoritma × 3 (ortalama olasılık, ortalama ikili, SD) + 2 topluluk × 2 + `EMsd` + 4 yenilik katmanı (`NT1`, `NT2`, `MIC_NT1`, `MIC_NT2`) = 37 dosya; görüntüleyiciye yüklenen 14 harita, `_prob` ile biten 8 bireysel, 4 algoritma ortalaması ve 2 topluluk haritasıdır. Haritadaki "test" noktaları bir 80/20 bölmeden değil, ilk mekânsal kattan gelir; bu nedenle birbirine bitişik bir küme oluştururlar (Şekil 6.7'deki turuncu küme). Değerler [0, 1]'e kırpılmış ham olasılıklardır; durum metnindeki `normalised` sözcüğü haritanın kendi aralığına gerildiği anlamına gelmez (`sdm_core.py:7884-7889`).

6.10 Çıktılar ve sonraki adıma geçenler

`<çıktı>/current/` klasörüne (`tabs/tab_models.py:1615`; `sdm_core.py:7895-8103`):

- `<tür>_current_<ALGO>_PA<n>_prob.tif` ve `_bin.tif`: her tam veri modeli (CV modelleri izdüşürülürse `_PA<n>_RUN<k>_…`).
- `<tür>_current_<ALGO>_mean_prob.tif`, `_mean_bin.tif`, `<ALGO>_sd.tif`: algoritma ortalamaları ve yayılımları (iki katmanlı modda).
- `<tür>_current_EMwmean_prob.tif`, `EMca_…`, ek kurallar ve `EMsd.tif` / `EMcv.tif` / `EMrange.tif`.
- `NT1`, `NT2`, `MIC_*` yenilik (ExDet) katmanları ve `projection_options.json` (Bölüm 8).

Çıktı kökünde: `evaluation_scores.csv` (ve `figures/` kopyası), `habitus_run_config.json` (CV tasarımı, kat başına blok boyu, SSB, kat boyutları, eşik ayarları, topluluk üyeleri ve ağırlıkları, hold-out topluluk skorları),

`occurrence_*.csv` ve aramalar yapıldıysa `tuning/tuning_<algo>.csv` (`tabs/tab_models.py:2056-2240`).

Sonraki sekmelere geçenler: modeller ve topluluk nesnesi, topluluk kuralları ve ağırlıkları, iki katmanlı mod ve izdüşürülen yinelemeler (⑤ Future Scenarios aynı kuralları kullanır, `tabs/tab_models.py:1539-1546`), eşikler (⑥ Range Change), harita dosyaları (⑦ Validation katman listeleri otomatik doldurulur, `tabs/tab_models.py:1853-1857`).

6.11 Uzman soruları

S1. Mekânsal blok boyu nasıl belirleniyor; blockCV ile aynı mı? Cevap: Hayır, aynı mantıkta ama sadeleştirilmiş. HABITUS varlık noktalarındaki her öngörücü için 15 sınıflı yönsüz ampirik variogram kurar, dış üçte bir sınıfların medyanını eşik alır, eşiğin %95'ine ilk ulaşılan uzaklığı menzil sayar ve öngörücüler üzerinden medyanı blok boyu yapar (`sdm_spatial.py:166-281`). blockCV ise raster hücrelerinden ya da noktalardan örnekleyip bir variogram modeli uydurarak menzili tahmin eder (Valavi vd. 2019). HABITUS'ta variogram modeli uydurulmaz ve değer yalnız varlık noktalarındaki öngörücü değerlerinden çıkar; sonuç blockCV'den farklı olabilir. Karşılaştırmalı bir değer isteniyorsa blockCV'de bulunan menzil `Block size:` alanına elle girilmelidir.

S2. Blok tasarımı yeterli bloğu kuramazsa ne oluyor; skorlar sessizce rastgele bölmeye mi dönüyor? Cevap: Önce blok altı kez yarıya indirilir ve bu not edilir; yine de istenen kat kadar blok yoksa ya da ikiden az kat her iki sınıfı içeriyorsa rastgele katlara düşülür, `degraded` işaretlenir, not `WARNING (design not applied):` ile başlar ve çalıştırma özetine `[BLOCKING FAILED - scores are optimistic]` eklenir (`sdm_spatial.py:454-495`; `sdm_core.py:5135-5140`, `5933-5934`). Sessiz bir geri dönüş yoktur; ancak kullanıcının günlüğü ya da raporu okuması gerekir.

S3. Haritayı çizen tam veri modelinin bağımsız bir skoru var mı? Cevap: Yoktur ve olamaz; tüm veriyle kurulmuştur. Program `Full` satırına aynı algoritma ve PA setinin CV kat skorlarının ortalamasını (ve kat eşiklerinin ortalamasını) yazar, `Score_source` = `cv_mean` (`sdm_core.py:5818-5853`). Topluluk üyelerini seçen ve ağırlıklayan da bu ortalamadır. Bu, biomod2'deki "tam modelin skoru = CV skorlarının özeti" yaklaşımına benzer.

S4. İkili eşik test verisinde mi seçiliyor? Eşiğe bağlı skorlar iyimser mi? Cevap: Varsayılan `train_cv`'de hayır: eşik, eğitim satırlarının üç iç katlı kat dışı tahminlerinde seçilip test katına uygulanır (`sdm_core.py:4876-4927`, `1193-1220`). `test` seçeneği eşiği test katında en iyiler; kod ve ipucu bunun TSS'yi tipik olarak 0,1–0,2 şişirdiğini belirtir. Bu fark, R'da `ecospat.max.tss` ya da biomod2'nin değerlendirme verisinde eşik araması yapan akışlarıyla karşılaştırmada önemlidir: HABITUS'un varsayılan TSS'si bunlardan sistematik olarak düşük çıkabilir.

S5. Topluluk skoru topluluğun kendi haritasından mı hesaplanıyor? Cevap: Hayır. Her PA × kat grubunda o katı görmemiş CV modelleri topluluk kuralıyla birleştirilir ve yalnız o katta skorlanır; üye seçimi ve ağırlıklar diğer katların skorlarından gelir (bir katı dışarıda bırakma) (`sdm_core.py:6463-6696`). Haritadan yeniden yerine koyma skoru (`evaluate`) bu tabloya hiç girmez (`sdm_core.py:6698-6706`). Bu tasarım, topluluğun seçim ve ağırlıklandırma adımını da doğrulamanın içine alır; biomod2'nin topluluk değerlendirmesinden daha temkinlidir.

S6. İki katmanlı topluluk neden var; biomod2'nin EMwmean'inden farkı ne? Cevap: biomod2'de her bireysel model (PA × koşu) topluluğa ayrı üye olarak girer. HABITUS varsayılan olarak önce algoritmanın yinelemelerini ortalar, sonra algoritmaları ağırlıklar (`sdm_core.py:6109-6132`, `6359-6376`). Böylece yinelemesi çok olan algoritma baskın olmaz ve algoritma içi kararsızlık ayrı bir SD haritasıyla görünür. Kutu kaldırılırsa düz birleşime dönülür; biomod2 ile karşılaştırma için bu seçenek kullanılabilir, ancak biomod2'nin ağırlık tanımı (sıralama tabanlı `decay` dâhil) birebir aynı değildir.

S7. Ağırlıklar nasıl hesaplanıyor; negatif bir skor ne oluyor? Cevap: $w_i = \max(s_i, 0)^p / \sum_j \max(s_j, 0)^p$; negatif skorlar 0'a çekilir, sonlu olmayanlar eşit pay alır, tüm ağırlıklar sıfırsa eşit ağırlık verilir (`sdm_core.py:5988-6005`). Örnek koşuda TSS ortalamaları 0,687 (GLM), 0,566 (RF), 0,611 (GAM), 0,672 (MaxEnt) iken ağırlıklar 0,271/0,223/0,241/0,265'tir; yani $p = 1$'de ağırlıklar birbirine yakındır. Ayrım isteniyorsa p artırılır.

S8. SSB nasıl hesaplanıyor ve dismo::ssb ile aynı mı? Cevap: Kat başına test varlıklarının ve test yokluklarının en yakın eğitim varlığına uzaklığı alınır, havuzlanır ve oranlanır (`sdm_spatial.py:808-881`). dismo::ssb de test-varlık→eğitim-varlık uzaklığını test-arka plan→eğitim-varlık uzaklığına oranlar (Hijmans 2012); HABITUS arka plan olarak test katındaki yalancı yoklukları kullanır ve 3000'den fazla noktayı tohumlu alt örnekler. Yorum eşikleri (0,7 ve 1,3) HABITUS'a özgüdür, yayımlanmış bir kesme noktası değildir.

S9. Ayar araması (tuning) sonrası skorlar iyimser mi? İç içe doğrulama yapılıyor mu? Cevap: Evet iyimserdir; iç içe doğrulama yapılmaz. En iyi nokta, raporlanan skorları üreten aynı katlarda seçilir; program bunu `SELECTION_BIAS_NOTE` ile günlüğe ve arayüze yazar ve iyimserliğin büyüklüğünün ölçülmediğini söyler (`sdm_tuning.py:68-78`; `tabs/tab_tuning.py:396`). Kullanıcı, ayarlanmış algoritmayı bağımsız bir test kümesiyle (`External test points (CSV)`) ya da ⑦ Validation'daki bağımsız haritayla sınayarak bu yanlılığı kısmen giderebilir.

S10. ENMeval'deki delta-AICc ölçütü neden yok? Cevap: elapid MaxEnt'i cezalı, ağırlıklı lojistik regresyon olarak uydurur ve Maxent olabilirliğini vermez; HABITUS bu nedenle AICc'yi uydurmak yerine hiç hesaplamaz (`sdm_tuning.py:38-46`, `62-66`). Seçim kat dışı omisyon ve AUC'ye dayanır (Warren ve Seifert 2011'in karmaşıklık uyarısı bu yolla yalnız dolaylı karşılanır).

S11. Dış test CSV'sinde `RUN1…RUN4` arasındaki standart sapma bir belirsizlik ölçüsü mü? Cevap: Hayır. Her koşu aynı tüm veriyle eğitilir ve aynı test noktalarında sınanır (`sdm_core.py:5417-5427`); fark yalnız algoritmanın kendi rastlantısallığından gelir. Program bunu rapora not olarak yazar (`sdm_core.py:5341-5347`). Belirsizlik için PA setleri arası fark ya da bloklu CV kullanılmalıdır.

S12. Hangi ayar haritayı en çok değiştirir? Cevap: Olasılık haritası için algoritma seçimi ve topluluğa giriş eşiği (§6.3, §6.6.4); ikili harita ve alan için eşik kuralı ve eşik kaynağı (§6.5.1–6.5.2), çünkü haritadaki eşik CV katlarında seçilen eşiklerin ortalamasıdır. CV tasarımı haritayı doğrudan değiştirmez ama skorlar üzerinden hem üye seçimini hem ağırlıkları hem de eşiği değiştirir. Örnek koşuda eşikler 0,06–0,21 aralığında olduğundan eşik kuralındaki küçük farklar alanı belirgin biçimde değiştirebilir; bunun duyarlılığı ölçülmek isteniyorsa ④ Evaluation'daki `Classification Thresholds` tablosu kullanılabilir (Bölüm 7).

6.12 Sık hatalar ve tuzaklar

- `random` tasarımıyla elde edilen yüksek AUC'yi mekânsal bloklu sonuçlarla aynı tabloda karşılaştırmak. Tasarım her satırın `_cv_strategy` alanında ve çalıştırma kaydında yazılıdır.
- `Min score to include model:` değerini AUC alışkanlığıyla 0,7 gibi yüksek tutup ölçütü TSS'ye çevirmek: hiçbir model geçmez ve `WARNING - NO COMBINED MAP:` görülür. Kullanıcı değer yazmamışsa varsayılan ölçütü izler; yazmışsa izlemez.
- `checkerboard2` ile `Cross-validation folds:` değerini 4'ün altına indirmek: dört katın bir kısmı hiç test edilmez.
- `Training data %:` değerini değiştirerek skorların değişmesini beklemek; CV başarılıysa değişmez (§6.4.6).
- `Test-fold threshold:` = `test` ile üretilen TSS'yi makalede bildirmek; arayüz bu seçimde turuncu uyarı gösterir.
- Haritadaki turuncu "test" noktalarını bir rastgele test kümesi sanmak; bunlar ilk mekânsal katın noktalarıdır.
- `Also project the cross-validation models` kapalıyken yalnız CV modelleri eşiği geçtiğinde topluluğun neden yazılmadığını anlamamak; ileti bu kutuyu işaretlemeyi önerir.
- Tek algoritmalı bir çalıştırmada `EMsd` aramak: en az iki üye gerekir.
- Ayar aramasında seçilen noktayı `Apply selected setting to Models tab` ile uygulamadan ana çalıştırmayı başlatmak; arama sonucu ayarları kendiliğinden değiştirmez. Çalıştırma kaydındaki `selected_options_used_in_this_run` alanı bunu gösterir.

6.13 Kaynaklar

Allouche, O., Tsoar, A., Kadmon, R., 2006. Assessing the accuracy of species distribution models: prevalence, kappa and the true skill statistic (TSS). Journal of Applied Ecology 43, 1223–1232. https://doi.org/10.1111/j.1365-2664.2006.01214.x

Araújo, M.B., New, M., 2007. Ensemble forecasting of species distributions. Trends in Ecology & Evolution 22, 42–47. https://doi.org/10.1016/j.tree.2006.09.010

Boyce, M.S., Vernier, P.R., Nielsen, S.E., Schmiegelow, F.K.A., 2002. Evaluating resource selection functions. Ecological Modelling 157, 281–300. https://doi.org/10.1016/S0304-3800(02)00200-4

Fielding, A.H., Bell, J.F., 1997. A review of methods for the assessment of prediction errors in conservation presence/absence models. Environmental Conservation 24, 38–49. https://doi.org/10.1017/S0376892997000088

Hijmans, R.J., 2012. Cross-validation of species distribution models: removing spatial sorting bias and calibration with a null model. Ecology 93, 679–688. https://doi.org/10.1890/11-0826.1

Hirzel, A.H., Le Lay, G., Helfer, V., Randin, C., Guisan, A., 2006. Evaluating the ability of habitat suitability models to predict species presences. Ecological Modelling 199, 142–152. https://doi.org/10.1016/j.ecolmodel.2006.05.017

Kass, J.M., Muscarella, R., Galante, P.J., Bohl, C.L., Pinilla-Buitrago, G.E., Boria, R.A., Soley-Guardia, M., Anderson, R.P., 2021. ENMeval 2.0: Redesigned for customizable and reproducible modeling of species' niches and distributions. Methods in Ecology and Evolution 12, 1602–1608. https://doi.org/10.1111/2041-210X.13628

Li, W., Guo, Q., 2013. How to assess the prediction accuracy of species presence–absence models without absence data? Ecography 36, 788–799. https://doi.org/10.1111/j.1600-0587.2013.07585.x

Liu, C., Berry, P.M., Dawson, T.P., Pearson, R.G., 2005. Selecting thresholds of occurrence in the prediction of species distributions. Ecography 28, 385–393. https://doi.org/10.1111/j.0906-7590.2005.03957.x

Marmion, M., Parviainen, M., Luoto, M., Heikkinen, R.K., Thuiller, W., 2009. Evaluation of consensus methods in predictive species distribution modelling. Diversity and Distributions 15, 59–69. https://doi.org/10.1111/j.1472-4642.2008.00491.x

Muscarella, R., Galante, P.J., Soley-Guardia, M., Boria, R.A., Kass, J.M., Uriarte, M., Anderson, R.P., 2014. ENMeval: An R package for conducting spatially independent evaluations and estimating optimal model complexity for Maxent ecological niche models. Methods in Ecology and Evolution 5, 1198–1205. https://doi.org/10.1111/2041-210X.12261

Pearson, R.G., Raxworthy, C.J., Nakamura, M., Peterson, A.T., 2007. Predicting species distributions from small numbers of occurrence records: a test case using cryptic geckos in Madagascar. Journal of Biogeography 34, 102–117. https://doi.org/10.1111/j.1365-2699.2006.01594.x

Roberts, D.R., Bahn, V., Ciuti, S., Boyce, M.S., Elith, J., Guillera-Arroita, G., Hauenstein, S., Lahoz-Monfort, J.J., Schröder, B., Thuiller, W., Warton, D.I., Wintle, B.A., Hartig, F., Dormann, C.F., 2017. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. Ecography 40, 913–929. https://doi.org/10.1111/ecog.02881

Valavi, R., Elith, J., Lahoz-Monfort, J.J., Guillera-Arroita, G., 2019. blockCV: An R package for generating spatially or environmentally separated folds for k-fold cross-validation of species distribution models. Methods in Ecology and Evolution 10, 225–232. https://doi.org/10.1111/2041-210X.13107

Velazco, S.J.E., Rose, M.B., de Andrade, A.F.A., Minoli, I., Franklin, J., 2022. flexsdm: An R package for supporting a comprehensive and flexible species distribution modelling workflow. Methods in Ecology and Evolution 13, 1661–1669. https://doi.org/10.1111/2041-210X.13874

Warren, D.L., Seifert, S.N., 2011. Ecological niche modeling in Maxent: the importance of model complexity and the performance of model selection criteria. Ecological Applications 21, 335–342. https://doi.org/10.1890/10-1171.1

7. Değerlendirme (④ Evaluation)

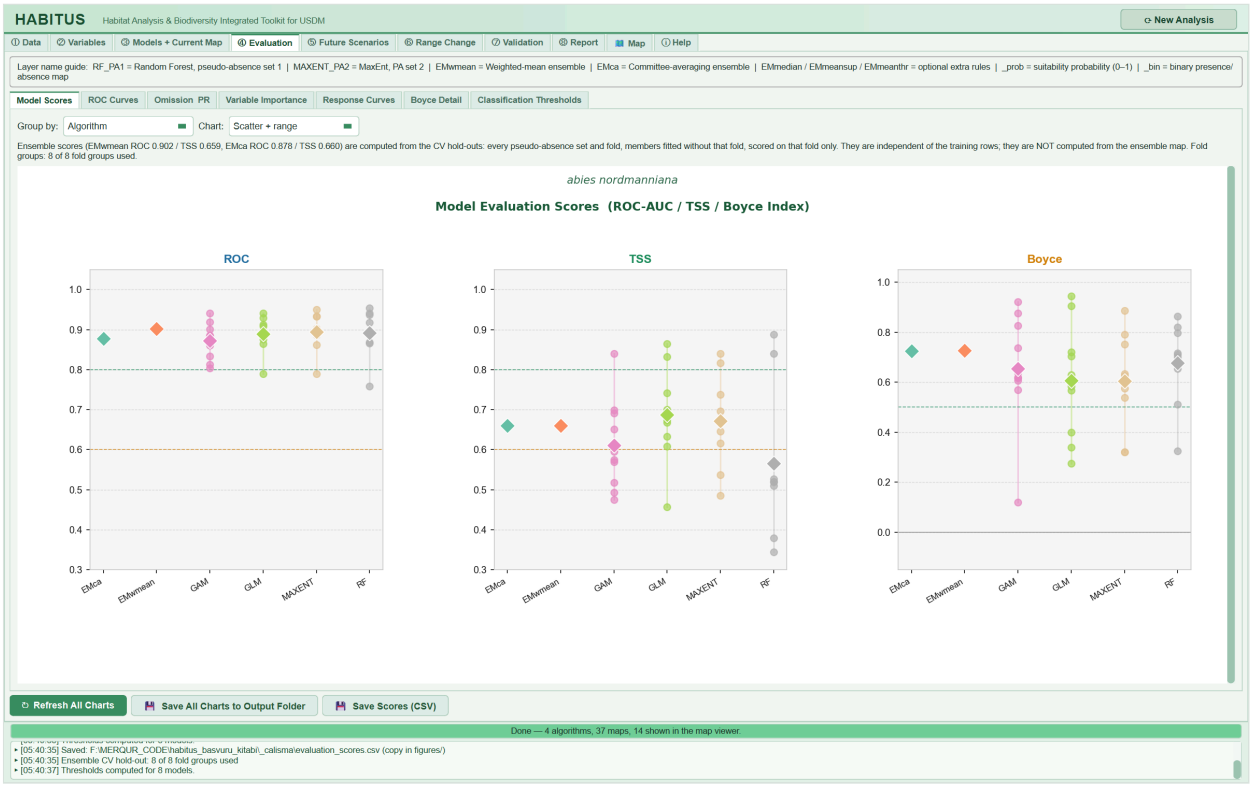
Sekme: ④ Evaluation · Kod: `tabs/tab_evaluation.py` (arayüz); `sdm_core.py` `evaluate_all`, `boyce_ecospat`, `compute_calibration`, `cv_roc_curves`, `cv_omission_pr_calibration`, `cv_boyce_curves`, `cv_threshold_performance`, `compute_response_curves`, `evaluation_scores_table` (hesap)

7.1 Bu sekme ne yapar

④ Evaluation, ③ Models'ta kurulan modellerin başarımını grafik ve tablolarla gösterir. Yeni bir model kurmaz, veriyi yeniden bölmez: bütün eğri ve skorlar ③ Models çalıştırması sırasında kaydedilen **kat dışı (out-of-fold, CV hold-out) tahminlerden** üretilir (`sdm_core.py:5547-5556`, `6826-7037`). Yalnız iki şey tam veri (`Full`) modellerinden hesaplanır: tepki eğrileri (§7.7) ve `Classification Thresholds` tablosundaki harita eşikleri (§7.9).

- **Girdi:** ③ Models'ın modelleyici (`modeler`) ve topluluk (`ensemble`) nesneleri, PA setleri, çıktı klasörü.
- **Çıktı:** ekranda yedi alt sekme; `figures/` klasörüne PNG şekiller (300 dpi), `evaluation_scores.csv`, `evaluation_omission_pr_calibration.csv`, `classification_thresholds.csv` (§7.11).
- **Kilit düzeni:** ③ Models çalıştırması bitince açılır (`tabs/tab_models.py:1848`). Sekme ilk gösterildiğinde tüm grafikler bir kez, ilerleme penceresi eşliğinde hesaplanır; aynı çalıştırma için sekme değiştirmek yeniden hesaplamaz (`tabs/tab_evaluation.py:1344-1371`).

7.2 Ekranın düzeni



Şekil 7.1. ④ Evaluation sekmesi Model Scores alt sekmesi açıkken: üstte katman adı rehberi, ortada yedi alt sekme, Group by = Algorithm ve Chart = Scatter + range ile ROC, TSS, Boyce panelleri (EMca ve EMwmean topluluk satırları dâhil), altta Refresh All Charts, Save All Charts to Output Folder ve Save Scores (CSV) düğmeleri.

Şekil 7.1. ④ Evaluation sekmesi Model Scores alt sekmesi açıkken: üstte katman adı rehberi, ortada yedi alt sekme, Group by = Algorithm ve Chart = Scatter + range ile ROC, TSS, Boyce panelleri (EMca ve EMwmean topluluk satırları dâhil), altta Refresh All Charts, Save All Charts to Output Folder ve Save Scores (CSV) düğmeleri.

Bölge	İçerik	Bu bölümde
Üst şerit	`Layer name guide:` katman adlarının anlamı (`RF_PA1`, `EMwmean`, `_prob`, `_bin` …)	§7.2
`Model Scores`	ROC, TSS, Boyce dağılımları	§7.3
`ROC Curves`	Kat dışı ROC eğrileri	§7.4
`Omission & PR`	Omisyon, kesinlik-duyarlılık, kalibrasyon	§7.5
`Variable Importance`	Permütasyon önemi	§7.6
`Response Curves`	Tepki eğrileri (marginal / partial / ice)	§7.7
`Boyce Detail`	Kat başına Boyce P/E eğrileri	§7.8
`Classification Thresholds`	On bir eşik kuralının eşikleri ve kat dışı başarımı	§7.9
Alt düğmeler	`↻ Refresh All Charts`, `💾 Save All Charts to Output Folder`, `💾 Save Scores (CSV)`	§7.10

Alt sekme şeridinde üçüncü sekmenin adı ekranda `Omission PR` olarak görünür; kodda adı `Omission & PR`'dır (`tabs/tab_evaluation.py:201`). Bu bölümde koddaki ad kullanılır.

Çok satırlı şekiller (skorlar, ROC, omisyon, tepki, Boyce) kendi kaydırma alanındadır; şeklin altında devamı varsa `▼ Scroll down for more of this figure` ipucu görünür (`tabs/tab_evaluation.py:69-101`).

7.3 Model skorları (Model Scores)

Bu alt sekme (Şekil 7.1) `evaluation_scores.csv` tablosundaki ROC, TSS ve Boyce sütunlarını üç panelde çizer (`tabs/tab_evaluation.py:229-353`). Tablonun kendisi her ölçütü içerir; ölçütlerin tanımı §7.3.2'dedir.

Arayüzdeki ad	Ne yapar	Varsayılan	Seçenekler	Kod
`Group by:`	Noktaları gruplayan sütun	`Algorithm`	`Algorithm` / `PA Set` / `CV Run`	`tabs/tab_evaluation.py:233-235`
`Chart:`	Grafik türü	`Scatter + range`	`Scatter + range` / `Box plot` / `Bar mean±SD`	`tabs/tab_evaluation.py:237-239`
Topluluk notu	Topluluk satırlarının hold-out skorlarını ve kat gruplarını yazar	—	—	`tabs/tab_evaluation.py:261-281`

7.3.1 Grafik ve gruplama

Ne işe yarar. `Scatter + range` her satırı bir nokta, grubun en küçük–en büyük aralığını bir çizgi ve ortalamayı bir elmas olarak çizer; `Box plot` kutu grafiği; `Bar mean±SD` ortalama ve standart sapma çubuğu çizer (`tabs/tab_evaluation.py:315-331`). ROC ve TSS panellerinde 0,8 (yeşil) ve 0,6 (turuncu) kesikli çizgileri, Boyce panelinde 0,5 ve 0 çizgileri vardır. Grup sayısı altıyı geçerse etiketler dik yazılır.

Ne zaman, nasıl ayarlanır. `Algorithm` algoritmaları karşılaştırır; `PA Set` yalancı yokluk setleri arasındaki farkı (arka plan çekiminin etkisini); `CV Run` katlar arasındaki farkı (bazı katların sistematik olarak zor olup olmadığını) gösterir.

Sonuca etkisi. Yalnız görünümü değiştirir; tablo ve dosyalar değişmez.

Dikkat. Grafik tablodaki **bütün** satırları kullanır: CV koşuları, CV ortalamasını taşıyan `Full` satırları ve topluluk satırları (`PA Set = Ensemble`). `Full` satırları ortalamanın kopyası olduğundan `Algorithm` gruplamasında bir algoritmanın dağılımına PA seti başına bir ortalama noktası daha eklenmiş olur; yayılımı okurken `evaluation_scores.csv`'de `Score_source = cv_fold` satırlarını esas alın. Eksen alt sınırları ROC ve TSS için 0,3, Boyce için −0,15'tir (`tabs/tab_evaluation.py:314`); bunun altındaki değerler panelde görünmez.

7.3.2 Skor tablosundaki ölçütler

Her satır `evaluate_all` ile bir modelin tahminlerinden hesaplanır (`sdm_core.py:1122-1257`). Eşiğe bağlı ölçütler, ③ Models'ta seçilen eşik kuralı ve eşik kaynağıyla (`train_cv` varsayılan) bulunan eşikte hesaplanır (Bölüm 6, §6.5). TP, FN, FP, TN karışıklık matrisinin hücreleridir; bir payda sıfırsa ölçüt `NaN` olur, 0 olmaz (`sdm_core.py:693-719`).

Sütun	Tanım	Eşiğe bağlı	Kod
`ROC`	ROC eğrisi altındaki alan (AUC)	hayır	`sdm_core.py:951-959`
`TSS`	Duyarlılık + özgüllük − 1 (Allouche vd. 2006)	evet	`sdm_core.py:671-679`

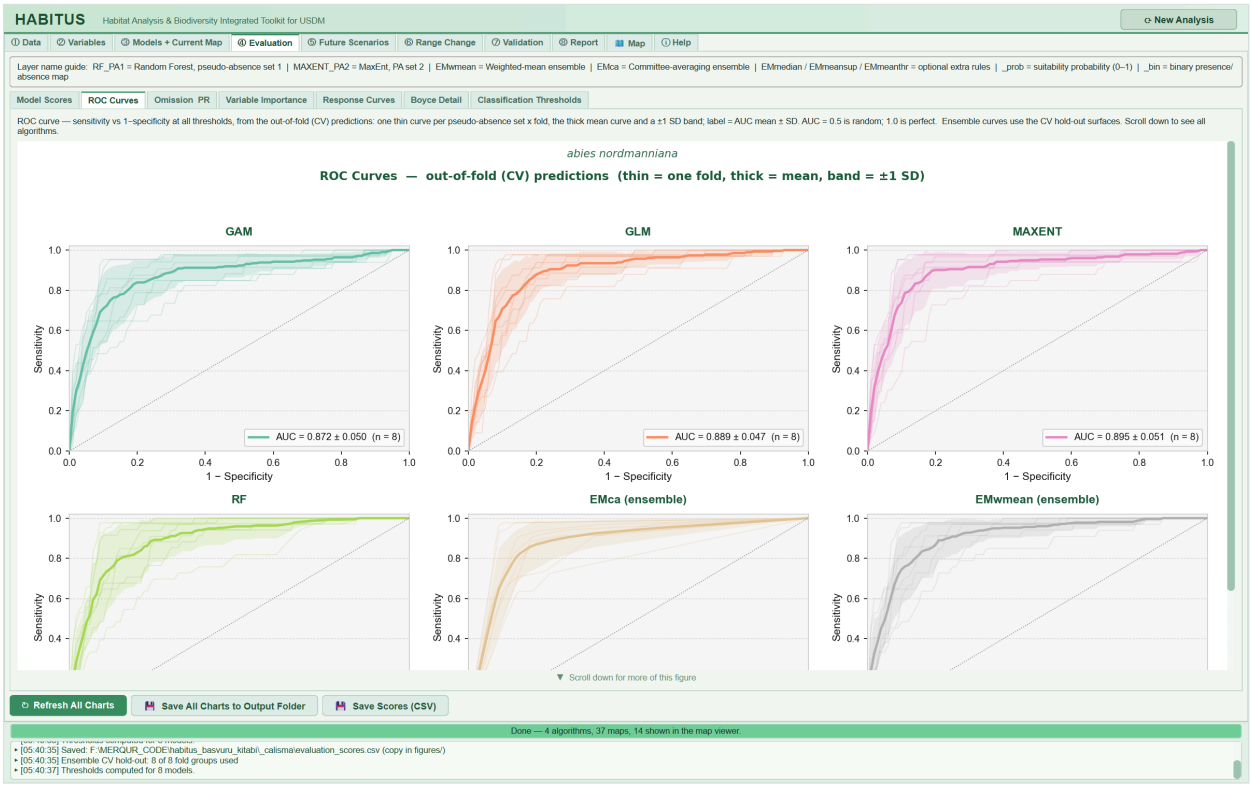
Sütun	Tanım	Eşiğe bağlı	Kod
`Boyce`	Sürekli Boyce indeksi (ecospat ya da legacy tanımı)	hayır	`sdm_core.py:475-664`
`Threshold`	Kullanılan ikili eşik	—	`sdm_core.py:1242`
`Prevalence`, `N_pres`, `N_abs`	Değerlendirme kümesinde varlık oranı ve sayıları	hayır	`sdm_core.py:1249-1252`
`PredPresFrac`	Eşiğin üstünde kalan satırların payı	evet	`sdm_core.py:1235-1236`
`Brier`	$\overline{(p-y)^2}$; küçük daha iyi	hayır	`sdm_core.py:998`
`CalSlope`	y'nin logit(p) üzerine lojistik regresyonunun eğimi; 1 iyi kalibrasyon, <1 aşırı güvenli (Pearce ve Ferrier 2000)	hayır	`sdm_core.py:1000-1013`
`Sensitivity`, `Specificity`	TP/(TP+FN), TN/(TN+FP)	evet	`sdm_core.py:709-710`
`Kappa`	Cohen kappa'sı (Fielding ve Bell 1997)	evet	`sdm_core.py:711-712`
`MCC`	Matthews korelasyon katsayısı	evet	`sdm_core.py:713-714`
`Jaccard`, `Sorensen`, `FPB`	TP/(TP+FP+FN); 2TP/(2TP+FP+FN); 2 × Jaccard (Li ve Guo 2013)	evet	`sdm_core.py:715-719`
`OR_MTP`, `OR_10p`	Test varlıklarının, eğitim varlıklarının en düşük / 10. yüzdelik tahmininin altında kalan payı	eşik referansa bağlı	`sdm_core.py:1095-1099`
`AUC_train`, `dAUC`	Eğitim verisindeki AUC; `AUC_train − ROC` (aşırı uyum farkı)	hayır	`sdm_core.py:1074-1084`
`OR_MTP_degenerate`, `OR_10p_degenerate`	Omisyon eşiği tahminlerin en küçüğüne eşit ya da altındaysa 1 (omisyon yapı gereği 0)	—	`sdm_core.py:1100-1114`
`Threshold_source`, `OR_reference`, `Boyce_formula`	Eşiğin, omisyon referansının ve Boyce tanımının kaynağı	—	`sdm_core.py:5877-5879`
`Score_source`	`cv_fold`, `cv_mean`, `random_split`, `cv_holdout`	—	`sdm_core.py:1038-1053`

`Score_source` değerleri: `cv_fold` bir CV koşusunun kendi tutulan katında skorudur; `cv_mean` tam veri modeline taşınan CV ortalamasıdır; `random_split` CV skoru oluşmamış bir tam veri modelinin rastgele bölme skorudur (iyimser); `cv_holdout` topluluk satırıdır (Bölüm 6, §6.6.7).

> **Not:** `compute_calibration` ayrıca kalibrasyon kesişimi (`CalIntercept`), en büyük ve beklenen kalibrasyon hatası (`MCE`, `ECE`) hesaplar (`sdm_core.py:981-1033`); bu üç değer `evaluation_scores.csv`'ye yazılmaz.

Örnek koşudan bir satır (GLM, PA1, RUN1): ROC 0,941, TSS 0,865, Boyce 0,399, eşik 0,177, yaygınlık 0,148 (43 varlık, 248 yokluk), duyarlılık 0,953, özgüllük 0,911, kalibrasyon eğimi 1,28, `AUC_train` 0,900 (`dAUC` = −0,041). Aynı algoritmanın RUN2 katında ROC 0,790, TSS 0,457'dir; katlar arası fark algoritmalar arası farktan büyüktür.

7.4 ROC eğrileri (ROC Curves)



Şekil 7.2. ROC Curves alt sekmesi: her algoritma ve topluluk kuralı için ayrı panel; ince çizgiler tek tek katlar, kalın çizgi ortalama eğri, gölge ±1 SD bandı.

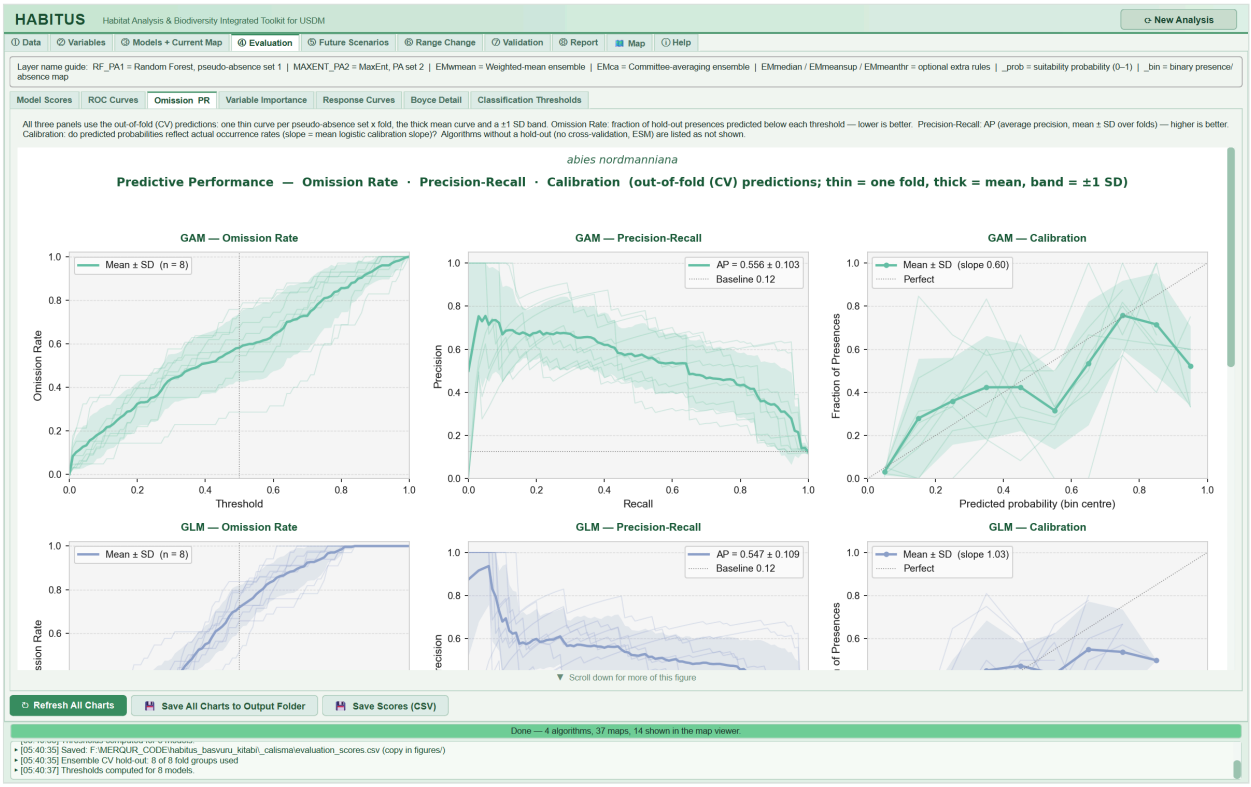
Şekil 7.2. ROC Curves alt sekmesi: her algoritma ve topluluk kuralı için ayrı panel; ince çizgiler tek tek katlar, kalın çizgi ortalama eğri, gölge ±1 SD bandı.

Ne işe yarar. Her algoritma için her PA seti × kat grubunun kat dışı tahminlerinden bir ROC eğrisi çizilir; eğriler 101 noktalı ortak bir yanlış pozitif oranı ızgarasına aradeğerlenir, ortalama eğri ve ±1 SD bandı hesaplanır; açıklama `AUC = ortalama ± SD (n = kat grubu sayısı)` biçimindedir (`cv_roc_curves`, `sdm_core.py:6826-6891`; `tabs/tab_evaluation.py:372-435`). Topluluk kuralları (`EMwmean (ensemble)` …) topluluğun hold-out yüzeylerinden çizilir (Bölüm 6, §6.6.7). Ayar yoktur.

Sonuca etkisi. Ekran ve `evaluation_roc_curves.png`; skorlar değişmez. Örnek koşuda sekiz kat grubu üzerinden AUC: GAM 0,872 ± 0,050, GLM 0,889 ± 0,047, MaxEnt 0,895 ± 0,051.

Dikkat. Tek sınıflı bir hold-out atlanır; hiç kat dışı tahmini olmayan algoritma şeklin altında `Not shown: …` notuyla listelenir (`sdm_core.py:6872-6876`; `tabs/tab_evaluation.py:427-429`). Yokluk sınıfı yalancı yokluk olduğundan AUC "varlık noktalarını arka plandan ayırma" başarısıdır; gerçek yokluklara karşı başarının ölçüsü değildir.

7.5 Omisyon, kesinlik-duyarlılık ve kalibrasyon (Omission & PR)



Şekil 7.3. Omission & PR alt sekmesi: her algoritma için bir satırda omisyon oranı, kesinlik-duyarlılık ve kalibrasyon panelleri, kat başına ince eğriler ve ortalama.

Şekil 7.3. Omission & PR alt sekmesi: her algoritma için bir satırda omisyon oranı, kesinlik-duyarlılık ve kalibrasyon panelleri, kat başına ince eğriler ve ortalama.

Ne işe yarar. Her algoritma için kat dışı tahminlerden üç panel çizilir (`cv_omission_pr_calibration`, `sdm_core.py:6926-6995`; `tabs/tab_evaluation.py:459-571`):

- **Omission Rate:** 0–1 arasında 101 eşikte, hold-out varlıklarından tahmini eşiğin altında kalanların payı. Düşük daha iyidir; 0,5'te kesikli dikey çizgi vardır.
- **Precision-Recall:** kesinlik-duyarlılık eğrisi, ortak duyarlılık ızgarasına aradeğerlenir; açıklama ortalama kesinlik (average precision, AP) ± SD'dir. Gri yatay çizgi taban kesinliktir (kat yaygınlıklarının ortalaması).
- **Calibration:** tahminler 10 eşit genişlikli kutuya ayrılır; her kutuda varlık payı hesaplanır (3'ten az satırlı kutular boş bırakılır); açıklama katların ortalama kalibrasyon eğimidir. Köşegen tam kalibrasyonu gösterir.

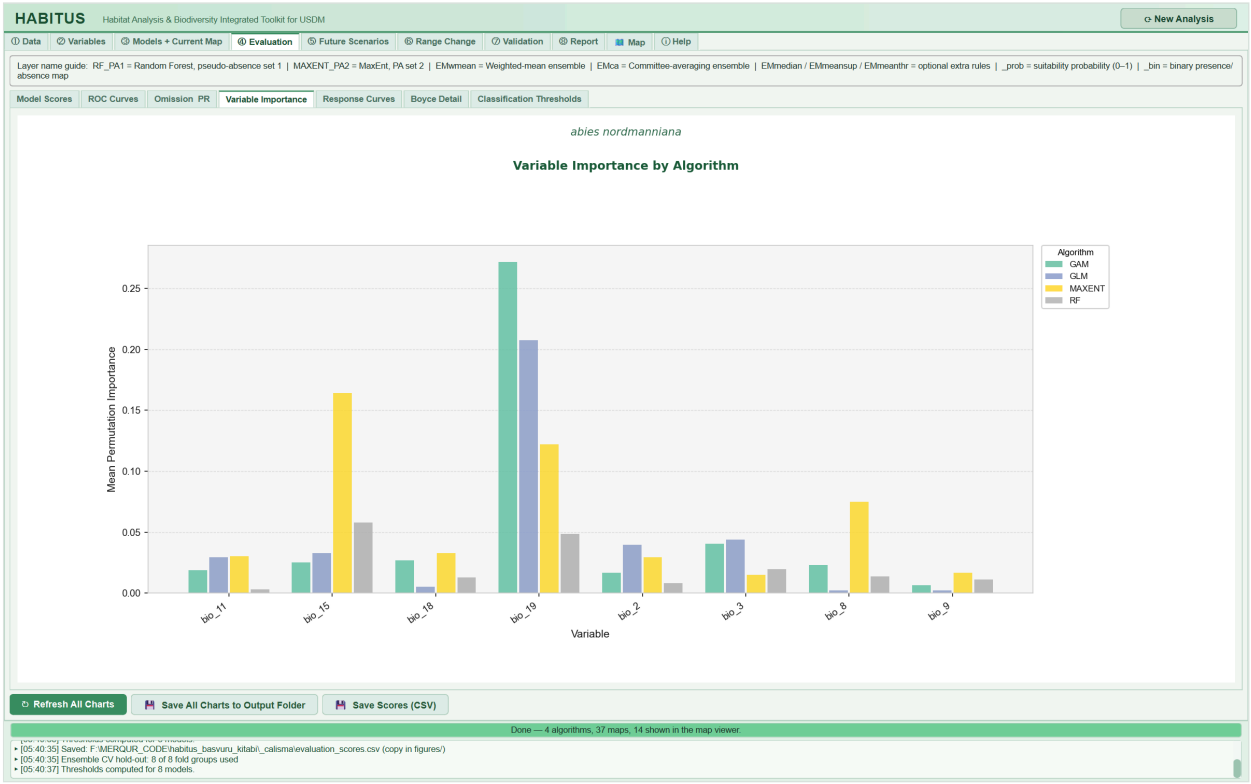
Sonuca etkisi. Ekran, `evaluation_omission_pr_calibration.png` ve her algoritma için 0,5 eşiğindeki ortalama omisyonu, AP'yi, taban kesinliği ve ortalama kalibrasyon eğimini içeren `figures/evaluation_omission_pr_calibration.csv` (`tabs/tab_evaluation.py:573-597`).

Örnek koşuda GAM için AP 0,556 ± 0,103 (taban 0,12) ve ortalama kalibrasyon eğimi 0,60, GLM için AP 0,547 ± 0,109 ve eğim 1,03'tür: iki model benzer biçimde ayırt eder, ama GAM'ın olasılıkları aşırı güvenlidir.

Dikkat. Yalancı yokluklu verilerde kesinlik ve kalibrasyon mutlak değer olarak yorumlanamaz: "yokluk" sınıfı arka plandır, olasılık ölçeği PA:varlık oranına bağlıdır (örnek koşuda 7:1, taban kesinlik yaklaşık 0,12). Bu

paneller algoritmaları ve katları **birbiriyle** karşılaştırmak içindir. Hold-out kümeleri küçük olduğundan (örnek koşuda kat başına 32–43 varlık) ±1 SD bandı bir güven aralığı değildir.

7.6 Değişken önemi (Variable Importance)



Şekil 7.4. Variable Importance alt sekmesi: her değişken için algoritma başına ortalama permütasyon önemi çubukları; açıklama kutusu grafiğin sağında.

Şekil 7.4. Variable Importance alt sekmesi: her değişken için algoritma başına ortalama permütasyon önemi çubukları; açıklama kutusu grafiğin sağında.

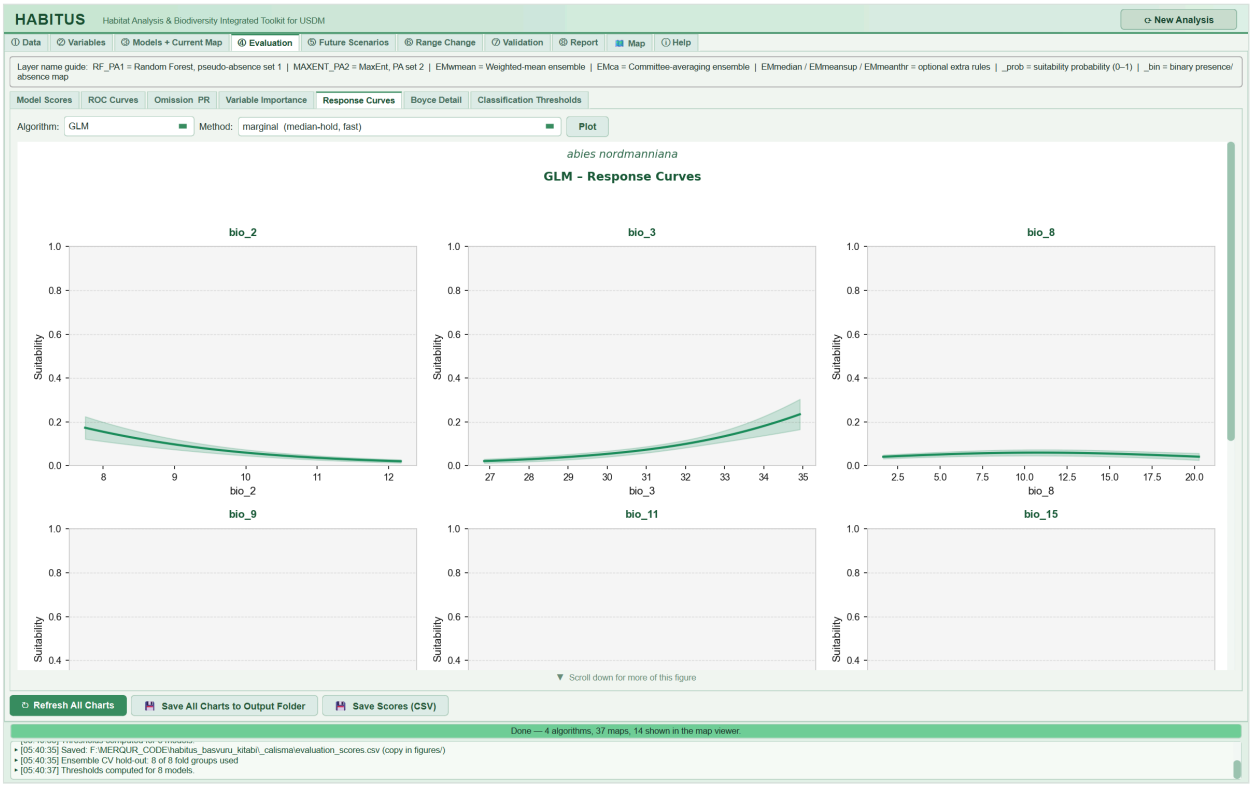
Ne işe yarar. ③ Models'ta her CV koşusunda, test katında hesaplanan permütasyon önemi (öngörücü karıştırıldığında AUC'deki düşüş) algoritma × değişken için tüm katlar ve PA setleri üzerinden ortalanır ve gruplu çubuk grafikte çizilir (`tabs/tab_evaluation.py:607-647`; hesap: `sdm_core.py:5562-5636`). Ayar yoktur; tekrar sayısı ③ Models'taki `Variable importance reps:` alanıdır (Bölüm 6, §6.4.7).

Ne zaman, nasıl ayarlanır. Önem değerleri algoritmalar arasında aynı ölçektedir (AUC düşüşü); yine de tree tabanlı ve doğrusal modellerin önem sıralamaları farklı olabilir, ortak sıralama için algoritmalar arası tutarlılığa bakın.

Sonuca etkisi. Ekran ve `evaluation_variable_importance.png`. Örnek koşuda GAM ve GLM'de en önemli değişken bio_19 (AUC düşüşü sırasıyla yaklaşık 0,27 ve 0,21), MaxEnt'te bio_15'tir (yaklaşık 0,16); RF'de hiçbir değişken 0,06'yı geçmez.

Dikkat. İlişkili öngörücüler önemi paylaşır: biri karıştırıldığında diğeri bilgiyi taşımaya devam eder ve ikisinin önemi de düşük görünebilir. Önem negatif çıkabilir (karıştırma AUC'yi rastlantıyla artırmıştır). `Variable importance reps:` = 0 ise grafik boş kalır. Tam veri modellerinin önemi hesaplanmaz; grafik yalnız CV koşularındandır.

7.7 Tepki eğrileri (Response Curves)



Şekil 7.5. Response Curves alt sekmesi: Algorithm, Method ve Plot denetimleri; seçilen algoritma için her sürekli öngörücüye bir panel, kalın çizgi ortalama tepki, gölge yinelemeler arası standart sapma; ikinci satır panellerin yalnız üst kısmı görünür, eğriler kaydırınca görünen 0,4'ün altındaki bölgededir.

Şekil 7.5. Response Curves alt sekmesi: Algorithm, Method ve Plot denetimleri; seçilen algoritma için her sürekli öngörücüye bir panel, kalın çizgi ortalama tepki, gölge yinelemeler arası standart sapma; ikinci satır panellerin yalnız üst kısmı görünür, eğriler kaydırınca görünen 0,4'ün altındaki bölgededir.

Arayüzdeki ad	Ne yapar	Varsayılan	Seçenekler	Kod
`Algorithm:`	Eğrileri çizilecek algoritma	ilk algoritma	çalıştırmadaki algoritmalar	`tabs/tab_evaluation.py:654-655`, `885-892`
`Method:`	Tepki eğrisi yöntemi	`marginal (median-hold, fast)`	`marginal` / `partial` / `ice`	`tabs/tab_evaluation.py:656-671`
`Plot`	Eğrileri hesaplar ve çizer	—	—	`tabs/tab_evaluation.py:672-673`

7.7.1 Tepki eğrisi yöntemi (Method)

Ne işe yarar. Her sürekli öngörücü, tüm PA setlerinin birleşik verisinde (varlıklar ve yalancı yokluklar) 5. ile 95. yüzdeliği arasında 100 noktada değiştirilir (`compute_response_curves`, `sdm_core.py:8354-8510`):

- `marginal`: diğer bütün öngörücüler birleşik verinin medyanında tutulur (biomod2'deki "medyan tutma" yaklaşımı).
- `partial`: kısmi bağımlılık (PDP; Friedman 2001): tohumlu seçilen en çok 500 gerçek satırda odak değişken her ızgara değerine eşitlenir ve tahminlerin ortalaması alınır.
- `ice`: bireysel koşullu beklenti (ICE; Goldstein vd. 2015): tohumlu seçilen 60 satırın her biri için ayrı bir eğri (ince çizgiler) çizilir; kalın çizgi bu eğrilerin ortalaması, yani o satırların PDP'sidir.

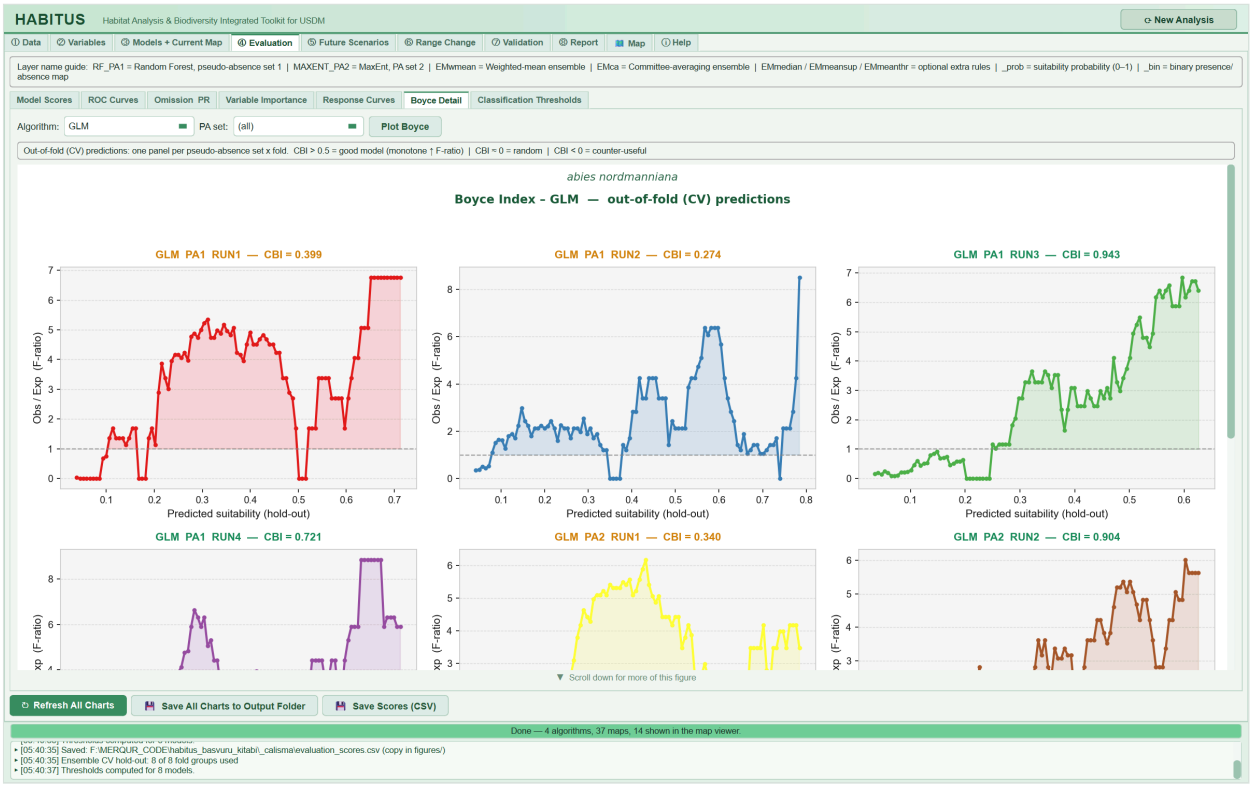
Her yöntemde algoritmanın **tüm kurulan modelleri** (her PA setinin CV koşuları ve tam veri modeli) ayrı ayrı tahmin edilir; kalın çizgi bunların ortalaması, gölge standart sapmasıdır (`sdm_core.py:8461-8506`). Kategorik öngörücüler için her sınıfa bir çubuk çizilir (diğer öngörücüler medyanda; 2–60 sınıflı değişkenler) (`sdm_core.py:8512-8553`).

Ne zaman, nasıl ayarlanır. `marginal` hızlıdır ve biomod2/maxnet çıktılarıyla karşılaştırılabilir. Öngörücüler arasında güçlü etkileşim ya da ilişki varsa medyan satırı gerçekçi olmayan bir öngörücü bileşimi olabilir; `partial` gerçek veri dağılımı üzerinden ortalar. Etkileşimi görmek için `ice` kullanılır: ince eğrilerin birbirinden farklı biçimde olması, odak değişkenin etkisinin diğer değişkenlere bağlı olduğunu gösterir.

Sonuca etkisi. Yalnız ekran ve `evaluation_response_<ALGO>.png`. `partial` yaklaşık 500 × 100 tahmin, `ice` 60 × 100 tahmin × model sayısı gerektirir; MaxEnt ve RF'de belirgin biçimde yavaştır. Sonuç aynı çalıştırma, yöntem ve algoritma için önbelleğe alınır; yeniden çizim tekrar hesaplamaz (`tabs/tab_evaluation.py:690-712`).

Dikkat. Eğriler ③ Models'taki `Clamp predictors …` ayarı uygulanmadan hesaplanır (tahmin işlevleri doğrudan çağrılır, `sdm_core.py:8419-8437`). Y ekseni 0–1 aralığına sabittir; yalancı yokluklu verilerde olasılık ölçeği düşük olduğundan eğriler sıkışık görünebilir (örnek koşuda GLM eğrileri 0,25'in altında kalır). Bir eğrinin gölgesi CV ve tam veri modellerinin farkını gösterir, bir güven aralığı değildir. Şekil 3 sütunludur; dokuzdan fazla öngörücüde alt satırlar için şekli kaydırın.

7.8 Boyce ayrıntısı (Boyce Detail)



Şekil 7.6. Boyce Detail alt sekmesi: seçilen algoritma için her PA seti × kat grubuna bir panel; P/E (F-ratio) eğrisi, 1 düzeyinde kesikli çizgi ve başlıkta o katın CBI değeri.

Şekil 7.6. Boyce Detail alt sekmesi: seçilen algoritma için her PA seti × kat grubuna bir panel; P/E (F-ratio) eğrisi, 1 düzeyinde kesikli çizgi ve başlıkta o katın CBI değeri.

Arayüzdeki ad	Ne yapar	Varsayılan	Seçenekler	Kod
`Algorithm:`	Algoritma	ilk algoritma	çalıştırmadaki algoritmalar	`tabs/tab_evaluation.py:774`
`PA set:`	Gösterilecek PA seti	`(all)`	`(all)`, `PA1`, `PA2`, …	`tabs/tab_evaluation.py:775`, `893-897`
`Plot Boyce`	Panelleri çizer	—	—	`tabs/tab_evaluation.py:776-777`

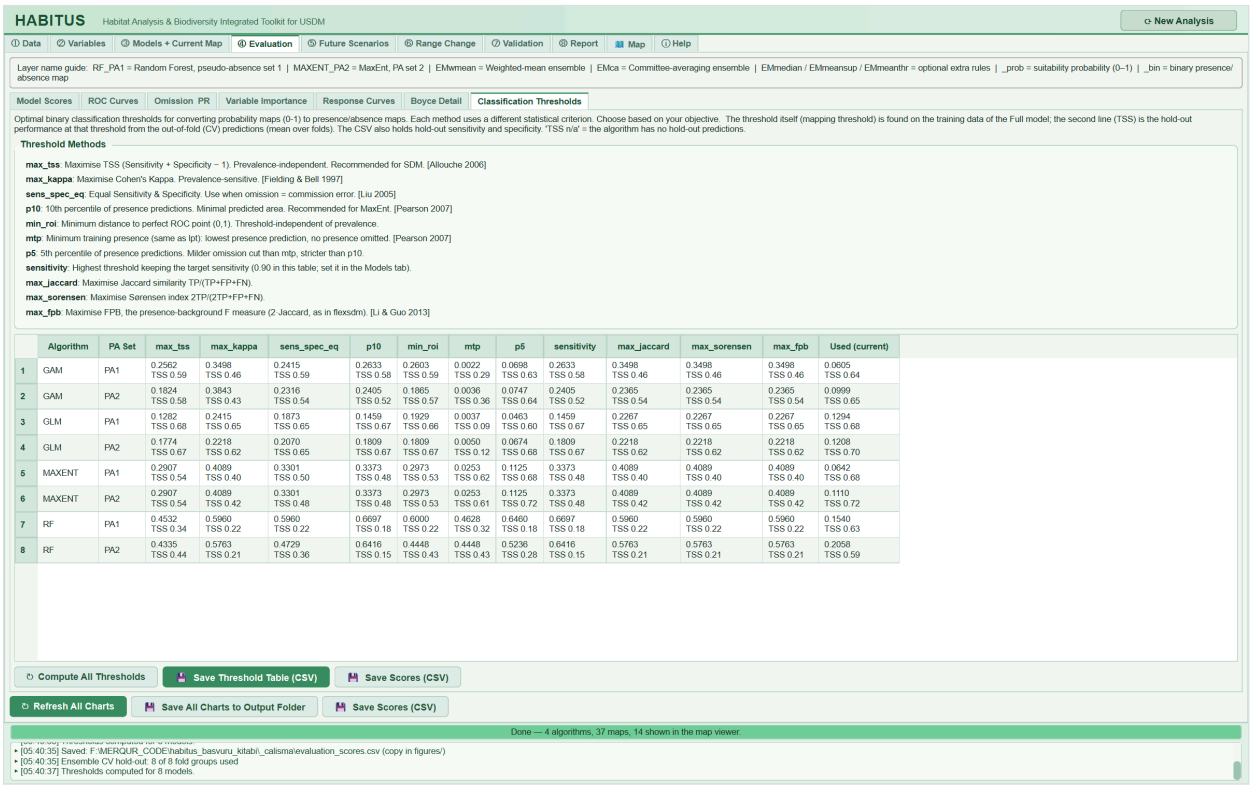
Ne işe yarar. Her PA seti × kat için hold-out tahminlerinden Boyce eğrisi çizilir: x ekseni pencere orta noktası, y ekseni $F = P_i/E_i$ oranıdır; 1'in üstündeki bölge gölgelenir; başlık rengi CBI ≥ 0,5 için yeşil, 0–0,5 için turuncu, 0'ın altı için kırmızıdır (`cv_boyce_curves`, `sdm_core.py:6998-7015`; `tabs/tab_evaluation.py:791-847`). Çağrılan işlev, skor tablosundaki Boyce değerini üreten çağrının aynısıdır; panel başlığındaki CBI tablodaki değerle örtüşür.

Ne zaman, nasıl ayarlanır. İyi bir modelde P/E eğrisi uygunluk arttıkça tekdüze yükselir ve düşük uygunlukta 1'in altında, yüksek uygunlukta 1'in üstündedir (Hirzel vd. 2006). Eğrinin tepeden sonra düşmesi, modelin en yüksek değerleri gereğinden geniş alana verdiğini; düz bir eğri ayırt etmenin zayıf olduğunu gösterir.

Sonuca etkisi. Ekran ve `evaluation_boyce_<ALGO>.png`. Örnek koşuda GLM'nin kat CBI değerleri 0,274 (PA1 RUN2) ile 0,943 (PA1 RUN3) arasında değişir; RUN1 ve RUN2 panellerinde P/E eğrisi orta uygunlukta tepe yapıp düşer.

Dikkat. Pencere genişliği tahmin aralığının onda biridir (ecospat tanımı); çok dar tahmin aralıklı modellerde (ör. BIOCLIM) pencereler birbirine çok yakın düşer. Kat başına varlık sayısı azsa eğri basamaklıdır; CBI'yi tek katta değil katların ortalamasında okuyun.

7.9 Sınıflandırma eşikleri (Classification Thresholds)



Şekil 7.7. Classification Thresholds alt sekmesi: eşik kurallarının açıklamaları ve her tam veri modeli için on bir kuralın eşiği (üst satır) ile o eşikteki kat dışı TSS (alt satır); son sütun haritada kullanılan eşik.

Şekil 7.7. Classification Thresholds alt sekmesi: eşik kurallarının açıklamaları ve her tam veri modeli için on bir kuralın eşiği (üst satır) ile o eşikteki kat dışı TSS (alt satır); son sütun haritada kullanılan eşik.

Arayüzdeki ad	Ne yapar	Kod
`Threshold Methods` kutusu	On bir kuralın kısa tanımları	`tabs/tab_evaluation.py:1132-1160`
Tablo	Satır: tam veri modeli (algoritma × PA seti); sütunlar: 11 kural ve `Used (current)`	`tabs/tab_evaluation.py:1162-1180`
`↻ Compute All Thresholds`	Tabloyu yeniden hesaplar	`tabs/tab_evaluation.py:1183-1185`
`💾 Save Threshold Table (CSV)`	`figures/classification_thresholds.csv` yazar	`tabs/tab_evaluation.py:1316-1342`

Arayüzdeki ad	Ne yapar	Kod
💾 Save Scores (CSV)	evaluation_scores.csv yazar (§7.10)	tabs/tab_evaluation.py :1188-1194

Ne işe yarar. Her tam veri modeli için on bir kuralın her biriyle (Bölüm 6, §6.5.1) bir **harita eşiği** bulunur ve hücrede iki satır yazılır: eşik ve bu eşiğin kat dışı tahminlerde verdiği TSS'nin katlar üzerinden ortalaması (tabs/tab_evaluation.py:1200-1314; sdm_core.py:7018-7037). İpucu ayrıca o eşikte kat dışı duyarlılık ve özgüllüğü, eğitim verisindeki ölçüt değerini ve "pixel >= t -> Present (1)" kuralını gösterir. Used (current) sütunu haritada fiilen kullanılan eşiktir. CSV her kural için eşik ile _holdout_Sens, _holdout_Spec, _holdout_TSS sütunlarını, kullanılan eşiği, kuralın adını ve kat sayısını içerir.

Ne zaman, nasıl ayarlanır. Eşik kuralı seçimini desteklemek için: hangi kuralın kat dışı TSS'yi koruyarak daha geniş ya da daha dar bir harita verdiğini görürsünüz. sensitivity sütunu her zaman 0,90 hedefiyle hesaplanır (yöntem açıklaması).

Sonuca etkisi. Bu tablo haritaları değiştirmez; kural ③ Models'ta değiştirilip modeller yeniden çalıştırılınca haritalar değişir.

Dikkat. Bu tablodaki eşikler ile haritada kullanılan eşik farklı yollarla bulunur:

- Tablodaki kural sütunları, tam veri modelinin **tüm PA setlerinin birleşik verisindeki** kendi (eğitimde görülmüş) tahminlerinden bulunur (tabs/tab_evaluation.py:1217-1259).
- Used (current) ise ③ Models'ta CV katlarında, seçilen eşik kaynağıyla (varsayılan train_cv) bulunan eşiklerin ortalamasıdır (sdm_core.py:5852-5853).

Bu nedenle max_tss sütunu ile Used (current) aynı kural için bile farklıdır. Örnek koşuda GAM PA1 için max_tss sütunu 0,2562 (kat dışı TSS 0,59), kullanılan eşik 0,0605'tir (kat dışı TSS 0,64). Eğitim verisini ezberleyen algoritmalarda fark büyür: RF PA1 için max_tss sütunu 0,4532 (kat dışı TSS 0,34), kullanılan eşik 0,1540'tır (kat dışı TSS 0,63). Kural karşılaştırması için alt satırdaki kat dışı TSS'ye bakın; eşik değerlerini haritanın eşiği olarak okumayın.

7.10 Alt düğmeler: yenileme, kaydetme

Arayüzdeki ad	Ne yapar	Kod
↻ Refresh All Charts	Açılır listeleri günceller ve yedi alt sekmenin tamamını yeniden hesaplar	tabs/tab_evaluation.py :851-872
💾 Save All Charts to Output Folder	Bütün şekilleri ve skor tablosunu figures/ klasörüne yazar; iptal edilebilir	tabs/tab_evaluation.py :939-1058
💾 Save Scores (CSV)	evaluation_scores.csv dosyasını çıktı kökene ve figures/ kopyasına yazar	tabs/tab_evaluation.py :1064-1108

Ne işe yarar. Save All Charts … sırayla skor, ROC, omisyon/PR/kalibrasyon ve önem şekillerini; ardından CV modeli olan her algoritma için tepki eğrisi (o anda seçili Method: ile) ve Boyce şeklini; en sonda skor tablosunu yazar (tabs/tab_evaluation.py:993-1030). Her şekil 300 dpi PNG olarak bir kez kaydedilir (tabs/tab_evaluation.py:923-937). Ekranda bir şekil çizildiğinde de aynı adla kaydedilir; bu yüzden düğmeye basılmadan figures/ klasöründe yalnız görüntülenen algoritmanın tepki ve Boyce şekilleri bulunur (örnek koşuda yalnız evaluation_response_GLM.png ve evaluation_boyce_GLM.png).

Dikkat. evaluation_scores.csv ③ Models bitince, sekme ilk açıldığında ve topluluk yeniden kurulduğunda kendiliğinden yazılır (tabs/tab_evaluation.py:1071-1081; tabs/tab_models.py:1840-1845); Save Scores

(CSV) yalnız elle yinelemek içindir. Çıktı klasörü yoksa şekil kaydedilemez ve günlüğe "Output directory not set — cannot save figure." yazılır. Kayıt sonrası bilgi penceresi figures/response_curves/ alt klasöründen söz eder; böyle bir klasör oluşturulmaz, algoritma başına tek birleşik şekil yazılır.

7.11 Çıktılar ve sonraki adıma geçenler

<çıktı>/figures/ klasöründe:

- evaluation_model_scores.png, evaluation_roc_curves.png, evaluation_omission_pr_calibration.png, evaluation_variable_importance.png
- evaluation_response_<ALGO>.png, evaluation_boyce_<ALGO>.png
- evaluation_omission_pr_calibration.csv, classification_thresholds.csv, evaluation_scores.csv (kopya)

Çıktı kökünde evaluation_scores.csv. Bu sekme modelleri, eşikleri ve haritaları değiştirmez; ⑧ Report aynı skor tablosunu ve şekilleri kullanır (Bölüm 11).

7.12 Uzman soruları

S1. Bu sekmedeki ROC, omisyon ve Boyce eğrileri bağımsız veriden mi? Model eğitim verisinde mi sınanıyor? Cevap: Eğriler yalnız ③ Models'ta kaydedilen CV hold-out tahminlerinden çizilir (modeler.cv_holdout; sdm_core.py:5547-5556); yeni bir bölme ya da eğitim satırlarında tahmin yapılmaz (sdm_core.py:6926-6932). Bağımsızlık derecesi ③ Models'taki CV tasarımına bağlıdır: mekânsal bloklarda testler eğitimden coğrafi olarak ayrıdır, random tasarımında değildir. İstisnalar tepki eğrileri ve Classification Thresholds tablosunun eşik sütunlarıdır.

S2. Topluluk (EMwmean) için verilen AUC topluluk haritasından mı hesaplanıyor? Cevap: Hayır. Topluluk satırları ve ROC panelleri, her katta o katı görmemiş CV modellerinin topluluk kuralıyla birleştirilmesi ve yalnız o katta skorlanmasıyla üretilir; üye seçimi ve ağırlıklar diğer katların skorlarından gelir (sdm_core.py:6516-6696). Skorlar alt sekmesinin notu bunu açıkça yazar: "… they are NOT computed from the ensemble map." (tabs/tab_evaluation.py:269-274).

S3. TSS kaç eşikte ve hangi veride hesaplanıyor; ecospat ya da biomod2 ile karşılaştırılabilir mi? Cevap: TSS, seçilen kuralın (varsayılan max_tss) eğitim satırlarının iç kat dışı tahminlerinde bulduğu eşikte, test katında hesaplanır (sdm_core.py:1193-1217). biomod2 ve ecospat.max.tss eşiği genellikle değerlendirme verisinin kendisinde en iyiler; bu "kâhin" eşik TSS'yi yükseltir. ③ Models'ta Test-fold threshold: = test seçilirse HABITUS da aynı yolu izler ve sayılar karşılaştırılabilir olur; ancak bu seçim iyimserdir (Bölüm 6, §6.5.2).

S4. Boyce indeksi ecospat ile birebir aynı mı? Cevap: Varsayılan ecospat formülü ecospat.boyce (ecospat 4.1.4) işlevinin satır satır aktarımıdır: pencere genişliği tahmin aralığının 1/10'u, 101 konum, kapalı pencereler, ardışık yinelenen P/E değerlerinin atılması ve Spearman korelasyonu (sdm_core.py:475-543). Fark, fit kümesinin seçimindedir: HABITUS değerlendirme katının tüm satırlarını (test varlıkları ve test yalancı yoklukları) kullanır (sdm_core.py:605-610); ecospat'ta fit çoğu uygulamada tüm çalışma alanının raster tahminleridir. Bu nedenle değerler aynı tahminlerle birebir, tam harita ile yaklaşık örtüşür.

S5. Kalibrasyon eğimi nasıl hesaplanıyor ve yalancı yokluklu veride ne anlama geliyor? Cevap: Gözlemin, tahminin logit'i üzerine (pratikte cezasız, C = 1e12) lojistik regresyonunun eğimidir; tahminler $[10^{-6}, 1-10^{-6}]$ aralığına kırpılır, en az 10 satır ve iki sınıf gerekir (sdm_core.py:989-1013; Pearce ve Ferrier 2000). Yalancı yokluklu veride olasılığın ölçeği PA:varlık oranına bağlıdır; eğim, gerçek varlık olasılığının

değil, modelin kat dışı verideki aşırı ya da eksik güveninin göstergesidir. Eğimin 1'den çok küçük olması (aşırı uyum) algoritmalar arası karşılaştırmada anlamlıdır.

S6. OR_10p neden bazen 0 çıkıyor ve bu iyi bir sonuç mu? Cevap: Her zaman değil. Omisyon eşiği değerlendirme kümesinin en düşük tahminine eşit ya da altındaysa hiçbir varlık eşiğin altına düşemez; omisyon yapı gereği 0'dır. Bu durum `OR_10p_degenerate` = 1 ile işaretlenir; katların en az yarısında olursa algoritma için uyarı yazılır (`sdm_core.py:1100-1114`, `5787-5816`). Bu genellikle BIOCLIM, sığ CART ve güçlü düzenlenmiş GLMNET'te görülür; bu algoritmaların omisyonu diğerleriyle karşılaştırılmamalıdır.

S7. Değişken önemi hangi yöntemle hesaplanıyor; MaxEnt'in yüzde katkısı gibi mi? Cevap: Hayır, permütasyon önemidir: test katında bir öngörücü karıştırılır ve AUC'deki düşüş ölçülür (`sdm_core.py:5564-5636`). MaxEnt'in "percent contribution" ve "permutation importance" çıktıları yüzdeye normalleştirilir ve eğitim verisinde hesaplanır; HABITUS normalleştirmez ve test katında hesaplar. scikit-learn modellerinde sabit tohum nedeniyle `Variable importance reps:` tekrarları aynı sonucu verir; etkin tekrar `n_repeats` = 3'tür.

S8. Tepki eğrileri hangi modellerden ve hangi veriden üretiliyor? Cevap: Algoritmanın tüm kurulan modellerinden (her PA setinin CV koşuları ve tam veri modeli); kalın çizgi bunların ortalaması, gölge standart sapmasıdır (`sdm_core.py:8461-8506`). Değişkenin aralığı ve medyan satır, tüm PA setlerinin varlık ve yalancı yokluk satırlarının birleşiminden alınır; aralık 5.–95. yüzdeliktir, uçlarda dışdeğerleme yoktur (`sdm_core.py:8405-8457`). biomod2'nin `bm_PlotResponseCurves` işlevindeki varsayılan yaklaşım da diğer değişkenleri sabit bir değerde (medyan) tutmaktır; değişken aralığı ve özetleme ayrıntıları iki programda aynı olmadığından eğriler biçimce benzer, sayıca birebir olmayabilir.

S9. Marginal, PDP ve ICE ne zaman farklı sonuç verir? Cevap: Öngörücüler arasında etkileşim ya da güçlü ilişki varsa. Marginal yalnız medyan satırının çevresini gösterir; PDP gerçek veri satırları üzerinden ortalar (Friedman 2001); ICE her satırın eğrisini ayrı gösterir ve PDP'nin ortalamayla gizlediği heterojenliği ortaya çıkarır (Goldstein vd. 2015). Doğrusal toplamsal bir modelde (ör. etkileşimsiz GLM) üçü aynı biçimi verir, düzeyleri farklı olabilir; ağaç modellerinde ve `interactions` terimli GLM'de ayrışırlar.

S10. Classification Thresholds tablosundaki `max_tss` neden haritadaki eşikle aynı değil? Cevap: Tablo eşiği, tam veri modelinin tüm PA setlerinin birleşik verisindeki (eğitimde görülmüş) tahminlerinden bulur (`tabs/tab_evaluation.py:1217-1259`); haritadaki eşik ise CV katlarında eşik kaynağına göre bulunan eşiklerin ortalamasıdır (`sdm_core.py:5852-5853`). Örnek koşuda RF PA1 için tablo 0,4532, harita 0,1540'tır; RF'nin eğitim tahminleri varlıklarda 1'e yakın olduğundan tablo eşiği yüksek çıkar ve kat dışı TSS 0,34'e düşer, haritanın eşiği 0,63 verir. Tablonun güvenilir kısmı alt satırdaki kat dışı TSS'dir.

S11. Algoritmalar arasındaki fark anlamlı mı? Bir test sunuluyor mu? Cevap: Hayır, istatistiksel bir karşılaştırma testi yoktur. Program katlar ve PA setleri üzerinden ortalama, SD ve aralığı verir (`sdm_core.py:5883-5919`) ve grafikler dağılımı gösterir. Katlar aynı veriden türediği için bağımsız değildir; basit bir t testi de geçerli olmaz. Örnek koşuda aynı algoritmanın katları arasındaki fark (GLM PA1 ROC 0,790–0,941) algoritmalar arasındaki farktan büyüktür; bu durumda sıralama yapmak yerine birden çok algoritmayı topluluğa almak daha savunulabilir.

S12. ESM ya da CV olmadan çalıştırılan bir algoritma bu sekmede nasıl görünür? Cevap: Kat dışı tahmini olmayan algoritmalar ROC, omisyon ve Boyce şekillerinde "Not shown" notuyla listelenir (`sdm_core.py:6872-6876`, `6974-6977`); `Classification Thresholds` hücreleri `TSS n/a` gösterir. ESM'nin hold-out tahmini kaydedilir ve normal biçimde görünür; yalnız iç kat dışı eğitim tahmini üretmediği için eşik kaynağı farklı etiketlenebilir (Bölüm 6, §6.5.2).

7.13 Sık hatalar ve tuzaklar

- `Model Scores` grafiğindeki dağılımı katlar arası yayılım sanmak: grafik `Full` (CV ortalaması) ve topluluk satırlarını da içerir.
- TSS'si 0,3'ün altındaki modeli grafikte görememek: eksen 0,3'ten başlar; değeri tabloda ya da CSV'de okuyun.
- `Classification Thresholds` tablosundaki eşik değerlerini haritanın eşiği sanmak; haritanın eşiği `Used (current)` sütunudur.
- Kalibrasyon ve kesinlik değerlerini gerçek varlık olasılığı gibi yorumlamak; ölçek PA:varlık oranına bağlıdır.
- Tepki eğrisinin gölgesini güven aralığı olarak raporlamak; yinelemeler arası SD'dir.
- `Save All Charts …` kullanmadan `figures/` klasöründe her algoritmanın tepki ve Boyce şekillerini beklemek.
- `Variable importance reps:` = 0 ile çalıştırıp önem grafiğinin neden boş olduğunu aramak.
- Tepki eğrilerinde kırpma (`Clamp`) uygulandığını varsaymak.

7.14 Kaynaklar

Allouche, O., Tsoar, A., Kadmon, R., 2006. Assessing the accuracy of species distribution models: prevalence, kappa and the true skill statistic (TSS). Journal of Applied Ecology 43, 1223–1232. https://doi.org/10.1111/j.1365-2664.2006.01214.x

Fielding, A.H., Bell, J.F., 1997. A review of methods for the assessment of prediction errors in conservation presence/absence models. Environmental Conservation 24, 38–49. https://doi.org/10.1017/S0376892997000088

Friedman, J.H., 2001. Greedy function approximation: A gradient boosting machine. The Annals of Statistics 29, 1189–1232. https://doi.org/10.1214/aos/1013203451

Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E., 2015. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. Journal of Computational and Graphical Statistics 24, 44–65. https://doi.org/10.1080/10618600.2014.907095

Hirzel, A.H., Le Lay, G., Helfer, V., Randin, C., Guisan, A., 2006. Evaluating the ability of habitat suitability models to predict species presences. Ecological Modelling 199, 142–152. https://doi.org/10.1016/j.ecolmodel.2006.05.017

Li, W., Guo, Q., 2013. How to assess the prediction accuracy of species presence–absence models without absence data? Ecography 36, 788–799. https://doi.org/10.1111/j.1600-0587.2013.07585.x

Pearce, J., Ferrier, S., 2000. Evaluating the predictive performance of habitat models developed using logistic regression. Ecological Modelling 133, 225–245. https://doi.org/10.1016/S0304-3800(00)00322-7

8. Gelecek senaryoları (⑤ Future Scenarios)

Sekme: ⑤ Future Scenarios · Kod: `tabs/tab_projection.py` (arayüz), `sdm_core.py` `Projector` (hesap), `sdm_novelty.py` (ExDet, MESS), `sdm_grid.py` (ızgara hizalama), `sdm_mask.py` (çalışma sınırına kırpma)

8.1 Bu sekme ne yapar

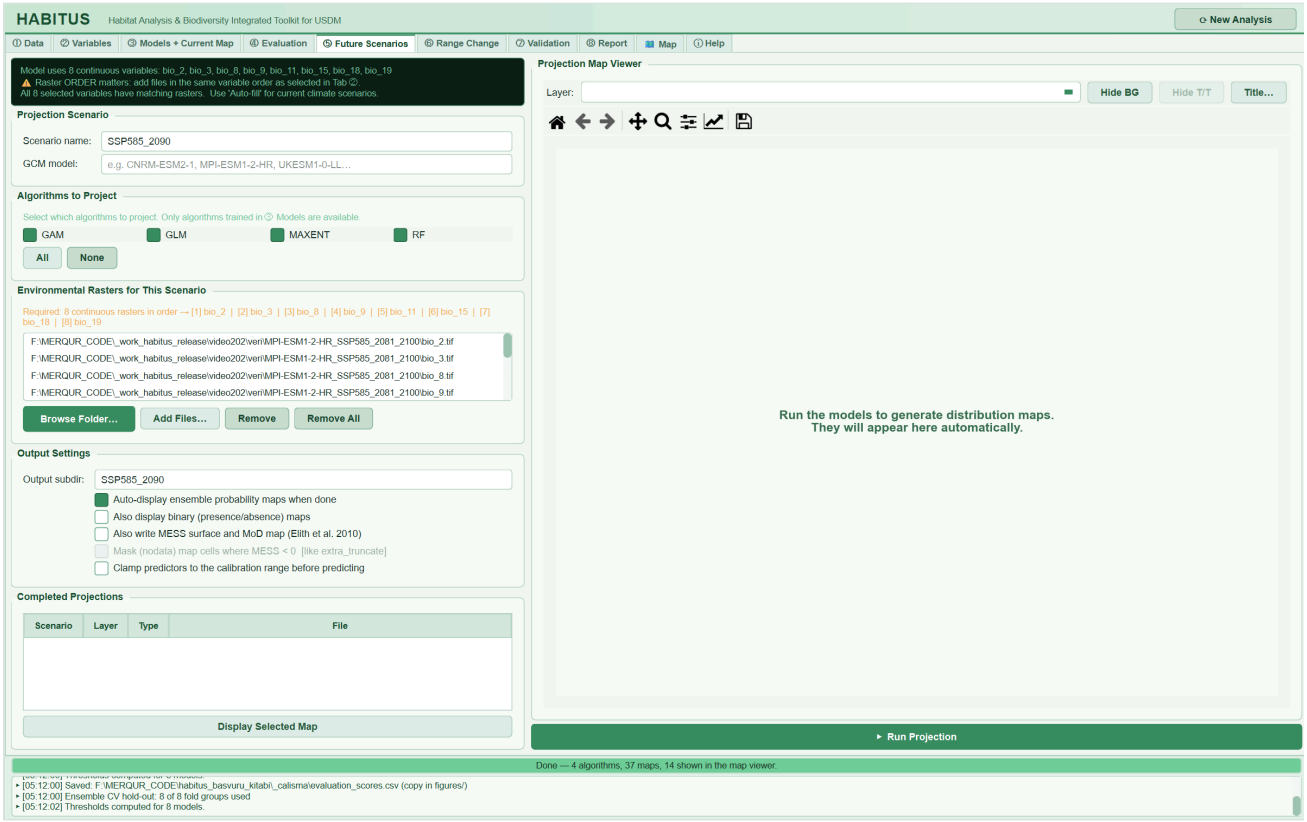
⑤ Future Scenarios sekmesi, ③ Models'ta kurulan modelleri başka bir ortam katmanı yığınına uygular; buna **projeksiyon** (aktarım) denir. En yaygın kullanım, bugünkü iklimle kurulmuş modeli bir iklim modelinin (GCM, genel dolaşım modeli) gelecek dönem katmanlarına uygulamaktır. Aynı sekme, modeli başka bir bölgeye ya da geçmiş bir döneme aktarmak için de kullanılır: program yalnızca "eğitimdeki değişkenlerle aynı adlı ve aynı sırada sürekli raster" bekler.

Sekme ③ Models tamamlanınca açılır (`tabs/tab_models.py:1849`). Bugünkü iklimin haritası ③'te zaten üretilmiştir (çıktı klasörünün `current\` alt klasörü, `tabs/tab_models.py:1719`); ⑤ bu haritanın gelecekteki karşılığını üretir.

- **Girdi:** her model değişkeni için bir sürekli raster (senaryo katmanları), projekte edilecek algoritmaların seçimi, isteğe bağlı MESS/MoD, MESS maskesi ve kırpma (clamping) seçenekleri.
- **Hesap:** her tam veri modeli (her algoritma × yalancı yokluk kümesi) yeni katmanlara uygulanır; algoritma içi ortalama ve yayılım haritaları, topluluk (ensemble) haritaları, ikili (0/1) haritalar ve ExDet yenilik yüzeyleri yazılır (`sdm_core.py:7733-8156`).
- **Çıktı:** senaryo alt klasöründe GeoTIFF dosyaları ve `<tür>_<senaryo>_projection_options.json` kaydı; `Completed Projections` tablosunda dosya listesi; sağdaki harita görüntüleyicide haritalar.
- **Açtığı sekme:** iş bitince ⑥ Range Change kilidi açılır (`tabs/tab_projection.py:672`).

8.2 Ekranın düzeni

Sekme iki bölmelidir: solda ayarlar, sağda harita görüntüleyici ve `▶ Run Projection` düğmesi (`tabs/tab_projection.py:98`, `ui_style.two_pane`).



Şekil 8.1. Future Scenarios sekmesi çalıştırmadan önce: üstte modelin beklediği sekiz değişkeni ve sıra uyarısını gösteren bilgi şeridi, solda Projection Scenario, Algorithms to Project, Environmental Rasters for This Scenario, Output Settings ve boş Completed Projections tablosu, sağda boş Projection Map Viewer.

Sol bölmedeki bölgeler yukarıdan aşağıya şunlardır:

Bölge	Arayüzdeki ad	Anlatıldığı yer
Bilgi şeridi	(başlıksız, koyu yeşil kutu)	§8.3
Senaryo adı ve GCM	`Projection Scenario`	§8.4
Algoritma seçimi	`Algorithms to Project`	§8.5
Raster listesi	`Environmental Rasters for This Scenario`	§8.6
Çıktı ve yöntem seçenekleri	`Output Settings`	§8.7, §8.8
Tamamlanan çıktılar	`Completed Projections`	§8.11
Harita	`Projection Map Viewer` (sağ bölme)	§8.12

8.3 Bilgi şeridi

Sekmenin en üstündeki koyu kutu, sekme her gösterildiğinde yeniden yazılır (`tabs/tab_projection.py:308-345`). Üç bilgi verir:

1. Modelin kullandığı sürekli değişkenler (`Model uses 8 continuous variables: bio_2, bio_3, …`).
2. Sıra uyarısı: `⚠ Raster ORDER matters: add files in the same variable order as selected in Tab ②.`
3. Eğitim rasterlarının değişkenlerle eşleşme durumu (`All 8 selected variables have matching rasters.` ya da `Warning: only k/n variables matched …`). Kategorik değişken varsa, bunların geleceğe taşınmadığını, bugünkü katmanın yeniden kullanıldığını söyleyen bir satır eklenir.

Hemen altındaki turuncu satır (`Required: 8 continuous rasters in order → [1] bio_2 | [2] bio_3 | …`) eklenecek dosyaların sırasını numaralı olarak gösterir (`tabs/tab_projection.py:340-343`).

> **Not:** Şeritte geçen `Use 'Auto-fill' for current climate scenarios.` ifadesine karşılık gelen bir düğme bu sekmede yoktur. Bugünkü iklim haritası ③ Models'ta kendiliğinden üretilir; bugünkü iklimi başka ayarlarla (ör. MESS açık) yeniden projekte etmek isteyen kullanıcı eğitim rasterlarının klasörünü `Browse Folder…` ile seçer (§8.6.2).

8.4 Senaryo adı ve GCM (Projection Scenario)

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Scenario name:`	Projeksiyonun adı; dosya adlarına, alt klasöre, rapora ve ⑥'daki listeye girer	`2050_SSP245`	serbest metin; boşsa `current`	`tabs/tab_projection.py:115`, `:505`
`GCM model:`	Harita başlığına yazılacak iklim modeli adı	boş	serbest metin	`tabs/tab_projection.py:119`, `:31-44`

8.4.1 Senaryo adı

- **Ne işe yarar.** Senaryo adı (`proj_name`) her çıktı dosyasının adına girer: `<tür>_<senaryo>_<katman>_prob.tif` (`sdm_core.py:7895`). `Output subdir:` boşsa çıktı klasörünün adı da budur (`tabs/tab_projection.py:544`). Ad, oturum durumunda (`state["projections"][ad]`) anahtar olarak tutulur; ⑥ Range Change listeleri, rapordaki yenilik tablosu ve çok türlü toplu çalıştırma bu anahtarla çalışır (`tabs/tab_projection.py:580`, `:636`).
- **Ne zaman, nasıl ayarlanır.** Ad, dönemi ve senaryoyu birlikte söylemelidir: `SSP585_2090`, `SSP245_2050_MPI`. Birden çok GCM kullanılıyorsa GCM'yi de ada katın; aksi halde iki GCM'nin çıktıları aynı klasöre ve aynı dosya adlarına yazılır.
- **Sonuca etkisi.** Hesaba etkisi yoktur; dosyaların ve kayıtların düzenini belirler.
- **Dikkat.** Varsayılan `2050_SSP245` metni değiştirilmeden bir SSP5-8.5 2090 katmanı projekte edilirse dosyalar yanlış adla yazılır; program adla içerik arasında denetim yapmaz. Aynı adla ikinci kez çalıştırmak aynı dosyaların üzerine yazar ve tablodaki satırları yineler. Adı `current` yapmak, ③'ün ürettiği bugünkü haritaların (aynı `current\` klasörü) ve oturumdaki `current` kaydının üzerine yazar.

8.4.2 GCM adı

- **Ne işe yarar.** Yalnızca görüntüleyicideki harita başlığına eklenir (`Future Distribution · MPI-ESM1-2-HR · SSP585_2090 · …`, `tabs/tab_projection.py:690-691`). Dosya adlarına, rapora ve ODMAP'a girmez.
- **Dikkat.** GCM bilgisinin rapor ve dosyalarda görünmesi isteniyorsa senaryo adına yazılmalıdır.

8.5 Projekte edilecek algoritmalar (Algorithms to Project)

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
algoritma kutuları (ör. `GAM`, `GLM`, `MAXENT`, `RF`)	Hangi algoritmanın modellerinin projekte edileceği	hepsi işaretli	③'te eğitilmiş algoritmalar	`tabs/tab_projection.py:288-306`
`All` / `None`	Hepsini işaretler / hiçbirini işaretlemez	—	—	`tabs/tab_projection.py:142-145`

- **Ne işe yarar.** Kutular sekme her açıldığında ③'te eğitilen modellerden (`modeler.fitted_models`) dörderli ızgara olarak kurulur. Seçilen algoritmaların **tam veri** modelleri ("Full", tüm veriyle yeniden kurulan model; her yalancı yokluk kümesi için bir tane) projekte edilir (`sdm_core.py:7850-7855`). ③'te `Also project the cross-validation models (more replicates, slower)` işaretlendiyse çapraz doğrulama modelleri de projekte edilir (`tabs/tab_models.py:1537`, `:1546`).
- **Ne zaman, nasıl ayarlanır.** Bugünkü harita ile gelecek haritası karşılaştırılacaksa (⑥ Range Change) iki haritanın **aynı algoritma kümesiyle** üretilmesi gerekir; bu yüzden ③'te kullanılan tüm algoritmaları işaretli bırakmak varsayılan ve önerilen seçimdir. Bir algoritmanın aktarımda kötü davrandığı biliniyorsa (ör. aralık dışında sıçrayan bir GAM) onu dışarıda bırakmak bir duyarlılık denemesi olarak yapılabilir.
- **Sonuca etkisi.** Seçilmeyen algoritma topluluk haritasına girmez; topluluk ağırlıkları yalnızca projekte edilen modeller üzerinden hesaplanır (`sdm_core.py:8028-8035`). Örnek koşuda dört algoritma seçildiğinde 8 model (4 algoritma × 2 yalancı yokluk kümesi) projekte edilmiş ve 37 raster yazılmıştır.
- **Dikkat.** Hiçbir kutu işaretli değilse `Select at least one algorithm to project.` iletisi çıkar. Seçilen algoritmaların hiçbiri ③'teki topluluk ölçütünü (`minimum score`) geçmiyorsa topluluk haritası yazılmaz ve günlüğe `WARNING: No full-data model reached the minimum …` uyarısı düşer (`sdm_core.py:8036-8054`).

8.6 Raster listesi ve sıra kuralı (Environmental Rasters for This Scenario)

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
raster listesi	Senaryonun sürekli katmanları, değişken sırasıyla	boş	çoklu seçim	`tabs/tab_projection.py:159-162`
`Browse Folder…`	Klasördeki rasterları değişken adlarıyla eşleyip listeyi doldurur	—	`.tif .tiff .asc .img .nc .vrt`	`tabs/tab_projection.py:166-170`, `:382-449`
`Add Files…`	Seçilen dosyaları listenin sonuna ekler	—	aynı uzantılar	`tabs/tab_projection.py:172`, `:451-457`
`Remove`	Seçili satırları siler	—	—	`tabs/tab_projection.py:174`, `:459-461`
`Remove All`	Listeyi ve **tüm projeksiyon sonuçlarını** siler, ⑥'yı kilitler	—	onay sorulur	`tabs/tab_projection.py:177-180`, `:463-489`

8.6.1 Sıra kuralı

- **Ne işe yarar.** Program sürekli rasterları **konumla** eşler: listedeki i'inci dosya, modelin i'inci sürekli değişkeni olarak okunur (`sdm_core.py:7335`, `:7407-7412`). Tek bir istisna vardır: listedeki dosyaların adları (temizlenmiş haliyle, ör. `bio_ (12)` → `bio_12`) model değişkenleriyle birebir aynı kümeyi oluşturuyorsa, sıra farklı da olsa dosyalar adla eşlenir (`sdm_core.py:7375-7379`). Bu, `bio_11, bio_15, bio_18, bio_19, bio_2, …` gibi alfabetik sıralı listelerin iki katmanı yer değiştirmesini önler.
- **Ne zaman, nasıl ayarlanır.** Gelecek katmanlarını eğitim katmanlarıyla **aynı dosya adlarıyla** (`bio_2.tif`, `bio_3.tif` …) saklamak en güvenli yoldur; o zaman ad eşlemesi devreye girer ve sıra hatası olanaksızlaşır. Dosya adları farklıysa (`wc2.1_2.5m_bioc_MPI_ssp585_2081-2100_02.tif` gibi), liste turuncu satırdaki sırayla birebir kurulmalıdır.
- **Sonuca etkisi.** Sıra hatası programı durdurmaz: iki değişken yer değiştirirse model, örneğin yağış mevsimselliğini sıcaklık sanır ve harita anlamsızlaşır. Bunu fark ettiren tek iz ExDet NT1 oranının olağandışı yüksekliğidir (§8.9).
- **Dikkat.** Dosya sayısı modelin sürekli değişken sayısına eşit olmalıdır; değilse `Raster count mismatch!` iletisi çıkar ve projeksiyon başlamaz (`tabs/tab_projection.py:525-534`, `sdm_core.py:7360-7366`). Kategorik değişkenler (arazi örtüsü, toprak sınıfı) bu listeye **eklenmez**: program onları ① Data'da yüklenen bugünkü kategorik rasterlardan alır ve gelecekte de aynı kabul eder (`sdm_core.py:7474-7483`, `:7414-7442`). Yani arazi örtüsünün değişmediği varsayılır.

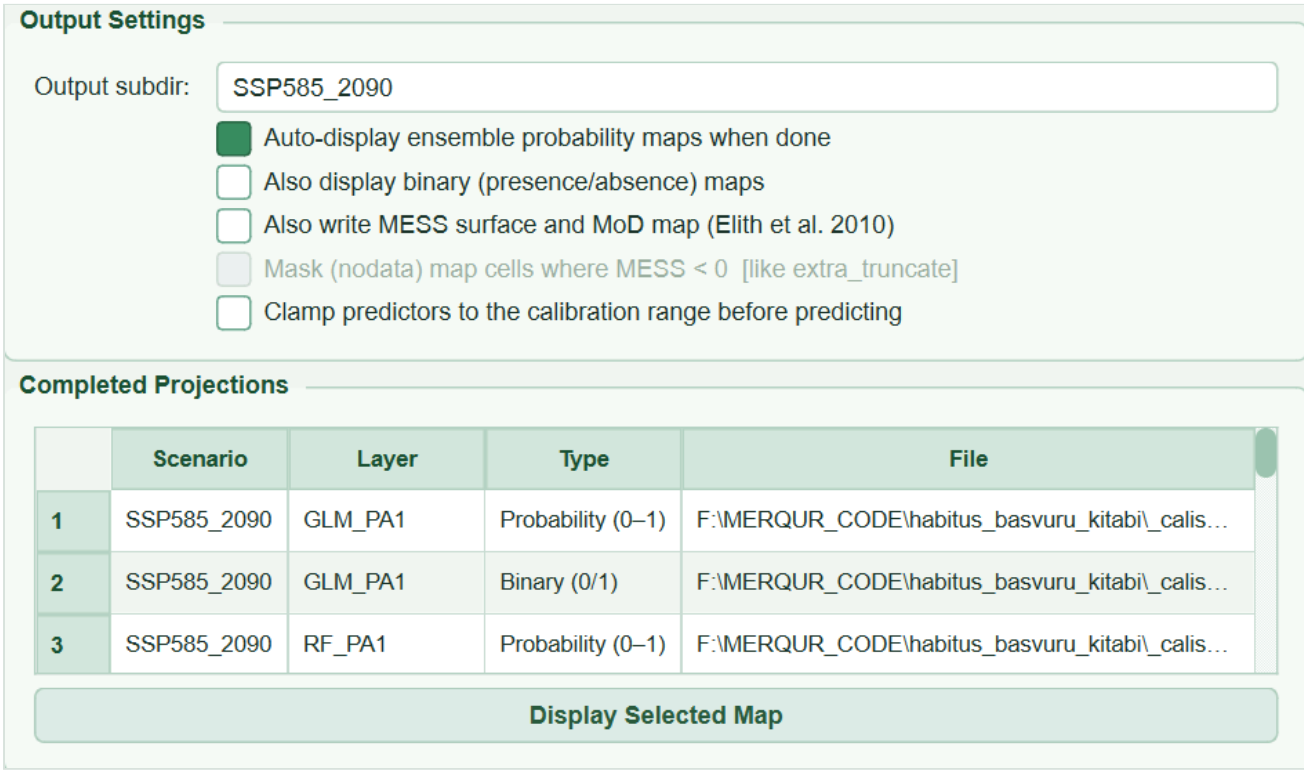
8.6.2 Klasörden ekleme (Browse Folder…)

- **Ne işe yarar.** Seçilen klasördeki raster dosyalarının adları küçük harfe çevrilip boşluk, parantez, tire ve nokta alt çizgiye indirgenir; her model değişkeni için sırasıyla (1) tam ad eşleşmesi, (2) dosya adının değişken adıyla başlaması, (3) değişken adının dosya adının içinde geçmesi aranır (`tabs/tab_projection.py:399-436`). Bulunanlar **değişken sırasıyla** listeye yazılır.
- **Ne zaman, nasıl ayarlanır.** Her senaryonun ayrı bir klasörde ve eğitimle aynı adlarla tutulduğu düzende tek tıkla doğru liste kurulur (örnek koşuda `MPI-ESM1-2-HR_SSP585_2081_2100\bio_2.tif …`).
- **Sonuca etkisi.** Doğru eşleşmede elle kurulan listeyle aynı sonucu verir.
- **Dikkat.** "Önek" ve "içerir" kuralları gevşektir: `bio_1` değişkeni için önce tam ad aranır, bulunamazsa `bio_10.tif`, `bio_11.tif` gibi `bio_1` ile başlayan ilk dosya alınabilir. Eşleşmeyen değişkenler `Unmatched Variables` penceresinde listelenir ve `Add Files…` ile eklenmesi istenir. Liste kurulduktan sonra turuncu satırla karşılaştırarak gözle denetleyin.

8.6.3 Dosya ekleme ve silme

`Add Files…` seçilen dosyaları listenin **sonuna** ekler; sıralamayı korumak kullanıcıya kalır. `Remove` yalnız listedeki satırları siler. `Remove All` ise onay sorduktan sonra listeyi temizler, oturumdaki tüm projeksiyon ve alan değişimi sonuçlarını sıfırlar, iki sekmenin haritalarını ve tablolarını boşaltır ve ⑥ Range Change'i yeniden kilitler (`tabs/tab_projection.py:475-489`). Diskteki dosyalar silinmez.

8.7 Çıktı ve görüntüleme ayarları (Output Settings)



Şekil 8.2. Projeksiyondan sonra sol bölmenin alt kısmı: Output Settings kutusunda örnek koşuda kullanılan seçenekler (yalnız Auto-display açık, MESS maskesi MESS kapalı olduğu için soluk) ve Completed Projections tablosunun ilk satırları.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Output subdir:`	Senaryo dosyalarının yazılacağı alt klasör	boş (senaryo adı kullanılır)	serbest metin	`tabs/tab_projection.py:197-198`, `:543-545`
`Auto-display ensemble probability maps when done`	İş bitince olasılık haritalarını görüntüleyiciye yükler	açık	açık / kapalı	`tabs/tab_projection.py:199-201`, `:669-670`
`Also display binary (presence/absence) maps`	İkili haritaları da yükler	kapalı	açık / kapalı	`tabs/tab_projection.py:202-203`, `:683-684`
`Also write MESS surface and MoD map (Elith et al. 2010)`	MESS ve MoD rasterlarını da yazar	kapalı	açık / kapalı	`tabs/tab_projection.py:204-214`
`Mask (nodata) map cells where MESS < 0 [like extra_truncate]`	MESS < 0 hücrelerini haritalarda nodata yapar	kapalı (MESS kapalıyken devre dışı)	açık / kapalı	`tabs/tab_projection.py:215-224`
`Clamp predictors to the calibration range before predicting`	Öngörücüleri tahminden önce eğitim aralığına kırpar	kapalı	açık / kapalı	`tabs/tab_projection.py:225-233`

8.7.1 Çıktı alt klasörü

- **Ne işe yarar.** Dosyalar `<çıktı klasörü>\<Output subdir>` altına yazılır; alan boşsa senaryo adı kullanılır (`tabs/tab_projection.py:544-545`).
- **Ne zaman, nasıl ayarlanır.** Çoğu durumda boş bırakılır. Aynı senaryonun farklı ayarlarla (ör. clamping açık/kapalı) iki çalıştırmasını karşılaştırmak için farklı alt klasör verilebilir; ancak dosya adları senaryo adından üretildiği için ⑥ ve rapor iki çalıştırmayı ancak senaryo adları da farklıysa ayırır.
- **Dikkat.** Rapor (⑧) gelecek haritalarının PNG'lerini bu alt klasörlerde arar (§11.5.7); senaryo PNG'lerini buraya kaydetmek raporun düzenini belirler.

8.7.2 Görüntüleme seçenekleri

`Auto-display …` açıkken iş bitince görüntüleyici temizlenir ve adı `_prob` ile biten **tüm** olasılık haritaları yüklenir: yalnız topluluk değil, tek modeller ve algoritma ortalamaları da (örnek koşuda 14 harita, `tabs/tab_projection.py:675-705`). `Also display binary …` açıksa `_bin` haritaları da eklenir. Bu iki seçenek hiçbir dosyayı değiştirmez; yalnızca görüntüleyicinin `Layer:` listesini doldurur.

8.8 Aktarım güvenliği seçenekleri: MESS, MESS maskesi, clamping

Bir model, eğitimde hiç görmediği ortam koşullarında da 0–1 arası bir değer üretir; bu değer görünüşte diğerleri kadar "kesin"dir. Kalibrasyon alanının ortam aralığı dar kaldıkça bu durum aktarımda daha sık görülür (Owens vd. 2013). HABITUS bu sorunu üç katmanda ele alır: ExDet her projeksiyonda kendiliğinden hesaplanır (§8.9); MESS isteğe bağlı ek tanıdır; MESS maskesi ve clamping ise haritanın kendisini değiştirir.

Üçü de aynı **kalibrasyon referans kümesini** kullanır: modellerin gerçekten kurulduğu satırlar, yani tüm yalancı yokluk kümelerindeki varlık ve yokluk noktalarının birleşimi (`sdm_core.py:7142-7171, :4929-4952`). Kategorik değişkenler referansa girmez.

8.8.1 MESS ve MoD (Also write MESS surface and MoD map)

- **Ne işe yarar.** Elith vd. (2010) tanımıyla her hücre ve her değişken için bir benzerlik değeri hesaplanır. Referans kümede değişken j için değeri p'ye eşit ya da küçük olanların yüzdesi F, en küçük değer $\min_j$, en büyük $\max_j$ olmak üzere (`sdm_novelty.py:377-449`):

$$s_j = \begin{cases} 100\,(p - \min_j)/(\max_j - \min_j) & F = 0 \\ 2F & 0 < F \le 50 \\ 2(100 - F) & 50 < F < 100 \\ 100\,(\max_j - p)/(\max_j - \min_j) & F = 100 \end{cases}$$

MESS, s_j'lerin en küçüğüdür; MoD (most dissimilar variable) bu en küçük değeri veren değişkenin sırasıdır. Yazılan dosyalar `<tür>_<senaryo>_MESS.tif` (float32) ve `<tür>_<senaryo>_MoD.tif` (int16; değişken adları raster etiketlerinde `CLASS_i` olarak saklanır) (`sdm_core.py:7263-7281`). - **Ne zaman, nasıl ayarlanır.** MESS, MaxEnt yazılımı ve dismo/flexsdm çıktılarıyla karşılaştırma gereken, hakemin "MESS haritası" istediği çalışmalarda açılır. Yöntem bakımından ExDet NT1 aynı bilgiyi zaten verir (§8.9). - **Sonuca etkisi.** Uygunluk haritalarını değiştirmez. Günlüğe `MESS: x% of projected cells have MESS < 0 …` satırı yazılır (`sdm_core.py:7286-7289`). Aynı referans küme ve aynı en küçük/en büyük değerler kullanıldığı için **MESS < 0 olan hücreler, ExDet NT1 < 0 olan hücrelerle aynıdır** (bir değer tam en büyük değere eşitse ikisi de 0 verir): örnek koşuda SSP585_2090 için NT1 < 0 oranı %55,4'tür; MESS açılsaydı aynı oran MESS < 0 için

raporlanacaktı. Tek fark, eğitimde sabit kalan bir değişkendir: NT1 onu yok sayar, MESS sabit değerden farklı her hücreye −100 verir (`sdm_novelty.py:216-225`, `:436-437`). - **Dikkat.** MESS tek değişkenlidir: tüm değişkenler tek tek aralık içindeyken birleşimin yeni olduğu hücreleri göremez; bunlar için ExDet NT2'ye bakılır. İpucu metni de bunu söyler: `MESS cannot see novel combinations of in-range values; ExDet NT2 can.`

8.8.2 MESS < 0 maskesi (Mask (nodata) map cells where MESS < 0)

- **Ne işe yarar.** Bütün çıktılar yazıldıktan sonra, MESS < 0 olan hücreler olasılık, ikili, algoritma ortalaması, algoritma yayılımı (`_sd`) ve topluluk yayılımı (`EMsd`, `EMcv`, `EMrange`) haritalarında nodata (−9999) yapılır (`sdm_core.py:7293-7325`, `:8112-8123`). ExDet ve MESS/MoD rasterları maskelenmez. flexsdm'deki `extra_truncate` işlevinin karşılığıdır (Velazco vd. 2022). Kutu, MESS kutusu işaretlenmeden etkinleşmez (`tabs/tab_projection.py:224`).
- **Ne zaman, nasıl ayarlanır.** Harita doğrudan bir karar için kullanılacaksa (koruma önceliği, alan hesabı) ve dışdeğerlenmiş (extrapolated) hücrelerin hiç yorumlanmaması isteniyorsa açılır. Bilimsel raporda çoğu zaman maskesiz harita ile NT1 haritası yan yana verilir ve maskeli hesap duyarlılık analizi olarak eklenir.
- **Sonuca etkisi.** Etkisi büyüktür. MESS < 0 ile NT1 < 0 aynı hücreler olduğu için örnek koşuda maske, SSP585_2090 haritasındaki hücrelerin %55,4'ünü nodata yapacaktı. ⑥ Range Change yalnız iki haritada da geçerli olan hücreleri sayar (`sdm_core.py:8292`); maskeli bir gelecek haritası, maskelenen hücreleri "kaybedilen" ya da "kazanılan" saymaz, onları hesaptan tümüyle çıkarır. Böylece bugünkü alan da yalnız maske dışında kalan bugünkü varlık hücrelerinden hesaplanır.
- **Dikkat.** MESS hesaplanamazsa maskeleme yapılmaz ve günlüğe `WARNING: MESS truncation was requested but MESS could not be computed; the maps of <senaryo> were NOT truncated.` yazılır (`sdm_core.py:8114-8119`). Yapılıp yapılmadığı `_projection_options.json` dosyasındaki `mess_truncate` ve `mess_truncated_files` alanlarından okunur.

8.8.3 Clamping (Clamp predictors to the calibration range before predicting)

- **Ne işe yarar.** Projeksiyondaki her sürekli değişken, tahminden önce referans kümenin en küçük ve en büyük değerine kırpılır (`np.clip` mantığı; `sdm_core.py:7808-7814`, `:4929-4960`). Tek modeller, algoritma ortalamaları ve topluluk bu kırpılmış değerlerle tahmin eder. ExDet ve MESS ise kırpılmamış ham değerlerle hesaplanır; yani yenilik haritaları gerçek dışdeğerlemeyi göstermeye devam eder.
- **Ne zaman, nasıl ayarlanır.** Esnek yanıt eğrisi kuran algoritmalar (GLM'nin ikinci dereceden terimleri, GAM eğrileri, MaxEnt menteşe/ürün özellikleri) aralık dışında yükselip alçalarak gerçekçi olmayan değerler üretebilir; ağaç tabanlı yöntemler (RF, GBM, BRT) ise aralık dışında zaten son değerde sabit kalır. Gelecek katmanlarında NT1 oranı yüksekse (örnek koşuda %55,4) clamping'li ve clamping'siz iki projeksiyon yapıp farkı raporlamak iyi bir uygulamadır. Bu, MaxEnt yazılımındaki "clamping" seçeneğinin ve biomod2'deki `do.clamping` / `clamping.mask` yaklaşımının mantığıdır.
- **Sonuca etkisi.** Aralık dışı hücrelerde uygunluk, aralığın kenarındaki değerin uygunluğuna sabitlenir. Ağaç tabanlı yöntemlerde etkisi küçüktür; GLM/GAM/MaxEnt'te dışdeğerlenmiş hücrelerde belirgin olabilir. ⑥'daki kazanım/kayıp oranları bu hücrelerden etkilenir.
- **Dikkat.** Clamping dışdeğerlemeyi ortadan kaldırmaz, yalnızca yanıtı sabitler; "aralık kenarındaki koşul aralık dışında da sürer" varsayımı yapılmış olur. ③ Models'taki clamping ayarı (`settings.models.clamp`) bugünkü haritayı yönetir; ⑤'teki kutu her senaryo için ayrıca seçilir ve `_projection_options.json` dosyasına yazılır. Karşılaştırılabilirlik için bugünkü ve gelecek haritalarında aynı seçimi kullanın.

8.9 ExDet yenilik yüzeyleri (her zaman yazılır)

Arayüzde bir kutusu yoktur: `Projector` her projeksiyonda ExDet'i çalıştırır (`write_novelty=True`, `sdm_core.py:7088`, `:7937-7943`). Yöntem Mesgaran vd. (2014) ExDet'idir ve iki tür yeniliği ayırır.

NT1 (tek değişkenli aralık yeniliği). Referans kümede değişken j'nin en küçüğü $\min_j$, en büyüğü $\max_j$ olmak üzere, hücre i için (`sdm_novelty.py:202-228`):

$$UD_{ij} = \frac{\min\left(P_{ij} - \min_j,\ \max_j - P_{ij},\ 0\right)}{\max_j - \min_j}, \qquad NT1_i = \sum_j UD_{ij}$$

Aralık içinde NT1 = 0, dışında negatiftir; mutlak değer büyüdükçe aralığın ne kadar dışında olunduğu artar.

NT2 (birleşim yeniliği). Yalnız NT1 = 0 olan hücrelerde, referans kümenin ortalaması ve kovaryansıyla Mahalanobis uzaklığının karesi D^2 hesaplanır ve referans kümedeki en büyük D^2'ye bölünür: $NT2_i = D^2_i / \max(D^2_{ref})$ (`sdm_novelty.py:249-290`). NT2 > 1, her değişkenin tek başına aralık içinde olduğu ama birleşimin eğitimde hiç görülmediği hücredir.

MIC (en etkili değişken). NT1 için en negatif UD_{ij}'yi veren değişken; NT2 için çıkarıldığında D^2'yi en çok düşüren değişken (Mesgaran vd. 2014, eş. 7; `sdm_novelty.py:231-235`, `:292-317`).

Yazılan dosyalar: `<tür>_<senaryo>_NT1.tif`, `_NT2.tif`, `_MIC_NT1.tif`, `_MIC_NT2.tif` (MIC değerleri değişken sırasıdır; adlar raster etiketlerinde `CLASS_i`; `sdm_core.py:7206-7233`). Günlüğe düz metinli bir özet yazılır (`sdm_novelty.py:640-692`). Örnek koşunun SSP585_2090 özeti:

Ölçü	current	SSP585_2090
NT1 < 0 (aralık dışı)	%23,4	%55,4
NT2 > 1 (yeni birleşim)	%0,0	%0,3
Her iki teste göre benzer	%76,6	%44,3
En etkili değişken, aralık yeniliği	bio_3 (%43), bio_15 (%24), bio_2 (%13)	bio_9 (%58), bio_3 (%17), bio_8 (%10)

NT1 oranı %25'i aşınca günlüğe `More than a quarter of the projection is outside the calibration range. Treat the suitability surface in those areas as an extrapolation, not a prediction.` uyarısı eklenir (`sdm_novelty.py:686-690`).

> **Dikkat:** Referans küme varlık ve yalancı yokluk noktalarıdır; bugünkü haritada bile hücrelerin %23,4'ünün NT1 < 0 çıkması bundandır: eğitim noktaları erişilebilir alan içinden (örnekte kayıtlar çevresinde 200 km) seçildiği için, çalışma alanının uzak köşeleri bugün de aralık dışındadır. Bu bir hata değil, haritanın o kısmının da dışdeğerleme olduğunu söyleyen bir bulgudur.
>
> **Not:** Kalibrasyon kovaryans matrisinin koşul sayısı 10^{12}'yi aşarsa (birbirinin neredeyse kopyası değişkenler) NT2 hesaplanmaz; neden günlükte ve raporda yazılır: `NT2 was not computed: the calibration covariance matrix is numerically singular …` (`sdm_novelty.py:62`, `:257-265`). NT1 bu durumda da yazılır.

8.10 Izgara hizalama, çalışma sınırına kırpma ve şeritli projeksiyon

Bu üç işlem kullanıcıdan ayar beklemez, ama sonucu doğrudan etkiler ve programın verdiği uyarıların anlamını bilmek gerekir.

8.10.1 Izgara hizalama

- **Ne işe yarar.** Her senaryo katmanı, eğitim ızgarasına (dönüşüm, satır-sütun sayısı, koordinat sistemi) **konumla** oturtulur (`sdm_grid.py:188-280`, `sdm_core.py:7486-7498`):
- ızgara aynıysa dosya olduğu gibi okunur;
- hücre boyu aynı ve kayma tam sayı hücre ise değerler interpolasyonsuz kaydırılır (`cell-shift`);
- başka her durumda `rasterio.warp.reproject` ile yeniden örneklenir: sürekli katmanlarda çift doğrusal (bilinear), kategoriklerde en yakın komşu (nearest).

Katmanın örtmediği eğitim hücreleri NaN olur ve sayılır; NaN'lı hücre projeksiyondan çıkar (bütün değişkenlerin sonlu olduğu hücreler tahmin edilir, `sdm_core.py:7799-7807`). - **Ne zaman, nasıl ayarlanır.** İndirilen GCM setleri eğitim katmanlarından bir hücre geniş ya da farklı kesilmiş olabilir; hizalama bunları güvenle bir araya getirir. En iyisi, gelecek katmanlarını eğitimle aynı çözünürlük ve kapsamda hazırlamaktır; o zaman hiçbir yeniden örnekleme yapılmaz. - **Sonuca etkisi.** Kaydırma ve özdeşlik değerleri değiştirmez; yeniden örnekleme (özellikle farklı çözünürlükten) değerleri yumuşatır. - **Dikkat.** Hizalanan her katman için günlüğe ve rapordaki `Processing Notices` bölümüne bir satır yazılır, örneğin `WARNING grid alignment: bio_2: 468x222 px grid realigned to 466x220 reference by position (cell-shift), offset -1 col / -1 row, 0 reference cell(s) not covered (0.0%)`. Bir katman eğitim ızgarasının %5'inden fazlasını örtmüyorsa ayrıca `… % of the reference grid is outside this layer's extent (> 5%); those cells are NaN and are left out of the projection. Check that the layer covers the study area.` uyarısı çıkar (`sdm_grid.py:40`, `:304-328`). Özet, `_projection_options.json` dosyasının `grid_alignment` alanında saklanır; örnek koşuda `n_realigned: 0` (sekiz katmanın hepsi eğitim ızgarasında).

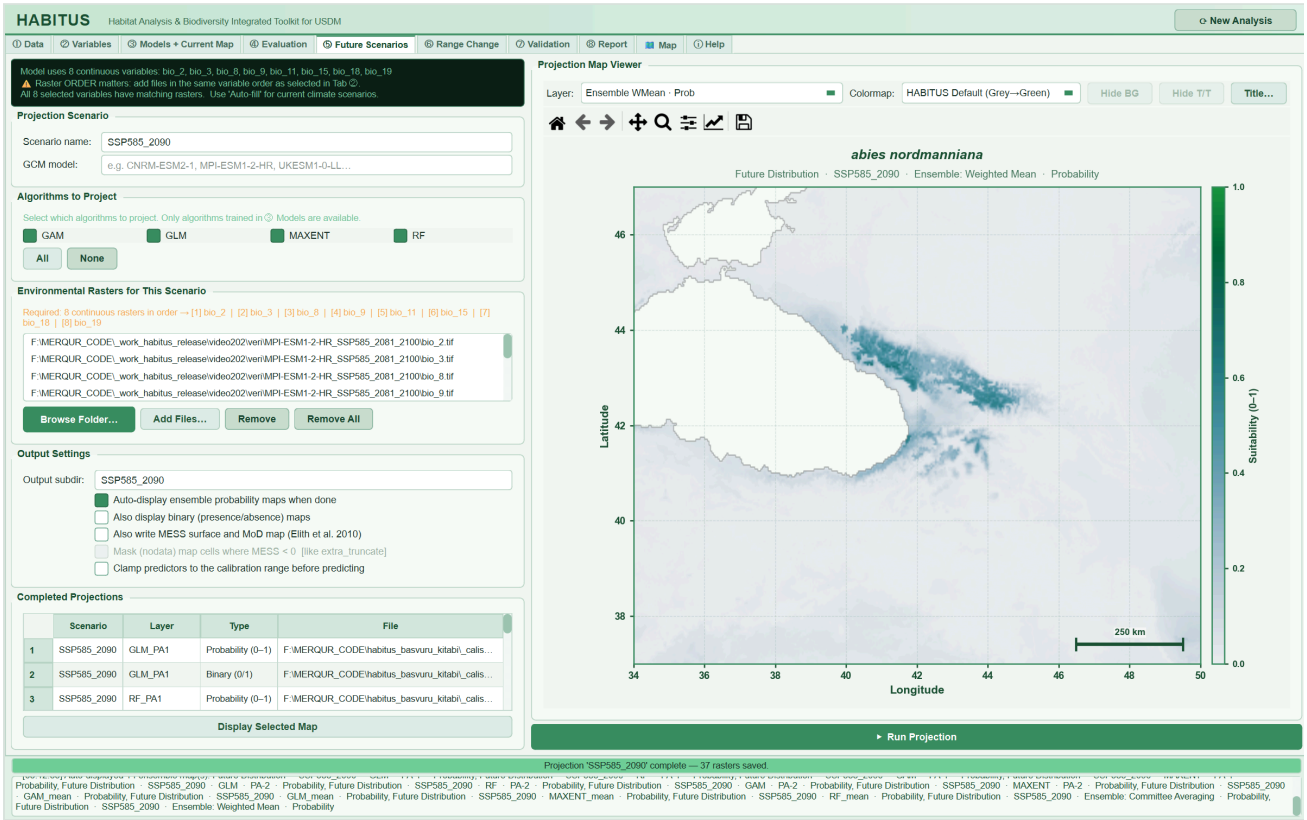
8.10.2 Çalışma sınırına kırpma

- **Ne işe yarar.** ① Data'da bir çalışma sınırı (study boundary) verildiyse, gelecek rasterları da aynı sınıra kırpılır ve **kırpılmış eğitim rasterlarının ızgarasına** yerleştirilir (`tabs/tab_projection.py:589-613`, `sdm_mask.py:304-418`). Kırpılmış kopyalar `<çıktı klasörü>\masked_inputs\<senaryo>\` altına özgün dosya adlarıyla yazılır; böylece ad eşlemesi bozulmaz. Listedeki dosyaların hepsi zaten eğitimde kullanılan (kırpılmış) dosyalarsa kırpma atlanır (`tabs/tab_projection.py:590-592`).
- **Sonuca etkisi.** Bugünkü ve gelecek haritaları hücre hücre aynı alanı kaplar; ⑥ benzerini benzerle karşılaştırır.
- **Dikkat.** Kırpma sırasında farklı hücre boyundaki katmanlar **en yakın komşu** ile eğitim ızgarasına taşınır ve günlüğe `Resampled to the training grid (nearest neighbour, another cell size): …` yazılır (`tabs/tab_projection.py:610-613`, `sdm_mask.py:273-301`). Sınır yokken aynı durumda sürekli katmanlar çift doğrusal yöntemle örneklenir (§8.10.1). Farklı çözünürlükteki gelecek katmanları kullanılacaksa, onları önceden eğitim çözünürlüğüne getirmek bu farkı ortadan kaldırır. Sınır katmanla örtüşmüyorsa ya da bir katmanın sınır içinde hiç verisi yoksa iş açık bir iletiyle durur (`sdm_mask.py:369-403`).

8.10.3 Şeritli projeksiyon (büyük raster)

- **Ne işe yarar.** Eğitim ızgarası 4 milyon hücreyi aşıyorsa (varsayılan şerit boyu 2 milyon hücre; ızgara bunun iki katından büyükse bölünür), raster satır şeritleri halinde projekte edilir ve şeritler aynı dosyalarda birleştirilir (`sdm_core.py:7514-7527`, `:7566-7682`). Her hücre tek başına tahmin edildiği için sonuç tek parça hesapla aynıdır; ExDet ve MESS özetleri şeritlerden sayımlar toplanarak elde edilir (`sdm_novelty.py:452-637`).
- **Ne zaman, nasıl ayarlanır.** Kullanıcı ayarı yoktur. Şerit boyu `HABITUS_PROJECTION_TILE_CELLS` ortam değişkeniyle değiştirilebilir (0 = hiç bölme). Bellek yetersizliği hatası alan kullanıcı bu değeri küçültebilir.
- **Sonuca etkisi.** Değerler aynıdır; süre biraz uzar, bellek kullanımı düşer. Günlüğe `Large raster (… cells): projecting in k strips of up to n rows to keep memory in check.` yazılır.
- **Dikkat.** Ara şerit dosyaları çıktı klasöründe `.habitus_strips_*` adlı geçici klasöre yazılır ve iş bitince silinir; çıktı diskinde son haritalar kadar boş yer gerekir. Eğitim ızgarasında olmayan katmanlar önce bir kez hizalanıp bu klasöre yazılır.

8.11 Çıktılar ve Completed Projections tablosu



Şekil 8.3. SSP585_2090 projeksiyonu tamamlandıktan sonra: Completed Projections tablosunda senaryo, katman, tür ve dosya yolu satırları; sağda ağırlıklı ortalama topluluk olasılık haritası (Ensemble WMean · Prob), alt şeritte 37 rasterın kaydedildiği iletisi.

Örnek koşuda `SSP585_2090\` klasörüne 37 raster ve bir JSON kaydı yazılmıştır:

Dosya (`<tür>_<senaryo>_…`)	İçerik	Kod
`<ALG>_PA<k>_prob.tif`, `_bin.tif`	Tek modelin olasılık ve ikili haritası (eşik: ③'te o model için bulunan eşik)	`sdm_core.py:7895-7916`
`<ALG>_mean_prob.tif`, `_mean_bin.tif`, `_sd.tif`	Algoritmanın kopyaları üzerinden ortalama, ikili ve standart sapma (③'te iki katmanlı topluluk açıksa)	`sdm_core.py:7964-8023`

Dosya (`<tür>_<senaryo>_…`)	İçerik	Kod
`EMwmean_prob.tif`, `EMca_prob.tif` ve `_bin`	Ağırlıklı ortalama ve komite ortalaması topluluk haritaları	`sdm_core.py:8055-8103`
`EMsd.tif` (ve seçildiyse `EMcv`, `EMrange`)	Topluluk üyeleri arasındaki yayılım; eşiklenmez	`sdm_core.py:8057-8068`
`NT1`, `NT2`, `MIC_NT1`, `MIC_NT2`	ExDet yüzeyleri	`sdm_core.py:7227-7233`
`MESS`, `MoD`	Yalnız MESS seçildiyse	`sdm_core.py:7279-7281`
`_projection_options.json`	clamp, mess, mess_truncate, maskelenen dosya sayısı, MESS < 0 oranı, ızgara hizalama özeti	`sdm_core.py:8126-8143`

Bütün olasılık haritaları float32, nodata −9999 ve 0–1 aralığındadır; harita kendi aralığına göre germe yapılmaz, böylece eşikler ve senaryolar karşılaştırılabilir kalır (`sdm_core.py:7884-7889`). Kaynak raster coğrafi koordinatlıysa (ya da koordinat sistemi yoksa) çıktıya EPSG:4326 atanır; projeksiyonlu kaynakta özgün sistem korunur (`sdm_core.py:7830-7837`). Topluluk ikili eşiği komite ortalamasında 0,5 (çoğunluk oyu), ağırlıklı ortalamada üyelerin eşiklerinin aynı ağırlıklarla ortalamasıdır (`sdm_core.py:6423-6446`).

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Completed Projections` tablosu	Her çıktı dosyası için bir satır: `Scenario`, `Layer`, `Type`, `File`	boş	satır seçimi	`tabs/tab_projection.py:243-257`, `:644-662`
`Display Selected Map`	Seçili satırların haritasını görüntüleyiciye yükler	—	—	`tabs/tab_projection.py:259-262`, `:709-733`

`Type` sütunu olasılık haritalarında `Probability (0–1)`, ikili haritalarda `Binary (0/1)` yazar; yenilik ve yayılım dosyalarında anahtarın son parçası (ör. `NT1`, `sd`) görünür.

> **Dikkat:** `Display Selected Map`, `Probability` yazmayan her satırı ikili renk ölçeğiyle çizer. NT1, NT2, MESS ya da `_sd` haritalarını incelemek için bu dosyaları bir CBS yazılımında ya da 🗺 Map sekmesinde açmak daha doğru bir görüntü verir.

8.12 Harita görüntüleyici (Projection Map Viewer)

Sağ bölmedeki görüntüleyici (`RasterMapWidget`, `tabs/tab_projection.py:265-272`) ③'teki bugünkü harita görüntüleyicisiyle aynıdır: `Layer:` listesi yüklenen katmanları kısa adlarıyla gösterir (`Ensemble WMean · Prob`, `GLM · PA1 · Bin`), `Colormap:` olasılık haritasının renk ölçeğini seçer, `Hide BG` arka plan ülke sınırlarını gizler, `Title…` başlığı düzenler; araç çubuğundaki disket simgesi haritayı dergi sütun genişliğinde kaydeder. Başlık, senaryo, GCM, topluluk yöntemi ve harita türünden kurulur (`tabs/tab_projection.py:31-44`). Görüntüleyicinin tüm denetimleri ve yayın çıktısı Bölüm 12'de anlatılır.

> **İpucu:** Gelecek haritasının PNG'sini rapora girmesi için senaryo alt klasörüne (ör. `SSP585_2090\`) kaydedin. Kaydetme penceresi varsayılan olarak çıktı klasörünün köküne açılır; köke kaydedilen PNG raporda "6. Current Distribution" bölümüne girer (§11.5.6).

8.13 Çıktılar ve sonraki adıma geçenler

- `state["projections"][<senaryo>]`: anahtar → dosya yolu sözlüğü. ⑥ Range Change'teki `Current proj. key` / `Future proj. key` listeleri bundan kurulur (yalnız adında `_bin` geçen katmanlar).
- `state["novelty"][<senaryo>]`: ExDet özeti; rapordaki yenilik tablosu (§11.5.8) ve ⑥'daki dışdeğerleme payı bunu ve `NT1.tif` dosyasını kullanır.
- `state["grid_alignment"]`, `state["projection_settings"]`: rapordaki işleme notları ve ODMAP.
- `run_log`: senaryonun raster listesi, algoritmaları ve seçenekleri; ⑧'deki çok türlü toplu çalıştırma bu kaydı yineler (`tabs/tab_projection.py:580-584`).

8.14 Uzman soruları

S1. Gelecek rasterları adla mı, sırayla mı eşleniyor? Sıra hatası nasıl önlenir? Cevap: Temel kural konumdur: i'inci dosya i'inci sürekli değişkendir (`sdm_core.py:7335`, `:7407`). Ancak dosya adları (temizlenmiş) model değişkenleriyle aynı kümeyi oluşturuyorsa program sırayı adlara göre düzeltir (`sdm_core.py:7375-7379`). Dosya adları farklıysa düzeltme yapılamaz. En sağlam yol, gelecek katmanlarını eğitimle aynı adlarla saklamak ve `Browse Folder…` kullanmaktır. biomod2'de `BIOMOD_Projection` yeni ortamı katman adlarıyla eşler; HABITUS'ta ad eşlemesi yalnız adlar tam örtüştüğünde devreye girer.

S2. ExDet'in referans kümesi nedir? MaxEnt'in arka plan noktaları neden içinde değil? Cevap: Referans, tüm yalancı yokluk kümelerindeki varlık + yokluk satırlarının birleşimidir (`sdm_core.py:7142-7171`). Mesgaran vd. (2014) referansın modelleme yaklaşımına göre seçilmesini önerir; varlık-arka plan modelinde modelin gördüğü ortam budur. MaxEnt ise ayrıca 10 000 arka plan noktasıyla (`maxent_background`, `sdm_core.py:3916`, `:4596-4598`) kurulur ve bu noktalar referansa katılmaz. Erişilebilir alan dışındaki noktalar her iki kümede de yoktur; ancak MaxEnt'in arka planı daha geniş bir ortam aralığını kapsayabileceği için ExDet, MaxEnt açısından biraz kötümser (daha çok yenilik gösteren) olabilir. Bu etki ölçülmemiştir.

S3. MESS ile ExDet NT1 farklı sonuç verir mi? Cevap: Aynı referansla, aralık dışı hücre kümesi aynıdır: MESS < 0 ancak bir değer en küçükten küçük ya da en büyükten büyükse oluşur (`sdm_novelty.py:431-435`); NT1 < 0 da tam olarak budur (`sdm_novelty.py:221-225`). Değerler farklıdır (MESS yüzde ölçeğinde en küçük benzerlik, NT1 aralığa göre normalleştirilmiş uzaklıkların toplamı), işaret aynıdır. MESS ayrıca aralık içindeki hücrelere 0–100 arası bir benzerlik verir; NT1 orada 0'dır. Fark eğitimde sabit kalan değişkende ortaya çıkar (§8.8.1).

S4. NT2 eşiği neden 1? Mahalanobis uzaklığı mı, karesi mi? Cevap: Kod D^2 (kare) hesaplar ve referans kümedeki en büyük D^2'ye böler (`sdm_novelty.py:271-290`). Oran 1'i aşan hücre, referans kümedeki en uç noktadan bile daha "uzaktadır"; bu, Mesgaran vd.'nin ölçütüdür. Kare ya da kök alınması 1 eşiğini değiştirmez (tekdüze dönüşüm). Kovaryans ters matrisi `pinv` ile alınır; koşul sayısı 10^{12}'yi aşarsa NT2 hesaplanmaz (`sdm_novelty.py:257-269`).

S5. Clamping ExDet ve MESS sonuçlarını değiştirir mi? Cevap: Hayır. Kırpma yalnız tahmin matrisine (`X_valid`) uygulanır; ExDet ve MESS ham `X_flat` üzerinden hesaplanır (`sdm_core.py:7808-7814`, `:7937-7955`). Böylece clamping açıkken de yenilik haritaları gerçek dışdeğerlemeyi gösterir. Kırpma aralığı clamping için de referans kümenin en küçük/en büyük değeridir (`sdm_core.py:4929-4952`).

S6. Hangi modeller projekte ediliyor; çapraz doğrulama modelleri nerede? Cevap: Varsayılan olarak her algoritma ve her yalancı yokluk kümesi için tüm veriyle yeniden kurulan "Full" model projekte edilir; bu biomod2'nin (Thuiller vd. 2009) `allData` davranışına karşılık gelir (`sdm_core.py:7846-7855`). ③'te `Also project the cross-validation models` seçilirse katlama modelleri de projekte edilir, algoritma ortalaması ve yayılımı daha çok kopyaya dayanır; dosya adları `PA1_RUN1` gibi koşu etiketini taşır (`sdm_core.py:7874`). Bu

seçim ⑤'te değil ③'te yapılır ve bugünkü haritayla tutarlılık için orada tutulur (`tabs/tab_models.py:1538-1546`).

S7. Gelecek ikili haritasının eşiği nereden geliyor? Gelecek verisinden yeniden mi hesaplanıyor? Cevap: Hayır. Her model için ③'te çapraz doğrulamadan bulunan eşik (`tss_thresholds`, varsayılan yöntem max TSS) aynen kullanılır (`sdm_core.py:7908`); topluluk eşiği ağırlıklı ortalamada üyelerin eşiklerinin ağırlıklı ortalaması, komite ortalamasında 0,5'tir (`sdm_core.py:6423-6446`). Gelecekte gözlem olmadığı için eşiğin yeniden kestirilmesi zaten olanaksızdır. Olasılık haritaları germeye uğramadığından bu eşik iki dönemde aynı ölçekte anlamlıdır (`sdm_core.py:7884-7889`).

S8. Kategorik değişkenler (arazi örtüsü) gelecekte nasıl ele alınıyor? Cevap: Kategorik rasterlar gelecek listesine eklenmez; ① Data'da yüklenen ve modelde kullanılan bugünkü kategorik rasterlar yeniden okunur (`sdm_core.py:7474-7483`). Yani kategorik değişkenlerin değişmediği varsayılır. Gelecek arazi örtüsü senaryosu kullanılmak isteniyorsa program bunu desteklemez; değişken sürekli bir katmana (ör. orman oranı) dönüştürülüp sürekli olarak modellenmelidir.

S9. Gelecek katmanları farklı çözünürlükte olursa ne olur? Cevap: Çalışma sınırı yoksa sürekli katmanlar eğitim ızgarasına çift doğrusal, kategorikler en yakın komşu ile yeniden örneklenir (`sdm_grid.py:263-280`); sınır varsa kırpma sırasında en yakın komşu kullanılır (`sdm_mask.py:273-301`). Her iki durumda uyarı günlüğe ve rapora yazılır. Kaba bir GCM katmanının ince ızgaraya örneklenmesi bilgi üretmez; yalnızca hücre sayısını eşitler. Eğitim ve projeksiyonun aynı çözünürlükte (gerekirse ölçek küçültme sonrası) hazırlanması önerilir.

S10. MESS < 0 maskesi açıkken alan değişimi neden farklı çıkar? Cevap: Maskelenen hücreler gelecek haritasında nodata olur; ⑥ yalnız iki haritada da geçerli olan hücreleri sayar (`sdm_core.py:8292`). Bu hücreler kayıp ya da kazanım değil, "hesap dışı" olur; hem bugünkü hem gelecek alan küçülür. Örnek koşuda bu, hücrelerin %55,4'ünün çıkarılması demektir. Maske açık ve kapalı iki sonucu birlikte raporlamak, sonucun dışdeğerlemeye ne kadar bağlı olduğunu gösterir. ⑥'daki `Accessible area without extrapolation` seçeneği aynı soruyu haritayı değiştirmeden yanıtlar (§9.6).

S11. Bir GCM yerine birden çok GCM'nin ortalaması nasıl yapılır? Cevap: Program GCM'ler arası topluluk ya da senaryo belirsizlik yüzeyi üretmez; ODMAP kaydı da bunu söyler (`odmap.py`, "Uncertainty through scenarios"). Her GCM ayrı senaryo adıyla projekte edilir; ortalama ve yayılım haritaları dışarıda (ör. R `terra::app`) hesaplanır. GCM'ler arası fark çoğu zaman algoritmalar arası farktan büyüktür; tek GCM ile yapılan çalışmada bu sınırlılık yazılmalıdır.

S12. Projeksiyon yeniden üretilebilir mi? Cevap: Projeksiyonda rastgele bileşen yoktur: aynı modeller ve aynı rasterlar aynı haritayı verir. Modellerin kendisi ① Data'daki tohumla belirlenir. Hangi seçeneklerle çalışıldığı `_projection_options.json` dosyasına (`sdm_core.py:8126-8143`), raster listesi ve algoritmalar ise oturumdaki `run_log`'a yazılır; raster yolları rapora ya da çalıştırma kaydına girmez, bu yüzden kullanılan GCM dosyalarının kaynağı yöntem metninde ayrıca belirtilmelidir.

8.15 Sık hatalar ve tuzaklar

- **Varsayılan senaryo adını değiştirmemek.** `2050_SSP245` adıyla SSP5-8.5 katmanları projekte edilirse bütün dosyalar ve rapor yanlış adı taşır.
- **Sırayı karıştırmak.** Ad eşlemesi devreye girmediğinde (dosya adları farklı) sıra hatası iletisiz geçer. Belirtisi, olağandışı yüksek NT1 oranı ve `Most influential predictor for range novelty` satırında beklenmedik değişkendir.
- **Kategorik değişkeni listeye eklemek.** `Raster count mismatch!` iletisi çıkar; kategorikler otomatik yeniden kullanılır.
- **Aynı adla ikinci GCM'yi çalıştırmak.** Dosyalar aynı klasörde üzerine yazılır. Her GCM için ayrı ad kullanın.
- **`Remove All`'u yalnız listeyi temizlemek sanmak.** Bu düğme oturumdaki tüm projeksiyon ve alan değişimi sonuçlarını sıfırlar ve ⑥'yı kilitler.
- **MESS maskesini haber vermeden kullanmak.** Maskeli harita ile yapılan alan hesabı dışdeğerlenmiş hücreleri dışarıda bırakır; yöntemde belirtilmezse okur kayıp oranını yanlış yorumlar.
- **Bugünkü ve gelecek haritasında farklı clamping.** ③'teki ve ⑤'teki clamping seçimi ayrı ayrı yapılır; farklıysa alan değişiminin bir kısmı yöntem farkından gelir.
- **NT1'i görmezden gelmek.** Örnek koşuda gelecek hücrelerinin yarısından çoğu aralık dışındadır; bu hücrelerdeki uygunluk bir tahmin değil dışdeğerlemedir.

8.16 Kaynaklar

Elith, J., Kearney, M., Phillips, S., 2010. The art of modelling range-shifting species. Methods in Ecology and Evolution 1, 330–342. https://doi.org/10.1111/j.2041-210X.2010.00036.x

Mesgaran, M.B., Cousens, R.D., Webber, B.L., 2014. Here be dragons: a tool for quantifying novelty due to covariate range and correlation change when projecting species distribution models. Diversity and Distributions 20, 1147–1159. https://doi.org/10.1111/ddi.12209

Owens, H.L., Campbell, L.P., Dornak, L.L., Saupe, E.E., Barve, N., Soberón, J., Ingenloff, K., Lira-Noriega, A., Hensz, C.M., Myers, C.E., Peterson, A.T., 2013. Constraints on interpretation of ecological niche models by limited environmental ranges on calibration areas. Ecological Modelling 263, 10–18. https://doi.org/10.1016/j.ecolmodel.2013.04.011

Thuiller, W., Lafourcade, B., Engler, R., Araújo, M.B., 2009. BIOMOD – a platform for ensemble forecasting of species distributions. Ecography 32, 369–373. https://doi.org/10.1111/j.1600-0587.2008.05742.x

Velazco, S.J.E., Rose, M.B., de Andrade, A.F.A., Minoli, I., Franklin, J., 2022. flexsdm: An R package for supporting a comprehensive and flexible species distribution modelling workflow. Methods in Ecology and Evolution 13, 1661–1669. https://doi.org/10.1111/2041-210X.13874

9. Alan değişimi (⑥ Range Change)

Sekme: ⑥ Range Change · Kod: `tabs/tab_range.py` (arayüz), `sdm_core.py` `RangeChangeAnalyzer` (hesap), `sdm_grid.py` (ızgara hizalama)

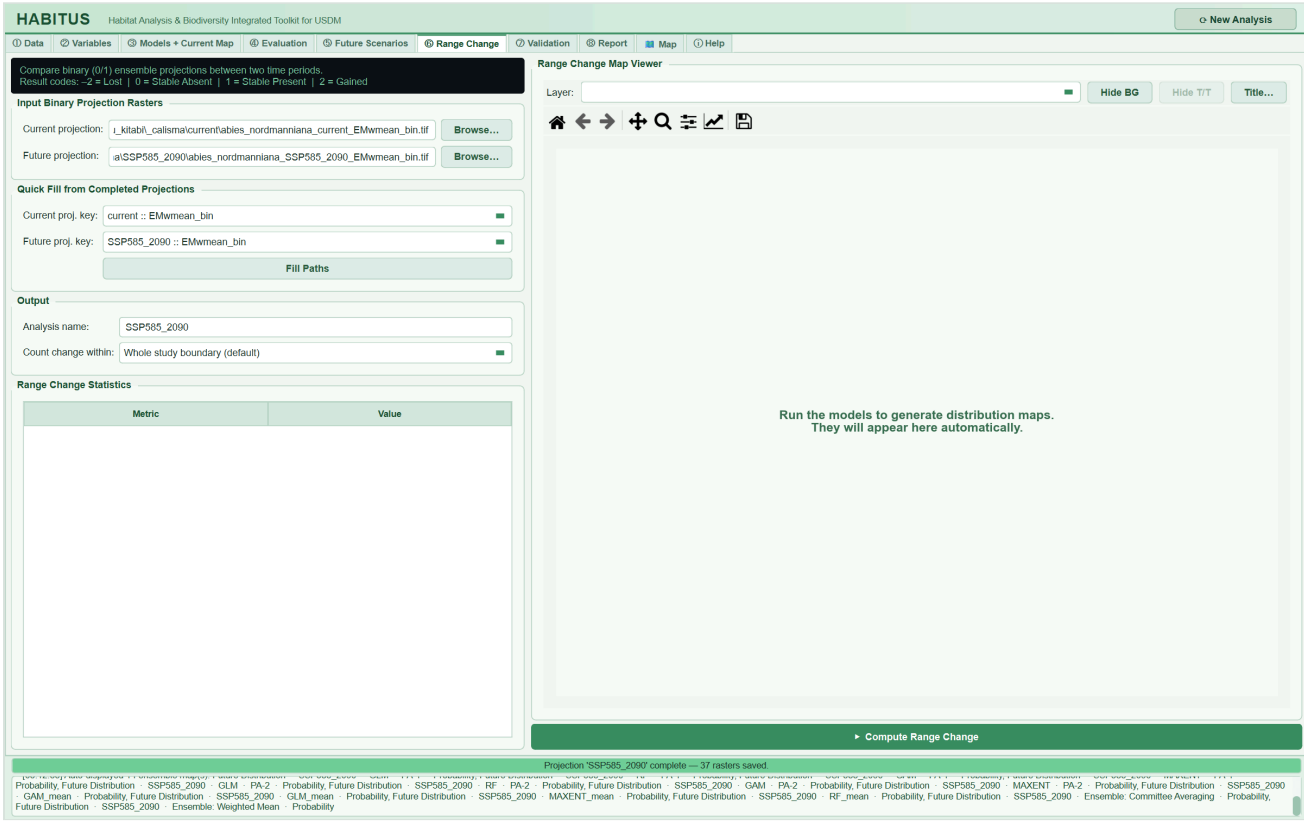
9.1 Bu sekme ne yapar

⑥ Range Change, iki ikili (0/1, yok/var) haritayı hücre hücre karşılaştırarak türün dağılım alanının nasıl değiştiğini sayar. Genellikle bugünkü topluluk haritası ile bir gelecek senaryosunun topluluk haritası karşılaştırılır; ama iki ikili harita aynı alanı kapladığı sürece (iki senaryo, iki algoritma, iki dönem) herhangi bir çift kullanılabilir. Hesap biomod2'deki (Thuiller vd. 2009) `BIOMOD_RangeSize` işlevinin karşılığıdır (`tabs/tab_range.py:1-5`, `sdm_core.py:8159-8163`).

Sekme ③ Models tamamlanınca açılır (`tabs/tab_models.py:1850`); listelerin gelecek haritalarıyla dolması için ⑤ Future Scenarios'ta en az bir projeksiyon yapılmış olmalıdır.

- **Girdi:** bugünkü ve gelecek ikili haritası; isteğe bağlı olarak ① Data'nın yazdığı erişilebilir alan raster'ı (`accessible_area.tif`) ve ⑤'in yazdığı ExDet NT1 yüzeyi.
- **Hesap:** her hücreye −2 (kaybedilen), 0 (sürekli yok), 1 (sürekli var), 2 (kazanılan) kodu verilir; sayımlar, yüzdeler ve net değişim hesaplanır (`sdm_core.py:8196-8347`).
- **Çıktı:** `<tür>_<analiz adı>_SRC.tif` alan değişimi haritası, `Range Change Statistics` tablosu, görüntüleyicide renkli harita ve ⑧ raporundaki "8. Range Change Analysis" bölümü.

9.2 Ekranın düzeni



Şekil 9.1. Range Change sekmesi çalıştırmadan önce: üstte sonuç kodlarını açıklayan şerit, solda Input Binary Projection Rasters (iki yol alanı), Quick Fill from Completed Projections (iki liste ve Fill Paths), Output (Analysis name, Count change within) ve boş Range Change Statistics tablosu; sağda boş Range Change Map Viewer.

Bölge	Arayüzdeki ad	Anlatıldığı yer
Bilgi şeridi	(başlıksız)	§9.1, §9.7
Dosya seçimi	`Input Binary Projection Rasters`	§9.3
Hızlı doldurma	`Quick Fill from Completed Projections`	§9.4
Çıktı ve kapsam	`Output`	§9.5, §9.6
İstatistik tablosu	`Range Change Statistics`	§9.8
Harita	`Range Change Map Viewer` (sağ bölme)	§9.9

9.3 İkili haritaların seçimi (Input Binary Projection Rasters)

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Current projection: + Browse…`	Başlangıç (bugünkü) ikili haritanın yolu	boş	`.tif .tiff .img .asc`	`tabs/tab_range.py:42-46, :167-173`
`Future projection: + Browse…`	Karşılaştırılacak (gelecek) ikili haritanın yolu	boş	aynı	`tabs/tab_range.py:48-52`

- **Ne işe yarar.** Bu iki alan hesaba giden tek girdidir; hızlı doldurma listeleri yalnızca bu alanları doldurur. Hesapta 1 "var", 0 "yok" sayılır; başka değerler (ör. olasılık 0,37) hiçbir sınıfa girmez (`sdm_core.py:8292-8296`).
- **Ne zaman, nasıl ayarlanır.** İki haritanın **aynı yöntemle** üretilmiş olması gerekir: aynı topluluk kuralı (`EMwmean_bin` ile `EMwmean_bin`), aynı algoritma kümesi, aynı eşik kuralı ve aynı clamping/maske seçimi. Farklı yöntemlerle üretilmiş iki haritanın farkı, iklim değişiminin değil yöntem farkının etkisini de taşır.
- **Sonuca etkisi.** Hangi haritanın seçildiği sonucu en çok etkileyen karardır (§9.10).
- **Dikkat.** `Browse…` ile seçilen dosyanın ikili olup olmadığı denetlenmez; yanlışlıkla bir `_prob.tif` seçilirse hücrelerin çoğu hiçbir sınıfa girmez ve sayımlar anlamsızlaşır. Her iki alan boşsa `Please specify both current and future binary rasters.` iletisi çıkar (`tabs/tab_range.py:200-202`).

9.4 Hızlı doldurma (Quick Fill from Completed Projections)

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Current proj. key:`	Oturumdaki ikili haritalardan bugünkünü seçer	`— select —`	`<senaryo> :: <katman>_bin`	`tabs/tab_range.py:62, :128-142`
`Future proj. key:`	Aynı listeden geleceği seçer	`— select —`	aynı	`tabs/tab_range.py:63`
`Fill Paths`	Seçilen anahtarların dosya yollarını üstteki alanlara yazar	—	—	`tabs/tab_range.py:64-66, :144-165`

- **Ne işe yarar.** Listeler, oturumda üretilen bütün projeksiyonların (③'teki `current` ve ⑤'teki her senaryo) adında `_bin` geçen katmanlarından kurulur (`tabs/tab_range.py:130-131`): tek modeller (`GLM_PA1_bin`), algoritma ortalamaları (`GLM_mean_bin`) ve topluluklar (`EMwmean_bin`, `EMca_bin`). Liste sekme her açıldığında yenilenir; değişmemişse kullanıcının seçimi korunur.

- **Ne zaman, nasıl ayarlanır.** Örnek koşuda `current :: EMwmean_bin` ve `SSP585_2090 :: EMwmean_bin` seçilip `Fill Paths`'e basılmıştır. Listeden seçmek, `Browse…`'a göre iki üstünlük taşır: yol hatası olmaz ve dışdeğerleme payı için gereken NT1 yüzeyi kesin olarak bulunur (§9.6.3). Ayrıca çok türlü toplu çalıştırma (⑧) listeden seçilen anahtarı yineler; elle seçilen yol ise yalnızca o türün dosyasını gösterir (`tabs/tab_range.py:206-211`).
- **Dikkat.** Listelerden biri `— select —` durumundayken `Fill Paths`'e basılırsa `Fill Paths takes the paths from the lists above. Choose the … binary map in the list first.` iletisi çıkar; listeler boşsa ③ ve ⑤'in çalıştırılması ya da `Browse` kullanılması önerilir (`tabs/tab_range.py:154-165`).

9.5 Analiz adı (Output · Analysis name)

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Analysis name:`	Çıktı dosyasının adı ve sonucun oturumdaki anahtarı	`SRC_current_future`	serbest metin	`tabs/tab_range.py:76-77`, `:216`, `:267`

Çıktı `<çıktı klasörü>\<tür>_<analiz adı>_SRC.tif` olarak yazılır (`tabs/tab_range.py:213-216`). Aynı adla ikinci çalıştırma hem dosyanın hem de oturumdaki sonucun üzerine yazar; rapor her ad için ayrı bir alt bölüm açar. Birden çok senaryo karşılaştırılıyorsa adı senaryoya göre verin (örnek koşuda `SSP585_2090`).

9.6 Değişimin sayıldığı kapsam (Count change within)

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Count change within:`	Başlık sayılarının ve haritanın kapsamı	`Whole study boundary (default)`	`Whole study boundary (default)` / `Accessible area` / `Accessible area without extrapolation`	`tabs/tab_range.py:80-94`, `sdm_core.py:8179-8184`

9.6.1 Üç kapsam

- **Ne işe yarar.** Seçilen kapsam, tablonun üst satırlarındaki sayıları ve haritada hangi hücrelerin kodlanacağını belirler (`sdm_core.py:8318-8322`):
- `Whole study boundary (default)`: iki haritada da geçerli olan her hücre.
- `Accessible area`: yalnız erişilebilir alan (① Data'daki `M`, türün ulaşabildiği alan) içindeki hücreler.
- `Accessible area without extrapolation`: erişilebilir alan içinde **ve** gelecek senaryosunda ExDet NT1 ≥ 0 olan (Mesgaran vd. 2014), yani hiçbir değişkenin eğitim aralığı dışına çıkmadığı hücreler (`sdm_core.py:8308-8311`).

Hangisi seçilirse seçilsin, girdiler mevcutsa öteki kapsamların net değişimi de tabloya ayrı satırlar olarak eklenir (`tabs/tab_range.py:285-288`). - **Ne zaman, nasıl ayarlanır.** Çalışma sınırı türün ulaşabileceği alandan çok genişse, öngörülen değişimin büyük kısmı hiçbir kaydın yakınında olmayan ve eğitim aralığı dışındaki hücrelerde toplanabilir (ipucu metni bunu söyler, `tabs/tab_range.py:85-93`). Erişilebilir alan kavramı Barve vd. (2011) BAM çerçevesine dayanır. Raporlanacak ana sayı için `Accessible area without extrapolation` en savunulabilir olanıdır; çalışma sınırının tamamı ise karşılaştırma için yanında verilir. - **Sonuca etkisi.** Örnek koşuda (`EMwmean_bin`) net değişim tüm sınırda **−46,13**, erişilebilir alanda **−30,25**, dışdeğerleme dışı erişilebilir alanda **−16,57** yüzdedir. Aynı iki harita, kapsama göre "yarı yarıya küçülme" ile "altıda bir küçülme" arasında değişen sonuçlar verir. - **Dikkat.** Erişilebilir alan yoksa ikinci ve üçüncü seçenek `No accessible area is available for this run, so change cannot be counted within it. …` iletisiyle durur;

NT1 yüzeyi bulunamazsa üçüncü seçenek `No ExDet NT1 surface was found next to the future map, …` iletisiyle durur (`tabs/tab_range.py:224-239`).

9.6.2 accessible_area.tif nereden gelir

① Data'da bir erişilebilir alan yöntemi (kayıtlar çevresinde tampon, dışbükey zarf ya da kullanıcı poligonu) seçilmişse, veri hazırlanırken bu maske eğitim ızgarasında 0/1 bir GeoTIFF olarak çıktı klasörüne `accessible_area.tif` adıyla yazılır (`tabs/tab_data.py:1536-1548`, `sdm_spatial.py:1065-1075`). Maske, seyreltme sonrası kalan varlık kayıtlarından kurulur (`sdm_core.py:2971-2974`). Örnek koşuda kayıtlar çevresinde 200 km'lik tampon kullanılmış ve maske raster kapsamının %40,0'ını kaplamıştır. Erişilebilir alan seçilmediyse (tampon 0 km) dosya yazılmaz.

9.6.3 NT1 yüzeyi nereden gelir

⑤ her projeksiyonda `<tür>_<senaryo>_NT1.tif` yazar (§8.9). ⑥, gelecek haritasının yolunu oturumdaki projeksiyon kayıtlarında arar ve bulduğu senaryonun NT1 dosyasını alır; bulamazsa gelecek haritasıyla aynı klasörde `_NT1.tif` ile biten **tek** bir dosya varsa onu kullanır (`tabs/tab_range.py:175-193`). Gelecek haritası başka bir klasörden `Browse…` ile seçildiyse ya da klasörde birden çok NT1 dosyası varsa NT1 bulunamaz.

> **Not:** Dışdeğerleme ölçütü yalnızca **gelecek** senaryosunun NT1'idir. Bugünkü haritada da dışdeğerlenmiş hücreler olabilir (örnek koşuda bugünkü haritanın %23,4'ü); bunlar bu kapsamda ayrıca dışlanmaz.

9.7 Sonuç kodları ve haritanın yazılması

Her hücre için, yalnız iki haritada da nodata olmayan hücrelerde (`valid`), şu kodlar yazılır (`sdm_core.py:8292-8296`, `:8319-8322`):

Kod	Ad	Koşul	Renk (görüntüleyici)
−2	Lost (kaybedilen)	bugün 1, gelecekte 0	kırmızı
0	Stable Absent (sürekli yok)	bugün 0, gelecekte 0	açık gri
1	Stable Present (sürekli var)	bugün 1, gelecekte 1	açık turkuaz
2	Gained (kazanılan)	bugün 0, gelecekte 1	yeşil

Seçilen kapsam dışındaki ve iki haritadan birinde nodata olan hücreler nodata kalır. Gelecek haritası bugünkü haritanın ızgarasında değilse önce en yakın komşu yöntemiyle konumla hizalanır (ikili harita olduğu için; `sdm_core.py:8242-8277`); erişilebilir alan ve NT1 de aynı şekilde bugünkü ızgaraya getirilir. Hizalama yapıldıysa uyarı günlüğe yazılır ve iş `Range change complete; the future raster was not on the current grid and was realigned by position (see log).` iletisiyle biter (`tabs/tab_range.py:311-318`). Hesap bellek bütçesine göre satır şeritleri halinde yapılır (64 MB; `sdm_core.py:8164-8177`); sonuç tek parça hesapla aynıdır.

9.8 İstatistik tablosu (Range Change Statistics)

Range Change Statistics

	Metric	Value
1	Counted within	whole study boundary
2	Lost pixels (present → absent)	4037
3	Stable present pixels	3065
4	Stable absent pixels	69308
5	Gained pixels (absent → present)	761
6	Total current range (pixels)	7102
7	Total future range (pixels)	3826
8	% Lost (of current range)	56.84
9	% Gained (of current range)	10.72
10	Net change (%)	-46.13
11	Net change, accessible area (%)	-30.25
12	Net change, accessible area without extrapolation …	-16.57
13	Changed cells in extrapolated conditions (%)	58.63

Şekil 9.2. Örnek koşunun Range Change Statistics tablosu (EMwmean_bin, tüm çalışma sınırı): kayıp, sürekli var, sürekli yok ve kazanılan hücre sayıları, bugünkü ve gelecek alan, yüzdeler, öteki kapsamların net değişimi ve değişen hücrelerin dışdeğerlenmiş koşullardaki payı.

Bütün sayılar **hücre** sayısıdır; alan birimine (km²) çevrilmez. Formüllerde L kaybedilen, G kazanılan, S_1 sürekli var, S_0 sürekli yok hücre sayısıdır; $C = L + S_1$ bugünkü alan, $F = G + S_1$ gelecek alandır (`sdm_core.py:8187-8193`, `:8312-8317`).

Satır (arayüzdeki ad)	Tanım	Örnek koşu
`Counted within`	Seçilen kapsamın adı	whole study boundary
`Lost pixels (present → absent)`	L	4037
`Stable present pixels`	S_1	3065
`Stable absent pixels`	S_0	69308
`Gained pixels (absent → present)`	G	761
`Total current range (pixels)`	$C = L + S_1$	7102
`Total future range (pixels)`	$F = G + S_1$	3826
`% Lost (of current range)`	$100\,L/C$	56,84
`% Gained (of current range)`	$100\,G/C$	10,72

Satır (arayüzdeki ad)	Tanım	Örnek koşu
`Net change (%)`	$100\,(F-C)/C = \%G - \%L$	−46,13
`Net change, accessible area (%)`	aynı formül, erişilebilir alan içinde	−30,25
`Net change, accessible area without extrapolation (%)`	aynı formül, erişilebilir alan ∩ NT1 ≥ 0	−16,57
`Changed cells in extrapolated conditions (%)`	$100\times$ (NT1 < 0 olan değişen hücreler) / (tüm değişen hücreler), değişen = $L+G$	58,63

Yüzdelerin paydası her zaman **bugünkü** alandır (C); $C=0$ ise payda 1 alınır (`max(tc, 1)`). Kazanım yüzdesi bu yüzden 100'ü aşabilir.

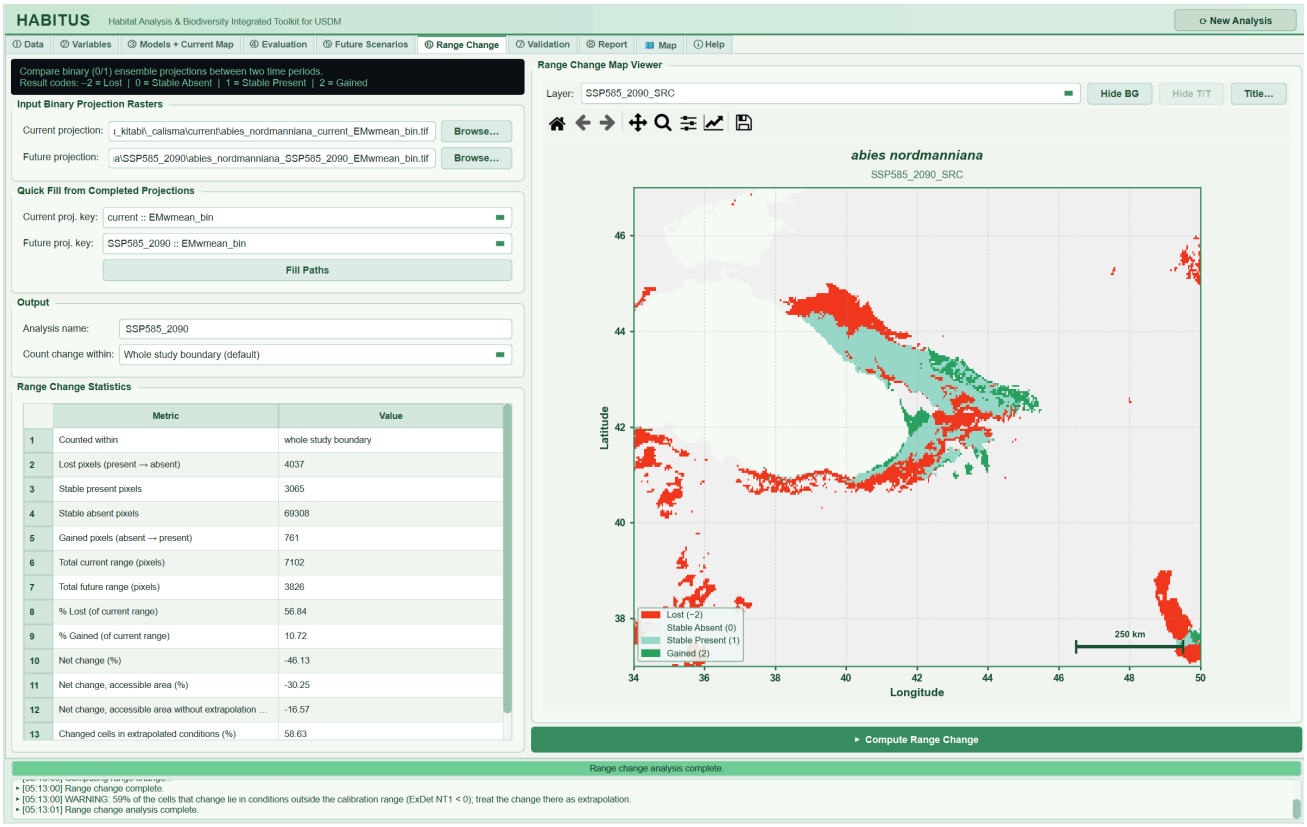
9.8.1 Öteki kapsamların net değişimi

- **Ne işe yarar.** Başlık sayıları seçilen kapsamda verilir; erişilebilir alan dosyası varsa `Net change, accessible area (%)`, ayrıca NT1 varsa `Net change, accessible area without extrapolation (%)` satırları kendiliğinden eklenir (`sdm_core.py:8330-8340`). Seçilen kapsamın kendi satırı tekrar edilmez.
- **Sonuca etkisi.** Okur, tek bir tablodan değişimin kapsama ne kadar duyarlı olduğunu görür. Örnek koşuda üç değer arasında yaklaşık 30 yüzde puanlık fark vardır.

9.8.2 Dışdeğerlenmiş değişim payı ve %25 uyarısı

- **Ne işe yarar.** `Changed cells in extrapolated conditions (%)`, kaybedilen ya da kazanılan hücrelerin ne kadarının gelecek senaryosunda en az bir değişkenin eğitim aralığı dışında kaldığı (NT1 < 0) yerlerde olduğunu söyler (`sdm_core.py:8299-8304`, `:8341-8343`). Bu oran, seçilen kapsamdan bağımsız olarak, iki haritada da geçerli olan bütün hücreler üzerinden hesaplanır.
- **Sonuca etkisi.** Oran %25 ya da üstündeyse günlüğe `WARNING: x% of the cells that change lie in conditions outside the calibration range (ExDet NT1 < 0); treat the change there as extrapolation.` yazılır (`tabs/tab_range.py:297-305`); raporda da aynı anlamda bir not çıkar (§11.5.9). Örnek koşuda oran %58,63'tür: değişimin yarısından fazlası modelin hiç görmediği koşullardadır.
- **Dikkat.** Bu satır yalnız NT1 bulunduğunda görünür; `Browse…` ile başka klasörden seçilen gelecek haritasında görünmez.

9.9 Harita



Şekil 9.3. Range Change sekmesi çalıştırıldıktan sonra: solda doldurulmuş yollar ve istatistik tablosu, sağda SSP585_2090_SRC haritası; kırmızı kaybedilen, açık turkuaz sürekli var, yeşil kazanılan hücreler, sol altta lejant ve 250 km ölçek çubuğu; alt şeritte değişen hücrelerin %59'unun dışdeğerlenmiş koşullarda olduğunu söyleyen uyarı.

İş bitince `<analiz adı>_SRC` katmanı `range_change` biçemiyle görüntüleyiciye yüklenir (`tabs/tab_range.py:307-309`). Renkler sabittir: kırmızı `#f03b20` (−2), açık gri `#f0f0f0` (0), açık turkuaz `#99d8c9` (1), yeşil `#2ca25f` (2) (`map_widget.py:142-146`). Görüntüleyicinin denetimleri ve yayın çıktısı Bölüm 12'de anlatılır. Örnek haritada kayıplar türün bugünkü alanının alçak ve batı kenarlarında, kazanımlar doğu uçta yoğunlaşır; çalışma alanının güneyindeki ve kuzeydoğusundaki dağınık kayıp lekeleri erişilebilir alanın dışındadır.

9.10 Hangi topluluk haritası seçilmeli: EMwmean_bin mi, EMca_bin mi?

İki topluluk ikili haritası farklı kurallarla eşiklenir (`sdm_core.py:6423-6446`):

- `EMwmean_bin`: ağırlıklı ortalama olasılık haritası, üyelerin kendi eşiklerinin aynı ağırlıklarla ortalaması olan bir eşikle kesilir.
- `EMca_bin`: komite ortalaması, yani "var" oyu veren kopyaların oranıdır; 0,5'te (çoğunluk oyu) kesilir.

Örnek koşunun aynı bugünkü/gelecek çifti için, tüm çalışma sınırında (betik: `sekil/_betik/09_alan_degisimi_karsilastirma.py`, programın `RangeChangeAnalyzer.compute` işleviyle; "ort." algoritma ortalamasının ikili haritası `<ALG>_mean_bin`, alanlar hücre sayısı, "erişil." erişilebilir alan, "dışd." dışdeğerleme):

İkili harita	Bugün	Gelecek	% Kayıp	% Kazanım	Net, sınır	Net, erişil.	Net, dışd. dışı	Dışd. değişim
EMwmean	7102	3826	56,84	10,72	−46,13	−30,25	−16,57	%58,6

İkili harita	Bugün	Gelecek	% Kayıp	% Kazanım	Net, sınır	Net, erişil.	Net, dışd. dışı	Dışd. değişim
EMca	7222	3161	65,12	8,89	−56,23	−45,38	−25,05	%60,2
GLM ort.	5764	1953	67,52	1,41	−66,12	−62,99	−45,69	%43,0
GAM ort.	10077	4194	72,58	14,20	−58,38	−41,24	−28,68	%68,4
MaxEnt ort.	9673	3856	68,58	8,45	−60,14	−43,56	−15,74	%61,3
RF ort.	7774	8264	58,01	64,32	+6,30	+20,58	−6,41	%78,6

- **Sonuca etkisi.** Topluluk kuralının değiştirilmesi tek başına net değişimi 10 puan oynatmıştır (−46 ile −56). Algoritmalar arasındaki fark çok daha büyüktür: RF genişleme (+6), GLM ağır daralma (−66) öngörür. RF'nin kazanımlarının %78,6'sı dışdeğerlenmiş koşullardadır; ağaç tabanlı yöntemlerin aralık dışında son değerde sabit kalması, gelecekte aralık dışına çıkan geniş alanları "uygun" bırakabilir.
- **Ne zaman, nasıl ayarlanır.** Ana sonuç için ③'te topluluk ölçütü olarak seçilen kuralın haritası kullanılır ve yöntemde adı yazılır. İki topluluk kuralının ve algoritma ortalamalarının sonuçları ek tabloda verilirse okur belirsizliğin kaynağını görür. Bu karşılaştırma için her harita ayrı bir `Analysis name` ile çalıştırılır.

9.11 Dağılım (dispersal) varsayımı

- **Ne işe yarar.** Program kazanılan hücreleri nerede olursa olsun sayar: gelecekte uygun hale gelen her hücreye türün ulaşabileceği varsayılır (**sınırsız dağılım**). Kayıplar her iki varsayımda da aynıdır.
- **Ne zaman, nasıl ayarlanır.** Ağaç türleri gibi yavaş yayılan türlerde **dağılım yok** varsayımı da raporlanmalıdır (Thuiller vd. 2005). Bu durumda gelecek alan yalnız sürekli var hücrelerinden oluşur: $F_{yok} = S_1$, net değişim 100 $(S_1 - C)/C = -\%L$. Örnek koşuda (`EMwmean_bin`, tüm sınır) bu, −46,13 yerine **−56,84** demektir. Program bu sayıyı ayrıca yazmaz; tablodaki `% Lost` satırının eksi işaretlisi dağılımsız net değişimdir.
- **Dikkat.** Erişilebilir alan kapsamı kısmen bir dağılım sınırı gibi davranır: alan dışındaki kazanımlar sayılmaz. Ancak erişilebilir alan bugünkü kayıtlardan kurulduğu için gelecekte bu alanın içinde kalan ama türün yüz yılda ulaşamayacağı hücreler yine sayılır. Belirli bir yayılma hızı (km/yıl) uygulayan bir seçenek yoktur.

9.12 Çıktılar ve sonraki adıma geçenler

- `<çıktı klasörü>\<tür>_<analiz adı>_SRC.tif`: float32, kodlar −2/0/1/2, nodata bugünkü haritanınki (yoksa −9999).
- `state["range_results"][<analiz adı>] = (istatistikler, dosya yolu)`: ⑧ raporu bu kaydı okur; ızgara uyarıları rapordaki işleme notlarına girer (`tabs/tab_report.py:1993-1998`).
- `run_log[<analiz adı>]`: anahtarlar, yollar ve kapsam; çok türlü toplu çalıştırma bu kaydı yineler (`tabs/tab_range.py:206-211`, `:240`).

9.13 Uzman soruları

S1. Hesap biomod2'nin BIOMOD_RangeSize işleviyle aynı mı? Cevap: Kodlama aynı mantıktadır (−2 kayıp, 0 sürekli yok, 1 sürekli var, 2 kazanım) ve yüzdeler bugünkü alana bölünür (`sdm_core.py:8187-8193`). Fark ek kapsamlardadır: biomod2 yalnız verilen haritaların tamamında sayar; HABITUS ayrıca erişilebilir alan ve dışdeğerleme dışı erişilebilir alan için net değişim ve değişimin dışdeğerlenmiş payını verir (`sdm_core.py:8297-8343`). Sonuçlar hücre sayısıdır; km² için hücre alanıyla çarpmak gerekir ve coğrafi koordinatlarda hücre alanı enleme göre değişir; program bu dönüşümü yapmaz.

S2. Net değişim neden "% Gained − % Lost"a eşit? Cevap: $F - C = (G + S_1) - (L + S_1) = G - L$; ikisi de C'ye bölündüğünden $\%Net = \%G - \%L$'dir. Örnek: 10,72 − 56,84 = −46,12 (yuvarlama farkıyla −46,13). Bu, tablonun kendi içinde tutarlılığını denetlemek için kullanılabilir.

S3. Erişilebilir alan dışındaki hücreler neden hesaba hiç girmiyor; kayıp sayılmaları gerekmez mi? Cevap: `Accessible area` seçildiğinde maske dışındaki hücreler dört sınıfın hiçbirine sayılmaz ve haritada nodata olur (`sdm_core.py:8312-8322`). Mantığı şudur: türün bugün ulaşamadığı bir hücrede "var" tahmini zaten gerçekleşmiş bir dağılımı değil potansiyel uygunluğu gösterir; bu hücrelerdeki "kayıp" gözlenmiş bir alanın kaybı değildir (Barve vd. 2011). Tüm sınırdaki sayım her zaman yapılır; ana kapsam başka bir seçenekse tabloya `Net change, whole study boundary (%)` satırı olarak eklenir (`sdm_core.py:8330-8340`, `tabs/tab_range.py:285-288`). Yalnız net değişim eklenir; tüm sınırdaki kayıp ve kazanım sayıları için analiz `Whole study boundary` ile ayrıca çalıştırılmalıdır.

S4. "Dışdeğerleme dışı" neden yalnız gelecek NT1'ine göre tanımlanmış? Cevap: Kod NT1 olarak ⑤'in gelecek senaryosu için yazdığı yüzeyi kullanır (`tabs/tab_range.py:175-193`, `sdm_core.py:8299-8311`). Bugünkü haritadaki dışdeğerlenmiş hücreler ayrıca dışlanmaz. Bugünkü haritanın da aralık dışı kısmını dışlamak isteyen kullanıcı `current\…_NT1.tif` ile gelecek NT1'ini dışarıda birleştirip maskelemelidir; programda bunun bir seçeneği yoktur.

S5. NT2 (yeni birleşim) değişim hesabında kullanılıyor mu? Cevap: Hayır; yalnız NT1 < 0 dışlanır (`sdm_core.py:8304`, `:8310-8311`). Örnek koşuda NT2 > 1 hücreleri gelecek senaryosunun yalnız %0,3'üdür, etkisi küçüktür. Bu oranın yüksek olduğu bir çalışmada NT2 yüzeyi dışarıda ek maske olarak kullanılabilir.

S6. Gelecek haritası farklı bir ızgaradaysa sonuç güvenilir mi? Cevap: Gelecek haritası bugünkü haritanın ızgarasına konumla hizalanır: aynı hücre boyunda tam sayı kayma interpolasyonsuz kaydırılır, değilse en yakın komşu ile yeniden örneklenir (`sdm_core.py:8242-8277`, `sdm_grid.py:244-280`). Hizalama günlükte ve raporun işleme notlarında yazılır. ⑤ 'te üretilen haritalar zaten eğitim ızgarasında olduğu için normal kullanımda hizalama gerekmez.

S7. MESS < 0 maskesi açık bir gelecek haritasıyla sonuç neden değişiyor? Cevap: Yalnız iki haritada da geçerli hücreler sayılır (`sdm_core.py:8292`). Maskelenen gelecek hücreleri hesaptan çıkar; bu hücrelerde bugün "var" olan alan da C'den düşer. Sonuç, "Accessible area without extrapolation" seçeneğine benzer ama aynı değildir: o seçenek ayrıca erişilebilir alanla kısıtlar ve haritayı değil sayımı maskeler.

S8. Eşik seçimi alan değişimini nasıl etkiler? Cevap: İkili haritalar ③'teki eşik kuralıyla (varsayılan max TSS) üretilir ve iki dönemde aynı eşik kullanılır (`sdm_core.py:7908`). Eşik düştükçe iki haritada da "var" alanı büyür; yüzdeler bugünkü alana bölündüğü için oransal değişim eşiğe göre farklı yönlerde oynayabilir. Örnek koşuda yalnız eşikleme kuralının farklı olduğu `EMwmean_bin` ile `EMca_bin` arasında net değişim 10 puan farklıdır (§9.10). Eşik duyarlılığı, ③'te farklı eşik kuralıyla yeniden kurup ⑥'yı yinelemekle sınanır.

S9. Bir hücrenin "var" olması türün orada olacağı anlamına mı gelir? Cevap: Hayır. İkili harita, uygunluk değerinin eşiği aştığı hücreleri gösterir; varlık-yalancı yokluk modelinde bu "iklimsel olarak uygun" demektir. Biyotik etkileşimler, arazi kullanımı ve dağılım sınırı modelde yoktur. Kazanım hücreleri ancak sınırsız dağılım varsayımı altında yeni alan sayılabilir (§9.11).

S10. Sonuç yeniden üretilebilir mi? Cevap: Hesapta rastgele bileşen yoktur; aynı iki harita aynı sayımı verir. Seçilen kapsam, analiz adı ve girdi yolları `run_log`'a, sonuçlar `state["range_results"]`'a yazılır; rapor her analiz için tabloyu ve haritayı içerir. Bu bölümdeki karşılaştırma tablosu `sekil/_betik/09_alan_degisimi_karsilastirma.py` ile programın kendi işlevi çağrılarak yeniden üretilebilir.

9.14 Sık hatalar ve tuzaklar

- **Farklı yöntemle üretilmiş iki haritayı karşılaştırmak.** Bugün için `EMwmean_bin`, gelecek için `EMca_bin` seçilirse değişimin bir kısmı yöntemden gelir.
- **`Browse…` ile olasılık haritası seçmek.** Program uyarmaz; sınıflara girmeyen hücreler sayılmaz.
- **Tüm sınır sonucunu tek sonuç olarak raporlamak.** Örnek koşuda tüm sınır −46, dışdeğerleme dışı erişilebilir alan −17 yüzdedir; hangisinin raporlandığı açıkça yazılmalıdır.
- **Dağılım varsayımını belirtmemek.** Tablodaki net değişim sınırsız dağılımı varsayar; dağılımsız değer `% Lost` satırının eksi işaretlisidir.
- **Değişen hücrelerin dışdeğerlenmiş payını görmezden gelmek.** %25 üzerindeki oran, sonucun önemli kısmının modelin hiç görmediği koşullara dayandığını söyler.
- **Hücre sayısını alan sanmak.** Tablo hücre sayar; km² için hücre alanıyla çarpılmalıdır (coğrafi ızgarada enleme göre değişir).
- **Başka klasörden seçilen gelecek haritası.** NT1 bulunamaz; dışdeğerleme payı satırı ve üçüncü kapsam kullanılamaz.

9.15 Kaynaklar

Barve, N., Barve, V., Jiménez-Valverde, A., Lira-Noriega, A., Maher, S.P., Peterson, A.T., Soberón, J., Villalobos, F., 2011. The crucial role of the accessible area in ecological niche modeling and species distribution modeling. Ecological Modelling 222, 1810–1819. https://doi.org/10.1016/j.ecolmodel.2011.02.011

Mesgaran, M.B., Cousens, R.D., Webber, B.L., 2014. Here be dragons: a tool for quantifying novelty due to covariate range and correlation change when projecting species distribution models. Diversity and Distributions 20, 1147–1159. https://doi.org/10.1111/ddi.12209

Thuiller, W., Lafourcade, B., Engler, R., Araújo, M.B., 2009. BIOMOD – a platform for ensemble forecasting of species distributions. Ecography 32, 369–373. https://doi.org/10.1111/j.1600-0587.2008.05742.x

Thuiller, W., Lavorel, S., Araújo, M.B., Sykes, M.T., Prentice, I.C., 2005. Climate change threats to plant diversity in Europe. Proceedings of the National Academy of Sciences 102, 8245–8250. https://doi.org/10.1073/pnas.0409902102

10. Bağımsız doğrulama (⑦ Validation)

Sekme: ⑦ Validation · Kod: `tabs/tab_validation.py` (arayüz ve hesap), `sdm_grid.py` `read_on_grid` (ızgara hizalama), `tabs/tab_report.py` `_s9_validation` (rapordaki bölüm)

10.1 Bu sekme ne yapar

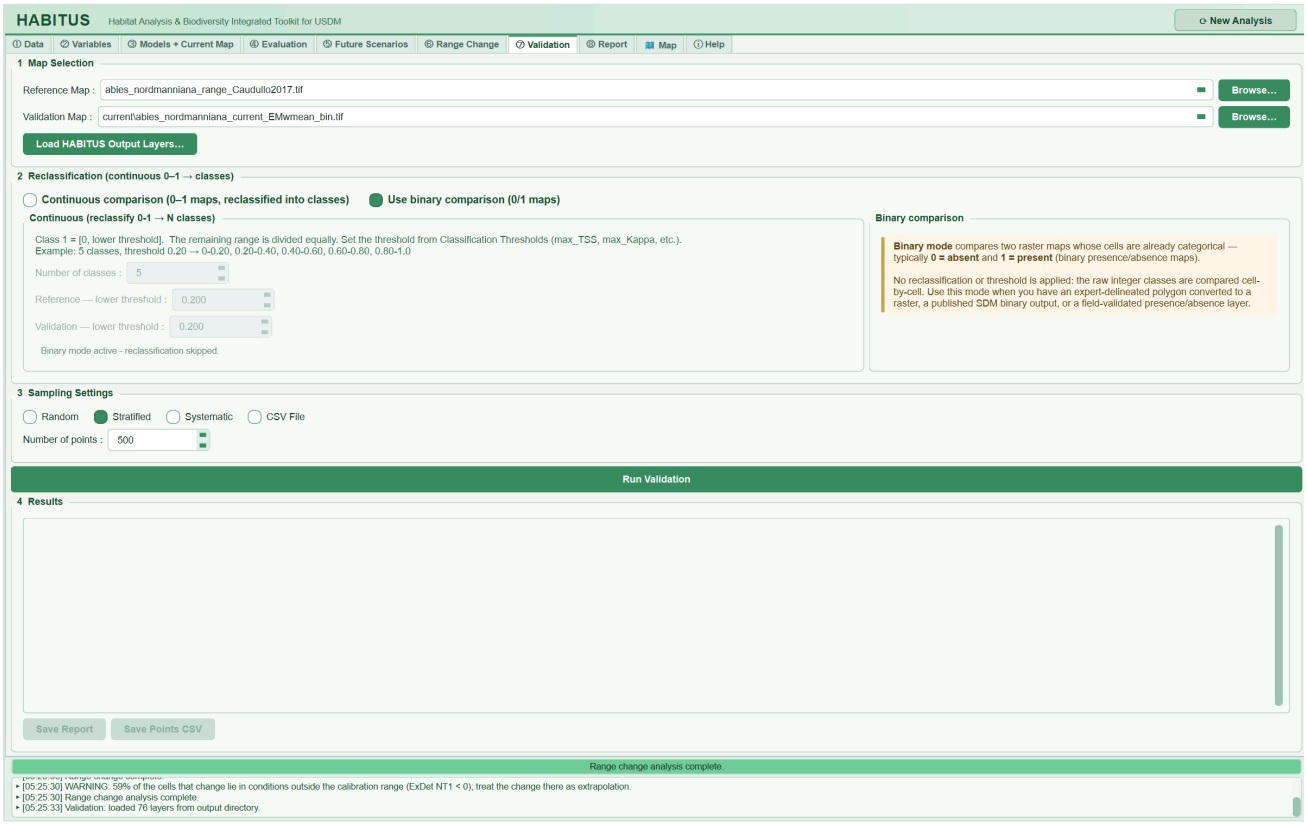
⑦ Validation, modelin ürettiği bir haritayı modelden bağımsız bir başvuru kaynağıyla karşılaştırır ve harita doğruluğunu ölçer. Başvuru kaynağı ya bir raster haritadır (uzman çizimi bir yayılış haritası, yayımlanmış bir dağılım haritası, saha verisinden üretilmiş bir varlık/yokluk katmanı) ya da noktaları ve başvuru değerleri bir CSV dosyasında verilen saha gözlemleridir. Karşılaştırma, harita üzerinden seçilen örnek noktalarda yapılır; sonuç bir karışıklık matrisi ve ondan türetilen ölçütlerdir.

- **Girdi:** iki GeoTIFF (başvuru ve doğrulanacak harita) ya da bir GeoTIFF ve bir CSV. Çoğu zaman doğrulanacak harita ③ Models'in yazdığı bir haritadır (örneğin topluluk ikili haritası `EMwmean_bin`).
- **Çıktı:** sonuç metni (ölçütler, karışıklık matrisi, sınıf başına ölçütler), isteğe bağlı rapor (`.txt`, `.json`, `.html`) ve nokta tablosu (`.csv`), yeniden sınıflanmış iki raster (`validation/` klasörü). Son doğrulama sonucu ⑧ Report'un 9. bölümüne geçer.
- **Kilit düzeni:** sekme, ③ Models çalışmayı bitirince açılır; aynı anda çıktı klasöründeki rasterlar iki açılır kutuya otomatik yüklenir (`tabs/tab_models.py:1848-1856`).

Bu sekmedeki doğrulama, ④ Evaluation'daki çapraz doğrulama skorlarından farklıdır: ④ modelin eğitim kayıtlarından ayrılan katlarla kendi verisi üzerinde nasıl ayırt ettiğini ölçer; ⑦ ise haritayı dışarıdan gelen bir kaynakla karşılaştırır. İkisi birbirinin yerine geçmez.

10.2 Ekranın düzeni

Sekme yukarıdan aşağıya dört numaralı gruptan oluşur.



Şekil 10.1. Validation sekmesi çalıştırılmadan önce: 1 Map Selection (başvuru haritası Caudullo vd. 2017, doğrulanan harita topluluk ikili haritası), 2 Reclassification (ikili mod seçili, sürekli mod alanları soluk), 3 Sampling Settings (Stratified, 500 nokta), Run Validation düğmesi ve boş 4 Results kutusu.

Bölge	Ne içerir	Alt bölüm
`1 Map Selection`	`Reference Map`, `Validation Map`, iki `Browse…`, `Load HABITUS Output Layers…`	§10.3
`2 Reclassification (continuous 0–1 → classes)`	İki mod düğmesi; solda `Continuous (reclassify 0-1 → N classes)` ayarları, sağda `Binary comparison` açıklaması	§10.4
`3 Sampling Settings`	`Random`, `Stratified`, `Systematic`, `CSV File`; CSV yolu; `Number of points`	§10.5
`Run Validation`	Karşılaştırmayı çalıştırır	§10.6
`4 Results`	Sonuç metni, `Save Report`, `Save Points CSV`	§10.6–10.8

Şekil 10.1 ve 10.2 örnek koşudan alınmıştır: *Abies nordmanniana*, başvuru haritası Caudullo vd. (2017) yayılış haritasının modelin ızgarasına çevrilmiş hâli (`abies_nordmanniana_range_Caudullo2017.tif`), doğrulanan harita topluluk ikili haritası (`current\abies_nordmanniana_current_EMwmean_bin.tif`).

10.3 Harita seçimi (1 Map Selection)

İki harita seçilir: **başvuru haritası** (doğru kabul edilen kaynak) ve **doğrulanacak harita** (modelin haritası). Arayüzde ikincisinin adı `Validation Map`'tir; bu bölümde "doğrulanan harita" diye anılır.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Reference Map :`	Başvuru rasterı	listedeki ilk katman	listedeki katmanlar	`tabs/tab_validation.py:64-72`
`Validation Map :`	Doğrulanan raster	listedeki ilk katman	listedeki katmanlar	`tabs/tab_validation.py:74-82`
`Browse…` (iki tane)	Bir GeoTIFF seçip ilgili listeye ekler ve seçer	—	`*.tif`, `*.tiff`	`tabs/tab_validation.py:341-350`
`Load HABITUS Output Layers…`	Çıktı klasöründeki bütün `.tif` dosyalarını iki listeye ekler	modeller bitince otomatik	düğme	`tabs/tab_validation.py:358-384`

10.3.1 Çıktı katmanlarını yükleme (`Load HABITUS Output Layers…`)

- **Ne işe yarar.** Çıktı klasörünü alt klasörleriyle birlikte tarar, bulduğu her `.tif` / `.tiff` dosyasını iki listeye **aynı sırayla** ve çıktı klasörüne göreli adla ekler (ör. `current\abies_nordmanniana_current_EMwmean_bin.tif`). Zaten listede olanı yinelemez. Günlüğe "Validation: loaded N layers from output directory." yazar; örnek koşuda 76 katman yüklenmiştir.
- **Ne zaman, nasıl ayarlanır.** ③ Models bitince program bu düğmeye kendisi basar; gelecek senaryoları ya da alan değişimi gibi sonradan yazılan haritaları listeye almak için yeniden basılır.
- **Sonuca etkisi.** Yalnız seçim kolaylığıdır.
- **Dikkat.** Liste çıktı klasöründeki bütün rasterları içerir: olasılık ve ikili haritalar, tek tek model haritaları, belirsizlik (`EMsd`), yenilik (`NT1`, `NT2`) ve erişilebilir alan rasterı. Doğru dosyayı seçmek kullanıcıya kalır. Çıktı klasörü yoksa "No output directory set yet. Run the analysis first." uyarısı çıkar. Başvuru haritası çoğu zaman çıktı klasöründe değildir; `Browse…` ile eklenir.

10.3.2 İki haritanın ızgarası

Karşılaştırma doğrulanan haritanın ızgarasında yapılır. Başvuru rasterı bu ızgaraya **konumuna göre** okunur (`tabs/tab_validation.py:422-455`, `sdm_grid.py:188-283`):

- Izgaralar aynıysa raster olduğu gibi okunur ve not çıkmaz.
- Hücre boyu ve koordinat sistemi aynı, yalnız tam sayıda hücre kaymışsa ara değer hesaplamadan kaydırılarak kopyalanır.
- Öteki her durumda `rasterio.warp.reproject` ile yeniden örneklenir: ikili modda en yakın komşu, sürekli modda iki doğrusal (bilinear) ara değer.

Hizalama yapıldıysa günlüğe ve sonuç metnine ("NOTE (grid)") şu biçimde bir not düşer: "Reference raster (W×H px) was not on the validation raster's grid (W×H px) and was realigned to it by position (…); N validation cell(s) not covered (x%)." Başvurunun kapsamadığı hücreler doğrulanan ızgaranın %5'inden fazlaysa ek uyarı çıkar: "… of the validation grid lies outside the reference raster (> 5%); those cells are not compared. Check that both maps cover the same area." (`tabs/tab_validation.py:447-452`, `sdm_grid.py:40`). Bu notlar raporun "Processing notices" bölümüne de geçer (`state["validation_notices"]`).

> **Dikkat:** Sürekli modda başvuru haritası iki doğrusal ara değerle yeniden örneklenir. Başvuru 0/1 bir haritaysa sınırlarda 0 ile 1 arasında ara değerler oluşur ve bunlar ara sınıflara düşer. İkili bir başvuruyu ikili modda karşılaştırın ya da modele hiç yeniden örnekleme gerekmeyecek biçimde, aynı ızgarada hazırlayın.

10.4 Karşılaştırma modu ve yeniden sınıflama (2 Reclassification)

İki mod vardır. Mod düğmeleri her iki durumda da etkindir; ikili mod seçilince sürekli mod ayarları soluklaşır ve önizleme satırı "Binary mode active - reclassification skipped." yazar (`tabs/tab_validation.py:191-198`, `316-321`).

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Continuous comparison (0–1 maps, reclassified into classes)`	İki haritayı N sınıfa yeniden sınıflayıp karşılaştırır	seçili	mod düğmesi	`tabs/tab_validation.py:100-103`
`Use binary comparison (0/1 maps)`	Hücre değerlerini yuvarlayıp olduğu gibi karşılaştırır	—	mod düğmesi	`tabs/tab_validation.py:102`, `599-603`, `672-677`
`Number of classes :`	Sürekli modda sınıf sayısı	5	2–10	`tabs/tab_validation.py:132-137`
`Reference — lower threshold :`	Başvuru haritasında 1. sınıfın üst sınırı	0.200	0.000–0.990, adım 0.01	`tabs/tab_validation.py:139-145`
`Validation — lower threshold :`	Doğrulanan haritada 1. sınıfın üst sınırı	0.200	0.000–0.990, adım 0.01	`tabs/tab_validation.py:147-153`
`Ref breaks:` / `Val breaks:` satırı	Sınıf sınırlarının canlı önizlemesi	—	—	`tabs/tab_validation.py:155-160`, `316-328`

10.4.1 Sürekli karşılaştırma (`Continuous comparison`)

- **Ne işe yarar.** Her iki harita 0–1 aralığındaki değerlerine göre N sınıfa ayrılır. Sınırlar: $[0,\ t,\ \ldots,\ 1]$; 1. sınıf $[0, t)$, kalan aralık t ile 1 arasında $N-1$ **eşit** parçaya bölünür; son sınıf 1'i de içerir (`tabs/tab_validation.py:330-339`, `461-474`). $t = 0$ ise sınırlar 0 ile 1 arasında eşit aralıklıdır. 0'ın altındaki ve 1'in üstündeki değerler hiçbir sınıfa girmez ve karşılaştırma dışı kalır. Sınıf etiketleri "C1 [0.000–0.200]" biçimindedir.
- **Ne zaman, nasıl ayarlanır.** İki haritanın da gerçekten sürekli (olasılık ya da uygunluk) olduğu durumlarda: örneğin iki modelin olasılık haritalarını ya da modelin olasılık haritasını başka bir çalışmanın uygunluk haritasıyla karşılaştırırken. Alt eşik olarak ④ Evaluation'ın `Classification Thresholds` tablosundaki bir eşik (max_TSS gibi) kullanılabilir; böylece 1. sınıf "uygun değil" anlamına gelir.
- **Sonuca etkisi.** Sınıf sayısı arttıkça tam eşleşme zorlaşır; genel doğruluk ve Kappa düşer. Başvuru ve doğrulanan harita için farklı alt eşik verilebilir; iki modelin farklı eşikleri varsa bu, sınıfların anlamını eşitler. Sürekli modda ayrıca noktalardaki ham değerlerle regresyon ölçütleri hesaplanır (§10.7.3).
- **Dikkat.** Başvuru 0/1 bir haritaysa yalnız 1. ve son sınıf dolar, ara sınıflar başvuruda boş kalır; karışıklık matrisinin köşegeni çoğunlukla 1. sınıfta toplanır ve ölçütler ikili moddakinden düşük çıkar. Bu iki mod aynı şeyi ölçmez. İkili bir başvuru için ikili modu kullanın.

10.4.2 İkili karşılaştırma (`Use binary comparison`)

- **Ne işe yarar.** İki haritanın hücre değerleri tam sayıya yuvarlanır ve hiçbir eşik uygulanmadan hücre hücre karşılaştırılır (`np.round`; `tabs/tab_validation.py:601`, `674`). Sınıflar veriden alınır ve "Class 0", "Class 1" diye etiketlenir. İki sınıf varsa sonuç metnine doğru/yanlış pozitif ve negatif sayıları, duyarlılık, özgüllük ve TSS eklenir (§10.7.2).
- **Ne zaman, nasıl ayarlanır.** Başvuru bir varlık/yokluk haritası, doğrulanan harita da modelin ikili haritası (`…_bin.tif`) olduğunda. Arayüzdeki açıklama üç tipik başvuru sayar: rastera çevrilmiş uzman çizimi bir poligon, yayımlanmış bir ikili SDM çıktısı, sahada doğrulanmış bir varlık/yokluk katmanı.
- **Sonuca etkisi.** Örnek koşuda (Caudullo vd. 2017 başvurusu, `EMwmean_bin`, tabakalı 500 nokta) genel doğruluk 0,866, Kappa 0,732'dir (Şekil 10.2).
- **Dikkat.** Bu modda değerler 0,5'ten yuvarlanır; ikili mod seçiliyken bir **olasılık** haritası (`…_prob.tif`) seçilirse harita sessizce 0,5 eşiğiyle ikiye ayrılmış olur, modelin kendi eşiğiyle değil. Tam 0,5 değeri de NumPy'nin yarımı çifte yuvarlama kuralı nedeniyle 0'a gider. İkili modda daima `…_bin.tif` dosyalarını seçin. Etiketler 0/1 dışındaysa (ör. 1/2) TP/FP/TN/FN ataması küçük kodun "yokluk" olduğu varsayımıyla yapılır (`tabs/tab_validation.py:908-911`).

10.5 Örnekleme tasarımı (3 Sampling Settings)

Karşılaştırma bütün hücrelerde değil, seçilen örnek noktalarda yapılır. Bu, doğruluk değerlendirmesinin yerleşik uygulamasıdır (Congalton 1991; Olofsson vd. 2014): sonuç, örneklemden kestirilen bir doğruluktur.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Random`	Başvurunun sınıflanmış hâlindeki geçerli hücrelerden yerine koymadan rastgele	seçili	yöntem düğmesi	`tabs/tab_validation.py:485-492`
`Stratified`	Başvurunun her sınıfından eşit sayıda hücre	—	yöntem düğmesi	`tabs/tab_validation.py:494-505`
`Systematic`	Rasterın üzerine düzgün aralıklı bir ızgara	—	yöntem düğmesi	`tabs/tab_validation.py:507-520`
`CSV File`	Başvuru değerleri bir CSV'deki noktalardan	—	yöntem düğmesi	`tabs/tab_validation.py:526-544`, `622-651`
`CSV (id, x, y, reference_value):` + `Browse`	CSV dosyasının yolu (yalnız `CSV File` seçiliyken görünür)	boş	`*.csv`	`tabs/tab_validation.py:218-229`
`Number of points :`	İstenen nokta sayısı	500	30–100 000, adım 50	`tabs/tab_validation.py:231-239`

10.5.1 Rastgele (`Random`)

- **Ne işe yarar.** Başvuru haritasının sınıflanmış hâlinde geçerli (NaN olmayan) bütün hücreler arasından `Number of points` kadar hücre yerine koymadan seçilir. Rastgele sayı üreteci koşu tohumundan (`random_seed`; bilinmiyorsa 42) türetilmiş yerel bir `numpy.random.default_rng`'dir; aynı tohum aynı noktaları verir ve sonuç metni "Random (seed 42)" yazar (`tabs/tab_validation.py:413-420`, `699-701`).
- **Ne zaman, nasıl ayarlanır.** Haritanın bütününde doğruluğu kestirmek için varsayılan ve en kolay yorumlanan tasarımdır: her hücrenin seçilme olasılığı eşittir, bu yüzden genel doğruluk ve Kappa doğrudan alan ağırlıklı kestirimlerdir.
- **Sonuca etkisi.** Nadir sınıf (tür çoğu zaman alanın küçük bir kısmında bulunur) az sayıda noktayla temsil edilir; o sınıfın üretici ve kullanıcı doğrulukları belirsiz olur.
- **Dikkat.** Seçilen bir hücre doğrulanan haritada NaN ise atılır; bu yüzden geçerli nokta sayısı istenenden az olabilir. Sonuç metnindeki `Points` satırı gerçekten kullanılan sayıyı yazar.

10.5.2 Tabakalı (`Stratified`)

- **Ne işe yarar.** Başvuru haritasının her sınıfından $\max(1, \lfloor n/K \rfloor)$ hücre seçilir (K sınıf sayısı); bir sınıfta bu kadar hücre yoksa hepsi alınır (`tabs/tab_validation.py:494-505`). Tohum, rastgele tasarımdakiyle aynıdır.
- **Ne zaman, nasıl ayarlanır.** Her sınıfın üretici ve kullanıcı doğruluğunu yeterli noktayla kestirmek gerektiğinde, özellikle varlık sınıfı alanın küçük bir kısmıysa.
- **Sonuca etkisi.** Sınıflar eşit sayıda noktayla temsil edildiği için genel doğruluk ve Kappa **alan ağırlıklı değildir**: her sınıf haritadaki payından bağımsız olarak aynı ağırlıkla sayılır. Örnek koşuda başvuru haritasının geçerli hücrelerinin yalnız %5,8'i varlıktır (77 171 hücrenin 4458'i); tabakalı 500 noktada ise yarı yarıya varlık ve yokluk vardır. Bu örneklemden genel doğruluk 0,866, Kappa 0,732 çıkmıştır. Aynı iki haritanın bütün ortak hücrelerinde (77 171 hücre) genel doğruluk 0,938, Kappa 0,553'tür (`_calisma/validation/` altındaki yeniden sınıflanmış iki rasterdan bu kitap için hesaplandı). Duyarlılık (0,78 ve 0,76) ve özgüllük (0,95 ve 0,95) ise iki hesapta birbirine yakındır, çünkü bunlar sınıf içi oranlardır.
- **Dikkat.** Tabakalı tasarımda genel doğruluğun ve Kappa'nın alan ağırlıklı kestirimi için her sınıfın sonucu haritadaki alan payıyla ağırlıklandırılmalıdır (Olofsson vd. 2014). Program bu ağırlıklandırmayı yapmaz; sınıf başına ölçütleri (üretici/kullanıcı doğruluğu, duyarlılık, özgüllük, TSS) kullanın ya da genel doğruluğu `Save Points CSV` çıktısı ve sınıf alanlarıyla kendiniz ağırlıklandırın. n, K'ya tam bölünmezse toplam nokta sayısı n'den az olur.

10.5.3 Sistematik (`Systematic`)

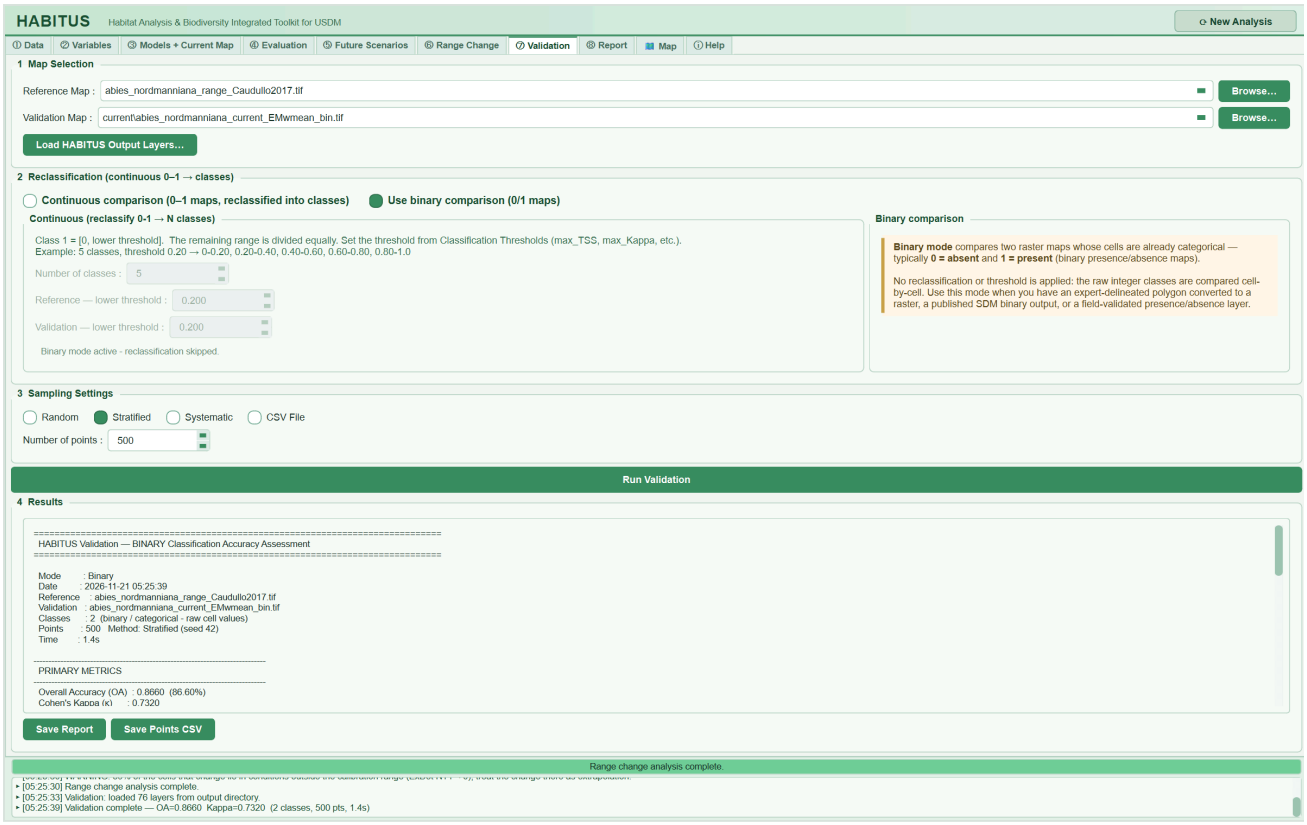
- **Ne işe yarar.** $g = \lfloor \sqrt{n} \rfloor$ olmak üzere, rasterın bütün satır ve sütun aralığına $g \times g$ düzgün aralıklı bir ızgara serilir; ızgara noktalarından başvuruda geçerli olanlar alınır, n'e ulaşılınca durulur (`tabs/tab_validation.py:507-520`). Rastgelelik yoktur; tohum kullanılmaz.
- **Ne zaman, nasıl ayarlanır.** Alanı düzenli kapsamak istendiğinde. Yakın noktaların mekânsal öz ilişkisini azaltır.
- **Sonuca etkisi.** Izgara rasterın bütün dikdörtgeni üzerine serildiği için NoData alanına (deniz, çalışma sınırının dışı) düşen noktalar atılır; geçerli nokta sayısı n'den çoğu zaman belirgin biçimde azdır. 500 istendiğinde ızgara 22 × 22 = 484 noktadır.
- **Dikkat.** Raster düzeninde periyodik bir yapı varsa (yol, ızgara biçimli arazi kullanımı) sistematik örnek bu yapıyla çakışabilir.

10.5.4 CSV noktaları (`csv File`)

- **Ne işe yarar.** Başvuru değerleri haritadan değil, bir CSV dosyasındaki noktalardan alınır. Dosya UTF-8 olmalı ve başlık satırında `x`, `y`, `reference_value` sütunlarını (isteğe bağlı `id`) içermelidir; adlar büyük/küçük harfe duyarlıdır (`tabs/tab_validation.py:526-544`). Her nokta doğrulanan rasterın dönüşümüyle hücreye çevrilir; raster dışına düşen, sayısal olmayan ya da sütunu eksik satırlar sessizce atlanır. `reference_value`, **başvuru sınıf sınırlarıyla** sınıflanır (`Reference — lower threshold` ve `Number of classes`); doğrulanan sınıf, doğrulanan haritada noktanın hücresinden okunur (`tabs/tab_validation.py:636-648`).
- **Ne zaman, nasıl ayarlanır.** Saha gözlemleri, ayrı toplanmış varlık/yokluk noktaları ya da başka bir çalışmanın nokta verisi ile doğrulama için. Koordinatlar doğrulanan rasterın koordinat sisteminde olmalıdır (örnek veride derece; boylam `x`, enlem `y`).
- **Sonuca etkisi.** `Number of points` bu yöntemde kullanılmaz; dosyadaki geçerli bütün noktalar kullanılır. Yeniden sınıflanmış rasterlar bu modda yazılmaz.
- **Dikkat.** Koordinat dönüşümü yapılmaz; farklı bir koordinat sistemindeki noktalar yanlış hücrelere düşer ya da raster dışında kalıp atılır. Ondalık ayırıcı nokta olmalıdır ("41,73" gibi değerler sayı olarak okunamaz ve satır atlanır). `reference_value` sürekli moddaki gibi 0–1 ölçeğinde sınıflandığından 0/1 saha verisi sürekli modda 1. ve son sınıfa düşer; bu verilerle ikili karşılaştırma için §10.10'daki tuzağa bakın.

10.6 Çalıştırma (Run Validation)

`Run Validation` işi arayüz iş parçacığında çalıştırır ve süresince adımları gösteren küçük bir pencere açar (`tabs/tab_validation.py:550-558`). Adımlar sonuç kutusuna "▸ " ile yazılır: doğrulanan rasterın geçerli hücre sayısı, en küçük/en büyük/ortalama değeri ve tekil değer sayısı; sınıf sınırları; sınıf başına hücre sayıları; başvuru için aynısı; örnekleme ve geçerli nokta sayısı. Ardından bu metin sonuç raporuyla değiştirilir (`tabs/tab_validation.py:963`). Doğrulanan harita seçilmemişse "Please select a Validation Map.", başvuru seçilmemişse "Please select a Reference Map.", ortak geçerli nokta yoksa "No valid overlapping points found." iletisi çıkar.



Şekil 10.2. Validation sekmesi çalıştırıldıktan sonra: ikili mod, tabakalı 500 nokta (seed 42); 4 Results kutusunda Mode, Classes, Points ve PRIMARY METRICS başlığı altında genel doğruluk 0,8660 ve Kappa 0,7320; Save Report ve Save Points CSV düğmeleri etkin.

10.7 Ölçütler ve sonuç metni

Örnek noktalardaki (başvuru sınıfı, doğrulanan sınıf) çiftlerinden scikit-learn ile bir karışıklık matrisi kurulur; satırlar başvuru, sütunlar doğrulanan haritadır (`tabs/tab_validation.py:749-779`). Sınıf listesi iki taraftaki gözlenen sınıfların birleşimidir.

10.7.1 Birincil ölçütler (`PRIMARY METRICS`)

Sonuç metnindeki ad	Tanım	Kod
`Overall Accuracy (OA)`	Doğru sınıflanan nokta oranı	`accuracy_score`, `tabs/tab_validation.py:754`
`Cohen's Kappa (κ)`	Rastlantısal uyum düzeltilmiş uyum (Cohen 1960)	`cohen_kappa_score`, `:755`
`F1-Score (Macro)` / `F1-Score (Weighted)`	Sınıf başına F1'in basit / destek ağırlıklı ortalaması	`f1_score`, `:756-757`
`Precision (Macro)` / `Recall (Macro)`	Sınıf başına kesinlik / duyarlılığın basit ortalaması	`:758-759`
`Kappa Interpretation`	Kappa'nın sözel yorumu (aşağıda)	`:900-904`

Bütün oranlarda payda sıfırsa sonuç 0 alınır (`zero_division=0`). Kappa,

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

ile hesaplanır; p_o gözlenen uyum (genel doğruluk), p_e iki haritanın sınıf paylarından beklenen rastlantısal uyumdur.

Kappa yorum ölçeği. Sonuç metni Kappa'yı şu ölçekle sözelleştirir (`tabs/tab_validation.py:900-904`): $\kappa < 0$ "Poor"; $< 0{,}20$ "Slight"; $< 0{,}40$ "Fair"; $< 0{,}60$ "Moderate"; $< 0{,}80$ "Substantial"; diğerleri "Almost Perfect". Bu, Landis ve Koch (1977)'un önerdiği ölçektir. Ölçek, gözlemciler arası uyum için önerilmiş bir uzlaşmadır; harita doğruluğu için kesin sınırlar değildir.

> **Dikkat:** Kappa, varlık sınıfının yaygınlığına (prevalans) bağlıdır (Allouche vd. 2006). Aynı iki harita farklı örnekleme tasarımlarıyla çok farklı Kappa verebilir: örnek koşuda tabakalı örnekte 0,732, bütün hücrelerde 0,553 (§10.5.2). Kappa'yı tek başına raporlamayın; duyarlılık, özgüllük ve TSS'yi ve kullanılan örnekleme tasarımını birlikte verin.

10.7.2 İkili özet (`BINARY SUMMARY`)

İkili modda ve tam iki sınıf varsa sonuç metnine şunlar eklenir (`tabs/tab_validation.py:906-922`): karışıklık matrisi [0, 1] sırasıyla okunur; `True Positive (TP)`, `False Positive (FP)`, `True Negative (TN)`, `False Negative (FN)`; `Sensitivity` = TP/(TP+FN); `Specificity` = TN/(TN+FP); `True Skill Stat TSS` = duyarlılık + özgüllük − 1 (Allouche vd. 2006). TSS prevalanstan bağımsızdır; ikili doğrulamada Kappa'dan daha güvenilir bir özettir.

Örnek koşunun karışıklık matrisi (tabakalı, 500 nokta; rapor dosyasından): başvuruda yokluk olan 250 noktanın 238'i modelde de yokluk, 12'si varlık; başvuruda varlık olan 250 noktanın 55'i modelde yokluk, 195'i varlıktır. Buradan duyarlılık 0,780, özgüllük 0,952, TSS 0,732 çıkar. Eşit sayıda varlık ve yokluk örneklendiğinde Kappa ile TSS aynı değeri alır; örnekteki eşitlik bundandır.

10.7.3 Regresyon ölçütleri (yalnız sürekli mod)

Sürekli modda noktalardaki **ham** değerler (başvuru ve doğrulanan, 0–1) üzerinden hesaplanır (`tabs/tab_validation.py:761-767, 923-927`): `R^2` (`r2_score`, belirleme katsayısı; Pearson r^2'si değildir ve negatif olabilir), `RMSE`, `MAE`, `Bias` = ortalama(doğrulanan − başvuru); pozitifse "(Over)", negatifse "(Under)" eklenir. Başvuru 0/1 bir haritaysa bu ölçütler bir olasılık haritasının 0/1'e uzaklığını ölçer ve sınırlı anlam taşır.

10.7.4 Karışıklık matrisi ve sınıf başına ölçütler

Sonuç metni üç tablo daha yazar (`tabs/tab_validation.py:929-958`):

- `CONFUSION MATRIX`: satırlar başvuru ("Ref \ Val"), sütunlar doğrulanan sınıflar.
- `PER-CLASS METRICS`: her sınıf için `Precision`, `Recall`, `F1`, `Support` (o sınıftaki başvuru noktası sayısı).
- `PRODUCER'S & USER'S ACCURACY`: üretici doğruluğu (başvurudaki bir sınıfın haritada doğru gösterilme oranı) = recall; kullanıcı doğruluğu (haritadaki bir sınıfın sahada doğru çıkma oranı) = precision (Congalton 1991).

Sonuç metninin başında mod, tarih, başvuru ve doğrulanan dosyanın adı, sınıf sayısı, sürekli modda iki eşik ve sınıf sınırları, nokta sayısı ve yöntem (tohumla), varsa ızgara notları ve süre yer alır.

10.8 Dışa aktarma

Arayüzdeki ad	Ne yapar	Varsayılan	Kod
`Save Report`	Sonucu uzantıya göre `.txt` (sonuç metni), `.json` (bütün sonuç sözlüğü) ya da `.html` (özet ve karışıklık matrisi) olarak kaydeder	`validation_report_<tarih_saat>`	`tabs/tab_validation.py:970-992`, `1021-1066`
`Save Points CSV`	Her noktayı kaydeder: `id`, `x`, `y`, `ref_value`, `val_value`, `ref_class`, `val_class`, `match`	`validation_points_<tarih_saat>.csv`	`tabs/tab_validation.py:994-1019`
(otomatik)	Raster–raster modunda `validation/Reclassify_ref_<ad>.tif` ve `validation/Reclassify_val_<ad>.tif`	—	`tabs/tab_validation.py:726-730`, `819-839`

İki düğme ilk başarılı çalıştırmaya kadar devre dışıdır. Yeniden sınıflanmış rasterlar float32 ve NoData = −9999 ile yazılır; başvuru hizalanmışsa doğrulanan rasterın coğrafi başvurusunu alır. JSON dosyası sınıf sınırlarını, karışıklık matrisini, sınıf raporunu, tohumu ve ızgara notlarını içerir; bir doğrulamanın sonradan denetlenmesi için en eksiksiz kayıttır.

⑧ Report'un 9. bölümü, sekmedeki **en son** doğrulama sonucunu kullanır (`tabs/tab_report.py:2269-2320`): mod, başvuru ve doğrulanan katman, nokta sayısı, genel doğruluk, Kappa, ağırlıklı F1, (varsa) R^2, RMSE, MAE, karışıklık matrisi ve yeniden sınıflanmış haritalar. Örnekleme yöntemi bu bölümde yer almaz; raporu yazarken ekleyin.

10.9 Uzman soruları

S1. Bu doğrulama neden "bağımsız" sayılıyor; ④ Evaluation'daki skorlardan farkı ne? ④ Evaluation, modelin eğitim kayıtlarından türetilen çapraz doğrulama katlarında hesaplanır; aynı kayıt kaynağı ve aynı sözde yokluk tasarımı kullanılır. ⑦ Validation ise haritayı dışarıdan gelen bir kaynakla karşılaştırır (`tabs/tab_validation.py:550-817`). Bağımsızlık başvurunun niteliğine bağlıdır: başvuru, modelin eğitim kayıtlarından türetilmiş bir yayılış haritasıysa (örneğin aynı herbaryum kayıtlarından çizilmişse) tam bağımsız sayılmaz. Program başvurunun kökenini denetlemez; bunu kullanıcı belirtmelidir.

S2. Tabakalı örneklemle bulunan genel doğruluk ve Kappa haritanın doğruluğunu yansıtır mı? Doğrudan yansıtmaz. `Stratified` her başvuru sınıfından eşit sayıda nokta alır (`tabs/tab_validation.py:494-505`) ve sonuçları alan payına göre ağırlıklandırmaz. Örnek koşuda tabakalı 500 noktada genel doğruluk 0,866 ve Kappa 0,732; bütün ortak hücrelerde 0,938 ve 0,553'tür (§10.5.2). Alan ağırlıklı kestirim için Olofsson vd. (2014)'ün tabakalı kestiricileri kullanılmalıdır; program bunları hesaplamaz. Sınıf içi oranlar (duyarlılık, özgüllük, TSS) tabakalı tasarımda da yansızdır ve raporlanması önerilir.

S3. Kappa yerine hangi ölçüt raporlanmalı? Kappa prevalansa bağlıdır (Allouche vd. 2006) ve harita doğruluğu literatüründe eleştirilmiştir; Pontius ve Millones (2011) miktar ve yerleşim uyuşmazlığını ayrı ayrı raporlamayı önerir. Program miktar/yerleşim uyuşmazlığını hesaplamaz. İkili modda TSS, duyarlılık ve özgüllük sonuç metnindedir (`tabs/tab_validation.py:906-922`); bunlar ve karışıklık matrisi raporlanırsa okur öteki ölçütleri kendisi hesaplayabilir. Miktar ve yerleşim uyuşmazlığı `Save Points CSV` çıktısından ya da yeniden sınıflanmış iki rasterdan hesaplanabilir.

S4. Başvuru ve doğrulanan haritanın ızgarası farklıysa ne olur? Başvuru, doğrulanan haritanın ızgarasına konumuna göre okunur: aynı ızgara olduğu gibi, tam hücre kayması kaydırmayla, öteki her durum yeniden örneklemeyle (ikili modda en yakın komşu, sürekli modda iki doğrusal) (`sdm_grid.py:188-283`). Yapılan işlem ve kapsanmayan hücre oranı sonuç metnine "NOTE (grid)" olarak yazılır; %5'i aşan kapsama açığı ayrıca

uyarılır (`tabs/tab_validation.py:436-452`). Yeniden örnekleme ölçek farkı içeriyorsa (örneğin 1 km'lik bir başvuru 4,6 km'lik model ızgarasına) en yakın komşu seçimi başvurunun sınır hücrelerini değiştirir; ölçek uyumu kullanıcının sorumluluğundadır.

S5. Örnekleme hangi alanda yapılıyor; erişilebilir alanla sınırlı mı? Hayır. `Random` ve `Stratified`, başvurunun sınıflanmış hâlinde geçerli olan bütün hücrelerden seçer; ardından doğrulanan haritada NaN olanlar atılır (`tabs/tab_validation.py:486-492`, `709-722`). Böylece örnekleme alanı, iki haritanın da değer taşıdığı bütün ızgaradır; ① Data'daki erişilebilir alanla sınırlanmaz. Erişilebilir alan dışındaki uzak, açıkça uygun olmayan hücreler kolay doğru sınıflanan yokluklardır ve genel doğruluğu yükseltir. Doğrulamayı erişilebilir alanla sınırlamak için başvuru haritasının bu alan dışını NoData yapmak gerekir.

S6. Sonuç tekrarlanabilir mi? `Random` ve `Stratified`, koşu tohumundan (`random_seed`; bilinmiyorsa 42) kurulan yerel bir üreteç kullanır (`tabs/tab_validation.py:413-420`, `699`); aynı tohum, aynı haritalar ve aynı nokta sayısı aynı noktaları verir. Tohum sonuç metnine ("Stratified (seed 42)") ve JSON raporuna yazılır. `Systematic` ve `CSV File` rastgelelik içermez.

S7. 500 nokta yeterli mi; sonucun belirsizliği nedir? Program genel doğruluk ya da Kappa için güven aralığı vermez. Sonuç tek bir örneklemden kestirildiği için örnekleme belirsizliği taşır. Basit rastgele örneklemde genel doğruluğun standart hatası yaklaşık $\sqrt{p(1-p)/n}$'dir; $p = 0{,}9$ ve $n = 500$ için bu yaklaşık 0,013'tür. Nadir sınıfın doğruluğu için tabakalı tasarım ya da daha çok nokta gerekir (Olofsson vd. 2014). Farklı tohumlarla tekrarlayarak ya da `Number of points`'i artırarak (en çok 100 000) değişkenliği görmek mümkündür.

S8. Bir olasılık haritasını ikili bir başvuruyla nasıl karşılaştırmalıyım? En doğru yol, olasılık haritasını modelin seçilen eşiğiyle ikiye ayırmış haritayı (`…_bin.tif`) ikili modda kullanmaktır. İkili modda bir olasılık haritası seçilirse değerler yuvarlanır, yani 0,5 eşiği uygulanmış olur (`tabs/tab_validation.py:601`); bu, modelin eşiği değildir. Sürekli modda ise 0/1 başvuru yalnız 1. ve son sınıfa düşer ve ölçütler iki şeyi karışık ölçer (§10.4.1).

S9. CSV noktalarıyla doğrulamada saha varlık/yokluk verisini nasıl kullanmalıyım? CSV yönteminde `reference_value` her zaman başvuru sınıf sınırlarıyla (sürekli moddaki gibi) sınıflanır (`tabs/tab_validation.py:642`); doğrulanan harita ise ikili modda yuvarlanmış 0/1 değerlerle okunur. İkili modda bu iki kodlama uyuşmaz (başvuru 1 ve N, harita 0 ve 1) ve sonuç anlamsız olur. Saha 0/1 verisini doğru karşılaştırmak için sürekli modu, `Number of classes` = 2 ve iki alt eşiği aynı değerde (ör. 0.500) kullanın: o zaman başvurudaki 0 ve 1 sırasıyla 1. ve 2. sınıfa, doğrulanan olasılık haritası da aynı eşikle iki sınıfa ayrılır. Doğrulanan harita ikili (`…_bin.tif`) ise eşik 0 ile 1 arasında herhangi bir değer olabilir.

S10. Başvuru haritası olarak bir yayılış (chorology) haritası kullanmanın sınırları nelerdir? Caudullo vd. (2017) gibi yayılış haritaları uzman çizimidir ve genellikle türün doğal yayılışının dış sınırını gösterir; sınırın içinde türün bulunmadığı yerler (uygun olmayan yükseklik, arazi kullanımı) de "varlık" sayılır. Bu, modelin uygun bulmadığı ama yayılış içinde kalan hücrelerde yanlış negatif sayısını artırır. Örnek koşuda modeldeki varlık hücreleri (7102) başvurudakinden (4458) fazladır; modelin "fazla" gösterdiği yerlerin bir kısmı iklimsel olarak uygun ama türün ulaşmadığı yerler olabilir. Bu nedenle bu karşılaştırma harita uyumunu ölçer; modelin biyolojik doğruluğunun kesin kanıtı değildir.

S11. Sonuç rapora otomatik geçiyor mu, ve ne eksik kalıyor? Evet; ⑧ Report'un 9. bölümü en son doğrulama sonucunu kullanır (`tabs/tab_report.py:2269-2320`). Rapor örnekleme yöntemini ve tohumu yazmaz; bu bilgi JSON ve metin raporunda vardır. Raporun yorum cümlesindeki "κ values above 0.60 are generally considered acceptable …" ifadesi genel bir kalıptır; kaynak göstermez ve Kappa'nın prevalans bağımlılığını hesaba katmaz. Raporu yayımlamadan önce örnekleme tasarımını ekleyin ve yorumu uyarlayın.

10.10 Sık hatalar ve tuzaklar

- **İkili modda olasılık haritası seçmek.** Değerler 0,5'ten yuvarlanır; modelin eşiği kullanılmaz. İkili modda `…_bin.tif` seçin.
- **İkili başvuruyu sürekli modda karşılaştırmak.** Ara sınıflar boş kalır, Kappa yapay olarak düşer (§10.4.1).
- **Tabakalı örneklemin Kappa'sını alan doğruluğu diye raporlamak.** Örnek koşuda 0,732 ile 0,553 arasındaki fark bundandır (§10.5.2). Tasarımı açıkça yazın, sınıf içi ölçütleri verin.
- **CSV'de yanlış sütun adı ya da ondalık virgül.** Sütunlar `x`, `y`, `reference_value` olmalı, sayılar noktalı yazılmalıdır; uymayan satırlar uyarı verilmeden atlanır. Hiç geçerli satır kalmazsa "No valid points in CSV." iletisi çıkar.
- **CSV koordinatlarının farklı koordinat sisteminde olması.** Dönüşüm yapılmaz; noktalar yanlış hücreye düşer ya da atılır.
- **İkili modda CSV ile 0/1 saha verisi.** Kodlar uyuşmaz (§10.9, S9); sürekli mod, 2 sınıf ve eşit eşikler kullanın.
- **`Systematic` ile istenen nokta sayısını beklemek.** Izgara bütün dikdörtgene serilir; NoData'ya düşenler atılır.
- **Başvuru ile doğrulanan haritanın kapsamlarının farklı olması.** "NOTE (grid)" notunu ve kapsama açığı uyarısını okuyun; karşılaştırma yalnız ortak hücrelerde yapılır.
- **Ardışık doğrulamalardan yalnız sonuncusunun rapora geçtiğini unutmak.** Raporda görmek istediğiniz karşılaştırmayı en son çalıştırın; öteki sonuçları `Save Report` ile ayrı kaydedin.

10.11 Kaynaklar

Allouche, O., Tsoar, A., Kadmon, R., 2006. Assessing the accuracy of species distribution models: prevalence, kappa and the true skill statistic (TSS). Journal of Applied Ecology 43, 1223–1232. https://doi.org/10.1111/j.1365-2664.2006.01214.x

Caudullo, G., Welk, E., San-Miguel-Ayanz, J., 2017. Chorological maps for the main European woody species. Data in Brief 12, 662–666. https://doi.org/10.1016/j.dib.2017.05.007

Cohen, J., 1960. A coefficient of agreement for nominal scales. Educational and Psychological Measurement 20, 37–46. https://doi.org/10.1177/001316446002000104

Congalton, R.G., 1991. A review of assessing the accuracy of classifications of remotely sensed data. Remote Sensing of Environment 37, 35–46. https://doi.org/10.1016/0034-4257(91)90048-B

Landis, J.R., Koch, G.G., 1977. The measurement of observer agreement for categorical data. Biometrics 33, 159–174. https://doi.org/10.2307/2529310

Olofsson, P., Foody, G.M., Herold, M., Stehman, S.V., Woodcock, C.E., Wulder, M.A., 2014. Good practices for estimating area and assessing accuracy of land change. Remote Sensing of Environment 148, 42–57. https://doi.org/10.1016/j.rse.2014.02.015

Pontius, R.G., Millones, M., 2011. Death to Kappa: birth of quantity disagreement and allocation disagreement for accuracy assessment. International Journal of Remote Sensing 32, 4407–4429. https://doi.org/10.1080/01431161.2011.552923

11. Rapor, ODMAP ve çok türlü çalışma (⑧ Report)

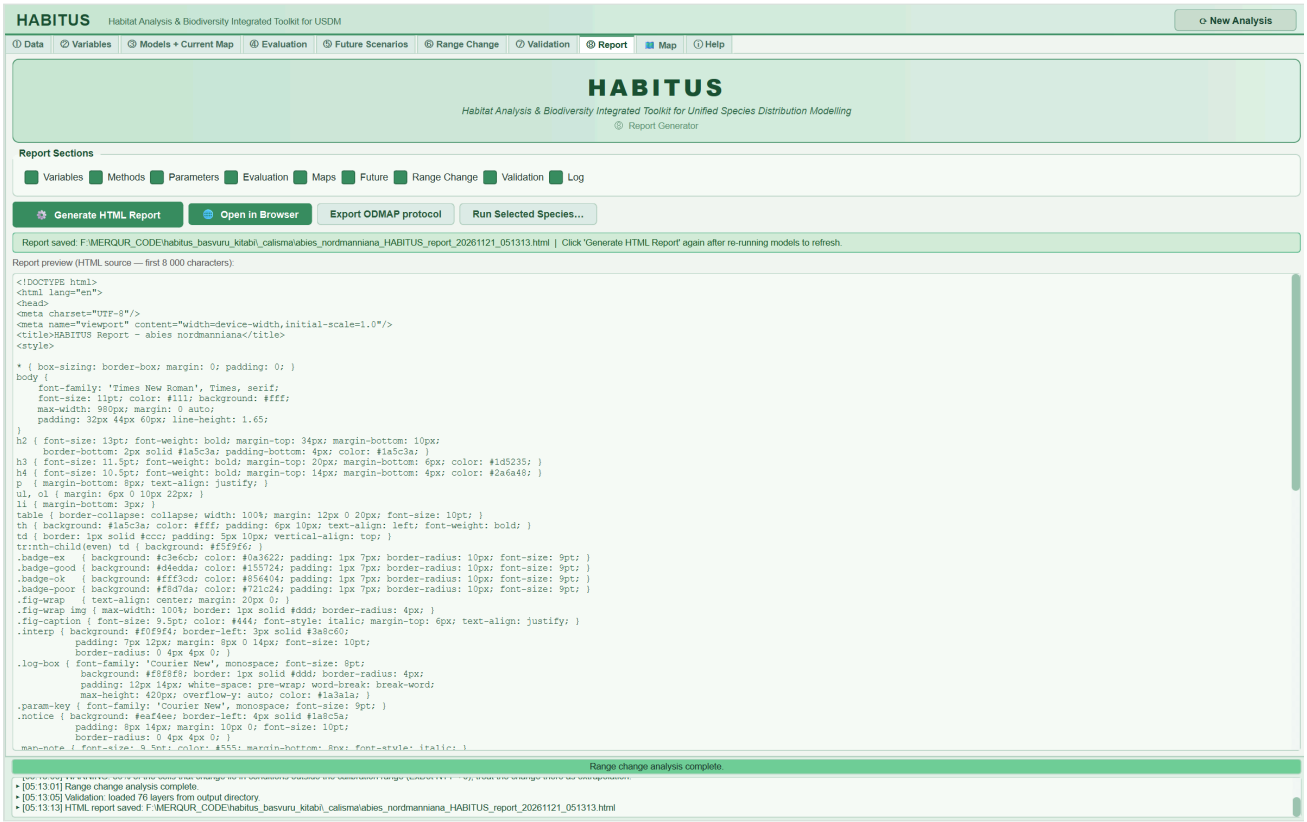
Sekme: ⑧ Report · Kod: `tabs/tab_report.py` (arayüz ve HTML rapor), `odmap.py` (ODMAP protokolü), `batch_species.py` (çok türlü toplu çalıştırma), `map_widget.py` ve `study_area_figure.py` (yayın haritaları)

11.1 Bu sekme ne yapar

⑧ Report, oturumdaki bütün sonuçları tek bir HTML dosyasında toplar; ayrıca ODMAP raporlama protokolünü kısmen doldurur ve çok türlü bir kayıt dosyasında tamamlanan iş akışını öteki türler için yineler. Sekme ③ Models tamamlanınca açılır (`tabs/tab_models.py:1852`).

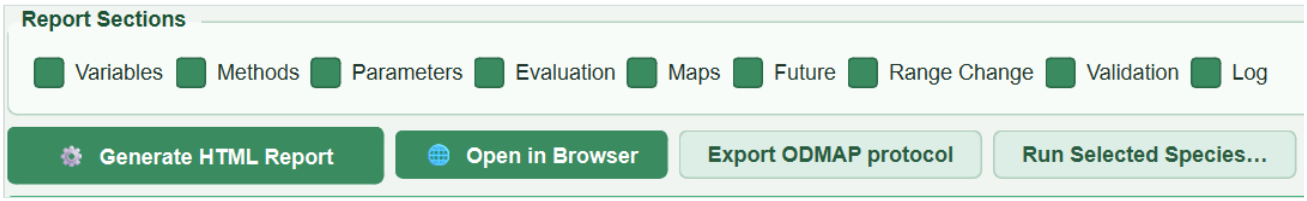
- **Girdi:** oturum durumu (modeller, topluluk, projeksiyonlar, alan değişimi, doğrulama sonuçları), çıktı klasöründeki `figures\*.png` grafikleri, kullanıcının harita görüntüleyiciden kaydettiği PNG haritalar, `habitus_run_config.json` ve oturum günlüğü (`.log`).
- **Çıktı:** `<tür>_HABITUS_report_<YYYYMMDD_HHMMSS>.html` (bütün görüntüler dosyanın içine gömülü), istenirse `odmap_protocol.md` ve `odmap_protocol.csv`, toplu çalıştırmada her tür için bir alt klasör ve `batch_summary.csv`.
- **Hesap:** Rapor yeni bir hesap yapmaz; var olan sonuçları okur, yorum cümleleri ve tablolar üretir. Haritaları yeniden çizdiği tek yer alan değişimi ve doğrulama haritalarıdır.

11.2 Ekranın düzeni



Şekil 11.1. Report sekmesi rapor üretildikten sonra: üstte HABITUS başlık kutusu, Report Sections kutusunda dokuz bölüm kutusu, altında Generate HTML Report, Open in Browser, Export ODMAP protocol ve Run Selected Species… düğmeleri, yeşil durum satırında kaydedilen dosyanın yolu, altta HTML kaynağının ilk 8000 karakterlik önizlemesi.

Sekme tek bölmelidir (`tabs/tab_report.py:376-465`). Yukarıdan aşağıya: başlık kutusu, `Report Sections` (bölüm seçimi, §11.3), dört düğme (§11.4, §11.6, §11.8), ilerleme çubuğu ve durum satırı, `Report preview (HTML source — first 8 000 characters):` önizleme kutusu.



Şekil 11.2. Report Sections kutusu ve düğme satırı yakından: Variables, Methods, Parameters, Evaluation, Maps, Future, Range Change, Validation ve Log kutularının hepsi varsayılan olarak işaretli.

11.3 Rapor seçenekleri (Report Sections)

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Variables`	2. bölüm: değişken taraması ve seçilen değişkenler	açık	açık / kapalı	`tabs/tab_report.py:400`, `:2416-2418`
`Methods`	3. bölüm: yöntem metni	açık	açık / kapalı	`:401`, `:2421-2423`
`Parameters`	4. bölüm: genel ayarlar ve algoritma ayarları tablosu	açık	açık / kapalı	`:402`, `:2426-2428`
`Evaluation`	5. bölüm: değerlendirme tabloları ve grafikleri	açık	açık / kapalı	`:403`, `:2431-2433`
`Maps`	6. bölüm: kullanıcının kaydettiği bugünkü harita PNG'leri	açık	açık / kapalı	`:404`, `:2436-2439`
`Future`	7. bölüm: senaryo klasörlerindeki harita PNG'leri	açık	açık / kapalı	`:405`, `:2443-2448`
`Range Change`	8. bölüm: alan değişimi tabloları ve haritaları	açık	açık / kapalı	`:406`, `:2464-2467`
`Validation`	9. bölüm: ⑦ Validation sonuçları	açık	açık / kapalı	`:407`, `:2470-2473`
`Log`	10. bölüm: oturum günlüğü	açık	açık / kapalı	`:408`, `:2476-2481`

- **Ne işe yarar.** Her kutu raporun bir bölümünü ekler ya da çıkarır. 1. bölüm (Study Summary), çevresel yenilik bölümü (ExDet) ve işleme notları kutulardan bağımsız olarak **her zaman** eklenir (`tabs/tab_report.py:2412-2413`, `:2450-2461`): bu üç bölüm haritaların nasıl okunacağını belirlediği için kapatılamaz.
- **Ne zaman, nasıl ayarlanır.** Ek materyal olarak eksiksiz rapor isteniyorsa hepsi açık bırakılır. Ana metne yalnız yöntem ve değerlendirme alınacaksa `Maps`, `Future` ve `Log` kapatılarak dosya küçültülebilir (gömülü PNG'ler ve uzun günlük boyutu büyütür; örnek rapor 5,5 MB'tır).
- **Sonuca etkisi.** Hiçbir hesabı değiştirmez. Bir kutu işaretli olsa da kaynağı yoksa bölüm yazılmaz: örneğin ⑦ Validation çalıştırılmadıysa 9. bölüm, harita PNG'si kaydedilmediyse 6. ve 7. bölümler rapora girmez (örnek raporda 6, 7 ve 9 yoktur).
- **Dikkat.** Kutular değişken seçimini, hangi senaryonun ya da hangi alan değişimi analizinin rapora gireceğini seçmez; oturumdaki **bütün** senaryolar ve alan değişimi analizleri yazılır.

11.4 Generate HTML Report ve Open in Browser

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
⚙ Generate HTML Report	Raporu baştan üretir ve çıktı klasörüne kaydeder	—	—	tabs/tab_report.py: 417-419, :2365-2539
🌐 Open in Browser	Son üretilen raporu sistemin varsayılan tarayıcısında açar	rapor üretilene dek devre dışı	—	:420-421, :2541-2547

Her basışta bütün çıktılar yeniden okunur ve rapor sıfırdan kurulur; durum satırı bunu söyler: `Each click re-reads all outputs and re-generates from scratch.` (`tabs/tab_report.py:449-451`). Dosya adı `<tür>_HABITUS_report_<zaman damgası>.html` olur; her üretim yeni bir dosya yazar, eskileri silinmez (`tabs/tab_report.py:2507-2510`). Çıktı klasörü yoksa bir kaydetme penceresi açılır. İş bitince `Report Ready` penceresi yolu gösterir ve önizleme kutusuna HTML kaynağının ilk 8000 karakteri yazılır. Örnek koşuda rapor `abies_nordmanniana_HABITUS_report_20261121_051313.html` adıyla kaydedilmiştir.

> **İpucu:** Model ya da senaryo yeniden çalıştırıldıysa rapor kendiliğinden güncellenmez; `Generate HTML Report`'a yeniden basılmalıdır.

11.5 HTML raporun bölümleri

Rapor İngilizcedir ve dergiye gönderilecek ek materyal taslağı olarak yazılmıştır; en üstteki not bunu açıkça söyler: `Results and interpretive text should be reviewed and adapted before submission to a scientific journal.` (`tabs/tab_report.py:2405-2409`). Bölümler ve içerikleri aşağıdadır.

11.5.1 Başlık ve 1. Study Summary

Başlıkta tür adı (italik) ve üretim zamanı vardır. 1. bölüm bir tablodur: hedef tür, süzme sonrası kayıt sayısı (ilk yalancı yokluk kümesindeki varlık sayısı), değişken sayısı, algoritmalar, toplam model sayısı, çıktı klasörü ve üretim tarihi (`tabs/tab_report.py:703-725`). Örnek koşuda 143 kayıt, 8 değişken, 4 algoritma.

11.5.2 2. Selected Variables and Multicollinearity Assessment

② Variables'taki tarama yöntemi (VIF ve |r| eşikleri, korelasyon yöntemi) bir paragrafla anlatılır; seçilen değişkenlerden VIF eşiğini aşan varsa adlarıyla bildirilir. `2.1 Variable Screening Table` her aday değişken için VIF, tek değişkenli AUC, VIF ve korelasyon uygunluğunu; `2.2 Final Selected Variables` seçilen değişkenleri listeler. `figures\` klasöründeki korelasyon ısı haritası (seçilenlerinki varsa o) ve VIF çubuk grafiği gömülür (`tabs/tab_report.py:729-878`).

11.5.3 3. Methods

Yöntem bölümü, oturumda gerçekten kullanılan ayarlardan cümle kurar (`tabs/tab_report.py:879-1205`):

- `3.1 Occurrence data`: süzme adımları ve kalan kayıt sayısı.
- `3.2 Environmental predictors`: seçilen değişkenler.
- `3.3 Absence and background data`: yokluk kaynağı (yalancı yokluk, kullanıcı yokluğu ya da ikisi), küme sayısı ve büyüklüğü; erişilebilir alan (tampon/zarf yöntemi ve kilometre değeri ya da kullanıcı poligonu ve kaç biriminin seçildiği; kapsamın yüzdesi); seyreltme mesafesi, kaynağı ve kalan kayıt; aynı hücre süzmesi.
- `3.4 Modelling algorithms`: algoritma açıklamaları ve kaynakları; çapraz doğrulama tasarımı (mekânsal blok, çevresel blok, öteki tasarımlar ya da rastgele) ve tasarımın skorları nasıl etkilediğine dair uyarı; mekânsal sıralama yanlılığı (SSB) ortalaması; birleştirilen kat sayısı.
- `3.5 Ensemble modelling`: iki katmanlı ya da tek katmanlı birleştirme, algoritma başına kopya sayıları, birleştirme kuralı.
- `3.6 Model evaluation`: AUC, TSS, Boyce, kalibrasyon ölçüleri ve yokluk kaynağına göre bir uyarı.
- `3.7 Reproducibility`: rastgele tohum ve `habitus_run_config.json` dosyasına gönderme.

11.5.4 4. Model Parameters

`Global Modelling Settings` tablosu (seçilen değişkenler, küme başına yalancı yokluk, yalancı yokluk stratejisi, küme sayısı, kat sayısı, eğitim/test oranı, permütasyon önem tekrarları) ve `4.1 Algorithm Settings` tablosu (her algoritmanın tam adı, kaynağı, açıklaması ve hiperparametreleri) (`tabs/tab_report.py:1209-1267`).

> **Dikkat:** Bu tablodaki `Pseudo-absences per replicate` ve `PA strategy` satırları, ① Data'da seçilen değerleri değil 1000 ve `random` varsayılanlarını gösterir; doğru değerler 3.3 bölümündeki metinde ve `habitus_run_config.json` dosyasında (`settings.background`) bulunur. `Train / Test split` satırı mekânsal blok tasarımında kullanılmaz; kat düzeni 3.4'te anlatılır.

11.5.5 5. Model Evaluation Results

Önce skorların nasıl okunacağını belirleyen uyarılar (rastgele katlama, mekânsal sıralama yanlılığı vb.) tablodan **önce** verilir (`tabs/tab_report.py:1271-1392`). Ardından: `5.0 Performance and its variability` (kopyalar arası yayılım), `5.1 Individual Model Scores` (her model ve kat için AUC, TSS, Boyce, eşik, yaygınlık, kalibrasyon eğimi ve yorum), `5.1a Ensemble Scores` (topluluğun çapraz doğrulama dışarıda bırakılan katlardaki skorları), `5.1b Threshold-based Metrics and Omission Rates` (Kappa, MCC, duyarlılık, özgüllük, Jaccard, Sørensen, FPB, omisyon oranları, ΔAUC), `5.2 Algorithm Summary` (algoritma ortalaması ± SD) ve `5.3` alt başlıklarında `figures\` klasöründeki değerlendirme, ROC, omisyon/kesinlik-duyarlılık/kalibrasyon, değişken önemi ve yanıt eğrisi grafikleri. Bu çıktıların ayrıntısı Bölüm 7'dedir.

11.5.6 6. Current Distribution

Rapor bugünkü harita GeoTIFF'lerini çizmez; kullanıcının harita görüntüleyiciden **kaydettiği PNG** dosyalarını gömer. Çıktı klasörünün **kökündeki** bütün PNG'ler alınır; adı `habitus_` ile başlayan ve `_study_area.png` ile bitenler hariçtir (`tabs/tab_report.py:1794-1880`). Harita türü dosya adından tahmin edilir (`prob` / `suitability` → uygunluk, `bin` → ikili, `range` / `src` → alan değişimi) ve buna göre hazır bir yorum paragrafı eklenir.

> **Dikkat:** ⑤ ve ⑥'daki görüntüleyicilerin kaydetme penceresi de varsayılan olarak çıktı klasörünün köküne açılır (`map_widget.py:612-620`). Oraya kaydedilen bir gelecek haritası bu bölümde "Current Distribution" başlığı altında görünür. Gelecek haritalarını senaryo alt klasörüne kaydedin (§11.5.7).

11.5.7 7. Future Scenarios

Çıktı klasöründeki her alt klasör (`figures`, `validation` ve `current` dışında) bir dönem sayılır; içinde PNG bulunan her alt klasör için `7.k <klasör adı>` başlığı açılır ve PNG'ler gömülür (`tabs/tab_report.py:1884-1973`). Klasör adı senaryo adıdır (⑤'teki `Output subdir`, boşsa senaryo adı). Hiç PNG yoksa bölüm yazılmaz.

11.5.8 7.1 Environmental Novelty (Extrapolation Detection)

Her projeksiyon (`current` önce, sonra senaryolar alfabetik) için NT1 < 0, NT2 > 1, iki teste göre benzer hücre yüzdesi ve aralık/birleşim yeniliğinde en etkili üç değişken tablolanır; NT2 hesaplanamadıysa nedeni yazılır (`tabs/tab_report.py:2024-2123`). Gelecek bölümü (7) yazılmadıysa başlık `7 Environmental Novelty …` olur. Yöntem Bölüm 8.9'da anlatılmıştır.

7 Environmental Novelty (Extrapolation Detection)

Correlative models return a suitability value everywhere, including where the environment differs from anything seen during calibration. Novelty was quantified with the ExDet method (Mesgaran et al. 2014), which separates two kinds that are routinely conflated.

NT1 marks cells lying outside the calibration range of at least one predictor; this is what a MESS surface detects. **NT2** marks cells in which every predictor is individually within range but the *combination* of values never occurred during calibration. A univariate check reports the second group as safe by construction, and they are not rare: in the worked example of Mesgaran et al., 6,617 of the 10,785 points passing the univariate test showed a distorted correlation structure. The **most influential covariate** columns name the predictors responsible.

Projection	NT1 < 0 (outside range)	NT2 > 1 (novel combination)	Analogous	Most influential (range novelty)	Most influential (combination novelty)
current	23.4 %	0.0 %	76.6 %	bio_3 (43%), bio_15 (24%), bio_2 (13%)	—
SSP585_2090	55.4 %	0.3 %	44.3 %	bio_9 (58%), bio_3 (17%), bio_8 (10%)	bio_3 (93%), bio_8 (3%), bio_19 (2%)

Percentages are shares of all projected cells and sum to 100 for each row. Suitability values in cells flagged by either test describe an extrapolation, not a prediction, and should not be interpreted as habitat suitability without further justification. Per-cell surfaces are written beside the projections as `<scenario>_NT1.tif`, `_NT2.tif` and the two `_MIC_` rasters, whose pixel values index the predictor list carried in the raster tags.

7.2 Processing Notices

The following conditions were detected while preparing inputs or producing maps. They do not stop the analysis but affect how the results should be read.

- **Sampling bias**: Presence records are used as collected; no target-group or bias surface was supplied, so sampling bias (uneven survey effort, e.g. near roads or settlements) is not corrected. Suitability then partly describes where the species was recorded, not only where it can live. Supply a target-group CSV or a bias raster in the Data tab if effort is uneven.

Şekil 11.3. Örnek HTML raporun yenilik kesiti: Environmental Novelty tablosu (current ve SSP585_2090 için NT1 < 0, NT2 > 1, benzer hücre oranı ve en etkili değişkenler) ve 7.2 Processing Notices listesindeki örnekleme yanlılığı notu.

11.5.9 7.2 Processing Notices (işleme notları)

Haritaların okunuşunu değiştiren ama işi durdurmayan durumlar burada maddelenir; yoksa bölüm yazılmaz (`tabs/tab_report.py:1981-2022`):

- projeksiyonlarda eğitim ızgarasına hizalanan katmanlar ve örtülmeyen hücre uyarıları (§8.10);
- alan değişiminde hizalanan gelecek haritası;
- örnekleme yanlılığı uyarısı (hedef grup ya da yanlılık yüzeyi verilmediyse; örnek raporda tek not budur);
- ayar araması (tuning) yapıldıysa seçim yanlılığı notu;
- topluluk haritası üretilemediyse nedeni;
- veri girişi ve doğrulama notları.

11.5.10 8. Range Change Analysis

Her alan değişimi analizi için bir alt başlık, ⑥'daki tabloyla aynı satırlar (kapsam, bugünkü ve gelecek alan, yüzdeler, hücre sayıları, öteki kapsamların net değişimi, dışdeğerlenmiş değişim payı), değişen hücrelerin %25'i ya da fazlası dışdeğerlenmiş koşullardaysa bir not, net değişime göre üç düzeyli bir yorum (−10'un altı daralma, +10'un üstü genişleme, arası kararlılık) ve `_SRC.tif` dosyasından yeniden çizilmiş renkli harita (`tabs/tab_report.py:2126-2265`).

8. Range Change Analysis

Species range change was quantified by comparing binary current and future distribution maps pixel by pixel. Each cell was classified into one of four categories: **Lost** (predicted present now, absent in future), **Stable Present**, **Stable Absent**, and **Gained** (absent now, present in future).

8. SSP585_2090

Metric	Value
Counted within	**whole study boundary**
Current range (cells)	**7102**
Future range (cells)	**3826**
Lost (%)	**56.84**
Gained (%)	**10.72**
Net change (%)	**-46.13**
Stable Present (cells)	**3065**
Stable Absent (cells)	**69308**
Lost (cells)	**4037**
Gained (cells)	**761**
Net change, accessible area (%)	**-30.25**
Net change, accessible area without extrapolation (%)	**-16.57**
Changed cells in extrapolated conditions (%)	**58.6**

59% of the cells that change lie in conditions outside the calibration range of at least one predictor (ExDet NT1 < 0). The change there is extrapolation; the change within the accessible area without extrapolation is the better supported estimate.

> Under the **SSP585_2090** scenario, the analysis indicates a projected range contraction of -46.1%. Lost habitat (56.8%) represents areas where current suitable conditions are projected to disappear, posing a climate risk. Gained habitat (10.7%) represents newly suitable areas that may serve as refugia or colonisation targets.

Şekil 11.4. Örnek HTML raporun 8. Range Change Analysis bölümü: SSP585_2090 tablosu, değişen hücrelerin %59'unun dışdeğerlenmiş koşullarda olduğunu söyleyen not ve net değişime göre kurulan yorum kutusu (harita bu kesitte gösterilmemiştir).

11.5.11 9. Validation

⑦ Validation'da bir doğrulama yapıldıysa: kip, başvuru ve doğrulama katmanları, nokta sayısı, genel doğruluk, Cohen κ, ağırlıklı F1 (varsa R^2, RMSE, MAE), karışıklık matrisi ve `validation\reclassify*.tif` dosyalarından en çok iki harita (`tabs/tab_report.py:2269-2359`). Ayrıntı Bölüm 10'dadır.

11.5.12 10. Session Log ve atıf

Çıktı klasöründeki `.log` dosyalarından adı en büyük olan (en yeni oturum) kaydırılabilir bir kutuya olduğu gibi yazılır (`tabs/tab_report.py:606-614`, `:2476-2481`). Raporun sonunda HABITUS için önerilen atıf bulunur (`tabs/tab_report.py:2484-2495`).

> **Not:** Günlük her oturumda yeni bir dosyaya yazılır. Program aynı çıktı klasöründe kapatılıp yeniden açıldıysa rapora yalnız son oturumun günlüğü girer; önceki oturumun kayıtları kendi `.log` dosyasında kalır.

11.6 ODMAP dışa aktarımı (Export ODMAP protocol)

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Export ODMAP protocol`	`odmap_protocol.md` ve `odmap_protocol.csv` dosyalarını çıktı klasörüne yazar	—	—	`tabs/tab_report.py:422-430`, `:537-570`

- **Ne işe yarar.** ODMAP (Overview, Data, Model, Assessment, Prediction), tür dağılım modellerinin raporlanması için önerilen standart protokoldür (Zurell vd. 2020). Düğme, protokolün Tablo 1'deki 79 öğesini satır satır kurar; `habitus_run_config.json` dosyasında ve oturumda kaydı bulunan öğeleri doldurur (`auto`), geri kalanları `user_required` olarak işaretler (`odmap.py:42-111`, `:156`, `:637-686`). Hiçbir değer uydurulmaz; ipucu metni de raporun kullanıcı bu satırları doldurana dek ODMAP uyumlu sayılmayacağını söyler.
- **Ne zaman, nasıl ayarlanır.** Makale gönderiminden önce; dosya yalnız düğmeye basınca yazılır, kendiliğinden yazılmaz (`tabs/tab_report.py:538-540`). Önce modeller ve projeksiyonlar tamamlanmalı, sonra dışa aktarılmalıdır.
- **Sonuca etkisi.** Hesabı etkilemez. Örnek koşuda 79 öğenin 46'sı otomatik doldurulmuş, 33'ü kullanıcıya bırakılmıştır. Durum satırı ve günlük şu iletiyi yazar: `ODMAP protocol written (79 elements: 46 filled automatically, 33 to be completed by you): …`.
- **Dikkat.** Çıktı klasörü yoksa `No output folder yet. Run the models first.` iletisi çıkar. `habitus_run_config.json` bulunamazsa bütün öğeler `user_required` olur ve iletiye bu not eklenir (`tabs/tab_report.py:552-564`).

Otomatik doldurulan başlıca öğeler (örnek koşudan): hedef çıktı (algoritmalar, ikili harita kuralı), tür adı, konum ve ızgara (satır × sütun, hücre boyu, koordinat sistemi), kapsam sınırının türü (erişilebilir alan tamponu), yanıt türü (varlık-yalancı yokluk), değişkenler, algoritmalar ve topluluk yöntemi, adımların özeti, yazılım ve paket sürümleri, örneklem büyüklüğü ve süzme gerekçeleri, yalancı yokluk ayarları ve tohum, eğitim/doğrulama bölümlemesi (blok boyu dahil), değişken ön seçimi, algoritma ayarları, aralık dışı tahmin ayarı (clamping), değişken önemi, topluluk ağırlıkları, mekânsal özilişkinin nasıl ele alındığı, eşik kuralı, performans istatistikleri, tahmin birimi, belirsizlik haritaları, aktarılan senaryolar ve yenilik ölçümü (ExDet; seçildiyse MESS ve maskesi).

Kullanıcıya bırakılan öğeler: yazarlar, iletişim, başlık, DOI; çalışmanın amacı; zaman kapsamı ve çözünürlüğü; gözlem türü; hipotezler ve varsayımlar; model karmaşıklığının gerekçesi; kod ve veri erişilebilirliği; taksonomik başvuru sistemi, ekolojik düzey, veri kaynakları, örnekleme tasarımı; bağımsız test verisi; değişkenlerin ve aktarım katmanlarının kaynakları, zaman ve mekân çözünürlükleri, ölçüm hataları; model katsayıları; zamansal özilişki ve iç içe veri; yanıt eğrileri ve uzman değerlendirmesinin sonucu; girdi verisi ve parametre belirsizliği. Çoklu doğrusallık öğesi, ② Variables'taki eşikler oturumda okunabildiğinde otomatik doldurulur (`odmap.py:487-495`).

11.7 Yayın haritaları

HABITUS haritaları ekrandaki görüntüleyicinin ekran görüntüsü olarak değil, dergi sütun genişliğinde yeniden çizilerek kaydedilir (`map_widget.py:597-668`). Bu kayıt, ③, ⑤ ve ⑥'daki harita görüntüleyicilerinin araç çubuğundaki disket simgesiyle; çalışma alanı haritası ise 🗺 Map sekmesindeki 💾 `Save study-area figure…` düğmesiyle yapılır (`tabs/tab_map.py:124-130`, `study_area_figure.py:293-313`). Her ikisi de aynı `Figure for publication` penceresini açar (`map_widget.py:180-238`).

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Width:`	Şeklin basılı genişliği	`Double column — 174 mm`	`Single column — 85 mm`, `1.5 column — 120 mm`, `Double column — 174 mm`	`map_widget.py: 112-116`, `:306`
`Resolution:`	Raster biçimlerde çözünürlük	`600 dpi`	`300 dpi`, `600 dpi`, `1200 dpi`	`map_widget.py: 117`, `:197-203`
`Title:`	Başlık metni (yalnız çalışma alanı haritasında)	tür adı	serbest metin	`map_widget.py: 209-212`
`Include the title above the map`	Başlığı haritanın üstüne yazar	açık (sonuç haritaları), kapalı (çalışma alanı)	açık / kapalı	`map_widget.py: 204-206`, `:306`, `tabs/tab_map.p y:139`
`Italic title (scientific name)`	Başlığı italik yazar (çalışma alanı)	tür adı varsa açık	açık / kapalı	`tabs/tab_map.p y:289`
`Locator map in the upper-left corner`	Konum göstergesi iç haritası (çalışma alanı)	açık	açık / kapalı	`tabs/tab_map.p y:290`
dosya türü	Kaydetme penceresindeki biçim	PNG	PNG, TIFF (LZW), PDF, SVG, EPS, JPEG	`map_widget.py: 156-177`

Basılı görünüm Arial 7–9 pt, beyaz arka plan ve ince çizgilerle çizilir; PDF/SVG/EPS'de yazı tipleri gömülür (`map_widget.py:126-131`, `:149-153`). Çözünürlük vektör biçimlerde (PDF, SVG, EPS) kullanılmaz. Haritanın yüksekliği coğrafi en-boy oranından (enlem kosinüs düzeltmesiyle) hesaplanır, en çok 9 inç ile sınırlanır (`map_widget.py:652-662`). Çalışma alanı haritası, pakete gömülü Natural Earth ülke sınırları üzerine kayıtları, sınırı, ölçek çubuğunu ve kuzey okunu çizer (`study_area_figure.py:1-10`). Görüntüleyicinin öteki denetimleri Bölüm 12'de anlatılır.

> **İpucu:** Raporun 6. ve 7. bölümleri yalnız PNG dosyalarını gömer. Dergi için PDF/TIFF kaydedilecekse aynı haritayı bir kez de PNG olarak (bugünkü harita çıktı klasörünün köküne, gelecek haritası senaryo alt klasörüne)

> kaydedin. Rapor metni bu haritalar için "300 DPI" yazar; kaydetme penceresinin varsayılanı 600 dpi'dır.

11.8 Çok türlü toplu çalıştırma (Run Selected Species…)

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Run Selected Species…`	Tamamlanan iş akışını ① Data'da işaretli öteki türler için aynı ayarlarla yineler	—	onay penceresi	`tabs/tab_report.py:431-438`, `:469-512`, `batch_species.py`

- **Ne işe yarar.** Birden çok tür içeren bir kayıt dosyasında, bir tür için iş akışı elle tamamlandıktan sonra, ① Data'da işaretli **öteki** türler sırayla işlenir. Her tür için yinelenen adımlar (`batch_species.py:145-159`): 1. ① Data: etkin tür değiştirilir ve veri hazırlığı aynı ayarlarla çalıştırılır; 2. ② Variables: ilk türde seçilen değişkenler (ve kategorik seçim) aynen atanır; bu türün kayıt sayısına göre değişken sayısı uyarısı günlüğe yazılır (`batch_species.py:201-224`); 3. ③ Models: sekmedeki ayarlar olduğu gibi kullanılarak modeller kurulur (bugünkü harita dahil); 4. ⑤ her senaryo: aynı raster listesi, algoritmalar, alt klasör, MESS, maske ve clamping seçimiyle (`batch_species.py:226-239`); 5. ⑥ her alan değişimi analizi: listeden seçilmiş anahtarlar (ör. `current :: EMwmean_bin`) bu türün dosyalarına çevrilerek, aynı analiz adı ve kapsamla (`batch_species.py:241-259`); 6. ⑧ HTML raporu.

Tür başına çıktı klasörü `<çıktı klasörü>\<tür>` biçimindedir (`tabs/tab_data.py:1115-1124`). - **Ne zaman, nasıl ayarlanır.** Önce bir tür için iş akışı eksiksiz ve dikkatle yapılır; ⑤ ve ⑥'da raporlanacak her analiz bir kez çalıştırılmış olmalıdır, çünkü toplu çalıştırma yalnız oturumda çalıştırılmış olanları yineler. Onay penceresi tekrarlanacak türleri, değişkenleri, senaryo ve alan değişimi sayısını listeler (`tabs/tab_report.py:489-499`). - **Sonuca etkisi.** Her tür elle yapılsaydı alacağı sonucu alır; program ayrı bir hat kullanmaz, sekmelerin kendisini sürer (`batch_species.py:12-15`). Değişken seçimi ise türe göre yeniden yapılmaz: ilk türün seçimi bütün türlere uygulanır. - **Dikkat.** Çalışma sırasında açılan bütün ileti pencereleri kendiliğinden yanıtlanır (soru pencerelerine `Yes`) ve günlüğe `[batch, …]` önekiyle yazılır (`batch_species.py:80-111`). Bir adım başarısız olursa o türün kalan adımları atlanır ve sıradaki türe geçilir. İş bitince `<çıktı klasörü>\batch_summary.csv` yazılır: `species`, `status` (`done` / `failed`), `failed_step`, `error` (hatanın kullanıcıya dönük son satırı), `output_dir`, `minutes` (`batch_species.py:263-302`). Bu dosya başka bir programda açıksa yazılamaz ve günlüğe not düşülür.

Toplu çalıştırmada **yinelenmeyenler**: ayar araması (tuning) adımı, ⑦ Validation, ODMAP dışa aktarımı ve harita PNG'lerinin kaydedilmesi. Bu yüzden toplu çalıştırmadaki türlerin raporlarında 6., 7. ve 9. bölümler bulunmaz. Ayrıca arayüzde toplu çalıştırmayı yarıda kesecek bir düğme yoktur; iş, bütün türler bitene dek sürer.

İletiler: tür işaretlenmemişse `The batch run is for an occurrence file with several species: tick the species to analyse in ① Data.`; iş akışı tamamlanmamışsa `Complete the workflow for one species first (at least ① Data, ② Variables and ③ Models); …`; işaretli tek tür zaten işlenmiş olansa `No other species is ticked in ① Data.` (`tabs/tab_report.py:474-487`).

11.9 Çıktılar

- `<tür>_HABITUS_report_<zaman>.html`: tek dosyalık, görüntüleri gömülü rapor.
- `odmap_protocol.md`, `odmap_protocol.csv`: kısmen doldurulmuş ODMAP (CSV sütunları `section, subsection, element, value, source`).
- Toplu çalıştırmada: `<tür>\` alt klasörleri ve `batch_summary.csv`.
- Kaydedilen yayın haritaları: kullanıcının seçtiği yolda.

11.10 Uzman soruları

S1. Rapordaki yöntem metni doğrudan makaleye konabilir mi? Cevap: Rapor kendisi bunu önermez: başındaki not metnin gözden geçirilip uyarlanması gerektiğini söyler (`tabs/tab_report.py:2405-2409`). Metin oturumdaki gerçek ayarlardan kurulur (ör. erişilebilir alan, seyreltme, katlama tasarımı ve SSB, `tabs/tab_report.py:930-1092`), ama ilke cümleleri ve atıflar genel kalıplardır; 4. bölümdeki yalancı yokluk satırları da varsayılan değerleri gösterir (§11.5.4). Sayılar `habitus_run_config.json` ile karşılaştırılmalıdır.

S2. ODMAP çıktısı "ODMAP uyumlu" sayılır mı? Cevap: Hayır. Modül yalnız Zurell vd. (2020) Tablo 1'deki öğeleri kullanır; Ek Tablo A1'deki tam soru listesi kullanılmamıştır ve dosya bunu açıkça yazar (`odmap.py:11-21, :661-664`). Örnek koşuda 33 öğe kullanıcı tarafından doldurulmalıdır. ODMAP Shiny uygulamasındaki form ile karşılaştırılarak tamamlanması önerilir.

S3. Raporda MESS ya da clamping seçimi görünüyor mu? Cevap: HTML raporda görünmez; yenilik bölümü yalnız ExDet sonuçlarını tablolar (`tabs/tab_report.py:2024-2123`). Seçimler ODMAP'ta ("Settings for extrapolation beyond sample range", "Quantification of novel environments", `odmap.py:466-508`) ve her senaryo klasöründeki `_projection_options.json` dosyasında bulunur. Makalede bu seçimlerin yöntem kısmında yazılması gerekir.

S4. Rapordaki ExDet yüzdelerinin paydası nedir? Cevap: Projeksiyonda tahmin edilen (bütün değişkenleri sonlu) hücrelerin tamamı; NT1 < 0, NT2 > 1 ve benzer yüzdeleri toplamı 100'dür (`sdm_novelty.py:100-125`, rapor notu `tabs/tab_report.py:2109-2118`). NT2 yüzdesi de NT2'nin hesaplandığı alt kümeye değil bütün hücrelere bölünür. Örnek koşuda SSP585_2090 için 55,4 + 0,3 + 44,3 = 100.

S5. Toplu çalıştırmada her tür için değişken seçimi yeniden yapılıyor mu? Cevap: Hayır; ilk türde ② Variables'ta onaylanan değişkenler aynen atanır (`batch_species.py:201-218`). Program yalnız bu türün kayıt sayısının bu kadar değişkene yetip yetmediğini günlüğe yazar (`note_selected_predictors`, `batch_species.py:219-224`). Türlerin ekolojisi farklıysa ya da kayıt sayıları çok farklıysa değişken seçimi türe göre elle yapılmalı ve türler ayrı çalıştırılmalıdır. biomod2'de toplu çalışma kullanıcının yazdığı döngüyle yapılır; HABITUS'taki karşılığı aynı ayarların bütün türlere uygulanmasıdır.

S6. Toplu çalıştırmada bir tür başarısız olursa diğerleri etkilenir mi? Cevap: Hayır. Başarısız adım kaydedilir, o türün kalan adımları atlanır, sıradaki türe geçilir (`batch_species.py:161-190, :263-274`). `batch_summary.csv` hangi adımda ve neden başarısız olduğunu yazar; hata metninden kullanıcıya dönük son satır alınır (`batch_species.py:26-51`). Tam hata izi günlükte `[batch, trace]` önekiyle kalır.

S7. Yayın haritalarında hangi çözünürlük seçilmeli? Cevap: Pencere ipucu renkli haritalar için genellikle 300 dpi, ince yazı ve çizgi içeren şekiller için 600 dpi ve üstünü önerir (`map_widget.py:201-203`); varsayılan 600 dpi'dır. Vektör biçimlerde (PDF, SVG, EPS) dpi kullanılmaz, raster katman yine görüntü olarak gömülür. Sütun genişlikleri Elsevier, Wiley ve Springer'in yaygın ölçüleridir (85, 120, 174 mm; `map_widget.py:111-116`); derginin kendi ölçüsü farklıysa en yakın genişlik seçilip dergi yönergesine göre ölçeklenmelidir.

S8. Rapordaki harita yorum paragrafları nereden geliyor? Cevap: Dosya adına göre seçilen sabit metinlerdir (`tabs/tab_report.py:1824-1865, :1935-1958`): örneğin olasılık haritası için yüksek uygunluk

alanlarının arazi çalışmasına öncelik verilmesi önerisi, gelecek haritası için ısınmayla kuzeye/yukarıya kayma cümlesi. Bu cümleler verinin kendisinden çıkarılmaz; makaleye alınmadan önce bulgulara göre yeniden yazılmalıdır. Alan değişimi yorumu ise net değişimin gerçek değerine (±%10 eşikleri) göre kurulur (`tabs/tab_report.py:2184-2196`).

S9. Rapor yeniden üretilebilir mi; hangi dosyalar yeterli? Cevap: Rapor her basışta oturumdaki sonuçlardan sıfırdan üretilir; oturum kapandıktan sonra aynı rapor yeniden üretilemez, çünkü modeller ve ExDet özetleri bellektedir. Yeniden üretilebilirlik için `habitus_run_config.json` (tüm ayarlar, tohum, paket sürümleri), senaryo klasörlerindeki `_projection_options.json` ve kaydedilen HTML rapor birlikte saklanmalıdır. 3.7 bölümü tohumu ve yapılandırma dosyasını yazar (`tabs/tab_report.py:1138-1151`).

S10. Raporda kaç senaryo ve kaç alan değişimi analizi görünür? Cevap: Oturumda yapılan hepsi. Yenilik tablosu `state["novelty"]`'deki her projeksiyonu, 8. bölüm `state["range_results"]`'taki her analizi yazar (`tabs/tab_report.py:2044-2073, :2140-2207`). Rapora girmesi istenmeyen bir deneme varsa ⑤'teki `Remove All` ile sonuçlar sıfırlanıp yalnız istenenler yeniden çalıştırılmalıdır.

11.11 Sık hatalar ve tuzaklar

- **Haritaları PNG olarak kaydetmemek.** Rapor GeoTIFF'lerden bugünkü ya da gelecek haritası çizmez; PNG kaydedilmediyse 6. ve 7. bölümler hiç çıkmaz.
- **Gelecek haritasını çıktı klasörünün köküne kaydetmek.** Raporda "6. Current Distribution" altında görünür.
- **4. bölümdeki yalancı yokluk satırlarına güvenmek.** Doğru değerler 3.3 bölümünde ve `habitus_run_config.json` dosyasındadır.
- **ODMAP dosyasını eksiksiz sanmak.** `user_required` satırları doldurulmadan dosya bir ODMAP raporu değildir.
- **Toplu çalıştırmayı tek türde tamamlanmamış iş akışıyla başlatmak.** Yalnız oturumda çalıştırılmış senaryolar ve alan değişimi analizleri yinelenir; ⑤'te hiç senaryo çalıştırılmadıysa toplu çalıştırmada da gelecek haritası üretilmez.
- **Toplu çalıştırmada değişken uyarısını gözden kaçırmak.** Kayıt sayısı az olan türlerde `[batch] ADVISORY:` satırları günlüğe düşer; değişken sayısı o tür için fazla olabilir.
- **Raporu yeniden üretmeyi unutmak.** Yeni bir senaryo ya da yeniden kurulan modeller rapora ancak `Generate HTML Report`'a yeniden basılınca girer; eski dosya silinmez, yeni zaman damgalı bir dosya yazılır.

11.12 Kaynaklar

Zurell, D., Franklin, J., König, C., Bouchet, P.J., Dormann, C.F., Elith, J., Fandos, G., Feng, X., Guillera-Arroita, G., Guisan, A., Lahoz-Monfort, J.J., Leitão, P.J., Park, D.S., Peterson, A.T., Rapacciuolo, G., Schmatz, D.R., Schröder, B., Serra-Diaz, J.M., Thuiller, W., Yates, K.L., Zimmermann, N.E., Merow, C., 2020. A standard protocol for reporting species distribution models. Ecography 43, 1261–1277. https://doi.org/10.1111/ecog.04960

12. Harita, Yardım ve çıktı dosyaları (🗺 Map, ⓘ Help)

Sekmeler: 🗺 Map, ⓘ Help · Kod: `tabs/tab_map.py`, `occurrence_map.py`, `study_area_figure.py`, `map_widget.py` (şekil seçenekleri), `tabs/tab_help.py`, `run_config.py`, `main_dialog.py` (oturum günlüğü), `main.py` (çökme günlüğü)

12.1 Bu bölüm ne anlatır

Bu bölüm iş akışının dışında duran iki sekmeyi ve bütün iş akışının ortak konularını anlatır:

- `🗺 Map` (harita) sekmesi: girdilerin çevrimiçi altlık harita üzerinde gösterimi ve yayın için çalışma alanı şekli (§12.3).
- `ⓘ Help` (yardım) sekmesi: programın içindeki İngilizce rehber ve güncelleme denetimi (§12.4).
- Çıktı klasörünün tam dosya kataloğu (§12.5), koşu kaydı `habitus_run_config.json` (§12.6), oturum ve çökme günlükleri (§12.7), ortam değişkenleri (§12.8), sorun giderme (§12.9) ve sistem gereksinimleri (§12.10).

İki sekme de kilitli değildir; program açıldığı andan itibaren kullanılabilir (`main_dialog.py:194-196`). Sekme şeridinde numaralı adımlardan sonra gelirler ve iş akışına girdi vermezler.

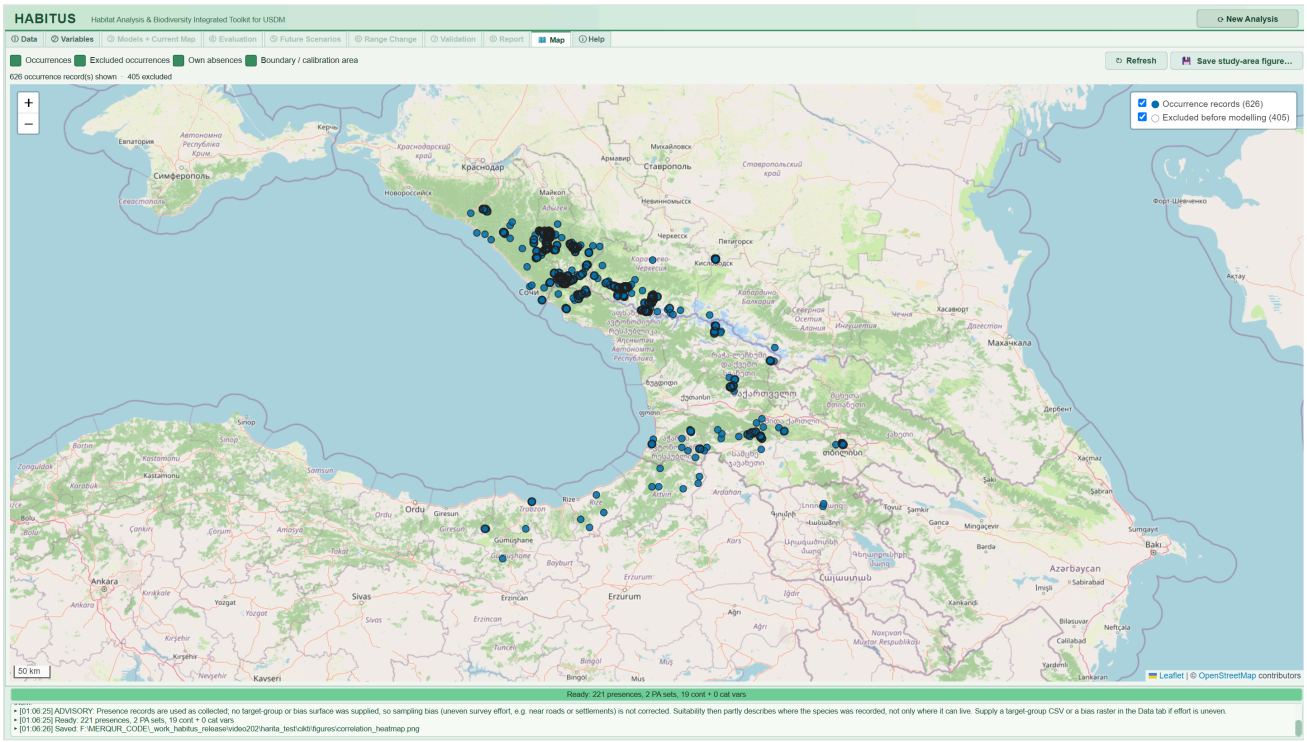
12.2 🗺 Map sekmesinin yeri

`🗺 Map`, herhangi bir model kurulmadan önce `① Data`'ya verilenleri harita üzerinde gösterir: sunum kayıtları, `① Data` çalıştırıldıktan sonra modellemeden önce dışarıda bırakılan kayıtlar, kullanıcının kendi yokluk kayıtları, çalışma sınırı ve kalibrasyon poligonu. Amacı yanlış yere düşmüş bir kaydı, ters yazılmış koordinatları ya da bölgeyle örtüşmeyen bir sınır katmanını modellemeden önce görmektir (`tabs/tab_map.py:1-11`).

Sekme her görünür olduğunda girdiler değişmişse haritayı yeniden kurar (`tabs/tab_map.py:141-169`). Model sonuçları burada gösterilmez; güncel ve gelecek haritaları `③`, `⑤` ve `⑥` sekmelerindeki harita görüntüleyicisindedir (Bölüm 6, 8, 9).

12.3 🗺 Map sekmesi

Ekranın üstünde katman kutuları ve iki düğme, altında bir durum satırı, onun altında harita bulunur (`tabs/tab_map.py:100-139`). Harita bir web görünümü içinde Leaflet kütüphanesiyle çizilir; Leaflet programla birlikte gelir, altlık döşemeleri ise `tile.openstreetmap.org` adresinden indirilir (`occurrence_map.py:45-50`, `92-95`).



Şekil 12.1. 🗺 Map sekmesi, örnek koşuda ① Data çalıştırıldıktan sonra: üstte dört katman kutusu, sağda ↻ Refresh ve 💾 Save study-area figure...; altında "626 occurrence record(s) shown · 405 excluded" durum satırı; OpenStreetMap altlığı üzerinde mavi sunum kayıtları ve içi boş gri dairelerle dışlanan kayıtlar; sağ üstte katman listesi, sol altta ölçek çubuğu.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Occurrences`	CSV'deki sunum kayıtlarını gösterir; çok türlü dosyada işaretli her tür ayrı renkte	açık	açık / kapalı	`tabs/tab_map.py:108`, `179-214`
`Excluded occurrences`	`① Data` çalıştırıldıktan sonra modellemeden önce dışlanan kayıtları gerekçesiyle gösterir	açık	açık / kapalı	`tabs/tab_map.py:109-113`, `216-225`
`Own absences`	`Own Absences` alt sekmesinde verilen yokluk CSV'sini siyah noktalarla gösterir	açık	açık / kapalı	`tabs/tab_map.py:114`, `227-243`
`Boundary / calibration area`	Çalışma sınırını (kırmızı) ve kalibrasyon poligonunu (mor, kesikli) çizer	açık	açık / kapalı	`tabs/tab_map.py:115`, `245-263`
`↻ Refresh`	Haritayı girdilerden yeniden kurar	—	düğme	`tabs/tab_map.py:120-122`
`💾 Save study-area figure...`	İşaretli katmanlardan yayın için bir konum haritası kaydeder	—	düğme; §12.3.2	`tabs/tab_map.py:123-130`, `268-311`

Kutulardan biri değiştiğinde harita hemen yeniden çizilir (`tabs/tab_map.py:116-117`). Bu kutular yalnız görüntüyü ve kaydedilen şekli etkiler; modellemeye giren veriyi değiştirmez.

12.3.1 Katmanların içeriği

- **Sunum kayıtları.** Sütunlar `① Data`'nın sezdiği adlarla, sezilmediyse ilk üç sütun olarak okunur. Tek türlü dosyada katmanın adı "Occurrence records"tır; çok türlü dosyada yalnız `① Data`'da işaretlenmiş türler gösterilir. Renkler renk körlüğüne uygun bir listeden alınır. Her noktanın açılır penceresi tür adını ve koordinatları verir (`tabs/tab_map.py:179-214`, `occurrence_map.py:40-42`).
- **Dışlanan kayıtlar.** `① Data` çalıştırılmadan bu katman boştur. Gerekçeler: "duplicate coordinate", "same raster cell as another record" (`Keep one record per raster cell` açıkken), "outside raster extent", "falls on NoData cell", "missing coordinate", "spatially thinned" ve yokluk dosyası için "absence record outside extent or on NoData" (`sdm_core.py:1853`, `1875`, `1899`, `1935`, `2779`, `2949`, `2966`). Örnek koşuda 626 kaydın 405'i aynı hücrede başka bir kayıt bulunduğu için dışlanmıştır (Şekil 12.1).
- **Koordinat dönüşümü.** Rasterlar projeksiyonlu bir koordinat sistemindeyse kayıtlar haritada gösterilmek için boylam-enleme dönüştürülür; dönüştürülemeyen tek bir kayıt atılır, ötekiler etkilenmez (`tabs/tab_map.py:35-71`). `① Data` çalıştırılmadan rasterların koordinat sistemi bilinmez; o zaman metre cinsinden koordinatlar için durum satırı "coordinates are not longitude/latitude in degrees and the rasters' CRS is not known yet: run ① Data first to place them correctly" uyarısını verir (`tabs/tab_map.py:196-199`).
- **Seyreltme.** Bir katmanda 20.000'den fazla nokta varsa yalnız görüntü için her k'ıncı nokta çizilir; katman listesindeki sayı çizilen nokta sayısıdır (`tabs/tab_map.py:22-24`, `313-326`).
- **Poligonlar.** Çalışma sınırı poligon katmanı ya da köşe koordinatlarıyla verilmiş bir dikdörtgen olabilir. Kalibrasyon poligonu yalnız `Accessible area` `polygon` seçildiğinde çizilir. 5000'den çok köşeli bir geometri görüntü için sadeleştirilir (`tabs/tab_map.py:74-95`, `245-263`).

Durum satırı gösterilen ve dışlanan kayıt sayısını, geçerli koordinatı olmayan kayıtları ve çizilemeyen katmanları yazar. Okunamayan bir CSV için harita yerine "The map could not be built: …" iletisi çıkar (`tabs/tab_map.py:154-169`).

> **Not:** Altlık döşemeleri ağ bağlantısı ister. Bağlantı yoksa noktalar ve poligonlar yine çizilir, altlık boş kalır. Program kendini OpenStreetMap döşeme sunucusuna "HABITUS/<sürüm>" kimliğiyle tanıtır (`occurrence_map.py:7-11, 53-58`). Web görünümü bileşeni (PyQt6-WebEngine) kurulu değilse sekme "The map needs PyQt6-WebEngine, which is not installed (pip install PyQt6-WebEngine)." iletisini gösterir (`occurrence_map.py:138-145`).

12.3.2 `Save study-area figure…` (çalışma alanı şekli)

Bu düğme, haritadaki işaretli katmanları web görünümünden bağımsız olarak, bir makaleye konabilecek bir konum haritası olarak çizer (`study_area_figure.py:1-10`). Ekrandaki OpenStreetMap döşemeleri ekran çözünürlüğündedir ve kendi lisans koşullarını taşır; kaydedilen şekil bunun yerine programla birlikte gelen Natural Earth ülke sınırlarını (kamu malı) kullanır ve ağ gerektirmez.

Düğmeye basınca önce "Figure for publication" seçenek penceresi, ardından kayıt penceresi açılır (`tabs/tab_map.py:268-311`).

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Width:`	Şeklin genişliği (dergi sütun genişlikleri)	`Double column — 174 mm`	`Single column — 85 mm`, `1.5 column — 120 mm`, `Double column — 174 mm`	`map_widget.py:112-116`, `193-196`; `tabs/tab_map.py:139`
`Resolution:`	Raster biçimlerde (PNG, TIFF, JPEG) çözünürlük	`600 dpi`	300, 600, 1200 dpi; PDF, SVG, EPS'de kullanılmaz	`map_widget.py:117`, `197-203`
`Title:`	Başlık metni	işaretli türlerin adları, yoksa tür adı	serbest metin	`tabs/tab_map.py:283-288`, `map_widget.py:209-212`
`Include the title above the map`	Başlığı haritanın üstüne yazar	kapalı	açık / kapalı	`tabs/tab_map.py:139`, `map_widget.py:204-206`
`Italic title (scientific name)`	Başlığı italik yazar	tür adı varsa açık	açık / kapalı	`tabs/tab_map.py:289`
`Locator map in the upper-left corner`	Sol üst köşeye çalışma kapsamını gösteren küçük bir konum haritası ekler	açık	açık / kapalı	`tabs/tab_map.py:290`, `study_area_figure.py:265-290`
Kayıt penceresinde dosya türü	Biçim, dosya uzantısından belirlenir	PNG	PNG, TIFF (LZW sıkıştırmalı), PDF, SVG, EPS (vektör), JPEG (kalite 95)	`map_widget.py:156-177`

Önerilen dosya adı çıktı klasöründe `<tür>_study_area.png`'dir (`tabs/tab_map.py:294-300`). Pencere son seçimleri oturum boyunca hatırlar (`tabs/tab_map.py:293`).

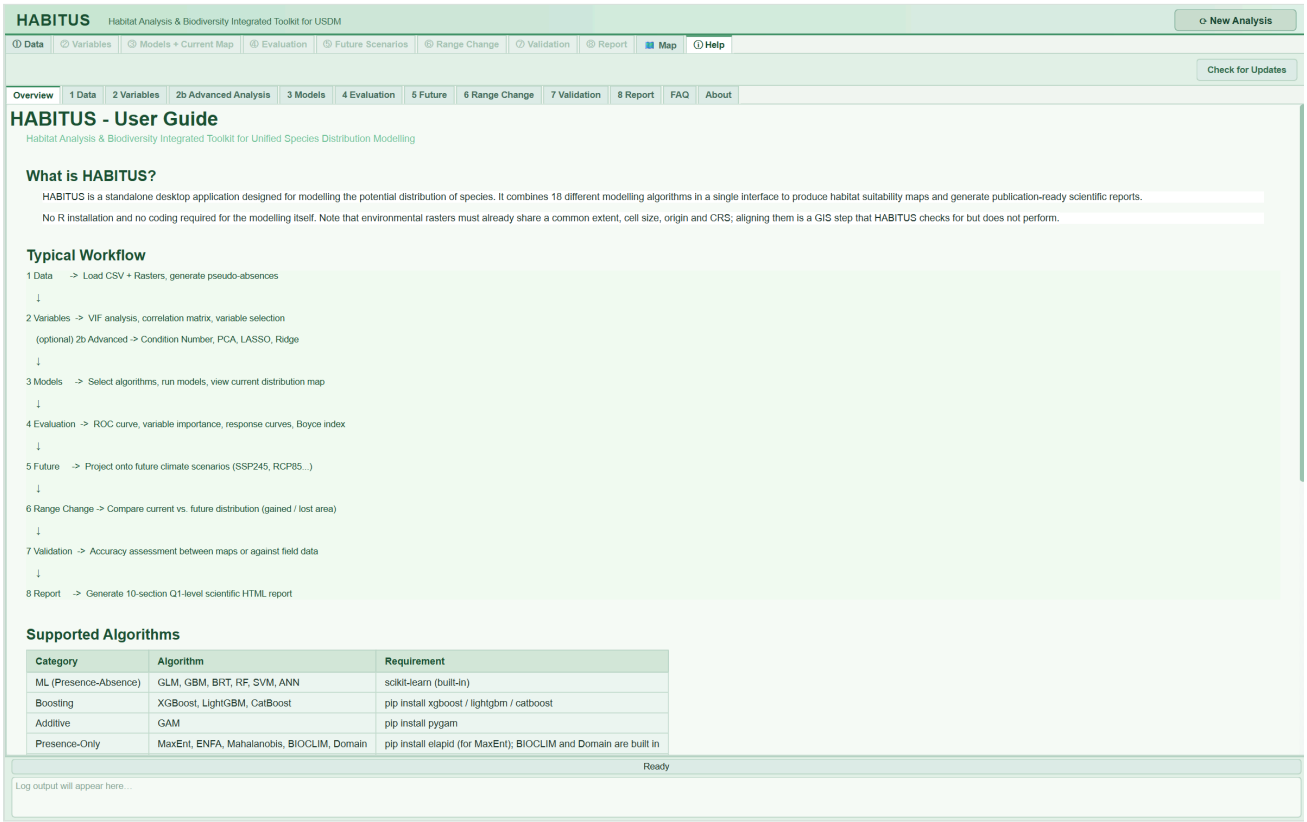
Şekil şu öğelerden oluşur (`study_area_figure.py:190-262`, `293-313`):

- **Kapsam.** Çizilen her şeyin çevresinde %8 pay; genişlik ya da yükseklik 0,5 dereceden azsa 0,5 dereceye genişletilir.
- **Zemin.** Natural Earth ülkeleri; kapsamın uzun kenarı 20 dereceden azsa 1:10 milyon, değilse 1:50 milyon ölçekli sınırlar. Deniz açık mavi, kara açık gridir.
- **Katmanlar.** Sınır ve kalibrasyon poligonları, sunum kayıtları, dışlanan kayıtlar (içi boş) ve kendi yokluklar; lejantta her katmanın adı ve "(n = <kayıt sayısı>)".
- **Harita öğeleri.** Derece etiketli eksenler, kesikli ızgara, ölçek çubuğu, kuzey oku; enlem ortasının kosinüsüne göre en-boy düzeltmesi.
- **Konum haritası.** Çalışma kapsamının 6 katı (kapsam 30 dereceyi aşarsa 3 katı) bir alan; kapsam kırmızı çerçeveyle gösterilir.
- **180° boylamı.** Veriler 180. boylamın iki yanındaysa ve bu çizginin üzerinden birbirine daha yakınsa batı boylamlarına 360° eklenir; bölge tek parça çizilir.
- **Yazı.** Arial (yoksa Helvetica, Liberation Sans, DejaVu Sans) 7-9 punto, beyaz zemin; PDF, SVG ve EPS'de yazı tipi gömülür (`map_widget.py:149-153`, `219`).

- **Ne işe yarar.** Bir makalenin "çalışma alanı" şeklini programdan çıkmadan, sütun genişliğinde ve gerçek çözünürlükte üretir.
- **Ne zaman, nasıl ayarlanır.** Dergi tek sütun şekil istiyorsa 85 mm, tam sayfa için 174 mm seçilir. Haritada ince yazı ve çizgi bulunduğundan 600 dpi çoğu dergi için yeterlidir; vektör biçim (PDF, EPS) isteniyorsa çözünürlük kullanılmaz. Çoğu dergi başlığı şekil altyazısında istediği için `Include the title above the map` kapalı gelir.
- **Sonuca etkisi.** Model sonucunu etkilemez. Şekil, Map sekmesinde işaretli katmanları çizer; bir katmanın kutusu kapalıysa şekilde de yoktur.
- **Dikkat.** Çizilecek bir şey yoksa "Nothing to draw yet: load records or a study boundary in ① Data." iletisi çıkar. Koordinatlar derece değilse şekil "The coordinates are not longitude/latitude in degrees … Run (1) Data first so that projected coordinates are converted, or check that longitude and latitude are not swapped." iletisiyle reddedilir (`study_area_figure.py:64-69`, `tabs/tab_map.py:278-282`). Seyreltilmiş katmanlarda (20.000'den çok nokta) şekil de seyreltilmiş noktaları çizer. Adı `_study_area.png` ile biten dosyalar `⑧ Report`'taki harita bölümüne alınmaz (`tabs/tab_report.py:1802-1803`).

12.4 ⓘ Help sekmesi

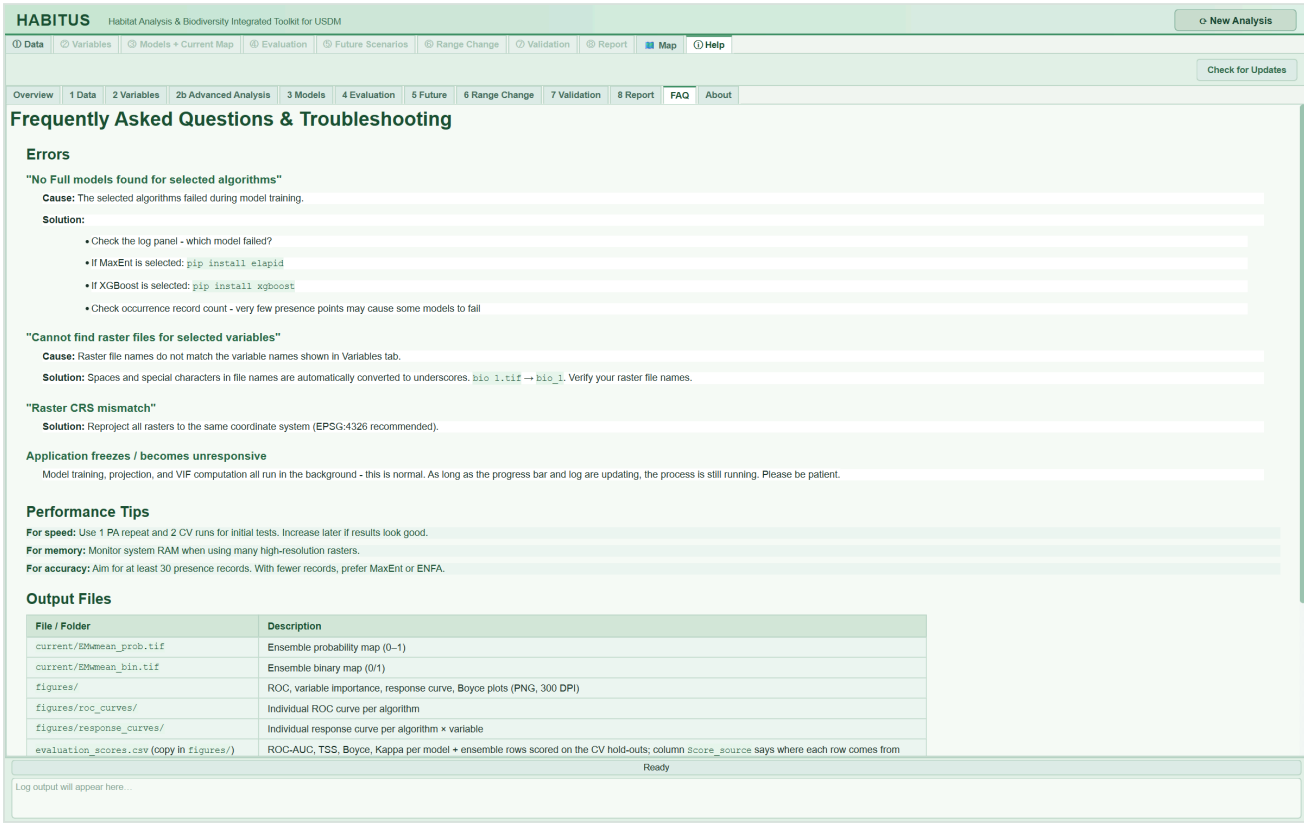
`ⓘ Help`, programın içine gömülü, yalnız İngilizce bir kullanıcı rehberidir (`tabs/tab_help.py:1000-1017`). Üst şeritte güncelleme düğmeleri, altında on iki sayfa sekmesi bulunur (`tabs/tab_help.py:1027-1101`).



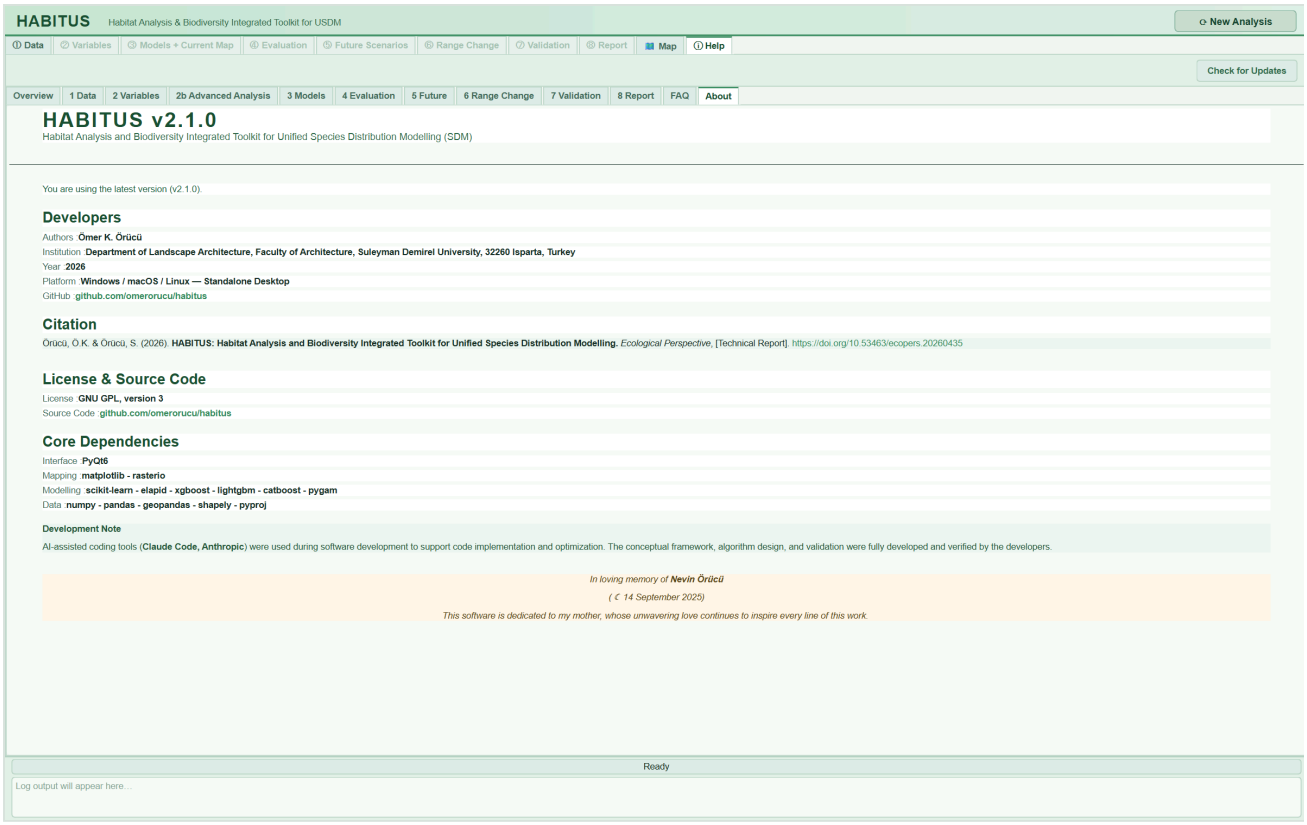
Şekil 12.2. ⓘ Help sekmesi, Overview sayfası: sağ üstte Check for Updates düğmesi, altında on iki sayfa sekmesi; sayfada "What is HABITUS?", sekiz adımlık "Typical Workflow" ve "Supported Algorithms" tablosu.

Sayfa	İçerik
`Overview`	Programın tanımı, sekiz adımlık iş akışı, algoritma tablosu ve gereken kütüphaneler, hızlı başlangıç
`1 Data`	Sunum CSV'si, rasterlar, arka plan ve sözde yokluk ayarları, çıktı klasörü

Sayfa	İçerik
`2 Variables`	VIF ve korelasyon, renk kodları, onay
`2b Advanced Analysis`	Koşul sayısı, PCA, LASSO, Ridge ve bunların birlikte okunması
`3 Models`	Algoritma seçimi, en az kayıt önerileri, eğitim ayarları, çapraz doğrulama tasarımının gerekçesi, iki katmanlı topluluk, başarım ölçütleri, güncel harita
`4 Evaluation`	Alt sekmeler, ROC, değişken önemi, tepki eğrileri, eşik yöntemleri
`5 Future`	Senaryo adı, algoritma seçimi, gelecek rasterlarının eklenmesi
`6 Range Change`	Sonuç kodları (-2, 0, 1, 2), dosya seçimi, sonuç tablosunun okunması
`7 Validation`	Harita seçimi, yeniden sınıflama, örnekleme, ölçütler, dışa aktarım, Kappa yorumu
`8 Report`	Rapor bölümleri, üretim ve dışa aktarım, ODMAP protokolü, harita kalitesi
`FAQ`	Sık hatalar, başarım ipuçları, çıktı dosyaları tablosu
`About`	Sürüm, geliştirici, atıf, lisans (GNU GPL sürüm 3), temel bağımlılıklar, güncelleme durumu



Şekil 12.3. ⓘ Help > FAQ sayfası: "Errors" başlığı altında sık hata iletileri ve çözümleri, "Performance Tips" ve "Output Files" tablosunun başı.



Şekil 12.4. ⓘ Help > About sayfası: "HABITUS v2.1.0" başlığı, güncelleme denetiminin sonucu ("You are using the latest version"), geliştirici, atıf ve DOI, lisans, temel bağımlılıklar.

Arayüzdeki ad	Ne yapar	Varsayılan	Aralık / seçenekler	Kod
`Check for Updates`	Yayın sayfasında yeni sürüm olup olmadığını elle sorar; sonuç `About` sayfasında görünür	—	düğme	`tabs/tab_help.py:1052-1059`, `1142-1152`
`Download Update`	Yeni sürüm bulunduğunda görünür; bu bilgisayarın kurulum dosyasını İndirilenler klasörüne indirir	gizli	düğme	`tabs/tab_help.py:1062-1071`, `1154-1248`

Program açılışta güncelleme denetimini kendiliğinden, arka planda yapar; `HABITUS_NO_UPDATE_CHECK` ortam değişkeni tanımlıysa yapmaz (`main_dialog.py:130-143`). Sorgu GitHub yayın sayfasına gider ve 10 saniyede yanıt gelmezse bırakılır (`updater.py:88-94`). `About` sayfası denetimin sonucunu bir şeritle gösterir: "Checking for updates...", "You are using the latest version (v...)", "Update Available - v..." ya da "Update check failed: ..." (`tabs/tab_help.py:1103-1140`). Yeni sürüm bulunursa günlük paneline de bir satır yazılır (`main_dialog.py:145-148`).

`Download Update` kurulum dosyasını indirir ama kurmaz: "Close HABITUS and run it to install the update. Your settings and results are not touched." (`tabs/tab_help.py:1212-1218`). İndirme başarısız olursa yayın sayfasının adresi verilir.

> **Not:** Help sayfaları kısa bir başvuru rehberidir. Kesin ayrıntı (varsayılan değerler, dosya adları) için bu kitap esas alınmalıdır; örneğin çıktı dosyalarının gerçek adları §12.5'tedir.

12.5 Çıktı klasörünün dosya kataloğu

Bütün çıktılar `① Data` > `Output Directory` klasörüne (ya da `③ Models`'te yazılan klasöre) gider (Bölüm 1.9). Çok türlü dosyada her tür kendi alt klasöründedir; o alt klasör aşağıdaki ağacın aynısını taşır, üst klasörde ise `batch_summary.csv` ve (sınır varsa) `masked_inputs/current/` bulunur.

Dosya adlarındaki `<tür>` öneki: projeksiyon, rapor ve şekil dosyalarında tür adındaki boşluklar alt çizgiye çevrilir (`abies_nordmanniana_current_EMwmean_prob.tif`); alan değişimi dosyasında tür adı olduğu gibi kalır (`abies nordmanniana_SSP585_2090_SRC.tif`) (`sdm_core.py:7655-7656`, `tabs/tab_range.py:213-216`). Tür adı bilinmiyorsa önek konmaz.

Aşağıdaki ağaç örnek koşunun klasöründen alınmış, kodla tamamlanmıştır:

```
<çıktı klasörü>/
  habitus_run_config.json              ③ koşu kaydı
  evaluation_scores.csv                ③ (+ evaluation_scores.<zaman>.bak.csv)
  .habitus_state.json                  CSV yedek denetimi için özet
  occurrence_train.csv                 ③ eğitim sunumları
  occurrence_test.csv                  ③ test sunumları
  occurrence_all_splits.csv            ③ ikisi birlikte
  occurrences_used.csv                 ① yalnız çok türlü dosyada
  accessible_area.tif                  ① erişilebilir alan
  background_maxent.csv                ① isteğe bağlı
  <tür>_<YYYYMMDD_HHMMSS>.log          oturum günlüğü
  <tür>_HABITUS_report_<zaman>.html    ⑧ rapor
  odmap_protocol.md / .csv             ⑧ ODMAP düğmesiyle
  <tür>_<analiz adı>_SRC.tif           ⑥ alan değişimi
  <tür>_study_area.png                 🗺 isteğe bağlı
  masked_inputs/current/               ① sınırla kırpılmış rasterlar
  masked_inputs/<senaryo>/             ⑤ sınırla kırpılmış senaryo rasterları
  current/                             ③ güncel iklim haritaları
  <senaryo>/                           ⑤ gelecek haritaları
  figures/                             ② ④ şekiller ve tablolar
  advanced_analysis/                   ② Advanced Analysis dışa aktarımı
  tuning/tuning_<algoritma>.csv        ③ Tuning
  validation/                          ⑦ yeniden sınıflanmış rasterlar
```

Dosya	Hangi adım	İçerik	Kod
`accessible_area.tif`	① Data	Arka planın çekildiği erişilebilir alan maskesi; `Accessible area` `none` ise yazılmaz. ⑥ Range Change kullanır	`tabs/tab_data.py:1540-1545`
`background_maxent.csv`	① Data	MaxEnt arka plan noktaları ve çevresel değerleri (MaxEnt alt sekmesinde istenirse)	`tabs/tab_data.py:1550-1559`
`occurrences_used.csv`	① Data	Çok türlü dosyada yalnız bu türün kayıtları	`tabs/tab_data.py:1503`
`masked_inputs/current/`, `masked_inputs/<senaryo>/`	① Data, ⑤ Future	Çalışma sınırına kırpılmış raster kopyaları	`tabs/tab_data.py:1520`, `tabs/tab_projection.py:607`

Dosya	Hangi adım	İçerik	Kod
`figures/correlation_heatmap.png`, `correlation_heatmap_selected.png`, `vif_barplot.png`	② Variables	Korelasyon ısı haritası (bütün ve seçili değişkenler), VIF grafiği	`tabs/tab_vif.py:1007`, `1151`, `1278`
`figures/high_correlation_pairs.csv`, `figures/variable_priority_ranking.csv`	② Variables	Yüksek korelasyonlu çiftler; VIF ve öncelik sıralaması	`tabs/tab_vif.py:1056`, `1091`
`advanced_analysis/`	② Advanced Analysis	`pca_variance.png/.csv`, `pca_loadings.csv`, `lasso_coefficient_paths.png`, `lasso_coefficients.csv`, `lasso_path.csv`, `ridge_coefficient_paths.png`, `ridge_coefficients.csv`, `ridge_path.csv`	`tabs/tab_vif_advanced.py:769-882`
`current/<tür>_current_<ALGO>_PA<n>_prob.tif`, `_bin.tif`	③ Models	Her algoritma ve sözde yokluk seti için olasılık (0-1) ve ikili (0/1) harita	`sdm_core.py:7895`, `7911`
`current/<tür>_current_<ALGO>_mean_prob.tif`, `_mean_bin.tif`, `_sd.tif`	③ Models	Algoritma içi tekrarların ortalaması ve standart sapması	`sdm_core.py:7981-8005`
`current/<tür>_current_EMwmean_prob.tif`, `EMca_prob.tif` (ve `_bin`)	③ Models	Topluluk: ağırlıklı ortalama ve oy birliği (committee averaging)	`sdm_core.py:8083`, `8098`
`current/<tür>_current_EMsd.tif`	③ Models	Topluluk üyeleri arasındaki yayılma (eşiklenmez)	`sdm_core.py:8056-8066`
`current/<tür>_current_NT1.tif`, `NT2.tif`, `MIC_NT1.tif`, `MIC_NT2.tif`	③ Models	Ekstrapolasyon tanısı (ExDet) ve en etkili değişken	`sdm_core.py:7205-7215`
`current/<tür>_current_projection_options.json`	③ Models	Projeksiyon seçenekleri (clamp, MESS) ve ızgara hizalama özeti	`sdm_core.py:7655-7662`
`evaluation_scores.csv` (kopyası `figures/` içinde)	③ Models, ④ Evaluation	Her model ve topluluk için ROC-AUC, TSS, Boyce, Kappa vb.; `Score_source` sütunu skorun kaynağını söyler	`tabs/tab_evaluation.py:1073-1107`
`occurrence_train.csv`, `occurrence_test.csv`, `occurrence_all_splits.csv`	③ Models	`longitude`, `latitude`, `presence`, `split` sütunlarıyla eğitim ve test noktaları	`tabs/tab_models.py:1915-1955`
`habitus_run_config.json`	③ Models	Koşu kaydı (§12.6)	`tabs/tab_models.py:1640-1657`
`tuning/tuning_<algoritma>.csv`	③ Tuning	Izgara araması tablosu	`tabs/tab_tuning.py:408-413`

Dosya	Hangi adım	İçerik	Kod
`figures/evaluation_model_scores.png`, `evaluation_roc_curves.png`, `evaluation_omission_pr_calibration.png/.csv`, `evaluation_variable_importance.png`, `evaluation_response_<ALGO>.png`, `evaluation_boyce_<ALGO>.png`	④ Evaluation	Değerlendirme şekilleri (300 dpi) ve omisyon-kalibrasyon tablosu	`tabs/tab_evaluation.py:353-847`, `923-937`
`figures/classification_thresholds.csv`	④ Evaluation	Her eşik yöntemiyle bulunan eşikler ve hold-out başarımı	`tabs/tab_evaluation.py:1323`
`<senaryo>/<tür>_<senaryo>_…tif`	⑤ Future	`current/` ile aynı dosya ailesi, senaryo adıyla	`sdm_core.py:7895-8138`
`<tür>_<analiz adı>_SRC.tif`	⑥ Range Change	Değişim haritası: -2 kayıp, 0 kararlı yokluk, 1 kararlı varlık, 2 kazanç	`tabs/tab_range.py:213-216`
`validation/Reclassify_ref_<ad>.tif`, `Reclassify_val_<ad>.tif`	⑦ Validation	Yeniden sınıflanmış referans ve model haritaları	`tabs/tab_validation.py:728-730`, `819-830`
`validation_report_<zaman>.txt/.json/.html`, `validation_points_<zaman>.csv`	⑦ Validation	Kullanıcı kaydederse doğrulama raporu ve örnek noktalar	`tabs/tab_validation.py:970-1005`
`<tür>_HABITUS_report_<zaman>.html`	⑧ Report	Tek dosyalık HTML rapor; şekiller dosyanın içine gömülüdür	`tabs/tab_report.py:2507-2510`
`odmap_protocol.md`, `odmap_protocol.csv`	⑧ Report	Kısmen doldurulmuş ODMAP protokolü	`odmap.py:647-686`
`batch_summary.csv`	⑧ Report (toplu)	Tür başına durum, başarısız adım, hata, klasör, süre	`batch_species.py:280`
`<tür>_<YYYYMMDD_HHMMSS>.log`	bütün adımlar	Oturum günlüğü (§12.7)	`main_dialog.py:612-645`

Raster biçimi. Projeksiyon rasterları GeoTIFF'tir, `float32`, NoData değeri -9999 (`sdm_core.py:7205-7215`). Olasılık haritaları 0-1 aralığındadır. Örnek koşunun `current/` klasöründe dört algoritma için 37 harita dosyası yazılmıştır (Şekil 1.3).

> **Dikkat:** Çıktı klasöründeki dosyaların üzerine yeni koşu yazar; önceki koşudan kalan ve yeni koşuda üretilmeyen dosyalar silinmez (Bölüm 1.6.2). `evaluation_scores.csv` elle düzenlenmişse üzerine yazılmadan önce `<ad>.<YYYYmmdd_HHMMSS>.bak.csv` adıyla yedeklenir; bunun için dosyanın son yazılış özeti `.habitus_state.json` dosyasında tutulur (`sdm_core.py:6752-6800`).

12.6 Koşu kaydı: `habitus_run_config.json`

Koşu kaydı, bir analizi yeniden kurmak için gereken bilgiyi tek bir makinece okunabilir dosyada toplar (`run_config.py:1-14`). ③ `Models` her çalıştığında, modelleme bittikten sonra çıktı klasörünün köküne yazılır (`tabs/tab_models.py:1640-1657`, `run_config.py:154-163`).

12.6.1 Üst düzey alanlar

Alan	İçerik	Örnek koşu	Kod
`habitus_version`	Program sürümü	"2.1.0"	`run_config.py:140`
`random_seed`	Koşunun tohumu	42	`run_config.py:141`
`started`, `finished`	Kaydın oluşturulma ve yazılma zamanı (ISO biçimi)	ikisi de "2026-11-21T05:11:12"	`run_config.py:97-98`, `142-143`
`settings`	Bölüm bölüm ayarlar (§12.6.2)	9 bölüm	`run_config.py:144`
`setting_source`	Her ayarın kaynağı: `default`, `user` ya da `recorded`	36 `default`, 2 `user`, 44 `recorded`	`run_config.py:79-92`, `105-119`
`defaults_left_unchanged`	Varsayılan bırakılan ayarların listesi: `{section, setting, value}`	36 öğe	`run_config.py:127-134`, `146-149`
`notes`	`ADVISORY:` notları ve çapraz doğrulama özeti	4 satır	`run_config.py:121-125`
`environment`	Python sürümü, işletim sistemi, işlemci mimarisi, bilimsel paketlerin sürümleri	Python 3.13.1, Windows 11, AMD64, numpy 2.4.6, scikit-learn 1.8.0, elapid 1.0.4 …	`run_config.py:61-74`

`started` ve `finished` modelleme işinin başlangıç ve bitişi değildir; kaydın oluşturulup yazıldığı anı gösterir. İşin süresi oturum günlüğündeki zaman damgalarından okunur.

Ayar kaynağı. Kod, kaydedilen her değeri programın varsayılanıyla karşılaştırır (`run_config.py:105-119`):

- `default`: değer programın varsayılanına eşittir;
- `user`: ayarın bir varsayılanı vardır ve kullanıcı değiştirmiştir;
- `recorded`: karşılaştırılacak bir varsayılan yoktur; ya kullanıcının verdiği bir girdidir (sunum dosyası, rasterlar) ya da koşu sırasında bulunan bir niceliktir (kullanılan sunum sayısı, blok boyutu, prevalans).

`defaults_left_unchanged` listesi, raporda da açıkça basılır; böylece hiçbir varsayılan, kullanıcının farkında olmadan bir yayına girmez (`run_config.py:87-91`).

12.6.2 `settings` bölümleri

Bölüm	Başlıca alanlar	Bölüm
`data`	sunum dosyası, tür, kullanılan ve dışlanan kayıt sayıları ve gerekçeleri, hücre başına bir kayıt, rasterlar, raster ızgarası (şekil, hücre boyutu, CRS), prevalans	2
`background`	yokluk kaynağı (`pseudo`, `user`, `user+pseudo`), strateji, tekrar ve nokta sayıları, MaxEnt arka planı, örnekleme yanlılığı düzeltmesi	3
`study_area`	erişilebilir alan yöntemi, tampon, kapsamın yüzde kaçı, kalibrasyon katmanı	3
`study_boundary`	sınır katmanı, poligon sayısı, kırpılmış ızgara	2
`thinning`	seyreltme uygulandı mı, mesafe, tutulan ve atılan kayıt	3
`spatial_structure`	uzaysal öz ilişki menzili ve yöntemi, çalışma kapsamı	5
`cross_validation`	tasarım, istenen kat sayısı, blok boyutu, her sözde yokluk setinde gerçekte kullanılan tasarım (`per_pa_set`)	5
`models`	algoritmalar ve bütün seçenekleri, eşik yöntemi, Boyce formülü, değişkenler, clamp, kategorik değişkenler, sınıf ağırlığı	5-6
`ensemble`	topluluk yöntemi, en düşük skor, ağırlıklar, yazılan yüzeyler, hold-out topluluk skorları	6
`tuning_<algoritma>`	Tuning yapıldıysa ızgara, kural ve seçilen nokta	6

Örnek koşudan: `data.n_presences_used` 143, `data.n_points_dropped` 483, `data.prevalence` 0,125; `notes` içinde "Cross-validation: PA1: spatial_block, 4 fold(s), blocks of 151.5 km [SSB=0.36] | PA2: ...".

> **Not:** Dosya `③ Models` bitince yazılır. Sonraki adımların ayarları burada değildir: projeksiyon seçenekleri her senaryo klasöründeki `<tür>_<senaryo>_projection_options.json` dosyasındadır; `⑥ Range Change` ve `⑦ Validation` ayarları oturum günlüğünde ve raporda bulunur. Programın iç sabitleri (ör. Tuning ızgarasının aralıkları dışındaki sabit değerler) kayda yazılmaz; kayıt arayüzden toplanan ve hesap motoruna geçirilen değerleri içerir.

12.6.3 Koşu kaydı ve ODMAP

ODMAP (Overview, Data, Model, Assessment, Prediction), tür dağılım modellerinin raporlanması için önerilmiş standart protokoldür (Zurell vd., 2020). `⑧ Report`'taki ODMAP düğmesi koşu kaydını okuyarak protokolün öğelerini doldurur (`odmap.py:1-22`, `155-200`; Bölüm 11):

- Kayıtta karşılığı olan öğeler (`auto`) doldurulur: kayıt sayıları, öngörücüler, çapraz doğrulama, algoritmalar, eşik, topluluk vb.
- Programın bilemeyeceği öğeler (yazarlar, hipotezler, veri kaynakları, model amacı) `user_required` olarak işaretlenir ve "Not recorded by HABITUS: to be completed by the user." metnini taşır (`odmap.py:38`, `153-154`).
- Koşu kaydı bulunamazsa protokol yine yazılır, ama her öğe `user_required` olur (`tabs/tab_report.py:545-565`).

Kısacası koşu kaydı protokolün kaynağıdır; ODMAP dosyası ise kaydın okunaklı ve standart biçimli bir özetidir. Protokol, kullanıcı eksik öğeleri tamamlamadan "ODMAP uyumlu rapor" sayılmaz.

12.7 Günlükler

12.7.1 Oturum günlüğü

Günlük panelinin her satırı, çıktı klasörü belli olduğu andan itibaren klasördeki `<tür>_<YYYYMMDD_HHMMSS>.log` dosyasına da yazılır (Bölüm 1.7; `main_dialog.py:612-645`, `790-801`). Dosyanın içinde:

- başlık: "HABITUS Analysis Log", tür, başlangıç zamanı;
- her ileti `[SS:DD:ss] ileti` biçiminde; okunan her raster, dışlanan kayıt sayıları, öz ilişki menzili, her modelin süresi, kaydedilen her dosya, uyarılar ve `ADVISORY:` notları;
- hatalarda tam hata metni `--- ERROR ---` başlığıyla;
- kapanışta "Closed" satırı.

Örnek koşunun günlüğü (`habitus_20261121_050108.log`) şöyle başlar: "[05:01:08] Reading raster 1 of 19: bio_1.tif" … "[05:01:09] ⚠ 405 of 626 presence records removed (405 in a cell that already has a record) — 221 retained, one per raster cell." … "[05:01:09] Spatial autocorrelation range = 151.5 km (variogram)." ® `Report`, raporun son bölümüne çıktı klasöründeki günlük dosyasını ekler (`tabs/tab_report.py:607-616`).

12.7.2 Çökme günlüğü

Program bir hata penceresi gösteremeden kapanırsa (yerel bir çökme ya da yakalanmamış bir hata) neden kullanıcı veri klasöründeki çökme günlüğüne yazılır (`main.py:253-290`):

Platform	Klasör
Windows	`%LOCALAPPDATA%\HABITUS\logs\`
macOS	`~/Library/Application Support/HABITUS/logs/`
Linux	`$XDG_DATA_HOME/HABITUS/logs/`, değişken yoksa `~/.local/share/HABITUS/logs/`

Her gün için bir dosya vardır (`crash_YYYYMMDD.log`); her açılış sürüm numarası ve saatle bir satır ekler. Yerel bir çökmede o andaki bütün iş parçacıklarının Python yığını yazılır. Bir çökmeyi bildirirken bu dosya oturum günlüğüyle birlikte gönderilmelidir.

12.8 Ortam değişkenleri

Ortam değişkenleri programı başlatmadan önce işletim sisteminde tanımlanır. Hiçbiri olağan kullanımda gerekmez.

Değişken	Etkisi	Varsayılan	Kod
`HABITUS_UI_SCALE`	Arayüz yazılarını ve denetimlerini orantılı büyütür ya da küçültür	1.0; 0.5-3.0 dışındaki ya da geçersiz değer 1.0 sayılır	`ui_style.py:30-38`
`HABITUS_PROJECTION_TILE_CELLS`	Büyük rasterların projeksiyonunda bir şeritteki hücre sayısı; 0 şeritlemeyi kapatır. Bu değerin iki katından küçük raster tek parça işlenir	2000000	`sdm_core.py:7512-7526`
`HABITUS_NO_UPDATE_CHECK`	Boş olmayan herhangi bir değer açılıştaki güncelleme denetimini kapatır (ağsız ortam)	tanımsız	`main_dialog.py:131-132`
`HABITUS_SAFE_GRAPHICS`	Yalnız Linux başlatma betiklerinde: `1` ise yazılım çizimi açılır (pencereler boş açılıyorsa)	tanımsız	`scripts/make_linux_archive.sh:71-75`
`QT_QPA_PLATFORM`	Linux betikleri: Wayland oturumunda tanımlı değilse `xcb` (XWayland) seçilir; kullanıcının değeri korunur	betik belirler	`scripts/make_linux_archive.sh:63-66`
`QTWEBENGINE_DISABLE_SANDBOX`	Linux betikleri `1` yapar; Map sekmesinin web görünümü taşınabilir klasörden çalışabilsin diye	betik belirler	`scripts/make_linux_archive.sh:67-70`
`PROJ_DATA`, `PROJ_LIB`	Geçerli bir koordinat veri tabanını göstermiyorsa açılışta program kendi kopyasına ayarlar	program denetler	`main.py:110-163`
`PYTHONHASHSEED`	Tanımlı değilse ⓘ `Data` çalışırken koşu tohumuna ayarlanır	tohum	`run_config.py:49`
`HABITUS_OWNER_PID`	İç kullanım: çalışan program kendi süreç numarasını yazar; ondan türeyen yardımcı süreçler ikinci bir pencere açmaz. Elle ayarlanmamalıdır	program yazar	`proc_guard.py:46`, `86-122`

12.8.1 `HABITUS_PROJECTION_TILE_CELLS`

- **Ne işe yarar.** Projeksiyonda raster satır şeritlerine bölünür; her şerit ayrı tahmin edilir ve dosyalar birleştirilir (`sdm_core.py:7514-7526`).
- **Ne zaman, nasıl ayarlanır.** Bellek yetersizliği hatası alınırsa değer küçültülür (ör. 500000). Bellek boldur ve hız önemliyse büyütülebilir.
- **Sonuca etkisi.** Kod açıklamasına göre şeritli projeksiyon tek parça projeksiyonla aynı dosyaları üretmek üzere tasarlanmıştır (`sdm_core.py:7531-7537`); şerit boyutu bellek ve süreyi değiştirir. Bu kitap için şeritli ve tek parça çıktılar ayrıca karşılaştırılmamıştır.
- **Dikkat.** Sayı olmayan bir değer varsayılana (2000000) döner (`sdm_core.py:7521-7524`).

12.9 Sorun giderme

Tam hata metni hata penceresindeki `Copy full message` düğmesiyle alınır ve oturum günlüğünde bulunur (Bölüm 1.7). Aşağıdaki iletiler koddan alınmıştır (başları verilmiştir).

İleti	Neden	Çözüm	Kod
"Please specify an output directory."	`Output Directory` boş	Klasör seçin ya da önce CSV seçip otomatik klasörü bekleyin	`tabs/tab_data.py:1370-1371`
"Environmental rasters are not on the same grid."	Rasterlardan biri boyut, hücre boyutu, orijin ya da CRS bakımından ilk rasterdan farklı; ileti farklı dosyayı adlandırır	Rasterları bir CBS programında ortak ızgaraya getirin ya da çalışma sınırı verin (Bölüm 2)	`sdm_core.py:1735-1742`
"No continuous environmental rasters were given."	Yalnız kategorik raster eklenmiş	En az bir sürekli raster ekleyin	`sdm_core.py:1554`
"Only N valid presence point(s) remain after removing …"	Kayıtların çoğu raster dışında ya da NoData'da; çoğunlukla CSV ile rasterların koordinat sistemi farklıdır	Koordinat sistemini ve boylam-enlem sırasını denetleyin; `🗺 Map`'te kayıtlara bakın	`sdm_core.py:2028-2036`
"No presence record can be used: the layer(s) … have no value (NoData) …"	Adı geçen katman bütün kayıtlarda NoData	Katmanı denetleyin ya da çıkarın	`sdm_core.py:2021`
"N of M coordinate pair(s) are not valid degrees"	Koordinatlar metre ya da sütunlar ters	CSV'yi düzeltin	`sdm_inputs.py:192`
"Only one predictor is loaded …"	Tek sürekli raster	`① Data`'da raster ekleyin	`tabs/tab_vif.py:369`
"… is too small for N requested points: the background was drawn from the FULL raster extent instead."	Erişilebilir alan istenen nokta sayısı için dar	Tamponu büyütün ya da nokta sayısını azaltın	`sdm_core.py:2845`
"Calibration polygon: NONE of the N presence records lies inside the polygon(s)."	Poligon başka bölgede ya da başka CRS'te	Poligonu denetleyin	`sdm_core.py:2657`
"WARNING: … were merged into others …"	Blok tasarımı istenen kat sayısını karşılayamadı	Kat sayısını azaltın ya da blok boyutunu küçültün (Bölüm 5)	`sdm_core.py:5131`
"WARNING (design not applied): …"	İstenen çapraz doğrulama tasarımı uygulanamadı, rastgele katlara geçildi; skorlar iyimser olabilir	Tasarımı ve kayıt dağılımını gözden geçirin (Bölüm 5)	`sdm_core.py:5139`
"No model passed the ensemble threshold …"	Hiçbir model topluluğa girme eşiğini geçemedi	En düşük skoru düşürün ya da başka ölçüt seçin (Bölüm 6)	`tabs/tab_models.py:1732`
"No Full models found for selected algorithms."	Seçilen algoritmalar eğitimde başarısız	Günlükteki `FAILED` satırlarına bakın; kütüphanenin kurulu olduğunu denetleyin	`sdm_core.py:7859`
"Raster count mismatch! …"	Gelecek rasterlarının sayısı modelin değişkenleriyle uyuşmuyor	Senaryo klasörünü ve dosya adlarını denetleyin (Bölüm 8)	`tabs/tab_projection.py:527`

İleti	Neden	Çözüm	Kod
"Cannot find raster files for N selected variable(s)"	Senaryo dosya adları değişken adlarıyla eşleşmiyor	Dosya adlarını ② `Variables`'taki adlarla eşleyin	`tabs/tab_models.py:1435`
"Complete the workflow for one species first …"	Toplu çalıştırma için önce bir türde ①-③ tamamlanmalı	Bir türü sonuna kadar çalıştırın (Bölüm 11)	`tabs/tab_report.py:481`
"ODMAP protocol: No output folder yet. Run the models first."	Çıktı klasörü yok	Önce ③ `Models`'i çalıştırın	`tabs/tab_report.py:554`
"… still running — close anyway?"	Çalışan iş varken pencere kapatılıyor	`Yes` işi iptal eder; bitmesini bekleyin	`main_dialog.py:293-301`
"Unable to allocate …" / "MemoryError"	Bellek yetersiz	Rasterları kırpın, nokta sayısını azaltın, `HABITUS_PROJECTION_TILE_CELLS`'i küçültün	`error_dialog.py:39-41`
"proj.db lacks DATABASE.LAYOUT.VERSION.MAJOR …"	Başka bir koordinat kütüphanesi kurulumu araya giriyor	Program açılışta kendi kopyasını seçer; sürerse `PROJ_DATA` ve `PROJ_LIB` değişkenlerini kaldırın	`main.py:82-110`
"Permission denied" / "being used by another process"	Çıktı dosyası başka programda açık	Dosyayı kapatıp yeniden çalıştırın	`error_dialog.py:51-53`
"The map needs PyQt6-WebEngine, which is not installed"	Map sekmesinin web bileşeni yok (kaynaktan çalıştırma)	`pip install PyQt6-WebEngine`	`occurrence_map.py:140-141`
Map sekmesinde altlık boş	Ağ bağlantısı yok ya da döşeme sunucusu yanıt vermiyor	Bağlantıyı denetleyin; noktalar ve `Save study-area figure…` ağsız çalışır	`occurrence_map.py:92-95`
Linux'ta pencereler boş açılıyor	Grafik sürücüsü sorunu	`HABITUS_SAFE_GRAPHICS=1` ile başlatın	`scripts/make_linux_archive.sh:71-75`
Program iletisiz kapandı	Yerel çökme	Çökme günlüğünü (§12.7.2) gönderin	`main.py:253-290`
Pencere yanıt vermiyor gibi	Uzun iş arka planda sürüyor	İlerleme çubuğu ve günlük güncelleniyorsa bekleyin	`threads.py`

İpucu: Bir kurulumun sağlam olup olmadığını denemek için önce `--selftest` çalıştırın (Bölüm 1.3). Rapor, eksik algoritma kütüphanelerini ve koordinat veri tabanı sorunlarını adıyla gösterir.

12.10 Sistem gereksinimleri

Öğe	Gereksinim	Kaynak
Windows	64 bit Windows 10 (sürüm 1809, yapı 17763) veya üstü; Windows 11	`habitus_setup.iss:33`, `44`
macOS	Apple Silicon ve Intel paketleri ayrı; en düşük sistem sürümü paketin üzerinde yazar	`habitus.spec:345-373`
Linux	64 bit; glibc 2.35 veya üstü olan dağıtımlar (paketler bazı sistem kitaplıklarını, ör. GL ve X11, sistemden alır)	`CHANGELOG.md`, Linux paketleri
Kaynaktan çalıştırma	Python 3.10 veya üstü ve `requirements.txt` bağımlılıkları	`pyproject.toml:11`
Bellek	Kodda bir alt sınır yoktur; gereken bellek raster boyutuna ve sayısına bağlıdır. Ölçülmüş bir örnek: dünyayı kaplayan 19 katman (2,5 dakika) bellekte yaklaşık 2,6 GB yer tutar	`CHANGELOG.md`
Ağ	Gerekmez. Yalnız Map sekmesinin altlık döşemeleri ve güncelleme denetimi ağa bağlanır	`occurrence_map.py:92`, `updater.py:88`
Ekran	Pencere ekranın %95'ine kadar açılır; küçük ekranda sekmeler kaydırılır. Yazılar `HABITUS_UI_SCALE` ile büyütülebilir	`main_dialog.py:113-121`, `ui_style.py:30-38`

Bellek gereksinimini en çok azaltan önlem, rasterları `① Data` > `Study Boundary` ile çalışma bölgesine kırpmaktır (Bölüm 2). Algoritmaların süreleri oturum günlüğüne yazılır; büyük bir analizden önce küçük bir deneme (bir sözde yokluk tekrarı, az algoritma) süreyi kestirmeye yardım eder.

12.11 Uzman soruları

S1. Bir başka araştırmacı bu çalışmayı yalnız çıktı klasöründen yeniden kurabilir mi? Cevap: Büyük ölçüde. `habitus_run_config.json` tohumu, girdi dosyalarının yollarını, bütün arayüz ayarlarını, gerçekte uygulanan çapraz doğrulama tasarımını ve paket sürümlerini taşır (`run_config.py:137-152`; `tabs/tab_models.py:2056-2242`). Girdi dosyalarının kendisi klasöre kopyalanmaz (çalışma sınırıyla kırpılan kopyalar dışında, `masked_inputs/`); yalnız yolları yazılır. Rasterların içeriğinin değişmediğini kanıtlayan bir sağlama özeti kayda yazılmaz. Projeksiyon ayarları ayrı `projection_options.json` dosyalarındadır. Tam yeniden kurulum için girdi verisi ayrıca arşivlenmelidir.

S2. `defaults_left_unchanged` neden ayrı bir liste? Cevap: Bir yayında hangi sayıların bilinçli seçildiğini, hangilerinin programın varsayılanı olarak kaldığını ayırmak için. Kod her ayarı varsayılanıyla karşılaştırır ve varsayılana eşit olanları bu listeye koyar (`run_config.py:105-134`); rapor bu listeyi basar. Örnek koşuda 36 ayar varsayılandadır. Hakem, örneğin sözde yokluk sayısının (1000) veya kat sayısının gerekçelendirilip gerekçelendirilmediğini bu listeden sorabilir.

S3. `recorded` kaynağı ne anlama gelir; `user` ile farkı ne? Cevap: `user` yalnız varsayılanı olan ve kullanıcının değiştirdiği ayarlar içindir. Varsayılanı olmayan değerler (girdi dosyaları, koşu sırasında hesaplanan blok boyutu, prevalans, kullanılan kayıt sayısı) `recorded` olarak işaretlenir (`run_config.py:79-92`, `113-118`). Böylece hesaplanan bir sayı kullanıcı seçimi gibi görünmez. Örnek koşuda 44 değer `recorded`, 2 değer `user`'dır.

S4. Koşu kaydı ODMAP raporunun yerine geçer mi? Cevap: Hayır. ODMAP (Zurell vd., 2020) yazarlık, amaç, hipotezler ve veri kaynakları gibi programın bilemeyeceği öğeler içerir. `odmap.py`, kayıtta karşılığı olan

öğeleri `auto` olarak doldurur, ötekileri `user_required` olarak bırakır ve hiçbir değeri uydurmaz (`odmap.py:14-22`, `153-154`). Kod açıklamasına göre öğeler makalenin Tablo 1'indeki öğelerdir; ek tablodaki ayrıntılı soru listesi temel alınmamıştır (`odmap.py:11-14`). Kullanıcının `user_required` satırları tamamlaması gerekir.

S5. Çıktı haritaları başka bir CBS programında doğrudan kullanılabilir mi; değerler nasıl yorumlanır? Cevap: Evet; GeoTIFF, `float32`, NoData -9999 olarak yazılır ve kaynak rasterların koordinat sistemini taşır (`sdm_core.py:7205-7215`). `_prob` dosyaları 0-1 aralığında uygunluktur; bu bir bulunma olasılığı değildir, çünkü sözde yokluklarla kurulan modellerde prevalans bilinmez (Elith ve Leathwick, 2009). `_bin` dosyaları seçilen eşikle elde edilmiş 0/1 haritalardır; eşik yöntemi `models.binary_threshold_method` alanındadır. `EMsd` ve `_sd` yüzeyleri uygunluk değil, modeller arası uyuşmazlıktır ve eşiklenmez (`sdm_core.py:8056-8066`).

S6. Ekrandaki harita görüntüsü ile kaydedilen GeoTIFF arasında fark var mı? Cevap: Görüntüleyicinin ve `Save study-area figure…`'ın ürettiği PNG, PDF gibi dosyalar resimdir; renk ölçeği, ölçek çubuğu ve en-boy düzeltmesi içerir ve analiz için kullanılmamalıdır. Sayısal analiz `current/` ve senaryo klasörlerindeki GeoTIFF'lerle yapılır. Map sekmesindeki noktalar 20.000'i aşınca yalnız görüntü için seyreltilir (`tabs/tab_map.py:22-24`); modelleme bu seyreltmeden etkilenmez.

S7. Program aynı klasöre ikinci kez yazdığında eski sonuçları nasıl ayırt ederim? Cevap: Program eski dosyaları silmez, aynı adlıların üzerine yazar. Ayırt etmenin yolları: koşu kaydının `finished` zamanı, oturum günlüğünün zaman damgalı adı ve raporun zaman damgalı adı. `evaluation_scores.csv` elle değiştirilmişse yedeklenir (`sdm_core.py:6764-6800`); öteki dosyalar yedeklenmez. Karşılaştırmalı koşular için her koşuya ayrı klasör vermek tek güvenilir yoldur (Bölüm 1.9).

S8. Yazılım ortamının (paket sürümleri) sonuca etkisi kayıt altında mı? Cevap: Evet, `environment` alanında Python sürümü, işletim sistemi, mimari ve numpy, pandas, scipy, scikit-learn, rasterio, matplotlib, xgboost, lightgbm, catboost, pygam, elapid, PyQt6 sürümleri yazılır (`run_config.py:61-74`). Aynı tohumla aynı sonucun beklenebilmesi bu sürümlerin de aynı olmasına bağlıdır; farklı bir scikit-learn ya da elapid sürümü aynı tohumla farklı sayılar verebilir. Paketli program sürümleri sabitlediği için aynı paket sürümüyle tekrarlamak en güvenilir yoldur.

S9. Çalışma alanı şeklindeki ölçek çubuğu ve en-boy oranı ne kadar doğru? Cevap: Şekil bir projeksiyon kullanmaz; boylam-enlem eksenleri, enlem ortasının kosinüsüne göre en-boy düzeltmesiyle çizilir (`study_area_figure.py:229-232`). Ölçek çubuğu harita görüntüleyicisinin yöntemini kullanır (`study_area_figure.py:241`). Bu, küçük ve orta ölçekli alanlar için yeterli bir yaklaşımdır; geniş enlem aralıklarında (ör. kıta ölçeği) ölçek çubuğu yalnız orta enlem için doğrudur. Eşit alanlı bir projeksiyon gerekiyorsa şekil bir CBS programında yeniden çizilmelidir.

S10. Güncelleme denetimi hangi bilgiyi dışarı gönderir; ağsız çalışılabilir mi? Cevap: Denetim yalnız GitHub'daki son yayının bilgisini ister (`updater.py:88-94`); analiz verisi gönderilmez. Map sekmesi altlık döşemelerini indirirken programın adını ve sürümünü kimlik olarak gönderir (`occurrence_map.py:53-58`). Program ağsız çalışır; güncelleme denetimi `HABITUS_NO_UPDATE_CHECK` ile kapatılabilir (`main_dialog.py:131-132`), Map sekmesinde yalnız altlık eksik kalır.

12.12 Sık hatalar ve tuzaklar

- **Map sekmesinde dışlanan kayıtları beklemek.** `Excluded occurrences` katmanı `① Data` çalıştırılmadan boştur.
- **Projeksiyonlu koordinatları Map sekmesinde yanlış yerde görmek.** Rasterların koordinat sistemi `① Data` çalıştırılınca bilinir; ondan önce metre cinsinden koordinatlar haritaya yerleştirilemez.
- **Ekran görüntüsünü yayın şekli olarak kullanmak.** OpenStreetMap döşemeleri ekran çözünürlüğündedir ve kendi lisans koşulları vardır; yayın için `Save study-area figure…` kullanın.
- **Help sayfalarını kesin başvuru saymak.** Help kısa bir rehberdir; dosya adları ve varsayılanlar için bu kitaptaki tablolar esastır.
- **Koşu kaydını tek kanıt saymak.** Kayıt `③ Models` anına aittir; projeksiyon, alan değişimi ve doğrulama ayarları başka dosyalardadır (projeksiyon seçenekleri, oturum günlüğü, rapor).
- **Çökmeden sonra yalnız oturum günlüğünü göndermek.** Yerel bir çökme oturum günlüğüne yazılamayabilir; çökme günlüğü (§12.7.2) ayrı bir dosyadır.
- **Ortam değişkenini program açıkken değiştirmek.** Değişkenler açılışta okunur; değişiklik için program yeniden başlatılmalıdır.

12.13 Kaynaklar

Elith, J., Leathwick, J.R., 2009. Species distribution models: ecological explanation and prediction across space and time. Annual Review of Ecology, Evolution, and Systematics 40, 677-697. https://doi.org/10.1146/annurev.ecolsys.110308.120159

Zurell, D., Franklin, J., König, C., Bouchet, P.J., Dormann, C.F., Elith, J., vd., 2020. A standard protocol for reporting species distribution models. Ecography 43, 1261-1277. https://doi.org/10.1111/ecog.04960