UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Architetture Dati

Use case: data quality for Machine Learning

Autori:
Emilio Toli - m.920337
Daniel Satriano - m.919053

Anno accademico:
2023/2024

Indice

1 Introduzione **1**
- 1.1 Tipi di rumore . . . 1
- 1.2 Progetto . . . 2
- 1.3 Dataset utilizzato . . . 2
  - 1.3.1 Descrizione delle feature . . . 2

2 Analisi del dataset **3**
  - 2.0.1 Distribuzione del target . . . 3
  - 2.0.2 Feature più importanti e correlazione . . . 4

3 Modello di classificazione: Decision Tree **9**
- 3.1 Split del dataset . . . 9
- 3.2 Addestramento del modello e analisi . . . 10
  - 3.2.1 Modello . . . 10
  - 3.2.2 Analisi delle prestazioni . . . 10
  - 3.2.3 Feature più importanti per il Decision Tree . . . 14

4 Rumore nel dataset **15**
- 4.1 Valori nulli . . . 16
  - 4.1.1 Rumore 1% . . . 16
  - 4.1.2 Rumore 2% . . . 17
  - 4.1.3 Rumore 3% . . . 18
  - 4.1.4 Rumore 50% . . . 19
  - 4.1.5 Rumore 100% . . . 20
  - 4.1.6 Evoluzione del modello . . . 21
  - 4.1.7 Analisi specifica dell'impatto delle feature sul modello . 22
- 4.2 Outliers . . . 26
  - 4.2.1 Rumore 1% . . . 26
  - 4.2.2 Rumore 2% . . . 28
  - 4.2.3 Rumore 3% . . . 30
  - 4.2.4 Rumore 50% . . . 32
  - 4.2.5 Rumore 100% . . . 34
  - 4.2.6 Evoluzione del modello . . . 35
  - 4.2.7 Analisi specifica dell'impatto delle feature sul modello . 36
- 4.3 Righe duplicate . . . 41
  - 4.3.1 Rumore 1% . . . 41
  - 4.3.2 Rumore 2% . . . 42
  - 4.3.3 Rumore 3% . . . 43
  - 4.3.4 Rumore al 50% . . . 44

4.3.5 Rumore al 100% 45

5 Conclusione **47**

Capitolo 1

Introduzione

Nel contesto del Machine Learning la qualità dei dati è un aspetto fondamentale per quanto riguarda le prestazioni dei modelli. Infatti, anche modelli molto avanzati e sofisticati non potranno mai superare i limiti imposti da una scarsa qualità del dataset, che influenza in modo negativo le prestazioni del modello. Questo principio è noto come "Garbage In - Garbage Out" e sottolinea la forte dipendenza che si ha nel machine learning dai dati in ingresso.

1.1 Tipi di rumore

Il rumore nei dati si riferisce a qualsiasi informazione errata presente in un set di informazioni, che può manifestarsi sotto varie forme, le principali sono:

- **Outliers**: dati che si discostano in modo significativo dagli altri punti nel dataset e possono essere causati spesso da errori di inserimento
- **Valori mancanti**: sono dati incompleti che potrebbero avere valori nulli all'interno dei campi
- **Righe duplicate**: è una tipologia di rumore che si verifica quando un dataset contiene più copie identiche dello stesso dato
- **Rumore Sistematico**: è una tipologia di errore che si presenta in modo sistematico e seguendo un pattern prevedibile e può essere causato ad esempio da errori provenienti da strumenti di misura
- **Label Noise**: è un errore che deriva da errori di etichettatura nei dati, facendo si che il modello apprenda a partire da informazioni sbagliate

Essi, se presenti ed in percentuali elevate, hanno come effetti diretti la riduzione delle performance, possibili errori nella fase di feature selection e potrebbero, in fase di produzione, contraddire i dati di training.

1.2 Progetto

Lo scopo di questo progetto è quello di analizzare le prestazioni di un modello di classificazione all'aumentare del rumore inserito in modo incrementale per poi confrontare i risultati ottenuti per valutare l'impatto che il rumore ha sulla qualità del modello.

1.3 Dataset utilizzato

Il set di dati utilizzato raccoglie 1500 istanze, di 32 feature ciascuna, relative alle misurazioni tumorali riguardanti tumore al seno, con relativa classificazione in tumore benigno oppure maligno.

1.3.1 Descrizione delle feature

- Feature 1: **Id**: è la colonna che indica l'identificativo univoco del paziente. Dato che non fornisce informazioni utili al modello e alla classificazione, pertanto è stata presa la decisione di rimuoverla dal set di dati
- Feature 2-11: Feature che indicano la media delle differenti misurazioni relative a ciascuna area della massa tumorale
- Feature 11-21: Informazioni relative all'errore standard (SE), ossia una misura di incertezza nelle stime della media
- Feature 21-31: Colonna che indica il valore massimo per ogni tipologia di misurazione, relativo all'insieme di sezioni appartenenti alla massa tumorale.
- Feature 32: **diagnosis**: è la colonna target e indica se il tumore è benigno (etichetta 'B', poi trasformata nel valore 1) oppure maligno (etichetta 'M', poi trasformata nel valore 0)

Capitolo 2

Analisi del dataset

In seguito al caricamento del dataset grazie all'ausilio della libreria *pandas*, si è provveduto all'analisi esplorativa dei dati al fine di poter analizzare quanto e se il dataset sia sbilanciato e di poter studiare la correlazione delle feature con il target, con il fine di poter stilare una classifica delle feature più importanti.

2.0.1 Distribuzione del target

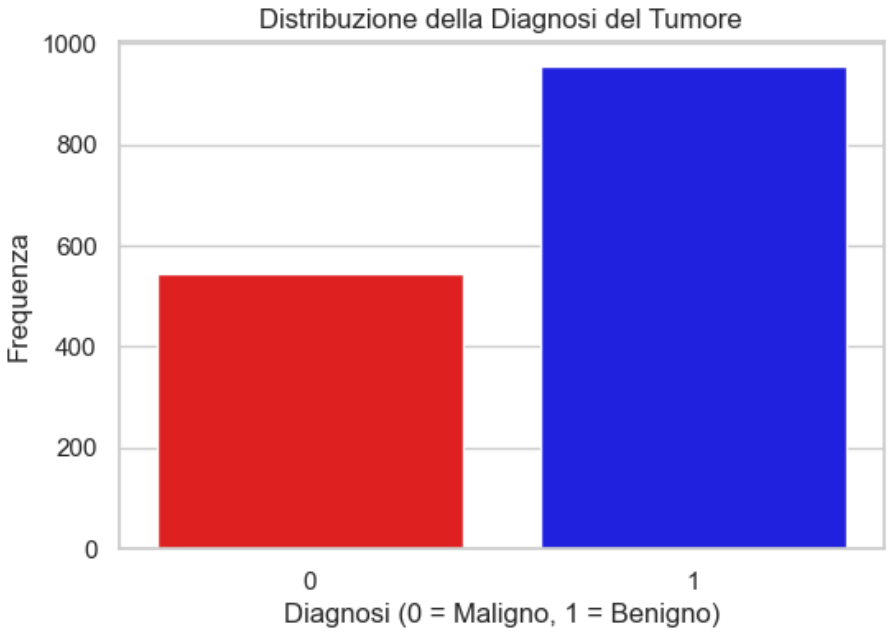

Figura 2.1: Distribuzione del target

Come si può notare vi è un lieve sbilanciamento tra le diagnosi di tumore maligno e benigno, con percentuali:

- Benigno: 63,7 %
- Maligno: 36,3 %

A fronte di questo, seppur non grave, sbilanciamento, possiamo affermare che l'accuratezza potrebbe non essere la metrica di valutazione migliore per questo modello, ma in questi è più opportuno focalizzarsi su metriche come Precision, Recall, F1-score e AUC, in quanto meno suscettibili allo sbilanciamento della distribuzione di valori del target.

2.0.2 Feature più importanti e correlazione

Lo studio della matrice di correlazione permette di studiare la correlazione che c'è tra coppie di feature del dataset, che può essere diretta, inversa o non correlata. In particolare nel grafico sottostante possiamo notare come le coppie che nella matrice presentano valori più chiari e sbiaditi abbiano una bassa correlazione tra loro o addirittura assente. Al contrario, quando si incontra un blu oppure rosso intenso ci si trova di fronte ad una forte correlazione, inversa (blu) oppure diretta (rossa).

Matrice di correlazione

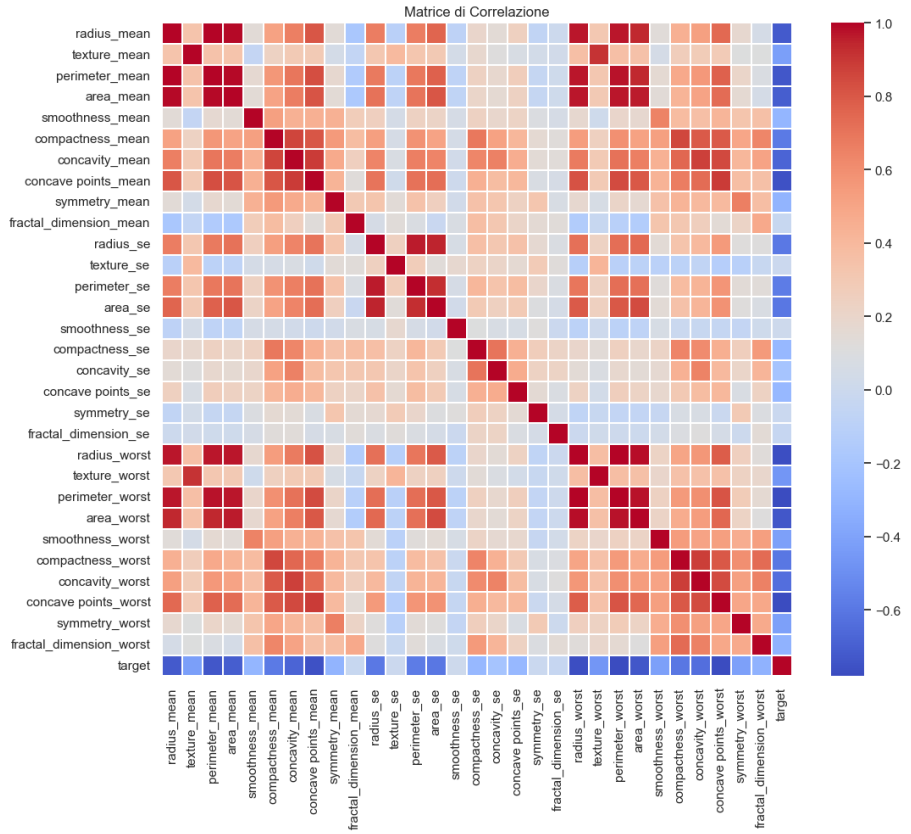

Figura 2.2: Correlazione tra feature

Da questo grafico si può dedurre come alcune variabili come 'area_worst" e "radius_worst" siano fortemente correlate tra loro, mentre "radius_mean" e "texture_mean" non lo siano. In seguito è importante anche capire quali siano le variabili con maggiore importanza: sapendo in precedenza di dover operare con un albero decisionale come classificatore, abbiamo potuto risalire alle variabili più importanti del nostro dataset utilizzando appunto un Decision Tree:

Correlazione delle feature con il target

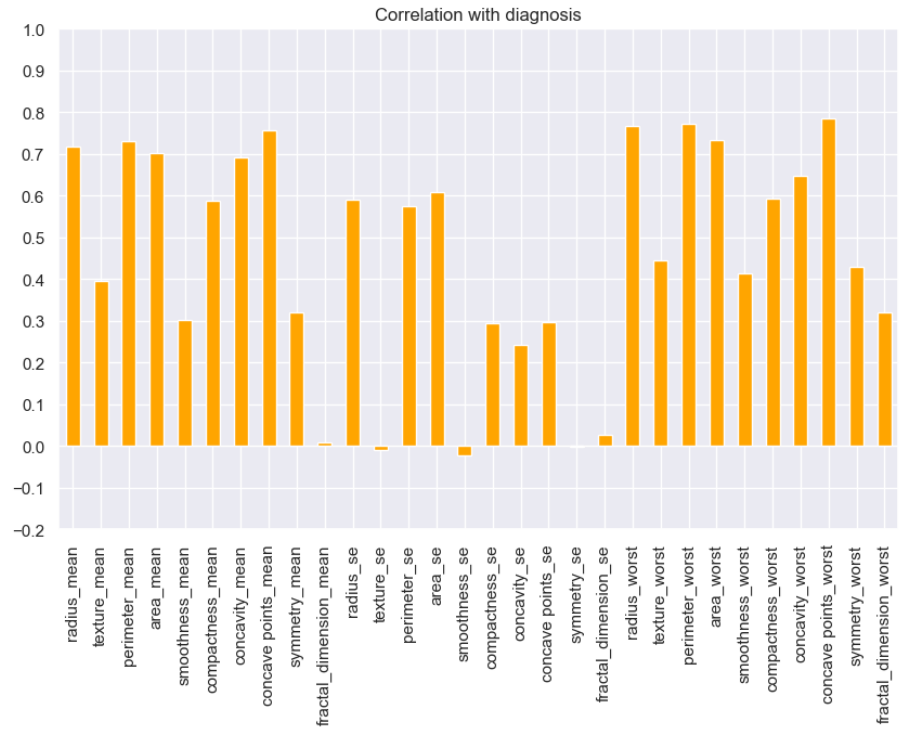

Figura 2.3: Correlazione delle feature con il target

Da questa rappresentazione possiamo notare come sono molteplici le feature con una alta correlazione con il target.

Feature più importanti

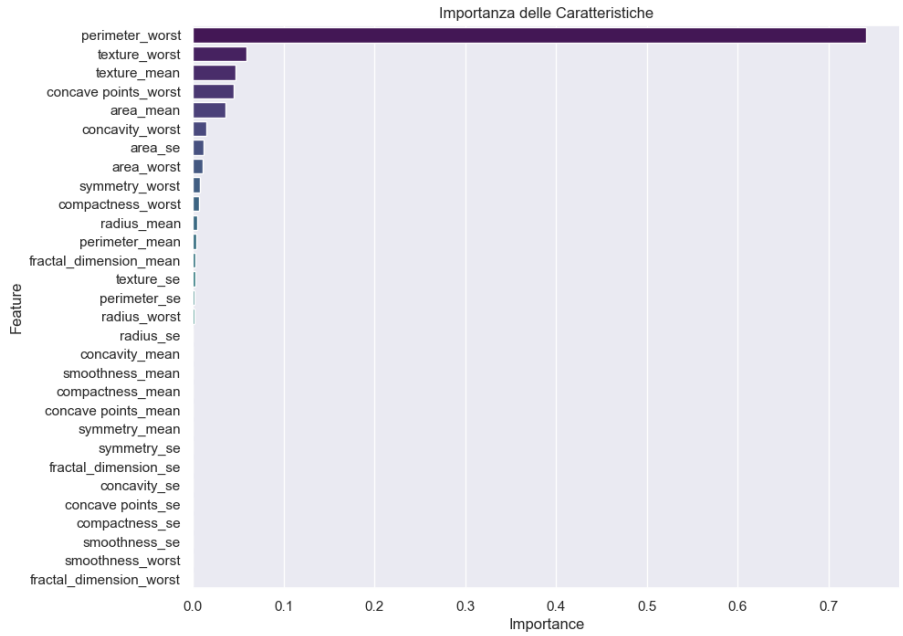

Figura 2.4: Feature più importanti Decision Tree

Grazie a questa rappresentazione grafica, si evince che le 5 feature più importanti nel dataset (non ancora separato in set di addestramento e set di test) siano in ordine: 'perimeter_worst' , 'texture_worst' , 'texture_mean' , 'concave points_worst' e 'area_mean' , con una netta maggiore importanza della prima feature. Questa maggiore importanza può essere dovuta alla forte correlazione della variabile con la malignità (e quindi con il target), rendendola particolarmente efficace nel separare i tumori maligni da quelli benigni. Inoltre la distribuzione delle feature in base al target è raccolta nei grafici sottostanti:

Distribuzione delle feature più importanti

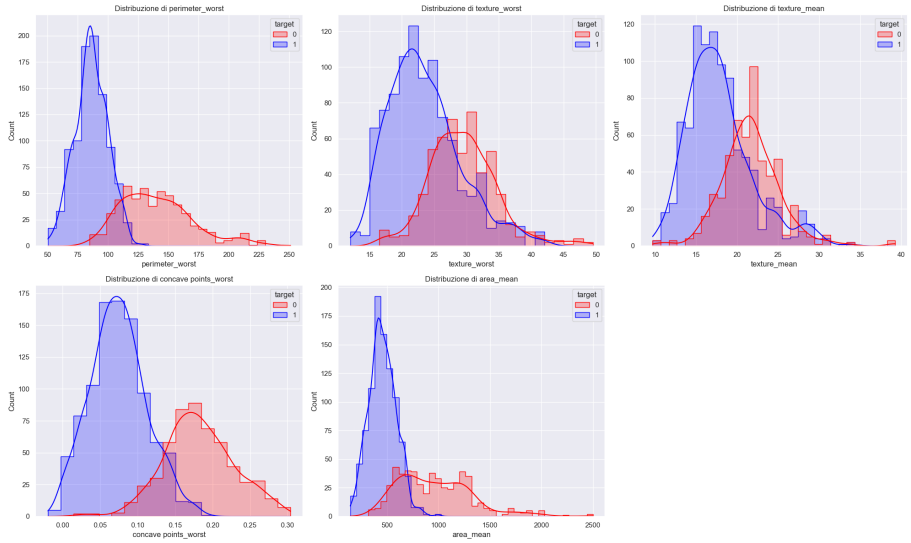

Figura 2.5: Distribuzione delle top 5 feature

Si può notare che per i cinque grafici illustrati sopra, vi siano tendenzialmente due zone ben distinte, in cui la distribuzione dei tumori benigni (grafico blu) ha maggiore distribuzione verso sinistra e quindi verso i valori inferiori relativi alla variabile, mentre la distribuzione dei tumori maligni(grafico rosso) sia in prevalenza orientata verso la zona destra e, di conseguenza, verso i valori maggiori.

Boxplot delle feature più importanti

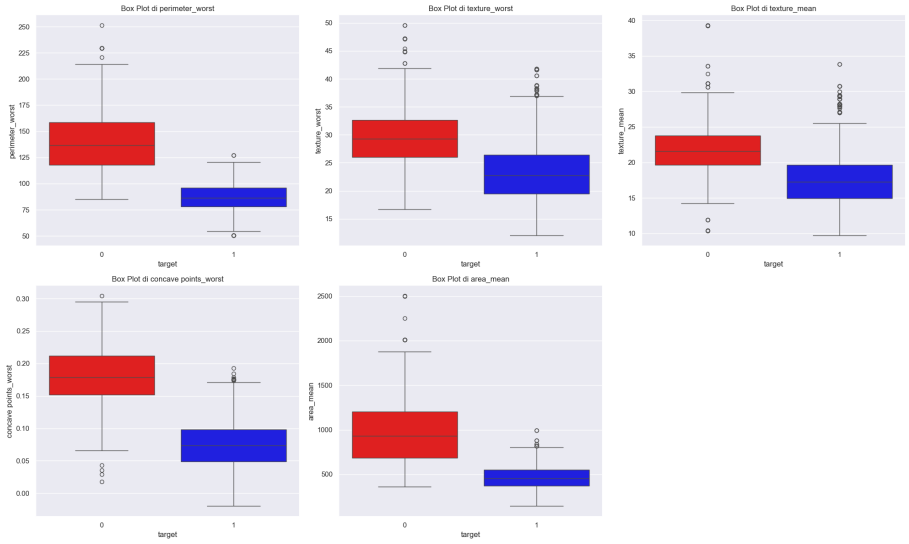

Figura 2.6: Boxplot delle 5 feature più importanti

Il boxplot è un diagramma utile per rappresentare graficamente la distribuzione di un insieme di dati, mettendo in evidenza caratteristiche chiave come la mediana, i quartili e gli outlier. Nel caso in oggetto sono stati identificati due boxplot per ogni feature, uno per ogni valore del target.

Elementi del box plot:

- Scatola: La scatola rappresenta la distribuzione centrale dei dati. La linea centrale all'interno della scatola rappresenta la mediana.
- Whisker: I baffi si estendono dalla scatola fino al massimo e al minimo valore dei dati che non sono considerati outlier.
- Quartili: il quartile superiore e il quartile inferiore sono i limiti della "scatola" e rappresentano il settantacinquesimo e il venticinquesimo percentile
- Outlier: I punti dati che si trovano al di fuori dei baffi sono considerati outlier e sono rappresentati da punti singoli.

Capitolo 3

Modello di classificazione: Decision Tree

Il decision tree è un algoritmo di machine learning supervisionato che costruisce un modello predittivo basandosi sull'importanza delle feature nel dataset, per generare delle regole decisionali, sulla base delle quali costruire una struttura ad albero, dove i nodi interni rappresentano le decisioni, mentre le foglie i valori delle classi. La capacità di gestire le variabili cliniche e la sua semplicità di lettura e comprensione, sono state fondamentali per la scelta di questo modello, abbinate alla robustezza che esso possiede anche quando si trova in situazioni in cui il dataset vede la distribuzione del target sbilanciata.

3.1 Split del dataset

Con lo scopo di addestrare il modello, è stata definita la percentuale di dati da utilizzare per l'addestramento: 70% , mentre la restante parte, il 30 %, appartiene al set di test del modello. Le dimensioni dei due set sono quindi di 1050 righe per l'addestramento e 450 per il test, con una distribuzione del target simile al dataset di partenza:

- **Training-set**: Classe 1 37,4% - Classe 0 62,6%
- **Test-set**: Classe 1 36,8% - Classe 0 63,2%

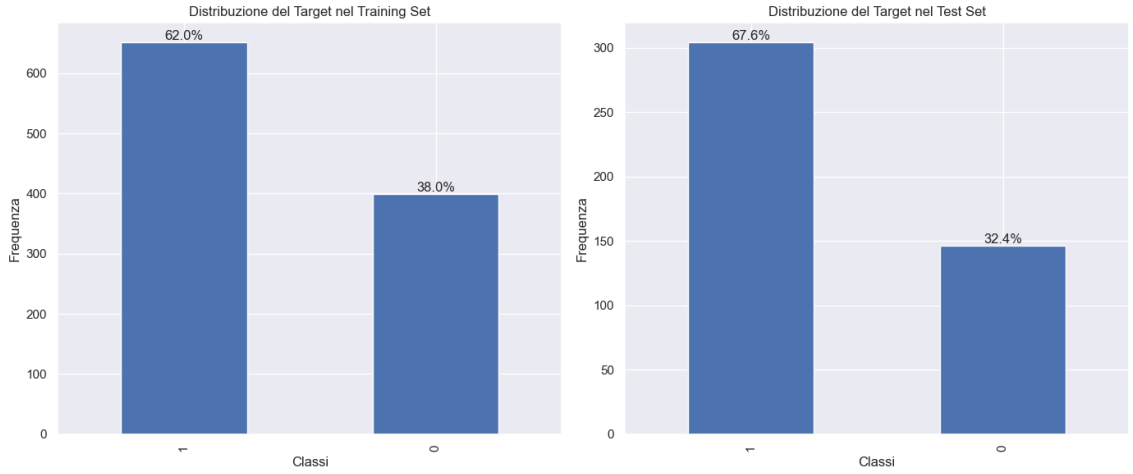

Figura 3.1: Distribuzione dei valori target all'interno del test set e del training set

3.2 Addestramento del modello e analisi

3.2.1 Modello

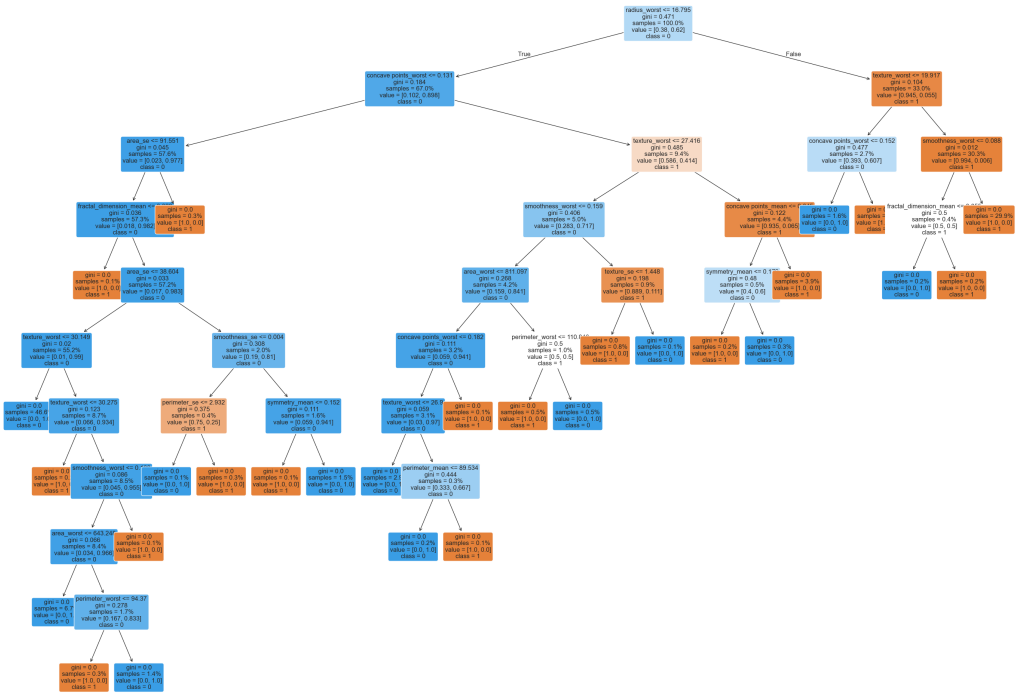

Figura 3.2: Decision Tree

3.2.2 Analisi delle prestazioni

In seguito all'addestramento del modello, avvenuto in un tempo di **0,003s**, sono stati calcolati i parametri di qualità del modello: Accuratezza, Precisione, Recall, F1_score e Curva roc con annessa AUC, con l'intento di paragonarli ai valori del modello addestrati con dati che presentano rumore.

- **Accuracy**: 96,22% - Il modello è in grado di classificare la quasi totalità dei campioni in modo corretto
- **Precision**: 97,04% - La percentuale di predizioni positive corrette
- **Recall**: 97,36% - Capacità del modello di identificare correttamente i casi positivi molto alta
- **F1_score**: 97,20% - Alta capacità di ridurre i falsi positivi ed identificare i positivi reali.

Il modello restituisce punteggi molto alti in tutte le metriche. Particolare attenzione, in questo caso, viene data al valore di Recall, perchè con un valore così alto è possibile minimizzare il numero di falsi negativi, che soprattutto in un contesto medico è molto importante che vengano evitati o minimizzati.

Viene adesso riportata anche la matrice di confusione del modello:

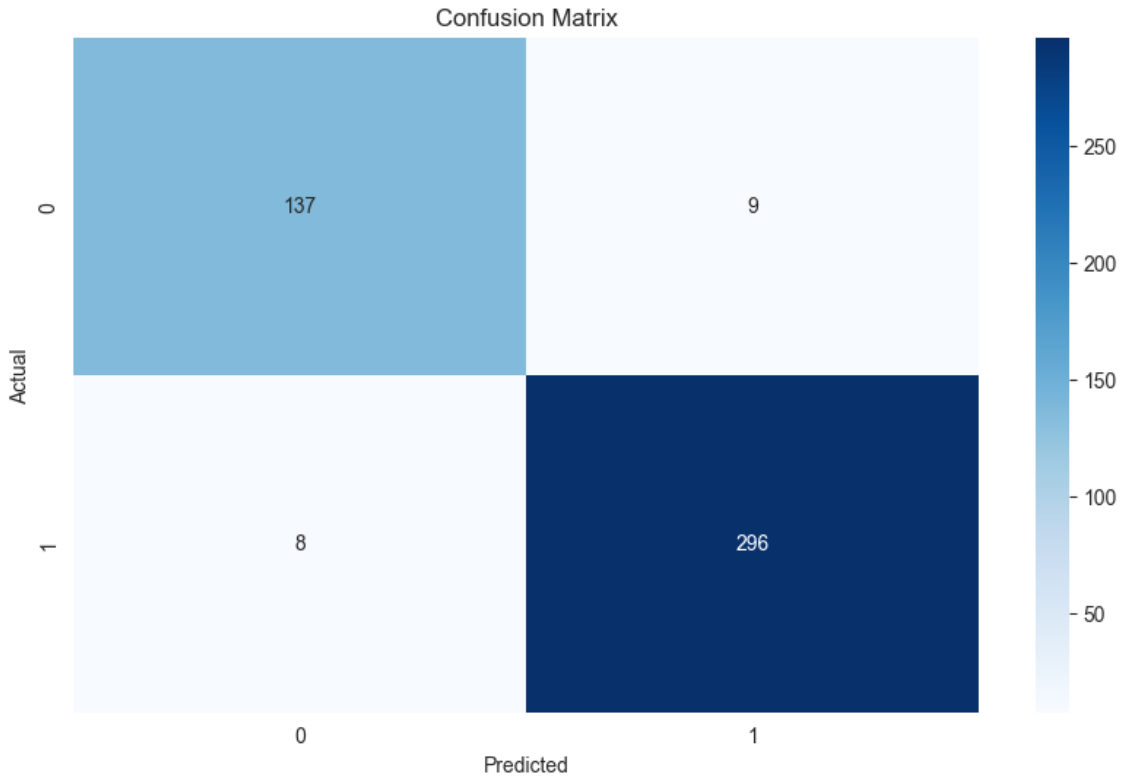

Figura 3.3: Matrice di confusione

Curva ROC

La curva ROC è uno strumento grafico che permette di valutare le prestazioni del modello, grazie alla valutazione della forma della curva e al valore dell'area che vi è sotto alla curva (AUC:Area Under the Curve). Un alto valore della AUC e il grafico che si avvicina all'angolo in alto a sinistra indicano che il modello ha buone capacità predittive, con un alto tasso di veri positivi ed un basso tasso di falsi positivi

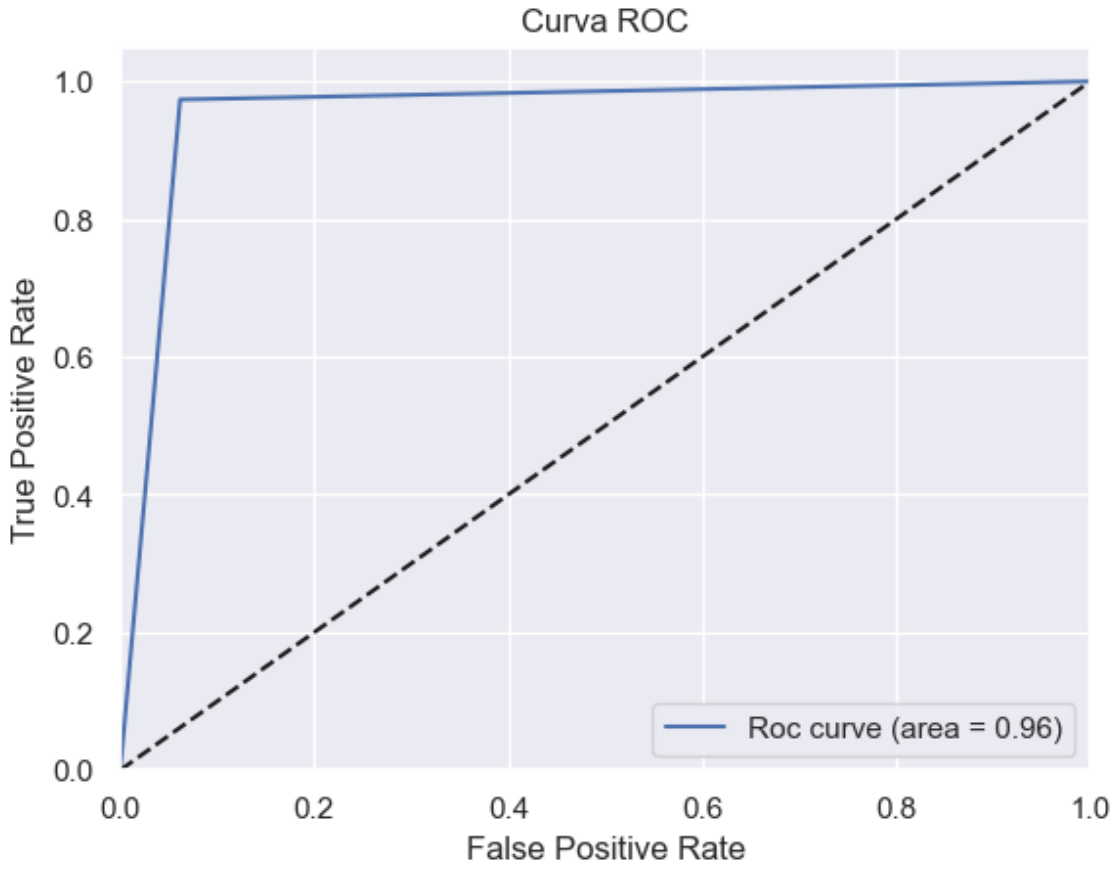

Figura 3.4: Curva ROC

Come si può notare, i valori della AUC sono molto elevati: 0,96, a fronte di un valore massimo di 1, mentre la curva si avvicina molto all'angolo in alto a sinistra del grafico, denotando così, una buona capacità di predizione.

K-fold cross validation

Con lo scopo di migliorare la precisione del calcolo delle metriche del modello, si è deciso di applicare la k-cross fold validation con un valore di k pari a 10. In questo modo i dati vengono in k sottoinsiemi e il modello viene addestrato sul sottoinsieme rimanente al fine di ottenere una valutazione più robusta che evita l'impatto di particolarità del set di addestramento. In seguito all'esecuzione del metodo i valori sono i seguenti:

- **Accuracy**: 95,90% - Il modello è in grado di classificare la quasi totalità dei campioni in modo corretto
- **Precision**: 97,40% - La percentuale di predizioni positive corrette
- **Recall**: 96,01% - Capacità del modello di identificare correttamente i casi positivi molto alta
- **F1_score**: 96,67% - Alta capacità di ridurre i falsi positivi ed identificare i positivi reali.

Paragonando i valori, si può notare un lieve calo per quanto riguarda accuratezza, precisione e f1 score, mentre un leggero miglioramento del parametro di precisione, indicando una buona precisione nelle predizioni.

3.2.3 Feature più importanti per il Decision Tree

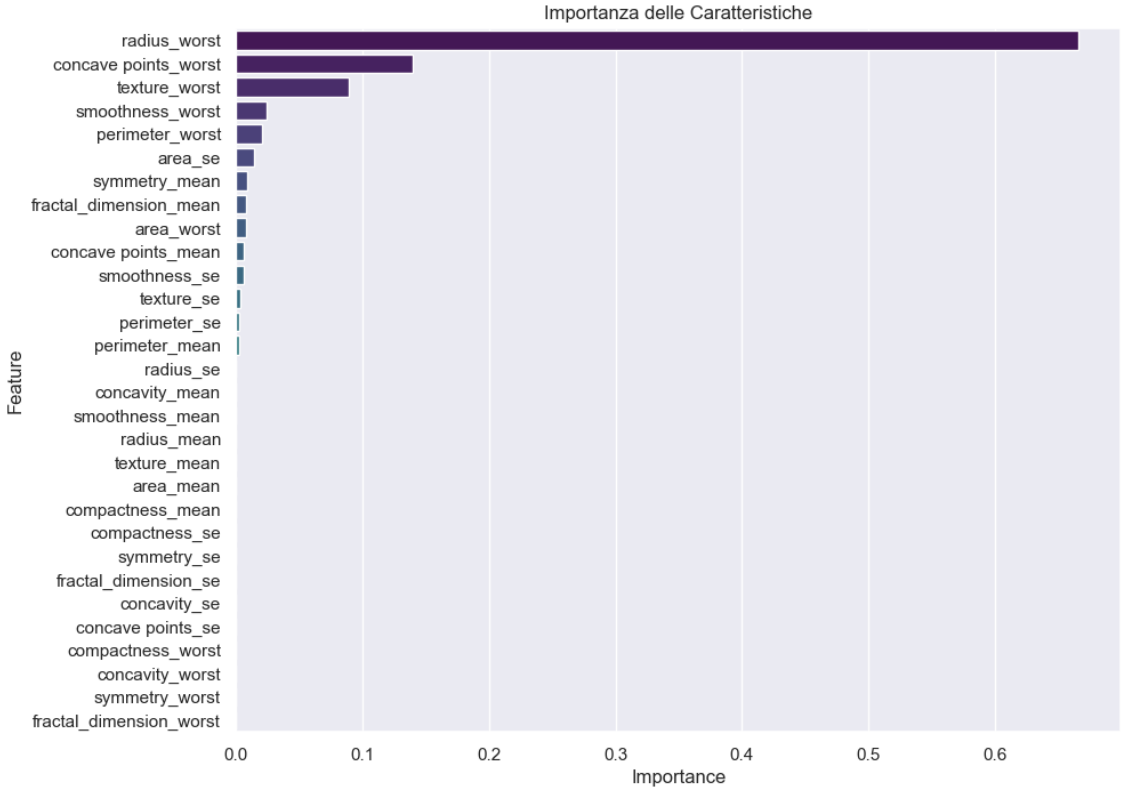

Figura 3.5: Feature più importanti per il decision tree sui dati di training

Vengono analizzate sul modello addestrato le feature più importanti, mostrando come per il modello addestrato sui dati di training le feature più importanti risultino **radius_worst, concave points_worst, texture_worst, smoothness_worst, perimeter_worst**. Risulta di grande importanza tenere traccia di queste informazioni per valutare come e se queste cambino nei passaggi successivi.

Anche in questo caso la feature 'radius_worst" ha una netta importanza per il modello, rispetto alle altre variabili, perchè, data la sua forte correlazione con il target, permette di separare efficacemente i tumori maligni da quelli benigni.

Capitolo 4

Rumore nel dataset

Come precedentemente introdotto, l'obiettivo di questo studio è quello di analizzare il comportamento del modello di classificazione e i cambiamenti che esso può subire a fronte dell'aggiunta di rumore all'interno dei dati di addestramento. A tale scopo si è deciso di introdurre i seguenti tre tipi di rumore all'interno del set di dati di addestramento: **Outliers**, **Valori nulli**, **Righe duplicate**.

Per introdurre il rumore nei dati, sono state create copie del set di addestramento del modello. Per ogni tipo e per ogni percentuale di rumore, è stata realizzata una copia separata del dataset. Questo approccio ha permesso di applicare il rumore in modo controllato e progressivo, facilitando un'analisi dettagliata degli effetti di ciascun tipo e livello di rumore sulle prestazioni del modello. La decisione sulla scelta delle percentuali è ricaduta su un approccio incrementale ispirato ad un approccio esponenziale: partendo dall'aggiunta dei dati alterati in bassissime percentuali vicine vicine tra loro, fino ad arrivare a valori alti, con i seguenti valori inseriti nell'array *percentuali_rumore*: 1% , 2% , 3% , 50% , 100% In questo modo si vuole catturare la panoramica sull'andamento dei comportamenti del modello sin dalla minima introduzione dei valori alterati, per vedere come ne viene influenzato.

4.1 Valori nulli

In questa sezione si analizzeranno le prestazioni del modello in seguito all'alterazione delle feature più importanti, che verranno rese nulle seguendo le percentuali pre-indicate.

4.1.1 Rumore 1%

In seguito alla manipolazione dell'1%, i valori riguardanti le metriche di valutazione del modello sono stati i seguenti:

Parametri di valutazione

- **Accuracy**: 96,00%
- **Precision**: 96,01%
- **Recall**: 96,00%
- **F1_score**: 96,00%
- **AUC**: 95,61%

Si nota, dunque un leggero calo delle prestazioni per quanto riguarda ogni metrica se paragonate al dataset privo di rumore: in particolare Precision, Recall e F1 Score che hanno subito la diminuzione di più di un punto percentuale.

Matrice di confusione

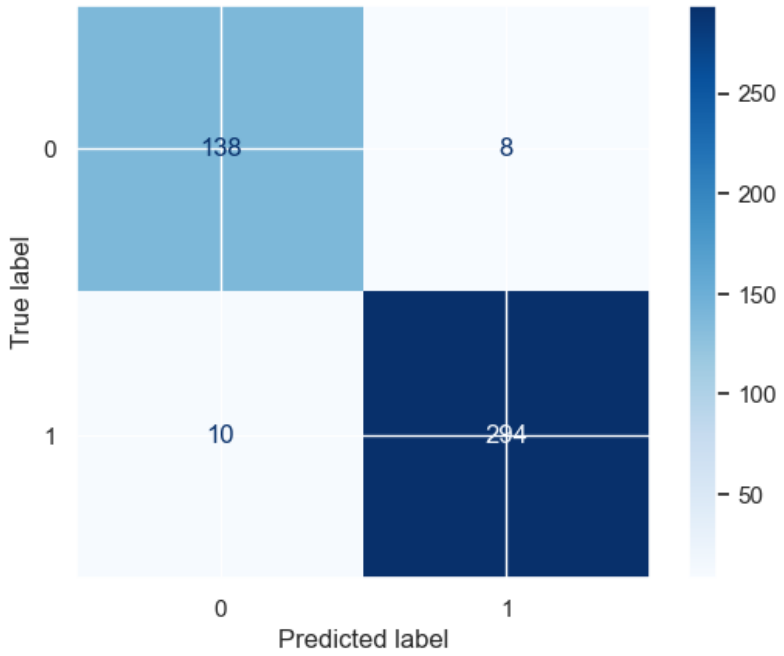

Figura 4.1: Matrice di confusione valori nulli 1%

4.1.2 Rumore 2%

All'aumentare di un punto percentuale del rumore, i parametri di valutazione hanno subito il seguente cambiamento:

Parametri di valutazione

- **Accuracy**: 96,89%
- **Precision**: 96,88%
- **Recall**: 96,89%
- **F1_score**: 96,88%
- **AUC**: 96,27%

Con la seguente matrice di confusione:

Matrice di confusione

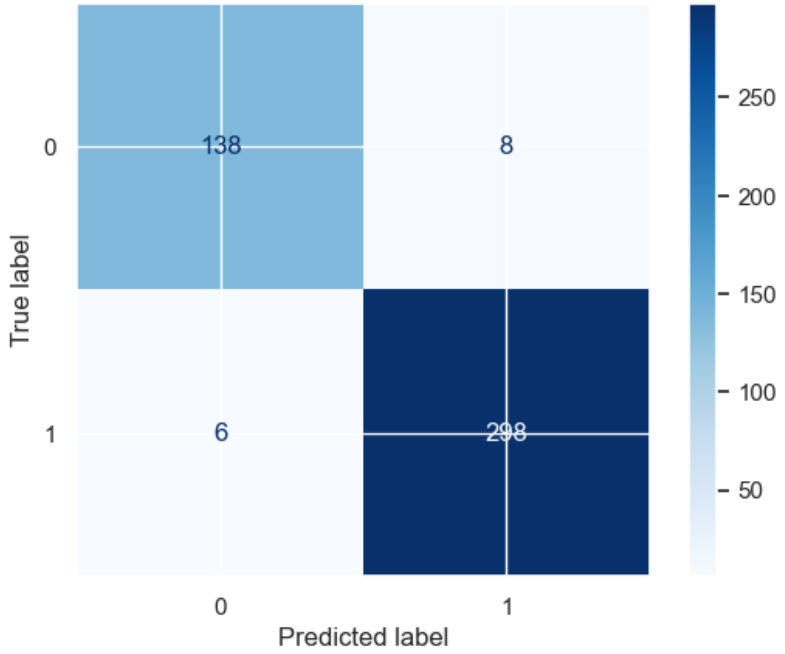

Figura 4.2: Matrice di confusione valori nulli 2%

Dai dati si evince un leggero miglioramento nelle performance nel momento in cui il rumore aumenta di un punto percentuale, con una diminuzione dei casi di falsi negativi che si può notare dalla matrice di confusione.

4.1.3 Rumore 3%

Al 3% di rumore i valori di Accuratezza, Precisione, Recall, F1 score e AUC sono i seguenti:

Parametri di valutazione

- **Accuracy**: 96,22%
- **Precision**: 96,21%
- **Recall**: 96,21%
- **F1_score**: 96,22%
- **AUC**: 96,22%

Matrice di confusione

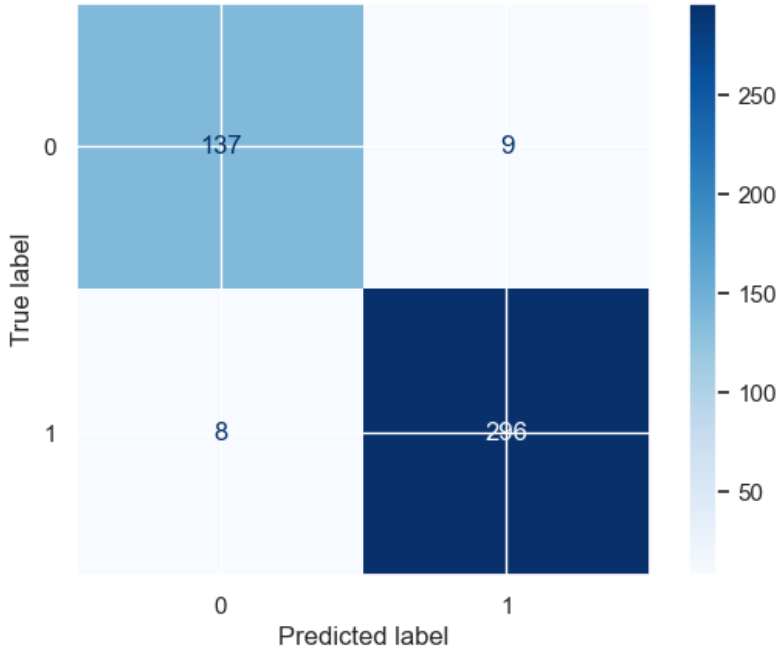

Figura 4.3: Matrice di confusione valori nulli 3%

In questo caso, si nota un nuovo lieve abbassamento delle prestazioni e un aumento, seppure leggero dei falsi positivi e dei falsi negativi

4.1.4 Rumore 50%

In questa sezione il rumore viene aumentato in modo significativo. Le feature più importanti in questo caso vengono sporcate per la metà delle istanze presenti nei dati.

Parametri di valutazione

- **Accuracy**: 97,7%
- **Precision**: 97,7%
- **Recall**: 97,7%
- **F1_score**: 97,7%
- **AUC**: 97,4%

Matrice di confusione

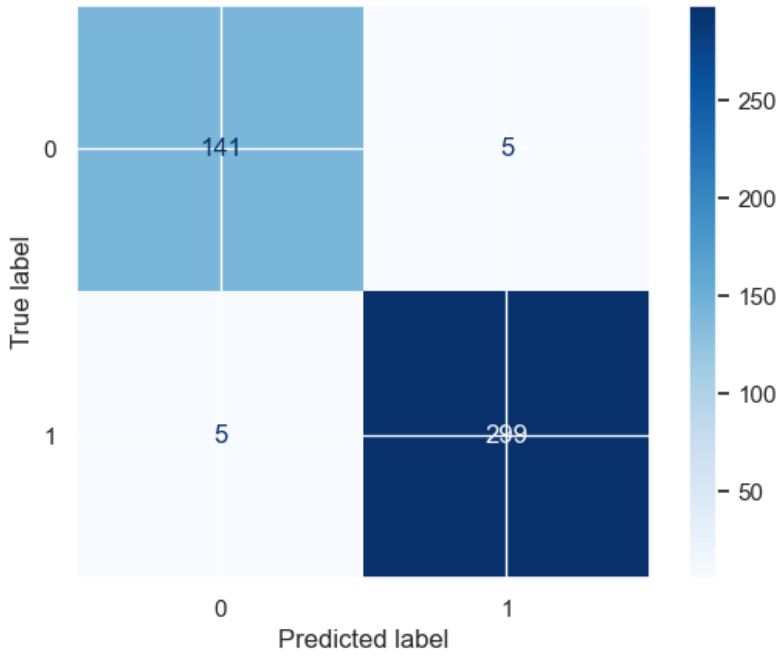

Figura 4.4: Matrice di confusione valori nulli 50%

Seppur le feature più importanti del modello siano state significativamente compromesse, le prestazioni del modello non sono peggiorate, ma al contrario. Ciò suggerisce che ci può essere stato qualche cambiamento interno al modello che necessita un'analisi più approfondita.

4.1.5 Rumore 100%

A questo passo le feature più importanti sono per la totalità prive di valore informativo, dato che sono state alterate e rese NaN.

Parametri di valutazione

- **Accuracy**: 97,33%
- **Precision**: 97,33%
- **Recall**: 97,33%
- **F1_score**: 97,32%
- **AUC**: 96,60%

Matrice di diffusione

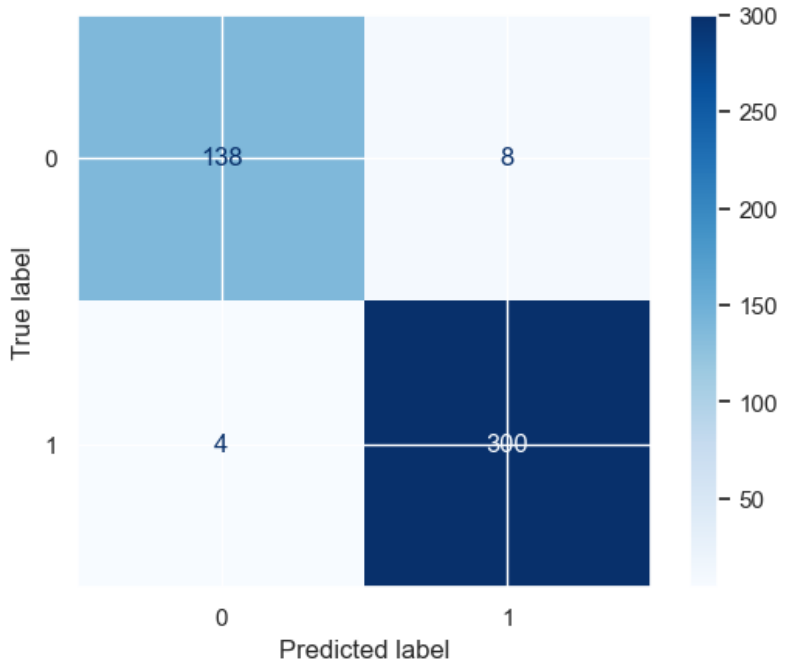

Figura 4.5: Matrice di confusione valori nulli 100%

4.1.6 Evoluzione del modello

Andamento dei parametri di qualità

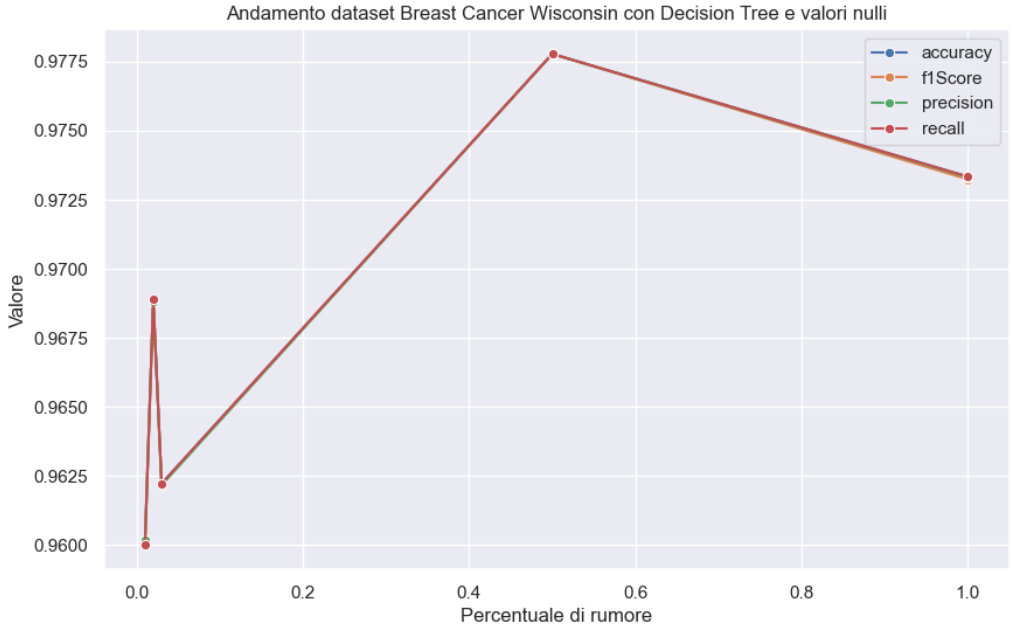

Figura 4.6: Metriche all'aumentare del rumore

Dal grafico che mostra l'andamento delle metriche all'aumentare delle righe nulle inserite si nota un aumento inaspettato delle prestazioni al passaggio tra l'1 e il 2 percento di rumore. In seguito all'aumento di un nuovo punto percentuale si osserva una nuova diminuzione, per poi andare a stabilizzare i valori verso valori nuovamente elevati, che sono in linea con quelli che si avevano prima dell'introduzione del rumore.

Per capire le motivazioni di questo fenomeno è necessaria una valutazione più nel dettaglio riguardante il comportamento del modello, per ciò che concerne le sue feature più importanti. A tale scopo si è deciso di studiare il cambiamento delle feature più significative in modo specifico. Si è optato per il seguente approccio, con l'introduzione del rumore a:

1. feature più importante per il modello
2. due tra le feature più importanti per il modello
3. le 5 feature più importanti per il modello

In tutti questi casi sono state inserite delle percentuali di rumore tali che evidenziassero il cambiamento del modello, giustificandone un comportamento differente.
(In giallo sono evidenziate le barre relative alle feature selezionate per l'osservazione)

4.1.7 Analisi specifica dell'impatto delle feature sul modello

La situazione di partenza (senza l'aggiunta di rumore) riguardante le feature più importanti era la seguente:

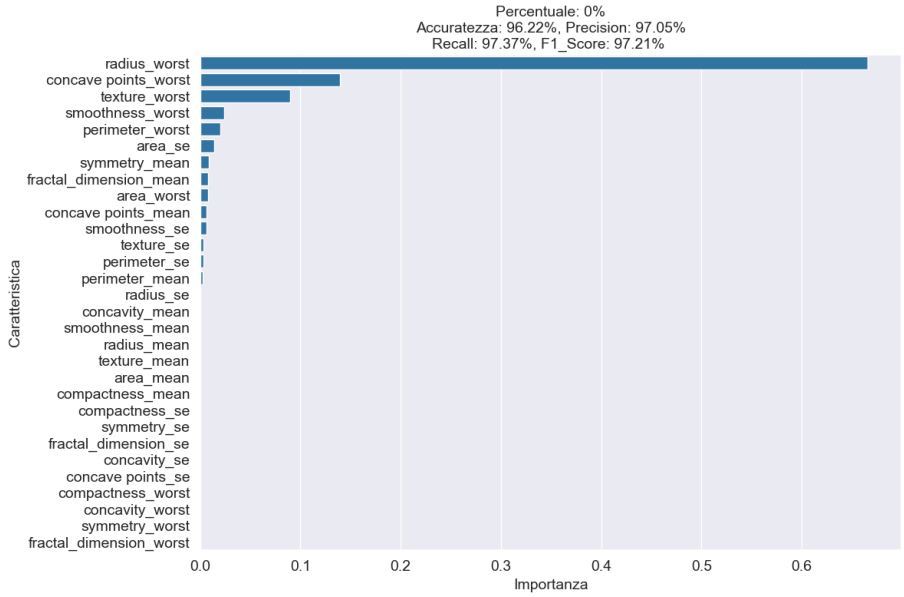

Figura 4.7: Feature più importanti per il decision tree sui dati di training

Aggiunta di rumore a radius_worst

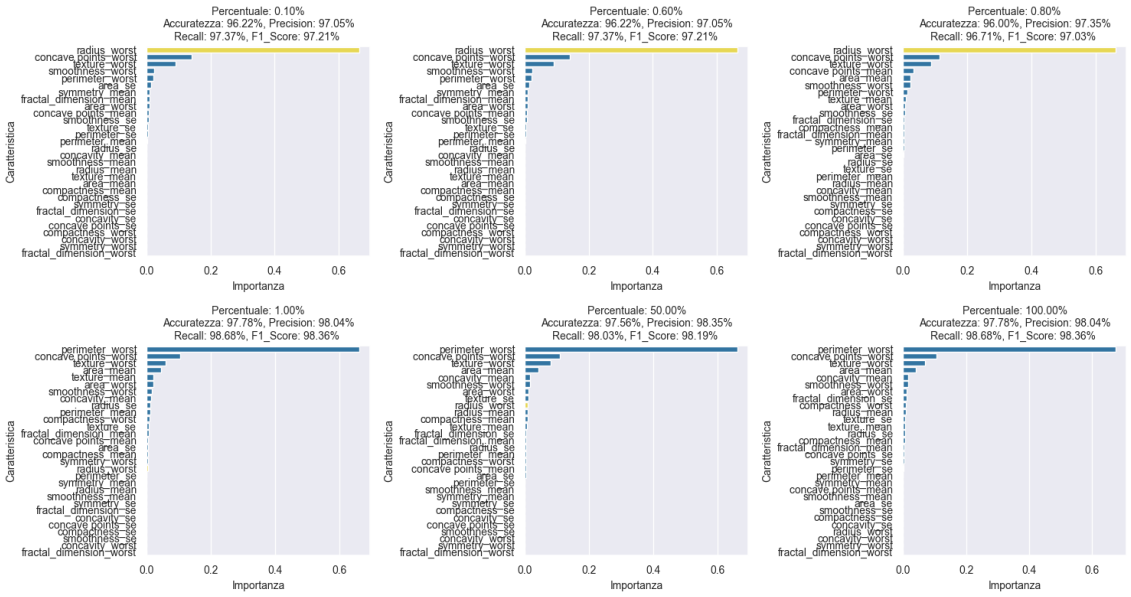

Figura 4.8: Cambiamento con rumore in radius_worst

Le percentuali di rumore inserite sono: 0.1%, 0.6%, 0.8%, 1%, 50%, 100%, possiamo notare che, man mano che aumenta il rumore, la feature "radius_worst" mantiene una certa rilevanza in termini di importanza nelle prime percentuali

di rumore (0.1%, 0.6%, 0.8%), evidenziando la sua robustezza e contributo significativo alla classificazione. Tuttavia, a partire dall'1% di rumore, la sua importanza inizia a diminuire drasticamente, mentre altre feature come perimeter_worst e concave_points_worst guadagnano centralità.
Al 100% di rumore, "radius_worst" sembra avere un peso minore, indicando che con rumore elevato, il modello dipende meno da questa feature, nonostante il mantenimento di buone metriche di performance come accuratezza, precisione e recall. Questo suggerisce che il rumore riduce la capacità del modello di sfruttare appieno questa variabile inizialmente rilevante.

Aggiunta di rumore a radius_worst e perimeter_worst

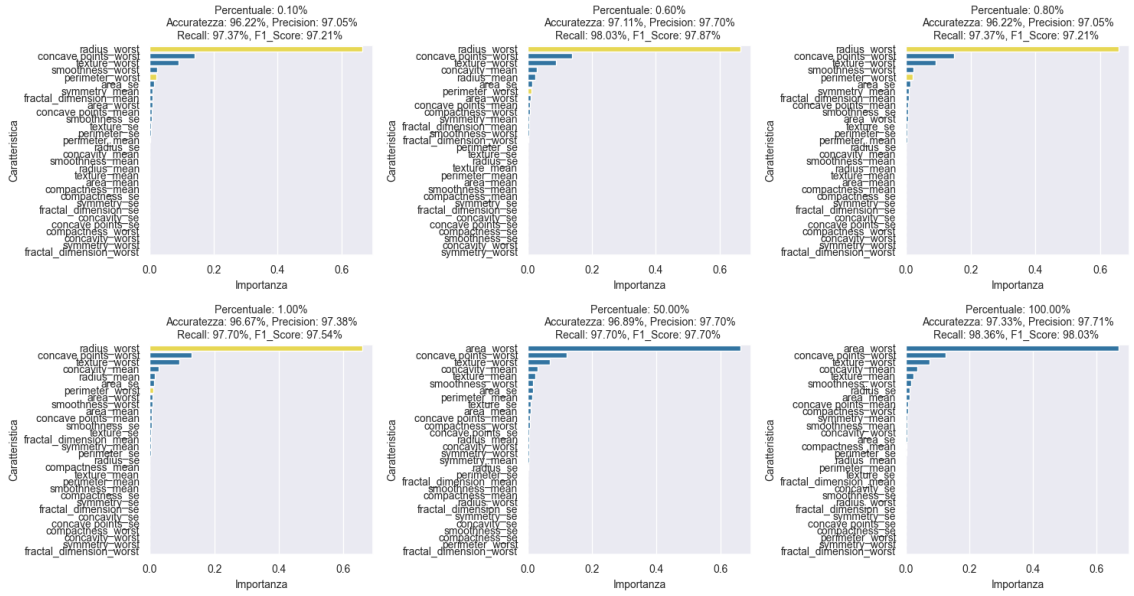

Figura 4.9: Cambiamento con rumore in radius_worst e perimeter_worst

Dai grafici si evince un comportamento simile a quello precedentemente studiato, ma con la differenza che a 100% di rumore su entrambe le feature l'algoritmo inizia a calare di poco l'accuratezza. Questo permette di poter concludere nonostante l'algoritmo riesce a gestire un rumore elevato su due feature a lungo andare più feature sporchiamo e più il modello degrada.
Di seguito lo studio sull'andamento nel caso di rumore su più feature del dataset.

Aggiunta di rumore a 10 feature

Le feature che hanno ricevuto l'alterazione sono: 'radius_worst', 'perimeter_worst', 'area_worst', 'concave points_worst', 'concave points_mean', 'concavity_mean', 'area_mean', 'perimeter_mean','radius_mean', 'area_se'

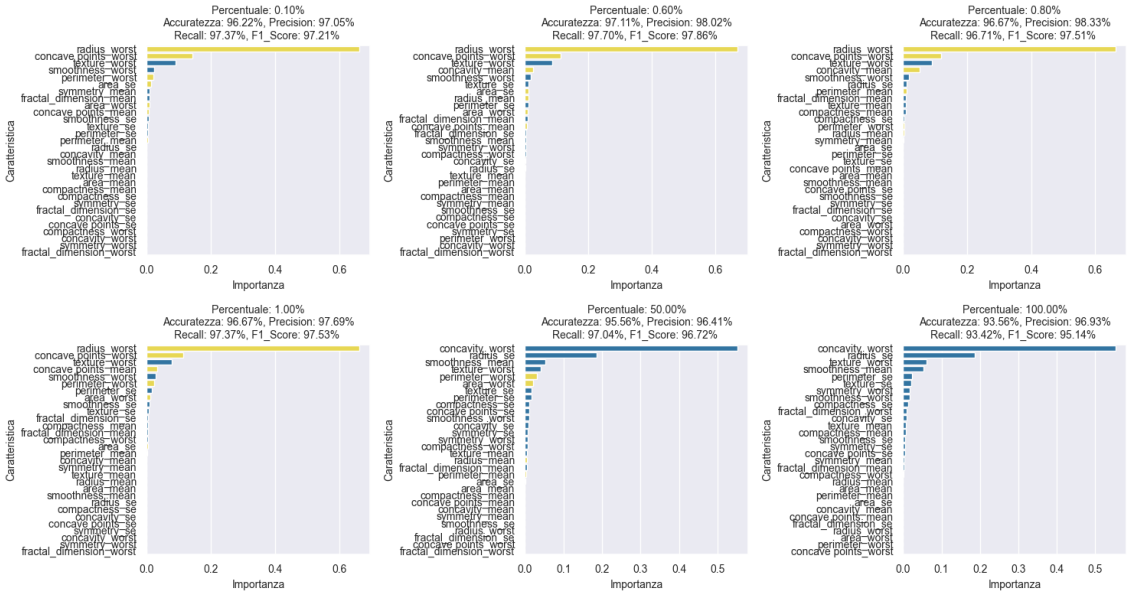

Figura 4.10: Cambiamento con rumore in 10 feature

Si osserva che nonostante il decision tree sia un algoritmo molto robusto e consolidato per la classificazione con il numero di feature che sono state sporcate (pari a 10) l'accuratezza del modello è calata in modo significativo per una qualsiasi percentuale di rumore al di sopra dell'1%, cosa prevedibile, ma ci fa capire l'importanza di un dataset pulito per l'addestramento di un modello.

Aggiunta di rumore a tutte le feature del dataset

In quest'ultima sezione del rumore NaN andiamo adesso, come esperimento, ad aggiungere del rumore progressivo a **tutte** le feature del dataset per vedere come si comporta il modello.

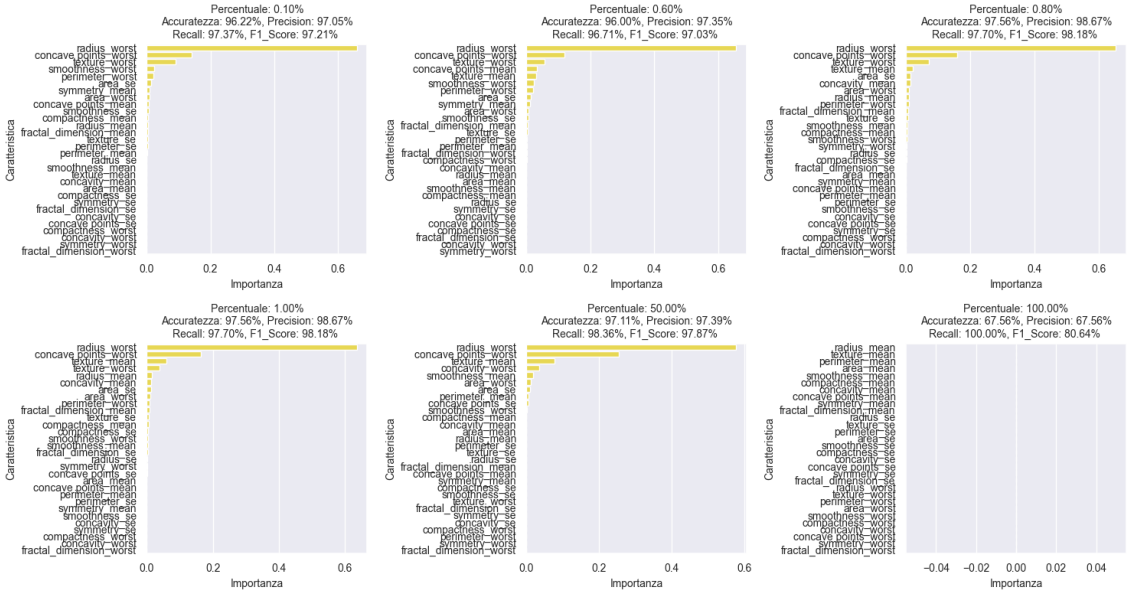

Figura 4.11: Cambiamento con rumore su tutte le feature

Le percentuali utilizzate sono uguali a quelle utilizzate precedentemente per calcolare gli altri grafici. Possiamo notare che l'accuratezza del modello fino al 50% è sorprendentemente robusta, mentre come ci si poteva aspettare al 100% il modello ha un calo totale delle prestazioni e diventa inaffidabile. Notiamo, inoltre, come al 100% il modello abbia (giustamente) smesso di considerare importanti le feature o comunque di avere una preferenza.
In conclusione questo ci fa capire oltre all'importanza di un dataset pulito, specialmente se consideriamo l'ambito del dataset scelto da noi, ovvero, l'ambito medico, ma anche quanto è robusto l'algoritmo Decision Tree Classifier che nonostante il 50% delle feature siano valori nulli (che ricordiamo esser inseriti casualmente all'interno del dataset) riesce comunque a predirre correttamente la maggior parte dei record.

4.2 Outliers

In questa sezione si analizzeranno le prestazioni del modello in seguito all'alterazione delle feature più importanti, che verranno rese outliers seguendo le percentuali pre-indicate.

4.2.1 Rumore 1%

In seguito alla manipolazione dell'1%, i valori riguardanti le metriche di valutazione del modello sono stati i seguenti:

Parametri di valutazione

- **Accuracy**: 96,22%
- **Precision**: 96,22%
- **Recall**: 96,00%
- **F1_score**: 96,00%
- **AUC**: 95,78%

Si nota, dunque un leggero calo delle prestazioni per quanto riguarda ogni metrica se paragonate al dataset privo di rumore.

Matrice di confusione

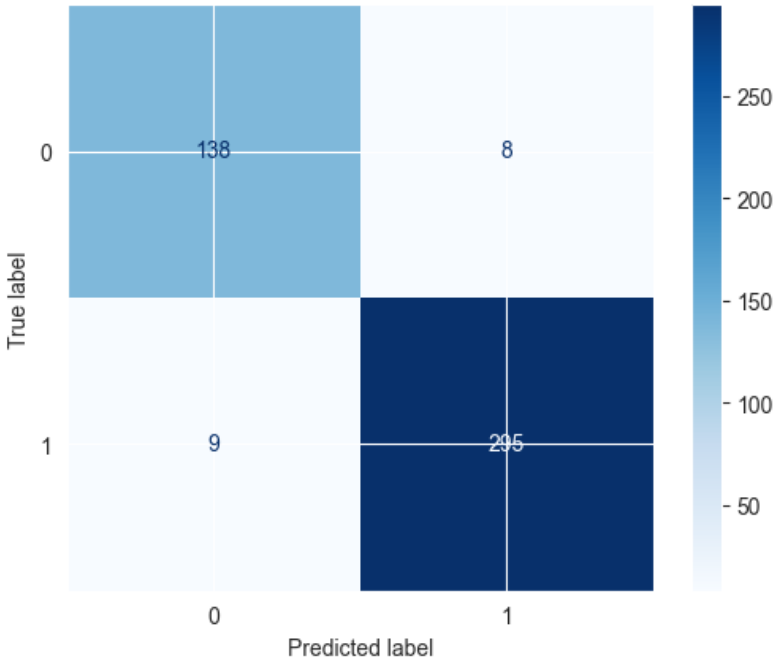

Figura 4.12: Matrice di confusione outliers all'1%

Confronto tra dataset pulito e sporcato

Andiamo adesso a mostrare i box plot per l'1% di outliers, i box plot sono particolarmente utili in questo caso in quanto permettono di guardare visivamente la quantità di outliers che aumenta e quindi se l'introduzione del rumore è avvenuta correttamente.
In questo caso andiamo a confrontare il dataset pulito "*clean*" e quello sporcato "*noisy*".

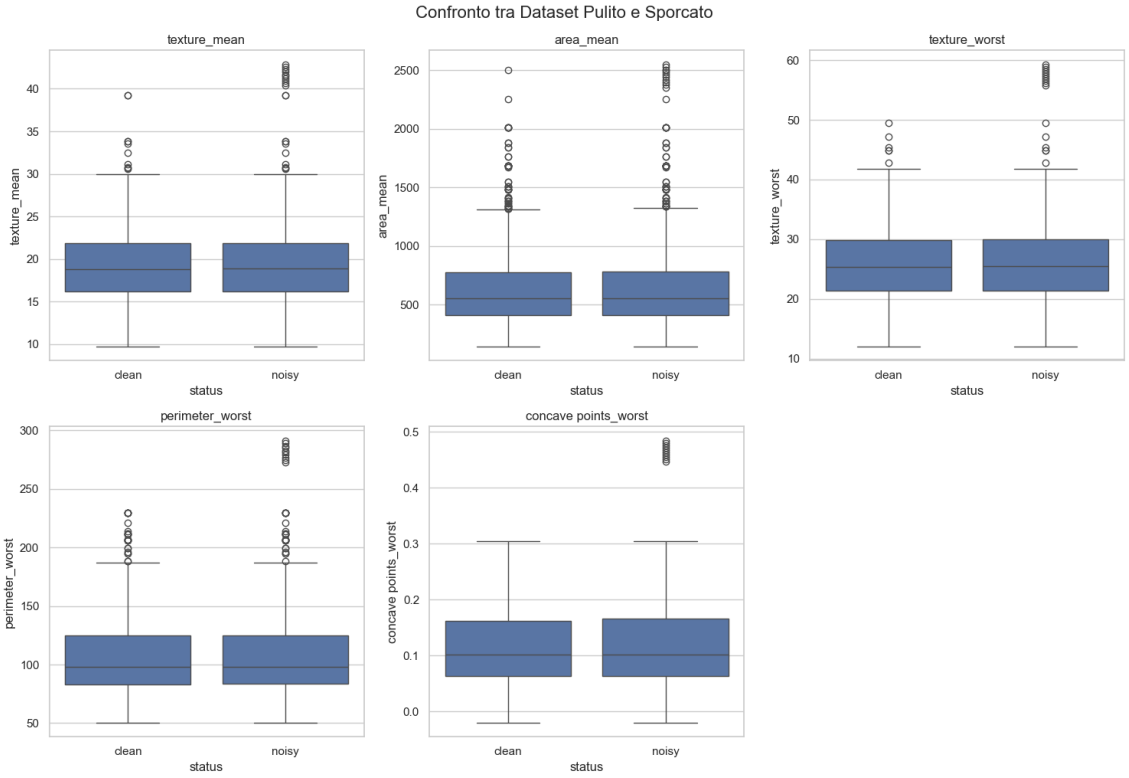

Figura 4.13: Box plot con outliers all'1%

4.2.2 Rumore 2%

All'aumentare di un punto percentuale del rumore, i parametri di valutazione hanno subito il seguente cambiamento:

Parametri di valutazione

- **Accuracy**: 97.11%
- **Precision**: 97.10%
- **Recall**: 97.11%
- **F1_score**: 97.10%
- **AUC**: 96.43%

Si nota, dunque un leggero aumento delle prestazioni per quanto riguarda ogni metrica in paragone alla situazione di rumore precedente.

Matrice di confusione

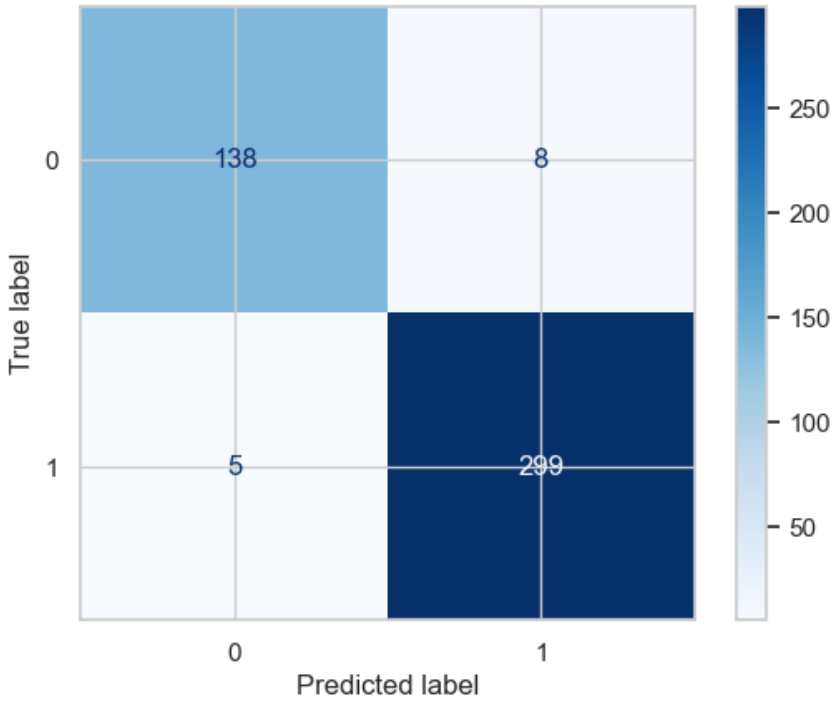

Figura 4.14: Matrice di confusione outliers al 2%

Confronto tra dataset pulito e sporcato

Verranno ora mostrati i box plot con rumore al 2%

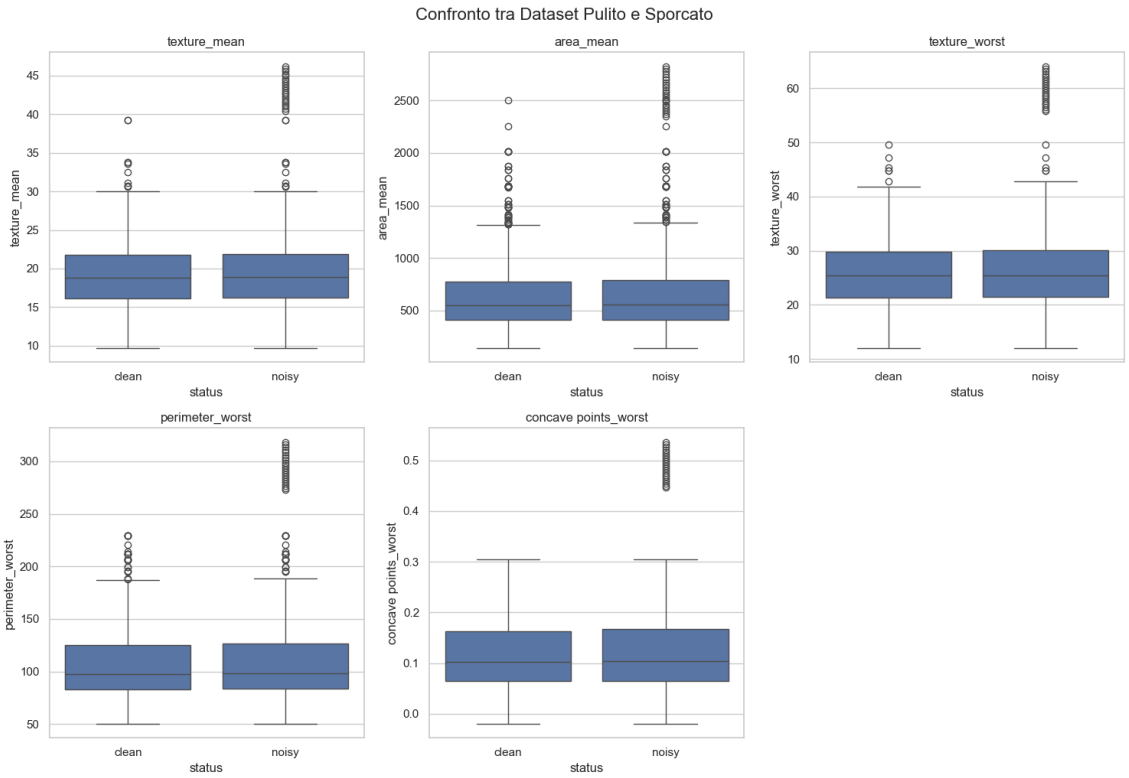

Figura 4.15: Box plot con outliers al 2%

4.2.3 Rumore 3%

All'aumentare di un punto percentuale del rumore, i parametri di valutazione hanno subito il seguente cambiamento:

Parametri di valutazione

- **Accuracy**: 96.44%
- **Precision**: 96.51%
- **Recall**: 96.44%
- **F1_score**: 96.46%
- **AUC**: 96.47%

Si nota, dunque una leggera diminuzione delle prestazioni per quanto riguarda ogni metrica in paragone alla situazione di rumore precedente.

Matrice di confusione

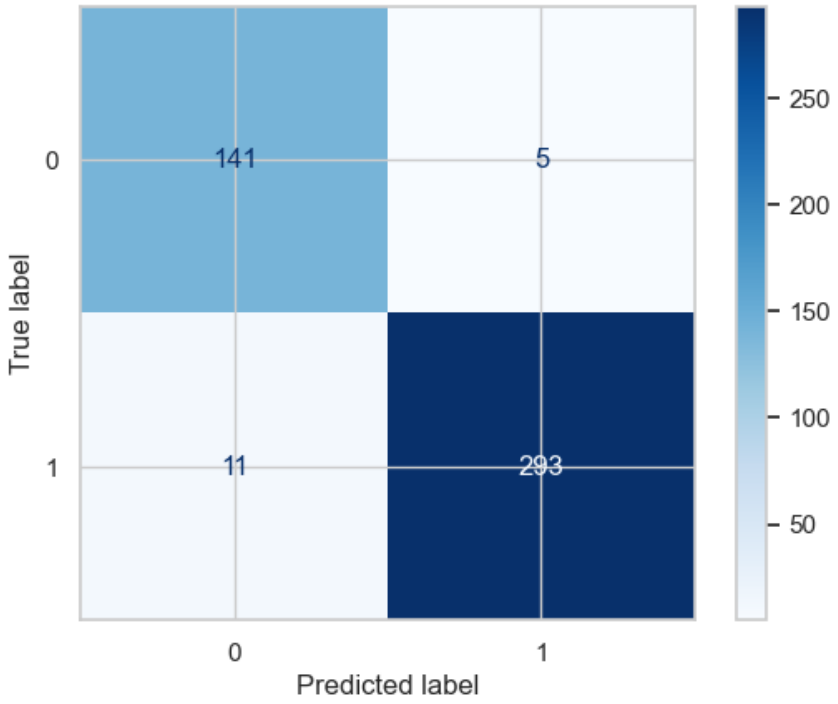

Figura 4.16: Matrice di confusione outliers al 3%

Confronto tra dataset pulito e sporcato

Verranno ora mostrati i box plot con rumore al 3%

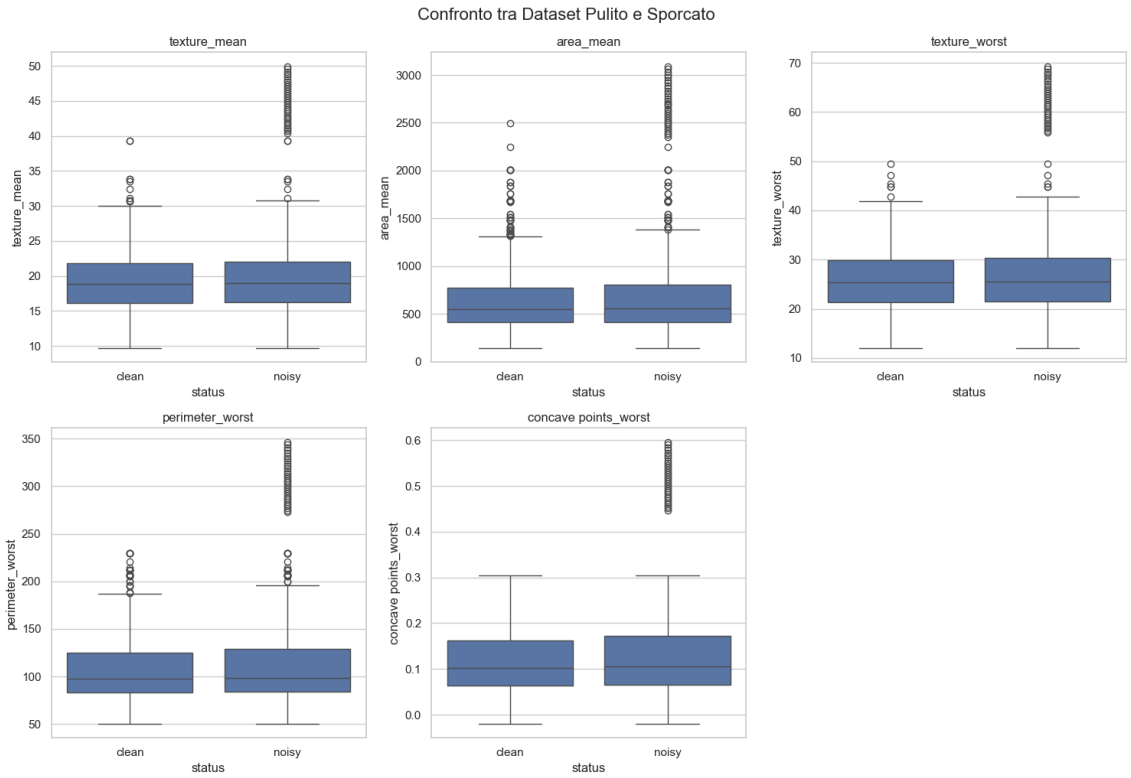

Figura 4.17: Box plot con outliers al 3%

4.2.4 Rumore 50%

Aumentando il rumore a 50 punti percentuale, i parametri di valutazione hanno subito il seguente cambiamento:

Parametri di valutazione

- **Accuracy**: 97.33%
- **Precision**: 97.32%
- **Recall**: 97.33%
- **F1_score**: 97.32%
- **AUC**: 96.78%

Si nota, dunque un leggero aumento delle prestazioni per quanto riguarda ogni metrica in paragone alla situazione di rumore precedente.

Matrice di confusione

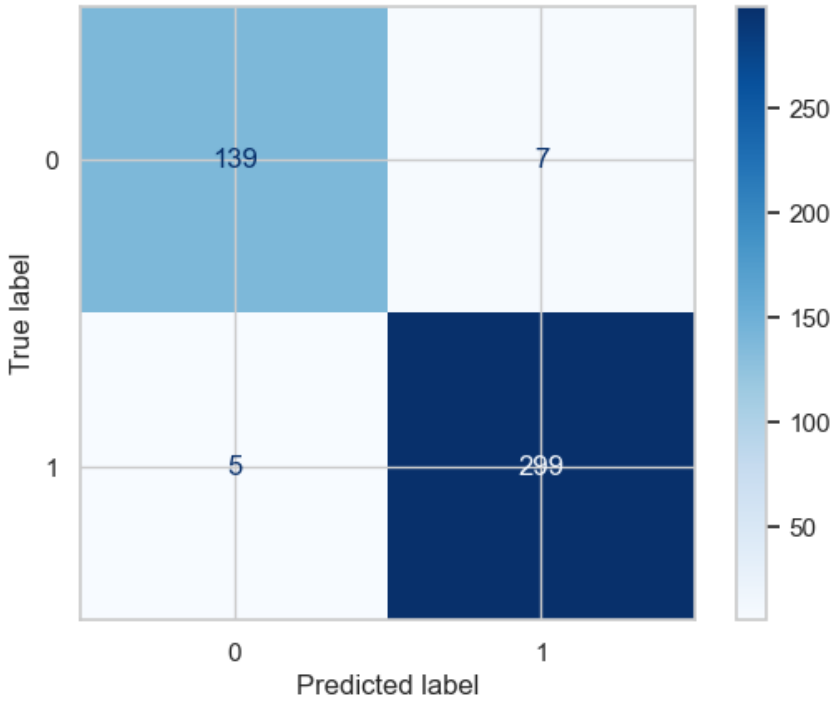

Figura 4.18: Matrice di confusione outliers al 50%

Confronto tra dataset pulito e sporcato

Verranno ora mostrati i box plot con rumore al 50%

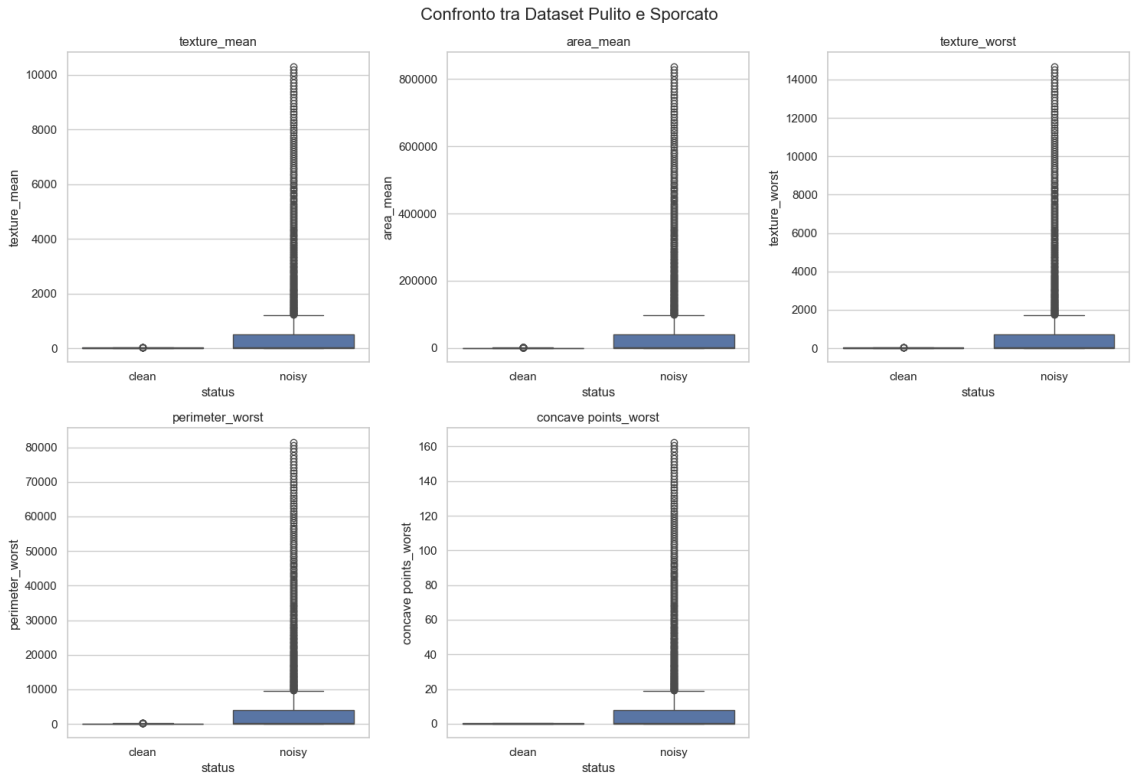

Figura 4.19: Box plot con outliers al 50%

4.2.5 Rumore 100%

Aumentando il rumore a 100 punti percentuale, i parametri di valutazione hanno subito il seguente cambiamento:

Parametri di valutazione

- **Accuracy**: 97.33%
- **Precision**: 97.32%
- **Recall**: 97.33%
- **F1_score**: 97.32%
- **AUC**: 96.78%

Possiamo notare che i valori sono rimasti pressochè identici, facendoci intuire che già al 50% se non prima il modello abbiamo scartato completamente le cinque feature che prima riteneva le più importanti.

Matrice di confusione

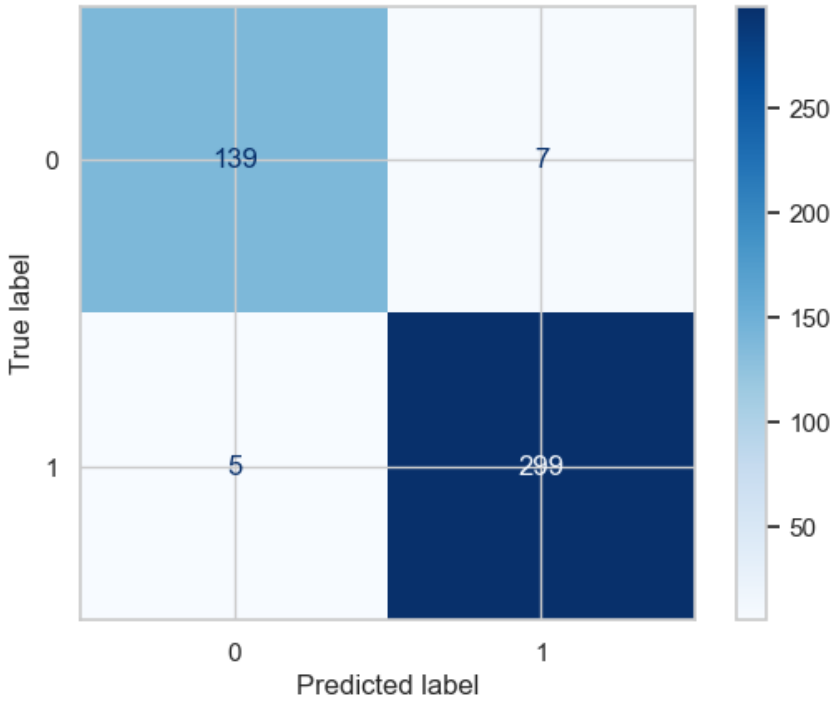

Figura 4.20: Matrice di confusione outliers al 100%

Confronto tra dataset pulito e sporcato

Verranno ora mostrati i box plot con rumore al 100%

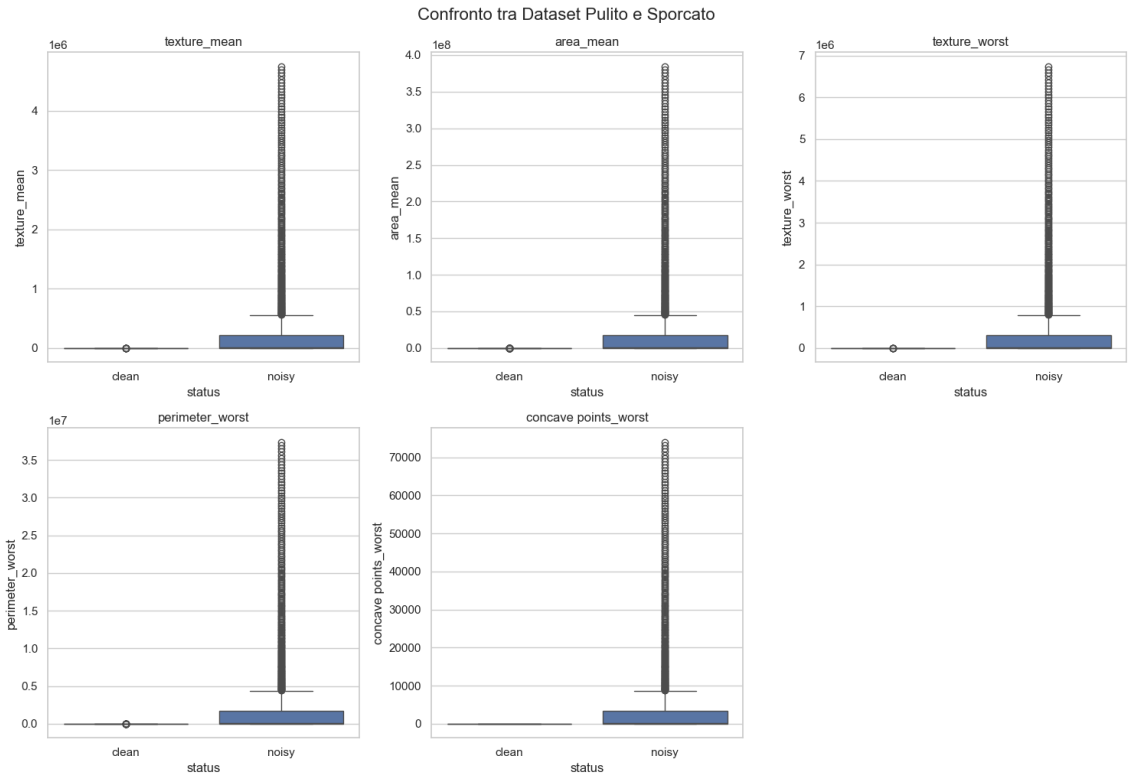

Figura 4.21: Box plot con outliers al 100%

4.2.6 Evoluzione del modello

Andamento dei parametri di qualità

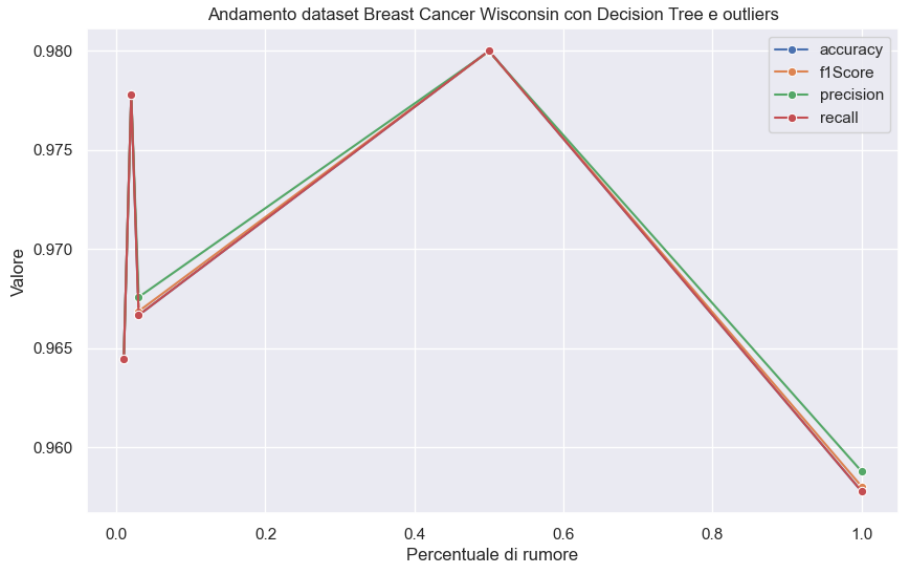

Figura 4.22: Metriche all'aumentare del rumore

4.2.7 Analisi specifica dell'impatto delle feature sul modello

Andiamo adessoa ad analizzare l'impatto delle feature sul modello e nello specifico e determinare quando il modello si accorge del rumore inserito all'interno delle feature.

Situazione di partenza Viene sotto riportato il grafico riguardante la situazione di partenza del modello (con dataset di train), dove nessuna feature è sporcata.

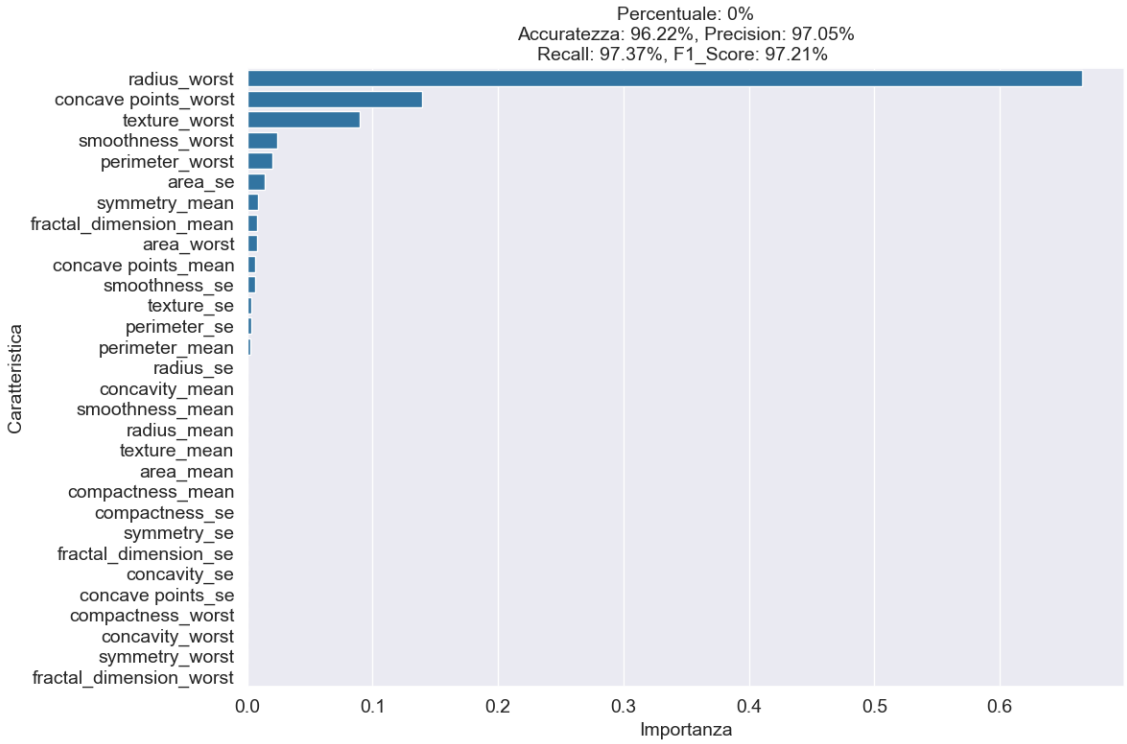

Figura 4.23: Situazione base delle feature

Aggiunta di rumore alla feature "radius_worst"

Andiamo adesso ad aggiungere dei valori di **outliers** alla feature radius_worst, siccome dalla base del nostro test abbiamo visto che questa è la feature più importante.

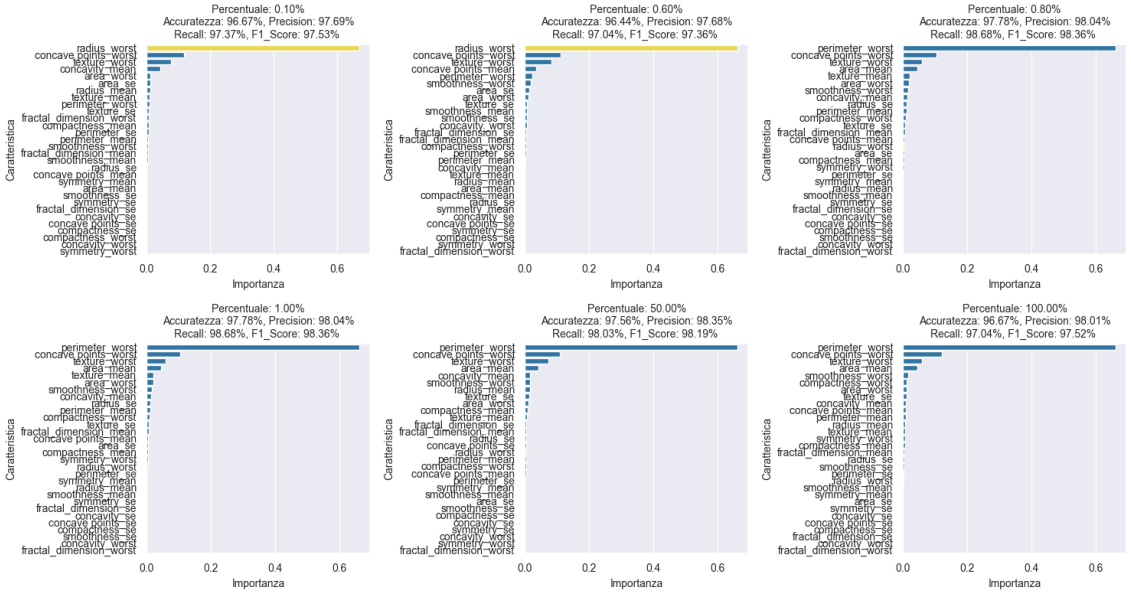

Figura 4.24: Classifica delle feature con aggiunta di outliers a radius_worst

Possiamo notare che rispetto alla base i valori nelle prime percentuali (0.1% e 0.6%) rimangono quasi invariati, già dallo 0.8% però i valori aumentano in quanto il modello non ritiene radius_worst importante come lo era nelle percentuali prima. A 100%, invece, come ci si potrebbe aspettare l'accuratezza e le altre metriche calano un pochino, sicuramente dovuto al fatto che tutti i valori delle feature che siamo andati a considerare sono adesso outliers.

Aggiunta di rumore alla feature "perimeter_worst" e "radius_worst"

Andiamo adesso ad aggiungere rumore anche a perimeter_worst per vedere come si comporta il modello avendo le prime due percentuali che ritiene più importanti sporcate prima con percentuali piccole e poi con grosse percentuali di rumore.

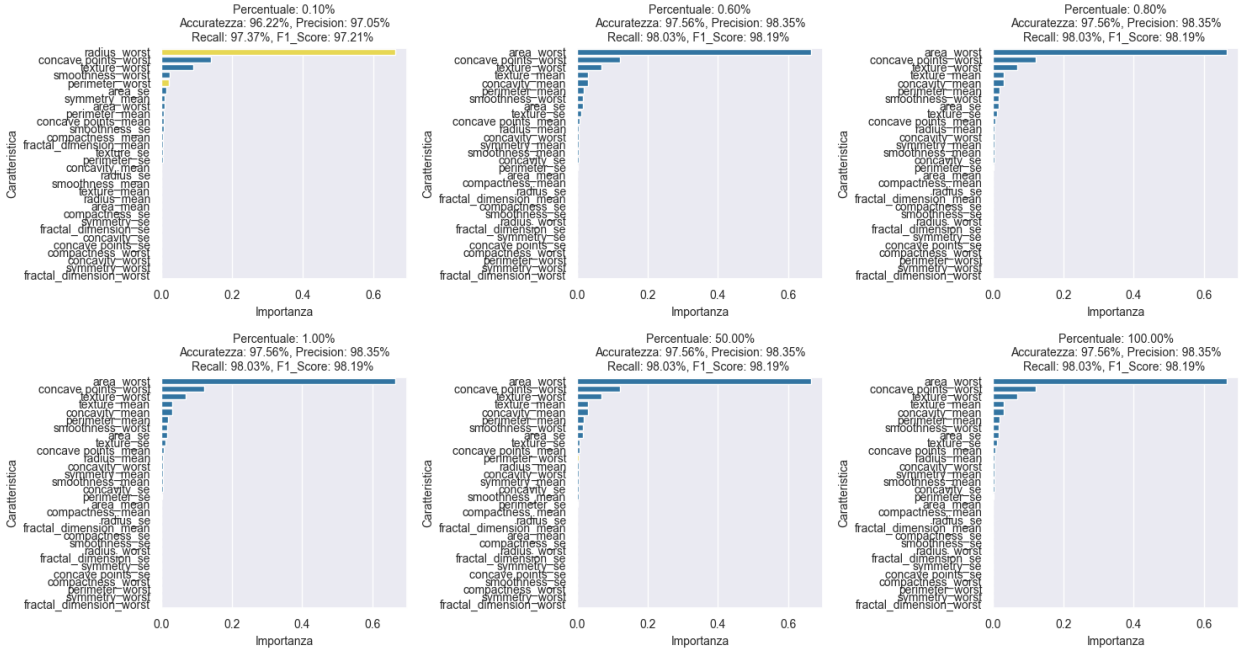

Figura 4.25: Classifica delle feature con aggiunta di rumore a perimeter_worst e radius_worst

Con due feature sporcate il modello non si comporta come ci saremmo aspettati, infatti, subito un calo minimo, praticamente trascurabile all'1% di rumore, ma dopodichè si riprendere a persino al 100% l'accuratezza rimane alta.

Aggiunta di rumore alle prime dieci feature che il modello prende in considerazione

Abbiamo adesso sporcato un totale di dieci feature e siamo arrivati ad una situazione dove nonostante avesse dieci feature sporcate il modello per percentuali piccole continua a comportarsi egregiamente, solo dal 50% in poi il modello subisce un calo nelle prestazioni

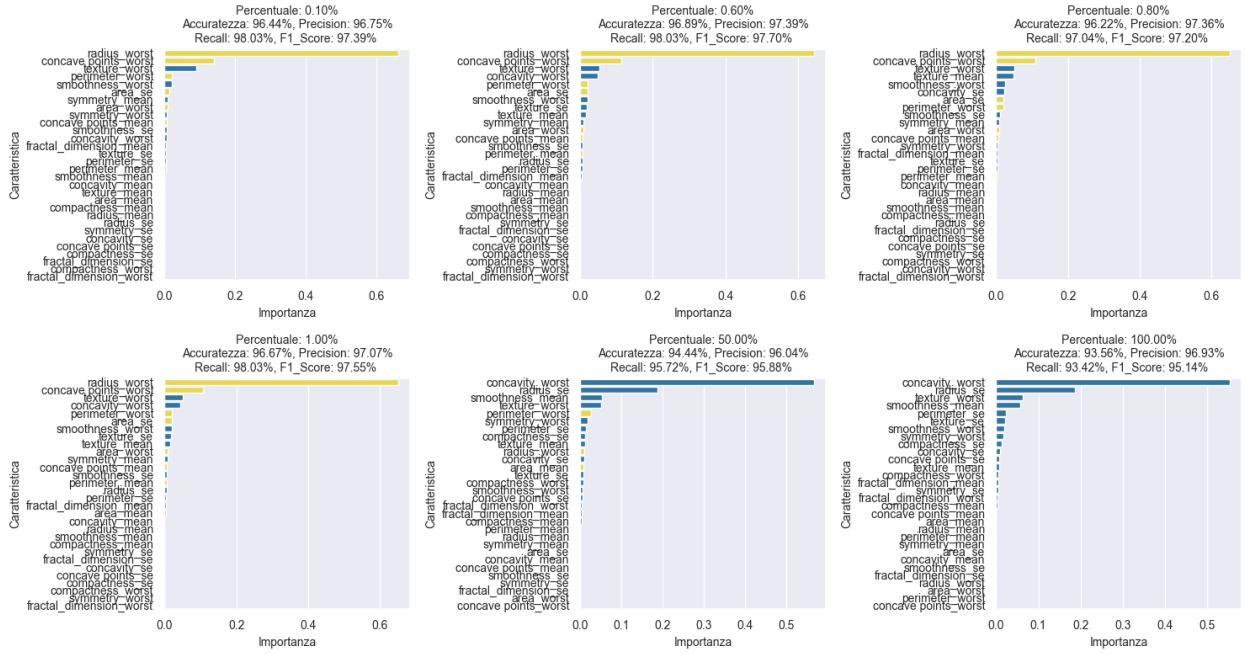

Figura 4.26: Classifica delle feature con aggiunta di rumore alle prime dieci feature che il modello prende in considerazione

(Per una rifermenti di quali feature sta andando a prendere basta vedere le prime 10 feature riportate nella situazione base. [Figura 4.23])

Aggiunta di rumore a tutte le feature del dataset

In quest'ultima sezione del rumore outliers andiamo adesso, come esperimento, ad aggiungere del rumore progressivo a **tutte** le feature del dataset per vedere come si comporta il modello.

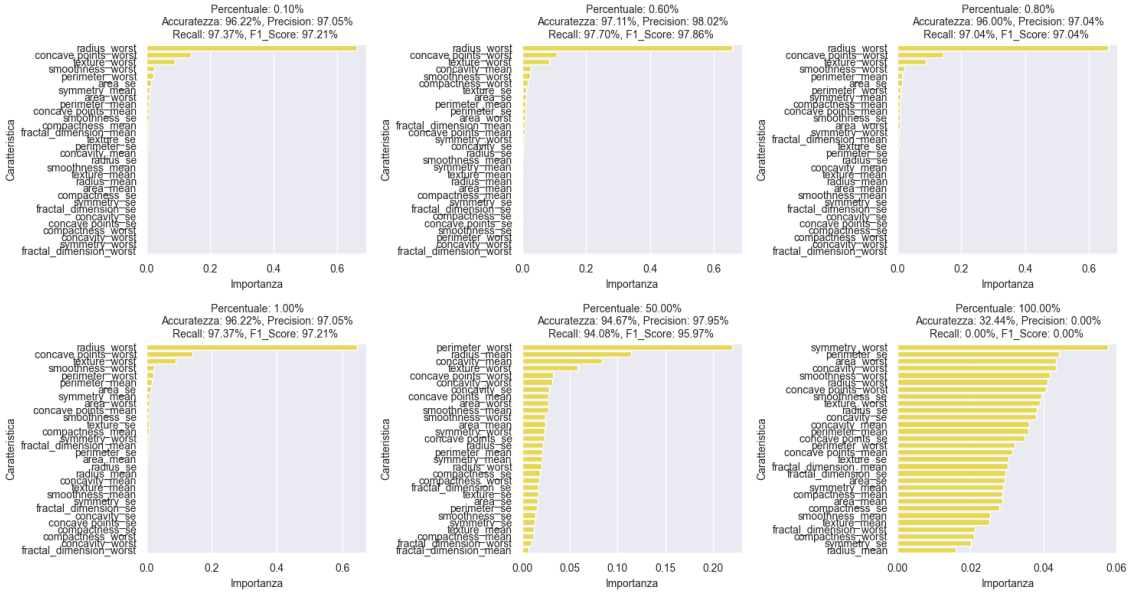

Figura 4.27: Cambiamento con rumore su tutte le feature

Questa situazione ci risulta più interessante rispetto alle altre in quanto possiamo andare a vedere come il modello gestisce le feature all'aumentare del rumore nell'intero dataset. Il modello per percentuali basse possiamo notare che riesce comunque ad avere un accuratezza straordinaria considerando che sappiamo con certezza che ci sono feature sporcate con outliers, già oltre all'1% però le cose si fanno interessanti, possiamo notare che il modello al 50% inizia già a cambiare la classifica delle feature che ritiene più importanti, cala la precisione, come anche le altre metriche, ma rimangono comunque al di sopra del 90%, che, sapendo che il dataset su cui sta venendo addestrato presenta per il 50% valori di outliers è sbalorditivo. Al 100% però, come ci si potrebbe aspettare il modello non sa più cosa considerare e cosa fare... le metriche di importanza delle feature sono tutte sballate e il modello è completamente incapace di predirre qualsiasi cosa, infatti possiamo notare che tutte le metriche tranne l'accuratezza sono a 0%.
L'accuratezza ovviamente non potrà mai scendere allo zero percento in quanto il modello comunque cerca di predirre qualcosa giocando un gioco di probabilità, dove ha il 50% di chance di sbagliare e il 50% di predirre correttamente, infatti a queste percentuali di rumore non è più una predizione, ma solo un gioco di probabilità e casualità.

4.3 Righe duplicate

Infine, viene analizzato il caso in cui vengono generate nuove righe, che sono dei duplicati di istanze già esistenti nel dataset.

4.3.1 Rumore 1%

Con la aggiunta dell'1% di righe duplicate si ha il seguente risultato in termini di prestazioni:

Parametri di valutazione

- **Accuracy**: 96,22%
- **Precision**: 96,21%
- **Recall**: 96,22%
- **F1_score**: 96,21%
- **AUC**: 95,60%

Matrice di confusione

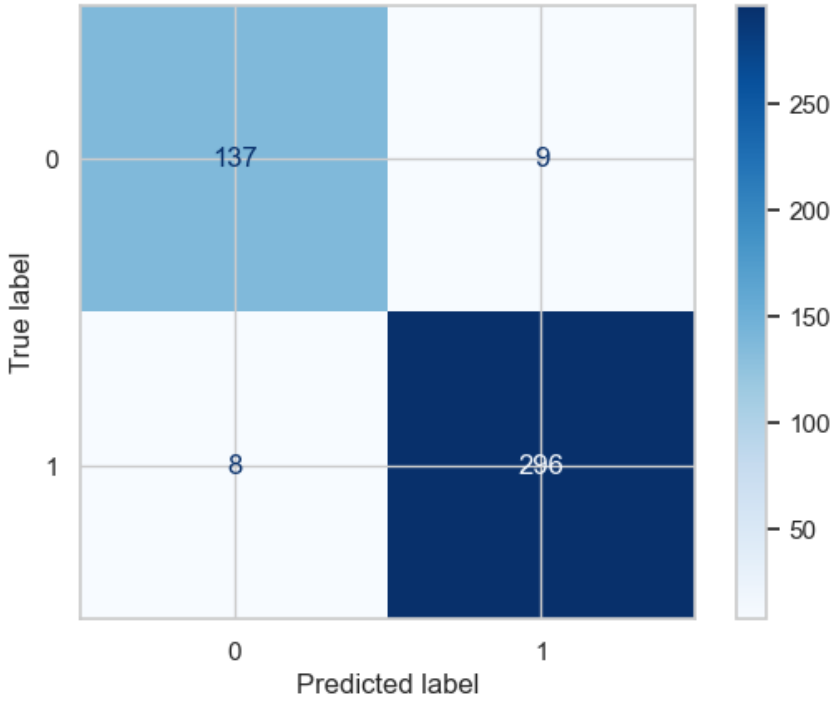

Figura 4.28: Matrice di confusione righe duplicate all' 1%

Da annotare anche in questo caso un leggero calo rispetto alle prestazioni del modello, addestrato su dati non rumorosi.

4.3.2 Rumore 2%

Parametri di valutazione

- **Accuracy**: 96,22%
- **Precision**: 96,21%
- **Recall**: 96,22%
- **F1_score**: 96,21%
- **AUC**: 95,60%

Matrice di confusione

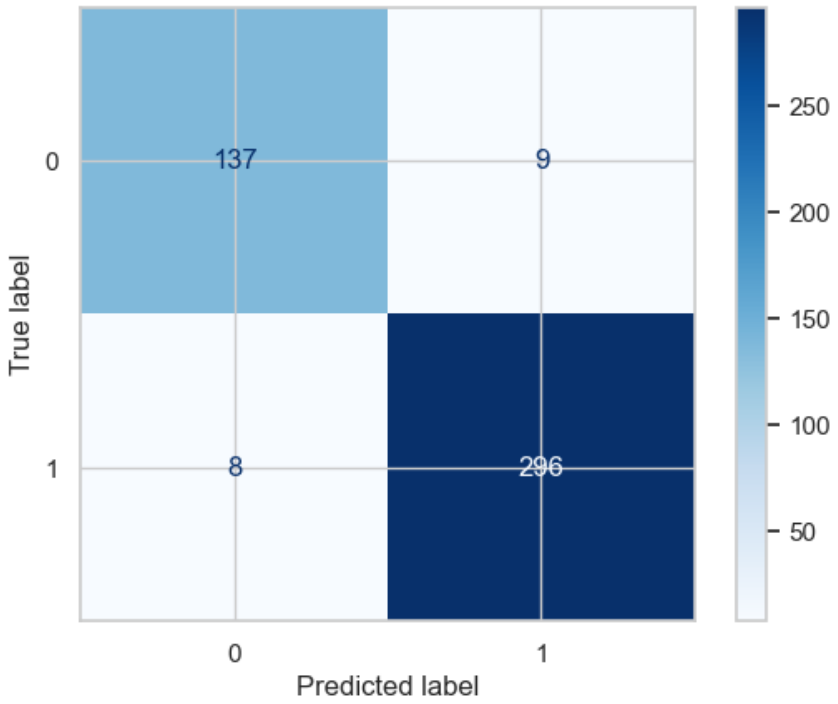

Figura 4.29: Matrice di confusione righe duplicate al 2%

Dalla comparazione tra i parametri relativi al rumore all'1% e al 2% possiamo notare che i valori per la prima volta rimangono esatattamente invariati.

4.3.3 Rumore 3%

Parametri di valutazione

- **Accuracy**: 97,11%
- **Precision**: 97,13%
- **Recall**: 97,11%
- **F1_score**: 97,12%
- **AUC**: 96,97%

Matrice di confusione

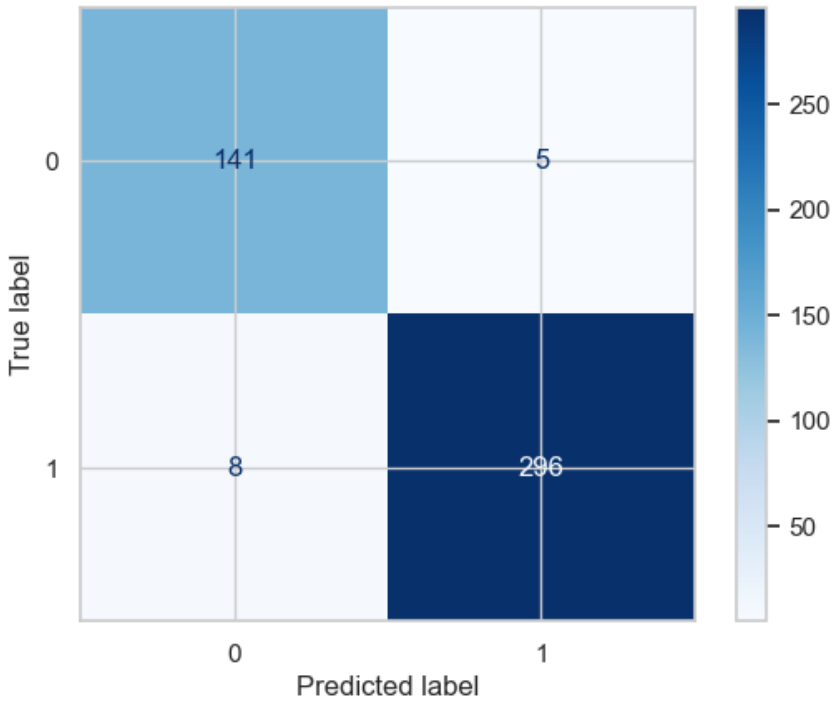

Figura 4.30: Matrice di confusione righe duplicate al 3%

Diversamente da quanto osservato nel passaggio precedente, in questo caso possiamo notare un cambiamento che risulta positivo per quanto riguarda le metriche di prestazione, che vedono i propri valori percentuali aumentare, ma anche diminuire il numero di falsi positivi del modello.

4.3.4 Rumore al 50%

Parametri di valutazione

- **Accuracy**: 97,11%
- **Precision**: 97,13%
- **Recall**: 97,11%
- **F1_score**: 97,12%
- **AUC**: 96,97%

Matrice di confusione

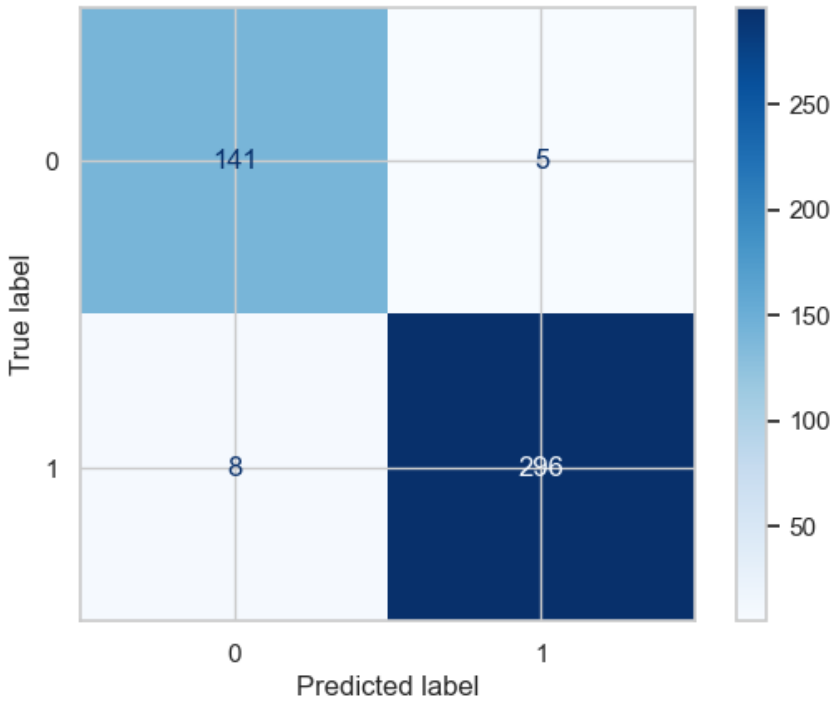

Figura 4.31: Matrice di confusione righe duplicate al 50%

4.3.5 Rumore al 100%

Parametri di valutazione

- **Accuracy**: 96,22%
- **Precision**: 96,21%
- **Recall**: 96,22%
- **F1_score**: 96,22%
- **AUC**: 95,60%

Matrice di confusione

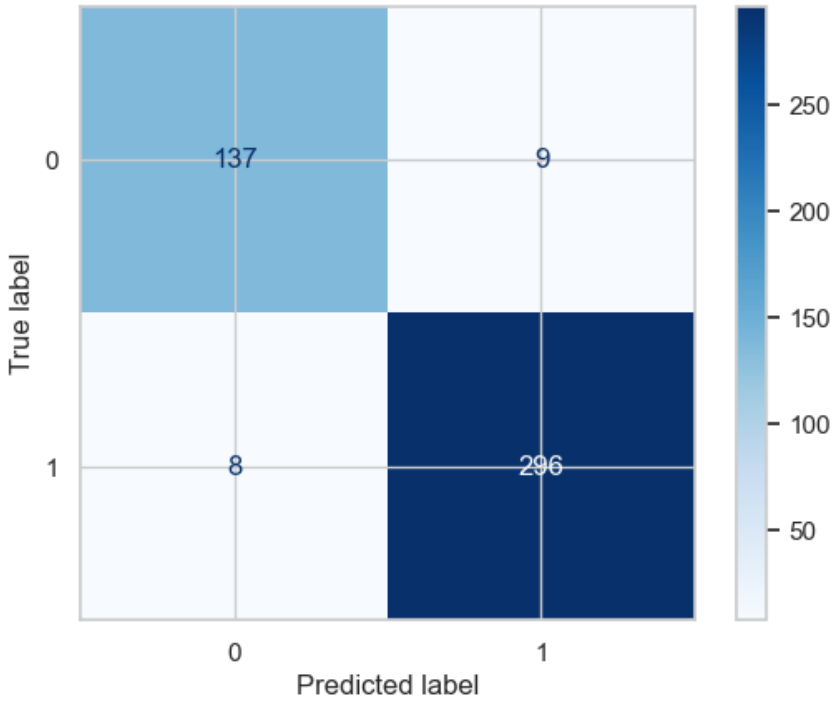

Figura 4.32: Matrice di confusione righe duplicate al 100%

Il risultato mostra un nuovo leggero calo delle prestazioni dimostrato dal calo dei parametri, che sorprendentemente tornano ai valori che si ottengono al 1% e al 2%

Variazioni delle feature più importanti

Per quanto riguarda lo studio sulle singole feature, si è deciso di non riportare i grafici come nei casi degli altri tipi di rumore, a causa della variazione nulla che si ha avuto all'interno del modello per le feature più importanti.

Andamento delle metriche con righe duplicate

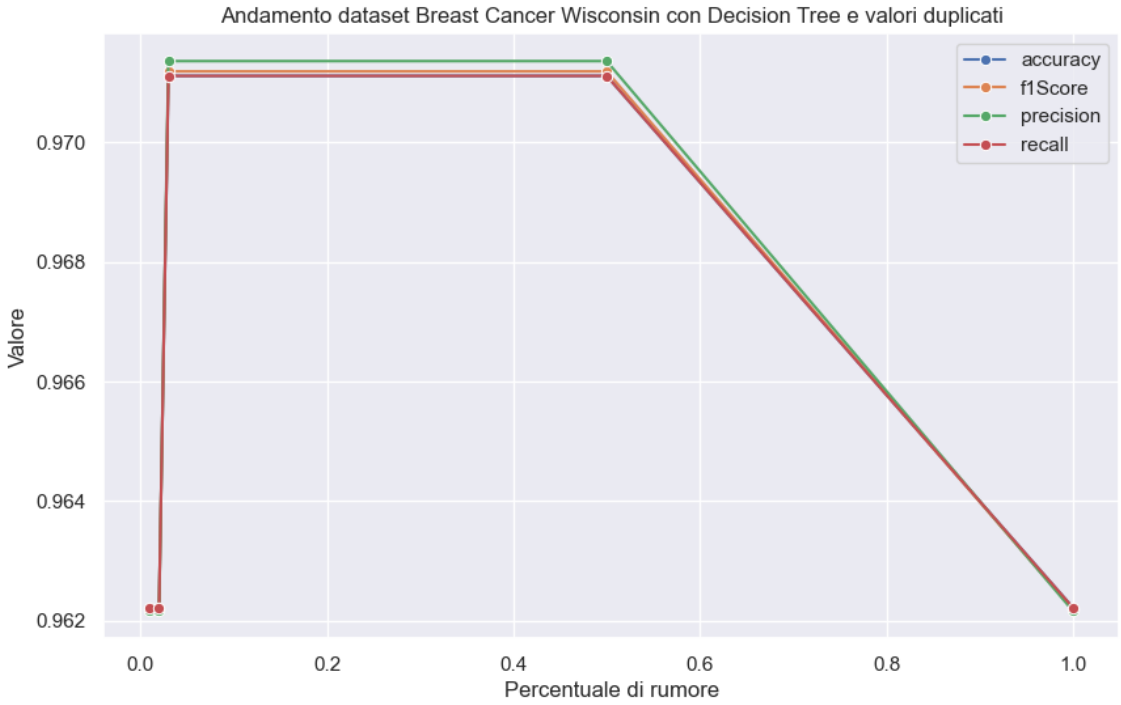

Figura 4.33: Andamento delle metriche con righe duplicate

Capitolo 5

Conclusione

Avendo ora a disposizione i dati degli esperimenti effettuati si può affermare che i valori dei parametri di valutazione del modello sono sempre rimasti molto alti. Analizzando i casi singolarmente, possiamo dire che per quanto riguarda l'introduzione di righe duplicate le prestazioni del modello rimangono stabili, con leggere variazioni in determinati intervalli. Questi risultati indicano che il modello è robusto rispetto alla presenza di righe duplicate, le quali non sembrano influire negativamente in maniera significativa. Anche con un'alta percentuale di dati duplicati, il modello mantiene performance stabili, suggerendo che le righe duplicate non rappresentano un problema per le prestazioni, pur non offrendo un beneficio aggiuntivo oltre una certa soglia.

I casi rimanenti (outliers e righe nulle) seguono, invece, un comportamento differente. Pur mantenendo prestazioni comunque elevate, il modello presenta performance altalenanti in entrambi i casi. L'alterazione dei dati con valori nulli ed outliers sembra quindi influenzare il decision tree, come si può notare anche dalla variazione delle variabili più importanti: quelle con rilevanza maggiore, perdono importanza a favore di altre feature, portando il modello ad eseguire la classificazione in base a queste ultime e non più delle feature "sporcate", che vengono sempre meno considerate con l'aumento della percentuale di presenza di rumore, fino ad essere ignorate.

Nel caso preso in considerazione si può affermare che il mantenimento di prestazioni così elevate è dovuto, oltre che dalla robustezza dell'algoritmo scelto per il modello, anche ad una alta quantità di variabili con forte correlazione con il target: ciò consente al modello di non calare di prestazioni con molta facilità anche a fronte di rumore inserito nelle feature più importanti.

Rimane comunque di vitale importanza la scelta di un dataset che presenti un rumore nullo o contenuto in quanto un rumore troppo elevato potrebbe comportare problemi di accuratezza dei risultati e ad ore di troubleshooting per capire le motivazioni dei risultati ottenuti. In generale siamo contenti dei risultati ottenuti dal modello e non ci aspettavamo una performance così alta. Le percentuali al di sopra del 50% possono essere considerate come uno stress-test del modello, in quanto un modello con così tanto rumore non sarebbe un modello da scegliere o comunque necessiterebbe di tanta pulizia dei dati