

# **CASO DI STUDIO ANNO 2024**

## **PREDIZIONE CANCRO ALLE OVAIE**

### **AUTORI**

Emanuele de Stena [matr.744397] [mail. [e.destena1@studenti.uniba.it](mailto:e.destena1@studenti.uniba.it)]

Federico Gemma [matr.741217] [mail. [f.gemma@studenti.uniba.it](mailto:f.gemma@studenti.uniba.it)]

Andrea Barbaro [matr.739797] [mail. [a.barbaro12@studenti.uniba.it](mailto:a.barbaro12@studenti.uniba.it)]

---

*Link Github per codice sorgente:*

<https://github.com/FeGem99/Caso-di-Studio-ICON.git>

## Sommario

<b>INTRODUZIONE .....</b>	3
<b>INFORMAZIONI TECNICHE SUL PROGETTO .....</b>	3
<b>ORGANIZZAZIONE DEL DATASET .....</b>	3
<b>ANALISI E PRE-ELABORAZIONE DEI DATI .....</b>	4
<b>OSSERVAZIONE GRAFICA DEI DATI .....</b>	6
<b>APPRENDIMENTO NON SUPERVISIONATO .....</b>	10
<i>Curve a gomito: .....</i>	10
<i>Valutazione qualità del cluster: .....</i>	14
<b>APPRENDIMENTO SUPERVISIONATO: .....</b>	15
<i>Cross Validation .....</i>	15
<i>KNeighbours Classifier.....</i>	16
<i>Curva di overfitting:.....</i>	18
<i>SVM.....</i>	18
<i>Curva di overfitting:.....</i>	20
<i>Logistic Regression.....</i>	20
<i>Curva di overfitting.....</i>	22
<i>Gradient Booster Classifier .....</i>	22
<i>Curva di overfitting:.....</i>	24
<i>Stratified K-Fold Validation:.....</i>	25
<b>RETE BAYESIANA: .....</b>	26
<i>Pre-processamento dei dati: .....</i>	26
<i>Creazione della rete:.....</i>	27
<i>Rappresentazione grafica della rete: .....</i>	28



UNIVERSITÀ  
DEGLI STUDI DI BARI  
ALDO MORO

## **INTRODUZIONE**

Il nostro progetto offre un supporto medico per la diagnosi del cancro alle ovaie, focalizzandosi sulla determinazione della natura maligna o benigna del tumore. L'obiettivo finale è evidenziare i fattori chiave che influenzano significativamente la diagnosi. Attraverso un'approfondita analisi statistica e l'applicazione di algoritmi avanzati di apprendimento automatico, sia supervisionato che non supervisionato, miriamo a individuare modelli di dati significativi.

# **INFORMAZIONI TECNICHE SUL PROGETTO**

Il progetto è stato realizzato con il linguaggio Python nella IDE Visual Studio Code.

Di seguito le librerie utilizzate:

- sklearn -- per gli algoritmi di apprendimento e la loro valutazione;
  - pandas, numpy e math -- per la manipolazione dei dati;
  - Seaborn -- per la rappresentazione grafica dei dati;
  - pgmpy -- per lavorare con modelli grafici.
  - networkx -- rappresentazione del modello bayesiano

## ORGANIZZAZIONE DEL DATASET

[Link: Predict Ovarian Cancer | Kaggle](#)

Il carcinoma ovarico è il sesto tumore più diagnosticato tra le donne, ed è il più grave a livello ginecologico, causando il 50% di morte. Nel 20 % dei casi in cui il tumore viene diagnosticato precocemente, la sopravvivenza a cinque anni aumenta notevolmente e questo rende particolarmente importante identificare dei marcatori della malattia nelle fasi iniziali. Lo scopo è quindi di individuare quali fattori possono favorire la diagnosi precoce per aumentare le probabilità di sopravvivenza.

Il dataset di riferimento è di 349 righe e 51 colonne:



## ANALISI E PRE-ELABORAZIONE DEI DATI

Dopo aver condotto un'analisi approfondita del dataset, è stato deciso di rinominare le colonne sostituendo gli acronimi medici con il relativo termine italiano. Successivamente, si è optato per ridimensionare il dataset eliminando le colonne ritenute poco utili per lo scopo principale.

```
queste sono le colonne nel nostro dataset
Index(['AFP', 'età', 'ALB', 'ALP', 'ALT', 'AST', 'BUN', 'calcio', 'CA125',
       'CA19-9', 'CA72-4', 'CEA', 'cloro', 'CO2CP', 'creatinina',
       'tipo_tumore', 'DBIL', 'num_eosinofili', 'perc_eosinofili', 'globulina',
       'ematocrito', 'HE4', 'HGB', 'K', 'num_linfociti', 'perc_linfociti',
       'MCH', 'MCV', 'Menopause', 'magnesio', 'rapp_eosinofili', 'PDW',
       'num_piastrine', 'num_globuli_rossi', 'RDW', 'TBIL', 'proteine_totali',
       'acido_urine'],
      dtype='object')
```

**numero righe: 349 numero colonne: 38**

Finita l'eliminazione di colonne superflue, ci si è concentrati sui valori nel dataset e sulla categorizzazione di essi. Nel dataset sono presenti valori continui, discreti e solo 3 campi di tipo Object (Figura 1).

(Figura 1)

AFP	object
età	int64
ALB	float64
ALP	float64
ALT	float64
AST	float64
BUN	float64
calcio	float64
CA125	object
CA19-9	object
CA72-4	float64
CEA	float64
cloro	float64
CO2CP	float64
creatinina	float64
tipo_tumore	int64
DBIL	float64
num_eosinofili	float64
perc_eosinofili	float64
globulina	float64
ematocrito	float64
HE4	float64
HGB	float64
K	float64
num_linfociti	float64
perc_linfociti	float64
MCH	float64
MCV	float64
Menopause	int64
magnesio	float64
rapp_eosinofili	float64
PDW	float64
num_piastrine	int64
num_globuli_rossi	float64
RDW	float64
TBIL	float64
proteine_totali	float64
acido_urine	float64

(Figura 2)

AFP	22
età	0
ALB	10
ALP	10
ALT	10
AST	10
BUN	0
calcio	0
CA125	17
CA19-9	24
CA72-4	240
CEA	22
cloro	0
CO2CP	1
creatinina	0
tipo_tumore	0
DBIL	10
num_eosinofili	0
perc_eosinofili	0
globulina	10
ematocrito	0
HE4	20
HGB	0
K	0
num_linfociti	0
perc_linfociti	0
MCH	0
MCV	0
Menopause	0
magnesio	0
rapp_eosinofili	91
PDW	2
num_piastrine	0
num_globuli_rossi	0
RDW	0
TBIL	10
proteine_totali	10
acido_urine	0

(Figura 3)

AFP	0
età	0
ALB	0
ALP	0
ALT	0
AST	0
BUN	0
calcio	0
CA125	0
CA19-9	0
CA72-4	0
CEA	0
cloro	0
CO2CP	0
creatinina	0
tipo_tumore	0
DBIL	0
num_eosinofili	0
perc_eosinofili	0
globulina	0
ematocrito	0
HE4	0
HGB	0
K	0
num_linfociti	0
perc_linfociti	0
MCH	0
MCV	0
Menopause	0
magnesio	0
rapp_eosinofili	0
PDW	0
num_piastrine	0
num_globuli_rossi	0
RDW	0
TBIL	0
proteine_totali	0
acido_urine	0

I suddetti valori di tipo Object si è provveduto a trasformarli in valori di tipo Float.

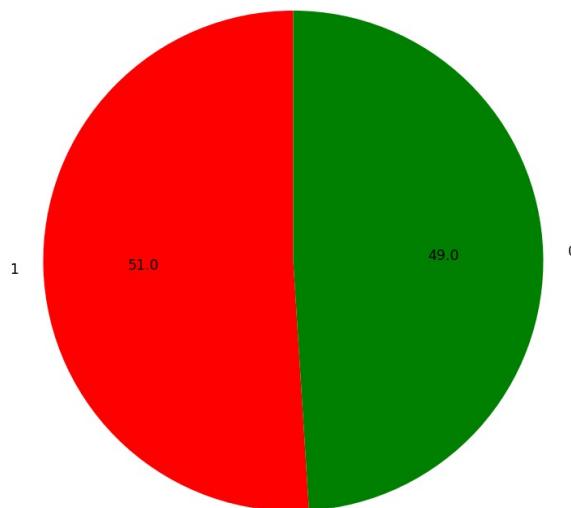
Inizialmente, è stata condotta un'analisi dei valori nulli per ciascuna colonna del dataset. Una volta identificata la presenza di tali valori ([Figura 2](#)), è stato adottato un approccio basato sull'imputazione utilizzando il KNN Imputer. Tale tecnica rappresenta un'applicazione specifica dell'algoritmo K-Nearest Neighbors, concepita per affrontare la problematica della mancanza di dati all'interno di un insieme di informazioni. Mediante il KNN Imputer, è stato possibile stimare e completare i valori mancanti basandosi sulla somiglianza dei casi circostanti utilizzando una metrica di distanza appropriata.

L'imputazione è il processo di stima o sostituzione di valori mancanti con valori stimati in base a informazioni disponibili. Per ogni istanza con un valore mancante, il KNN Imputer calcola le distanze rispetto a tutte le altre istanze nel dataset in base a una metrica di distanza specificata.

Effettuando nuovamente la verifica dei valori nulli, si nota come tutti i campi sono stati popolati ([Figura 3](#)).

Successivamente, è stato utilizzato un grafico a torta per comprendere la distribuzione del dataset in relazione alla variabile target "Tipo\_Tumore", con due valori distinti: 0, che indica un tumore benigno, e 1, un tumore maligno.

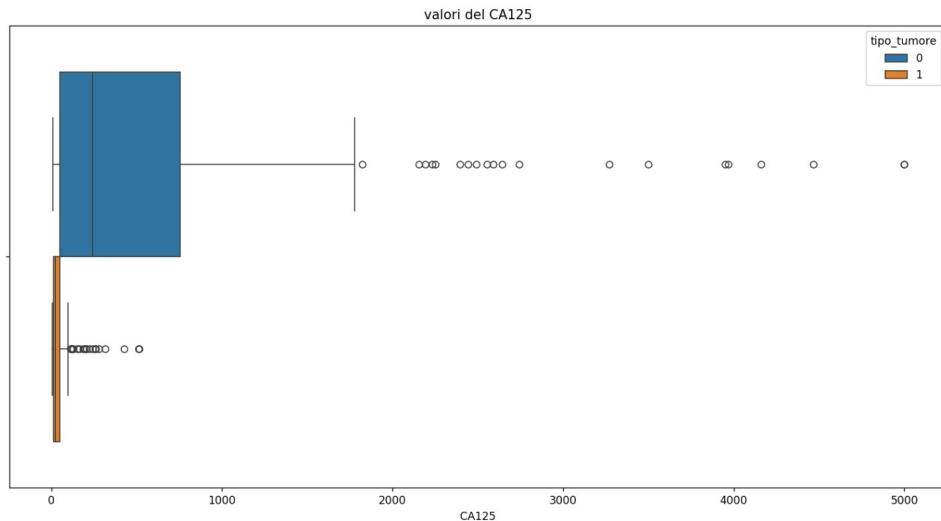
grafico delle distinzioni cancro maligno e benigno



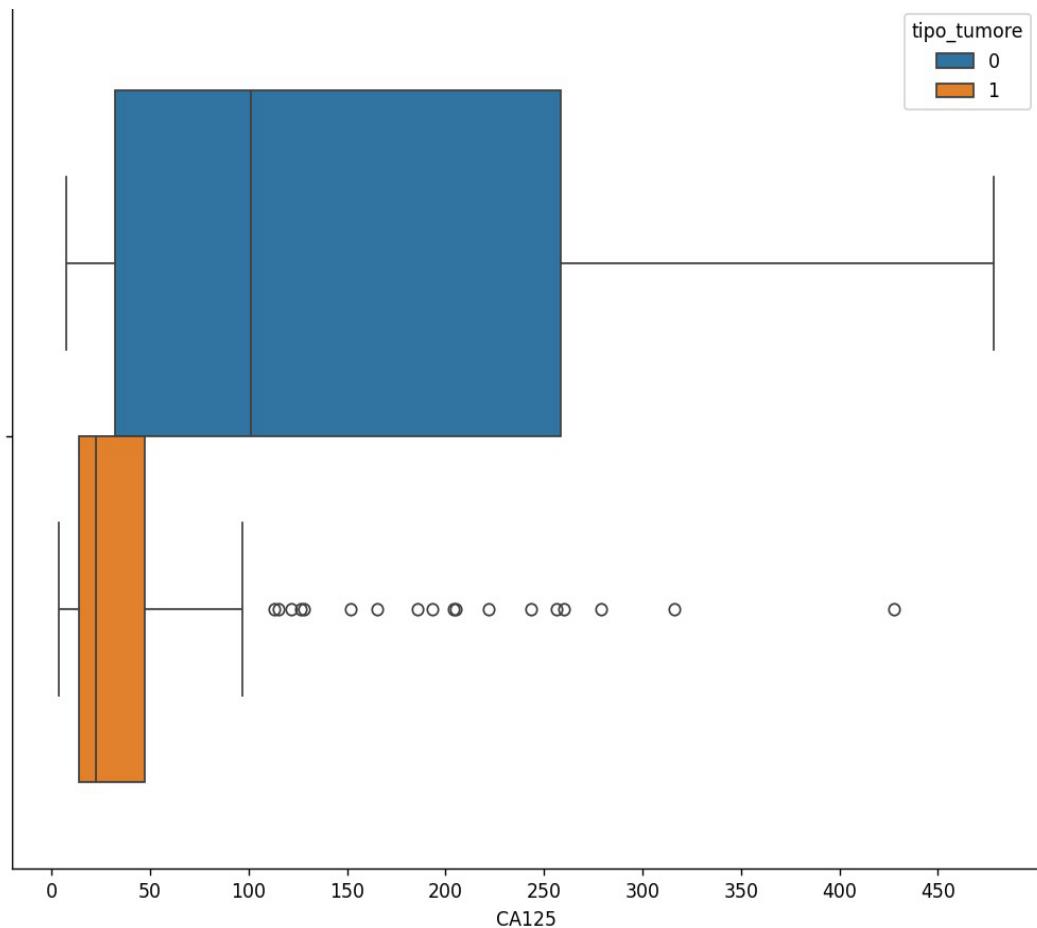
## OSSERVAZIONE GRAFICA DEI DATI

Svolgendo osservazioni grafiche basate sulla tabella che indica se il tumore è benigno o maligno, è emerso che, rispetto agli altri elementi nel dataset, il CA125, il Ca72-4 e il rapporto degli eosinofili hanno avuto un impatto significativo sulla diagnosi benigna o maligna.

Il grafico dell'antigene CA125 è mostrato di seguito:

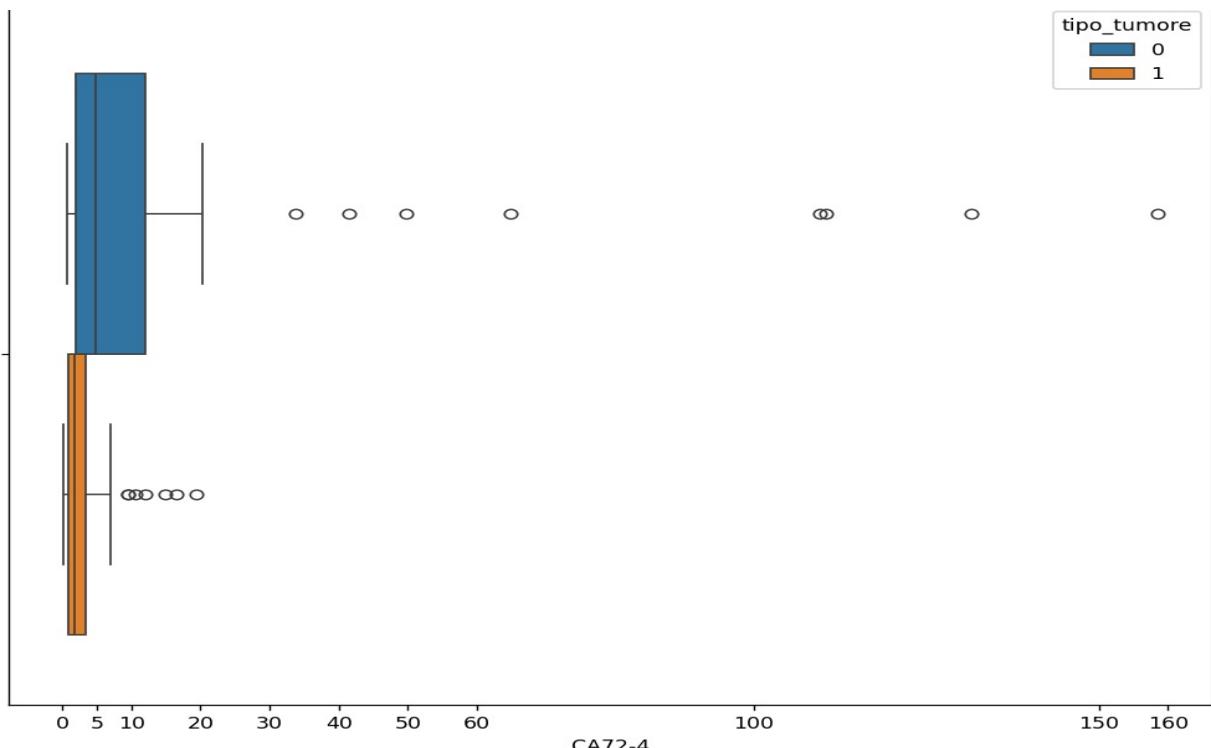


Analizzando più nel dettaglio i risultati di tale grafico risulta:



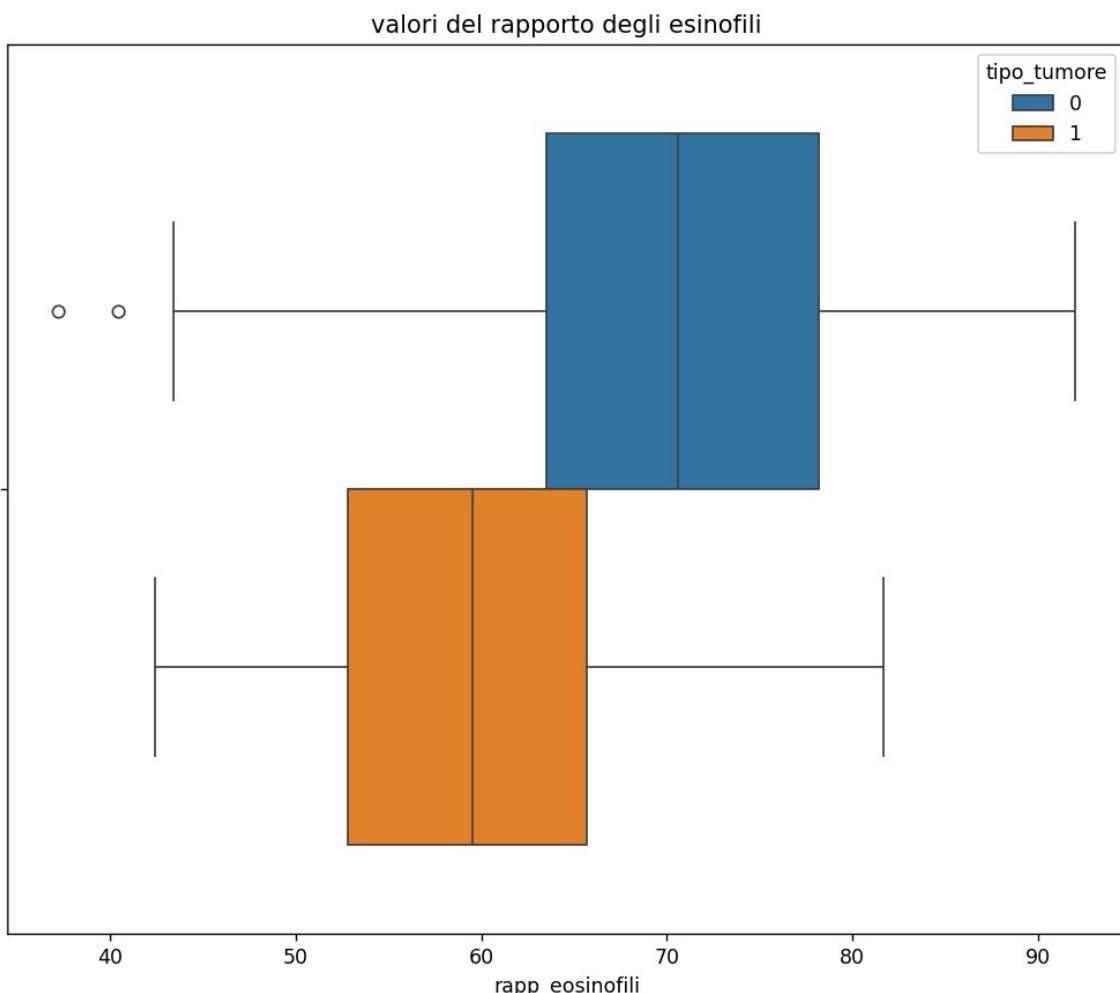
Il valore dell'antigene CA125 da 0 a circa 25 indica in genere tumori maligni, mentre i valori da 26 a 50 indicano tumori benigni o maligni. Invece, quando il valore è superiore a 50, il tumore è generalmente benigno. Il primo grafico mostra che ci sono molti casi rari (fino a 1000) in cui un valore di CA125, che normalmente indica una diagnosi benigna, è invece anormalmente maligno.

Si analizza ora il grafico relativo all'antigene carboidrato CA72-4:



L'analisi dell'incidenza del CA72-4 nella diagnosi del cancro è stata condotta utilizzando il seguente grafico. Si è osservato che se l'antigene assume un valore compreso tra 0 e 2, la diagnosi è maligna. Allo stesso modo, nell'intero intervallo da 0 a 10, si evidenzia la possibilità di tumori maligni. Al di fuori di questo intervallo, si sono verificati casi anomali con valori che raggiungono fino a 20. Per quanto concerne la certezza che il cancro sia benigno, si evince dall'intervallo compreso tra 2 e 15. Tuttavia, per quanto riguarda la probabilità che il cancro sia benigno, l'intervallo analizzato è tra 0 e 20. È importante notare che, anche in questo caso, si sono verificati casi anomali con valori che arrivano fino a 160.

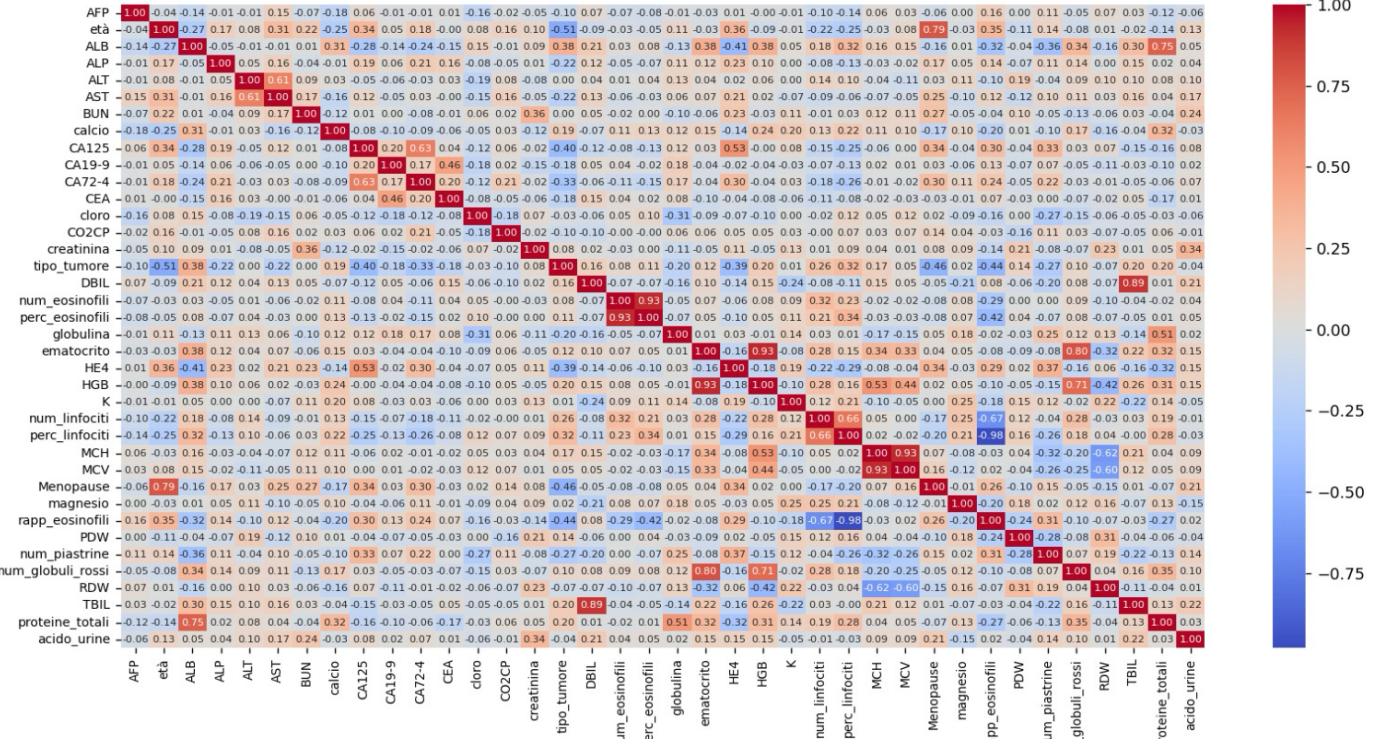
Si passa ora all'interpretazione del dato del rapporto degli eosinofili:



Questo grafico illustra i valori del rapporto degli eosinofili e il loro impatto sulla diagnosi di cancro. Dall'analisi del grafico emerge che la presenza di un cancro maligno è evidente nei valori compresi tra 53 e 67. Allo stesso modo, si osserva la probabilità di malignità nell'intero intervallo da 43 a 82. Non sono stati riscontrati casi anomali al di fuori di questo intervallo. Per quanto riguarda la probabilità che il cancro sia benigno, si deduce dall'intervallo compreso tra 62 e 78. Tuttavia, l'intervallo analitico da 42 a 95 rappresenta la probabilità che il cancro sia benigno. È da notare che in questo intervallo si riscontrano casi anomali con valori inferiori a quaranta.



Per concludere l'analisi grafica dei dati, è stata creata una **heatmap** che rappresenta la correlazione tra ogni elemento del dataset al fine di determinare quali attributi fossero più correlati tra loro.



## APPRENDIMENTO NON SUPERVISIONATO

È una categoria di algoritmi di machine learning, dove il modello deve cercare di trarre informazioni dai dati in input, senza prima essere etichettati. L'obiettivo principale è quello di addestrare il modello in modo da individuare strutture e relazioni tra i dati non etichettati. L'algoritmo cerca di estrarre informazioni in modo automatico, quindi "non supervisionato", individuando cluster di dati simili.

standardizzare e normalizzare i dati sono passaggi cruciali quando si lavora con algoritmi di machine learning, poiché aiutano a rendere uniformi le scale dei dati e facilitano la comparazione tra di essi. Una volta ottenuti due nuovi dataset attraverso le tecniche di standardizzazione e normalizzazione, è possibile applicare la tecnica del clustering. Questa tecnica consiste nel raggruppare i dati all'interno dello stesso cluster in modo che siano quanto più simili tra loro rispetto ai dati in altri cluster.

la strategia **mean** viene utilizzata come parametro nella creazione di un oggetto SimpleImputer. Il SimpleImputer è un trasformatore che sostituisce i valori mancanti con un valore specifico, e la strategia mean indica che si vuole sostituire i valori mancanti con la media dei valori presenti nella stessa colonna.

### *Curve a gomito:*

Per suddividere il dataset in cluster utilizzando la tecnica di **k-means**, è stata impiegata la metodologia della "curva a gomito" al fine di individuare il numero ottimale di cluster da utilizzare nei dati. Questo è stato realizzato mediante l'algoritmo di k-means, variando il parametro k da un minimo di 2 a un massimo di 10. I valori di k sono stati selezionati osservando i grafici delle curve a gomito, dai quali emerge che, aumentando il numero di cluster, il dataset si stabilizza.

Per ciascun valore di k, è stata calcolata l'inerzia o la somma delle distanze al quadrato. Maggiore è l'inerzia, migliore sarà il valore di k assegnato.

Servendosi del coefficiente di silhouette, si è valutata la qualità dei cluster ottenuti dall'algoritmo k-means. Il coefficiente può assumere valori nell'intervallo che va da 1 a -1. In particolare:

- il valore 1, indica che i cluster sono separati correttamente e chiaramente distinti
- valore 0, indica una sovrapposizione di cluster e quindi una distanza non significativa.
- Il valore -1 invece, indica che i punti sono stati assegnati in modo errato al cluster.

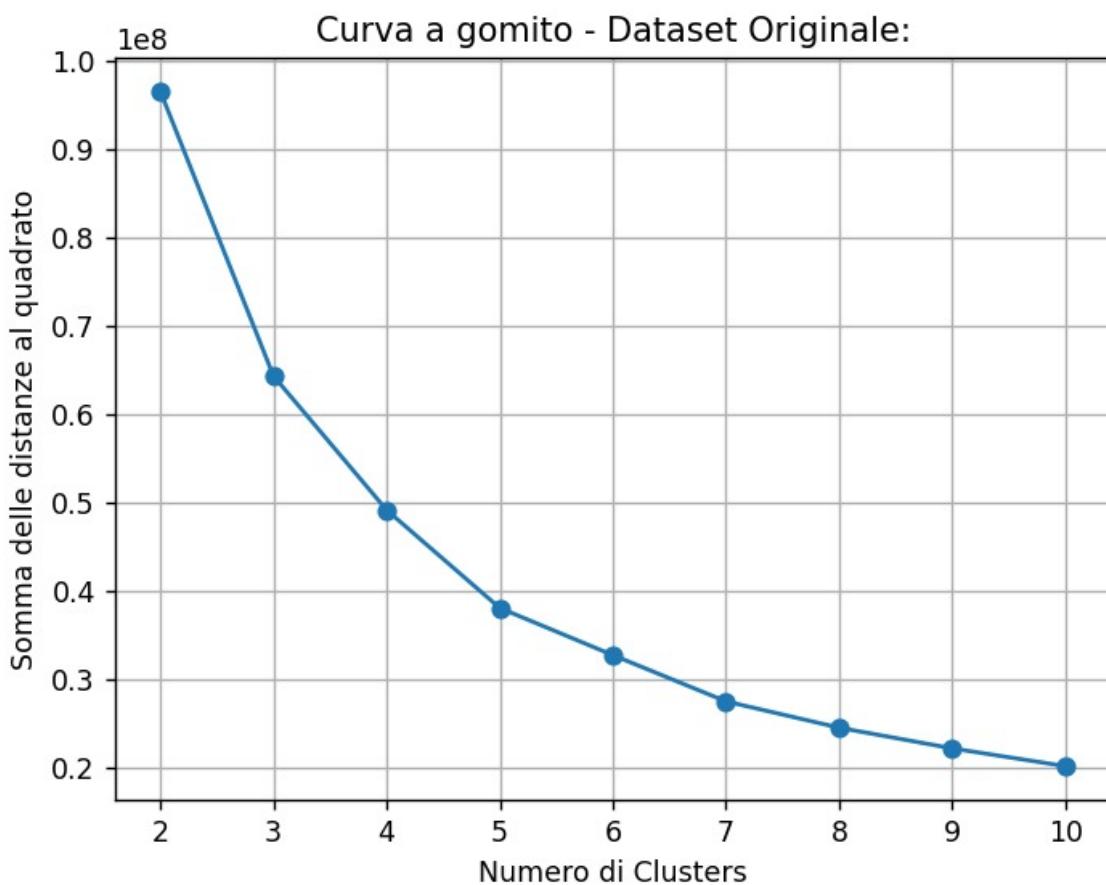
Il coefficiente di silhouette viene calcolato attraverso la formula:

**Score= (p-q) / max (p, q)** dove:

p indica la distanza media dei punti nel cluster più vicino (a cui il punto non appartiene)  
q indica la distanza media intra-cluster da tutti i punti del proprio cluster.

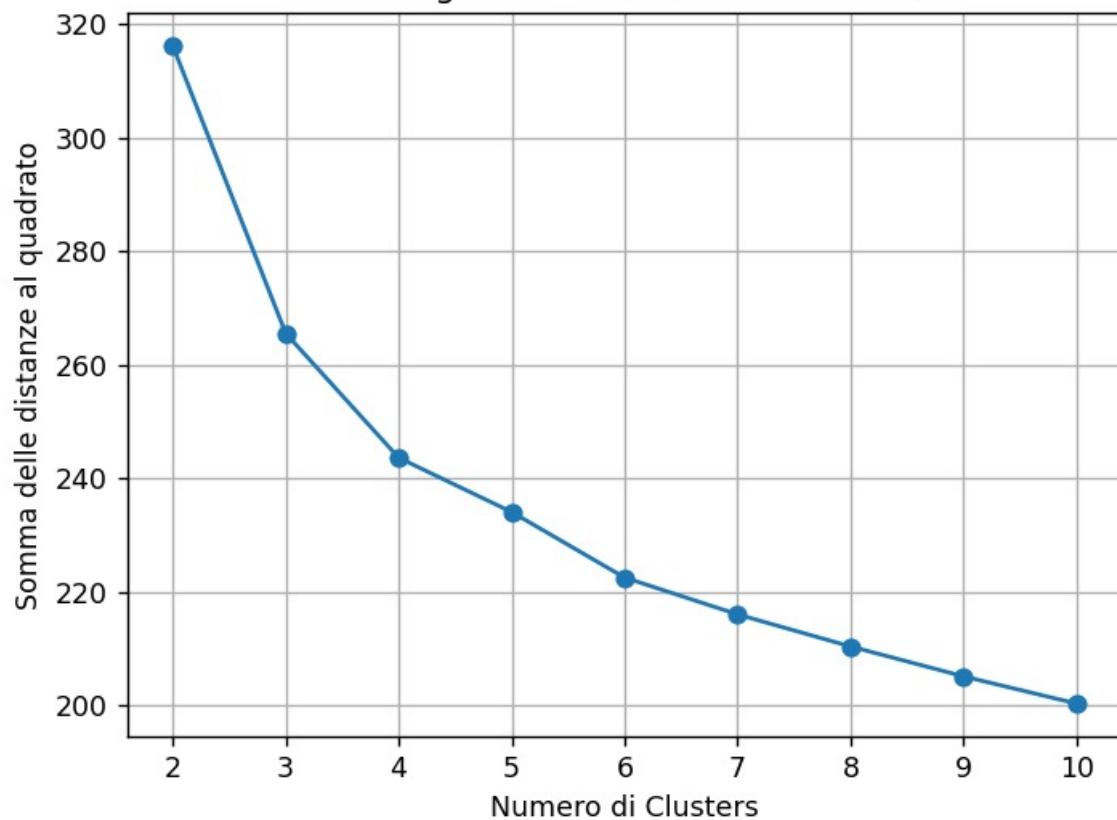
Ovviamente più è alto il valore di silhouette, migliore sarà la separazione e la distinzione dei cluster.

Di seguito vengono riportati i grafici ottenuti:



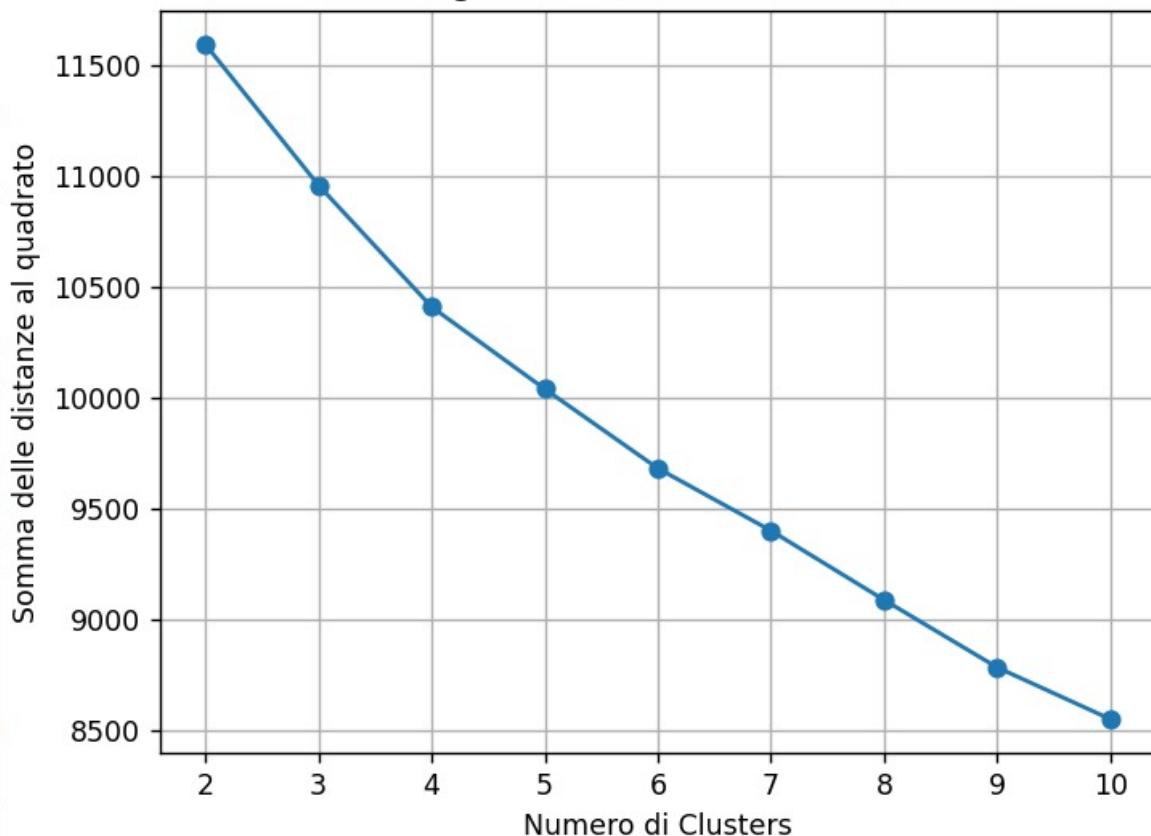
```
Con n_clusters=2, il valore di silhouette 0.8211209561920404
Con n_clusters=3, il valore di silhouette 0.786053236432206
Con n_clusters=4, il valore di silhouette 0.780157653826814
Con n_clusters=5, il valore di silhouette 0.715021366105295
Con n_clusters=6, il valore di silhouette 0.5342946810593641
Con n_clusters=7, il valore di silhouette 0.5231286869628646
Con n_clusters=8, il valore di silhouette 0.530864167191578
Con n_clusters=9, il valore di silhouette 0.5283834486122293
Con n_clusters=10, il valore di silhouette 0.5269859363962943
```

Curva a gomito - Dataset Normalizzato:



```
Con n_clusters=2, il valore di silhouette 0.2639535624623705
Con n_clusters=3, il valore di silhouette 0.25549204420578087
Con n_clusters=4, il valore di silhouette 0.2667716960915175
Con n_clusters=5, il valore di silhouette 0.15543202024341607
Con n_clusters=6, il valore di silhouette 0.1418514020395862
Con n_clusters=7, il valore di silhouette 0.1450426856069733
Con n_clusters=8, il valore di silhouette 0.14881854571877343
Con n_clusters=9, il valore di silhouette 0.13037390188636885
Con n_clusters=10, il valore di silhouette 0.12492697782987577
```

Curva a gomito - Dataset Standardizzato:



```
Con n_clusters=2, il valore di silhouette 0.16223609840000205
Con n_clusters=3, il valore di silhouette 0.13625065322156824
Con n_clusters=4, il valore di silhouette 0.08381097697268962
Con n_clusters=5, il valore di silhouette 0.0620239406793462
Con n_clusters=6, il valore di silhouette 0.03925530387093242
Con n_clusters=7, il valore di silhouette 0.052688560797330905
Con n_clusters=8, il valore di silhouette 0.04283458072437865
Con n_clusters=9, il valore di silhouette 0.050986218356188814
Con n_clusters=10, il valore di silhouette 0.04522564122796838
```

Il "gomito" nella curva a gomito rappresenta il punto in cui l'inerzia smette di diminuire drasticamente e inizia a stabilizzarsi. Questa flessione della curva è indicativa del numero ottimale di cluster da considerare, ovvero il valore di  $k$ . Oltre questo punto, l'aggiunta di ulteriori cluster non fornisce miglioramenti significativi al modello.

Si è osservato che i valori di  $k$  assumono un aumento più significativo nell'intervallo compreso tra 2 e 5. Pertanto, per confrontare questi risultati con le metriche di complessità, omogeneità e v-measure, tale intervallo è stato preso in considerazione.

## Valutazione qualità del cluster:

Valutazione standardizzato:

```
Con n_clusters=2:  
Omogeneità : 0.31204720717768053  
Completezza : 0.3445604090472601  
V_measure : 0.3274988309315097  
Con n_clusters=3:  
Omogeneità : 0.33478517243828276  
Completezza : 0.25652467969587617  
V_measure : 0.2904759960169721  
Con n_clusters=4:  
Omogeneità : 0.25441440716463465  
Completezza : 0.138980932528114  
V_measure : 0.179761923890325  
Con n_clusters=5:  
Omogeneità : 0.32376496863239235  
Completezza : 0.14462621218522395  
V_measure : 0.19993929420205536
```

Valutazione normalizzato:

```
Con n_clusters=2:  
Omogeneità : 0.9745921742404425  
Completezza : 0.9744536147936436  
V_measure : 0.9745228895918843  
Con n_clusters=3:  
Omogeneità : 0.7584256258172644  
Completezza : 0.4968450337034148  
V_measure : 0.6003804880846965  
Con n_clusters=4:  
Omogeneità : 1.0  
Completezza : 0.5655924228613445  
V_measure : 0.7225283089038503  
Con n_clusters=5:  
Omogeneità : 1.0  
Completezza : 0.4948711318982329  
V_measure : 0.6620920309964519
```

Valutazione dataset:

```
Con n_clusters=2:  
Omogeneità : 0.09121950045040488  
Completezza : 0.22081155970945532  
V_measure : 0.12910458439651437  
Con n_clusters=3:  
Omogeneità : 0.11862962898989422  
Completezza : 0.20892811005080175  
V_measure : 0.15133249028689316  
Con n_clusters=4:  
Omogeneità : 0.1581806551615859  
Completezza : 0.20653229184863775  
V_measure : 0.17915137616872406  
Con n_clusters=5:  
Omogeneità : 0.16937270246138592  
Completezza : 0.197699310049381  
V_measure : 0.18244303720558327
```

Si è ottenuto un punteggio migliore per la suddivisione dei cluster con  $k = 2$  dalle nostre valutazioni utilizzando il metodo del gomito. Tuttavia, nonostante ciò, si è scelto di suddividere il dataset in tre cluster come decisione di progetto. Questa scelta è stata presa perché optare per un numero di cluster pari a 2 significherebbe dividere il dataset semplicemente in casi benigni e maligni, il che risulterebbe in una suddivisione troppo ovvia. Invece, con  $k = 3$ , si evidenzia che all'interno del dataset esiste un pattern di dati che va oltre la semplice distinzione tra casi benigni e maligni, suggerendo la presenza di una struttura più complessa e rilevante rispetto alla variabile target.

Per quanto riguarda le metriche utilizzate per la valutazione della qualità del clustering, ci siamo basati su diversi parametri, tra cui completezza, v-measure e omogeneità.

L'omogeneità misura quanto ciascun cluster è composto da istanze di una sola classe, ossia se tutti i punti in un cluster appartengono alla stessa classe.

La completezza misura quanto tutte le istanze di una classe sono assegnate allo stesso cluster. Indica quindi se il clustering raccoglie tutte le istanze della stessa classe, assumendo valori tra 1 e 0. Più è vicino all'uno, più indica completezza.

La V-Measure è una metrica che combina sia l'omogeneità che la completezza in un'unica misura bilanciata. È particolarmente utile quando si devono tenere conto di entrambi gli aspetti contemporaneamente. Anche questo parametro può assumere valori tra 0 e 1, e più è vicino all'1, più indica equilibrio tra omogeneità e completezza.

Si è applicato il KMeans su tutti i dataset a nostra disposizione, assegnando a k il valore 3, come precedentemente specificato.

## APPRENDIMENTO SUPERVISIONATO:

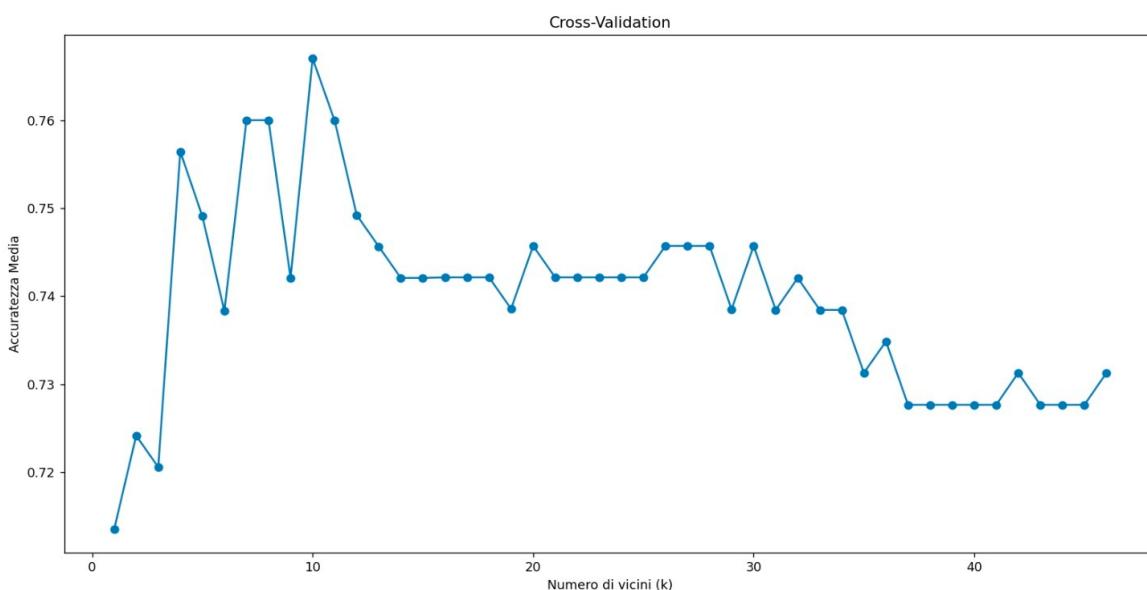
È una categoria di algoritmi che fa parte della famiglia dell'apprendimento automatico, in cui un modello viene addestrato secondo un insieme di dati in input etichettati. Durante la fase di addestramento, il modello riceve dei dati in input insieme alle corrispondenti etichette. Lo scopo del modello è quello di mappare il flusso di dati in ingresso correttamente, per permettere di effettuare delle previsioni accurate su nuovi tipi di dati non visti.

La predizione dovrà essere mirata sulla possibilità che il tumore sia di natura benigna o maligna. Per prima cosa, si è suddiviso il dataset in Training set e Test set. Sono entrambi dei sottoinsiemi del dataset principale, in particolare il Training set verrà utilizzato per addestrare il modello, per creare relazioni e pattern tra i dati in ingresso e le rispettive etichette, cercando di generalizzare le caratteristiche per fare previsioni accurate. Invece il Test set, svolge un compito di verifica dell'apprendimento del modello nella fase precedente. Utilizza un set di dati che non è stato utilizzato nella fase di addestramento, e serve per valutare quanto il modello abbia imparato a riconoscere correttamente i nuovi dati. Questo step è fondamentale, per evitare il fenomeno dell'over-fitting, ovvero quando il modello memorizza solamente le peculiarità di quel particolare training set, senza sforzarsi ad apprendere di generalità.

### Cross Validation

È una tecnica utilizzata nell'apprendimento supervisionato per valutare le prestazioni di un modello su dati diversi da quelli utilizzati durante il suo addestramento. Fornisce stime più affidabili riducendo la casualità di suddivisione dei dati.

Si è applicata la cross-validation su un range da 1 a 47 al fine di determinare il valore ottimale da assegnare a k. Dal grafico ottenuto, emerge che la migliore accuratezza viene restituita con k = 10. Il valore massimo del range è 47, poiché corrisponde al numero di colonne presenti nel dataset.



Come si vede dal grafico, il k selezionato risulta essere il migliore.

## KNeighbours Classifier

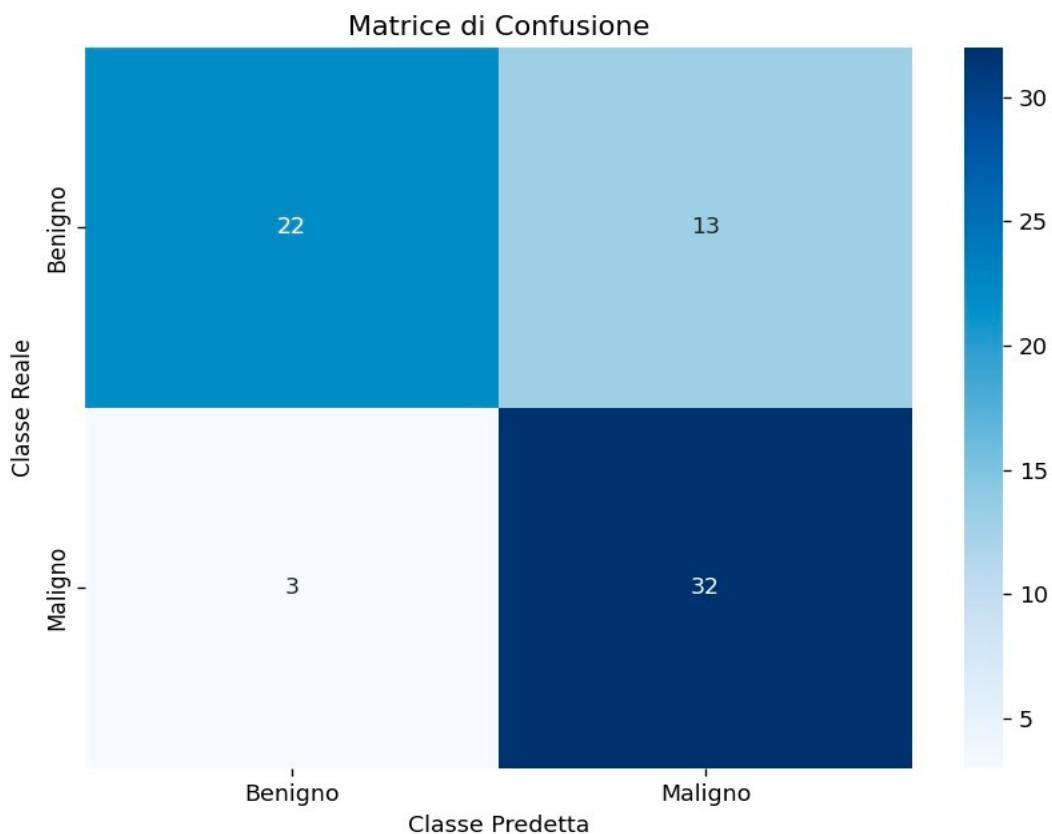
L'algoritmo K-Nearest neighbors (K-NN) è basato sul concetto che oggetti simili, si trovano vicini nello spazio. Per determinare l'intorno da considerare per la classificazione di un nuovo oggetto, si utilizza K, un valore intero e positivo. Lo spazio viene suddiviso in base alle posizioni e alle caratteristiche degli oggetti di apprendimento.

In questa fase del progetto si è scelto di utilizzare il dataset standardizzato, poiché maggiormente compatibile per gli algoritmi basati sulla distanza. Si è applicato il metodo `KNeighborsClassifier()` assegnando a k il valore scelto precedentemente:

Il livello di accuratezza con il quale il modello ha predetto correttamente i casi di tumore è stato:

Accuratezza: 0.7714

La seguente tabella aiuta a visualizzare i falsi positivi e i falsi negativi, dati utili per stimare la precisione del modello:



Si è effettuata una valutazione complessiva del classificatore, per verificare il livello di precision, recall e F1:

Classification Report per K-Nearest Neighbors (k=10):				
	precision	recall	f1-score	support
Benigno	0.88	0.63	0.73	35
Maligno	0.71	0.91	0.80	35
accuracy			0.77	70
macro avg	0.80	0.77	0.77	70
weighted avg	0.80	0.77	0.77	70

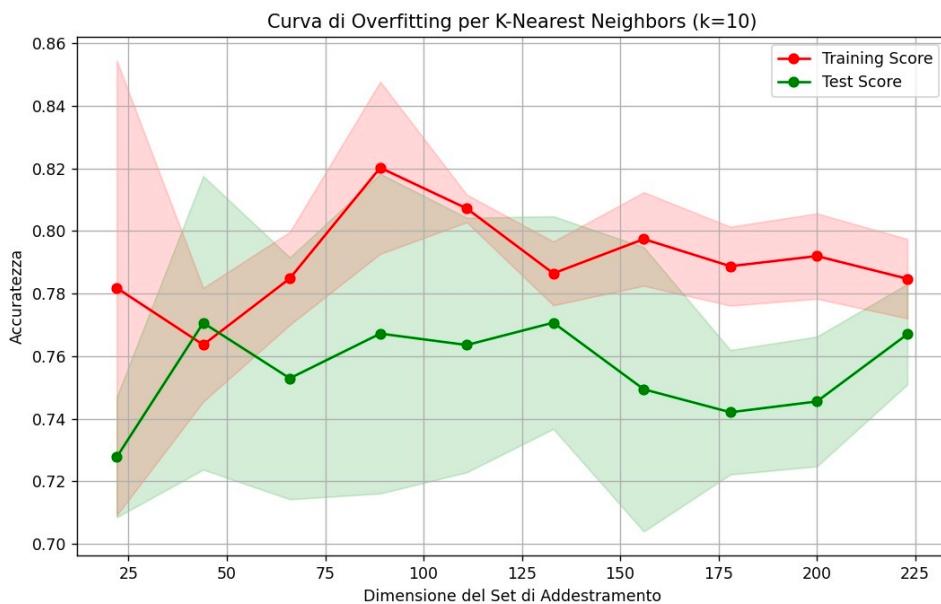
Questi risultati sono relativi all'utilizzo del KNeighbors con il valore di k assegnato a seguito della cross-validation.

Il seguente grafico mostra come l'accuratezza del Training Set e del Test Set, sono molto simili.



### Curva di overfitting:

La curva di overfitting mostra come le prestazioni del nostro modello, variano al variare del set di addestramento. Si è somministrato il set di addestramento in maniera progressiva, prima analizzando solo il 10%, poi il 20%, poi il 30% e così fino alla totalità set di addestramento. Di seguito:



Si osserva come le linee del training set e del test set tendano a convergere all'aumentare della somministrazione del test di addestramento, implicando la stabilità delle performance del modello. I riempimenti blu e arancio indicano la variazione standard dell'accuratezza su diverse ripetizioni della cross-validation. Inizialmente si osserva una certa instabilità del modello (tra il 10% e il 20% del training set), che si stabilizza verso la fine del processo di addestramento.

### SVM

L'algoritmo di classificazione SVM (Support Vector Machine), è un algoritmo di apprendimento automatico, utilizzato principalmente per compiti di classificazione e regressione. Facendo parte della branca di apprendimento automatico, ha necessità di suddividere anch'esso i dati in Training Set e Test set. L'obiettivo principale di un SVM è trovare l'iperpiano ottimale che separa i dati di addestramento in diverse classi. L'iperpiano è scelto in modo che la distanza tra i punti più vicini delle classi (chiamati vettori di supporto) sia massimizzata. Questa distanza è chiamata margine, e l'iperpiano che massimizza il margine è detto "iperpiano di massimo margine".

È stato utilizzato il metodo SVC(), che prende in input il kernel "rbf", uno dei più utilizzati in questa tipologia di algoritmo poiché gestisce in modo efficiente la complessità dei dati non lineari, mappando i dati in uno spazio di dimensioni superiori. È stata definita una

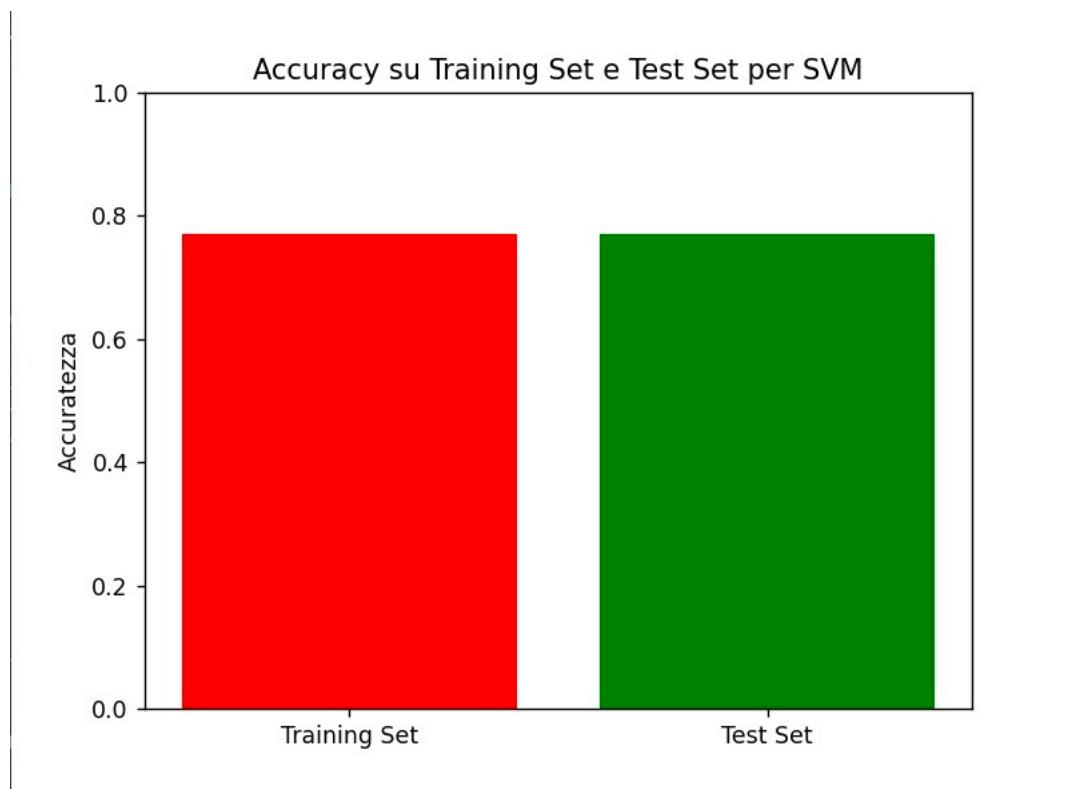
griglia di parametri ottimali inserendo una serie di valori per "C" e per "gamma". In particolare, per C è stato scelto un range da 0.1 a 100, mentre per gamma sono stati considerati i valori "Scale", "Auto", e un intervallo da 0.001 a 10. Attraverso l'oggetto GridSearchCV\_cv(), sono state provate tutte le combinazioni di tali valori, restituendoci la migliore combinazione di valori, che sono "C"=100 e "gamma"=0.1.

Di seguito la valutazione dei risultati ottenuti.

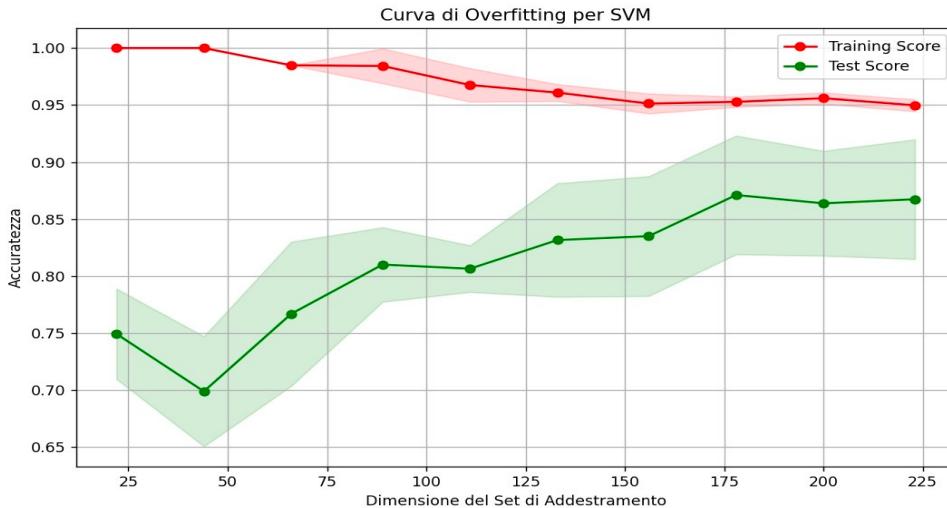
Questi sono i valori di Accuratezza, Precision, Recall e F1 ottenuti valutando questo metodo:

Report di Classificazione:				
	precision	recall	f1-score	support
0	0.85	0.66	0.74	35
1	0.72	0.89	0.79	35
accuracy			0.77	70
macro avg	0.79	0.77	0.77	70
weighted avg	0.79	0.77	0.77	70

Si può notare come l'accuratezza del training set e del test set siano molto simili alla totalità del set di addestramento.



### *Curva di overfitting:*



Inizialmente, le curve sembrano essere molto distanti tra loro, ma man mano che si procede con la somministrazione del set di addestramento, il modello tende ad avvicinare le curve. Si nota come, alla somministrazione della totalità del set di addestramento, il modello tende a stabilizzarsi. Questo fenomeno è tipico durante il processo di addestramento di modelli di machine learning, in cui l'errore di addestramento e di validazione tende a convergere man mano che il modello viene esposto a più dati e apprende dalle variazioni nel set di addestramento.

### *Logistic Regression*

La regressione logistica (Logistic Regression) è un modello di regressione utilizzato per problemi di classificazione binaria, ovvero quando la variabile dipendente è di tipo binario. Nonostante il nome contenga la parola "regressione", la regressione logistica è un algoritmo di classificazione e non di regressione nel senso tradizionale del termine. Infatti, il suo scopo principale è quello di stimare la probabilità che una determinata istanza appartenga a una delle due classi binarie.

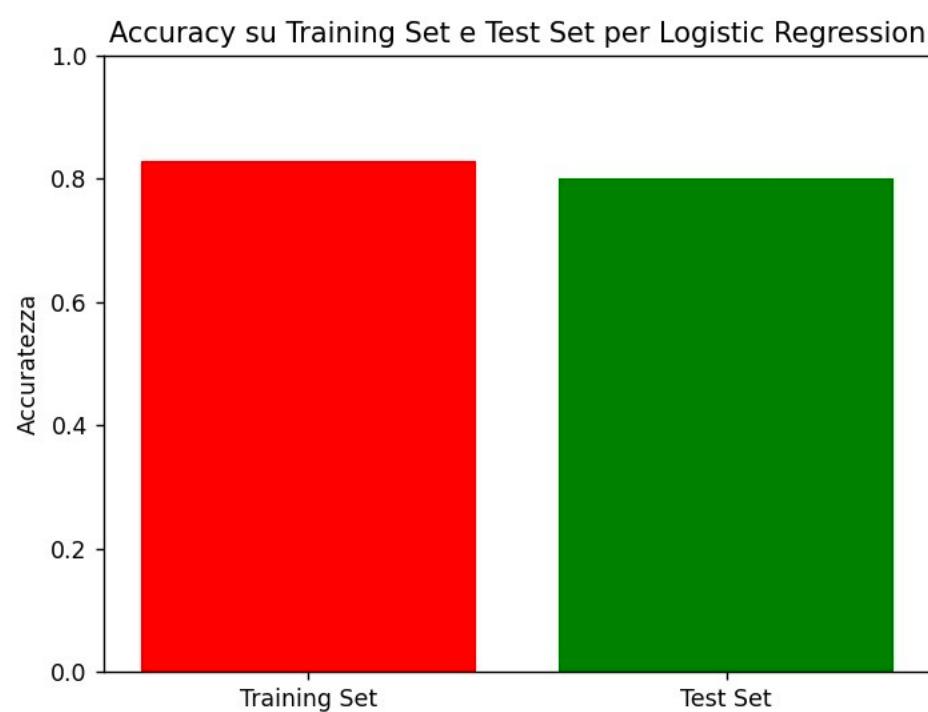
Utilizzando questo modello, si è ottenuta la seguente valutazione:

- Accuratezza:
- **Accuratezza del modello di logistic regression: 0.8000**



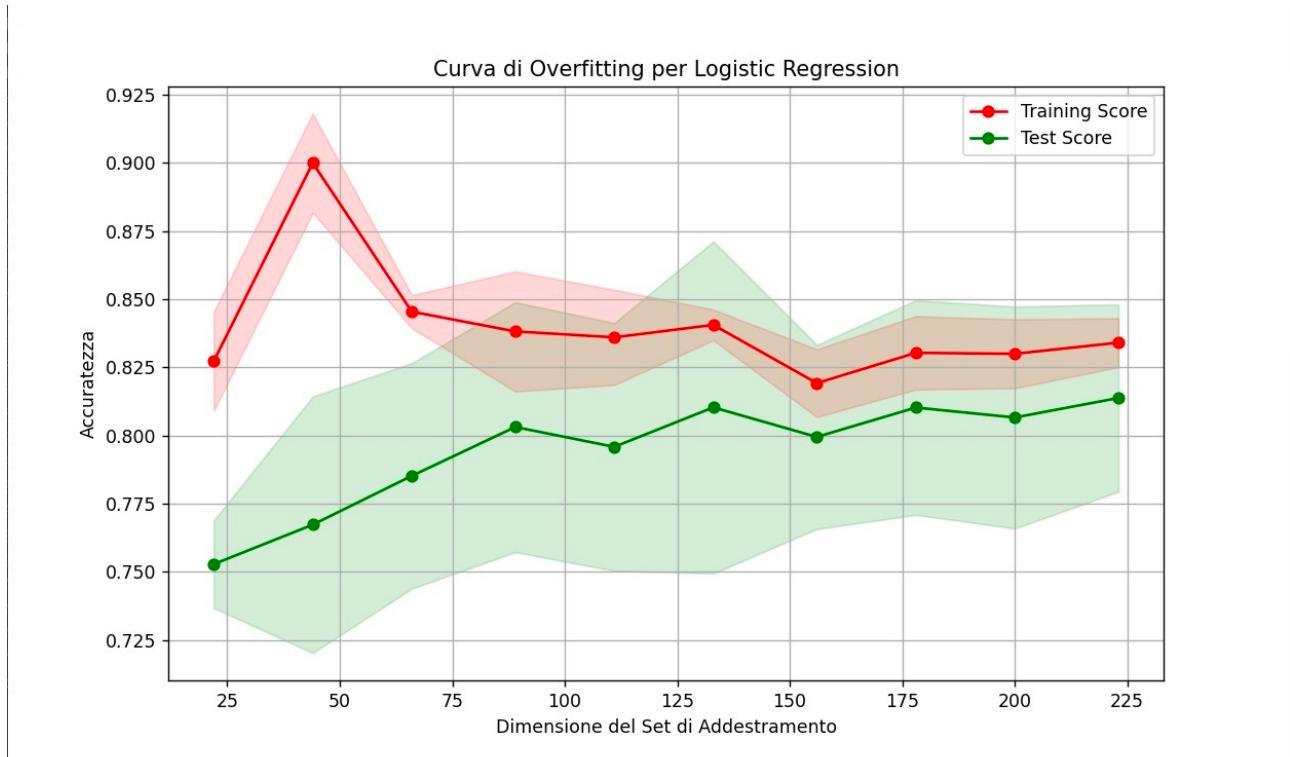
#### Classification Report per Logistic Regression:

	precision	recall	f1-score	support
Benigno	0.84	0.74	0.79	35
Maligno	0.77	0.86	0.81	35
accuracy			0.80	70
macro avg	0.80	0.80	0.80	70
weighted avg	0.80	0.80	0.80	70



Anche in questo caso si nota come l'accuratezza del training set e del test set, sono simili alla totalità del set di addestramento.

### *Curva di overfitting:*



Dal grafico si può evincere che in un primo momento la distanza tra il training set e il test set sia ampia; tuttavia, questa distanza tende a ridursi man mano che viene somministrato il set di addestramento. Alla somministrazione della totalità del set di addestramento, si osserva che il modello presenta stabilità, indicando che l'errore di addestramento e di validazione tende a convergere e il modello ha imparato a generalizzare bene sui dati.

### *Gradient Booster Classifier*

Il Gradient Boosting Classifier è un algoritmo di apprendimento automatico basati sulla tecnica di boosting e che utilizza il gradiente della funzione di perdita durante il processo di addestramento. Un classificatore di Gradient Boosting sfrutta il concetto di boosting per migliorare gradualmente la precisione del modello. Il processo di addestramento avviene in modo iterativo, aggiungendo uno alla volta i modelli deboli alla combinazione esistente. Ciascun modello successivo è addestrato per correggere gli errori residui commessi dai modelli precedenti.

Per determinare i valori da assegnare ai parametri del metodo `GradientBoosterClassifier()`, si è scelto di utilizzare anche in questo caso l'algoritmo `GridSearchCV()`, impostando come range di valori per il parametro “`n_estimators`” i valori 50, 100 e 150, per il parametro “`learning_rate`” i valori 0.01, 0.1 e 0.2 e infine per il parametro

“max\_depth” i valori 3, 5 e 7. Questi valori sono stati scelti a seguito di ricerche in modo da mantenere stabile il modello.

I migliori valori trovati dall’algoritmo sono: “n\_estimators” il valore intermedio 100 perché:

- inserendo un numero di estimatori alto, aumenta la complessità del modello, ma se si superasse un determinato valore, potrebbe portare solo rendimenti decrescenti.
- Inserendo un numero di estimatori basso, si ha una sotto stimazione del modello.

Anche per il parametro learning\_rate, che scala la contribuzione di ciascun albero, abbiamo assegnato un valore intermedio di 0.01, dato che:

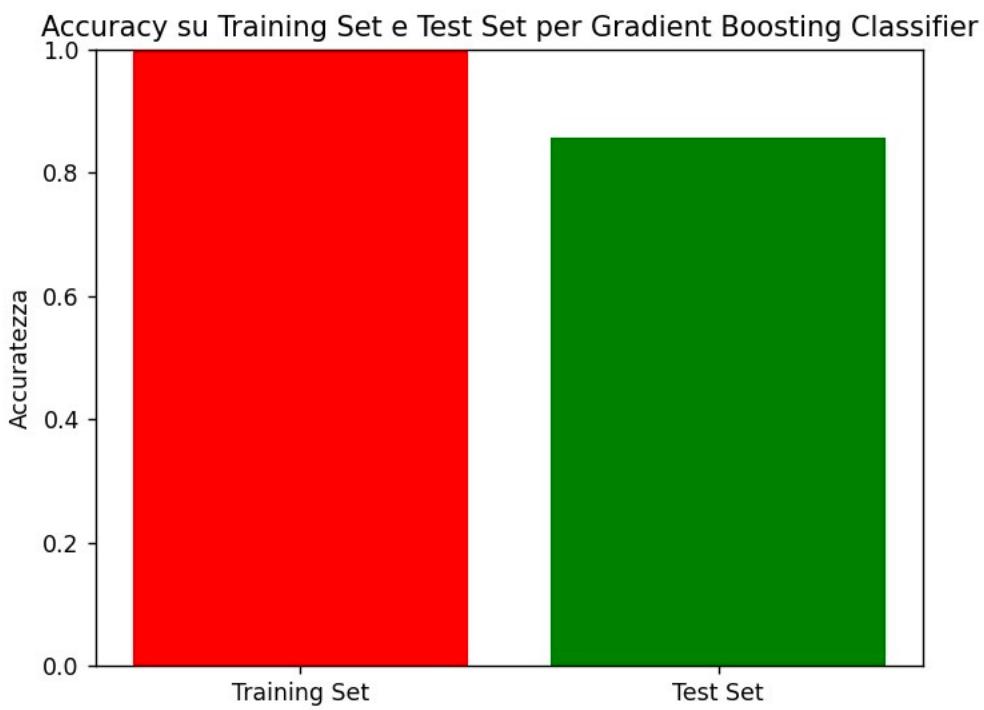
- Valori troppo alti possono portare a Overfitting.
- Valori troppo bassi rendono il modello più robusto, ma implica che bisogna aumentare il numero di alberi (n\_estimators) per raggiungere buone prestazioni.

Infine, anche il parametro max\_depth, che determina la profondità di ogni albero decisionale nell’algoritmo di boosting, abbiamo assegnato il valore di 3, dato che:

- Un valore troppo alto di profondità degli alberi potrebbe generare un’enorme complessità di calcolo, limitandone le prestazioni;
- Un valore troppo basso rende gli alberi troppo semplici e meno profondi, influenzando negativamente i risultati.

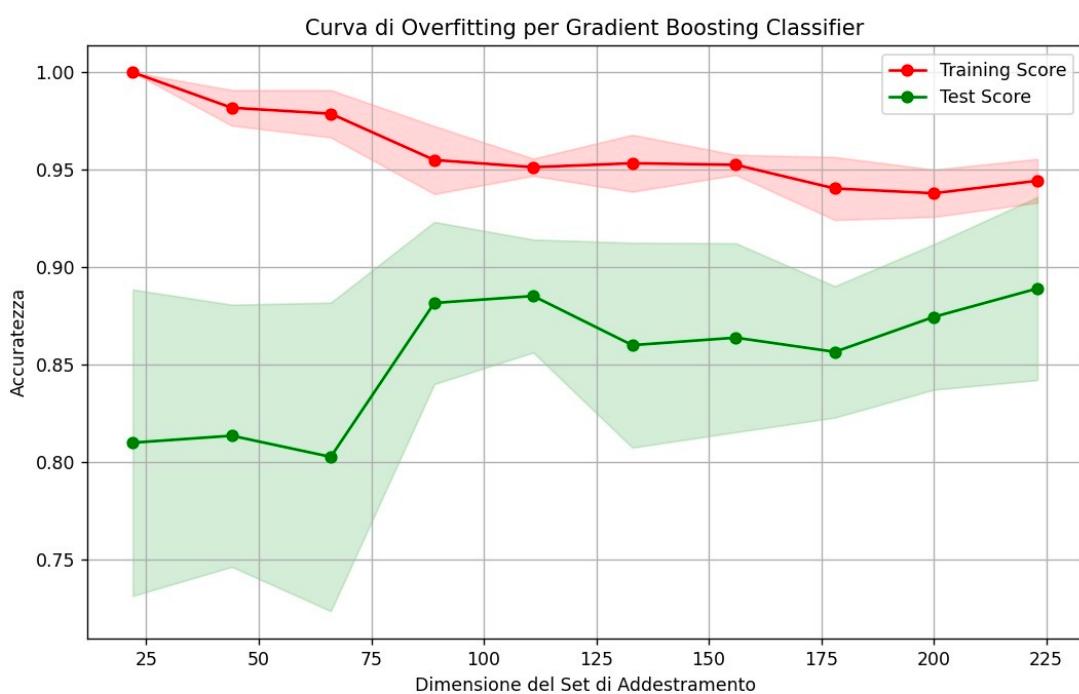
Di seguito si riportano le valutazioni effettuate su questo modello:

Report di Classificazione:				
	precision	recall	f1-score	support
0	0.90	0.77	0.83	35
1	0.80	0.91	0.85	35
accuracy			0.84	70
macro avg	0.85	0.84	0.84	70
weighted avg	0.85	0.84	0.84	70



In questo caso, l'accuratezza risulta essere 1 nel training set a differenza del test set, che è più bassa.

### *Curva di overfitting:*



In questo grafico, si nota che il training set ha un valore costante pari a 1 per l'intera durata del processo di addestramento, indicando un'accuratezza del 100% sui dati di addestramento. Nel frattempo, il test set mostra un andamento iniziale caratterizzato da una flessione, ma successivamente cresce, raggiungendo un buon valore di accuratezza verso la fine della somministrazione del set di addestramento. Questo indica che il modello ha inizialmente faticato a generalizzare sui dati di test, ma con il passare del tempo ha imparato a fare previsioni più accurate.

### *Stratified K-Fold Validation:*

La Stratified K-Fold Cross-Validation viene spesso utilizzata per valutare le prestazioni di un modello considerando diverse suddivisioni (fold) del dataset. Si è scelto il valore di suddivisioni pari a 3, poiché la divisione del dataset in 3 cluster, come nell'apprendimento non supervisionato, risulta essere la miglior suddivisione.

Si valutano i risultati dei migliori 4 classificatori sulla base delle metriche di precision, recall e F1.

K-Neighbors:

Classification Report durante la stratified K-Fold Cross-Validation:				
	precision	recall	f1-score	support
Benigno	0.82	0.59	0.68	136
Maligno	0.69	0.87	0.77	143
accuracy			0.73	279
macro avg	0.75	0.73	0.73	279
weighted avg	0.75	0.73	0.73	279

SVM:

Classification Report durante la Stratified K-Fold Cross-Validation per SVM:				
	precision	recall	f1-score	support
Benigno	0.84	0.87	0.90	133
Maligno	0.80	0.84	0.96	143
accuracy			0.88	279
macro avg	0.90	0.88	0.88	279
weighted avg	0.80	0.89	0.88	279

Logistic Regression:

Classification Report during Stratified K-Fold Cross-Validation for Logistic Regression:				
	precision	recall	f1-score	support
Benigno	0.82	0.75	0.78	136
Maligno	0.78	0.85	0.81	143
accuracy			0.80	279
macro avg	0.80	0.80	0.80	279
weighted avg	0.80	0.80	0.80	279

Gradient Booster Classifier:

Classification Report during Stratified K-Fold Cross-Validation for Gradient Boosting Classifier:				
	precision	recall	f1-score	support
Benigno	0.96	0.80	0.87	136
Maligno	0.84	0.97	0.90	143
accuracy			0.89	279
macro avg	0.90	0.88	0.88	279
weighted avg	0.89	0.89	0.88	279

Valutazioni finali dei 4 algoritmi:

Analizzando i valori ottenuti dai quattro modelli sopra citati e confrontando le diverse curve di overfitting ottenute per ciascun modello, è stato concluso che il modello più stabile rispetto agli altri tre è sicuramente la regressione logistica.

Tuttavia, confrontando i valori dei report di accuratezza, precisione, recall e f1 per ogni algoritmo, sia nella cross-validation che nella stratified cross-validation, il modello che ha restituito le predizioni più accurate è stato il Gradient Boosting Classifier. Questo suggerisce che, nonostante la regressione logistica sia stata più stabile rispetto agli altri modelli in termini di overfitting, il Gradient Boosting Classifier ha dimostrato di avere prestazioni migliori in termini di predizione accurata delle classi.

## RETE BAYESIANA:

È stata implementata una rete bayesiana per capire, date tutte le caratteristiche (di quelle specificate nel dataset) di un soggetto, quanto questo abbia la probabilità di un tumore benigno o maligno.

Una rete bayesiana è un grafico aciclico diretto, dove:

- i nodi, rappresentano le variabili
- gli archi rappresentano le relazioni di dipendenza statistica tra le variabili e le distribuzioni locali di probabilità dei nodi figlio rispetto ai valori dei nodi genitori.

## Pre-processamento dei dati:

È stato necessario trasformare tutti i valori float in Int, per facilitare il processamento dei dati e la costruzione della rete stessa. Infatti, l'algoritmo supporta pienamente solo questo tipo di valori.

Predisposizione della struttura:

Utilizzando il metodo HillClimbSearch() secondo il metodo di scoring fornito, abbiamo stimato la struttura DAG (Grafo Aciclico Diretto), che ha restituito punteggio ottimale. Nel caso in questione si è calcolato il punteggio che misura quanto una data variabile è "influenzata" da una data lista di potenziali genitori, attraverso il metodo K2Score(), che utilizza la distribuzione di Dirichlet con iperparametri impostati a 1.

Dopo aver definito la struttura della rete bayesiana, abbiamo proceduto con un'analisi dettagliata della sua configurazione. Sono stati stampati i nodi e gli archi presenti nella rete, fornendo così una panoramica completa delle variabili coinvolte e delle loro relazioni.

```
Nodi nella rete bayesiana: tipo_tumore, Menopause, PDW, num_linfociti, magnesio, ALP, HE4, CA19-9, ALT, creatinina, BUN, calcio, K, cloro, proteine_totali, ALB, globulina, CA125, acido_urine, CA72-4, CEA, DBIL, TBIL, perc_linfociti, rapp_eosinofili, MCH, MCV, età, num_piastrine, HGB, num_globuli_rossi
```

```
Archi nella rete bayesiana: ('tipo_tumore', 'Menopause'), ('tipo_tumore', 'PDW'), ('tipo_tumore', 'num_linfociti'), ('tipo_tumore', 'magnesio'), ('Menopause', 'età'), ('num_linfociti', 'perc_linfociti'), ('num_linfociti', 'creatinina'), ('ALP', 'HE4'), ('HE4', 'CA19-9'), ('CA19-9', 'CA125'), ('CA19-9', 'HE4'), ('ALT', 'creatinina'), ('creatinina', 'ALP'), ('BUN', 'ALP'), ('BUN', 'Menopause'), ('calcio', 'K'), ('calcio', 'cloro'), ('calcio', 'proteine_totali'), ('calcio', 'magnesio'), ('calcio', 'num_linfociti'), ('calcio', 'BUN'), ('calcio', 'ALB'), ('calcio', 'globulina'), ('proteine_totali', 'acido_urine'), ('CA125', 'acido_urine'), ('CA125', 'PDW'), ('CA125', 'proteine_totali'), ('acido_urine', 'età'), ('acido_urine', 'rapp_eosinofili'), ('acido_urine', 'perc_linfociti'), ('CA72-4', 'tipo_tumore'), ('CEA', 'CA19-9'), ('DBIL', 'TBIL'), ('perc_linfociti', 'rapp_eosinofili'), ('rapp_eosinofili', 'num_piastrine'), ('MCH', 'MCV'), ('num_piastrine', 'TBIL'), ('num_piastrine', 'MCV'), ('num_piastrine', 'HGB'), ('num_piastrine', 'K'), ('num_piastrine', 'cloro'), ('num_piastrine', 'ALB'), ('num_piastrine', 'globulina'), ('num_globuli_rossi', 'HGB')
```

Sono stati valutati i parametri di accuratezza, precision, recall e F1 della rete bayesiana, e si sono ottenuti i seguenti valori:

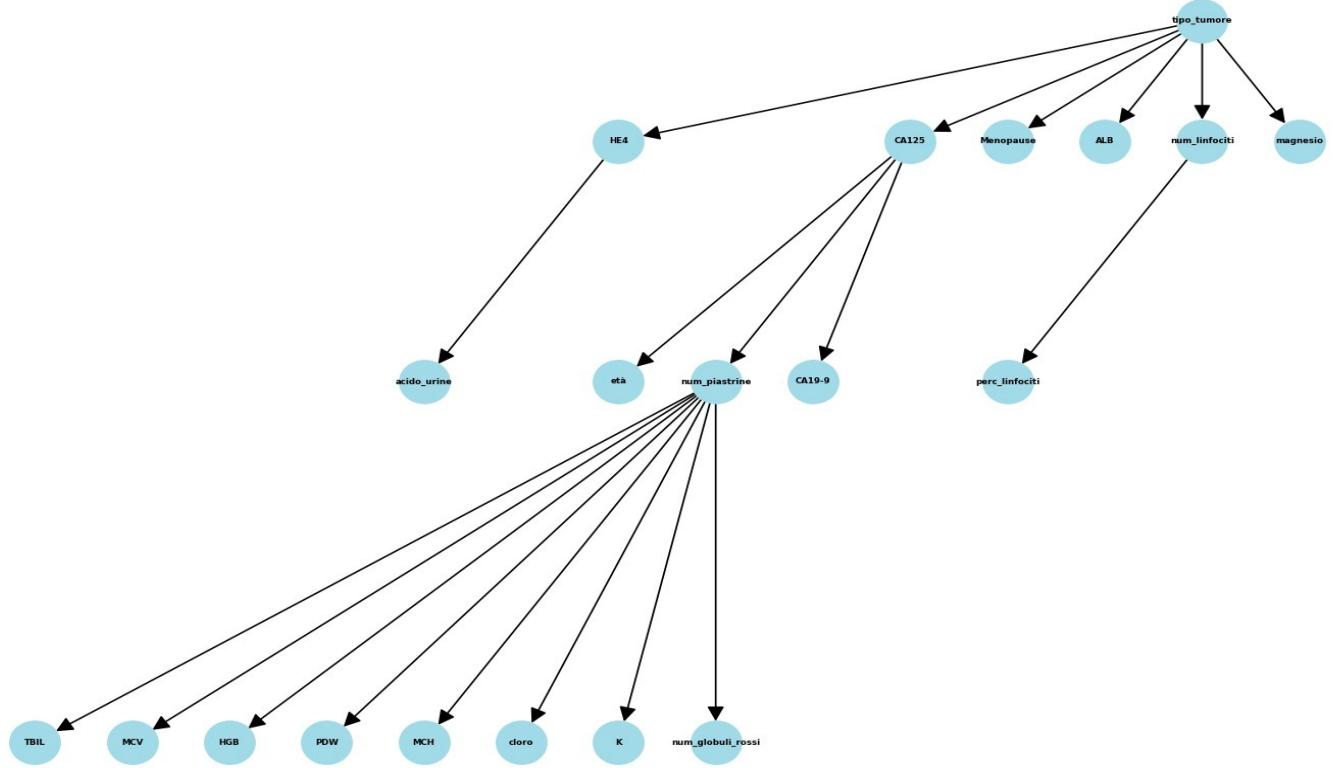
```
Valutazioni della rete Bayesiana
Accuracy: 0.7611
Precision: 0.81
Recall: 0.77
F1 Score: 0.63
```

### *Creazione della rete:*

Utilizzando il metodo BayesianNetwork(), è stata creata la rete, inserendo come argomenti gli archi e i nodi ottenuti.

Per una visualizzazione più chiara della rete, è stato estratto un sotto-grafo radicato nella variabile "tipo\_tumore" e sono state eseguite operazioni di disposizione ottimizzata dei nodi.

Rappresentazione grafica della rete:



Come ultimo passaggio, si è calcolata la probabilità che un individuo con determinati valori, abbia un tumore benigno o maligno. Il bayesian Estimator è lo stimatore che abbiamo utilizzato.

Con la struttura della rete definita, sono state aggiunte le variabili al modello e le sono state aggiornate con le nuove Conditional Probability Distributions (CPD) calcolate dal dataset. Questo passaggio è fondamentale per garantire che il modello rifletta accuratamente le relazioni probabilistiche tra le variabili di interesse.

Infine, sono state eseguite operazioni di inferenza e previsione sulla rete bayesiana per stimare la probabilità di avere un tumore benigno o maligno.

Caso benigno:

```
Probabilità per una donna di avere un tumore benigno:
+-----+
| tipo_tumore | phi(tipo_tumore) |
+=====+=====+
| tipo_tumore(0) | 0.9999 |
+-----+
| tipo_tumore(1) | 0.0001 |
+-----+
```

Caso maligno:

```
Probabilità per una donna di avere un tumore maligno:
+-----+
| tipo_tumore | phi(tipo_tumore) |
+=====+=====+
| tipo_tumore(0) | 0.0009 |
+-----+
| tipo_tumore(1) | 0.9991 |
+-----+
```