Politecnico di Torino

Corso di Laurea in Ingegneria Biomedica
A.a. 2024/2025
Sessione di Laurea luglio 2025

Sviluppo di un biomarcatore tramite intelligenza artificiale, basato su immagini di anatomia patologica digitale, per la predizione della risposta patologica nei pazienti con cancro del retto.

Relatore:
Gabriella Balestra
Correlatore:
Rosati Samanta

Candidato:
Tullio Maria Sborgia

SOMMARIO

L'obiettivo principale di questa tesi è lo sviluppo di un possibile biomarcatore predittivo della risposta patologica alla terapia neoadiuvante nei pazienti affetti da cancro del retto, a partire da immagini di anatomia patologica digitale. Il lavoro è stato condotto su due versioni di un dataset di immagini ottenute da biopsie endoscopiche: un dataset completo, contenente tutte le tiles, e una versione filtrata, da cui sono state escluse le tiles corrispondenti a tessuto di background. In entrambi i casi, le immagini erano già state pre-processate e rappresentate tramite feature quantitative estratte con tecniche di elaborazione delle immagini.

È stata sviluppata una pipeline modulare che include fasi di pulizia, normalizzazione, aggregazione, selezione delle feature e classificazione supervisionata. Le feature sono state aggregate a livello di paziente tramite medie o statistiche descrittive. La selezione delle feature è stata condotta utilizzando metodi statistici e non statistici, tra cui Recursive Feature Elimination (RFE), MRMR (Minimum Redundancy Maximum Relevance), correlazione assoluta con la variabile target, Affinity Propagation e Principal Component Analysis (PCA). Successivamente, sono stati addestrati diversi classificatori, tra cui Support Vector Machine, Random Forest, Gradient Boosting, Naïve Bayes, Stochastic Gradient Descent e K-Nearest Neighbors, valutando l'efficacia di ciascuna combinazione metodo di selezione–classificatore.

Le performance predittive sono state valutate tramite validazione incrociata ripetuta, considerando metodi di bilanciamento delle classi. I risultati hanno mostrato che le migliori performance sono state ottenute sul dataset filtrato (privo di tiles che contengono background), in particolare con la combinazione MRMR–Random Forest, che ha raggiunto valori elevati di AUC (area under the ROC curve) e bilanciamento tra sensibilità e specificità. Anche la combinazione PCA–Naïve Bayes ha mostrato buoni risultati, pur con minore stabilità. L'analisi ha evidenziato che l'esclusione delle tiles di background può migliorare la qualità dell'apprendimento automatico, presumibilmente riducendo il rumore informativo.

Per favorire l'interpretabilità dei modelli, sono stati impiegati gli SHAP values (SHapley Additive exPlanations), che hanno permesso di analizzare l'influenza delle singole feature nel processo decisionale dei classificatori. Tuttavia, le feature identificate come più rilevanti da SHAP non coincidono pienamente con quelle selezionate durante la fase di feature selection, suggerendo una possibile presenza di outlier o ridondanze che alterano la percezione di importanza da parte dei modelli.

Nel complesso, le analisi condotte rappresentano un passo verso lo sviluppo di strumenti automatici e interpretabili basati su dati di anatomia patologica digitale, con potenziali applicazioni cliniche nell'individuazione di biomarcatori predittivi di risposta terapeutica.

Indice

Introduzione

Il tumore del colon retto (CRC) rappresenta ancora oggi una delle neoplasie più importanti al livello mondiale, detenendo il terzo posto (10% rispetto agli altri siti tumorali) in termini di incidenza preceduto dal tumore alla mammella (11.7%) e dal cancro ai polmoni (11.4%). La malattia è maggiormente diffusa in persone fra i 60 e i 75 anni, con poche distinzioni fra uomini e donne, ma dati più recenti hanno registrato un aumento annuo dello 0,4 per cento dei casi di tumore in individui con meno di 50 anni di età e pertanto non coperti dallo screening. Questi ultimi riguardano soggetti molto giovani, fino ai 30 anni, e sono considerati casi "sporadici" attualmente oggetto di studio da parte dei ricercatori. La diagnosi precoce attraverso lo screening è uno dei principali strumenti per migliorare le possibilità di cura. Al di là dello screening, la diagnosi prevede la raccolta di informazioni sulla storia clinica personale e familiare del paziente seguita da una visita medica ma per essere completa è necessario, tuttavia, ricorrere a esami strumentali. L'esame più importante è la colonscopia che permette di prelevare direttamente un pezzo di tessuto (biopsia) e sottoporlo subito ad analisi istologica. Ecografia, tomografia computerizzata (TC) e risonanza magnetica sono utilizzate poi per valutare l'estensione del tumore stesso e la presenza o meno di metastasi a distanza. Il trattamento più comune e più efficace per il tumore del colon-retto, soprattutto se la malattia è negli stadi iniziali è la chirurgia spesso preceduta da chemioterapia e radioterapia, al fine sia di meglio prevenire recidive locali sia di effettuare interventi più conservativi [1].

Studi recenti hanno evidenziato un maggiore beneficio terapeutico si ottiene intervenendo nella fase preoperatoria. Questo ha portato allo sviluppo dell'approccio total neoadjuvant treatment (TNT) che si focalizza nell'intensificare il trattamento chemioterapico o eliminare completamente la fase chemioterapica. Uno degli obiettivi principali della terapia preoperatoria nel trattamento del cancro del retto è la riduzione dello stadio e delle dimensioni del tumore, al fine di migliorare il controllo locale della malattia. Tuttavia, il rischio di sviluppare metastasi a distanza nel tempo rappresenta ancora una sfida clinica significativa. Per contrastare tale rischio di recidiva, recenti ricerche hanno esplorato l'efficacia dell'introduzione precoce della chemioterapia sistemica nelle fasi iniziali del percorso terapeutico. In questo contesto, studi clinici randomizzati e controllati (RCT) hanno evidenziato che l'approccio del trattamento totale neoadiuvante (Total Neoadjuvant Therapy, TNT) è in grado di apportare benefici clinicamente rilevanti, tra cui un incremento della sopravvivenza libera da malattia (disease-free survival, DFS), un aumento dei tassi di risposta patologica completa (pathological complete response, pCR) e una maggiore aderenza al completamento del ciclo chemioterapico. Questi risultati suggeriscono un impatto positivo del TNT su diversi esiti oncologici rilevanti.

In questo scenario uno dei principali studi è il NO-CUT Trial che si concentra sul trattamento del carcinoma del retto medio-basso localmente avanzato, proponendo un approccio innovativo rispetto al trattamento standard. In questo studio, i pazienti ricevono una terapia totale neoadiuvante (TNT) e, in caso di remissione clinica completa confermata da esami strumentali e biopsia, non vengono operati ma seguiti con monitoraggi ravvicinati, secondo il modello del Non-Operative Management (NOM). Su 180 pazienti trattati, circa il 25% ha evitato la chirurgia mantenendo la remissione nel tempo, con una sopravvivenza a 30 mesi del 97%, libera da metastasi.
Parallelamente, lo studio integra analisi multiomiche (radiopatomica, genomica e trascrittomica), esaminando sia il DNA tumorale che quello circolante (tramite biopsia liquida) e l'RNA tumorale, con l'obiettivo di identificare biomarcatori predittivi della risposta alla terapia. Questi dati permettono di prevedere con maggiore precisione i pazienti candidabili alla strategia NOM o che potrebbero trarre beneficio da trattamenti alternativi. Il progetto ha coinvolto quattro centri oncologici italiani ed è

sostenuto da importanti istituzioni di ricerca, dimostrando l'efficacia di un approccio integrato tra clinica e ricerca traslazionale [2].

I biomarcatori, molecolari, cellulari o radiografici, svolgono funzioni fondamentali nella pratica oncologica: diagnostica, prognosi, predizione della risposta ai trattamenti e screening [3]. Nel CRC, esempi consolidati includono mutazioni di KRAS/BRAF, marcatori di instabilità microsatellitare (MSI), ctDNA, e cellule tumorali circolanti (CTC) [4]. Tuttavia, la sensibilità e specificità di molti biomarcatori attuali sono ancora limitate, rendendo necessaria l'integrazione di tecniche avanzate per migliorare il loro valore diagnostico e prognostico.

In merito a ciò, ci sono diversi studi che dimostrano l'efficacia di queste features come biomarcatori. Tra le ricerche più promettenti nell'ambito della radiomica applicata al carcinoma del retto, va segnalato il lavoro di Rosati, Giannini et al., che ha esplorato l'impiego di tecniche di intelligenza artificiale per estrarre e analizzare caratteristiche radiomiche da immagini di risonanza magnetica pretrattamento. L'obiettivo dello studio è la definizione di biomarcatori in grado di predire la risposta completa alla terapia neoadiuvante, contribuendo così alla personalizzazione del percorso terapeutico. In parallelo, Rosati, Giannini et al. hanno dimostrato che l'ottimizzazione congiunta di un sottoinsieme di feature e dei parametri dei classificatori, applicata su diversi set di imaging (T2w, DWI, PET), può migliorare significativamente le performance predittive pretrattamento [5].
La presente tesi si basa sul lavoro di Ilaria Oliva, che ha sviluppato una pipeline automatizzata per l'identificazione di potenziali biomarcatori radiomici predittivi della risposta al trattamento neoadiuvante nel carcinoma del retto. Il suo approccio si è focalizzato sull'analisi di un dataset radiomico pretrattamento costituito esclusivamente da tiles tumorali, con le regioni di background già escluse, e ha previsto una serie di passaggi standardizzati che includono la normalizzazione, la selezione delle feature e l'addestramento di diversi classificatori, con particolare attenzione alla confrontabilità dei risultati tra pazienti [6].

In continuità con questo lavoro, ma con l'obiettivo di estendere le possibilità esplorative offerte dalla radiomica, la seguente tesi si propone di ricostruire e ampliare la pipeline proposta, introducendo l'utilizzo comparativo di due dataset distinti: uno completo e uno depurato dalle regioni di background. Questa scelta mira ad approfondire il ruolo delle informazioni spaziali e a valutare se l'inclusione o l'esclusione del background possa influenzare l'individuazione di biomarcatori significativi, contribuendo così a migliorare la precisione predittiva dei modelli e l'interpretabilità clinica dei risultati.

Dataset e metodi

Dataset e feature

Il dataset utilizzato in questo lavoro deriva da uno studio precedente in cui, a partire da immagini istologiche digitali di pazienti (298) affetti da cancro del retto, è stata effettuata una segmentazione dei tessuti tumorali tramite reti neurali convoluzionali. Le immagini, suddivise in tiles ad alta risoluzione, sono state normalizzate e processate da una rete VGG19 opportunamente addestrata per identificare le regioni di interesse. Su tali regioni, è stata successivamente effettuata l'estrazione di feature radiomiche mediante la libreria PyRadiomics [7], con un'ulteriore fase di aggregazione e selezione delle feature per paziente. Questo processo ha portato alla generazione di file contenenti, per ciascun paziente, le feature associate alle rispettive tiles.
I pazienti si dividono in due categorie:

- Pazienti I/F: molti pazienti presentano due file diversi: I e F. Questo perché per lo stesso paziente sono state effettuate due biopsie una all'inizio e una alla fine del colon retto. Di conseguenza, essendo state analizzate entrambe le biopsie, sono stati generati due file diversi;
- Pazienti restanti: presentano una sola biopsia, di conseguenza un solo file relativo.

Ogni file .csv rappresenta la singola biopsia di un paziente e al suo interno sono presenti tutte le tiles relative alla biopsia del paziente (righe) e tutte le feature estratte dalle tiles di quel paziente (colonne). Le feature estrette sono 93 sono così suddivise:

- 18 features **First order:** 10th percentile, 90th percentile, energy, entropy, interquartile range, kurtosis, minimum, maximum, mean, median, range, mean absolute deviation, robust mean absolute deviation, root mean squared, skewness, total energy, variance e uniformity
- 24 features **Gray Level Cooccurence Matrix (GLCM):** autocorrelation, cluster prominence, cluster shade, cluster tendency, contrast, correlation, difference average, difference entropy, difference variance, inverse difference (Id), inverse difference normalized (Idn), informational measure of correlation (Imc1), informational measure of correlation (Imc2), inverse variance, joint average, joint energy, joint entropy, maximal correlation coefficient (MCC), maximum probability, sum average, sum entropy e sum squares
- 16 features **Gray Level Run Length Matrix (GLRLM):** gray level non uniformity, gray level non uniformity normalized, gray level variance, high gray level run emphasis, long run emphasis, long run high gray level emphasis, long run low gray level emphasis, low gray level run emphasis, run entropy, run length non uniformity, run length non uniformity normalized, run percentage, run variance, short run emphasis, short run high gray level emphasis e short run low gray level emphasis
- 16 features **Gray Level Size Zone Matrix (GLSZM)**: gray level non uniformity, gray level non uniformity normalized, gray level variance, high gray level zone emphasis, large area emphasis, large area high gray level emphasis, large area low gray level emphasis, low gray level zone emphasis, size zone non uniformity, size zone non uniformity normalized, small area emphasis, small area high gray level emphasis, small area low gray level emphasis, zone entropy, zone percentage e zone variance;
- 5 features **Neighbouring Gray Tone Difference Matrix** (**NGTDM**): Busyness, Coarseness, Complexity, Contrast e Strength

- 14 features **Gray Level Dependence Matrix (GLDM):** dependence entropy, dependence non uniformity, dependence non uniformity normalized, dependence variance, gray level non uniformity, gray level variance, high gray level emphasis, large dependence emphasis, large dependence high gray level emphasis, large dependence low gray level emphasis, low gray level emphasis, small dependence emphasis, small dependence high gray level emphasis e small dependence low gray level emphasis.

Oltre alle feature principali estratte, ogni tiles di ogni paziente presenta anche dei metadati inerenti alle coordinate delle feature e le informazioni delle versioni dei software usati nel precedente lavoro. Per quantificare la differenza tra i due dataset, completo e privo di background, in termini di dimensione, è stata calcolata la media del numero di tiles per paziente. Nel dataset completo, ogni paziente presenta in media circa 892 tiles, mentre nel dataset privato del background la media scende a circa 511, con una riduzione media del 42,7%. Questa differenza è attesa, in quanto la rimozione delle tile appartenenti esclusivamente al background riduce sensibilmente la quantità di informazione disponibile.

Si segnala la presenza di un numero molto limitato di pazienti con meno di 20 tiles (< 2% del totale), mantenuti comunque nel dataset per coerenza con l'approccio già adottato in letteratura e per evitare di ridurre ulteriormente il numero di campioni, già contenuto.

Metodi di feature selection e classificatori

Per rendere i futuri risultati coerenti sono stati utilizzati, dove possibile, gli stessi metodi di feature selection e gli stessi classificatori. Di seguito un rapido elenco e descrizione dei metodi e dei classificatori.

Metodi di feature selection

La feature selection è un passaggio fondamentale nell'ambito dell'apprendimento automatico, in particolare quando si lavora con dataset ad alta dimensionalità, come quelli derivati dall'analisi di immagini biomediche. L'obiettivo principale è identificare un sottoinsieme ottimale di variabili (feature) rilevanti per il compito predittivo, migliorando le prestazioni del modello, riducendo il rischio di overfitting e migliorando l'interpretabilità dei risultati. I metodi di feature selection si suddividono comunemente in tre categorie principali: filter, wrapper e embedded methods. I metodi filter valutano l'importanza delle feature indipendentemente dal modello di classificazione, mentre i metodi wrapper sfruttano le performance del modello per selezionare le feature. Gli embedded methods, infine, eseguono la selezione delle feature durante il processo di addestramento del modello stesso. Una panoramica approfondita delle tecniche di selezione delle variabili, dei loro vantaggi, limiti e implicazioni pratiche è stata fornita da Guyon ed Elisseeff [8], i quali hanno evidenziato come una buona selezione delle feature possa migliorare significativamente l'efficienza computazionale e l'accuratezza predittiva, soprattutto in contesti ad alta dimensionalità come la bioinformatica o la diagnostica medica assistita da computer. Di seguito vengono riportati i vari metodi di feature selection usati in questa tesi:

- **Recursive Feature Elimination (RFE):** è un metodo wrapper di feature selection (F.S.) alla cui base c'è un modello di machine learning per stimare la stabilità di ogni feature (per rimanere coerenti si è mantenuto l'utilizzo di "Random Forest"). L'algoritmo, introdotto da Guyon et al. [9], utilizza un valore di "importanza" per rimuovere in modo ricorsivo le feature

meno rilevanti fino ad ottenere un dataset contenente solo le variabili più significative. Nel caso del "Random Forest" l'importanza delle feature è basata sulla diminuzione dell'impurità, che misura l'omogeneità di un gruppo di elementi rispetto le due classi, ed è calcolata tramite il Gini Index [eq.2]:

$$Gini\ Index\ =\ 1 - \sum_{i=1}^{c} p_i^2$$

Equazione 1 Gini Index

Dove C è il numero delle classi e pi la proporzione delle feature appartenenti alla classe i-esima nel nodo corrente. Più il valore dell'indice è vicino allo zero, più campioni appartengono alla stessa classe.

- **Minimum Redundancy – Maximum Relevance (MRMR):** è un metodo che seleziona le feature che hanno un'alta correlazione con la classe e bassa correlazione tra di loro, eliminando le feature altamente ridondanti. La mutua informazione e la mutua informazione media sono state usate rispettivamente per la misura di rilevanza e di ridondanza. La mutua informazione di due feature è la misura della dipendenza reciproca tra le due feature stesse, ovvero quantità di informazioni ottenute da una variabile osservando l'altra, ed è calcolata con la seguente equazione [eq.2]:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right)$$

Equazione 2 Mutua Informazione

Dove:

- p(x,y) è la probabilità congiunta di X=x e Y=y
- p(x) è la probabilità marginale di X=x, dove la probabilità marginale sta per il valore associato alla probabilità di un singolo evento o di una singola variabile, ignorando ogni altra variabile o condizione nel sistema
- p(y) è la probabilità marginale di Y=y

Per quanto riguarda questo metodo di F.S. nel lavoro precedente era stato implementato un algoritmo di ChiMerge per discretizzare i valori delle feature in quanto la Mutual Information si calcola per valori discreti. In questo studio invece si è optato per l'utilizzo della libreria "Feature Engine" che permette il calcolo dell'MRMR anche con variabili continue. Utilizzando il criterio FCQ (F-statistic Correlation Quotient), il metodo valuta la rilevanza delle feature attraverso la statistica F (che misura la dipendenza tra ciascuna feature e il target) e la ridondanza in base alla correlazione tra le feature candidate e quelle già selezionate. L'algoritmo seleziona iterativamente le feature massimizzando il rapporto tra rilevanza e ridondanza, migliorando così la qualità informativa del set finale [10].

- **Affinity Propagation:** è un metodo di clustering che si occupa di raggruppare dati simili tra loro senza la necessità di dover specificare il numero di cluster sin dall'inizio. Questo algoritmo, introdotto da Frey e Dueck [11], è stato usato appunto per fare clustering delle feature e selezionarle attraverso la riduzione della dimensionalità. Inizialmente ogni punto del dataset è considerato come un possibile centro del cluster, definito esemplare. In seguito, in modo iterativo, ogni punto inizierà a mandare "messaggi" a tutti gli altri punti in modo da identificare quale punto possa rappresentarli meglio ed essere "l'esemplare" adatto come

centro del cluster. I "messaggi" che i punti si inviano si aggiornano iterativamente fino a trovare un equilibrio e si dividono in:

- **Responsabilità:** quanto è adatto un certo punto a fare da centro per un altro.
- **Disponibilità:** quanto è disponibile un punto ad accettare un altro come centro.

Quando il sistema converge vengono scelti degli esemplari come centri dei cluster e tutti i restanti punti vengono assegnati al cluster che li rappresenta di più.
La metrica che si usa per decidere quanto un punto è simile ad un altro è la misura di similarità che, nell'implementazione classica, è calcolata tramite la distanza euclidea negativa [eq.3]:

$$s(i,k) = -\| x_i - x_k \|^2$$

Equazione 3 Matrice di Similarità

Questi valori formano una **matrice di similarità SSS**, dove ogni elemento $s(i,k)$ rappresenta quanto il punto i è simile al punto k.

- **Correlazione assoluta tra feature e target:** questo metodo calcola appunto la correlazione assoluta tra le feature e la variabile target, quindi la classe. Essendo le feature continue è stata usata la correlazione di Pearson [eq.4]:

$$r = \frac{Cov(X,Y)}{(\sigma X \cdot \sigma Y)}$$

Equazione 4 Correlazione Assoluta

Dove:

- $Cov(X,Y)$ calcola la covarianza tra X e Y
- $\sigma X, \sigma Y$ rappresentano la deviazione standard di X e Y

- **Analisi delle Componenti Principali (PCA):** l'analisi delle componenti principali è una tecnica di riduzione della dimensionalità nell'ambito della statistica multivariata, per questo è stata considerata e implementata a parte in questo lavoro. Questa tecnica si basa sul creare delle feature artificiali, definite componenti principali, costituite dalla combinazione lineare delle feature iniziali del dataset. Come prima cosa viene calcolata la matrice di covarianza (o di correlazione) tra le variabili originali, da cui si ottengono gli autovalori, che indicano quanta variabilità viene spiegata da ogni componente, e gli autovettori, che definiscono la direzione delle componenti principali nel nuovo spazio. Gli autovettori forniscono i pesi utilizzati per combinare le variabili originali e ottenere le componenti, le quali sono calcolate in modo che siano ortogonali tra loro e quindi non correlate. La prima componente principale raccoglie la maggiore variabilità possibile dei dati, la seconda raccoglie quella che resta, e così via. In questo studio, sono state mantenute solo le componenti necessarie a spiegare almeno il 90% della variabilità totale del dataset, come proposto nella formulazione classica della PCA [12].

Classificatori

Di seguito un elenco dei classificatori usati in questa tesi (per coerenza con la pipeline precedente sono stati usati gli stessi classificatori del lavoro precedente):

- **Naïve Bayes:** Il classificatore Naïve Bayes è un algoritmo di apprendimento supervisionato basato sul teorema di Bayes, ampiamente descritto da Murphy [13] nel contesto dell'apprendimento probabilistico. Il teorema di Bayes viene utilizzato per determinare la probabilità condizionata di un evento che si è già verificato [eq.5]:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Equazione 5 Probabilità Condizionata

Dove:

- P(A) e P(B) sono le probabilità degli eventi A e B
- P(A|B) è la probabilità dell'evento A quando si verifica l'evento B
- P(B|A) è la probabilità dell'evento B quando si verifica A

Nel contesto della classificazione binaria, l'algoritmo Naïve Bayes stima la probabilità che un determinato campione appartenga a una classe piuttosto che all'altra. È definito "naïve" (cioè, ingenuo) perché parte dall'ipotesi semplificativa che tutte le feature siano indipendenti tra loro. Anche se questa condizione raramente si verifica nei dati reali, l'approccio risulta comunque molto efficace e veloce, soprattutto in presenza di dataset di dimensioni contenute. Esistono diverse varianti di Naïve Bayes, a seconda del tipo di dati utilizzati nel dataset. In questo studio si utilizzano il Gaussian Naïve Bayes in quanto adatto per dati numerici continui.

- **K-Nearest Neighbours (KNN):** il KNN è un algoritmo di apprendimento supervisionato che necessita a priori la scelta del parametro K e si basa sul principio che correla la similitudine dei dati alla loro vicinanza nello spazio delle feature. L'algoritmo, introdotto originariamente da Cover e Hart nel 1967 [14], assume che i punti più vicini a un'osservazione siano quelli che probabilmente condividono la stessa classe o un valore simile. Per determinare la vicinanza ai valori K più vicini possono essere usate diverse equazioni relative alla distanza. Le distanze usate in questo lavoro sono:

  - Distanza Euclidea [eq.6]: misura la distanza "lineare" nello spazio.

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_{i)}^{\,2}}$$

Equazione 6 Distanza Euclidea

  - Distanza Manhattan [eq7]: misura la distanza in termini assoluti.

$$d(x,y) = \sum_{i=1}^{n} |\, x_i - y_i|$$

Equazione 7Distanza Manhattan

Nel modello K-Nearest Neighbors (KNN), la scelta del parametro k è cruciale per ottenere buoni risultati. Un valore di k troppo basso rende il modello sensibile al rumore e può portare a classificazioni instabili. Al contrario, un valore troppo alto tende a diluire le decisioni, introducendo un bias che può compromettere la precisione del modello. La classe da assegnare

a un nuovo campione viene determinata attraverso il cosiddetto Majority Voting, cioè in base alla classe più frequente tra i k vicini più prossimi.

- **Support Vector Machine:** Il SVM è un algoritmo di classificazione supervisionata introdotto da Cortes e Vapnik nel 1995 [15], che cerca di separare i dati trovando una linea (o un iperpiano) che divida al meglio le diverse classi. Questo iperpiano viene scelto in modo da massimizzare la distanza (margine) tra i punti delle due classi, così da rendere la separazione il più chiara possibile. Quando i dati non sono separabili in modo lineare, l'algoritmo utilizza delle funzioni chiamate kernel per trasformare i dati in uno spazio con più dimensioni, spazio in cui diventa possibile trovare una separazione lineare, senza calcolare direttamente le nuove coordinate (kernel trick) [16].
  I kernel utilizzati sono:

  - Lineare: per dati linearmente separabili[eq.8]

$$\boldsymbol{K(x,y) = x \cdot y}$$

Equazione 8 Kernel Lineare

  - Polinomiale: per relazioni più complesse [eq.9]

$$\boldsymbol{K(x,y) = (x \cdot y + c)^d}$$

Equazione 9 Kernel Polinomiale

  - Radial Basis Function (RBF): per separare dati non linearmente separabili [eq.10]:

$$\boldsymbol{K(x,y) = e^{-\gamma||x-y||^2}}$$

Equazione 10 Kernel RBF

  - Sigmoid: usata come funzione di attivazione nelle reti neurali [eq.11]

$$\boldsymbol{K(x,y) = \tanh(x \cdot y + c)}$$

Equazione 11 Kernel Sigmoid

- **Decision tree:** Il Decision Tree (DT) è un algoritmo di apprendimento supervisionato con struttura ad albero. Il nodo radice rappresenta l'intero dataset, il quale viene successivamente suddiviso in sottoinsiemi più piccoli (nodi decisionali) sulla base di una regola di suddivisione. Tale regola seleziona la feature e la soglia ottimali in grado di massimizzare la purezza dei nodi figli. L'obiettivo è infatti ridurre l'impurità, ovvero la probabilità che un'istanza venga classificata erroneamente. Ogni ramo dell'albero rappresenta una decisione, mentre le foglie finali rappresentano le classi previste. La suddivisione si interrompe quando tutti i campioni di un nodo appartengono alla stessa classe (nodo "puro") oppure quando si verificano condizioni di arresto predefinite, come la profondità massima dell'albero. L'impurità può essere calcolata con diversi criteri, tra cui l'indice di Gini e l'entropia [17] [eq.12]:

$$Entropia = -\sum_{i=1}^{c} p_i log_2(p_i)$$

Equazione 12 Entropia

metodo è stato introdotto da Quinlan con l'algoritmo ID3 [18], da cui sono derivate versioni migliorate come C4.5 e CART. Studi successivi hanno contribuito a un'analisi più sistematica delle tecniche di classificazione basate su alberi [19].

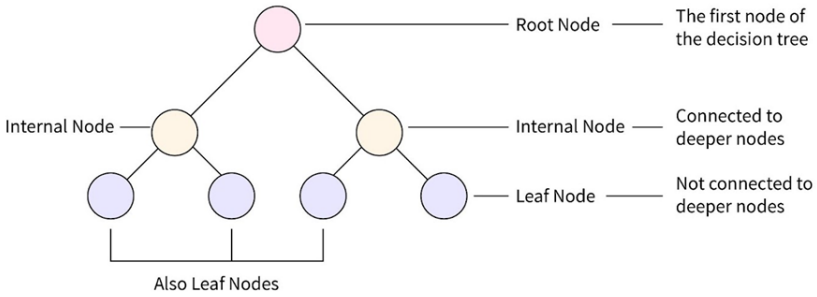


Figura 1 Modello Decision Tree

- **Ensamble Learning Methods**: Gli ensamble learning models, o modelli di apprendimento d'insieme, sono gruppi di modelli, uguali o diversi, che vengono allenati sullo stesso dataset, per poi combinarne le predizioni allo scopo di di ottenere migliori prestazioni rispetto al singolo modello. Gli ensamble learning models utilizzati in questo studio sono:

    - Bagging Decision Tree
    - Boosting Decision Tree

  Il **Bagging Decision Tree**, introdotto da Breiman [20], è un metodo che unisce più alberi decisionali per ottenere un modello più preciso e stabile. Ogni albero viene addestrato su un sottoinsieme casuale del dataset (con campionamento con ripetizione, chiamato bootstrap), e la predizione finale è la media (o la maggioranza) delle predizioni dei singoli alberi. Questo metodo aiuta a ridurre l'overfitting, cioè il rischio che il modello si adatti troppo ai dati di addestramento.
  Il **Boosting Decision Tree**, proposto da Freund e Schapire [21], combina più alberi decisionali semplici e poco profondi (detti "deboli") in sequenza. Ogni nuovo albero viene costruito per correggere gli errori fatti da quelli precedenti, dando più importanza agli esempi difficili. L'obiettivo è migliorare la precisione e ridurre il bias del modello.
  In questo studio sono stati utilizzati due algoritmi di boosting:

    - AdaBoost (Adaptive Boosting): aumenta il peso dei dati che vengono classificati male, così il modello successivo li considera più importanti.
    - Gradient Boosting: non usa i pesi, ma migliora il modello cercando di minimizzare una funzione di errore (loss function) in ogni passaggio.

  La validità empirica di queste tecniche è stata dimostrata da Dietterich [22], che ha condotto un confronto sistematico tra bagging, boosting e altre strategie ensemble. Una panoramica moderna e orientata all'applicazione è fornita da Rahman e Tasnim [23], che illustrano l'efficacia dei modelli ensemble in diversi domini, evidenziandone i vantaggi in termini di generalizzazione.

- **Random forest:** Il Random Forest, sviluppato da Breiman [24], è una variante del metodo di Bagging che utilizza molti alberi decisionali per migliorare l'accuratezza del modello. Ogni albero viene addestrato su un sottoinsieme casuale del dataset, ma con una particolarità in più: a ogni nodo dell'albero, viene selezionato casualmente un sottoinsieme di feature per decidere come suddividerlo. Questo passaggio aumenta la diversità tra gli alberi e aiuta a ridurre l'overfitting, cioè il rischio che il modello si adatti troppo ai dati di partenza. Durante la fase di classificazione, il modello raccoglie le previsioni di tutti gli alberi e sceglie la classe più votata, cioè quella indicata più spesso dagli alberi. Una trattazione teorica approfondita è offerta da Biau e Scornet [25], che esplorano l'interpretabilità, la consistenza statistica e le performance del Random Forest su problemi reali.

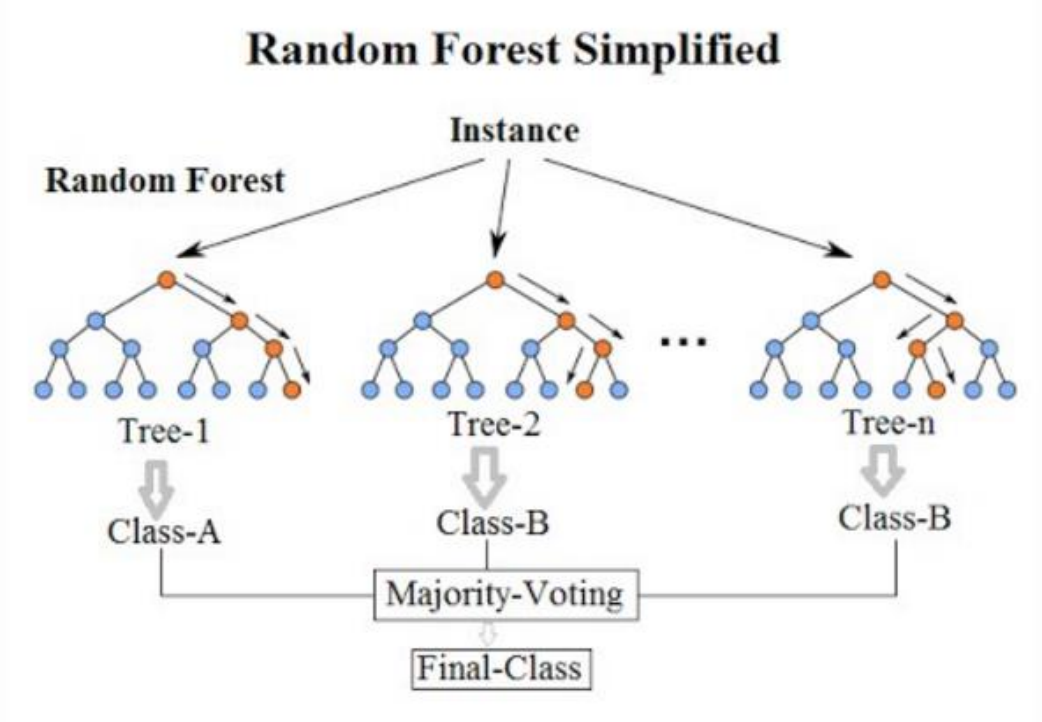


Figura 2 Modello Random Forest

- **Stochastic Gradient Descent (SGD) Classifier:** L'SGD è un classificatore che costruisce un confine lineare per separare le classi, ovvero assume che il legame tra le feature e il risultato possa essere rappresentato da una retta (o un iperpiano nei dati multidimensionali). Viene chiamato "stocastico" perché, durante l'addestramento, non usa tutto il dataset insieme, ma lavora su piccoli gruppi di dati scelti casualmente (mini-batch). Questo approccio accelera il processo di apprendimento e consente di sfuggire a minimi locali [26], ovvero quei punti in cui la funzione di errore sembra minima, ma non lo è davvero su tutta la scala.

Programma e iter

In questa sezione si vuole spiegare nella maniera più semplice, precisa e dettagliata di come sono stati progettati il programma e le sue funzioni per implementare la pipeline di questo lavoro. Il software di questa tesi, programmato in linguaggio Python, è stato progettato a moduli in modo da rendere più semplice e intuitivo l'uso del programma, il debug e la possibilità in futuro di futuri ampliamenti del

codice stesso. Il “programma principale” è la struttura portante da cui poi vengono chiamati gli altri x “sotto programmi” che effettivamente eseguono il codice relativo al modulo di appartenenza. Per fare ciò è stata creata una funzione apposita che, attraverso l’input dell’utente, abilita le varie flag per ogni modulo. La prima richiesta da parte del programma principale consiste nel richiede all’utente se vuole eseguire il programma per intero (all) oppure solo parti di esso (steps), le altre richieste consistono in quale modulo l’utente vuole utilizzare e quindi quale flag voler attivare. In questo modo l’utente ha la scelta di poter lanciare l’intero programma o lanciare il programma passo per passo analizzando i risultati ad ogni step. Inoltre, ogni modulo e ogni sottoprogramma è stato creato, attraverso i comandi “try” - “exept”, in modo tale da fornire un errore con relativo log(resoconto), grazie alla libreria “logging”, tramite il quale è possibile prendere visione in maniera dettagliata di incongruenze nel codice. Per possibili utilizzi futuri di questo programma, e relativi sottoprogrammi, è consigliato l’uso “steps” con l’esecuzione di un modulo per volta, soprattutto se con l’aggiunta di modifiche. Inoltre, il codice è stato progettato affinché i moduli possano operare indipendentemente ma con la presenza dei necessari input, per questo si consiglia di eseguire i moduli nell’ordine in cui sono posti in modo da fornire sempre i giusti input per ogni modulo. Per quanto riguarda la pipeline, e quindi l’iter teorico-pratico, di questo lavoro si è cercato di mantenere il più possibile quello del lavoro precedente, ove possibile, per dare la massima coerenza possibile ai risultati. Infine, è stato sviluppato un sistema di salvataggio di tutti i dati in modo generare risultati il più ordinati possibile.

Preprocessing

Il preprocessing rappresenta una fase fondamentale in quasi tutti gli studi, poiché fornire un dataset pulito, privo di errori, metadati e outlier è essenziale per ottenere risultati attendibili.
Il modulo dedicato al preprocessing è stato implementato secondo la seguente struttura:

Data cleaning

In questa prima parte l’intero dataset è stato ripulito dai metadati che comprendono le coordinate di ogni tiles e le informazioni relative alle versioni dei software usati per l’estrazione delle feature. Sono inoltre state eliminate alcune features risultate instabili (che verranno descritte nel prossimo paragrafo), ed è stata applicata la normalizzazione a tutti i dati di tutte le feature attraverso l’utilizzo del “MinMaxScaler” [27], per rimanere coerenti con il lavoro precedente. Questa trasformazione, che rimappa le feature nell’intervallo [0,1], è stata scelta in virtù della sua capacità di conservare la distribuzione originale dei dati, preservando allo stesso tempo la sensibilità a variazioni relative tra le feature [28], [29]. Come dimostrano Sinsomboonthong (2022), il MinMaxScaler può migliorare significativamente la performance delle reti neurali in presenza di vari range di feature [28]. Normalizzare i dati è necessario in quanto diversi algoritmi, in particolare quelli basati su distanze (es. KNN o SVM), necessitano di dati normalizzati in input. Infine, il dataset è stato diviso in base alla tipologia dei pazienti I ed F e salvati in due cartelle separate, mentre i pazienti restanti sono stati copiati e salvati in entrambi, i nuovi dataset sono: **dataset I** (192 pazienti) e **dataset F** (193 pazienti). In questo studio, si è deciso di proseguire l’analisi esclusivamente sul **dataset F**, tuttavia il programma è stato strutturato in modo da consentire l’analisi su entrambi. Per comodità di consultazione, ogni paziente è stato salvato singolarmente in una cartella specifica. Questa fase è gestita dal sottoprogramma **preprocessing**.

Analisi outliers

Il prossimo passo è quello dell'analisi degli otliers, cioè di possibili dati anomali all'interno del dataset che possono potare a distorsioni della distribuzione dei dati. Per iniziare lo studio degli outliers è stata analizzata la distribuzione per verificare se la distribuzione fosse normale. Per fare ciò è stato scritto un codice apposito che implementa il test di "Shapiro-Wilk" [30] [31] per ogni feature di ogni paziente. Le distribuzioni delle feature mostrano tutte un p-value < 0.05, derivante dal test, da cui la conferma che i dati non presentano una distribuzione normale ed è quindi impossibile usare la metrica Z-score come metodo per individuare gli outliers. Per questo si è passati ad implementare il metodo IQR (Inter Quartile Range). Il metodo IQR si basa sul calcolo dell'intervallo interquartile e sulla definizione di limiti oltre i quali i dati vengono considerati anomali. Le equazioni per il calcolo dell'IQR sono [eq.13]:

$$IQR = Q_3 - Q_1$$

Equazione 13 Inter Quartile Range

Con:

- $Q_1 = 25^{th}$ percentile
- $Q_3 = 75^{th}$ percentile

Tramite questi calcoli è possibile definire:

- $Limite\ inferiore = Q_1 - (k \cdot IQR)$
- $Limite\ superiore = Q_3 + (k \cdot IQR)$

In questo modo tutti i dati compresi al di fuori dei limiti imposti sono considerati outliers. Il k, invece, è una costante che permette di variare l'estensione del range, più è piccolo più si è restrittivi nel considerare dati come appartenenti all'insieme di dati, viceversa si è più permissivi nel considerare più dati come appartenenti all'insieme di dati. Il valore di **k** utilizzato è pari a **1.5**, un valore standard che rappresenta un buon compromesso tra sensibilità e specificità nel rilevamento degli outlier.
In parallelo a questo lavoro è stato condotto uno studio su possibili variabili instabili tra i risultati delle biopsie iniziali e finali dei relativi pazienti. Da questo studio sono state individuate le seguenti feature instabili:

- original_firstorder 90Percentile
- original_firstorder Maximum
- original_firstorder Minimum
- original_firstorder Range
- original_firstorder Uniformity
- original_glcm JointEnergy
- original_glcm MaximumProbability
- original_glszm LargeAreaEmphasis
- original_glszm LargeAreaHighGrayLevelEmphasis
- original_glszm LargeAreaLowGrayLevelEmphasis
- original_glszm ZoneVariance
- original_gldm GrayLevelNonUniformity
- original_glrlm LongRunEmphasis
- original_glrlm LongRunHighGrayLevelEmphasis
- original_glrlm LongRunLowGrayLevelEmphasis
- original_glrlm RunVariance

Le 16 feature sono state eliminate, passando quindi da 93 a **77 features**, da tutti i pazienti anche quelli non facenti parte della categoria I/F per avere un dataset con pazienti che presentassero gli stessi dati. Questo argomento viene riportato in questa sezione in quanto la maggior parte di queste feature sono state individuate dal sottoprogramma "analisi_outliers_IQR" come quelle che presentavano la maggior percentuale di outliers. Così facendo sono state eliminate tutte le features che presentavano una percentuale >10% di outliers, mentre sono state mantenute tutte le restanti che presentano una media del 5%, e solo alcune lo superano. Alla luce di ciò, lo **studio degli outlier si è concluso**, evitando ulteriori trasformazioni o rimozioni.
I risultati di questo studio sono arrivati in concomitanza con l'analisi degli outliers quindi per completezza riportiamo quelle che erano le intenzioni e le azioni che sarebbero state svolte se le feature non fossero state eliminate. Brevemente, l'intenzione era quella di trasformare la distribuzione delle feature che presentavano un andamento molto simile tra di loro in modo da renderele quanto più possibile vicino ad una distribuzione gaussiana. Le feature in questione, che presentavano una percentuale di outliers >10%, hanno valori molto "intensi" in un range "stretto" da somigliare molto da dei "picchi". La trasformazione che era stata presa in considerazione era quella di "Yeo-Jonshon" che permette l'utilizzo anche di valori negativi.

Aggregazione dataset

Un nodo fondamentale di questa pipiline è quello di arrivare ad avere un dataset unico che comprenda tutti i pazienti con i relativi valori per ogni feature. Essendo partiti da un dataset che comprende tutti i singoli pazienti con tutte le informazioni estratte da ogni singola tiles del soggetto, è necessario dunque basarsi su di un "metodo di aggregazione" delle feature in modo da poter unificare tutti i pazienti. Rimanendo fedeli alla precedente tesi i metodi di aggregazione usati sono:

- **Aggregzione per media:** è il metodo più semplice e diretto in cui per ogni singolo paziente viene calcolata la media di tutti i valori delle tiles per ogni feature. In questo modo il dataset risultante consiste in una tabella che presenta i pazienti per righe e il valore medio delle tiles di ogni feature per colonne.

- **Aggregazione per statistiche descrittive dell'istogramma:** questo metodo di aggregazione è più complesso in quanto va a calcolare tutte le statistiche descrittive dell'istogramma per ogni feature del singolo paziente. Le metriche in questione sono:
  Media, Mediana, Skewness, Kurtosis, Percentili, range interquartili (IQR) ed Entropia.

La funzione addetta a questa operazione è "aggregazione".

Aggiunta classi

Come ultima operazione di questo modulo è stata associata la classe **"Label"** ai due dataset aggregati in modo che ogni paziente corrisponda alla classe di appartenenza **"responder" (0)** o **"not responder" (1).** Alcuni pazienti non avendo una classe di appartenenza sono stati eliminati dal dataset rimanendo con un numero di pazienti finale di 137. La funzione che esegue questa operazione è "aggiunta_classi".

Divisione dataset

Il prossimo modulo, gestito dal sottoprogramma “divisione_dataset”, prevede la divisione del dataset secondo la metodologia usata nel precedente progetto. La divisione è stata ottenuta tramite l’utilizzo di “train_test_split” modulo della libreria “sklearn”. La divisione consiste in:

- Construction set: ottenuto prendendo l’80% del dataset e comprende 53 soggetti not responder (classe 0) e 56 soggetti responder (classe 1)
- Test set: ottenuto con il rimanente 20% che comprende 7 soggetti not responder e 21 soggetti responder.

Il Construction-set è stato ulteriormente separato per generare:

- Training-set: costituito dal 75% del Construction set che comprende 39 soggetti not responder e 42 soggetti responder
- Validation-set: costituito dal 25% restante del Construction-set che comprende 17 soggetti not responder e 11 soggetti responder.

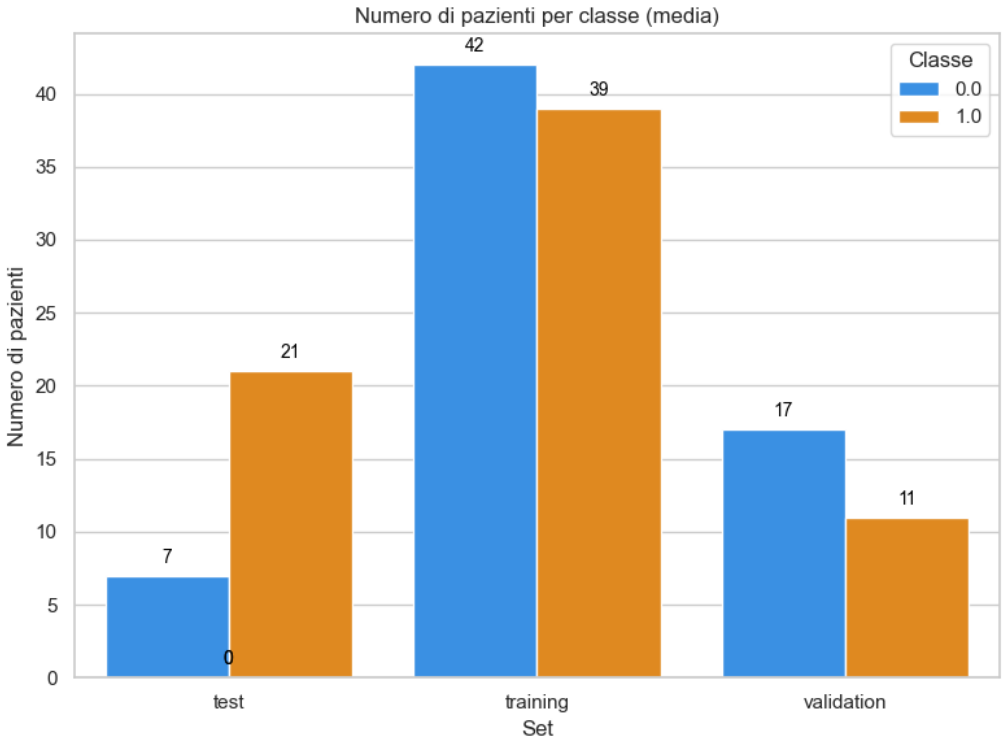


Figura 3 Suddivisione Dataset

La Figura mostra la distribuzione dei pazienti nei tre set (training, validation e test), suddivisi per classe (responder e non responder). Si osserva un leggero sbilanciamento nel test set, generatosi a seguito della suddivisione casuale del dataset aggregato. Poiché il training set risulta bilanciato, l’apprendimento del modello non è stato influenzato da tale squilibrio. Inoltre, per una valutazione affidabile delle prestazioni sono state utilizzate metriche robuste come l'F1-score e l'AUC-ROC. I dataset così ottenuti sono stati salvati in una cartella apposita in modo tale da essere usati come input per i futuri moduli, per prendere visione dei risultati alla fine dell’esecuzione del modulo e per mantenere i dataset aggregati originali, privi di modifiche, pronti per essere usati nel modulo di feature selection.

Feature selection

Il pacchetto dei metodi di F.S. sono stati trattati al livello teorico in precedenza, qui andremo ad approfondirli di più al livello pratico e nelle modifiche che sono state apportate. Questo modulo è stato gestito dal sottoprogramma "feature_selection_originale" all'inizio del quale è stata rimossa la colonna "Label" ed è stata considerata come "indice" la colonna degli "ID paziente", in questo modo entrambe le colonne non sono state considerate come feature.

Affinity propagation e Correlazione Assoluta

L'Affinity propagation è stata implementata usando il modulo relativo "AffinityPropagation" della lbreria "Sklearn.cluster" [32] mentre l'Abs. Corr. è stata implementata principalmente attraverso la funzione "corrwith" della libreria "numpy". Un appunto da fare è sull'Abs. Corr. che prevede come input il numero di feature da selezionare in output. Per ovviare a questo è stato deciso di impostare una soglia fissa decisionale sul numero di feature da considerare pari all' 80% del valore della correlazione calcolata tra feature e target.

Minimum Redundancy – Maximum Relevance (MRMR)

Una prima differenza la possiamo trovare nell'applicazione del metodo di MRMR. In questo caso, come già citato in precedenza, è stata utilizzata la libreria "feature_engine" che ha permesso il calcolo diretto dell'MRMR delle feature che, essendo continue, avrebbero necessitato di essere discretizzate. Il metodo che permette il calcolo su funzioni continue è "FCQ" e i parametri usati sono:

mrmr_selector = MRMR (method="FCQ", scoring="roc_auc", regression=False, random_state=42)

Recursive Feature Elimination (RFE)

Il problema principale nell'applicazione dei metodi di F.S. è stato con l'RFE. Questo perché l'intento è quello di modificare il meno possibile la pipeline di lavoro in modo da rendere i risultati il più coerenti possibile ma l'iter prevedeva l'utilizzo della combinazione di RFE con bootstrap sampling e cross validation. Bootstap sampling è una tecnica basata sul ricampionamento iterativo (numero iterazioni usate nel progetto precedente 100) del dataset in modo da poter variare ad ogni ciclo il subset di pazienti selezionato. La cross validation invece viene utilizzata per andare a valutare le performance del modello alla base dell'RFE (il random forest in questo caso). Questa tecnica prevede la divisione in X fold (in questo caso 5) del dataset in input di cui X-1 sono considerati come training set e l'ultimo è considerato come test set per il classificatore alla base dell'RFE. Così facendo ad ogni iterazione vengono calcolate le performance (accuracy nel nostro caso) dei random forest di ogni fold andando a premiare quello con le metriche più alte.
Il problema di questo insieme di tecniche è il numero di modelli, e quindi il costo computazionale, che il calcolatore deve elaborare per applicare effettivamente il metodo RFE. Sono state tentate diverse opzioni tra cui: l'effettivo utilizzo di RFE+bootstrap+cross validation, il solo utilizzo del modulo RFECV con cross validation integrata e l'utilizzo di RFE standard con la cross validation implementata a parte attraverso "StratifiedKFold" [33].
Il computer usato in questa tesi presenta un processore con frequenza massima di clock di 2.7GHz con 4 core e le tempistiche registrate sono:

- RFECV: per dataset media (77 feature) ci sono voluti circa 2000 s quindi circa 30 min. Se fosse stato calcolato anche per il dataset statistiche (616 feature) il tempo stimato è di 4 ore
- RFE+boostrap+CV: le stime sui tempi si aggirano intorno alle 40 ore per il dataset media e 400 ore per il dataset statistiche
- RFE+CV: circa 4 min per il dataset media e circa 30 min per il dataset statistiche

Oltre alle stime e alle prove è stato tenuto conto anche dei risultati del precedente lavoro in cui, con l'utilizzo di RFE+bootstrap+CV, sono state selezionate 13 feature tra le oltre 700 del dataset statistiche. Questo è dovuto alla severità con cui il metodo RFE, basato sul Random Forest, seleziona le feature. In base a tutte queste considerazioni si è quindi optato per l'utilizzo del RFE con cross validation che garantisce un buon compromesso tra accuratezza e tempi di esecuzione, mantenendo una pipeline ripetibile anche su macchine meno performanti.
Durante la cross-validation è stata registrata, per ciascuna feature, la frequenza con cui veniva selezionata nei diversi fold. Questo ha permesso di costruire un grafico delle frequenze di selezione, utile per valutare la stabilità delle feature selezionate e supportare visivamente la fase decisionale. Per individuare un criterio oggettivo di selezione, è stato utilizzato il modulo "KneeLocator" della libreria kneed [34], che permette di identificare automaticamente il cosiddetto ginocchio (knee point) nella curva delle frequenze ordinate. Questo punto corrisponde a un cambiamento marcato nella pendenza del grafico e rappresenta un dislivello che separa le feature più stabili (frequenze elevate) da quelle meno rilevanti (frequenze più basse). In altre parole, identifica il punto oltre il quale la frequenza di selezione cala in modo significativo. Tuttavia, in presenza di frequenze distribuite in modo relativamente uniforme o con dislivelli poco marcati, il "KneeLocator" può non individuare alcun punto significativo o può restituire una soglia troppo restrittiva, che rischia di escludere anche feature potenzialmente rilevanti. Per ovviare a questa limitazione è stato introdotto un meccanismo di fallback: nel caso in cui la soglia identificata dal "KneeLocator" porti alla selezione di un numero molto ridotto di feature (pari a 0 o al massimo 5), viene applicata una soglia alternativa più permissiva, definita come [eq.14]:

$$Soglia\ forzata = \max(frequenza) - 0.4$$

Equazione 14 Soglia Forzata

Questo approccio consente di garantire una selezione meno severa quando necessario, mantenendo al tempo stesso un criterio replicabile e flessibile.

Classificatori

Nel seguente modulo sono stati allenati, validati e testati tutti i classificatori, trattati teoricamente in precedenza, per ogni metodo di feature selection. L'intera sezione è gestita dalla funzione "training_validation_test". Il primo passo è stato quello di fornire ad ogni classificatore i rispettivi set con le sole feature selezionate dal singolo metodo di feature selection. Fatto questo si passa alla fase di:

- **Training:** la fase di allenamento sul training set avviene contemporaneamente alla fase di tuning dei parametri di ogni classificatore. Più precisamente è stata preparata una "lista" di range di ogni iper-parametro di ogni classificatore. Questa lista è stata poi passata alla funzione **"GridSearchCV"** [35] che sceglie i migliori parametri utilizzando la cross validation (5 fold) il cui parametro decisionale è: l'area sotto la curva ROC (**AUC-ROC**).

- **Validation:** in seguito ogni classificatore allenato e impostato con i migliori parametri viene poi valutato al livello di performance utilizzando il validation set. In questa parte di codice vengono calcolate tutte le metriche possibili per valutare le prime performance dei classificatori e inizialmente come parametro decisionale per la scelta del miglior classificatore è stato usato questo score composito [eq.15]:

$$\text{score} = 0.5*\text{F1\_score\_classe\_1} + 0.3*\text{auc\_roc} + 0.2*\text{balanced\_accuracy}$$

Equazione 15 Score Composito

La scelta di questo score è stata guidata dalla necessità di valorizzare in particolare la **corretta individuazione dei pazienti "responder" (classe 1)**, che rappresentano la classe di maggiore interesse clinico. In molti contesti medici, infatti, è più critico **perdere un responder** (falso negativo) che includere erroneamente un non-responder (falso positivo). Le metriche sono state selezionate per i seguenti motivi:

  - **F1-score della classe 1 (peso 0.5)**: è una metrica focalizzata sulla classe positiva e bilancia precision e recall ed è particolarmente utile quando è importante ridurre falsi negativi [36]. È stata assegnata una **maggior importanza** perché rappresenta direttamente la capacità del modello di identificare correttamente i responder senza eccedere nei falsi positivi;
  - **AUC-ROC (peso 0.3)**: valuta l'abilità discriminativa generale del modello indipendentemente dalla soglia [37]. Il suo utilizzo consente di valutare la **discriminazione generale** del classificatore;
  - **Balanced accuracy (peso 0.2)**: media della sensitivity (recall della classe 1) e della specificity (recall della classe 0). È stata introdotta per tenere conto di eventuali **sbilanciamenti di classe** e per garantire che anche la classe 0 venga considerata in fase di valutazione.

Nonostante la motivazione clinica forte, si è deciso di non utilizzare lo score composito come criterio definitivo per la selezione del miglior classificatore, principalmente per la dipendenza dell'F1-score dalla soglia di classificazione e perché i migliori modelli, secondo la metrica F1-score, presentavano bassi valori di AUC ROC. Questo rende l'F1-score meno stabile e poco adatto a confrontare modelli che restituiscono probabilità predittive (es. modelli probabilistici). Ad esempio, uno dei migliori classificatori individuati tramite lo "score" presentava un valore AUC ROC sul test set di 0.13, il che è contraddittorio. In letteratura si sottolinea infatti che l'AUC-ROC è una metrica più robusta e stabile, soprattutto quando si valutano modelli probabilistici e diversi threshold [38][39]. Per questo alla fine si è optato come parametro decisionale l'AUC ROC in quanto è risultata essere più robusta è stabile. Infine, tutti i risultati, le metriche e i possibili plot di questo modulo sono stati salvati in una cartella apposita per un'ordinata consultazione.

- **Test:** una volta scelti e salvati i migliori classificatori e i loro modelli si è passato a testarli con il test set, il quale è rimasto intoccato fino ad ora per evitare la perdita della sua funzione. Le metriche calcolate per valutare le performance del test set (che corrispondono anche a quelle calcolate per training e validation set) sono:

- AUC-ROC [eq.16]:

$$\boldsymbol{AUC} = \int_0^1 TPR(FPR)dFPR$$

Equazione 16 AUC-ROC

Dove:

- TPR (True Positive Rate) = $\frac{TP}{TP+NF}$
- FPR (False Positive Rate) = $\frac{FP}{FP+TN}$

- Accuracy [eq.17]:

$$\boldsymbol{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Equazione 17 Accuracy

- Balanced Accuracy [eq.18]:

$$\boldsymbol{Balanced\ Accuracy} = \frac{\boldsymbol{Sensitivity + Specificity}}{\mathbf{2}}$$

Equazione 18 Balanced Accuracy

Con:

- $\boldsymbol{Sensitivity} = \frac{TP}{TP+FN}$
- $\boldsymbol{Specificity} = \frac{TN}{TN+FP}$

- Recall per entrambe le classi[eq.19]:

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

Equazione 19 Recall

- Precision per entrambe le classi [eq.20]:

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

Equazione 20 Precision

- F1-score per entrambe le classi [eq.21]:

$$\boldsymbol{F1 - score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Equazione 21 F1 - Score

(TP: True Positives, TN: True Negatives, FP: False Positives, FN: False Negatives)

Sono inoltre state calcolati e plottati: la curva roc, la curva precision-recall e la matrice di confusione.

Shap values

Gli SHAP values (SHapley Additive exPlanations) sono uno strumento di interpretazione dei modelli di machine learning basato sulla teoria dei giochi cooperativi. In particolare, si ispirano al concetto di valore di Shapley, una misura usata in teoria dei giochi per valutare il contributo di ciascun giocatore (in questo caso, ogni feature) al risultato di un gioco (in questo caso, la predizione del modello). In parole semplici, gli SHAP values quantificano quanto ogni singola feature contribuisce ad aumentare o diminuire la probabilità predetta per una determinata osservazione (ad esempio, la probabilità che un paziente sia classificato come responder). Il calcolo degli SHAP values si basa sull'idea di confrontare tutte le possibili combinazioni di feature (sottoinsiemi) e osservare come cambia la predizione del modello quando una feature viene aggiunta al sottoinsieme. Per ogni feature, si considera la media del suo contributo marginale in tutti i possibili scenari. Questo approccio garantisce una ripartizione equa e coerente del risultato finale (la predizione) tra tutte le feature [40].
Poiché il calcolo esatto è computazionalmente costoso, vengono utilizzate delle approssimazioni efficienti (come Kernel SHAP per modelli generici, o Tree SHAP per modelli basati su alberi) che mantengono buone proprietà interpretative e sono scalabili anche su dataset complessi [41].
Gli SHAP values sono uno degli strumenti più affidabili e diffusi per valutare l'interpretabilità locale e globale di un modello predittivo:

- A livello locale: mostrano come le singole feature hanno influenzato la predizione di un singolo paziente, rendendo il modello trasparente caso per caso.
- A livello globale: aggregando gli SHAP values su tutto il dataset, si ottiene un ranking delle feature più importanti in media, fornendo una visione generale del comportamento del modello.

Questo è particolarmente utile in ambito clinico, dove non è sufficiente ottenere un'accurata classificazione, ma è fondamentale capire il motivo per cui un paziente è stato classificato in un certo modo, per poter prendere decisioni cliniche più consapevoli e giustificabili.
In numerosi studi biomedicali, gli SHAP values sono stati utilizzati per analizzare modelli predittivi applicati a dati di immagini, segnali, omiche o cartelle cliniche, proprio per la loro capacità di rendere interpretabili i modelli complessi, come le reti neurali o gli ensemble di alberi [42][43]. In questi contesti, gli SHAP values consentono di identificare pattern fisiopatologici rilevanti e di validare l'affidabilità delle decisioni del modello.
Nel presente lavoro, l'aggiunta dell'analisi tramite SHAP values ha permesso di:

- Confermare le feature più rilevanti secondo il modello, a supporto dei risultati ottenuti nella fase di feature selection.
- Fornire una giustificazione interpretabile delle classificazioni dei pazienti responder e non-responder.
- Rendere il modello più trasparente e potenzialmente più accettabile in ambito clinico, dove la spiegabilità è un requisito fondamentale.

Cross validation esterna

La cross-validazione esterna (in questo contesto) è una tecnica di valutazione in cui i modelli già selezionati e ottimizzati (sia in termini di classificatore che di feature) vengono testati su diverse

suddivisioni del dataset completo per ottenere una stima più robusta e affidabile delle loro prestazioni medie. Non si tratta di una nested cross-validation, bensì di una valutazione a posteriori delle performance dei modelli finali, basata su una suddivisione in k-fold (in questo caso 5) del dataset totale.
Nel nostro caso, una volta selezionato:

- il metodo di feature selection più performante,
- il modello classificatore più adatto

abbiamo unito i dataset di training, validation e test per creare un dataset completo aggregato a livello paziente. Su questo dataset abbiamo poi applicato una Stratified K-Fold Cross-Validation a 5 fold, in cui:

- il dataset viene suddiviso in cinque parti mantenendo la proporzione delle classi (stratificazione),
- a turno, 4 fold vengono usati per addestrare il modello, e 1 fold per testarlo,
- il modello viene ricalibrato da zero ad ogni split, ma senza effettuare nuovamente la feature selection o l'ottimizzazione (queste erano già state fatte a monte).

Le metriche (accuratezza, F1, AUC, ecc.) vengono poi calcolate per ciascun fold e mediate per ottenere una valutazione più stabile. In questo modo abbiamo potuto calcolare anche le deviazioni standard delle prestazioni. Abbiamo scelto di applicare questa forma di cross-validazione "esterna" per i seguenti motivi:

- Valutare la stabilità delle performance del modello selezionato su più suddivisioni del dataset, piuttosto che basarsi su un'unica train-test split.
- Calcolare la deviazione standard delle metriche, utile per confrontare i modelli non solo sulla media delle prestazioni, ma anche sulla loro robustezza.
- Evitare di sovrastimare le performance, dato che i modelli erano stati inizialmente validati su un solo test set. La cross-validazione a 5 fold ci ha fornito una misura più generalizzabile delle loro capacità predittive.

L'approccio adottato si colloca tra una semplice validazione hold-out e una nested cross-validation. Non è completamente indipendente dal processo di selezione (dato che le feature sono state selezionate *prima* di questo ciclo), ma rappresenta comunque un buon compromesso per:

- ottenere una valutazione post-selezione
- senza dover rientrare in una complessa (e computazionalmente costosa) nested CV [44][45]

Nonostante non si tratti di nested CV, studi recenti evidenziano che la cross-validation esterna rappresenti un buon compromesso tra accuratezza delle stime e costi computazionali [46][47].

Risultati

Dataset completo: risultati metodo di aggregazione calcolo media di ogni feature.

Affinity propagation

Le feature selezionate dal metodo sono:

- mean_original_firstorder_TotalEnergy
- mean_original_glcm_MCC
- mean_original_glrlm_LowGrayLevelRunEmphasis
- mean_original_gldm_GrayLevelVariance
- mean_original_gldm_SmallDependenceEmphasis
- mean_original_gldm_SmallDependenceHighGrayLevelEmphasis

Il tuning dei parametri dei classificatori è riportato nella seguente tabella (tab.1):

Classificatori	Parametri Tuning
Adapting Boosting	learning_rate: 0.5, n_estimators: 50
Bagging DT	bootstrap: True, max_samples: 0.5, n_estimators: 10
Decision Tree	criterion: entropy, max_depth: 4, min_samples_split: 10
Gradient Boosting	learning_rate: 0.01, max_depth: 3, n_estimators: 50
KNN	n_neighbors: 7, p: 2, weights: uniform
Naive Bayes	Default
Random Forest	max_depth: 10, max_features: sqrt, min_samples_split: 2, n_estimators: 50
SGD	alpha: 0.001, loss: log_loss, penalty: l1
SVM Linear	C: 0.01, gamma: scale
SVM Polinomial	C: 1, degree: 2, gamma: scale
SVM Kernel RBF	C: 0.1, gamma: scale
SVM Sigmoid	C: 0.1, gamma: scale

Tabella 1Parametri classificatori

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC (fig.4) e accuracy (fig.5).

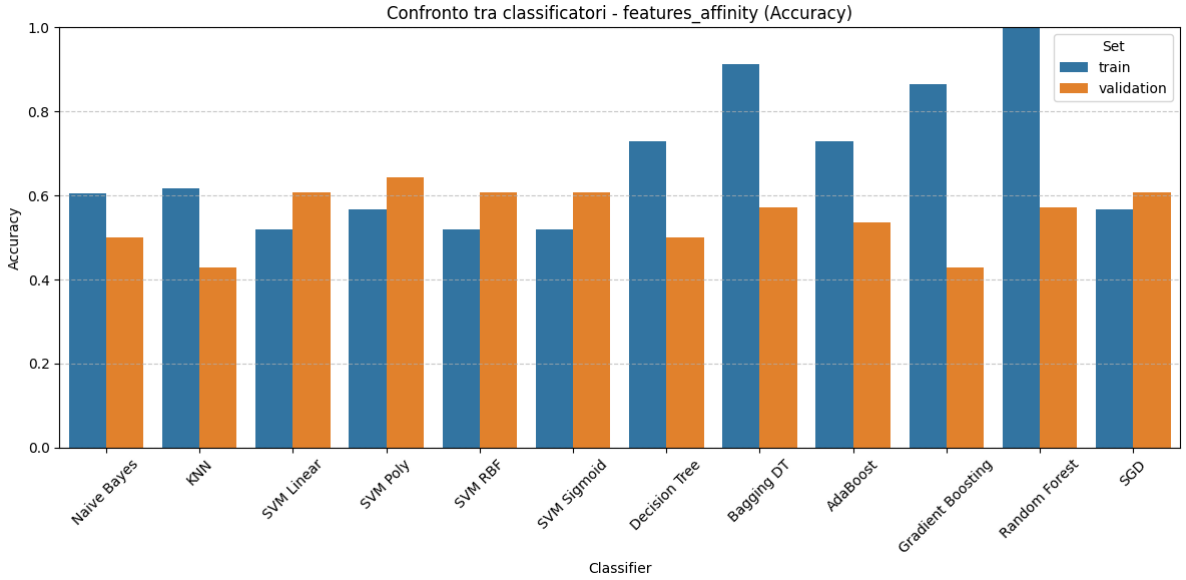


Figura 4 Valori AUC-ROC Validation Affinity

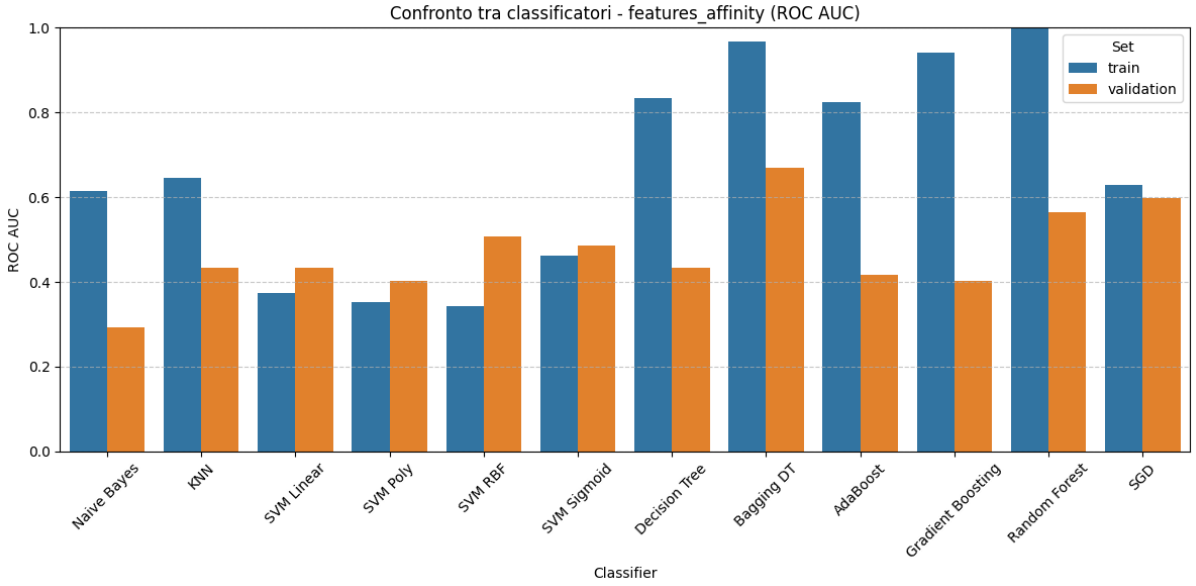


Figura 5 Valori Acccuracy Validation Affinity

Il classificatore, per questo metodo di feature selection, scelto in base allo "score" calcolato sulle performance del validation set è il Bagging DT.

Correlazione assoluta tra feature e traget

Le feature selezionate dal metodo sono:

- mean_original_firstorder_Energy
- mean_original_firstorder_Entropy
- mean_original_firstorder_TotalEnergy
- mean_original_glcm_ClusterProminence

- mean_original_glcm_Correlation
- mean_original_glcm_Imc2
- mean_original_glcm_MCC
- mean_original_glrlm_GrayLevelVariance
- mean_original_glrlm_LowGrayLevelRunEmphasis
- mean_original_glrlm_ShortRunLowGrayLevelEmphasis
- mean_original_glszm_GrayLevelVariance
- mean_original_glszm_LowGrayLevelZoneEmphasis
- mean_original_glszm_SmallAreaLowGrayLevelEmphasis
- mean_original_ngtdm_Busyness
- mean_original_ngtdm_Contrast
- mean_original_gldm_SmallDependenceLowGrayLevelEmphasis

Nella figura (fig.6) sottostante possiamo vedere il grafico dei valori di correlazione delle feature selezionate.

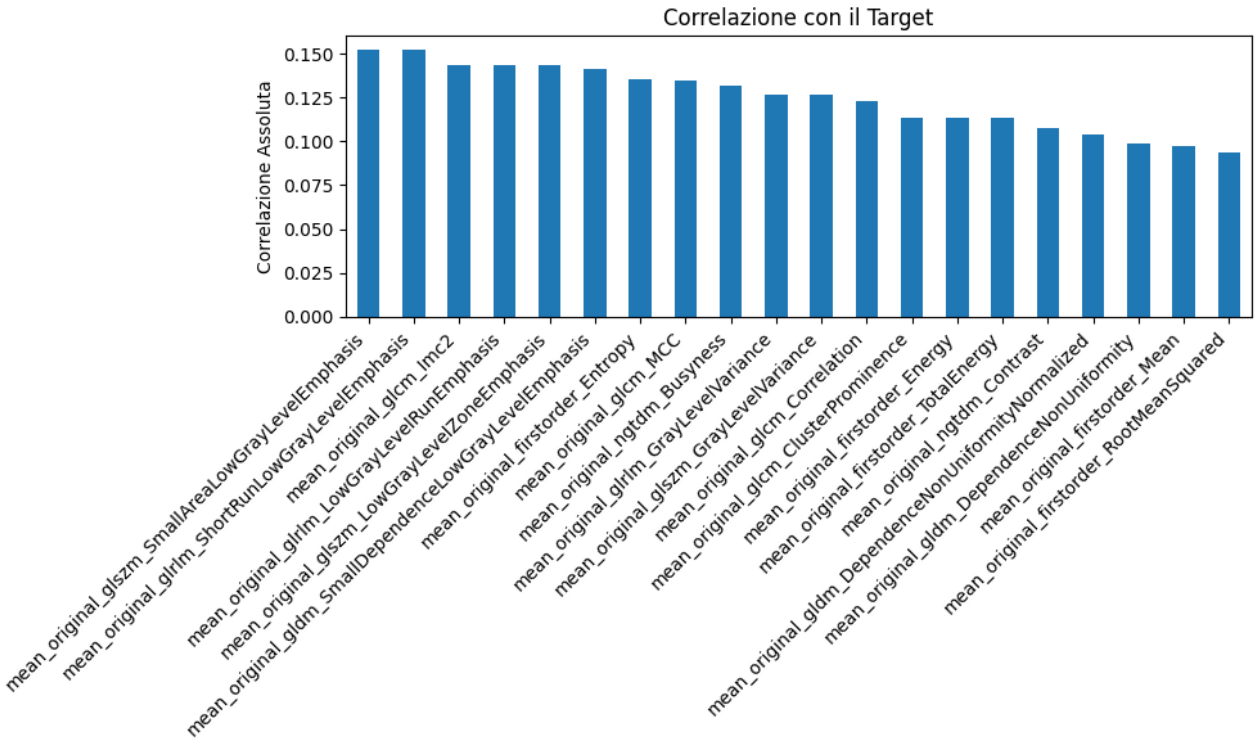


Figura 6 Valori Correlazione Assoluta

Il tuning dei parametri dei classificatori è riportato nella seguente tabella (tab.2):

Classificatori	Parametri Tuning
Adapting Boosting	learning_rate: 1, n_estimators: 50
Bagging DT	bootstrap: True, max_samples: 0.5, n_estimators: 50
Decision Tree	criterion: gini, max_depth: 6, min_samples_split: 10

Gradient Boosting	learning_rate: 0.01, max_depth: 7, n_estimators: 50
KNN	n_neighbors: 9, p: 2, weights: uniform
Naive Bayes	Default
Random Forest	max_depth: None, max_features: sqrt, min_samples_split: 2, n_estimators: 50
SGD	alpha: 1e-05, loss: hinge, penalty: l2
SVM Linear	C: 10, gamma: scale
SVM Polinomial	C: 10, degree: 2, gamma: scale
SVM Kernel RBF	C: 1, gamma: scale
SVM Sigmoid	C: 10, gamma: scale

Tabella 2 Prametri classificatori Corr.Ass.

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC (fig.7) e accuracy (fig.8).

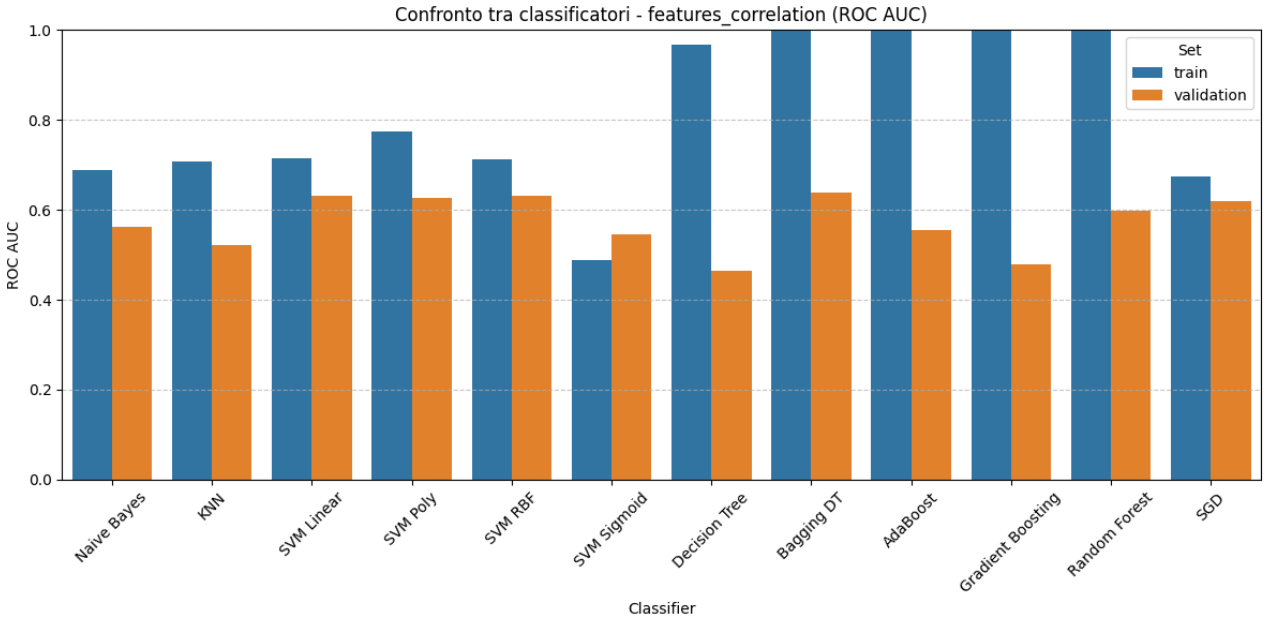


Figura 7 AUC-ROC Validation Corr.Ass.

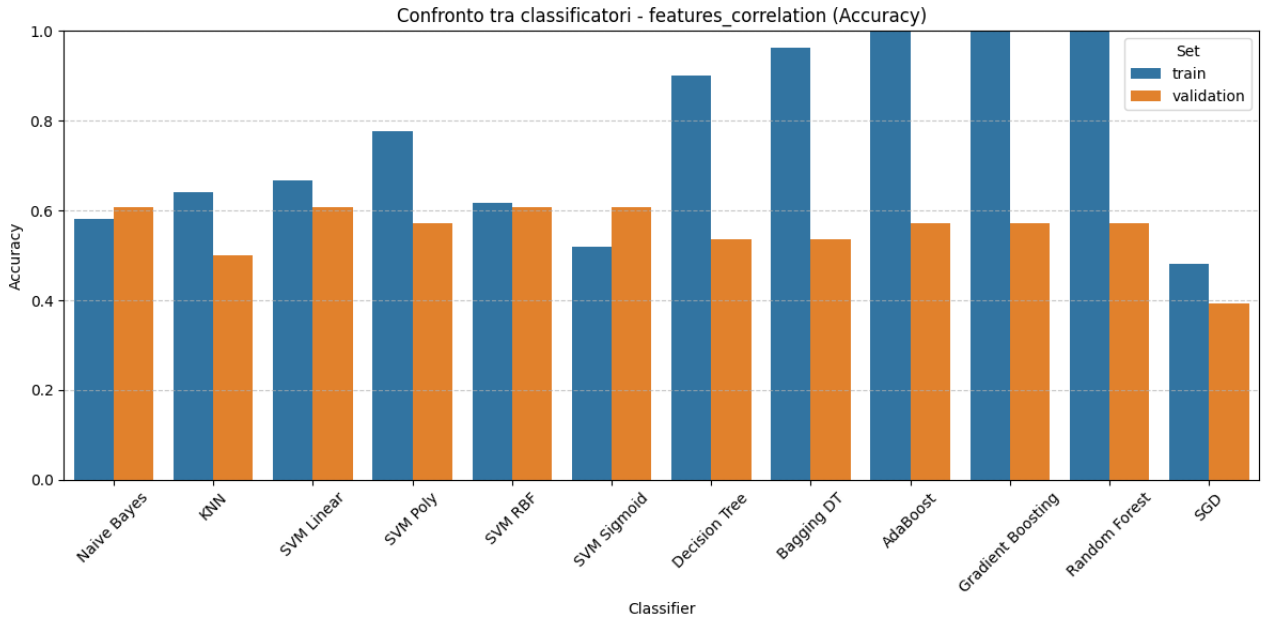


Figura 8 Accuracy Validation Corr.Ass.

Il classificatore, per questo metodo di feature selection, scelto in base allo "score" calcolato sulle performance del validation set è il Bagging DT.

Minimun Redundancy – Maximum Relevance (MRMR)

Le feature selezionate da questo metodo sono:

- mean_original_firstorder_Entropy
- mean_original_firstorder_Skewness
- mean_original_glcm_ClusterProminence
- mean_original_glcm_Correlation
- mean_original_glcm_Imc2
- mean_original_glcm_MCC
- mean_original_glrlm_GrayLevelVariance
- mean_original_glrlm_LowGrayLevelRunEmphasis
- mean_original_glrlm_ShortRunLowGrayLevelEmphasis
- mean_original_glszm_GrayLevelVariance
- mean_original_glszm_LowGrayLevelZoneEmphasis
- mean_original_glszm_SmallAreaLowGrayLevelEmphasis
- mean_original_ngtdm_Busyness
- mean_original_ngtdm_Contrast
- mean_original_gldm_SmallDependenceLowGrayLevelEmphasis

Nella figura (fig.9) sottostante mostriamo la rilevanza delle top 100 feature (in questo caso 77 perché è il massimo numero di feature nel dataset aggregato per media) calcolata con il metodo MRMR.

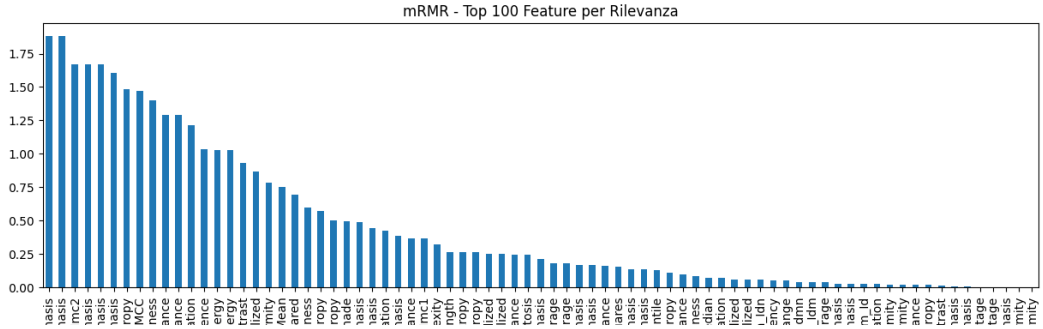


Figura 9 Valori Rilevanza MRMR

Il tuning dei parametri dei classificatori è riportato nella seguente tabella (tab.3):

Classificatori	Parametri Tuning
Adapting Boosting	learning_rate: 1, n_estimators: 50
Bagging DT	bootstrap: False, max_samples: 1, n_estimators: 10
Decision Tree	criterion: gini, max_depth: None, min_samples_split: 10
Gradient Boosting	learning_rate: 0.01, max_depth: 7, n_estimators: 50
KNN	n_neighbors: 9, p: 2, weights: uniform
Naive Bayes	Default
Random Forest	max_depth: None, max_features: sqrt, min_samples_split: 5, n_estimators: 50
SGD	alpha: 1e-05, loss: hinge, penalty: l1
SVM Linear	C: 1, gamma: scale
SVM Polinomial	C: 1, degree: 4, gamma: auto
SVM Kernel RBF	C: 100, gamma: scale
SVM Sigmoid	C: 0.1, gamma: scale

Tabella 3 Parametri Clssificatori MRMR

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC (fig.10) e accuracy (fig.11).

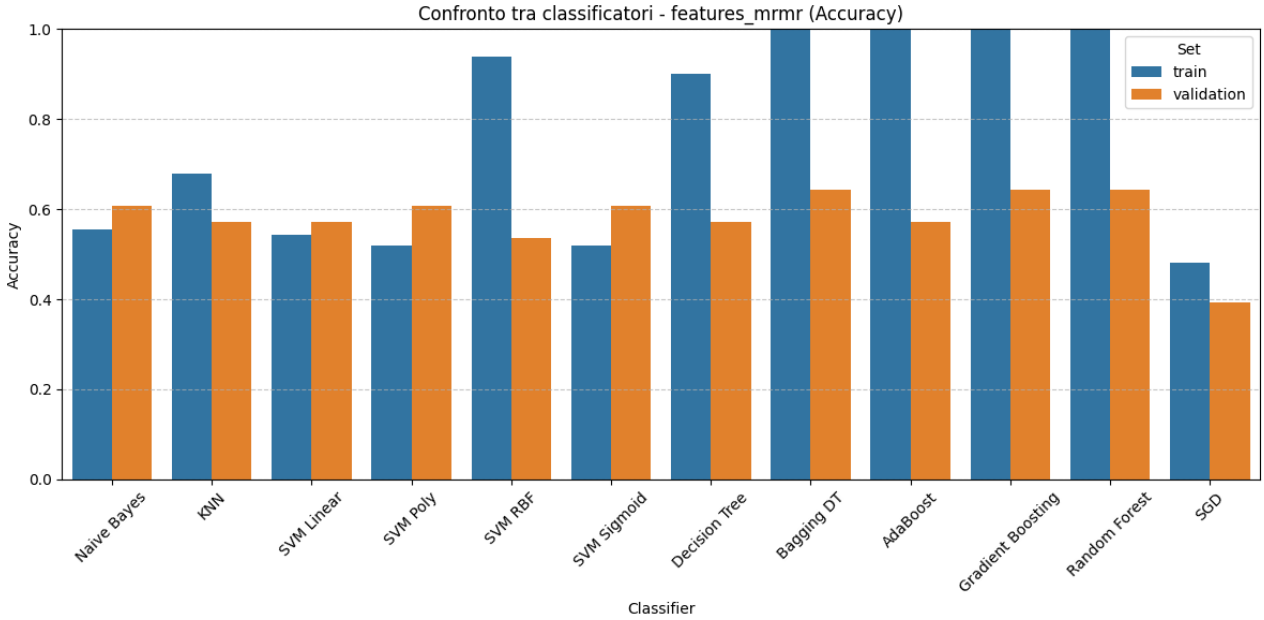
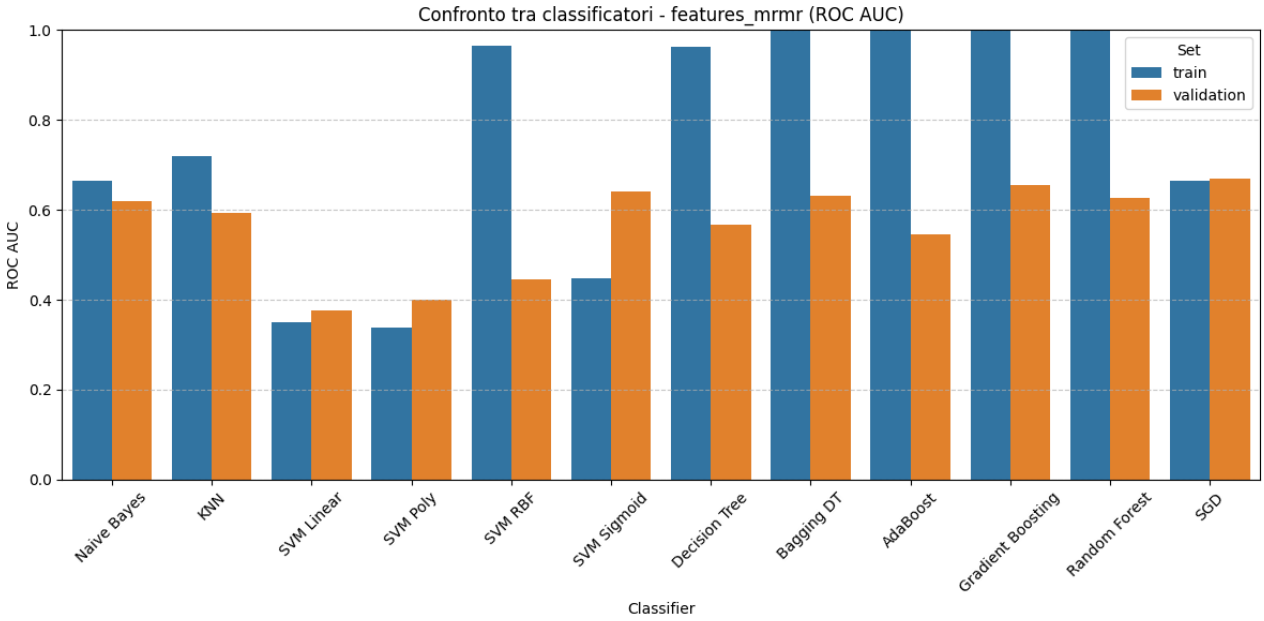


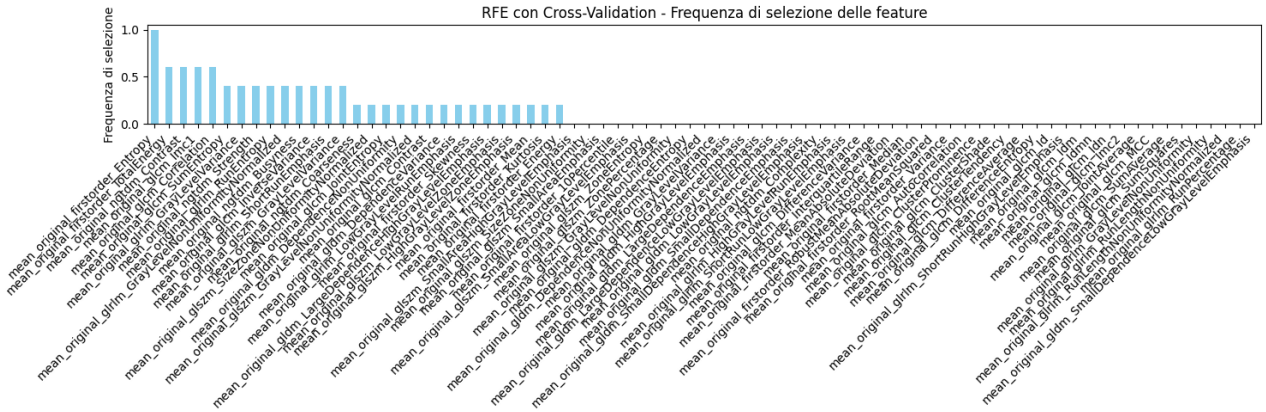
Figura 10 AUC-ROC Validation MRMR



Il classificatore, per questo metodo di feature selection, scelto in base allo "score" calcolato sulle performance del validation set è il SGD.

RFE+CV:

Il grafico ottenuto (fig.11) dalle frequenze di selezione è il seguente:



Le feature selezionate sono:

- mean_original_firstorder_Entropy
- mean_original_firstorder_TotalEnergy
- mean_original_ngtdm_Contrast
- mean_original_glcm_Imc1
- mean_original_glcm_Correlation

Il tuning dei parametri dei classificatori è riportato nella seguente tabella (tab.4):

Figura 11 Frequenze di selezione

Classificatori	Parametri Tuning
Adapting Boosting	learning_rate: 0.5, n_estimators: 200
Bagging DT	bootstrap: False, max_samples: 0.7, n_estimators: 10
Decision Tree	criterion: gini, max_depth: 8, min_samples_split: 2
Gradient Boosting	learning_rate: 0.01, max_depth: 7, n_estimators: 150
KNN	n_neighbors: 9, p: 1, weights: uniform
Naive Bayes	Default
Random Forest	max_depth: 10, max_features: sqrt, min_samples_split: 5, n_estimators: 50
SGD	alpha: 0.001, loss: hinge, penalty: l1
SVM Linear	C: 10, gamma: scale

SVM Polinomial	C: 0.1, degree: 4, gamma: scale
SVM Kernel RBF	C: 1, gamma: scale
SVM Sigmoid	C: 10, gamma: scale

Tabella 4 Parametri classificatori RFE

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC (fig.12) e accuracy (fig.13).

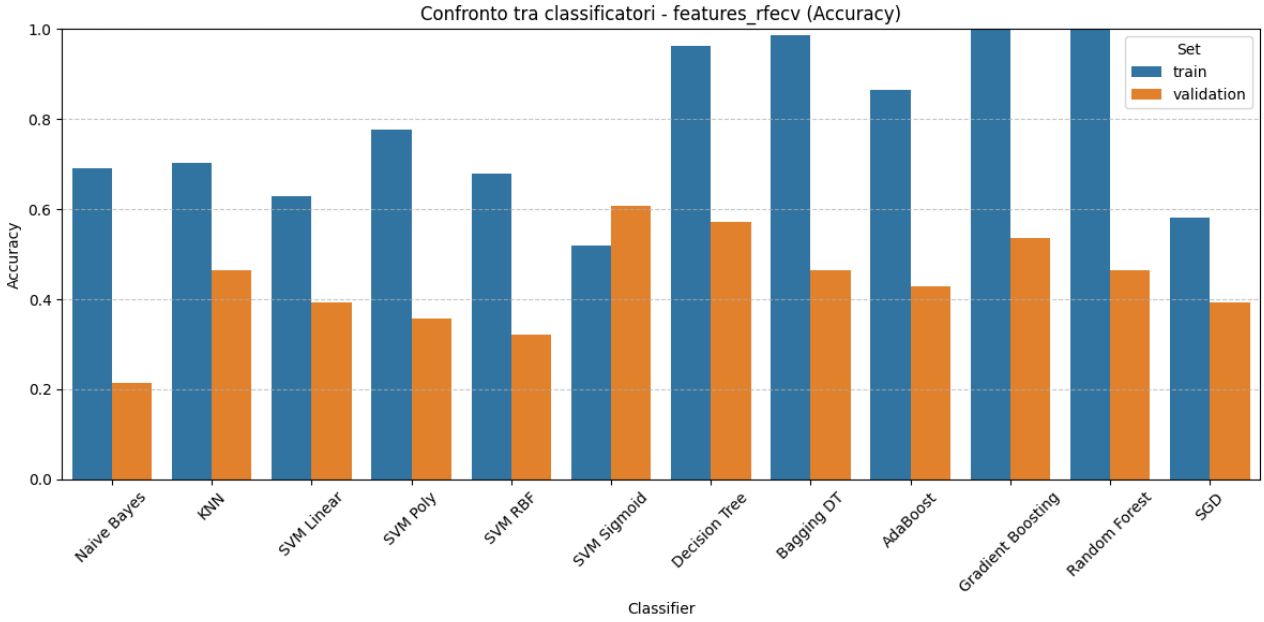


Figura 12 AUC-ROC Validation RFE

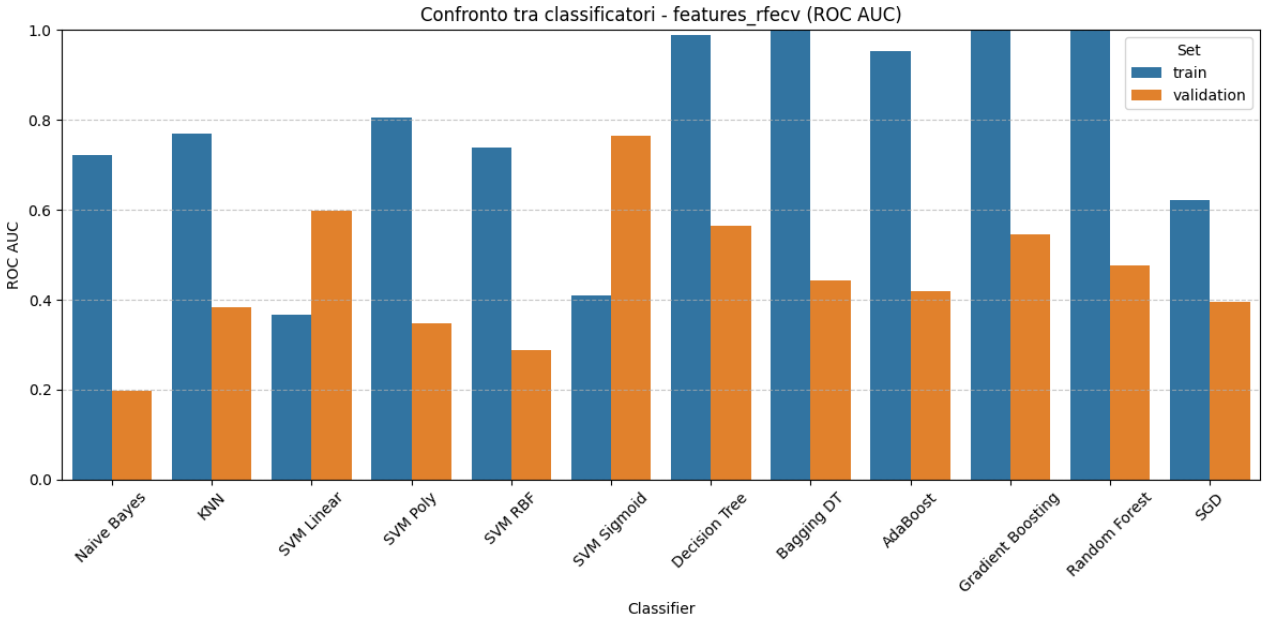


Figura 13 Accuracy Validation RFE

Il classificatore, per questo metodo di feature selection, scelto in base allo "score" calcolato sulle performance del validation set è il SVM Sigmoid.

Risultati test set

Il test set è stato riservato per validare i soli classificatori migliori di ogni metodo di feature selection. Nella seguente tabella (tab.5) vengono riportate le metriche principali calcolate:

clf_nome	Bagging DT	Bagging DT	SGD	SVM Sigmoid
roc_auc	0,765306122	0,517006803	0,673469	0,455782313
accuracy	0,535714286	0,5	0,75	0,25
bal_accuracy	0,642857143	0,523809524	0,5	0,5
F1_score_1	0,580645161	0,588235294	0,857143	0
F1_score_0	0,48	0,363636364	0	0,4
precision_0	0,333333333	0,266666667	0	0,25
precision_1	0,9	0,769230769	0,75	0
recall_0	0,857142857	0,571428571	0	1
recall_1	0,428571429	0,476190476	1	0

Tabella 5 Risultati metriche test Media

Nel seguente grafico (fig.14) vengono riportati i valori di riferimento di ciascuno dei classificatori in base ai valori risultanti dal test set:

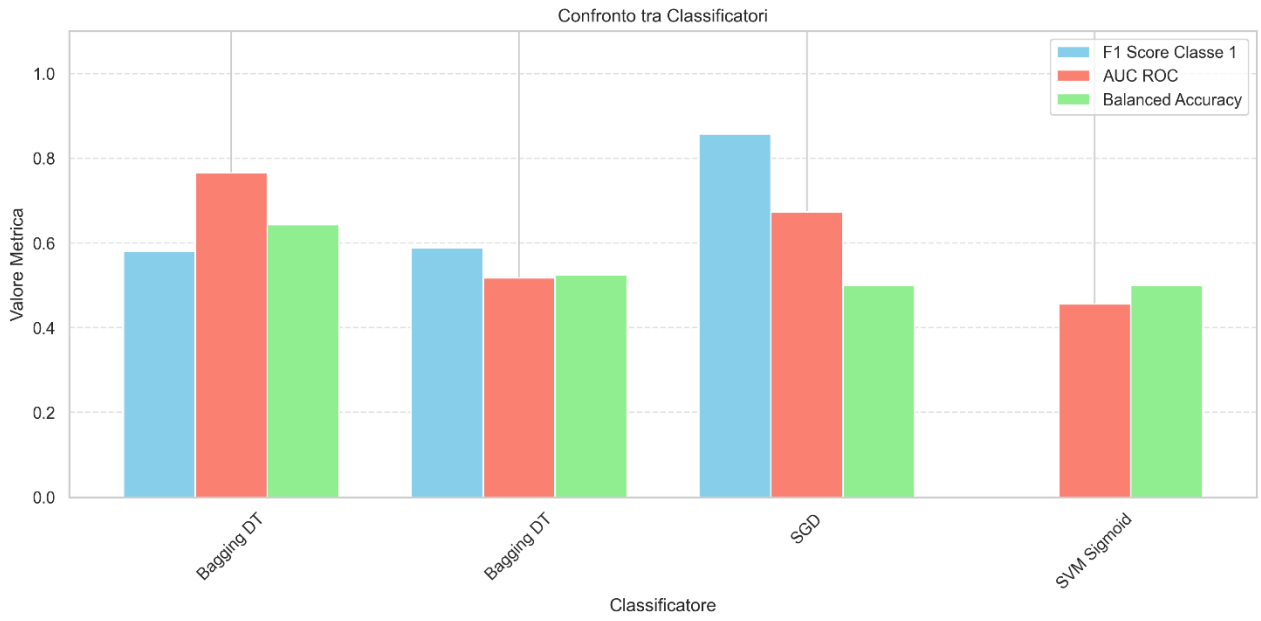


Figura 14 Principali metriche test

Le curve ROC risultanti dal test set sono riportate nella seguente tabella (tab.6):

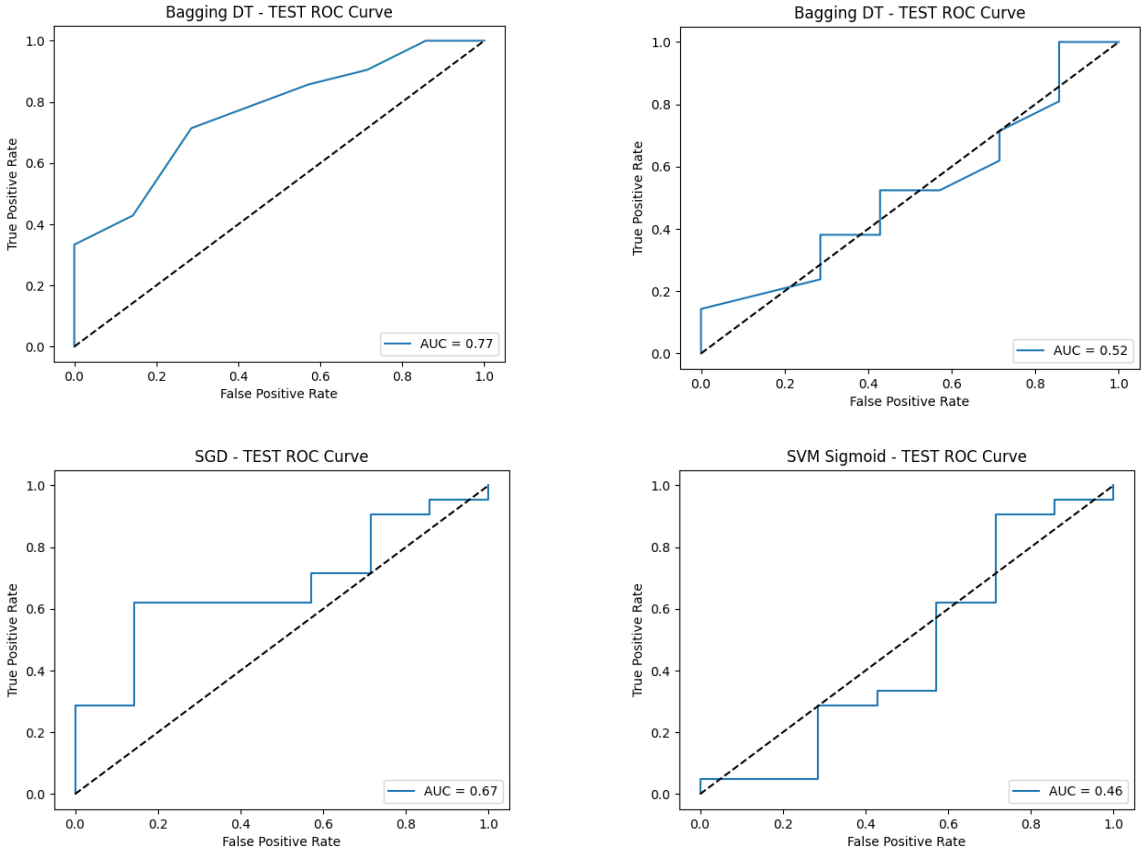
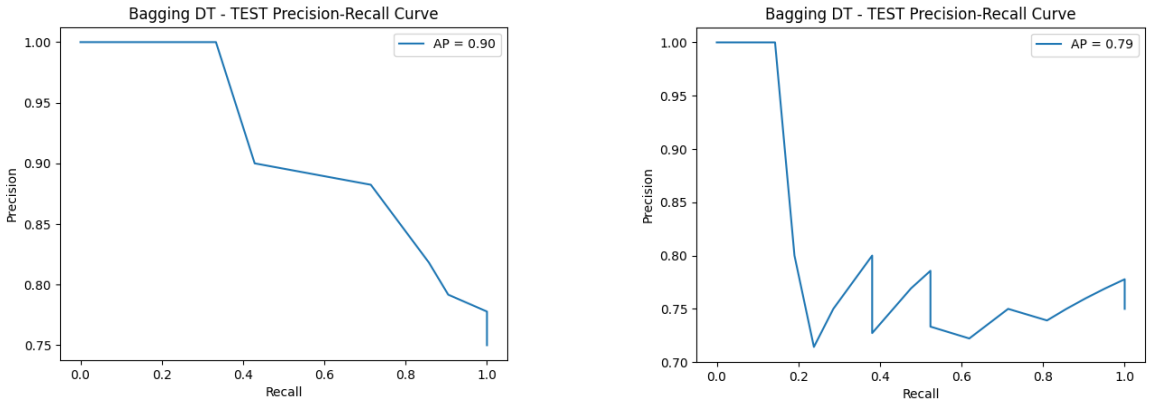
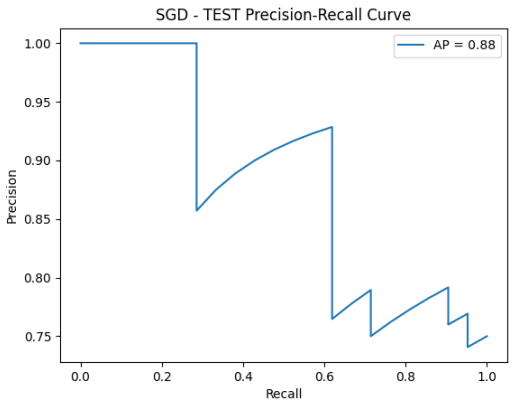


Tabella 6 Curve ROC tes set

Le curve Precision-Recall risultati dal test set sono riportati nella seguente tabella (tab.7):





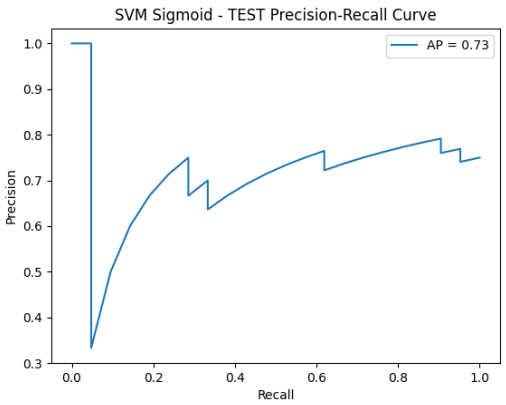


Tabella 7 Curve Precision-Recall test set

Dataset completo: risultati metodo di aggregazione calcolo statistiche descrittive dell'istogramma per ogni feature.

Affinity propagation

Le feature selezionate da questo metodo sono riportate interamente nell'appendice, in questa sezione mostriamo solo le prime dieci:

- mean_original_firstorder_Entropy
- mean_original_glcm_SumAverage
- mean_original_gldm_GrayLevelVariance
- skew_original_firstorder_Energy
- skew_original_firstorder_Entropy
- skew_original_firstorder_Kurtosis
- skew_original_firstorder_RobustMeanAbsoluteDeviation
- skew_original_glcm_Correlation
- skew_original_glcm_DifferenceVariance
- skew_original_glcm_Id

Il tuning dei parametri è riportato nella seguente tabella (tab.8):

Classificatori	Parametri Tuning
Adapting Boosting	learning_rate: 0.1, n_estimators: 100
Bagging DT	bootstrap: False, max_samples: 0.5, n_estimators: 10
Decision Tree	criterion: gini, max_depth: None, min_samples_split: 2
Gradient Boosting	learning_rate: 0.01, max_depth: 5, n_estimators: 150

KNN	n_neighbors: 3, p: 2, weights: distance
Naive Bayes	Default
Random Forest	max_depth: None, max_features: log2, min_samples_split: 2, n_estimators: 50
SGD	alpha: 0.001, loss: log_loss, penalty: elasticnet
SVM Linear	C: 0.1, gamma: scale
SVM Polinomial	C: 1, degree: 3, gamma: auto
SVM Kernel RBF	C: 1, gamma: 0.1
SVM Sigmoid	C: 10, gamma: auto

Tabella 8 Parametri Classificatori Affinity

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC (fig.15) e accuracy (fig.16).

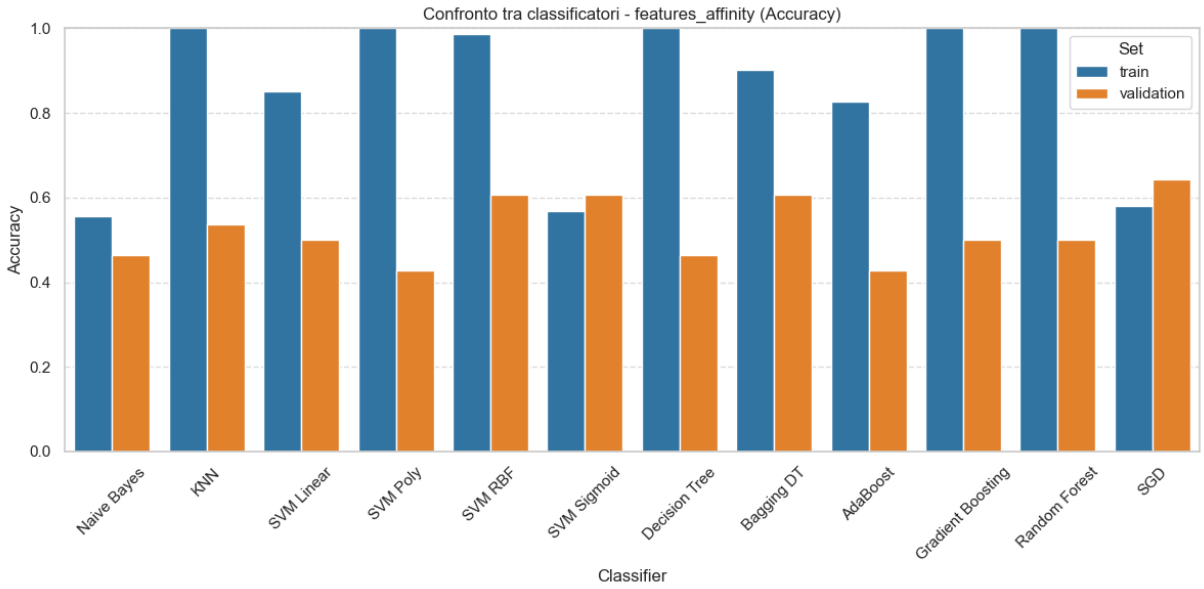


Figura 15AUC-ROC Affinity stats

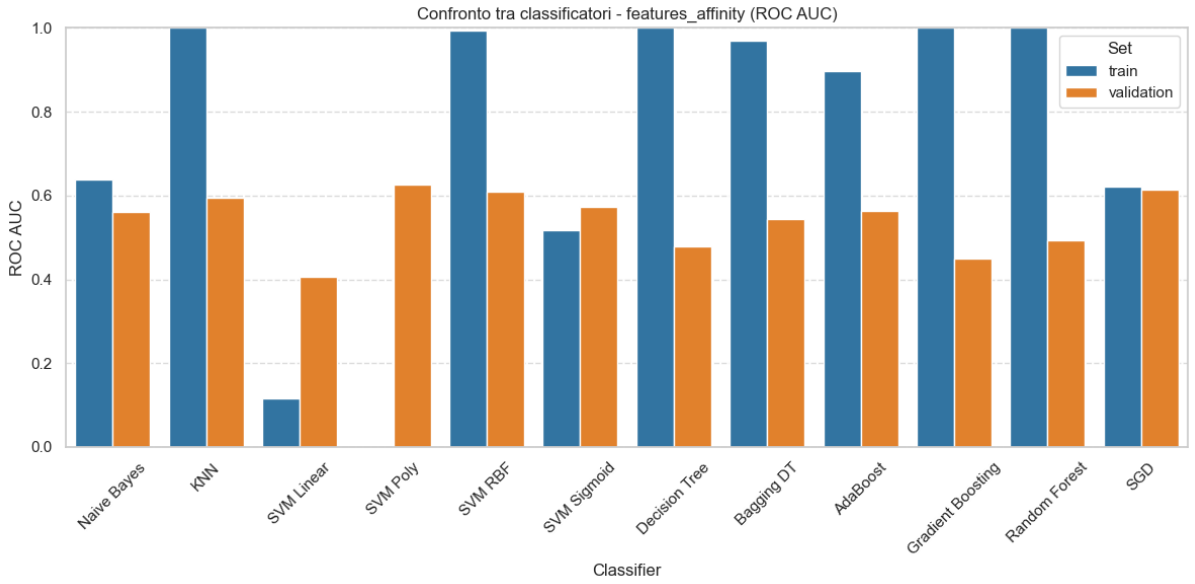


Figura 16 Accuracy Affinity stats

Il classificatore, per questo metodo di feature selection, scelto in base allo "score" calcolato sulle performance del validation set è SVM Polinomial.

Correlazione assoluta tra feature e target

Il grafico (fig.17) dei valori di correlazione è il seguente:

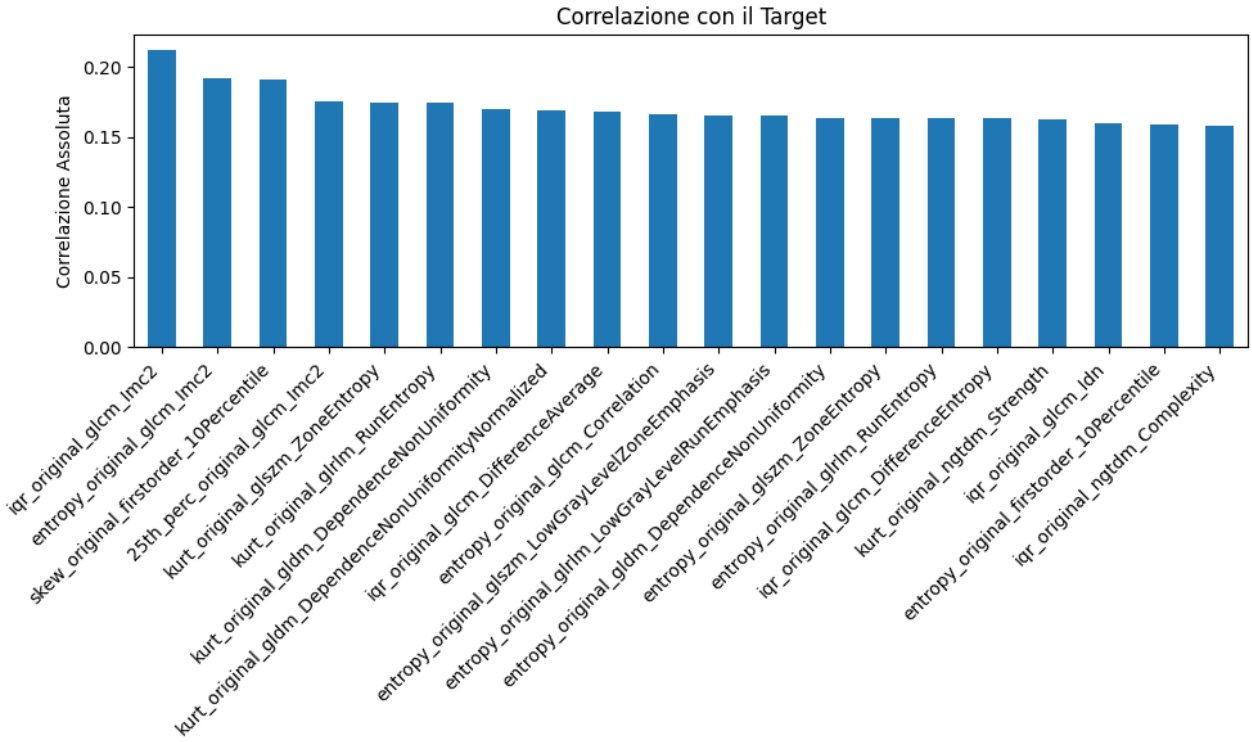


Figura 17 Valori Correlazione stats

Le 126 feature selezionate sono riportate in appendice mentre qui mostriamo solo le prime dieci:

- mean_original_firstorder_Entropy
- mean_original_glcm_Correlation
- mean_original_glcm_Imc2
- mean_original_glcm_MCC
- mean_original_glrlm_GrayLevelVariance
- mean_original_glrlm_LowGrayLevelRunEmphasis
- mean_original_glrlm_ShortRunLowGrayLevelEmphasis
- mean_original_glszm_GrayLevelVariance
- mean_original_glszm_LowGrayLevelZoneEmphasis
- mean_original_glszm_SmallAreaLowGrayLevelEmphasis

Nella prossima tabella (tab.9) sono riportati i parametri dei classificatori in seguito al tuning:

Classificatori	Parametri Tuning
Adapting Boosting	learning_rate: 0.1, n_estimators: 100
Bagging DT	bootstrap: True, max_samples: 0.7, n_estimators: 10
Decision Tree	criterion: entropy, max_depth: None, min_samples_split: 2
Gradient Boosting	learning_rate: 0.1, max_depth: 3, n_estimators: 150
KNN	n_neighbors: 9, p: 2, weights: distance
Naive Bayes	Default
Random Forest	max_depth: None, max_features: sqrt, min_samples_split: 5, n_estimators: 50
SGD	alpha: 0.001, loss: log_loss, penalty: elasticnet
SVM Linear	C: 0.01, gamma: scale
SVM Polinomial	C: 1, degree: 2, gamma: auto
SVM Kernel RBF	C: 0.1, gamma: 0.001
SVM Sigmoid	C: 10, gamma: scale

Tabella 9 Parametri Classificatori Corr.Ass stats

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC (fig.18) e accuracy (fig.19).

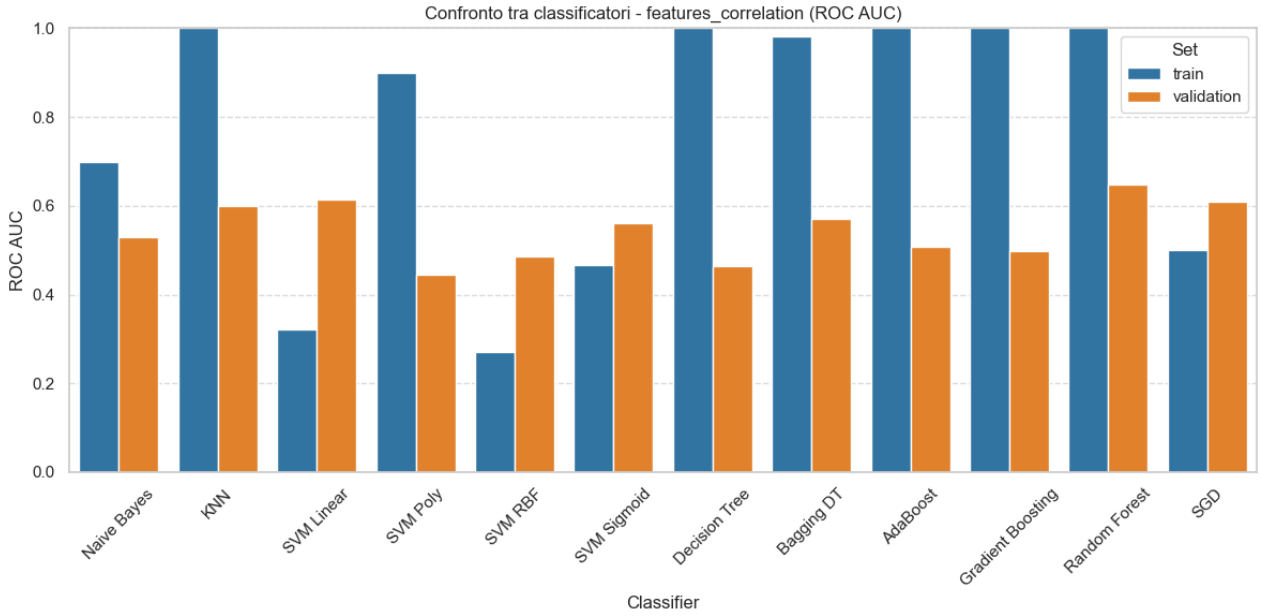


Figura 18 AUC-ROC Corr.Ass. stats

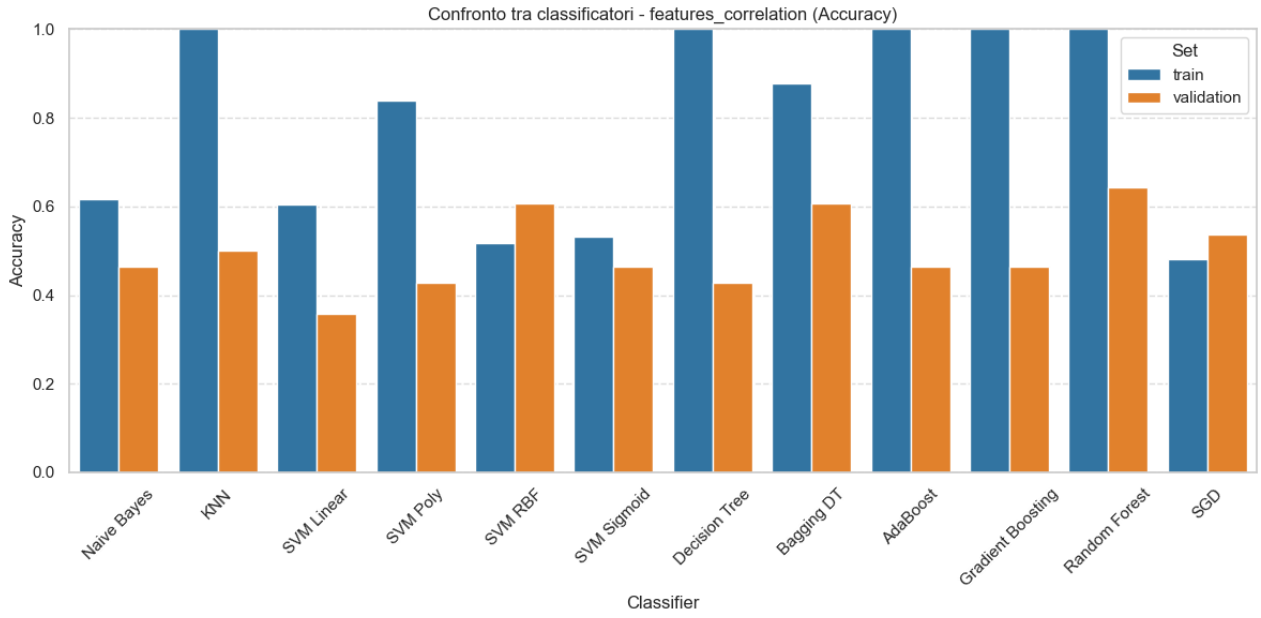


Figura 19 Accuracy Corr.Ass stats

Il classificatore, per questo metodo di feature selection, scelto in base allo “score” calcolato sulle performance del validation set è il Random Forest.

Minimun Redundancy – Maximum Relevance (MRMR)

Riportiamo il grafico (fig.20) dei valori di rilevanza:

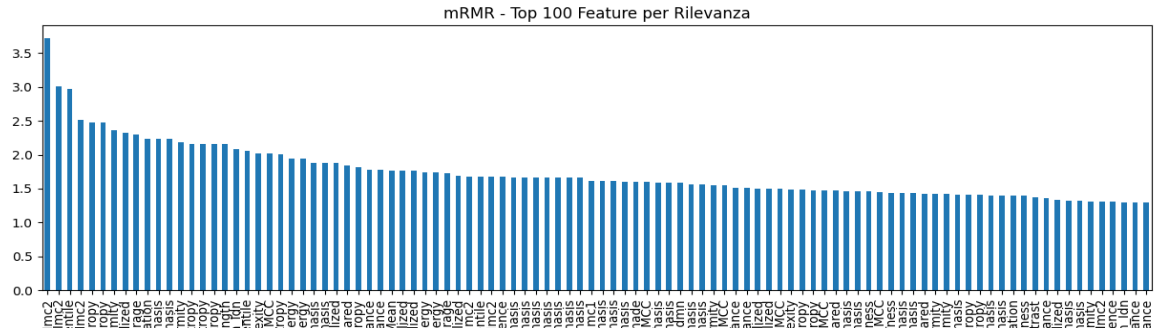


Figura 20 Valori Rilevanza MRMR stats

L'elenco completo delle feature selezionate dal modello è riportato nell'appendice, di seguito mostriamo le prime 10:

- mean_original_firstorder_Entropy
- mean_original_glcm_ClusterProminence
- mean_original_glcm_Correlation
- mean_original_glcm_Imc2
- mean_original_glcm_MCC
- mean_original_glrlm_GrayLevelVariance
- mean_original_glrlm_LowGrayLevelRunEmphasis
- mean_original_glrlm_ShortRunLowGrayLevelEmphasis
- mean_original_glszm_GrayLevelVariance
- mean_original_glszm_LowGrayLevelZoneEmphasis

Nella prossima tabella (tab.10) verranno mostrati i valori dei parametri dei classificatori in seguito al tuning:

Classificatori	Parametri Tuning
Adapting Boosting	learning_rate: 0.1, n_estimators: 200
Bagging DT	bootstrap: True, max_samples: 0.5, n_estimators: 10
Decision Tree	criterion: entropy, max_depth: None, min_samples_split: 10
Gradient Boosting	learning_rate: 0.05, max_depth: 7, n_estimators: 150
KNN	n_neighbors: 5, p: 2, weights: distance
Naive Bayes	Default
Random Forest	max_depth: None, max_features: sqrt, min_samples_split: 2, n_estimators: 50

SGD	alpha: 0.0001, loss: hinge, penalty: l2
SVM Linear	C: 0.1, gamma: scale
SVM Polinomial	C: 10, degree: 2, gamma: scale
SVM Kernel RBF	C: 1, gamma: 0.001
SVM Sigmoid	C: 10, gamma: scale

Tabella 10 Parametri Classificatori MRMR stats

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC (fig.21) e accuracy (fig.22).

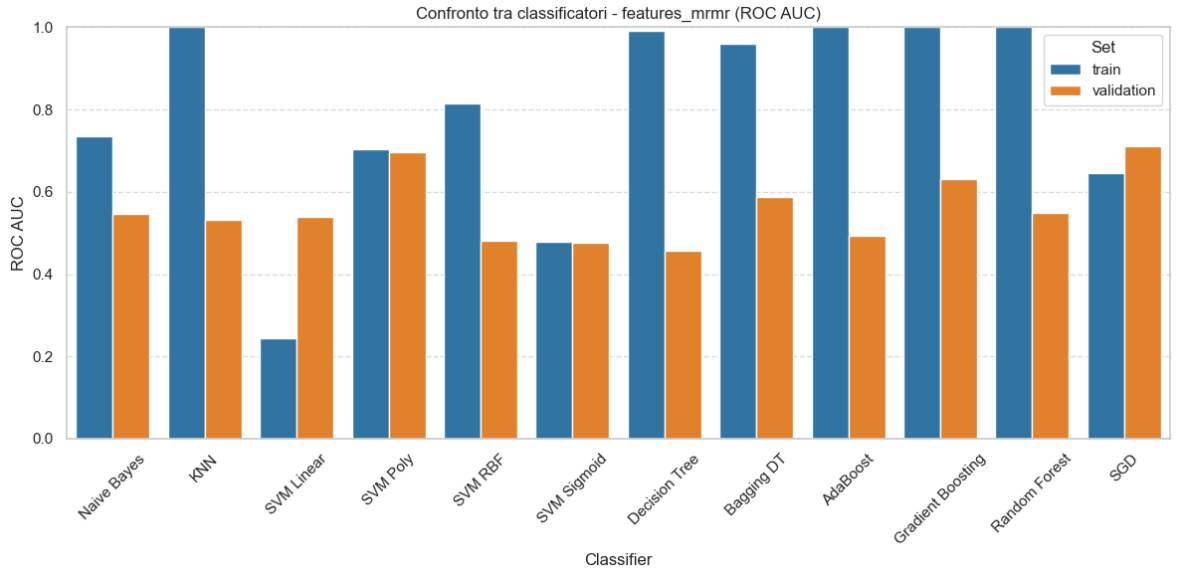


Figura 21 AUC-ROC MRMR stat

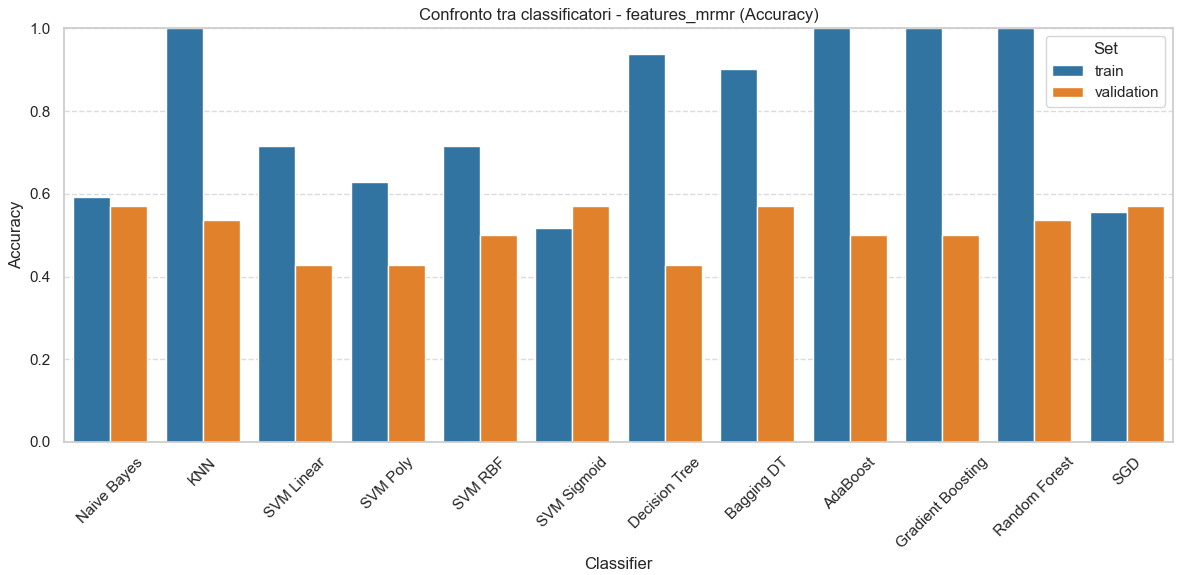


Figura 22 Accuracy MRMR stats

Il classificatore, per questo metodo di feature selection, scelto in base allo “score” calcolato sulle performance del validation set è SGD.

Recursive Feature Elimination (RFE)

Nel prossimo grafico (fig.23-24) mostriamo i valori di frequenza di selezione delle feature:

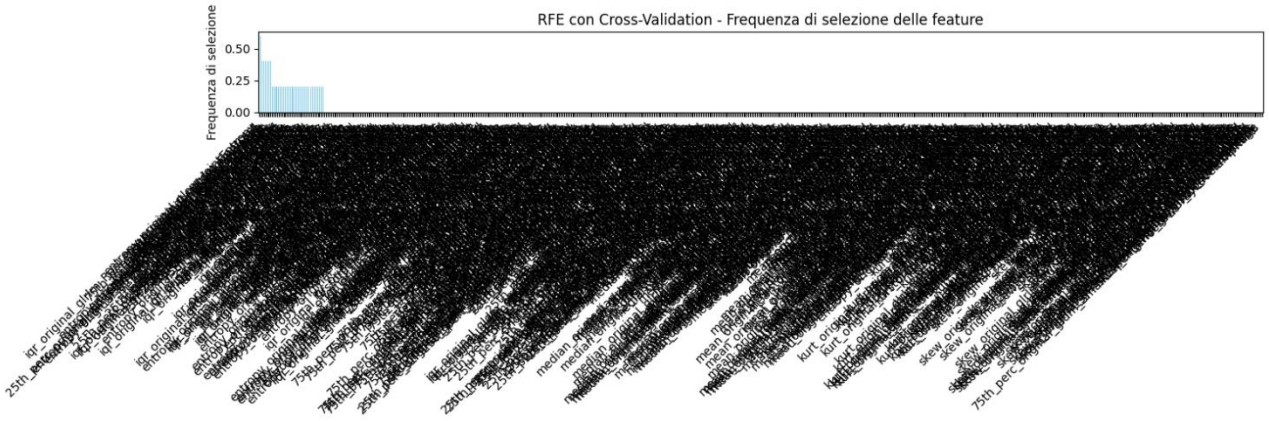


Figura 23 Frequenze di selezione RFE stats

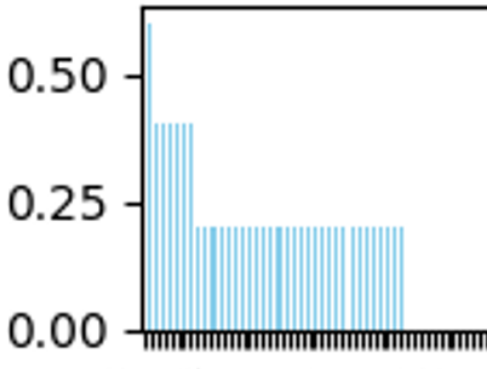


Figura 24 Zoom sulle frequenze

Le feature selezionate sono:

- entropy_original_glcm_JointEntropy
- mean_original_ngtdm_Contrast
- iqr_original_glrlm_RunLengthNonUniformityNormalized
- iqr_original_ngtdm_Busyness
- 75th_perc_original_glcm_InverseVariance
- 75th_perc_original_glcm_Imc1
- median_original_ngtdm_Contrast
- mean_original_ngtdm_Busyness

Nella prossima tabella vengono riportati i parametri dei classificatori dopo il tuning.

Classificatori	Parametri Tuning
Adapting Boosting	learning_rate: 1, n_estimators: 200
Bagging DT	bootstrap: True, max_samples: 0.5, n_estimators: 10
Decision Tree	criterion: entropy, max_depth: None, min_samples_split: 10
Gradient Boosting	learning_rate: 0.05, max_depth: 3, n_estimators: 150
KNN	n_neighbors: 7, p: 1, weights: distance
Naive Bayes	Default
Random Forest	max_depth: None, max_features: sqrt, min_samples_split: 5, n_estimators: 50
SGD	alpha: 0.001, loss: log_loss, penalty: l1
SVM Linear	C: 100, gamma: scale
SVM Polinomial	C: 10, degree: 3, gamma: scale
SVM Kernel RBF	C: 100, gamma: auto
SVM Sigmoid	C: 1, gamma: scale

Tabella 11Parametri Classificatori RFE stats

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC (fig.25) e accuracy (fig.26).

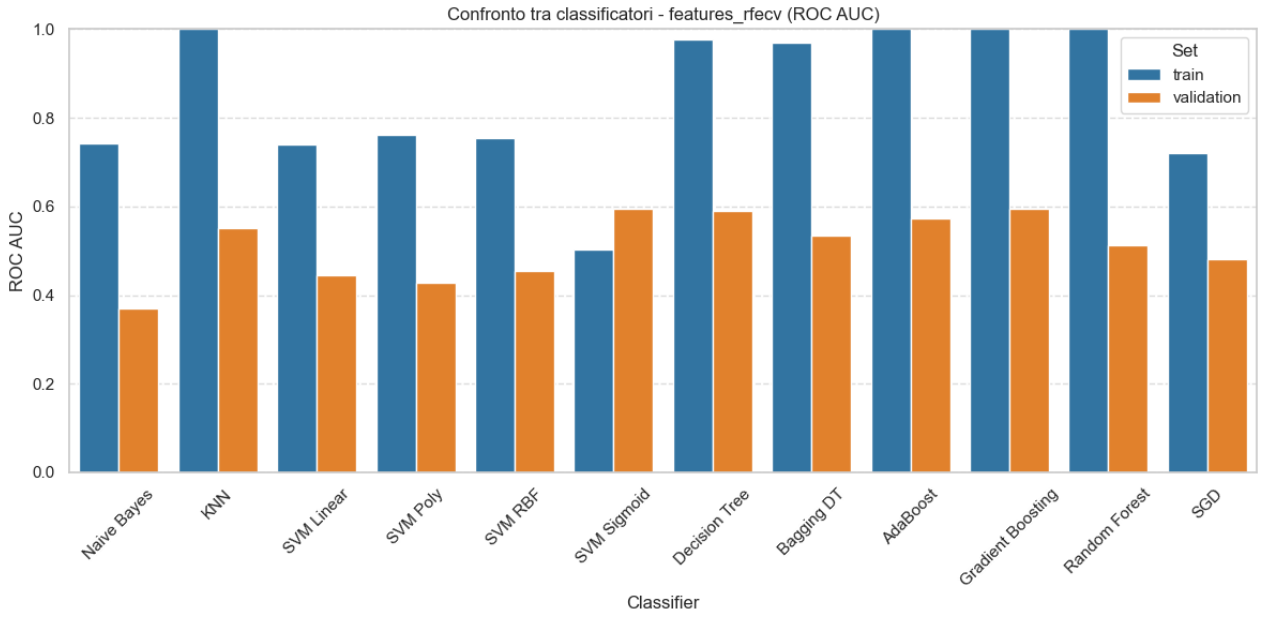


Figura 25AUC-ROC RFE stats

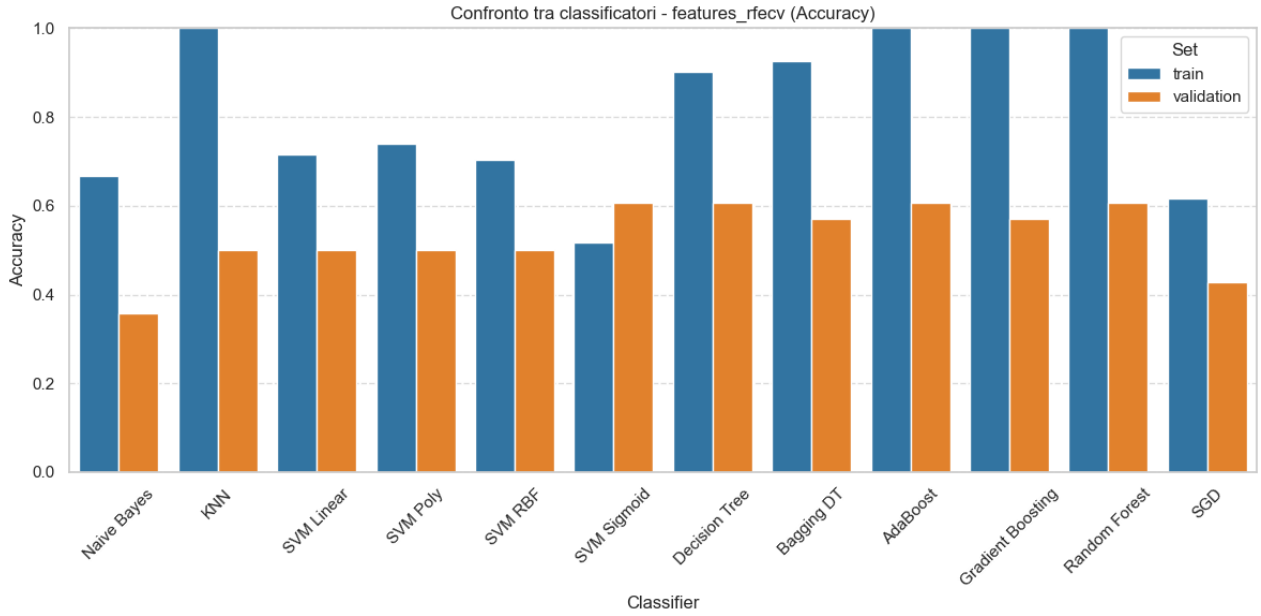


Figura 26 Accuracy RFE stats

Il classificatore, per questo metodo di feature selection, scelto in base al valore si AUC ROC, calcolato sulle performance del validation set, è il SVM Sigmoid.

Risultati test set

Nella seguente tabella (tab.12) vengono riportate le metriche calcolate sul test set dei migliori classificatori:

clf_nome	SVM Poly	Random Forest	SGD	SVM Sigmoid
roc_auc	0,5034014	0,462585034	0,319728	0,578231293
accuracy	0,4642857	0,5	0,607143	0,25
bal_accuracy	0,452381	0,476190476	0,452381	0,5
F1_score_1	0,5714286	0,611111111	0,744186	0
F1_score_0	0,2857143	0,3	0,153846	0,4
precision_0	0,2142857	0,230769231	0,166667	0,25
precision_1	0,7142857	0,733333333	0,727273	0
recall_0	0,4285714	0,428571429	0,142857	1
recall_1	0,4761905	0,523809524	0,761905	0

Tabella 12 Risultati metriche test set

Nel prossimo grafico (fig.27) vengono riportati i valori di riferimento di ciascuno dei classificatori in base ai valori risultanti dal test set:

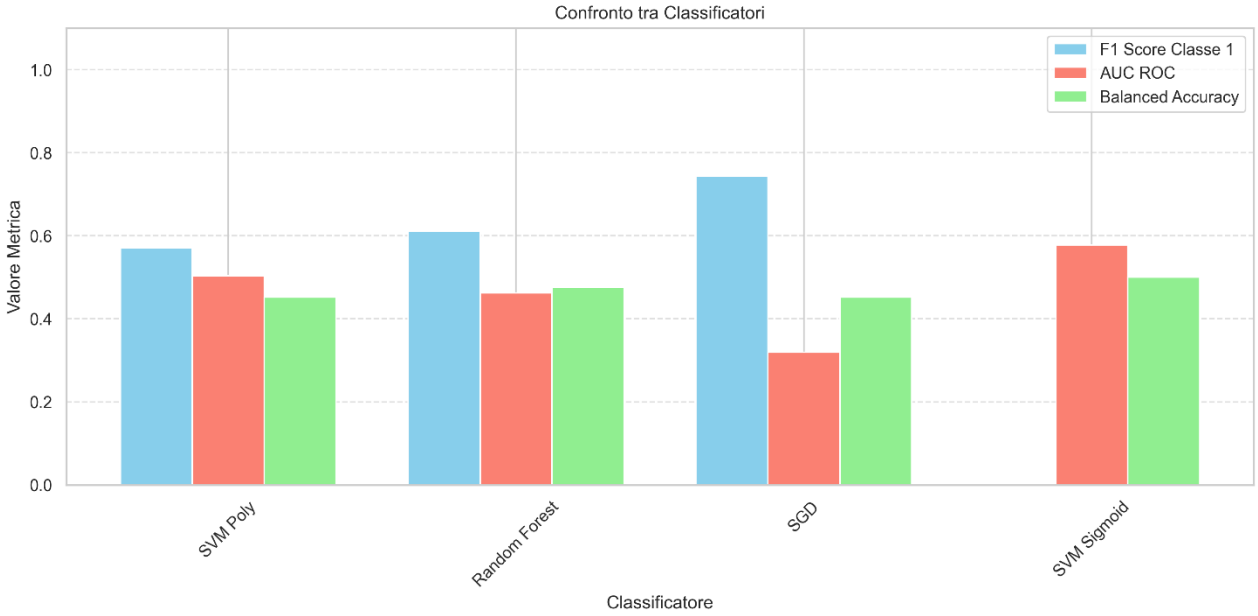


Figura 27Metriche principali test set stats

Nella prossima tabella (tab.13) vengono mostrate le curve ROC risultanti dal test set:

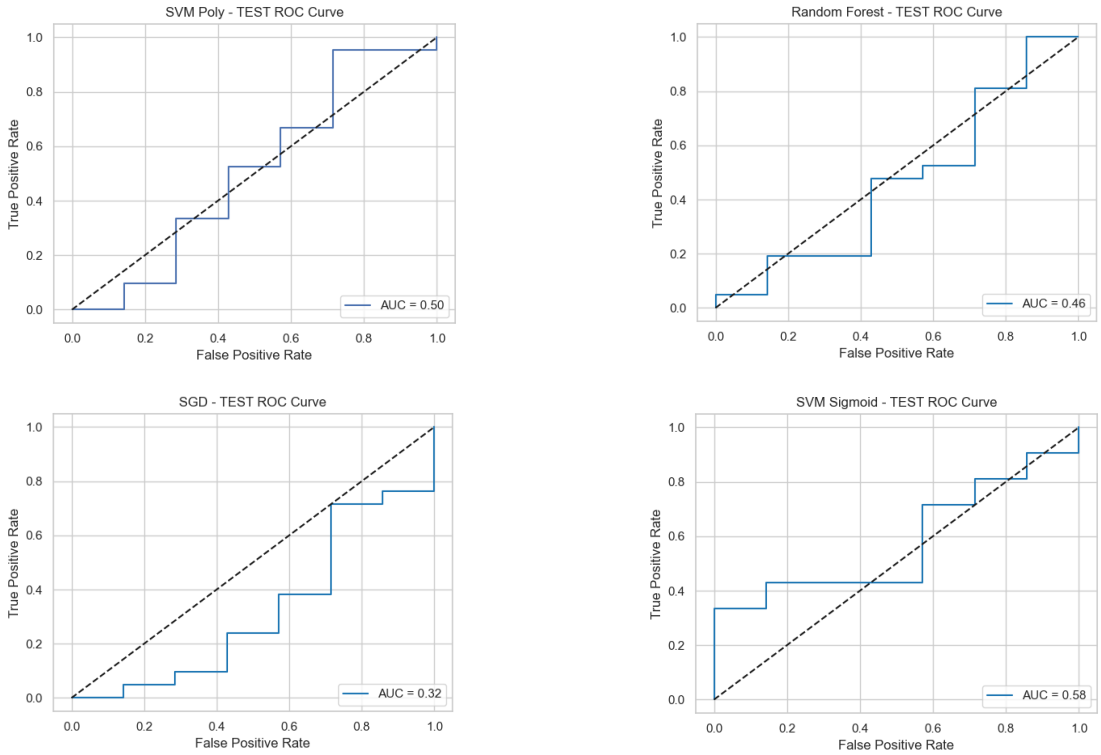


Tabella 13Curve ROC test set stats

Nella prossima tabella (tab.14) vengono mostrate le curve Precision-Recall risultanti dal test set:

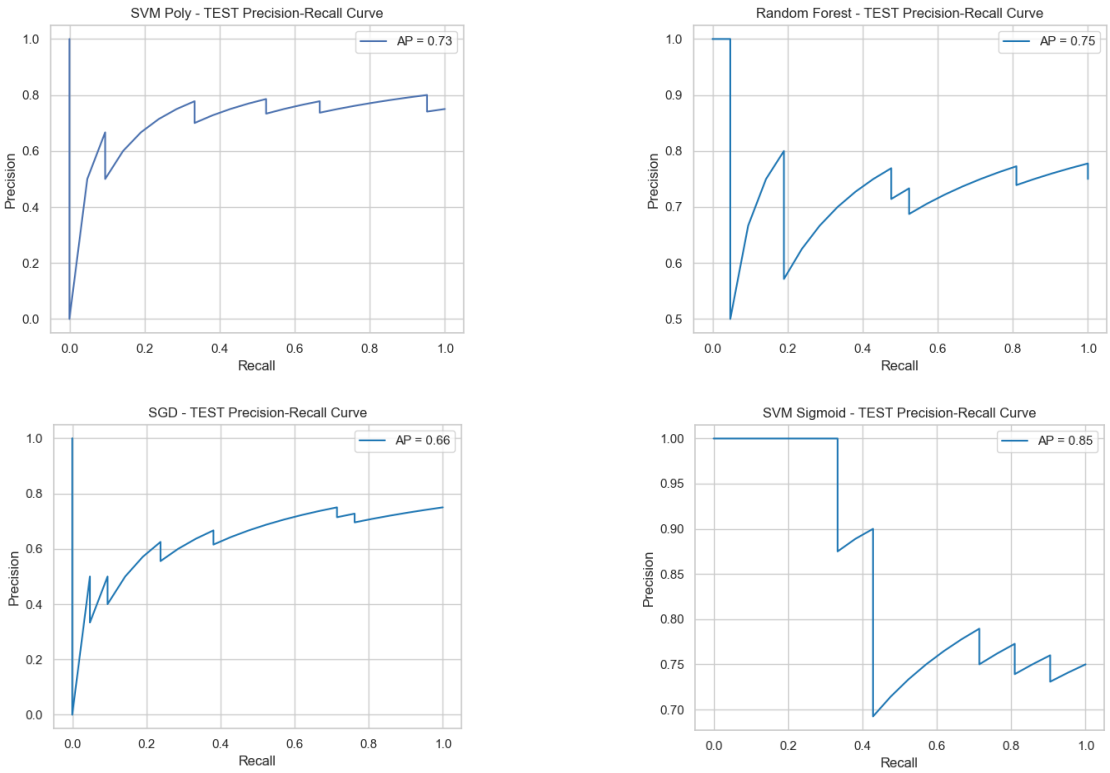


Tabella 14 Curve Precision-Recall test set stats

Risultati PCA: dataset completo

Metodo di aggregazione: calcolo della media di ogni feature

Il seguente grafico (fig.28) mostra la varianza delle feature con la soglia al 90%:

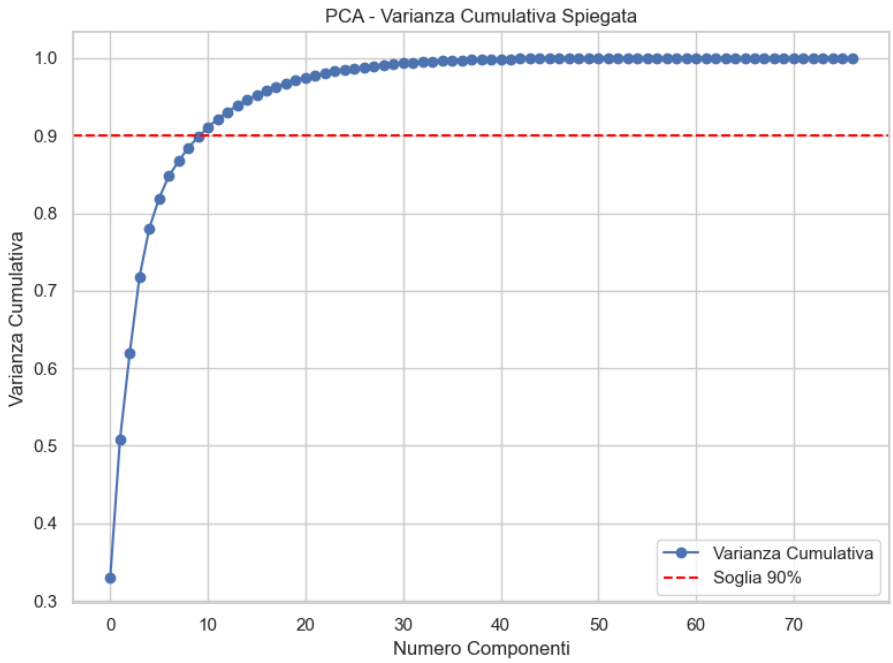


Figura 28 Varianza PCA media

I parametri dei classificatori risultanti dal tuning sono riportati nella seguente tabella (tab.15):

Classificatori	Parametri Tuning
Naive Bayes	Default
SGD	alpha: 1e-05, loss: hinge, penalty: l1
SVM Linear	C: 1, gamma: scale

Tabella 15Parametri classificatori PCA media

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AU ROC (fig.28) e accuracy (fig.29).

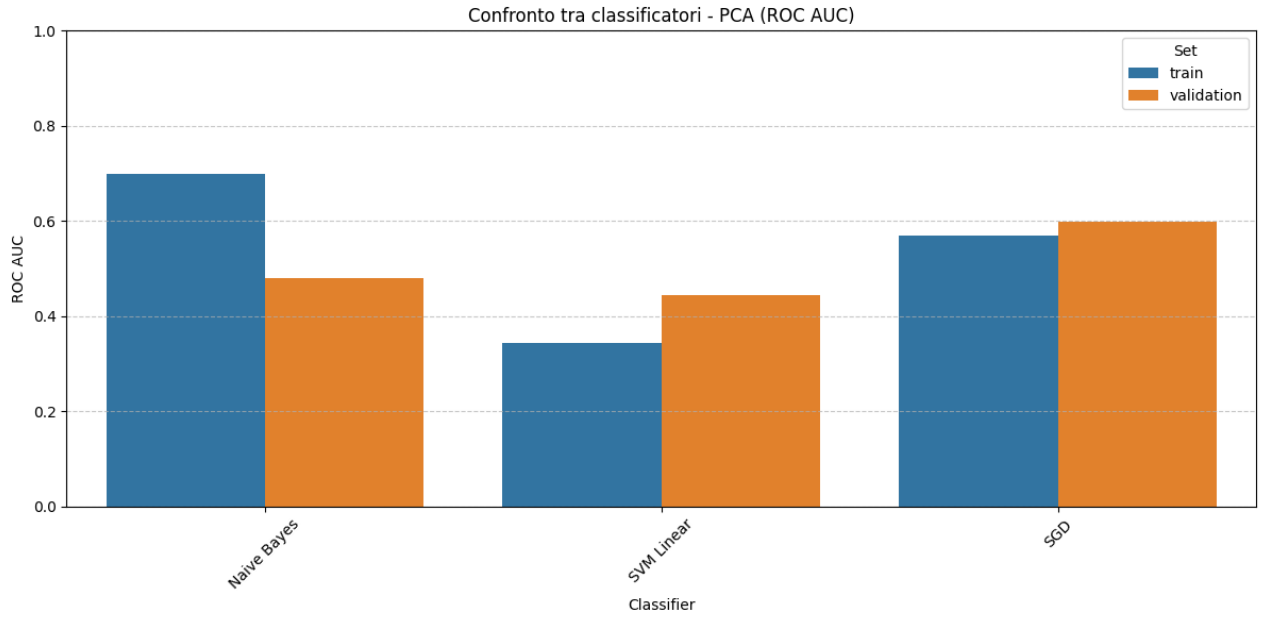


Figura 29AUC-ROC PCA media

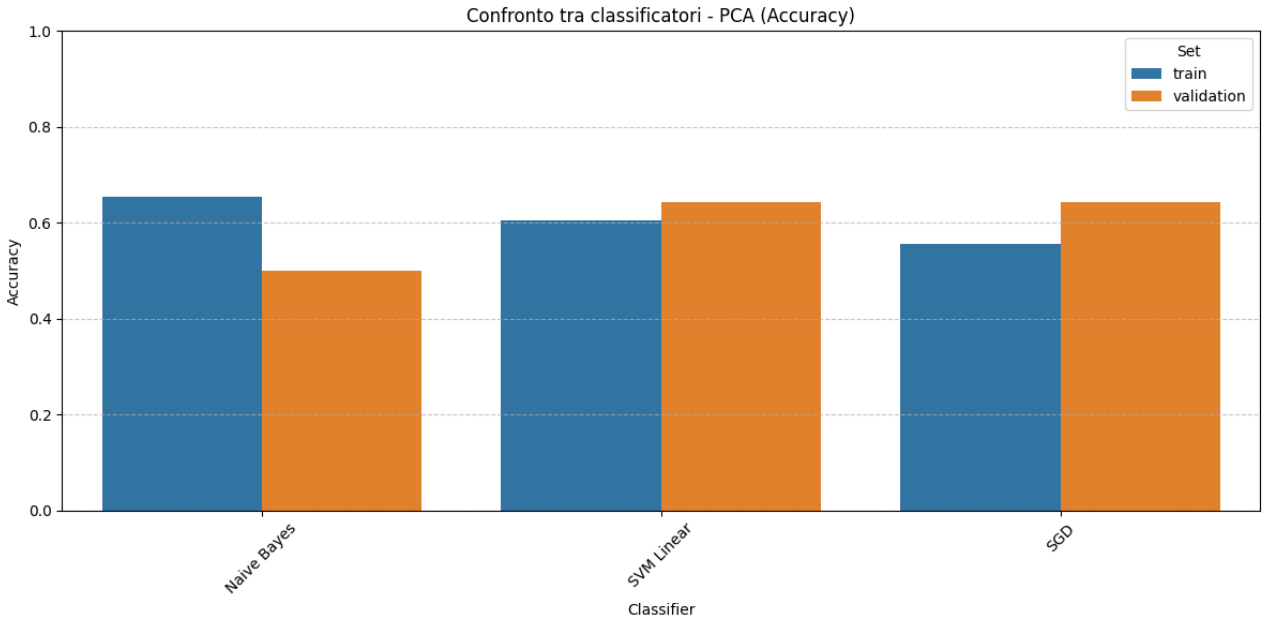


Figura 30 Accuracy PCA media

Il classificatore, per questo metodo di feature selection, scelto in base allo "score" calcolato sulle performance del validation set è SGD. I risultati delle metriche calcolate sul test set sono riportate nella seguente tabella (tab.16):

roc_auc	0,462585034
accuracy	0,428571429
bal_accuracy	0,523809524
F1_score_1	0,466666667
F1_score_0	0,384615385
precision_0	0,263157895
precision_1	0,777777778
recall_0	0,714285714
recall_1	0,333333333

Tabella 16 Risultati metriche PCA media

Di seguito riportiamo i grafici della ROC curve e curva Precision-Recall (tab.17):

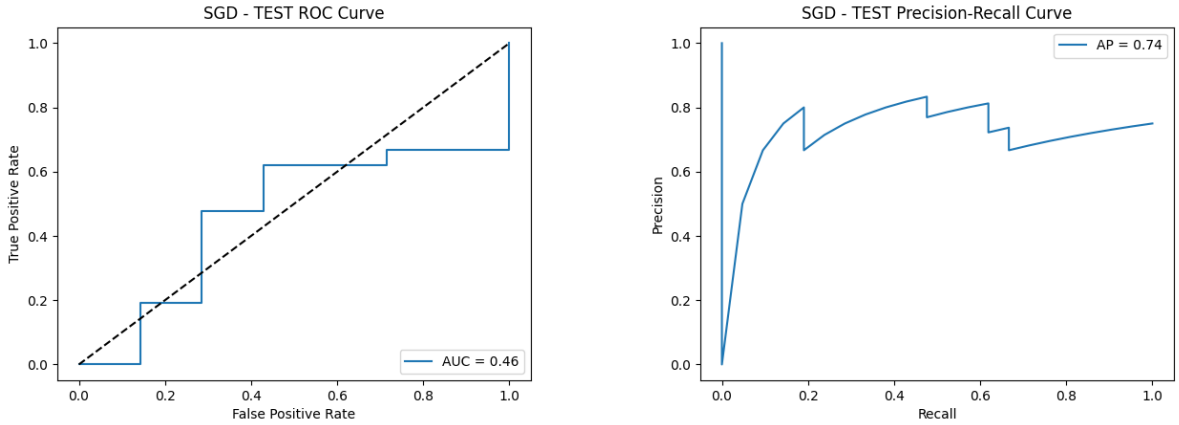


Tabella 17Curve ROC PCA media

Metodo di aggregazione: calcolo delle statistiche descrittive dell'istogramma per ogni feature.

Il seguente grafico (fig.31) mostra la varianza delle feature con la soglia al 90%:

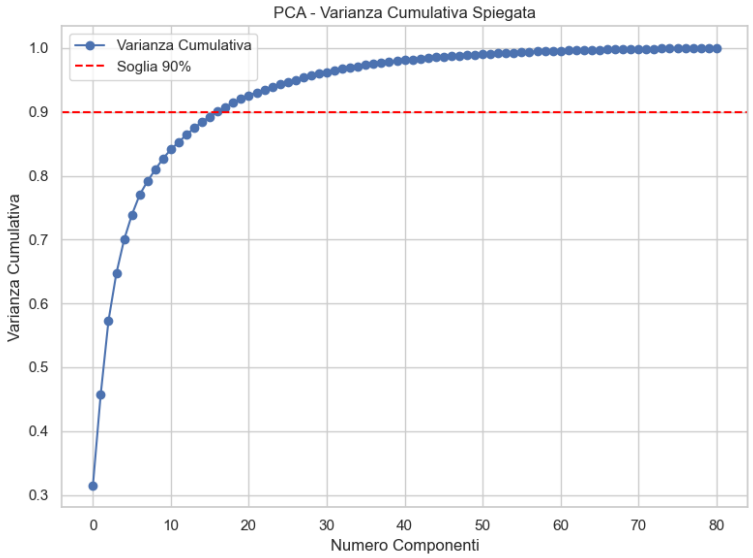


Figura 31 Varianza PCA stats

I parametri dei classificatori risultanti dal tuning sono riportati nella seguente tabella (tab.18):

Classificatori	Parametri Tuning
Naive Bayes	Default
SGD	alpha: 0.001, loss: log_loss, penalty: l1
SVM Linear	C: 1, gamma: scale

Tabella 18Parametri classificatori PCA stats

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC (fig.32) e accuracy (fig.33).

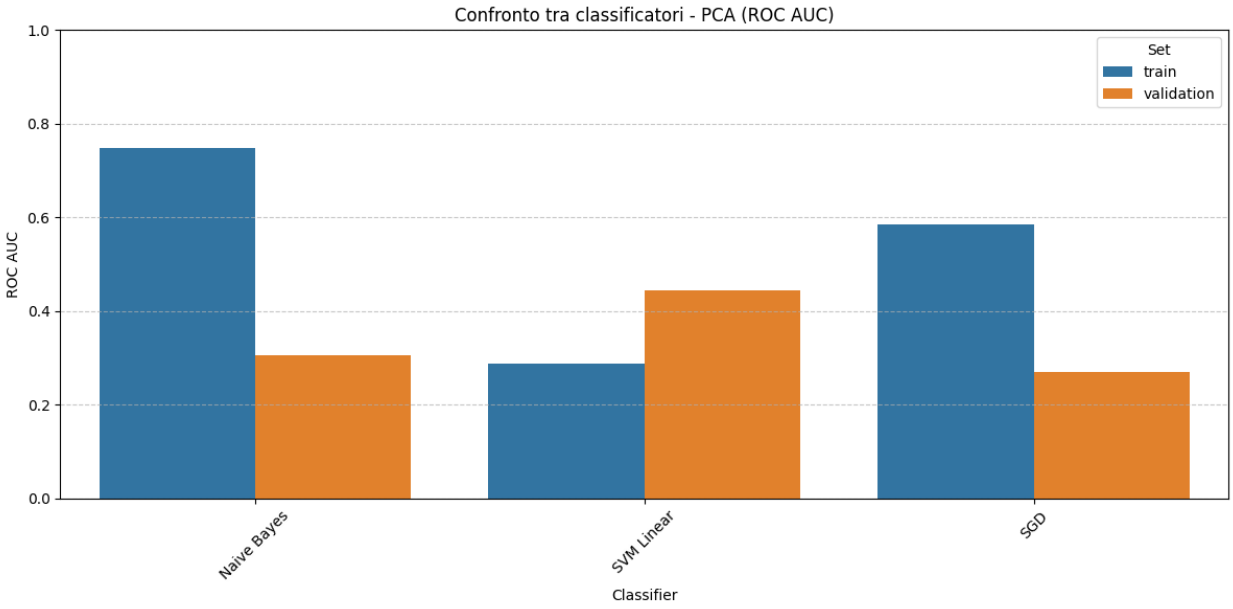


Figura 32AUC-ROC PCA stats

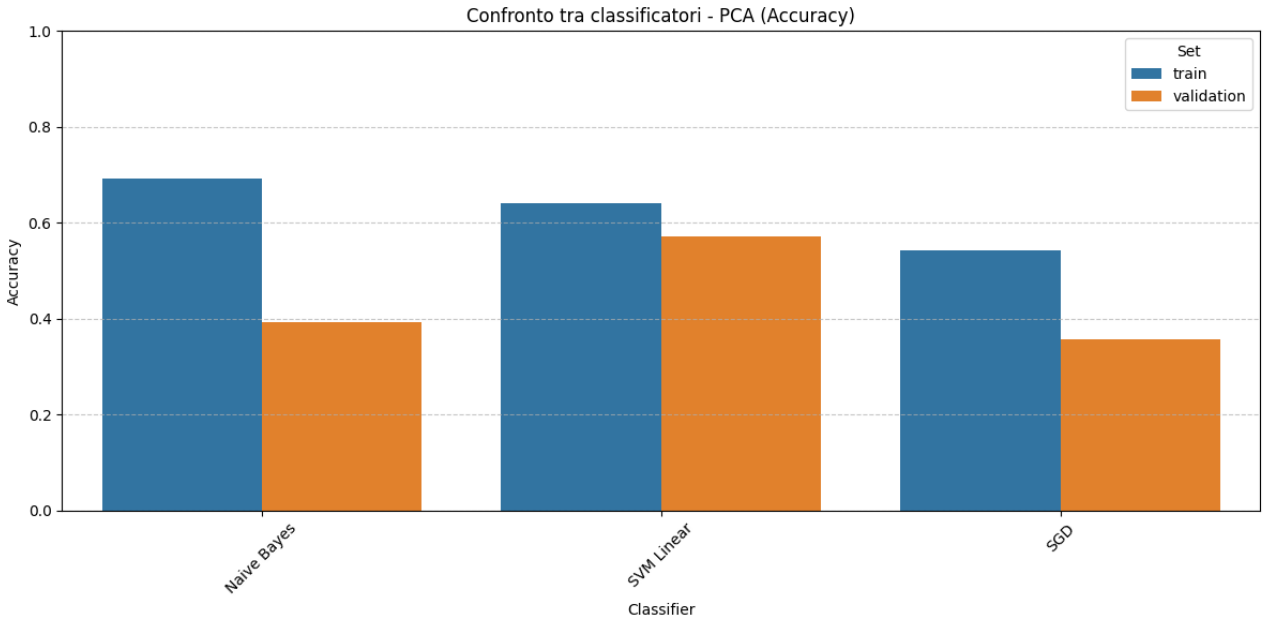


Figura 33 Accuracy PCA stats

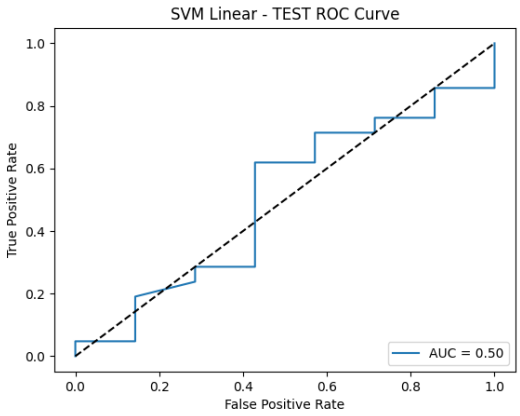
Il classificatore, per questo metodo di feature selection, scelto in base all'AUC ROC calcolato sulle performance del validation set è SVM Linear. I risultati delle metriche calcolate sul test set sono (tab.19):

roc_auc	0,5
accuracy	0,607142857
bal_accuracy	0,547619048
F1_score_1	0,717948718
F1_score_0	0,352941176
precision_0	0,3
precision_1	0,777777778
recall_0	0,428571429

recall_1	0,666666667

Tabella 19 Risultati metriche PCA stats test set

Di seguito riportiamo i grafici della ROC curve e curva Precision-Recall (tab.20):



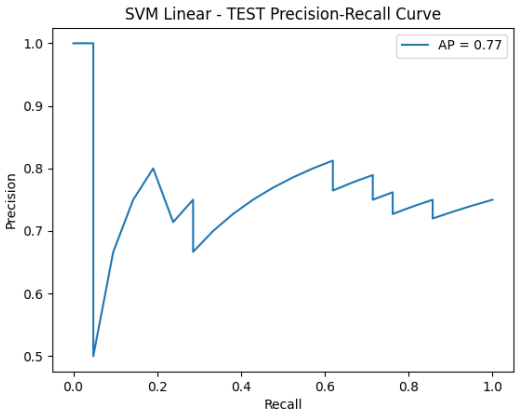


Tabella 20 Curve ROC e Precison-Recall PCA stats

Dataset privo di background: risultati metodo di aggregazione calcolo media per ogni feature.

Affinity propagation

Le feature selezionate dal metodo sono:

- mean_original_glcm_ClusterTendency
- mean_original_glcm_MCC
- mean_original_glrlm_GrayLevelVariance
- mean_original_ngtdm_Coarseness
- mean_original_ngtdm_Complexity
- mean_original_gldm_HighGrayLevelEmphasis
- mean_original_gldm_SmallDependenceEmphasis

Nella seguente tabella (tab.21) riportiamo i parametri dei classificatori scelti in seguito al tuning:

Classificatori	Parametri Tuning
Adapting Boosting	learning_rate: 0.01, n_estimators: 100
Bagging DT	bootstrap: False, max_samples: 0.1, n_estimators: 10
Decision Tree	criterion: gini, max_depth: None, min_samples_split: 5
Gradient Boosting	learning_rate: 0.01, max_depth: 7, n_estimators: 50
KNN	n_neighbors: 5, p: 1, weights: uniform

Naive Bayes	Default
Random Forest	max_depth: None, max_features: sqrt, min_samples_split: 2, n_estimators: 100
SGD	alpha: 1e-05, loss: log_loss, penalty: l1
SVM Linear	C: 100, gamma: scale
SVM Polinomial	C: 1, degree: 3, gamma: scale
SVM Kernel RBF	C: 100, gamma: scale
SVM Sigmoid	C: 0.1, gamma: scale

Tabella 21Parametri classificatori Affinity media

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC (fig.34) e accuracy (fig.35).

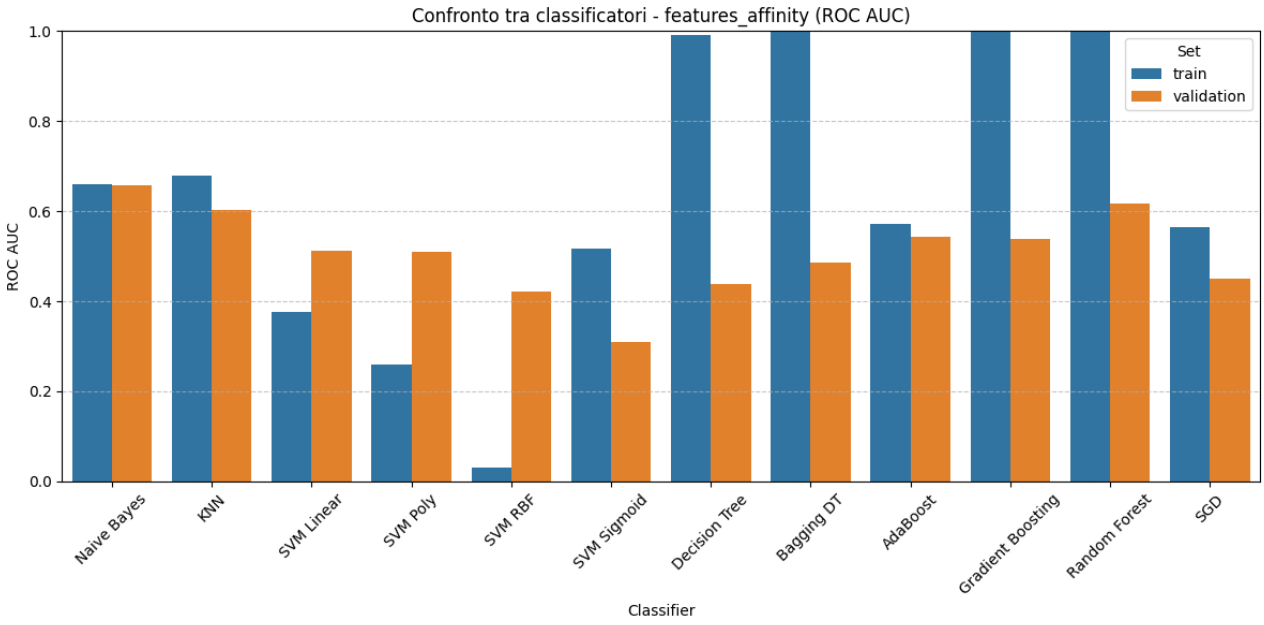


Figura 34 AUC-ROC Affinity media

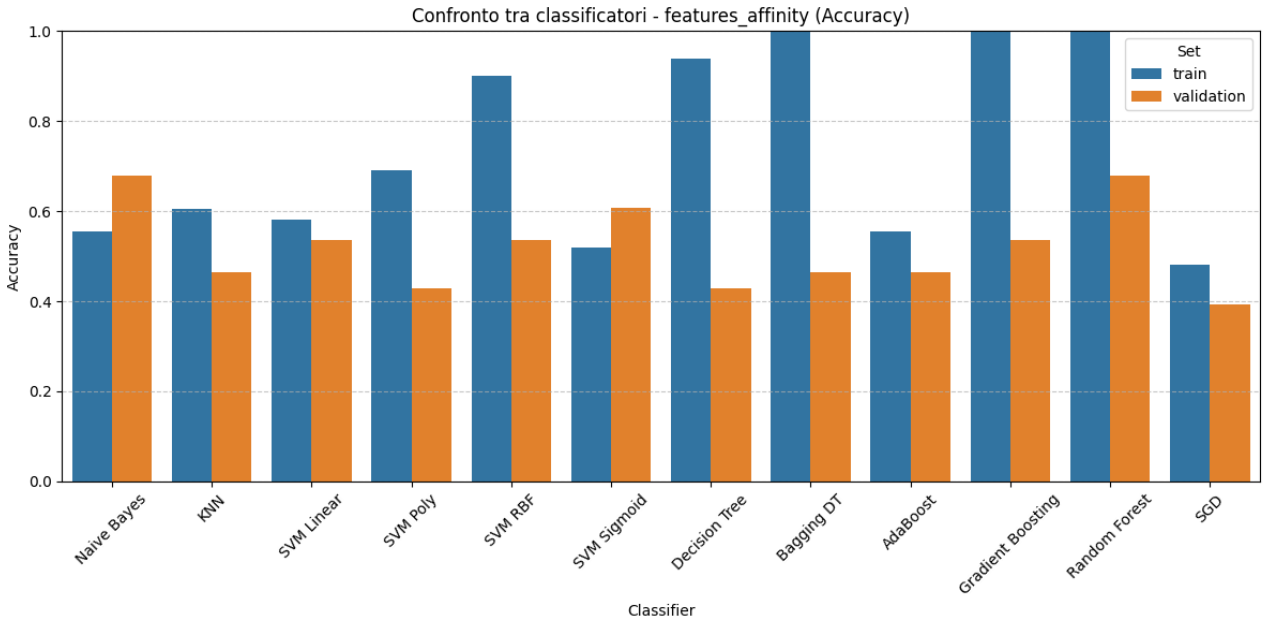


Figura 35 Accuracy Affinity media

Il classificatore, per questo metodo di feature selection, scelto in base alla metrica AUC ROC calcolato sulle performance del validation set è Naive Bayes.

Correlazione assoluta tra feature e target

Nel seguente grafico (fig.36) riportiamo il grafico dei valori di correlazione:

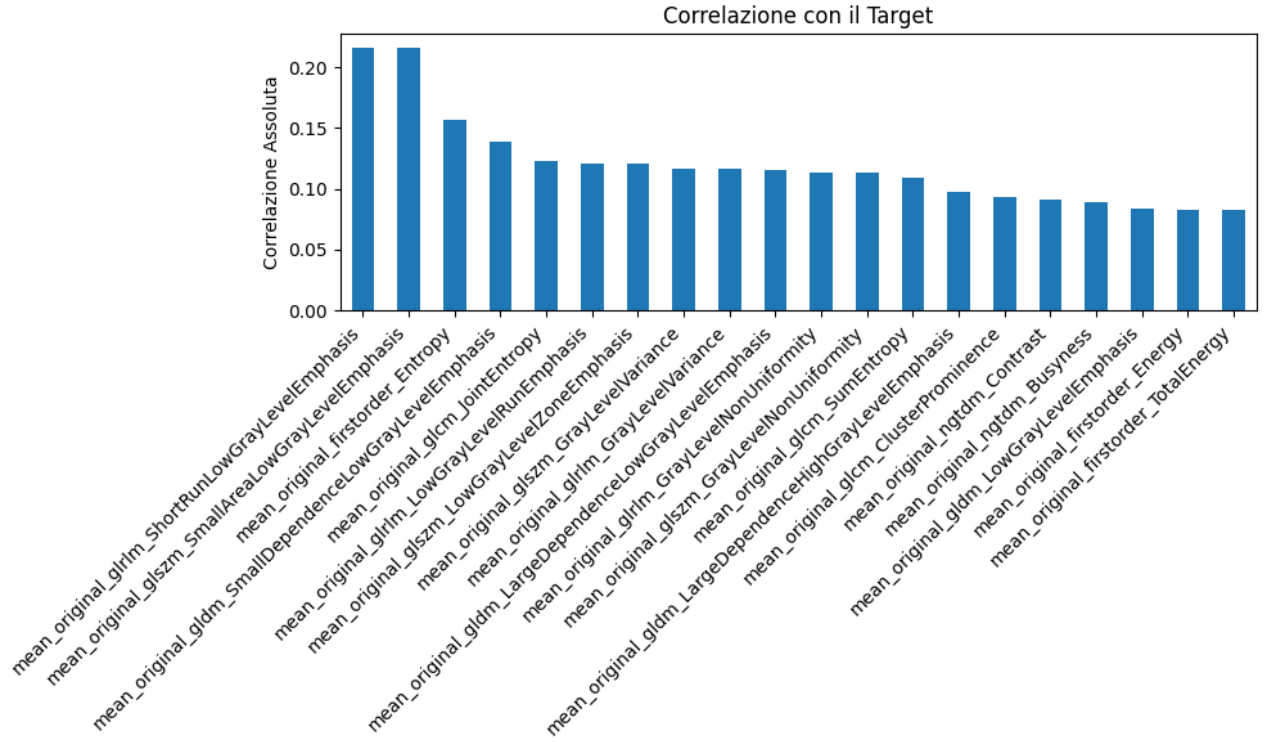


Figura 36 Valori Correlazione media

Le feature selezionate dal metodo sono:

- mean_original_firstorder_Entropy
- mean_original_glcm_ClusterProminence
- mean_original_glcm_JointEntropy
- mean_original_glcm_SumEntropy
- mean_original_glrlm_GrayLevelNonUniformity
- mean_original_glrlm_GrayLevelVariance
- mean_original_glrlm_LowGrayLevelRunEmphasis
- mean_original_glrlm_ShortRunLowGrayLevelEmphasis
- mean_original_glszm_GrayLevelNonUniformity
- mean_original_glszm_GrayLevelVariance
- mean_original_glszm_LowGrayLevelZoneEmphasis
- mean_original_glszm_SmallAreaLowGrayLevelEmphasis
- mean_original_ngtdm_Contrast
- mean_original_gldm_LargeDependenceHighGrayLevelEmphasis
- mean_original_gldm_LargeDependenceLowGrayLevelEmphasis
- mean_original_gldm_SmallDependenceLowGrayLevelEmphasis

Nella seguente tabella (tab.22) vengono riportati i valori dei parametri dei classificatori in seguito al tuning:

Classificatori	Parametri Tuning
Adapting Boosting	learning_rate: 0.01, n_estimators: 100
Bagging DT	bootstrap: False, max_samples: 0.5, n_estimators: 10
Decision Tree	criterion: gini, max_depth: 4, min_samples_split: 5
Gradient Boosting	learning_rate: 0.05, max_depth: 7, n_estimators: 100
KNN	n_neighbors: 5, p: 2, weights: uniform
Naive Bayes	Default
Random Forest	max_depth: None, max_features: sqrt, min_samples_split: 5, n_estimators: 200
SGD	alpha: 0.001, loss: log_loss, penalty: l1
SVM Linear	C: 100, gamma: scale
SVM Polinomial	C: 10, degree: 2, gamma: scale
SVM Kernel RBF	C: 100, gamma: auto
SVM Sigmoid	C: 0.1, gamma: scale

Tabella 22 Parametri Classificatori Corr.Ass.media

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC (fig.37) e accuracy (fig.38).

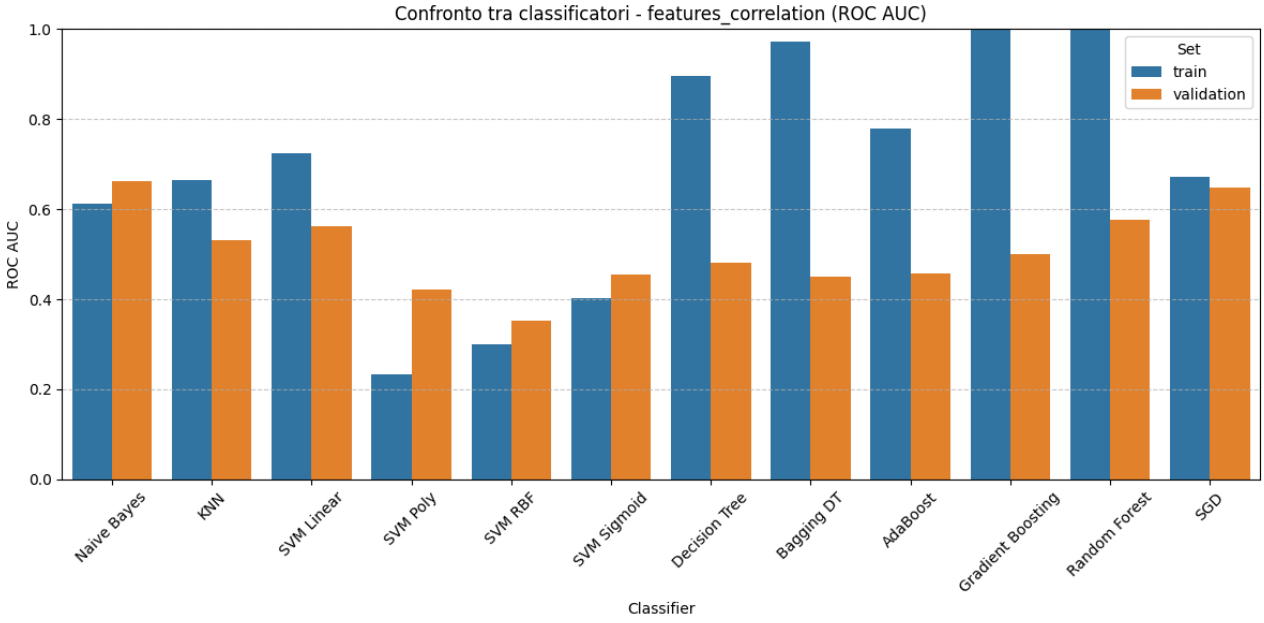


Figura 37 AUC-ROC Corr.Ass media

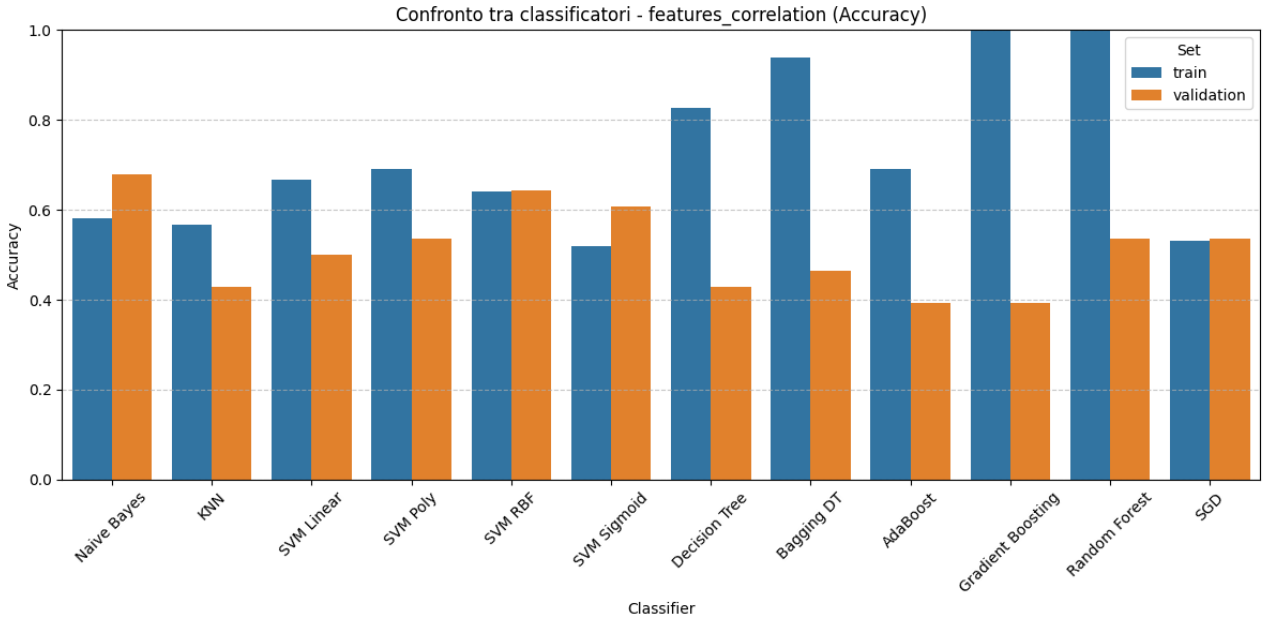


Figura 38 Accuracy Corr.Ass. media

Il classificatore, per questo metodo di feature selection, scelto in base allo “score” calcolato sulle performance del validation set è il Naive Bayes.

Minimun Redundancy – Maximum Relevance (MRMR)

Nel seguente grafico (fig.39) vengono riportati i valori di rilevanza delle feature:

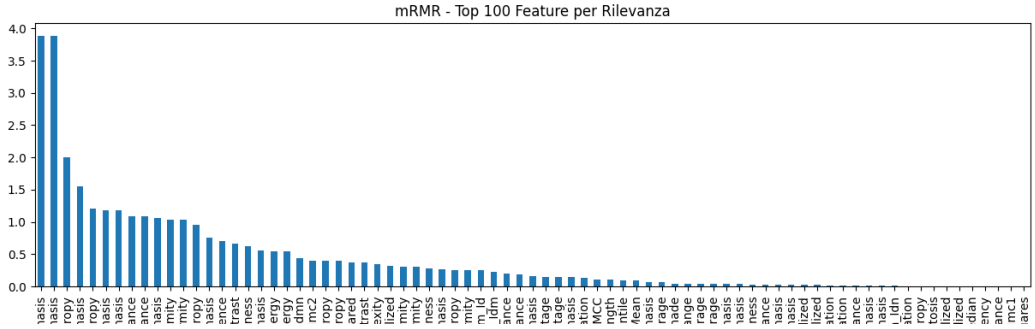


Figura 39 Valori rilevanza MRMR media

Le feature selezionate da questo metodo sono:

- mean_original_firstorder_Energy
- mean_original_firstorder_Entropy
- mean_original_glcm_JointEntropy
- mean_original_glcm_SumEntropy
- mean_original_glrlm_GrayLevelNonUniformity
- mean_original_glrlm_GrayLevelVariance
- mean_original_glrlm_LowGrayLevelRunEmphasis
- mean_original_glrlm_ShortRunLowGrayLevelEmphasis
- mean_original_glszm_GrayLevelNonUniformity
- mean_original_glszm_GrayLevelVariance
- mean_original_glszm_SmallAreaLowGrayLevelEmphasis
- mean_original_ngtdm_Busyness
- mean_original_ngtdm_Complexity
- mean_original_gldm_LargeDependenceLowGrayLevelEmphasis
- mean_original_gldm_SmallDependenceLowGrayLevelEmphasis

Nella seguente tabella (tab.23) vengono riportati i valori dei parametri dei classificatori in seguito al tuning:

Classificatori	Parametri Tuning
Adapting Boosting	learning_rate: 1, n_estimators: 100
Bagging DT	bootstrap: True, max_samples: 0.5, n_estimators: 50
Decision Tree	criterion: gini, max_depth: 4, min_samples_split: 2
Gradient Boosting	learning_rate: 0.05, max_depth: 5, n_estimators: 50
KNN	n_neighbors: 3, p: 1, weights: uniform

Naive Bayes	Default
Random Forest	max_depth: None, max_features: sqrt, min_samples_split: 2, n_estimators: 50
SGD	alpha: 1e-05, loss: hinge, penalty: l2
SVM Linear	C: 0.1, gamma: scale
SVM Polinomial	C: 10, degree: 4, gamma: scale
SVM Kernel RBF	C: 100, gamma: scale
SVM Sigmoid	C: 1, gamma: auto

Tabella 23Parametri Classificatori MRMR media

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC (fig.40) e accuracy (fig.41).

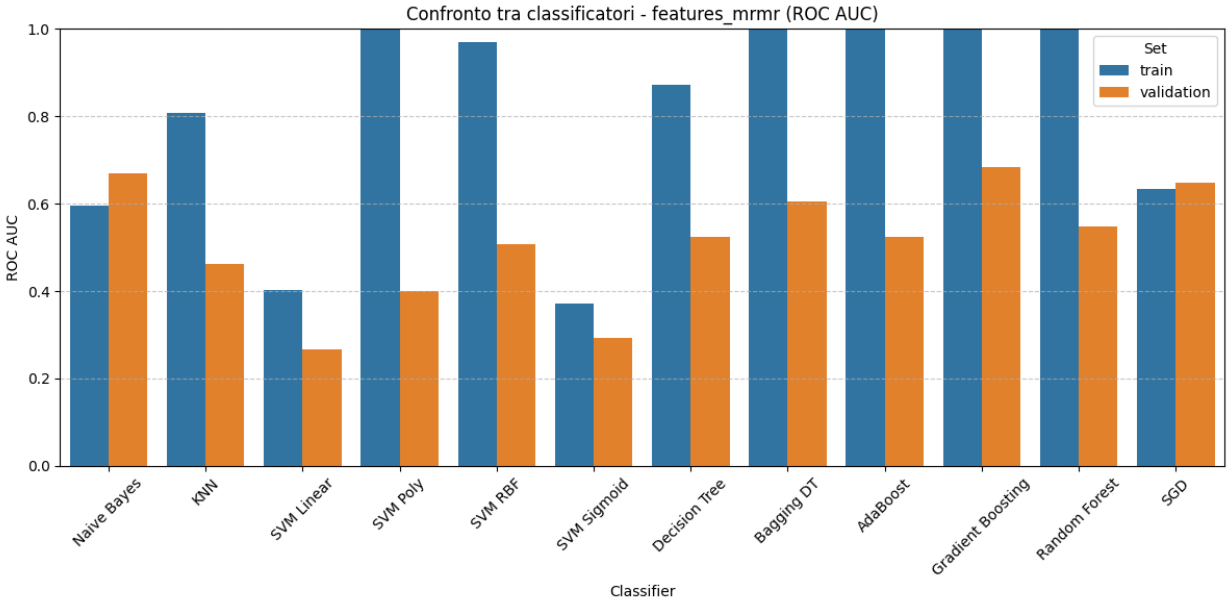


Figura 40 AUC-ROC MRMR media

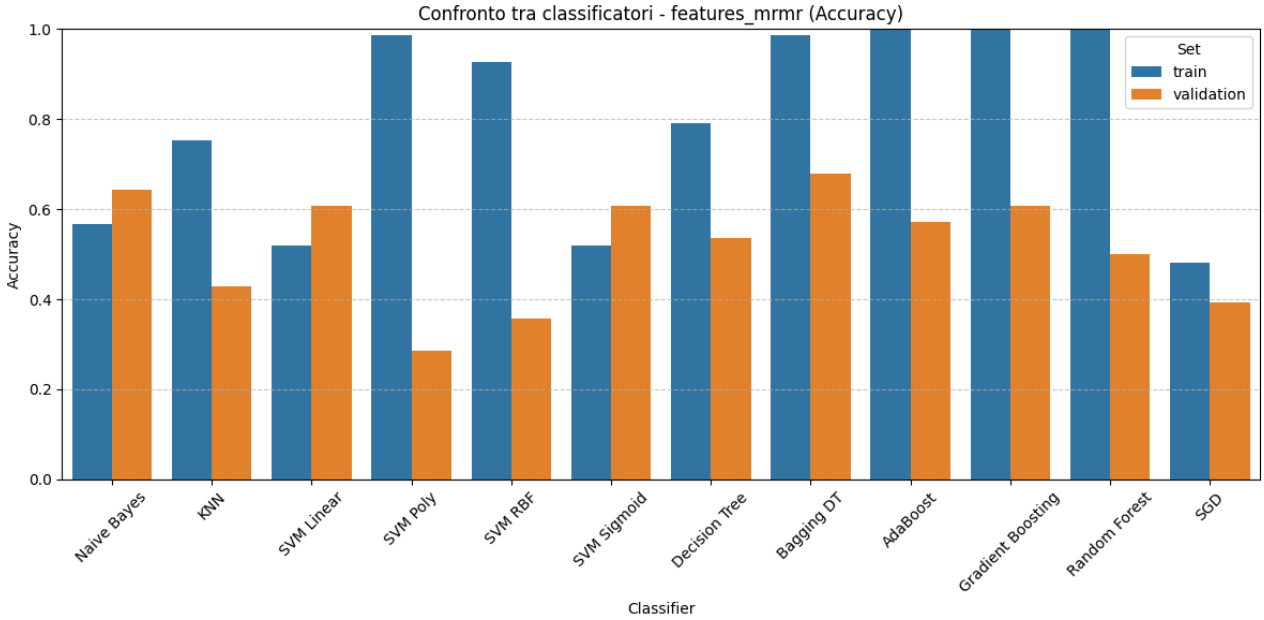


Figura 41Accuracy MRMR media

Il classificatore, per questo metodo di feature selection, scelto in base al valore di AUC ROC calcolato sulle performance del validation set è Gradient Boosting.

Recursive Feature Elimination (RFE)

Nel seguente grafico (fig. 42) sono riportate le frequenze di selezione di ogni feature:

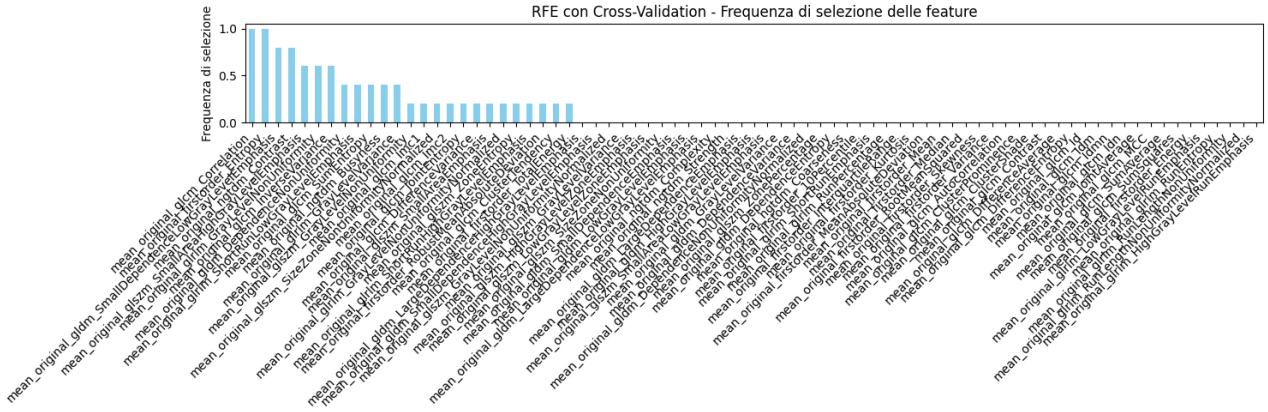


Figura 42 Frequenze di selezione RFE media

Le feature selezionate dal metodo sono:

- mean_original_glcm_Correlation
- mean_original_firstorder_Entropy
- mean_original_gldm_SmallDependenceLowGrayLevelEmphasis
- mean_original_ngtdm_Contrast
- mean_original_glszm_SmallAreaHighGrayLevelEmphasis
- mean_original_glrlm_GrayLevelNonUniformity
- mean_original_glcm_InverseVariance

Nella seguente (tab.24) tabella vengono riportati i valori dei parametri dei classificatori in seguito al tuning:

Classificatori	Parametri Tuning
Adapting Boosting	learning_rate: 1, n_estimators: 50
Bagging DT	bootstrap: True, max_samples: 0.5, n_estimators: 10
Decision Tree	criterion: entropy, max_depth: None, min_samples_split: 5
Gradient Boosting	learning_rate: 0.1, max_depth: 7, n_estimators: 50
KNN	n_neighbors: 7, p: 1, weights: uniform
Naive Bayes	Default
Random Forest	max_depth: None, max_features: sqrt, min_samples_split: 5, n_estimators: 50
SGD	alpha: 0.001, loss: log_loss, penalty: l1
SVM Linear	C: 100, gamma: scale
SVM Polinomial	C: 0.1, degree: 3, gamma: scale
SVM Kernel RBF	C: 100, gamma: scale
SVM Sigmoid	C: 1, gamma: auto

Tabella 24 Parametri classificatori RFE media

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC (fig.43) e accuracy (fig.44).

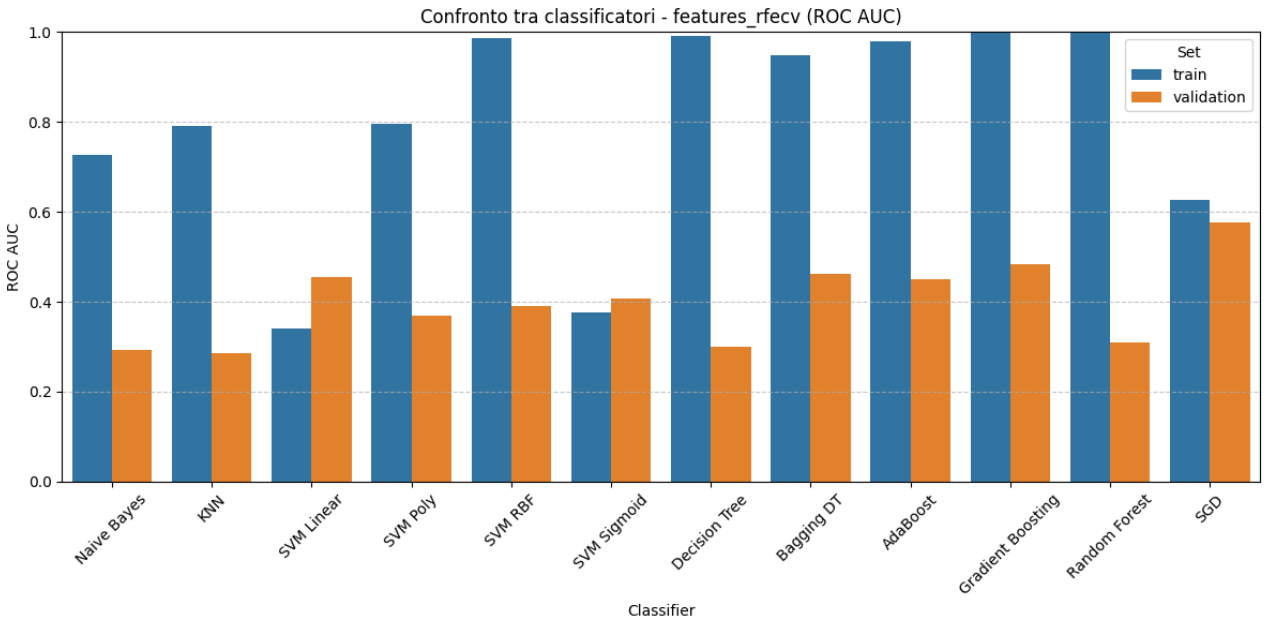


Figura 43AUC-ROC RFE media

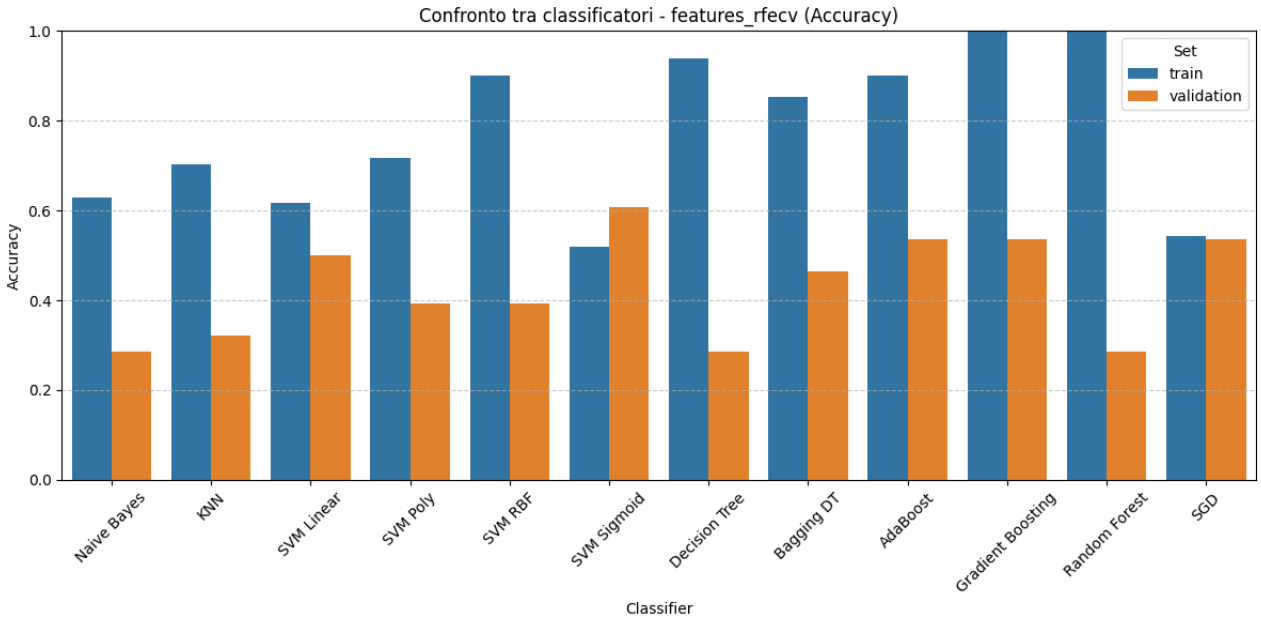


Figura 44 ccuracy RFE media

Il classificatore, per questo metodo di feature selection, scelto in base al valore di AUC ROC calcolato sulle performance del validation set è SGD.

Risultati test set

Nella seguente tabella (tab.25) vengono riportate le metriche calcolate sul test set dei migliori classificatori:

clf_nome	Naive Bayes	Naive Bayes	Gradient Boosting	SGD
roc_auc	0,421768707	0,510204082	0,489795918	0,700680272
accuracy	0,5	0,428571429	0,392857143	0,25
bal_accuracy	0,476190476	0,476190476	0,404761905	0,5
F1_score_1	0,611111111	0,5	0,484848485	0
F1_score_0	0,3	0,333333333	0,260869565	0,4
precision_0	0,230769231	0,235294118	0,1875	0,25
precision_1	0,733333333	0,727272727	0,666666667	0
recall_0	0,428571429	0,571428571	0,428571429	1
recall_1	0,523809524	0,380952381	0,380952381	0

Tabella 25RIsultati metriche test set Dataset NO backgorund media

Nel seguente grafico (fig.45) vengono riportati i valori di riferimento di ciascuno dei classificatori in base ai valori risultanti dal test set:

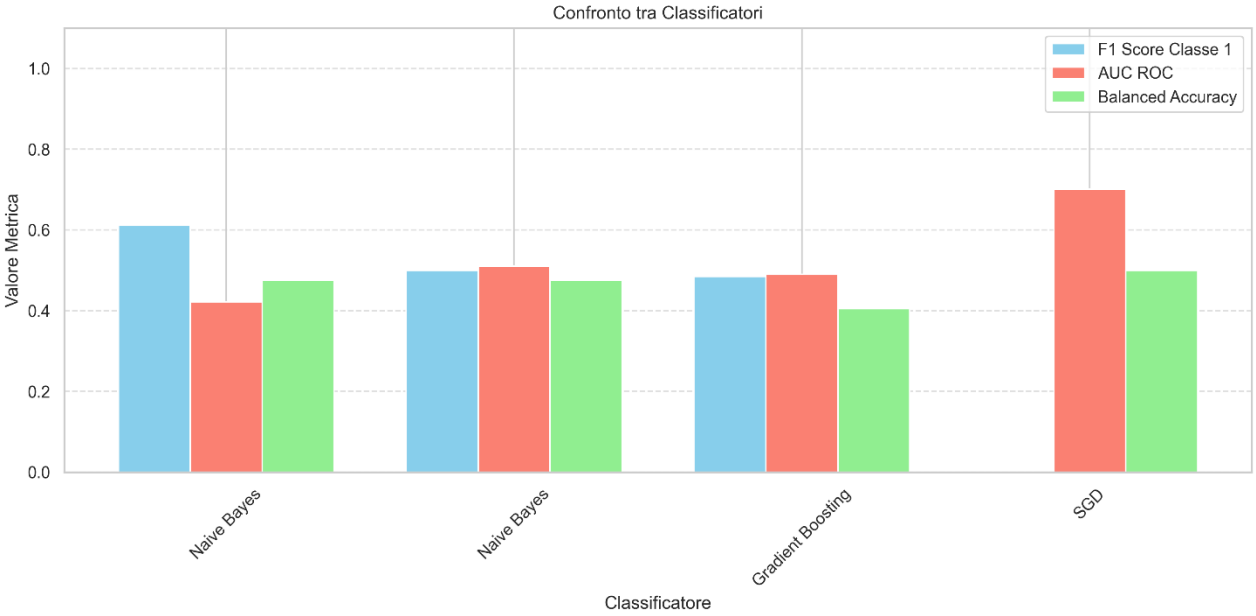


Figura 45 Principali metriche test set media, dataset no background

Di seguito riportiamo i grafici della ROC (tab.26) curve e curva Precision-Recall(tab.27):

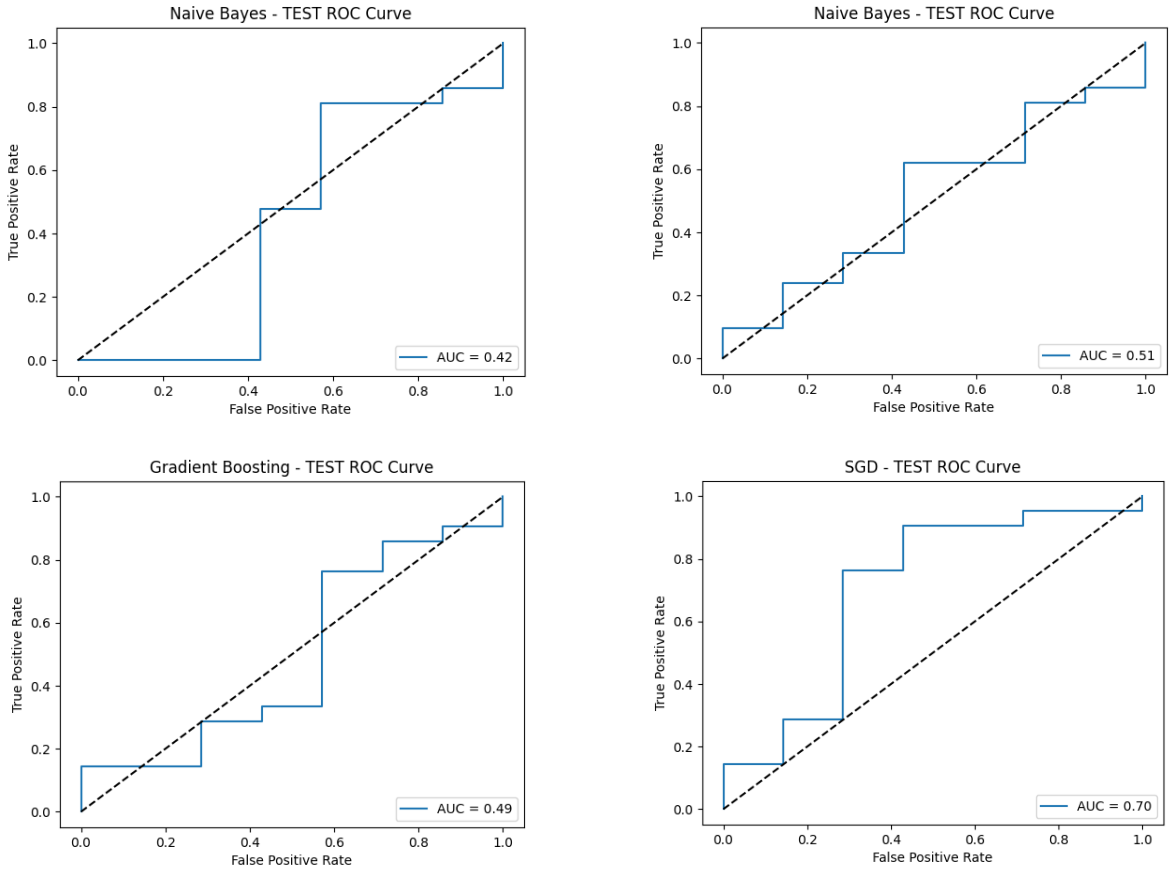


Tabella 26 Curve ROC test set media dataset no background

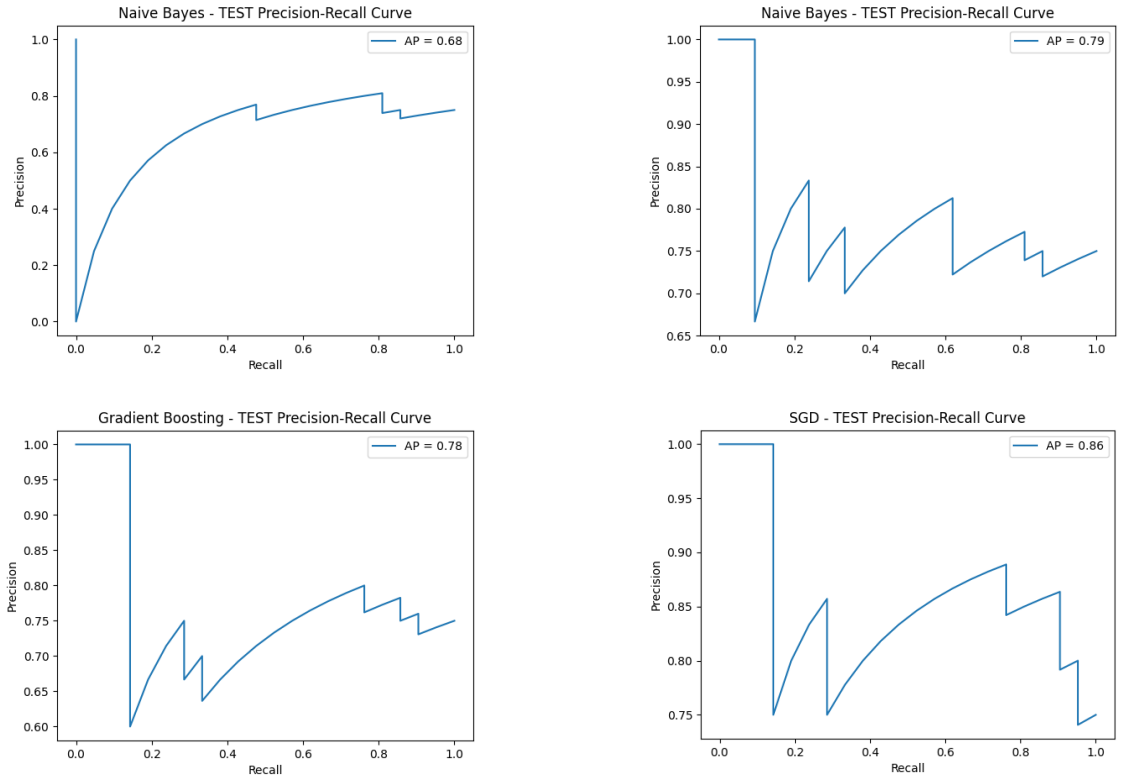


Tabella 27Curve Precision-Recall test set media dataset no backgorund

Dataset privo di background: risultati metodo di aggregazione calcolo statistiche descrittive dell'istogramma per ogni feature.

Affinity Propagation

L'intero elenco delle feature selezionate da questo metodo è riportato nell'indice. Di seguito vengono riportate le prime dieci feature:

- mean_original_firstorder_Entropy
- mean_original_glcm_SumAverage
- mean_original_glcm_SumSquares
- skew_original_firstorder_Entropy
- skew_original_firstorder_Kurtosis
- skew_original_firstorder_RobustMeanAbsoluteDeviation
- skew_original_glcm_Contrast
- skew_original_glcm_Correlation
- skew_original_glcm_Id
- skew_original_glrlm_GrayLevelNonUniformityNormalized

Nella seguente tabella (tab.28) vengono riportati i parametri dei classificatori in seguito al tuning:

Classificatori	Parametri Tuning
Adapting Boosting	learning_rate: 0.01, n_estimators: 100
Bagging DT	bootstrap: True, max_samples: 0.5, n_estimators: 50
Decision Tree	criterion: entropy, max_depth: 4, min_samples_split: 2
Gradient Boosting	learning_rate: 0.05, max_depth: 7, n_estimators: 50
KNN	n_neighbors: 5, p: 1, weights: uniform
Naive Bayes	Default
Random Forest	max_depth: None, max_features: sqrt, min_samples_split: 2, n_estimators: 50
SGD	alpha: 1e-05, loss: log_loss, penalty: l2
SVM Linear	C: 1, gamma: scale
SVM Polinomial	C: 10, degree: 4, gamma: auto
SVM Kernel RBF	C: 10, gamma: 0.01
SVM Sigmoid	C: 0.1, gamma: auto

Tabella 28 Parametri Classificatori Affinity statistiche

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC e accuracy.

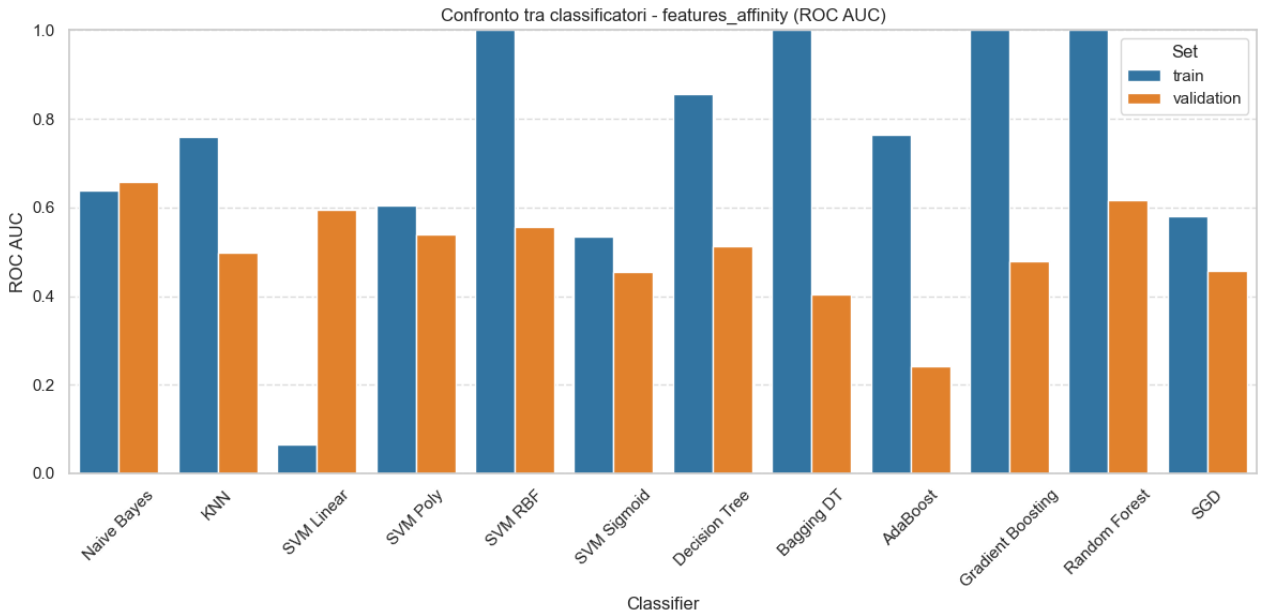


Figura 46AUC-ROC Affinity statistiche no backgorund

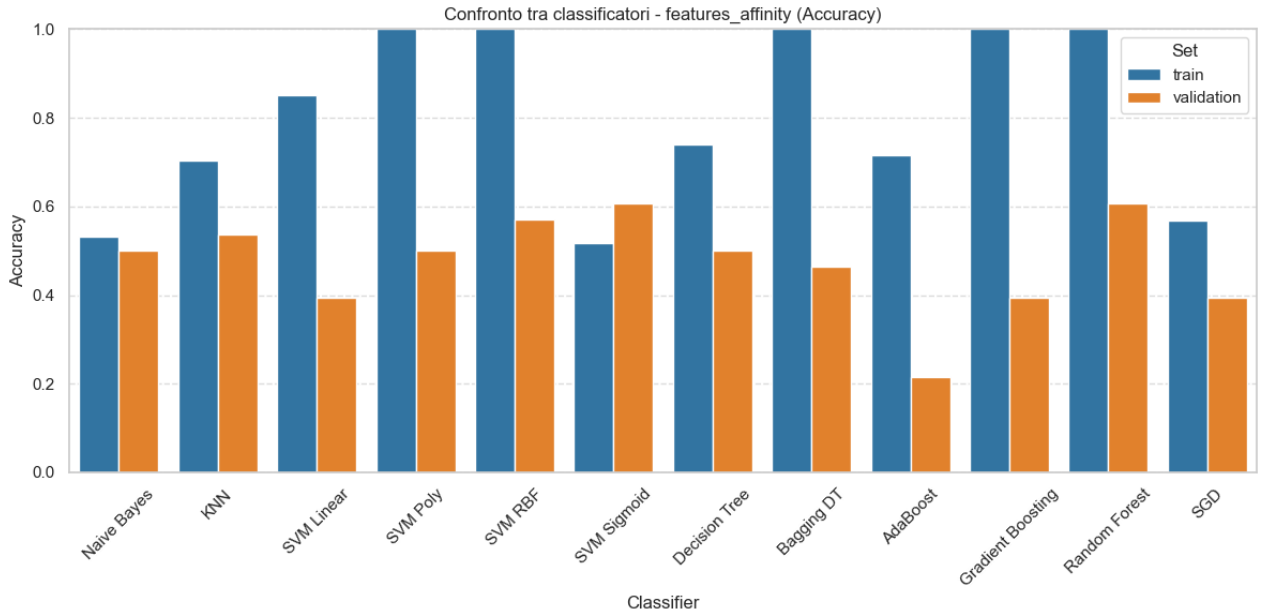


Figura 47Accuracy Affinity statistiche no background

Il classificatore, per questo metodo di feature selection, scelto in base al valore di AUC ROC calcolato sulle performance del validation set è Naive Bayes.

Correlazione assoluta tra feature e target

Nel seguente grafico (fig.48) vengono mostrati i valori di correlazione delle feature:

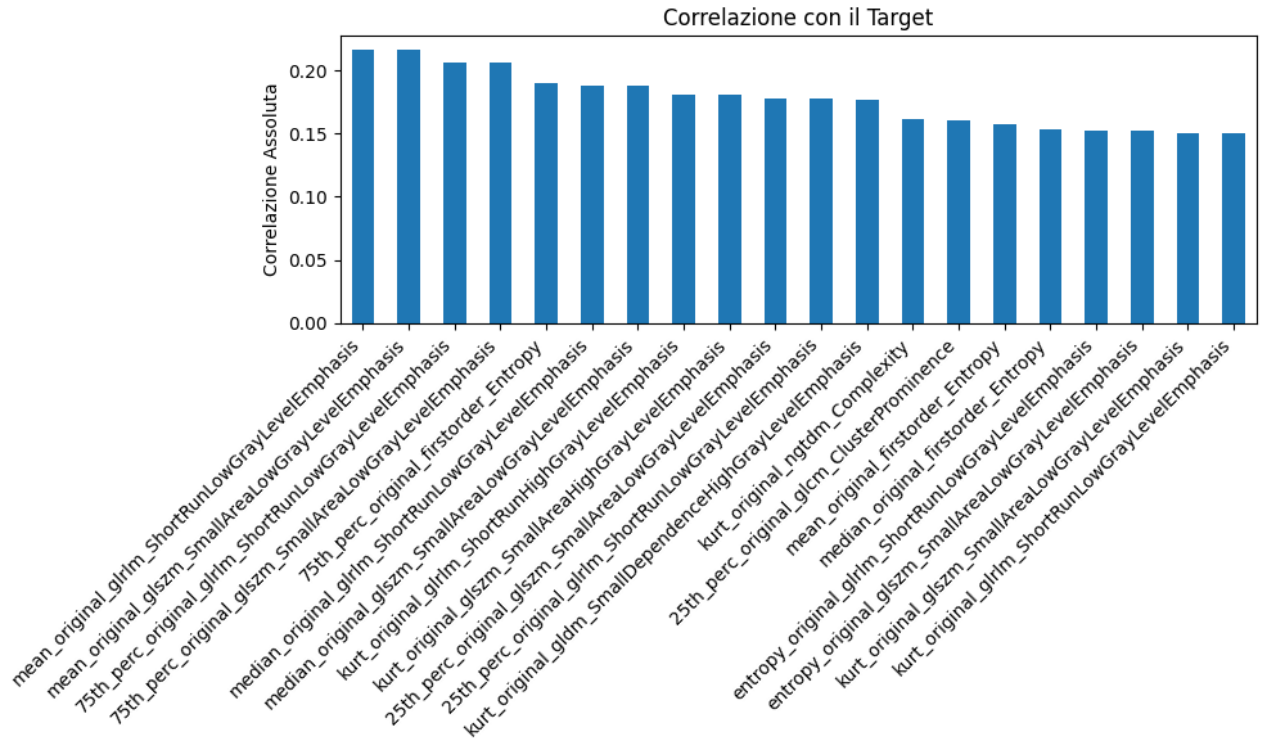


Figura 48Valori Correlazione statistiche no backgorund

L’elenco delle feature selezionate è riportato nell’indice qui riportiamo le prime dieci feature:

- mean_original_firstorder_Entropy
- mean_original_glcm_JointEntropy
- mean_original_glcm_SumEntropy
- mean_original_glrlm_GrayLevelNonUniformity
- mean_original_glrlm_GrayLevelVariance
- mean_original_glrlm_LowGrayLevelRunEmphasis
- mean_original_glrlm_ShortRunLowGrayLevelEmphasis
- mean_original_glszm_GrayLevelNonUniformity
- mean_original_glszm_GrayLevelVariance
- mean_original_glszm_LowGrayLevelZoneEmphasis

Nella seguente tabella riportiamo i valori dei parametri dei classificatori in seguito al tuning:

Classificatori	Parametri Tuning
Adapting Boosting	learning_rate: 0.01, n_estimators: 200
Bagging DT	bootstrap: True, max_samples: 0.5, n_estimators: 50
Decision Tree	criterion: gini, max_depth: 4, min_samples_split: 2
Gradient Boosting	learning_rate: 0.01, max_depth: 7, n_estimators: 150
KNN	n_neighbors: 3, p: 1, weights: distance
Naive Bayes	Default
Random Forest	max_depth: None, max_features: sqrt, min_samples_split: 2, n_estimators: 50
SGD	alpha: 1e-05, loss: log_loss, penalty: l2
SVM Linear	C: 100, gamma: scale
SVM Polinomial	C: 1, degree: 4, gamma: scale
SVM Kernel RBF	C: 0.1, gamma: 0.1
SVM Sigmoid	C: 1, gamma: auto

Tabella 29Parametri Classificatori Affinity Statistiche no background

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC (fig.49) e accuracy (fig.50).

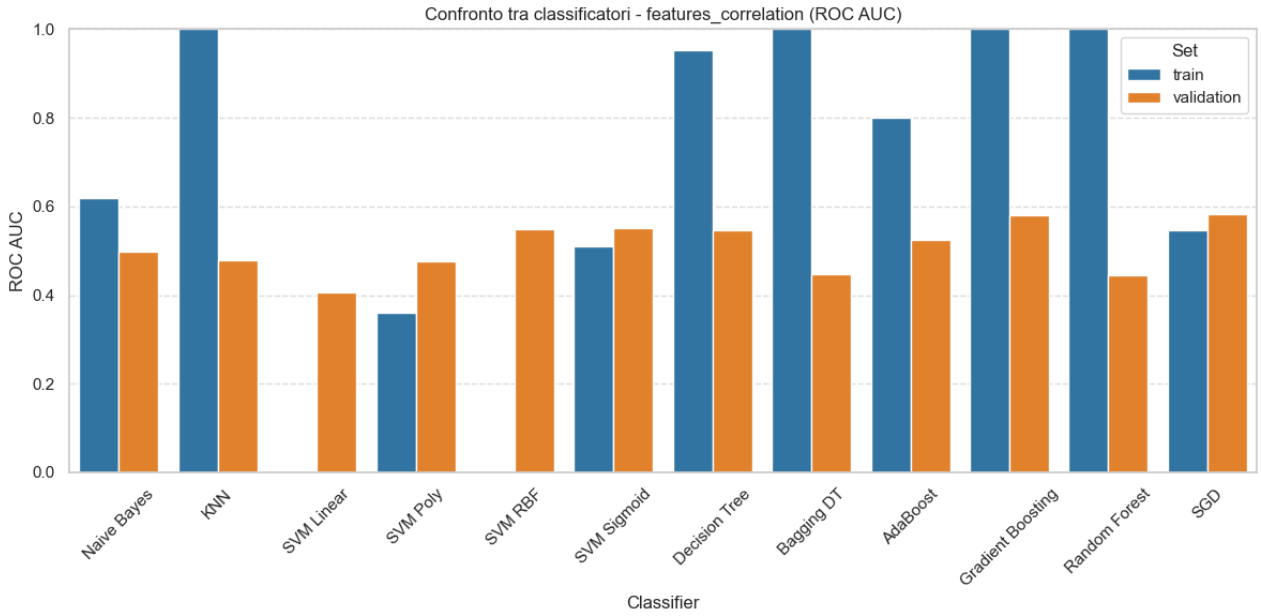


Figura 49AUC-ROC Corr.Ass. statistiche no backgorund

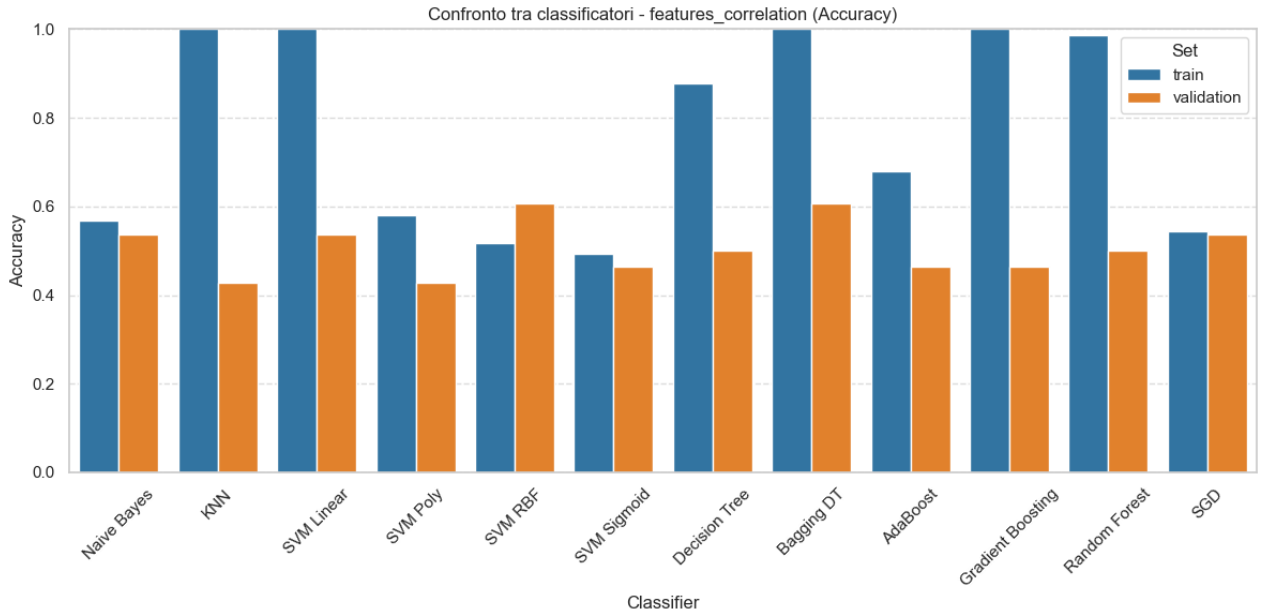


Figura 50 Accuracy Corr.Ass. statistiche no background

Il classificatore, per questo metodo di feature selection, scelto in base allo “score” calcolato sulle performance del validation set è il Naive bayes.

Minimun Redundancy – Maximum Relevance (MRMR)

Nel seguente grafico (fig.51) vengono mostrati i valori delle rilevanze delle feature:

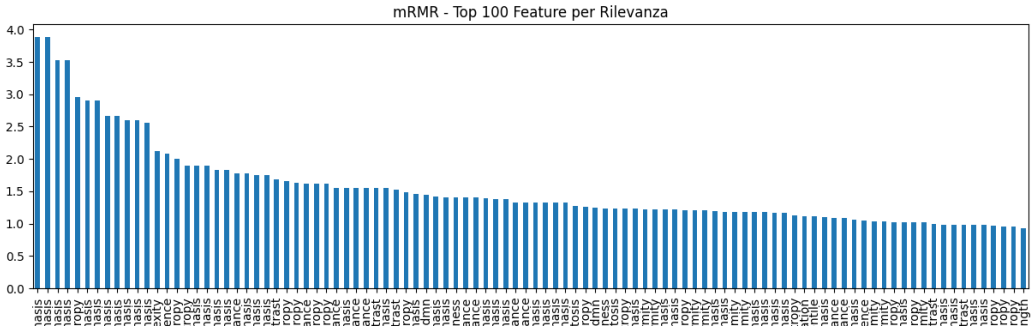


Figura 51 Valori Rilevanze MRMR statistiche no background

L'elenco completo delle feature è riportato nell'appendice mentre di seguito vengono riportate le prime dieci feature:

- mean_original_firstorder_Entropy
- mean_original_glcm_JointEntropy
- mean_original_glcm_SumEntropy
- mean_original_glrlm_GrayLevelNonUniformity
- mean_original_glrlm_GrayLevelVariance
- mean_original_glrlm_LowGrayLevelRunEmphasis
- mean_original_glrlm_ShortRunLowGrayLevelEmphasis
- mean_original_glszm_GrayLevelNonUniformity
- mean_original_glszm_GrayLevelVariance
- mean_original_glszm_LowGrayLevelZoneEmphasis

Nella seguente tabella vengono riportati i parametri dei classificatori in seguito al tuning:

Classificatori	Parametri Tuning
Adapting Boosting	learning_rate: 1, n_estimators: 50
Bagging DT	bootstrap: True, max_samples: 0.5, n_estimators: 50
Decision Tree	criterion: entropy, max_depth: None, min_samples_split: 5
Gradient Boosting	learning_rate: 0.01, max_depth: 3, n_estimators: 150
KNN	n_neighbors: 5, p: 2, weights: uniform
Naive Bayes	Default
Random Forest	max_depth: None, max_features: log2, min_samples_split: 5, n_estimators: 100

SGD	alpha: 0.0001, loss: log_loss, penalty: l2
SVM Linear	C: 100, gamma: scale
SVM Polinomial	C: 10, degree: 2, gamma: auto
SVM Kernel RBF	C: 10, gamma: 0.1
SVM Sigmoid	C: 1, gamma: auto

Tabella 30Parametri Classificatori MRMR statistiche no background

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC (fig.52) e accuracy (fig.53).

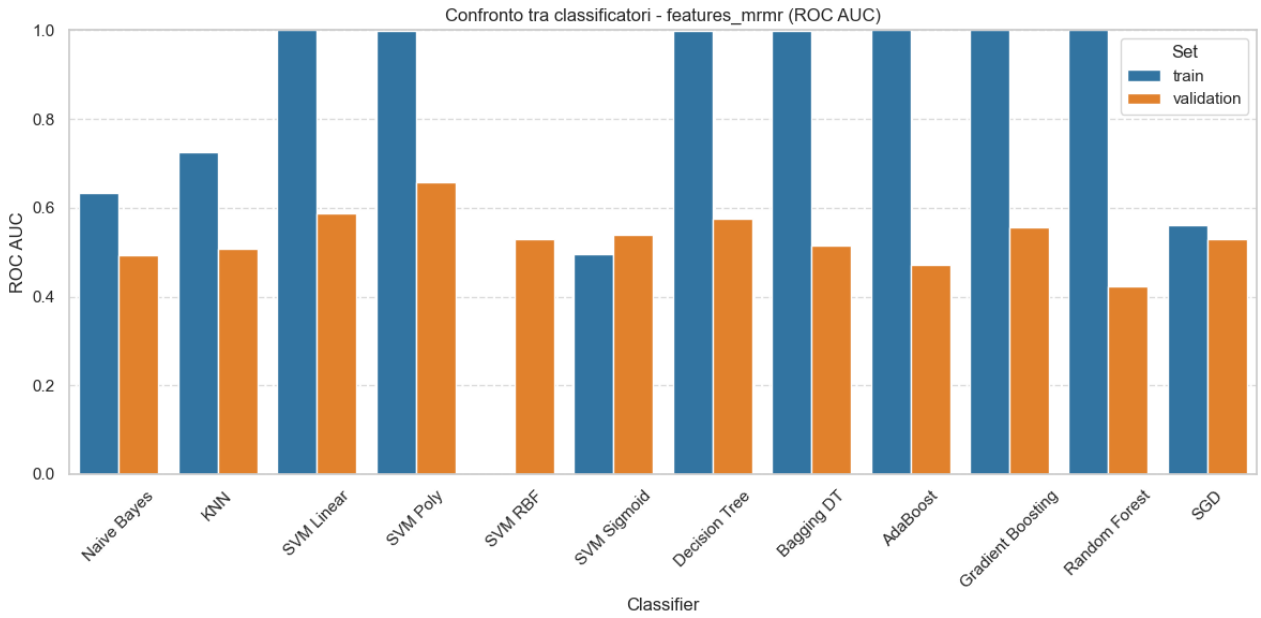


Figura 52AUC-ROC MRMR statistiche no background

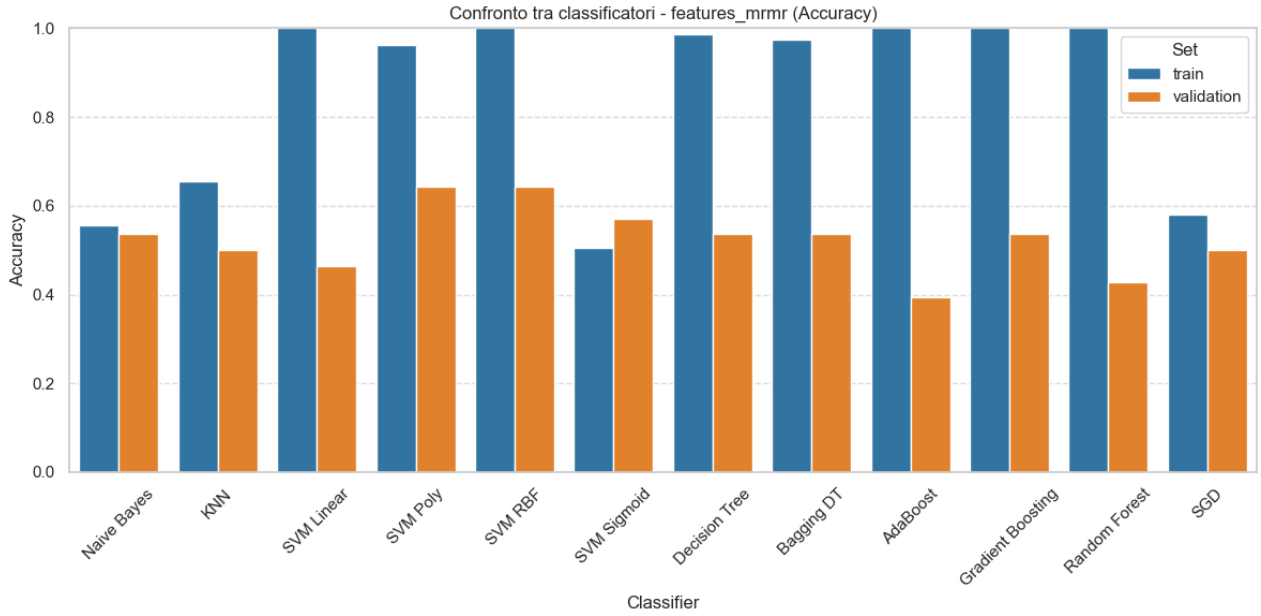


Figura 53Accuracy MRMR statistiche no background

Il classificatore, per questo metodo di feature selection, scelto in base al valore di AUC ROC calcolato sulle performance del validation set è SVM Polinomial.

Recursive Feature Elimination (RFE)

Nel seguente grafico (fig.54-55)vengono riportate le frequenze di selezione delle feature:

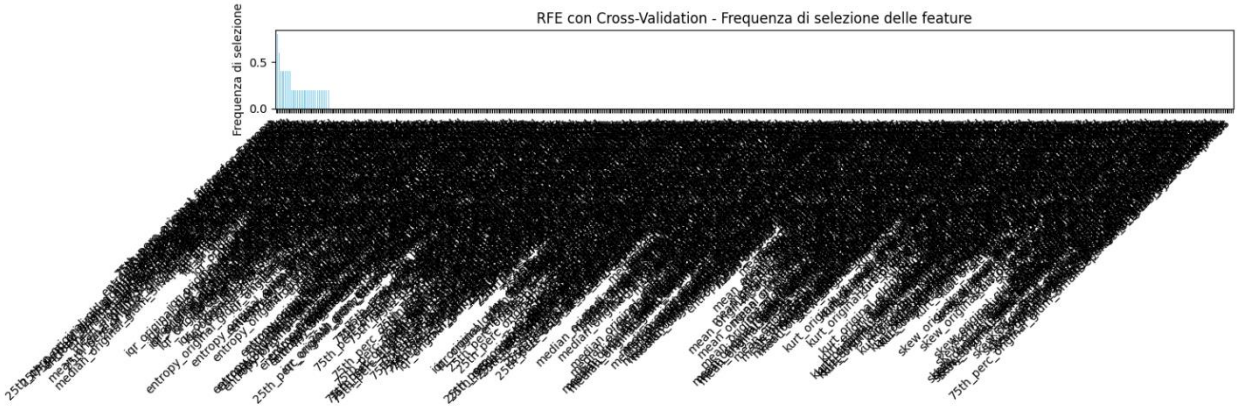


Figura 54 Frequenze di selezione RFE statostiche no backgorund

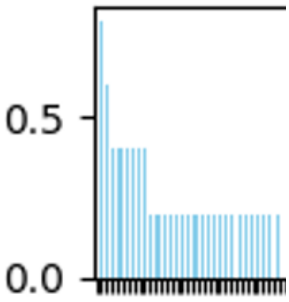


Figura 55 Zoom grafico frequenze

In questo caso il KneeDetector ha stabilito una soglia di 0.6 che ha selezionato solamente 2 feature. Questo comportamento troppo restrittivo ha attivato il fallback che ha stabilito una nuova soglia a 0.2. In questo caso la soglia impostata risulta troppo permissiva e per questo si è optato per una soglia manuale impostata a 0.4. In questo modo le feature selezionate dal metodo sono:

- 75th_perc_original_firstorder_Entropy
- 75th_perc_original_glcm_SumEntropy
- 25th_perc_original_ngtdm_Contrast
- entropy_original_glrlm_ShortRunEmphasis
- 75th_perc_original_glcm_Correlation
- kurt_original_glcm_Imc1
- 25th_perc_original_glcm_InverseVariance
- median_original_ngtdm_Contrast
- 25th_perc_original_gldm_SmallDependenceLowGrayLevelEmphasis
- 25th_perc_original_gldm_DependenceNonUniformityNormalized

Nella seguente tabella vengono riportati i valori dei parametri dei classificatori in seguito al tuning:

Classificatori	Parametri Tuning
Adapting Boosting	learning_rate: 1, n_estimators: 200
Bagging DT	bootstrap: True, max_samples: 0.7, n_estimators: 50
Decision Tree	criterion: entropy, max_depth: None, min_samples_split: 5
Gradient Boosting	learning_rate: 0.1, max_depth: 3, n_estimators: 150
KNN	n_neighbors: 9, p: 2, weights: uniform
Naive Bayes	Default
Random Forest	max_depth: None, max_features: sqrt, min_samples_split: 5, n_estimators: 50
SGD	alpha: 0.0001, loss: hinge, penalty: elasticnet
SVM Linear	C: 10, gamma: scale
SVM Polinomial	C: 0.1, degree: 4, gamma: scale

SVM Kernel RBF	C: 100, gamma: scale
SVM Sigmoid	C: 1, gamma: scale

Tabella 31Parametri CLassificatori RFE statistiche no background

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC (fig.56) e accuracy (fig.57).

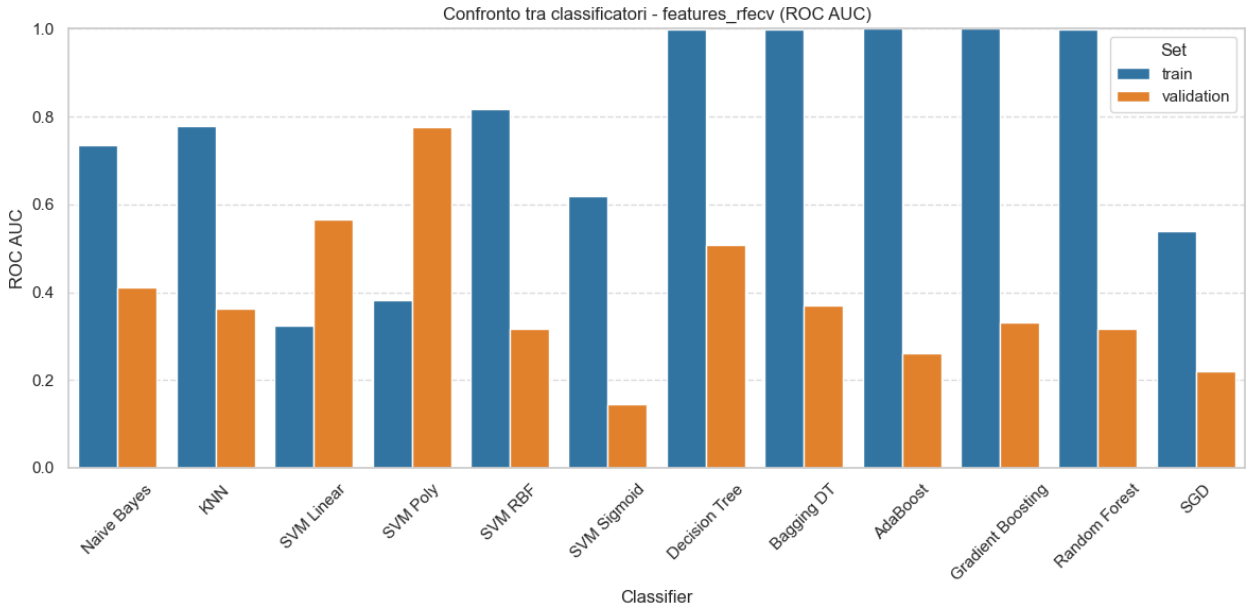


Figura 56AUC-ROC RFE statistiche no background

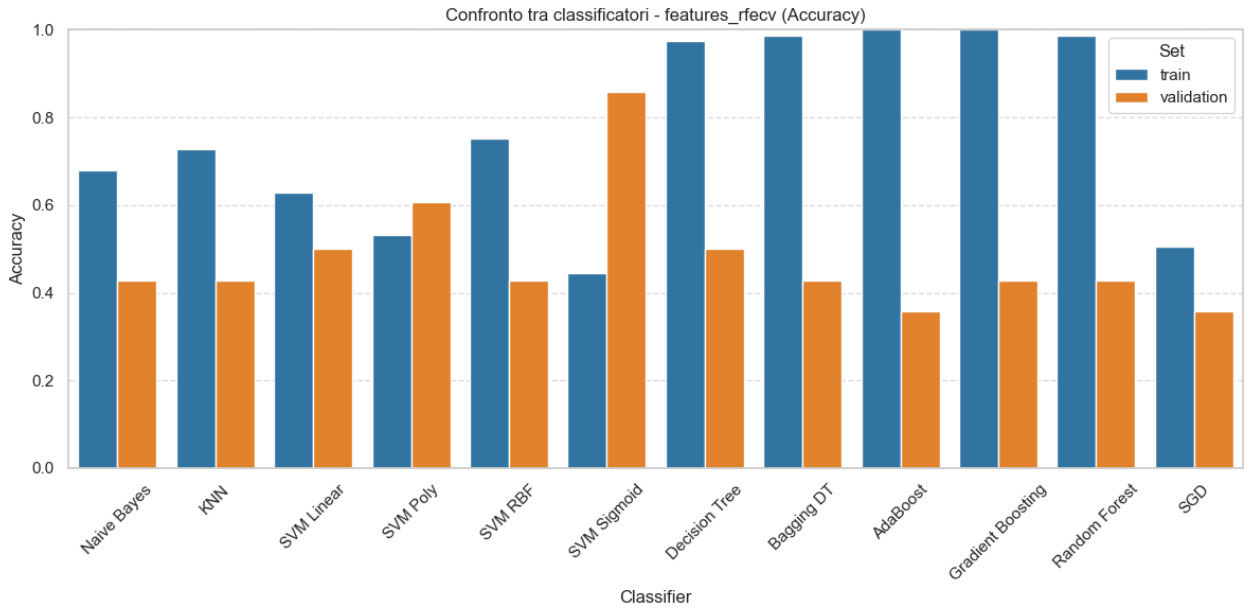


Figura 57Accuracy RFE statistiche no background

Il classificatore, per questo metodo di feature selection, scelto in base al valore di AUC ROC calcolato sulle performance del validation set è SVM Polinomial.

Risultati test

Nella seguente tabella vengono riportati le principali metriche dei classificatori calcolate sul test set.

clf_nome	Naive Bayes	SGD	SVM Poly	SVM Poly
roc_auc	0,727891156	0,482993197	0,319727891	0,619047619
accuracy	0,75	0,642857143	0,357142857	0,25
bal_accuracy	0,595238095	0,571428571	0,333333333	0,5
F1_score_1	0,844444444	0,75	0,470588235	0
F1_score_0	0,363636364	0,375	0,181818182	0,4
precision_0	0,5	0,333333333	0,133333333	0,25
precision_1	0,791666667	0,789473684	0,615384615	0
recall_0	0,285714286	0,428571429	0,285714286	1
recall_1	0,904761905	0,714285714	0,380952381	0

Tabella 32RIsultati metriche test set statistiche no background

Nel seguente grafico (fig.58) vengono riportati i valori di riferimento di ciascuno dei classificatori in base ai valori risultanti dal test set:

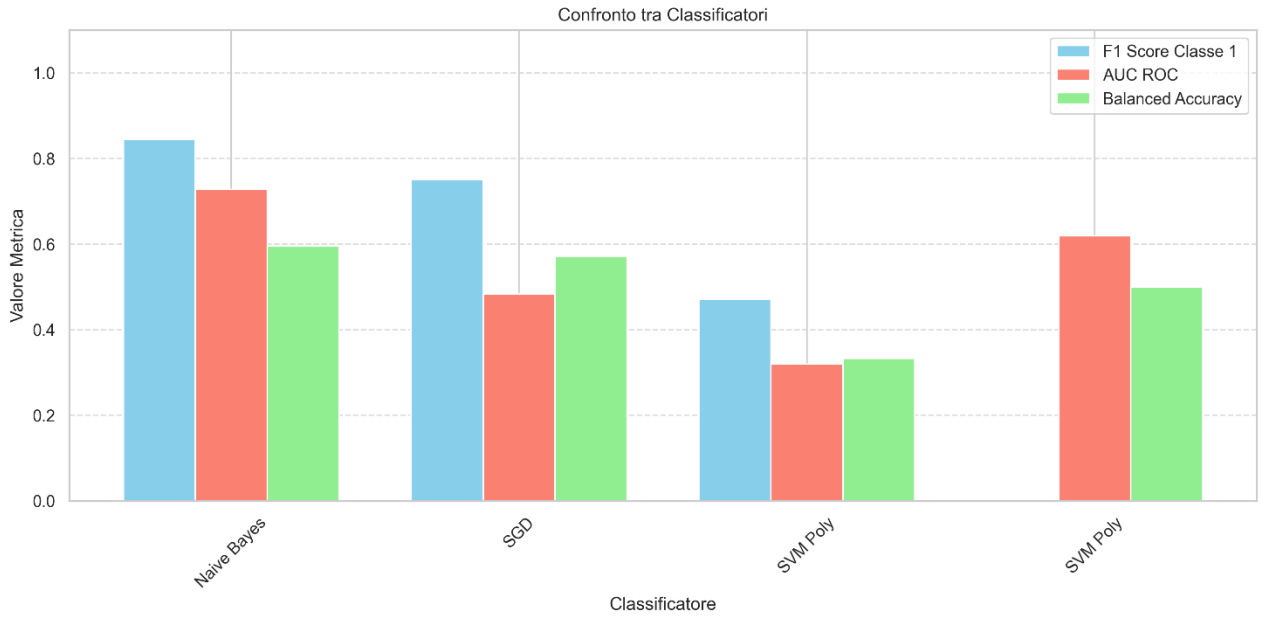
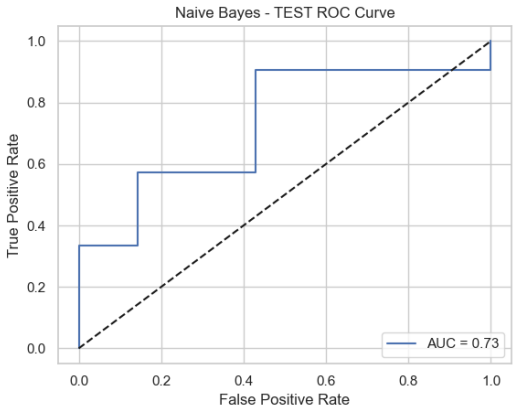
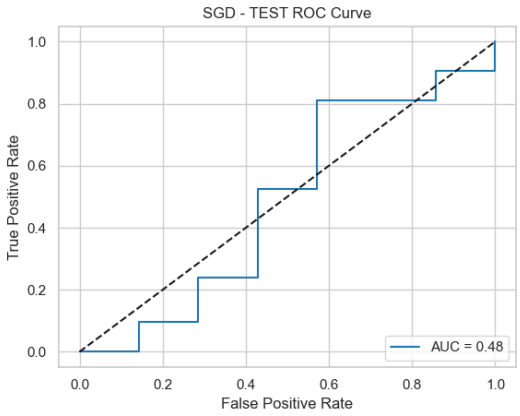
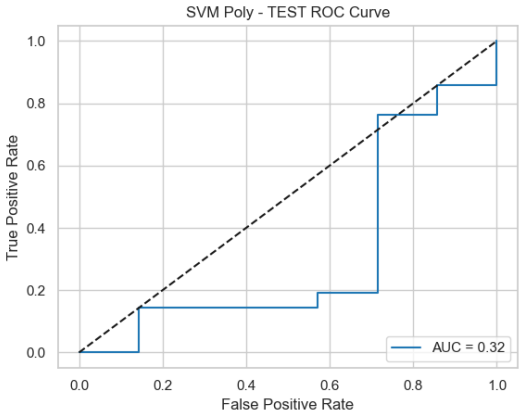


Figura 58RIsultati metriche principali test set statistiche no background

Di seguito vengono mostrati grafici delle curve ROC (tab.34) e delle curve Precision-Recall (tab.) calcolate sul test set dei migliori classificatori:







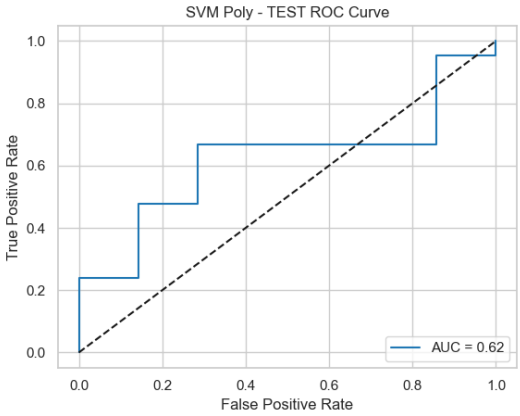
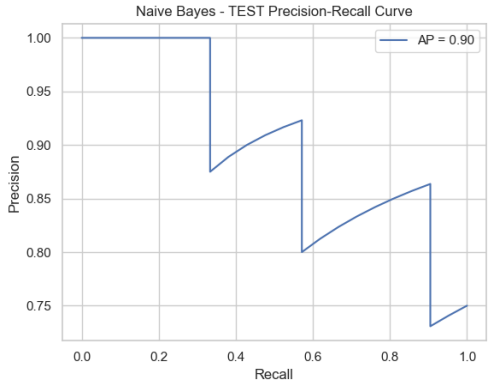
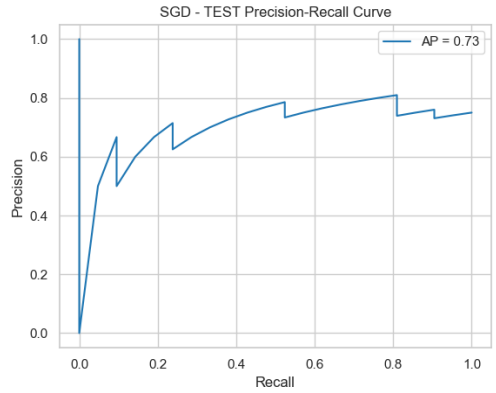
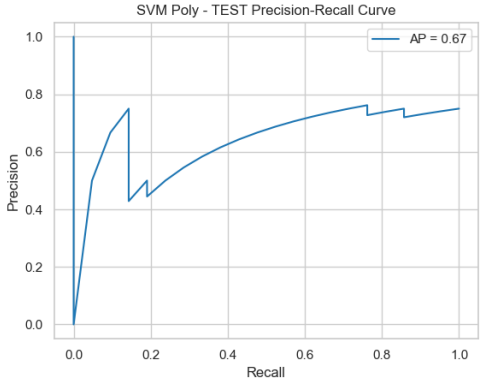


Tabella 33Curve ROC test set statistiche no background







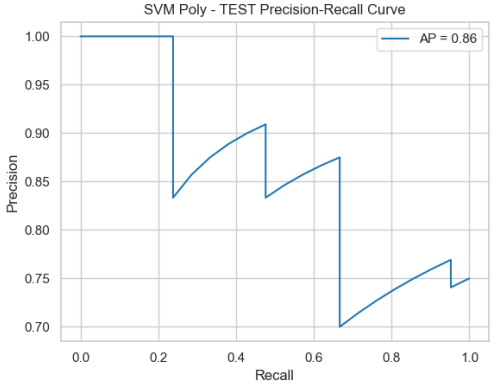


Tabella 34 Curve Precision-Recall test set statistiche no background

Risultati PCA: dataset privo di background

Metodo di aggregazione: calcolo della media di ogni feature

Il seguente grafico (fig.59) mostra la varianza delle feature con la soglia al 90%:

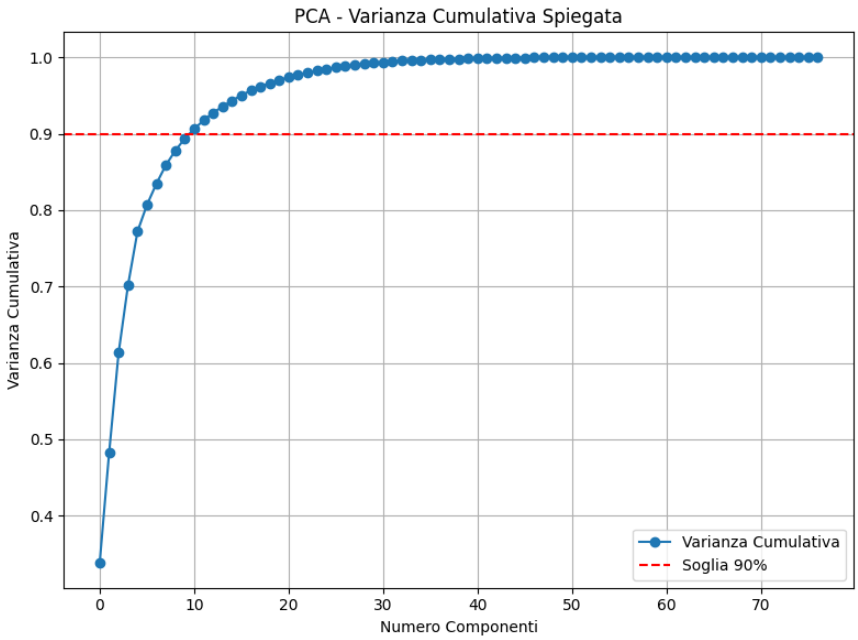


Figura 59VArianza PCA media no background

Nella seguente tabella vengono mostrati i parametri dei classificatori in seguito al tuning:

Classificatori	Parametri Tuning
Naive Bayes	Default
SGD	alpha: 0.001, loss: log_loss, penalty: l1
SVM Linear	C: 1, gamma: scale

Tabella 35Parametri Classificatori PCA media no backgorund

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC (fig.60) e accuracy (fig.61).

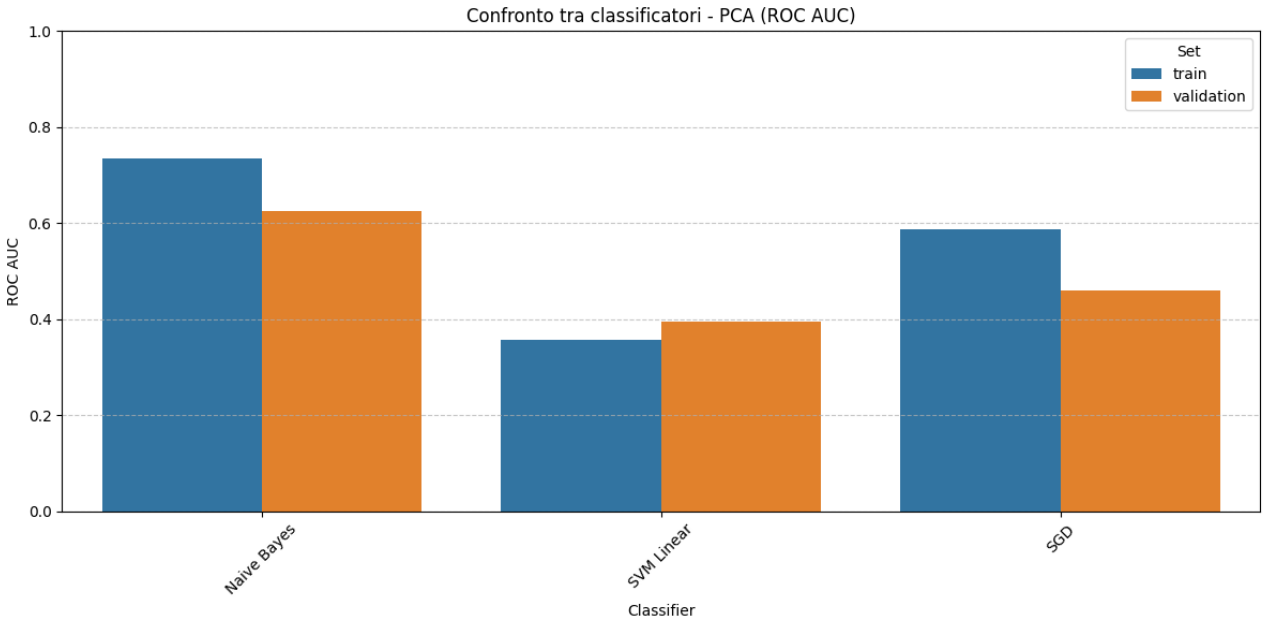


Figura 60AUC-ROC PCA media no background

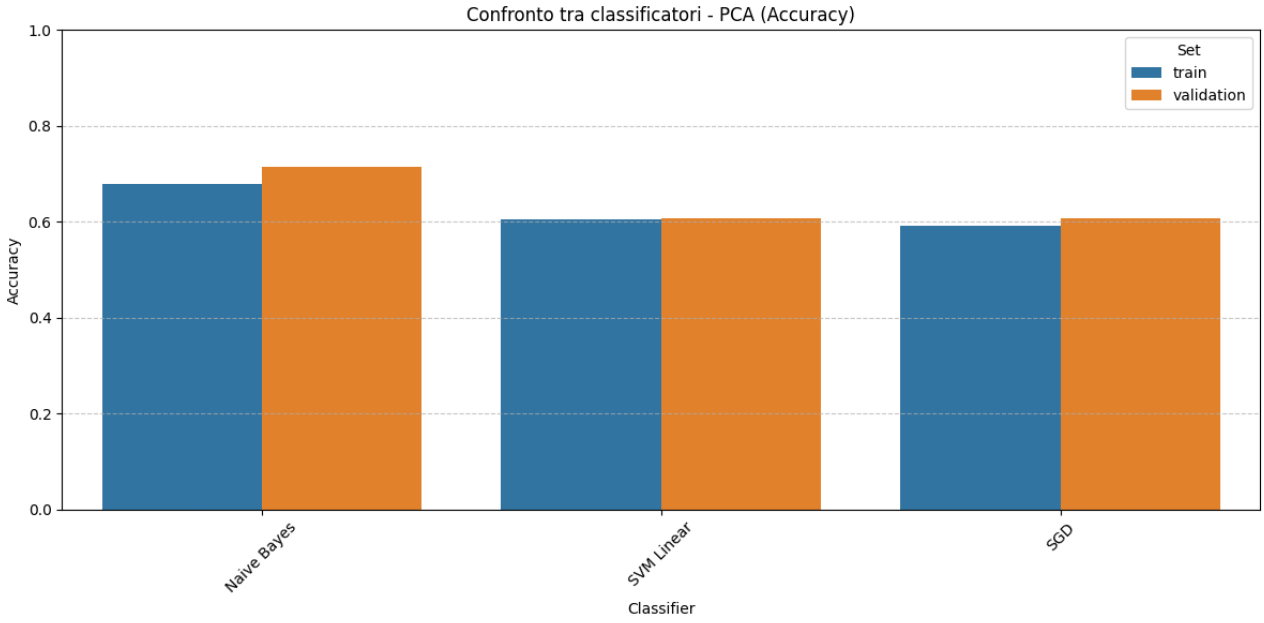


Figura 61Accuracy PCA media no background

Il classificatore, per questo metodo di feature selection, scelto in base all'AUC ROC calcolato sulle performance del validation set è SVM Linear. I risultati delle metriche calcolate sul test set sono riportate nella seguente tabella:

roc_auc	0,496598639
accuracy	0,535714286
bal_accuracy	0,547619048
F1_score_1	0,628571429
F1_score_0	0,380952381
precision_0	0,285714286
precision_1	0,785714286

recall_0	0,571428571
recall_1	0,523809524

Tabella 36 Risultati metriche PCA media no background

Di seguito vengono mostrati i grafici della curva ROC e della curva Precision-Recall (tab.37):

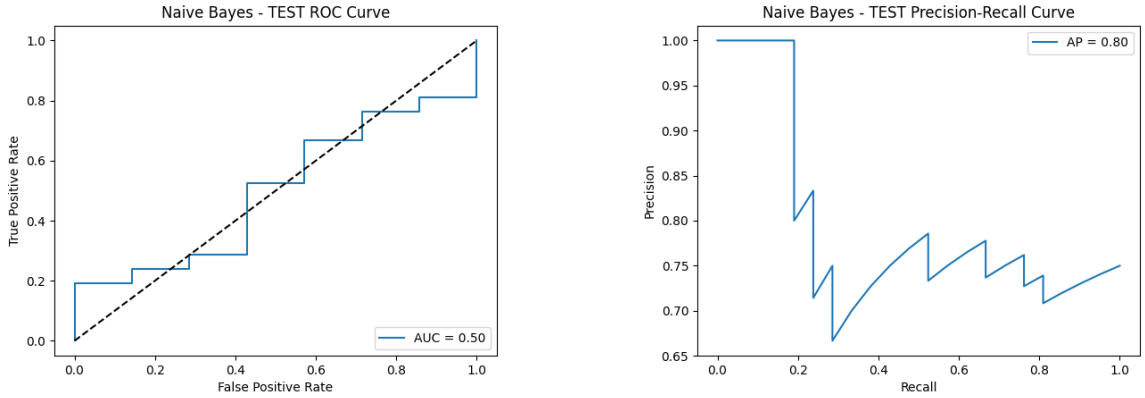


Tabella 37 Curve ROC e Precision-Recall PCA media no background

Metodo di aggregazione: calcolo delle statistiche descrittive dell'istogramma per ogni feature.

Il seguente grafico (fig.62) mostra la varianza delle feature con la soglia al 90%:

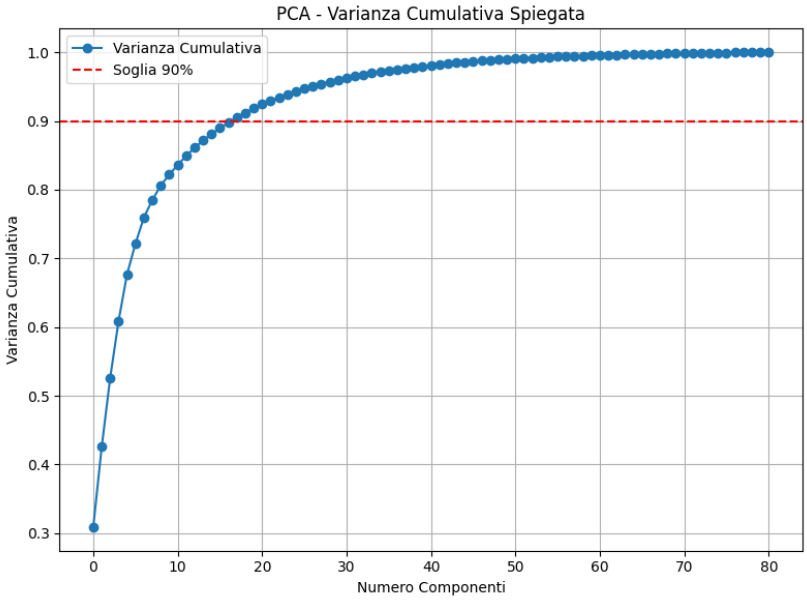


Figura 62Varianza PCA statistiche no Background

Nella seguente tabella (tab.38) vengono mostrati i parametri dei classificatori in seguito al tuning:

Classificatori	Parametri Tuning
Naive Bayes	Default
SGD	alpha: 0.001, loss: log_loss, penalty: elasticnet

SVM Linear	C: 100, gamma: scale

Tabella 38Parametri Classificatori PCA statistiche no background

Di seguito vediamo le performance dei classificatori sul training e validation set delle metriche AUC ROC (fig.63) e accuracy (fig.64).

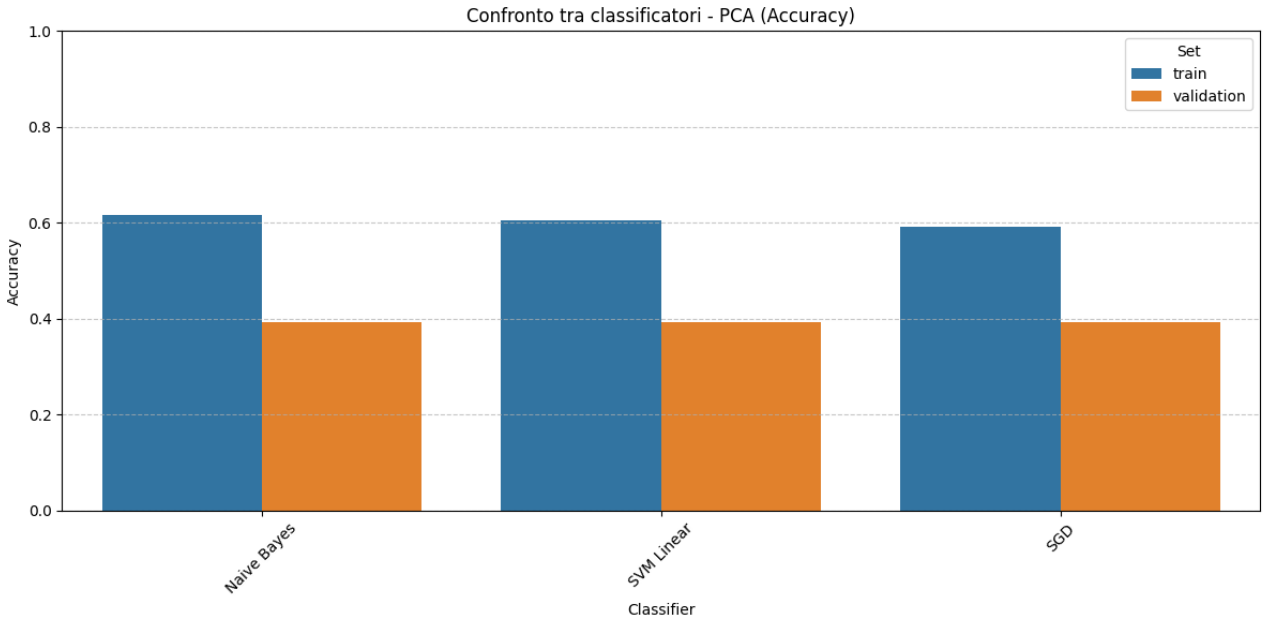


Figura 63 AUC-ROC PCA sttistiche no background

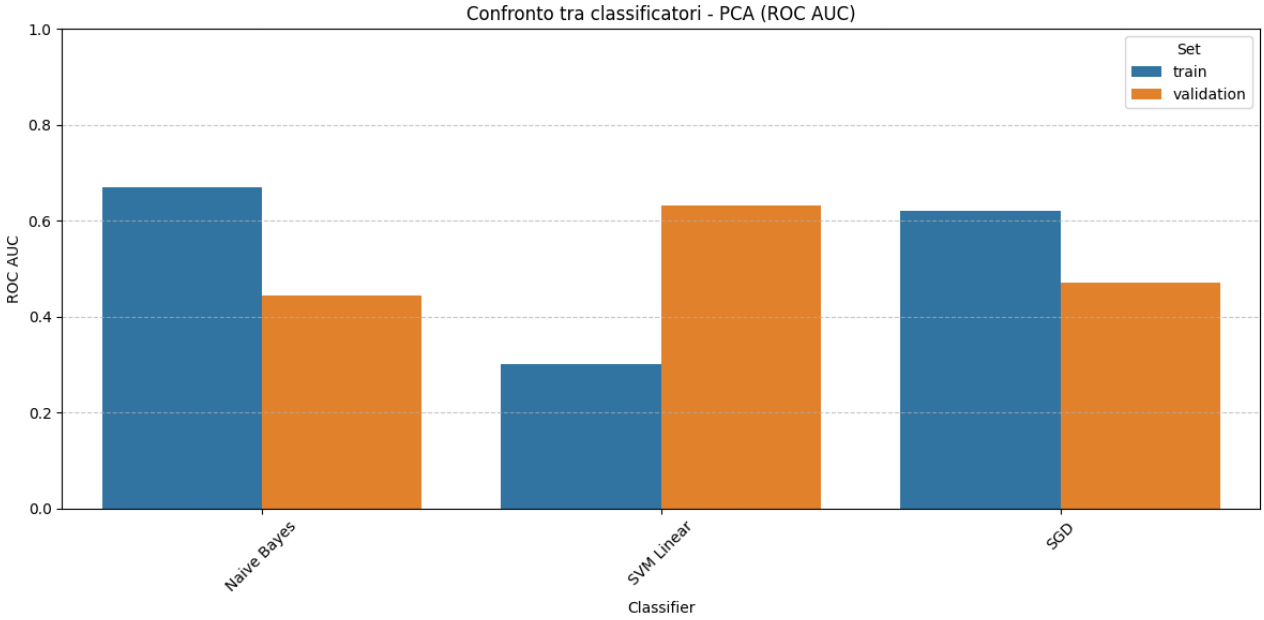


Figura 64Accuracy PCA statistiche no background

Il classificatore, per questo metodo di feature selection, scelto in base all'AUC ROC calcolato sulle performance del validation set è SVM Linear. I risultati delle metriche calcolate sul test set sono riportati nella seguente tabella (tab.39):

roc_auc	0,544217687
accuracy	0,464285714
bal_accuracy	0,404761905
F1_score_1	0,594594595
F1_score_0	0,210526316
precision_0	0,166666667
precision_1	0,6875
recall_0	0,285714286
recall_1	0,523809524

Tabella 39Risultati metriche PCA test set statistiche no background

Di seguito (tab.40) vengono mostrati i grafici della curva ROC e della curva Precision-Recall risultanti dal test set:

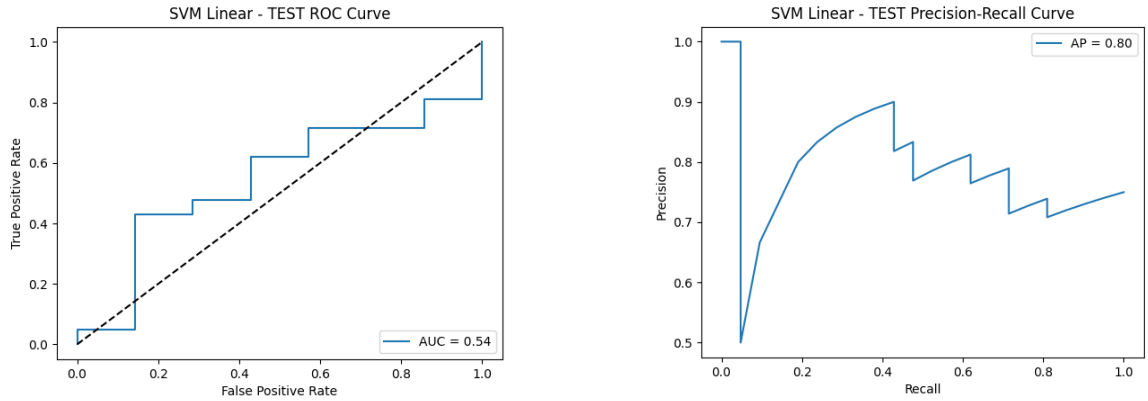


Tabella 40 Curve ROC e Precision-Recall PCA statistiche test set no background

Risultati cross-validation esterna

Dataset completo: metodo aggregazione media

Nella seguente tabella (tab.41) vengono mostrate le metriche dei migliori classificatori calcolate con la cross validation esterna:

Feature_Selection	features_affinity	features_correlation	features_mrmr	features_rfecv
Classifier	Bagging DT	Bagging DT	SGD	SVM Sigmoid
accuracy	0.495 ± 0.104	0.517 ± 0.068	0.511 ± 0.033	0.518 ± 0.011
precision	0.523 ± 0.133	0.533 ± 0.053	0.315 ± 0.258	0.518 ± 0.011
recall	0.435 ± 0.106	0.564 ± 0.136	0.557 ± 0.462	1.000 ± 0.000
f1	0.471 ± 0.107	0.541 ± 0.082	0.402 ± 0.330	0.683 ± 0.010
balanced_accuracy	0.499 ± 0.106	0.517 ± 0.070	0.502 ± 0.022	0.500 ± 0.000
roc_auc	0.512 ± 0.101	0.529 ± 0.148	nan ± nan	0.438 ± 0.073
tp	6.200 ± 1.600	8.000 ± 1.897	8.000 ± 6.663	14.200 ± 0.400
tn	7.400 ± 1.855	6.200 ± 1.720	6.000 ± 6.164	0.000 ± 0.000
fp	5.800 ± 2.040	7.000 ± 1.789	7.200 ± 5.913	13.200 ± 0.400
fn	8.000 ± 1.414	6.200 ± 1.939	6.200 ± 6.462	0.000 ± 0.000

specificity	0.563 ± 0.148	0.470 ± 0.134	0.446 ± 0.455	0.000 ± 0.000

Tabella 41CV esterna dataset completo media

Dataset completo: metodo aggregazione statistiche descrittive dell'istogramma

Nella seguente tabella (tab.42) vengono mostrate le metriche dei migliori classificatori calcolate con la cross validation esterna:

feature_selection	features_affinity	features_correlation	features_mrmr	features_rfecv
classifier	SVM Poly	Random Forest	SGD	SVM Sigmoid
accuracy	0.518 ± 0.073	0.562 ± 0.072	0.488 ± 0.041	0.518 ± 0.011
precision	0.529 ± 0.054	0.581 ± 0.074	0.478 ± 0.080	0.518 ± 0.011
recall	0.524 ± 0.153	0.590 ± 0.108	0.435 ± 0.346	1.000 ± 0.000
f1	0.521 ± 0.101	0.580 ± 0.075	0.392 ± 0.230	0.683 ± 0.010
balanced_accuracy	0.520 ± 0.071	0.562 ± 0.072	0.494 ± 0.039	0.500 ± 0.000
roc_auc	0.480 ± 0.071	0.564 ± 0.090	nan ± nan	0.517 ± 0.035
tp	7.400 ± 2.059	8.400 ± 1.625	6.200 ± 4.874	14.200 ± 0.400
tn	6.800 ± 0.748	7.000 ± 1.673	7.200 ± 4.020	0.000 ± 0.000
fp	6.400 ± 0.800	6.200 ± 1.939	6.000 ± 4.336	13.200 ± 0.400
fn	6.800 ± 2.315	5.800 ± 1.470	8.000 ± 4.817	0.000 ± 0.000
specificity	0.515 ± 0.058	0.533 ± 0.136	0.553 ± 0.311	0.000 ± 0.000

Tabella 42CV esterna dataset completo statistiche

Dataset privo di background: metodo di aggregazione media

Nella seguente tabella (tab.43) vengono mostrate le metriche dei migliori classificatori calcolate con la cross validation esterna:

Feature_Selection	features_affinity	features_correlation	features_mrmr	features_rfecv
Classifier	Naive Bayes	Naive Bayes	Gradient Boosting	SGD
accuracy	0.497 ± 0.067	0.553 ± 0.105	0.561 ± 0.060	0.497 ± 0.083
precision	0.516 ± 0.075	0.560 ± 0.089	0.575 ± 0.070	0.406 ± 0.210
recall	0.550 ± 0.123	0.632 ± 0.108	0.619 ± 0.109	0.629 ± 0.354
f1	0.526 ± 0.081	0.594 ± 0.098	0.590 ± 0.071	0.490 ± 0.260
balanced_accuracy	0.494 ± 0.069	0.550 ± 0.105	0.561 ± 0.061	0.491 ± 0.075
roc_auc	0.479 ± 0.048	0.540 ± 0.094	0.536 ± 0.068	0.461 ± 0.103
tp	7.800 ± 1.720	9.000 ± 1.673	8.800 ± 1.600	9.000 ± 5.177
tn	5.800 ± 1.939	6.200 ± 1.470	6.600 ± 1.625	4.600 ± 4.454
fp	7.400 ± 1.855	7.000 ± 1.265	6.600 ± 1.855	8.600 ± 4.630
fn	6.400 ± 1.744	5.200 ± 1.470	5.400 ± 1.497	5.200 ± 4.956
specificity	0.438 ± 0.145	0.468 ± 0.103	0.502 ± 0.131	0.353 ± 0.344

Tabella 43CV esterna dataset no background media

Dataset privo di background: metodo aggregazione statistiche descrittive dell'istogramma

Nella seguente tabella (tab.44) vengono mostrate le metriche dei migliori classificatori calcolate con la cross validation esterna:

Feature_Selection	features_affinity	features_correlation	features_mrmr	features_rfecv
Classifier	Naive Bayes	SGD	SVM Poly	SVM Poly
accuracy	0.539 ± 0.108	0.510 ± 0.073	0.444 ± 0.094	0.526 ± 0.019
precision	0.533 ± 0.068	0.538 ± 0.096	0.468 ± 0.085	0.523 ± 0.014
recall	0.787 ± 0.164	0.543 ± 0.262	0.491 ± 0.110	0.986 ± 0.029
f1	0.634 ± 0.101	0.508 ± 0.131	0.476 ± 0.089	0.683 ± 0.014
balanced_accuracy	0.528 ± 0.109	0.505 ± 0.074	0.441 ± 0.094	0.508 ± 0.015
roc_auc	0.528 ± 0.080	nan ± nan	0.570 ± 0.109	0.530 ± 0.072
tp	11.200 ± 2.482	7.800 ± 4.020	7.000 ± 1.673	14.000 ± 0.632
tn	3.600 ± 1.356	6.200 ± 3.868	5.200 ± 2.040	0.400 ± 0.490
fp	9.600 ± 1.020	7.000 ± 3.742	8.000 ± 1.673	12.800 ± 0.748
fn	3.000 ± 2.280	6.400 ± 3.666	7.200 ± 1.470	0.200 ± 0.400
specificity	0.270 ± 0.093	0.467 ± 0.291	0.390 ± 0.139	0.031 ± 0.038

Tabella 44 CV esterna dataset no background statstiche

Conclusioni

Visione complessiva

Presa visione di tutti i risultati riportiamo in seguito una tabella riassuntiva (tab.45) delle migliori coppie metodo di feature selection – classificatore per avere una visione complessiva:

Dataset	Aggregazione	Metodo F.S.	Classificatore	Performance (AUC ROC)
Completo	Media	Affinity Propagation	Bagging DT	0,77
Completo	Statistiche	RFE+CV	SVM Sigmoid	0.58
No Background	Media	RFE+CV	SGD	0.70
No Background	Statistiche	Affinity Propagation	Naïve Bayes	0.73

Tabella 45Migliori Coppie F.S. - Classificatore

Confrontando i risultati ottenuti nei due dataset (completo vs. no background), si osserva che, sebbene le performance non siano particolarmente elevate in entrambi i casi, il dataset privo di background ha prodotto il classificatore con la performance più alta (AUC ROC = 0.73) nel caso di aggregazione

tramite statistiche descrittive. Questo risultato può essere interpretato alla luce della presenza di tiles di background nel dataset completo, che introducono rumore informativo nelle distribuzioni aggregate delle feature. Infatti, l'inclusione di aree prive di contenuto istologicamente rilevante può alterare la rappresentazione delle biopsie, ostacolando la capacità del classificatore di distinguere correttamente le classi. Al contrario, il dataset privo di background, pur contenendo un numero inferiore di tiles, appare più informativo e meno soggetto a confondenti. Questo si riflette nella maggiore stabilità dei risultati e nella performance lievemente superiore nel caso dell'aggregazione per statistiche. In conclusione, pur con una differenza modesta, si può affermare che il dataset senza background offre risultati più solidi e potenzialmente più affidabili per la classificazione, confermando l'importanza di una selezione accurata delle regioni d'interesse (ROI) nelle immagini digitali. Nelle seguenti riportiamo le matrici di confusione e le relative metriche delle migliori coppie viste in precedenza:

	Predicted	
True	Classe 0	Classe 1
Classe 0	6	1
Classe 1	12	9

Tabella 46 Affinity + Bagging DT

	Predicted	
True	Classe 0	Classe 1
Classe 0	17	0
Classe 1	11	0

Tabella 47RFE + SVM SIgmoid

	Predicted	
True	Classe 0	Classe 1
Classe 0	15	2
Classe 1	11	0

Tabella 48 RFE+SGD

	Predicted	
True	Classe 0	Classe 1
Classe 0	2	5
Classe 1	2	19

Tabella 49 Affinity + Naive Baies

Metodo F.S	**Classificatore**	**Sensitivity %**	**Specificity %**	**PPV %**	**NPV %**	**AUC ROC**
Affinity Propagation	Bagging DT	0,86	0,43	0,33	0,90	0,77
RFE+CV	SVM Sigmoid	1,00	0,00	0,09	Nan	0.58
RFE+CV	SGD	0,88	0,00	0,61	0,00	0.70
Affinity Propagation	Naive Bayes	0,29	0,90	0,50	0,79	0.73

Tabella 50 Metriche ricavate dalle matrici di confusione delle migliori coppie

Per quanto riguarda i risultati complessivi ricavati dal metodo PCA abbiamo la seguente tabella (tab.46):

Dataset	Aggregazione	Classificatore	Performance (AUC ROC)
Completo	Media	SGD	0.46
Completo	Statistiche	SVM Linear	0.50
No Background	Media	Naïve Bayes	0.50
No Backgorund	Statistiche	SVM Linear	0.54

Tabella 51 Migliori classificatori metodo PCA

L'applicazione del PCA come tecnica di riduzione dimensionale non ha portato a miglioramenti significativi delle performance. In tutti i casi testati, i valori di AUC ROC risultano prossimi o inferiori al caso random (0.50), con un massimo di 0.54. Questo comportamento può essere spiegato dal fatto che il PCA, pur essendo utile per ridurre la dimensionalità preservando la varianza, non considera l'informazione di classe, essendo una tecnica unsupervised. Di conseguenza, le componenti principali ottenute non sono necessariamente le più discriminative ai fini della classificazione, e possono aver escluso feature rilevanti per la separazione tra le classi. In questo contesto, dove le feature sono già derivate da aggregazioni statistiche o medie di immagini complesse, è plausibile che la variabilità massima catturata dal PCA non coincida con la variabilità utile per distinguere le classi. Per questa ragione, il PCA si è dimostrato non adatto in questo caso specifico come strumento di preprocessing per la classificazione.

Feature selezionate e valori SHAP

Per ottenere una visione d'insieme più qualitativa e comparativa sulle feature selezionate, è stata condotta un'analisi esplorativa sulla frequenza con cui ciascuna feature compare tra le prime dieci selezionate da ogni metodo di feature selection, sia per l'aggregazione tramite media che tramite statistiche descrittive. A tale scopo è stato implementato un programma in Python che legge i file prodotti dai metodi di selezione, estrae le prime dieci feature da ciascun file, e ne calcola la frequenza di occorrenza. Il risultato è rappresentato tramite un grafico a barre che mostra quali feature sono risultate più frequentemente selezionate all'interno di ciascun approccio di aggregazione.

Di seguito vengono riportati i grafici ottenuti dal programma (fig.65-66):

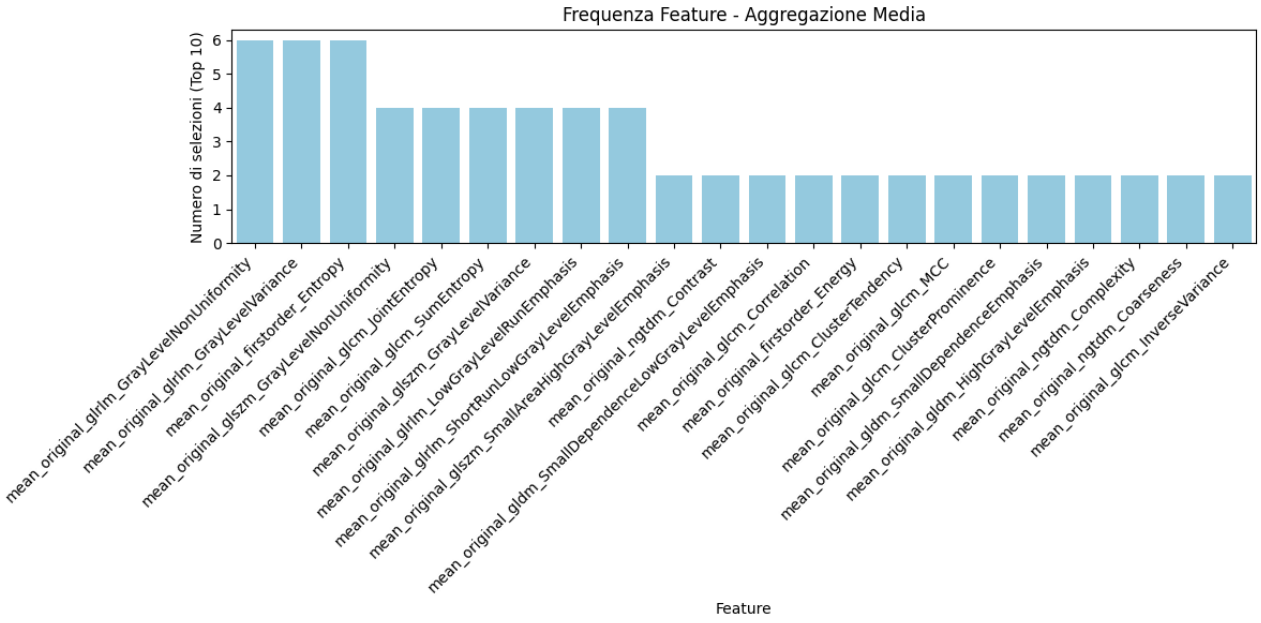


Figura 65 Feature selezionate metodo aggregazione media

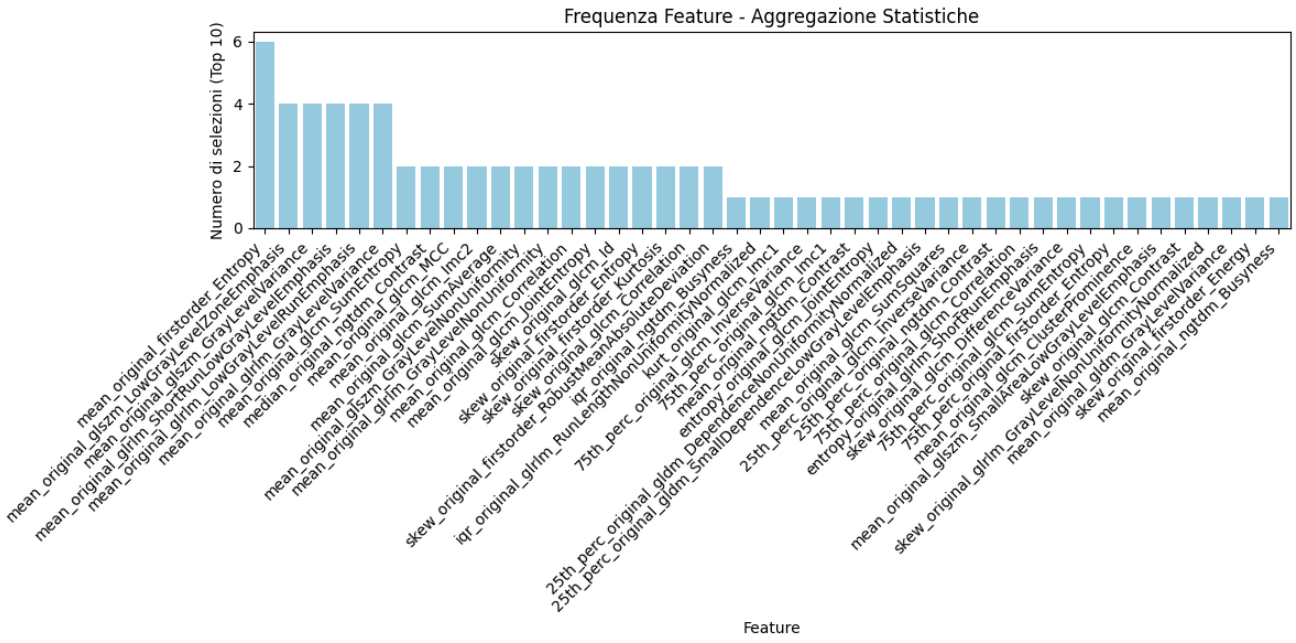


Figura 66 Feature selezonate metodo aggregazione statistiche descrittive

Lista feature con più alta frequenza (frequenza > 4) di selezione per il metodo aggregato media:

- mean_original_glrlm_GrayLevelNonUniformity
- mean_original_glrlm_GrayLevelVariance
- mean_original_firstorder_Entropy
- mean_original_glszm_GrayLevelNonUniformity
- mean_original_glcm_JointEntropy
- mean_original_glcm_SumEntropy
- mean_original_glszm_GrayLevelVariance
- mean_original_glrlm_LowGrayLevelRunEmphasis
- mean_original_glrlm_ShortRunLowGrayLevelEmphasis

Lista feature con più alta frequenza (frequenza > 4) di selezione per il metodo aggregato statistiche descrittive:

- mean_original_firstorder_Entropy
- mean_original_glszm_LowGrayLevelZoneEmphasis
- mean_original_glszm_GrayLevelVariance
- mean_original_glrlm_ShortRunLowGrayLevelEmphasis
- mean_original_glrlm_LowGrayLevelRunEmphasis
- mean_original_glrlm_GrayLevelVariance

In seguito all'analisi delle coppie migliori di metodi di feature selection e classificatori, è stata eseguita un'interpretazione dei modelli attraverso i valori SHAP (SHapley Additive exPlanations), focalizzandosi sulla Classe 1 (positiva). I summary plot dei valori SHAP consentono di visualizzare, per ciascuna feature, la distribuzione dell'impatto sul modello in funzione del valore assunto (rappresentato dalla scala cromatica), fornendo così una misura locale dell'importanza di ciascuna variabile nel processo decisionale del classificatore. Tuttavia, non è emersa una corrispondenza evidente tra le feature ritenute più importanti dai valori SHAP e quelle selezionate con maggiore frequenza dai metodi di feature selection. Le feature selezionate più frequentemente, ad esempio, per

il dataset senza background aggregato tramite statistiche descrittive, tendono a essere medie di descrittori statistici o strutturali (es. mean_original_firstorder_Entropy, mean_original_glszm_LowGrayLevelZoneEmphasis, mean_original_glrlm_GrayLevelVariance). Al contrario, nei summary plot dei valori SHAP relativi al modello Affinity Propagation + Naïve Bayes (accuratezza: 0.73), risultano più influenti feature legate a misure puntuali di kurtosi, skewness ed entropie locali, come kurt_original_ngtdm_Strength o kurt_original_glcm_DifferenceEntropy. Questa discrepanza potrebbe essere spiegata dal fatto che i metodi di feature selection applicati si basano su criteri globali o aggregati (es. correlazione, ridondanza, varianza), mentre i valori SHAP forniscono una valutazione locale dell'importanza delle feature, influenzata dal singolo paziente o caso. Ne consegue che una feature statisticamente rilevante su larga scala non è necessariamente decisiva per la predizione di singoli campioni, e viceversa. Inoltre, un'osservazione rilevante riguarda la presenza, nei summary plot, di valori isolati fortemente distanti dalla massa dei punti. Questi valori potrebbero costituire outlier nel dominio SHAP: singole osservazioni il cui impatto sul modello è estremamente elevato (sia in positivo che in negativo). Tali punti — caratterizzati da colori saturi e posizionati lontano dalla distribuzione centrale — potrebbero condizionare l'importanza apparente di una feature, facendola risultare tra le più influenti anche in presenza di un effetto marginale medio debole. A supporto di questa ipotesi, si nota come l'intensità dei valori SHAP vari considerevolmente tra i modelli:

- Per la maggior parte dei classificatori, l'impatto delle feature risulta contenuto, con valori compresi nell'intervallo ±0.0X
- Tuttavia, in pochi casi, emergono picchi molto elevati (es. SHAP value massimo ≈ 6000 con correlazione assoluta + SGD, oppure ≈ 3.6 con MRMR + SGD).

Tali picchi anomali potrebbero derivare da singoli esempi fortemente influenti, suggerendo la necessità di un'analisi più approfondita sui valori SHAP, volta a identificare e valutare l'effetto di eventuali outlier. In loro assenza, altre feature — meno influenzate da questi valori estremi — potrebbero assumere un ruolo relativamente più importante, e in tal modo risultare più allineate con quelle selezionate tramite feature selection. In sintesi, l'analisi congiunta dei metodi di selezione delle feature e dei valori SHAP ha evidenziato la complementarità di queste due prospettive: i metodi statistici offrono una visione aggregata e replicabile, mentre i valori SHAP forniscono insight interpretativi sull'inferenza del modello. Tuttavia, la presenza di possibili outlier nei valori SHAP e la variabilità nelle intensità suggeriscono di approfondire l'analisi futura con strumenti specifici, come il calcolo di media, varianza e skewness dei valori SHAP per ogni feature, oppure l'utilizzo di boxplot o density plot dei valori SHAP per meglio visualizzare la distribuzione e la presenza di anomalie.
Di seguito vengono mostrati i summary plot SHAP relativi alle migliori combinazioni metodo di feature selection - classificatore ottenute nei due scenari:

- Dataset completo aggregato con media: Affinity Propagation + Bagging Decision Tree (accuracy: 0.77) (fig.):

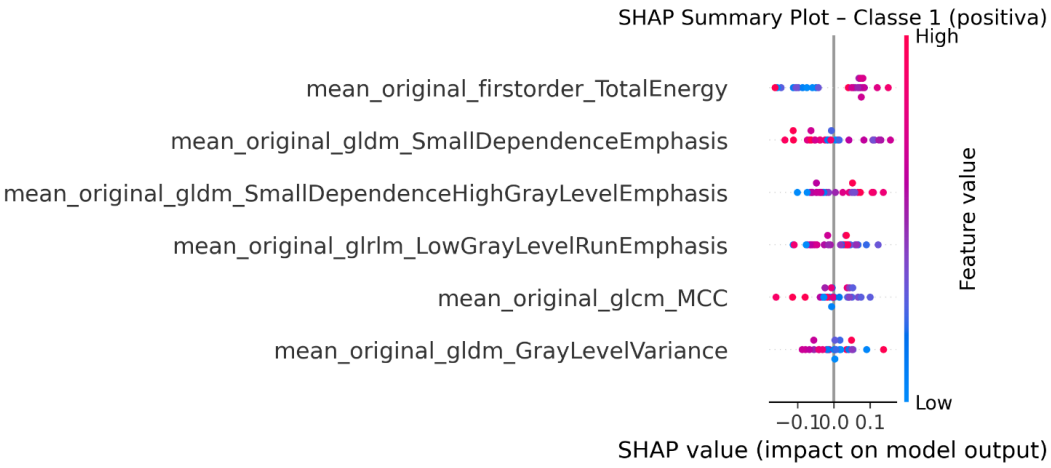


Figura 67 SHAP summary Affinity - Bagging DT

- Dataset senza background aggregato con statistiche: Affinity Propagation + Naïve Bayes (accuracy: 0.73):

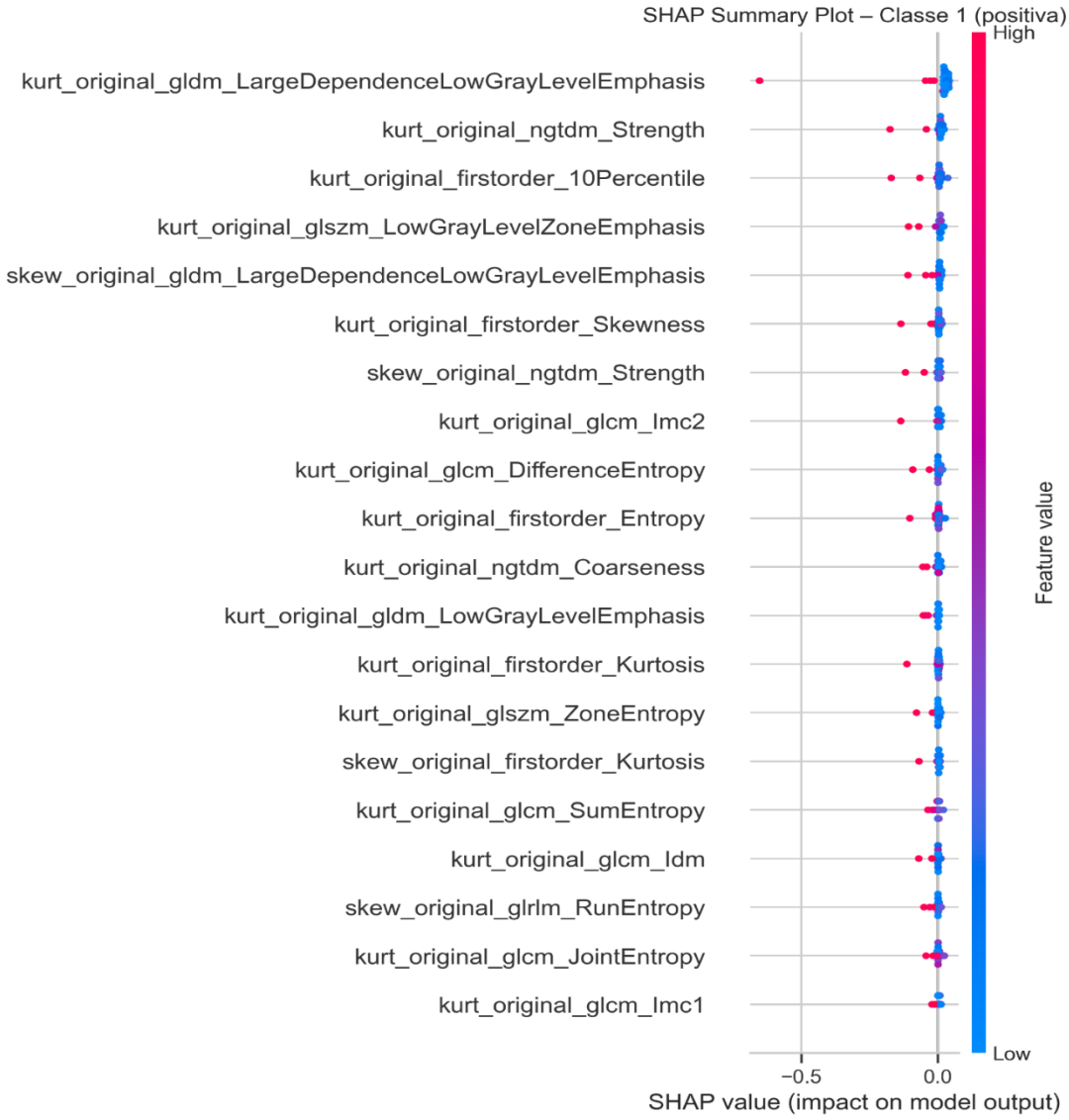


Figura 68 SHAP summary Affinity - Naive Bayes

Sviluppi futuri

Il lavoro svolto ha messo in luce diverse prospettive di sviluppo per approfondimenti futuri. Un primo passo fondamentale consiste nell'ampliamento del dataset, sia in termini di numerosità dei pazienti che di varietà dei casi clinici. In particolare, sarebbe utile evitare la presenza di pazienti con più biopsie, che ha richiesto in questo studio una suddivisione del dataset e l'esclusione di alcune feature instabili potenzialmente informative. Parallelamente, un'estensione del pool di metodi di feature selection e classificatori permetterebbe di esplorare combinazioni alternative, potenzialmente più efficaci, e di effettuare confronti più ampi tra le tecniche. In tale direzione, un'infrastruttura computazionale più performante rappresenterebbe un vantaggio significativo: ad esempio, l'adozione di workstation ad alte prestazioni basate su GPU di ultima generazione (come quelle recentemente introdotte da NVIDIA) consentirebbe di gestire dataset più ampi e algoritmi più complessi. Inoltre, un approfondimento dell'analisi dei valori SHAP potrebbe fornire nuove evidenze interpretative. In particolare, l'individuazione e l'eventuale rimozione di outlier tra i valori SHAP — ad esempio tramite metodi statistici o visuali — potrebbe modificare la gerarchia delle feature ritenute importanti, e favorire una maggiore convergenza tra le feature selezionate e quelle interpretativamente rilevanti. Dal punto di vista metodologico, infine, sarebbe interessante esplorare tecniche alternative di Explainable AI (XAI), come LIME (Local Interpretable Model-agnostic Explanations) [48], che fornisce spiegazioni locali basate su perturbazioni delle feature, o approcci basati sulla causalità o sull'attenzione neurale. L'integrazione di questi metodi può migliorare la trasparenza e l'affidabilità dell'intelligenza artificiale in ambito medico [49], e potenziare il valore clinico delle pipeline proposte.

Appendice

Elenco intero delle feature selezionate dal metodo Affinity Propagation per il dataset completo. Metodo di aggregazione: calcolo statistiche descrittive dell'istogramma di ogni feature.

- mean_original_firstorder_Entropy
- mean_original_glcm_SumAverage
- mean_original_gldm_GrayLevelVariance
- skew_original_firstorder_Energy
- skew_original_firstorder_Entropy
- skew_original_firstorder_Kurtosis
- skew_original_firstorder_RobustMeanAbsoluteDeviation
- skew_original_glcm_Correlation
- skew_original_glcm_DifferenceVariance
- skew_original_glcm_Id
- skew_original_glrlm_GrayLevelNonUniformityNormalized
- skew_original_glrlm_LowGrayLevelRunEmphasis
- skew_original_glrlm_RunEntropy
- skew_original_glrlm_RunLengthNonUniformityNormalized
- skew_original_glrlm_ShortRunLowGrayLevelEmphasis
- skew_original_glszm_GrayLevelNonUniformityNormalized
- skew_original_glszm_LowGrayLevelZoneEmphasis
- skew_original_glszm_SmallAreaEmphasis
- skew_original_glszm_ZoneEntropy
- skew_original_ngtdm_Strength
- skew_original_gldm_LargeDependenceLowGrayLevelEmphasis
- skew_original_gldm_LowGrayLevelEmphasis
- skew_original_gldm_SmallDependenceEmphasis
- skew_original_gldm_SmallDependenceHighGrayLevelEmphasis
- kurt_original_firstorder_10Percentile
- kurt_original_firstorder_Energy
- kurt_original_firstorder_Entropy
- kurt_original_firstorder_Kurtosis
- kurt_original_firstorder_MeanAbsoluteDeviation
- kurt_original_firstorder_Mean
- kurt_original_firstorder_Skewness
- kurt_original_glcm_ClusterShade
- kurt_original_glcm_Contrast
- kurt_original_glcm_Correlation
- kurt_original_glcm_DifferenceAverage
- kurt_original_glcm_DifferenceEntropy
- kurt_original_glcm_DifferenceVariance
- kurt_original_glcm_Idm
- kurt_original_glcm_Idmn
- kurt_original_glcm_Imc1
- kurt_original_glcm_Imc2
- kurt_original_glcm_JointAverage

- kurt_original_glcm_JointEntropy
- kurt_original_glcm_MCC
- kurt_original_glcm_SumEntropy
- kurt_original_glrlm_GrayLevelNonUniformity
- kurt_original_glrlm_RunEntropy
- kurt_original_glrlm_ShortRunEmphasis
- kurt_original_glrlm_ShortRunLowGrayLevelEmphasis
- kurt_original_glszm_GrayLevelNonUniformity
- kurt_original_glszm_GrayLevelNonUniformityNormalized
- kurt_original_glszm_GrayLevelVariance
- kurt_original_glszm_LowGrayLevelZoneEmphasis
- kurt_original_glszm_SmallAreaHighGrayLevelEmphasis
- kurt_original_glszm_SmallAreaLowGrayLevelEmphasis
- kurt_original_ngtdm_Busyness
- kurt_original_ngtdm_Coarseness
- kurt_original_ngtdm_Complexity
- kurt_original_ngtdm_Contrast
- kurt_original_ngtdm_Strength
- kurt_original_gldm_DependenceEntropy
- kurt_original_gldm_DependenceNonUniformity
- kurt_original_gldm_GrayLevelVariance
- kurt_original_gldm_LargeDependenceHighGrayLevelEmphasis
- kurt_original_gldm_LargeDependenceLowGrayLevelEmphasis
- kurt_original_gldm_LowGrayLevelEmphasis
- kurt_original_gldm_SmallDependenceLowGrayLevelEmphasis
- entropy_original_firstorder_Variance
- entropy_original_glrlm_LowGrayLevelRunEmphasis
- iqr_original_glrlm_GrayLevelVariance

Elenco delle 126 feature selezionate dal metodo Correlazione assoluta tra feature e target per il dataset completo. Metodo di aggregazione: calcolo statistiche descrittive dell'istogramma di ogni feature.

- mean_original_firstorder_Entropy
- mean_original_glcm_Correlation
- mean_original_glcm_Imc2
- mean_original_glcm_MCC
- mean_original_glrlm_GrayLevelVariance
- mean_original_glrlm_LowGrayLevelRunEmphasis
- mean_original_glrlm_ShortRunLowGrayLevelEmphasis
- mean_original_glszm_GrayLevelVariance
- mean_original_glszm_LowGrayLevelZoneEmphasis
- mean_original_glszm_SmallAreaLowGrayLevelEmphasis
- mean_original_ngtdm_Busyness
- mean_original_gldm_SmallDependenceLowGrayLevelEmphasis
- median_original_firstorder_Energy
- median_original_firstorder_Entropy
- median_original_firstorder_Mean

- median_original_firstorder_RootMeanSquared
- median_original_firstorder_TotalEnergy
- median_original_glcm_ClusterProminence
- median_original_glcm_Correlation
- median_original_glcm_Imc2
- median_original_glcm_MCC
- median_original_glrlm_GrayLevelVariance
- median_original_glrlm_LowGrayLevelRunEmphasis
- median_original_glrlm_ShortRunLowGrayLevelEmphasis
- median_original_glszm_GrayLevelVariance
- median_original_glszm_LowGrayLevelZoneEmphasis
- median_original_glszm_SmallAreaLowGrayLevelEmphasis
- median_original_gldm_SmallDependenceLowGrayLevelEmphasis
- skew_original_firstorder_10Percentile
- skew_original_glcm_Imc2
- skew_original_glcm_MCC
- skew_original_glrlm_ShortRunEmphasis
- skew_original_glszm_SmallAreaEmphasis
- kurt_original_firstorder_10Percentile
- kurt_original_firstorder_Mean
- kurt_original_firstorder_RootMeanSquared
- kurt_original_glcm_DifferenceEntropy
- kurt_original_glcm_Id
- kurt_original_glcm_Idm
- kurt_original_glrlm_LowGrayLevelRunEmphasis
- kurt_original_glrlm_RunEntropy
- kurt_original_glrlm_RunLengthNonUniformity
- kurt_original_glrlm_RunLengthNonUniformityNormalized
- kurt_original_glrlm_ShortRunEmphasis
- kurt_original_glszm_LowGrayLevelZoneEmphasis
- kurt_original_glszm_SizeZoneNonUniformity
- kurt_original_glszm_SizeZoneNonUniformityNormalized
- kurt_original_glszm_SmallAreaEmphasis
- kurt_original_glszm_ZoneEntropy
- kurt_original_ngtdm_Coarseness
- kurt_original_ngtdm_Strength
- kurt_original_gldm_DependenceNonUniformity
- kurt_original_gldm_DependenceNonUniformityNormalized
- kurt_original_gldm_SmallDependenceEmphasis
- entropy_original_firstorder_10Percentile
- entropy_original_firstorder_Entropy
- entropy_original_glcm_Correlation
- entropy_original_glcm_DifferenceAverage
- entropy_original_glcm_DifferenceEntropy
- entropy_original_glcm_Idmn
- entropy_original_glcm_Idn
- entropy_original_glcm_Imc2
- entropy_original_glcm_MCC

- entropy_original_glrlm_LowGrayLevelRunEmphasis
- entropy_original_glrlm_RunEntropy
- entropy_original_glrlm_ShortRunEmphasis
- entropy_original_glrlm_ShortRunHighGrayLevelEmphasis
- entropy_original_glrlm_ShortRunLowGrayLevelEmphasis
- entropy_original_glszm_LowGrayLevelZoneEmphasis
- entropy_original_glszm_SmallAreaEmphasis
- entropy_original_glszm_SmallAreaHighGrayLevelEmphasis
- entropy_original_glszm_SmallAreaLowGrayLevelEmphasis
- entropy_original_glszm_ZoneEntropy
- entropy_original_ngtdm_Complexity
- entropy_original_gldm_DependenceNonUniformity
- entropy_original_gldm_DependenceNonUniformityNormalized
- iqr_original_firstorder_10Percentile
- iqr_original_glcm_ClusterShade
- iqr_original_glcm_Contrast
- iqr_original_glcm_DifferenceAverage
- iqr_original_glcm_DifferenceEntropy
- iqr_original_glcm_Idmn
- iqr_original_glcm_Idn
- iqr_original_glcm_Imc1
- iqr_original_glcm_Imc2
- iqr_original_glrlm_RunEntropy
- iqr_original_glrlm_RunLengthNonUniformityNormalized
- iqr_original_glrlm_ShortRunEmphasis
- iqr_original_glrlm_ShortRunHighGrayLevelEmphasis
- iqr_original_glszm_SizeZoneNonUniformityNormalized
- iqr_original_glszm_SmallAreaEmphasis
- iqr_original_glszm_SmallAreaHighGrayLevelEmphasis
- iqr_original_glszm_ZoneEntropy
- iqr_original_ngtdm_Complexity
- iqr_original_gldm_DependenceNonUniformity
- iqr_original_gldm_DependenceNonUniformityNormalized
- iqr_original_gldm_SmallDependenceHighGrayLevelEmphasis
- 25th_perc_original_glcm_ClusterProminence
- 25th_perc_original_glcm_ClusterShade
- 25th_perc_original_glcm_Correlation
- 25th_perc_original_glcm_Imc2
- 25th_perc_original_glcm_MCC
- 25th_perc_original_glrlm_ShortRunLowGrayLevelEmphasis
- 25th_perc_original_glszm_SmallAreaLowGrayLevelEmphasis
- 25th_perc_original_ngtdm_Busyness
- 25th_perc_original_ngtdm_Contrast
- 25th_perc_original_gldm_DependenceVariance
- 25th_perc_original_gldm_SmallDependenceLowGrayLevelEmphasis
- 75th_perc_original_firstorder_Energy
- 75th_perc_original_firstorder_Entropy
- 75th_perc_original_firstorder_Mean

- 75th_perc_original_firstorder_RootMeanSquared
- 75th_perc_original_firstorder_TotalEnergy
- 75th_perc_original_glcm_Imc1
- 75th_perc_original_glcm_MCC
- 75th_perc_original_glrlm_GrayLevelVariance
- 75th_perc_original_glrlm_LowGrayLevelRunEmphasis
- 75th_perc_original_glrlm_ShortRunLowGrayLevelEmphasis
- 75th_perc_original_glszm_GrayLevelVariance
- 75th_perc_original_glszm_LowGrayLevelZoneEmphasis
- 75th_perc_original_glszm_SmallAreaLowGrayLevelEmphasis
- 75th_perc_original_ngtdm_Busyness
- 75th_perc_original_gldm_DependenceNonUniformity
- 75th_perc_original_gldm_DependenceNonUniformityNormalized
- 75th_perc_original_gldm_SmallDependenceLowGrayLevelEmphasis

Elenco completo delle feature selezionate dal metodo MRMR con il dataset completo. Metodo di aggregazione: calcolo statistiche descrittive dell'istogramma di ogni feature.

- mean_original_firstorder_Entropy
- mean_original_glcm_ClusterProminence
- mean_original_glcm_Correlation
- mean_original_glcm_Imc2
- mean_original_glcm_MCC
- mean_original_glrlm_GrayLevelVariance
- mean_original_glrlm_LowGrayLevelRunEmphasis
- mean_original_glrlm_ShortRunLowGrayLevelEmphasis
- mean_original_glszm_GrayLevelVariance
- mean_original_glszm_LowGrayLevelZoneEmphasis
- mean_original_glszm_SmallAreaLowGrayLevelEmphasis
- mean_original_ngtdm_Busyness
- mean_original_ngtdm_Contrast
- mean_original_gldm_SmallDependenceLowGrayLevelEmphasis
- median_original_firstorder_Energy
- median_original_firstorder_Entropy
- median_original_firstorder_Mean
- median_original_firstorder_RootMeanSquared
- median_original_firstorder_Skewness
- median_original_firstorder_TotalEnergy
- median_original_glcm_ClusterProminence
- median_original_glcm_Correlation
- median_original_glcm_Imc2
- median_original_glcm_MCC
- median_original_glrlm_GrayLevelVariance
- median_original_glrlm_LowGrayLevelRunEmphasis
- median_original_glrlm_ShortRunLowGrayLevelEmphasis
- median_original_glszm_GrayLevelVariance
- median_original_glszm_LowGrayLevelZoneEmphasis
- median_original_glszm_SmallAreaLowGrayLevelEmphasis

- median_original_ngtdm_Contrast
- median_original_gldm_SmallDependenceLowGrayLevelEmphasis
- skew_original_firstorder_10Percentile
- skew_original_glcm_Imc2
- skew_original_glcm_MCC
- kurt_original_firstorder_10Percentile
- kurt_original_firstorder_RootMeanSquared
- kurt_original_firstorder_Skewness
- kurt_original_glcm_DifferenceEntropy
- kurt_original_glrlm_GrayLevelNonUniformity
- kurt_original_glrlm_LowGrayLevelRunEmphasis
- kurt_original_glrlm_RunEntropy
- kurt_original_glrlm_RunLengthNonUniformity
- kurt_original_glrlm_RunLengthNonUniformityNormalized
- kurt_original_glrlm_ShortRunEmphasis
- kurt_original_glszm_GrayLevelNonUniformity
- kurt_original_glszm_LowGrayLevelZoneEmphasis
- kurt_original_glszm_SizeZoneNonUniformityNormalized
- kurt_original_glszm_SmallAreaEmphasis
- kurt_original_glszm_ZoneEntropy
- kurt_original_ngtdm_Coarseness
- kurt_original_ngtdm_Strength
- kurt_original_gldm_DependenceNonUniformity
- kurt_original_gldm_DependenceNonUniformityNormalized
- kurt_original_gldm_LowGrayLevelEmphasis
- entropy_original_firstorder_10Percentile
- entropy_original_firstorder_Entropy
- entropy_original_glcm_Correlation
- entropy_original_glcm_DifferenceAverage
- entropy_original_glcm_DifferenceEntropy
- entropy_original_glcm_Idmn
- entropy_original_glcm_Imc2
- entropy_original_glcm_MCC
- entropy_original_glrlm_LowGrayLevelRunEmphasis
- entropy_original_glrlm_RunEntropy
- entropy_original_glrlm_ShortRunHighGrayLevelEmphasis
- entropy_original_glrlm_ShortRunLowGrayLevelEmphasis
- entropy_original_glszm_LowGrayLevelZoneEmphasis
- entropy_original_glszm_SmallAreaHighGrayLevelEmphasis
- entropy_original_glszm_SmallAreaLowGrayLevelEmphasis
- entropy_original_glszm_ZoneEntropy
- entropy_original_ngtdm_Complexity
- entropy_original_gldm_DependenceNonUniformity
- entropy_original_gldm_DependenceNonUniformityNormalized
- entropy_original_gldm_SmallDependenceHighGrayLevelEmphasis
- iqr_original_firstorder_10Percentile
- iqr_original_glcm_ClusterShade
- iqr_original_glcm_Contrast

- iqr_original_glcm_DifferenceAverage
- iqr_original_glcm_DifferenceEntropy
- iqr_original_glcm_Idmn
- iqr_original_glcm_Idn
- iqr_original_glcm_Imc2
- iqr_original_glrlm_RunEntropy
- iqr_original_glrlm_RunLengthNonUniformityNormalized
- iqr_original_glrlm_ShortRunEmphasis
- iqr_original_glrlm_ShortRunHighGrayLevelEmphasis
- iqr_original_glszm_SizeZoneNonUniformityNormalized
- iqr_original_glszm_SmallAreaEmphasis
- iqr_original_glszm_SmallAreaHighGrayLevelEmphasis
- iqr_original_glszm_ZoneEntropy
- iqr_original_ngtdm_Complexity
- iqr_original_gldm_DependenceNonUniformity
- iqr_original_gldm_DependenceNonUniformityNormalized
- iqr_original_gldm_SmallDependenceHighGrayLevelEmphasis
- 25th_perc_original_firstorder_Skewness
- 25th_perc_original_glcm_ClusterProminence
- 25th_perc_original_glcm_ClusterShade
- 25th_perc_original_glcm_Correlation
- 25th_perc_original_glcm_Imc2
- 25th_perc_original_glcm_MCC
- 25th_perc_original_glrlm_GrayLevelVariance
- 25th_perc_original_glrlm_ShortRunLowGrayLevelEmphasis
- 25th_perc_original_glszm_GrayLevelVariance
- 25th_perc_original_glszm_SmallAreaLowGrayLevelEmphasis
- 25th_perc_original_ngtdm_Busyness
- 25th_perc_original_ngtdm_Contrast
- 25th_perc_original_gldm_DependenceVariance
- 25th_perc_original_gldm_SmallDependenceLowGrayLevelEmphasis
- 75th_perc_original_firstorder_Energy
- 75th_perc_original_firstorder_Entropy
- 75th_perc_original_firstorder_Mean
- 75th_perc_original_firstorder_RootMeanSquared
- 75th_perc_original_firstorder_TotalEnergy
- 75th_perc_original_glcm_Imc1
- 75th_perc_original_glcm_MCC
- 75th_perc_original_glrlm_GrayLevelVariance
- 75th_perc_original_glrlm_ShortRunLowGrayLevelEmphasis
- 75th_perc_original_glszm_GrayLevelVariance
- 75th_perc_original_glszm_SmallAreaLowGrayLevelEmphasis
- 75th_perc_original_ngtdm_Busyness
- 75th_perc_original_gldm_DependenceNonUniformity
- 75th_perc_original_gldm_DependenceNonUniformityNormalized

Elenco completo delle feature selezionate dal metodo Affinity Propagation con il dataset privo di backgorund. Metodo di aggregazione: calcolo statistiche descrittive dell'istogramma di ogni feature.

- mean_original_firstorder_Entropy
- mean_original_glcm_SumAverage
- mean_original_glcm_SumSquares
- skew_original_firstorder_Entropy
- skew_original_firstorder_Kurtosis
- skew_original_firstorder_RobustMeanAbsoluteDeviation
- skew_original_glcm_Contrast
- skew_original_glcm_Correlation
- skew_original_glcm_Id
- skew_original_glrlm_GrayLevelNonUniformityNormalized
- skew_original_glrlm_LowGrayLevelRunEmphasis
- skew_original_glrlm_RunEntropy
- skew_original_glrlm_RunLengthNonUniformityNormalized
- skew_original_glrlm_ShortRunLowGrayLevelEmphasis
- skew_original_glszm_GrayLevelNonUniformityNormalized
- skew_original_glszm_SmallAreaEmphasis
- skew_original_glszm_ZoneEntropy
- skew_original_ngtdm_Strength
- skew_original_gldm_HighGrayLevelEmphasis
- skew_original_gldm_LargeDependenceLowGrayLevelEmphasis
- skew_original_gldm_LowGrayLevelEmphasis
- skew_original_gldm_SmallDependenceEmphasis
- kurt_original_firstorder_10Percentile
- kurt_original_firstorder_Entropy
- kurt_original_firstorder_Kurtosis
- kurt_original_firstorder_RobustMeanAbsoluteDeviation
- kurt_original_firstorder_RootMeanSquared
- kurt_original_firstorder_Skewness
- kurt_original_firstorder_TotalEnergy
- kurt_original_glcm_ClusterProminence
- kurt_original_glcm_ClusterShade
- kurt_original_glcm_Contrast
- kurt_original_glcm_Correlation
- kurt_original_glcm_DifferenceAverage
- kurt_original_glcm_DifferenceEntropy
- kurt_original_glcm_DifferenceVariance
- kurt_original_glcm_Idm
- kurt_original_glcm_Idmn
- kurt_original_glcm_Imc1
- kurt_original_glcm_Imc2
- kurt_original_glcm_JointEntropy
- kurt_original_glcm_MCC
- kurt_original_glcm_SumEntropy
- kurt_original_glrlm_GrayLevelNonUniformity

- kurt_original_glrlm_RunLengthNonUniformityNormalized
- kurt_original_glrlm_RunPercentage
- kurt_original_glrlm_ShortRunEmphasis
- kurt_original_glrlm_ShortRunHighGrayLevelEmphasis
- kurt_original_glrlm_ShortRunLowGrayLevelEmphasis
- kurt_original_glszm_GrayLevelNonUniformity
- kurt_original_glszm_GrayLevelNonUniformityNormalized
- kurt_original_glszm_GrayLevelVariance
- kurt_original_glszm_LowGrayLevelZoneEmphasis
- kurt_original_glszm_SizeZoneNonUniformityNormalized
- kurt_original_glszm_SmallAreaLowGrayLevelEmphasis
- kurt_original_glszm_ZoneEntropy
- kurt_original_ngtdm_Busyness
- kurt_original_ngtdm_Coarseness
- kurt_original_ngtdm_Complexity
- kurt_original_ngtdm_Contrast
- kurt_original_ngtdm_Strength
- kurt_original_gldm_DependenceEntropy
- kurt_original_gldm_DependenceNonUniformity
- kurt_original_gldm_HighGrayLevelEmphasis
- kurt_original_gldm_LargeDependenceHighGrayLevelEmphasis
- kurt_original_gldm_LargeDependenceLowGrayLevelEmphasis
- kurt_original_gldm_LowGrayLevelEmphasis
- kurt_original_gldm_SmallDependenceLowGrayLevelEmphasis
- entropy_original_glcm_DifferenceEntropy
- entropy_original_gldm_GrayLevelVariance
- iqr_original_firstorder_Entropy

Elenco completo delle feature selezionate dal metodo Correlazione Assoluta con il dataset privo di backgorud. Metodo di aggregazione: calcolo statistiche descrittive dell'istogramma di ogni feature.

- mean_original_firstorder_Entropy
- mean_original_glcm_JointEntropy
- mean_original_glcm_SumEntropy
- mean_original_glrlm_GrayLevelNonUniformity
- mean_original_glrlm_GrayLevelVariance
- mean_original_glrlm_LowGrayLevelRunEmphasis
- mean_original_glrlm_ShortRunLowGrayLevelEmphasis
- mean_original_glszm_GrayLevelNonUniformity
- mean_original_glszm_GrayLevelVariance
- mean_original_glszm_LowGrayLevelZoneEmphasis
- mean_original_glszm_SmallAreaLowGrayLevelEmphasis
- mean_original_gldm_LargeDependenceHighGrayLevelEmphasis
- mean_original_gldm_LargeDependenceLowGrayLevelEmphasis
- mean_original_gldm_SmallDependenceLowGrayLevelEmphasis
- median_original_firstorder_Entropy
- median_original_glcm_ClusterProminence
- median_original_glcm_JointEntropy

- median_original_glcm_SumEntropy
- median_original_glrlm_GrayLevelNonUniformity
- median_original_glrlm_GrayLevelVariance
- median_original_glrlm_LowGrayLevelRunEmphasis
- median_original_glrlm_ShortRunLowGrayLevelEmphasis
- median_original_glszm_GrayLevelNonUniformity
- median_original_glszm_GrayLevelVariance
- median_original_glszm_LowGrayLevelZoneEmphasis
- median_original_glszm_SmallAreaLowGrayLevelEmphasis
- median_original_ngtdm_Contrast
- median_original_gldm_LargeDependenceHighGrayLevelEmphasis
- median_original_gldm_LargeDependenceLowGrayLevelEmphasis
- median_original_gldm_SmallDependenceLowGrayLevelEmphasis
- skew_original_firstorder_Kurtosis
- skew_original_glcm_DifferenceEntropy
- skew_original_glcm_JointEntropy
- skew_original_glrlm_GrayLevelVariance
- skew_original_glrlm_LowGrayLevelRunEmphasis
- skew_original_glrlm_ShortRunLowGrayLevelEmphasis
- skew_original_glszm_GrayLevelVariance
- skew_original_glszm_LowGrayLevelZoneEmphasis
- skew_original_glszm_SmallAreaLowGrayLevelEmphasis
- skew_original_gldm_LargeDependenceHighGrayLevelEmphasis
- skew_original_gldm_LargeDependenceLowGrayLevelEmphasis
- skew_original_gldm_LowGrayLevelEmphasis
- kurt_original_firstorder_10Percentile
- kurt_original_firstorder_Entropy
- kurt_original_firstorder_Kurtosis
- kurt_original_firstorder_Skewness
- kurt_original_glcm_ClusterShade
- kurt_original_glcm_Contrast
- kurt_original_glcm_DifferenceEntropy
- kurt_original_glcm_DifferenceVariance
- kurt_original_glcm_Idmn
- kurt_original_glcm_Imc2
- kurt_original_glcm_JointEntropy
- kurt_original_glcm_SumEntropy
- kurt_original_glrlm_HighGrayLevelRunEmphasis
- kurt_original_glrlm_LowGrayLevelRunEmphasis
- kurt_original_glrlm_RunEntropy
- kurt_original_glrlm_RunLengthNonUniformityNormalized
- kurt_original_glrlm_ShortRunEmphasis
- kurt_original_glrlm_ShortRunHighGrayLevelEmphasis
- kurt_original_glrlm_ShortRunLowGrayLevelEmphasis
- kurt_original_glszm_HighGrayLevelZoneEmphasis
- kurt_original_glszm_LowGrayLevelZoneEmphasis
- kurt_original_glszm_SizeZoneNonUniformityNormalized
- kurt_original_glszm_SmallAreaEmphasis

- kurt_original_glszm_SmallAreaHighGrayLevelEmphasis
- kurt_original_glszm_SmallAreaLowGrayLevelEmphasis
- kurt_original_glszm_ZoneEntropy
- kurt_original_ngtdm_Coarseness
- kurt_original_ngtdm_Complexity
- kurt_original_ngtdm_Strength
- kurt_original_gldm_LargeDependenceLowGrayLevelEmphasis
- kurt_original_gldm_LowGrayLevelEmphasis
- kurt_original_gldm_SmallDependenceHighGrayLevelEmphasis
- kurt_original_gldm_SmallDependenceLowGrayLevelEmphasis
- entropy_original_firstorder_Entropy
- entropy_original_glcm_Contrast
- entropy_original_glcm_DifferenceVariance
- entropy_original_glcm_Idmn
- entropy_original_glcm_Imc2
- entropy_original_glrlm_ShortRunLowGrayLevelEmphasis
- entropy_original_glszm_SmallAreaLowGrayLevelEmphasis
- entropy_original_ngtdm_Complexity
- iqr_original_firstorder_MeanAbsoluteDeviation
- iqr_original_firstorder_Median
- iqr_original_firstorder_Variance
- iqr_original_glcm_ClusterShade
- iqr_original_glcm_Contrast
- iqr_original_glcm_DifferenceAverage
- iqr_original_glcm_DifferenceVariance
- iqr_original_glcm_Idmn
- iqr_original_glcm_Imc2
- iqr_original_glrlm_GrayLevelNonUniformity
- iqr_original_glrlm_ShortRunHighGrayLevelEmphasis
- iqr_original_glrlm_ShortRunLowGrayLevelEmphasis
- iqr_original_glszm_GrayLevelNonUniformity
- iqr_original_glszm_SmallAreaHighGrayLevelEmphasis
- iqr_original_glszm_SmallAreaLowGrayLevelEmphasis
- iqr_original_ngtdm_Complexity
- iqr_original_gldm_DependenceNonUniformity
- iqr_original_gldm_DependenceNonUniformityNormalized
- iqr_original_gldm_SmallDependenceHighGrayLevelEmphasis
- 25th_perc_original_firstorder_Entropy
- 25th_perc_original_glcm_ClusterProminence
- 25th_perc_original_glcm_Idmn
- 25th_perc_original_glrlm_GrayLevelVariance
- 25th_perc_original_glrlm_LowGrayLevelRunEmphasis
- 25th_perc_original_glrlm_ShortRunLowGrayLevelEmphasis
- 25th_perc_original_glszm_GrayLevelVariance
- 25th_perc_original_glszm_LowGrayLevelZoneEmphasis
- 25th_perc_original_glszm_SmallAreaLowGrayLevelEmphasis
- 25th_perc_original_ngtdm_Contrast
- 25th_perc_original_gldm_LargeDependenceHighGrayLevelEmphasis

- 25th_perc_original_gldm_LargeDependenceLowGrayLevelEmphasis
- 25th_perc_original_gldm_LowGrayLevelEmphasis
- 25th_perc_original_gldm_SmallDependenceLowGrayLevelEmphasis
- 75th_perc_original_firstorder_Entropy
- 75th_perc_original_glcm_JointEntropy
- 75th_perc_original_glcm_SumEntropy
- 75th_perc_original_glrlm_GrayLevelNonUniformity
- 75th_perc_original_glrlm_ShortRunLowGrayLevelEmphasis
- 75th_perc_original_glszm_GrayLevelNonUniformity
- 75th_perc_original_glszm_SmallAreaLowGrayLevelEmphasis
- 75th_perc_original_gldm_SmallDependenceLowGrayLevelEmphasis

Elenco completo delle feature selezionate dal metodo MRMR con il dataset privo di backgorund. Metodo di aggregazione: calcolo statistiche descrittive dell'istogramma di ogni feature.

- mean_original_firstorder_Entropy
- mean_original_glcm_JointEntropy
- mean_original_glcm_SumEntropy
- mean_original_glrlm_GrayLevelNonUniformity
- mean_original_glrlm_GrayLevelVariance
- mean_original_glrlm_LowGrayLevelRunEmphasis
- mean_original_glrlm_ShortRunLowGrayLevelEmphasis
- mean_original_glszm_GrayLevelNonUniformity
- mean_original_glszm_GrayLevelVariance
- mean_original_glszm_LowGrayLevelZoneEmphasis
- mean_original_glszm_SmallAreaLowGrayLevelEmphasis
- mean_original_ngtdm_Busyness
- mean_original_ngtdm_Contrast
- mean_original_gldm_LargeDependenceHighGrayLevelEmphasis
- mean_original_gldm_LargeDependenceLowGrayLevelEmphasis
- mean_original_gldm_SmallDependenceLowGrayLevelEmphasis
- median_original_firstorder_Energy
- median_original_firstorder_Entropy
- median_original_firstorder_TotalEnergy
- median_original_firstorder_Variance
- median_original_glcm_ClusterProminence
- median_original_glcm_JointEntropy
- median_original_glcm_SumEntropy
- median_original_glrlm_GrayLevelNonUniformity
- median_original_glrlm_GrayLevelVariance
- median_original_glrlm_ShortRunLowGrayLevelEmphasis
- median_original_glszm_GrayLevelNonUniformity
- median_original_glszm_GrayLevelVariance
- median_original_glszm_SmallAreaLowGrayLevelEmphasis
- median_original_ngtdm_Contrast
- median_original_gldm_LargeDependenceHighGrayLevelEmphasis
- median_original_gldm_LargeDependenceLowGrayLevelEmphasis
- median_original_gldm_SmallDependenceLowGrayLevelEmphasis

- skew_original_firstorder_Kurtosis
- skew_original_glrlm_GrayLevelVariance
- skew_original_glrlm_LowGrayLevelRunEmphasis
- skew_original_glrlm_ShortRunLowGrayLevelEmphasis
- skew_original_glszm_GrayLevelVariance
- skew_original_glszm_LowGrayLevelZoneEmphasis
- skew_original_glszm_SmallAreaLowGrayLevelEmphasis
- skew_original_gldm_LargeDependenceLowGrayLevelEmphasis
- skew_original_gldm_LowGrayLevelEmphasis
- kurt_original_firstorder_10Percentile
- kurt_original_firstorder_Entropy
- kurt_original_firstorder_Kurtosis
- kurt_original_firstorder_Skewness
- kurt_original_glcm_Contrast
- kurt_original_glcm_DifferenceEntropy
- kurt_original_glcm_DifferenceVariance
- kurt_original_glcm_Idmn
- kurt_original_glcm_Imc2
- kurt_original_glcm_JointEntropy
- kurt_original_glcm_SumEntropy
- kurt_original_glrlm_HighGrayLevelRunEmphasis
- kurt_original_glrlm_LowGrayLevelRunEmphasis
- kurt_original_glrlm_RunEntropy
- kurt_original_glrlm_ShortRunEmphasis
- kurt_original_glrlm_ShortRunHighGrayLevelEmphasis
- kurt_original_glrlm_ShortRunLowGrayLevelEmphasis
- kurt_original_glszm_HighGrayLevelZoneEmphasis
- kurt_original_glszm_LowGrayLevelZoneEmphasis
- kurt_original_glszm_SmallAreaEmphasis
- kurt_original_glszm_SmallAreaHighGrayLevelEmphasis
- kurt_original_glszm_SmallAreaLowGrayLevelEmphasis
- kurt_original_glszm_ZoneEntropy
- kurt_original_ngtdm_Coarseness
- kurt_original_ngtdm_Complexity
- kurt_original_gldm_DependenceNonUniformityNormalized
- kurt_original_gldm_DependenceVariance
- kurt_original_gldm_LargeDependenceLowGrayLevelEmphasis
- kurt_original_gldm_LowGrayLevelEmphasis
- kurt_original_gldm_SmallDependenceHighGrayLevelEmphasis
- kurt_original_gldm_SmallDependenceLowGrayLevelEmphasis
- entropy_original_firstorder_10Percentile
- entropy_original_firstorder_Entropy
- entropy_original_glcm_Contrast
- entropy_original_glcm_DifferenceVariance
- entropy_original_glcm_Idmn
- entropy_original_glcm_Imc2
- entropy_original_glrlm_ShortRunLowGrayLevelEmphasis
- entropy_original_glszm_SmallAreaLowGrayLevelEmphasis

- entropy_original_ngtdm_Complexity
- entropy_original_gldm_DependenceVariance
- iqr_original_firstorder_MeanAbsoluteDeviation
- iqr_original_firstorder_Median
- iqr_original_firstorder_Variance
- iqr_original_glcm_Contrast
- iqr_original_glcm_Idmn
- iqr_original_glcm_Imc2
- iqr_original_glrlm_GrayLevelNonUniformity
- iqr_original_glrlm_ShortRunHighGrayLevelEmphasis
- iqr_original_glrlm_ShortRunLowGrayLevelEmphasis
- iqr_original_glszm_GrayLevelNonUniformity
- iqr_original_glszm_SmallAreaHighGrayLevelEmphasis
- iqr_original_glszm_SmallAreaLowGrayLevelEmphasis
- iqr_original_ngtdm_Complexity
- iqr_original_gldm_DependenceNonUniformity
- iqr_original_gldm_DependenceNonUniformityNormalized
- iqr_original_gldm_SmallDependenceHighGrayLevelEmphasis
- 25th_perc_original_firstorder_Entropy
- 25th_perc_original_firstorder_Variance
- 25th_perc_original_glcm_ClusterProminence
- 25th_perc_original_glcm_Idmn
- 25th_perc_original_glrlm_GrayLevelVariance
- 25th_perc_original_glrlm_LowGrayLevelRunEmphasis
- 25th_perc_original_glrlm_ShortRunLowGrayLevelEmphasis
- 25th_perc_original_glszm_GrayLevelVariance
- 25th_perc_original_glszm_LowGrayLevelZoneEmphasis
- 25th_perc_original_glszm_SmallAreaLowGrayLevelEmphasis
- 25th_perc_original_ngtdm_Busyness
- 25th_perc_original_ngtdm_Contrast
- 25th_perc_original_gldm_LargeDependenceHighGrayLevelEmphasis
- 25th_perc_original_gldm_LargeDependenceLowGrayLevelEmphasis
- 25th_perc_original_gldm_SmallDependenceLowGrayLevelEmphasis
- 75th_perc_original_firstorder_Entropy
- 75th_perc_original_glcm_Contrast
- 75th_perc_original_glcm_JointEntropy
- 75th_perc_original_glcm_SumEntropy
- 75th_perc_original_glrlm_GrayLevelNonUniformity
- 75th_perc_original_glrlm_ShortRunLowGrayLevelEmphasis
- 75th_perc_original_glszm_GrayLevelNonUniformity
- 75th_perc_original_glszm_SmallAreaLowGrayLevelEmphasis
- 75th_perc_original_gldm_SmallDependenceLowGrayLevelEmphasis

Bibliografia

[1] https://www.airc.it/cancro/informazioni-tumori/guida-ai-tumori/tumore-colon-retto

[2] https://www.ospedaleniguarda.it/news/leggi/tumore-del-retto-localmente-avanzato-remissione-completa-per-1-persona-su-4-anche-senza-chirurgia-niguarda-promotore-dello-studio-no-cut

[3] https://en.wikipedia.org/wiki/Artificial_intelligence_in_healthcare?utm

[4] Hoseini et al. (MDPI 2024), Biomarkers in CRC: Actual and Future Perspectives – Biopsia liquida, CTC, ctDNA, uso dell'AI per migliorare sens/spec

[5] Rosati, Giannini et al. "Development and validation of an AI-based pathomics biomarker to predict response to first-line treatment in metastatic colorectal cancers" Paper No. ICBB 122 DOI: 10.11159/icbb24.122

[6] Oliva I. Biomarcatore radiopatomico basato sull'intelligenza artificiale per la predizione della risposta patologica nei pazienti con cancro del retto = A radio-pathomics AI-based biomarker to predict pathological response in patients with rectal cancer [Master's thesis]. Politecnico di Torino; 2024. Available from: https://webthesis.biblio.polito.it/33670/

[7] https://pyradiomics.readthedocs.io/en/latest/

[8] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," J. Mach. Learn. Res., vol. 3, pp. 1157–1182, 2003

[9] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," Machine Learning, vol. 46, no. 1–3, pp. 389–422, 2002.

[10] https://feature-engine.trainindata.com/en/1.8.x/user_guide/selection/MRMR.html

[11] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," Science, vol. 315, no. 5814, pp. 972–976, 2007

[12] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," Science, vol. 315, no. 5814, pp. 972–976, 2007.

[13] K. P. Murphy, Machine Learning: A Probabilistic Perspective. MIT Press, 2012.

[14] T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21–27, 1967.

[15] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.

[16] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," Data Mining and Knowledge Discovery, vol. 2, no. 2, pp. 121–167, 1998.

[17] C. E. Shannon, "A Mathematical Theory of Communication," Bell System Technical Journal, vol. 27, no. 3, pp. 379–423, 1948.

[18] J. R. Quinlan, "Induction of decision trees," Machine Learning, vol. 1, no. 1, pp. 81–106, 1986.

[19] P. J. R. Hepp et al., “Classification and Regression Trees, Bagging, and Boosting,” Computer Science Reviews, 2005.

[20] L. Breiman, “Bagging predictors,” Machine Learning, vol. 24, no. 2, pp. 123–140, 1996.

[21] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” J. Comput. Syst. Sci., vol. 55, no. 1, pp. 119–139, 1997.

[22] T. G. Dietterich, “An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization,” Machine Learning, vol. 40, pp. 139–157, 2000.

[23] A. Rahman and S. Tasnim, “Ensemble classifiers and their applications: A review,” arXiv preprint arXiv:1412.1847, 2014.

[24] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[25] G. Biau and E. Scornet, “A random forest guided tour,” Test, vol. 25, no. 2, pp. 197–227, 2016.

[26] L. Bottou, “Stochastic Gradient Descent Tricks,” in *Neural Networks: Tricks of the Trade*, Springer, 2012, pp. 421–436.

[27] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[28] S. Sinsomboonthong, “Performance Comparison of New Adjusted Min-Max with Decimal Scaling and Statistical Column Normalization Methods for Artificial Neural Network Classification,” Int. J. Math. & Math. Sci., 2022.

[29] L. B. V. de Amorim, G. D. C. Cavalcanti, and R. M. O. Cruz, “The choice of scaling technique matters for classification performance,” 2022.

[30] https://en.wikipedia.org/wiki/Shapiro%E2%80%93Wilk_test

[31] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.shapiro.html

[32] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AffinityPropagation.html

[33] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html

[34] https://kneed.readthedocs.io/en/stable/api.html#kneelocator

[35] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[36] Cao et al. (2020) mostrano l’importanza di considerare criteri compositi (F1, MCC) per una valutazione completa della performance

[37] Drury (2017) sottolinea la robustezza dell’AUC-ROC di fronte a dataset sbilanciati

[38] Il thread su Stack Overflow (2018) evidenzia la differenza tra F1 (soglia dipendente) e AUC (valutazione globale sul ranking)

[39] La review di Naidu et al. (2023) conferma che l'AUC-ROC è una metrica riconosciuta e standardizzata in contesti clinici e industriali

[40] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," arXiv preprint arXiv:1705.07874, 2017.

[41] E. Štrumbelj and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions," Knowl. Inf. Syst., vol. 41, no. 3, pp. 647–665, 2014.

[42] C. Molnar, Interpretable Machine Learning: A Guide for Making Black Box Models Explainable, 2nd ed. [Online]. Available: https://christophm.github.io/interpretable-ml-book/, 2022.

[43] V. Viswan, L. Velayudhan, G. B. Williams, and B. Vázquez-Rodríguez, "Interpreting artificial intelligence models: A systematic review on the application of LIME and SHAP in Alzheimer's disease detection," Brain Inform., vol. 11, no. 1, p. 3, 2024.

[44] D. Berrar, "Cross-validation," in Encyclopedia of Bioinformatics and Computational Biology, S. Ranganathan, M. Gribskov, K. Nakai, and C. Schönbach, Eds. Academic Press, 2019, vol. 1, pp. 542–545.

[45] J. J. Eertink, B. Geurts, A. I. J. Arens, et al., "External validation of PET radiomic features in diffuse large B-cell lymphoma: A simulation study," EJNMMI Res., vol. 12, no. 1, p. 38, 2022.

[46] M. Drury, "Cross-validation for model evaluation," Towards Data Science, 2017. [Online]. Available: https://towardsdatascience.com/cross-validation-for-model-evaluation-30a06bd2b117

[47] T. Takada, M. Yamasaki, K. Kanemitsu, et al., "Internal–external cross-validation helped to evaluate the generalizability of prediction models," J. Clin. Epidemiol., vol. 136, pp. 179–187, 2021.

[48] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135–1144).

[49] Holzinger, A., Biemann, C., Pattichis, C. S., & Kell, D. B. (2017). What do we need to build explainable AI systems for the medical domain? Review. arXiv preprint arXiv:1712.09923