Ingegneria della Conoscenza
Heart Disease
Prof. Nicola Fanizzi

Gruppo di Lavoro
Notaro Lorenzo – 724931 – l.notaro1@studenti.uniba.it
Piergianni Giada – 717065 – g.piergianni4@studenti.uniba.it

Repository GitHub:
https://github.com/lnotaro1/IconHeartDisease-PiergianniNotaro.git

Sommario

INTRODUZIONE 1

INFORMAZIONI TECNICHE 1

ANALISI DEL DATASET 2

ESPLORAZIONE GRAFICA DELLE VARIABILI CATEGORICHE 5

Variabili categoriche considerate singolarmente 5

1. Sex 5
2. Cp 5
3. Fbs 6
4. Restecg 7
5. Exng 7
6. Slp 8
7. Caa 8
8. Thall 9
9. Output 9

Variabili categoriche considerate in relazione tra le stesse 10

1. Sex - Output 10
2. Cp – Output 11
3. Fbs - Output 11
4. Restecg - Output 12
5. Exng - Output 13
6. Slp - Output 13
7. Caa - Output 14
8. Thall - Output 14

ESPLORAZIONE GRAFICA DELLE VARIABILI NUMERICHE 15

Variabili numeriche considerate singolarmente 15

1. Age 15
2. Trtbps 15
3. Chol 16
4. Thalachh 16
5. Oldpeak 17

Variabili categoriche considerate in relazione tra le stesse 18

1. Age – Cholestoral – Output 18

2. Age – Resting Blood Pressure - Output 18
3. Age – Maximum Heart Rate - Output 19
Matrice di correlazione 19
Standardizzazione delle variabili 20
APPRENDIMENTO SUPERVISIONATO 22
Matrice di confusione 23
Curva roc (receiver operating characteristics) 24
Metriche 25
Logistic regression 25
No Iperparametri 26
Con Iperparametri 27
Grafici a confronto 29
K-nearest neighbors 30
No Iperparametri 31
Con Iperparametri 33
Grafici a confronto 35
Support vector machines 36
No Iperparametri 37
Con Iperparametri 38
Grafici a confronto 41
Decison tree 42
No Iperparametri 42
Con Iperparametri 43
Grafici a confronto 46
Random forest 47
No Iperparametri 47
Con Iperparametri 48
Grafici a confronto 51
Conclusioni 51
Metriche senza l'uso di iperparametri a confronto 52
Metriche con l'uso di iperparametri a confronto 54
BAYESIAN NETWORK 56

INTRODUZIONE

L'obiettivo primario del progetto è fornire supporto medico sia per la diagnosi preventiva di problemi cardiaci, aiutando gli specialisti a distinguere se un paziente soffre o meno di una condizione cardiaca, sia per identificare le caratteristiche condivise tra i pazienti affetti da tali disturbi. In questo modo, si può valutare quanto siano simili tra loro gli individui all'interno del campione di dati.

INFORMAZIONI TECNICHE

Il linguaggio di programmazione utilizzato è Python v. 3.11.6
Il software è Microsoft Visual Studio Code
Le librerie principali che si incontrano nel codice python sono:

- numpy →libreria che offre un supporto matematico e semplifica la gestione di calcoli matematici mediante array e funzioni matematiche
- pandas → libreria utilizzata per la gestione (lettura, pulizia, preparazione) di insiemi di dati strutturati in tabelle
- matplotlib → libreria usata per la visualizzazione dei dati python in grafici e personalizzazione degli stessi
- seaborn → libreria utilizzata per la visualizzazione di dati python (soprattutto dati statistici e categorici). Utilizzata spesso in combinazione con matplotlib per ottenere il massimo controllo e flessibilità nella creazione di grafici personalizzati
- sklearn → libreria utilizzata per l'apprendimento automatico. Essa fornisce strumenti efficaci per la modellazione statistica, la classificazione, la regressione, l'ottimizzazione dei parametri dei modelli
- pgmpy → libreria Python specializzata nella costruzione, l'addestramento e l'inferenza di modelli grafici probabilistici (PGM), che includono reti Bayesiane
- plotly → Plotly è una libreria di visualizzazione dei dati. Permette la creazione di grafici interattivi e dinamici, che possono essere incorporati in pagine web o in applicazioni

ANALISI DEL DATASET

Per ogni individuo il dataset raccoglie le seguenti informazioni:

- age
- sex
- cp (chest pain type)
- trtbps (resting blood pressure in mm Hg)
- chol (cholestoral in mg/dL fetched via BMI sensor)
- fbs (fasting blood sugar)
- restecg (resting ecg results)
- thalachh (maximum heart rate during exercise)
- exng (exercise induced angina)
- oldpeak (ST-segment depression)
- slp (slope of ST segment)
- caa (number of major vessels colored by fluoroscopy)
- thall (thalassemia type)
- output (risk of heart disease)

In particolare:

- sex - sex (1 = male; 0 = female)
- cp - chest pain type (0 = typical angina; 1 = atypical angina; 2 = non-anginal pain; 3 = asymptomatic)
- fbs - fasting blood sugar (0 = <= 120 mg/dL; 1 = > 120 mg/dL)
- restecg - resting ecg results (0 = normal; 1 = having ST-T wave abnormality; 1 = probable or definite Left Ventricular Hypertrophy)
- exng - exercise induced angina (1 = yes; 0 = no)
- slp - slope of ST segment (0 = Downsloping; 1 = Flat; 2 = Upsloping diagnosis)
- ca - number of major vessels (0-3) colored by flourosopy
- thall - thalessemia type (0 = None; 1 = Fixed Defect; 2 = Reversible Defect; 3 = Thalassemia)
- output – risk of heart disease (0 = no; 1 = yes)

Quindi per ogni individuo del dataset si conosce:

- age età
- sex genere
- cp tipo di dolore toracico
  Può assumere i seguenti valori:
  - 0 = angina tipica
  - 1 = angina atipica
  - 2 = dolore non anginoso
  - 3 = asintomatico
- trtbps pressione arteriosa a riposo misurata in mm Hg (millimetri di mercurio)
- chol colesterolo in mg/dL rilevato tramite sensore BMI
- fbs glicemia a digiuno
  Può assumere i seguenti valori:
  - 0 = <= 120 mg/dL;
  - 1 = > 120 mg/dL
- restecg risultati dell'elettrocardiogramma a riposo
  Può assumere i seguenti valori:
  - 0 = normale
  - 1 = è presente un'anomalia dell'onda ST-T
  - 2 = vi è una probabile o definita ipertrofia ventricolare sinistra
- thalachh frequenza cardiaca massima durante l'esercizio
- exng angina da sforzo
  Può assumere i seguenti valori:
  - 0 = no
  - 1 = si
- oldpeak sottoslivellamento del tratto ST
- slp pendenza del tratto ST
  Può assumere i seguenti valori:
  - 0 = inclinazione discendente
  - 1 = piatta
  - 2 = inclinazione ascendente
- caa numero di vasi principali colorati mediante fluoroscopia

- thall tipo di talassemia
  Può assumere i seguenti valori:
  - 0 = nessuna talassemia
  - 1 = difetto fisso
  - 2 = difetto reversibile
  - 3 = talassemia
- output rischio problema cardiaco
  Può assumere i seguenti valori:
  - 0 = no
  - 1 = si

Il dataset è formato da 303 righe e 14 colonne e tutti i valori sono di due tipi: interi o float; dopo aver ottenuto queste informazioni è stato controllato se all'interno del dataset ci fossero elementi duplicati oppure elementi nulli:

```
Number of null data:
 age         0
sex          0
cp           0
trtbps       0
chol         0
fbs          0
restecg      0
thalachh     0
exng         0
oldpeak      0
slp          0
caa          0
thall        0
output       0
dtype: int64
Number of duplicated data:  1
Number of duplicated data:  0
```

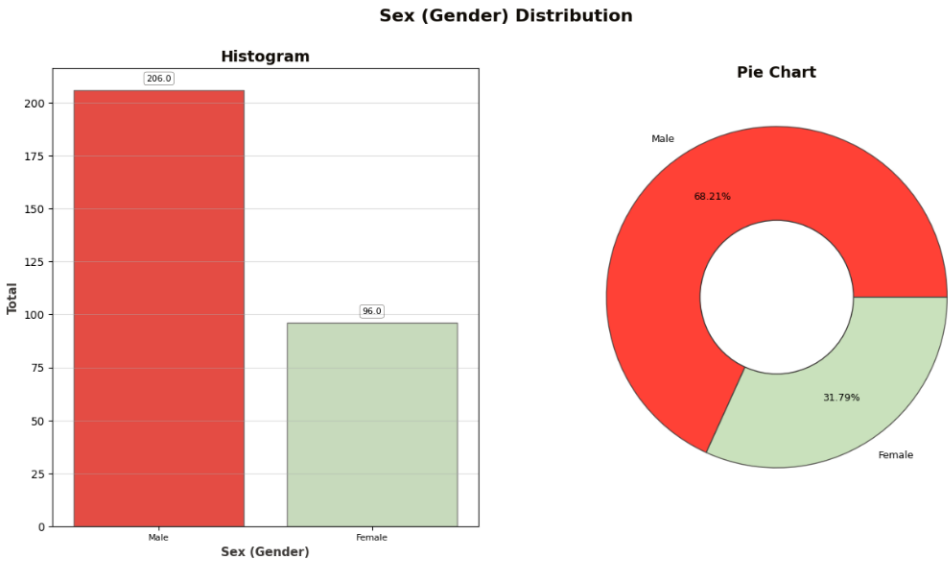
Si è quindi concluso che non ci sono dati nulli, ma c'è un valore duplicato che è stato successivamente rimosso.
Si possono riconoscere nel dataset due tipi di variabili: numeriche e categoriche.
Le variabili categoriche sono: sex, cp, fbs , restecg, exng, slp, caa, thall, output; le variabili numeriche sono: age, trtbps, chol, thalachh, oldpeak.
Per poter approfondire lo studio sul dataset è stata fatta un'esplorazione grafica dei dati.

ESPLORAZIONE GRAFICA DELLE VARIABILI CATEGORICHE

Variabili categoriche considerate singolarmente

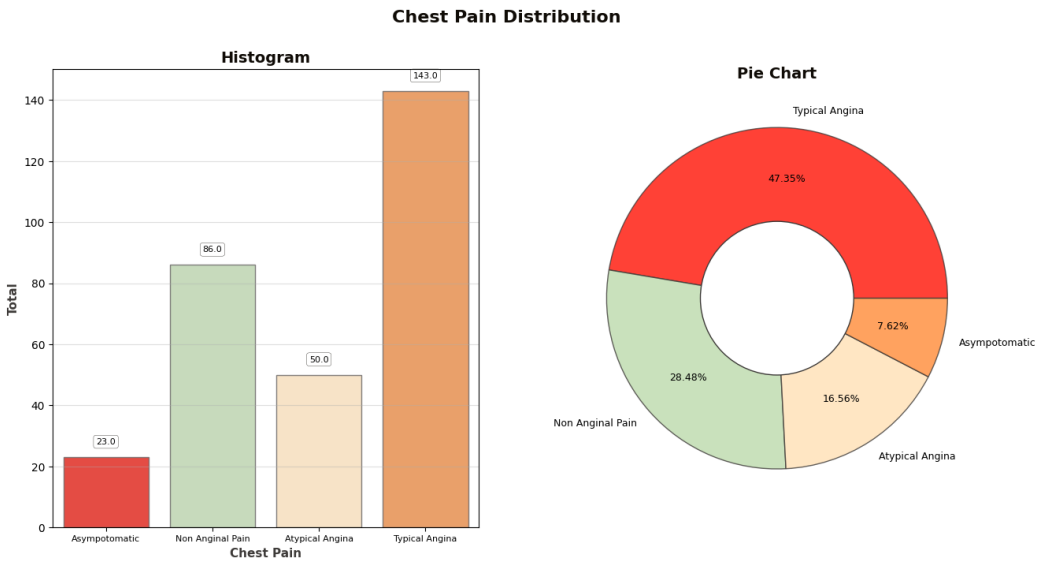
Si sono considerate le diverse variabili categoriche singolarmente e in correlazione tra loro.

1. Sex



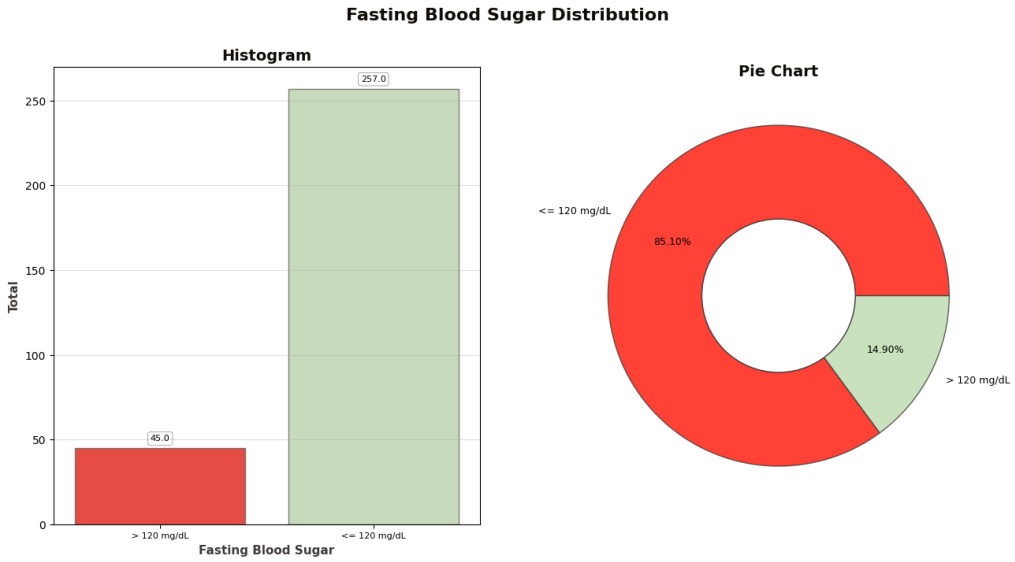
Gli individui di sesso maschile rappresentano il 68,21%, più precisamente sono 206, invece quelli di sesso femminile rappresentano il 31,79%, sono presenti 96 donne nel dataset. Possiamo concludere quindi che più della metà delle persone sono maschi.

2. Cp



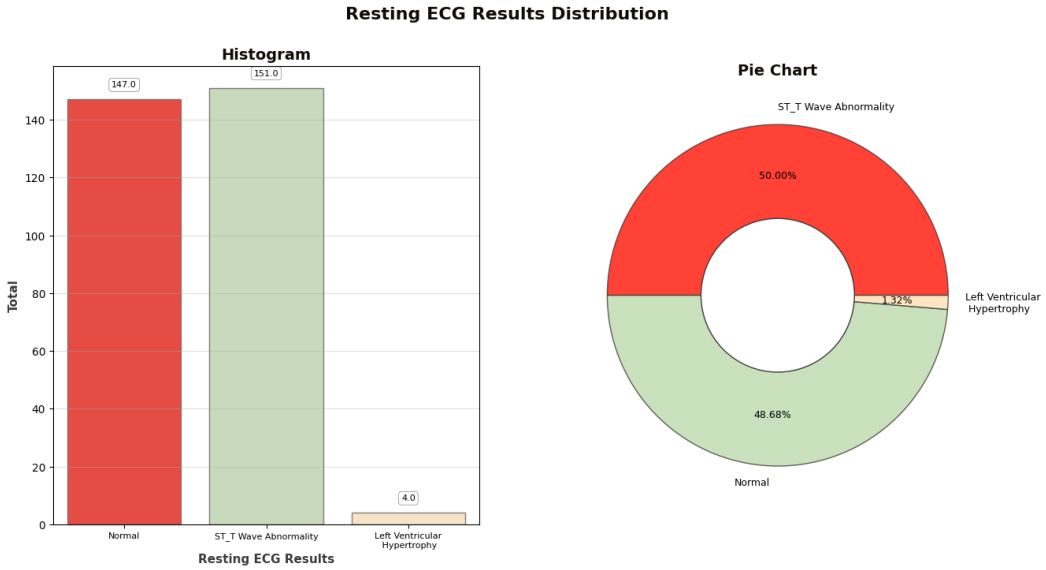
Il 47,35% delle persone, più precisamente 143, presenta un'angina tipica, questo valore indica che il paziente sta sperimentando un tipo di dolore toracico che è tipico di un'angina pectoris. L'angina è un dolore o disagio al torace causato da una diminuzione temporanea del flusso sanguigno al cuore, di solito a causa di arterie coronarie occluse.
Il 28,48% dei pazienti, cioè 86 di questi, ha un dolore non anginoso, ossia un dolore non causato da problemi cardiaci
Il 16,56% dei pazienti, in dettaglio 50 pazienti, presentano un'angina atipica.
Solo 23 pazienti, cioè il 7,62% del totale è asintomatico.

3. Fbs



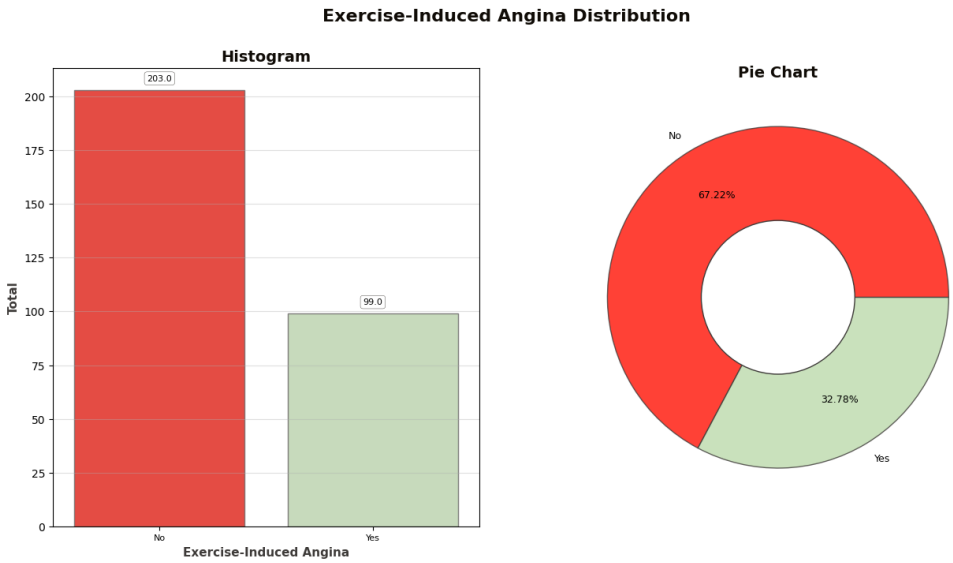
Più della metà dei pazienti, più precisamente l'85,10%, ossia 257 pazienti, presenta un valore <= 120 mg/dL. Il 14,90% dei pazienti ha un valore > 120 mg/dL.

4. Restecg



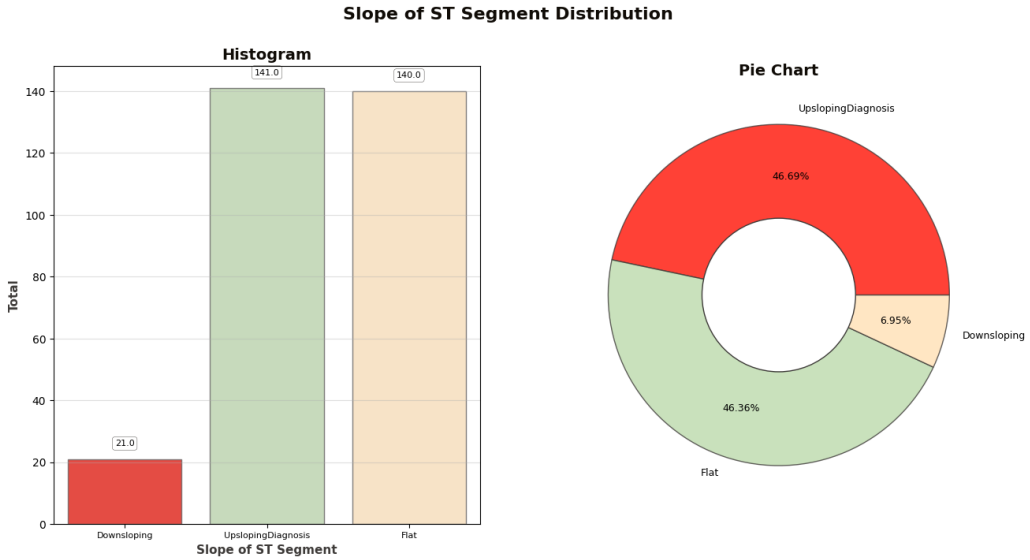
Il 50,00% dei pazienti (151) presentano un'anomalia dell'onda ST-T, queste anomalie possono essere indicative di problemi cardiaci.
Il 48,68% dei pazienti (147) hanno un ECG registrato durante il riposo che mostra un ritmo cardiaco e una morfologia delle onde tipiche e senza anomalie.
Il restante 1,32% dei pazienti (4) ha una probabile o definita ipertrofia ventricolare sinistra che è spesso associata a condizioni come l'ipertensione.

5. Exng



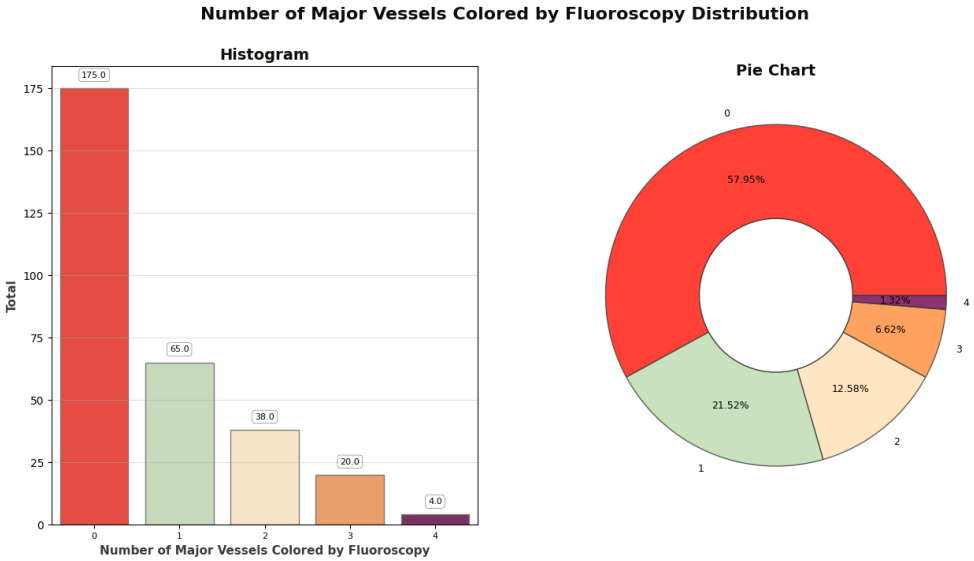
Il 67,22% dei pazienti non sperimenta angina durante l'esercizio fisico o uno sforzo. Il restante 32,78% presenta angina, dolore toracico o disagio, durante uno sforzo oppure durante un esercizio fisico.

6. Slp

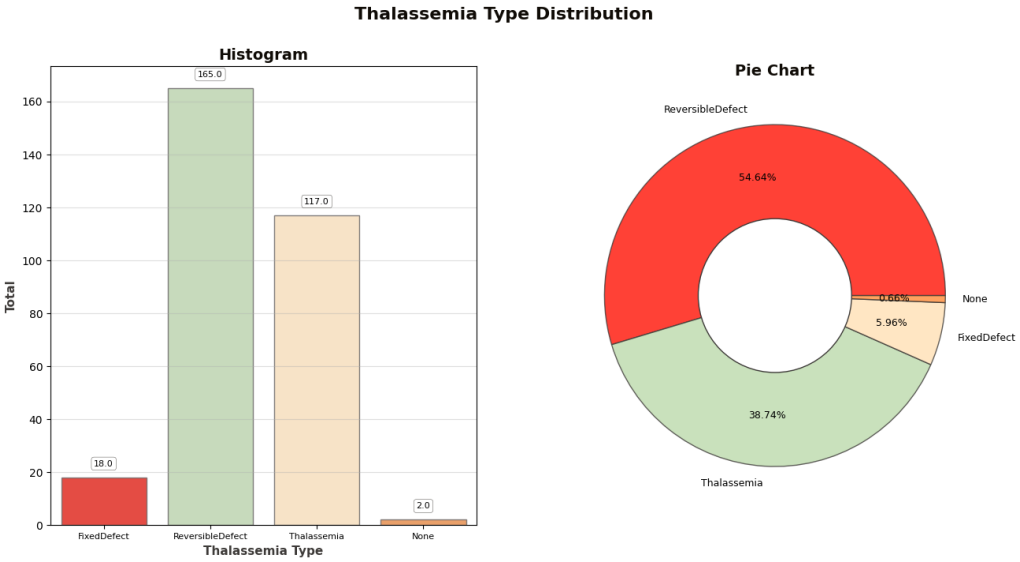


Il 46,69% dei pazienti ha un'inclinazione ascendente del tratto ST, cioè il segmento ST è inclinato verso l'alto rispetto la linea base.
Il 46,36% dei pazienti presenta un tratto ST piatto.
Solo il 6,95% dei pazienti ha un'inclinazione discendente del tratto ST, cioè, è inclinato verso il basso rispetto alla linea base.

7. Caa

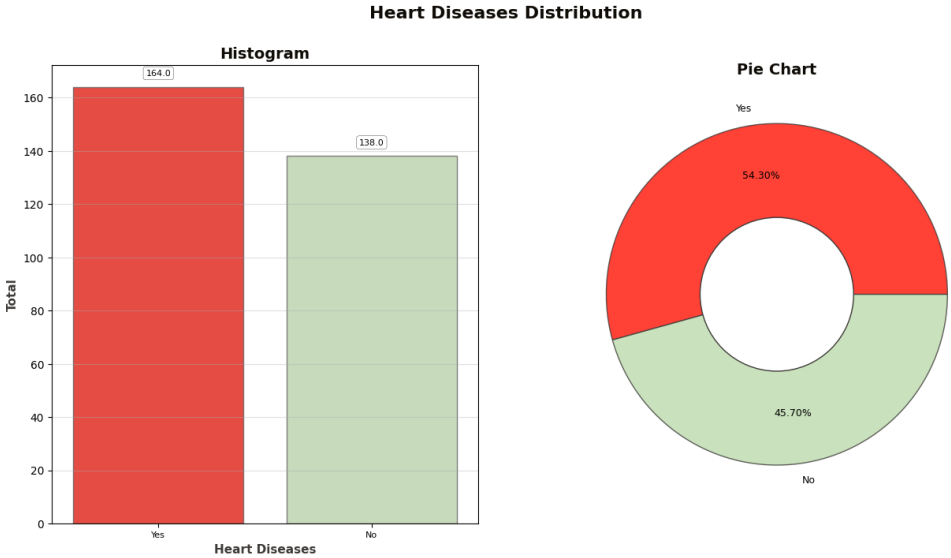


8. Thall



Lo 0,66% dei pazienti non presenta talassemia, cioè non è presente alcun difetto genetico nella produzione di emoglobina.
Il 5,69% dei pazienti presenta un difetto fisso, cioè un difetto fisso nella produzione di emoglobina.
Il 56,64% dei pazienti presenta un difetto reversibile, cioè la presenza di un difetto nella produzione di emoglobina che è reversibile.
Il 38,74% dei pazienti ha talassemia quindi il paziente ha una condizione ereditaria che comporta difetti nella produzione di emoglobina.

9. Output

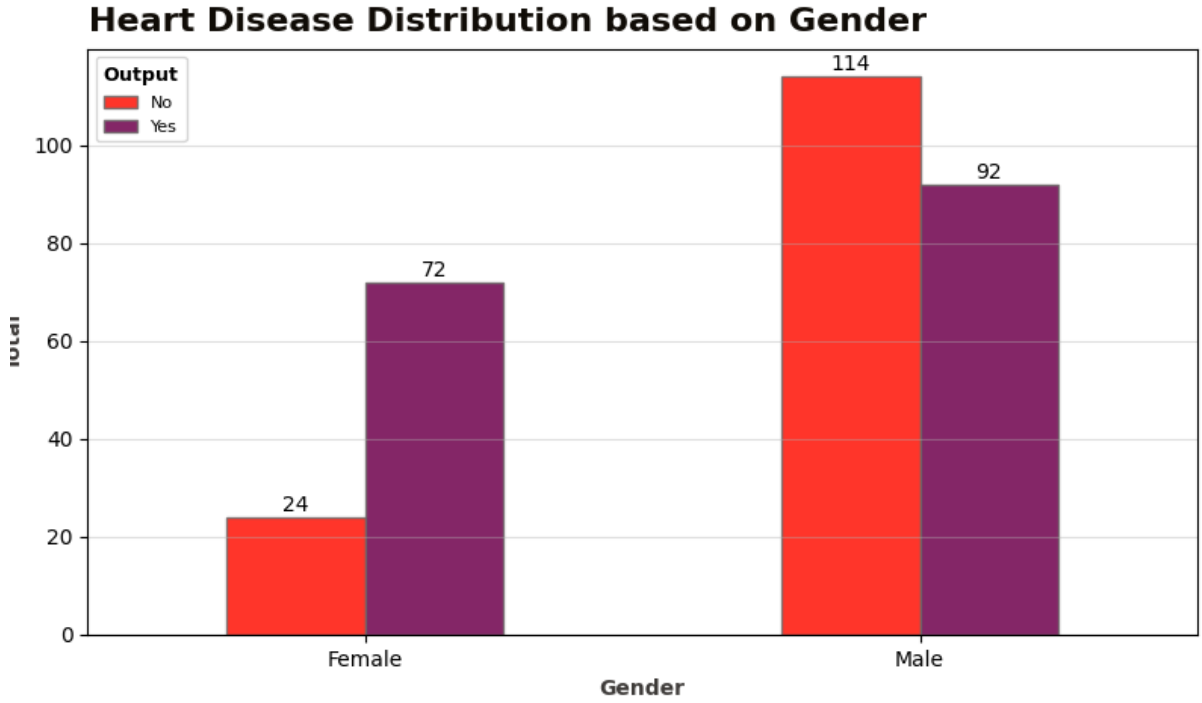


Il 54,30% dei pazienti presenta un problema cardiaco mentre il restante 45,70% non presenta un problema cardiaco.

Variabili categoriche considerate in relazione tra le stesse

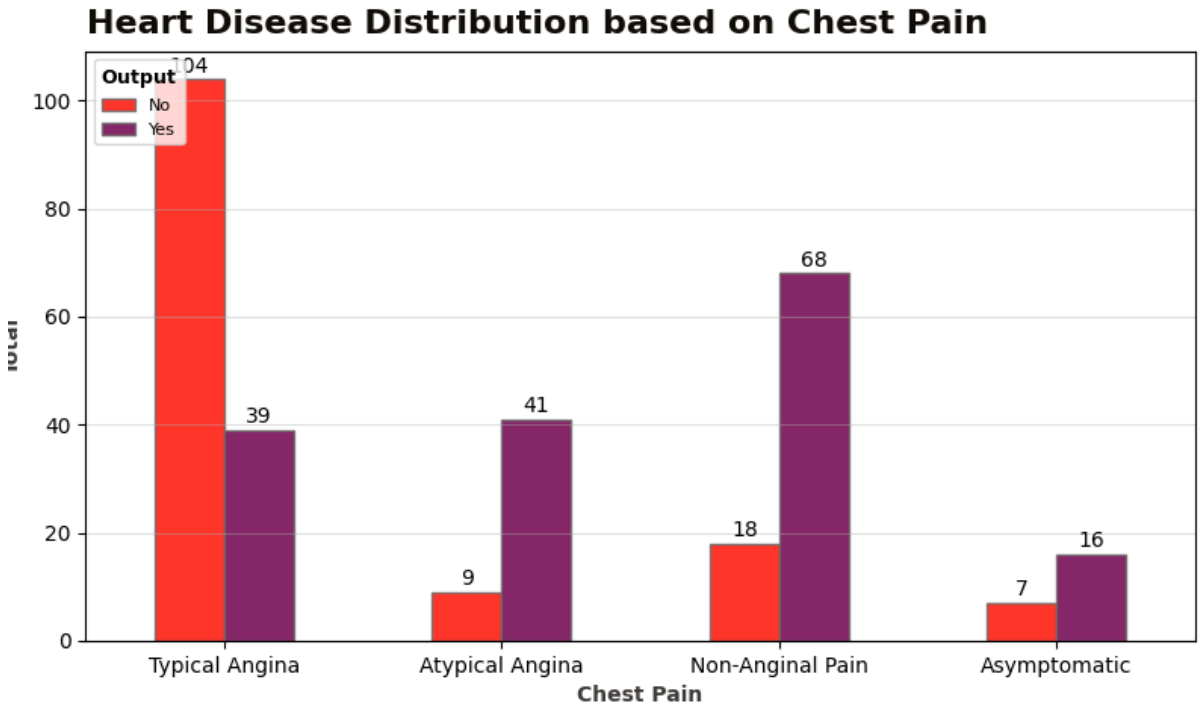
Successivamente abbiamo creato dei grafici per studiare la relazione che esiste tra queste variabili categoriche e la variabile 'output'.

1. Sex - Output



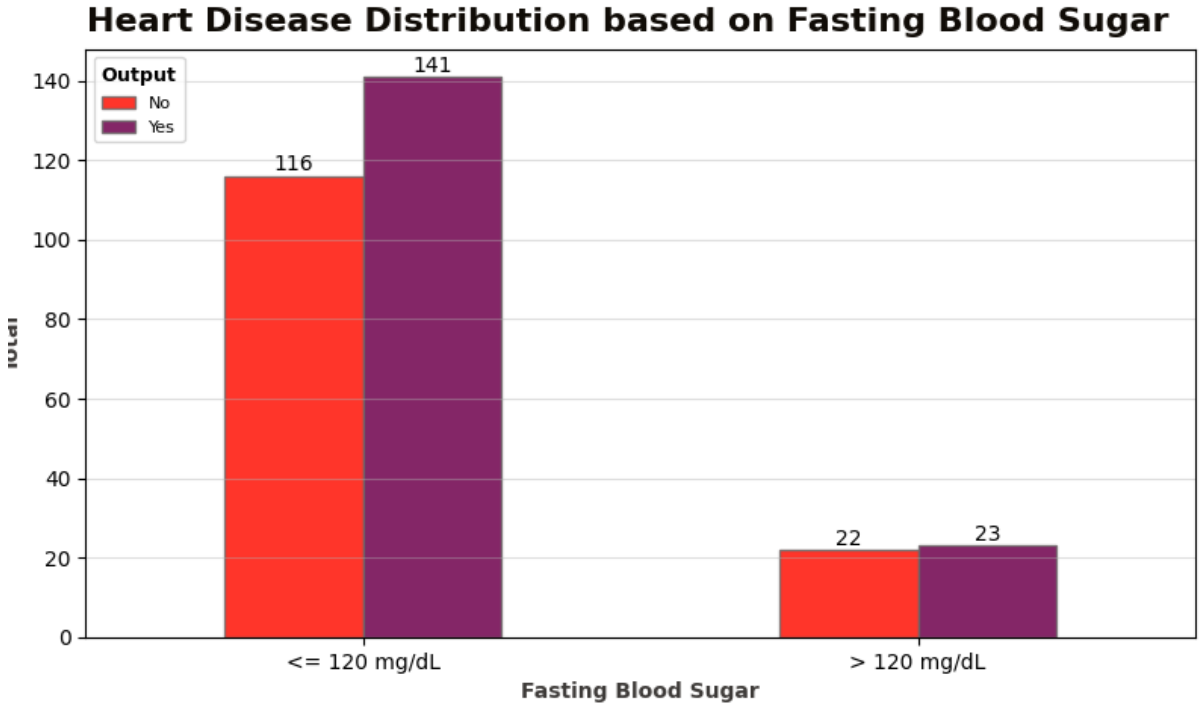
Dal grafico si può notare come gli uomini siano più portati a sviluppare dei disturbi cardiaci rispetto alle donne

2. Cp – Output



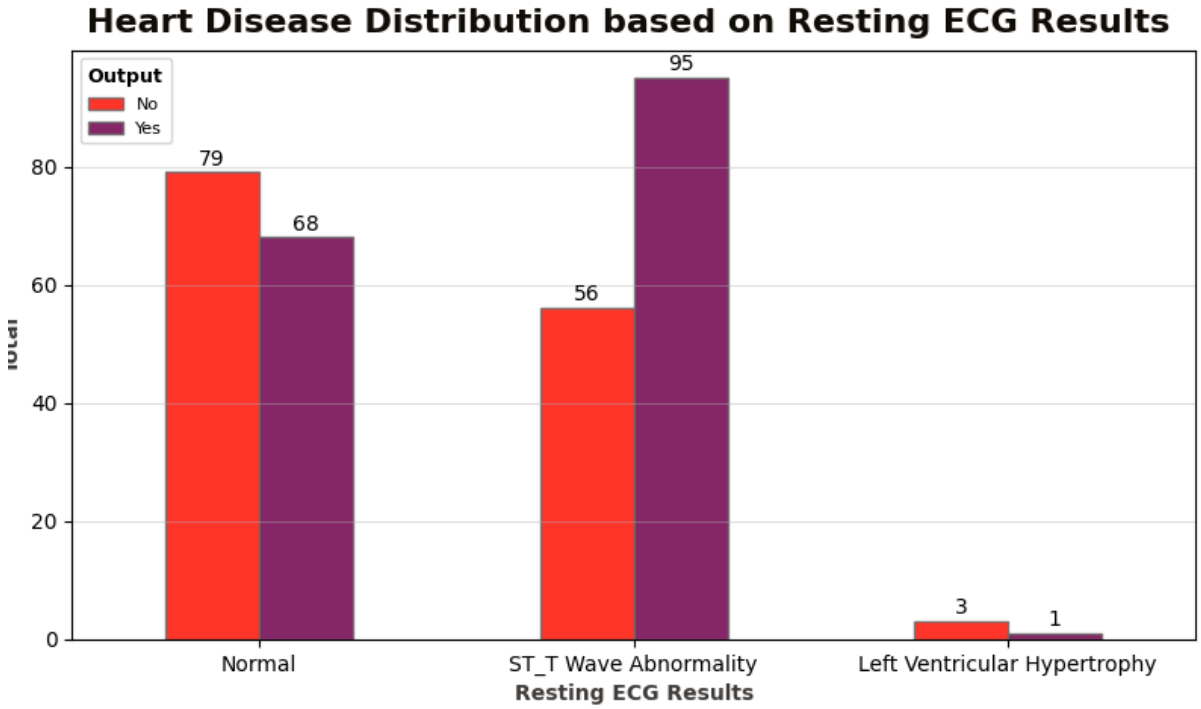
Dai risultati forniti dal grafico si può concludere che coloro che presentano un dolore non anginoso sono più portati ad avere un problema cardiaco. Inoltre, si può notare che coloro che presentano un'angina tipica hanno meno probabilità di sviluppare un problema cardiaco.

3. Fbs - Output



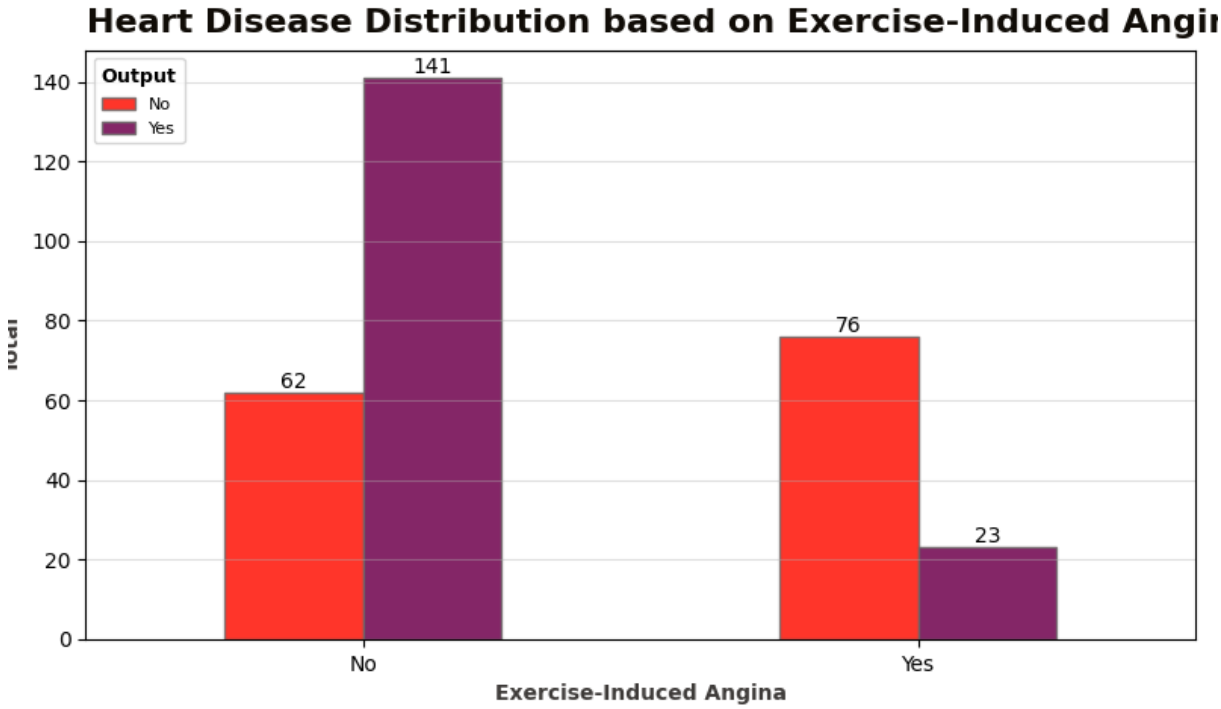
I dati a disposizione in questo caso sono piuttosto equilibrati, si può concludere però che coloro che presentano un valore minore o uguale a 120 mg/dL sono leggermente più portati ad avere un problema cardiaco.

4. Restecg - Output



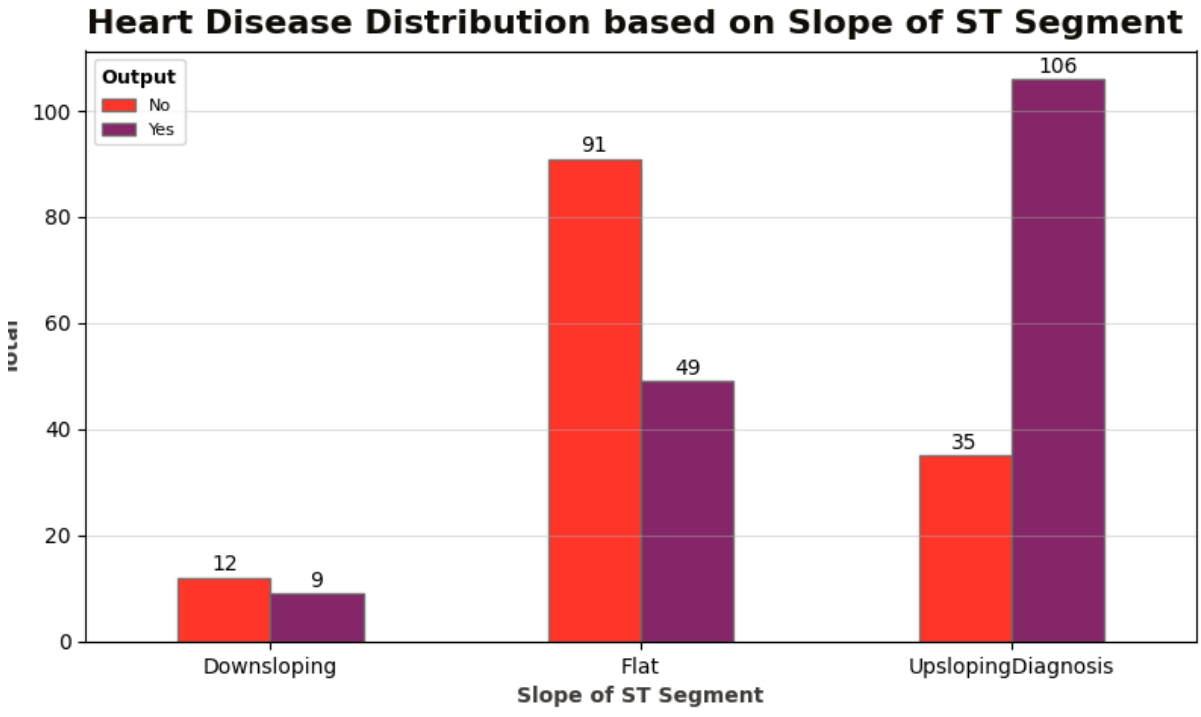
I pazienti presenti nel dataset che presentano un'anormalità dell'onda ST-T sono più portati ad avere un problema cardiaco.

5. Exng - Output



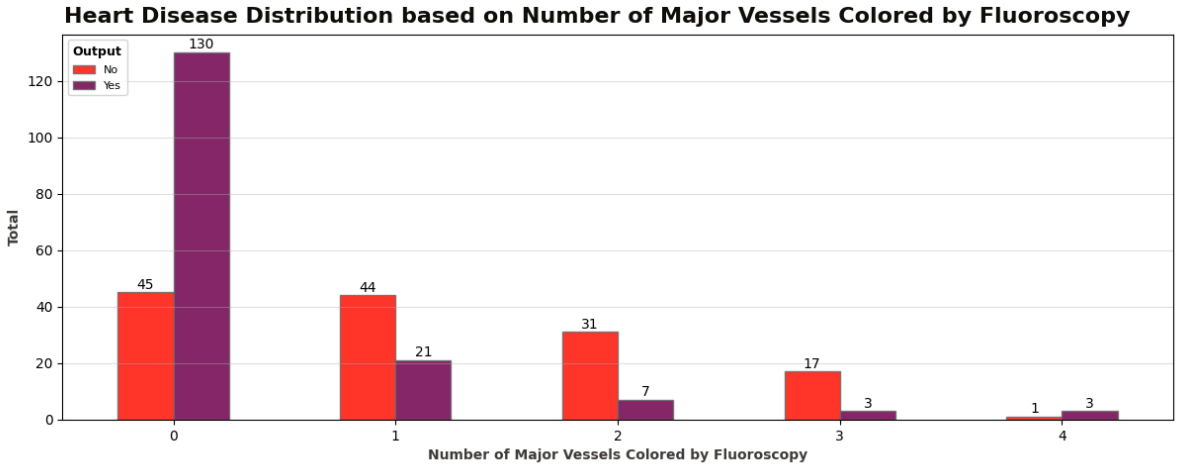
La probabilità di non avere un problema cardiaco per coloro che presentano un'angina da sforzo è più alta rispetto a coloro che ce l'hanno.

6. Slp - Output

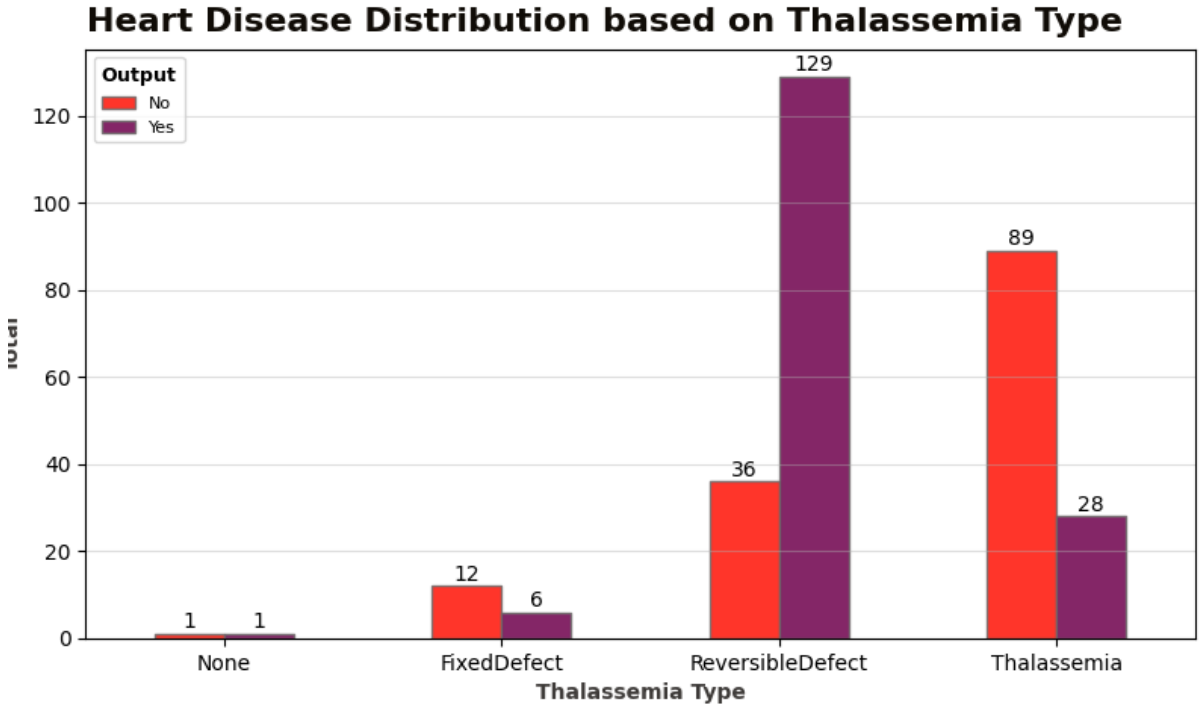


Coloro che hanno un'inclinazione ascendente del tratto ST, cioè il segmento ST inclinato verso l'alto rispetto la linea base, sono più portati a sviluppare un problema cardiaco rispetto a coloro che hanno un'inclinazione discendente del tratto o che presentano un tratto ST piatto.

7. Caa - Output



8. Thall - Output



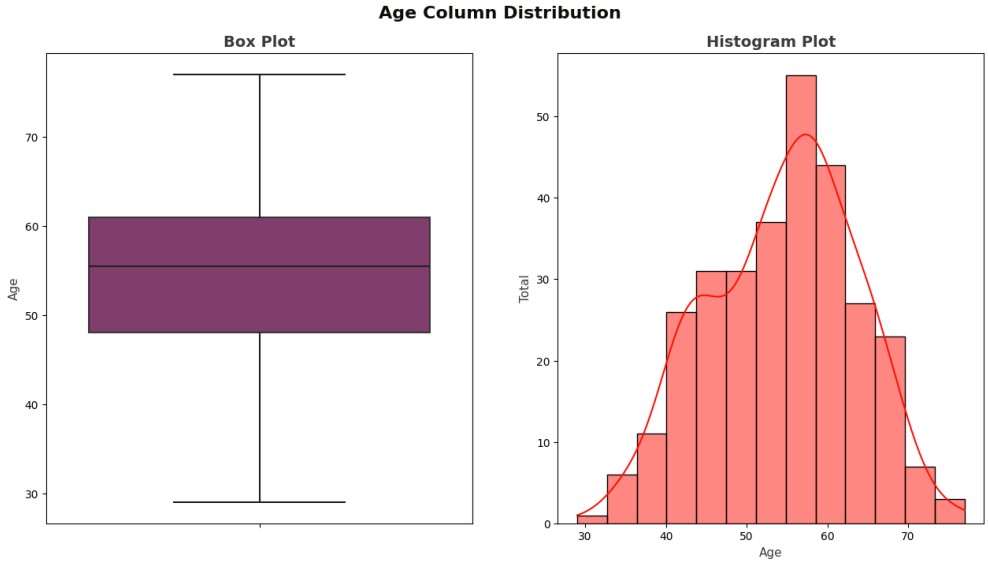
I pazienti che presentano un difetto reversibile, cioè la presenza di un difetto nella produzione di emoglobina che è reversibile, hanno più probabilità di sviluppare un problema cardiaco.

ESPLORAZIONE GRAFICA DELLE VARIABILI NUMERICHE

Variabili numeriche considerate singolarmente

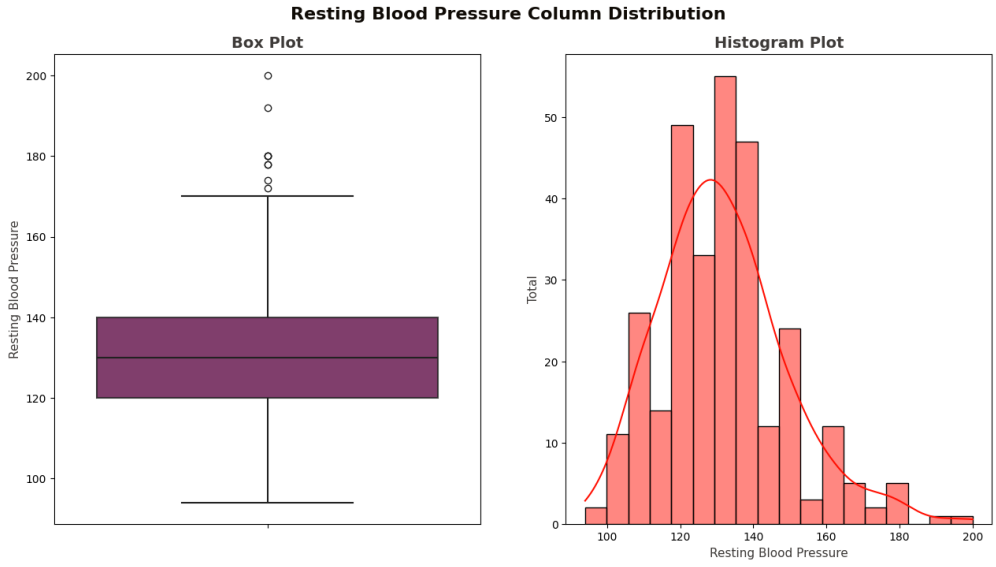
Si sono considerate le diverse variabili numeriche singolarmente e in correlazione tra loro.

1. Age



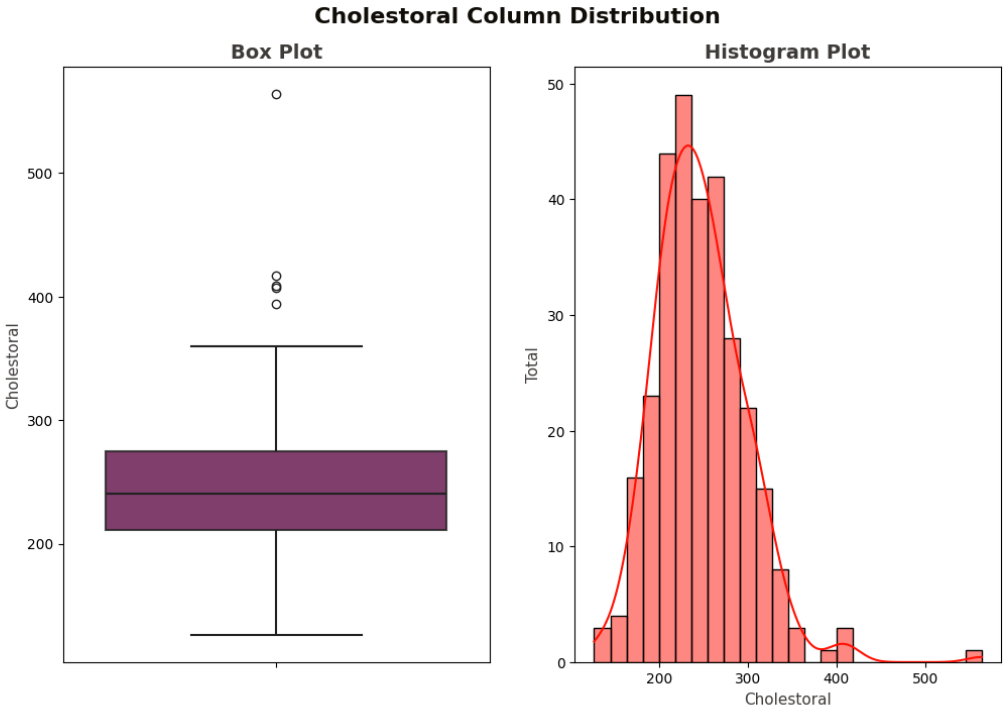
La maggior parte dei pazienti all'interno del dataset ha un'età compresa tra i 50 e i 60 anni.

2. Trtbps



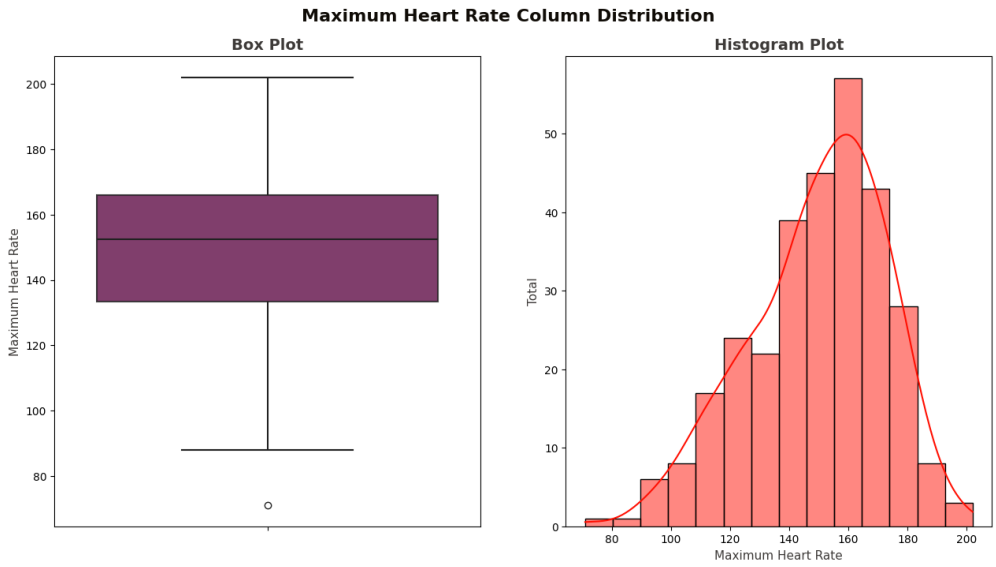
La pressione arteriosa a riposo misurata in mm Hg (millimetri di mercurio) della maggior parte dei pazienti presenti nel dataset è compresa tra i 120 e i 140.

3. Chol



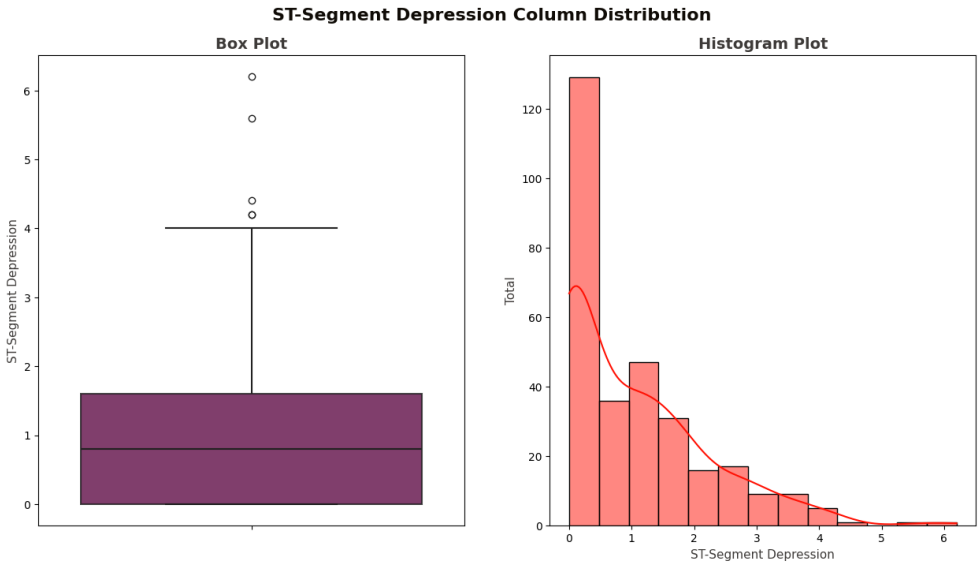
La maggior parte dei pazienti ha un valore di colesterolo in mg/dL, rilevato tramite sensore BMI, compreso tra 200 e 300.

4. Thalachh



La maggior parte dei valori per quanto riguarda la variabile ‘thalachh’ , che rappresenta il valore di frequenza cardiaca massima durante l’esercizio, sono compresi tra 140 e 170.

5. Oldpeak

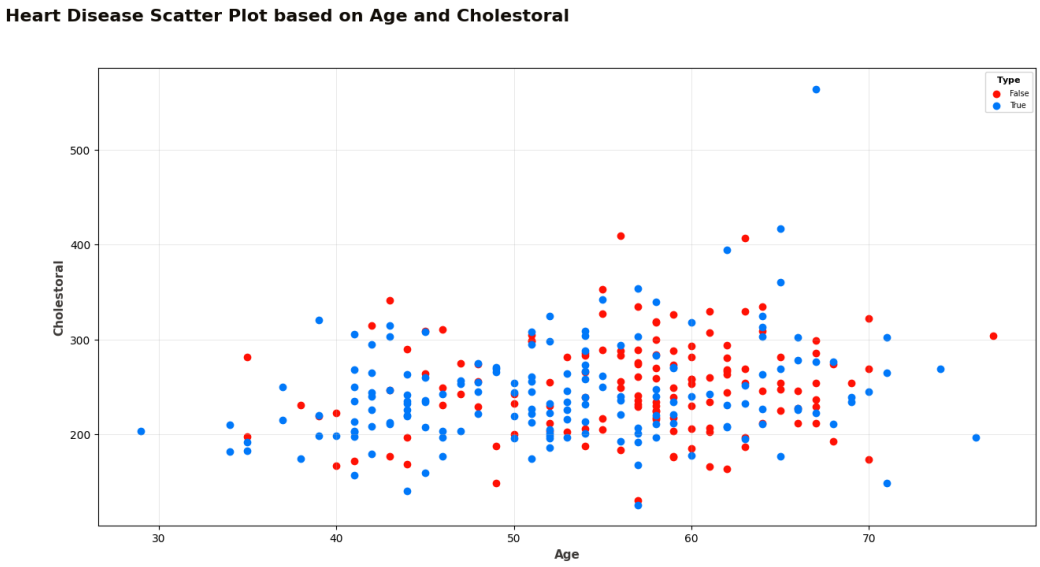


La maggior parte dei valori per quanto riguarda il sottoslivellamento del tratto ST sono compresi tra 0 e 2.

Variabili categoriche considerate in relazione tra le stesse

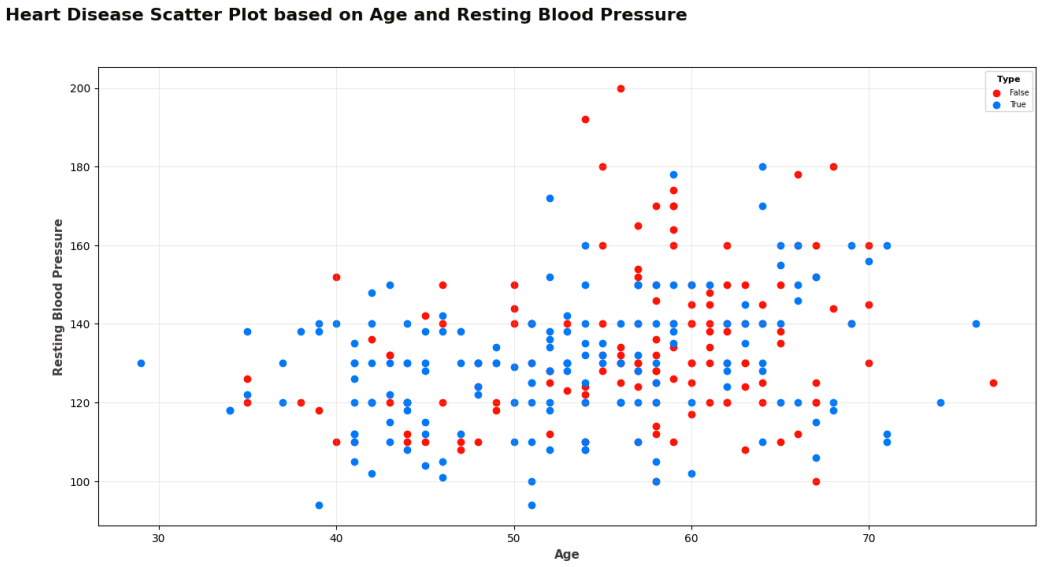
Successivamente abbiamo creato dei grafici per studiare la relazione che esiste tra queste variabili numeriche e la variabile 'output'.

1. Age – Cholestoral – Output



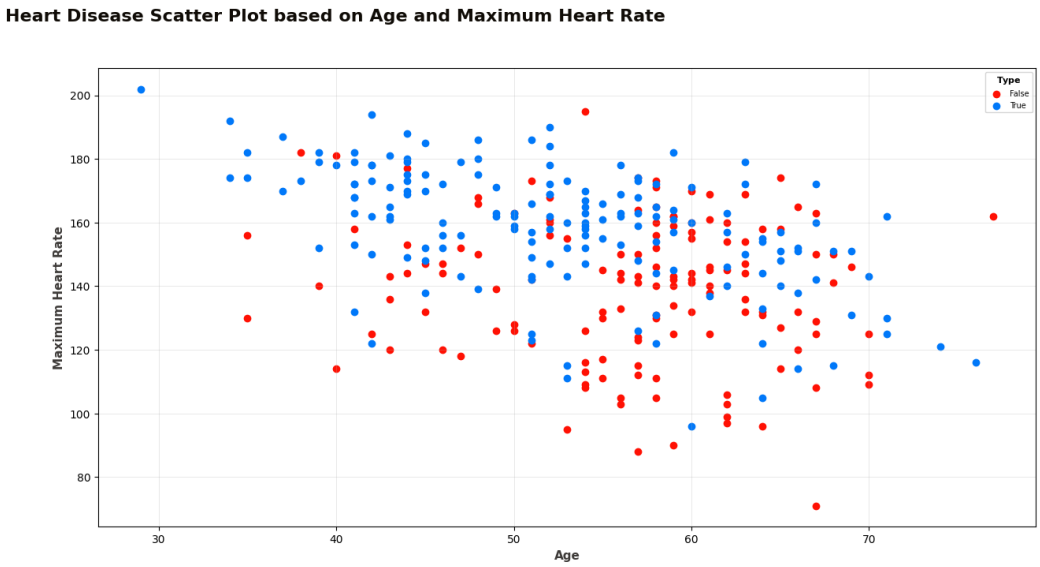
Coloro che hanno un valore della variabile 'chol', relativa al colesterolo, compresa tra 200 e 300 e un'età compresa tra 40 e 55 ha una maggiore probabilità di sviluppare un problema cardiaco, rispetto a quelli che hanno un'età maggiore di 55 con lo stesso valore di colesterolo.

2. Age – Resting Blood Pressure - Output



Coloro che hanno un valore della pressione arteriosa a riposo misurata in mm Hg (millimetri di mercurio), compresa tra 110 e 140 e un'età compresa tra 40 e 65 ha una maggiore probabilità di sviluppare un problema cardiaco.

3. Age – Maximum Heart Rate - Output

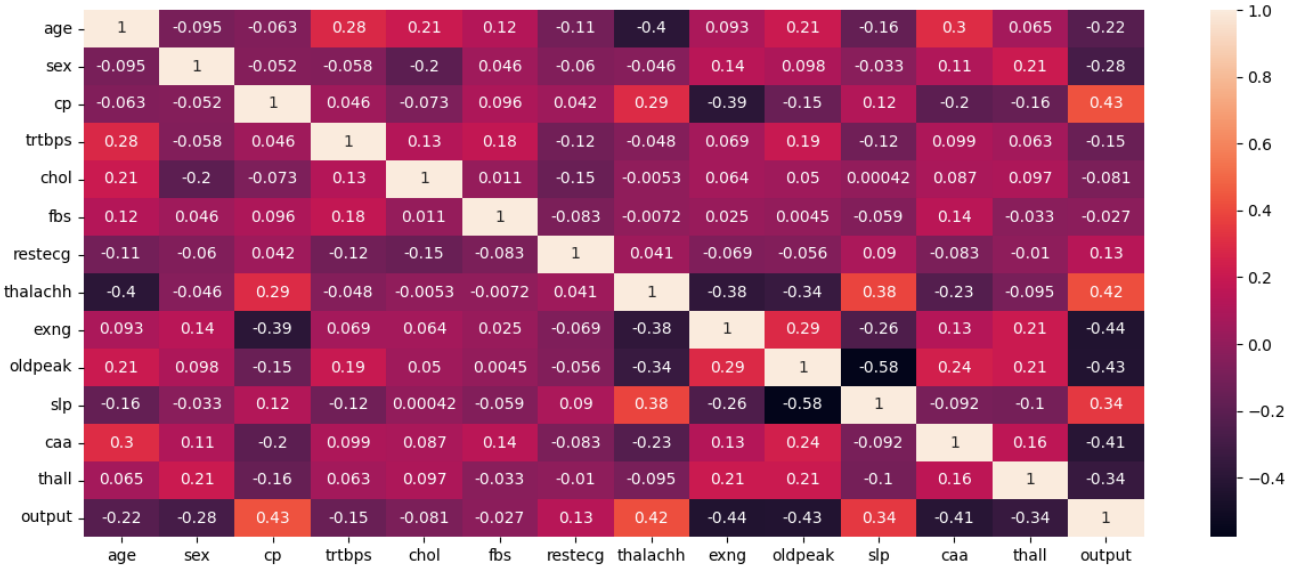


Coloro che hanno una frequenza cardiaca massima durante l'esercizio tra 140 e 190 e un'età compresa tra 40 e 60 ha una probabiltà più alta di presentare un disturbo cardiaco.

Matrice di correlazione

L'esplorazione grafica si è conclusa visualizzando la matrice di correlazione.
Una matrice di correlazione è una tabella che mostra i coefficienti di correlazione tra le variabili. Essa è composta da righe e colonne che mostrano le variabili. Ogni cella della tabella contiene il coefficiente di correlazione.
La correlazione è una misura che indica la relazione lineare fra due variabili casuali.

È compresa sempre fra -1 e 1, dove:
- -1 significa che le due variabili hanno una relazione lineare inversa, vale a dire che all'aumentare di una, l'altra diminuisce
- 1 significa che le due variabili hanno una relazione lineare diretta, vale a dire che all'aumentare di una aumenta anche l'altra
- 0 significa che non è possibile stabilire fra le due variabili un andamento lineare



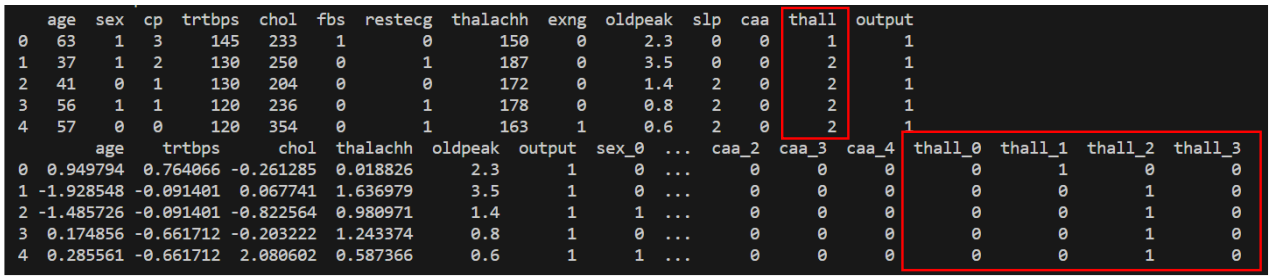
Standardizzazione delle variabili

Successivamente, per non apportare modifiche al dataset, è stata creata una copia di quest'ultimo per procedere con la fase di preelaborazione dei dati. Si è infatti deciso di standardizzare le variabili numeriche e le variabili categoriche.
Per le variabili numeriche è stato utilizzato lo StandardScaler() che effettua la standardizzazione delle feature portando il valore della media a 0 e deviazione standard a 1.
Con lo StandardScaler() il valore standard di un campione x è calcolato come:

$$z = (x - u) / s$$

dove u è la media dei campioni di addestramento e s è la deviazione standard dei campioni di addestramento.

Per le variabili categoriche è stata utilizzata la funzione *get_dummies()* che le converte in variabili dummy. Ciascuna variabile categorica viene trasformata in tante variabili binarie quante sono i differenti valori.



	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

	age	trtbps	chol	thalachh	oldpeak	output	sex_0	...	caa_2	caa_3	caa_4	thall_0	thall_1	thall_2	thall_3
0	0.949794	0.764066	-0.261285	0.018826	2.3	1	0	...	0	0	0	0	1	0	0
1	-1.928548	-0.091401	0.067741	1.636979	3.5	1	0	...	0	0	0	0	0	1	0
2	-1.485726	-0.091401	-0.822564	0.980971	1.4	1	1	...	0	0	0	0	0	1	0
3	0.174856	-0.661712	-0.203222	1.243374	0.8	1	0	...	0	0	0	0	0	1	0
4	0.285561	-0.661712	2.080602	0.587366	0.6	1	1	...	0	0	0	0	0	1	0

Nell'immagine è mostrato un esempio di come compare un variabile (thall in questo caso) dopo che è stata convertita in variabili dummy (thall_0, thall_1, thall_2, thall_3); si può notare che c'è una variabile dummy per ogni valore che può assumere la variabile thall.

APPRENDIMENTO SUPERVISIONATO

L'apprendimento supervisionato è una tecnica di apprendimento automatico che si concentra sull'addestramento di un sistema in modo da consentirgli di effettuare previsioni o classificazioni automatiche in risposta a determinati input. Queste previsioni o classificazioni sono basate su un insieme di dati di addestramento costituito da esempi che contengono sia gli input che i corrispondenti output desiderati. Questi esempi di addestramento sono forniti al sistema inizialmente. L'obiettivo dell'apprendimento supervisionato è far sì che il sistema possa fare previsioni accurate su nuovi dati o input che non sono stati precedentemente osservati.
Si parla di apprendimento supervisionato perché si hanno dati etichettati, ossia un insieme di dati in cui ogni esempio o campione di dati è associato a un'etichetta o a una categoria che indica l'output desiderato o la classe a cui appartiene. Queste etichette forniscono informazioni sul "corretto" risultato o sulla categoria a cui dovrebbe appartenere ciascun esempio di dati.
L'apprendimento supervisionato può essere distinto in due tipi di problemi: classificazione e regressione.
La differenza principale tra regressione e classificazione risiede nell'output previsto. La regressione prevede valori numerici continui, mentre la classificazione prevede categorie discrete.
La classificazione utilizza un algoritmo per assegnare accuratamente i dati di test a categorie specifiche. Riconosce delle entità specifiche all'interno del set di dati e cerca di trarre delle conclusioni su come queste entità dovrebbero essere etichettate o definite. La classificazione è utilizzata quando l'obiettivo è assegnare una categoria o una classe a ciascun esempio di input.
La regressione viene utilizzata per comprendere la relazione tra variabili dipendenti e indipendenti. La regressione è utilizzata quando l'obiettivo è prevedere un valore numerico continuo come output.
In questo caso di studio sono stati utilizzati cinque algoritmi di apprendimento supervisionato:

- Logistic Regression
- K-Nearest Neighbors
- Support Vector Machines

- Decision Tree
- Random Forest

Per ognuno di essi sono state prodotti due grafici:

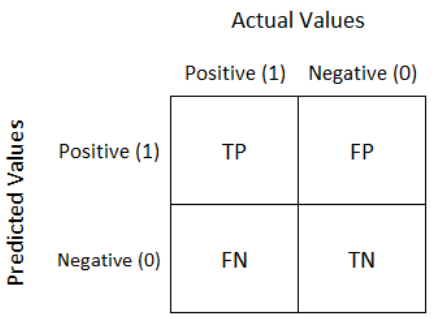
- Matrice di confusione
- Curva ROC

Ogni algoritmo di apprendimento è stato valutato prima e dopo l'ottimizzazione degli iperparametri. L'obiettivo è stato infatti ricercare gli iperparametri che massimizzassero le prestazioni del modello e questo processo è stato eseguito utilizzando la grid search.

Matrice di confusione

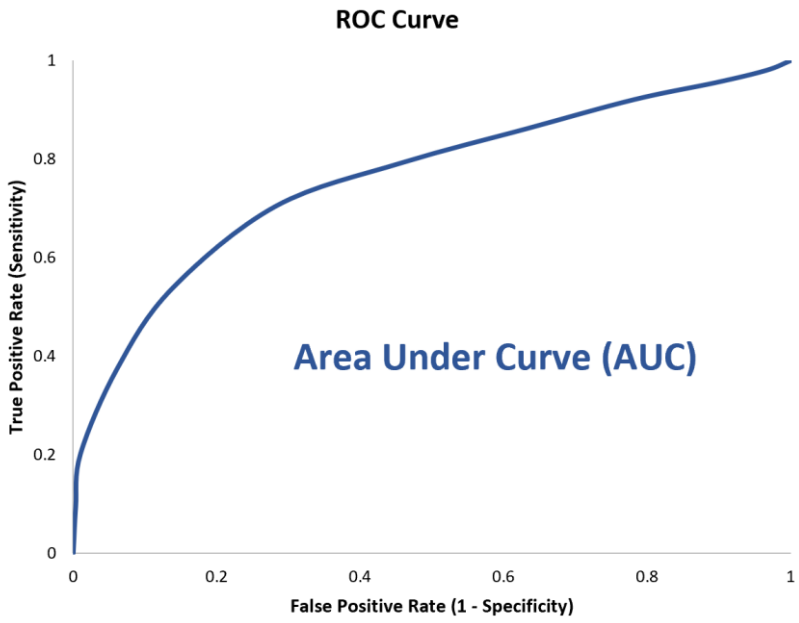
Una matrice di confusione è una tabella utilizzata nella classificazione dei modelli di machine learning per valutare le prestazioni di un algoritmo di classificazione. Essa mostra la relazione tra le previsioni del modello e le etichette reali dei dati. La matrice di confusione è particolarmente utile quando si lavora con problemi di classificazione binaria, ma può essere estesa per problemi di classificazione multi-classe.
Una matrice di confusione è generalmente composta da quattro parti principali:

- True Positives (TP): Questo rappresenta il numero di casi in cui il modello ha correttamente previsto una classe positiva.
- True Negatives (TN): Questo rappresenta il numero di casi in cui il modello ha correttamente previsto una classe negativa.
- False Positives (FP): Questo rappresenta il numero di casi in cui il modello ha erroneamente previsto una classe positiva quando in realtà era negativa.
- False Negatives (FN): Questo rappresenta il numero di casi in cui il modello ha erroneamente previsto una classe negativa quando in realtà era positiva.



Curva roc (receiver operating characteristics)

La ROC è una curva di probabilità e l'AUC è una misura numerica utilizzata per valutare le prestazioni di un classificatore binario basato sulla curva ROC. L'AUC rappresenta l'area sottesa dalla curva ROC ed è comunemente utilizzato come indicatore di quanto bene un classificatore possa discriminare tra le classi positive e negative. Maggiore è l'AUC, migliore è la capacità predittiva del modello.
Sull'asse delle ordinate della curva ROC, si rappresenta il True Positive Rate, TPR, che è il numero di campioni positivi correttamente classificati rispetto al totale dei campioni positivi; sull'asse delle ascisse, si rappresenta il False Positive Rate, FPR, che è il numero di campioni negativi erroneamente classificati come positivi rispetto al totale dei campioni negativi.
L'AUC è un valore compreso tra 0 e 1. Un valore pari 0 indica che il classificatore è completamente inadeguato e non è in grado di discriminare tra le classi. Un valore pari a 1 indica che il classificatore è perfetto e può separare completamente le classi senza errori. Quando l'AUC è pari a 0,7, significa che c'è il 70% di possibilità che il modello sia in grado di distinguere tra classe positiva e classe negativa. Quando l'AUC è circa 0,5, il modello non ha alcuna capacità di discriminazione.



Metriche

Logistic regression

La regressione logistica è un metodo statistico utilizzato per costruire modelli di machine learning in cui la variabile dipendente è dicotomica: cioè binaria. La regressione logistica è chiamata così perché utilizza la funzione logit (log-odds) per modellare la probabilità che un'osservazione appartenga a una delle due categorie della variabile dipendente. Questo modello appiattisce la funzione di regressione lineare utilizzando la funzione sigmoide.
Funzione della regressione lineare:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n$$

Funzione sigmoide:

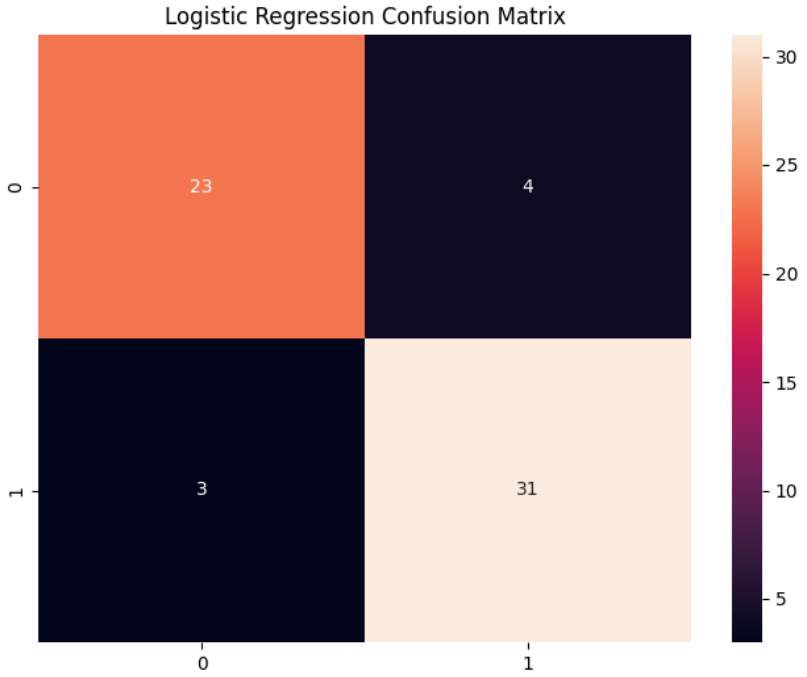
$$p = \frac{1}{1 + e^{-y}}$$

In questo caso specifico si sono ottenuti i seguenti risultati:

```
Logistic Regression
ACCURACY :  0.8852459016393442
F1       :  0.8985507246376812
PRECISION:  0.8857142857142857
RECALL   :  0.9117647058823529
```

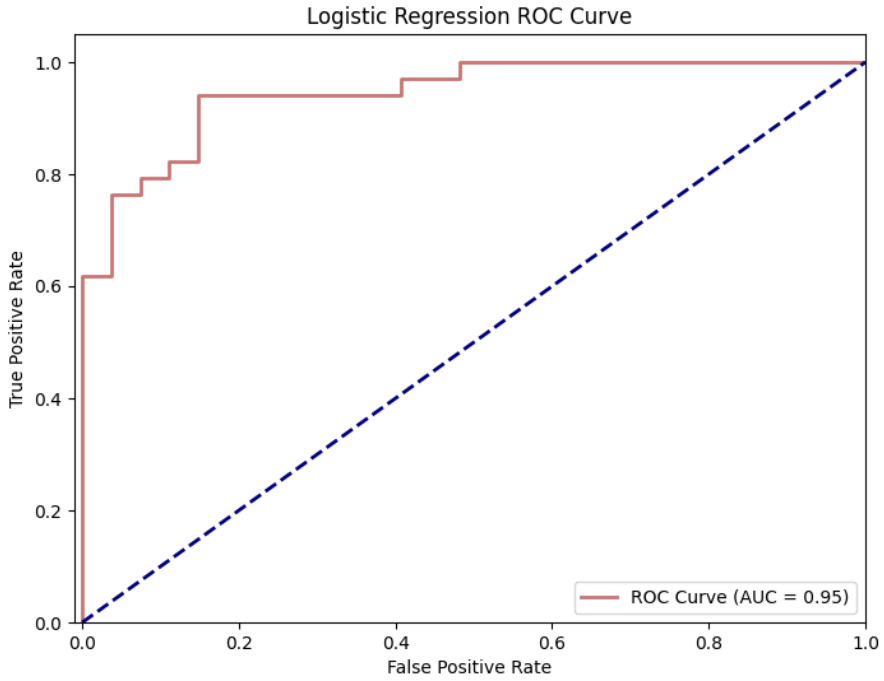
No Iperparametri

La matrice di confusione risultante è la seguente:



Questo modello ha previsto correttamente un problema cardiaco per 23 pazienti (true positive), mentre ha previsto erroneamente un problema cardiaco per 4 pazienti (false positive) che invece non lo hanno. Ha previsto correttamente che 31 persone (true negative) non hanno alcun problema cardiaco, mentre ha previsto erroneamente che 3 persone (false negative) non presentano un problema cardiaco.

La curva ROC e il valore AUC risultanti sono i seguenti:



Si può notare che la Logistic Regression è un buon modello di classificazione; infatti, il valore AUC è pari a 0,95.

Con Iperparametri

Per l'ottimizzazione del modello sono stati scelti i seguenti valori per gli iperparametri:

```
param_grid = [
    {'C' : [0.001, 0.01, 0.1, 1, 10, 100, 1000],'penalty' : ['l2'],'max_iter' : [100,1000,5000]},
    {'C' : [0.001, 0.01, 0.1, 1, 10, 100, 1000], 'penalty' : ['l1'],'max_iter' : [100,1000,5000]}
    ]

lg=LogisticRegression(solver='liblinear')
```

Solver: algoritmo utilizzato nei problemi di ottimizzazione *(default = 'lbfgs')*

C: parametro di regolarizzazione che controlla la forza della regolarizzazione *(default = 1.0)*

Penalty: parametro che specifica il tipo di regolarizzazione da utilizzare *(default = l2)*

- 'l1': ha una "funzione di selezione" incorporata che aiuta a selezionare le feature più rilevanti per il modello, riducendo il numero di feature utilizzate
- 'l2': non ha una funzione di selezione incorporata come L1, ma riduce l'ampiezza dei coefficienti, limitando così l'overfitting

Max_iter: parametro che definisce il numero massimo di iterazioni per la convergenza *(default = 100)*

In questo caso il solver utilizzato è il 'liblinear' perché è il più indicato per dataset relativamente piccoli, come quello scelto in questo caso di studio. Il solver 'liblinear' ha condizionato la scelta dei due valori per penalty: l1, l2.

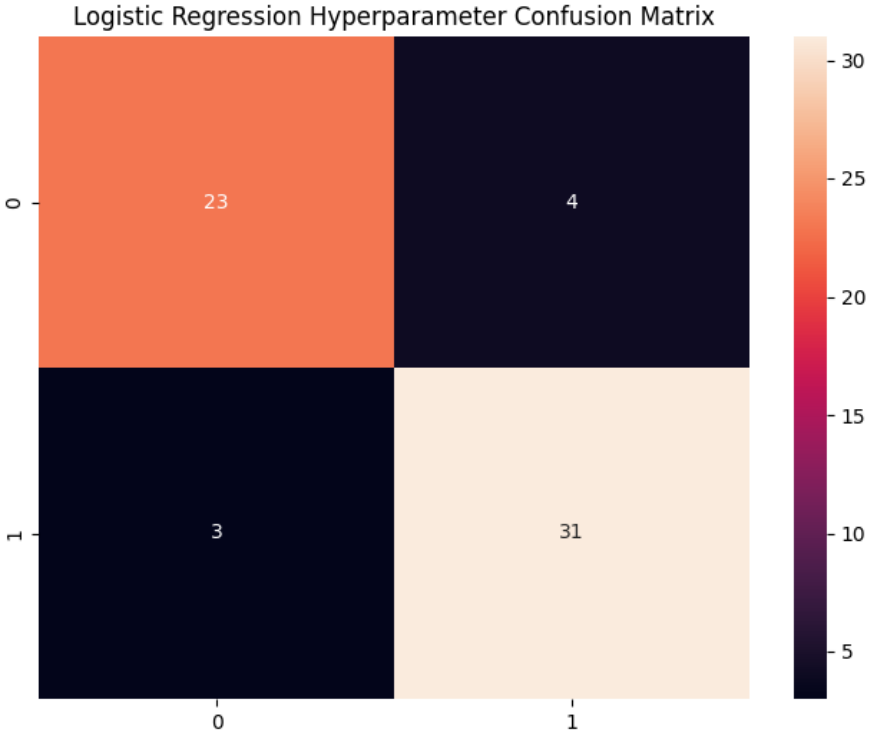
Al fine di selezionare i migliori iperparametri è stato utilizzato il GridSearchCV() che ha dato questi risultati:

```
Best hyperparameters Logistic Regression:  {'C': 1, 'max_iter': 100, 'penalty': 'l2'}
```

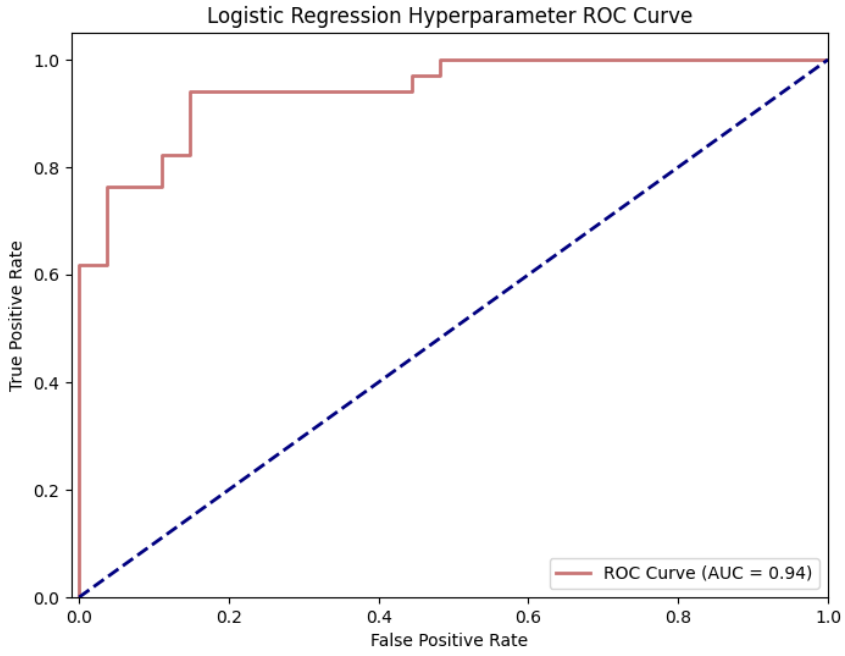
In questo caso specifico si sono ottenuti i seguenti risultati:

```
Logistic Regression Hyperparameter
ACCURACY :  0.8852459016393442
F1       :  0.8985507246376812
PRECISION:  0.8857142857142857
RECALL   :  0.9117647058823529
```

La matrice di confusione risultante è la seguente:



La curva ROC e il valore AUC risultanti sono i seguenti:

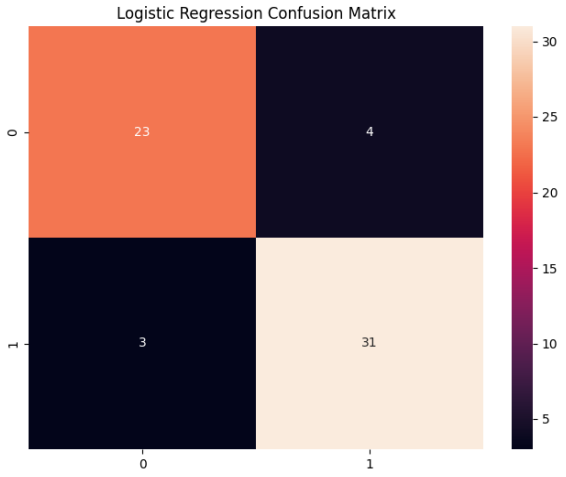


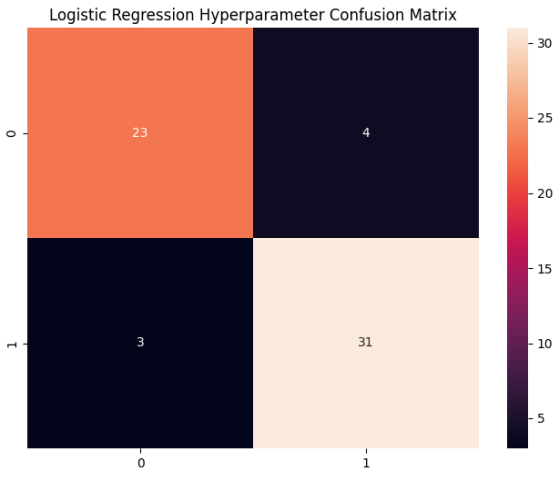
L'ottimizzazione del modello con iperparametri non ha portato miglioramenti notevoli, i risultati ottenuti sono pressoché simili.

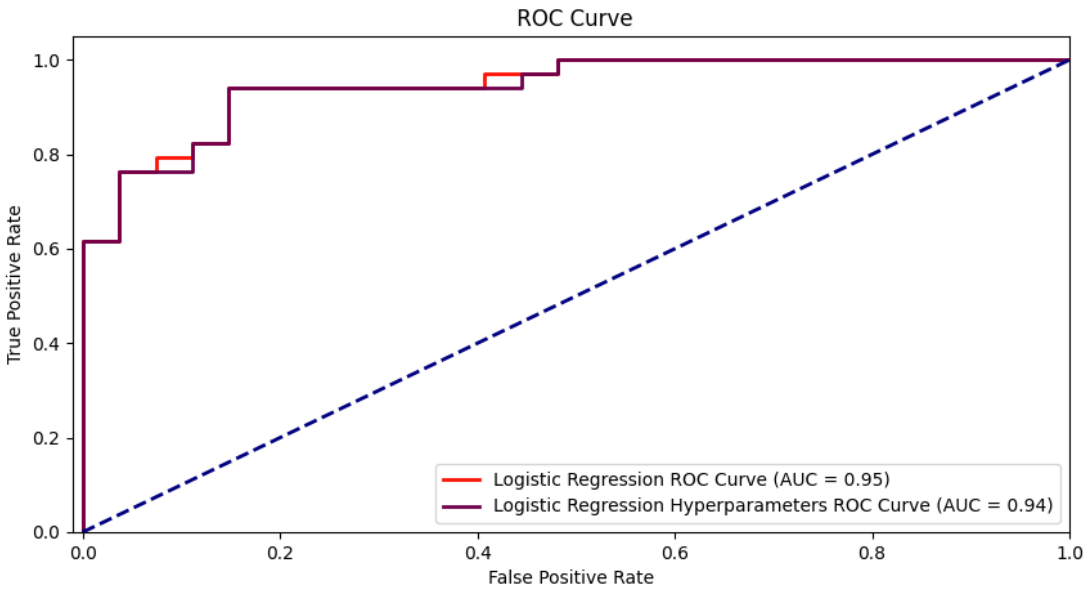
Grafici a confronto

```
Logistic Regression
ACCURACY :  0.8852459016393442
F1       :  0.8985507246376812
PRECISION:  0.8857142857142857
RECALL   :  0.9117647058823529
```

```
Logistic Regression Hyperparameter
ACCURACY :  0.8852459016393442
F1       :  0.8985507246376812
PRECISION:  0.8857142857142857
RECALL   :  0.9117647058823529
```







K-nearest neighbors

Il K-nearest neighbor, noto anche come algoritmo KNN, è un algoritmo che classifica i punti dati in base alla loro vicinanza e associazione con altri dati disponibili. Questo algoritmo presuppone che punti dati simili possano essere trovati uno vicino all'altro, 'k' indica appunto il numero di punti più vicini a quello da classificare, ed è un intero positivo. Inizialmente avviene la definizione del parametro k, poi viene calcolata la distanza tra punti dati. Quelle più comuni sono la distanza euclidea o la distanza di Manhattan e infine assegna una categoria basandosi sulla categoria più frequente.
Quindi il primo passo è la definizione del parametro k e in questo caso specifico il k è stato scelto applicando la Cross-Validation.

In particolare, in questo caso è stata utilizzata una 5-fold cross-validation

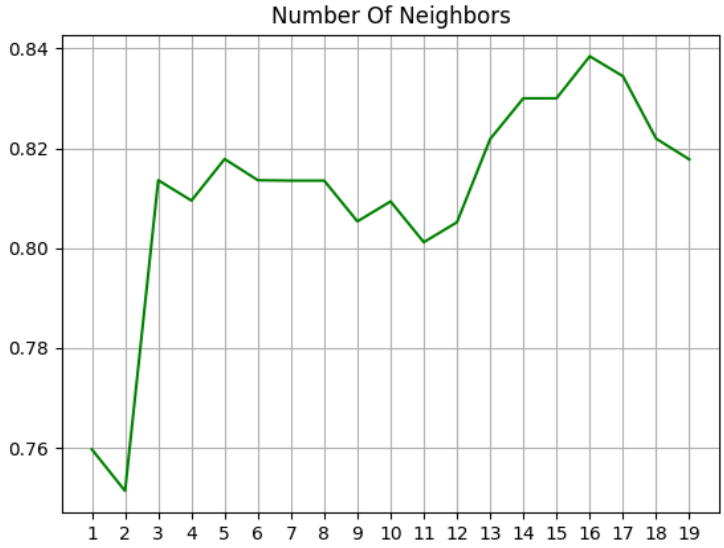
```
classifiers.append("KNearestNeighbors")
k_values = [i for i in range (1,20)]
scores = []

for k in k_values:
    knn = KNeighborsClassifier(n_neighbors=k)
    score = cross_val_score(knn, X_train, y_train, cv = 5)
    scores.append(np.mean(score))

plt.plot(k_values, scores, color='g')
plt.xticks(ticks=k_values, labels=k_values)
plt.title("Number Of Neighbors")
plt.grid()
plt.show()

best_index = np.argmax(scores)
best_k = k_values[best_index]
```

Il ciclo for viene utilizzato per iterare attraverso i valori di "k", si è scelto un range di valori tra 1 e 19, e per ciascun valore di "k", è stato creato un oggetto KNeighborsClassifier con quel valore specifico di "k". Quindi per ogni iterazione, è stato addestrato un classificatore KNN con il valore specifico di "k" e poi è stato valutato il modello su ciascuno dei 5 fold, le porzioni di train set e test set sono differenti ad ogni iterazione. Per ciascuna iterazione, è stato calcolatolo lo score, in questo caso è stata utilizzata come metrica l'accuracy. La media di questi score è stata memorizzata nella lista scores. Alla fine del ciclo for, è stato determinato il valore di "k" che ha ottenuto il punteggio medio più alto.

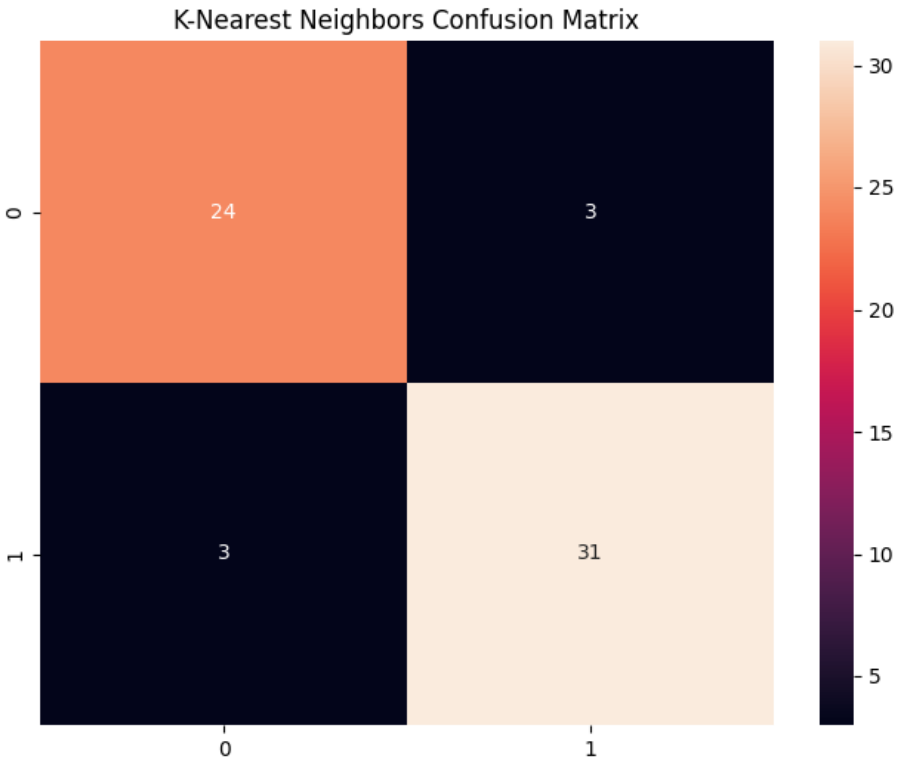


In questo caso è stato scelto k = 16.
Si sono ottenuti i seguenti risultati:

```
K-Nearest Neighbors
ACCURACY :  0.9016393442622951
F1       :  0.9117647058823528
PRECISION:  0.9117647058823529
RECALL   :  0.9117647058823529
```

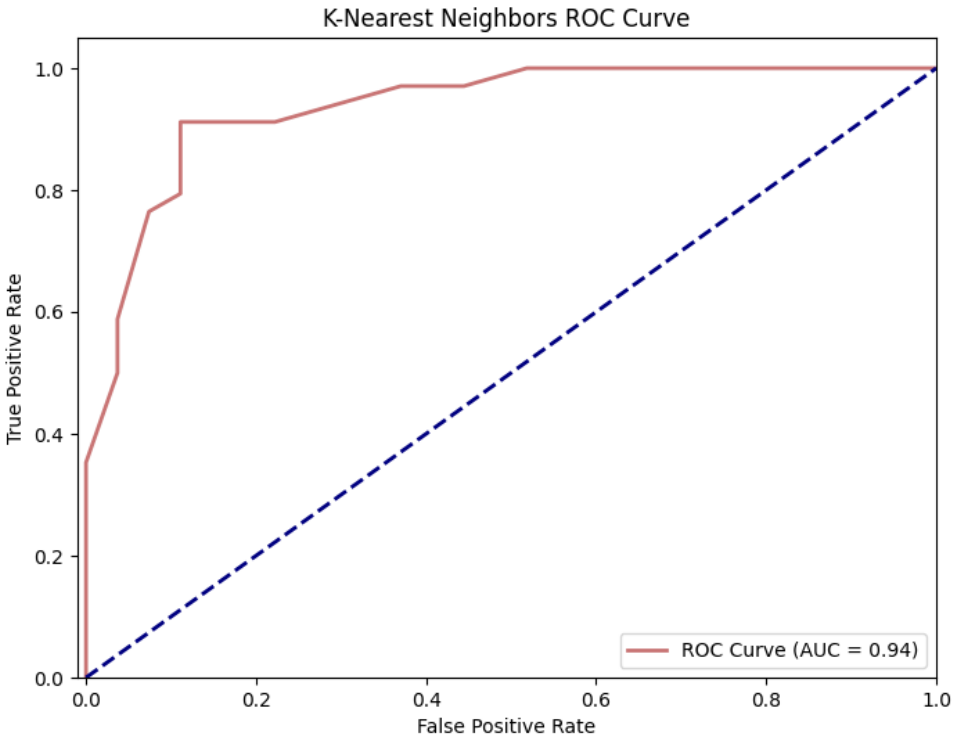
No Iperparametri

La matrice di confusione risultante è la seguente:



Questo modello ha previsto correttamente un problema cardiaco per 24 pazienti (true positive), mentre ha previsto erroneamente un problema cardiaco per 3 pazienti (false positive) che invece non lo hanno. Ha previsto correttamente che 31 persone (true negative) non hanno alcun problema cardiaco, mentre ha previsto erroneamente che 3 persone (false negative) non presentano un problema cardiaco.

La curva ROC e il valore AUC risultanti sono i seguenti:



Si può notare che il K-Nearest Neighbors è un buon modello di classificazione; infatti, il valore AUC è pari a 0,94.

Con Iperparametri

Per l'ottimizzazione del modello sono stati scelti i seguenti valori per gli iperparametri:

```
param_grid = {
    'weights':['uniform','distance'],
    'algorithm':['auto','ball_tree','kd_tree','brute'],
    'leaf_size':[10,20,30,40,50],
    'p':[1,2],
    'metric': ['euclidean','manhattan','minkowski'],
}
```

Weights: specifica il tipo di peso da assegnare ai vicini durante il calcolo delle predizioni *(default ='uniform')*

- 'uniform': pesi uniformi. Tutti i vicini hanno lo stesso peso.
- 'distance': pesa i punti in base all'inverso della loro distanza. In questo caso, i vicini più prossimi avranno un'influenza maggiore rispetto ai vicini più lontani.

Algorithm: specifica l'algoritmo da utilizzare per trovare i vicini più vicini (*default = 'auto'*)

- 'auto' : cercherà di selezionare l'algoritmo più adatto in modo automatico, in base al dataset e agli altri iperparametri forniti.
- 'ball_tree' : Utilizzerà l'algoritmo BallTree
- 'kd_tree' : Utilizzerà l'algoritmo KDTree
- 'brute' : Utilizzerà una ricerca in forza bruta

Leaf_size: specifica la dimensione delle foglie passata a BallTree o KDTree *(default = 30)*

P: specifica la metrica di distanza da usare *(default = 2)*.

- '1': distanza di Manhattan
- '2': distanza euclidea

Metric: specifica la metrica di distanza da utilizzare *(default = minkwski)*.

- 'euclidean'
- 'manhattan'
- 'minkowski'

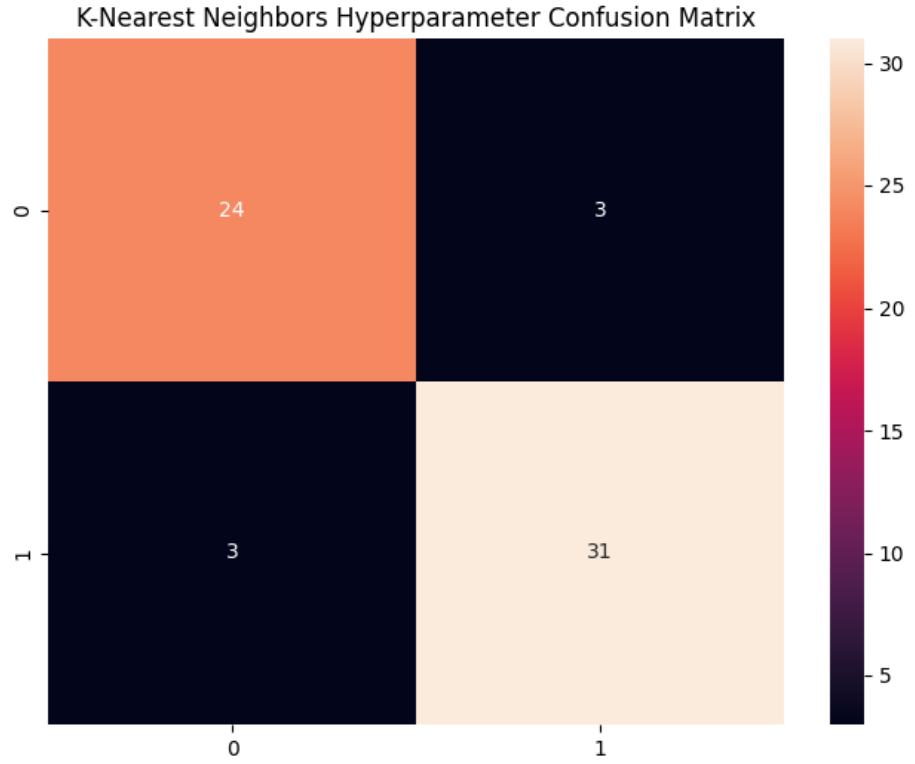
Al fine di selezionare i migliori iperparametri è stato utilizzato il GridSearchCV() che ha dato questi risultati:

```
Best hyperparameters K-Nearest Neighbors:  {'algorithm': 'auto', 'leaf_size': 10, 'metric': 'euclidean', 'p': 1, 'weights': 'uniform'}
```

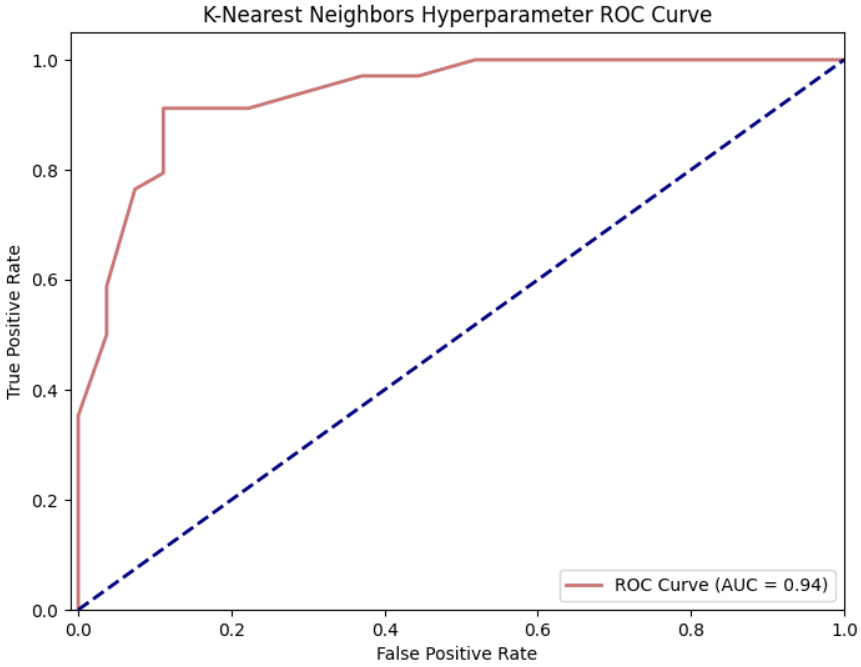
In questo caso specifico si sono ottenuti i seguenti risultati

```
K-Nearest Neighbors Hyperparameter
ACCURACY :  0.9016393442622951
F1       :  0.9117647058823528
PRECISION:  0.9117647058823529
RECALL   :  0.9117647058823529
```

La matrice di confusione risultante è la seguente:



La curva ROC e il valore AUC risultanti sono i seguenti:

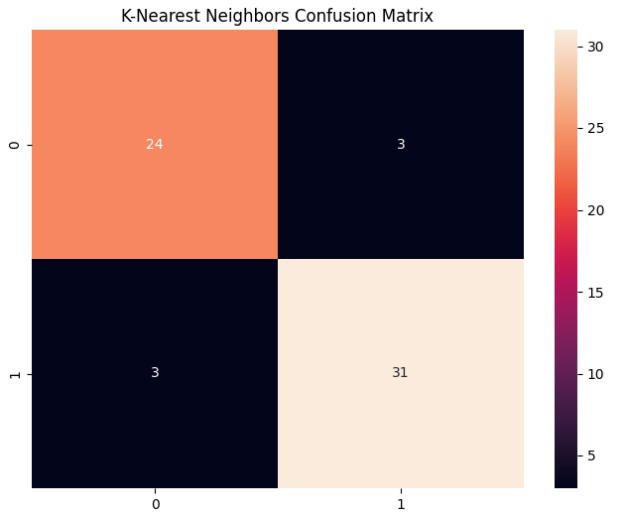


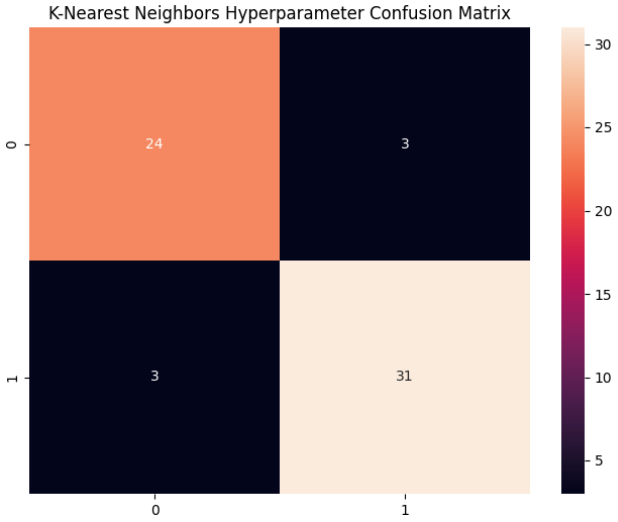
L'ottimizzazione del modello con iperparametri non ha portato miglioramenti, i risultati ottenuti sono uguali.

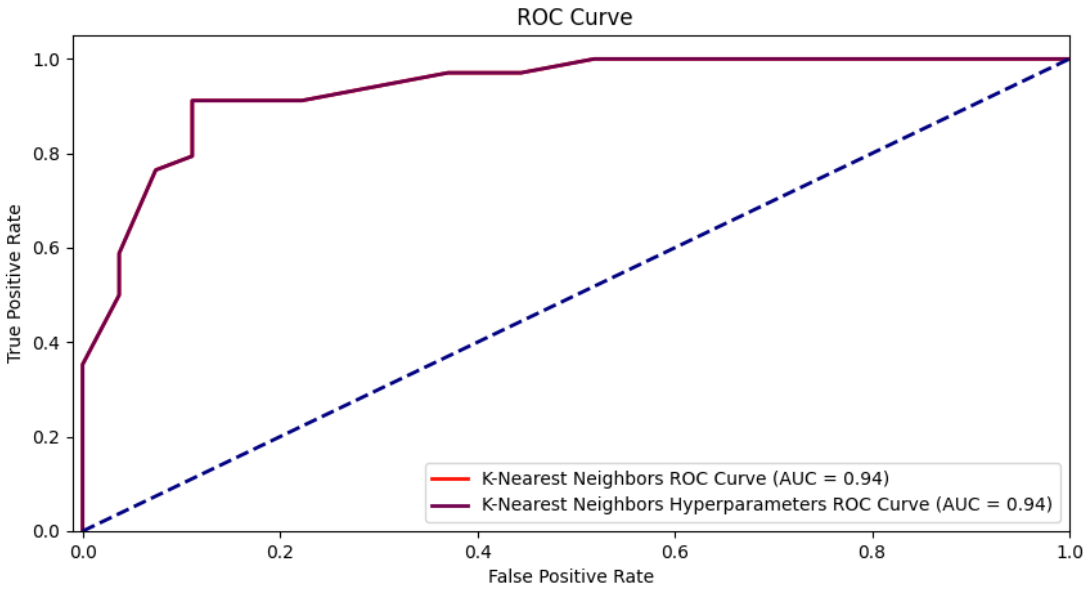
Grafici a confronto

```
K-Nearest Neighbors
ACCURACY :  0.9016393442622951
F1       :  0.9117647058823528
PRECISION:  0.9117647058823529
RECALL   :  0.9117647058823529
```

```
K-Nearest Neighbors Hyperparameter
ACCURACY :  0.9016393442622951
F1       :  0.9117647058823528
PRECISION:  0.9117647058823529
RECALL   :  0.9117647058823529
```







Support vector machines

Il Support Vector Machine (SVM) fa parte degli algoritmi di apprendimento. Ogni istanza di addestramento è rappresentata come un vettore n-dimensionale, dove n rappresenta il numero di caratteristiche o attributi. L'obiettivo dell'algoritmo è trovare l'iperpiano che massimizza il margine tra le due classi; quindi, l'obiettivo è creare la migliore linea o confine decisionale in grado di separare lo spazio n-dimensionale in classi, in modo da poter facilmente inserire il nuovo punto dati nella categoria corretta. Dopo aver trovato l'iperpiano ottimale, è possibile utilizzarlo per classificare nuovi dati (nel caso di classificazione) o prevedere valori (nel caso di regressione) assegnando loro una classe o un valore in base a quale lato dell'iperpiano si trovino.
Le Support Vector Machines (SVM) possono essere classificate in due categorie principali: SVM lineari e SVM non lineari.

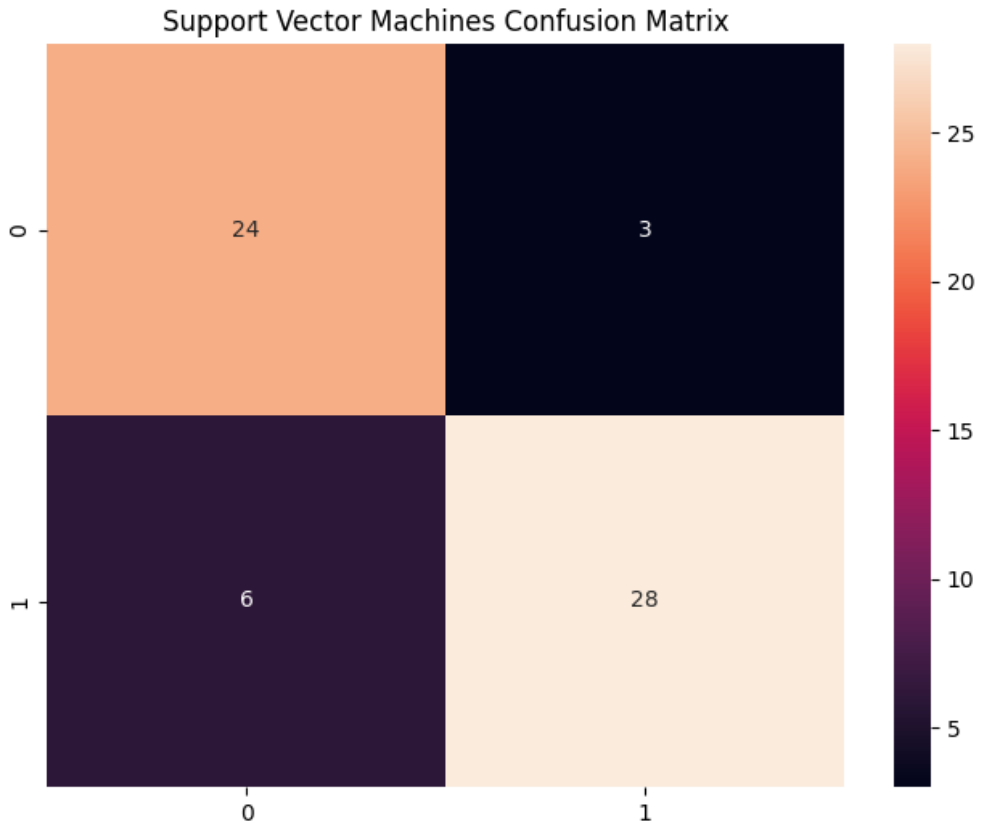
SVM lineari:
Le SVM lineari cercano di trovare un iperpiano lineare che separa chiaramente i punti dati in diverse classi e sono efficaci quando i dati sono linearmente separabili, il che significa che è possibile tracciare una linea retta o un piano per separare chiaramente le classi.

SVM non lineare:
Le SVM non lineari affrontano il problema dei dati che non sono linearmente separabili. In caso di dati non linearmente separabili, gli SVM possono utilizzare una funzione kernel per trasformare i dati in uno spazio di dimensioni superiori in cui possono diventare separabili linearmente.
In questo caso specifico si sono ottenuti i seguenti risultati

```
Support Vector Machines
ACCURACY :  0.8524590163934426
F1       :  0.8615384615384616
PRECISION:  0.9032258064516129
RECALL   :  0.8235294117647058
```

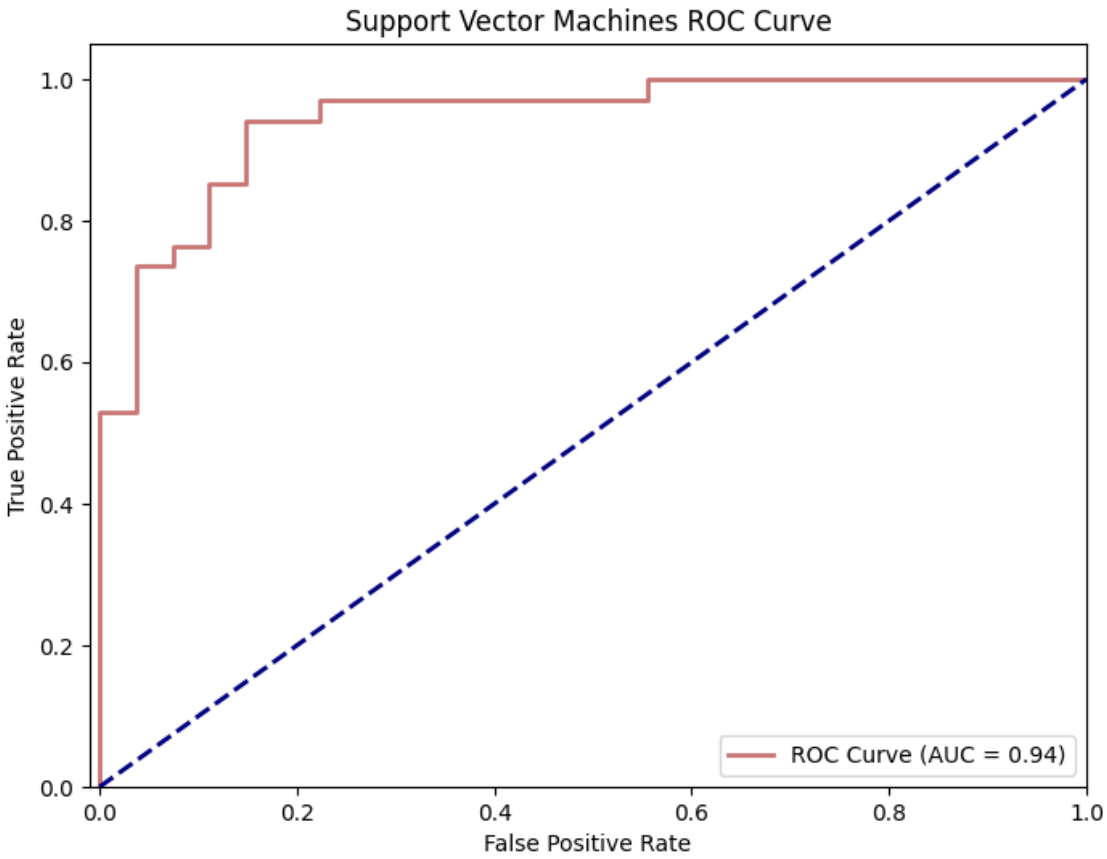
No Iperparametri

La matrice di confusione risultante è la seguente:



Questo modello ha previsto correttamente un problema cardiaco per 24 pazienti (true positive), mentre ha previsto erroneamente un problema cardiaco per 3 pazienti (false positive) che invece non lo hanno. Ha previsto correttamente che 28 persone (true negative) non hanno alcun problema cardiaco, mentre ha previsto erroneamente che 6 persone (false negative) non presentano un problema cardiaco.

La curva ROC e il valore AUC risultanti sono i seguenti:



Si può notare che il Support Vector Machines è un buon modello di classificazione; infatti, il valore AUC è pari a 0,94.

Con Iperparametri

Per l'ottimizzazione del modello sono stati scelti i seguenti valori per gli iperparametri:

```
parameters = [ {'C':[1, 10, 100, 1000], 'kernel':['linear']},
               {'C':[1, 10, 100, 1000], 'kernel':['rbf'], 'gamma':[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]},
               {'C':[1, 10, 100, 1000], 'kernel':['poly'], 'degree': [2,3,4] ,'gamma':[0.01,0.02,0.03,0.04,0.05]}
             ]
svc=SVC(probability=True)
```

C: Parametro di regolarizzazione. La forza della regolarizzazione è inversamente proporzionale a C. Deve essere strettamente positiva

Kernel: Specifica il tipo di kernel da utilizzare nell'algoritmo *(default='rbf')*

- 'Linear'
- 'Poly'
- 'Rbf'

Gamma: Coefficiente del kernel per 'rbf', 'poly' *(default= 'scale')*

Probability: *(default= 'False')*

- 'True': indica che si desidera essere in grado di ottenere stime di probabilità dal modello addestrato

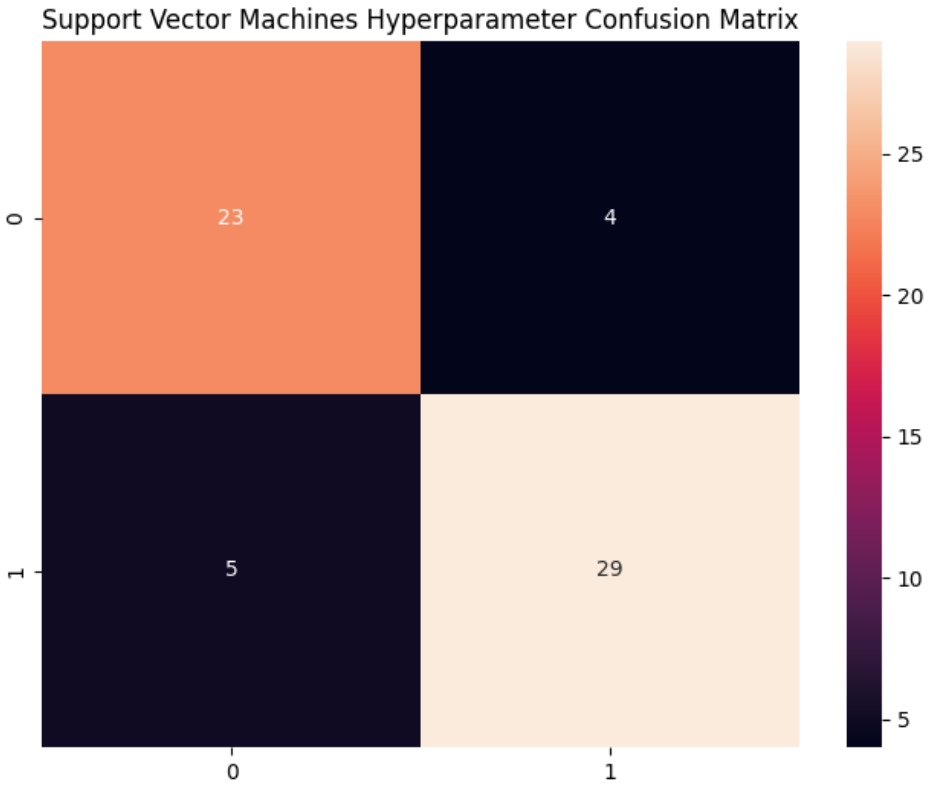
Al fine di selezionare i migliori iperparametri è stato utilizzato il GridSearchCV() che ha dato questi risultati:

```
Best hyperparameters Support Vector Machines:  {'C': 1, 'kernel': 'linear'}
```

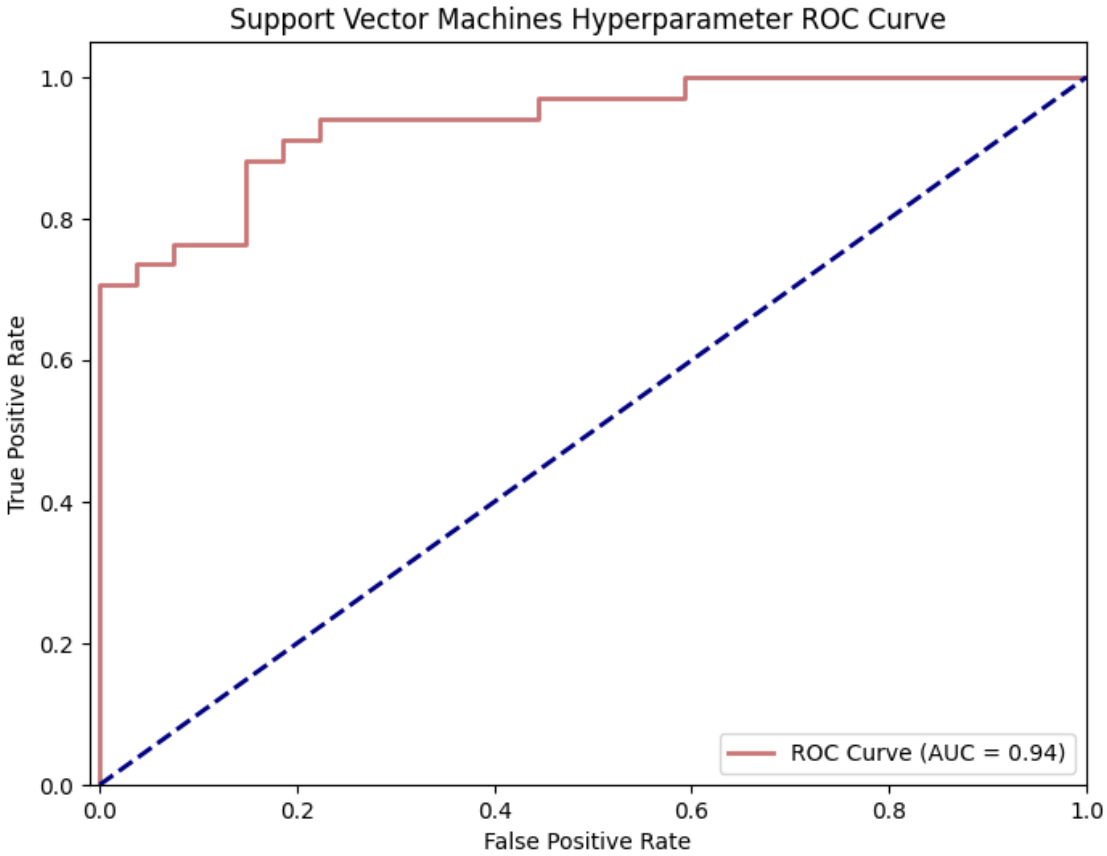
In questo caso specifico si sono ottenuti i seguenti risultati:

```
Support Vector Machines Hyperparameter
ACCURACY :  0.8524590163934426
F1       :  0.8656716417910447
PRECISION:  0.8787878787878788
RECALL   :  0.8529411764705882
```

La matrice di confusione risultante è la seguente:



La curva ROC e il valore AUC risultanti sono i seguenti:

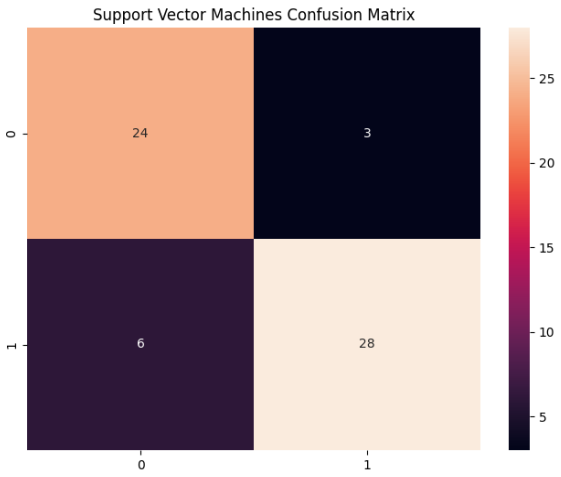


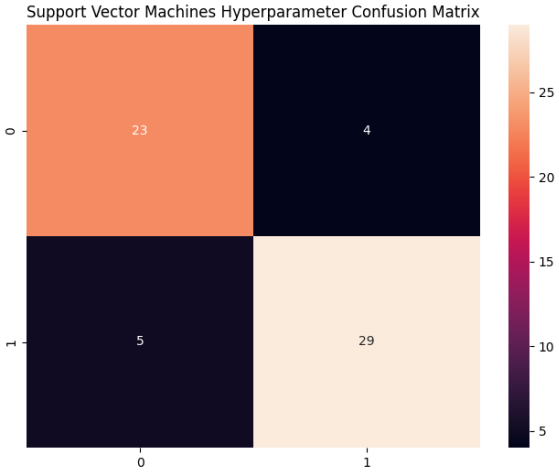
L'ottimizzazione del modello con iperparametri non ha portato miglioramenti notevoli, i risultati ottenuti sono pressoché simili.

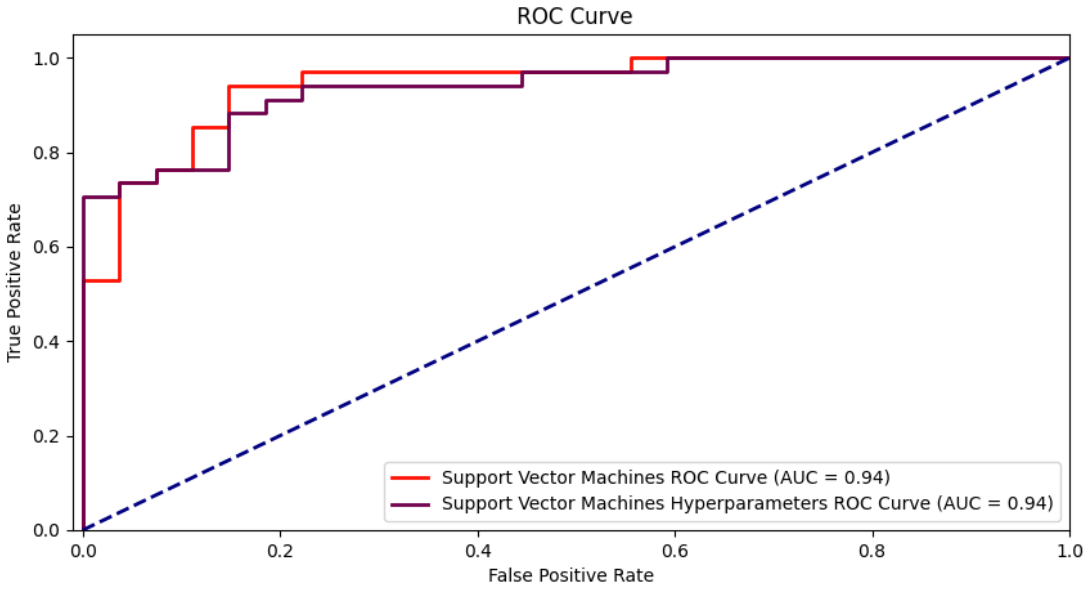
Grafici a confronto

```
Support Vector Machines
ACCURACY :  0.8524590163934426
F1       :  0.8615384615384616
PRECISION:  0.9032258064516129
RECALL   :  0.8235294117647058
```

```
Support Vector Machines Hyperparameter
ACCURACY :  0.8524590163934426
F1       :  0.8656716417910447
PRECISION:  0.8787878787878788
RECALL   :  0.8529411764705882
```







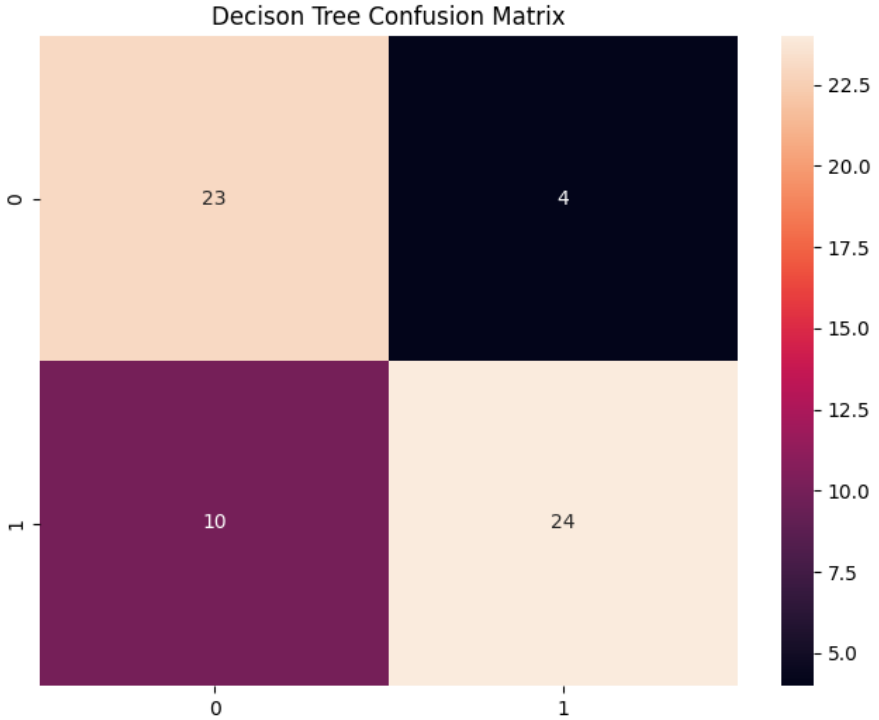
Decison tree

Il decision tree è un modello di apprendimento automatico utilizzato per la classificazione e la regressione. L'obiettivo di un albero decisionale è suddividere un set di dati basandosi su regole decisionali derivate dai dati stessi. Questo processo crea una struttura a forma di albero dove i nodi foglia rappresentano le classi per problemi di classificazione o valori numerici per problemi di regressione. I nodi foglia sono etichettati con la classe o il valore previsti.
In questo caso specifico si sono ottenuti i seguenti risultati:

```
Decision Tree
ACCURACY :  0.7704918032786885
F1       :  0.7741935483870968
PRECISION:  0.8571428571428571
RECALL   :  0.7058823529411765
```

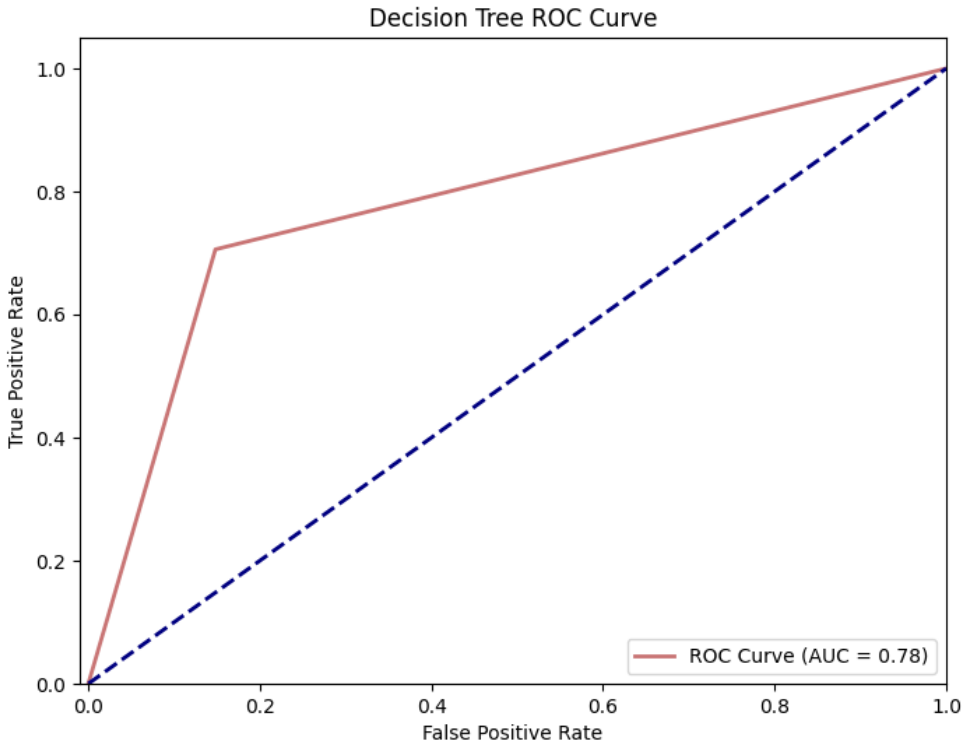
No Iperparametri

La matrice di confusione risultante è la seguente:



Questo modello ha previsto correttamente un problema cardiaco per 23 pazienti (true positive), mentre ha previsto erroneamente un problema cardiaco per 4 pazienti (false positive) che invece non lo hanno. Ha previsto correttamente che 24 persone (true negative) non hanno alcun problema cardiaco, mentre ha previsto erroneamente che 10 persone (false negative) non presentano un problema cardiaco.

La curva ROC e il valore AUC risultanti sono i seguenti:



Si può notare che il Decision Tree è un discreto modello di classificazione; infatti, il valore AUC è pari a 0,78

Con Iperparametri

Per l'ottimizzazione del modello sono stati scelti i seguenti valori per gli iperparametri:

```
param_grid = {
    'criterion' : ['gini', 'entropy','log_loss'],
    'splitter': ['best','random'],
    'max_depth': [10, 20, 30, 40, 50],
    'min_samples_split': [2, 5, 10, 20],
    'min_samples_leaf': [1, 2, 4, 7]
}
```

Criterion: determina la funzione di misura della qualità delle divisioni fatte nell'albero decisionale *(default = 'gini')*

- 'gini': misura l'impurità dei nodi utilizzando l'indice di Gini. Misura l'impurità calcolando la probabilità che un campione estratto casualmente da un nodo sia etichettato in modo scorretto. Un indice di Gini più alto indica una maggiore impurità

- 'entropia': misura l'impurità utilizzando l'entropia, è una misura della casualità o dell'incertezza all'interno di un insieme di dati
- 'log_loss': entropia logaritmica

Splitter: strategia utilizzata per scegliere la suddivisione in ciascun nodo *(default = 'best')*

- 'best': per scegliere la migliore suddivisione
- 'random': per scegliere la migliore suddivisione casuale.

Max_depth: valore che rappresenta la profondità massima dell'albero *(default = 0)*

Min_samples_split: Il numero minimo di campioni richiesti per dividere un nodo interno *(default = 2)*

Min_samples_leaf: Il numero minimo di campioni richiesti per trovarsi in un nodo foglia *(default = 1)*

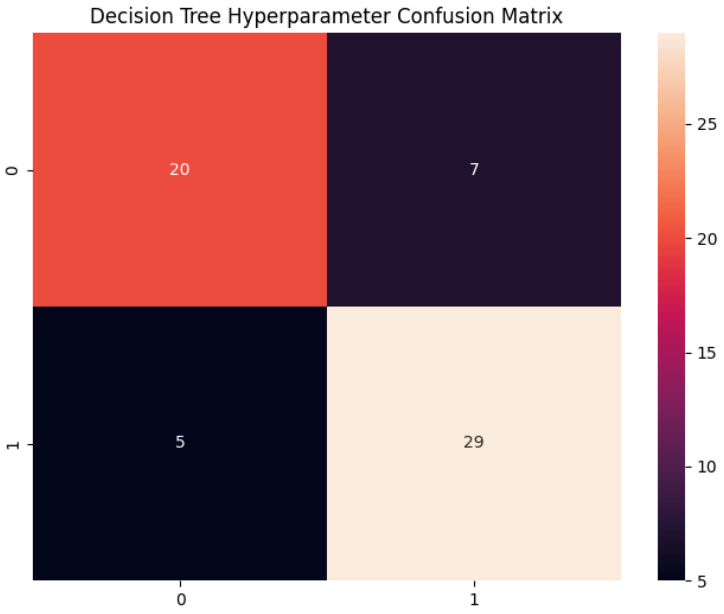
A fine di selezionare i migliori iperparametri è stato utilizzato il GridSearchCV() che ha dato questi risultati:

```
Best hyperparameters Decision Tree:  {'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 7, 'min_samples_split': 20, 'splitter': 'random'}
```

In questo caso specifico si sono ottenuti i seguenti risultati:

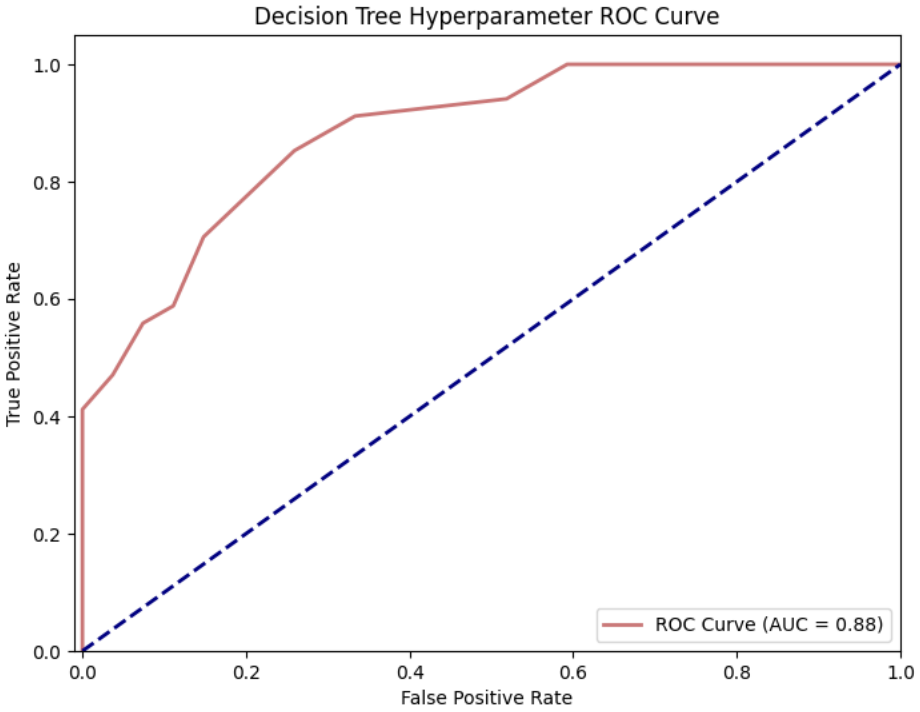
```
Decision Tree Hyperparameter
ACCURACY :  0.8032786885245902
F1       :  0.8285714285714286
PRECISION:  0.8055555555555556
RECALL   :  0.8529411764705882
```

La matrice di confusione risultante è la seguente:



Si può notare che la matrice di confusione, a differenza della precedente, ha subito dei cambiamenti. Un cambiamento notevole si può notare nella predizione dei False Negative e dei True Negative, rispettivamente 5 e 29 (prima dell'ottimizzazione: 10 e 24).

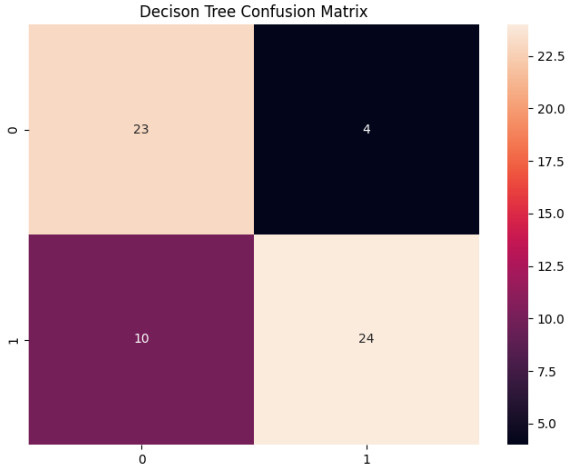
La curva ROC e il valore AUC risultanti sono i seguenti:

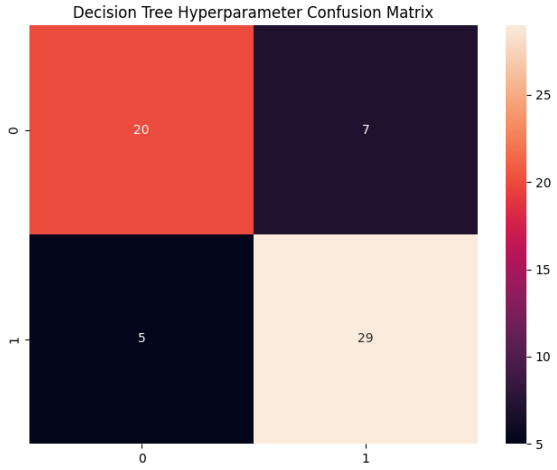


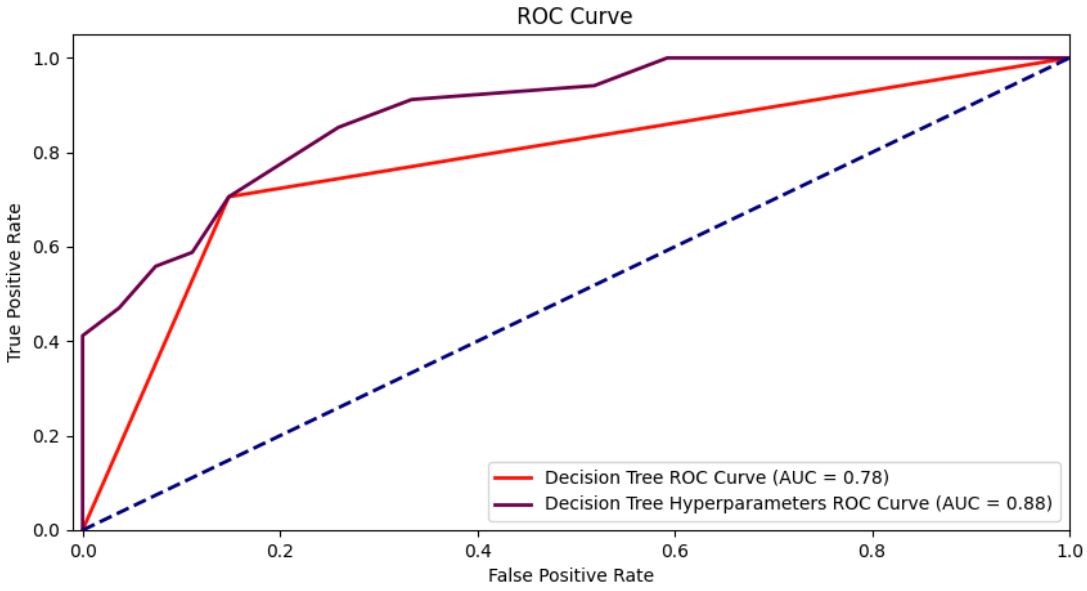
Grafici a confronto

```
Decision Tree
ACCURACY :  0.7704918032786885
F1       :  0.7741935483870968
PRECISION:  0.8571428571428571
RECALL   :  0.7058823529411765
```

```
Decision Tree Hyperparameter
ACCURACY :  0.8032786885245902
F1       :  0.8285714285714286
PRECISION:  0.8055555555555556
RECALL   :  0.8529411764705882
```







Grazie alla curva ROC si può notare il notevole incremento del modello Decision Tree ottenuto ottimizzando gli iperparametri.

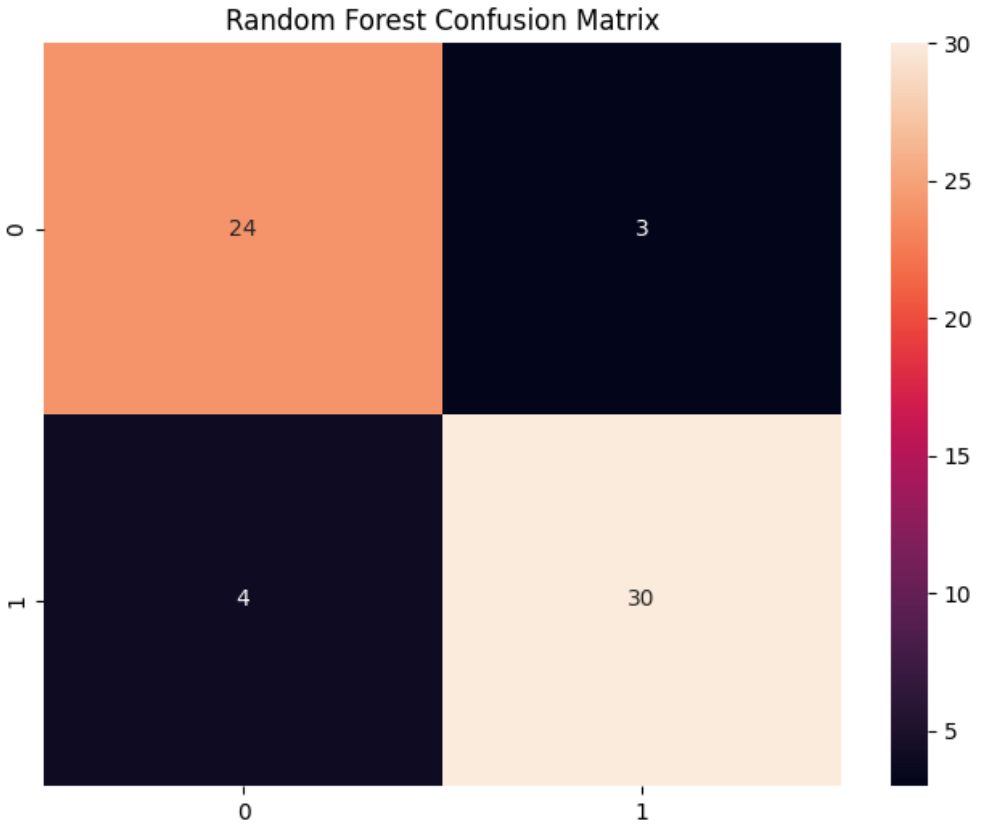
Random forest

Random Forest è un algoritmo di machine learning basato su alberi che sfrutta la potenza di più alberi decisionali per prendere decisioni. L'algoritmo costruisce un gran numero di alberi decisionali individuali, su diversi sottoinsiemi del dataset originale, che operano come un insieme. Ogni singolo albero nel Random Forest fa una previsione di classe e la classe che è stata predetta maggiormente diventa il risultato che dà il modello.
In questo caso specifico si sono ottenuti i seguenti risultati

```
Random Forest
ACCURACY :  0.8852459016393442
F1       :  0.8955223880597014
PRECISION:  0.9090909090909091
RECALL   :  0.8823529411764706
```

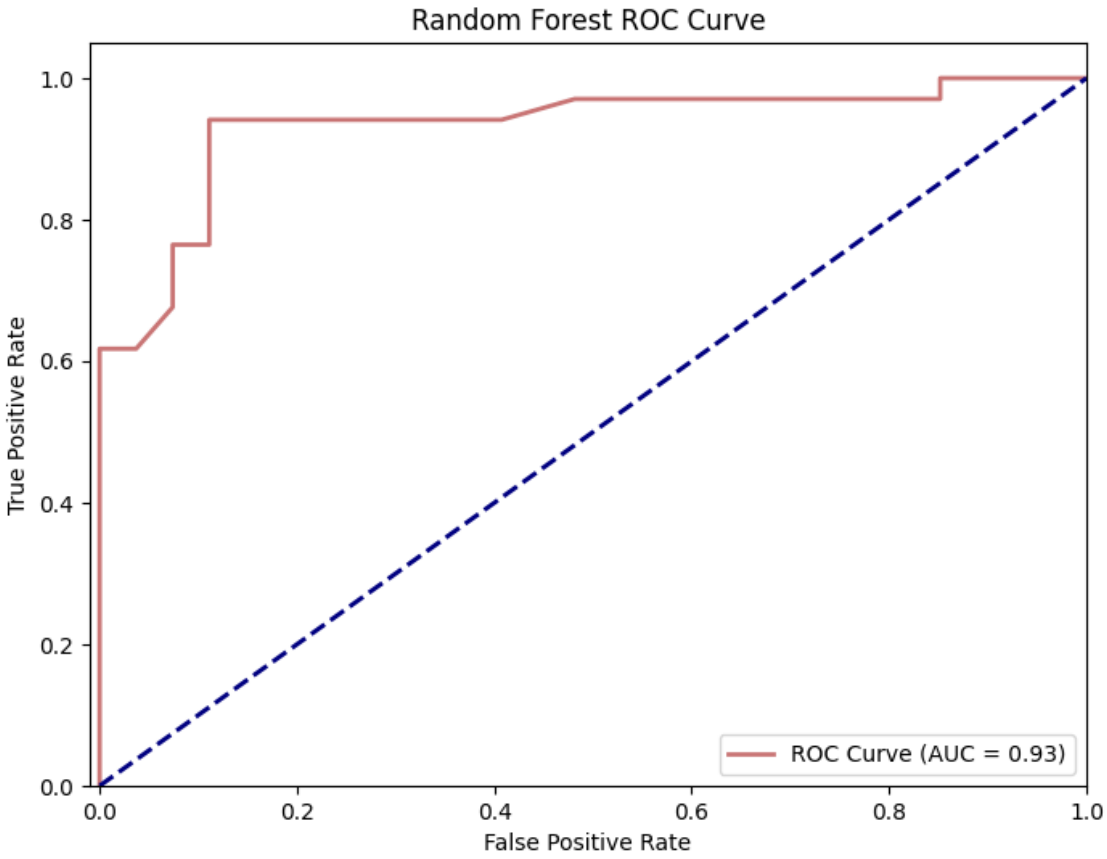
No Iperparametri

La matrice di confusione risultante è la seguente:



Questo modello ha previsto correttamente un problema cardiaco per 24 pazienti (true positive), mentre ha previsto erroneamente un problema cardiaco per 3 pazienti (false positive) che invece non lo hanno. Ha previsto correttamente che 30 persone (true negative) non hanno alcun problema cardiaco, mentre ha previsto erroneamente che 4 persone (false negative) non presentano un problema cardiaco.

La curva ROC e il valore AUC risultanti sono i seguenti:



Si può notare che il Random Forest è un buon modello di classificazione; infatti, il valore AUC è pari a 0,93

Con Iperparametri

Per l'ottimizzazione del modello sono stati scelti i seguenti valori per gli iperparametri:

```
param_grid = { 'criterion' : ['gini', 'entropy','log_loss'],
               'n_estimators': [25, 50, 75, 100, 150, 200, 250] }
```

Criterion: determina la funzione di misura della qualità delle divisioni fatte nell'albero decisionale *(default = 'gini')*

- 'gini': misura l'impurità dei nodi utilizzando l'indice di Gini. Misura l'impurità calcolando la probabilità che un campione estratto casualmente da un nodo sia etichettato in modo scorretto. Un indice di Gini più alto indica una maggiore impurità
- 'entropia': misura l'impurità utilizzando l'entropia, è una misura della casualità o dell'incertezza all'interno di un insieme di dati
- 'log_loss': entropia logaritmica

N_estimators: indica il numero di alberi nella foresta *(default = 100)*

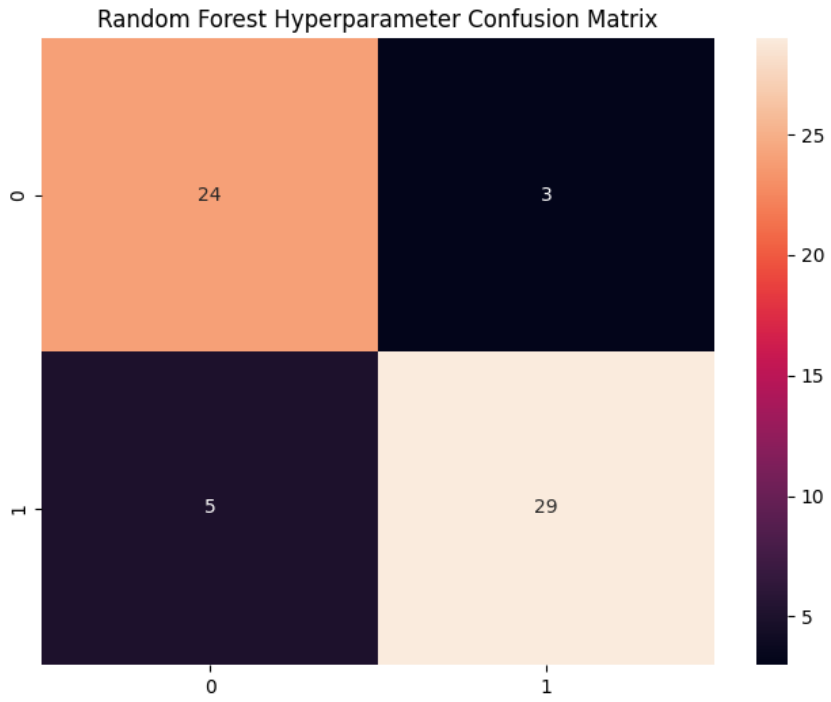
Al fine di selezionare i migliori iperparametri è stato utilizzato il GridSearchCV() che ha dato questi risultati:

```
Best hyperparameters Random Forest:  {'criterion': 'log_loss', 'n_estimators': 100}
```

In questo caso specifico si sono ottenuti i seguenti risultati:

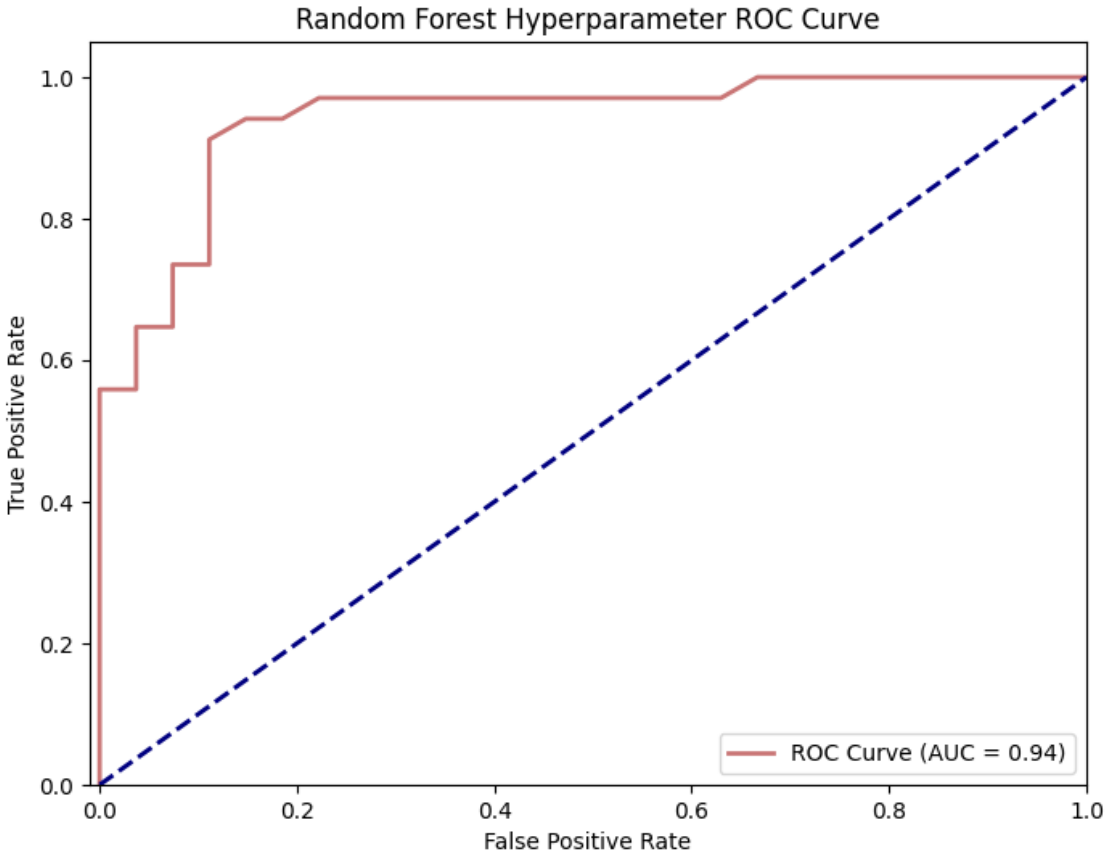
```
Random Forest Hyperparameter
ACCURACY :  0.8688524590163934
F1       :  0.8787878787878787
PRECISION:  0.90625
RECALL   :  0.8529411764705882
```

La matrice di confusione risultante è la seguente:



L’ottimizzazione del modello con iperparametri non ha portato miglioramenti notevoli, i risultati ottenuti sono pressoché simili.

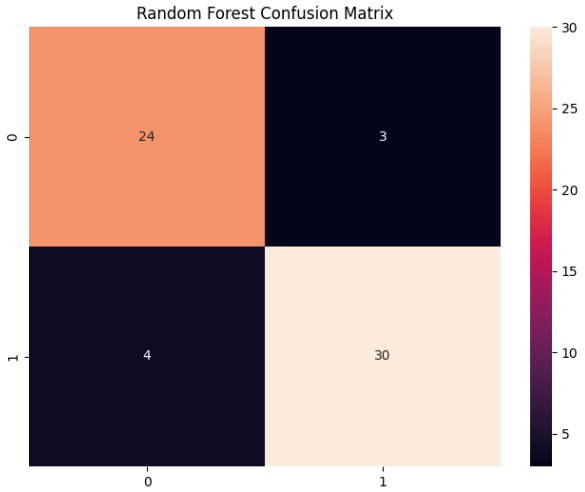
La curva ROC e il valore AUC risultanti sono i seguenti:

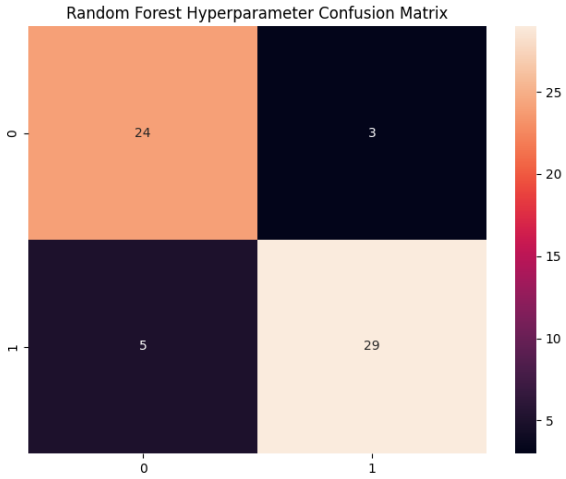


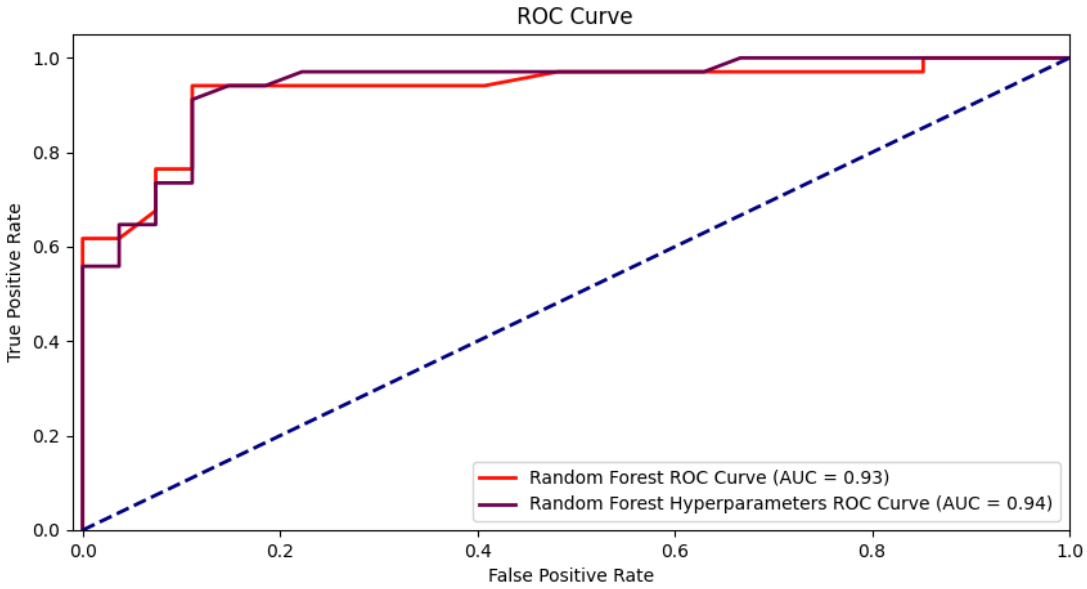
Grafici a confronto

```
Random Forest
ACCURACY :  0.8852459016393442
F1       :  0.8955223880597014
PRECISION:  0.9090909090909091
RECALL   :  0.8823529411764706
```

```
Random Forest Hyperparameter
ACCURACY :  0.8688524590163934
F1       :  0.8787878787878787
PRECISION:  0.90625
RECALL   :  0.8529411764705882
```





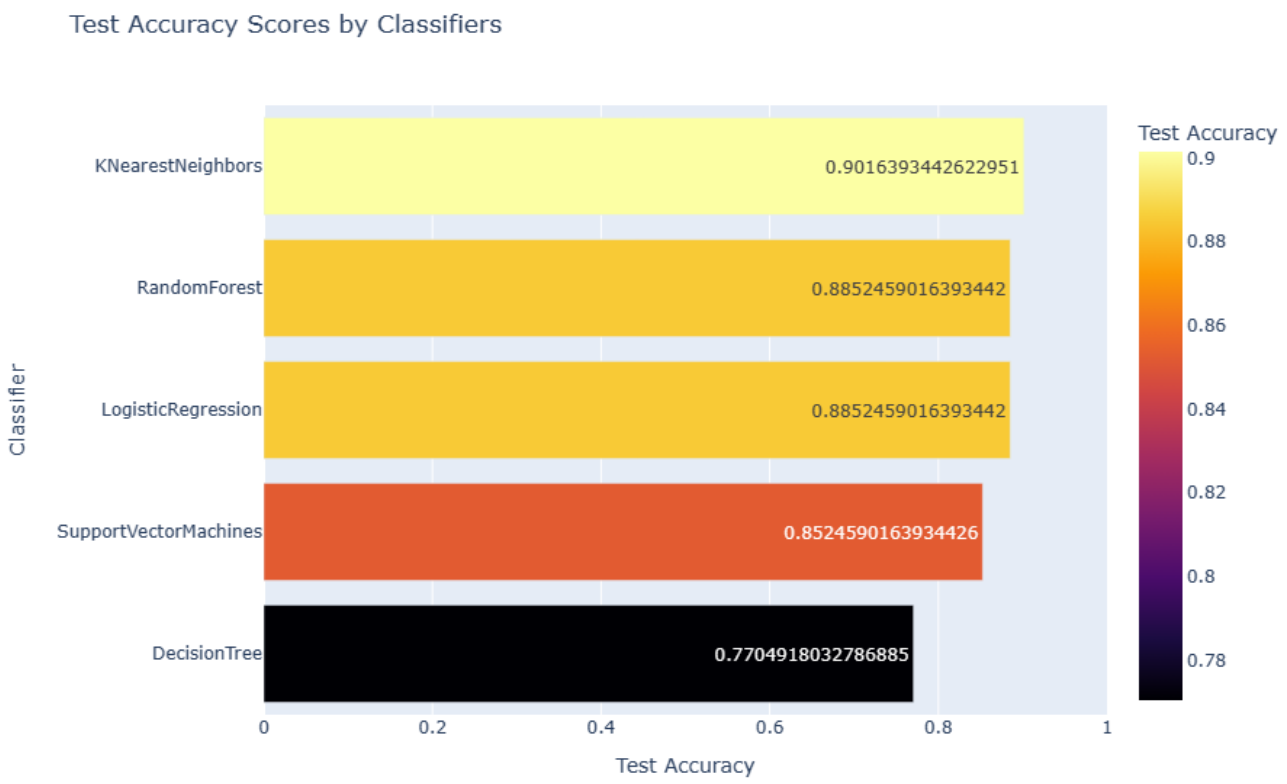


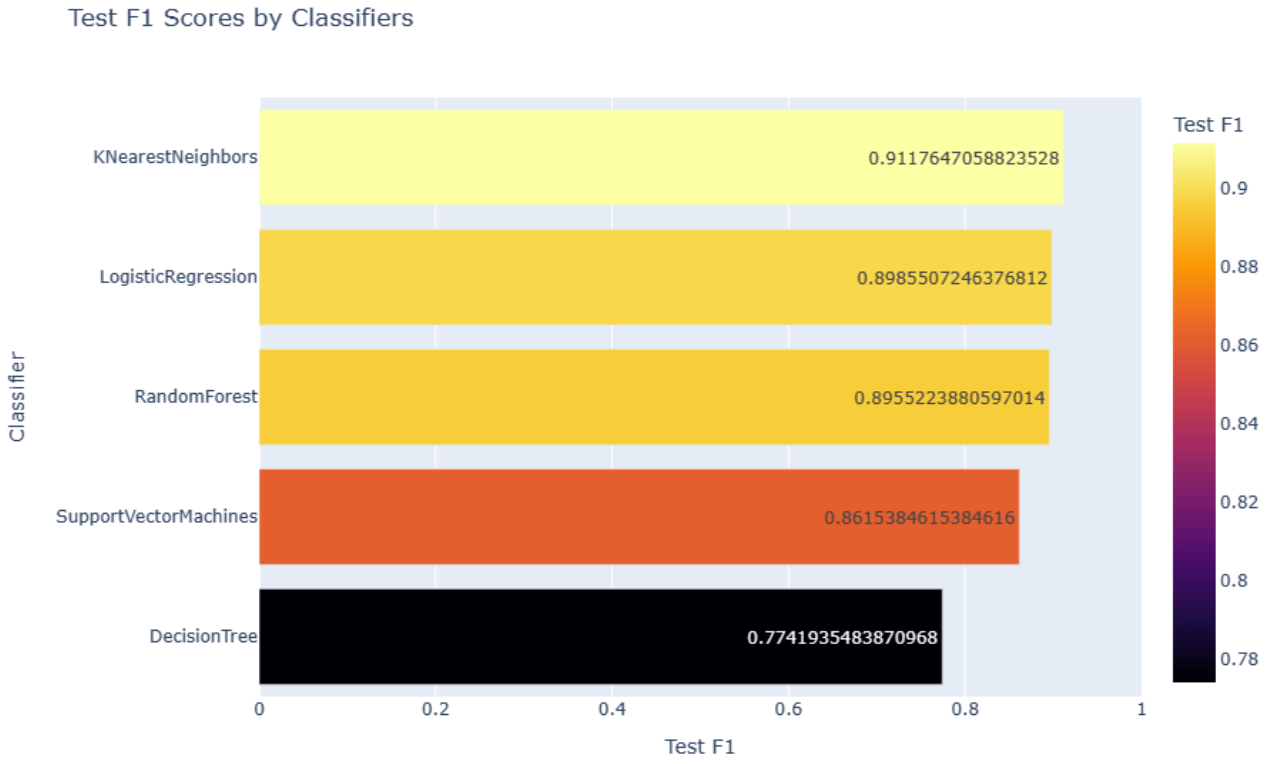
Grazie alla curva ROC si può notare un leggero incremento del modello Random Forest ottenuto ottimizzando gli iperparametri.

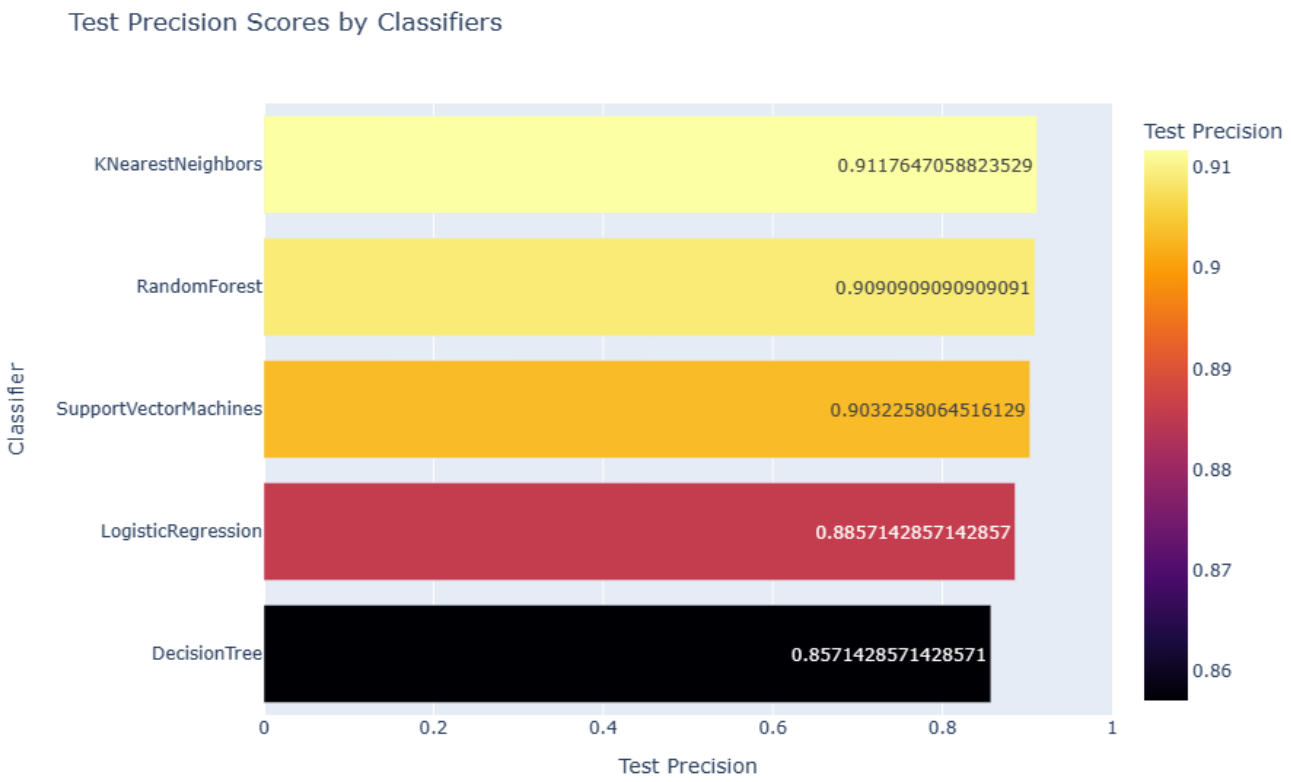
Conclusioni

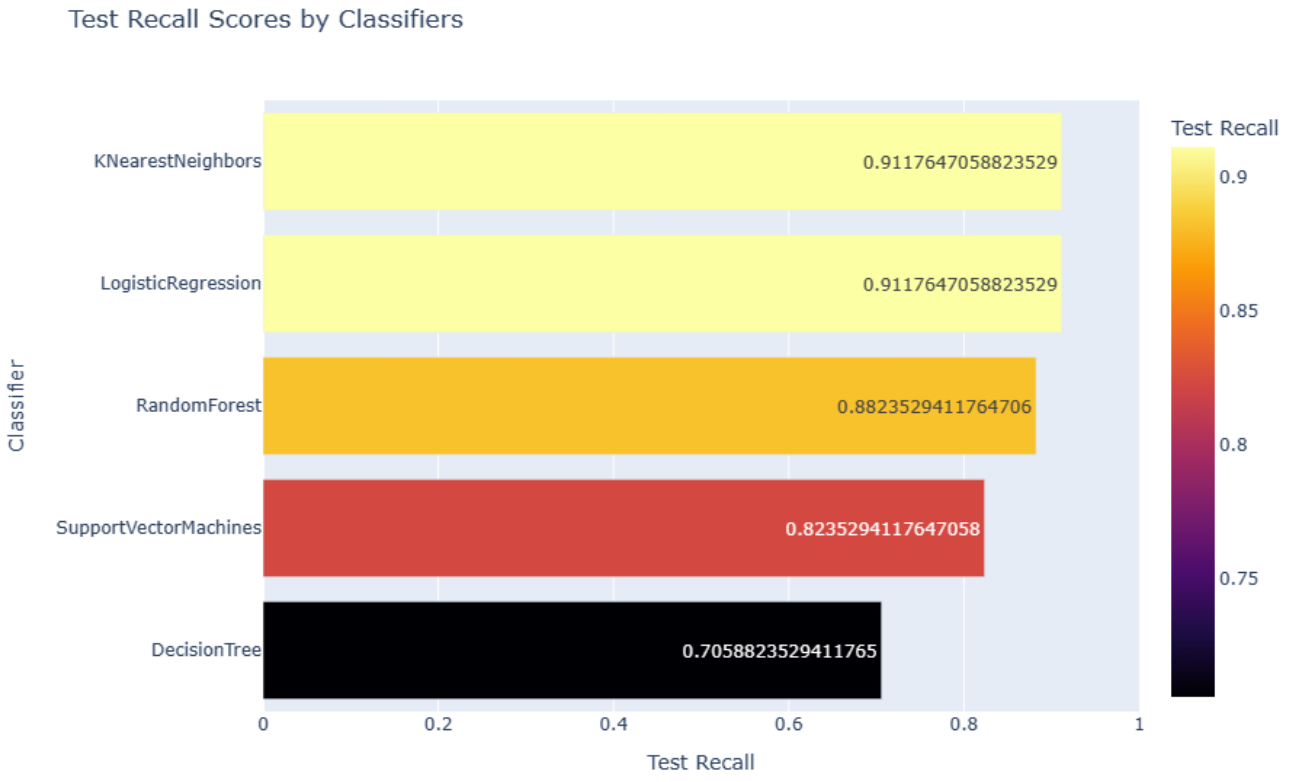
Si può concludere che l'ottimizzazione con gli iperparametri non ha portato cambiamenti notevoli in tutti i modelli. L'unico cambiamento significativo si può notare con il modello Decision Tree.

Metriche senza l'uso di iperparametri a confronto



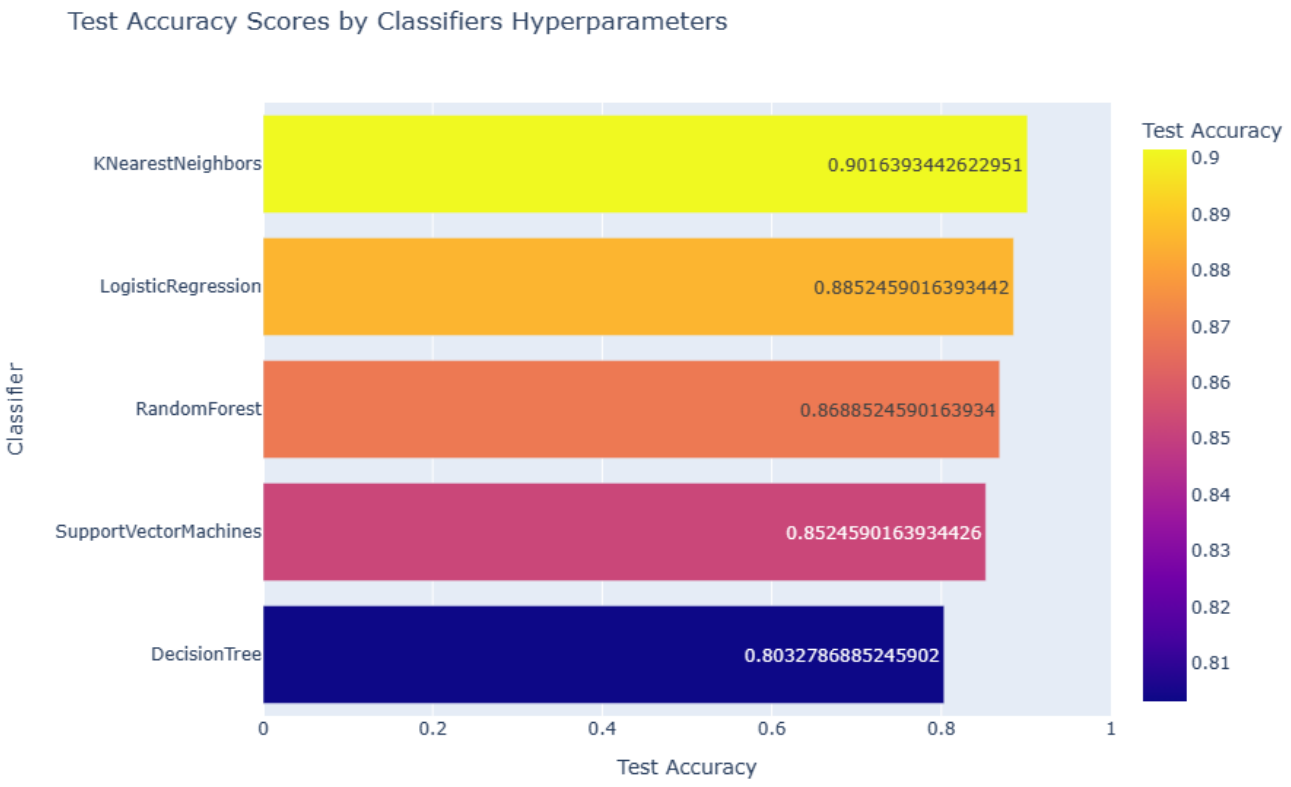


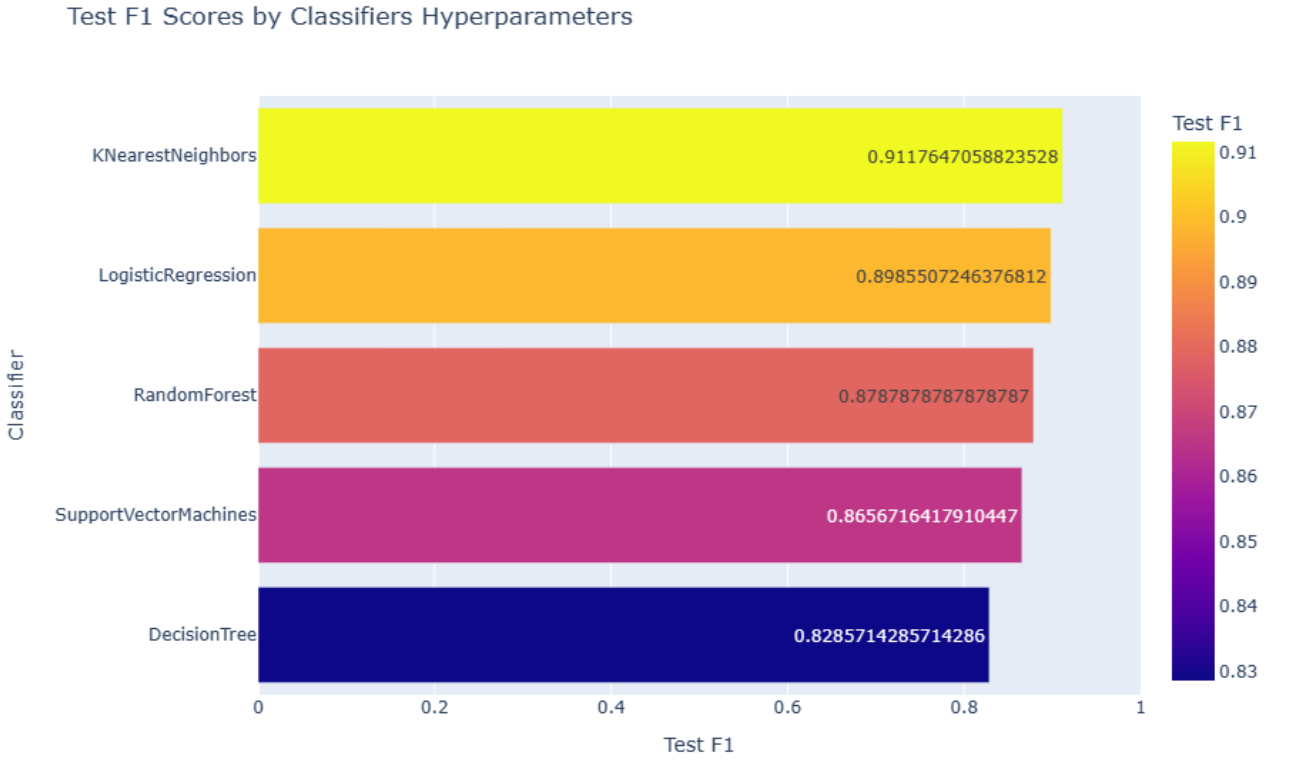


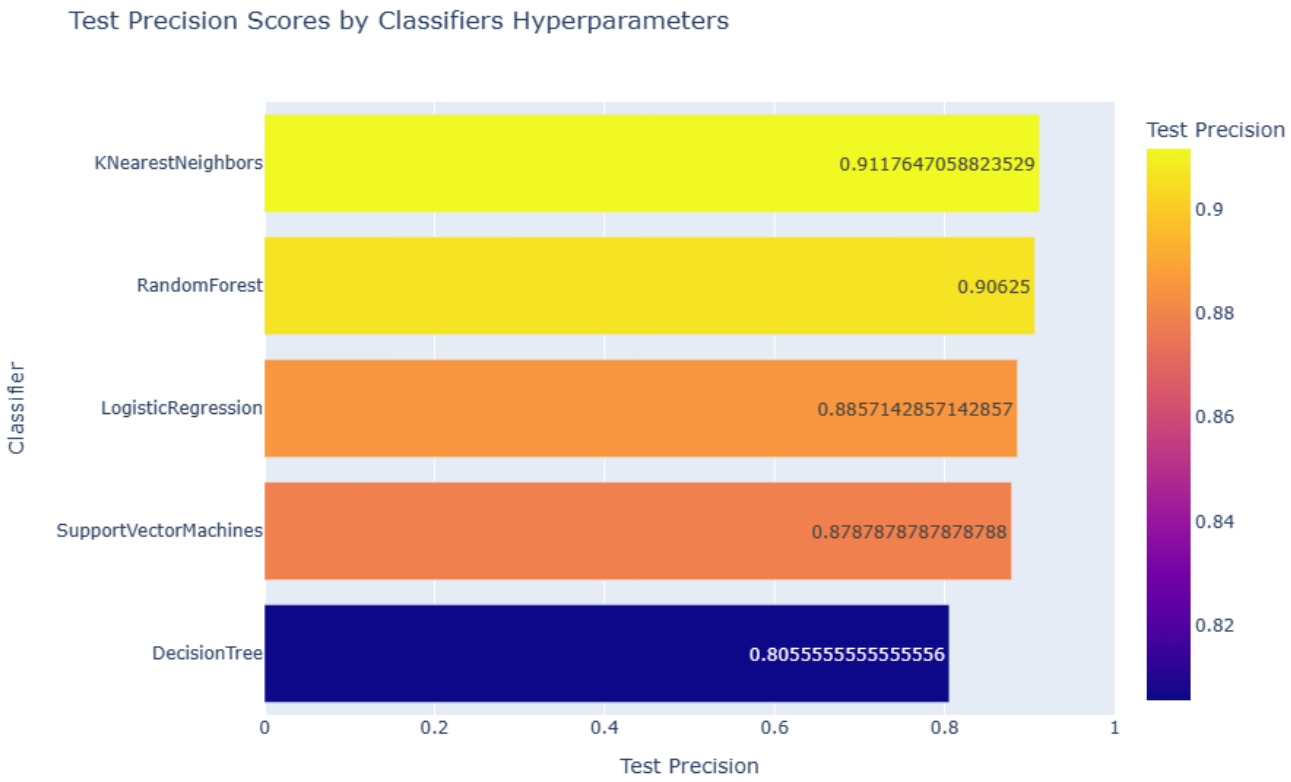


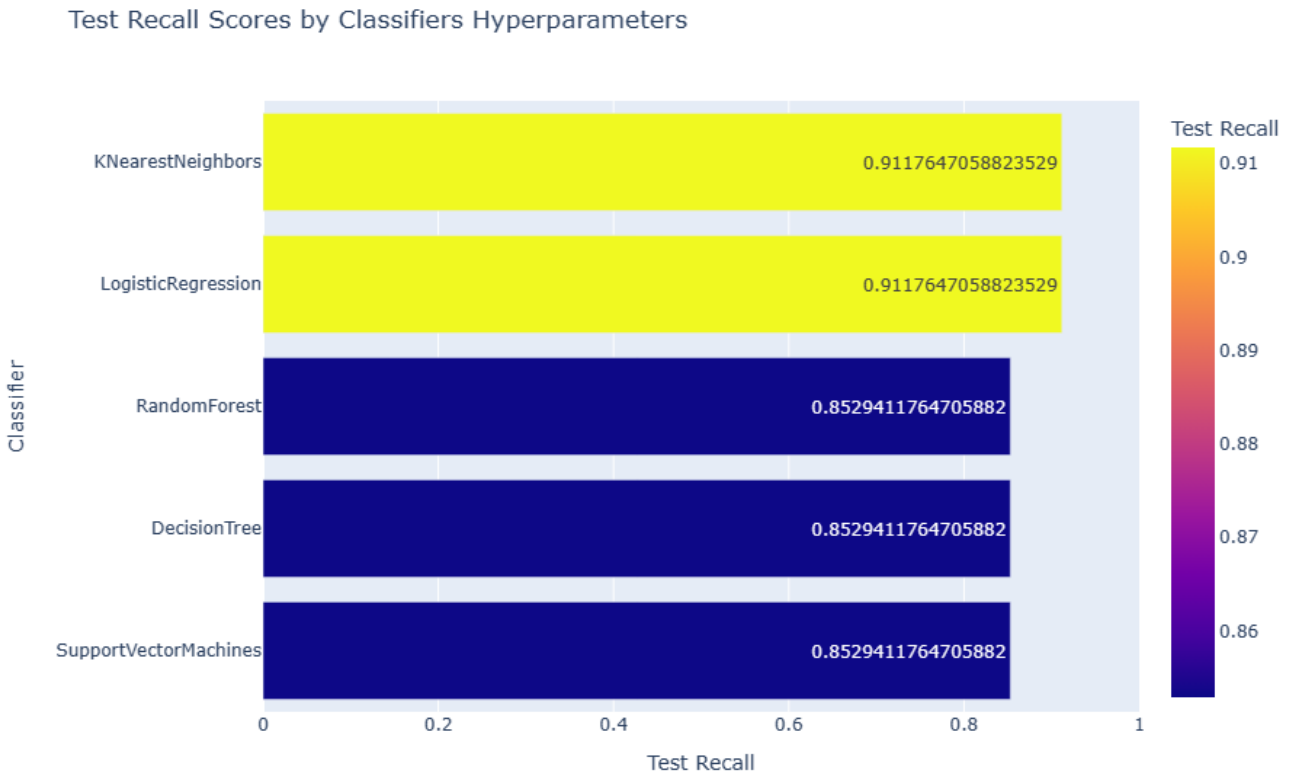
Vedendo i risultati delle metriche che sono state utilizzate per valutare i modelli, senza l'ottimizzazione con iperparametri, è stato possibile concludere che il modello K-Nearest Neighbours risulta esssere il migliore in tutte le metriche. Il modello che presenta i valori più bassi in tutte le metriche risulta essere il Decision Tree.

Metriche con l'uso di iperparametri a confronto









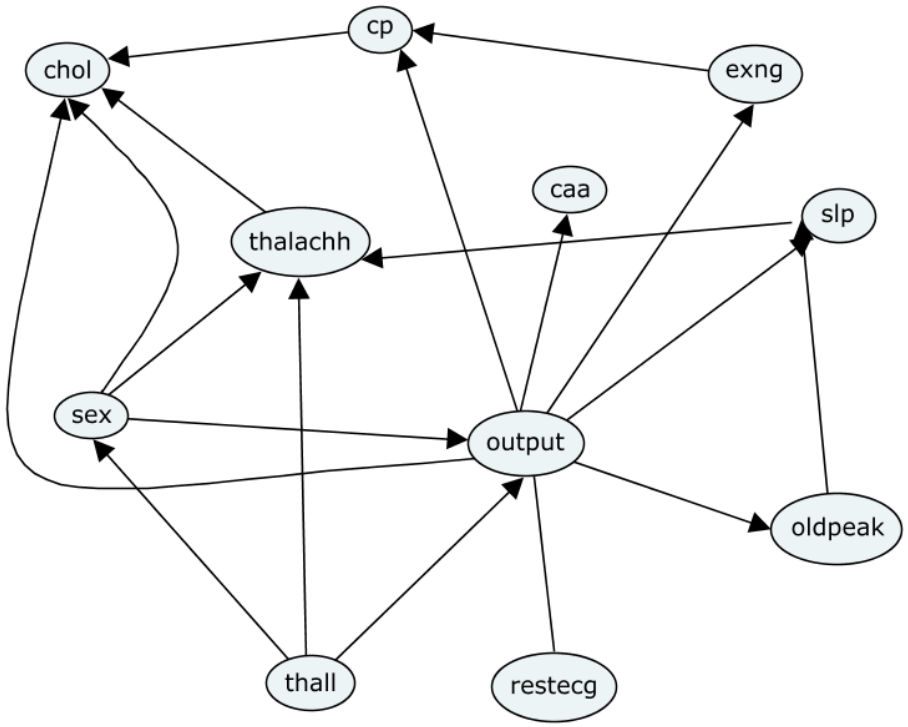
Come già detto precedentemente, l'ottimizzazione dei modelli con iperparametri, non ha portato a notevoli cambiamenti. Quindi, come previsto, il K-Nearest Neighbours anche in questo caso è il modello migliore. Si può notare che anche nel caso di ottimizzazione del modello il Decision Tree risulta essere il modello peggiore.

BAYESIAN NETWORK

È stata implementata una rete bayesiana per capire, in base alle caratteristiche specifiche di un soggetto, quanto questo abbia la probabilità di avere o meno un disturbo cardiaco.
La rete bayesiana è rappresentata mediante grafi diretti aciclici (DAG). Nel grafo, i nodi rappresentano variabili e gli archi rappresentano relazioni probabilistiche tra di esse.
Nel nostro caso specifico è stata utilizzata la funzione Bayesian Network(), ed è stato creato un oggetto `HillClimbSearch` per condurre una ricerca del miglior modello di rete bayesiana utilizzando il punteggio K2.

La rete bayesiana ottenuta presenta i seguenti nodi e archi :

- Nodi : ['age', 'sex', 'cp', 'trtbps', 'chol', 'fbs', 'restecg', 'thalachh', 'exng', 'oldpeak', 'slp', 'caa', 'thall', 'output']
- Archi : [('sex', 'output'), ('sex', 'thalachh'), ('sex', 'chol'), ('cp', 'chol'), ('thalachh', 'chol'), ('exng', 'cp'), ('oldpeak', 'slp'), ('slp', 'thalachh'), ('thall', 'output'), ('thall', 'sex'), ('thall', 'thalachh'), ('output', 'cp'), ('output', 'caa'), ('output', 'oldpeak'), ('output', 'exng'), ('output', 'slp'), ('output', 'chol'), ('output', 'restecg')]



Sono state poi utilizzate le capacità di inferenza della rete bayesiana per calcolare le probabilità condizionali.
Sono effettuate due query:

- La prima query calcola la probabilità di un individuo di non avere un problema cardiaco (`output`) dati i valori delle variabili di evidenza specificate.
- La seconda query calcola la probabilità di un individuo di avere un problema cardiaco (`output`) dati i valori delle variabili di evidenza specificate.

```
Probabilità per un individuo di non avere un problema cardiaco:
+-----------+---------------+
| output    |   phi(output) |
+===========+===============+
| output(0) |        0.0010 |
+-----------+---------------+
| output(1) |        0.9990 |
+-----------+---------------+


Probabilità per un individuo di avere un problema cardiaco:
+-----------+---------------+
| output    |   phi(output) |
+===========+===============+
| output(0) |        0.9966 |
+-----------+---------------+
| output(1) |        0.0034 |
+-----------+---------------+
```