Metodi di machine learning per la classificazione di tipi di foreste

Giulia de Innocentiis[1,2*] *mat*.865084 Giorgia Faccanoni[1,3] *mat*.871869

8 Luglio 2024

Sommario

Nel presente articolo sono stati testati metodi di machine learning per la classificazione di appezzamenti di terreno in diverse classi attraverso algoritmi come il Random Forest, Support Vector Machine e le reti neurali. Si sono ottenuti risultati soddisfacenti con tutti gli algoritmi, ma l'esito migliore è stato riscontrato attraverso Support Vector Machine ottenendo un'accuracy del 84.73%. Inoltre è stata sviluppata un'analisi di clustering utilizzando modelli come l'Iterative distance-based, EM-mixture model e DBSCAN con l'obiettivo di individuare gruppi omogenei dei dati. Gli algoritmi EM-mixture-model e DBSCAN sono stati in grado di raggruppare le osservazioni in base alla loro posizione geografica in modo più ottimale rispetto al k-means.

1 Introduzione

Durante questo progetto sono state svolte delle analisi su dati che si riferiscono a quattro aree di terreno situate nella foresta nazionale Roosevelt nel nord del Colorado (USA). L'obiettivo è stato quello di classificare ogni singola osservazione in base alla tipologia di piante che si trovano in esse, ovvero individuare a quale diversa tipologia di foresta, tra le 7 presenti, appartenga ogni istanza. L'analisi è stata condotta addestrando algoritmi di machine learning, che attraverso le diverse variabili esplicative disponibili nel dataset, sono in grado di assegnare ogni osservazione alla tipologia di foresta a cui appartengono.

Un secondo obiettivo è stato quello di applicare algoritmi di tipo non supervisionato in grado di individuare gruppi omogenei al loro interno ed eterogenei tra loro. L'interesse di questa indagine è stata quella di individuare gruppi in cui i pezzi di terreno analizzati siano posizionati geograficamente vicini, ovvero appartengano a una delle 4 aree selvagge in cui è divisa la foresta nazionale Roosevelt.

2 Materiali

Il dataset proviene dall'Istituto Geologico (USGS) e dal servizio forestale degli Stati Uniti (USFS) e i dati si riferiscono a quattro aree selvagge situate nella foresta nazionale Roosevelt nel Colorado settentrionale. Queste aree vengono definite selvagge poiché non sono influenzate dall'uomo e i tipi di foreste identificate sono il risultato di processi ecologici naturali e non di pratiche di gestione forestale. Il dataset originario è composto da 581012 osservazioni, ma il dataset utilizzato in questa analisi è stato selezionato da una competizione su Kaggle ed è una versione ridotta di quella originale. Esso è composto da 15120 osservazioni e da 54 variabili, di cui 10 variabili quantitative, 4 variabili binarie per i tipi di aree selvagge e 40 variabili binarie per i tipi di suolo. Ciascuna osservazione si riferisce a un pezzo di terreno forestale 30mt x 30mt. L'obiettivo dell'analisi è quello di classificare ogni pezzo di terreno nella tipologia di foresta a cui appartiene.

Per ogni osservazione sono state rilevate le seguenti variabili:

1. *Elevation*: Altitudine sul livello del mare in metri;

*1: Data preprocessing, 2: Classification, 3: Clustering

2. *Aspect* : Angolo in gradi azimuth;
3. *Slope*: Pendenza in gradi;
4. *Horizontal_Distance_To_Hydrology*: Distanza orizzontale tra l'osservazione e il corso d'acqua più vicino in metri;
5. *Vertical_Distance_To_Hydrology*: Distanza verticale tra l'osservazione e il corso d'acqua più vicino in metri;
6. *Horizontal_Distance_To_Roadways*: Distanza orizzontale tra l'osservazione e la strada più vicino in metri;
7. *Hillshade_9am*: Rilievo dell'ombra alle 9.00: è espressa da un indice che assume valori tra 0 e 255;
8. *Hillshade_Noon*: Rilievo dell'ombra alle 12.00: è espressa da un indice che assume valori tra 0 e 255;
9. *Hillshade_3pm*: Rilievo dell'ombra alle 15.00: è espressa da un indice che assume valori tra 0 e 255;
10. *Horizontal_Distance_To_Fire_Points*: Distanza orizzontale tra l'osservazione e il punto di innesco degli incendi più vicino in metri;
11. *Wilderness_Area*: Tipologia di area selvatica: assume 4 valori diversi;
12. *Soil_Type*: Tipologia di suolo: assume 40 valori diversi;
13. *Cover_Type*: Tipologia di foresta: assume 7 valori diversi.

Le variabili quantitative utilizzate sono tutte di tipo topografico e in particolare si è analizzato il significato delle variabile *Aspect*, *Slope* e *Elevation*. Ogni valore delle variabili Aspect e Slope è stato calcolato con riferimento ad un punto di origine diverso, per ogni area selvatica di appartenenza. La *Slope* è calcolata come il rapporto della differenza tra l'altitudine del punto di origine e quella dell'osservazione e della distanza tra i due punti. L'*Aspect* rappresenta invece l'angolo azimutale tra il punto cardinale Nord e la posizione geografica di ogni osservazione. Per comprendere meglio questa variabile si fa riferimento alla Figura 3. In essa viene rappresentata l'ombra provocata dalla posizione del sole alle 15.00 del pomeriggio e si nota che assume valori opposti in base alla posizione del punto rispetto al nord, la quale è rappresentata attraverso la variabile *Aspect*.

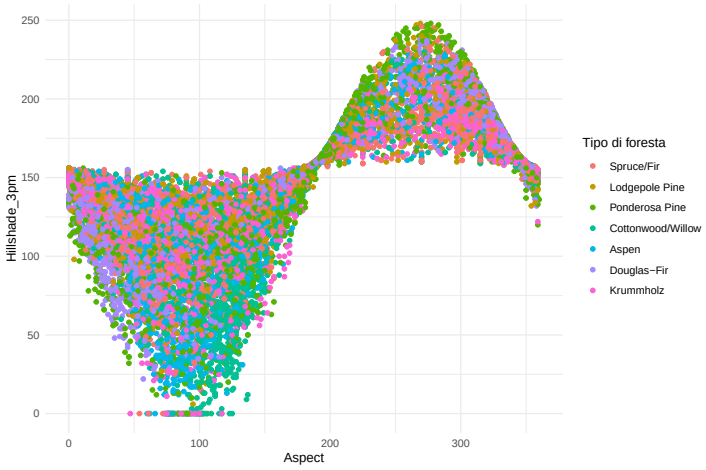

Figura 1: Scatterplot tra Aspect e Hillshade_3pm

La variabile Wilderness Area definisce la tipologia di area selvatica a cui appartiene l'osservazione e che può essere di 4 tipi:

1. Rawah;
2. Neota;
3. Comanche Peak;
4. Cache la Poudre.

La variabile Cover Type è la variabile target e definisce la tipologia di foresta a cui appartiene l'osservazione e può essere di 7 tipi:

1. Spruce/Fir
2. Lodgepole Pine
3. Ponderosa Pine
4. Cottonwood/Willow
5. Aspen
6. Douglas-Fir
7. Krummholz.

3 Risultati

Questa sezione è dedicata a mostrare i risultati ottenuti.

3.1 Data preprocessing

Gli obiettivi di quest'analisi sono due: il primo è quello di classificare le osservazioni in 7 differenti tipi di foreste e per questo è stato controllato e confermato che le classi fossero bilanciate. Il secondo obiettivo consiste nell'applicare algoritmi di clustering per raggruppare le osservazioni in gruppi omogenei tra loro; per questo secondo risultato sono state modificate alcune variabili del dataset come verrà illustrato di seguito. Innanzitutto, dopo aver accertato l'assenza di dati mancanti, sono state svolte alcune analisi descrittive dei dati. Per verificare la multicollinearità del dataset è stata calcolata la correlazione tra le variabili qualitative, ma nessuna di esse è fortemente correlata con le altre, quindi si è deciso di mantenerle tutte. Considerato che in quest'analisi sono stati utilizzati metodi di clustering basati sulle distanze è stato opportuno normalizzare le variabili quantitative.

Si è passato poi ad un'analisi di ogni singola variabile. Il grafico in Figura 2 rappresenta la distribuzione della variabile *Elevation* separata per ogni classe e si può notare che ogni diversa tipologia di foresta si trova ad un'altitudine sul livello del mare diversa. Si può inoltre notare che la classe 3 e 6 assumono valori molto simili tra loro e anche la classe 1 e 2 non si differenziano di molto, il problema della differenziazione tra queste classi verrà poi riscontrato negli algoritmi utilizzati.

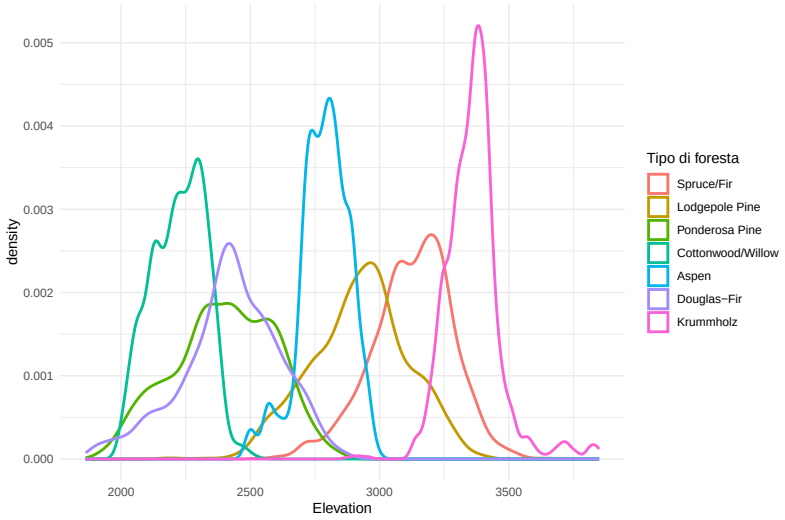

Figura 2: Distribuzione di Elevation per classe

Un'altra variabile in cui si nota questa forte differenza tra le classi è la *Horizontal_distance_to_Roadways* ovvero la distanza in metri di ogni terreno analizzato dalla strada asfaltata più vicina, questo è mostrato in Figura 3. Anche in questo caso si nota che le classi che assumono valori simili sono la prima, la seconda e la settima che si trovano ad una distanza dalla strada superiore a 1 km; mentre il resto delle classi si trova ad una distanza inferiore.

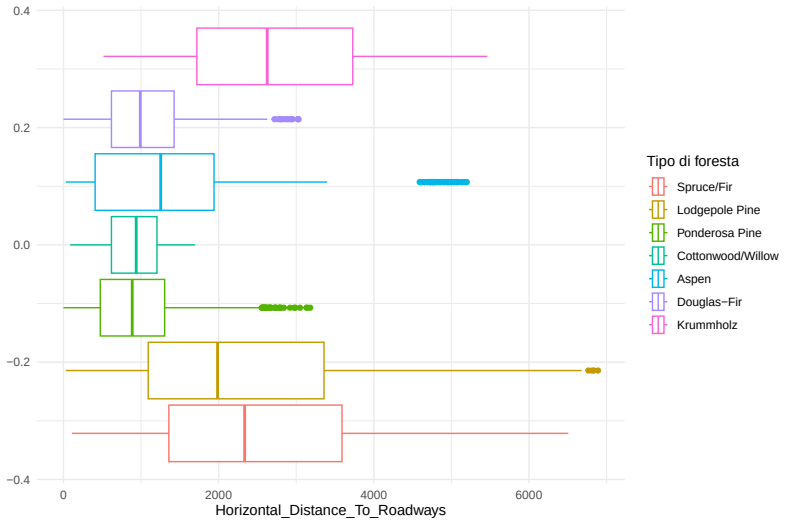

Figura 3: Distribuzione di Horizontal Distance To Roadways per classe

Per quanto riguarda l'obiettivo di clustering, si è modificato il dataset aggiungendo delle variabili propedeutiche alla tipologia di algoritmi utilizzati. Gli algoritmi di clustering si basano sul concetto di distanza tra osservazioni e per migliorare il funzionamento di questi algoritmi sono state aggiunte al dataset due variabili: le coordinate geografiche in formato UTM di ogni osservazione, ricavate a partire dalle variabili *Slope*, *Aspect* ed *Elevation*. Procedendo all'analisi esplorativa del nuovo dataset si è subito notato tramite la rappresentazione geografica, mostrata in Figura 4, la distinzione dei punti in quattro aree che corrispondono alle quattro aree identificate dalla variabile *Wilderness_Area*. Tuttavia, il dataset a disposizione contiene osservazioni fortemente sbilanciate rispetto a questa variabile, percui si

è creato un nuovo dataset, a partire dal dataset originario, con 2000 osservazioni per ogni area selvaggia di appartenenza. Il dataset ottenuto in questo modo ha quindi 8000 osservazioni.

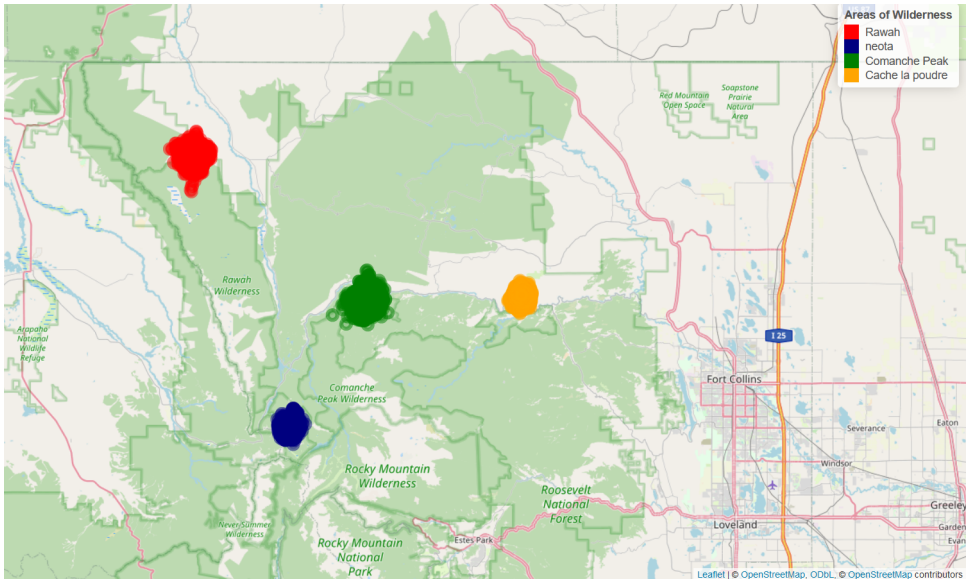

Figura 4: Mappa dell'area del parco Roosevelt

3.2 Classification

L'obiettivo di classificazione in quest'analisi consiste nel classificare ogni osservazione nella categoria di foresta di appartenenza, identificata dalla variabile *Cover_Type*, sulla base delle restanti. Per questo scopo sono stati utilizzati algoritmi di apprendimento supervisionati che, attraverso dati etichettati, sono in grado di classificare e fare previsione. La tecnica supervisionata cerca di identificare le relazioni tra variabili indipendenti e dipendenti e di costruire un modello che mostri queste dipendenze. Si è deciso di dividere il dataset composto da 15120 osservazioni in training e test set, includendo nel training set il'90% delle osservazioni. Inoltre, per cercare gli iperparametri in grado di minimizzare l'errore di generalizzazione sono stati utilizzati metodi di Repeated Cross Validation sul training set.

3.2.1 Random Forest

Il Random Forest è un algoritmo composto da una collezione di alberi decisionali e strutturato in due fasi principali. Una prima fase di *bootstrapping*, in cui vengono creati nuovi training bootstrap di eguale dimensione al dataset originale tramite campionamento con reinserimento delle osservazioni e un sottocampione casuale delle covariate del dataset. Una successiva fase di *aggregation* in cui ogni nuova osservazione viene assegnata alla classe di appartenenza per ogni albero creato e viene assegnata alla classe maggioritaria. Con questo algoritmo sono stati ottenuti risultati ottimali utilizzando come valori degli iperparametri quelli di default, ovvero un numero di alberi nella foresta apri a 500 e un numero di variabili da campionare nella fase di *bootstrapping* pari alla radice quadrata delle variabili complessive, ovvero 4. Si è ottenuta un'accuracy sul training set pari al 100%, e un'accuracy sul test del 78.18%. Per valutare quali classi sono le più problematiche da identificare si è calcolata la metrica *F1-score* definita come la media armonica tra la precisione e il recall. Essa viene utilizzata come misura statistica per valutare le prestazioni di un modello e indica la capacità equilibrata del modello di individuare i casi della classe di interesse (*recall*) e di essere accurato con i casi che individua (*precision*). Può assumere valori compresi tra 0 e 1, se il punteggio F1 aumenta, significa che il modello ha aumentato le performance per

la *precision*, per il *recall* o per entrambi. La metrica *F1-score* si calcola attraverso la seguente formula:

$$F1\text{-}score = \frac{2 \times precision \times recall}{precision + recall}.$$

I valori della metrica calcolati per ogni tipologia di foresta ottenuti sono riportati in Tabella 1. Si può notare che la foreste di tipo *Aspen* sono quelle che riescono ad essere classificate meglio mentre le foreste di tipo *Lodgepole Pine* e *Ponderosa Pine* presentano i valori più bassi. Nelle Figure 2 e 3 si può notare infatti come la categoria *Aspen* abbia una distribuzione per la variabile Elevation completamente distinta dalle altre e anche per le altre variabili ha valori che si distinugono. La classe *Lodgepole Pine* ha caratteristiche molto simili alla classe *Spruce Fir* mentre la classe *Ponderosa Pine* ha caratteristiche simili alla classe *Douglas Fir*. Risulta quindi difficile per queste due classi essere discriminate dalle loro simili con conseguenza di un alto numero di falsi positivi e falsi negativi e quindi una diminuzione delle metriche *recall* e *specificity*.

Tipo di Foresta	F1 Score
Spruce/Fir	0.8637
Lodgepole Pine	0.8016
Ponderosa Pine	0.8351
Cottonwood/Willow	0.8636
Aspen	0.9183
Douglas-Fir	0.8861
Krummholz	0.8727

Tabella 1: Punteggio *F1 Score* nel Random Forest

Per valutare la capacità discriminatoria delle variabili è stata inoltre calcolata una misura di importanza delle variabili. Per ogni variabile questa misura è calcolata come la diminuzione media su tutti gli alberi dell'impurità dei nodi creati dividendo in base alla stessa variabile. Nell'analisi di classificazione l'impurità dei nodi è misurata dall'indice di Gini. Come si può notare in Figura 5, le variabili con più potere discriminatorio sono l'altitudine sul livello del mare, *Elevation*, come ci si poteva aspettare dall'analisi esplorativa iniziale del dataset, e la tipologia del suolo, *soil*.

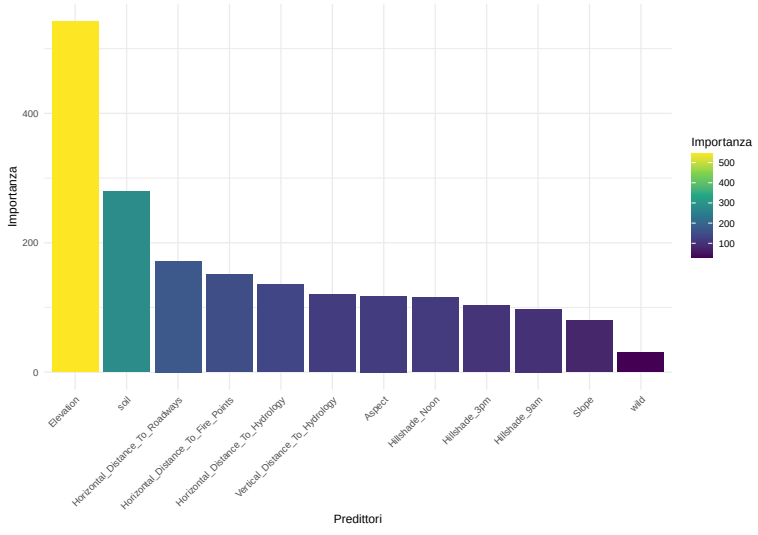

Figura 5: Importanza dei predittori nel Random Forest

3.2.2 Support Vector Machine

Le Support Vector Machine(SVM) sono un algoritmo che ha l'obiettivo di separare i dati tramite degli iperpiani. In questo caso i dati non sono linearmente separabili quindi si utilizzano le funzioni kernel per mappare le osservazioni in un *feature space* dove possono essere separate linearmente. Inoltre si è utilizzata una versione soft dell'algoritmo che prevede la presenza di un errore che tenga conto della misclassificazione dei dati. Per un algoritmo di questo tipo sono necessari quindi due iperparametri: la funzione kernel e il fattore di regolalizzazione C. Si è utilizzato un approccio di tipo Automated Machine Learning per selezionarli e si sono ottenuti i risultati migliori in corrispondenza di un kernel di tipo radiale con parametro γ=0.121 e fattore di regolarizzazione C=63.9. Con questi parametri si è ottenuto un errore empirico molto basso, pari a 0.04, e un errore di generalizzazione pari a 0.1664. Nonostante la differenza tra i due il modello non ricade in problematiche di overfitting dato che nel test set si ottiene un'accuracy pari al 84.73%, valore più elevato di quella ottenuta nel Random Forest e inoltre il numero di support vectors identificati dall'algoritmo sono 6752, circa la metà del numero di osservazioni. Come per il Random Forest, anche in questo caso si è voluto verificare la capacità di discriminazione tra le classi del modello tramite la metrica *F1score*. Dai risultati mostrati in Tabella 3 si può notare come in questo caso i risultati in generale siano più alti; le foreste di tipo *Krummholz* e *Cottonwood/Willow* sono quelle classificate meglio, con un valore della metrica quasi pari a 1. I valori della metrica più bassi si osservano nella classe *Lodgepole Pine* e nella classe *Spruce Fir* che rimangono le più problematiche da classificare. Per quanto riguarda la discriminazione tra la classe *Ponderosa Pine* e la classe *Douglas Fir* questo algoritmo ha una capacità discriminatoria più elevata rispetto al Random Forest.

Tipo di Foresta	F1 Score
Spruce/Fir	0.8181
Lodgepole Pine	0.8331
Ponderosa Pine	0.8985
Cottonwood/Willow	0.9607
Aspen	0.9543
Douglas-Fir	0.8821
Krummholz	0.9630

Tabella 2: Punteggio *F1 Score* nelle SVM

3.2.3 Reti Neurali

Come ultimo algoritmo di classificazione, sono state utilizzate le reti neurali. Le reti neurali sono algoritmi composti da nodi (*neuroni*) e connessioni tra essi, che in base alla loro struttura possono essere di diversi tipi. A partire da un strato di input, in cui i neuroni sono rappresentati dalle variabili del dataset grezzo, vengono costruite connessioni tra i neuroni negli strati nascosti (*hidden layers*) tramite le funzioni di attivazione scelte e vengono poi restituite nello strato finale (*softmax layer*), in un problema di classificazione, le probabilità di appartenenza dell'istanza a ogni classe. La potenza delle reti neurali risiede proprio nell'uso di funzioni di attivazione complesse che sono in grado di cogliere relazioni non lineari tra i dati. Per questa analisi è stato utilizzato il dataset composto solo da variabili numeriche ed è stata allenata una rete neurale con 2 strati nascosti. Dopo diversi tentativi si sono ottenuti risultati ottimali con una rete neurale definita dagli iperparametri mostrati in Tabella 3.

Iperparametri	Valori
Numero di Hidden Layer	2
Funzione di attivazione	ReLU
Dropout	0.4
Epoche	100
Batch Size	30
Validation Size	20%
Numero di neuroni per layer	50
Funzione di perdita	Categorial crossentropy
Algoritmo discesa del gradiente	ADAM

Tabella 3: Iperparametri per la rete neurale

Sul training si è ottenuta una perdita pari a 0.7238 e un'accuracy del 70.09%, con un errore empirico pari quindi a 0.2991. Sul validition si è ottenuta invece una perdita simile, pari a 0.8376 e un'accuracy del 58.89%, con un'errore di generalizzazione pari a 0.411. Dopo aver fatto diagnostica sulla rete neurale e aver controllato l'assenza di pesi nulli e di parti della rete danneggiate, si è applicato il modello sul test set, ottenendo un'accuracy pari al 71.83%. Rispetto agli altri algoritmi di classificazione, le reti neurali hanno una capacità di generalizzazione più alta poiché l'accuracy sul training e sul test sono molto simili. Tuttavia, come mostrato in Tabella 4, per le classi problematiche da classificare, ovvero *Lodgepole Pine* e *Ponderosa Pine*, il modello ha performance peggiori rispetto al Random Forest e alle Support Vector Machine, ottenendo per queste due classi un punteggio F1 molto basso.

Tipo di Foresta	F1 Score
Spruce/Fir	0.7378
Lodgepole Pine	0.6262
Ponderosa Pine	0.5264
Cottonwood/Willow	0.9302
Aspen	0.9265
Douglas-Fir	0.8385
Krummholz	0.9660

Tabella 4: Punteggio *F1 Score* nelle reti neurali

3.3 Clustering

Nell'analisi di clustering è stato utilizzato il dataset contenente 8000 osservazioni e con l'aggiunta delle nuove variabili, latitudine e longitudine. L'obiettivo dell'analisi è stato quello di individuare dei cluster che contenessero osservazioni simili tra loro e analizzare i dati all'interno di ogni gruppo in modo da individuare il motivo per cui gli algoritmi hanno unito certe osservazioni rispetto ad altre. In questa parte sono stati utilizzati metodi di machine learning non supervisionati, in cui la variabile target è stata ignorata e trattandosi di algoritmi basati sulla distanza, sono state utilizzate soltanto variabili quantitative in cui i dati sono stati standardizzati. Come mostrato dalla mappa precedente in Figura 4, le osservazioni risultano ben separate rispetto alla loro area di appartenenza, per questo motivo i risultati mostrati sono riferiti alla variabile *Wilderness_Area*. I metodi utilizzati sono stati:

3.3.1 Iterative distance-based

L'iterative distance-based è un algoritmo di tipo *crispy* ovvero che assegna ogni osservazione ad un solo cluster e non si basa sulla probabilità di appartenenza ad ogni gruppo. Si è utilizzato un algoritmo di tipo *k-means*, il quale si basa sulla distanza euclidea calcolata ad ogni passo tra le osservazioni e i

rappresentanti di ogni gruppo. L'unico iperparametro da scegliere in questo caso è stato k, ovvero il numero di cluster, che sulla base della metrica Silhouette è stato scelto pari a 4. I risultati dell'algoritmo sono mostrati in Figura 6, si può notare come l'algoritmo è in grado di individuare 4 gruppi, due dei quali rappresentano in modo ottimale le aree selvagge *Rawah* e *Cauche la Poudre*, mentre il terzo e quarto gruppo non riescono a distinguere tra le osservazioni appartenenti all'area *Neota* e all'area *Cauche la Poudre*. Questo problema può essere dovuto dal fatto che le due aree selvagge che l'algoritmo non riesce ad individuare sono più vicine tra loro rispetto alle altre aree, come si può vedere dalla mappa in Figura 4. Inoltre la prima e l'ultima classe assumono valori diversi in alcune variabili esplicative. Ad esempio la variabile *Horizontal_Distance_To_Roadways* per l'area *Neota* assume un range di valori molto più ampio rispetto al resto delle aree. Mentre per quanto riguarda l'area *Cauche la Poudre* le variabili come *Horizontal_Distance_To_Fire_Points* e *Elevation* assumono valori molto diversi rispetto alle altre aree.

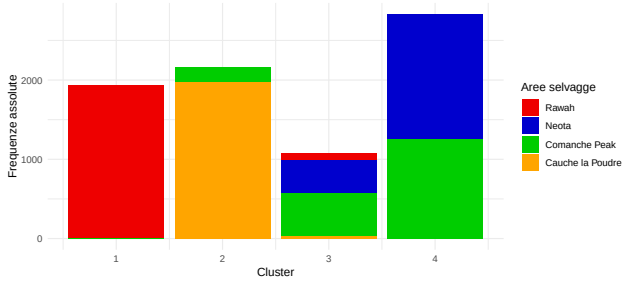

Figura 6: Cluster con K-means

3.3.2 EM-Mixture Model

Il metodo EM-Mixture model è un algoritmo probabilistico, in cui per ogni osservazione si calcola la probabilità di appartenere a un determinato cluster e si assegna il dato al gruppo con probabilità più elevata. L'idea di base dell'algoritmo è che i dati siano stati generati da una mistura di k funzioni di probabilità normali e come centroide di ogni k-esimo gruppo venga considerata la media della normale. I cluster formati possono assumere forme ellittiche a differenza dell'algoritmo k-means che si basa su gruppi a forma sferica. L'unico iperparametro da inserire è stato il numero di gruppi k impostato a 4, scelto in base alla metrica Silhouette. Nel grafico in Figura 7 vengono mostrati i risultati, si nota che l'algoritmo crea 4 cluster che individuano perfettamente l'area selvaggia di appartenenza. Per evitare che questo fosse soltanto un caso si è verificato attraverso nuovi dataset con classi sbilanciate rispetto alle aree selvagge il funzionamento dell'algoritmo. Su tutti i nuovi dataset creati l'algoritmo EM crea cluster in cui vengono raggruppate osservazioni appartenenti alla stessa area selvaggia.

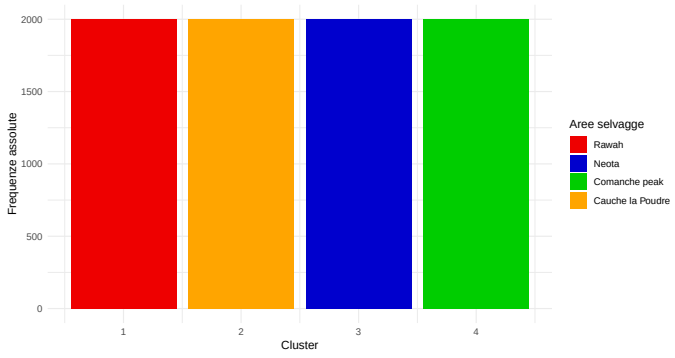

Figura 7: Cluster con EM

3.3.3 DBSCAN

L'ultimo algoritmo utilizzato per l'analisi di clustering è stato il DBSCAN. L'idea principale di quest'algoritmo è quella di individuare aree dense di punti, la densità in un punto è definita dal numero di punti che stanno all'interno della sfera centrata nel punto di raggio *epsilon*. La densità di ogni sfera definisce la classificazioni dei punti in:

1. *Core points*: quando il numero di punti all'interno della sfera è maggiore di una certa soglia *minPts*.
2. *Border points*: quando il numero di punti all'interno della sfera è minore di una certa soglia *minPts* e almeno uno di essi è un *Core point*.
3. *Noise points*: quando il numero di punti all'interno della sfera è minore di una certa soglia *minPts* e nessuno di essi è un *Core point*.

Una volta identificati i punti, l'algoritmo raggruppa i *Core points* ad una distanza minore di *epsilon*, mentre i *Border points* vengono classificati in base al *Core point* che contengono. In questo caso ci sono due iperparametri da impostare. *Epsilon* è stato fissato a 0.9 osservando dove il grafico delle distanze ordinate dell'algoritmo KNN ha un significativo cambio di pendenza. Mentre il parametro *minPts* si è scelto un valore pari al doppio del numero delle variabili utilizzare ovvero 14. I risultati del DBSCAN sono presentati in Figura 8. Si può notare che esso crea 4 gruppi che rappresentano in modo ottimale le 4 aree selvagge, mentre nel gruppo che identifica come 0 inserisce tutti i *Noise points*. In particolare individua perfettamente le osservazioni che appartengono all'area *Neota* a causa di alcune variabili che assumono valori diversi rispetto al resto delle osservazioni, ad esempio per la variabile *Elevation* i dati assumono valori in media più elevati poiché si trovano ad un altitudine superiore. Anche l'area *Cauche la Poudre* viene individuata in modo ottimale, infatti si può notare che per le variabili *Horizontal_Distance_To_Fire_Points* e *Elevation* assume valori diversi rispetto alle altre aree.

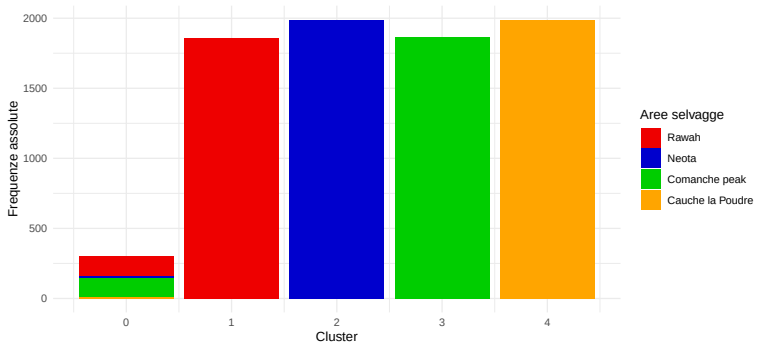

Figura 8: Cluster con DBSCAN

Nel seguente grafico in Figura 9 vengono evidenziati i *Noise points* rispetto alle coordinate geografiche e si può notare che essi stiano per la maggior parte ai lati dei 4 gruppi evidenziati. Non tutti sono così individuabili sui lati poiché il grafico è stato creato considerando solo due variabili, mentre l'algoritmo è stato implementato considerando 7 variabili.

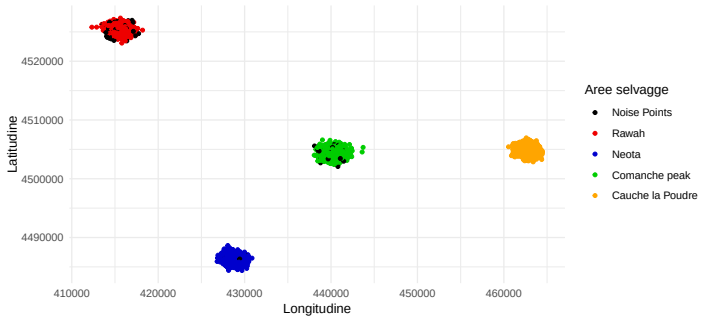

Figura 9: Noise points

3.4 Ulteriori risultati

Uno dei principali problemi riscontrati durante l'analisi di classificazione è stata la discriminazione delle foreste *Ponderosa Pine* dalle foreste *Douglas-Fir* e la distinzione delle foreste *Spruce/Fir* dalle foreste *Lodgepole Pine*. Dato che attraverso le SVM si sono ottenuti risultati migliori nella distinzione tra la classe *Ponderosa Pine* e la classe *Douglas-Fir*, si è deciso di focalizzarsi sulla distinzione delle due altre classi problematiche. Si è quindi condotta un'analisi di classificazione binaria solo per queste due classi tramite Random Forest e SVM. Tuttavia, sono stati osservati solo leggeri miglioramenti per cui si sono cercati i motivi di questo problema nella natura delle variabili, conducendo un'analisi esplorativa sulle osservazioni di queste due classi. Si è notato che le osservazioni classificate erroneamente presentano per diverse variabili, come *Elevation*, *Horizontal Distance To Roadways* e *Vertical Distance To Hydrology*, valori più vicini alla classe in cui sono stati classificate erroneamente rispetto alla classe di appartenenza reale. Ad esempio, in Figura 10, si sono confrontate in uno spazio bidimensionale, solo tramite le variabili *Elevation* e *Horizontal Distance To Roadways*, le osservazioni classificate correttamente e quelle classificate erroneamente. Si nota come i valori delle osservazioni delle due classi sono mischiati e quindi nessun algoritmo usato è in grado di separarli in modo adeguato.

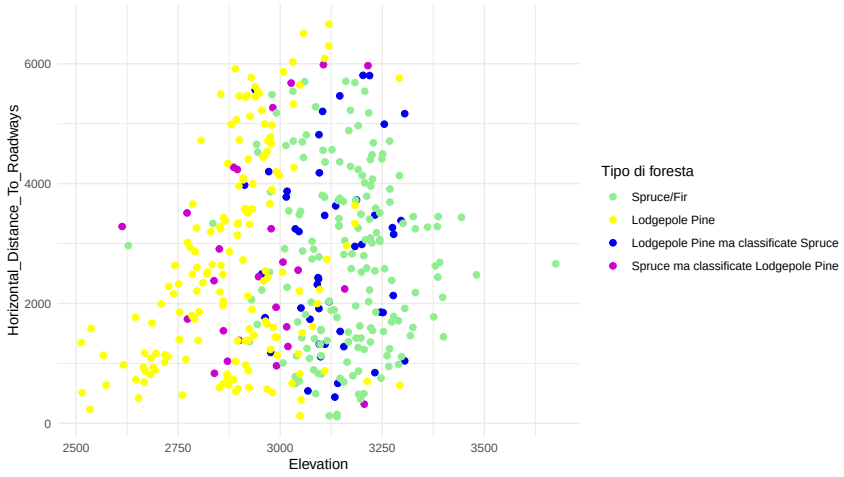

Figura 10: Scatterplot tra Elevation e Horizontal_Distance_To_Roadways

4 Conclusione

In questo articolo sono stati sviluppati 3 possibili soluzioni al problema di classificazione di ogni appezzamento di terreno nelle diverse tipologie di foreste. I risultati migliori sono stati ottenuti tramite

Support Vector Machine, con un'accuracy pari al 84.73% e una buona capacità discriminatoria tra tutte le classi.

Per quanto riguarda l'analisi di clustering, sono state sviluppate 3 tecniche diverse e tutti i modelli utilizzati hanno identificato relazioni tra le osservazioni, in modo da poterle raggruppare in diversi cluster. Dato che i metodi utilizzati si basano su distanze, è risultato evidente come i cluster costruiti sono fortemente influenzati dalla posizione geografica delle osservazioni.

Riferimenti bibliografici

[1] Competion Kaggle,

`https://www.kaggle.com/competitions/forest-cover-type-prediction/data?select=train.csv`

[2] Y. Idelbayev, *Assignment 1. Predicting cover of forest*,

`http://jmcauley.ucsd.edu/cse258/projects/wi15/Yerlan_Idelbayev.pdf`

[3] Sconosciuto,

`https://github.com/pratikbarjatya/INSAID-Assignment/blob/master/ML/ForestCoverTypePrediction/Forest%20Cover%20Type%20Prediction.ipynb`

[4] Dataset CoverType,

`https://archive.ics.uci.edu/dataset/31/covertype`.

[5] Sconosciuto, *Roosevelt National Forest*, Wikipedia,

`https://en.wikipedia.org/wiki/Roosevelt_National_Forest`.

[6] Sconosciuto, Coordinate UTM,

`https://www.youmath.it/domande-a-risposte/view/8085-coordinate-utm.html`.