Sezione 2 – classificazione usando i modelli di machine learning : SVM, RF & LDA

Dopo aver concluso la sezione relativa alla creazione della tabella partendo dalle coordinate di latitudine e longitudine degli ulivi risulta necessario classificarli utilizzando appositi algoritmi di apprendimento automatico.

A tal proposito sono stati utilizzati 3 algoritmi per effettuare la classificazione:

1. SVM (Support Vector Machine)
2. RF (Random Forest)
3. LDA (Linear Discriminant Analysis)

Prima di spiegare in modo dettagliato il funzionamento e la logica di ognuno di questi algoritmi e, quindi, passare all’applicazione degli stessi sono stati effettuati dei passaggi preliminari di preprocessing del dataset a disposizione.

Il dataset in questione (afferente agli ulivi) è costituito dalle coordinate di latitudine e longitudine originali degli ulivi, dalle colture corrispondenti (ogliarola barese, nociara, leccino, altro) e dalle 47 bande di frequenza.

Tuttavia, risulta necessario precisare che non sono stati presi in considerazione tutti i 329 alberi del dataset originale di partenza, ma solo i 177 ottenuti dopo il processo di estrazione delle chiome trattato nella prima parte della relazione.

Dal dataset risultante (tabella Excel) sono state utilizzate tutte le 47 bande di frequenza e le relative colture per procedere con la classificazione degli ulivi.

Dapprima è stata effettuata la normalizzazione usando la z-score normalization sulle varie bande singolarmente di tutte le 47 a disposizione.

Tale dataset è stato suddiviso dapprima in X e Y dove X rappresentano i dati del modello (bande) mentre Y le labels (colture).

Successivamente è stato partizionato il dataset usando la strategia di ‘Holdout’ che prevede di dividere il dataset una sola volta rispettivamente in training set e test set, in particolare la divisione è stata fatta considerando l’80% dei data points come training set e il restante 20% come test set; infine per preservare la proporzione tra le varie classi (ogliarola barese, nociara, leccino, altro) e favorirne la classificazione è stata utilizzata la stratificazione.

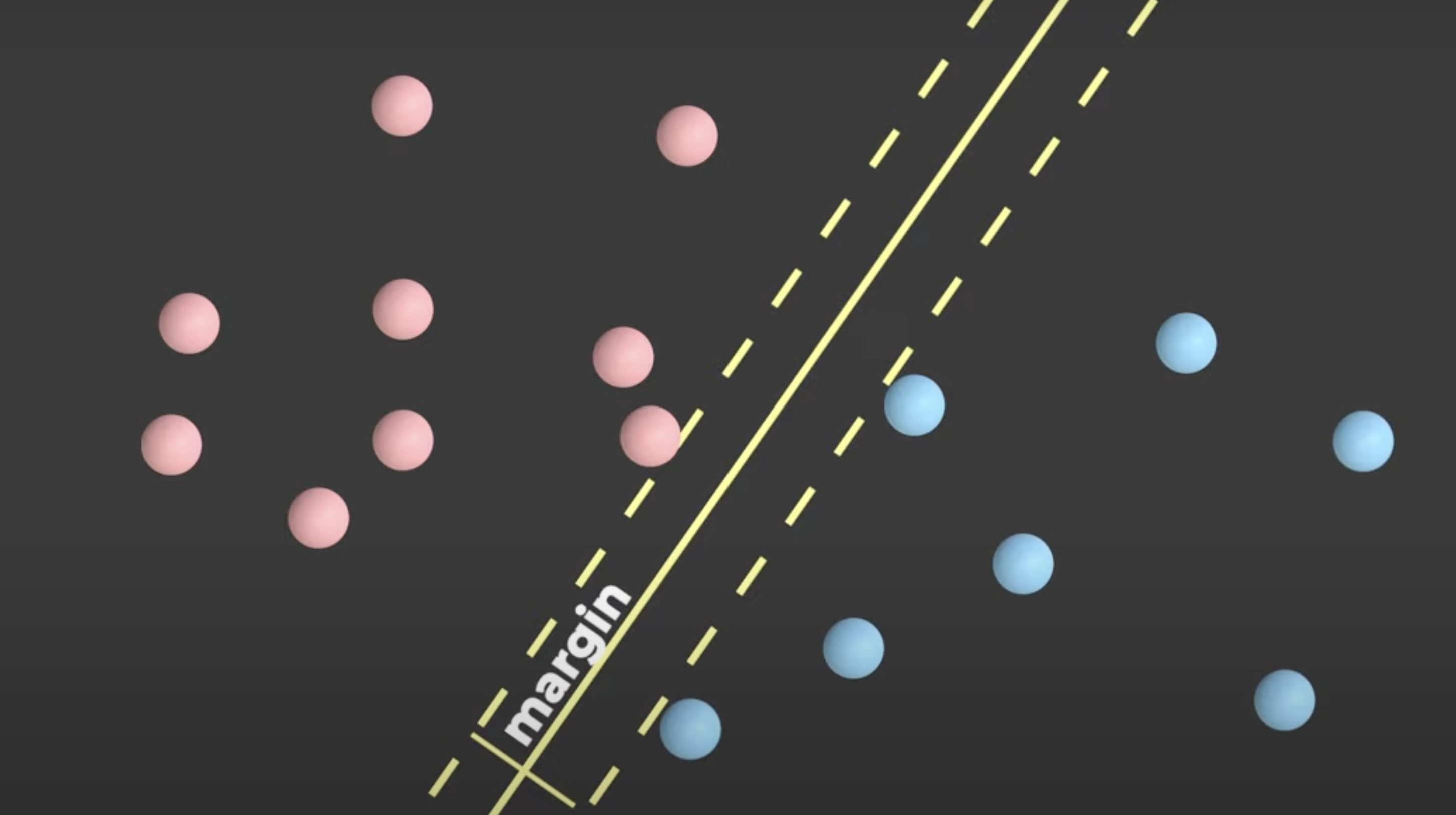
Solo aver effettuato il preprocessing del dataset è stato possibile passare all’applicazione dei modelli di machine learning.

Il primo modello scelto è stato SVM (Macchina a vettori di supporto), il cui algoritmo è descritto nel dettaglio nel paper ‘’Support Vector Network’’ pubblicato nel 1995 da

In estrema sintesi l’idea proposta nel paper è quella di trovare un iperpiano separatore che massimizzi la distanza tra le varie classi di dati.

(nel nostro caso tale iperpiano è costituito dalle 4 classi di interesse ovvero: ogliarola barese, nociara, leccino, altro).

Tale iperpiano viene tracciato solo dopo aver rappresentato in uno spazio altamente dimensionale le features.



Tale figura mostra l’idea dell’algoritmo, il pallino rosso e quello blu giacenti sulla linea tratteggiata sono detti ‘vettori di supporto’ mentre il margine è la linea retta che massimizza la distanza tra i vettori di supporto.

Tale algoritmo è stato implementato utilizzando la libreria offerta da Matlab e sono stati testati 4 modelli di SVM che differiscono a seconda della funzione di kernel (kernel gaussiano, polinomiale e rbf) utilizzate.

Le funzioni di kernel sono estremamente utili quando si vuole realizzare una classificazione di dati non linearmente separabili; tuttavia, nel nostro caso specifico, è risultato più performante utilizzare SVM senza l’uso del kernel trick.

Il secondo modello utilizzato è stato il Random Forest che rappresenta un’estensione del classification tree singolo, superandone le limitazioni prestazionali.

Nell’algoritmo random forest vengono generati più alberi di classificazione per evitare l’overfitting (ovvero l’incapacità di generalizzare la previsione su nuovi dati) provocato dall’uso di un solo albero di classificazione al tal proposito il nome di ‘foresta’.

Il termine ‘casuale’ deriva dal fatto che i ‘bootstrap’ datasets (ovvero i dataset costruiti a partire dal dataset originale) su cui sono addestrati gli altri alberi di classificazione ottenuti casualmente effettuando un random sampling con reinserimento dal dataset originale.

I dati non presenti nel bootstrap dataset vengono utilizzati come validation dall’albero di classificazione addestrato sul bootstrap corrente.

Tale algoritmo viene spiegato nel dettaglio nel paper: ‘Random Forest’ pubblicato da Leo Breiman nel 2001.

Immagine che contiene schermata, Policromia, tabellone

Descrizione generata automaticamente

Tale figura chiarisce il processo di creazione dei bootstrap datasets su cui sono addestrati i vari alberi di classificazione, ad esempio se consideriamo il primo boostrap dataset (prima colonna) verrà usato come test il primo data point e il terzo data point.

Il numero di alberi di classificazione da utilizzare per creare la foresta casuale viene scelta a priori dall’utente mediante cross-validation per evidenziare la configurazione ottimale della classificazione finale.

Infine, l’ultimo algoritmo utilizzato per la classificazione è stato LDA (Linear Discriminant Analysis).

Tale algoritmo è stato teorizzato ed è spiegato nel dettaglio nel paper di R. A. Fisher : ‘The Use of Multiple Measurements in Taxonomic Problems’.

A differenza dei precedenti algoritmi LDA si concentra sul trovare un nuovo piano dimensionalmente minore rispetto a quello iniziale, in cui risulta più semplice effettuare la classificazione.

Il termine ‘linear’ fa riferimento al tipo di classificatore che separa le classi il quale è lineare esattamente come nel SVM senza l’uso del kernel trick, questo rappresenta il metodo standard dell’algoritmo LDA in cui viene assunta che la variabilità dei dati all’interno di ciascuna classe sia uguale in ogni direzione.

Il termine ‘discriminant’ fa riferimento alla formula del discriminante per la risoluzione dell’algoritmo.

L’ esecuzione dell’algoritmo per la classificazione degli ulivi è stata eseguita utilizzato il discriminante lineare (‘DiscrimType : linear’), il discrimante diagonale (in cui viene assunto la matrice di covarianza delle classi siano matrici diagonali, (‘DiscrimType : diaglinear’) e infine il discriminante pseudolineare che si colloca come compromesso degli approcci precedenti. (‘DiscrimType : pseudolinear’).

Il risultato migliore in termini di accuratezza e precisione è stato ottenuto utilizzando il discrimante di tipo lineare.

Immagine che contiene testo, cerchio, schermata, Elementi grafici

Descrizione generata automaticamente

Esempio di LDA calcolata utilizzare il discrimante lineare.