Breast Cancer Prediction

Chiara Coscarelli
matr. 0512113869

January 10, 2024

Contents

1 Introduzione

Il cancro al seno (BC) è uno dei tumori più comuni tra le donne in tutto il mondo e, secondo le statistiche globali, rappresenta la maggior parte dei nuovi casi di cancro e dei decessi correlati al cancro, rendendolo un problema di salute pubblica significativo nella società odierna. La diagnosi precoce del BC può migliorare significativamente la prognosi e le possibilità di sopravvivenza, poiché può promuovere un trattamento clinico tempestivo per i pazienti. Una classificazione più accurata dei tumori benigni può evitare che i pazienti si sottopongano a trattamenti non necessari. Pertanto, la diagnosi corretta di BC e la classificazione dei pazienti in gruppi maligni o benigni è oggetto di molte ricerche. Il codice, la presentazione e la documentazione, si possono trovare presso tale repository https://github.com/chiaracos/BreastCancerPrediction

1.1 Obbiettivi

Questo progetto di Fondamenti di Intelligenza Artificiale, il mio primo approccio in questo ambito, mira a osservare quali caratteristiche sono più utili nel predire il cancro maligno o benigno e a vedere le tendenze generali che potrebbero aiutarci nella selezione del modello. L'obiettivo è classificare se il cancro al seno è benigno o maligno. Ho scelto questa tematica perché relativamente semplice, e quindi in grado di farmi apprendere il più possibile le basi del Machine Learning. Gli obiettivi principali includono:

- L'analisi approfondita dei dati estrapolati da un dataset.
- L'identificazione di feature associate alle diagnosi.
- L'implementazione di un modello di apprendimento in grado di calcolare la probabilità che un tumore al seno sia benigno o maligno.

1.2 Specifica PEAS

- Performance (misure di prestazione adottate per valutare l'operato di un agente), nel mio caso verranno valutate la precisione di classificazione e l'accuratezza.
- Environment (elementi che formano l'ambiente), nel mio caso è costituito dai dati clinici dei pazienti, inclusi parametri e misure relative al cancro al seno.
- Actuators (attuatori disponibili all'agente per intraprendere le azioni), nel mio caso sarà la capacità di predirre la presenza o assenza di cancro al seno.
- Sensors (sensori attraverso i quali l'agente riceve gli input percettivi), nel mio caso l'agente va ad acquisire dati clinici del paziente, tra cui risultati di esami, misure e caratteristiche correlate al cancro al seno.

1.3 Caratteristiche dell'ambiente

L'ambiente è:

- Singolo agente.
- Stocastico: le diagnosi possono essere influenzate da fattori difficilmente prevedibili o che presentano una certa variabilità.
- Parzialmente osservabile: non si ha accesso a tutte le informazioni complete relative ai pazienti.
- Dinamico: il cancro al seno è una malattia che evolve nel tempo. La progressione della malattia può variare da paziente a paziente, e le caratteristiche biologiche del tumore possono cambiare nel corso del tempo.
- Discreto: le variabili assumono valori in un limitato intervallo; alcune, invece, assumono valori distinti e separati.
- Sequenziale: le predizioni passate sulla risposta al trattamento possono fornire un contesto importante per valutare l'efficacia delle terapie successive.

1.4 Analisi del problema

La previsione del cancro al seno è un problema di apprendimento supervisionato, più in particolare di classificazione.Nelle successive sezioni vado a descrivere tutte le problematiche che ho affrontato. Per quanto riguarda le tecnologie che ho utilizzato per lo sviluppo del progetto, abbiamo:

- Python (in dettaglio le librerie per ML, come sickitLearn, Pandas, ecc.),
- JupyterNotebook all'interno dell'IDE PyCharm
- GitHub per il versionamento,
- Overleaf e Canva per documentazione e presentazione.

2 Data Understending

Tale fase é composta da più punti:

- Acquisizione dei dati
- Esplorazione dei dati
- Analisi della qualità dei dati

2.1 Acquisizione dei dati

L'acquisizione dei dati è il processo di raccolta, ed organizzazione dei dati necessari per andare a creare un modello di ML. Con gli obiettivi chiari e definiti, sono andato alla ricerca di un dataset fino a trovarne uno molto interessante su Kaggle.Il dataset preso in considerazione è presente al link:https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data

2.2 Esplorazione dei dati

Il dataset in esame contiene 569 righe e 32 colonne. ' Diagnosi ' è la colonna che useremo per predire, ovvero che dice se il cancro è M = maligno o B = benigno. In questa fase vado ad analizzare, o meglio esplorare nel dettaglio i dati per comprenderli meglio. Prima di tutto sono andata a fare una panoramica del dataset, andando a conteggiare quanti sample per ogni classe (B/M) fossero presenti.

2.3 Analisi della qualità dei dati

In questa fase vado ad analizzare i problemi di qualità dei dati rilevati durante la fase di esplorazione. Possiamo notare che la quantità di diagnosi B, è maggiore rispetto alla quantità di diagnosi M. Difatti, andando ad approfondire, è risultata la presenza di:

- 357 B (benigne)
- 212 M (maligne)

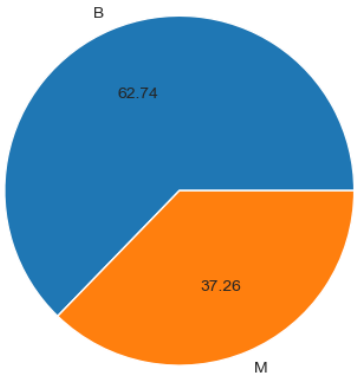


Figure 1: percentuali di frequenza delle classi

Questo indica un forte sbilanciamento dei dati, ed è quindi un problema che dovrà essere risolto. Altri problemi riscontrati:

- una colonna del dataset totalmente nulla.

Questo problema sarà risolto nella sezione successiva, ovvero nella Data Preparation.

3 Data Preparation

Tale fase mira a rendere i dati adatti per l'utilizzo nelle fasi successive del processo. Questo processo include più punti:

- data cleaning
- feature scaling
- feature selection
- data balancing

Quindi l'output di questa fase sarà un insieme di dati di input, che saranno utilizzati durante la modellazione dell'algoritmo di ML.

3.1 Data Cleaning

In questa fase vanno corretti i problemi individuati in fase di Data understanding: nel mio caso la presenza di una colonna nulla; per poi passare alla trasformazione dei dati in modo da poter essere utilizzati da un algoritmo di apprendimento. Quindi, innanzitutto, ho eliminato completamente la colonna dal dataset poichè era completamente vuota. Successivamente ho sostituito le stringhe "B" e "M" della colonna diagnosi rispettivamente con i valori 0 e 1, per garantire la compatibilità con gli algoritmi di classificazione binaria. Infine, ho controllato se fossero presenti valori nulli e valori duplicati.Nel mio caso non ce ne sono.

3.2 Feature scaling

In questa fase si vanno ad utilizzare delle tecniche per normalizzare o scalare i valori delle caratteristiche in modo da avere una scala uniforme. Questo serve per evitare che caratteristiche con scale molto diverse influenzino negativamente gli algoritmi di apprendimento. Ciò può essere ottenuto normalizzando.Prima di scalare i dati però ho fatto lo split del Dataset, ovvero ho diviso l'insieme di partenza in due sottoinsiemi: uno per l'addestramento ed uno per il testing per prevenire problemi di data Leakage. Infatti, se si esegue lo scaling prima di suddividere il dataset in set di addestramento e testing, potrebbe verificarsi un problema noto come "data leakage" (fuga di dati). Lo scaling basato sull'intero dataset potrebbe incorporare informazioni del set di testing nel set di addestramento, influenzando erroneamente le prestazioni del modello durante la fase di valutazione.

3.3 Feature selection

La feature selection è il processo in cui si va a scegliere un sottoinsieme delle caratteristiche più rilevanti dai dati originali per andare a ridurre la complessità del modello. Qui entra in gioco il Feature Engineering ovvero il processo nel quale il progettista utilizza la propria conoscenza del dominio per determinare le feature e dare piu o meno enfasi ad esse.

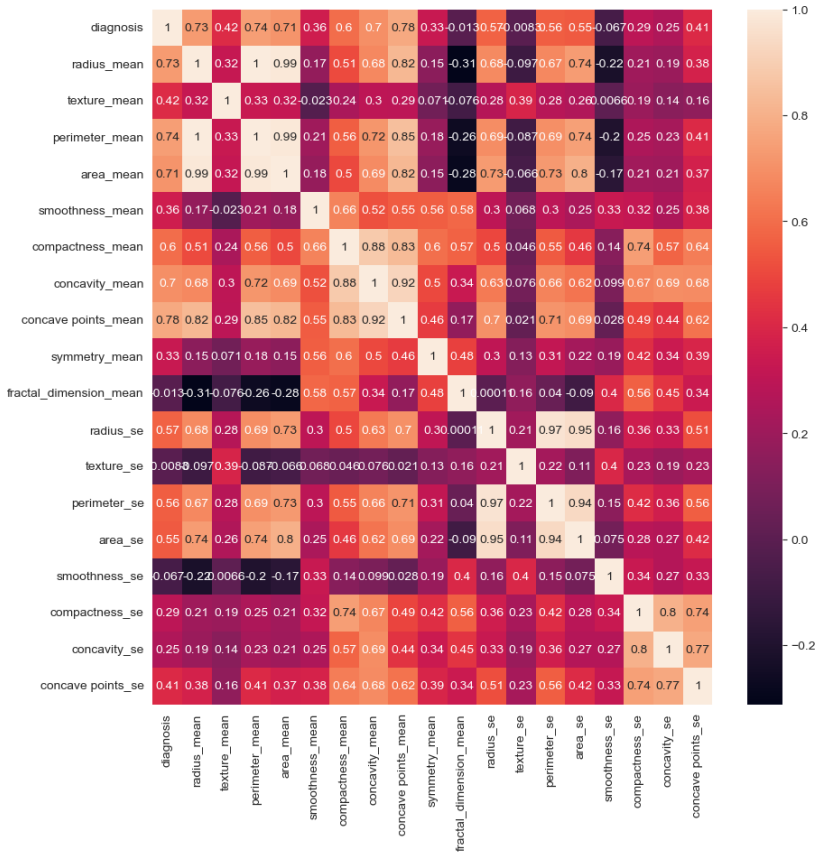


Figure 2: matrice di correlazione

Per calcolare la correlazione tra le variabili, possiamo usare un'ulteriore strumento: la heatmap, una mappa che utilizza colori per visualizzare i valori dei coefficienti di correlazione tra le diverse coppie di variabili nel dataset, consentendo di individuare facilmente relazioni tra di esse.

- Le celle più scure o più chiare indicano correlazioni più forti o più deboli, rispettivamente.
- Le variabili sono fortemente correlate tra loro se hanno valori vicini a 1 o -1, mentre hanno una bassa correlazione se hanno valori vicini a 0.

Nel mio caso, dalla matrice di correlazione possiamo notare che ci sono alcune variabili altamente correlate, ovvero che contengono valori simili che possono portare problemi di multicollinearità nei modelli, quindi possiamo considerare di eliminarne una.Per scegliere quale tra queste eliminare sono andata a verificare come ciascuna delle variabili è correlata con la variabile dipendente, e ho deciso

di mantenere la variabile maggiormente correlata alla variabile dipendente. In questo modo, si può portare il modello ad un miglioramento delle prestazioni. Tra le variabili altamente correlate abbiamo:

- radius-mean, perimeter-mean e area-mean sono correlati tra loro e utilizziamo solo perimeter-mean.
- compactness-mean, concavity-mean and concave points-mean sono correlati tra loro e utilizziamo solo concave points-mean
- radius-se, perimeter-se and area-se sono correlati tra loro e utilizziamo solo perimeter-se
- radius-worst, perimeter-worst and area-worst sono correlati tra loro e utilizziamo solo area-worst.
- Compactness-worst, concavity-worst and concave points-worst sono correlati tra loro e utilizziamo solo concavity-worst
- Compactness-se, concavity-se and concave points-se sono correlati tra loro e utilizziamo solo concavity-se.
- texture-mean, texture-worst sono correlati tra loro e utilizziamo solo texture-mean.

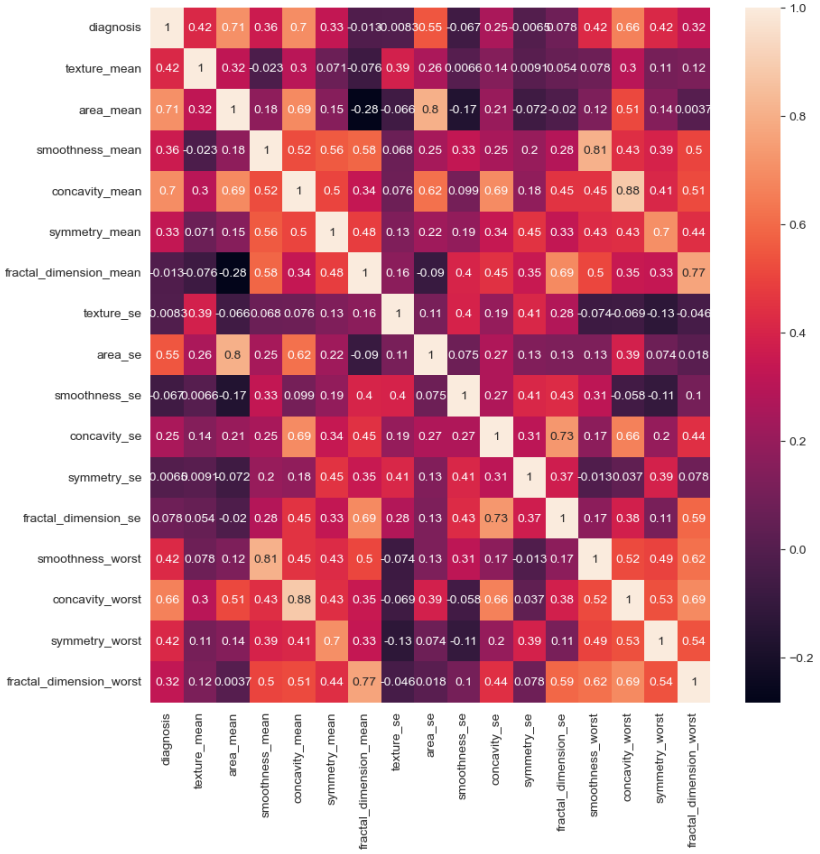


Figure 3: Matrice di correlazione dopo aver eliminato le feature correlate

3.4 Data Balancing

Il Data Balancing si riferisce alle tecniche usate per convertire un dataset sbilanciato in un uno bilanciato.Avere un dataset sbilanciato può portare diversi problemi, tra questi previsioni sbilanciate e innaccurate, nonchè problematiche di overfitting. Le tecniche applicabili sono due:

- Undersampling
- Oversampling

Nel mio caso, il dataset è fortemente sbilanciato, con una presenza maggiore di istanze della classe Benigna. Quindi le possibilità sono due:

- Eliminare istanze della classe benigna
- Aumentare le istanze della classe maligna

Tra le due, per evitare troppa duplicazione e quindi probabile overfitting, ho deciso di optare per la prima opzione, andando a considerare l'uso di Undersampling con RandomUnsersampler. Ho usato RandomUndersampling perché un dataset sbilanciato può portare a problemi di apprendimento, quindi il modello potrebbe avere difficoltà a identificare correttamente la classe minoritaria. Random Undersampling affronta questo problema riducendo casualmente il numero di campioni della classe maggioritaria, aiutando a bilanciare le proporzioni tra le classi.

4 Data Modeling

Nella sezione precedente ho preparato i dati in modo da essere dati in input ad un algortimo, quindi può iniziare la fase di modeling. Per prima cosa va selezionato l'algoritmo da utilizzare per poi passare alla fase di addestramento, dove si addestra il modello e di conseguenza si descrivono i risultati ottenuti.

4.1 Scelta dell'algoritmo

Breast Cancer Prediction va a trattare un problema di apprendimento supervisionato, nel dettaglio un problema di classificazione. Difatti verrà fornito all'algoritmo un insieme di dati con rispettivo valore della variabile target, ed inoltre la variabile target potrà assumere solamente un numero discreto di valori, nello specifico solamente due: 0 per "B" e 1 per "M". Quindi i possibili algoritmi utilizzabili sono i seguenti: Regressione Logistica, Support Vector Machines, Kernel SVM, Nearest Neighbor, Random Forest, Naive Bayes e gli Decision Tree(DTC). Tuttavia ho concentrato la mia scelta solo sui due algortimi visti a lezione, ovvero:

- Naive Bayes
- Decision Tree

Questi sono due algoritmi che operano in maniera differente per arrivare alla classificazione: infatti il primo considera le caratteristiche della nuova istanza da classificare e calcola la probabilità che queste facciano parte di una classe tramite l'applicazione del teorema di Bayes, mentre il secondo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni; il tutto sfruttando i concetti di Entropia e Information-Gain utili per andare a suddividere l'insieme di partenza. In linea generale, gli Alberi Decisionali sono noti per la loro flessibilità nel catturare relazioni complesse nei dati, mentre il classificatore Naive Bayes eccelle per la sua velocità di addestramento. Siccome non sono in grado di prevedere quale si adatti meglio al mio problema, ho adottato la valutazione empirica: ovvero l'addestramento del modello prima con un Naive Bayes e successivamente con Decision Tree, per poi valutare e confrontare i risultati ottenuti. In questo modo riesco a determinare quale algoritmo si comporta meglio rispetto al mio problema ed ai miei obiettivi.

4.2 Addestramento

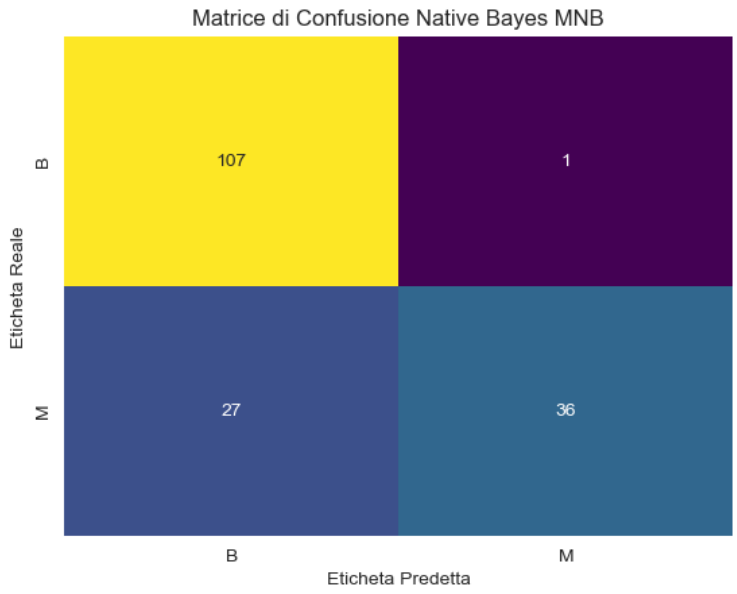
In questa sezione procedo ad addestrare il modello in base all'algoritmo scelto in precedenza. Nel mio caso, come già detto, procedo prima all'addestramento usando Naive Bayes, per poi fare lo stesso con Decision Tree. Infine valuterò le prestazioni ottenute e le metterò a confronto. Tutto ciò mi consentirà di capire quale algoritmo si adatta meglio al problema, e va di conseguenza a fare delle predizioni migliori.

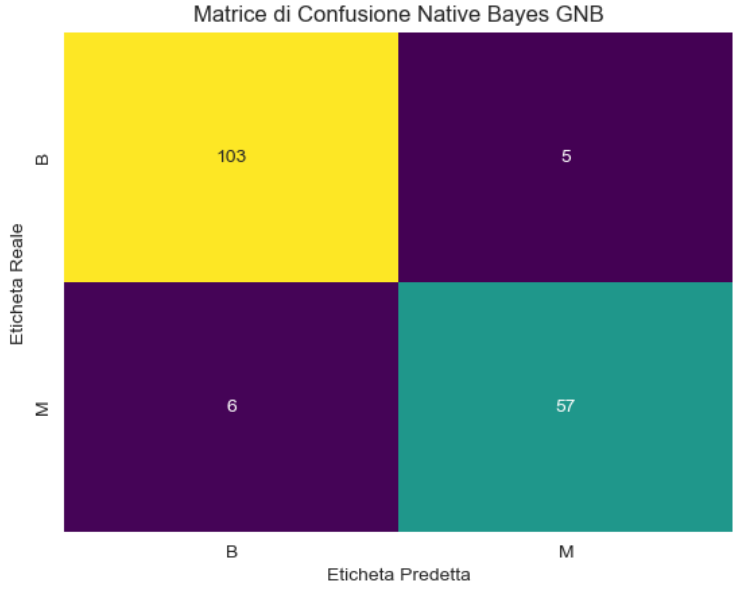
4.3 Native Bayes

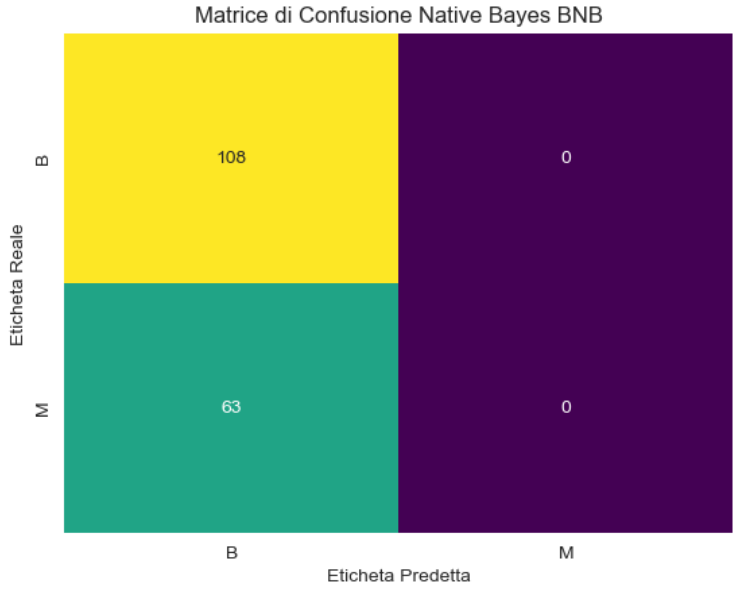
L'algoritmo Naive Bayes è un algoritmo di classificazione basato sul teorema di Bayes, che riguarda la probabilità condizionata. Si parla di "naive" perchè tale algortimo considera l'indipendenza tra le caratteristiche, ciò semplifica i calcoli e rende l'algoritmo veloce. Ci sono diverse varianti del Naive Bayes, tra cui:

- Multinomial Naive Bayes (MNB)
- Gaussian Naive Bayes (GNB)
- Bernoulli Naive Bayes (BNB)

Tali differiscono nelle loro ipotesi sulla distribuzione delle caratteristiche, ovvero la probabilità delle diverse features condizionate alle classi di output. Ad esempio l'MNB è particolarmente utile per problemi di classificazione di testi, il GNB per dati continui, ed infine il BNB per dati binari. Anche qui ho adottato la valutazione empirica, andandoli a provare tutti. Queste sono le matrici di confusione che ho ottenuto:







Successivamente, a partire dalle matrici di confusione, che esprimono al loro interno il numero di True Positive (alto a sinistra), True Negative (basso a destra), False Positive (alto a destra), False Negative (basso a sinistra) sono andato anche a calcolare alcune metriche di valutazione, ovvero Precisione,Recall e Accuratezza; che vanno calcolate proprio usando gli indicatori espressi dalle matrici. Il mio obbiettivo è andare a massimizzare tutte e tre le metriche. Risultati

ottenuti:

Variante	Precisione	Recall	Accuratezza
BNB	1.0	0.0	0.631578947368421
GNB	0.9193548387096774	0.9047619047619048	0.935672514619883
MNB	0.972972972972973	0.5714285714285714	0.8362573099415205

Andiamo a fare un'analisi dei risultati ottenuti, ma prima vediamo cosa esprimono nel mio caso tali indicatori.

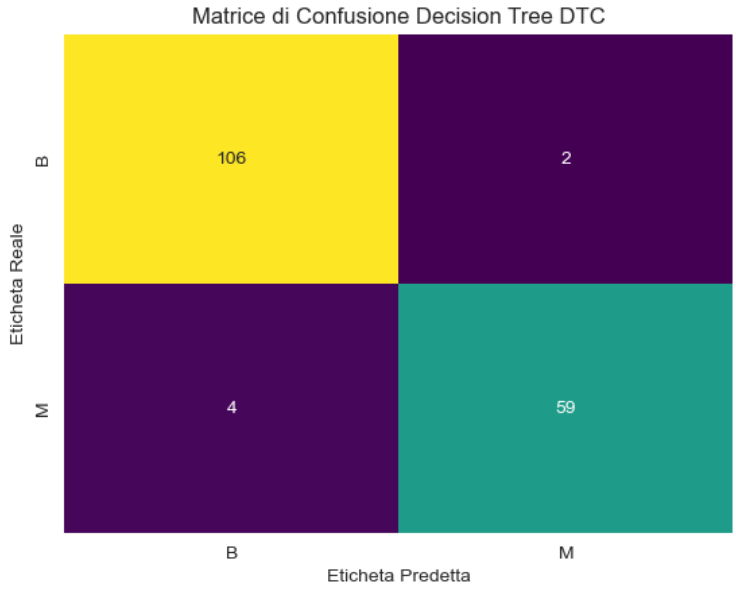
- La recall misura la capacità del modello di identificare correttamente i casi positivi rispetto al totale dei casi positivi reali.
- La precision misura la proporzione di casi positivi effettivamente positivi tra tutte le predizioni positive fatte dal modello.
- L'accuratezza misura la proporzione complessiva di predizioni corrette fatte dal modello rispetto al totale delle predizioni.

Da come emerge dai valori ottenuti, il classificatore Naive Bayes GNB ha ottenuto ottimi risultati su tutte le metriche. Una particolarità osservabile è che il Naive Bayes BNB ha ottenuto una precisione pari a 1 questo perchè non ci sono falsi positivi nel suo output,a sfavore di una recall pari a 0. In altre parole, non riesce a catturare alcuna delle istanze appartenenti alla classe positiva (True Positive, TP). Nel mio caso, il fatto che la recall nel Naive Bayes BNB è pari a

0, significa che il modello non è riuscito a catturare nessun caso positivo (nessun caso di tumore maligno) tra quelli effettivamente presenti nel dataset.Per questo motivo questo modello non supererà la fase di validazione.Quindi tra i due rimanenti prediligo il classificatore Native Bayes GNG.

4.4 Decision Tree DTC

Il DTC è un classificatore che opera attraverso una struttura ad albero. Ogni nodo rappresenta un sottoinsieme delle caratteristiche, e i rami che si dipartono da ogni nodo indicano i possibili valori di quelle caratteristiche. Il processo di costruzione dell'albero inizia con la scelta della caratteristica che meglio divide il set di dati in classi omogenee, ciò viene fatto utilizzando misure come l'entropia o l'information-gain (misura quanto un attributo divide bene il dataset). Una volta individuata la caratteristica deve essere creato un nodo e diviso il set di dati in base ai possibili valori di quella caratteristica. Ciò viene quindi ripetuto su ciascun sottoinsieme di dati, creando ulteriori nodi e rami fin quando viene raggiunto un criterio di stop o si arriva ad avere dei set puri, ovvero composti esclusivamente da istanze di un'unica classe. Dopo l'addestramento ho testato il modello, e questa è la matrice di confusione ottenuta:



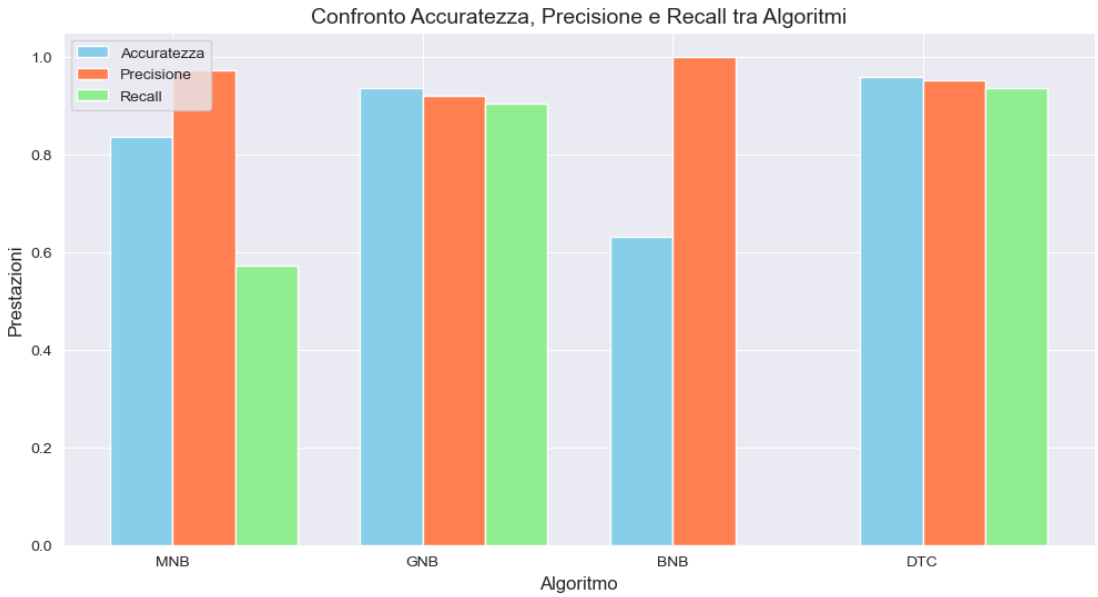
Possiamo notare che l'algoritmo DTC ha un importante parametro: maxDepth, ovvero il numero massimo di livelli creati per l'albero. Per trovare la profondità massima ottimale ho rieseguito l'algoritmo più volte, andando a variare il parametro. Ho osservato che le performance in termini di precisione/accuratezza/recall sono aumentate fino ad una profondità pari a 3, per poi andare in stallo o peggiorare.

4.5 GNG VS DTC

Visualizziamo e confrontiamo i dati del Native Bayes GNG e Decision Tree(DTC).

Variante	Precisione	Recall	Accuratezza
GNB	0.9193548387096774	0.9047619047619048	0.935672514619883
DTC	0.9672131147540983	0.9365079365079365	0.9649122807017544

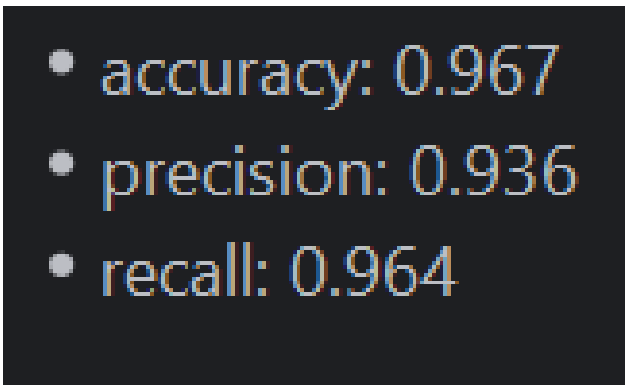
Come si può notare i risultati ottenuti usando il GNG non sono dei migliori. Quindi la discrepanza tra le due valutazioni potrebbe essere data dal differente modo di operare dei due algoritmi. Gli alberi decisionali sono flessibili nel modellare relazioni complesse tra le feature e la variabile target. Come abbiamo visto il nostro dataset ha feature complesse e relazioni non facilmente rappresentabili da regole decisionali semplici quindi il Decision Tree è in grado di catturare al meglio queste complessità. Il Native Bayes, invece, semplifica la complessità del problema assumendo un'indipendenza tra le varie caratteristiche. Questo è un grafico che riporta i risultati ottenuti da tutti gli addestramenti svolti:



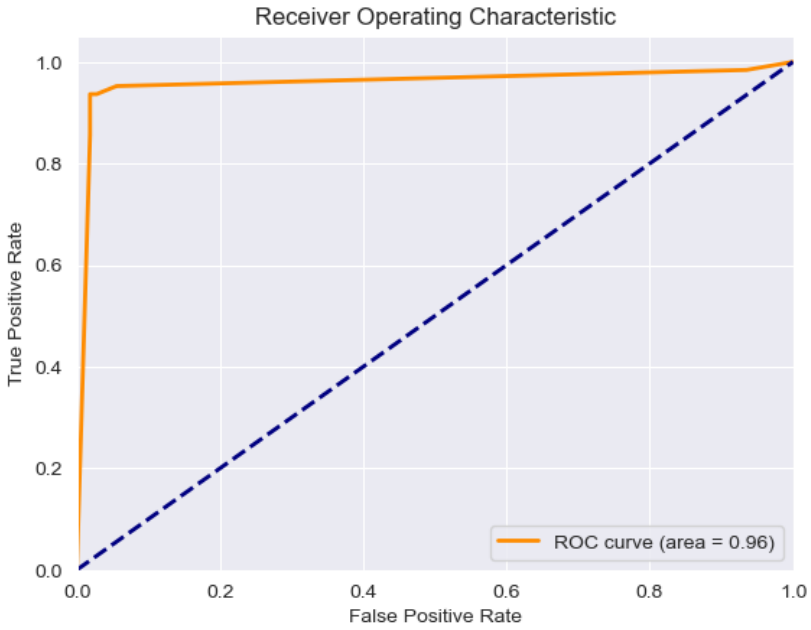
Possiamo notare che l'algoritmo che si è comportato meglio è stato il Decision Tree(DTC)

5 Evaluation

In questa fase si va a valutare se i risultati sono chiari, se sono in linea con gli obiettivi e se rivelano delle prospettive aggiuntive alle quali non si era pensato, quindi si va a verificare la consistenza e la solidità dell'intero processo. Vado quindi ad esaminare più nel dettaglio i risultati che ho ottenuto dall'addestramento e dal testing del Decision Tree, che ha avuto le seguenti performance:



Per avere più chiara la bontà del modello ho effettuato un'ulteriore valutazione attraverso la ROC-AUC curve, che consente di visualizzare il trade-off tra sensibilità(true positive rate) e specificità(true negative rate).



La linea diagonale rappresenta il risultato di una scelta casuale, quindi un qualunque modello utile deve essere al suo di sopra. Inoltre un buon modello deve posizionarsi in alto a sinistra, ovvero deve avere sensibilità elevata e basso tasso di falsi positivi, cio'e il modello deve riuscire correttamente a predirre che un paziente ha il cancro al seno senza generare falsi positivi, ovvero predirre che un paziente ha il cancro al seno, ma in realtà non lo ha. Ultima non meno importante la AUC (Area sotto la curva) che fornisce una misura generale delle prestazioni del modello, ad esempio un'area di 1.0 indica un modello perfetto, mentre un'area di 0.5 indica una scelta casuale. Il mio modello ha ottenuto un valore pari a 0.96, che è un ottimo risultato. Quindi posso considerare la costruzione del modello e in generale dell'approccio completa.

6 Conclusioni

In conclusione, posso affermare con certezza che il completamento di questo progetto è stata un'esperienza gratificante e formativa. Attraverso la sfida di esplorare nuovi concetti nel campo del machine learning, ho migliorato le mie competenze, dal perfezionamento del linguaggio Python fino all'applicazione di librerie specifiche.

7 Rifermienti bibliografici

From Wikipedia. Decision tree learning. From Wikipedia. Naive bayes classifier.