

# Diagnosi di malattie cardiache tramite tecniche di Machine Learning

## 1. Introduzione

Le malattie cardiache sono la prima causa di morte in tutto il mondo: secondo le stime ogni anno sono più di 17 milioni le morti legate a cardiopatie, circa il 31% di tutti i decessi. In questo contesto si rende fondamentale sviluppare approcci accurati e affidabili per fare una diagnosi precoce e gestire con anticipo la malattia.

Lo scopo di questo progetto è quello di utilizzare dei modelli di machine learning per realizzare un classificatore binario che sia in grado, sulla base dei dati clinici dei pazienti, di diagnosticare malattie cardiache potenzialmente fatali.

Il dataset utilizzato è stato reperito dal sito [Kaggle](#) ed è stato realizzato unendo i dati di diversi dataset indipendenti tra loro, combinati tramite 11 features in comune. I cinque dataset utilizzati sono i seguenti:

Dataset	# istanze
<a href="#">Hungarian Institute of Cardiology, Budapest: Andras Janosi, M.D.</a>	294
<a href="#">University Hospital, Zurich and Basel, Switzerland: William Steinbrunn, M.D. and Matthias Pfisterer, M.D.</a>	123
<a href="#">V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.</a>	503
<a href="#">Statlog Heart Dataset</a>	270
<b>Totale</b>	1190

Le 11 features che vanno a comporre il dataset sono le seguenti:

- Età
- Sesso (0 = Femmina, 1 = Maschio)
- Chest Pain Type (1 = dolore anginoso tipico, 2 = dolore anginoso atipico, 3 = dolore non anginoso, 4 = nessun dolore)
- Pressione sanguigna a riposo (mm/Hg)
- Colesterolo (mg/dL)
- Glicemia (1 se > 120 mg/dL, 0 altrimenti)
- Elettrocardiogramma a riposo (0 = normale, 1 = anomalia ST-T, 2 = ipertrofia ventricolare)
- Battiti cardiaci massimi
- Angina pectoris dopo esercizi (0 = assente, 1 = presente)
- Sottoslivellamento ST (mm)
- Inclinazione tangente ST durante esercizi (0 = normale, 1 = ascendente, 2 = orizzontale, 3 = discendente)

È presente inoltre l'etichetta target, che corrisponde a 1 in caso l'individuo sia a rischio e 0 in caso contrario.

## 2. Analisi esplorativa del dataset

Il dataset è composto da 1190 istanze, 11 covariate ed è provvisto di una variabile target.

Le covariate sono sia di tipo numerico che categorico:

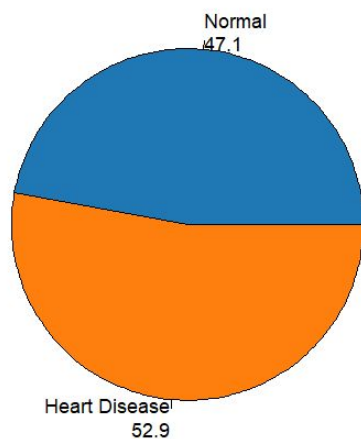
Covariate numeriche					
	Età	Pressione	Colesterolo	Max Heart Rate	Sottoslivellamento ST
Min	28	0	0	60	-2.6
Max	77	200.0	603.0	202.0	6.2
Media	53.72	132.2	210.4	139.7	0.92
Varianza	9.36	18.37	101.42	25.52	1.09
1° Quart.	47	120.0	188.0	121.0	0.0
Mediana	54	130.0	229.0	140.5	0.6
3° Quart	60	140.0	269.8	160.0	1.6

Covariate categoriche						
	Sesso	Chest pain type	Glicemia	Elettro-cardio-gramma	Angina pectoris	Inclinazione ST
# categorie	2	4	2	3	2	4
Categoria max	Maschio	Nessun dolore	<120 mg/dL	Normale	Assente	Orizzontale

Da questa prima analisi dei dati emerge la presenza di outliers in diverse covariate, come “Pressione” (che ha valore minimo 0 mm/Hg) e “Colesterolo” (che ha minimo 0 mg/dL e massimo 603 mg/dL). Prima di utilizzare il dataset, si renderà dunque necessario filtrare i dati per rimuovere gli errori presenti.

L’analisi delle distribuzioni delle covariate fornisce utili informazioni sulla qualità delle variabili contenute nel dataset:

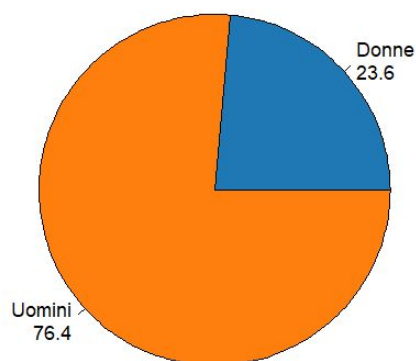
#### **Distribuzione della variabile target:**



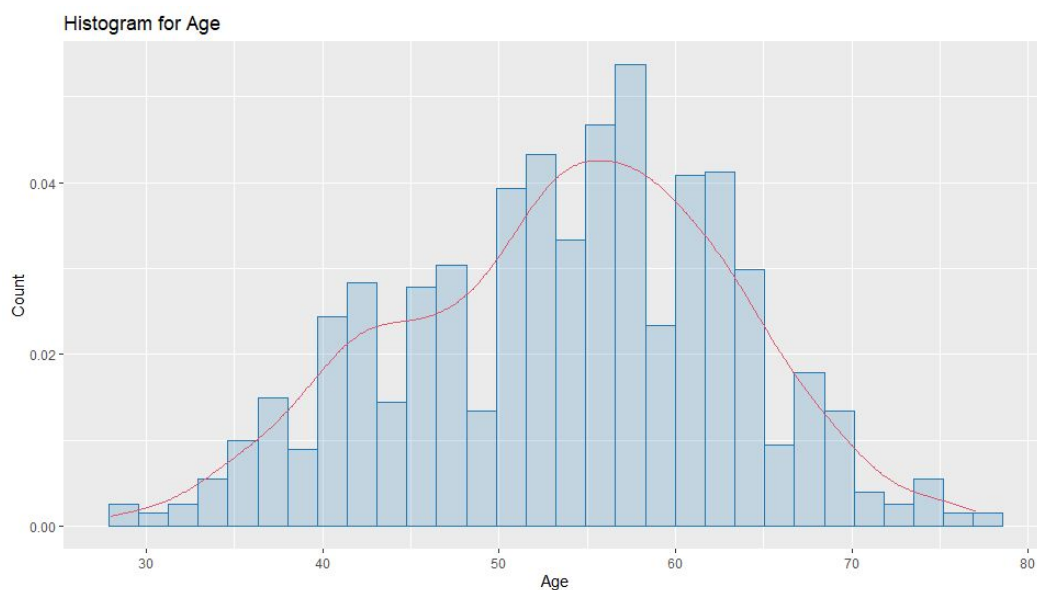
Normal	Heart Disease
561	629

Il dataset si presenta bilanciato, avendo il 47.1% di pazienti sani e il 52.9% di pazienti malati.

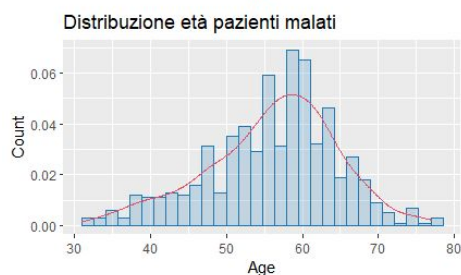
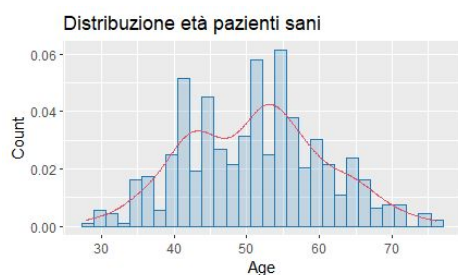
### Distribuzione in base a sesso ed età:



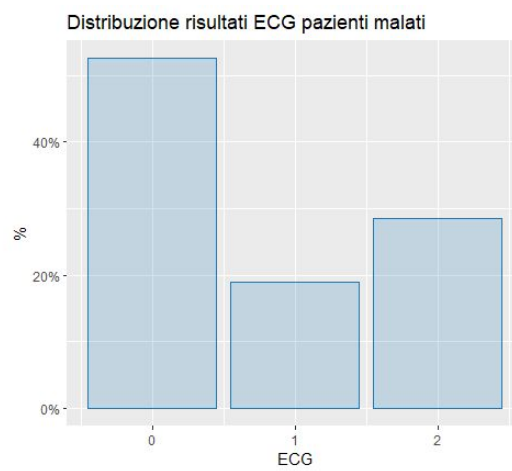
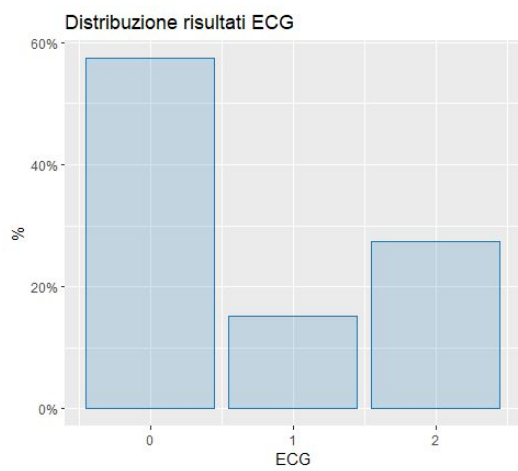
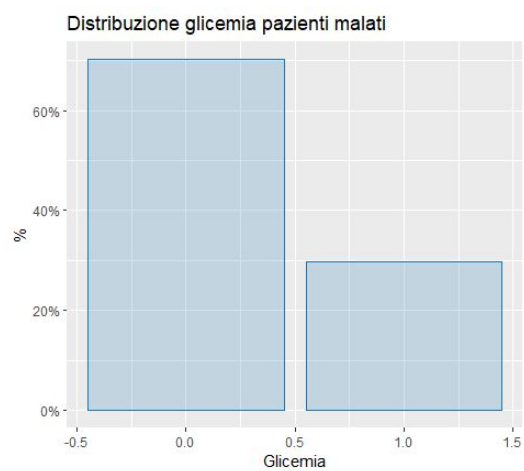
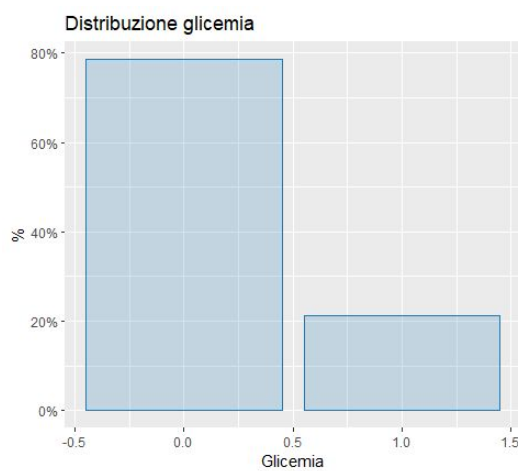
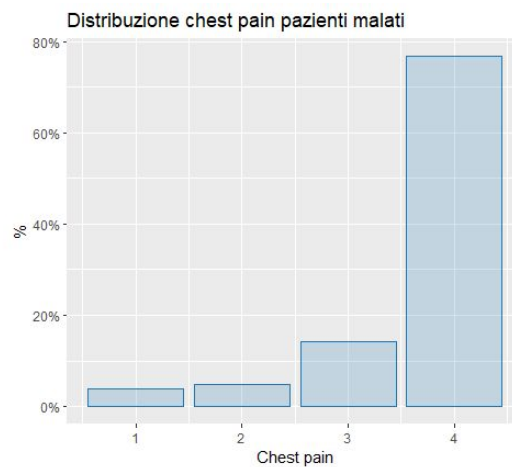
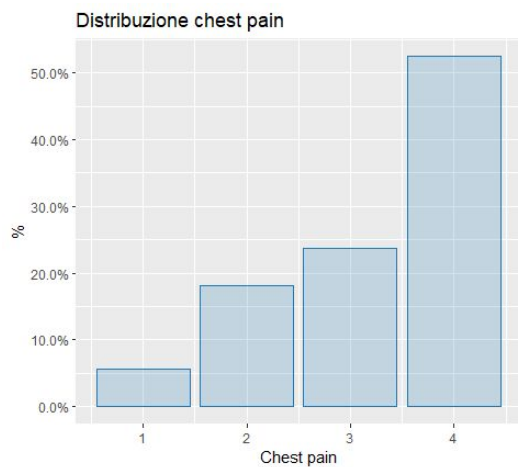
Uomini	Donne
909	281

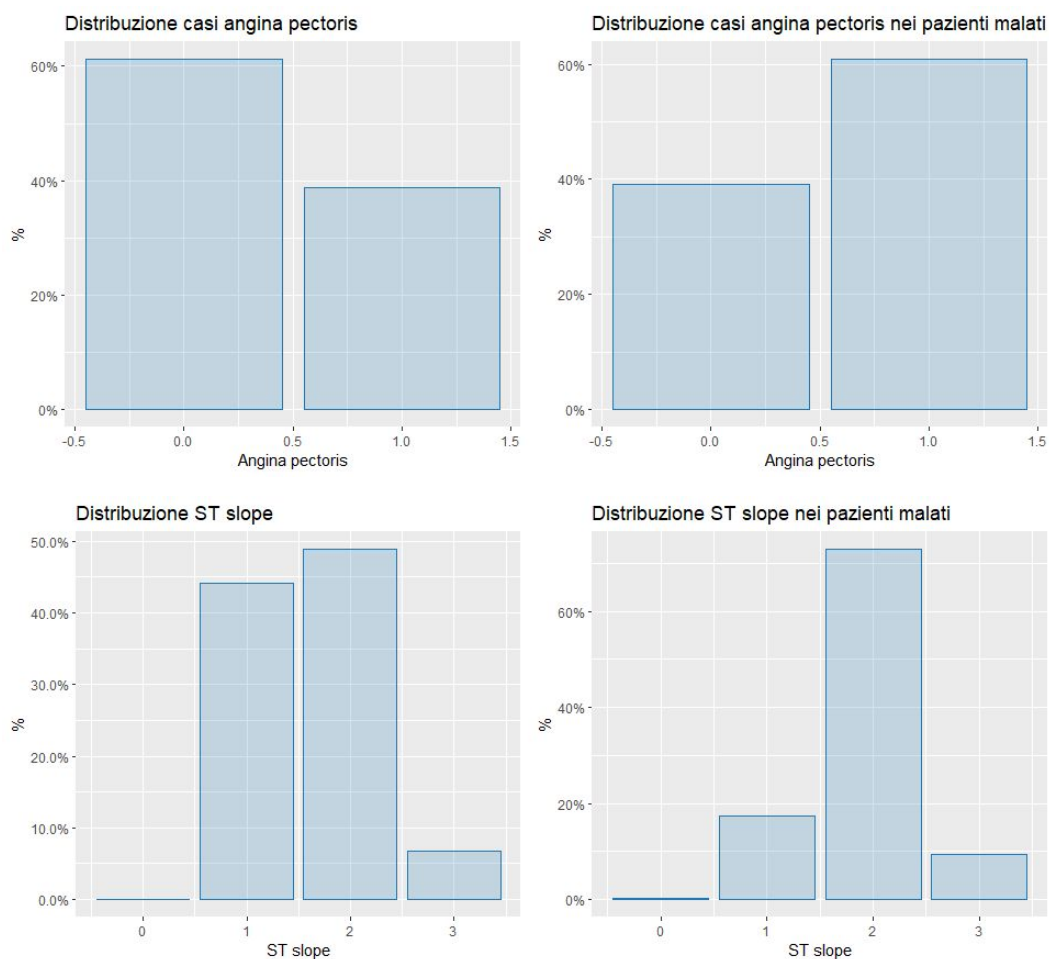


La distribuzione degli individui in base al sesso è fortemente sbilanciata verso il genere maschile, mentre le fasce di età risultano ben rappresentate. Come si mostra nei grafici seguenti, inoltre, la presenza di malattie cardiache è più alta in individui di genere maschile e aumenta leggermente all'aumentare dell'età.



## Distribuzione delle variabili categoriche:





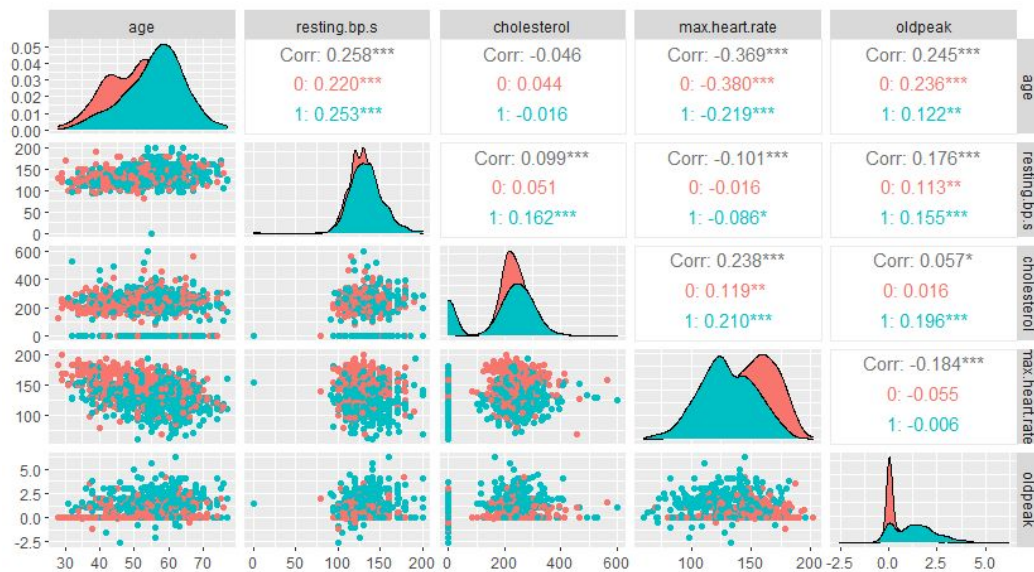
Dai grafici relativi alla covariata “Chest pain type”, emerge che la casistica più frequente è quella di tipo 4 (nessun dolore). Il fatto che questo trend si presenti anche nei soli casi con malattie cardiache potrebbe apparire un’incongruenza, in realtà questo si può ricondurre ai casi di infarto silente. [Uno studio](#) del 2009 condotto dal Duke University Medical Center ha mostrato che il 35% dei soggetti con arteriopatia coronarica mostrava segni di un precedente infarto non diagnosticato.

Anche la covariata indicante i risultati dell’elettrocardiogramma non presenta particolari differenze nei due grafici mostrati. Dunque, a discapito di quella che è la convinzione popolare, questa variabile non sembra essere una buona discriminante per diagnosticare malattie cardiache.

Per quanto riguarda la covariata relativa ai livelli glicemici nel sangue, si osserva che nei casi con malattie cardiache aumenta la percentuale di individui con un livello superiore a 120 mg/dL. Lo stesso accade nel caso della variabile “Angina pectoris”, la cui occorrenza aumenta negli individui a rischio.

Infine, il grafico relativo alla covariata “ST slope” mostra, nei casi con malattie cardiache, una maggiore occorrenza del valore 2 (tangente orizzontale) a discapito del valore 1 (tangente discendente).

### Analisi delle covariate numeriche:



Per quanto riguarda le covariate di tipo numerico, dai grafici di distribuzione e dagli scatterplot emerge nuovamente un aumento del fattore di rischio all'aumentare dell'età.

Per quanto riguarda la covariata "Oldpeak", si osserva che i casi di pazienti sani si concentrano in valori prossimi allo 0, mentre l'istogramma si allarga quando si analizzano i soli individui a rischio.

Tuttavia, per quanto si possano individuare lievi differenze tra istanze con target 0 e 1 negli scatterplot, nessuna coppia di covariate numeriche appare in grado di distinguere nettamente le due classi.

### Conclusioni:

Dall'analisi esplorativa del dataset è emersa innanzitutto la presenza di outliers, che rendono necessario un filtraggio dei dati per rimuovere gli errori presenti.

Non sempre è corretto rimuovere gli outliers, poichè potrebbe trattarsi di osservazioni reali e addirittura tra le più interessanti. Quando però si è certi che questi siano scaturiti da un errore nell'inserimento dei dati (come nel caso dei valori minimi di pressione e colesterolo), è buona norma rimuoverli.

Si è inoltre concluso che, dato il numero e il tipo di covariate a disposizione, non si rende necessario l'utilizzo di tecniche di riduzione delle features (come la Principal Component Analysis). La PCA in ogni caso non sarebbe la tecnica più indicata, vista la presenza di variabili sia numeriche che categoriche. Il problema potrebbe essere risolto ricorrendo a strategie di encoding delle variabili categoriche (ad esempio One-Hot Encoding) oppure utilizzando tecniche

alternative alla PCA, tuttavia si ritiene che questo non sia necessario data la già ridotta dimensione dello spazio delle features.

### 3. Individuazione e rimozione degli outliers

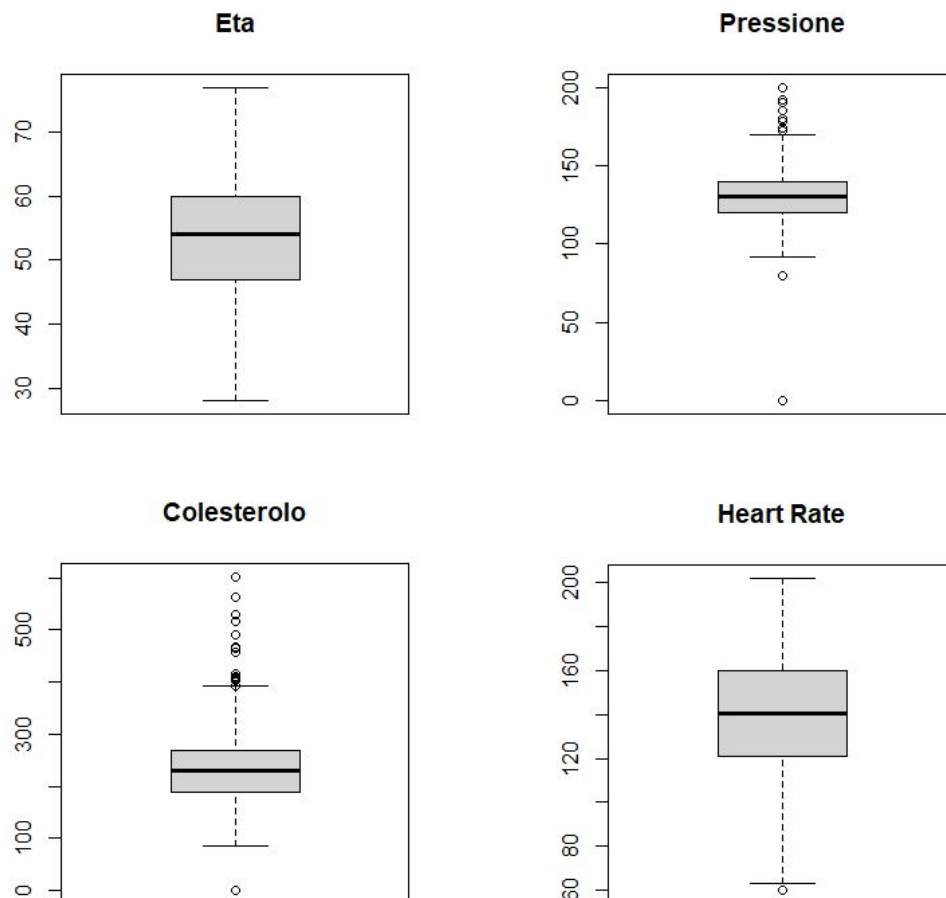
La presenza di outliers, se questi non vengono individuati e gestiti, può intaccare le performance dei modelli utilizzati (specialmente nei modelli di regressione). Come già detto, non sempre tali valori sono da eliminare: potrebbe trattarsi di dati corretti e potrebbero anche essere osservazioni di grande importanza per le sperimentazioni condotte.

Quando però si tratta di evidenti errori nella registrazione dei dati allora si rende necessaria la loro rimozione.

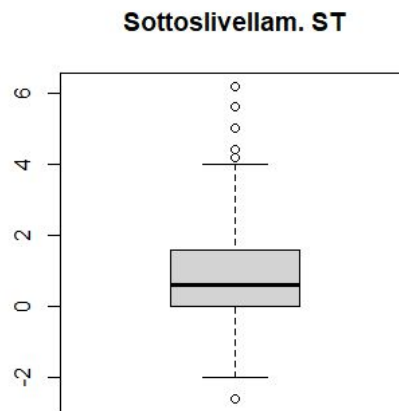
Esistono varie tecniche per individuare gli outliers, in questo contesto per farlo ci si baserà sullo scarto interquartile.

Se  $Q1 = 1^{\circ}$  quartile,  $Q3 = 3^{\circ}$  quartile e  $IQR = Q3 - Q1$ , allora un outlier è un punto che si trova sotto  $[Q1 - 1.5 * IQR]$  oppure sopra  $[Q3 + 1.5 * IQR]$  (il valore 1.5 è scelto a priori e regola l'ampiezza dell'intervallo).

Un modo semplice per individuare gli outliers è quello di visualizzare il boxplot delle variabili:







Dai boxplot si può osservare che, come già notato nell'analisi precedentemente svolta, solamente le covariate "Pressione" e "Colesterolo" hanno degli evidenti errori di registrazione dei dati (valori di pressione e colesterolo pari a 0).

Per quanto riguarda gli altri outliers si giunge alla conclusione che, per quanto si tratti di valori estremi, si tratta comunque di registrazioni plausibili. Pertanto tali valori non verranno rimossi.

Si procede dunque alla sola eliminazione degli outliers minori di  $[Q1 - 1.5 \cdot IQR]$  per le covariate "Pressione" e "Colesterolo".

Così facendo sono state rimosse 172 osservazioni, portando il dataset a 1018 istanze.

## 4. Training di Naive Bayes

Naive Bayes è un modello di apprendimento supervisionato in grado di effettuare classificazioni basate sul Teorema di Bayes.

L'algoritmo basa il suo funzionamento sull'assunzione che le caratteristiche siano indipendenti le une dalle altre, ovvero che la correlazione tra le features sia bassa. Il Teorema di Bayes permette di calcolare per ogni istanza la probabilità di appartenenza ad una classe. Nello specifico, calcola la probabilità di un evento A condizionata a un altro evento B.

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

Per ogni caratteristica vengono dunque calcolate le probabilità semplici e le probabilità condizionate (in questo caso B assume i possibili valori di target). A questo punto la predizione su una nuova istanza verrà effettuata semplicemente effettuando il prodotto tra le probabilità precedentemente calcolate.

Sono diversi i vantaggi del classificatore Naive Bayes:

- Semplicità di utilizzo: si adatta bene sia a dati numerici che categorici e non ha iperparametri da selezionare
- Semplicità computazionale: garantisce alta velocità sia in fase di training che di predizione
- Robustezza rispetto al rumore
- Scalabilità

Dall'analisi preliminare dei dati è emerso che le covariate sono ben distribuite e che la correlazione bassa. Assume dunque un senso la scelta di effettuare degli esperimenti con il classificatore Naive Bayes.

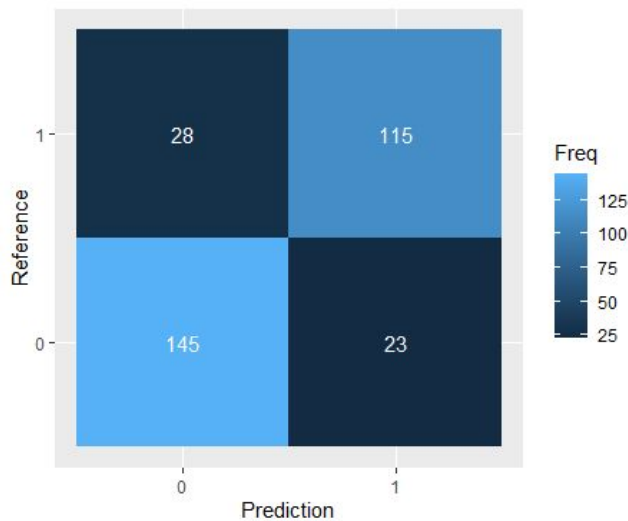
### **Training ed evaluation del modello**

Per effettuare il training del modello, si è innanzitutto diviso il dataset in due parti:

- Trainset: pari al 70% di tutte le istanze, viene utilizzato come input dell'algoritmo di Naive Bayes
- Testset: pari al restante 30% di tutte le istanze, viene utilizzato per testare l'accuratezza del modello predittivo creato

Dopo che il test sarà stato effettuato, si sfrutteranno anche le restanti istanze del testset per addestrare il modello definitivo.

Effettuando la predizione sul testset i risultati ottenuti sono i seguenti:

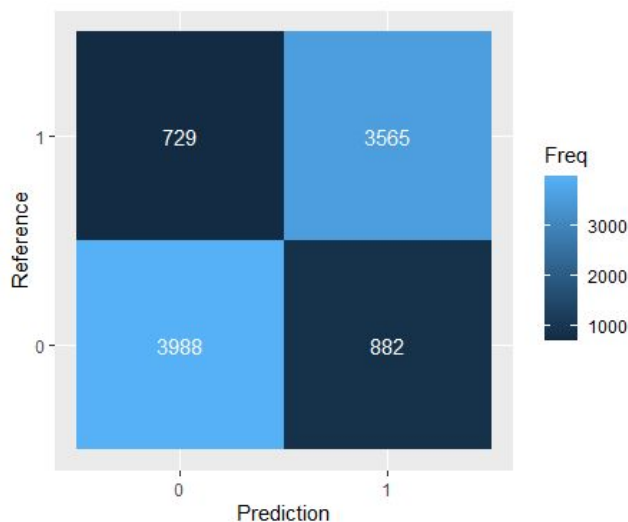


Accuracy = 83.6%

Precision = 79.9%

Recall = 82.8%

Un'ulteriore stima sulle performance può essere effettuata per mezzo di una 10-fold cross evaluation, i cui risultati sono i seguenti:



Accuracy = 82.4%

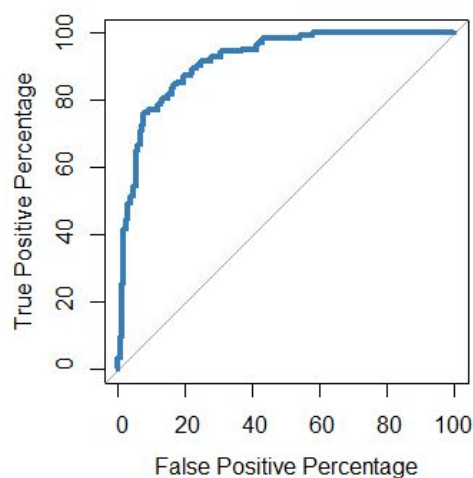
Precision = 80.1%

Recall = 83%

F-measure = 81.6%

95% CI : (0.78, 0.87)

Mentre la curva ROC relativa al modello è la seguente:



AUC = 91.8%

## 5. Training dell'albero di decisione (CART)

L'albero di decisione è un modello di apprendimento supervisionato che cerca di effettuare una classificazione costruendo un albero decisionale in cui:

- Ogni nodo interno rappresenta una variabile
- Un arco verso un nodo figlio rappresenta un possibile valore per la variabile del nodo genitore
- Una foglia corrisponde al valore predetto per la classe, ottenuto a partire dai valori dei nodi che la precedono

L'albero così descritto può essere realizzato, a partire dal dataset fornito, utilizzando diversi criteri.

Nel caso dell'algoritmo CART la metrica utilizzata è l'indice di Gini, il quale offre una misura della eterogeneità di una distribuzione statistica a partire dai valori delle frequenze relative associate alle k modalità di una generica variabile X:

$$I = 1 - \sum_{i=1}^k f_i^2$$

dove  $f_i$  sono le frequenze relative alle k modalità di X

Sono diversi i vantaggi dell'utilizzo degli alberi decisionali:

- Rispetto ad altri algoritmi, è più flessibile rispetto al tipo di dati che gli vengono forniti (ad es. accetta sia variabili numeriche che categoriche)
- È robusto rispetto ad outliers e dati mancanti
- È facilmente interpretabile

In particolare l'interpretabilità dell'albero costituisce un vantaggio non da poco, soprattutto in ambito medico.

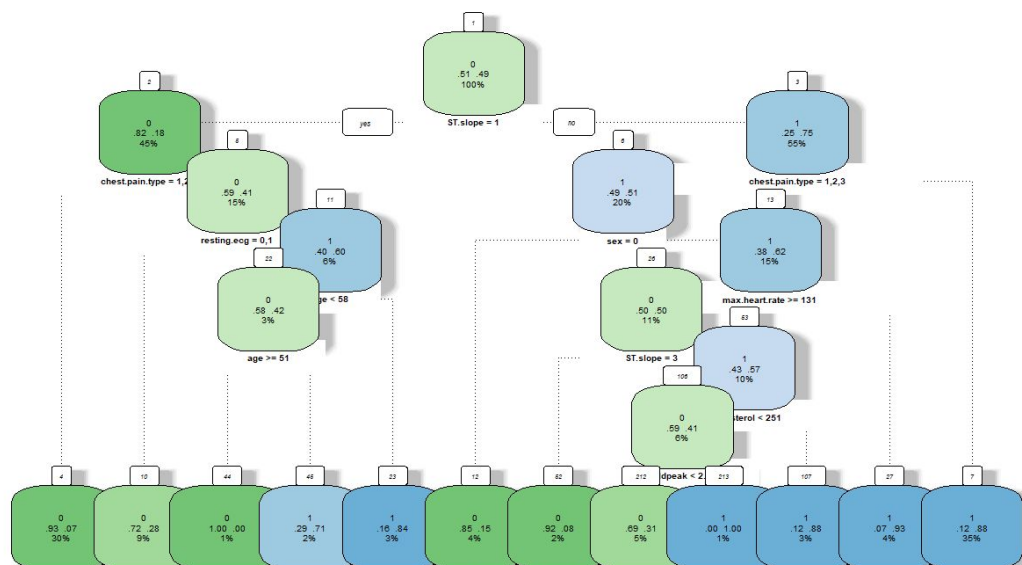
### **Training del modello:**

Per effettuare il training del modello, si è innanzitutto diviso il dataset in due parti:

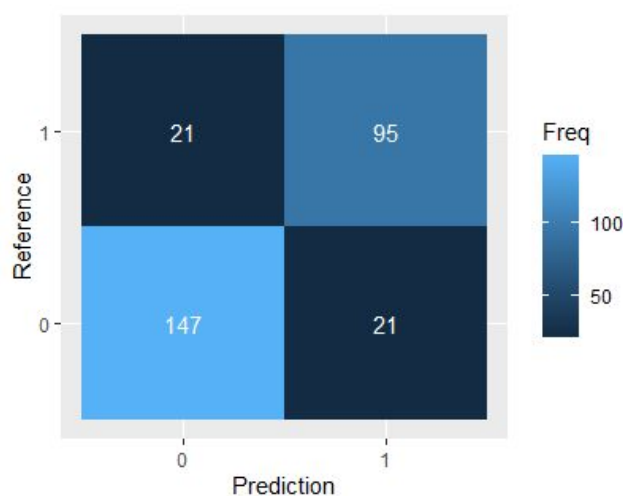
- Trainset: pari al 70% di tutte le istanze, viene utilizzato come input dell'algoritmo per produrre l'albero
- Testset: pari al restante 30% di tutte le istanze, viene utilizzato per testare l'accuratezza del modello predittivo creato

La suddivisione avviene in maniera casuale, pertanto ogni esecuzione del codice produrrà un albero con performance leggermente diverse.

Quando il test sarà effettuato, si sfrutteranno anche le restanti istanze del testset per addestrare il modello definitivo.



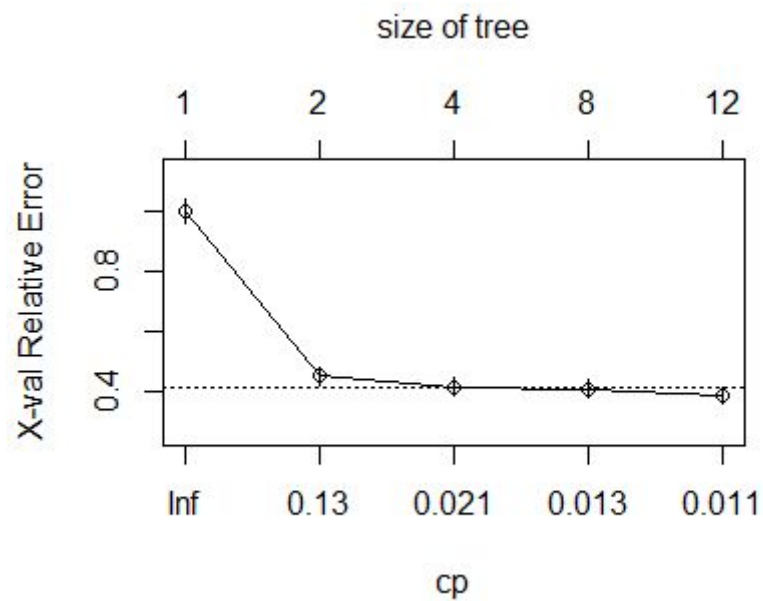
L'albero di decisione mostrato in figura è stato ottenuto dal training del modello, effettuato nelle modalità appena descritte, e ottiene le seguenti performance:



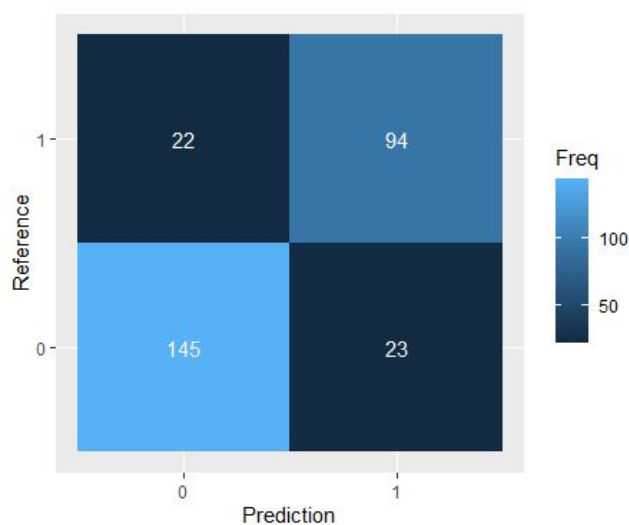
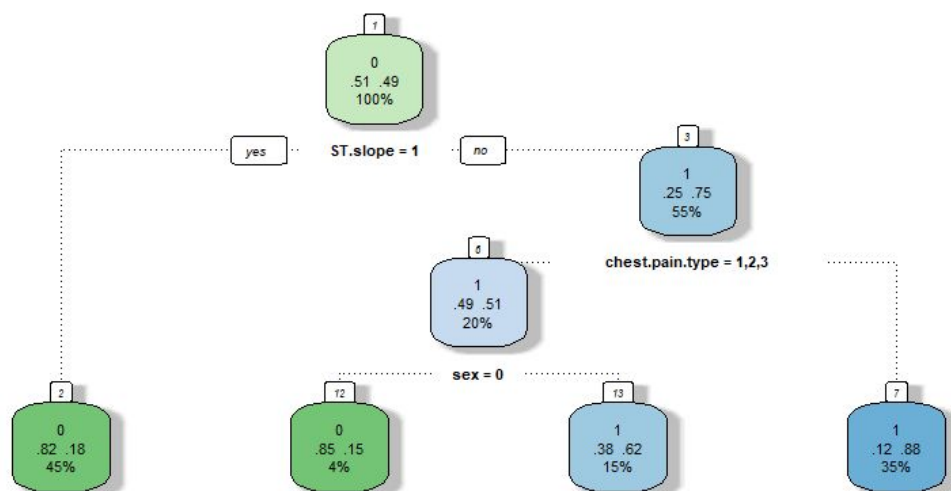
Accuracy = 85.2%  
Precision = 81.9%  
Recall = 81.9%

Spesso è buona norma ottimizzare l'albero di decisione, limitandone la profondità massima, per ridurre la complessità e la probabilità di overfitting. Questo processo è detto pruning e basa la scelta sul parametro di complessità CP:

	CP	nsplit	rel error	xerror	xstd
1	0.554017	0	1.00000	1.00000	0.037519
2	0.031856	1	0.44598	0.44875	0.031124
3	0.013850	3	0.38227	0.41274	0.030187
4	0.012465	7	0.31302	0.40720	0.030035
5	0.010000	11	0.26039	0.38504	0.029404



Osservando il grafico tra CP e xerror, scegliamo un  $cp = 0.02$  per ottenere un buon compromesso tra complessità dell'albero e performance.  
L'albero risultante è dunque il seguente:

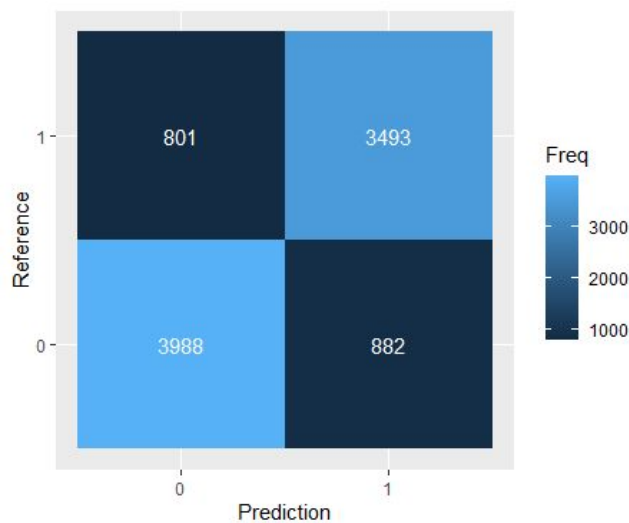


Accuracy = 84.2%  
Precision = 80.3%  
Recall = 81%

### Model evaluation:

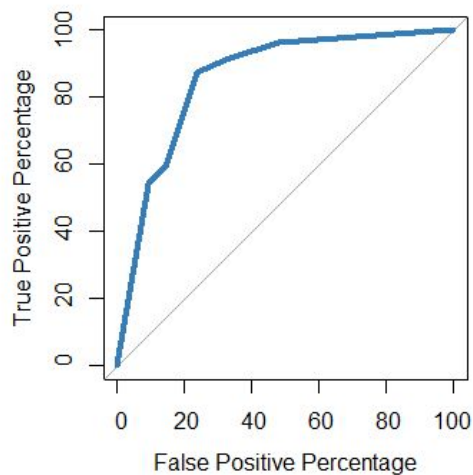
Per valutare le performance del modello realizzato, oltre ai test eseguiti sul testset, si può eseguire una k-fold-cross-validation (utile soprattutto in caso di dataset molto piccoli).

Nel caso in esame, è stata eseguita una 10-cross-fold-validation, i cui risultati sono i seguenti:



Accuracy = 81.5%  
Precision = 80%  
Recall = 80.7%  
F-measure = 80.4%  
95% CI : (0.79, 0.88)

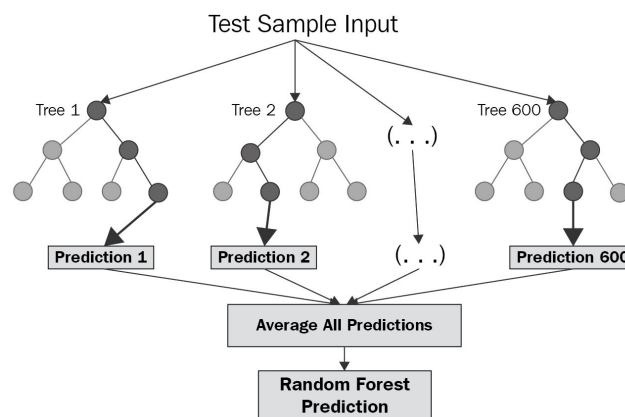
Mentre la curva ROC relativa al modello è la seguente:



AUC = 85.6%

## 6. Training di Random Forest

Random Forest appartiene alla classe dei metodi di apprendimento d'insieme, ovvero quei metodi che usano modelli multipli per ottenere una migliore prestazione predittiva rispetto ai modelli singoli da cui è costituito. In particolare Random Forest viene ottenuto tramite bagging di alberi di decisione: l'algoritmo dunque addestra diversi alberi di decisione a partire da subset del trainset ed effettua predizioni interrogando tutti gli alberi e scegliendo la risposta più popolare (o facendo la media delle risposte).

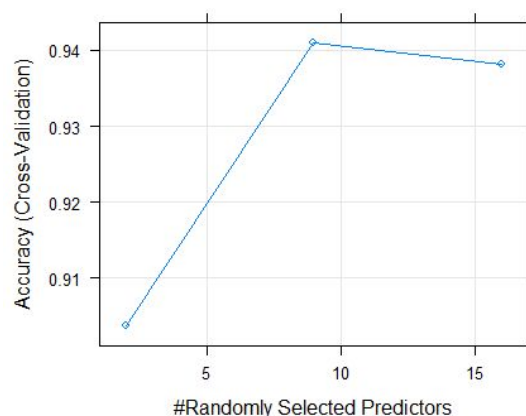


Random Forest appare dunque come un'evoluzione di Decision Tree, presentando performance migliori e riducendo il rischio di overfitting.

Per ottenere ciò tuttavia si rinuncia alla interpretabilità di Decision Tree e si passa a un modello più oneroso in termini di tempo.

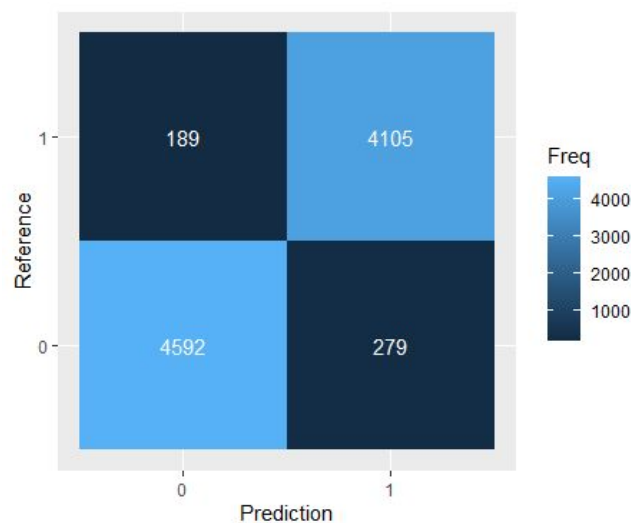
### Training ed evaluation del modello:

Il training del modello richiede la selezione del parametro `mtry`, ovvero il numero di predittori che verranno usati per il training degli alberi (selezionati randomicamente). Per scegliere il parametro più adatto, semplicemente si allenano diversi modelli con diversi parametri `mtry` e si seleziona quello che garantisce una accuracy più alta



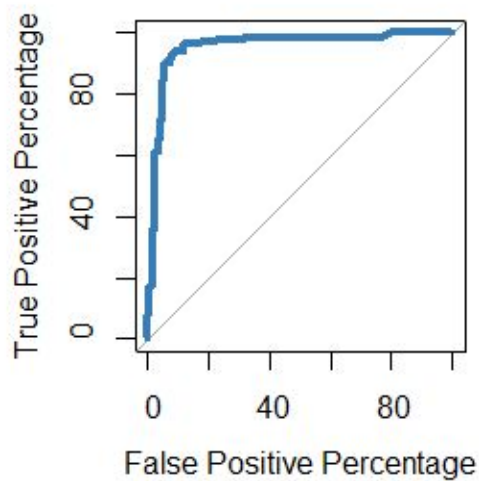


Dal grafico emerge che il valore di mtry ottimale è pari a 9.  
Per misurare le performance del modello si esegue una 10-fold-cross-validation, i cui risultati sono i seguenti:



Accuracy = 94.9%  
Precision = 93.6%  
Recall = 95.6  
F-measure = 94.6  
95% CI : (0.89, 0.95)

Mentre la curva ROC relativa al modello è la seguente:

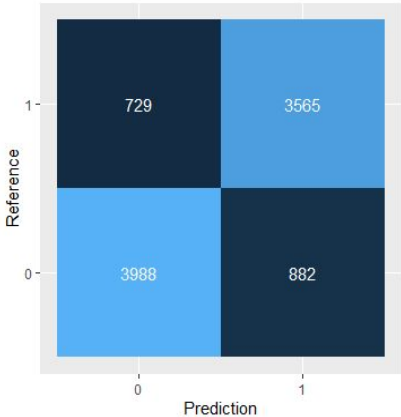
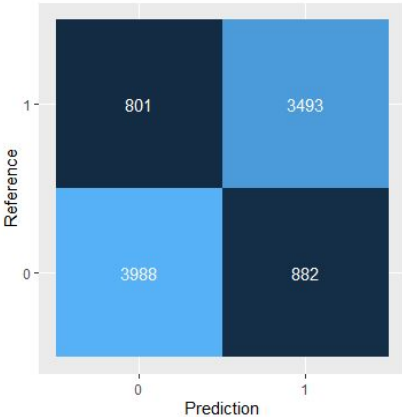
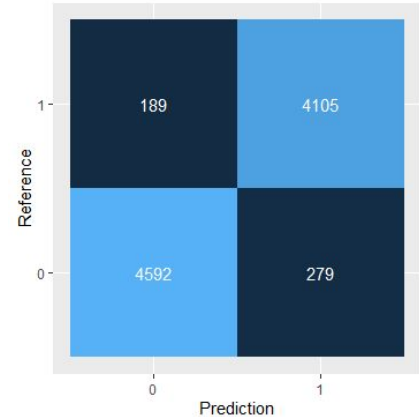


AUC = 95.2%

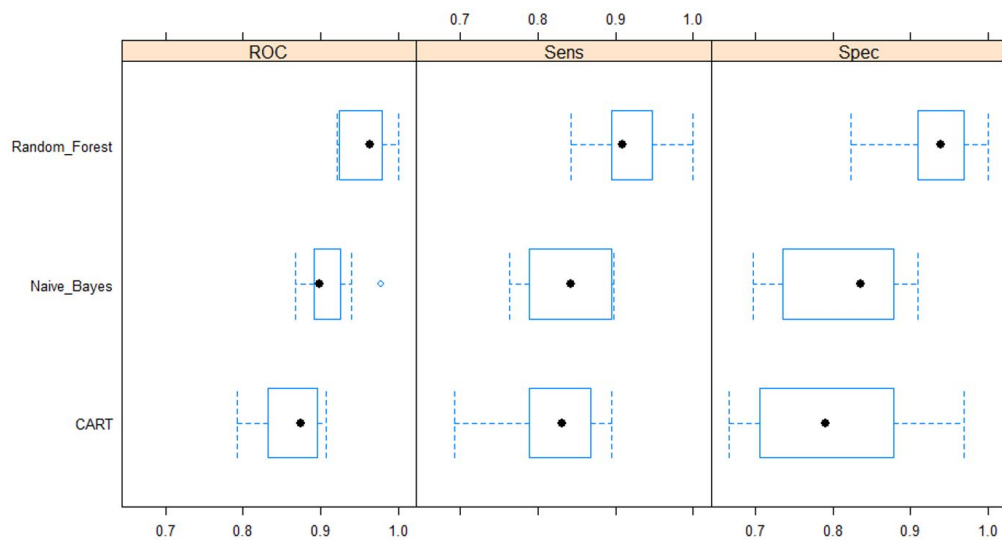
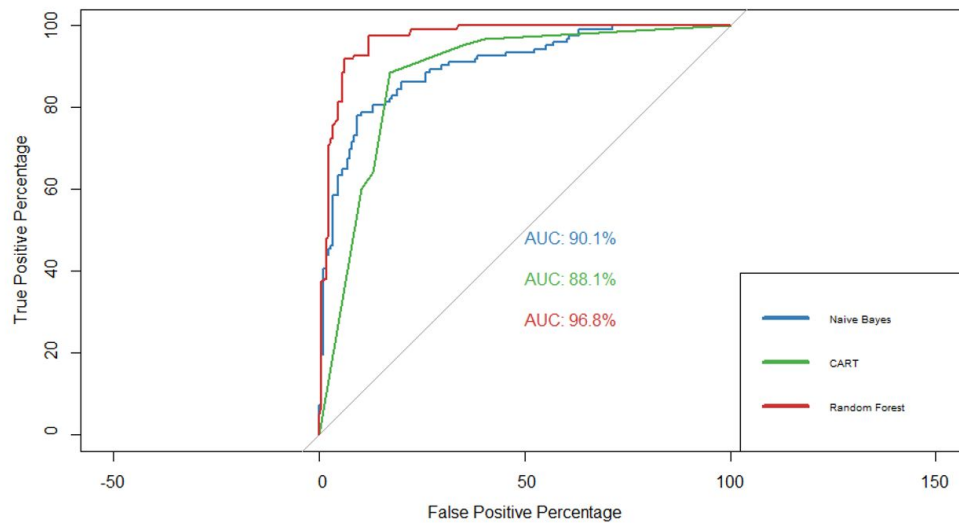
## 7. Analisi dei dati ottenuti

Per effettuare un confronto tra i tre modelli realizzati, si ricorre alle misure di performance già calcolate precedentemente, ad ulteriori visualizzazioni grafiche e ai risultati in termini di tempi nella 10-fold cross validation.

Di seguito la tabella comparativa dei tre modelli:

Naive Bayes	Decision Tree (CART)	Random Forest
		
Accuracy = 82.4%	Accuracy = 81.5%	Accuracy = 94.9%
Precision = 80.1%	Precision = 80%	Precision = 93.6%
Recall = 83%	Recall = 80.7%	Recall = 95.6
F-measure = 81.6%	F-measure = 80.4%	F-measure = 94.6
95% CI : (0.78, 0.87)	95% CI : (0.79, 0.88)	95% CI : (0.89, 0.95)

## CURVE ROC



## TEMPI

	Everything	FinalModel	Prediction
Naive_Bayes	3.57	0.01	NA
CART	0.75	0.02	NA
Random_Forest	14.75	0.42	NA

Dal punto di vista delle performance, Naive Bayes e Decision Tree (CART) appaiono molto paragonabili, con Naive Bayes che risulta leggermente migliore nel confronto tra le curve ROC (e tra le relativa AUC).

Anche dal punto di vista dei tempi i due modelli sono pressoché paragonabili (almeno per quanto riguarda il training del modello definitivo).

Random Forest ottiene invece performance nettamente superiori sotto tutti gli aspetti. Dal punto di vista del tempo tuttavia, si tratta di un modello molto più oneroso rispetto agli altri due.

## **8. Conclusioni**

Questo progetto è partito con l'intento di realizzare un modello in grado di realizzare un classificatore binario che fosse in grado, sulla base dei dati clinici dei pazienti, di diagnosticare malattie cardiache.

Dopo un'analisi preliminare dei dati, si è scelto di procedere con l'addestramento di tre diversi modelli di apprendimento: Naive Bayes, Decision Tree e Random Forest.

Sulla base dei risultati di performance ottenuti dagli esperimenti eseguiti sui modelli, Random Forest è risultato essere il modello più efficace ma anche quello più oneroso in termini di tempo. Naive Bayes e Decision Tree hanno invece ottenuto risultati paragonabili.

Alla luce di questo, si ritiene che il modello più consigliabile fra i tre sia proprio Random Forest che, nonostante la bassa interpretabilità e l'onerosità computazionale, ha ottenuto risultati nettamente superiori agli altri algoritmi. Tra gli altri due modelli invece, la preferenza ricade su Decision Tree poiché quest'ultimo garantisce un'ottima interpretabilità e, allo stesso tempo, performance paragonabili a quelle di Naive Bayes.