Università degli Studi di Salerno

DIPARTIMENTO DI INFORMATICA

HEART TO HEART

HEART DISEASE PREDICTOR

Progetto di Fondamenti di Intelligenza Artificiale

Anno Accademico *2023/2024*

Link Repository

https://github.com/OddlyHod/HTH

Docente:

Palomba Fabio

Partecipanti:

Amendola Alfredo -
Di Tella Nazaro -
Xu Xin Yu -

Sommario

Capitolo 1

Introduzione al Contesto

1.1 Definizione del Contesto

L'insufficienza cardiaca o scompenso cardiaco è una condizione per cui il cuore non riesce a pompare sangue in quantità sufficiente da soddisfare le esigenze dell'organismo. L'insufficienza cardiaca non si manifesta all'improvviso ma si sviluppa lentamente, spesso nell'arco di anni. L'insufficienza cardiaca è una patologia molto diffusa: colpisce infatti circa 14 milioni di europei.
In Italia, lo scompenso riguarda il 2% della popolazione, circa 1.200.000 di pazienti con una crescita media del 2,3% nei prossimi 10 anni.

Sia l'Insufficienza cardiaca acuta che quello cronica sono associate ad una elevata mortalità e al rischio di andare incontro a frequenti ospedalizzazioni ed inoltre ha un effetto negativo sulla qualità della vita.

L'insufficienza cardiaca si accompagna a *sintomi* caratteristici:

- Dispnea (mancanza di fiato);
- Ortopnea (difficoltà a respirare quando si è distesi);
- Tosse frequente;
- Gonfiore (edema) di piedi, caviglie e gambe;
- Debolezza generale, affaticamento o stanchezza;
- Perdita di appetito;
- Senso di ripienezza o tensione addominale.

1.2 Obiettivi

L'obiettivo che il progetto HTH si pone è quello di ridurre al minimo l'errore umano creando e sviluppando un modello di intelligenza artificiale per predire uno scompenso cardiaco.

Tramite lo sviluppo di questo modello si prova ad automatizzare la diagnosi di uno scompenso cardiaco lasciando al medico più tempo per concentrarsi sul trattamento.

Il sistema proposto utilizzerà una varia gamma di attributi numerici e categorici, variabili che spaziano dall'età ai valori del colesterolo spaziando per il tipo di dolore che si accusa al petto.
Utilizzando un set di variabili più eterogenee si riduce la probabilità di un falso positivo/negativo.

Uno dei focus, se non ***il*** focus, di questo progetto sarà la fase di validazione del modello, verranno valutate affidabilità e precisione e questi valori verranno messi a confronto con strumenti di diagnosi tradizionali.

1.3 Contesto Applicativo

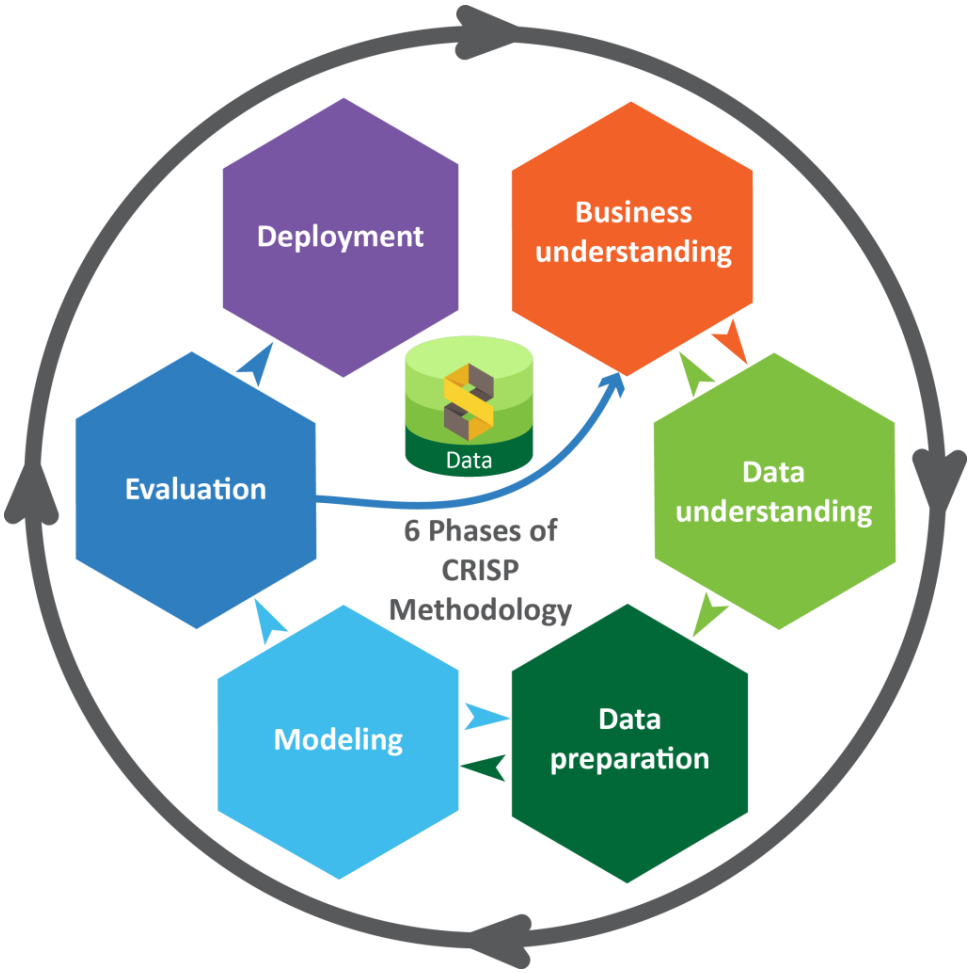
Questo progetto non solo si pone come scopo la creazione di un modello fatto e finito per il contesto clinico, ma vede, in una sua integrazione ed implementazione futura per il possibile utilizzo diretto da parte del pubblico, integrandolo all'interno di un applicativo web.

Questo potrebbe portare a vantaggi esponenziali, diagnosticando una patologia precocemente, velocemente e soprattutto remotamente.

2.1 CRISP – DM

CRISP-DM è l'acronimo di Cross-Industry Standard Process for Data Mining, è un process model che mette a disposizione un approccio strutturato a progetti di data mining.

Il modello si compone di 6 diverse fasi, ognuna specializzata per un ambito e tutte le fasi possono essere eseguite in maniera scollegata, ovvero, è possibile seguire un determinato percorso tra fasi ma è anche possibile effettuare il backtracking e tornare ad una fase precedente.



2.2 Specifiche P.E.A.S.

PEAS è l'acronimo in Inglese di Performance Environment Actuators Sensors. È utilizzato per raggruppare in un unico termine le caratteristiche dell'ambiente operativo.

2.2.1 Performance

Misura di prestazione adottata per valutare l'operato del modello.
Nel nostro caso le misure di prestazione sono i valori di ***precision***, ***accuracy***, ***recall*** ed ***f1-score***.

2.2.2 Environment

L'ambiente in cui opera il modello.
Nel nostro caso, il modello opera in un contesto clinico ovvero nell'insieme di tutti gli EHS[1], le cartelle elettroniche dei pazienti.

Le caratteristiche dell'Environment sono:

- Completamente Osservabile (Poiché conosco tutte le informazioni riguardo all'EHS)
- Stocastico (Poiché lo stato successivo è influenzato da quelli precedenti)
- Episodico (Poiché ogni previsione è a sé stante)
- Discreto (Poiché il risultato è o affermativo o negativo)
- Singolo (Poiché il modello non è multi-agente)

2.2.3 Actuators

Gli attuatori disponibili all'agente per intraprendere le azioni.
Nel nostro caso, i risultati della valutazione.

2.2.4 Sensors

I sensori attraverso i quali l'ambiente riceve gli input percettivi.
Che nel nostro caso sono i valori predittivi del modello, ovvero i valori sui quali il modello effettuerà le sue predizioni.

2.3 Business Understanding

La fase iniziale del CRISP – DM è fondamentale per la raccolta dei requisiti e la definizione degli obiettivi di business che si intende raggiungere.

La fase di business prevede la definizione dei business success criteria, ovvero i criteri secondo i quelli potremo accertare che il sistema è costruito in linea agli obiettivi di business.

In questa fase vengono anche selezionate le tecnologie ed i tool necessari al raggiungimento dei business success criteria.

- ***Obiettivo di Business***
  L'obiettivo di business è quello di stimare se un paziente, a partire dai suoi dati clinici, è o non è affetto da scompenso cardiaco
- ***Risorse***
  Per creare ed addestrare il nostro modello abbiamo bisogno di un dataset, che prenderemo dal sito Keggle[2], che mette disposizione vari dataset. Nel nostro caso utilizzeremo un dataset che mette tratta cartelle cliniche di oltre mille pazienti. (Per il trattamento dei dati cliccare qui)
- ***Rischi***
  Uno dei rischi principali è la poca accuratezza del modello che sarebbe causata da un dataset che presenta poca eterogeneità, nel nostro caso questo problema verrà analizzato ed eventualmente trattato nelle fasi successive.
- ***Tecnologie***
  Per l'analisi, la modellazione, l'addestramento e la visualizzazione grafica dei dati e del modello verranno utilizzate varie tecnologie, come ad esempio *Python* in concomitanza di varie librerie, come *pandas*, *numpy*, *pyplot* e *seaborn* per le informazioni sui dati ed *sklearn*[3] per la fase di *feature engineering* e la fase di *modeling*.

2.4 Data Understanding

La seconda fase del CRISP - DM consiste nell'identificazione, collezione ed analisi dei dataset. Innanzitutto, quindi, vengono *acquisiti i dati* necessari al raggiungimento degli obiettivi di business e tecnici. I dati verranno poi caricati in un tool di *analisi dei dati*, quindi documentati ed esaminati.
Successivamente si passa alla fase di *esplorazione dei dati*, durante la quale vengono visualizzati ed infine la fase di *qualità dei dati*, dove vengono identificati eventuali problemi di qualità (come ad esempio dati mancanti).

2.4.1 Acquisizione dei Dati

Il dataset per l'addestramento (e la valutazione) del modello è stato reperito in formato csv da keggle. Tutti i dati sono cartelle elettroniche di pazienti (EHS) e quindi contengono i risultati delle analisi effettuate per paziente (ECG ed analisi del sangue).

2.4.2 Analisi dei Dati

Il dataset, presenta circa 1000 cartelle cliniche e quindi 1000 pazienti e le feature per ogni paziente sono le seguenti:

- Age: Età del paziente; [Il numero di anni]
- Sex: Sesso del paziente; [M: Male, F: Female]
- ChestPainType: Tipo di dolore al petto; [TA, ATA, NAP, ASY][4]
- RestingBP: Pressione sanguigna a riposo; [in mm/Hg]
- Cholesterol: Valore del colesterolo; [in mm/dl]
- FastingBS: Livello di zucchero nel sangue a digiuno; [1 se è maggiore di 120, 0 altrimenti]
- RestingECG: Valori dell'ECG a riposo; [Normal, ST, LVH][5]
- MaxHR: Il valore massimo della freq. Cardiaca; [Valore tra 60 e 202]
- ExerciseAngina: Angina indotta da esercizi; [Y: Sì, N: No]
- Oldpeak: Sottolivellamento del tratto ST; [Valore Numerico]
- ST_Slope: Pendenza dal picco dell'ST; [Up: upsloping, Flat: flat, Down: downsloping]
- HeartDisease: Classe di Output; [1: Scompenso, 0: Normale]

Age	Sex	CPT	RestingBP	Cholesterol	FastingBS	RestECG	MaxHR	ExcAng	OldPeak	OPS	HD
62	F	TA	160	193	0	Normal	116	N	0	Up	0
…	…	…	…	…	…	…	…	…	…	…	…

Un esempio di paziente in forma tabellare

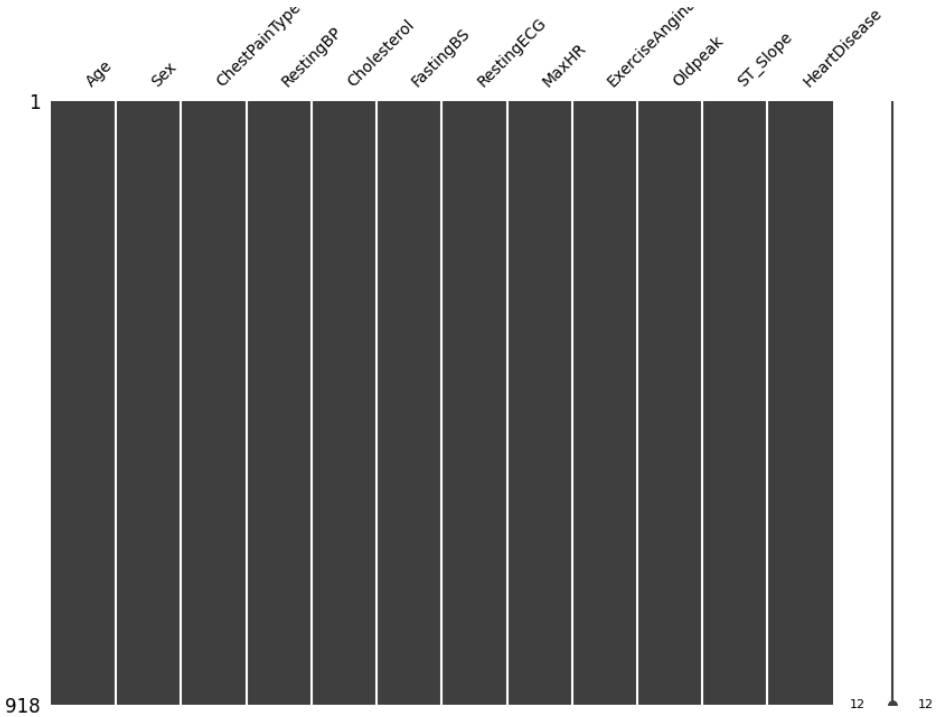
2.4.3 Esplorazione dei Dati

In questa fase, verrà effettuata la visualizzazione dei dati per trarre conclusioni sulla completezza, sulla distribuzione e sulla correlazione dei dati.

In primis, ci conviene esplorare il dataset alla ricerca di possibili valori nulli.
(Per la fase di esplorazione ci avvarremo dell'utilizzo di Python)

Grafico a Griglia

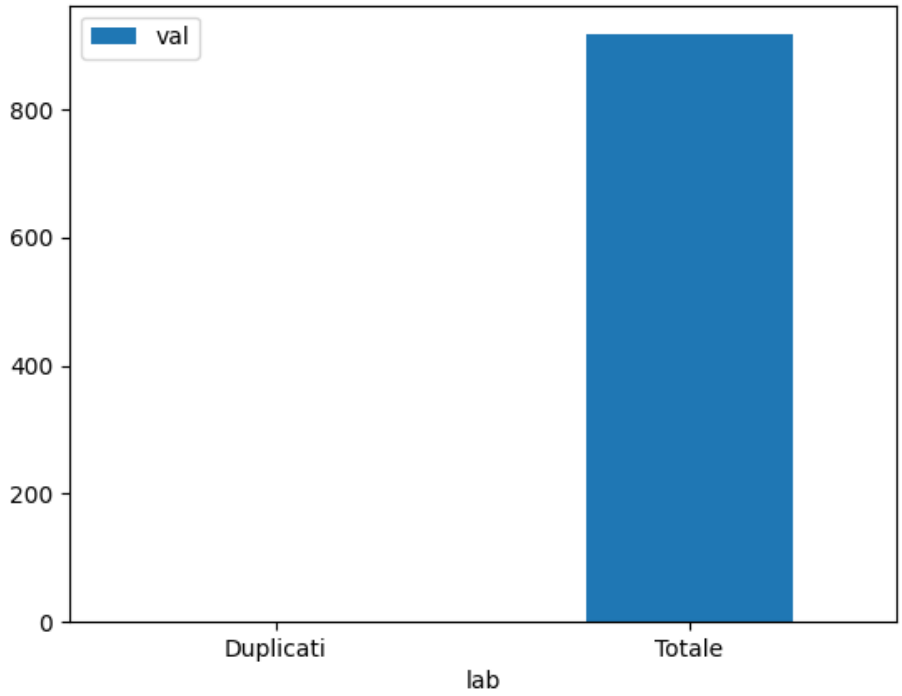
Con questo grafico a griglia, ci viene evidenziata la presenza o meno di valori nulli.



Nel nostro caso non ci sono valori nulli.

Grafico Duplicati

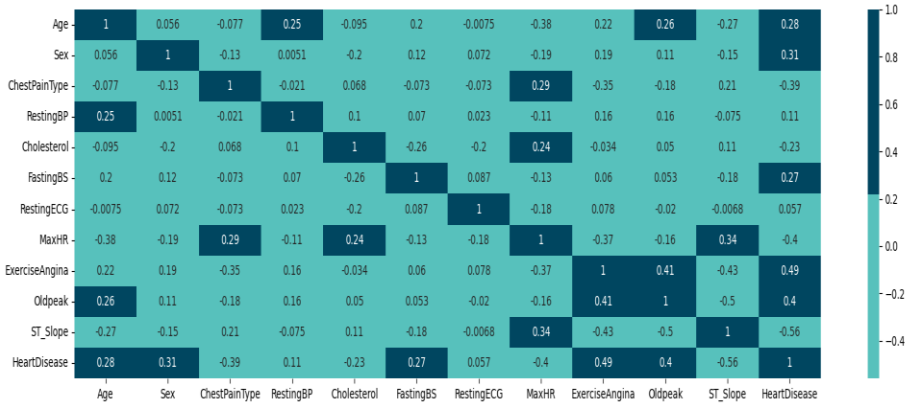
Da una semplice analisi utilizzando la funzione duplicate() di pandas, ci accorgiamo che:



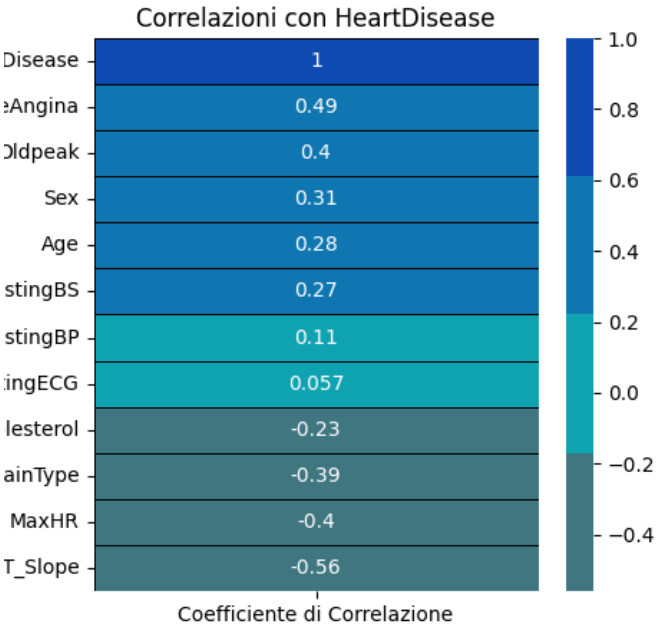
Non sono presenti duplicati.

Grafico Correlazione Valori

Con questa serie di grafici analizzeremo le correlazioni, se presenti, tra Features e Variabile di Target:



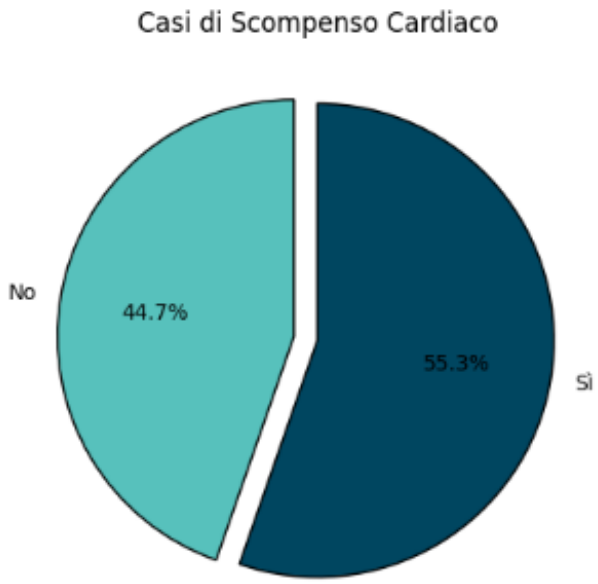
Di questa matrice, selezioniamo solo i valori che sono in correlazione che HeartDisease:



Si nota quindi che la maggior parte delle variabili di feature sono in correlazione con la variabile di target, in particolare, ExcerciseAngina.

Grafico Distribuzione Totale

Con questo grafico a torta, andremo a valutare se il dataset è bilanciato o meno.

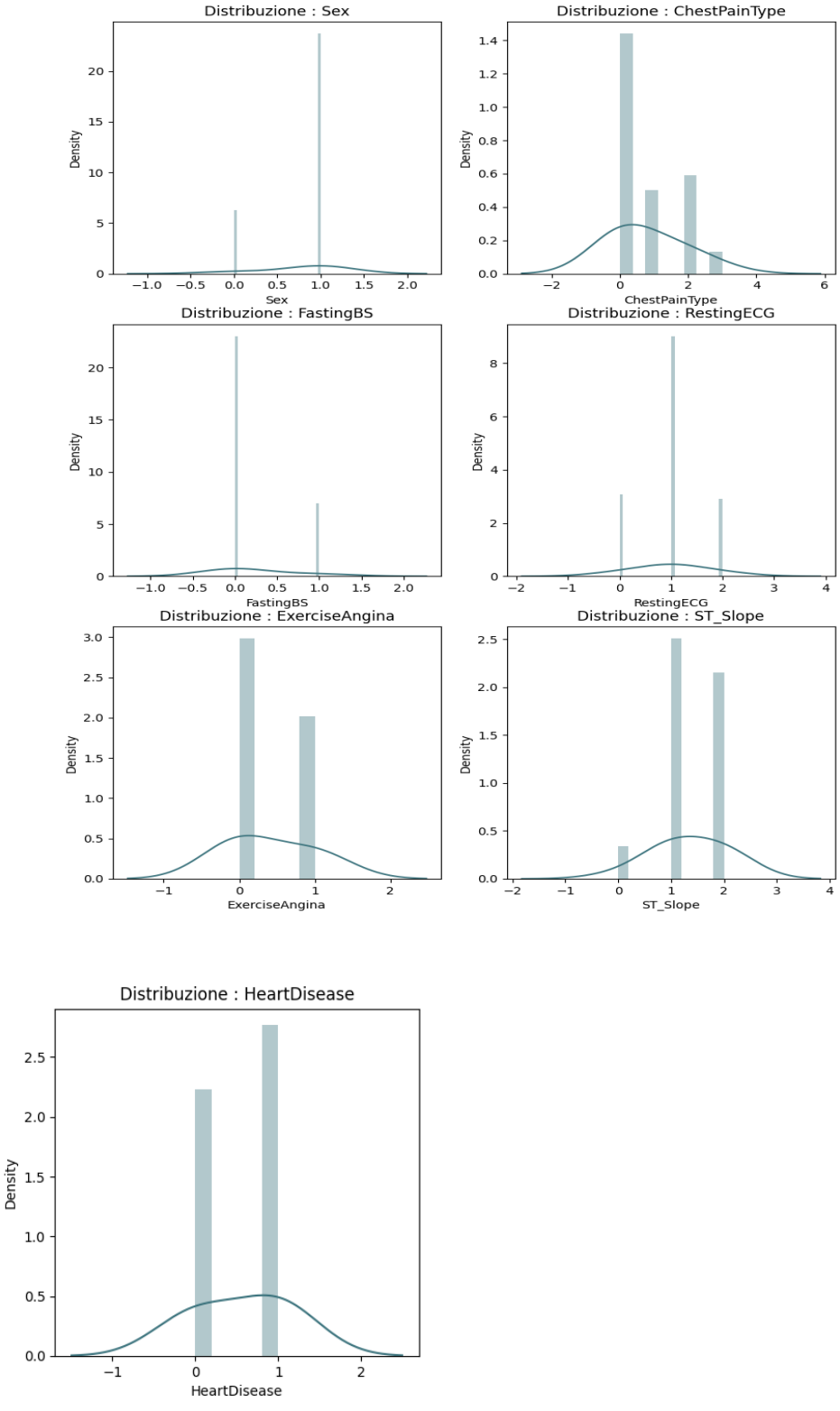


Da come si può notare, i casi si bilanciano, quindi il dataset è bilanciato.

Distribuzione dei dati

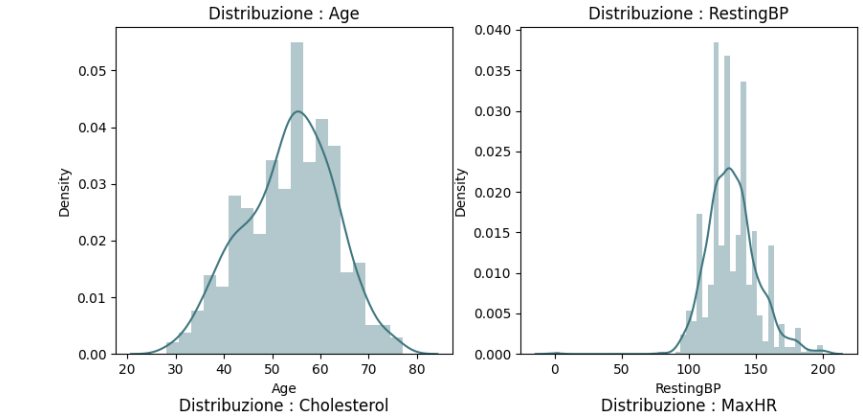
Adesso andremo a visualizzare le varie distribuzioni dei dati, in modo tale da visualizzare eventuali scompensi in termini di valori, che dovranno essere poi corretti in fase di Data Balancing.

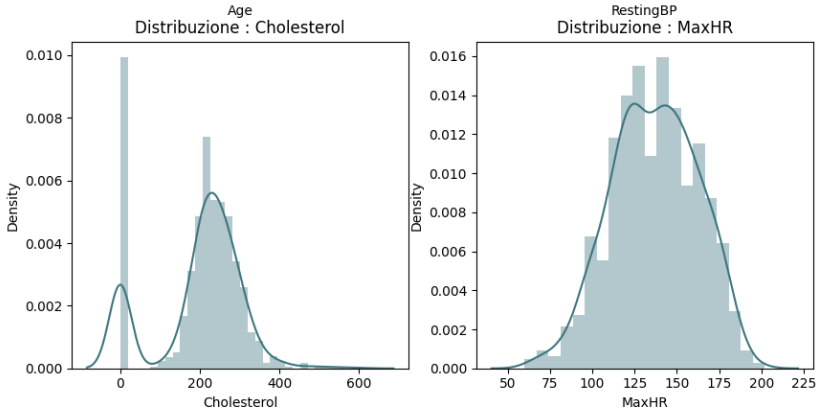
Features Categoriche

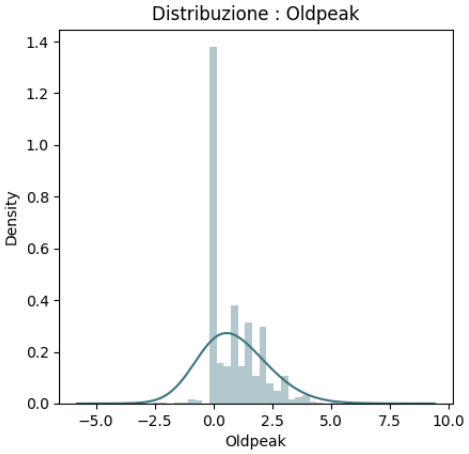


Tutte le features categoriche sono ***normalmente distribuite***.

Features Numeriche





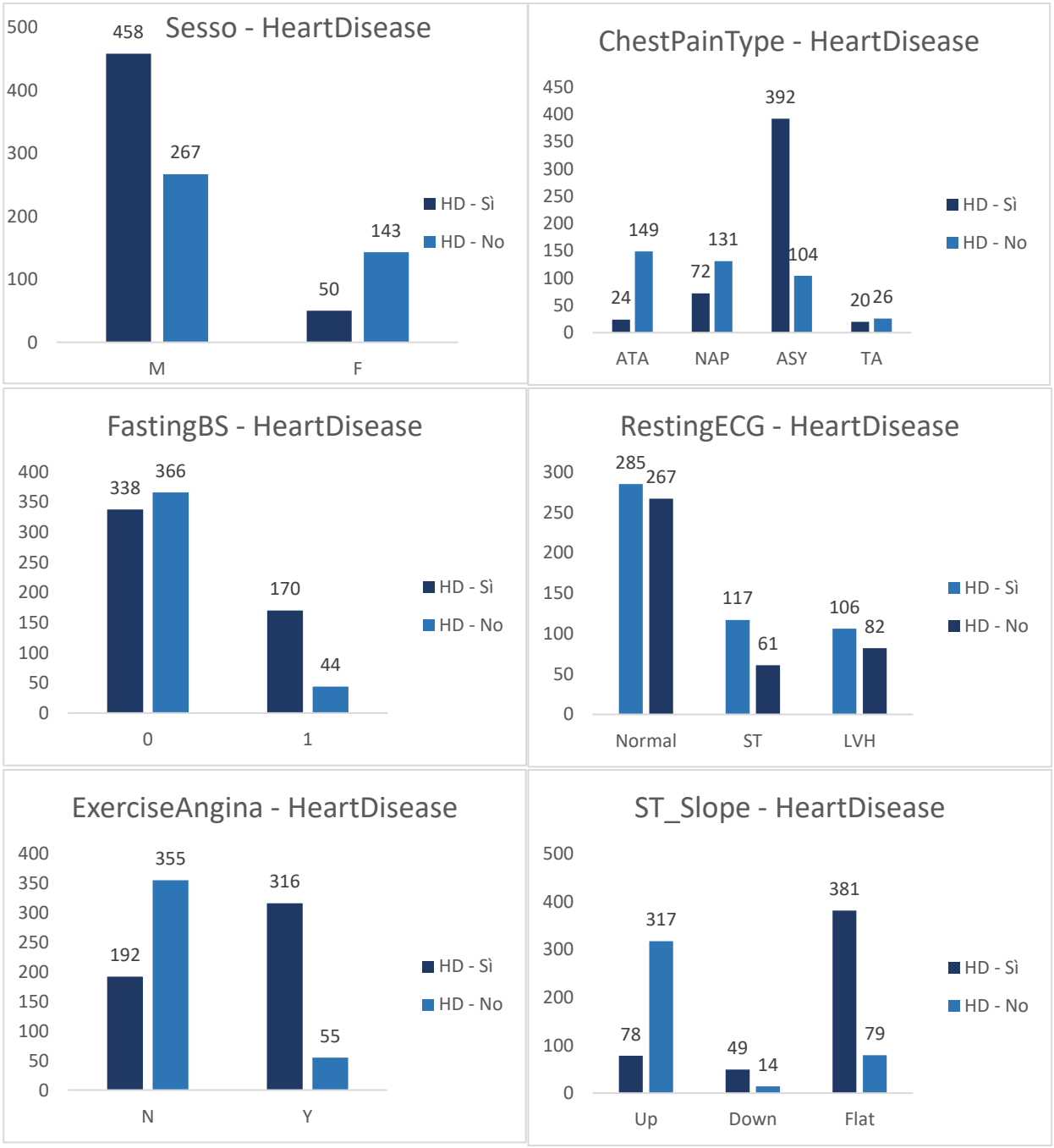


Dai grafici, si nota che l'unico parametro a non avere una distribuzione normale è Oldpeak, che ha una distribuzione skewed, ovvero presenta una notevole asimmetria centrale.

Distribuzione dei Dati rispetto Variabile Target

Infine, in questa fase viene messa in corrispondenza la correlazione tra features categoriche/numeriche con la variabile di target

Features Categoriche



Da questa serie di grafici è possibile notare che gli uomini hanno, probabilmente, una predisposizione agli scompensi cardiaci, che la maggioranza dei casi di scompenso cardiaco non comportano dolore al petto e che l'angina aumenta la probabilità di contrarre uno scompenso cardiaco.

Features Numeriche

Tutte le features numeriche sono distribuite normalmente rispetto ad HeartDisease.

2.4.4 Qualità dei Dati

Da come si è potuto notare dalle fasi precedenti, non vi sono valori nulli, né scompensi del dataset, l'unico problema che sorge, è la scala dei dati, che dovranno essere standardizzati e normalizzati nella fase seguente, la fase di Data Preparation.

Capitolo 3

Data Preparation – Preparazione dei Dati

3.1 Data Preparation

L'obiettivo di questa fase è quello di preparare i dati in modo tale da poter essere utilizzati nelle successive fasi del processo. In primis si includono i processi di pulizia dei dati, poi si selezionano le features che hanno più potere predittivo ed infine i dati vengono

3.1.1 Data Cleaning

Poiché il dataset non presenza istanze vuote e/o valori nulli, questa fase non è necessaria per il corretto procedimento del processo.

3.1.2 Feature Scaling

Per parlare di feature selection, si passa prima per la fase di feature scaling, ovvero, dato che un modello di predizione non può prendere come input valori non numerici e che un modello di predizione tratta i dati senza curarsi della loro unità, è necessaria una fase di Standardizzazione/Normalizzazione.

Viene utilizzata la standardizzazione, per valori che sono normalmente distribuiti, il contrario invece per la normalizzazione.

Nel nostro caso normalizzeremo solo ed esclusivamente *Oldpeak.*

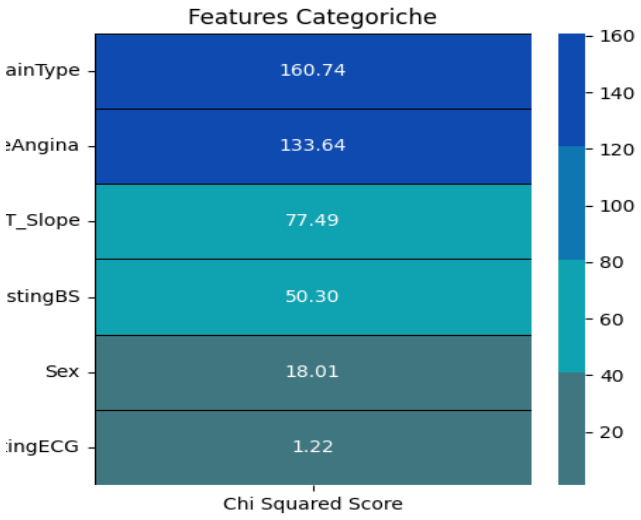
Andando ad utilizzare sklearn come libreria per standardizzare (StandardScaler) e normalizzare (MinMaxScaler), il risultato è il seguente:

Age	Sex	CPT	RestingBP	Cholesterol	FastingBS	RestECG	MaxHR	ExcAng	OldPeak	OPS	HD
-1.43	1	1	0.41	0.83	0	1	1.38	0	0.30	2	0
...	...	...	...	...	...	...	...	...	...	...	...

3.2 Feature Selection

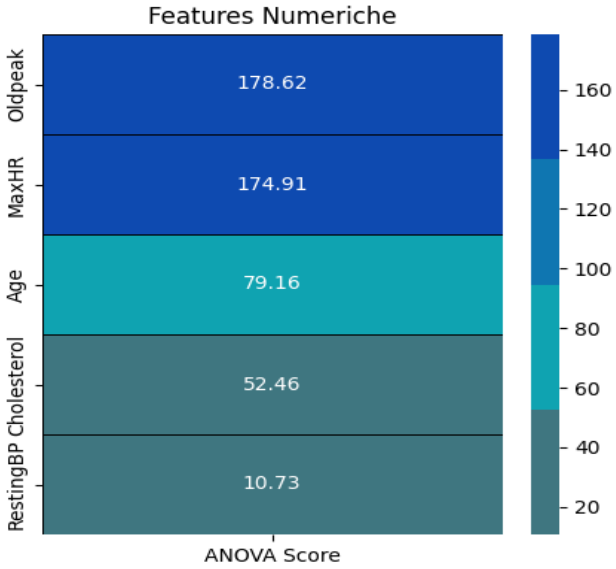
In questa fase utilizzeremo le matrici di correlazione precedentemente ricavate nella fase di Esplorazione dei Dati ed utilizzeremo anche il test del Chi Quadrato, che è un test statistico che serve per determinare se e quali features categoriche influenzano o meno la distribuzione dei dati. Mentre per quanto riguarda le features numeriche utilizzeremo l'ANOVA test, che è un altro test statistico.

3.2.1 Features Categoriche



Da come si può notare, tutte le features categoriche presentano un buon indice di indipendenza tranne per "RestingECG", che quindi non verrà considerata per l'addestramento del modello.

3.2.2 Features Numeriche



Mentre nel caso delle features numeriche, l'unica che presenta un indice relativamente inferiore agli altri è "RestingBP" che quindi non verrà considerata per l'addestramento del modello.

3.3 Data Balancing

La fase di data balancing è necessaria quando si presentano scompensi di casi notevoli all'interno di un dataset, questo perché il modello di ML potrebbe lavorare peggio nella realtà se addestrato su dati prevalentemente di un determinato tipo.

A questo problema si risponde con l'undersampling o con l'oversampling:

- Undersampling: Rimuovere casualmente istanze all'interno della classe di maggioranza.
- Oversampling: Aggiungere copie di istanze all'interno della classe di minoranza.

Queste due tecniche però potrebbero presentare dei problemi, la prima potrebbe causare problemi di rimozione di istanze particolarmente importanti per l'apprendimento del modello; mentre la seconda potrebbe creare *overfitting* ovvero quando il modello risponde accuratamente per dati pregressi ma in modo errato per dati nuovi, questo perché ***ha imparato a "memoria"***.

Nel nostro caso però, avendo un dataset pressoché ***bilanciato*** (la differenza di casi è di un 5% trascurabile) non adopereremo nessuna delle due tecniche. In alternativa si potrebbe pensare di utilizzare l'undersampling poiché statisticamente non ci sono casi ***particolarmente*** rilevanti.

4.2 Introduzione

Dopo la fase di analisi e preprocessing dei dati, vi sussegue la fase di modellazione dei dati; ovvero la fase in cui si andrà a creare un vero e proprio modello. In primis, si sceglie la *tecnica* che si adatta meglio ai dati, ed in secundis si passerà alla fase di *addestramento*, durante la quale verranno configurati i parametri, verrà addestrato il modello e si commenteranno i risultati ottenuti.

Procederemo utilizzando varie tecniche: Naive Bayes, Logistic Regression, Decision Trees e Random Forest.

4.2 Naïve Bayes

Naïve Bayes è un algoritmo classificatore basato sul teorema di Bayes. Assume l'indipendenza tra le features e calcola la probabilità di un input di appartenere ad una determinata classe.

Si chiama "Naïve" proprio perché assume che le variabili siano indipendenti tra loro, quindi non valuterà l'utilità della combinazione di features, proprio perché è basato sul teorema di Bayes.

$$P(A|B) = \frac{\mathrm{P(B\,|\,A)P(A)}}{P(B)};$$

Dove $P(A)$ ed $P(B)$ rappresentano la probabilità di osservare A e B indipendentemente dall'altro, mentre $P(A|B)$ rappresenta la probabilità di osservare A, avendo già osservato B e viceversa per $P(B|A)$.

Dividendo il dataset in training set e prediction set (80/20), si andrà ad utilizzare poi la libreria scikit-learn per implementare un algoritmo Gaussian Naive Bayes.

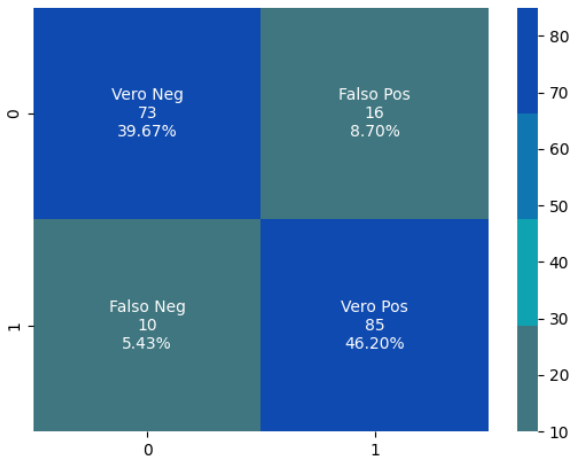
I risultati sono i seguenti:

Su un totale casi di 918, di cui 734 di training e 184 di test, abbiamo ottenuto

Accuracy: 85.87%

	Precision	***Recall***	***F1 – Score***
0 – No Hd	0.88	0.82	0.85
1 – HD	0.84	0.89	0.87

4.2.1 Matrice di Confusione



Grazie a questa matrice è possibile appunto calcolare l'accuratezza delle predizioni. Su un totale di 184 casi, ci sono state 26 predizioni errate.

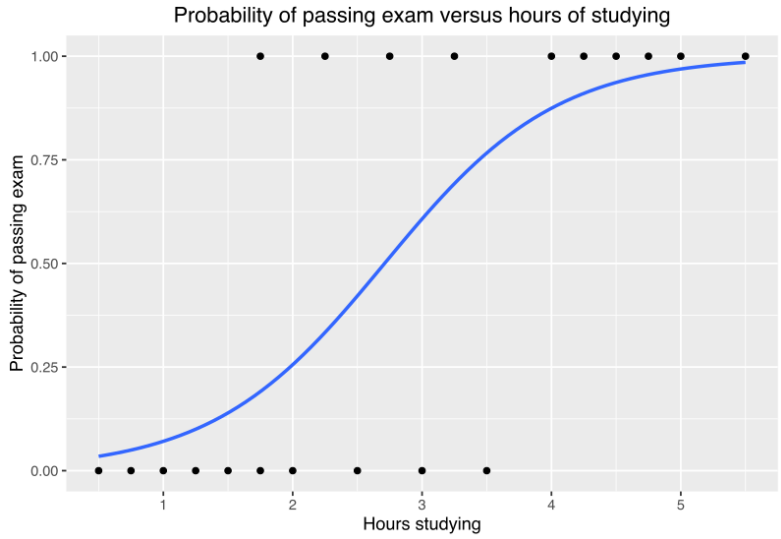
4.3 Logistic Regression

La logistic regression è un tipo di modello statistico utilizzato per la classificazione. La logistic regression stima la probabilità che un evento accada basandosi su di un dataset di variabili indipendenti.

In genere si utilizza quando il valore da predirre dev'essere Vero/Falso, Happy/Sad, Sì/No, come nel nostro caso.

$$P = \frac{1}{1+e^{-(\beta_0 + \beta_1 x)}};$$

Questa formula deriva dalla formula di best-fit della Linear Regression $(\beta_0 + \beta_1 x)$ ma modificata, in modo tale da avere la retta in [0, 1], risultando in una funzione sigmoidale.



Questo grafico mostra la sigmoidale risultante di una Linear Regression.

Un piccolo grafico che rappresenta la retta risultante che mostra la *probabilità di passare un esame* in funzione delle *ore studiate*.

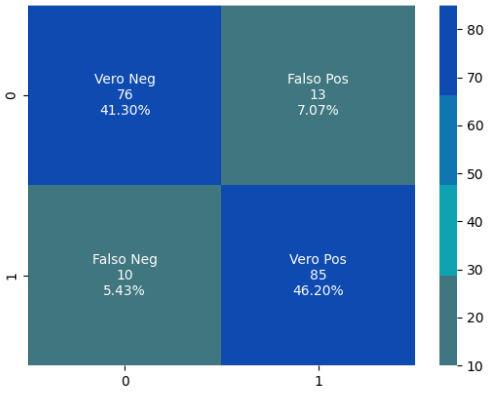
Utilizzando LogisticRegression() della libreria scikit-learn, i risultati sono i seguenti:

Su un totale casi di 918, di cui 734 di training e 184 di test, abbiamo ottenuto

Accuracy: 87.50%

	Precision	***Recall***	***F1 – Score***
0 – No Hd	0.88	0.85	0.87
1 – HD	0.87	0.89	0.88

4.3.1 Matrice di Confusione

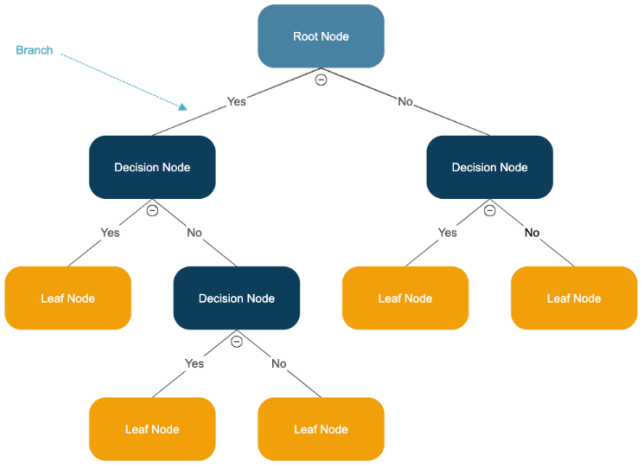


Su un totale di 184 casi, ci sono state 23 predizioni errate.

4.4 Decision Tree

Un Decision Tree è un algoritmo di apprendimento supervisionato, utilizzato sia per attività di classificazione che di regressione. È una struttura gerarchica che si costituisce di un nodo radice, rami, nodi interni e foglie. L'obiettivo di un DT è quello di suddividere il dataset in maniera ricorsiva, ricorrendo sempre sugli attributi che hanno più potenzialità predittive, in base al valore dell'entropia e dell'information gain di una determinata feature, che sia essa numerica o categorica.

L'albero ricorre finché non viene raggiunta una stop-condition, che può essere ad esempio quando tutti gli elementi del sotto-albero attuale hanno tutte istanze che appartengono alla stessa classe restituendo un modello decisionale.



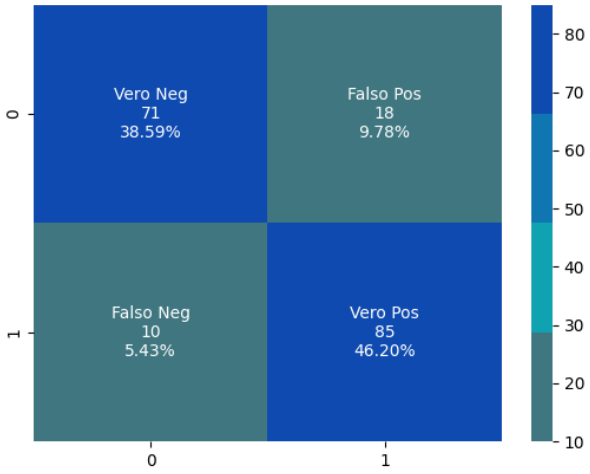
Utilizzando DecisionTreeClassifier() della libreria scikit-learn, con parametro $\max\_depth = 4$ (ovvero se non si raggiunge la purezza dell'informazione entro il 4 livello dell'albero, si interrompe la ricorsione)
i risultati sono i seguenti:

Su un totale casi di 918, di cui 734 di training e 184 di test, abbiamo ottenuto

Accuracy: 84.78%

	Precision	***Recall***	***F1 – Score***
0 – No Hd	0.88	0.80	0.84
1 – HD	0.83	0.89	0.86

4.4.1 Matrice di Confusione

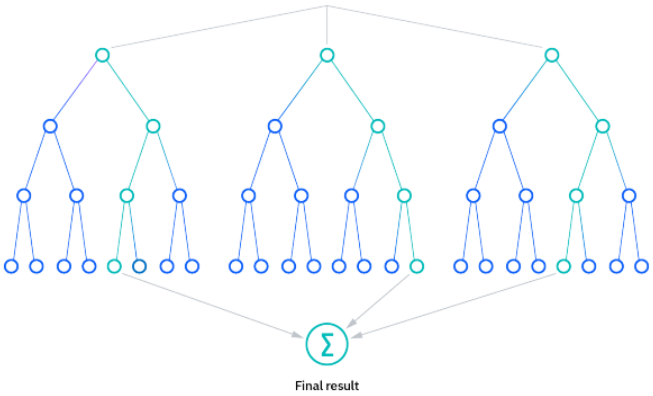


Su un totale di 184 casi, ci sono state 28 predizioni errate.

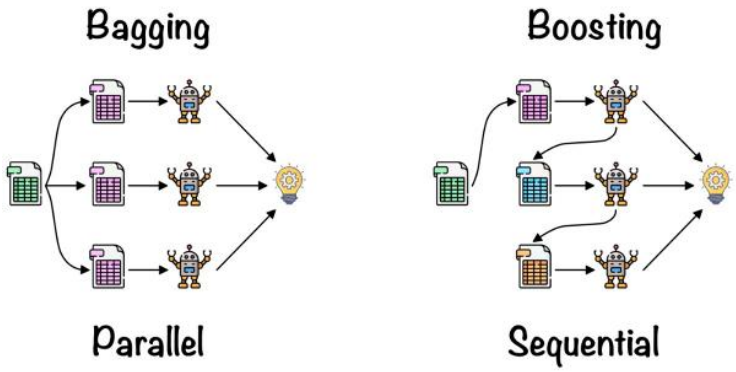
4.5 Random Forest

Random Forest è un algoritmo di machine learning che utilizza l'ensemble learning method, ovvero un metodo che combina l'output di più Decision Trees per raggiungere un singolo risultato. Si utilizza sia per problemi di classificazione che di regressione.

L'idea è quella di creare un insieme di alberi decisionali, ognuno dei quali addestrato su di un sottoinsieme casuale ed indipendente del dateset di training.



Utilizza due tipi di metodi per il raggiungimento di un risultato il *Bagging* ed il *Boosting*. Durante la fase di predizione ogni albero genera un risultato e, in base alla tecnica scelta, si costruisce un output finale.



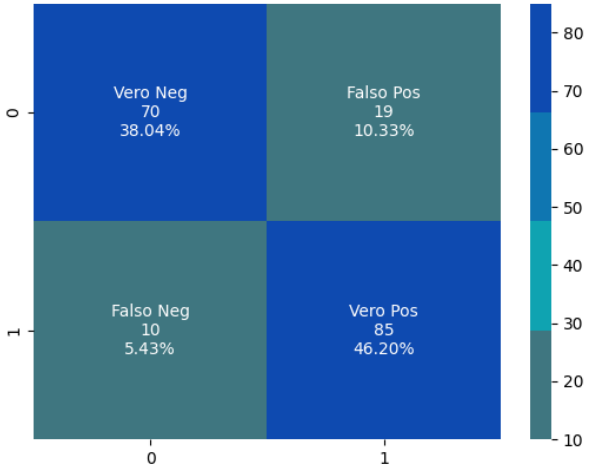
Utilizzando RandomForestClassifier() della libreria scikit-learn, con parametro $\max\_depth = 4$
i risultati sono i seguenti:

Su un totale casi di 918, di cui 734 di training e 184 di test, abbiamo ottenuto

Accuracy: 84.24%

	Precision	***Recall***	***F1 – Score***
0 – No Hd	0.88	0.79	0.83
1 – HD	0.82	0.89	0.85

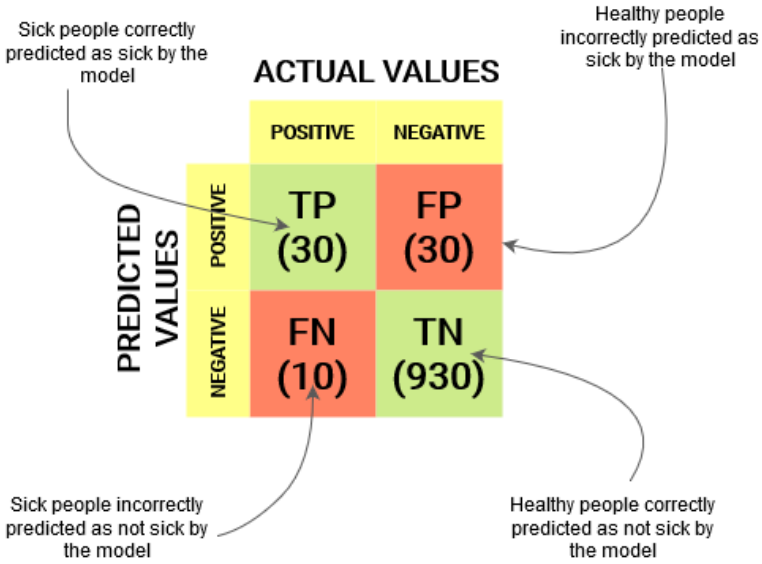
4.5.1 Matrice di Confusione



Su un totale di 184 casi, ci sono state 29 predizioni errate.

5.1 Metriche di Valutazione

Una volta aver creato i modelli, bisogna valutare i risultati ottenuti e per fare ciò ci siamo avvalsi della matrice di confusione, che ci permette di calcolare varie metriche di valutazione.



Le metriche di valutazione in questione sono:

- Accuracy – Rappresenta il rapporto tra tutte le predizioni corrette e il numero di entry del dataset;

$$Accuracy = \frac{TP + TN}{P + N}$$

- Recall – Rappresenta la percentuale di predizioni positive nel totale dei positivi;

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

- Precision – Rappresenta la percentuale dei positivi predetti correttamente;

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

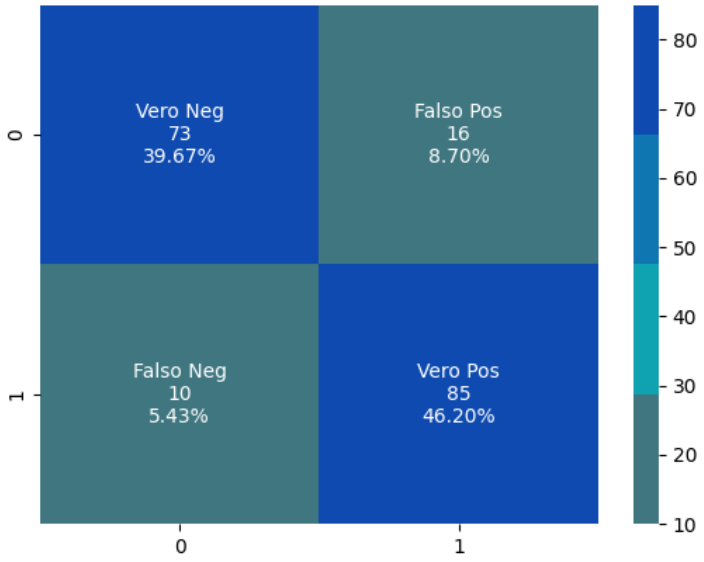
- F1 Score - Rappresenta la media armonica delle metriche Precision e Recall. Variando da 0 a 1.

$$FS = 2 * \frac{Recall * Precision}{Recall + Precision}$$

5.2 Valutazione dei 4 Modelli

Avendo calcolato le matrici e le metriche per ogni modello è giunto adesso il momento di trarre le somme, evidenziando quelli che sono pro e contro di ogni modello ma soprattutto analizzando i risultati ottenuti:

5.2.1 Naive Bayes



Accuracy: 85.87%

	Precision	***Recall***	***F1 – Score***
0 – No Hd	0.88	0.82	0.85
1 – HD	0.84	0.89	0.87

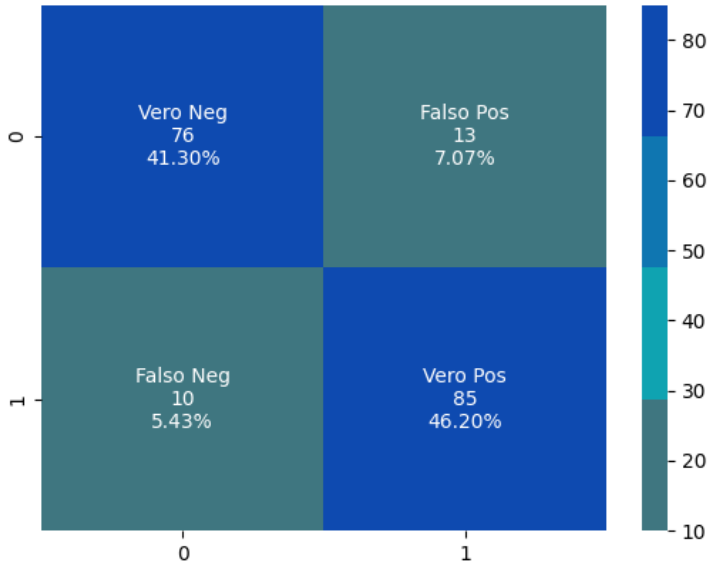
PRO

Da com'è possibile notare NB presenta sia un'accuracy che un F1Score dell'86% rendendo i risultati di questo modello tra i più alti. È anche il più facile da implementare nonché uno dei più veloci in termini di esecuzione

CONTRO

Però Naive Bayes presenta alcuni contro da non sorvolare, uno di questi è appunto l'indipendenza che assume tra le varie features, questo è un dettaglio non banale poiché approcciando le features in maniera dipendente si può ridurre l'errore.

5.2.2 Logistic Regression



Accuracy: 87.50%

	Precision	***Recall***	***F1 – Score***
0 – No Hd	0.88	0.85	0.87
1 – HD	0.87	0.89	0.88

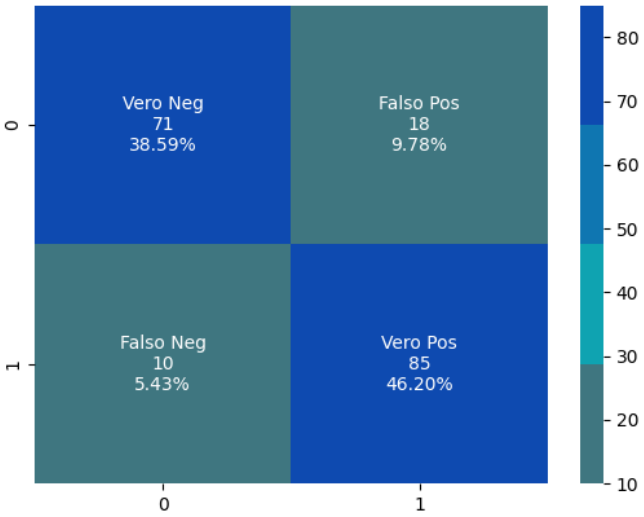
PRO

Logistic Regression è la tecnica con l'accuracy e l'F1Score più alti, entrambi 87.50%. È facile da implementare ed efficiente nella fase di training. Ed è particolarmente performante quando si trattano features linearmente separabili tra di loro.

CONTRO

È una tecnica che presenta overfitting con dati multidimensionali ma soprattutto, fallisce nell'analizzare livelli di dipendenza tra features complessi.

5.2.3 Decision Tree



Accuracy: 84.78%

	Precision	***Recall***	***F1 – Score***
0 – No Hd	0.88	0.80	0.84
1 – HD	0.83	0.89	0.86

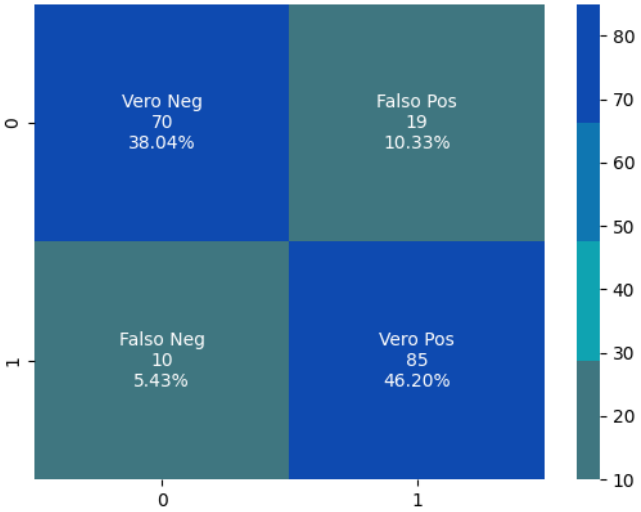
PRO

Facile da costruire ed utilizzare, facile da seguire anche da un punto di vista logico/visivo.

CONTRO

Potrebbe portare a problemi di Overfitting in fase di addestramento in quanto non ha meccanismi per terminare la sua esecuzione, creando, a lungo andare, complesse regole di decisione. Può anche essere altamente time consuming alla presenza di multiple features indipendenti.

5.2.4 Random Forest



Accuracy: 84.24%

	Precision	***Recall***	***F1 – Score***
0 – No Hd	0.88	0.79	0.83
1 – HD	0.82	0.89	0.85

PRO

Così come i DT, in genere, richiedono meno sforzo nella fase di data preparation, in quanto lavorano bene sia con dati normalizzati/standardizzati che non. Sono anche influenzati meno dalla presenza di outlier.

CONTRO

Non sono facilmente spiegabili in quanto non offrono completa visibilità per ogni sotto – albero e possono anche essere dispendiosi in termini di risorse computazionali.

Inoltre, offrono poco controllo sulla personalizzazione del modello.

5.3 Scelta dell'Algoritmo

La scelta dell'algoritmo, una volta aver analizzato pro, contro e prestazioni, è presto fatta, la Logistic Regression, nel nostro caso, vince questo contest, sia per quanto riguarda le performance, avendo un accuracy ed un F1Score più alto, ma anche per quanto riguarda la semplicità implementativa.

Capitolo 6

Conclusioni

6.1 Tirare le Somme

Per concludere questa analisi finale del lavoro svolto sui dati e sui modelli, c'è da fare un piccolo appunto.

Abbiamo analizzato il dataset da un punto di vista statistico, analizzando quelle che erano le correlazioni, le distribuzioni e le performance da un punto di vista numerico.

Abbiamo però trascurato, ovviamente, il lato medico della vicenda, stiamo comunque parlando di pazienti ed ogni feature considerata può essere o meno collegata e correlata a decine di altri fattori che possono essere più o meno correlati alla presenza di scompensi cardiaci.

Facendo un riscontro con alcune conoscenze nel settore, degli esempi lampanti possono essere RestingECG e RestingBP, che noi non abbiamo considerato poiché statisticamente parlando, erano relativamente poco correlati alla presenza di scompensi cardiaci; nel mondo reale però è bene se non necessario considerarle poiché sono collegate a problemi di ipertensione, sintomo quasi sempre presente per quanto riguarda scompensi cardiaci.

Un altro esempio è quello del MaxHR, che noi abbiamo considerato ma che è quasi sempre legato all'età del paziente e non alla presenza o meno di scompensi cardiaci.

Questo per dire che, integrando questo lavoro con un lavoro sui dati, anche dal punto di vista medico, si potrebbero avere performance ancora migliori di quelle ricavate.

Infine, vogliamo dire di essere rimasti estremamente contenti e soddisfatti del lavoro svolto, essendo questo stato il nostro primo Progetto con la 'P' maiuscola ma soprattutto il nostro primo approccio pratico, vero e proprio al mondo dell'IA.

Abbiamo imparato molto, sia teoricamente che praticamente, basti pensare che prima di questo progetto, non avevamo mai toccato con mano né Python benché meno sapevamo quale fosse l'ordine operazionale da seguire per lo svolgimento di un progetto di tale mole.

6.3 Glossario

In questo paragrafo tratteremo quelli che sono stati i termini di micro-lingua che sono stati utilizzati nel documento:

1: Electronic Health Records;

2: Sito web dove sono presenti innumerevoli dataset e pubblicazioni circa la data science;

3: Librerie Python per l'analisi, la visualizzazione e la modellazione dei dati;

4: Differenti tipi di dolore al petto Typical Angina, Atypical Angina, Non – Anginal Pain e Asymptomatic;

5: Differenti valori dell'ECG a riposo ST (anormalità onda ST-T) e LVH (probabile o sicura ipertrofia ventricolare).

6.3 Bibliografia e Sitografia

Le fonti utilizzate per la stesura e la realizzazione di questo progetto sono state:

- *[1] Dataset*

https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

- *[2] Dataset Info*

https://towardsdatascience.com/heart-disease-uci-diagnosis-prediction-b1943ee835a7

- *[3] PEAS Definition*

https://www.okpedia.it/peas

- *[4] DM CRISP Overview*

https://www.ibm.com/docs/it/spss-modeler/18.2.2?topic=dm-crisp-help-overview

- *[5] Decision Tree Overview*

https://www.smartdraw.com/decision-tree/

- *[6] Decision Tree*

https://www.researchgate.net/

- *[7] Random Forest Overview*

https://www.ibm.com/topics/random-forest

- *[8] Confusion Matrix Overview*

https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/

6.4 Ringraziamenti

Infine, in quest'ultimo paragrafo, ci teniamo a ringraziare in primis il professore ed i tutor, per averci accompagnato lungo tutto il percorso.

In secundis, i medici che ci hanno offerto un punto di vista sanitario sui valori ed i parametri presenti nel dataset.