Argomento: Classificazione per il rilevamento dei rischi ambientali

Corso: Machine Learning per applicazioni ambientali

Obiettivi di apprendimento

Al termine di questa unità, gli studenti saranno in grado di:

- Applicare metodi di apprendimento supervisionato per classificare rischi e pericoli ambientali.
- Valutare i modelli di classificazione mediante precision, recall, F1-score e metriche AUC.
- Implementare e interpretare algoritmi quali regressione logistica, SVM, Random Forest e CNN.
- Utilizzare i risultati dei modelli per supportare il monitoraggio ambientale e le decisioni relative alla gestione del rischio.

Introduzione

I modelli di classificazione rappresentano uno strumento fondamentale nella scienza dei dati ambientali, poiché permettono di assegnare osservazioni eterogenee a categorie discrete di rischio, ad esempio contaminato o non contaminato, presenza o assenza di una fioritura algale nociva, oppure condizioni sicure o pericolose. Diversamente dalla regressione, che predice valori continui (ad esempio la concentrazione di inquinanti), la classificazione produce etichette categoriali direttamente utilizzabili nella valutazione del rischio.

I dataset ambientali integrano generalmente diverse fonti di informazione, tra cui:

- Parametri fisico-chimici (ad esempio pH, ossigeno disciolto e carico di nutrienti);
- Fattori meteorologici (ad esempio temperatura, vento e precipitazioni);
- Descrittori dell’uso del suolo e della topografia;
- Indici di telerilevamento e misurazioni iperspettrali.

Esempi di classificazione in questo contesto comprendono stabilire se un campione d’acqua sia contaminato o pulito, assegnare la qualità dell’aria a categorie normative, distinguere le

tipologie di copertura del suolo da immagini satellitari e segnalare letture dei sensori normali o anomale.

Questi problemi sono complessi perché spesso coinvolgono numerose variabili predittive correlate, dipendenze temporali o spaziali che violano le ipotesi di indipendenza e un forte sbilanciamento tra classi, dato che gli eventi pericolosi sono generalmente meno frequenti delle condizioni sicure. Per affrontare tali difficoltà sono necessarie procedure solide di pre-elaborazione (imputazione dei valori mancanti, normalizzazione e codifica delle variabili categoriali) e strategie di validazione adeguate (ad esempio cross-validation a blocchi spaziali o temporali), così da prevenire la contaminazione informativa (*data leakage*) e ottenere stime realistiche della capacità di generalizzazione [4].

Oltre all'accuratezza predittiva, sono fondamentali l'interpretabilità e la quantificazione dell'incertezza. Le analisi dell'importanza delle variabili permettono di verificare che i modelli si basino su fattori ecologicamente significativi (ad esempio carichi di nutrienti o anomalie della temperatura), mentre la dispersione delle predizioni dei modelli ensemble e gli intervalli di previsione forniscono misure dell'incertezza utili per decisioni consapevoli del rischio. Infine, i cambiamenti ricorrenti nella distribuzione dei dati (dovuti a cicli stagionali, aggiornamenti dei sensori o eventi nuovi) richiedono riaddestramento periodico, transfer learning e valutazione continua per mantenere prestazioni affidabili nel tempo [4, 5, 7].

Tipologie di algoritmi

Nell'analisi dei dati ambientali sono comunemente impiegate diverse famiglie di algoritmi di classificazione:

- Regressione logistica: modello probabilistico lineare che fornisce coefficienti interpretabili per classificazioni binarie o multiclasse.
- Support Vector Machine (SVM): classificatori basati sul margine che utilizzano kernel per separare le classi in spazi lineari o non lineari.
- Alberi decisionali: modelli gerarchici che suddividono i dati in classi mediante soglie successive basate sulle caratteristiche.
- Random Forest e Gradient Boosting: insiemi di alberi decisionali che aumentano la capacità predittiva e riducono la varianza.
- Reti neurali convoluzionali (CNN): modelli di deep learning che estraggono automaticamente caratteristiche gerarchiche da immagini o dati spettrali.

Metriche di valutazione

Per valutare i modelli di classificazione in ambito ambientale non basta considerare l'accuratezza complessiva, poiché i fenomeni ecologici pericolosi sono spesso rari e i dataset fortemente sbilanciati.

Accuratezza (Accuracy)

- Proporzione di campioni classificati correttamente sul totale.
- Utile quando le classi sono bilanciate.
- Limite: può risultare fuorviante con dati sbilanciati (ad esempio, con il 95% di casi sicuri e il 5% pericolosi, un classificatore banale che predice sempre «sicuro» raggiunge già il 95% di accuratezza).

Precision

- Quota delle predizioni positive che corrispondono a veri positivi.
- Una precision elevata indica pochi falsi allarmi.
- Importante quando i falsi positivi comportano costi elevati (ad esempio interventi di bonifica non necessari).

Recall (sensibilità)

- Quota dei casi realmente positivi identificati correttamente.
- Un recall elevato implica pochi eventi pericolosi non rilevati.
- Essenziale nel monitoraggio ambientale, dove i falsi negativi (ad esempio sversamenti di petrolio non individuati o contaminazioni non rilevate) possono avere conseguenze gravi.

F1-score

- Media armonica di precision e recall.
- Bilancia l'esigenza di rilevare i pericoli con quella di evitare falsi allarmi.
- È ampiamente utilizzato con dataset sbilanciati per sintetizzare le prestazioni.

ROC–AUC (Receiver Operating Characteristic)

- Misura il compromesso tra tasso di veri positivi e tasso di falsi positivi al variare della soglia.
- Offre una valutazione complessiva delle prestazioni del classificatore.
- Limite: può apparire eccessivamente favorevole in situazioni di forte sbilanciamento.

PR–AUC (Precision–Recall)

- Descrive la relazione tra precision e recall al variare delle soglie.
- È più informativa della ROC–AUC quando le anomalie sono rare.
- È particolarmente rilevante nel rilevamento dei rischi ecologici, in cui gli eventi pericolosi sono poco frequenti ma hanno un impatto elevato.

Problemi ambientali

Molte delle principali sfide ambientali possono essere formulate come problemi di classificazione, per i quali algoritmi diversi offrono punti di forza complementari [4, 5].

Sistemi acquatici

Compiti: rilevamento di fioriture algali nocive, classificazione della contaminazione e dello stato trofico.

Algoritmi adatti:

- Regressione logistica: interpretabile per variabili predittive chimiche e fisiche [2];
- Limite: poco efficace in presenza di effetti non lineari.
- Random Forest/Gradient Boosting: modellano interazioni complesse in dataset idrochimici eterogenei [4];

Limite: minore interpretabilità.

Sistemi atmosferici

Compiti: classificazione della qualità dell'aria e rilevamento di pennacchi di inquinamento industriale [4].

Algoritmi adatti:

- SVM: efficaci con dati dei sensori ad alta dimensionalità [4];
  Limite: costo computazionale su larga scala.
- Gradient Boosting: elevate prestazioni predittive con dati sbilanciati sulla qualità dell'aria [5];

  Limite: richiede un'attenta regolazione degli iperparametri.

Foreste e uso del suolo

Compiti: mappatura del rischio di incendi boschivi, classificazione dello stress della vegetazione e rilevamento della deforestazione [4, 5].

Algoritmi adatti:

- Alberi decisionali: semplici e interpretabili per soglie legate alla vegetazione;

  Limite: inclini all'overfitting.
- Random Forest: prestazioni robuste con predittori eterogenei relativi all'uso del suolo [5];

  Limite: logica decisionale meno trasparente.
- CNN: ideali per immagini di telerilevamento della vegetazione e della copertura del suolo;
  Limite: richiedono grandi quantità di dati.

Salute degli ecosistemi

Compiti: rilevamento di specie invasive mediante DNA ambientale (eDNA) e classificazione delle categorie di rischio di malattia nella fauna selvatica [5, 7].

Algoritmi adatti:

- SVM: adatte a dati genomici o eDNA ad alta dimensionalità [4];
  Limite: minore interpretabilità.
- Regressione logistica: modello di riferimento interpretabile per i fattori di rischio delle malattie [4];
  Limite: prestazioni inferiori in presenza di interazioni complesse.

In questi ambiti, la scelta dell'algoritmo richiede di bilanciare accuratezza predittiva, interpretabilità ed efficienza computazionale. I modelli ensemble e di deep learning eccellono con dati complessi e ad alta dimensionalità [5], mentre i modelli lineari e quelli basati su alberi offrono trasparenza e robustezza in compiti più semplici [4].

Rilevamento di elementi pericolosi o anomali

Le anomalie ambientali corrispondono spesso a eventi rari, inattesi e localizzati nello spazio, che comportano rischi ecologici significativi. Tra gli esempi vi sono improvvisi picchi di particolato, fioriture algali nocive, diffusione di specie invasive o presenza di microplastiche e metalli pesanti nei sistemi acquatici e terrestri. Una difficoltà fondamentale è l'estremo

sbilanciamento delle classi: gli eventi pericolosi possono rappresentare meno del 5% delle osservazioni, risultando difficili da rilevare con classificatori convenzionali. Inoltre, le anomalie dipendono spesso dal contesto locale: senza un'attenta validazione, i modelli addestrati in una regione potrebbero non generalizzare ad altre regioni.

Elenco dei metodi

Per individuare anomalie nei dati ambientali si utilizzano diversi approcci:

- Rilevamento statistico degli outlier: identifica valori estremi sulla base di ipotesi distribuzionali.
- Ricampionamento e apprendimento sensibile ai costi: riequilibra le classi mediante sovracampionamento, sottocampionamento o penalità più elevate per i falsi negativi.
- Apprendimento ensemble: Random Forest, Gradient Boosting e modelli combinati (*stacking*) integrano più classificatori per aumentare la robustezza rispetto al rumore.
- Rilevamento delle anomalie mediante deep learning: modelli convoluzionali o basati su autoencoder individuano configurazioni non lineari e caratteristiche spaziali o spettrali insolite.

Esempi di metodi specifici

Tre tecniche rappresentative mostrano come i diversi approcci affrontino concretamente il rilevamento delle anomalie:

- One-Class SVM: apprende una frontiera decisionale che racchiude i dati «normali» e segnala come anomalie i campioni esterni a tale regione.
  Vantaggi: efficace in spazi ad alta dimensionalità e robusta rispetto a livelli moderati di rumore.
  Limiti: sensibile alla scelta del kernel e degli iperparametri; computazionalmente onerosa su dataset grandi.
  Applicazioni: rilevamento di episodi anomali di qualità dell'aria e identificazione di specie invasive mediante campioni di eDNA [6, 7].
- Random Forest: insieme di alberi decisionali che coglie interazioni complesse tra variabili e individua osservazioni anomale sulla base della confidenza delle classificazioni.
  Vantaggi: scalabile, resistente all'overfitting e in grado di gestire input eterogenei.

Limiti: meno interpretabile degli alberi semplici; le prestazioni dipendono dalla regolazione del modello e dal bilanciamento dei campioni.

Applicazioni: rilevamento di suoli contaminati mediante analisi chimiche e identificazione di profili insoliti della qualità dell'acqua [3].

- Autoencoder: reti neurali addestrate a ricostruire gli input; un errore di ricostruzione elevato segnala un'anomalia.

  Vantaggi: modellano dipendenze non lineari complesse e sono adatti al monitoraggio continuo.

  Limiti: richiedono sufficienti dati «normali» per l'addestramento, sono computazionalmente impegnativi e meno interpretabili.

  Applicazioni: rilevamento in tempo reale di anomalie nel monitoraggio della qualità dell'acqua [1].

Caso di studio: analisi del rischio legato alla distribuzione di una specie mediante dati di presenza GBIF e predittori bioclimatici WorldClim

Obiettivo

Questo caso di studio illustra una procedura riproducibile di rilevamento dei rischi ambientali basata sulla modellazione della distribuzione delle specie (*species distribution modeling*, SDM). L'obiettivo è individuare le condizioni ambientali associate alla presenza di una specie di interesse e costruire un modello di machine learning che ne stimi la probabilità di presenza in un'area di studio. Nell'ottica del rischio ecologico, le aree con elevata idoneità prevista possono essere considerate prioritarie per il monitoraggio e la prevenzione, soprattutto nel caso di specie invasive o vettori di malattie.

La procedura è implementata in Python utilizzando soltanto librerie standard per la scienza dei dati e fonti di dati aperte.

Librerie utilizzate

Utilizzeremo un insieme essenziale di librerie Python ampiamente diffuse per recuperare e manipolare dati, estrarre valori dai raster, applicare il machine learning e visualizzare i risultati.

Gli strumenti scelti sono volutamente semplici e basati su pacchetti consolidati della scienza dei dati applicata. Di seguito ne riportiamo l'elenco e le finalità:

- **requests:** consente di accedere alle API web (GBIF) e scaricare i layer climatici (WorldClim).
- **pandas** e **numpy:** servono per pulire e filtrare i dati e costruire le variabili predittive.
- rasterio: permette di leggere raster GeoTIFF ed estrarre variabili ambientali in corrispondenza di coordinate geografiche.
- **scikit-learn:** mette a disposizione modelli di classificazione e metriche di valutazione.
- **matplotlib:** consente di creare grafici semplici e interpretare i modelli.

Questa scelta favorisce la trasparenza e rende la procedura facilmente eseguibile in ambienti standard quali Jupyter Notebook o Google Colab.

Fonte dei dati

In questo tutorial pratico ci concentriamo su "***Vespa velutina***" (calabrone asiatico), una specie invasiva diffusasi in vari paesi europei che può minacciare la biodiversità locale e l'apicoltura, perché si nutre di impollinatori come le api. Individuare le aree in cui la specie potrebbe comparire in futuro costituisce quindi un esempio concreto di rilevamento del rischio ambientale. L'obiettivo del tutorial è costruire un semplice modello di machine learning che apprenda le condizioni ambientali associate alla presenza della specie e produca un punteggio di idoneità dell'habitat o di rischio per nuove località. Più precisamente, il dataset comprende osservazioni di presenza di "***Vespa velutina***" (coordinate geografiche e metadati delle osservazioni) per l'area e il periodo selezionati (2000–2024), scaricate mediante l'API pubblica GBIF. Per costruire le variabili di input, ogni punto di presenza viene associato a predittori climatici estratti dalle variabili bioclimatiche WorldClim v2.1: layer raster globali che riassumono i pattern di temperatura e precipitazioni su lunghi periodi e sono comunemente utilizzati nella modellazione della distribuzione delle specie. Il dataset finale associa quindi ogni località a indicatori climatici (ad esempio temperatura media annua e stagionalità delle precipitazioni) e permette una classificazione binaria basata su punti di presenza e di pseudo-assenza (*background*) campionati nella stessa regione.

Fonti dei dati: GBIF Occurrence API https://techdocs.gbif.org/en/openapi/v1/occurrence e WorldClim v2.1 Bioclim https://www.worldclim.org/data/worldclim21.html

Tipologie di dati

Il tutorial combina due tipologie comuni di dati geospaziali, spesso utilizzate insieme nelle analisi ambientali: dati puntuali di occorrenza (luoghi in cui è stato osservato un organismo) e dati raster (layer spaziali che descrivono le condizioni ambientali nell'intera regione).

- **Dati di occorrenza (**osservazioni puntuali): il primo dataset è una tabella di record GBIF relativi a *Vespa velutina*. Nel codice, ogni osservazione occupa una riga con le coordinate geografiche (decimalLatitude, decimalLongitude) e metadati di base: gbifID, year, basisOfRecord, datasetKey e, quando disponibile, coordinateUncertaintyInMeters. Questi punti indicano luoghi di presenza (etichetta 1), ossia località in cui la specie è stata effettivamente segnalata. Poiché GBIF fornisce prevalentemente dati di sola presenza, i record non comprendono direttamente assenze confermate.

```
# ------------------------------
SPECIES_NAME = "Vespa velutina"  # invasive Asian hornet example
# Italy bounding box (lon_min, lat_min, lon_max, lat_max)
ROI = dict(lon_min=6.6, lat_min=36.6, lon_max=18.6, lat_max=47.2)

YEAR_FROM, YEAR_TO = 2000, 2024

taxon_key, match_json = gbif_taxon_key(SPECIES_NAME)

pres = fetch_gbif_occurrences(
    taxon_key, ROI, YEAR_FROM, YEAR_TO,
    max_points=6000
)

pres = pres.dropna(subset=["decimalLatitude", "decimalLongitude"]).copy()
pres = pres.drop_duplicates(subset=["decimalLatitude", "decimalLongitude"]).copy()

pres["label"] = 1
pres.head()
```

1 to 5 of 5 entries Filter

index	gbifID	species	scientificName	year	decimalLatitude	decimalLongitude	basisOfRecord	datasetKey	coordinateUncertaintyInMeters	label
0	4510393598	Vespa velutina	Vespa velutina Lepeletier, 1836	2024	46.551558	6.637408	HUMAN_OBSERVATION	50c9509d-22c7-4a22-a47d-8c48425ef4a7	5.0	1
1	5219609581	Vespa	Vespa velutina	2024	47.01748	6.94541	HUMAN_OBSERVATION	81981d98-e27f-4155-	3535.0	1

- **Dati raster (layer ambientali a griglia):** il secondo dataset è costituito dai raster bioclimatici WorldClim v2.1, memorizzati come file GeoTIFF. Ogni raster è una griglia che copre la regione, nella quale ciascuna cella contiene un valore climatico. In questo tutorial utilizziamo sei predittori: BIO1, BIO4, BIO5, BIO12, BIO14 e BIO15, che corrisponderanno alle colonne bio1, bio4, bio5, bio12, bio14 e bio15. I valori vengono estratti mediante rasterio.

```
# WorldClim bioclim variables to use.
# https://www.worldclim.org/data/bioclim.html
# BIO1: Annual Mean Temperature
# BIO4: Temperature Seasonality (standard deviation ×100)
# BIO5: Max Temperature of Warmest Month
# BIO12: Annual Precipitation
# BIO14: Precipitation of Driest Month
# BIO15: Precipitation Seasonality (Coefficient of Variation)
BIO_VARS = [1, 4, 5, 12, 14, 15]

# We can download where we will be downloading our data
WC_URL = "https://geodata.ucdavis.edu/climate/worldclim/2_1/base/wc2.1_2.5m_bio.zip"

# ... download and unzip ...we need to keep only the files related to the BIO_VARS

bio_files = {}
for k in BIO_VARS:
    path = os.path.join(extract_dir, f"wc2.1_2.5m_bio_{k}.tif")
    bio_files[k] = path

bio_files
```

```
{1: 'data_sdm/wc2.1_2.5m_bio/wc2.1_2.5m_bio_1.tif',
 4: 'data_sdm/wc2.1_2.5m_bio/wc2.1_2.5m_bio_4.tif',
 5: 'data_sdm/wc2.1_2.5m_bio/wc2.1_2.5m_bio_5.tif',
 12: 'data_sdm/wc2.1_2.5m_bio/wc2.1_2.5m_bio_12.tif',
 14: 'data_sdm/wc2.1_2.5m_bio/wc2.1_2.5m_bio_14.tif',
 15: 'data_sdm/wc2.1_2.5m_bio/wc2.1_2.5m_bio_15.tif'}
```

- Combinazione delle due tipologie di dati (estrazione dei valori raster nei punti): ogni punto GBIF viene arricchito con valori raster; per ogni coordinata (lon, lat) si legge il valore del corrispondente pixel in ciascun raster WorldClim e lo si aggiunge come nuova variabile. Si ottiene così una tabella standard per il machine learning, nella quale ogni riga rappresenta una località e ogni predittore una variabile climatica.

```
def sample_raster_values(raster_path, lons, lats):
    values = np.empty(len(lons), dtype=float)
    with rasterio.open(raster_path) as src:
        nodata = src.nodata
        for i, (x, y) in enumerate(zip(lons, lats)):
            v = next(src.sample([(x, y)]))[0]
            values[i] = np.nan if (nodata is not None and v == nodata) else float(v)
    return values

def add_bioclim(df, bio_files):
    out = df.copy()
    lons = out["decimalLongitude"].to_numpy()
    lats = out["decimalLatitude"].to_numpy()
    for k, path in bio_files.items():
        out[f"bio{k}"] = sample_raster_values(path, lons, lats)
    return out

pres_feat = add_bioclim(pres, bio_files)
pres_feat.head()
```

1 to 5 of 5 entries Filter

index	gbifID	species	scientificName	year	decimalLatitude	decimalLongitude	basisOfRecord	datasetKey	coordinateUncertaintyInMeters	label	bio1	bio4	bio5
0	4510393598	Vespa velutina	Vespa velutina Lepeletier, 1836	2024	46.551558	6.637408	HUMAN_OBSERVATION	50c9509d-22c7-4a22-a47d-8c48425ef4a7	5.0	1	8.567000389099121	648.1673583984375	23.148000717163086
1	5219609581	Vespa velutina	Vespa velutina Lepeletier, 1836	2024	47.01748	6.94541	HUMAN_OBSERVATION	81981d98-e27f-4155-9a94-9eae9bbad2be	3535.0	1	7.342833518981934	627.8450927734375	21.215999603271484

Poiché GBIF fornisce prevalentemente osservazioni di sola presenza, il tutorial genera nella stessa regione punti di *background* e assegna loro l'etichetta 0. Tali punti vengono utilizzati come approssimazione delle località di non presenza, consentendo di costruire un dataset per la classificazione binaria.

```
bg = random_points_bbox(n_bg, ROI, seed=42)
bg["label"] = 0

bg_feat = add_bioclim(bg, bio_files)
bg_feat.head()
```

	decimalLongitude	decimalLatitude	label	bio1	bio4	bio5	bio12	bio14	bio15
0	15.887473	46.851860	0	9.194166	768.208618	24.972	842.0	30.0	38.745869
1	11.866541	43.871469	0	10.660334	650.370911	25.180	1430.0	64.0	30.592279
2	16.903175	37.555253	0	NaN	NaN	NaN	NaN	NaN	NaN
3	14.968416	43.947677	0	NaN	NaN	NaN	NaN	NaN	NaN
4	7.730128	42.998380	0	NaN	NaN	NaN	NaN	NaN	NaN

Pulizia dei dati

I dataset di occorrenza richiedono spesso una pulizia preliminare perché possono contenere coordinate duplicate (una stessa località segnalata più volte), coordinate incerte con bassa precisione spaziale e, più raramente, punti esterni all'area di interesse. Poiché il tutorial mira ad associare le condizioni ambientali alla presenza della specie, record imprecisi o ripetuti potrebbero introdurre distorsioni nell'analisi e nel modello. La procedura applica quindi filtri standard: elimina i record senza latitudine o longitudine, conserva soltanto le osservazioni comprese nel riquadro geografico dell'area di studio (ROI), rimuove le coppie di coordinate duplicate e, quando possibile, esclude facoltativamente i punti con incertezza geografica molto elevata.

Tabella finale per l'addestramento del modello

I punti di presenza e quelli di background, entrambi corredati dei predittori BIO estratti, vengono riuniti in un unico dataset di addestramento. La matrice delle variabili di input X contiene le variabili bioclimatiche selezionate da WorldClim (predittori BIO), mentre il vettore obiettivo y contiene le etichette binarie che distinguono presenza e pseudo-assenza. Si ottiene così un dataset standard di apprendimento supervisionato adatto all'addestramento di classificatori.

```
# --------------------------------
# 7) Build final ML table (X,y)
# --------------------------------
feature_cols = [f"bio{k}" for k in BIO_VARS]
cols = ["label"] + feature_cols

data = pd.concat([pres_feat[cols], bg_feat[cols]], ignore_index=True)
data = data.dropna().copy()

X = data[feature_cols].to_numpy()
y = data["label"].to_numpy()

data["label"].value_counts()
```

label	count
0	3778
1	1566

dtype: int64

Suddivisione training/test e modello di riferimento (regressione logistica)

Per verificare la capacità del modello di generalizzare a dati mai osservati, il dataset viene suddiviso in insiemi di addestramento e test. Si utilizza una suddivisione stratificata per mantenere in entrambi gli insiemi la stessa proporzione di punti di presenza e pseudo-assenza, evitando che le stime delle prestazioni siano alterate dallo sbilanciamento tra classi. Come primo modello di riferimento si impiega la regressione logistica, semplice e interpretabile, capace di produrre previsioni probabilistiche e adatta come termine di confronto per la classificazione binaria. Metriche di valutazione: poiché i dataset ecologici di presenza sono spesso sbilanciati, le prestazioni sono misurate non soltanto con l'accuracy, ma anche con ROC-AUC, precision, recall, F1-score e matrice di confusione. Queste metriche consentono un'interpretazione più significativa della capacità predittiva nel rilevamento del rischio ambientale.

```
# ------------------------------
# 8) Train/test split + Logistic Regression
# ------------------------------
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=0, stratify=y
)

logreg = LogisticRegression(max_iter=3000, class_weight="balanced")
logreg.fit(X_train, y_train)

proba_lr = logreg.predict_proba(X_test)[:, 1]
pred_lr = (proba_lr >= 0.5).astype(int)

print("LogReg ROC-AUC:", roc_auc_score(y_test, proba_lr))
print(classification_report(y_test, pred_lr, digits=3))
print("Confusion matrix:\n", confusion_matrix(y_test, pred_lr))
```

```
LogReg ROC-AUC: 0.9246512131422616
              precision    recall  f1-score   support

           0      0.952     0.820     0.881       945
           1      0.674     0.900     0.771       391

    accuracy                          0.844      1336
   macro avg      0.813     0.860     0.826      1336
weighted avg      0.871     0.844     0.849      1336

Confusion matrix:
 [[775 170]
 [ 39 352]]
```

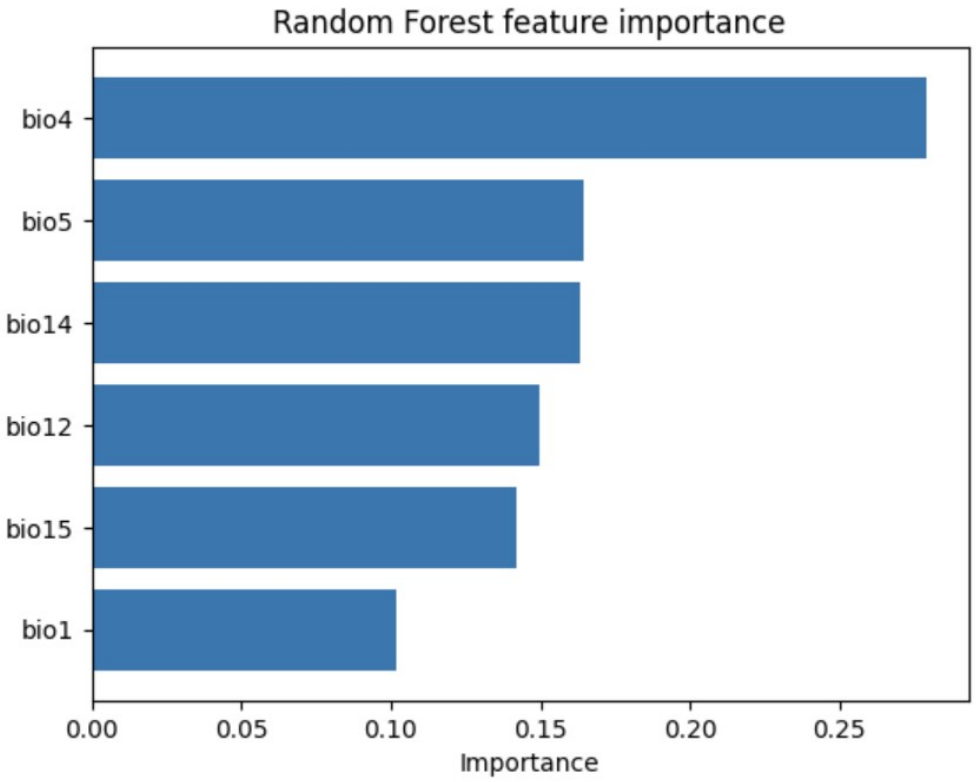
Random Forest e importanza delle variabili

La Random Forest viene introdotta come secondo modello perché le relazioni tra specie e ambiente sono spesso non lineari e possono comportare interazioni tra predittori. Questo algoritmo offre generalmente buone prestazioni con dataset ambientali tabulari e gestisce frontiere decisionali complesse. Inoltre, fornisce una forma pratica di interpretabilità tramite l'importanza delle variabili (feature importance), che evidenzia quali predittori climatici contribuiscono maggiormente a distinguere i punti di presenza da quelli di pseudo-assenza.

Ciò facilita l'interpretazione ecologica, ad esempio individuando associazioni con condizioni più calde, sensibilità alle precipitazioni della stagione secca o vincoli legati alla stagionalità. Nel complesso, questa fase rafforza l'approccio a un rilevamento del rischio interpretabile.

```
# ------------------------------
# 9) Random Forest + feature importance
# ------------------------------
rf = RandomForestClassifier(
    n_estimators=400,
    random_state=0,
    class_weight="balanced_subsample",
    n_jobs=-1
)
rf.fit(X_train, y_train)

proba_rf = rf.predict_proba(X_test)[:, 1]
pred_rf = (proba_rf >= 0.5).astype(int)

print("RF ROC-AUC:", roc_auc_score(y_test, proba_rf))
print(classification_report(y_test, pred_rf, digits=3))
print("Confusion matrix:\n", confusion_matrix(y_test, pred_rf))

# Importance plot
importances = rf.feature_importances_
order = np.argsort(importances)

plt.figure()
plt.barh([feature_cols[i] for i in order], importances[order])
plt.title("Random Forest feature importance")
plt.xlabel("Importance")
plt.show()
```

```
RF ROC-AUC: 0.988926778440845
              precision    recall  f1-score   support

           0      0.970     0.972     0.971       945
           1      0.933     0.928     0.931       391

    accuracy                          0.960      1336
   macro avg      0.952     0.950     0.951      1336
weighted avg      0.960     0.960     0.960      1336

Confusion matrix:
 [[919  26]
 [ 28 363]]
```



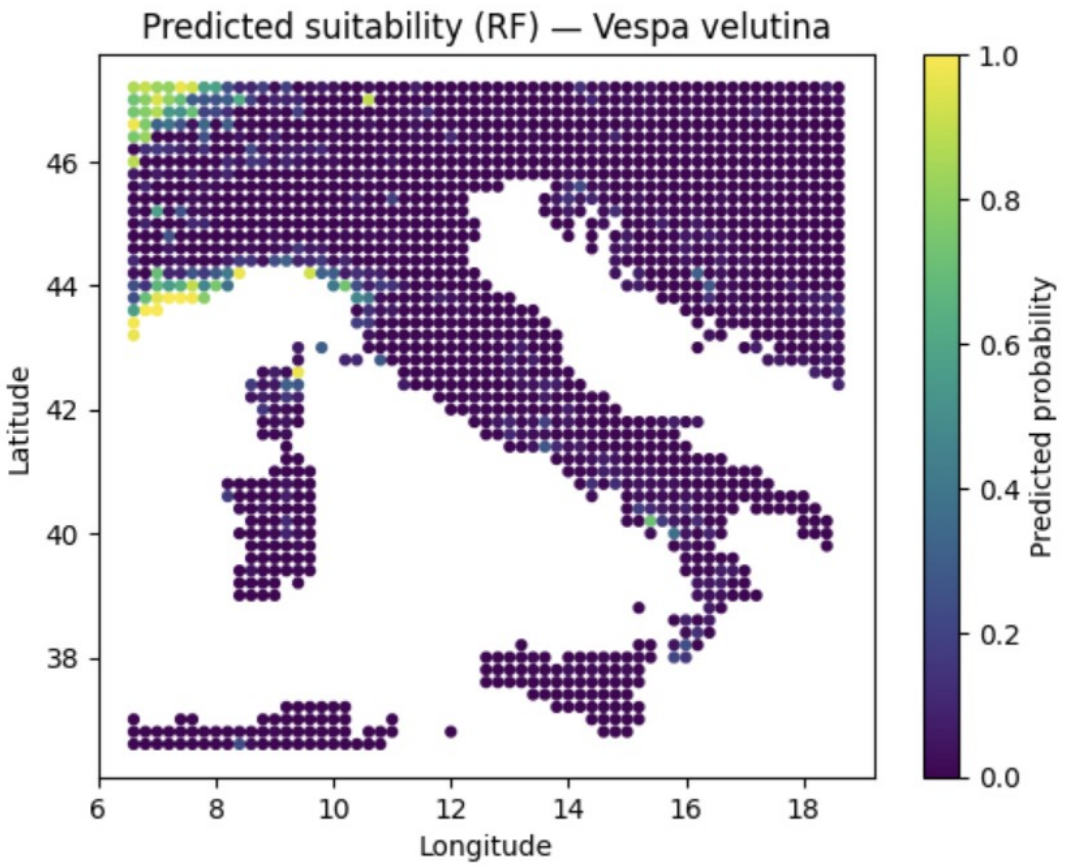
Il modello Random Forest ottiene prestazioni molto elevate su questo dataset, con una ROC–AUC di 0,99 e un'accuratezza complessiva di 0,96: i predittori climatici distinguono chiaramente i punti di presenza da quelli di pseudo-assenza. La matrice di confusione mostra pochi errori (26 falsi positivi e 28 falsi negativi) e il rapporto di classificazione conferma valori elevati di precision e recall per entrambe le classi. Nel grafico dell'importanza delle variabili, BIO4 (stagionalità della temperatura) risulta la più influente, suggerendo che la variazione della temperatura nel corso dell'anno sia un fattore importante per distinguere condizioni idonee e non idonee alla presenza di Vespa velutina. Tra gli altri predittori rilevanti figurano BIO5 (temperatura massima

del mese più caldo) e le variabili legate alle precipitazioni (BIO12, BIO14, BIO15), mentre BIO1 (temperatura media annua), da sola, sembra meno informativa. I risultati suggeriscono che l'idoneità dell'habitat per Vespa velutina dipenda non soltanto dai valori climatici medi, ma anche dalla stagionalità e dagli estremi, coerentemente con le aspettative ecologiche sull'insediamento delle specie invasive.

Predizione spaziale

Un vantaggio fondamentale della modellazione della distribuzione delle specie è la possibilità di applicare il modello addestrato a località prive di campionamento. Per mostrarlo, viene generata una griglia di punti sull'area di interesse e a ciascun punto viene assegnato un punteggio tramite il classificatore, producendo una superficie di predizione spaziale. La probabilità stimata può essere interpretata come indice di idoneità dell'habitat, indicatore approssimativo della probabilità di insediamento o criterio di priorità per il monitoraggio. Valori elevati indicano zone in cui la specie potrebbe plausibilmente persistere, anche in assenza di segnalazioni pregresse.

```
# ------------------------------
# 10) Optional: quick suitability map on a coarse grid
#     (no geopandas; just a scatter heatmap)
# ------------------------------
def make_grid(roi, step_deg=0.2):
    lons = np.arange(roi["lon_min"], roi["lon_max"] + 1e-9, step_deg)
    lats = np.arange(roi["lat_min"], roi["lat_max"] + 1e-9, step_deg)
    grid = [(x, y) for y in lats for x in lons]
    return pd.DataFrame(grid, columns=["decimalLongitude", "decimalLatitude"])

grid = make_grid(ROI, step_deg=0.2)
grid_feat = add_bioclim(grid, bio_files)
grid_feat = grid_feat.dropna(subset=feature_cols).copy()

Xg = grid_feat[feature_cols].to_numpy()
grid_feat["suitability"] = rf.predict_proba(Xg)[:, 1]

plt.figure()
plt.scatter(grid_feat["decimalLongitude"], grid_feat["decimalLatitude"], c=grid_feat["suitability"], s=12)
plt.title(f"Predicted suitability (RF) — {SPECIES_NAME}")
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.colorbar(label="Predicted probability")
plt.show()
```



Esercitazione

- Ripeti la procedura del tutorial scegliendo una specie diversa (ad esempio una specie invasiva, un impollinatore o una specie rilevante per il tuo territorio).
- Sostituisci SPECIES_NAME nel codice, modifica la ROI per la regione scelta e mantieni l'intervallo temporale 2000–2024. Esegui l'intera procedura: scarica i record GBIF (punti di presenza), pulisci le coordinate, genera punti di pseudo-assenza, estrai gli stessi predittori WorldClim (bio1, bio4, bio5, bio12, bio14, bio15) e addestra sia la regressione logistica sia la Random Forest. Assicurati che la ricerca GBIF restituisca un numero sufficiente di record (almeno circa 200 punti dopo la pulizia).
- Nel breve rapporto includi: (i) il numero di punti di presenza utilizzati; (ii) ROC–AUC e matrice di confusione per entrambi i modelli; (iii) il grafico dell'importanza delle variabili della Random Forest.

- Infine, rispondi alle domande seguenti: quale modello funziona meglio e perché? Quale variabile BIO è la più importante per la specie scelta? I risultati suggeriscono che la specie sia condizionata maggiormente dalla temperatura oppure dalle precipitazioni?

Verifica dell'apprendimento

1. Qual è l'obiettivo principale dei modelli di classificazione nel rilevamento dei rischi ambientali?
   A. Predire variabili ambientali continue, come temperatura o concentrazioni di inquinanti
   B. Raggruppare le osservazioni ambientali in cluster privi di etichette
   C. Assegnare le osservazioni a categorie di rischio predefinite (ad esempio pericoloso o sicuro)
   D. Ridurre la dimensionalità dei dataset ambientali

2. Perché l'accuratezza complessiva spesso non è sufficiente per valutare i modelli di classificazione nelle applicazioni ambientali?
   A. Perché i dati ambientali sono sempre rumorosi
   B. Perché l'accuracy non può essere calcolata per problemi multiclasse
   C. Perché i dataset ambientali sono spesso molto sbilanciati e gli eventi pericolosi sono rari
   D. Perché l'accuracy non tiene conto dell'interpretabilità del modello

3. Quale metrica è più importante se si vuole ridurre al minimo il numero di pericoli ambientali non rilevati (falsi negativi)?
   A. Precision
   B. Accuracy
   C. Recall (sensibilità)
   D. ROC–AUC

4. Quale algoritmo è particolarmente adatto a individuare sversamenti di petrolio o rifiuti marini nelle immagini satellitari o acquisite da droni?
   A. Regressione logistica
   B. Support Vector Machine

C. Random Forest
D. Reti neurali convoluzionali (CNN)

5. Quale descrizione rappresenta meglio il ruolo dei risultati dei modelli di classificazione nella gestione dei rischi ambientali?
   A. Sostituiscono completamente il giudizio degli esperti
   B. Forniscono previsioni esatte delle condizioni ambientali future
   C. Supportano le decisioni identificando e assegnando priorità alle categorie di rischio
   D. Eliminano l'incertezza nei sistemi di monitoraggio

Fonti aggiuntive

- Tutorial introduttivi sulla previsione con Python (ad esempio scikit-learn e pandas).
- Machine Learning - A First Course for Engineers and Scientists, Andreas Lindholm, Niklas Wahlström, Fredrik Lindsten, and Thomas B. Schön
- Deep Learning with Python, Second Edition, François Chollet
- Applications of Deep Neural Networks with Keras, Jeff Heaton
- Understanding Machine Learning: From Theory to Algorithms, Shai Shalev-Shwartz and Shai Ben-David
- Understanding Machine Learning: From Theory to Algorithms, Shai Shalev-Shwartz and Shai Ben-David

Bibliografia

[1] El-Shafeiy, E., Alsabaan, M., Ibrahem, M. I., & Elwahsh, H. (2023). Real-time anomaly detection for water quality sensor monitoring based on multivariate deep learning technique. Sensors, 23(20), 8613.

[2] Hu, X., Luo, H., Guo, M., & Wang, J. (2022). Ecological technology evaluation model and its application based on Logistic Regression. Ecological Indicators, 136, 108641.

[3] Madani, A., Hagage, M., & Elbeih, S. F. (2022). Random Forest and Logistic Regression algorithms for prediction of groundwater contamination using ammonia concentration. Arabian Journal of Geosciences, 15(20), 1619.

[4] Maxwell, A. E., Warner, T. A., & Fang, F. (2018). Implementation of machine-learning classification in remote sensing: An applied review. International Journal of Remote Sensing, 39 (9), 2784–2817.

[5] Pichler, M., & Hartig, F. (2023). Machine learning and deep learning—A review for ecologists. Methods in Ecology and Evolution, 14 (4), 994–1016.

[6] Scholkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. Neural Computation, 13(7), 1443–1471.

[7] Zinger, L., Bonin, A., Alsos, I.G., Ba´lint, M., Bik, H., Boyer, F., Chariton, A.A., Creer, S., Coissac, E., Deagle, B.E. & De Barba, M., (2019). DNA metabarcoding—Need for robust experimental designs to draw sound ecological conclusions. Molecular ecology, 28(8), pp.1857-1862.