Report del progetto di Machine Learning

Ferrario Tommaso Matr. 869005 (@TommasoFerrario18)

Terzi Telemaco Matr. 865981(@Tezze2001)

Vendramini Simone Matr. 866229(@Svendra4MySelf)

15 febbraio 2024

Indice

Capitolo 1

Introduzione

Questo è un progetto per l'esame di Machine Learning del primo anno del corso di laurea magistrale in informatica dell'Università degli Studi di Milano-Bicocca.

L'intero progetto si basa sul riconoscimento della presenza di un tumore al cervello data l'immagine di una risonanza magnetica.

Il dataset scelto per questo progetto è scaricabile dal seguente link ed è composto da un insieme di features estratte dalle immagini ottenute dalle risonanze magnetiche del cervello di diversi pazienti.

Per il riconoscimento del tumore sono stati allenati i seguenti modelli di machine learning:

- **SVM**: è stato scelto questo modello vista la buona capacità teorica nel generalizzare.
- **Gaussian Naive Bayes**: è stato scelto questo modello dal momento che permette di modellare le probabilità esplicitamente.
- **Rete neurale**: è stato scelto questo modello per confrontare i primi due con una soluzione neurale.

L'obiettivo sarà quello di trovare il modello migliore che riduca al minimo i falsi negativi, mantenendo comunque una buona precisione sui veri negativi.

Infine, la struttura dell'elaborato è delineata dai seguenti capitoli:

- **Introduzione**: descrizione del dominio e presentazione dei modelli che verranno presi in considerazione per questo progetto.
- **Dataset**: descrizione di come è stato costruito il dataset a partire dalle immagini, ovvero come sono state ricavate le features, e analisi esplorativa.
- **Rete neurale**: descrizione e analisi delle performance della rete.
- **SVM**: descrizione e analisi delle performance delle SVM.
- **Gaussian Naive Bayes**: descrizione e analisi delle performance per Gaussian Naive Bayes.
- **Analisi dei risultati**: analisi comparata dei risultati tra i tre modelli considerati.
- **Conclusioni**: conclusioni sull'elaborato.

Capitolo 2

Dataset

2.1 Struttura del dataset

Il dataset è composto da 13 features estratte da un set di 3762 immagini in bianco e nero, ciascuna immagine è stata prodotta dalla risonanza magnetica del cervello di diversi pazienti. Di conseguenza, si hanno un totale di 3762 istanze, ognuna etichettata con un valore categorico che rappresenta la presenza o meno del tumore al cervello. L'etichetta è presente sotto la colonna *Class* e assume i seguenti valori:

- **Presenza del tumore**: $T = 1$
- **Assenza del tumore**: $T = 0$

Le features vengono già date e si assumono che siano corrette rispetto alle risonanze magnetiche del dateset[1]. Più precisamente le features si distinguono in:

1. **First Order Features**: forniscono informazioni legate alle distribuzione dei livelli di grigio dell'immagine. Queste features corrispondono alle statistiche descrittive calcolate sui valori di ciascun pixel dell'immagine e corrispondono a:
    - **Media**
    - **Varianza**
    - **Deviazione standard**
    - **Indice di asimmetria**
    - **Indice di kurtosis**
2. **Second Order Features**: forniscono informazioni a livello di composizione della texture dell'immagine e si dividono in:
    - **Contrast**: misura la differenza tra i livelli di grigio tra diverse parti dell'immagine. Maggiore sarà il valore allora maggiore sarà la deviazione standard dei livelli di grigio nell'immagine.
    - **Energy**: fornisce informazioni sulla texture e sulla complessità. Maggiore sarà il valore di Energy, allora maggiore sarà il contrasto oppure più dettagliata sarà la texture.
    - **ASM**: misura quanto sono distribuiti uniformemente i livelli di grigio nell'immagine. Maggiore sarà il valore allora più uniforme sarà la distribuzione dei livelli di grigio nell'immagine, quindi la variabilità dei livelli di grigio è ridotta.
    - **Entropy**: misura la randomicità dei livelli di grigio, quindi l'entropia sarà massima quando tutti i livelli di grigio equalmente probabili (randomness). Più precisamente immagini con un ampio range di valori dei pixel e distribuzioni uniformi di intensità tendono a aumentare il valore dell'entropia.
    - **Homogeneous**: misura quanto sono uniformi i livelli di grigio. Più alto sarà l'indice allora minore sarà il contrasto dell'immagine.
    - **Dissimilarity**: misura quanto differiscono diverse regioni dell'immagine. Un valore alto indica che si hanno molte differenze tra diverse regioni della stessa immagine, quindi più complessa sarà la texture.
    - **Correlation**: misura la correlazione dei livelli di grigio tra diverse regioni della stessa immagine.
    - **Coarseness**: misura il grado di variazione o di irregolarità dei livelli di grigio, quindi misura la finezza o la granularità della texture.

2.2 Analisi descrittiva

Caricato il dataset, è stato eseguito un controllo per verificare che non ci fossero valori nulli. In questo caso non sono stati trovati valori nulli, quindi non è stato necessario eseguire alcuna operazione per la gestione di tali valori. In secondo luogo è stato controllato se il dataset contiene dei duplicati e sono stati trovati un totale di 63 duplicati e di conseguenza sono stati tutti rimossi.

Successivamente, si è proceduto controllando se le classi del dataset sono sbilanciate, per fare ciò è stato creato un instogramma che mostra la frequenza dei valori della colonna *Class* (visibile nella figura 2.1).

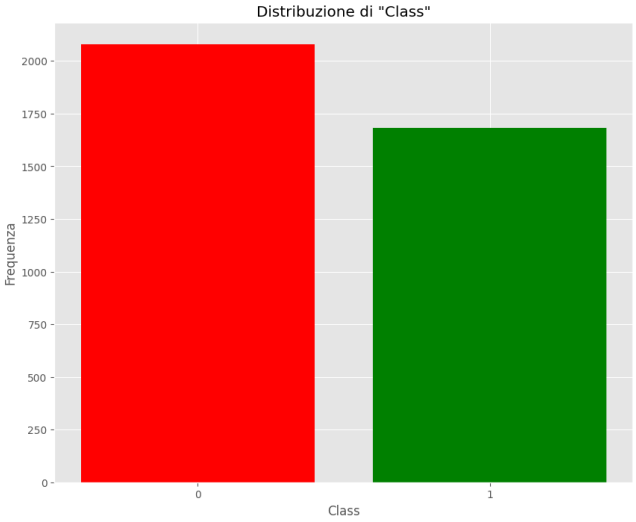


Figura 2.1: Distribuzione delle classi

Dall'istogramma si evidenza che le classi sono abbastanza bilanciate, infatti, il dataset è composto dal 45% di esempi positivi, mentre il 55% è composto da esempi negativi.

Successivamente sono stati costruiti 13 istogrammi, uno per ogni features in modo tale da analizzare visivamente la loro distribuzione (i grafici sono visibili nella figura 2.2).

Da questi grafici si evince che le features *Energy*, *ASM*, *Homogeneity*, *Entropy* e *Coarseness* non seguono una distribuzione normale, a differenza delle altre feature che hanno un andamento simile ad una gaussiana. In ogni caso, anche se non seguono l'ipotesi di normalità, si è deciso di non rimuoverli dal dataset tenendo presente che non si stanno rispettando le assunzioni dei modelli che vogliamo utilizzare. In aggiunta, dal grafico si può notare che le features con una distribuizione simile ad una normale non sono standardizzate, questa affermazione viene anche confermata dal calcolo delle statistiche descrittive mostrate nella tabella 2.1.

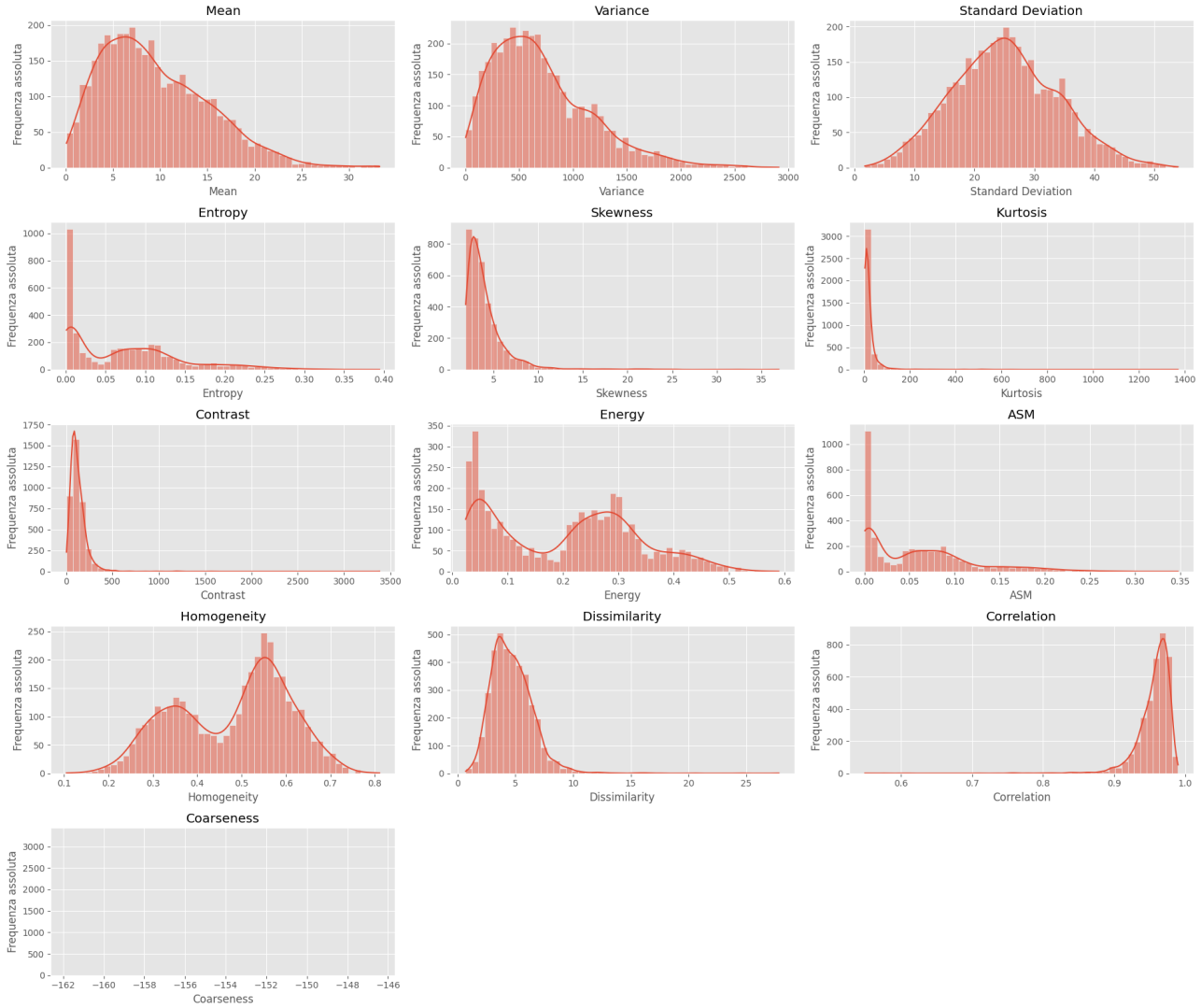


Figura 2.2: Barplot delle features

Di conseguenza sarà opportuno standardizzare le features per rispettare le assunzioni di SVM e della rete neurale. Per quanto riguarda Gaussian Naive Bayes non è necessario effettuare l'operazione sopracitata, dal momento che nel calcolo della probabilità si sfrutta la formula della Gaussiana che standardizza autonomamente.

Dal calcolo delle statistiche descrittive si può osservare che la feature *Coarseness* assume un valore quasi insignificativo tendente a 0, quindi è stato pensato di convertire questa feature ad una scala logaritmica, permettendo di aumentare la significatività dei valori. Nonostante questa trasformazione, la feature presenta una deviazione standard nulla quindi questo suggerisce la sua esclusione perché sicuramente non sarà la feature discriminante.

Per la fase di analisi risulta cruciale effettuare uno studio sulla potenzialità di discriminazione dei dati. Per fare ciò sono stati prodotti un totale di 13 grafici, uno per ogni feature, ciascuno composto da due boxplot rappresentanti i percentili delle feature separati per le classi 0 e 1. I grafici sono visibili nella figura 2.3.

	Mean	Variance	Standard Deviation	Entropy	Skewness	Kurtosis
count	3699	3699	3699	3699	3699	3699
mean	9.473354	710.895793	25.174138	0.072940	4.108362	24.422551
std	5.732700	468.154274	8.785183	0.069914	2.559163	56.292660
min	0.078659	3.145628	1.773592	0.000882	1.886014	3.942402
25%	4.965988	362.568474	19.041231	0.006662	2.621447	7.265711
50%	8.468414	624.708056	24.994160	0.065681	3.422625	12.370334
75%	13.184586	967.036275	31.097207	0.112694	4.662941	22.760735
max	33.239975	2910.581879	53.949809	0.394539	36.931294	1371.640060

(a) Statistiche descrittive delle feature *Mean*, *Variance*, *Standard Deviation*, *Entropy*, *Skewness* e *Kurtosis*.

	Contrast	Energy	ASM	Homogeneity	Dissimilarity	Correlation	Coarseness
count	3699	3699	3699	3699	3699	3699	3699
mean	128.119746	0.203546	0.058080	0.478442	4.702774	0.955697	7.458341e-155
std	110.168137	0.129047	0.057973	0.127971	1.856688	0.026061	0.000000e+00
min	3.194733	0.024731	0.000612	0.105490	0.681121	0.549426	7.458341e-155
25%	72.057782	0.068793	0.004732	0.364279	3.413266	0.946879	7.458341e-155
50%	107.075103	0.223482	0.049944	0.511894	4.486111	0.961567	7.458341e-155
75%	161.199093	0.298110	0.088870	0.575239	5.725644	0.971315	7.458341e-155
max	3382.574163	0.589682	0.347725	0.810921	27.827751	0.989972	7.458341e-155

(b) Statistiche descrittive delle feature *Contrast*, *Energy*, *ASM*, *Homogeneity*, *Dissimilarity*, *Correlation* e *Coarseness*.

Tabella 2.1: Statistiche descrittive degli attributi

Dai boxplot si può esservare che nel dataset sono presenti notevoli outliers, in aggiunta le distribuzioni delle features separate per classi si sovrappongono quasi tutte, eccetto per *Entropy*, *Energy*, *ASM* e *Homogeneity*. Questo implica il fatto che potenzialmente sono le più discriminanti rispetto alle altre features. In aggiunta, i grafici confermano che la *Coarseness* è costante, quindi dal momento che non può essere un attributo discriminate si può rimuovere dal dataset.

Infine, è stato effettuato un confronto tra features estratte 2 a 2 per poter analizzare se le classi sono separabili linearmente considerando gruppi di due feature. In questo modo sono state calcolate tutte le cobinazioni di feature, per ogni combinazione è stato prodotto un grafico cartesiano e un'istanza sarà disegnata nel grafico mediante un punto. Il punto esprime 2 informazioni:

- il colore del punto specifica la classe dell'istanza
- le coordinate saranno i valori delle due features considerate

In aggiunta, sono stati costruiti due grafici per ogni combinazione, per identificare quante istanze di classi diverse si sofrappongo. Tutti grafici vengono mostrati nella figura 2.4

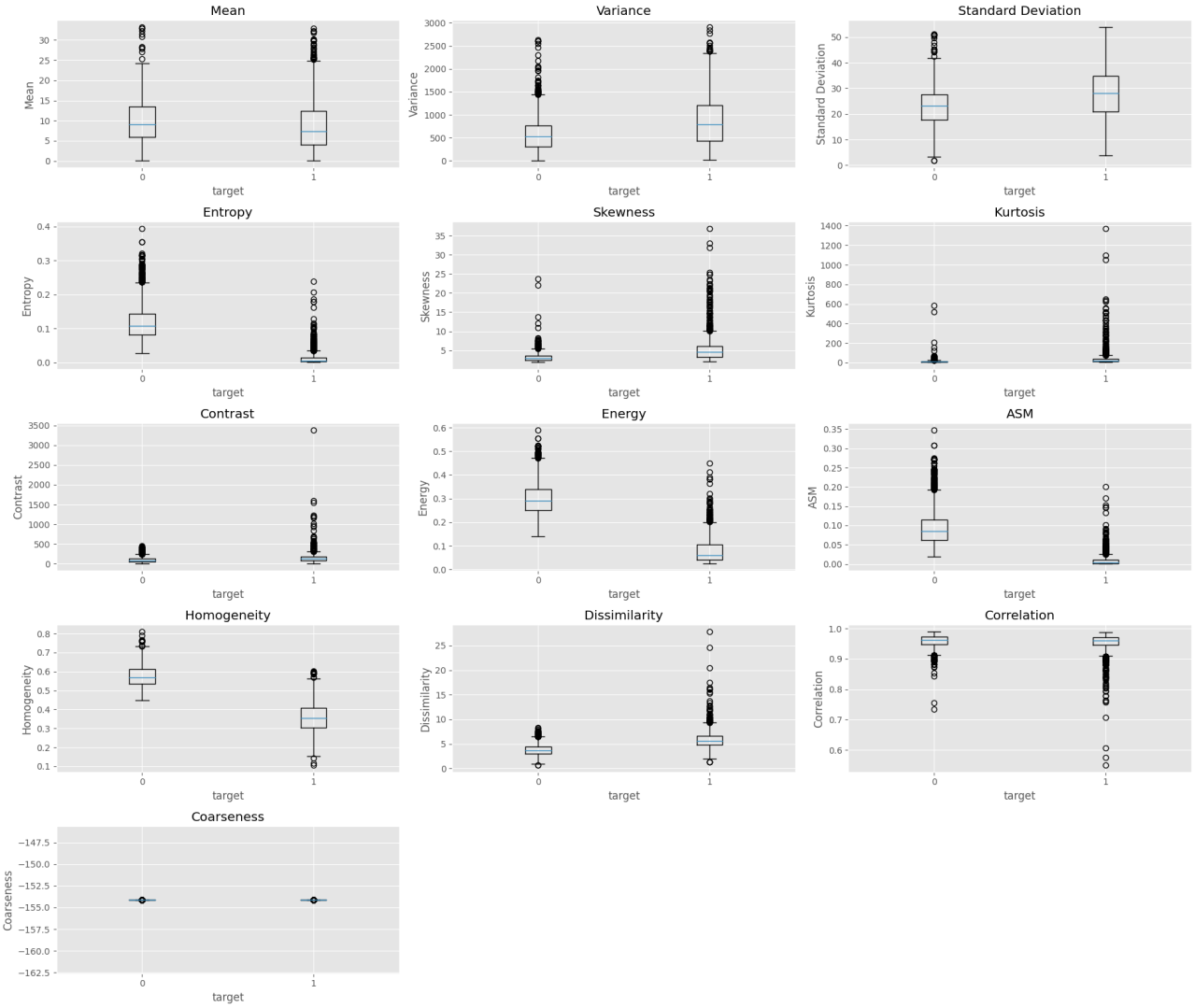


Figura 2.3: Barplot delle features

I grafici evidenziano il fatto che per ogni coppia di features si ha almeno una lieve sovrapposizione delle nuvole di punti rappresentanti le due classi, questo significa che in due dimensioni le classi non sono linearmente separabili a meno di accettare notevoli errori. In ogni caso, si pososno osservare le coppie con meno sovrapposizioni tra classi che in questo caso sono:

- *Entropy* e *Mean*
- *Skewness* e *Mean*
- *Skewness* e *Entropy*
- *Contrast* e *Entropy*
- *Correlation* e *Entropy*

Osservando queste coppie hanno un numero di istanze sovrapposte ridotto, allora si può affermare che le SVM con un buon kernel potrebbero ottenere degli ottimi risultati nella classi ficazione. Inoltre, questi grafici permettono anticipare dei primi studi sulla correlazione come la presenza di una correlazione logaritmica tra *Skewness* e *Mean*.

2.2.1 Analisi delle correlazioni

Il passaggio successivo è stato quello di analizzare le correlazioni tra le feature dal momento che un primo modo per ridurre la dimensionalità del dataset è attraverso il mantenimento di solo una feature tra tutte quelle correlate.

Perciò per prima cosa è stata prodotta una matrice di correlazione, riportata in figura 2.5, attraverso la quale è stato possibile osservare le correlazioni tra le feature.

Dall'analisi di questa matrice, si possono osservare diverse correlazioni tra le feature. Innanzitutto, si può notare una forte correlazione positiva tra le feature *Mean*, *Variance* e *Standard deviation*. Questa correlazione è facilmente spiegabile analizzando le immagini prodotte dalle risonanze magnetiche. Infatti, essendo in bianco e nero, se la media tende a 1 (colore bianco) allora la varianza e la deviazione standard aumentano, perché sono presenti diversi pixel bianchi. Questo comporta che le transizioni dal nero assoluto al bianco assoluto necessitano di regioni di pixel maggiore rispetto ad una transizione tra nero assoluto e grigio (0.5).

Invece, la correlazione tra varianza è deviazione standard facilmente spiegabile perché la deviazione standard è la radice quadrata della varianza, quindi sono misure dipendenti.

Una seconda forte correlazione positiva si può osservare tra le feature che misurano l'**uniformità dei livelli di grigio** dei pixel, più precisamente tra le feature *Entropy*, *ASM*, *Homogeneity* ed *Energy*. Queste feature quantificano delle informazioni legate alla texture dell'immagine, quindi la forte correlazione positiva può essere spiegata analizzando le texture delle immagini su cui vengono calcolate. Più precisamente se si ha un valore molto alto della feature *Entropy*, significa che la texture non è uniforme, ovvero si hanno strutture complesse e irregolari, quindi più uniforme sarà la distribuzione dei livelli di grigio, aumentando l'indice di *ASM*, comportando di conseguenza un aumento delle variazioni di intensità dei livelli di grigio, aumentando di conseguenza anche l'indice di *Energy* e *Homogeneity*.

Al tempo stesso, la matrice di correlazione evidenza una forte correlazione positiva tra gli indici che misurano la **morfologia della distribuzione dei livelli di grigio**, ovvero le feature di *Skewness* e *Kurtosis*. Questa dipendenza implica il fatto che più la distribuzione è leptokurtica (Kurtosis grande), ovvero la frequenza dei livelli di grigio dei pixel si concentrano interamente vicino alla media/mediana/ moda, allora più grande sarà la Skewness, ovvero maggiore sarà la tendenza ad avere frequenze di livelli di grigio più vicino al bianco (coda di destra più altra rispetto alla coda di sinistra).

La matrice della correlazione evidenzia anche una correlazione positiva tra le feature di *Contrast* e *Dissimilarity*, ovvero maggiore sarà il contrasto e maggiore sarà la complessità della texture.

In aggiunta dalla matrice si evidenza che le features di *Dissimilarity* e *Homogeneity* sono correlate negativo, dal momento che misurano una dissimilarità tra i livelli di grigio delle regioni e l'altra misura la loro l'omogeneità.

Dalla correlazione delle features è possibile ridurre la dimensionalità del dataset considerando sono le seguenti features:

- **Mean**
- **Entropy**
- **Skewness**
- **Contrast**
- **Correlation**

A puro scopo didattico è stato pensato di eseguire i modelli non solo sul dataset semplificato eliminando le correlazioni, ma anche applicando l'algoritmo PCA.

2.3 PCA

Precedentemente è stato presentato un primo modo per ridurre la dimensionalità dei dati basandoci sull'analisi delle correlazioni. In seguito, è stato pensato di provare ad utilizzare un metodo di trasformazione delle feature per ridurre la loro dimensionalità e successivamente analizzare i risultati ottenuti. La scelta sul metodo da utilizzare è ricaduta su PCA.

Prima di applicare la PCA, è stato necessario standardizzare le feature, questa operazione è stata fatta per evitare che le feature con varianza maggiore abbiano un peso maggiore rispetto alle altre. Senza standardizzare le feature, la PCA potrebbe non essere in grado di trovare le direzioni di massima varianza.

La prima parte dell'analisi è stata quella di trovare il corretto numero di componenti da utilizzare per la PCA. Questo è stato fatto attraverso l'osservazione della percentuale di varianza spiegata per ogni componente.

Per svolgere questa operazione sono state utilizzate solamente le feature numeriche del dataset, quindi sono state escluse le colonne *Image* e *Class*.

Rimosse le colonne non necessarie, è stato possibile computare la PCA utilizzando la libreria *sklearn* e successivamente è stato possibile osservare la percentuale di varianza spiegata per ogni componente, riportata in figura 2.6.

Dall'analisi della percentuale di varianza spiegata per ogni componente, si può osservare che le prime 3 componenti spiegano circa l'85% della varianza dei dati. Questo ci ha permesso di ridurre la dimensionalità del dataset a soli 3 attributi, permettendo di rappresentare i dati in uno spazio a 3 dimensioni.

Dalla figura 2.7 si può osservare che i dati ottenuti dalla PCA sembrano essere separabili con un piano.

2.4 Preparazione dei dati

Terminata la fase di analisi dei dati, si è passati alla fase di preparazione dei dati per l'addestramento dei modelli.

Partendo dal dataset utilizzato per l'analisi, si è proceduto con la derivazione da esso di due dataset distinti: uno ottenuto tramite analisi esplorativa e attraverso lo studio delle correlazioni tra gli attributi e l'altro ottenuto tramite Principal Component Analysis (PCA).

A livello implementativo, i due dataset sono stati rinominati nel seguente modo:

- `dataset_corr`: per il dataset ottenuto tramite analisi esplorativa e studio delle correlazioni;
- `dataset_pca`: per il dataset ottenuto tramite PCA.
- `dataset`: per il dataset originale.

La scelta di utilizzare due dataset distinti è stata fatta per confrontare i due approcci e valutare quale dei due fosse più adatto per l'addestramento dei modelli.

Ottenuti i due dataset, si è proceduto con la suddivisione di ciascuno di essi in due sottoinsiemi: uno per l'addestramento dei modelli e l'altro per la validazione dei modelli. Nello specifico, si è scelto di suddividere i dataset in modo tale da avere l'80% delle istanze per l'addestramento e il 20% per il test.

La parte di dati dedicata all'addestramento dei modelli è stata utilizzata anche per una fase di *cross-validation* per la scelta dei parametri migliori per la rete neurale e per le SVM. Per quanto riguarda la fase di ricerca degli iperparametri migliori, si è scelto di utilizzare una k-fold cross-validation con k=5.

La suddivisione dei dataset in training e test set è stata effettuata in modo strutturato, ovvero mantenendo la stessa proporzione di istanze per ciascuna classe in entrambi i set. Questo è stato fatto per evitare che i modelli addestrati fossero influenzati da una distribuzione sbilanciata delle classi.

Infine, per la rete neurale e per la SVM, si è proceduto con la standardizzazione dei dati, definendo due nuove varianti dei dataset `dataset_corr` e `dataset_pca`:

- `dataset_corr_std`: dataset `dataset_corr` standardizzato;
- `dataset_pca_std`: dataset `dataset_pca` standardizzato.

La standardizzazione dei dati è stata effettuata in quanto la rete neurale e la SVM sono modelli che possono essere influenzati dalla distribuzione dei dati.

Nella tabella 2.2 è presentare un breve riassunto di come sono stati preparati i dati per l'addestramento dei modelli e quale dataset è stato utilizzato per ciascun modello.

Nome del dataset	Operazioni applicate	Utilizzato per i seguenti modelli
dataset_corr	Riduzione della dimensionalità utilizzando l'analisi della correlazione	GNB
dateset_corr_std	dataset_corr con la standardizzazione dei dati	SVM e NN
dateset_pca	dataset_corr_std applicando l'algoritmo PCA	GNB
dateset_pca_std	dataset_pca con la standardizzazione dei dati	SVM e NN

Tabella 2.2: Riassunto delle operazioni effettuate sui dataset e utilizzo dei dataset per i modelli.

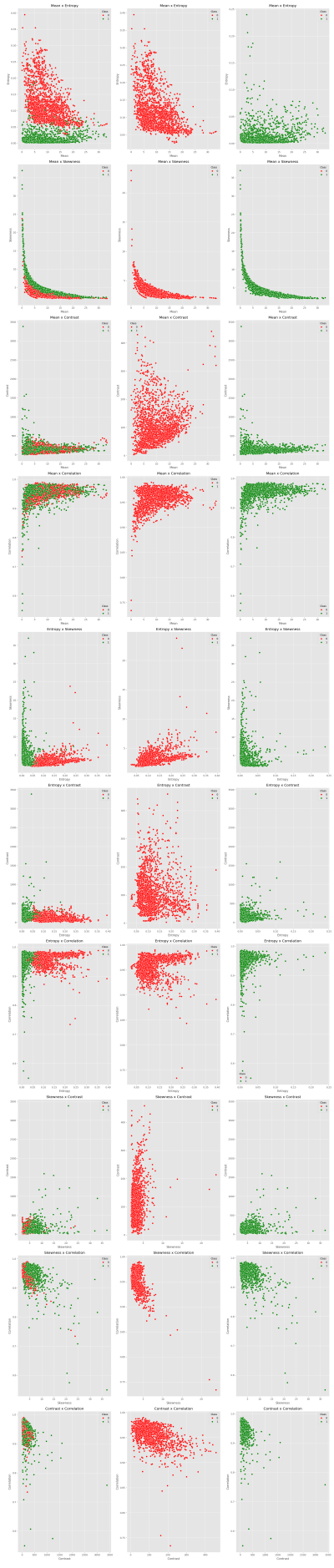


Figura 2.4: Scatterplot di tutte le combinazioni di features

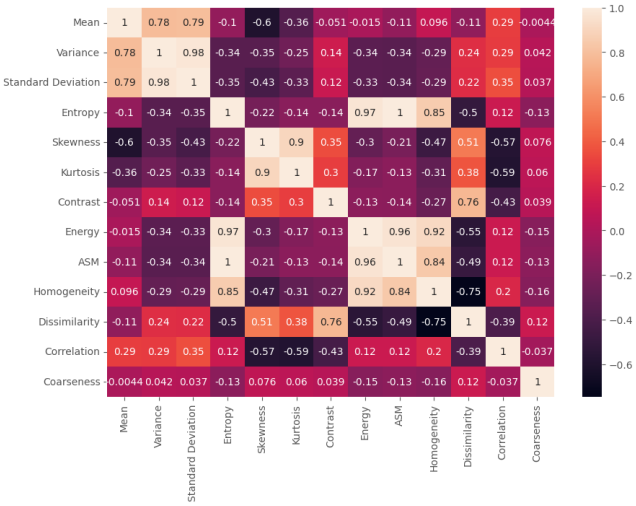


Figura 2.5: Matrice di correlazione

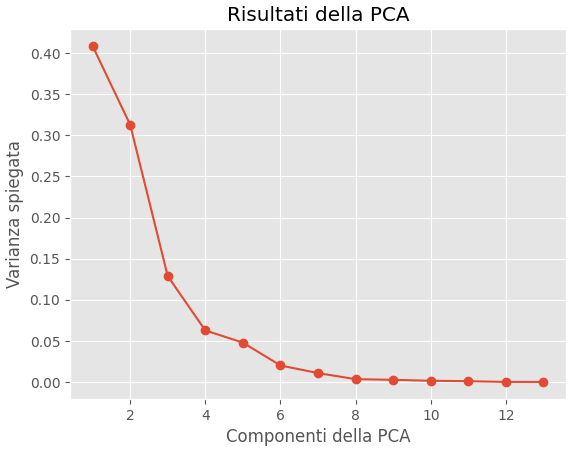


Figura 2.6: Percentuale di varianza spiegata per ogni componente

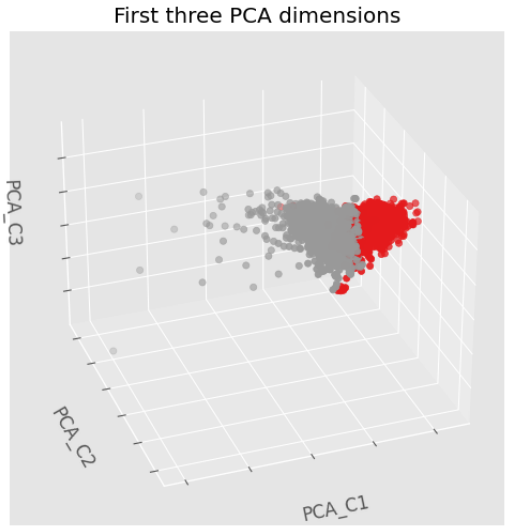


Figura 2.7: Scatter plot a 3 dimensioni

Capitolo 3

Rete neurale

La fase di analisi presentata nella sezione 2.2 ha permesso di acquisire informazioni utili sulla struttura del dataset e di conseguenza permettere la selezione di un modello adatto a svolgere il compito di classificazione.

In questo capitolo verrà presentato uno dei tre modelli che sono stati realizzati per svolgere il compito di classificazione, ovvero la **rete neurale**. Nello specifico, in questo capitolo si andranno a presentare i passaggi che sono stati effettuati per la realizzazione di questo modello, partendo dalla preparazione dei dati, passando per la definizione della struttura della rete neurale, fino ad arrivare ai risultati ottenuti. In un secondo momento, si andranno a confrontare i risultati ottenuti con quelli ottenuti da un modello addestrato con PCA.

3.1 Preparazione dei dati

Prima di passare alla presentazione della rete neurale nel dettaglio, è importante delineare il processo di preparazione dei dati necessario per l'addestramento della stessa.

La preparazione dei dati è stata eseguita attraverso una serie di operazioni mirate a renderli idonei per l'addestramento della rete neurale. Le fasi principali sono state le seguenti:

- **Standardizzazione dei dati**: Ogni caratteristica è stata trasformata in modo tale che la loro media fosse pari a 0 e la deviazione standard fosse 1. Questo passaggio è stato cruciale per garantire che la rete neurale non fosse influenzata da valori di input su differenti scale.
- **Suddivisione del dataset in training set e test set**: Il dataset è stato diviso in due parti: il training set e il test set. Questo processo è stato fondamentale per valutare le prestazioni della rete neurale su dati non utilizzati durante la fase di addestramento. Il training set ha rappresentato l'80% dei dati, mentre il test set il restante 20%.

Essendo il dataset non perfettamente bilanciato (55% dati negativi e 45% dati positivi) l'operazione di suddivisione è stata effettuata in modo da mantenere la stessa percentuale di dati positivi e negativi in entrambi i set. Lo scopo di questa scelta è quello di evitare che il modello sia addestrata su un dataset sbilanciato e di conseguenza non sia in grado di generalizzare correttamente.

3.2 Struttura della rete neurale

La fase di definizione della struttura della rete neurale è stata effettuata attraverso una serie di passaggi. Inizialmente, è stata effettuata un'analisi dei dati in modo tale da selezionare un sottoinsieme di feature le quali sono state utilizzate come input della rete neurale. Questo sottoinsieme è stato selezionato in modo tale da garantire che la rete neurale fosse in grado di discriminare in modo efficace le due classi.

In seguito, è stata effettuata una fase di grid search per valutare la combinazione migliore di iperparametri per la rete neurale. Questa fase è stata effettuata attraverso una cross validation a 5 fold, prendendo in considerazione solamente i dati del training set.

Dai risultati ottenuti dalla fase di analisi e dal dominio del problema, si è scelto di utilizzare una rete con una struttura di dimensioni ridotte, in modo tale da ridurre le possibilità che la rete neurale soffra di overfitting.

Per svolgere il compito di classificazione si è scelto di utilizzare una rete neurale feedforward, la cui struttura, a meno del layer di input e di output, è stata definita attraverso il processo di grid search.

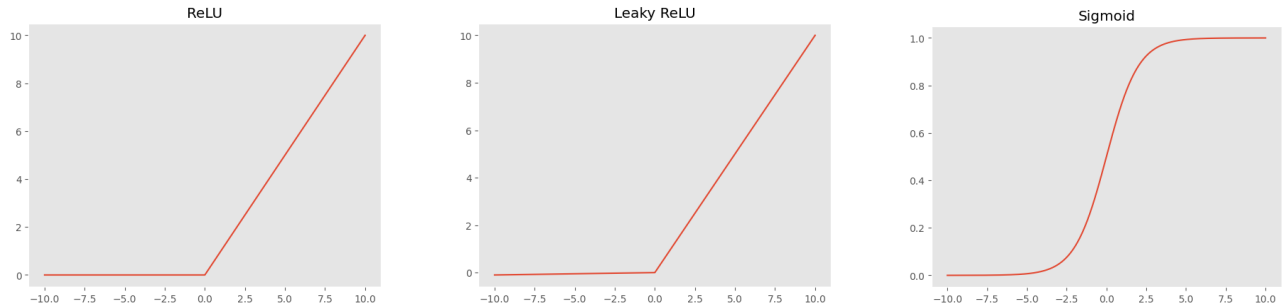
3.2.1 Ottimizzazione degli iperparametri

Come già accennato in precedenza, la ricerca degli iperparametri della rete neurale è stata effettuata attraverso un processo di grid search. Questo processo ha permesso di valutare le prestazioni della rete neurale al variare della funzione di attivazione, del numero di layer nascosti e del numero di neuroni per ogni layer nascosto.

Visti i risultati ottenuti nella fase di analisi e la volontà di mantenere i tempi di addestramento bassi, si è scelto di mantenere una struttura di dimensioni ridotte per la rete neurale. Per questo motivo, l'operazione di grid search è stata effettuata prendendo in considerazione un numero di neuroni per layer tra 5, 10 mentre il numero di layer nascosti è stato valutato tra 1 e 2.

Per quanto riguarda la funzione di attivazione, sono state valutate le seguenti funzioni di attivazione:

- *ReLU*
- *Leaky ReLU*
- *sigmoid*



(a) ReLU (b) Leaky ReLU (c) Sigmoide

Figura 3.1: Funzioni di attivazione utilizzate nella fase di grid search

Durante il processo di grid search, per ogni modello che è stato addestrato, sono state raccolte delle informazioni relative all'accuratezza, al tempo di addestramento richiesto. In aggiunta a queste informazioni, dato che ogni modello è stato addestrato attraverso una cross validation a 5 fold, sono stati calcolati gli intervalli di confidenza al 90% per ogni modello addestrato.

Ottenuti i risultati, si è proceduto con l'analisi di questi, in modo tale da definire la struttura della rete neurale. Per effettuare questa valutazione sono state utilizzate le misure precedentemente citate.

Il modello selezionato è stato scelto in base al seguente criterio:

$$Modello = 2 * Accuratezza + 2 * Tempo\ di\ addestramento + 1 * Intervalli\ di\ confidenza$$

Le misure di accuratezza e tempo di addestramento si riferiscono alla media calcolata attraverso la cross validation.

Nello specifico, sono stati utilizzati i seguenti pesi: 2 per l'accuratezza media, 2 per il tempo di addestramento medio e 1 per gli intervalli di confidenza. Questi pesi sono stati scelti in modo tale da dare più importanza all'accuratezza media e al tempo di addestramento medio, in quanto sono le due misure che permettono di valutare le prestazioni della rete neurale, mentre gli intervalli di confidenza sono stati utilizzati per valutare la variabilità delle prestazioni.

Per verificare la validità del modello scelto si è proceduto con il confronto di esso con la rete che ha ottenuto la migliore accuratezza e quella che ha ottenuto il tempo di addestramento minore, ottenendo i risultati riportati in tabella 3.1.

Modello	Accuratezza	Tempo di addestramento
Tempo di addestramento minore	97.9%	1.05s
Accuratezza maggiore	99.0%	14.43s
Modello scelto	98.6%	2.59s

Tabella 3.1: Risultati ottenuti dalla fase di grid search

Dai valori riportati nella tabella 3.1 si può notare che il notare che il modello che è stato selezionato fornisce un compromesso tra accuratezza e tempo di addestramento. Nello specifico, perdendo lo 0.4% di accuratezza si è ottenuto un tempo di addestramento minore di circa 12 secondi.

3.2.2 Definizione della struttura della rete neurale

Dalla fase di analisi è stato selezionato un sottoinsieme di feature le quali sono state utilizzate come input della rete neurale. Questo sottoinsieme è composto da 5 elementi, il che ha permesso di definire la struttura del layer di input della rete neurale, questo primo strato è composto da 5 neuroni, uno per ogni feature selezionata.

I risultati ottenuti dalla fase di grid search hanno permesso di definire la struttura della rete neurale. In particolare, la rete neurale è composta da 1 layer di input, 2 layer nascosti e 1 layer di output.

I layer nascosti sono composti nel seguente modo:

- Il primo layer nascosto è composto da 10 neuroni, in cui la funzione di attivazione è la funzione ReLU 3.1a.
- Il secondo layer nascosto è composto da 5 neuroni, in cui la funzione di attivazione è la funzione ReLU 3.1a.

Per concludere la descrizione della struttura della rete neurale, è necessario specificare come è composto l'ultimo layer, ovvero quello di output. Vista la natura del problema di classificazione, il layer di output è composto da un solo neurone, in cui la funzione di attivazione è la funzione sigmoide 3.1c.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.1}$$

Questa scelta è dovuta al fatto che tale funzione restituisce un valore compreso tra 0 e 1, il che permette di interpretare l'output della rete neurale come la probabilità che l'input appartenga alla classe positiva.

La struttura della rete neurale è riassunta nella figura 3.2.

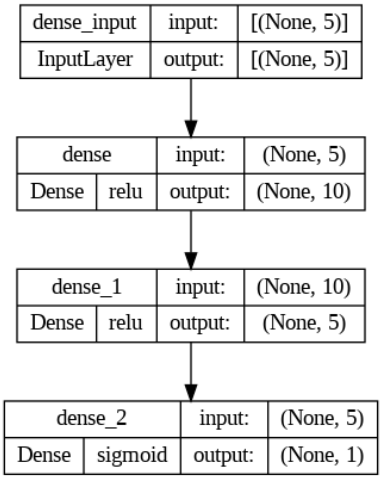


Figura 3.2: Struttura della rete neurale

3.2.3 Altri iperparametri

Oltre alla ricerca della struttura della rete neurale, la fase di grid search è stata utilizzata per valutare l'algoritmo di ottimizzazione, il numero di epoche e la dimensione del batch.

Per quanto riguarda l'algoritmo di ottimizzazione, il confronto è stato eseguito tra *Adam* e *SGD*, mentre per il numero di epoche e la dimensione del batch sono stati valutati i valori 100, 300 per il numero di epoche e 50, 100, 300 per la dimensione del batch.

I risultati ottenuti dalla fase di grid search hanno permesso di definire i valori degli iperparametri che hanno permesso di ottenere i migliori risultati. In particolare, l'algoritmo di ottimizzazione scelto è *Adam*, mentre il numero di epoche e la dimensione del batch sono stati impostati a 100 e 100 rispettivamente.

In questa fase è stato necessario definire la funzione di perdita. Si è scelta la *binary crossentropy* in quanto adatta a problemi di classificazione binaria. La scelta di questa loss è dovuta alla natura del problema di classificazione che si vuole risolvere.

3.3 Addestramento della rete neurale

La fase di addestramento della rete neurale è stata effettuata utilizzando il training set precedentemente definito. L'addestramento della rete neurale è stato effettuato utilizzando la libreria *Keras* in quanto permette di definire e addestrare reti neurali in modo intuitivo.

3.4 Risultati

Vista il dominio del problema, ovvero la classificazione di dati medici, si è deciso di modificare manualmente il valore della soglia per la predizione del tumore. Questa scelta è stata fatta in quanto si è voluto ridurre al minimo il numero di falsi negativi, ovvero il numero di casi in cui il modello predice l'assenza di tumore quando in realtà è presente.

Per realizzare questa operazione si è scelto di impostare il valore di threshold a 0.3, in modo tale da ridurre il numero di falsi negativi. Questa scelta è stata fatta in quanto si è voluto dare più importanza al valore di richiamo, il quale permette di valutare la capacità del modello di individuare i veri positivi.

Fatta questa precisazione, si può procedere con la presentazione dei risultati ottenuti. Utilizzando i dati del test set, è stato possibile valutare le prestazioni della rete neurale addestrata. In particolare, sono state calcolate le seguenti metriche:

- Accuratezza
- Precisione
- Richiamo
- F1 score

Oltre al calcolo di queste metriche, si è deciso di realizzare la curva ROC per il modello e di rappresentare la matrice di confusione. Nella tabella 3.2 sono presentati i risultati ottenuti dal modello addestrato.

Metrica	**Accuratezza**	**Precisione**	**Richiamo**	**F1 score**
Valore	98.93 %	98.52 %	99.10 %	98.81 %

Tabella 3.2: Risultati ottenuti dal modello addestrato

Dai valori riportati nella tabella 3.2 si può notare che la rete neurale ha ottenuto dei valori delle metriche molto alti. Questo comportamento è giustificato dal fatto che in fase di analisi è stato possibile notare che le feature selezionate sono in grado di discriminare in modo efficace le due classi.

In aggiunta al calcolo di queste metriche, è stata calcolata la matrice di confusione per il modello addestrato. La matrice di confusione ottenuta è riportata in figura 3.3.

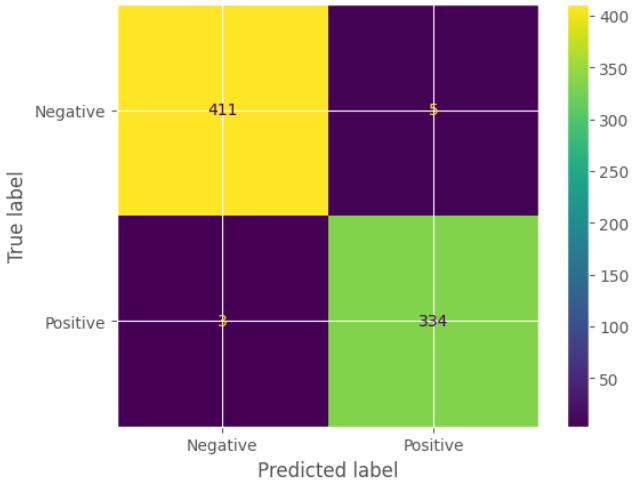


Figura 3.3: Matrice di confusione ottenuta dal modello addestrato

Dalla matrice di confusione è possibile confermare i risultati ottenuti dalle metriche calcolate in precedenza. Inoltre, avendo corretto manualmente il valore della soglia, si è riusciti a ridurre il numero di falsi negativi, il che ha permesso di aumentare il valore del richiamo.

Per concludere questa prima parte di analisi dei risultati, è stata realizzata la curva ROC per il modello addestrato. La curva ROC ottenuta è riportata in figura 3.4. Oltre alla curva ROC è stata calcolata l'area sotto la curva, la quale ha ottenuto un valore di 1.00. Questo valore ci permette di affermare che il modello addestrato si avvicina molto alla perfetta classificazione.

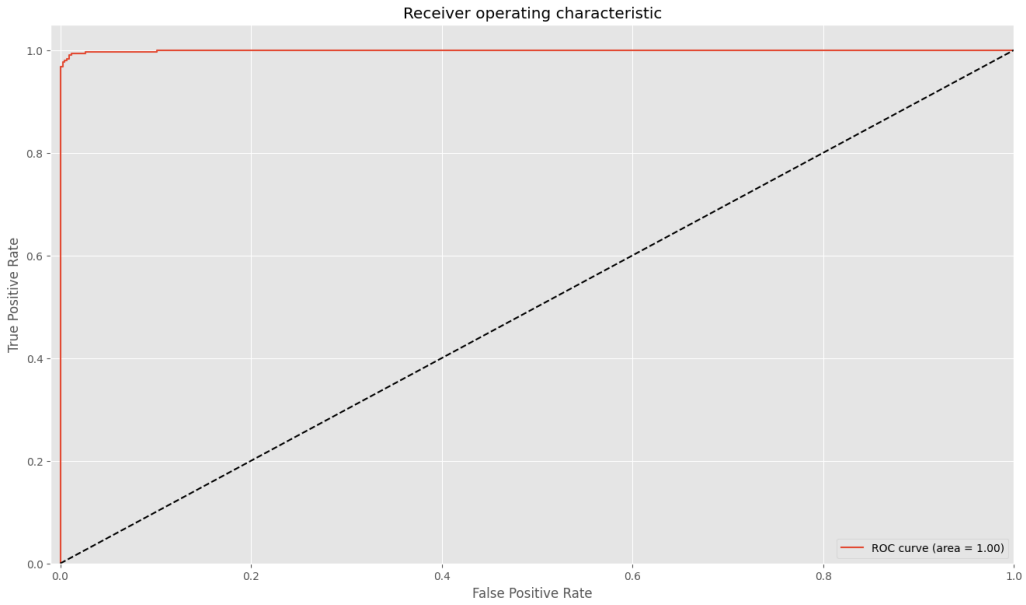


Figura 3.4: Curva ROC ottenuta dal modello addestrato

K-fold validation

Per avere una visione più chiara dei risultati ottenuti, si è deciso di effettuare una valutazione del modello attraverso 10 fold di cross validation. In questo processo ogni modello che è stato addestrato è stato valutato attraverso le metriche di accuratezza, precisione, richiamo e F1 score.

Per svolgere questa operazione è stato utilizzato il dataset completo, ovvero senza alcuna suddivisione in training set e test set.

Anche per questa operazione è stato utilizzato il valore di threshold precedentemente definito, ovvero 0.3.

Sui risultati ottenuti da questo processo sono stati calcolati gli intervalli di confidenza al 90%. I risultati ottenuti sono riportati in tabella 3.3.

Metrica	Valore Medio	Intervallo di confidenza
Accuratezza	98.27 %	[97.98%, 98.55%]
Precisione	97.99 %	[97.47%, 98.52%]
Richiamo	98.15 %	[97.49%, 98.81%]
F1 score	98.06 %	[97.75%, 98.38%]

Tabella 3.3: Risultati ottenuti dalla cross validation

Gli intervalli ottenuti sono stati successivamente rappresentati in un grafico riportato in figura 3.5. Questo grafico permette di avere una visione più chiara dei risultati ottenuti dalla cross validation.

3.5 Modello addestrato con PCA

Per verificare se i risultati ottenuti dal modello addestrato sulle feature da noi selezionate siano effettivamente dovuti alla struttura delle feature e non a una fortunata selezione, si è deciso di addestrare un modello con le feature ottenute attraverso la PCA.

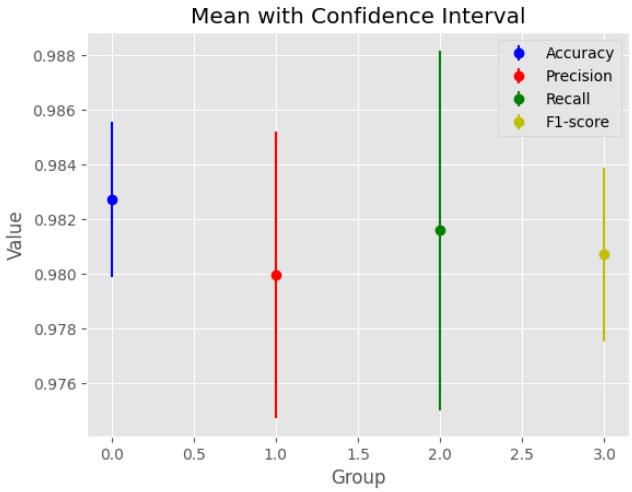


Figura 3.5: Risultati ottenuti dalla cross validation

Il dataset ottenuto attraverso la PCA, descritto sella sezione 2.3, è stato diviso in training set e test set in modo tale da mantenere la stessa percentuale di dati positivi e negativi in entrambi i set. Oltre a questa operazione, i dati sono stati standardizzati.

3.5.1 Struttura

Come per il modello addestrato con le feature selezionate manualmente, anche per questo modello è stata effettuata una fase di grid search per valutare la combinazione migliore di iperparametri per la rete neurale.

Il processo utilizzato in questa fase è analogo a quello utilizzato per il modello precedente, sia a livello di iperparametri che di valutazione del modello.

Come fatto in precedenza, il modello selezionato è stato confrontato con il modello che ha ottenuto la migliore accuratezza e quello che ha ottenuto il tempo di addestramento minore. I risultati ottenuti sono riportati in tabella 3.4.

Modello	Accuratezza	Tempo di addestramento
Tempo di addestramento minore	96.9%	1.06s
Accuratezza maggiore	98.0%	22.20s
Modello scelto	97.9%	1.16s

Tabella 3.4: Risultati ottenuti dalla fase di grid search

Anche in questo caso, come per il precedente, il modello che è stato selezionato rappresenta un compromesso tra accuratezza e tempo di addestramento. In particolare, perdendo lo 0.1% di accuratezza si è ottenuto un tempo di addestramento minore di circa 21 secondi.

I risultati ottenuti dalla fase di grid search hanno permesso di definire la struttura della rete neurale. In particolare, la rete neurale è composta da 1 layer di input, 1 layer nascosto e 1 layer di output.

Il layer di input è composto da 3 neuroni, uno per ogni componente principale ottenuta attraverso la PCA. Questo primo strato è stato definito in questo modo in quanto il dataset ottenuto attraverso la PCA è composto da 3 feature.

Il layer nascosto è composto da 10 neuroni, in cui la funzione di attivazione è la funzione ReLU 3.1a.

Il layer di output è lo stesso utilizzato per il modello addestrato con le feature selezionate manualmente, ovvero è composto da un solo neurone, in cui la funzione di attivazione è la funzione sigmoide 3.1c.

3.5.2 Risultati

Addestrato il modello con i dati ottenuti attraverso la PCA, si è proceduto con la valutazione delle prestazioni del modello utilizzando le stesse metriche utilizzate in precedenza e il test set.

I risultati ottenuti dal modello addestrato con la PCA sono riportati in tabella 3.5 e sono confrontati con quelli ottenuti dal modello addestrato con le feature selezionate manualmente.

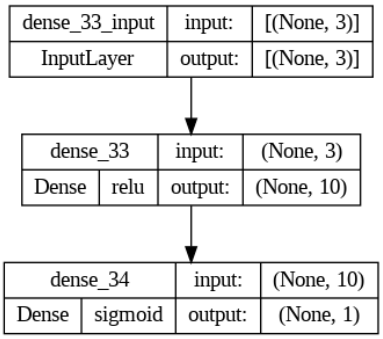


Figura 3.6: Struttura della rete neurale addestrata con PCA

Modello	Accuratezza	F1	Precision	Recall
Rete senza PCA	98.80%	98.65%	99.10%	98.65%
Rete con PCA	98.27%	98.07%	97.92%	98.21%

Tabella 3.5: Risultati ottenuti dai modelli addestrati con PCA

Dai valori riportati nella tabella 3.5 si può notare che il modello addestrato con le feature selezionate manualmente ha ottenuto dei risultati migliori rispetto a quello addestrato con la PCA.

È stata anche calcolata la matrice di confusione per il modello addestrato con la PCA. La matrice di confusione ottenuta è riportata in figura 3.7.

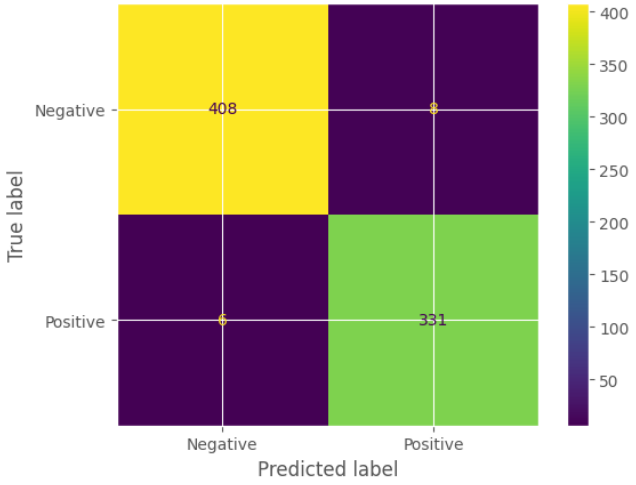


Figura 3.7: Matrice di confusione ottenuta dal modello addestrato con PCA

In aggiunta a queste metriche, i modelli sono stati confrontati attraverso la 10-fold cross validation. I risultati ottenuti sono riportati in tabella 3.6.

Oltre a confrontare le performance dei modelli utilizzando le metriche, si è deciso di confrontare le curve ROC dei due modelli. I risultati ottenuti sono riportati in figura 3.8.

Da questa figura si può notare che la differenza tra i due modelli è minima. Entrambi i modelli si avvicinano molto alla perfetta classificazione. Si può notare una differenza maggiore tra i due modelli osservando gli intervalli di confidenza. In particolare, si riesce a osservare una differenza di ampiezza degli intervalli di confidenza tra i due modelli.

Il modello ottenuto attraverso la PCA ha un intervallo di confidenza più ampio rispetto a quello ottenuto senza PCA, anche se il valore medio delle metriche è leggermente superiore nel caso di PCA.

Nella figura 3.9 sono riportati gli intervalli di confidenza ottenuti dai modelli addestrati con e senza PCA. Da questa figura si può notare che l'intervallo di confidenza ottenuto dal modello addestrato con PCA, rappresentato dalla linea di colore rosso, è più ampio rispetto a quello ottenuto dal modello addestrato senza PCA, rappresentato dalla linea di colore blu.

Metrica	Valore Medio	Intervallo di confidenza con PCA	Intervallo di confidenza senza PCA
Accuratezza	98.35 %	[97.97%, 98.72%]	[97.98%, 98.55%]
Precisione	98.05 %	[97.51%, 98.58%]	[97.47%, 98.52%]
Richiamo	98.27 %	[97.62%, 98.93%]	[97.49%, 98.81%]
F1 score	98.15 %	[97.73%, 98.58%]	[97.75%, 98.38%]

Tabella 3.6: Risultati ottenuti dalla cross validation

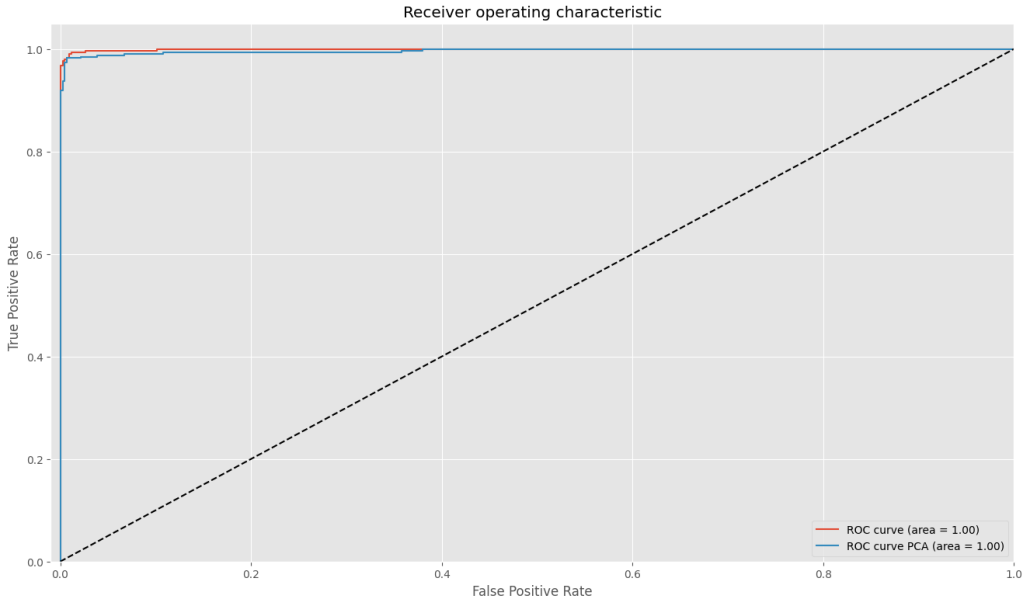
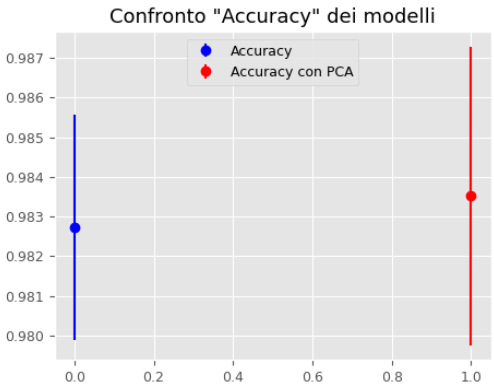
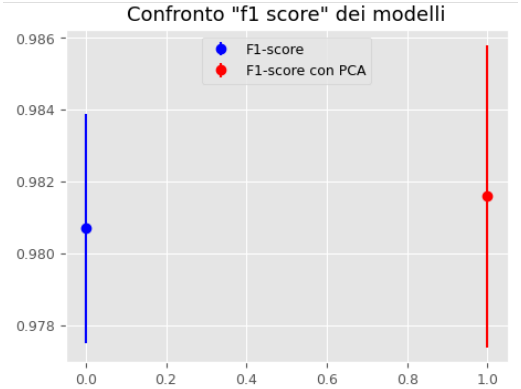


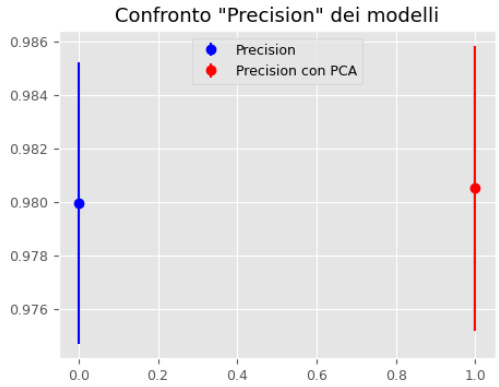
Figura 3.8: Confronto curve ROC tra il modello addestrato con e senza PCA



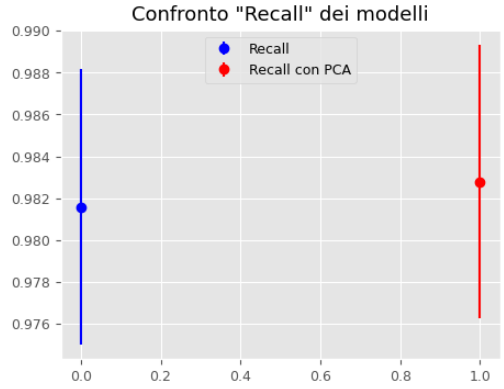
(a) Accuracy



(b) F1 score



(c) Precision



(d) Recall

Figura 3.9: Intervalli di confidenza ottenuti dai modelli addestrati con e senza PCA

Capitolo 4

Gaussian Naive Bayes

La precedente fase di analisi ha permesso di acquisire informazioni utili sulla struttura del dataset e di conseguenza permettere la selezione di un modello adatto a svolgere il compito di classificazione.

In questo capitolo verranno presentati tutti i risultati ottenuti dall'apprendimento e dalle valutazioni effettuate sul modello Gaussian Naive Bayes. Da notare che si sta utilizzando Gaussian Naive Bayes pur sapendo che non tutte le features derivano da una distribuzione normale, siamo consci del fatto che non si stanno rispettando le assunzioni del modello.

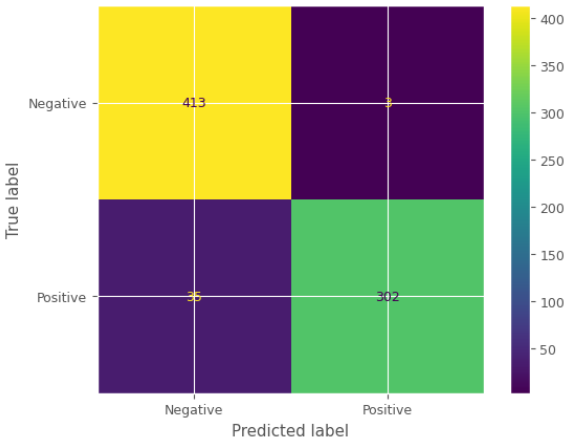
4.1 Addestramento di Gaussian Naive Bayes

Sono stati creati due modelli allenati rispettivamente sul training set di Dateset-corr e Dateset-pca. In aggiunta, non è stato effettuato nessuno studio degli iperparametri dal momento Gaussian Naive Bayes non possiede degli iperparametri che devono essere stimati.

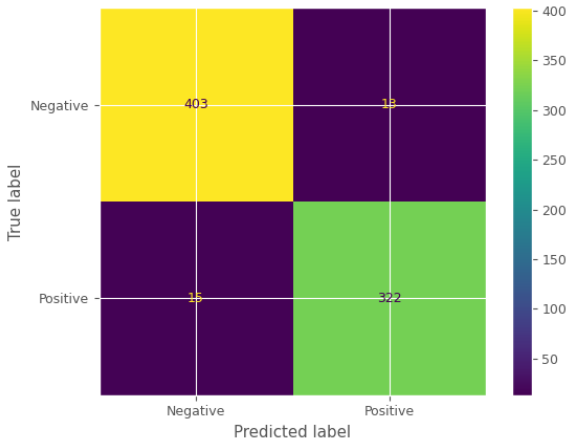
4.2 Risultati

Dati i due modelli addestrati precedentemente, sono state effettuate le previsioni sui test set dei rispettivi dataset utilizzati nell'apprendimento e, infine, sono state calcolate le metriche di valutazione.

Per prima cosa è stata calcolata la matrice di confusione per ciascun modello, visibile nella figura 4.1.



(a) Gaussian Naive Bayes allenato su Dataset-corr



(b) Gaussian Naive Bayes allenato su Dataset-pca

Figura 4.1: matrici di confusione per Gaussian Naive Bayes

Le matrici di confusione evidenziano come i modelli generalizzano molto bene, più precisamente il modello allenato su Dataset-corr ha un'accuracy del 95%, al contrario, quello allenato su Dataset-pca che raggiunge un'accuracy del 96%. Questo denota che ridurre le dimensioni del dataset utilizzando pca non porta a significativi miglioramenti sulle predizioni. Questo mancato miglioramento può essere dovuto al fatto che si è già trovata una buona ipotesi vicina alla funzione generatrice del dataset.

Dalle matrici di confusione oltre all'accuracy si possono calcolare le metriche di precision, recall e F1-score, i loro valori sono visionabili nella tabella 4.1.

Metrica	Valore sul training set di Dataset-corr	Valore sul training set di Dataset-pca
Accuracy	95%	96%
Precision	90%	96%
Recall	99%	96%
F1-score	94%	96%

Tabella 4.1: Metriche Gaussian Naive Bayes

Dalle metriche di valutazione si può notare come su Dataset-corr si ha una precision minore rispetto alla recall, quindi significa che si tende ad associare la presenza di un tumore anche quando non è presente. Al contrario su Dataset-pca aumenta la precision diminuendo la recall, questo significa che sarà più affidabile su un riscontro positivo rispetto ad un riscontro negativo. In aggiunta, si può notare come F1-score è migliore nel modello allenato su Dataset-pca, questo potrebbe suggerire che, utilizzando pca, miglira la qualità delle predizioni. Il problema è che la metrica in questione calcola la media armonica tra precision e recall, ma nel dominio applicativo che si sta considerando, la recall assume maggior importanza dal momento che l'obiettivo è eliminare il falsi negativi e ammettere dei falsi positivi, perciò si dovrà massimizzare la recall.

Oltre alle metriche di valutazione sono stati prodotti i grafici delle curve ROC di entrami i modelli, mostrati in figura 4.2

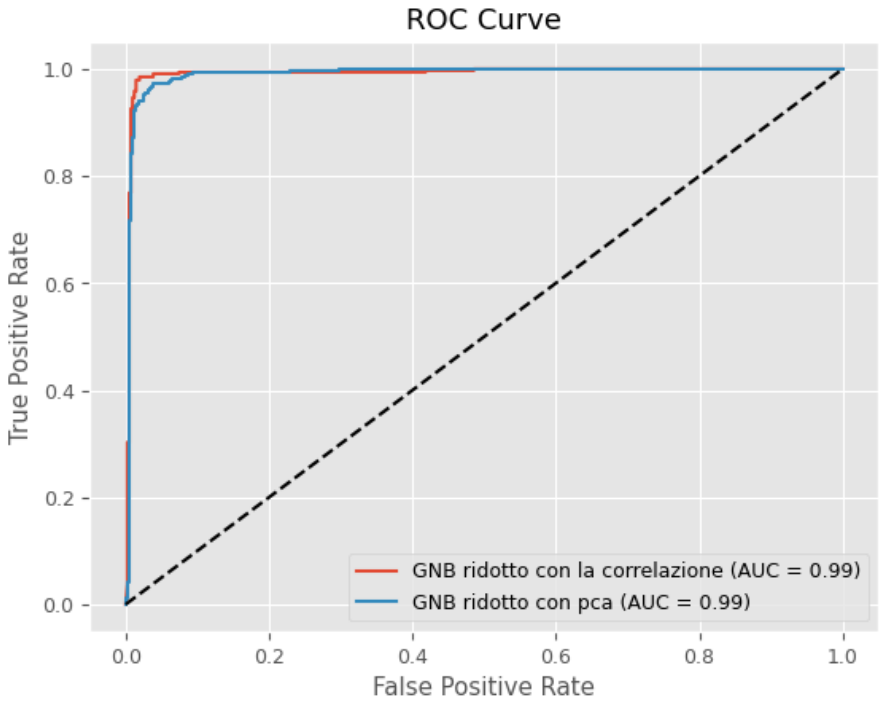


Figura 4.2: curve ROC

Le curve ROC permettono di confrontare i due modelli sfruttando la metrica AUC che coincide con la misura dell'area sottesa alla curva del modello. Come si può notare i due modelli hanno delle differenze non molto significative, infatti entrambi hanno un'area AUC uguale, questo lascia intendere che siano identici i due modelli. In realtà, AUC non considera la diversa importanza delle classi, infatti, come anticipato precedentemente, nel dominio in questione un errore di falso positivo ha un peso minore rispetto ad un faso negativo e questo criterio di valutazione non viene considerato dalla metrica.

Bisogna specificare che questi studi sono stati realizzati su modelli che sono stati allenati su un dataset di medie dimensioni ($\leq$ 10000 esempi), perciò è stato pensato di effettuare una valutazione dei modelli utilizzando la 10-fold stratified cross validation per poi ottenere gli intervali di confidenza delle metriche di valutazione (vedi figura 4.3).

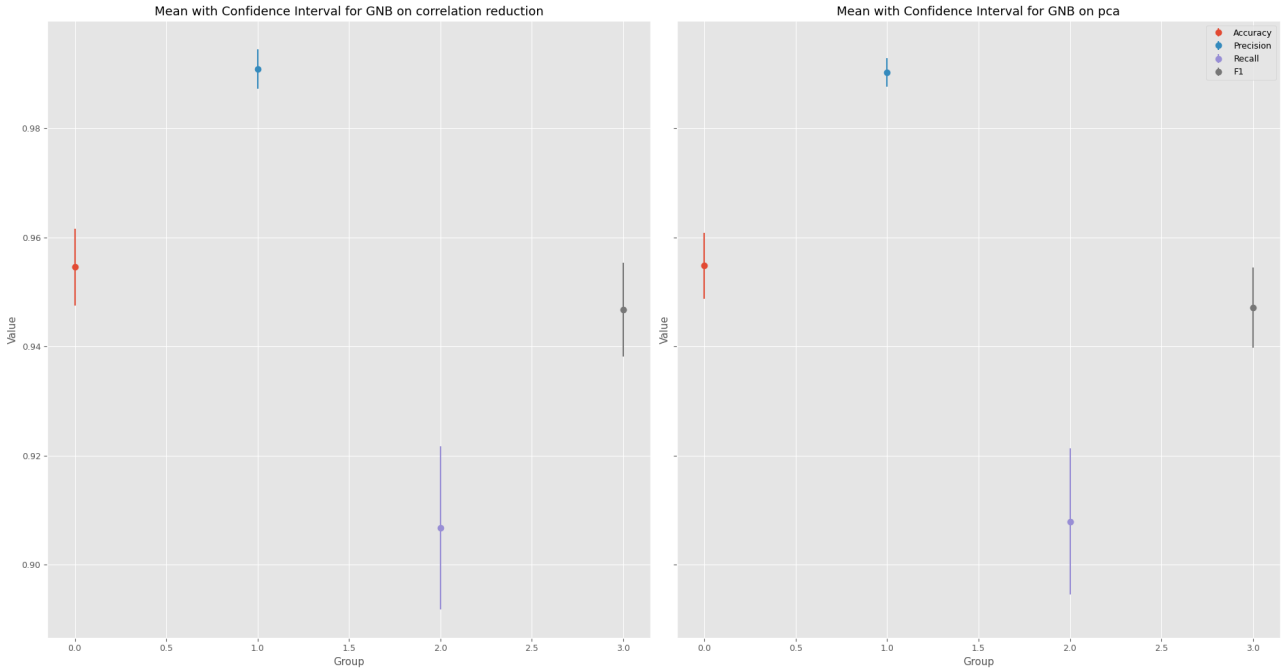


Figura 4.3: Intervalli di confidenza

Dagli intervalli di confidenza non ci sono significative differenze tra i due modelli, quindi i ragionamenti svolti precedemente sono confermati.

Bibliografia

[1] Namita Aggarwal e RK Agrawal. “First and second order statistics features for classification of magnetic resonance brain images”. In: (2012).