

Report del progetto di Machine Learning

Ferrario Tommaso Matr. 869005 (@TommasoFerrario18)

Terzi Telemaco Matr. 865981(@Tezze2001)

Vendramini Simone Matr. 866229(@Svendra4MySelf)

11 febbraio 2024

Indice

1	Introduzione	2
2	Dataset	3
2.1	Struttura del dataset	3
2.2	Analisi descrittiva	4
2.2.1	Analisi delle correlazioni	5
2.3	PCA	7
2.4	Preparazione dei dati per i vari modelli	8
3	Rete neurale	9
3.1	Preparazione dei dati	9
3.2	Struttura della rete neurale	9
3.3	Addestramento della rete neurale	11
3.4	Risultati	11
3.5	Modello addestrato con PCA	13
4	Gaussian Naive Bayes	16
4.1	Addestramento di Gaussian Naive Bayes	16
4.2	Risultati	16

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.

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.
- **Naive Bayes Gaussiano:** è stato scelto questo modello dal momento che è l'unico ad essere probabilistico.
- **Rete neurale:** è stato scelto questo modello per confrontare i primi due con una soluzione neurale.

La relazione è stata suddivisa nei 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.
- **Naive Bayes Gaussiano:** descrizione e analisi delle performance per Naive Bayes Gaussian.
- **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 un set di 3762 immagini, ognuna delle quali è stata ottenuta dalla risonanza magnetica del cervello. Ad ogni immagine è stata associata un'etichetta la quale permette di rappresentare la presenza o meno di un tumore al cervello. Nel dataset è presente una colonna, *Class*, nella quale sono presenti le etichette, ovvero:

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

Oltre alla colonna *Class*, il dataset è composto da altre 13 colonne, nelle quali sono presenti le feature calcolate sulle immagini. Tali features sono state ottenute calcolando i **momenti Hu** sulle immagini della risonanza magnetica. I momenti Hu catturano le informazioni di base sull'immagine come l'area dell'oggetto, il centroide, l'orientamento e altre proprietà.

Gli attributi di cui è composto il dataset possono essere suddivisi in 2 gruppi[1]:

1. **First Order Features:** forniscono informazioni legate alla distribuzione dei livelli di grigio dell'immagine. Queste features corrispondono alle statistiche descrittive calcolate sui valori di ciascun pixel dell'immagine. Le statistiche descrittive calcolate sono:
 - **Media**
 - **Varianza**
 - **Deviazione standard**
 - **Indice di asimmetria**
 - **Indice di kurtosis**
2. **Second Order Features:** forniscono informazioni a livello di composizione della texture dell'immagine.
 - **Contrasto:** misura la variazione locale dei livelli di grigio dei pixel. Maggiore sarà il valore allora maggiore sarà il contrasto dell'immagine.
 - **Energia:** misura l'eterogeneità o la variazione dell'intensità dell'immagine. Più piccolo è il valore allora meno variazioni di intensità e più omogenea sarà la texture, al contrario, più la texture è irregolare allora maggiore sarà il valore.
 - **ASM:** misura quanto sono distribuiti uniformemente i livelli di grigio nell'immagine. Più i livelli di grigio sono uniformemente distribuiti minore sarà il valore dell'indice.
 - **Entropia:** misura la randomicità dei livelli di grigio, quindi più piccolo è il valore, più la texture sarà uniforme.
 - **Homogeneous:** misura secondaria del contrasto. Più alto sarà l'indice allora minore sarà il contrasto dell'immagine.
 - **Dissimilarity:** misura quanto spesso differenti combinazioni dei valori di intensità dei pixel occorrono nell'immagine. Un valore alto indica che l'immagine ha una maggior variazione delle intensità dei pixel vicini, quindi più complessa sarà la texture.

- **Correlation:** misura la correlazione tra pixel nelle due diverse direzioni.
- **Coarseness:** misura quanto l'immagine è composta da regioni di intensità omogenea, ovvero quanto è granulare o fine la texture.

2.2 Analisi descrittiva

Il dataset è formato da un totale 3762 istanze, ciascuna delle quali è descritta da 15 attributi così suddivisi:

- 13 rappresentano le feature calcolate sulle immagini della risonanza magnetica. Tutti questi attributi sono di tipo *float*.
- 1, ovvero l'attributo *Image*, rappresenta l'identificativo dell'immagine a cui si riferisce l'istanza.
- 1, ovvero l'attributo *Class*, rappresenta l'etichetta associata all'immagine. Questo attributo è stato convertito da *intero* a *categorico* per poter effettuare la classificazione.

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.

Successivamente, si è proceduto con il calcolo delle statistiche descrittive per gli attributi numerici presenti nel dataset. Questa operazione ha permesso di ottenere le informazioni riportate nella tabella 2.1.

	Mean	Variance	Standard Deviation	Entropy	Skewness	Kurtosis
count	3762	3762	3762	3762	3762	3762
mean	9.488890	711.101063	25.182271	0.073603	4.102727	24.389071
std	5.728022	467.466896	8.773526	0.070269	2.560940	56.434747
min	0.078659	3.145628	1.773592	0.000882	1.886014	3.942402
25%	4.982395	363.225459	19.058475	0.006856	2.620203	7.252852
50%	8.477531	622.580417	24.951560	0.066628	3.422210	12.359088
75%	13.212723	966.954319	31.095889	0.113284	4.651737	22.640304
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	3762	3762	3762	3762	3762	3762	3762
mean	127.961459	0.204705	0.058632	0.479252	4.698498	0.955767	7.458341e-155
std	109.499601	0.129352	0.058300	0.127929	1.850173	0.026157	0.000000e+00
min	3.194733	0.024731	0.000612	0.105490	0.681121	0.549426	7.458341e-155
25%	72.125208	0.069617	0.004847	0.364973	3.412363	0.947138	7.458341e-155
50%	106.737418	0.225496	0.050849	0.512551	4.482404	0.961610	7.458341e-155
75%	161.059006	0.298901	0.089342	0.575557	5.723821	0.971355	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

Questa operazione ha permesso di ottenere alcune informazioni sul dataset. Innanzitutto, ha permesso di osservare che i valori associati agli attributi non sono standardizzati dal momento che nessun dato ha media 0 e deviazione standard 1.

La standardizzazione delle feature è stata fatta, utilizzando l'equazione 2.1, solo per le SVM e il modello neurale, mentre per Gaussian Bayes la standardizzazione viene fatta in automatico dalla libreria del modello.

$$F_{\mu,\sigma} = \frac{F - \mu}{\sigma} \quad (2.1)$$

In aggiunta alle statistiche descrittive, per ogni feature si è realizzato un istogramma attraverso il quale è stato possibile realizzare una prima analisi visiva sulla distribuzione dei valori. Questa operazione è stata svolta per verificare se le feature seguono una distribuzione normale.

TODO:immagine

Da questi grafici si evince che le feature *Energy*, *Homogeneity* e *Coarseness* non seguono una distribuzione normale, a differenza delle altre feature che possono essere considerate normali. In ogni caso, anche se non seguono l'ipotesi di normalità, si è deciso comunque di utilizzarli, anche se stiamo andando contro le assunzioni dei modelli che vogliamo utilizzare.

Da questa analisi, è stato possibile osservare che la feature *Coarseness* assume valori molto bassi, quindi è stato pensato di convertire questa feature ad una scala logaritmica, permettendo di aumentare la significatività dei valori.

Nonostante questa operazione, la feature *Coarseness* presenta una deviazione standard pari a circa 0.02, valore non particolarmente significativo, quindi questo ci ha permesso di escludere questa feature perché sicuramente non sarà la feature più discriminante.

Oltre all'analisi descrittiva delle feature, è stato eseguito anche un'analisi sulle etichette, ovvero sulle classi. Questo è stato fatto per verificare se le classi sono bilanciate, ovvero se il numero di esempi positivi è simile al numero di esempi negativi.

In questo caso, realizzando un istogramma, riportato in figura 2.1, che rappresenta le frequenze assolute delle classi. Da questo si può notare che la distribuzione di esse è bilanciata, infatti, circa il 55% degli esempi sono negativi (assenza del tumore), contro circa il 45% degli esempi sono positivi (presenza del tumore).

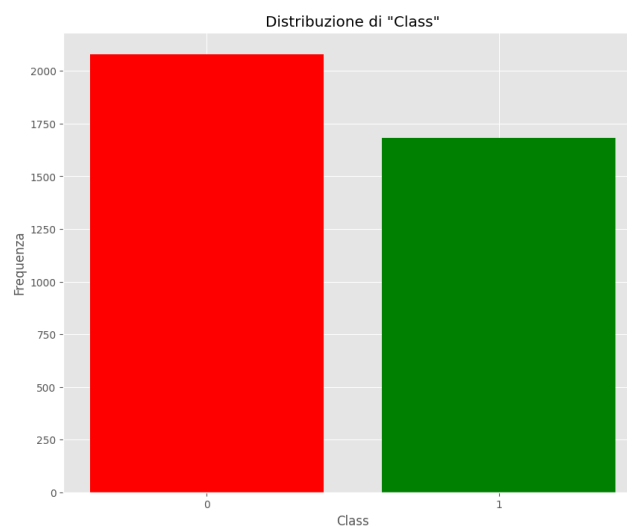


Figura 2.1: Distribuzione delle classi

Non è già stato detto prima? Successivamente risulta importante analizzare se le distribuzioni si avvicinano ad una normale standard, di conseguenza sono stati disegnati i barplot delle frequenze di ciascuna feature. Dai grafici si evince che tutte le feature eccetto *Energy* e *Homogeneity* sono distribuite normalmente.

Dal barplot delle features creato precedentemente, si possono ottenere ulteriori informazioni sulla distribuzione potenzialmente normale delle altre feature. Infatti, la distribuzione della *Standard deviation* è simmetrica, mentre tutte le altre presentano una asimmetria.

Ricapitolando, da questo primo studio descrittivo è stato modificato il dataset rimuovendo la feature *Coarseness*.

TODO:immagine

2.2.1 Analisi delle correlazioni

Il passaggio successivo è stato quello di analizzare le correlazioni tra le feature. Questo è stato fatto in modo tale da ridurre la dimensionalità dei dati, nello specifico sono state suddivise le feature in gruppi di feature altamente correlate tra loro, e in un secondo momento è stata scelta una feature per ogni gruppo.

Per fare ciò, è stata realizzata una matrice di correlazione, riportata in figura 2.2, 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

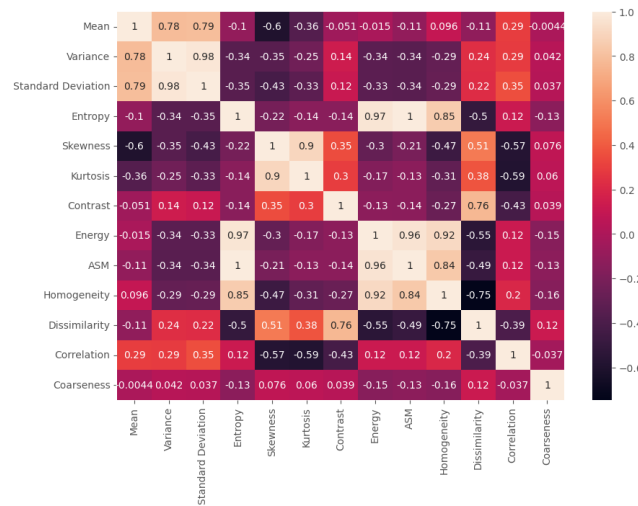


Figura 2.2: Matrice di correlazione

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 e deviazione standard facilmente spiegabile perché la deviazione standard è la radice quadrata della varianza ($SD[P] = \sqrt{VAR[P]}$).

TODO:immagine

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 meno 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*, infine, (**cosa c'entra Homogeneity?**).

Al tempo stesso, la matrice di correlazione evidenzia una forte correlazione positiva tra gli indici che misurano la **morfologia della distribuzione**, 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 alta 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 evidenzia che le features di *Dissimilarity* e *Homogeneity* sono correlate negativamente, ovvero maggiore sarà il contrasto allora minore è la complessità della texture.

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 2.3.

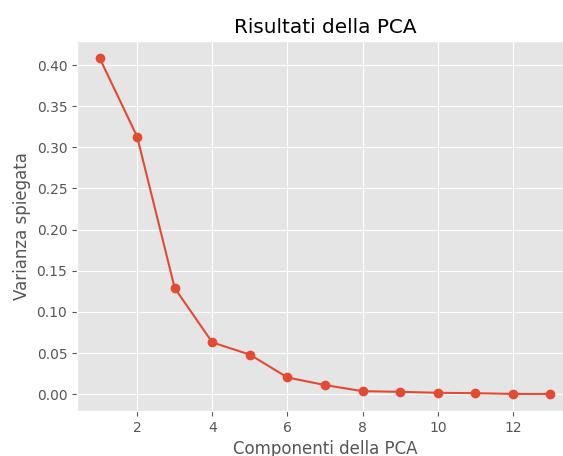


Figura 2.3: Percentuale di varianza spiegata per ogni componente

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.

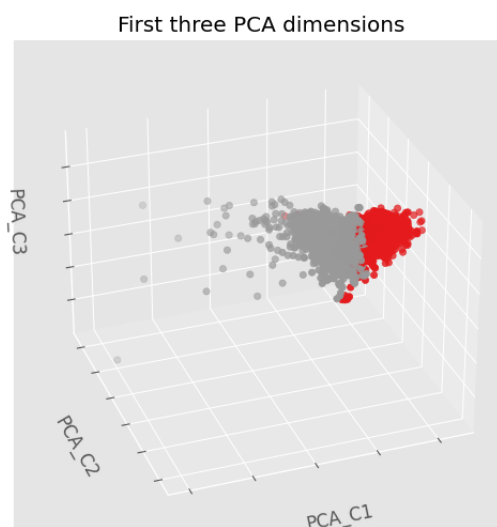


Figura 2.4: Scatter plot a 3 dimensioni

Dalla figura 2.4 si può osservare che i dati ottenuti dalla PCA sembrano essere separabili con un piano.

2.4 Preparazione dei dati per i vari modelli

In seguito a tutti i ragionamenti precedenti, dal primo dataset sono stati costruiti altri due datasets, il primo ridotto mediante la correlazione (Dataset-corr), mentre il secondo ridotto mediante PCA (Dataset-pca). La scelta di eseguire i modelli utilizzando due metodi per ridurre la dimensionalità è stata fatta solo per scoprire se ha davvero senso, in questo contesto, scomodare un algoritmo di riduzione di dimensionalità, come PCA, oppure ci si può limitare solo a considerare feature non fortemente correlate.

Successivamente su Dataset-corr e Dataset-pca sono state applicate delle operazioni di suddivisione degli esempi in due insiemi:

- **training set:** contenente l'80% degli esempi del dataset e viene utilizzato per effettuare la ricerca degli iperparametri migliori e, una volta trovati, si riutilizza questo insieme di esempi per allenare il modello con gli iperparametri migliori appena trovati. Per lo studio degli iperparametri è stata effettuata una 10-fold cross validation stratified.
- **test set:** contenente l'80% degli esempi del dataset rimanenti e viene utilizzato per effettuare la valutazione del modello.

Logicamente, la suddivisione tra training set e test set è stata realizzata in modo strutturato, ovvero nel training set e nel test set è stato preservato lo stesso bilanciamento delle classi presenti nel dataset di partenza, in modo tale da evitare di introdurre dei bias su modello.

In aggiunta alle operazioni di preprocessing introdotte precedentemente, è stata applicata anche un'operazione di standardizzazione dei dati definendo due nuovi dataset: Dataset-corr-std e Dataset-pca-std. La standardizzazione dei dati è stata necessaria dal momento che sono stati utilizzati dei modelli che assumono di ricevere in input dei dati standardizzati, come le reti neurali (NN) e le support vector machine (SVM).

Nella tabella 2.2 si può avere una ricapitolazione di tutti i dataset e delle operazioni effettuate per ottenerli.

Nome del dataset	Operazioni applicate	Utilizzato per i seguenti modelli
Dataset-corr	Riduzione della dimensionalità utilizzando l'analisi della correlazione	Gaussian Naive Bayes
Dataset-corr-std	Dataset-corr con la standardizzazione dei dati	SVM e NN
Dataset-PCA	Dataset-corr-std applicando l'algoritmo PCA	Gaussian Naive Bayes
Dataset-PCA-std	Dataset-PCA con la standardizzazione dei dati	SVM e NN

Tabella 2.2: Tabella riassuntiva dei dataset

Capitolo 3

Rete neurale

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 verrà presentato uno dei tre modelli che sono stati realizzati per svolgere il compito di classificazione, ovvero la rete neurale. Nello specifico 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.

3.1 Preparazione dei dati

Prima di passare alla presentazione della rete neurale nel dettaglio, risulta necessario specificare come sono stati preparati i dati per l'addestramento della rete neurale.

La prima operazione svolta sui dati è stata un'operazione di standardizzazione, ovvero per ogni feature è stata calcolata la media e la deviazione standard e questi valori sono stati utilizzati per standardizzare i dati.

Eseguendo questa operazione i dati sono stati trasformati in modo tale che la loro media sia 0 e la loro deviazione standard sia 1. Questa operazione è stata eseguita per garantire che la rete neurale non sia influenzata da valori di input con scale diverse.

La seconda operazione svolta sul dataset è stata la suddivisione in training set e test set. Questa operazione è stata fatta per poter utilizzare una parte dei dati per addestrare la rete neurale e una parte dei dati per valutare le prestazioni della rete neurale. La suddivisione del dataset è stata effettuata in modo tale che il training set contenesse il 80% dei dati, mentre il test set contenesse il 20% dei dati.

Questa operazione è stata effettuata in modo da mantenere la stessa percentuale di dati positivi e negativi in entrambi i set, con lo scopo di evitare che la rete neurale sia addestrata su un dataset sbilanciato e di conseguenza non sia in grado di generalizzare correttamente.

3.2 Struttura della rete neurale

Per definire la struttura della rete neurale sono stati utilizzati i risultati ottenuti dalla fase di analisi a cui si è aggiunta una fase di grid search. Questa fase è stata effettuata per valutare la combinazione migliore di iperparametri per la rete neurale.

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 evitare l'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 un processo di grid search.

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, in cui la funzione di attivazione è stata definita come il resto delle funzioni di attivazione.

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. Questo

processo è stato effettuata attraverso una cross validation a 5 fold, prendendo in considerazione solamente i dati del training set.

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, 50, 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*

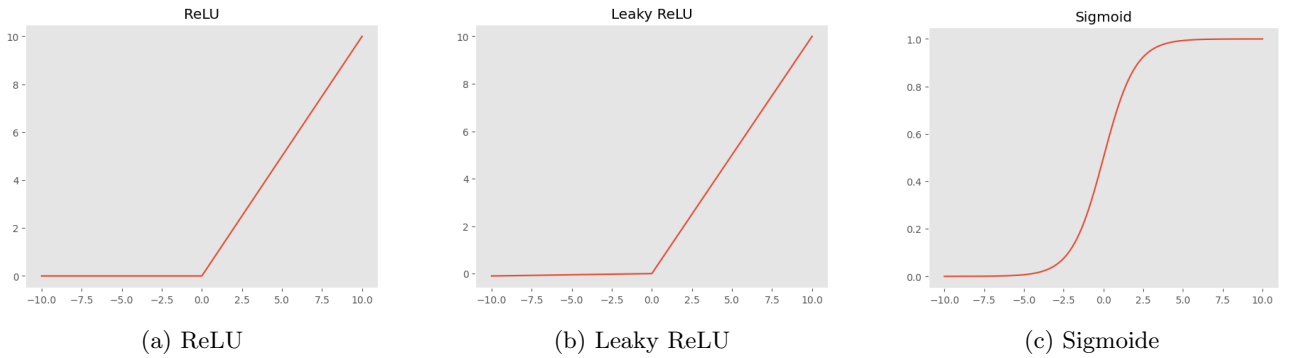


Figura 3.1: Funzioni di attivazione utilizzate nella fase di grid search

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, sono stati calcolati gli intervalli di confidenza al 95% per entrambe le misure.

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: si è scelto il modello che ha ottenuto il valore più alto combinando, attraverso dei pesi, i valori relativi all'accuratezza media, al tempo di addestramento medio e agli intervalli di confidenza ottenuti. Per questa operazione si è scelto di assegnare un peso maggiore all'accuratezza media ottenuta dalla cross validation e al tempo di addestramento medio.

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.

Per verificare la validità del modello scelto si è proceduto con il confronto tra il modello che ha ottenuto la migliore accuratezza e il modello 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	98.1%	0.96s
Accuratezza maggiore	99.5%	24.54s
Modello scelto	98.7%	2.7s

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 buon compromesso tra accuratezza e tempo di addestramento. Nello specifico, perdendo lo 0.8% di accuratezza si è ottenuto un tempo di addestramento minore di 21.84s secondi.

Definizione della struttura della rete neurale

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 nascosto è composto da 50 neuroni, in cui la funzione di attivazione è la funzione Leaky ReLU.

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. 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.

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

La struttura della rete neurale è riassunta nella figura 3.2.

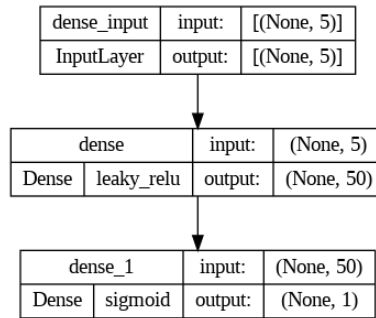


Figura 3.2: Struttura della rete neurale

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	98.80 %	99.10 %	98.21 %	98.65 %

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.

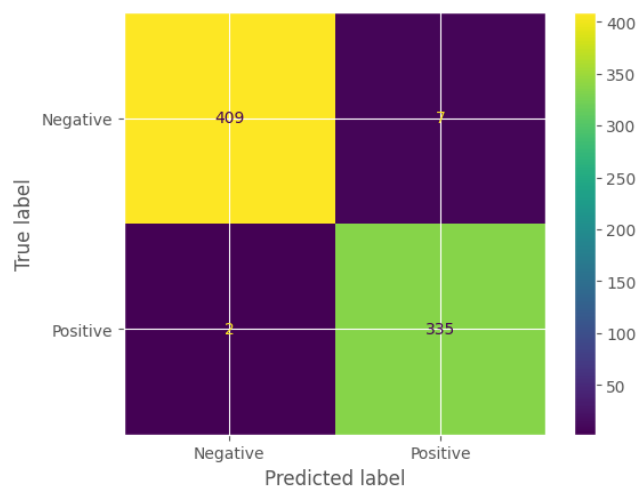


Figura 3.3: Matrice di confusione ottenuta dal modello addestrato

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.

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.

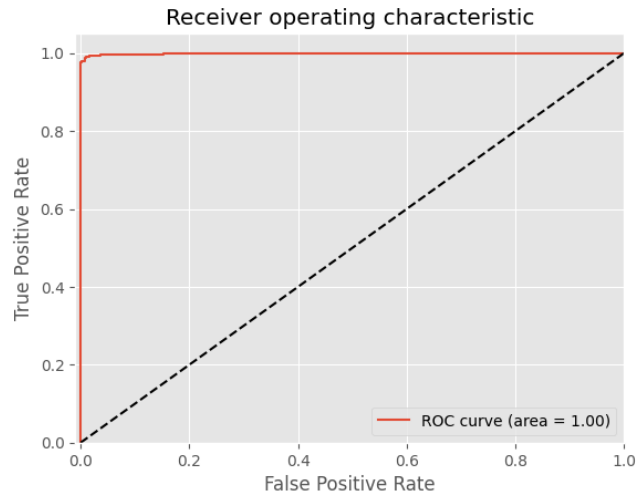


Figura 3.4: Curva ROC ottenuta dal modello addestrato

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	98.19 %	[97.71%, 98.67%]
Precisione	97.88 %	[97.16%, 98.59%]
Richiamo	98.09 %	[97.19%, 99.00%]
F1 score	97.97 %	[97.43%, 98.52%]

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.

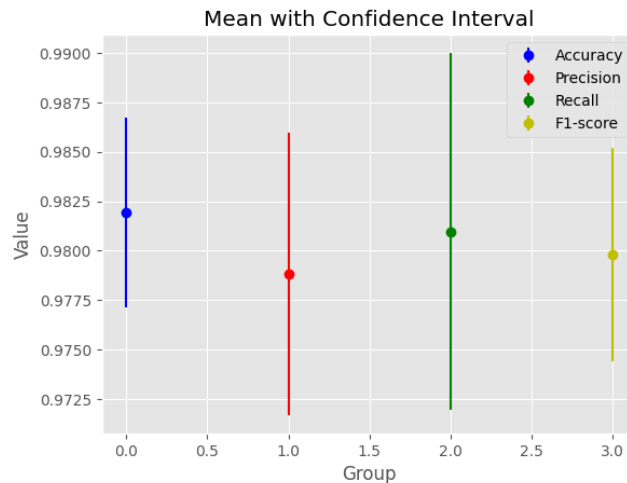


Figura 3.5: Risultati ottenuti dalla cross validation

3.5 Modello addestrato con PCA

A puro scopo didattico, si è deciso di addestrare una rete neurale sul dataset ottenuto attraverso la PCA. Per la precisione, si sono addestrati due modelli, uno con gli stessi iperparametri del modello addestrato in precedenza

e uno con iperparametri che sono stati stimati attraverso un processo di grid search.

Il dataset ottenuto attraverso la PCA, descritto nella 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.

Il meccanismo di grid search utilizzato per la ricerca degli iperparametri è stato lo stesso utilizzato per il modello addestrato in precedenza. Anche gli iperparametri valutati sono stati gli stessi.

I risultati ottenuti dai modelli addestrati sono riportati in tabella 3.4.

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

Tabella 3.4: Risultati ottenuti dai modelli addestrati con PCA

Per confrontare i modelli ottenuti con e senza PCA, sono state utilizzate, oltre alle metriche precedentemente presentate, anche le curve ROC e gli intervalli di confidenza. I risultati ottenuti sono riportati in figura 3.6.

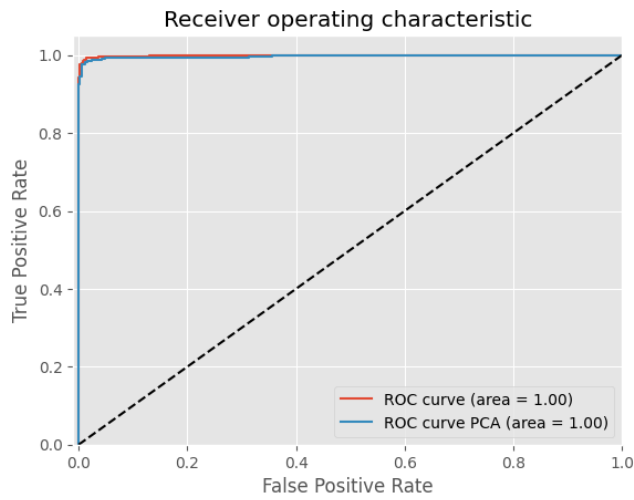
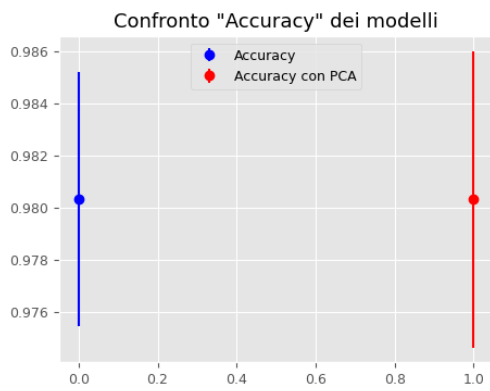


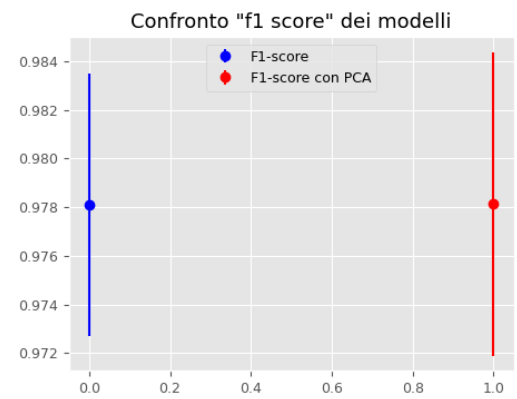
Figura 3.6: Confronto curve ROC tra il modello addestrato con e senza PCA

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 può notare che il modello addestrato con PCA ha un intervallo di confidenza più ampio rispetto al modello addestrato senza PCA.

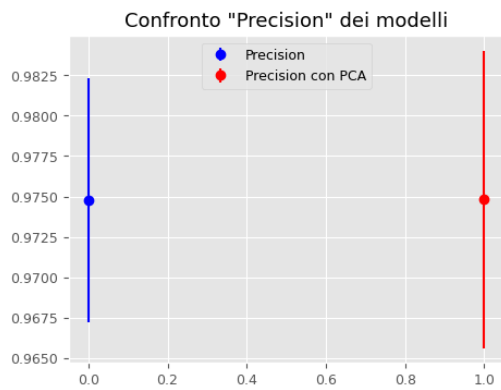
Nella figura 3.7 sono riportati gli intervalli di confidenza ottenuti dai modelli addestrati con e senza PCA. Da questa figura si può notare che l'intervallo di confidenza ottenuto dal modello addestrato con PCA, rappresentato dalla linea di colore rosso, è più ampio rispetto a quello ottenuto dal modello addestrato senza PCA, rappresentato dalla linea di colore blu.



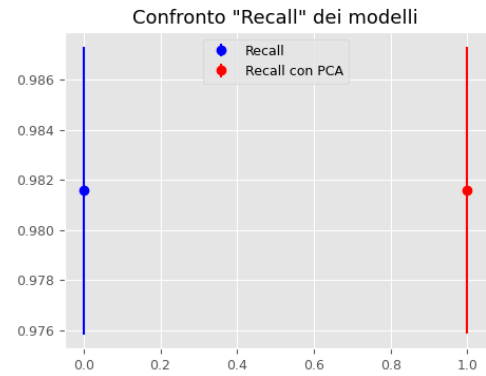
(a) Accuracy



(b) F1 score



(c) Precision



(d) Recall

Figura 3.7: 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.

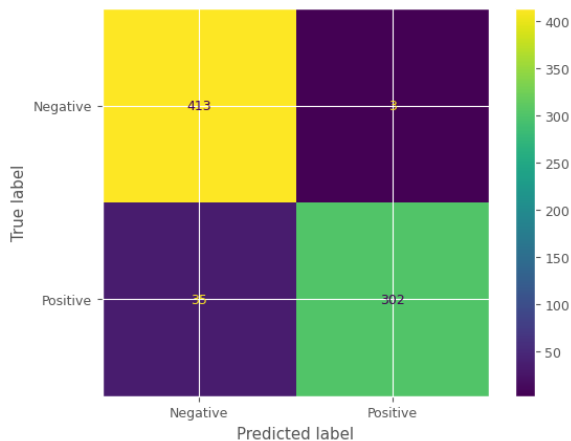
4.1 Addestramento di Gaussian Naive Bayes

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

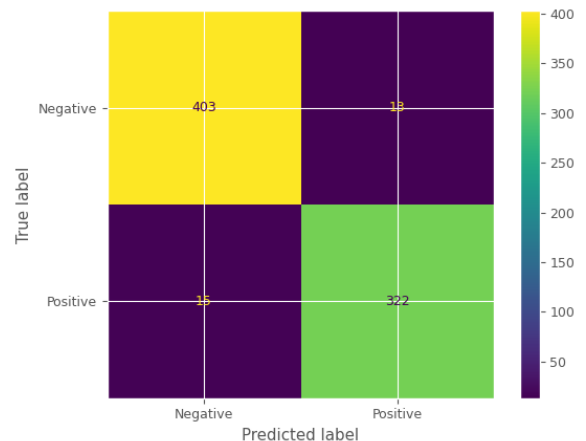
4.2 Risultati

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 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 nelle tabelle 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

Il modello addestrato in precedenza è stato valutato sui dati che compongono il test set. In particolare sono state valutate le seguenti metriche: accuracy, precision, recall e 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 4.1 vengono mostrati i valori delle metriche, mentre nella figura

Complessivamente i risultati ottenuti sull'apprendimento sono molto positivi, questo è dovuto principalmente dal fatto che il dataset è facilmente separabile linearmente.

Bibliografia

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