



DIPARTIMENTO DI INFORMATICA
Corso di Laurea Magistrale in Informatica
Esame di Machine Learning

Breast Cancer Prediction

Pallini Vincenzo - 907303
Nicolò Nicholas Zagami - 829888

Anno Accademico 2023 - 2024

Indice

1	Introduzione	2
2	Preparazione del dataset	3
2.1	Operazioni di null check e refactoring	4
3	Analisi esplorativa del dataset	5
3.1	Analisi variabili	5
3.2	Analisi correlazioni	9
3.3	Correlazione tra variabili dello stesso tipo	12
3.4	Principal Component Analysis (PCA)	15
3.5	Applicazione della PCA e i suoi risultati	15
3.6	Gli obiettivi dello studio	17
4	Modelli	18
4.1	Decision tree	18
4.2	Reti neurali	24
5	Esperimenti eseguiti	27
5.1	Matrici di confusione	27
5.2	Accuratezza	28
5.3	Precision, Recall e F1-Measure	28
5.4	Curve ROC e AUC	29
5.5	10-fold cross-validation	30
5.6	Tempi di computazione	32
6	Conclusioni	33

1 Introduzione

Il cancro al seno è uno dei tumori più gravi. Ogni anno miete centinaia di migliaia di vite. La diagnosi precoce del cancro al seno svolge un ruolo importante nel successo del trattamento e nel salvare la vita di migliaia di pazienti ogni anno. Tuttavia, gli approcci convenzionali sono limitati nel fornire tale capacità.

I metodi di apprendimento automatico per la diagnosi possono aumentare significativamente la velocità di elaborazione e su larga scala possono rendere la diagnosi significativamente più economica. Queste tecniche utilizzano algoritmi per “apprendere” dai dati e fare previsioni o decisioni senza essere esplicitamente programmati per farlo. Nel contesto del cancro al seno, l’apprendimento automatico può essere utilizzato per analizzare un gran numero di caratteristiche dei tumori, come dimensioni, forma, texture, ecc., e prevedere se un tumore è benigno o maligno.

Questo non solo può aumentare la velocità di elaborazione, permettendo ai medici di diagnosticare più pazienti in meno tempo, ma può anche rendere la diagnosi significativamente più economica su larga scala. Inoltre, poiché gli algoritmi di apprendimento automatico migliorano con l’esperienza, la loro precisione può aumentare nel tempo, potenzialmente portando a meno falsi positivi e migliorando ulteriormente l’efficacia della diagnosi precoce.

la relazione seguirà la seguente struttura:

- **caricamento del dataset**, in particolare vengono effettuate delle analisi preliminari sul dataset preso in esame.
- **analisi esplorativa**, dove ci siamo soffermati sull’analisi multivariata delle feature all’interno del dataset; facendo delle riflessioni rispetto alla diagnosi, ovvero il nostro target.
- **principal component analysis**, un importante strumento di dimensionality reduction.
- **modelli**, in questa sezione faremo riferimento alle scelte che ci hanno portato a scegliere i modelli utilizzati per questo studio.
- **esperimenti eseguiti**, dove vengono illustrati i risultati degli esperimenti condotti con i modelli.
- **conclusioni**

2 Preparazione del dataset

Il dataset analizzato in questo elaborato, Breast Cancer Wisconsin Diagnostic (WDBC), è tratto dai dati raccolti dal Wisconsin Breast Cancer Diagnostic Database. In particolare questi dati sono stati raccolti al fine di eseguire una indagine sulla diagnosi del cancro al seno. I campioni sono donne con un tumore al seno, al momento dell'intervista. Il dataset di partenza è disponibile presso il sito UCI. Il problema che cerchiamo di risolvere applicando due diversi modelli predittivi è quello di prevedere se il tumore è maligno o benigno, in base alle caratteristiche delle cellule tumorali.

Il dataset del cancro al seno del Wisconsin (WDBC) è un dataset statistico di 569 campioni di cancro al seno. Ogni istanza contiene 32 variabili, che corrispondono a :

- *id*: Variabile ID univoca del paziente
- *diagnosis*: Variabile target M - Malignant (Cancerous), B - Benign (Non-cancerous)

Poi seguono tre misure (media, deviazione standard, caso peggiore o valore più alto) per le dieci caratteristiche del nucleo cellulare:

- *radius*: media delle distanze dal centro ai punti sul perimetro
- *texture*: Deviazione standard dei valori in scala di grigi
- *perimeter*: Perimetro
- *area*: Area
- *smoothness*: Variazione locale delle lunghezze dei raggi
- *compactness*: Data dalla seguente formula $perimeter^2/area - 1$
- *concavity*: Gravità delle porzioni concave del contorno
- *concave points*: Numero di porzioni concave del contorno
- *symmetry*: Simmetria
- *fractal dimension*: "coastline approximation" - 1

Come prima detto quindi le 10 caratteristiche sono presenti nel nostro dataset nei loro valori di media, standard error e caso peggiore o valore più alto. Con una prima vista della tipologia delle variabili si evidenzia come tutte le nostre variabili tranne il target siano di tipologia *float*.

<i>id</i>	<i>int64</i>	
<i>diagnosis</i>	<i>object</i>	
<i>radius_mean</i>	<i>float64</i>	<i>concavity_se</i> <i>float64</i>
<i>texture_mean</i>	<i>float64</i>	<i>concave points_se</i> <i>float64</i>
<i>perimeter_mean</i>	<i>float64</i>	<i>symmetry_se</i> <i>float64</i>
<i>area_mean</i>	<i>float64</i>	<i>fractal_dimension_se</i> <i>float64</i>
<i>smoothness_mean</i>	<i>float64</i>	<i>radius_worst</i> <i>float64</i>
<i>compactness_mean</i>	<i>float64</i>	<i>texture_worst</i> <i>float64</i>
<i>concavity_mean</i>	<i>float64</i>	<i>perimeter_worst</i> <i>float64</i>
<i>concave points_mean</i>	<i>float64</i>	<i>area_worst</i> <i>float64</i>
<i>symmetry_mean</i>	<i>float64</i>	<i>smoothness_worst</i> <i>float64</i>
<i>fractal_dimension_mean</i>	<i>float64</i>	<i>compactness_worst</i> <i>float64</i>
<i>radius_se</i>	<i>float64</i>	<i>concavity_worst</i> <i>float64</i>
<i>texture_se</i>	<i>float64</i>	<i>concave points_worst</i> <i>float64</i>
<i>perimeter_se</i>	<i>float64</i>	<i>symmetry_worst</i> <i>float64</i>
<i>area_se</i>	<i>float64</i>	<i>fractal_dimension_worst</i> <i>float64</i>
<i>smoothness_se</i>	<i>float64</i>	Unnamed: 32 <i>float64</i>
<i>compactness_se</i>	<i>float64</i>	<i>dtype: object</i>

Fig. 1: Tipologia delle variabili

2.1 Operazioni di null check e refactoring

Prima di iniziare l'analisi esplorativa è necessario verificare la presenza di variabili **null** e fare del refactoring se necessario, andando a convertire il tipo di alcune variabili piuttosto che eliminarle definitivamente poiché non utili in fase di analisi.

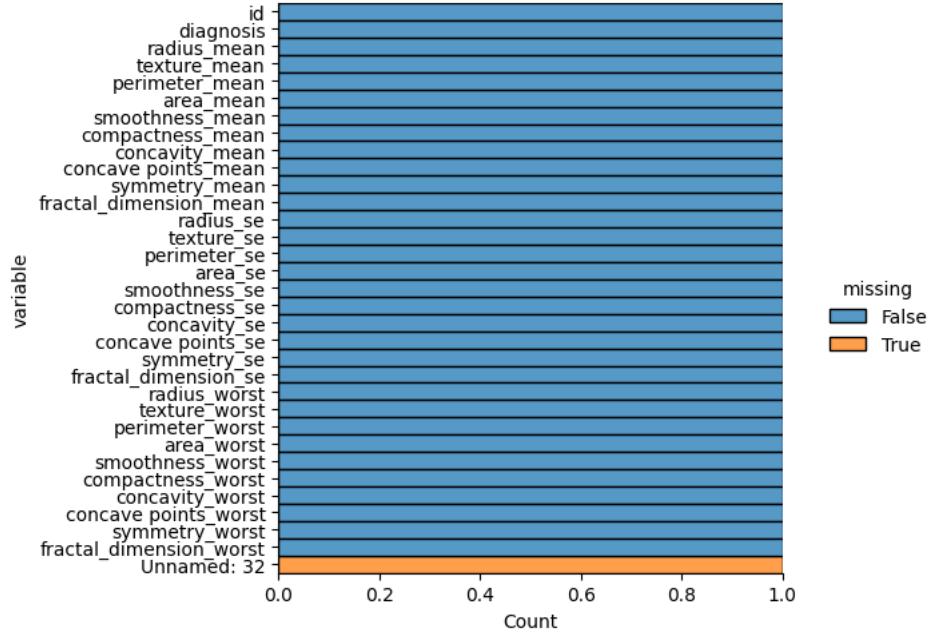


Fig. 2: Null check delle variabili presenti all'interno del dataset

Notiamo che l'ultima variabile oltre a non avere una label ben definita è completamente nulla, motivo per cui decidiamo di rimuoverla. Assieme all'ultima variabile verrà rimossa anche la variabile *id* poiché non utile in fase di analisi.

Dopo il refactoring il dataset risulta essere il seguente:

diagnosis	object		
radius_mean	float64	concavity_se	float64
texture_mean	float64	concave points_se	float64
perimeter_mean	float64	symmetry_se	float64
area_mean	float64	fractal_dimension_se	float64
smoothness_mean	float64	radius_worst	float64
compactness_mean	float64	texture_worst	float64
concavity_mean	float64	perimeter_worst	float64
concave points_mean	float64	area_worst	float64
symmetry_mean	float64	smoothness_worst	float64
fractal_dimension_mean	float64	compactness_worst	float64
radius_se	float64	concavity_worst	float64
texture_se	float64	concave points_worst	float64
perimeter_se	float64	symmetry_worst	float64
area_se	float64	fractal_dimension_worst	float64
smoothness_se	float64	Unnamed: 32	float64
compactness_se	float64	dtype: object	

Fig. 3: Tipologia delle variabili dopo il refactoring

3 Analisi esplorativa del dataset

Per iniziare la nostra analisi, volevamo prima avere una migliore comprensione dei dati in modo da poterli preparare adeguatamente per i nostri modelli. Come si può vedere nella Figura 5 sotto, abbiamo più informazioni sulle cellule benigne che sulle cellule maligne. Per essere più specifici, abbiamo 357 cellule benigne e 212 cellule maligne. Poiché siamo preoccupati per il tasso di errore di tipo II, in un mondo ideale vorremmo avere più informazioni sulle cellule maligne. Tuttavia, non essendo questo il caso, è importante tenerne conto perché potrebbe portare a un bias nel nostro modello se non lo consideriamo.

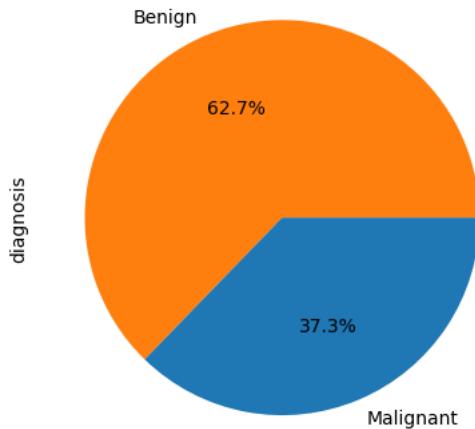


Fig. 4: Bilanciamento generale del target Diagnosi

3.1 Analisi variabili

Come visto in precedenza, all'interno del dataset sono presenti 3 caratteristiche diverse, per ognuna di queste è stato creato un boxplot coinvolgendo le relative variabili e mettendole in relazione rispetto alla diagnosi.

L'utilizzo dei boxplot non è casuale, infatti:

- **Visualizzare le distribuzioni:** Avere una visualizzazione chiara delle distribuzioni delle variabili è fondamentale per studiare al meglio le due diagnosi.
- **Confronto:** rapidità nel confronto tra le due diagnosi rispetto alle variabili divise per caratteristiche.
- **Outliers:** all'interno di un dataset possono essere presenti dei valori anomali anche detti outliers, i boxplot mi permettono di visualizzarli in modo chiaro e facilmente interpretabile.

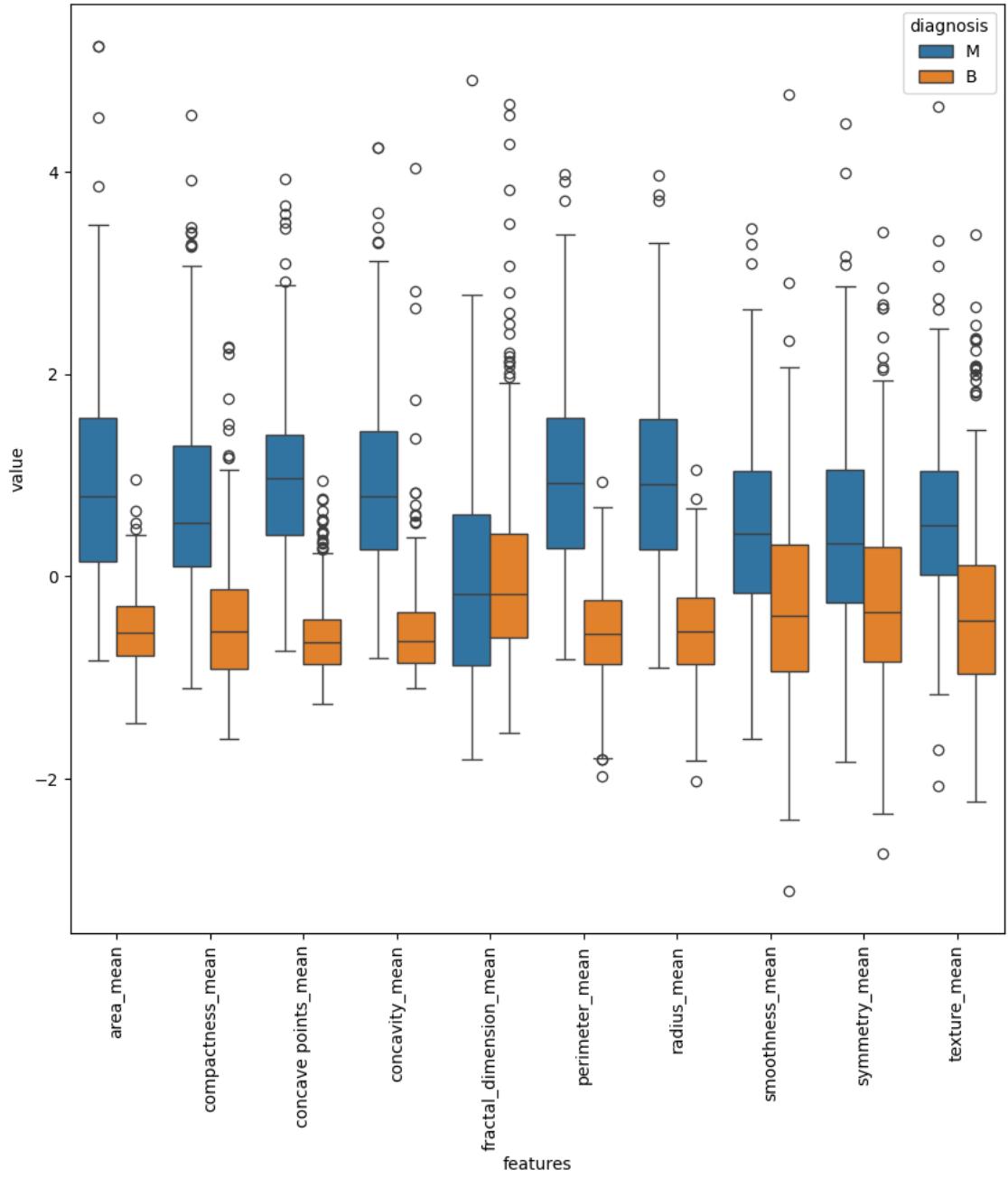


Fig. 5: Variabili mean rispetto al target

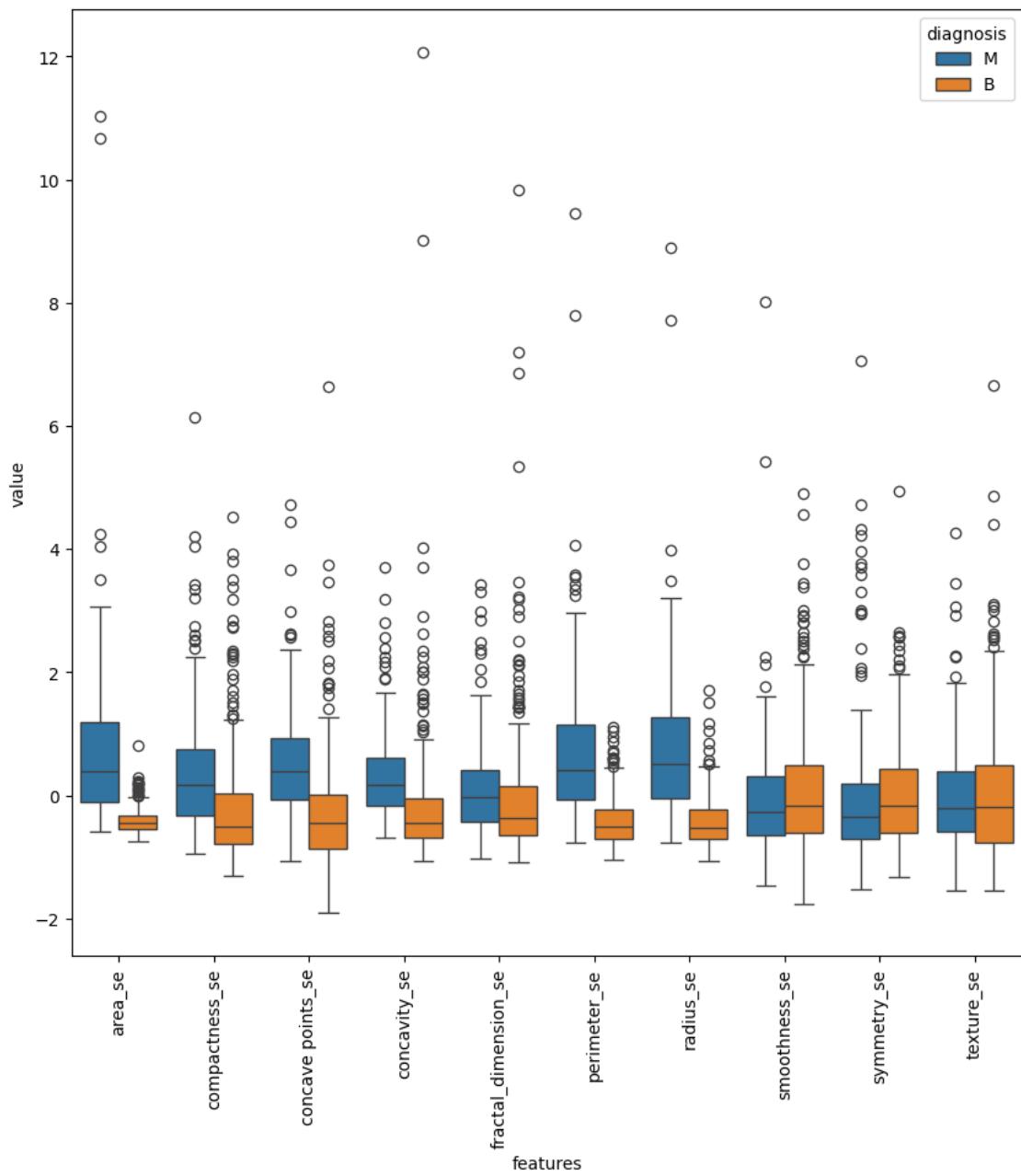


Fig. 6: Variabili SE rispetto al target

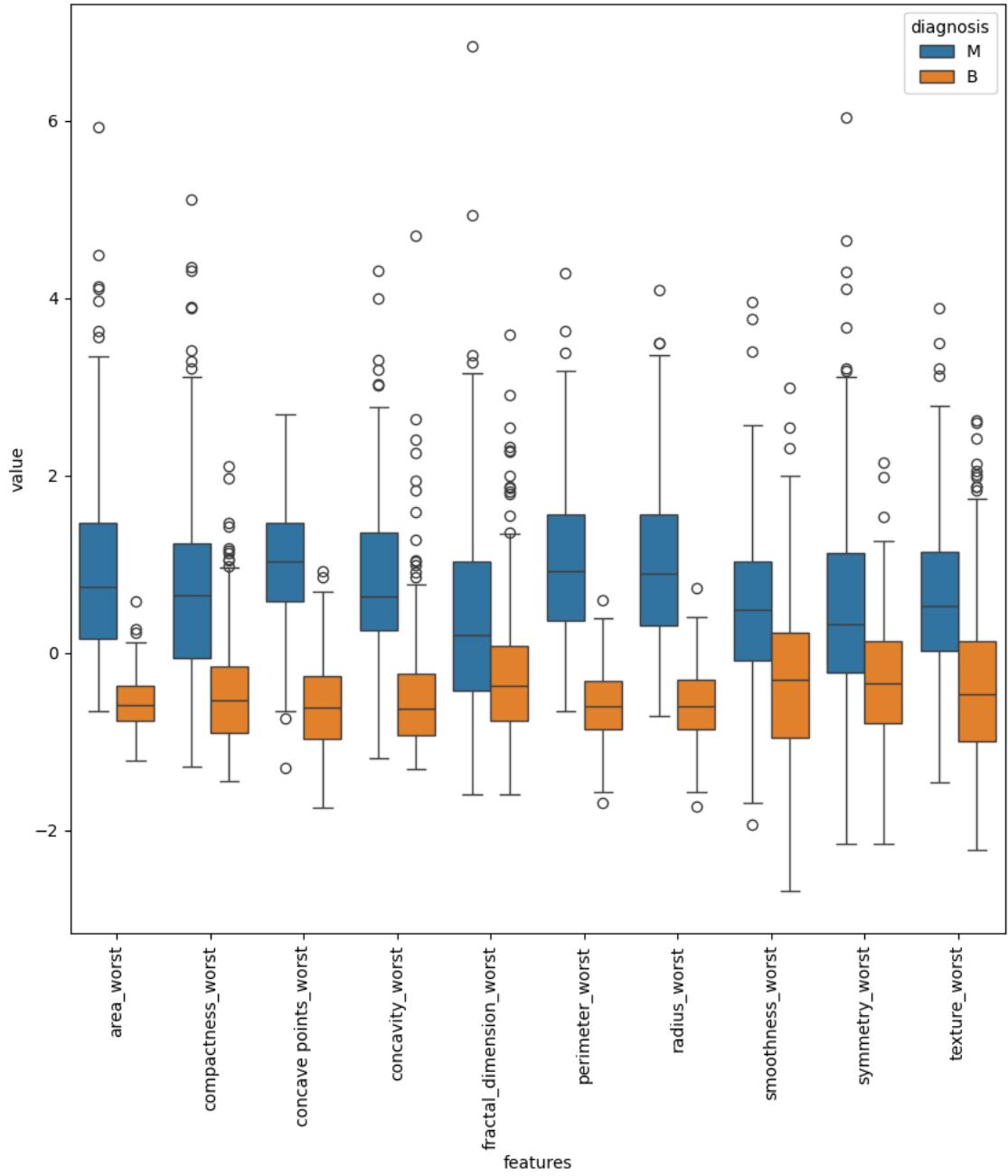


Fig. 7: Variabili Worst rispetto al target

La seguente analisi del dataset mostra che, in generale, i tumori maligni hanno valori più alti in tutti gli attributi rispetto ai tumori benigni. Facciamo alcune riflessioni:

- **I tumori maligni tendono ad essere più invasivi e/o grandi**, come ci viene dimostrato dai valori delle variabili che rappresentano l'*area*, il *perimetro* e il *raggio*. Infatti i tumori maligni sono in grado di distruggere la capsula in cui sono contenuti e ciò gli permette di andare in circolo nell'organismo.
- **Aggressività e anomalia**: Prendendo come riferimento le stesse variabili citate precedentemente si può notare che nel boxplot della caratteristica "worst" i tumori maligni tendono ad

avere valori molto più anomali, questo ci suggerisce che tendono ad essere molto più aggressivi verso l'organismo in cui sono contenuti.

- **Irregolarità:** Oltre ad essere anomali i tumori maligni sono anche molto più irregolari rispetto ai tumori benigni.
- **Compattezza:** i tumori con una densità maggiore di cellule sono più difficili da curare, con riferimento al nostro dataset abbiamo potuto notare che i tumori maligni tendono ad essere anche più compatti (o densi) rispetto a quelli benigni.

Ricapitolando, i tumori maligni nel dataset mostrano valori più elevati in quasi tutte le variabili analizzate rispetto ai tumori benigni, indicando una maggiore **invasività**, **dimensione** e **aggressività**.

3.2 Analisi correlazioni

Per analizzare la correlazione che esiste tra le coppie di variabili del dataset creiamo un *heatmap* il cui input è la *matrice delle correlazioni*. La relazione che intercorre tra coppie di variabili viene rappresentata attraverso un coefficiente, il quale può essere:

- **Positivo**, al crescere di una variabile crescerà anche l'altra
- **Negativo**, al crescere di una variabile l'altra diminuirà
- **Vicino allo zero**, in questo caso l'associazione tra le variabili risulta essere debole se non assente

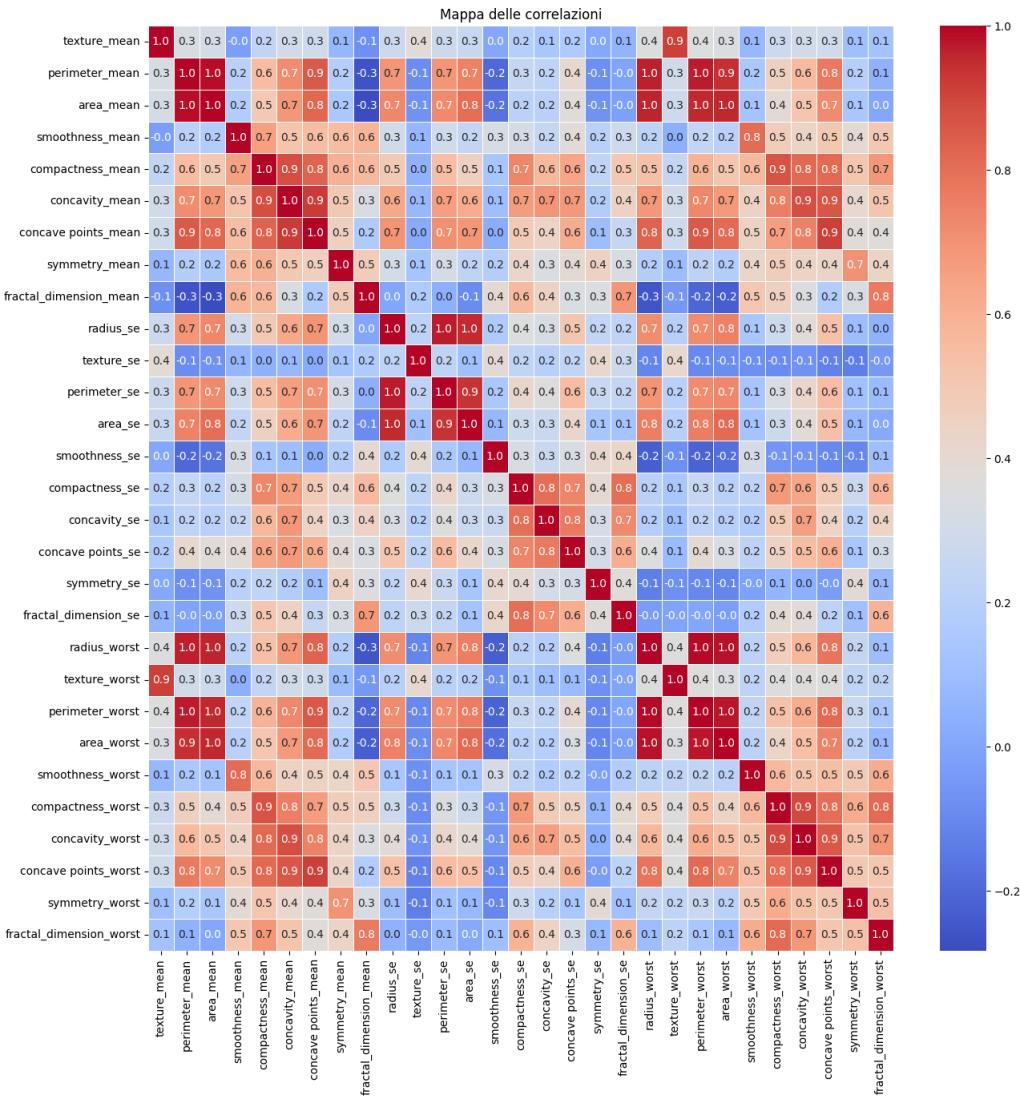


Fig. 8: Mappa delle correlazioni Mean, SE, Worst

- All'interno del nostro dataset troviamo un elevato numero di variabili altamente correlate, questo inevitabilmente introduce **multicollinearità**; rendendo difficile distinguere l'effetto individuale di ciascuna variabile sulla variabile dipendente.
- Notiamo inoltre che c'è una forte correlazione tra le variabili della caratteristica "mean" e "worst".

Quanto detto poco fa è verificabile nella mappa delle variabili altamente correlate:

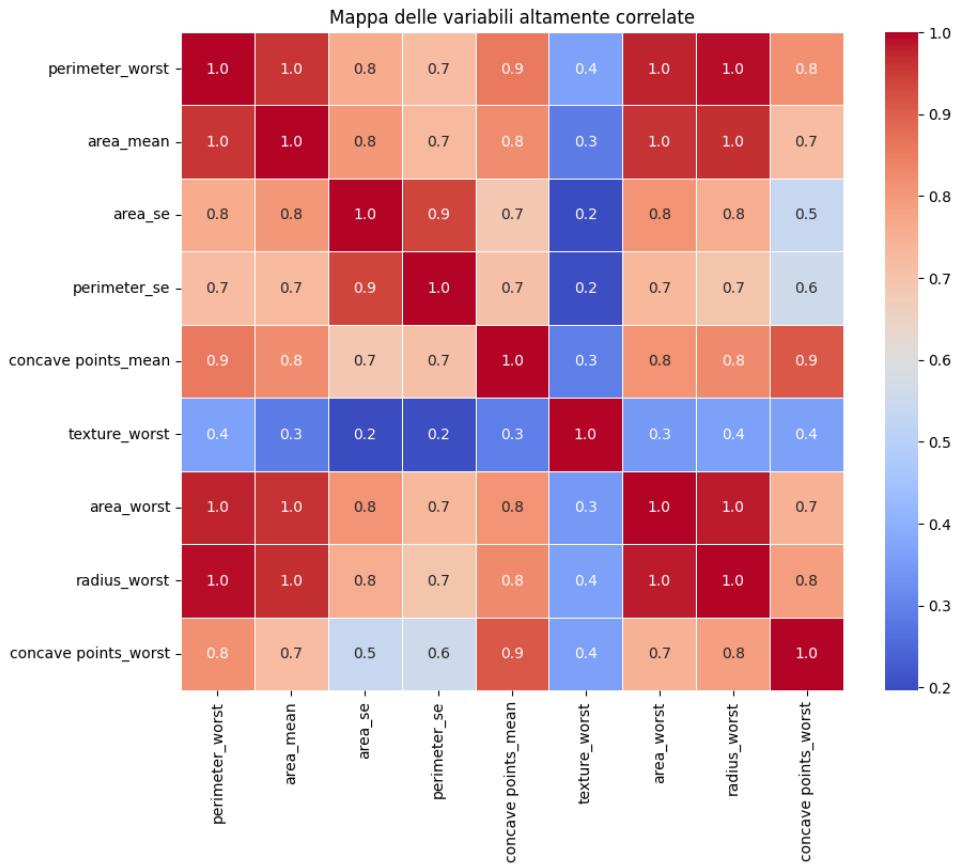


Fig. 9: Correlazioni variabili SE

E' importante notare che le correlazioni non sono causali. Ciò significa che il fatto che due variabili siano correlate non significa che una causi l'altra. Tuttavia, le correlazioni possono fornire indizi utili per comprendere le relazioni tra le variabili.

3.3 Correlazione tra variabili dello stesso tipo

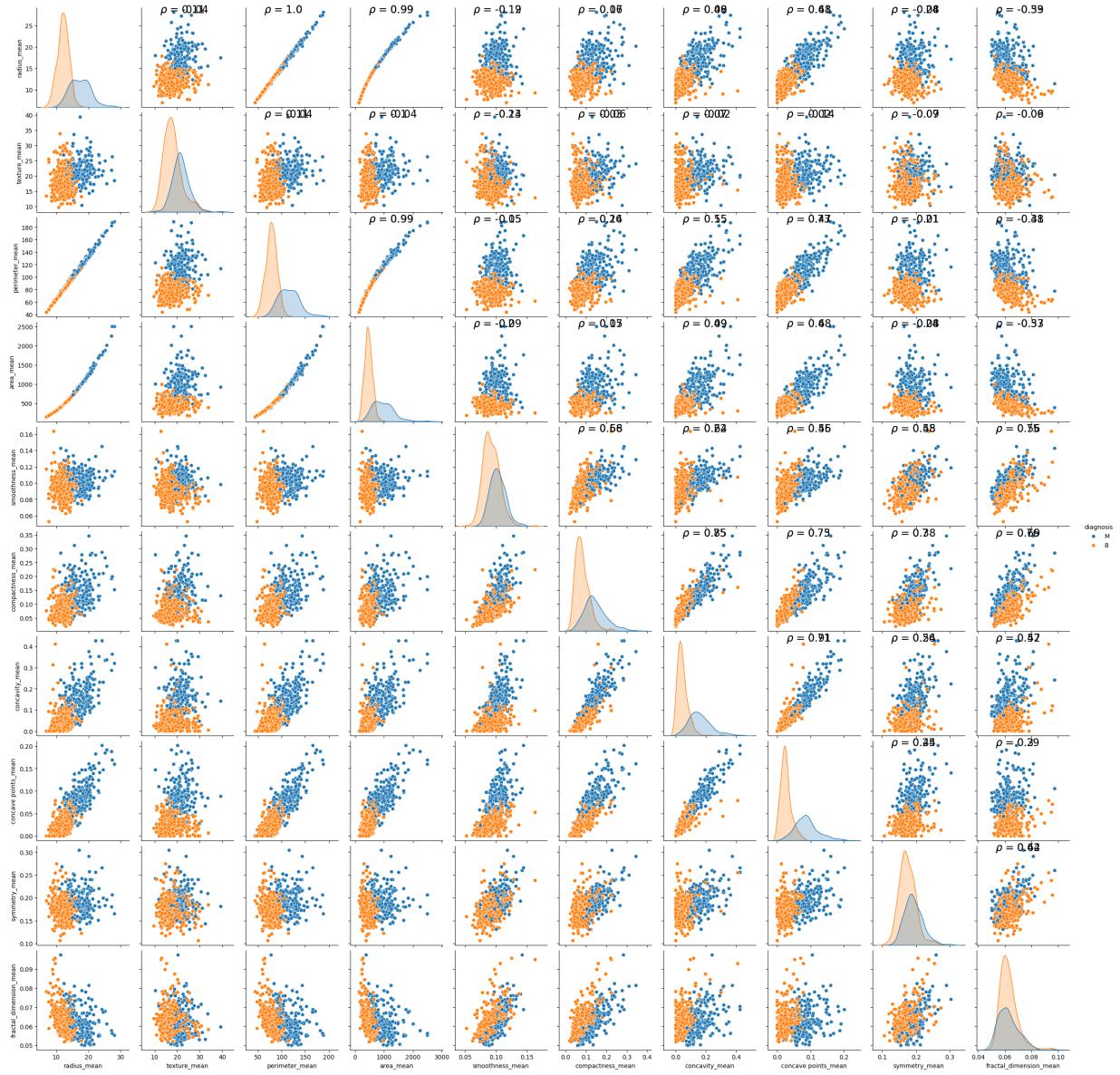


Fig. 10: Correlazioni variabili Mean



Fig. 11: Correlazioni variabili SE

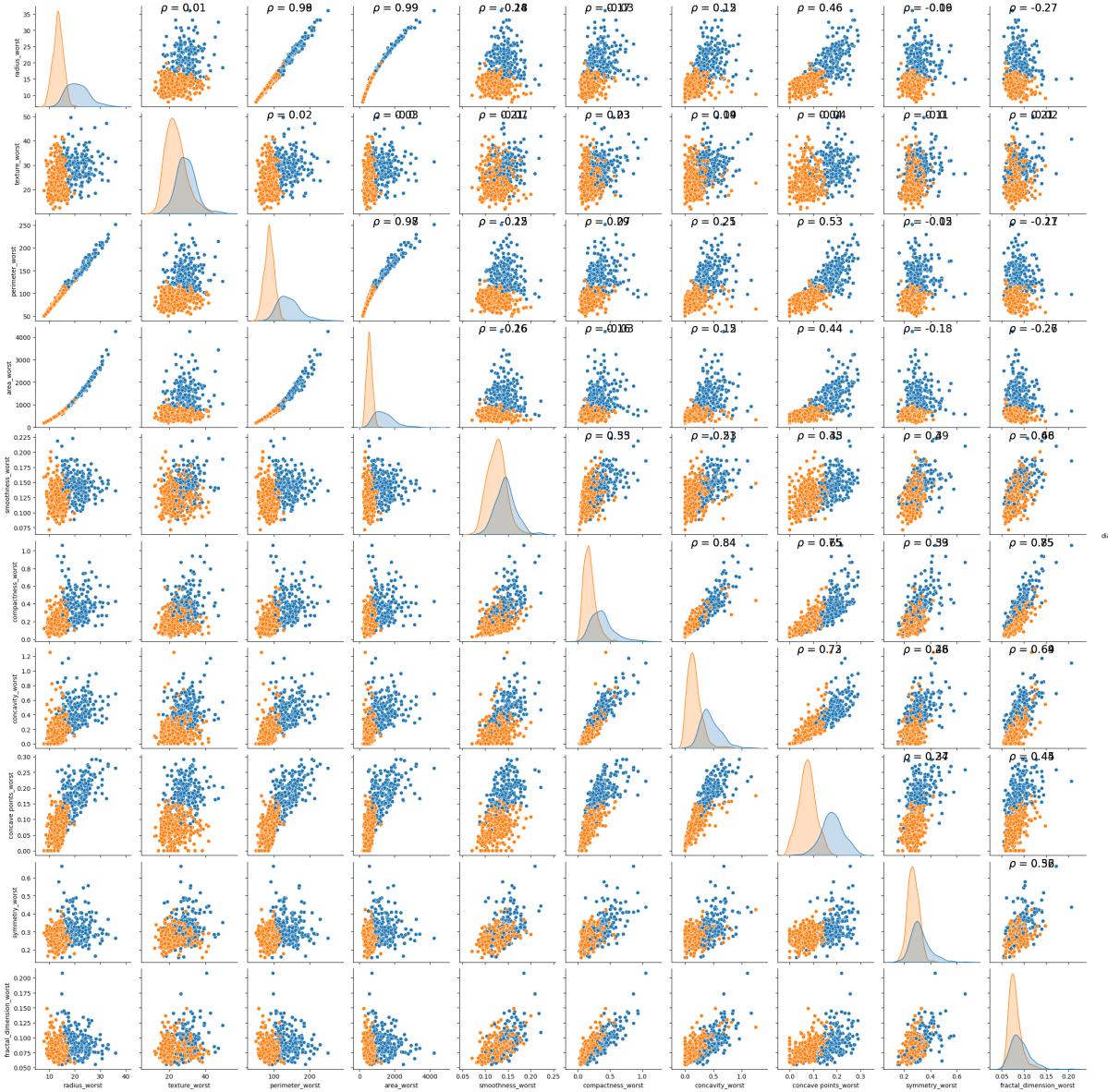


Fig. 12: Correlazioni variabili Worst

Dai grafici appena mostrati possiamo concludere che:

- Il sottogruppo mean è caratterizzato da correlazioni positive tra le variabili. Ciò suggerisce che le variabili sono correlate tra loro e condividono la stessa tendenza. Ad esempio, il diametro massimo è positivamente correlato con l'area, il che suggerisce che i tumori con un diametro maggiore tendono ad avere anche un'area maggiore.
- Il sottogruppo se è caratterizzato da correlazioni negative tra le variabili. Ciò suggerisce che le variabili sono correlate tra loro, ma in modo opposto. Ad esempio, la concavità è negativamente correlata con la concavità globale, il che suggerisce che i tumori con una concavità maggiore tendono ad avere una concavità globale minore.
- Il sottogruppo worst è caratterizzato da correlazioni positive e negative tra le variabili. Ciò suggerisce che le variabili sono correlate tra loro, ma in modo complesso. Ad esempio, il diametro massimo è positivamente correlato con l'area, ma negativamente correlato con la concavità.

3.4 Principal Component Analysis (PCA)

La *Principal Component Analysis* (o PCA) è una tecnica di riduzione della dimensionalità che ci permette di ottenere una serie di vantaggi:

- **riduzione del numero di variabili:** il nostro dataset presenta un alto numero di variabili, riducendole attraverso la PCA manteniamo la maggior parte della varianza, semplificando però notevolmente l'analisi e la visualizzazione dei dati.

Inoltre la presenza di un elevato numero di variabili porta all'*overfitting*: ovvero ad un sovradattamento durante la fase di addestramento del modello, vediamo come la PCA è in grado di mitigare questo problema.

- **riduzione del rumore:** la PCA tende a mantenere solo informazioni rilevanti, ovvero quelle che "spiegano" la maggior parte della varianza nei dati.
- **riduzione della multicollinearità:** la PCA trasforma il dataset originale in un nuovo sistema di coordinate dove le variabili create (principal components) non sono altamente correlate.
- **Semplificazione del modello:** come detto in precedenza, la presenza di troppe variabili porta il modello in una condizione di overfitting.
- **Migliore generalizzazione:** in presenza di un modello semplificato (con meno variabili) e di una riduzione del rumore il modello avrà una capacità migliore nel riconoscere i pattern rilevanti.
- **efficienze dei modelli:** riducendo la dimensionalità si può migliorare l'efficienza e le prestazioni degli algoritmi di apprendimento.

3.5 Applicazione della PCA e i suoi risultati

Prima di applicare la PCA dobbiamo standardizzare i dati, questo passaggio è essenziale per garantire che tutte le variabili abbiano lo stesso peso nell'analisi. Successivamente applichiamo la PCA e andiamo a visualizzare l'*explained variance* per ogni *principal components*.

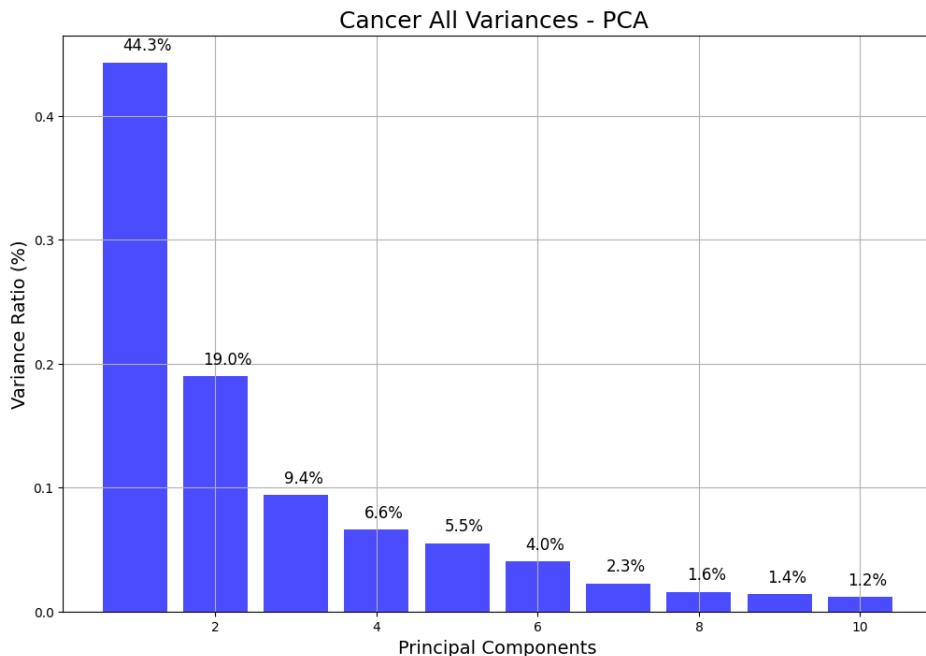


Fig. 13: Explained variance delle prime 10 PCs

Ogni principal component rappresenta una percentuale di *explained variance* della varianza totale del dataset di partenza.

Ora andiamo a rappresentare graficamente il contributo delle variabili nelle prime due principal components, le quali rappresentano la maggior parte di *explained variance*.

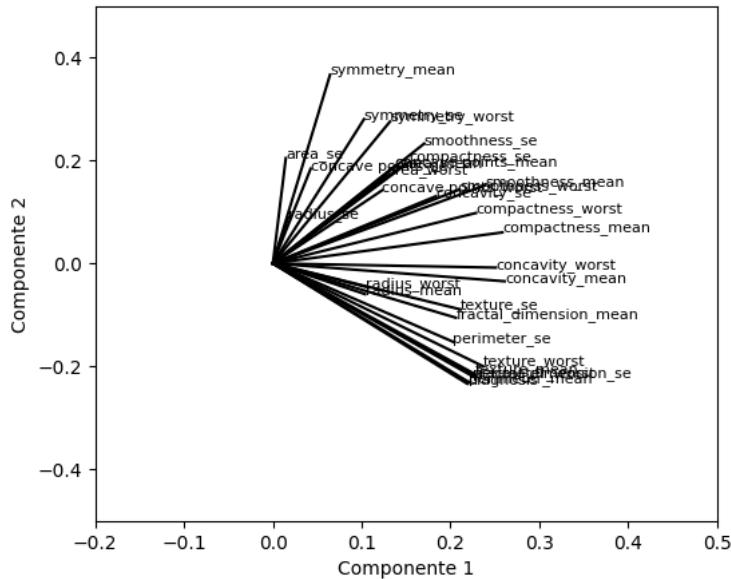


Fig. 14: Contributo delle variabili nelle prime due principal components

Come dicevamo precedentemente la PCA è uno strumento in grado di facilitare la visualizzazione dei dati, ho scelto quindi di rappresentare una serie di confronti fra le principal components che ho ottenuto:

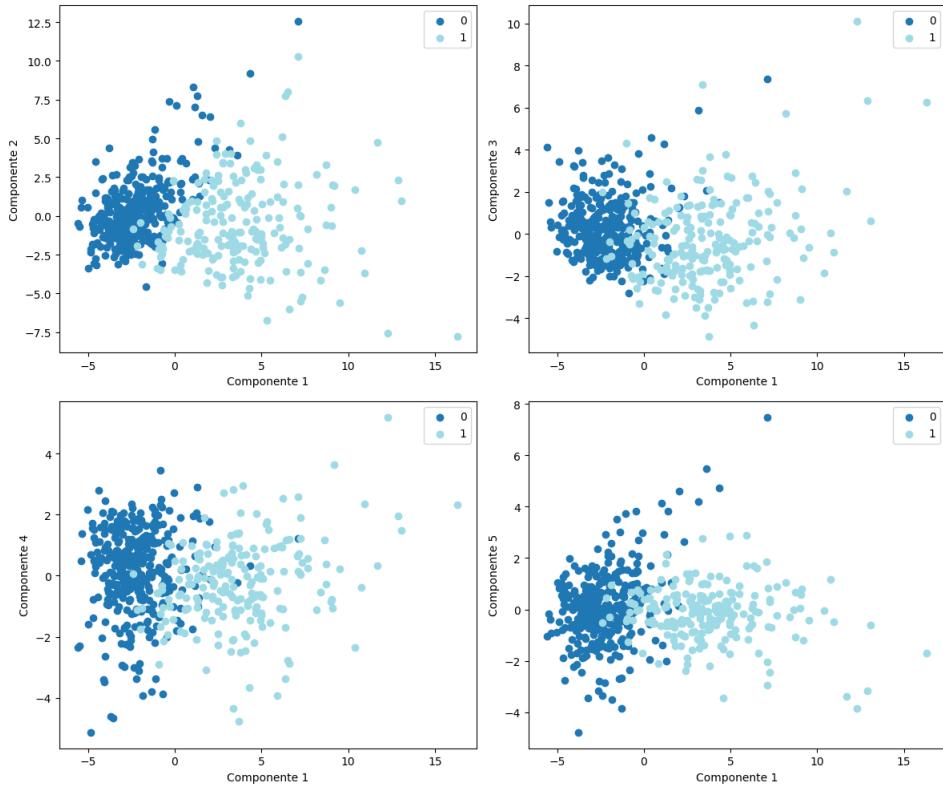


Fig. 15: Confronto fra le principal components

Da questo grafico possiamo notare che i dati assumono una separazione abbastanza netta in due cluster distinti; questo grado di separazione ci porta a scegliere di utilizzare il classificatore *Decision Tree*.

3.6 Gli obiettivi dello studio

Analizzati i risultati della PCA decidiamo di prendere in esame due dataset diversi:

- Il primo dataset sarà composto dalle prime **sei principal components**, la sua *explained variance* è pari a **88,8%**
- Il secondo dataset sarà composto dalle prime **nove principal components**, la sua *explained variance* è pari a **94,1%**

da cui analizzeremo una serie di fattori:

- Risposta dei modelli a fronte di:
 - Dimensionalità differenti
 - *Explained variance* differenti
- Performance, in termini di tempo ma anche di capacità predittive
- Presenza di *overfitting* nei risultati finali

4 Modelli

Fino a questo punto, abbiamo eseguito un'analisi esplorativa del dataset e abbiamo descritto l'applicazione della PCA. Ora passeremo a illustrare i modelli di Machine Learning utilizzati per prevedere se un tumore è benigno o maligno.

Si procederà con il seguente ordine: Inizieremo descrivendo il modello decision tree in primo luogo. Successivamente, forniremo dettagli sulle reti neurali impiegate. Condivideremo alcuni dettagli sull'implementazione dei modelli nel linguaggio Python. Infine, concluderemo con una sintesi dei modelli implementati. I risultati dei nostri esperimenti condotti con questi modelli saranno presentati nel capitolo successivo.

Ricordiamo che per l'addestramento dei modelli sono stati applicati due versioni dello stesso dataset, ovvero quello della PCA ma con due Explained Variance differenti: per verificare quale dei due generalizza meglio i nostri dati.

4.1 Decision tree

Motivazioni della scelta del modello

- I dati hanno una buona separazione come possiamo vedere nel grafico del clustering (quello nella PCA)
- La variabile target diagnosis è categorica binaria (Maligno o Benigno), perfettamente gestibile da un modello di classificazione come un Decision Tree
- per migliorare la generalizzazione e la performance del modello usiamo la potatura (o “pruning”) che è un processo che rimuove parte dell'albero decisionale, riducendo la sua complessità e aiutando a prevenire l'overfitting.

4.1.1 Training con dataset PCA con 6 features

Per prevenire l'overfitting è stata utilizzata una tecnica di post pruning, quindi andiamo a calcolare il pruning path:

Il quale consiste in un insieme di valori alpha (identificati come `ccp_alpha`) e le relative impurities. Le impurities possono essere rappresentate da più metriche, come ad esempio il Gini: il quale descrive quanta "incertezza" è presente nella classe di appartenenza; una classe rappresentata da soli esempi "Benign" avrà un'incertezza pari a 0, e così via, poi andremo ad allenare il modello Decision Tree al variare di `ccp_alpha`. Nel seguente grafico è possibile notare la il numero di nodi e la profondità dell'albero al variare del `ccp_alpha`.

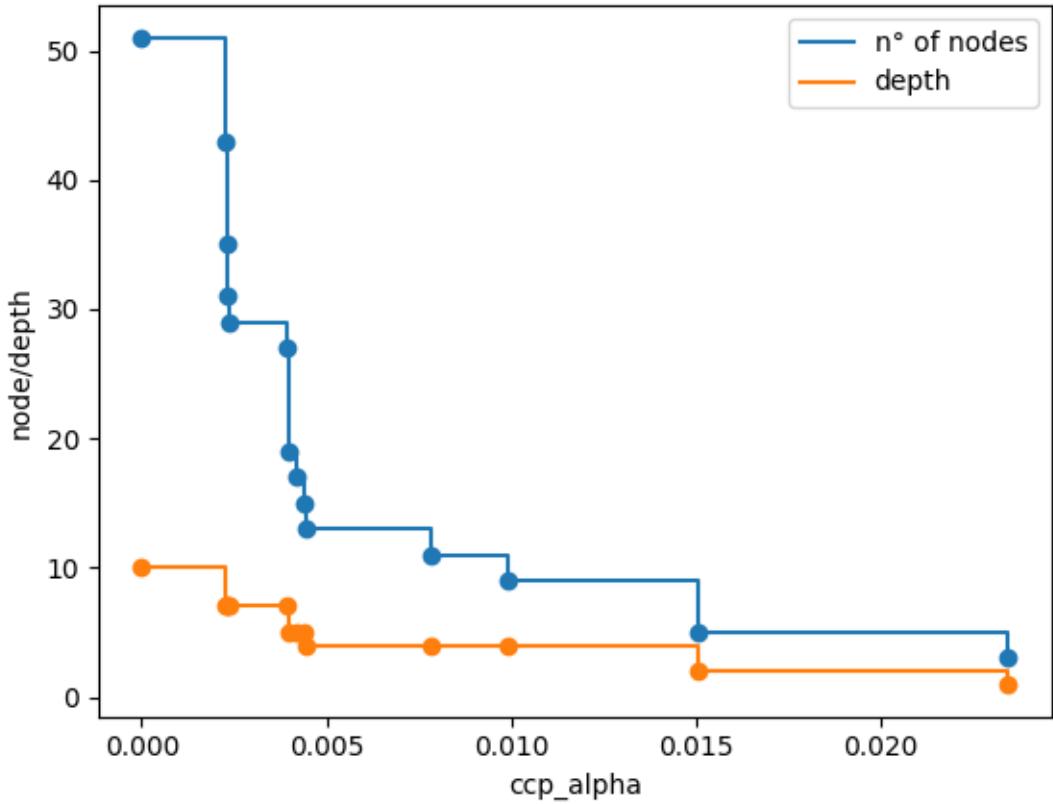


Fig. 16: pruning path

Ora andiamo a valutare l'accuracy dei classificatori inseriti nella lista `clfs`; creo altre due liste che mi serviranno a plottare in un grafico le accuracy relative alla fase di training e testing.

Il plot finale ci aiuta quindi a ricavare il valore di `ccp_alpha` ottimale: il quale massimizza le prestazioni del modello sul set di test, di fatto mitigando l'overfitting.

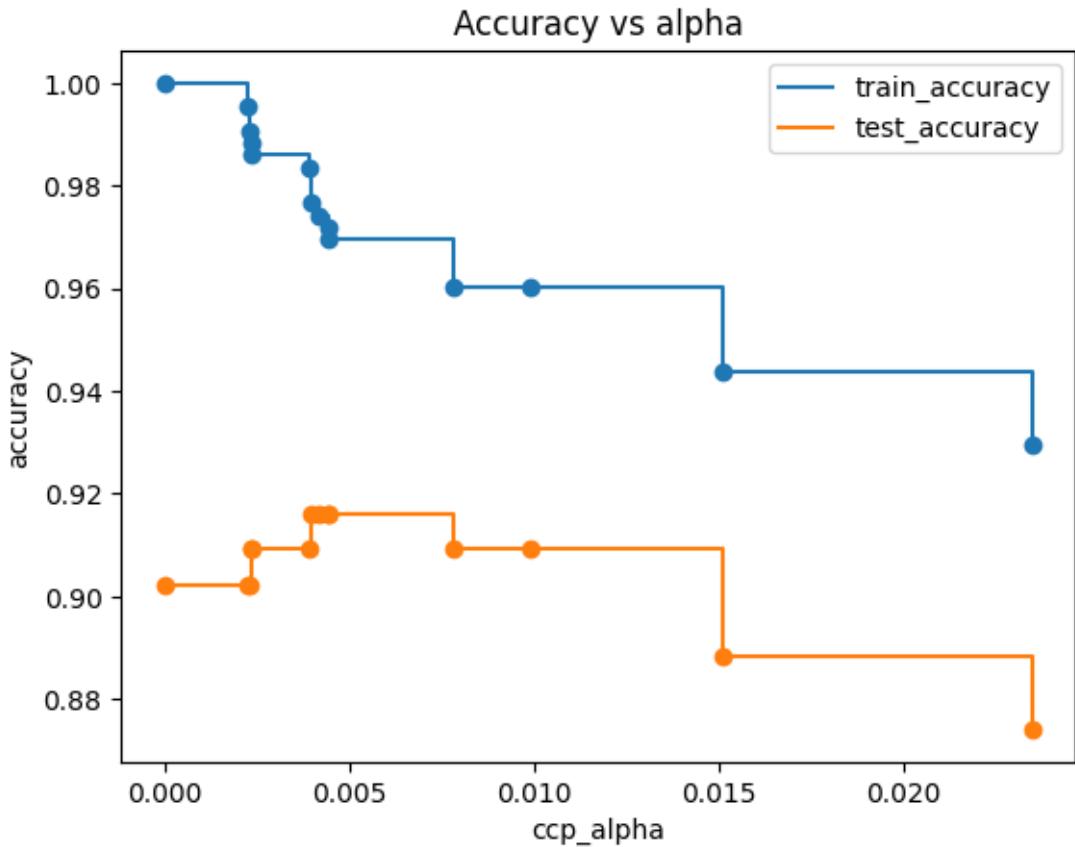


Fig. 17: valore di `ccp_alpha` ottimale

Una volta scelto un alpha che adeguato alleno nuovamente il decision tree e analizzo i risultati ottenuti.

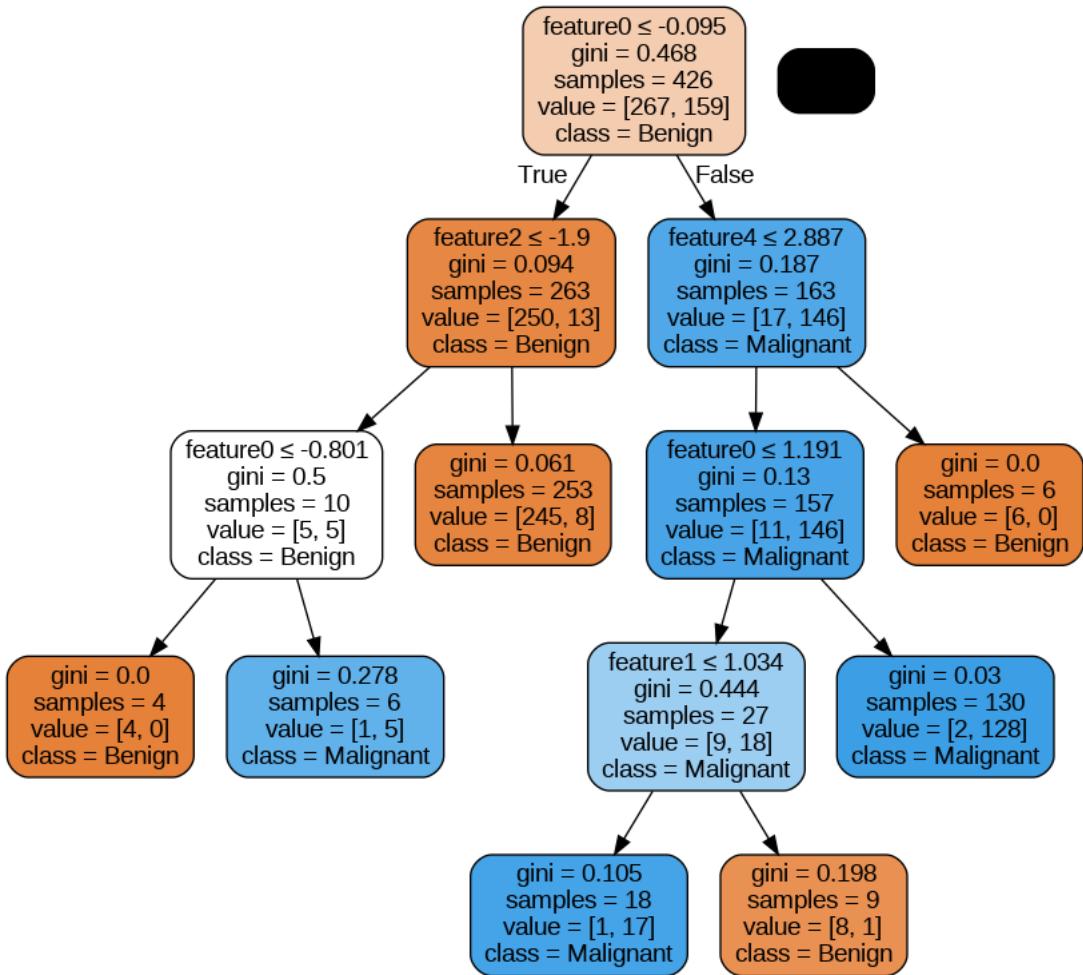


Fig. 18: Decision tree 6 features

4.1.2 Training con dataset PCA con 9 features

nel seguente grafico è possibile notare la il numero di nodi e la profondità dell'albero al variare del `ccp_alpha`.

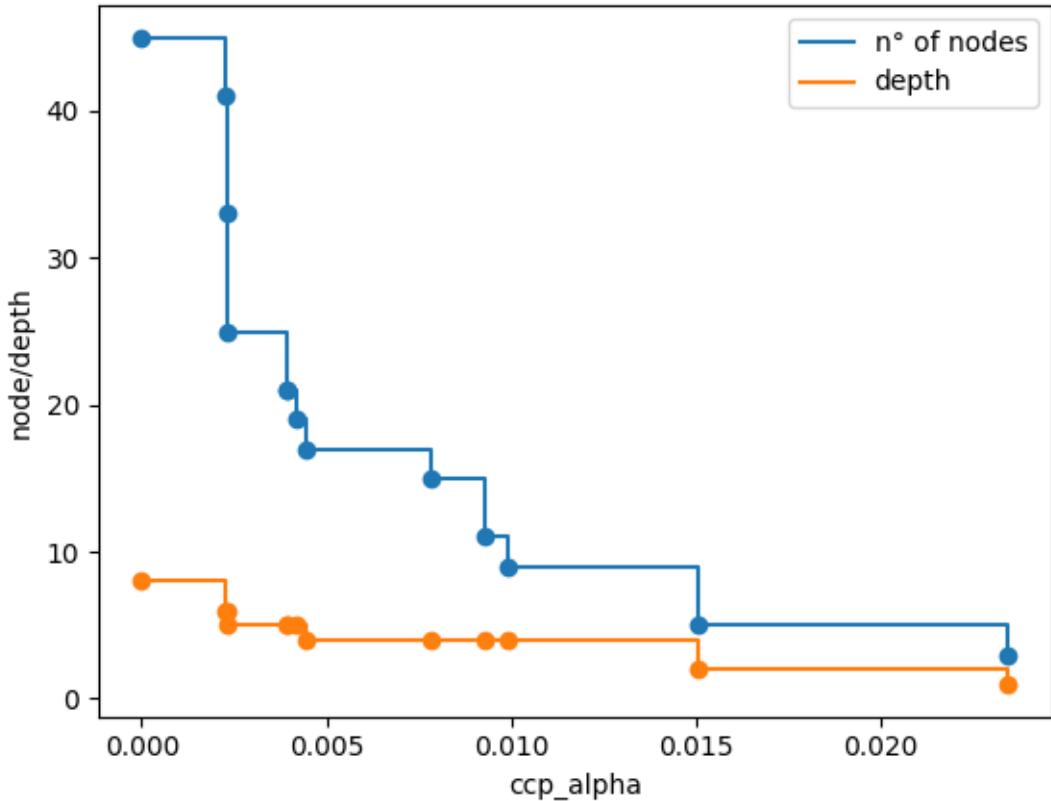


Fig. 19: pruning path

Ora andiamo a valutare l'accuracy dei classificatori inseriti nella lista `clfs`; creo altre due liste che mi serviranno a plottare in un grafico le accuracy relative alla fase di training e testing.

Il plot finale ci aiuta quindi a ricavare il valore di `ccp_alpha` ottimale: il quale massimizza le prestazioni del modello sul set di test, di fatto mitigando l'overfitting.

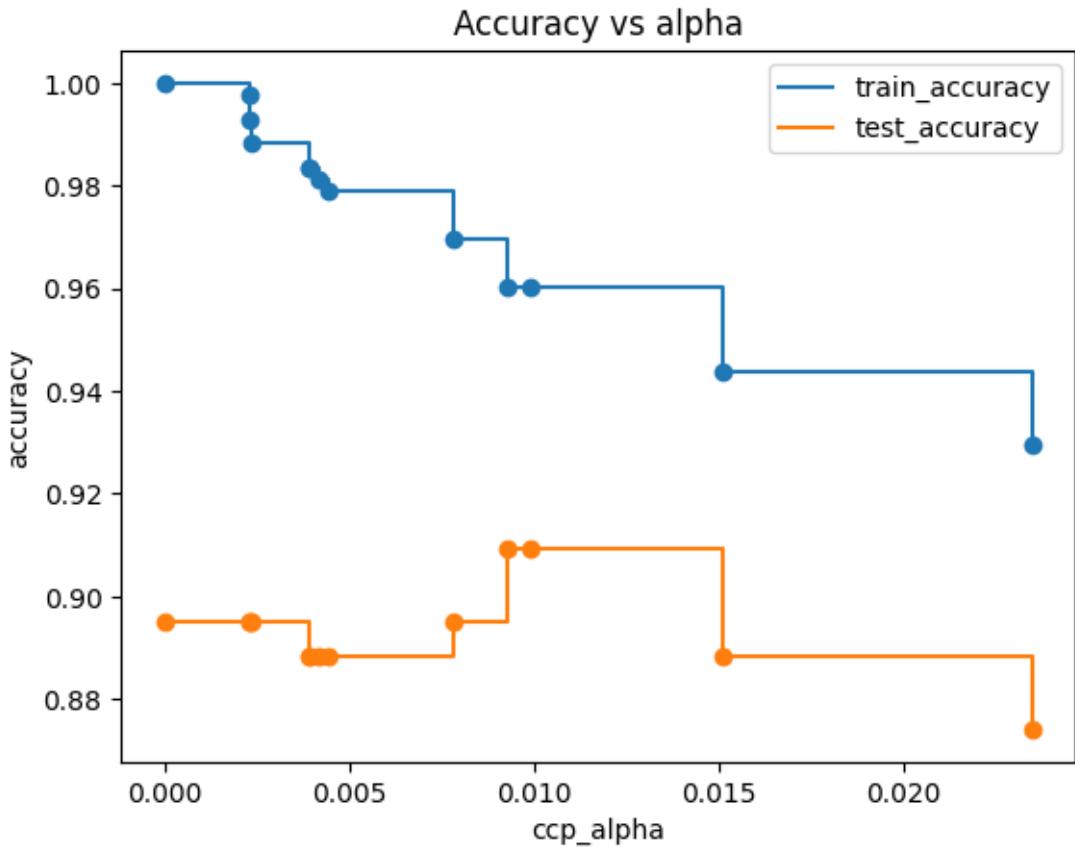


Fig. 20: valore di `ccp_alpha` ottimale

Una volta scelto un alpha che adeguato alleno nuovamente il decision tree e analizzo i risultati ottenuti.

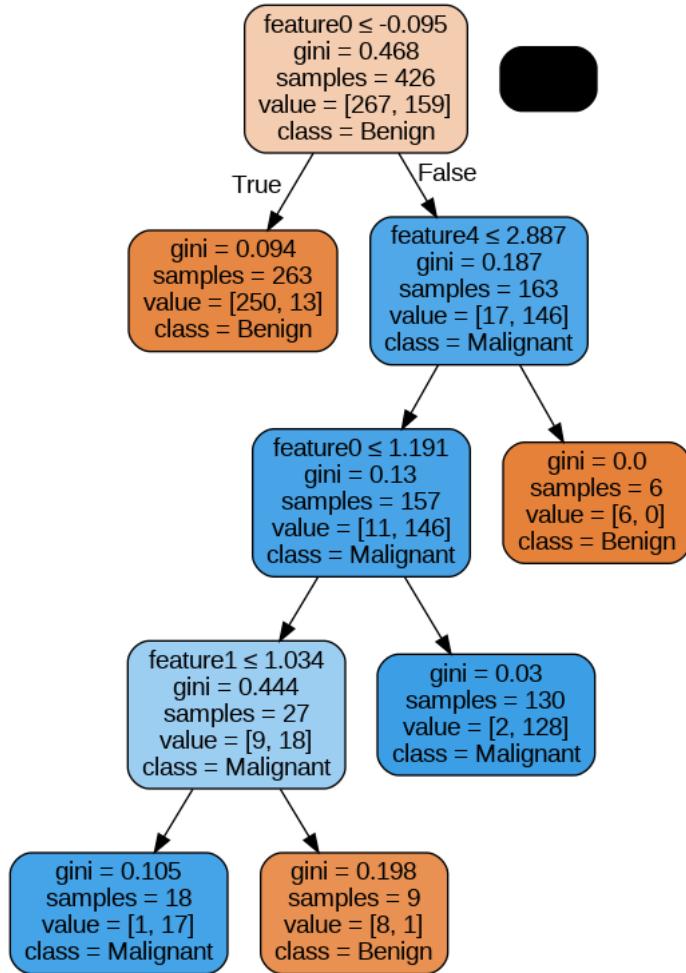


Fig. 21: Decision tree 9 features

4.2 Reti neurali

Il secondo modello predittivo che abbiamo deciso di utilizzare è la Rete Neurale. Le ragioni per questa scelta sono le seguenti:

- Le reti neurali sono estremamente tolleranti agli errori e al rumore;
- Sono capaci di classificare pattern complessi che non sono linearmente separabili, motivo per cui abbiamo preferito questo modello al percepitrone semplice;
- Possono essere facilmente aggiornate con nuove osservazioni;

Dettagli sulle reti neurali

Come accennato in precedenza, le reti neurali sono modelli di apprendimento supervisionato che, a differenza del percepitrone semplice che ha un solo neurone e può classificare solo insiemi linearmente indipendenti, permettono la creazione di reti di neuroni (da qui il nome) organizzate in diversi livelli. Questo rende le reti neurali adatte a lavorare su insiemi complessi non linearmente separabili.

Più specificamente, le reti neurali sono rappresentate graficamente come grafi, dove ogni nodo rappresenta un neurone e i nodi sono collegati tra loro da archi orientati e pesati:

- Ci sono neuroni di input, uno per ogni covariata da considerare per il calcolo del valore della variabile target;

- I neuroni di input sono collegati tramite archi pesati a tutti i neuroni definiti per il primo livello della rete. Inoltre, ogni neurone dei diversi livelli riceve in input anche un valore da un neurone aggiuntivo. Questo valore è detto soglia, e viene considerato per capire se tale neurone è da considerarsi attivo oppure no;
- I neuroni del primo livello sono collegati con archi pesati ai neuroni del secondo livello, e così via fino ad arrivare all'ultimo livello, dove sono presenti i neuroni di output;
- Per determinare se un neurone è attivo o meno si utilizza una specifica funzione di attivazione. Questa funzione, presi in input i segnali ricevuti dal neurone, calcola il suo valore. Se questo valore supera una certa soglia, allora il neurone sarà considerato attivo, altrimenti disattivo;
- L'ultimo livello della rete presenta tanti neuroni quanti sono i possibili valori assumibili dalla variabile target, nel caso questa fosse una variabile categorica come nel nostro caso. Ognuno di questi neuroni di output, data in input alla rete una istanza del problema, fornirà la probabilità che l'istanza sia classificata in quel tal modo;
- Le reti neurali possono essere facilmente aggiornate con nuove osservazioni;

Per il calcolo dei pesi ottimali da associare ai diversi archi si utilizza la strategia di back propagation. Questa strategia prevede l'inizializzazione casuale di tutti i pesi; successivamente, viene considerata ogni istanza del training set e per questa istanza viene calcolato tramite la rete il possibile valore per la variabile target: se il valore calcolato dalla rete coincide con quello effettivo, allora non avvengono modifiche; se invece il valore calcolato è diverso da quello atteso, si calcola l'errore e si propaga il calcolo all'indietro a partire dai neuroni di output, aggiustando progressivamente i pesi dei diversi archi. Per quanto riguarda la funzione di attivazione considerata nei nostri esperimenti, abbiamo utilizzato la funzione logistica, definita come segue:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

$$x = \sum_{i=1}^n WiSi \quad (2)$$

Il valore x viene calcolato come sommatoria pesata dei valori forniti dai neuroni in input. Si noti che questa sommatoria considera anche il valore di soglia, perché viene fornito anch'esso in input da un neurone definito appositamente.

4.2.1 Reti neurali utilizzate

Nella nostra analisi, abbiamo impiegato due diverse architetture di reti neurali per la classificazione. La prima rete, è composta da un input layer che accetta 6 Features, seguito da due strati nascosti con 16 e 8 neuroni rispettivamente, e un output layer con un singolo neurone con funzione di attivazione sigmoidale per produrre un output binario. La seconda rete, si differenzia per l'input layer che gestisce 9 Features, mantenendo la stessa struttura degli strati nascosti e dell'output layer della prima rete. Entrambe le reti utilizzano la funzione di attivazione ReLU per gli strati nascosti e la funzione sigmoidale per l'output, ottimizzate con Adam e addestrate con la funzione di perdita `binary_crossentropy`.

Di seguito mostriamo le 2 reti utilizzate per l'addestramento:

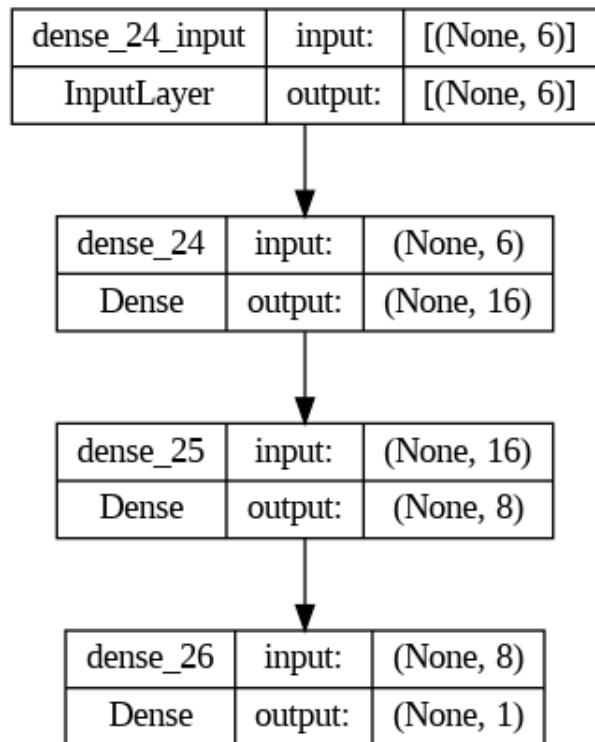


Fig. 22: Rete Neurale per il dataset PCA con 6 features

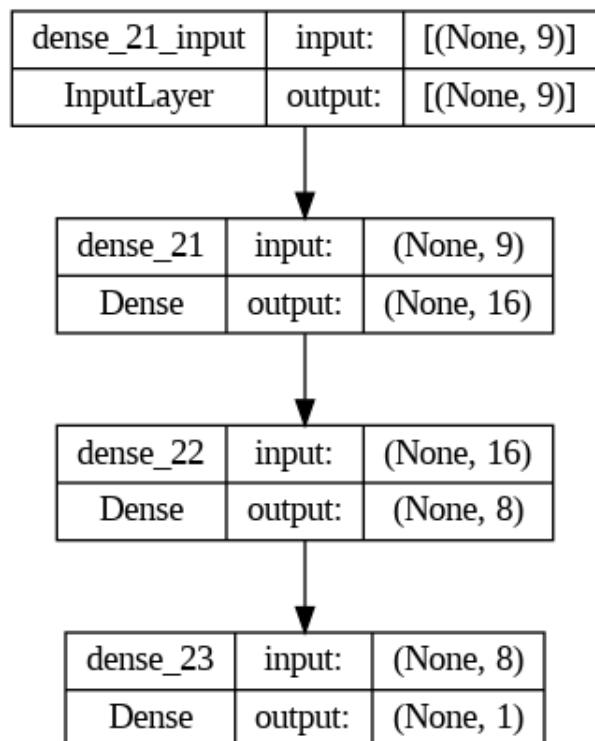


Fig. 23: Rete Neurale per il dataset PCA con 9 features

5 Esperimenti eseguiti

In questo capitolo illustreremo i risultati ottenuti dagli esperimenti condotti con i modelli descritti nei paragrafi precedenti. Si seguirà la seguente sequenza. Prima si examineranno le matrici di confusione prodotte dai modelli. Poi si calcolerà l'accuratezza dei modelli. Dopo si presenteranno altre misure di performance relative alla capacità di classificazione dei modelli. Poi si mostreranno le curve ROC e i corrispondenti valori AUC e la 10-fold cross-validation. Quindi si dettaglieranno i tempi di computazione dei modelli. Infine si fornirà un confronto generale tra alberi decisionali e reti neurali.

5.1 Matrici di confusione

Anzitutto vengono presentate le matrici di confusione derivanti dalle predizioni effettuate dai modelli. Vediamo ora separatamente le matrici di confusione generate (i) del Decision Tree e (ii) dalle Reti Neurali

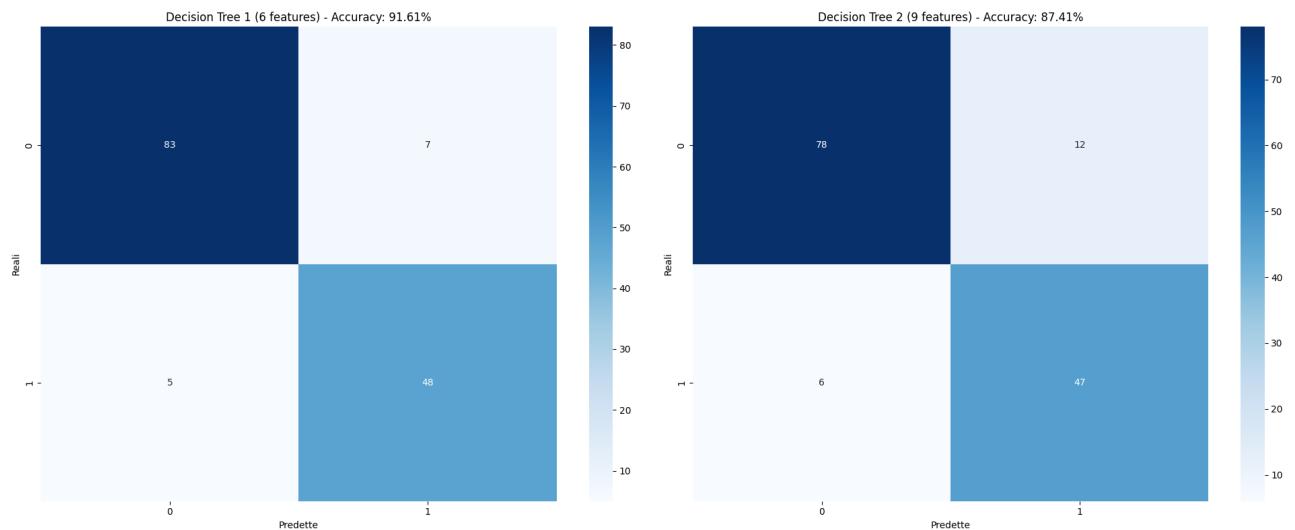


Fig. 24: Matrici di confusione modelli Decision Tree

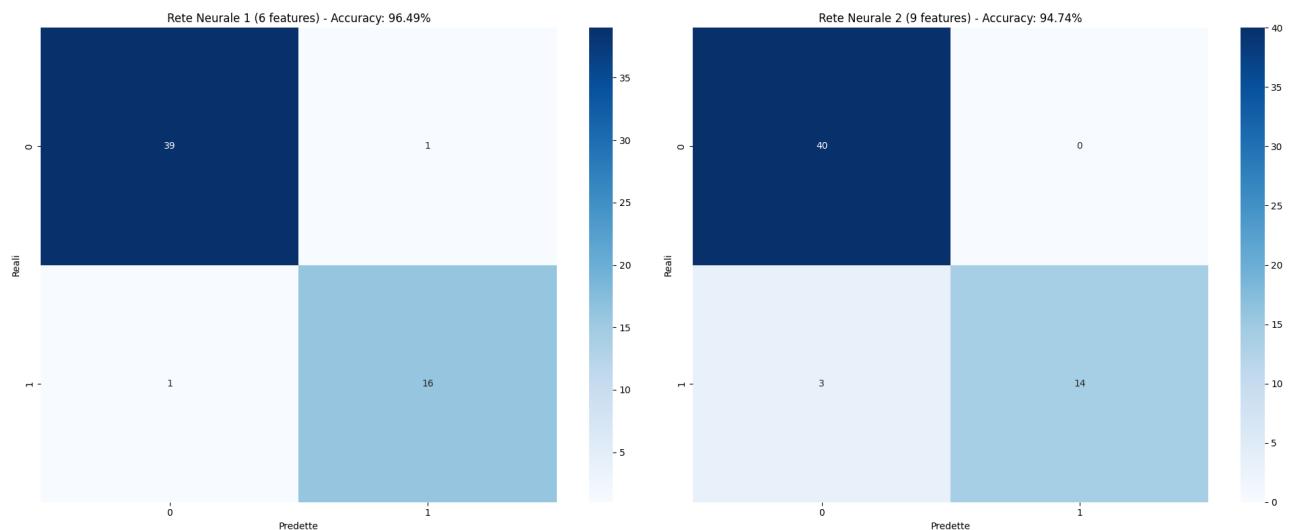


Fig. 25: Matrici di confusione Reti Neurali

5.2 Accuratezza

La proporzione di istanze classificate correttamente su tutte le istanze è misurata dall'accuratezza. Dalla matrice di confusione del modello si può calcolare facilmente l'accuratezza. Nel caso di un problema binario l'accuratezza è data da:

$$\text{Accuratezza} = \frac{TP + TN}{TP + TN + FP + FN}$$

Dove TP e TN sono le istanze corrette per la classe positiva e negativa, mentre FP e FN sono le istanze sbagliate. L'accuratezza si può ottenere dalla matrice di confusione sommando gli elementi della diagonale principale e dividendo per il totale degli elementi della matrice. Questo metodo si può usare anche per l'accuratezza di un modello in un problema multi-classe.

L'accuratezza ottenuta dal Decision Tree è la seguente:

	Decision tree (6 features)	Decision tree (9 features)
Accuracy	0.916083916083916	0.8741258741258742

Fig. 26: Accuracy Decision Tree

L'accuratezza ottenuta dalle Reti Neurali è la seguente:

	Rete neurale (6 features)	Rete neurale (9 features)
Accuracy	0.9824561476707458	0.9473684430122375

Fig. 27: Accuracy Rete neurale

Rispetto a K-means, le reti neurali hanno ottenuto risultati migliori. Verranno successivamente analizzati i tempi computazionali relativi ai due modelli per capire quale effettivamente è il migliore per il problema analizzato.

5.3 Precision, Recall e F1-Measure

Dalla matrice di confusione si possono calcolare altre misure di performance (oltre all'accuratezza). In particolare: La Precision indica quanto il modello è preciso nel classificare le istanze di una certa classe. Quindi c'è una Precision per ogni classe. La Precision di un problema binario si può calcolare come:

$$\text{Precision} = \frac{TP}{TP + FP}$$

I valori di TP e FP si trovano nella matrice di confusione. In questo modo la matrice di confusione permette il calcolo della Precision di ogni classe. E questo vale anche per i problemi multi-classe con le opportune modifiche.

Ecco una possibile riscrittura del testo con lo stesso significato:

La Recall è una metrica che indica la capacità del modello di individuare tutte le istanze di una certa classe in un dataset. Infatti la Recall di un problema binario è data da:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Anche questa metrica si può ottenere dalla matrice di confusione. E questo vale anche per i problemi multi-classe.

La metrica F1-Measure per un problema binario è data da:

$$F1 - \text{Measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

La metrica F1-Measure è la media armonica di Precision e Recall. Il valore di questa metrica è alto quando i valori di Precision e Recall del modello per una classe sono simili. Questa metrica si può calcolare anche per problemi multi-classe, conoscendo i valori di Precision e Recall.

I valori ottenuti dal Decision Tree per queste misure di performance sono i seguenti:

	Decision tree (6 features)	Decision tree (9 features)
Precision Malignant	0.87	0.80
Precision Benign	0.94	0.93
Precision Macro Average	0.91	0.86
Recall Malignant	0.91	0.89
Recall Benign	0.92	0.87
Recall Macro Average	0.91	0.88
F1 Malignant	0.89	0.84
F1 Benign	0.93	0.90
F1 Macro Average	0.91	0.87

Fig. 28: Misure di performance di Decision Tree

I valori ottenuti dalle Reti Neurali per queste misure di performance sono i seguenti:

	Rete neurale (6 features)	Rete neurale (9 features)
Precision Malignant	0.84	1.00
Precision Benign	0.97	0.95
Precision Macro Average	0.91	0.98
Recall Malignant	0.94	0.88
Recall Benign	0.93	1.00
Recall Macro Average	0.93	0.94
F1 Malignant	0.89	0.94
F1 Benign	0.95	0.98
F1 Macro Average	0.92	0.96

Fig. 29: Misure di performance della Rete neurale

L'aggregazione dei valori di performance è stata effettuata tramite Macro Average, come fatto per Decision Tree. Complessivamente i valori registrati sono molto simili a quelli osservati con Decision Tree.

5.4 Curve ROC e AUC

In questo paragrafo si analizzeranno le curve ROC e i valori AUC dei modelli. Prima di farlo diamo una definizione (almeno intuitiva) di cosa siano le curve ROC e i valori AUC. In particolare:

- La curva ROC mostra la performance di un classificatore binario (in cui la popolazione è divisa tra la classe positiva e la classe negativa). Più precisamente, mostra il valore di TPR (relativo alla frazione di veri positivi di una classe) rispetto al valore di FPR (relativo alla frazione di falsi positivi di una classe) al variare di una soglia. Il valore di TPR viene anche chiamato Sensitivity, mentre il valore di FPR può essere anche calcolato come $1 - \text{Specificity}$;
- L'area sotto la curva ROC prende il nome di AUC (acronimo di Area Under Curve). Il valore di AUC può essere interpretato come la probabilità che un'istanza scelta a caso dalla popolazione dei positivi sia classificata con un valore maggiore rispetto a quello ottenuto scegliendo a caso un'istanza dalla popolazione dei negativi.

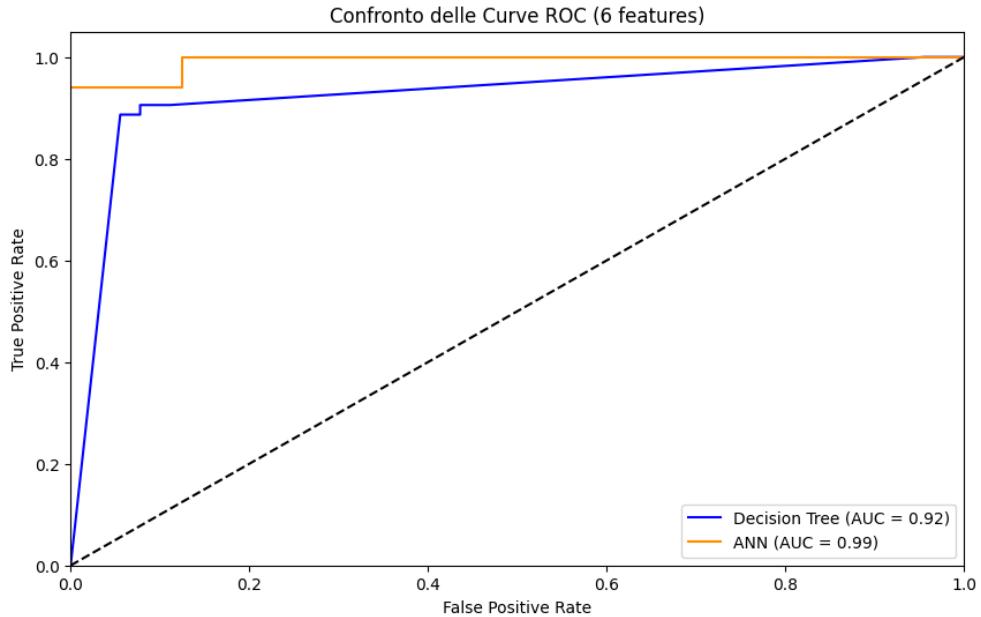


Fig. 30: Confronto curve ROC modelli 6 features

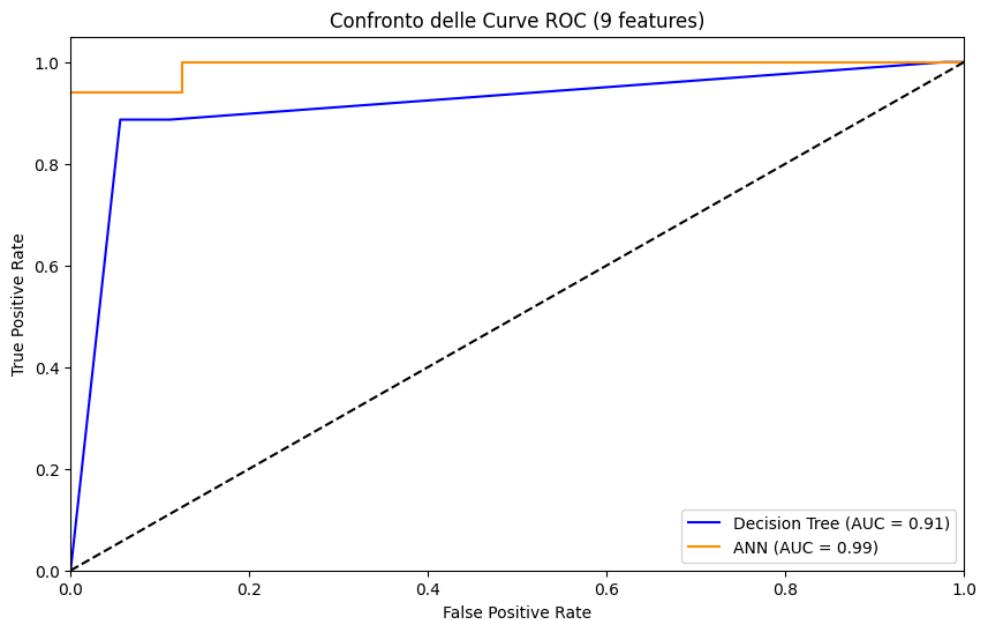


Fig. 31: Confronto curve ROC modelli 9 features

5.5 10-fold cross-validation

La 10-fold cross-validation è una tecnica per valutare la capacità di generalizzazione di un modello. Il dataset viene diviso in 10 parti (o "fold") di dimensioni uguali. Per ogni iterazione, 9 fold vengono usati per addestrare il modello e 1 fold viene usato come set di validazione per testare il modello. Questo processo viene ripetuto 10 volte, cambiando ogni volta il fold di validazione.

I vantaggi di questo metodo includono:

- Utilizzo efficiente dei dati: Ogni osservazione viene usata sia per l'addestramento che per la validazione, massimizzando l'uso dei dati disponibili.
- Riduzione della varianza: Dato che il processo di validazione viene ripetuto 10 volte su set di dati diversi, si riduce il rischio che la performance del modello sia influenzata da una particolare divisione dei dati.

Gli intervalli di confidenza forniscono una stima dell'incertezza associata alla media dell'accuracy ottenuta dalla cross-validation. In altre parole, danno un'idea di quanto ci si può aspettare che la media dell'accuracy vari se il processo di cross-validation fosse ripetuto su campioni diversi dello stesso popolazione. Questo è utile per avere una misura della stabilità della stima dell'accuracy.

Di seguito vengono mostrati i risultati della 10 cross-validation e gli intervalli di confidenza per i 2 modelli:

Accuracy per fold: [Intervallo di confidenza al 90%:
0.9298245614035088,	(0.9249608500347222,
0.9649122807017544,	0.9555529344264309)
0.9824561403508771,	
0.9122807017543859,	
0.9649122807017544,	
0.8947368421052632,	
0.9298245614035088,	
0.9473684210526315,	
0.9298245614035088,	
0.9464285714285714]	

Fig. 32: 10 cross-validation e intervalli di confidenza Decision Tree 6 features

Accuracy per fold: [Intervallo di confidenza al 90%:
0.9473684210526315,	(0.9206912743508598,
0.9122807017543859,	0.9457247657493907)
0.9473684210526315,	
0.9298245614035088,	
0.9649122807017544,	
0.8947368421052632,	
0.9473684210526315,	
0.9473684210526315,	
0.9122807017543859,	
0.9285714285714286]	

Fig. 33: 10 cross-validation e intervalli di confidenza Decision Tree 9 features

Accuracy per fold: [Intervallo di confidenza al 90%:
0.9473684210526315,	(0.9490846271836341,
1.0,	0.9842487061496991)
0.9824561403508771,	
0.9649122807017544,	
0.9649122807017544,	
0.9298245614035088,	
1.0,	
0.9122807017543859,	
0.9649122807017544,	
1.0]	

Fig. 34: 10 cross-validation e intervalli di confidenza Rete Neurale 6 features

Accuracy per fold: [Intervallo di confidenza al 90%:
0.9473684210526315,	{0.9480012374752675,
0.9649122807017544,	0.9747431234269883}
0.9824561403508771,	
0.9649122807017544,	
0.9473684210526315,	
0.9298245614035088,	
1.0,	
0.9298245614035088,	
0.9649122807017544,	
0.9821428571428571]	

Fig. 35: 10 cross-validation e intervalli di confidenza Rete Neurale 9 features

5.6 Tempi di computazione

Infine si riportano i tempi di computazione dei modelli.

	Tempo (secondi)
Rete neurale (6 features)	13.86
Rete neurale (9 features)	21.11
Decision tree (6 features)	0.01
Decision tree (9 features)	0.01

Fig. 36: Confronto delle performance tra i modelli

6 Conclusioni

Nella presente relazione, abbiamo esplorato l'applicazione di tecniche di Machine Learning per la diagnosi del cancro al seno, un'area di ricerca critica data la prevalenza e la gravità di questa malattia. Attraverso un'analisi dettagliata del dataset Breast Cancer Wisconsin Diagnostic (WDBC), abbiamo implementato e valutato due modelli predittivi principali: Decision Tree e Reti Neurali.

L'analisi esplorativa ha rivelato differenze significative tra tumori benigni e maligni in termini di caratteristiche come dimensione, forma e texture, sottolineando l'importanza di queste variabili nella diagnosi. La Principal Component Analysis (PCA) è stata utilizzata per ridurre la dimensionalità del dataset, mantenendo al contempo la maggior parte delle informazioni rilevanti, il che ha permesso di semplificare i modelli e ridurre i tempi di addestramento senza compromettere significativamente l'accuratezza.

I risultati degli esperimenti hanno dimostrato che entrambi i modelli hanno raggiunto un'alta accuratezza nella classificazione dei tumori, con le Reti Neurali che hanno mostrato una leggera superiorità rispetto ai Decision Tree. Tuttavia, è importante notare che i Decision Tree hanno offerto il vantaggio di una maggiore interpretabilità e tempi di addestramento notevolmente inferiori.

La 10-fold cross-validation ha confermato la robustezza dei modelli, fornendo stime affidabili della loro capacità di generalizzazione. Inoltre, l'analisi delle curve ROC e dei valori AUC ha ulteriormente validato l'efficacia dei modelli nel distinguere tra tumori benigni e maligni.

In conclusione, questo studio ha dimostrato il potenziale delle tecniche di Machine Learning nella diagnosi precoce del cancro al seno, offrendo strumenti preziosi per assistere i professionisti medici. Tuttavia, è fondamentale continuare a esplorare e sviluppare ulteriormente questi modelli, integrando nuovi dati e tecniche, per migliorare la precisione e l'efficienza della diagnosi. La ricerca futura potrebbe includere l'esplorazione di altri modelli di Machine Learning, l'ottimizzazione dei parametri esistenti e l'analisi di nuove caratteristiche dei dati per arricchire ulteriormente la nostra comprensione e capacità di diagnosi del cancro al seno.