Progetto Machine Learning

Group Name: I neuroni de-pensanti

Description of the domain and objectives

Il progetto si concentra sull'analisi dei vini, considerando le loro caratteristiche chimico-fisiche. Il dataset utilizzato contiene diversi campioni di vino, ciascuno descritto tramite variabili numeriche come il contenuto alcolico, il livello di acidità e la densità. Ogni riga del dataset rappresenta un singolo campione di vino con le proprie misurazioni specifiche.

L'obiettivo principale del progetto è utilizzare queste informazioni per costruire due modelli di machine learning in grado di prevedere il colore del vino, distinguendo tra vino rosso e vino bianco, e confrontare successivamente le prestazioni dei due modelli. Si tratta quindi di un problema di classificazione supervisionata, in cui i modelli devono imparare a riconoscere il colore del vino a partire dalle proprietà chimico-fisiche misurate. Questo consente di valutare quanto le caratteristiche dei vini siano informative per identificarne automaticamente la tipologia.

Design choices for creating the data set, hypotheses and assumptions

Il dataset utilizzato contiene numerosi campioni di vino con diverse caratteristiche chimico-fisiche. Per l'analisi è stata scelta come target la variabile colore.

Nel costruire il dataset sono state fatte alcune scelte progettuali e ipotesi: ogni campione è stato considerato indipendente, le misurazioni sono state assunte affidabili e si è ipotizzato che le feature disponibili fossero sufficientemente informative per consentire la classificazione tra vini rossi e bianchi.

Description of the exploratory analysis

Dopo aver effettuato una pulizia preliminare dei dati, rimuovendo tutte le osservazioni duplicate e verificando che non fossero presenti valori mancanti, il dataset risultante contiene 5320 osservazioni. Queste osservazioni sono descritte da 13 variabili: 11 covariate numeriche che rappresentano caratteristiche chimico-fisiche dei vini, il target binario `type` che indica se un vino è rosso o bianco, e una variabile di qualità, che in questo progetto non verrà utilizzata in quanto l'obiettivo principale è la classificazione tra vini rossi e bianchi.

Feature	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	type
Dtype	`float`	`float`	`float`	`float`	`float`	`float`	`float`	`float`	`float`	`float`	`float`	`int`	`object`

Table 1: Trasposizione delle proprietà del dataset

Un primo passo fondamentale nell'analisi esplorativa dei dati (Exploratory Data Analysis, EDA) è stato quello di osservare la distribuzione della variabile target. La distribuzione dei vini rossi e bianchi mostra un leggero sbilanciamento: circa il 75% delle osservazioni riguarda vini bianchi, mentre il 25% riguarda vini rossi, come evidenziato nella Figura 1. Questo sbilanciamento non è così marcato da compromettere l'uso di modelli di classificazione standard, ma è importante tenerlo presente, poiché potrebbe influenzare alcune metriche di valutazione, come l'accuratezza complessiva o nella scelta di metriche alternative (ad esempio F1-score o AUC).

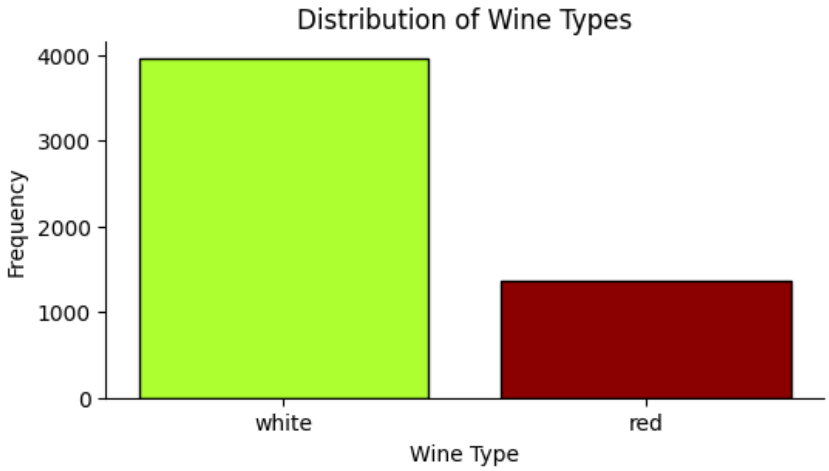

Figure 1: Distribuzione delle classi del target `type`.

Successivamente, l'analisi si è concentrata sulle covariate. Osservando le distribuzioni delle variabili, si nota che la maggior parte di esse è unimodale, cioè presenta un solo picco. Tuttavia, la forma delle distribuzioni non è simmetrica: molte covariate mostrano asimmetria positiva, il che significa che la maggior parte dei valori è concentrata nella parte bassa della scala, con alcuni valori più alti che rappresentano degli outlier. Questo fenomeno è comune nei dati chimici, dove alcune sostanze possono avere concentrazioni molto più elevate in pochi campioni rispetto alla maggioranza.

Inoltre, le covariate sono misurate su scale differenti. Alcune variabili hanno valori che si aggirano intorno all'unità, altre hanno valori più grandi o più piccoli di ordini di grandezza differenti. Questa differenza di scala è molto importante da considerare perché alcuni algoritmi di machine learning, come le reti neurali, sono sensibili alla scala delle variabili: se non vengono normalizzate, le variabili con valori più grandi potrebbero dominare l'apprendimento rispetto a quelle con valori più piccoli. La Figura 2 mostra le distribuzioni delle principali covariate del dataset.

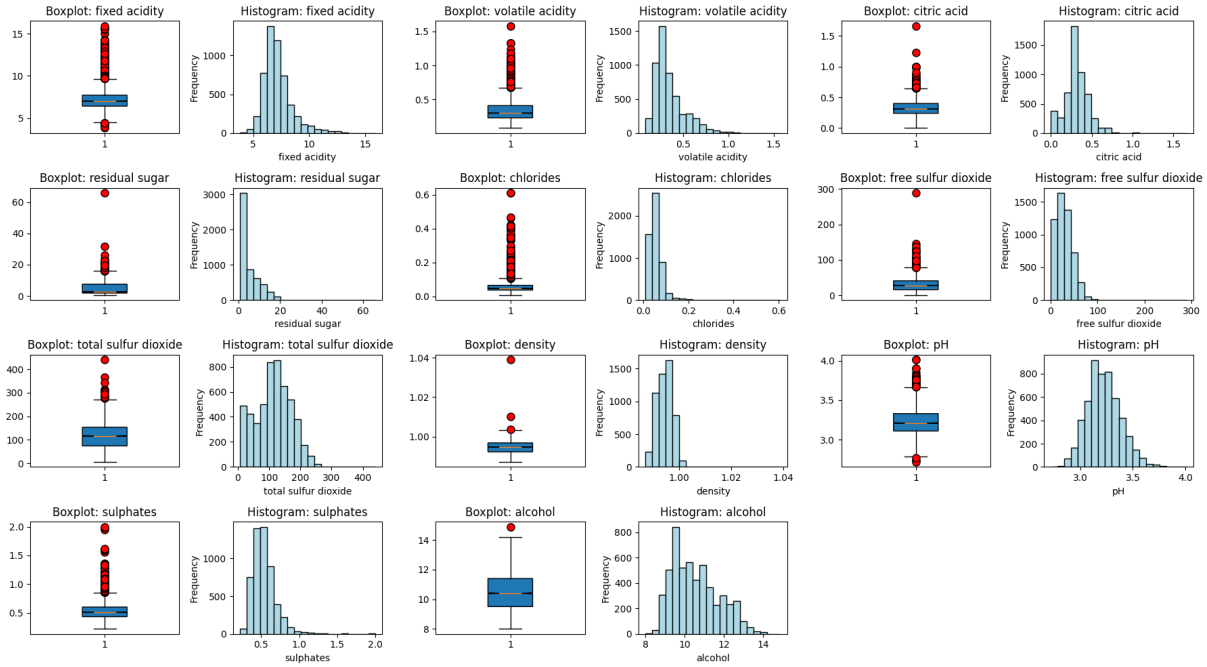

Figure 2: Distribuzioni delle covariate.

Per approfondire ulteriormente la comprensione del dataset, sono stati realizzati boxplot delle covariate separati per classe di vino (Figura 3). Questi grafici permettono di confrontare le distribuzioni dei

valori tra vini rossi e bianchi. Dai boxplot emerge che nessuna singola variabile è in grado di separare completamente le due classi: non esiste una covariata che assuma valori esclusivamente per i vini rossi o solo per i bianchi. Tuttavia, alcune covariate mostrano una certa capacità discriminante. In particolare, le variabili legate al contenuto di diossido di zolfo sembrano distinguere parzialmente le due classi: i vini rossi tendono ad avere valori più bassi, mentre i vini bianchi valori generalmente più elevati. Questo risultato è in linea con quanto si sa sul vino: i vini bianchi di solito contengono più diossido di zolfo perché ne hanno bisogno di più per conservarsi meglio.

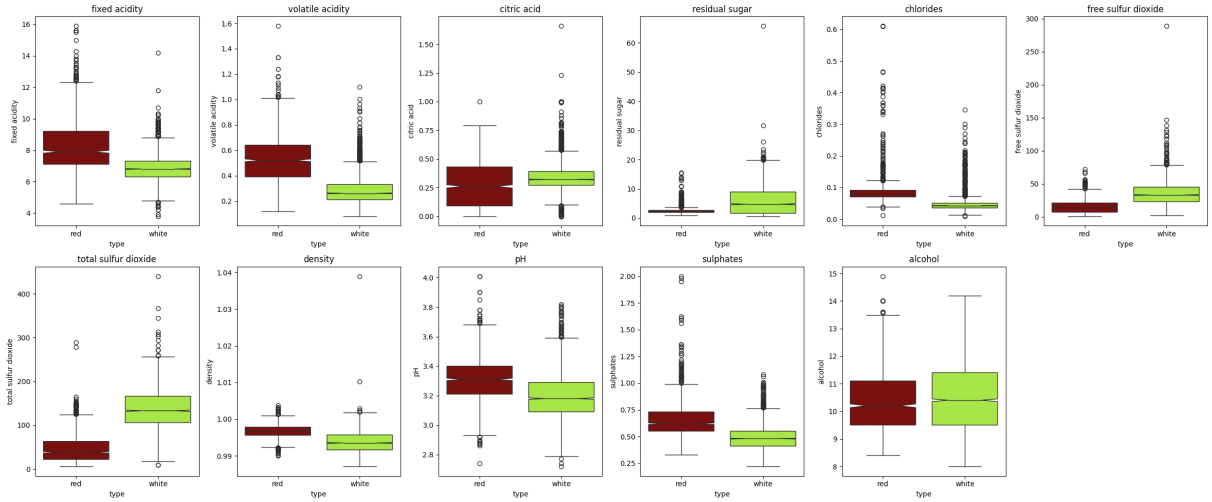

Figure 3: Boxplot delle covariate in base alla classe del vino.

Un altro passo fondamentale dell'EDA è stato calcolare la matrice di correlazione tra le covariate (Figura 4). La correlazione misura quanto due variabili siano linearmente legate: valori vicini a 1 o -1 indicano forte correlazione positiva o negativa, mentre valori vicino a 0 indicano scarsa correlazione. Nel nostro dataset, la maggior parte delle covariate presenta correlazioni medio-basse, il che suggerisce che le informazioni contenute nelle variabili siano in gran parte indipendenti.

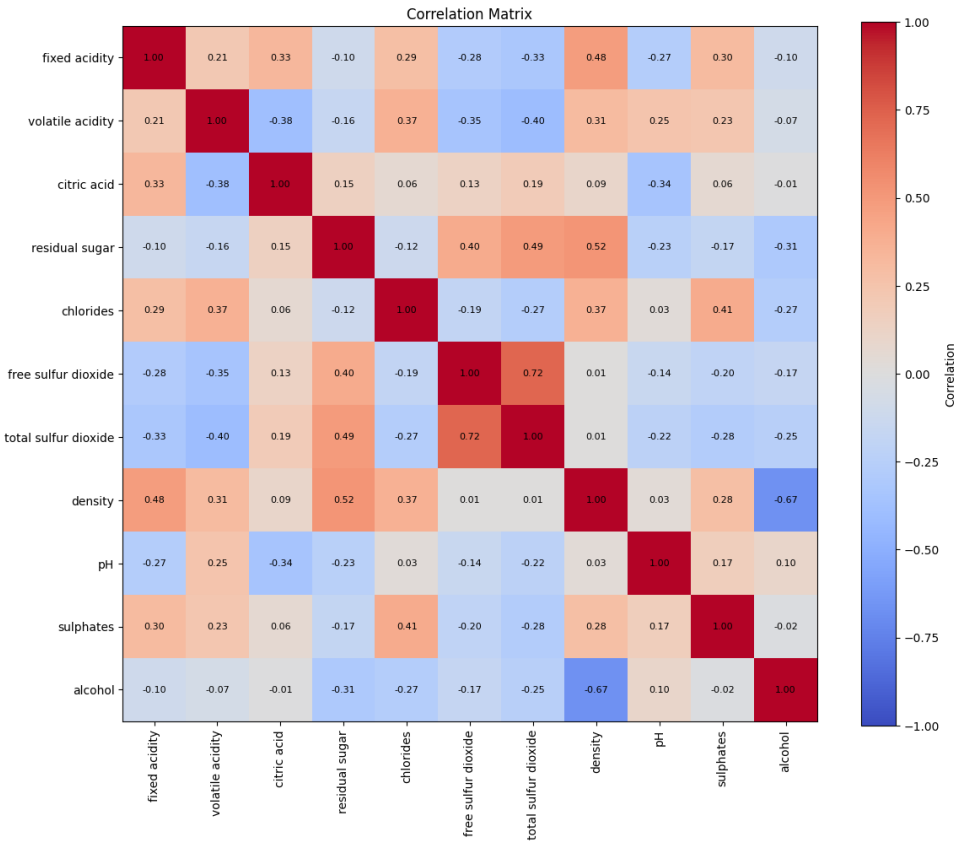

Figure 4: Matrice di correlazione tra le covariate.

Poiché le correlazioni erano generalmente basse, l'uso di tecniche di riduzione dimensionale come la PCA (Principal Component Analysis) non si è rivelato particolarmente utile. La PCA cerca di combinare variabili correlate in componenti principali che sintetizzano la maggior parte della varianza. Tuttavia, quando le covariate non sono fortemente correlate, la PCA non riduce significativamente la dimensionalità né semplifica l'interpretazione dei dati, come mostrato nella Figura 5.

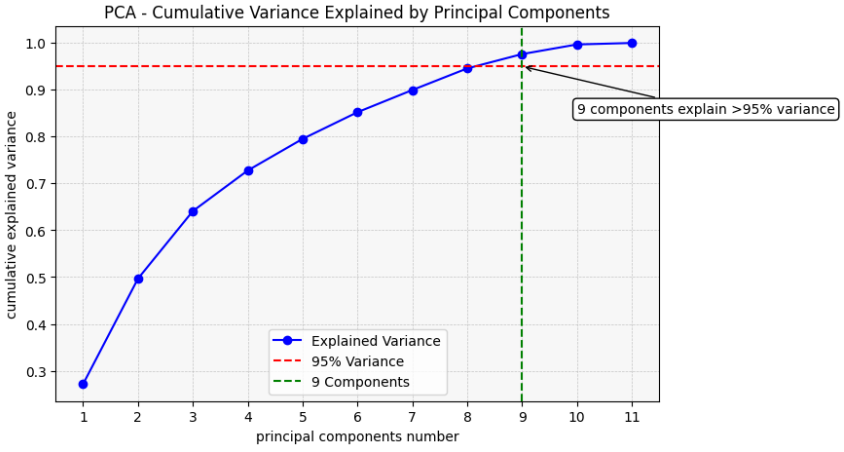

Figure 5: Risultato della PCA sulle covariate.

Analizzando più in dettaglio le coppie di variabili con correlazione relativamente più alta, si osserva che esistono alcune relazioni notevoli. Ad esempio, le variabili `free sulfur dioxide` e `total sulfur dioxide` sono correlate tra loro, così come la coppia `alcohol` e `density`. La prima coppia rappresenta variabili concettualmente simili: il diossido di zolfo libero è una parte del diossido di zolfo totale. Per

ridurre la ridondanza informativa, queste variabili sono state combinate in un unico indicatore, **sulfur dioxide free/tot ratio**, calcolando il rapporto tra le due quantità. La seconda coppia, al contrario, riguarda variabili concettualmente distinte e quindi è stata mantenuta separata, nonostante la correlazione moderata, per non perdere informazioni importanti. La Figura 6 mostra la matrice di correlazione aggiornata dopo la sostituzione delle variabili sul diossido di zolfo.

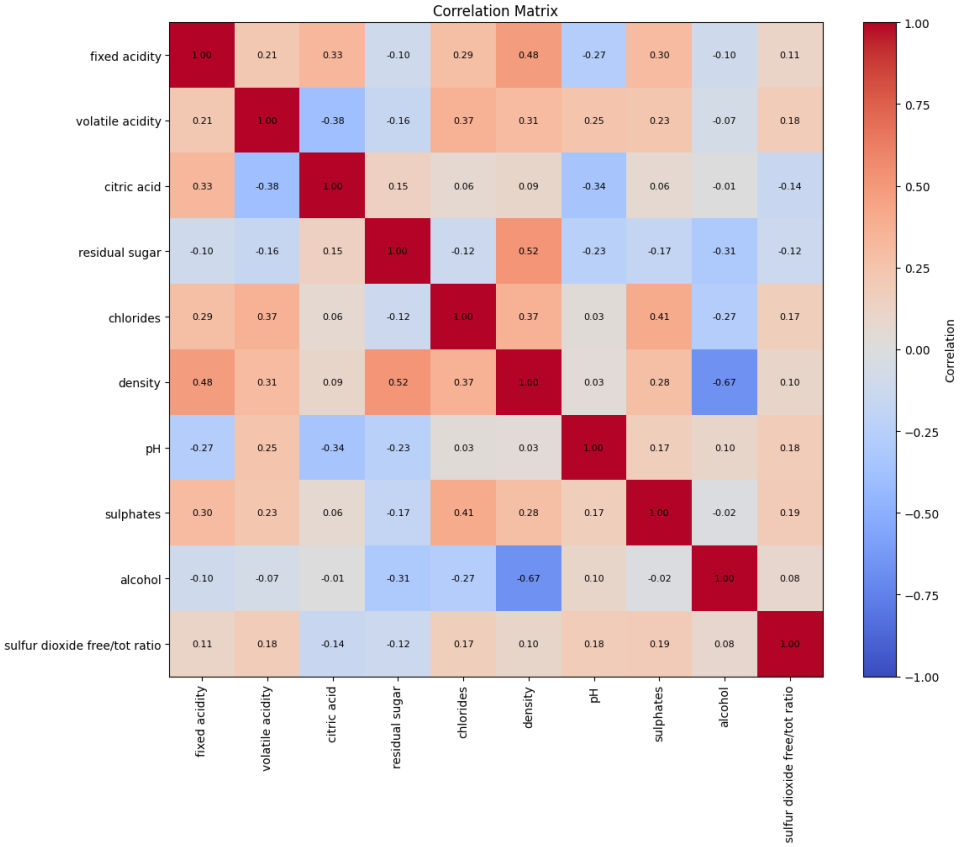

Figure 6: Matrice di correlazione dopo la sostituzione delle variabili relative al diossido di zolfo.

Models and Rationale

Nel progetto vengono utilizzati due modelli di apprendimento automatico supervisionato:

- un *Albero di decisione*;
- una *Multi-layer perceptron*.

L'obiettivo dell'analisi non è limitato al raggiungimento di elevate prestazioni predittive, ma include anche aspetti fondamentali come la stabilità del modello, la capacità di generalizzazione e l'interpretabilità dei risultati.

In particolare, l'albero di decisione è stato scelto per la sua struttura semplice e facilmente interpretabile. Questo modello consente infatti di comprendere in modo diretto il processo decisionale, traducendo i dati in regole chiare e leggibili.

Tuttavia, è noto che gli alberi di decisione possono soffrire di overfitting, soprattutto in presenza di profondità elevate. Per questo motivo è stato adottato un approccio di validazione nella selezione degli iperparametri, con l'obiettivo di controllare la complessità del modello e migliorare la sua capacità di generalizzazione su dati non visti.

Parallelamente, è stata implementata una Multi-Layer Perceptron (MLP) per sfruttare la sua capacità di catturare relazioni spesso invisibili agli alberi di decisione. Anche se è più difficile da interpretare rispetto all'albero, la rete neurale è più potente nel riconoscere legami complessi tra i dati del dataset.

Per prevenire il rischio di overfitting e instabilità del gradiente, l'architettura è stata ottimizzata attraverso l'inserimento di layer di Dropout e una normalizzazione dei dati. Anche in questo caso, è stata utilizzata una validazione sistematica degli iperparametri tramite grid search, come il learning rate e la densità dei neuroni, per garantire un modello robusto e massimizzare la recall.

Experimental Campaign

Il dataset è stato inizialmente suddiviso in tre sottoinsiemi distinti:

- training set (60%);
- validation set (20%);
- test set (20%).

I tre sottoinsiemi sono stati utilizzati ugualmente in entrambi i modelli per ottenere un confronto più accurato tra di essi. In entrambi i modelli, dopo aver trovato i parametri ottimali, è stato adestrato un nuovo modello unendo il training set e validation set per avere una base di dati più ampia per l'apprendimento finale, migliorando così la capacità di generalizzazione del modello prima della valutazione definitiva sul test set. Per garantire la completa riproducibilità degli esperimenti, è stato utilizzato un seed fisso (random_state=1) per la suddivisione dei dati e per l'inizializzazione del Decision Tree. Inoltre, è stato impostato il seed globale di TensorFlow (tf.random.set_seed(1)) per assicurare la costanza dei pesi iniziali e del comportamento dei layer della MLP.

Decision Tree

La suddivisione del dataset in training set e validation set ha permesso di valutare diverse configurazioni di iperparametri e selezionare la soluzione più equilibrata. In particolare, sono state considerate le seguenti combinazioni:

- max_depth: [2, 3, 4, 5, None];
- min_samples_split: [2, 5, 10];
- min_samples_leaf: [1, 3, 5];
- ccp_alpha: [0.0, 0.005, 0.01].

Per ciascuna configurazione sono state calcolate la **training accuracy**, la **validation accuracy** e la loro differenza (**train-val_diff**).

Il modello finale è stato selezionato imponendo i seguenti criteri:

- train-val_diff inferiore a 0.01;
- profondità massima maggiore di 3;
- numero minimo di campioni per split pari a 10.

Una differenza contenuta tra l'accuracy di training e di validation, insieme a un numero minimo di campioni per split, permette di limitare l'overfitting, evitando che l'albero apprenda regole instabili o basate sul rumore dei dati. Imporre inoltre una profondità minima dell'albero garantisce una capacità espressiva adeguata del modello. In questo modo si riduce il rischio di underfitting, evitando la costruzione di un classificatore eccessivamente semplice.

Si è scelto di utilizzare l'indice di Gini perché è più semplice da calcolare e fornisce risultati del tutto comparabili all'entropia nella costruzione degli alberi di decisione. Inoltre, su dataset sbilanciati, Gini tende a essere più stabile e veloce, rendendolo particolarmente adatto a questo contesto.

Questo approccio consente di concentrarsi su configurazioni realistiche e favorire una migliore capacità di generalizzazione. Inoltre, rende più coerente e significativo il confronto tra i modelli analizzati.

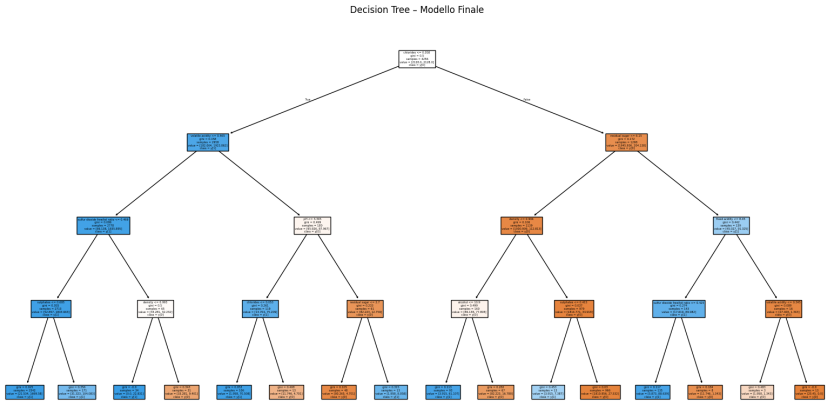

Figure 7: Albero di decisione.

MLP

Come specificato nelle sezioni precedenti, la suddivisione del dataset è stata definita a priori ed è stata utilizzata in modo coerente sia per l'albero di decisione sia per la rete neurale.

Pre-elaborazione e normalizzazione

Prima della fase di addestramento, i dati sono stati sottoposti a un processo di pulizia e trasformazione. Inizialmente, è stata applicata la normalizzazione tramite StandardScaler per uniformare tutte le covariate chimico-fisiche su una scala comune. Questo passaggio è cruciale per il Multi-Layer Perceptron, al fine di evitare che variabili con intervalli numerici più ampi influenzassero eccessivamente il calcolo del gradiente e l'aggiornamento dei pesi. Successivamente, la variabile target categorica "type" è stata codificata in formato binario per consentire l'elaborazione da parte del modello.

Validazione e ottimizzazione degli iperparametri

Analogamente a quanto fatto per l'albero di decisione, l'analisi non si è limitata a una configurazione standard, ma ha previsto una ricerca sistematica della combinazione ottimale di parametri mediante un approccio di grid search. Nella fase di validazione è stata testata una combinazione di diversi elementi, tra cui:

- numero di layers: [1, 2, 3, 4];
- numero di neuroni per leyer: [8, 10, 16, 32];
- learning rate: [0.01, 0.001, 0.0005];
- batch size: [128, 32, 64];
- epoche: [20, 30, 40, 50];
- dropout rate: [0.1, 0.2, 0.3].

I vari modelli sono stati generati seguendo una struttura interna uguale per tutti, che prevede l'utilizzo di Adam come ottimizzatore, scelto per le sue prestazioni migliori rispetto a SGD. Poiché la rete riceve in ingresso dieci feature, il layer di input è composto da altrettanti neuroni. Successivamente, vengono creati un numero variabile di layer nascosti con funzione di attivazione ReLU, scelta dettata dalla natura del dataset che, contenendo esclusivamente dati positivi, consente di evitare la saturazione dei neuroni durante l'apprendimento. In conformità con l'implementazione in Keras, a ogni layer nascosto segue un livello di dropout per prevenire l'overfitting. L'architettura si conclude con un singolo neurone di output dotato di funzione di attivazione sigmoide, scelta coerente con la rappresentazione binaria della classe

target. L'ottimizzazione tramite Grid Search ha richiesto circa 6 ore e mezza; per tale ragione, il relativo codice è stato commentato inserendo staticamente i parametri ottimali nel modello finale. I risultati completi della ricerca restano comunque consultabili nel file `risultati_grid_search_vini.csv` allegato alla cartella di progetto.

Per ogni configurazione è stata adottata la funzione di loss binary cross-entropy e sono state monitorate diverse metriche prestazionali:

- **Accuracy**
- **Precision**
- **Recall**
- **ROC** (Receiver Operating Characteristic)
- **PR** (Precision-Recall-curve)

Questi parametri sono stati fondamentali per la selezione del miglior modello, effettuata sulla base di tre criteri discriminanti, ordinati gerarchicamente: la recall, il numero di layer nascosti e il numero di neuroni. Una volta individuata la combinazione ottimale tramite grid search, è stato addestrato il modello finale procedendo infine alla valutazione delle performance definitive sul test set.

Analisi dei risultati

Valutazione delle prestazioni

La valutazione dei modelli finali è stata effettuata utilizzando diverse metriche, evitando di basarsi esclusivamente sull'accuracy. Questo permette di ottenere una visione più completa ed equilibrata delle prestazioni del classificatori.

In particolare, sono state analizzate:

- precision, recall e F1-score, per valutare le prestazioni su ciascuna classe;
- la matrice di confusione, utile per comprendere la distribuzione degli errori;
- la curva ROC e il valore AUC, per misurare la capacità discriminativa del modello;

inoltre per i due modelli sono state condotte due analisi dedicate in base al modello :

- l'albero di decisione: l'importanza delle feature, per analizzare il contributo delle variabili di input.
- MLP: Analisi dei Campioni Misclassificati.

I risultati mostrano prestazioni simili sui set di training e validation, indicando una buona capacità di generalizzazione in entrambi i modelli. L'analisi delle metriche evidenzia inoltre un comportamento equilibrato dei classificatori.

Albero di decisione

Valutazione delle prestazioni e interpretabilità

Il modello finale di **Decision Tree** presenta le seguenti prestazioni:

- **Train Accuracy**: 0.9624;
- **Test Accuracy**: 0.9455;
- **Tempo di training**: 0.02 s;
- **Tempo di esecuzione della Grid-Search**: 9.00 s;
- **Iperparametri selezionati**:
  - Criterio di slit: Gini
  - max_depth = 4;
  - min_samples_split = 10;
  - min_samples_leaf = 3;
  - ccp_alpha = 0.0;

Metriche di classificazione

Dal **classification report** [2] sul test set emergono risultati complessivamente positivi. La classe *red* presenta un valore di recall elevato, indicando che la maggior parte dei campioni viene correttamente identificata, sebbene con la presenza di alcuni elementi classificati erroneamente. La classe *white*, invece, mostra valori molto alti sia di precision che di recall, confermando una classificazione accurata della classe maggioritaria.

L'accuracy complessiva pari a 0.9455, insieme ai valori di macro e weighted F1-score, indica un buon bilanciamento generale delle prestazioni. Inoltre, la differenza minima tra training e validation accuracy suggerisce che il modello non soffre di overfitting significativo.

Classe	Precision	Recall	F1-score	Support
Red	0.882	0.934	0.907	272
White	0.977	0.957	0.967	792
Accuracy	0.951	0.951	0.951	1064
Macro Avg	0.929	0.945	0.937	1064
Weighted Avg	0.953	0.951	0.952	1064

Table 2: Classification report del modello Decision Tree sul test set

Confusion matrix e AUC

La matrice di confusione [8] conferma le osservazioni precedenti, mostrando un numero contenuto di errori sia per la classe *red* che per la classe *white*.

Il valore di **AUC** pari a 0.974 evidenzia un'ottima capacità del modello di distinguere correttamente tra le due classi, rafforzando l'affidabilità complessiva del classificatore.

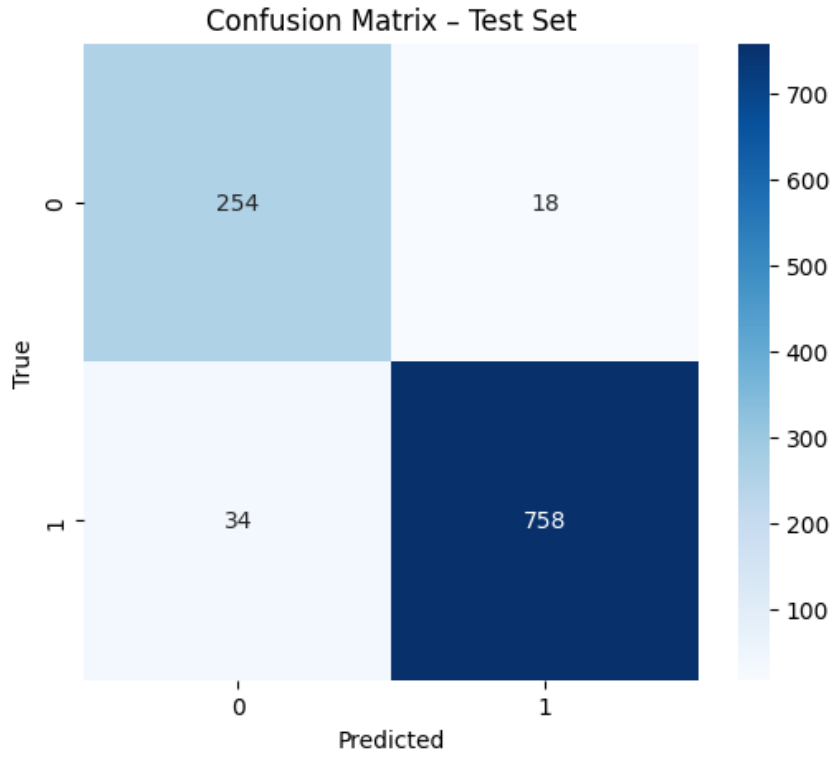

Figure 8: Confusion matrix

Analisi delle feature importance

L'analisi delle **feature importance** [11] mostra che la variabile **Chlorides** domina il processo decisionale, con un'importanza prossima all'80%. Questo indica che il modello si affida principalmente a questa feature per effettuare la classificazione.

Le restanti variabili hanno un contributo decisamente più contenuto, tutte inferiore al 10%, suggerendo che il loro impatto sulle decisioni finali è marginale.

Interpretazione: La forte predominanza di Chlorides suggerisce che alcune feature potrebbero essere ridondanti o poco informative. Questo risultato fornisce spunti utili per future attività di semplificazione del modello o di feature engineering.

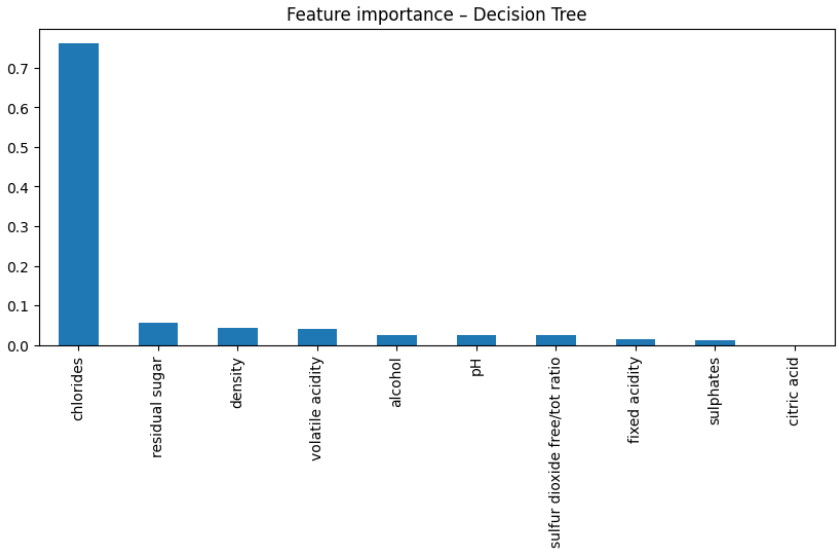

Figure 9: Importanza delle feature

MLP

L'analisi condotta tramite la tecnica di *Grid Search* ha permesso di valutare il comportamento del classificatore MLP al variare degli iperparametri fondamentali. Di seguito vengono analizzate le dinamiche di apprendimento e i criteri che hanno portato alla selezione della configurazione finale.

Criterio di Selezione del Modello

Per la scelta del modello ottimale, i risultati della *Grid Search* sono stati ordinati prioritariamente in base alla **Recall**. Tale scelta è motivata dalla volontà di minimizzare i falsi negativi, garantendo che quasi la totalità dei campioni della classe di interesse venga identificata correttamente.

A parità di Recall, si è scelto di prediligere la semplicità strutturale della rete. Si è scelto quindi il modello con il minor numero di livelli (*layers*) e, successivamente, il minor numero di neuroni per strato, al fine di ridurre la complessità computazionale e il rischio di overfitting, mantenendo comunque prestazioni eccellenti.

Selezione del Modello Ottimale

Sulla base dei criteri, citati precedentemente, è stata selezionata la seguente configurazione:

- layer: 1
- neurons: 8
- learning_rate: 0.01
- batch_size: 128
- epochs: 40
- dropout_rate: 0.3

Nota sulla regolarizzazione: Sebbene la Grid Search suggerisse un valore ottimale di Dropout pari a 0.1, si è scelto di incrementarlo a 0.3. Tale decisione è stata supportata dall'analisi della matrice di correlazione, la quale mostra che il parametro di dropout non influisce in maniera determinante sulle prestazioni complessive, portando a variazioni dell'accuratezza e recall contenute nell'ordine dello 0.001. L'obiettivo di questa scelta è limitare ulteriormente il rischio di *overfitting* e garantire una maggiore robustezza del modello su dati non visti, pur mantenendo un'architettura estremamente snella (soli 8 neuroni).

Performance del Modello Finale

L'analisi della correlazione tra iperparametri e metriche di performance ha evidenziato come il *learning rate* e il numero di neuroni siano i fattori con il maggiore impatto positivo sull'accuratezza. Sulla base di queste evidenze e dei risultati della Grid Search, è stata definita la configurazione ottimale per l'addestramento definitivo.

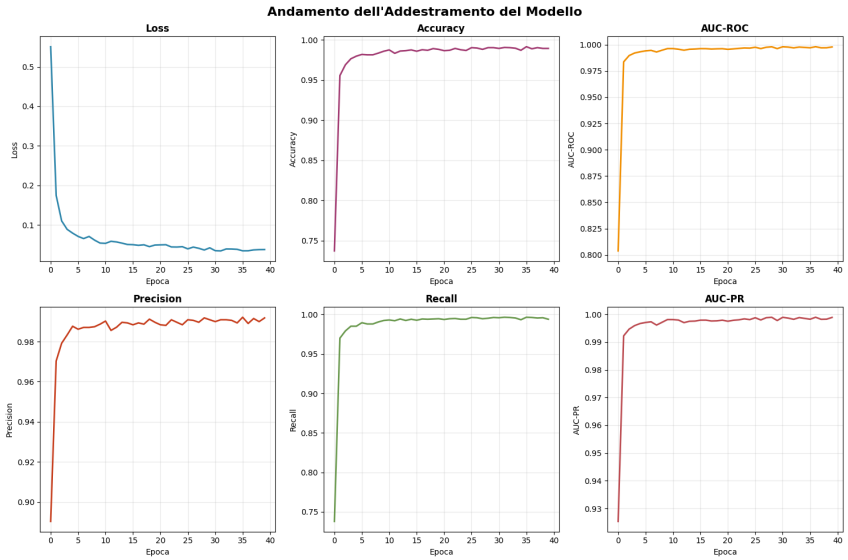

Figure 10: dropout vs accuracy

Una volta riaddestrato il modello con i parametri selezionati, si sono ottenuti risultati d'eccellenza, con un'accuratezza globale del **99%**. Il report di classificazione mostra una precisione e una recall bilanciate per entrambe le tipologie di vino:

Classe	**Precision**	**Recall**	**F1-score**	**Support**
Red	1.00	0.96	0.98	272
White	0.99	1.00	0.99	792
Accuracy			**0.99**	1064

Table 3: Report di classificazione del modello MLP finale.

La **Matrice di Confusione** conferma l'efficacia del classificatore:

- Solo **1 vini bianchi** è stato classificato erroneamente come rosso.
- Solo **11 vini rossi** sono stati classificati erroneamente come bianchi.

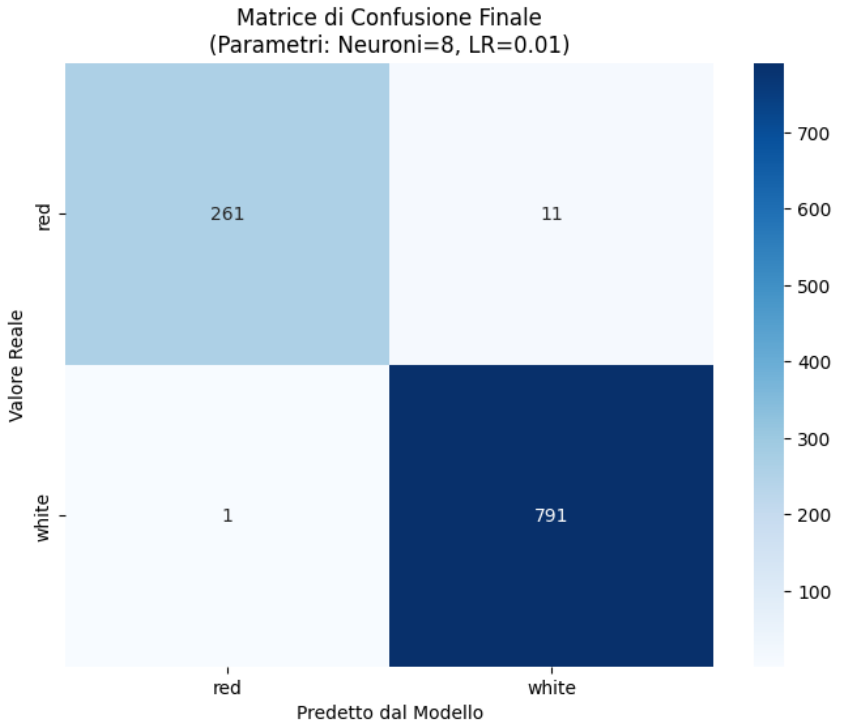

Figure 11: matrice di confusione

L'adozione della recall come metrica primaria, unita a una ricerca sistematica tramite *Grid Search*, ha permesso di gestire efficacemente lo sbilanciamento del dataset (75% vini bianchi, 25% rossi), garantendo un'elevata attenzione alla classe minoritaria. Nonostante l'assenza di tecniche specifiche per gestire dataset sbilanciati o della Cross-validation (generalmente indicata per dataset più limitati rispetto alle 5320 osservazioni correnti), l'architettura MLP ha dimostrato una capacità di generalizzazione ottimale. Il raggiungimento di un'accuratezza globale del 99% e di una precisione superiore al 98.8% sul test set conferma che, per questo specifico task, ulteriori interventi di bilanciamento non erano determinanti per il successo del modello.

Analisi Approfondita dei Campioni Misclassificati

Per completare l'analisi delle performance, è stata effettuata un'indagine puntuale sui **12 errori** totali commessi dal modello. L'obiettivo è verificare se tali errori siano dovuti a limiti strutturali della rete o alla presenza di campioni con caratteristiche chimiche atipiche (outliers).

Index	Reale	Predetto	F.Acid	V.Acid	C.Acid	R.Sug	Chlor	Dens	pH	Sulph	Alc	S02 ratio
6230	red	white	-0.98	-1.75	0.60	-0.17	-0.95	-2.41	-0.58	-0.46	0.43	-0.50
4376	red	white	-0.41	-1.80	0.45	-0.46	-0.91	-2.32	-0.13	-0.70	0.80	-1.07
2415	red	white	0.05	-1.91	0.76	1.91	-1.06	-1.50	-0.97	-0.81	1.17	-1.88
1938	red	white	-0.98	-1.91	0.40	-0.76	-0.79	-1.77	0.78	1.12	-0.21	-0.22
5691	red	white	-1.10	-0.16	-0.37	-0.09	-0.59	-1.91	-0.84	-1.52	-1.14	-1.37
4456	red	white	-0.98	-1.31	0.19	1.02	-1.24	-3.40	-0.45	-0.34	2.00	0.16
4814	red	white	-0.70	-2.13	0.29	-1.13	0.32	-3.28	-0.77	-0.11	0.99	-0.07
4083	white	red	-0.51	0.19	-1.42	0.33	0.18	0.73	-0.17	-0.00	-1.31	0.35
5452	red	white	-0.98	-1.14	-0.06	1.54	-1.16	-2.53	0.58	0.42	1.91	-1.07
2146	red	white	-0.35	-1.69	0.76	-0.46	-0.73	-2.41	-0.06	-0.99	1.26	-1.72
1814	red	white	-1.45	-1.47	-0.17	4.94	-0.89	-0.33	0.00	-1.63	-0.95	-0.13
824	red	white	-0.70	-0.27	0.04	0.20	-0.41	0.06	-0.45	-0.75	-0.12	-0.04

Table 4: Analisi qualitativa dei campioni misclassificati sul test set.

Considerazioni sull'analisi dei dati L'esame puntuale dei campioni misclassificati permette di comprendere come gli errori commessi dalla rete non siano riconducibili a un limite strutturale del modello, quanto piuttosto alla natura intrinsecamente ambigua di alcuni dati.

Un caso emblematico è rappresentato dal campione 4083, l'unico vino bianco identificato come rosso. Analizzando i suoi parametri, si nota un valore di *density* (0.73) e di *chlorides* (0.18) positivo nella scala standardizzata, accompagnato da un basso contenuto alcolico (-1.31). Trattandosi di caratteristiche fisiche che solitamente contraddistinguono i vini rossi nel dataset, la rete ha correttamente interpretato lo stimolo secondo le correlazioni apprese, nonostante l'etichetta reale fosse differente.

In modo speculare, si osserva la presenza di "profili chimici invertiti" nella quasi totalità dei restanti errori (vini rossi classificati come bianchi). Casi come gli indici 4456, 4814 e 6230 presentano valori di densità estremamente bassi (rispettivamente -3.40, -3.28 e -2.41) e cloruri significativamente negativi, ricalcando quasi perfettamente la "firma" tipica dei vini bianchi.

In definitiva, i 12 errori rilevati rappresentano una quota marginale pari all'1.13% del test set totale (composto da 1064 campioni). L'analisi qualitativa conferma che ci troviamo di fronte a veri e propri outliers o campioni con caratteristiche contraddittorie. L'elevata capacità di generalizzazione della rete neurale è dunque confermata dalla sua capacità di ignorare il rumore di fondo, mantenendo una precisione superiore al 98.8% sulla stragrande maggioranza della popolazione analizzata.

Conclusioni

L'obiettivo di questo progetto era valutare l'efficacia di due modelli di classificazione supervisionata nel distinguere vini rossi e bianchi a partire da caratteristiche chimico-fisiche.

Prestazioni Predittive

Dal punto di vista delle prestazioni, la rete neurale MLP si è dimostrata superiore. Il modello finale ha raggiunto un'accuratezza pari al 99%, con valori di precision e recall elevati e ben bilanciati per entrambe le classi. Il numero di errori sul test set risulta estremamente ridotto, indicando un'eccellente capacità di generalizzazione.

L'albero di decisione, pur ottenendo risultati leggermente inferiori, ha comunque mostrato prestazioni solide, con un'accuratezza sul test set pari a 0.9455 e un valore di AUC pari a 0.97. La ridotta differenza tra training e test accuracy suggerisce l'assenza di overfitting significativo, confermando la bontà del processo di selezione degli iperparametri.

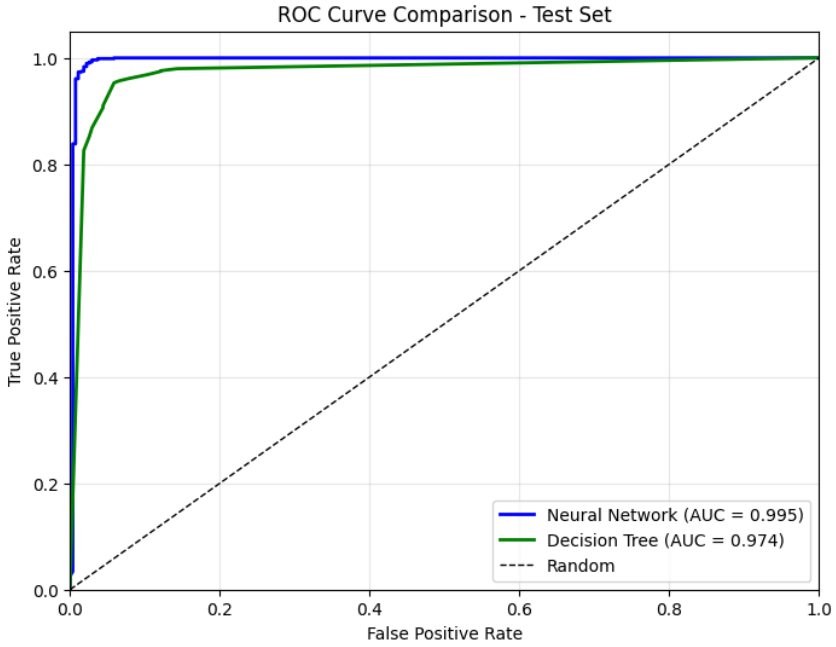

Figure 12: Comparazione delle due curve ROC

Gestione dello Sbilanciamento delle Classi

Il dataset presenta uno sbilanciamento tra le due classi, con una prevalenza di vini bianchi. In questo contesto, la scelta di utilizzare la recall come criterio principale per la selezione del modello MLP si è rivelata efficace, permettendo di garantire un'adeguata attenzione alla classe minoritaria.

L'albero di decisione mostra comunque un comportamento equilibrato, ma risulta leggermente più influenzato dalla classe dominante rispetto alla rete neurale.

Interpretabilità e Complessità del Modello

Uno dei principali vantaggi dell'albero di decisione è la sua elevata interpretabilità. Il modello fornisce regole decisionali semplici e comprensibili, rendendo trasparente il processo di classificazione. L'analisi delle feature importance evidenzia inoltre come poche variabili, in particolare *Chlorides*, guidino la maggior parte delle decisioni, fornendo spunti utili per future attività di feature selection.

La rete neurale MLP, al contrario, presenta una struttura meno interpretabile, poiché le decisioni emergono dalla combinazione non lineare dei pesi appresi nei layer interni. Tuttavia, questa maggiore complessità consente al modello di catturare relazioni più sofisticate tra le feature, traducendosi in prestazioni superiori.

Robustezza e Analisi degli Errori

L'analisi dei campioni misclassificati dalla rete neurale mostra che gli errori residui sono riconducibili principalmente a campioni chimicamente ambigui o outliers, caratterizzati da profili che si collocano in una zona di confine tra le due classi. Tali errori non sembrano derivare da limiti strutturali del modello, ma piuttosto dalla natura intrinsecamente complessa di alcuni dati.

Questo risultato suggerisce che la MLP abbia raggiunto un livello di performance prossimo al limite informativo del dataset.

Costi Computazionali

Dal punto di vista computazionale, l'albero di decisione risulta estremamente efficiente, con tempi di addestramento molto ridotti e una semplicità di implementazione che lo rende adatto a contesti con risorse limitate o in cui la rapidità di esecuzione è un requisito fondamentale.

La rete neurale, pur adottando un'architettura volutamente semplice, richiede una fase di addestramento più onerosa e una maggiore attenzione nella fase di pre-elaborazione e tuning degli iperparametri.

Considerazioni Finali

In conclusione, il confronto tra i due modelli evidenzia un compromesso tra semplicità e prestazioni. L'albero di decisione è facile da interpretare, veloce e trasparente, ed è quindi adatto quando è importante comprendere il processo decisionale. La rete neurale MLP offre prestazioni migliori e una maggiore capacità di gestire casi complessi, risultando preferibile quando l'obiettivo principale è massimizzare l'accuratezza.

Tuttavia, data la relativa semplicità del dataset, l'utilizzo di una rete neurale può risultare eccessivo, pur confermando l'elevata qualità dei dati attraverso le ottime prestazioni ottenute. Entrambi i modelli si dimostrano quindi adeguati al problema, con la MLP più performante e l'albero di decisione più interpretabile.