

UNIVERSITÀ POLITECNICA DELLE MARCHE

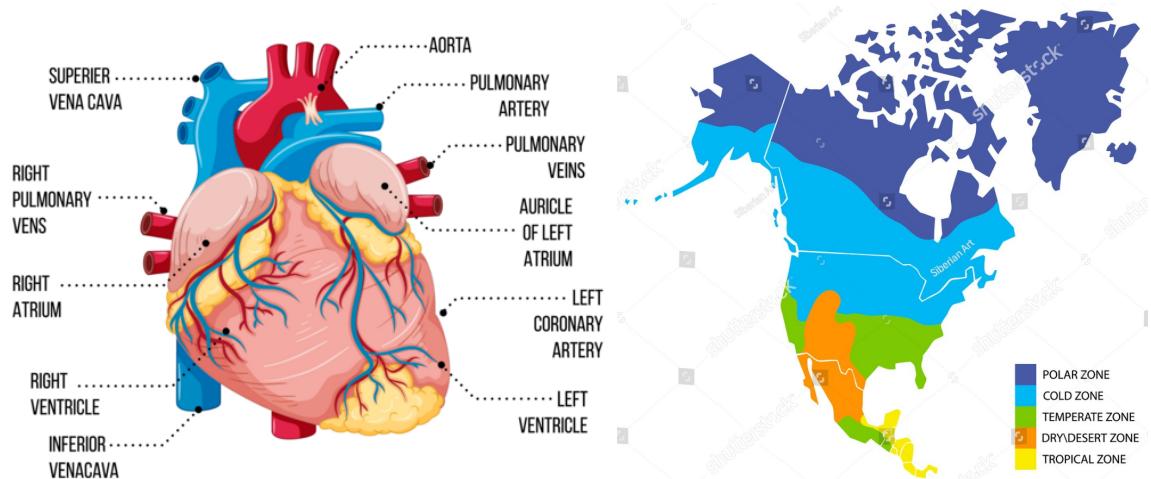
FACOLTÀ DI INGEGNERIA



*Corso di Laurea Magistrale in  
Ingegneria Informatica e dell'Automazione*

*Classificazione e Clustering di un dataset di infarti  
Forecasting Meteo Nord America*

*Progetto Data Science*



Docenti:

**PROF. URSINO DOMENICO**  
**DOTT. MARCHETTI MICHELE**

Studenti:

**CARDONI LORENZO**  
**EL MECHRI RAHMI**  
**NOVELLI GIOVANNI**

ANNO ACCADEMICO 2023-2024

# Indice

<b>1 Heart Attack Analysis</b>	<b>3</b>
1.1 Introduzione Dataset . . . . .	3
1.2 ETL . . . . .	5
1.3 Data Visualization . . . . .	6
<b>2 Clustering</b>	<b>16</b>
2.1 Scikit-Learn . . . . .	16
2.2 Preparazione Dataset . . . . .	17
2.3 Clustering Bidimensionale . . . . .	17
2.3.1 K-Means . . . . .	17
2.3.2 Clustering Gerarchico . . . . .	20
2.4 Clustering Multidimensionale - PCA . . . . .	21
2.4.1 DBSCAN . . . . .	22
2.4.2 K-Means . . . . .	24
2.5 Interpretazione dei risultati dei cluster- K-means . . . . .	25
2.6 Profilazione dei Cluster . . . . .	28
<b>3 Classificazione</b>	<b>30</b>
3.1 Training e Risultati della classificazione . . . . .	30
3.2 Grid Search . . . . .	36
<b>4 Forecasting</b>	<b>41</b>
4.1 Dataset Weather North America . . . . .	41
4.2 Librerie utilizzate . . . . .	43
4.3 ETL . . . . .	43
4.3.1 Analisi della serie temporale . . . . .	44
4.4 SARIMA . . . . .	45
4.4.1 Risultati con Aggregazione Bisettimanale . . . . .	46
4.4.2 Risultati con Aggregazione Trisettimanale . . . . .	48
4.4.3 Risultati con Aggregazione Mensile . . . . .	49
4.5 Analisi dei risultati e conclusioni . . . . .	51

# 1 Heart Attack Analysis

In questa sezione viene presentato il dataset riguardante l'analisi e la previsione degli attacchi cardiaci. Il dataset utilizzato per questa analisi è stato ottenuto dal sito Kaggle, disponibile al seguente ([link](#)). Esso contiene informazioni dettagliate su vari fattori di rischio associati agli attacchi cardiaci, compresi parametri demografici e clinici.

Prima di procedere con l'analisi approfondita, verranno eseguite operazioni di ETL (Estrazione, Trasformazione e Caricamento) e visualizzazione dei dati per preparare il dataset per l'analisi. Queste operazioni ci permetteranno di comprendere meglio la struttura dei dati, identificare eventuali valori mancanti e scoprire correlazioni significative.

Successivamente, ci concentreremo ampiamente sull'applicazione di tecniche di clustering e classificazione. Queste tecniche ci aiuteranno a identificare schemi nei dati e a costruire modelli predittivi per classificare i pazienti a rischio di attacco cardiaco. Attraverso l'uso di algoritmi come K-means e DBSCAN per il clustering, e metodi di classificazione, analizzeremo i risultati per fornire insights utili per la prevenzione e il trattamento degli attacchi cardiaci.

## 1.1 Introduzione Dataset

Di seguito sono riportate le tabelle principali con gli attributi che caratterizzano ciascuna di esse e la relativa descrizione, questo per fornire una migliore comprensione di quelli che saranno poi gli attributi utilizzati nelle successive fasi di analisi.

Attributo	Descrizione
age	Età del paziente
sex	Sesso del paziente
cp	Tipologia di Dolore al Petto: Angina tipica; Angina atipica; Dolore Non Anginoso; Asintomatico
trtbps	Pressione sanguigna a riposo (in mm Hg)
chol	Colesterolo in mg/dl rilevato dal sensore BMI
fbs	Glicemia a digiuno > 120 mg/dl
rest_ecg	Risultati di elettrocardiogramma a riposo: Normale; presenta un'anomalia dell'onda ST-T (inversioni dell'onda T e/o innalzamento o depressione del tratto ST > 0,05 mV); mostra una probabile o certa ipertrofia ventricolare sinistra secondo i criteri di Estes
thalach	Massima Frequenza Cardiaca raggiunta
exang	Angina indotta da esercizio fisico
old_peak	Depressione ST indotta dall'esercizio fisico rispetto al riposo
slp	Pendenza del segmento ST di picco da sforzo: non inclinato; piatto; in discesa
caa	Numero di vasi principali
thall	Talassemia: nullo; difetto fisso; normale; difetto reversibile
output	Diagnosi di cardiopatia (stato di malattia angiografica): restringimento del diametro < 50%. minori possibilità di malattie cardiache; 50% di restringimento del diametro. maggiore probabilità di malattie cardiache

**Tabella 1.1:** Heart.csv

## Breve descrizione dei termini medici usati negli attributi

1. Angina: dolore al petto dovuto alla riduzione del flusso sanguigno ai muscoli del cuore. Esistono 3 tipi di angina: angina stabile, angina instabile e angina variante.
2. Colesterolo: sostanza cerosa presente nelle cellule del corpo e appartenente a un gruppo di molecole organiche chiamate lipidi. Esistono 3 tipi di colesterolo: le lipoproteine ad alta densità (HDL), note come “colesterolo buono”, le lipoproteine a bassa densità (LDL), note come “colesterolo cattivo”, e le lipoproteine a bassissima densità (VLDL) che, come dice il nome, sono particelle a bassa densità che trasportano i trigliceridi nel sangue.
3. Glicemia: Un livello di glicemia a digiuno superiore a 120 mg/dl indica iperglicemia, che può suggerire prediabete (glicemia tra 100 e 125 mg/dl) o diabete mellito di tipo 2 (glicemia di 126 mg/dl o superiore in due test separati). Può anche indicare diabete mellito di tipo 1, caratterizzato dalla mancanza di produzione di insulina. È essenziale fare ulteriori esami e consultare un medico per una diagnosi precisa e un trattamento appropriato.
4. ECG: abbreviazione di elettrocardiogramma, è un esame di routine eseguito di solito per controllare l'attività elettrica del cuore.
5. Depressione ST: un tipo di anomalia del segmento ST. Il segmento ST è la parte piatta e isoelettrica dell'ECG e rappresenta l'intervallo tra la depolarizzazione e la ripolarizzazione ventricolare.
6. Talassemia: è una malattia genetica del sangue caratterizzata da un tasso di emoglobina inferiore al normale.

## 1.2 ETL

Una volta scaricato il dataset e caricato su Jupyter Notebook, sono state effettuate operazioni di pulizia e visualizzazione dei dati mediante le librerie `pandas`, `matplotlib` e `seaborn`. I risultati ottenuti sono mostrati in Figura 1.1

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
--- 
 0   age        303 non-null    int64  
 1   sex        303 non-null    int64  
 2   cp         303 non-null    int64  
 3   trtbps     303 non-null    int64  
 4   chol       303 non-null    int64  
 5   fbs        303 non-null    int64  
 6   restecg    303 non-null    int64  
 7   thalachh   303 non-null    int64  
 8   exng       303 non-null    int64  
 9   oldpeak    303 non-null    float64 
 10  slp        303 non-null    int64  
 11  caa        303 non-null    int64  
 12  thall      303 non-null    int64  
 13  output     303 non-null    int64  
dtypes: float64(1), int64(13)
memory usage: 33.3 KB

```

	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

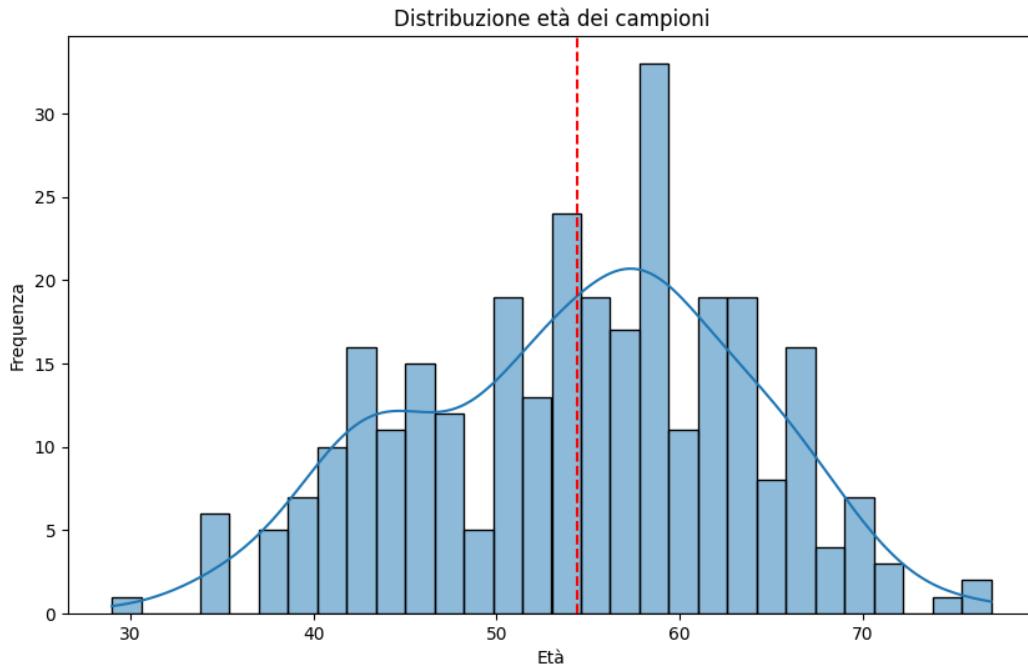
**Figura 1.1:** Dataset Heart Attack

Da quanto si evince sopra, il dataset è composto da 303 pazienti e per ogni paziente 14 features. Inoltre non ci sono valori NaN, né variabili di tipo categorico, quindi gli attributi sono già nel formato corretto per le operazioni di Clustering e Classificazione. Dato che sono 14 features distinte, non si è effettuata nessuna aggregazione.

## 1.3 Data Visualization

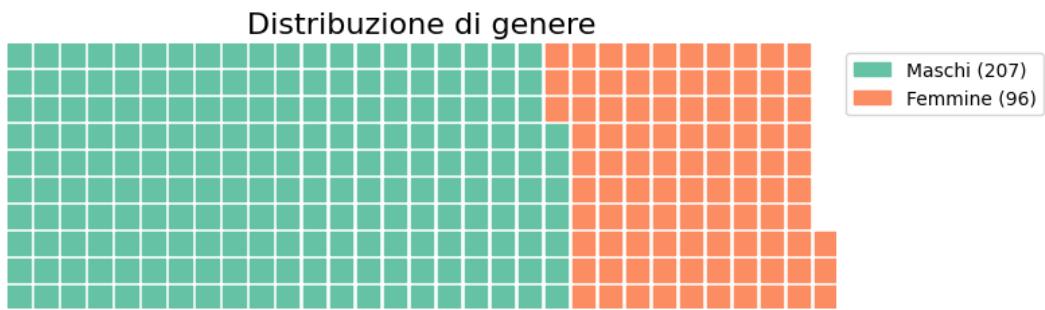
L’analisi descrittiva del dataset mira a fornire una panoramica sulla distribuzione dei vari attributi presenti. Per una migliore comprensione del dataset, sono stati creati vari grafici per visualizzare gli attributi più importanti. La Figura 1.2 riporta le distribuzioni generali degli attributi del dataset con degli istogrammi.

**Figura 1.2:** Istogrammi



**Figura 1.3:** Distribuzione dell'età dei campioni

La Figura 1.3 riportata illustra la distribuzione delle età all'interno del campione esaminato, evidenziando come il gruppo maggiormente rappresentato sia costituito da individui di età compresa tra i 40 e i 67 anni. Questo intervallo comprende la maggior parte dei soggetti nel dataset. L'età media calcolata è di circa 55 anni, il che sottolinea ulteriormente la concentrazione di casi in età medio-avanzata.



**Figura 1.4:** Percentuale di maschi e femmine

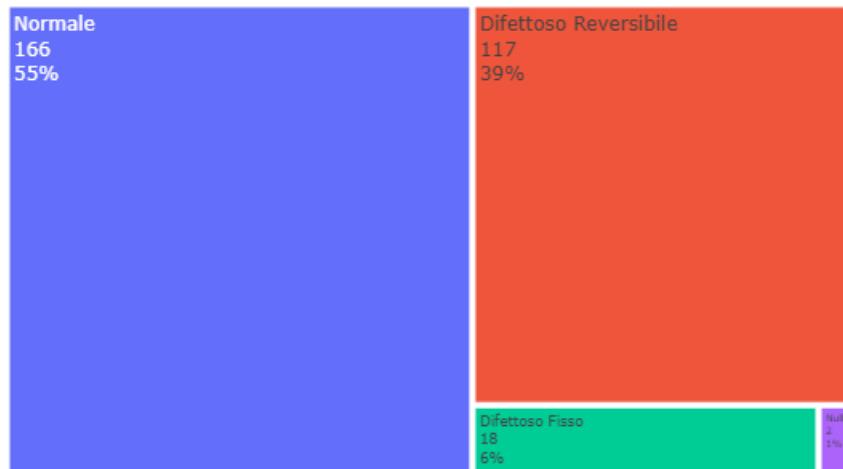
Il campione è prevalentemente maschile con un circa 70% di uomini campionati e un circa 30% di donne campionate (Figura 1.4).

Il diagramma a mappa ad albero, rappresentato in Figura 1.5, visualizza chiaramente la distribuzione percentuale dei diversi tipi di dolore al petto rilevati nel campione. La categoria predominante è l'Angina tipica, che rappresenta il 47% dei casi, seguita dal Dolore non Anginoso con il 39%. L'Angina atipica è riscontrata nel 17% dei soggetti, mentre una minoranza dell'8% risulta asintomatica.



**Figura 1.5:** Mappa ad albero per il tipo di dolore toracico (cp)

In Figura 1.6, il diagramma a mappa ad albero dei risultati di talassemia fornisce una chiara rappresentazione delle diverse condizioni riscontrate nel campione analizzato. La maggior parte dei soggetti, pari al 55%, presenta un livello normale di talassemia, indicativo di una condizione priva di anomalie significative. Un consistente 39% del campione, tuttavia, mostra un difetto reversibile, il che potrebbe implicare una condizione clinica che potrebbe migliorare con il trattamento o col tempo. Un ulteriore 6% dei soggetti è affetto da un difetto fisso, una condizione potenzialmente più grave e non soggetta a miglioramenti spontanei. Infine, un piccolo ma rilevante 1% del campione presenta altre anomalie, che meritano un'analisi più approfondita per comprendere le loro implicazioni cliniche.



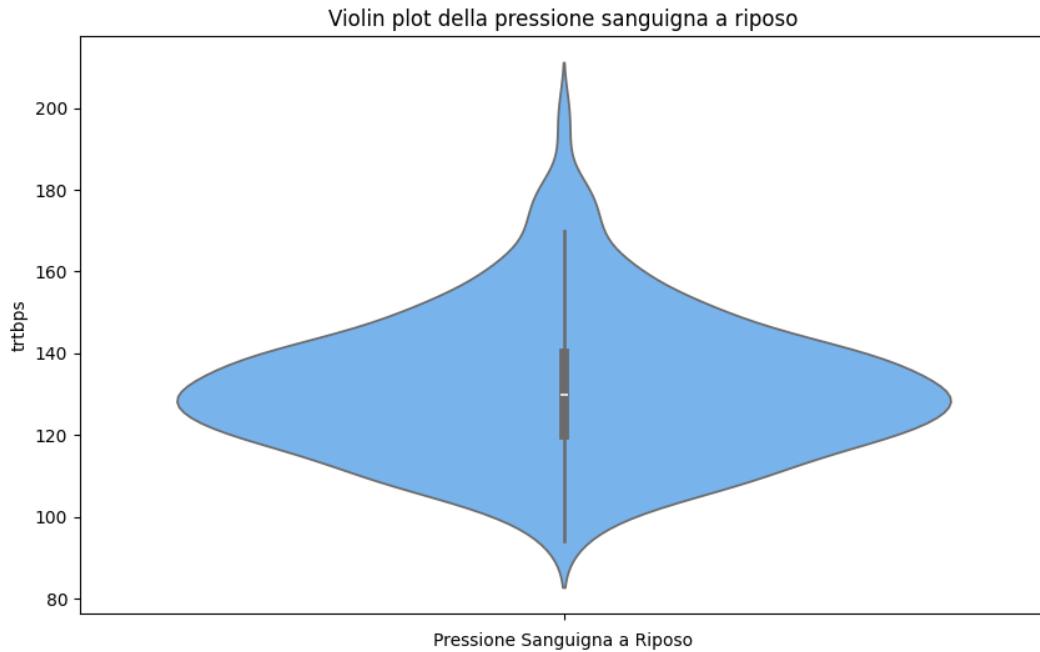
**Figura 1.6:** Mappa ad albero per thall

La Figura 1.7 illustra l'andamento della pressione sanguigna a riposo (trbps) tramite un diagramma a violino, che mostra la distribuzione dei valori nel campione analizzato.

Un diagramma a violino è una rappresentazione grafica che combina le caratteristiche di un box plot e di un kernel density plot, consentendo di visualizzare la distribuzione dei dati e la loro densità.

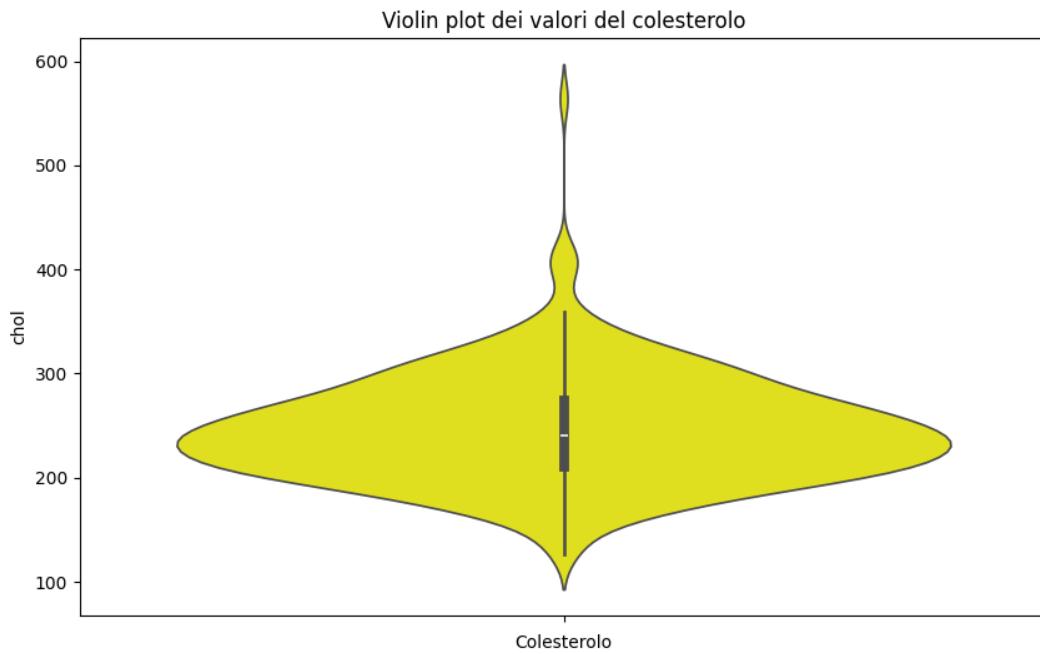
Si osserva una marcata concentrazione dei valori tra i 120 e 140 mmHg, evidenziando una

tendenza centrale in questa fascia. Questo suggerisce che la maggior parte degli individui nel campione ha una pressione sanguigna a riposo all'interno di questo intervallo, che è considerato tipico o leggermente elevato, a seconda del contesto clinico.



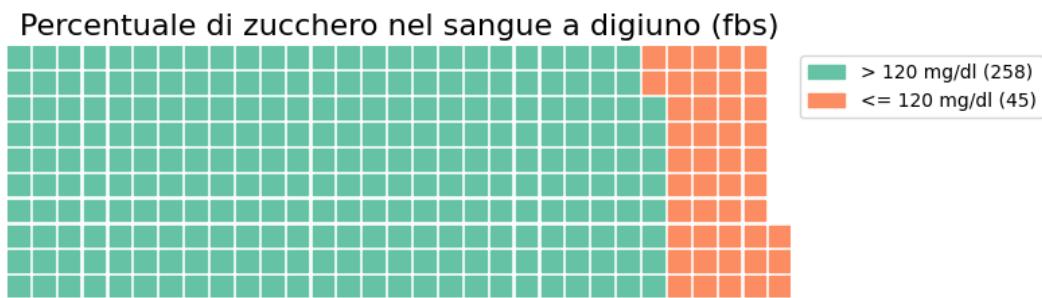
**Figura 1.7:** Andamento della pressione sanguigna a riposo

La Figura 1.8 rappresenta l'andamento dei valori del colesterolo (chol) nel campione analizzato, utilizzando un diagramma a violino. Si nota una concentrazione prevalente di valori tra i 180 e i 320 mg/dL, con una distribuzione che riflette le diverse condizioni lipidiche dei soggetti esaminati. Questo intervallo suggerisce la presenza di un'ampia variabilità nei livelli di colesterolo, con una predominanza di valori che rientrano in un range moderatamente elevato. Il diagramma a violino, grazie alla sua capacità di visualizzare la densità dei dati, consente di individuare con precisione le aree di maggiore concentrazione e di analizzare eventuali deviazioni dalla norma.



**Figura 1.8:** Andamento dei valori del colesterolo

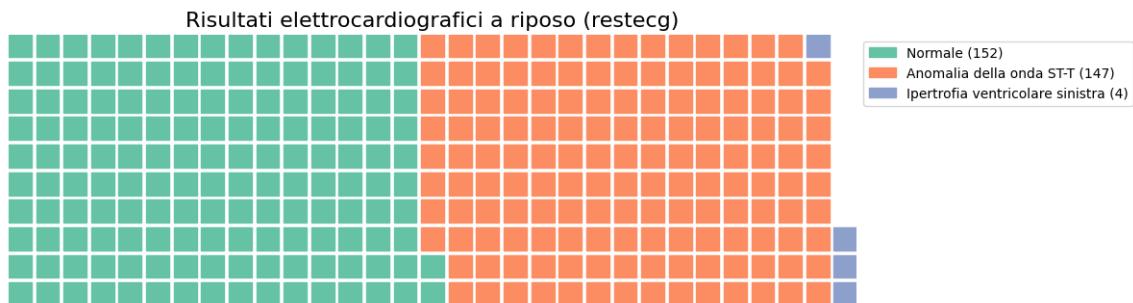
La Figura 1.9 mostra la distribuzione dei livelli di zucchero nel sangue (fbs) nel campione analizzato, rappresentata tramite un Waffle Diagram. Dai dati emerge che solo il 14,9% dei soggetti presenta livelli di zucchero nel sangue inferiori o uguali a 120 mg/dL, mentre una netta maggioranza, pari all'86,1%, registra valori superiori a 120 mg/dL. Questa rappresentazione visiva sottolinea una prevalenza significativa di individui con livelli di glicemia elevati, il che potrebbe indicare una diffusa condizione di iperglicemia nel campione studiato.



**Figura 1.9:** Distribuzione dei livelli di zucchero nel sangue (fbs)

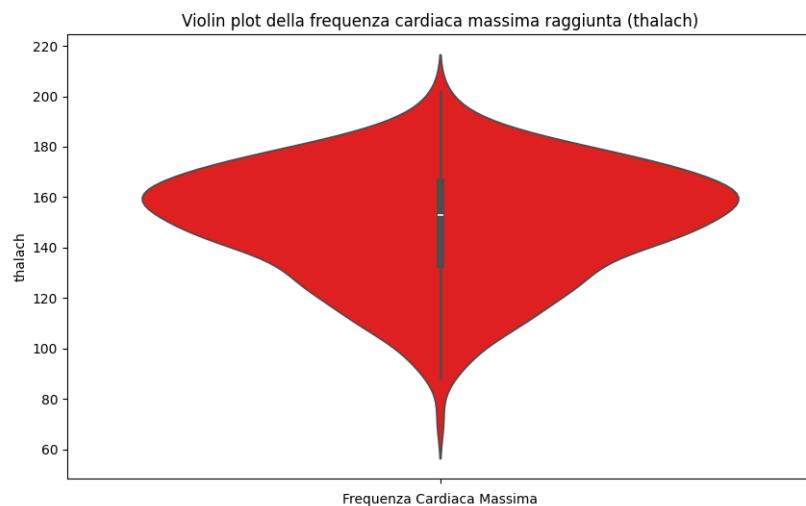
La Figura 1.10 rappresenta la distribuzione dei risultati dell'ECG a riposo (restecg) nel campione esaminato, utilizzando un Waffle Diagram. I dati indicano che il 50,2% dei soggetti presenta un ECG normale, mentre il 48,5% mostra un'anomalia dell'onda ST-T, un segnale potenzialmente indicativo di ischemia o altre condizioni cardiache. Un ulteriore 1,3% del campione evidenzia un'ipertrofia ventricolare sinistra, una condizione associata a un ispessimento del muscolo cardiaco. Questa distribuzione suggerisce una significativa presenza di

anomalie elettrocardiografiche nel campione, con quasi la metà dei soggetti che mostrano deviazioni rispetto ai valori normali.



**Figura 1.10:** Distribuzione dei risultati dell'ECG a riposo (restecg)

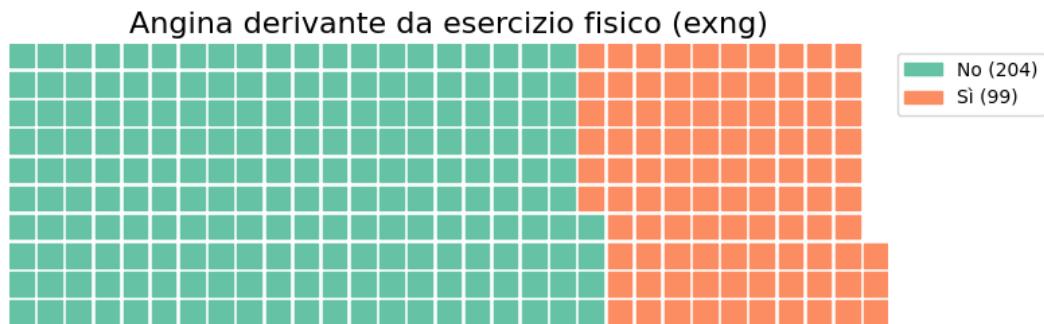
La Figura 1.11 illustra la distribuzione della frequenza cardiaca massima (thalach) ottenuta nel campione, rappresentata mediante un diagramma a violino. Questo tipo di grafico permette di visualizzare sia la distribuzione che la densità dei valori, offrendo una visione dettagliata delle variazioni all'interno del campione. Si osserva che la maggior parte dei valori si concentra tra i 130 e i 180 battiti per minuto, indicando che molti soggetti raggiungono una frequenza cardiaca massima in questo intervallo. Questa concentrazione potrebbe riflettere una normale risposta cardiovascolare durante lo sforzo fisico o altre condizioni di stress.



**Figura 1.11:** Distribuzione della frequenza cardiaca massima (thalach)

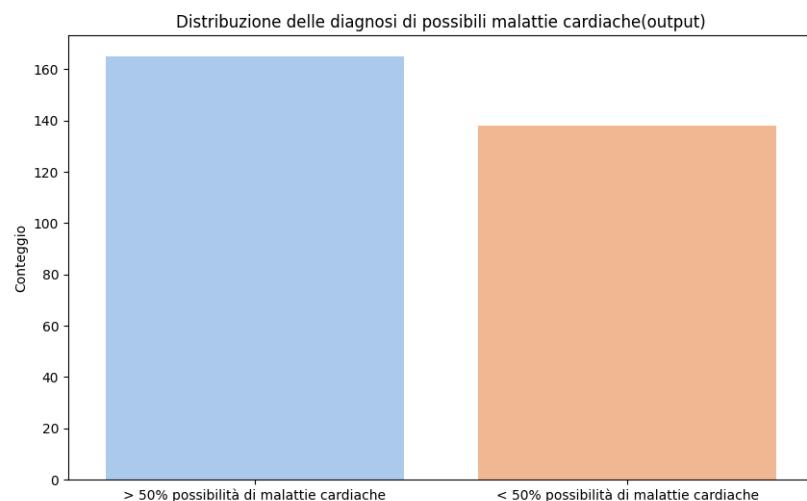
La Figura 1.12 mostra la distribuzione della variabile exang, rappresentata tramite un Waffle Diagram. I dati rivelano che il 32,7% dei soggetti ha manifestato angina derivante da sforzo fisico, mentre il restante 67,3% non ha riportato angina legata a sforzo fisico. Questo grafico evidenzia una prevalenza significativa di individui che non sviluppano sintomi anginosi in risposta all'attività fisica, suggerendo che, per la maggior parte del campione, l'angina non è provocata da sforzi fisici. Il Waffle Diagram fornisce una chiara rappresentazione delle

proporzioni tra i due gruppi, facilitando la comprensione dell'incidenza dell'angina da sforzo nel contesto dell'analisi clinica del campione.

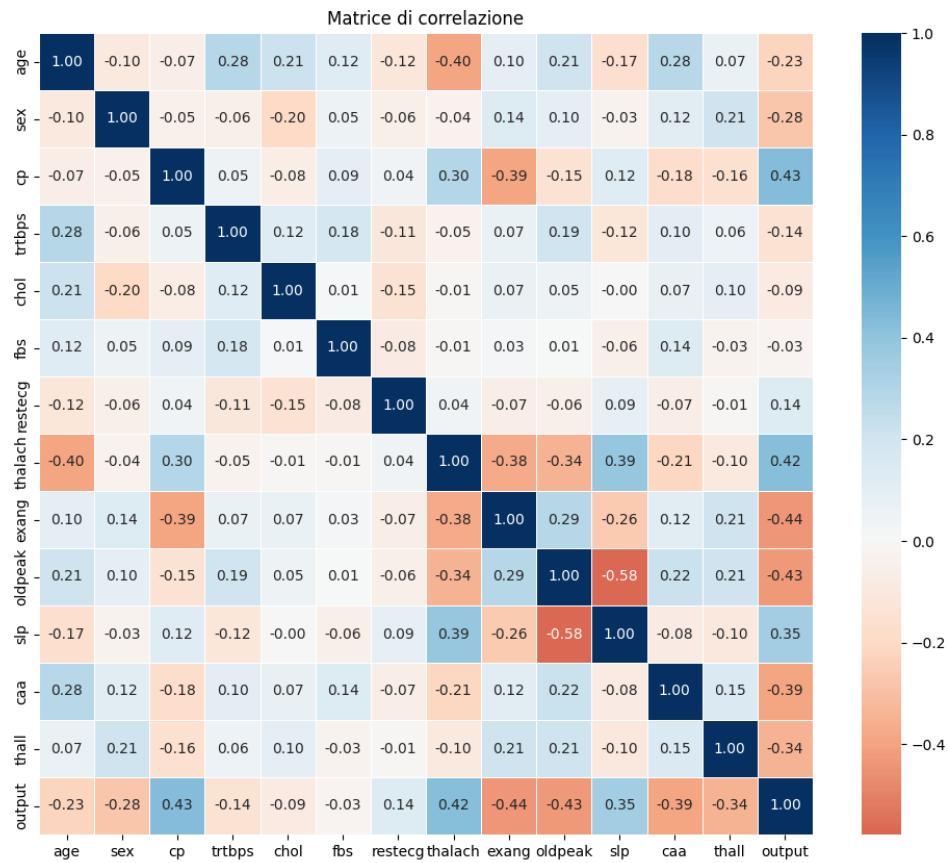


**Figura 1.12:** Distribuzione della variabile ‘exang’

La Figura 1.13 illustra la distribuzione dell'output predittivo, rappresentata tramite un grafico a barre. I dati mostrano che 160 campioni presentano una probabilità superiore al 50% di sviluppare malattie cardiache, mentre quasi 140 campioni hanno una probabilità inferiore al 50%. Questo grafico evidenzia una leggera prevalenza di soggetti con un rischio elevato di malattie cardiache all'interno del campione. La rappresentazione a barre permette di visualizzare chiaramente la differenza numerica tra i due gruppi, fornendo un quadro immediato della distribuzione delle probabilità di malattia nel campione analizzato e facilitando l'identificazione delle aree a maggior rischio.



**Figura 1.13:** Distribuzione dell'output



**Figura 1.14:** Matrice di correlazione tra tutti i dati

La Figura 1.14 presenta la matrice di correlazione tra tutte le variabili del dataset, rappresentata come ultimo grafico dell’analisi. I valori di correlazione, che indicano la forza e la direzione della relazione lineare tra due variabili, variano da -0.6 a 0.4, escludendo i valori sulla diagonale che sono pari a 1.00, poiché rappresentano la correlazione di ciascuna variabile con sé stessa.

Tra le correlazioni negative più significative, notiamo una forte correlazione inversa tra oldpeak e slp con un valore di -0.58, suggerendo che un aumento dell’oldpeak è associato a una diminuzione dei valori di slp. Altre correlazioni negative rilevanti includono thalach con age (-0.40), exang con cp (-0.39), output con exang (-0.44), output con oldpeak (-0.43), output con caa (-0.39), e output con thall (-0.34). Questi valori negativi indicano che, ad esempio, un aumento dell’output è associato a una riduzione nei valori di exang o oldpeak, suggerendo potenziali relazioni inverse tra queste variabili.

Per quanto riguarda le correlazioni positive, con valori superiori a 0.3, si evidenzia una correlazione di 0.30 tra thalach e cp, una correlazione di 0.39 tra slp e thalach, e una correlazione di 0.43 tra output e cp. Inoltre, output mostra correlazioni positive con thalach (0.42) e

slp (0.35). Queste correlazioni positive indicano che variabili come cp, thalach, e slp tendono a variare nella stessa direzione rispetto all'output, suggerendo che aumenti in queste variabili sono associati a un aumento della probabilità di malattia cardiaca (rappresentata da output).

La matrice di correlazione offre una panoramica completa delle interrelazioni tra le variabili, facilitando l'identificazione di quelle che possono avere un impatto significativo sull'output, e dunque sulla probabilità di malattie cardiache nel campione analizzato.

# 2 Clustering

Dopo aver completato le analisi descrittive sul dataset relativo agli arresti cardiaci, al fine di comprendere meglio le caratteristiche intrinseche dei dati, si procede con l'applicazione di tecniche di clustering. Questa metodologia di apprendimento non supervisionato consente di raggruppare pazienti con profili simili, basandosi su variabili come età, frequenza cardiaca massima ed altre caratteristiche mediche. L'obiettivo di questa analisi è quello di fornire informazioni utili per migliorare i trattamenti e la prevenzione. Inoltre, l'analisi dei cluster può aiutare i medici a prendere decisioni più informate, consentendo interventi sanitari più specifici e personalizzati, con l'obiettivo di ottimizzare le risorse e migliorare i risultati per i pazienti.

## 2.1 Scikit-Learn

Scikit-learn è una delle librerie principali di Python per il machine learning, basata su NumPy, SciPy e Matplotlib. Questa libreria consente di eseguire una vasta gamma di task di machine learning, tra cui:

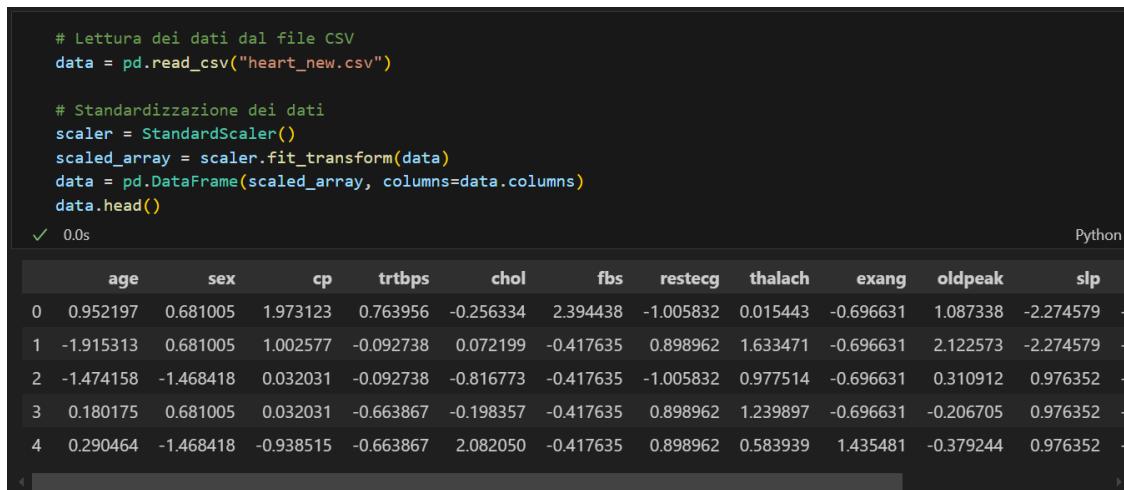
- **Classificazione:** Utilizza algoritmi come SVM per classificazioni a grandi dimensioni, K-nearest neighbors, random forest, e altri.
- **Regessione:** Simile alla classificazione, ma gli attributi da predire sono numerici anziché categorici.
- **Clustering:** Consiste nel raggruppamento di elementi privi di una label precisa, lasciando all'utente il compito di scoprire cosa accomuna gli elementi raggruppati in base alle loro caratteristiche.



Figura 2.1: Logo Scikit Learn

## 2.2 Preparazione Dataset

Prima di procedere alle operazioni di clustering, sono state svolte alcune analisi necessarie sul dataset per renderlo conforme alle necessità degli algoritmi che verranno utilizzati. Questo dataset contiene già tutte variabili di tipo numerico, così da non avere problemi con le possibili variabili categoriche che il modello di Machine Learning non accetta in input. Inoltre, avendo constatato che i valori degli attributi appartenessero a scale di grandezze diverse, è stata effettuata un'operazione di standardizzazione, necessaria al fine di poter dare in input, ai modelli di clustering, dati con metriche simili. Per quest'ultima operazione si è scelto di utilizzare lo StandardScaler del modello preprocessing di ScikitLearn. La Figura 2.2 mostra il dataset al termine delle operazioni di preparazione, idoneo per l'applicazione delle successive tecniche di clustering K-Means e DBSCAN



```
# Lettura dei dati dal file CSV
data = pd.read_csv("heart_new.csv")

# Standardizzazione dei dati
scaler = StandardScaler()
scaled_array = scaler.fit_transform(data)
data = pd.DataFrame(scaled_array, columns=data.columns)
data.head()

✓ 0.0s
```

The screenshot shows a Jupyter Notebook cell with the following content:

```
# Lettura dei dati dal file CSV
data = pd.read_csv("heart_new.csv")

# Standardizzazione dei dati
scaler = StandardScaler()
scaled_array = scaler.fit_transform(data)
data = pd.DataFrame(scaled_array, columns=data.columns)
data.head()

✓ 0.0s
```

Below the code, a preview of the DataFrame is shown:

	age	sex	cp	trtbps	chol	fbp	restecg	thalach	exang	oldpeak	sip
0	0.952197	0.681005	1.973123	0.763956	-0.256334	2.394438	-1.005832	0.015443	-0.696631	1.087338	-2.274579
1	-1.915313	0.681005	1.002577	-0.092738	0.072199	-0.417635	0.898962	1.633471	-0.696631	2.122573	-2.274579
2	-1.474158	-1.468418	0.032031	-0.092738	-0.816773	-0.417635	-1.005832	0.977514	-0.696631	0.310912	0.976352
3	0.180175	0.681005	0.032031	-0.663867	-0.198357	-0.417635	0.898962	1.239897	-0.696631	-0.206705	0.976352
4	0.290464	-1.468418	-0.938515	-0.663867	2.082050	-0.417635	0.898962	0.583939	1.435481	-0.379244	0.976352

**Figura 2.2:** Standardizzazione Dataset

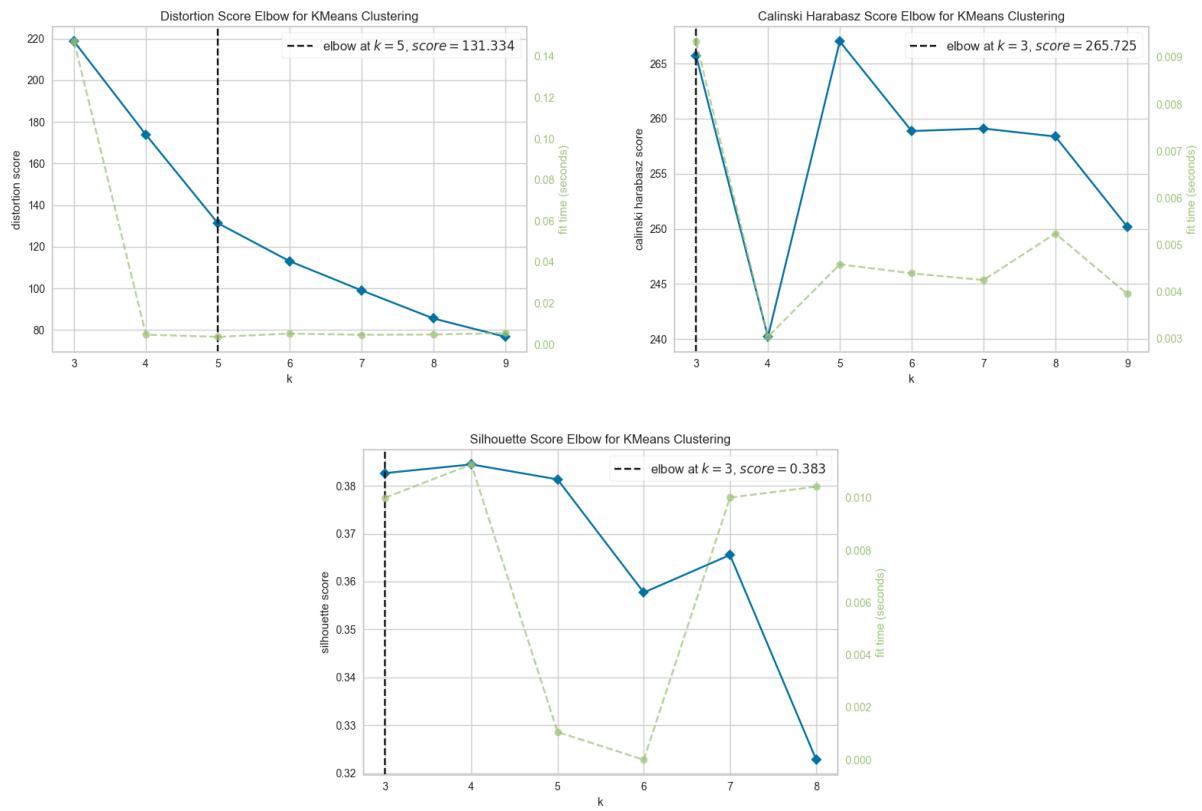
## 2.3 Clustering Bidimensionale

Il primo tipo di clustering che si vuole effettuare prende in considerazione solamente due features specifiche del dataset. Basandosi sull'osservazione della heatmap in Figura 1.14, sono stati scelti gli attributi "age" e "thalach", rispettivamente l'età del paziente e la frequenza cardiaca massima, in quanto soggetti a una correlazione molto elevata rispetto alla combinazione tra gli attributi restanti.

### 2.3.1 K-Means

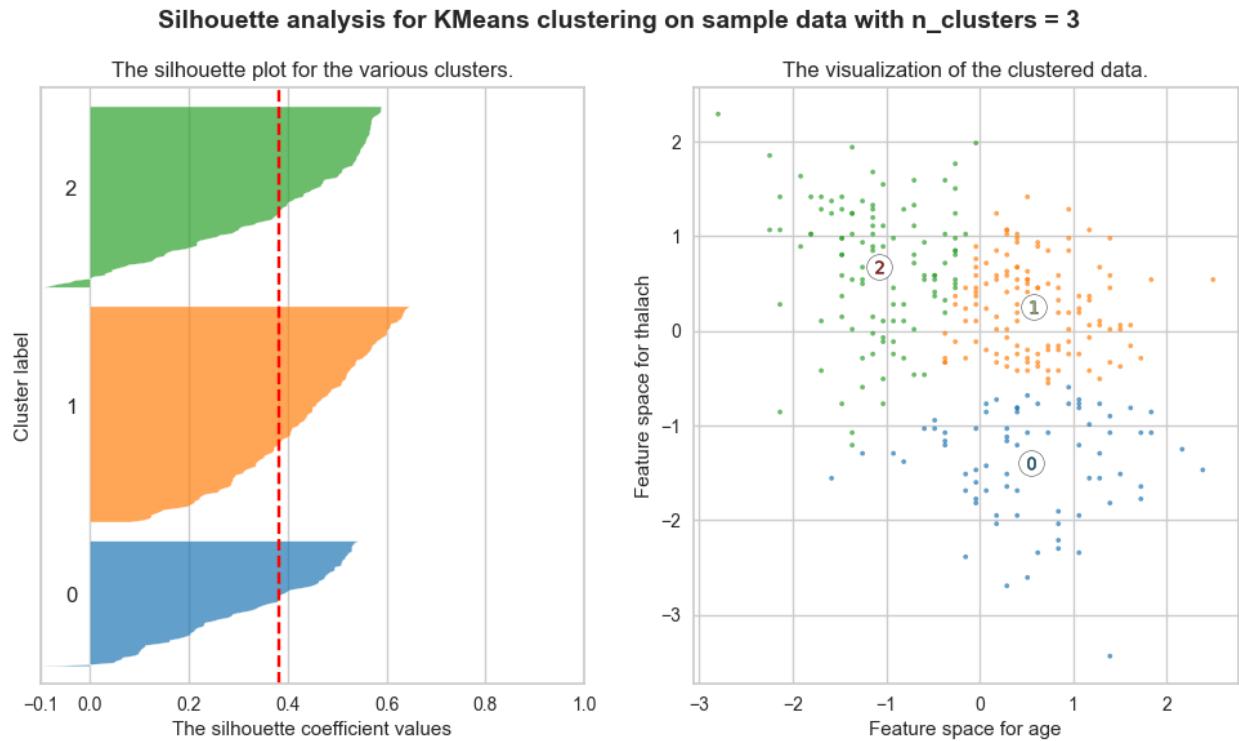
Una volta scelti i due attributi su cui effettuare la prima analisi di clustering, si è scelto di utilizzare l'algoritmo K-Means. La caratteristica fondamentale di tale tecnica è che essa richiede di conoscere in anticipo il parametro  $k$ , ovvero il numero di cluster in cui si intende segmentare i dati. Al fine di individuare tale informazione, quindi, è stato utilizzato, tramite la libreria Yellowbrick, il cosiddetto metodo del gomito (Elbow Method), un'euristica che ricava, tramite iterazioni del K-Means, il parametro  $k$  ottimale. In questo caso, sono state applicate tre tecniche dell'elbow method che utilizzano metriche diverse:

- Distorsione: misura la somma delle distanze al quadrato tra i punti e il centroide del cluster. Si cerca un punto di "gomito" nella curva, dove la riduzione della distorsione diminuisce significativamente, suggerendo il valore ottimale di k.
- Indice di Calinski-Harabasz: valuta la coesione e la separazione dei cluster; valori più alti indicano cluster ben definiti. L'elbow method cerca un valore ottimale in cui il miglioramento diventa marginale.
- Indice di Silhouette: misura la coerenza dei cluster confrontando la distanza media tra punti all'interno dello stesso cluster e tra cluster diversi. Il valore di k ottimale è suggerito dal punto di "gomito" della curva dell'indice di silhouette.



**Figura 2.3:** Metodi Elbow K-Means

I grafici ottenuti sono mostrati nella Figura 2.3. Si può notare che il metodo della distorsione dà un  $K = 5$ , mentre calinski e silhoutte danno un  $k = 3$ . A livello di codice, si è andato a confrontare i valori di k suggeriti da ciascuna metrica e seleziona il più comune come valore ottimale di k. Dunque, a questo punto si esegue il K-Means con numero di cluster preimpostato a 3. Il risultato del clustering e le silhouette ottenute sono mostrate in Figura 2.4



**Figura 2.4:** K-Means Clustering

Come è possibile notare dalla Figura 2.4, i tre cluster sono distribuiti lungo il grafo, ognuno centrato nel proprio centroide, evidenziato con un pallino bianco contenente il numero del cluster. Da questa preliminare rappresentazione, si può distinguere come ogni cluster contenga un sottoinsieme differente di pazienti:

- Cluster 0: contiene pazienti adulti ed anziani con frequenza cardiaca massina sotto alla media
- Cluster 1: contiene pazienti adulti ed anziani con frequenza cardiaca massina sopra alla media
- Cluster 2: contiene pazienti giovani ed adulti con frequenza cardiaca massina sopra alla media

Tuttavia, si ricorda che sugli assi non sono riportati i valori assoluti delle età e della frequenza cardiaca massina ma i valori normalizzati a seguito della precedente operazione di standardizzazione.

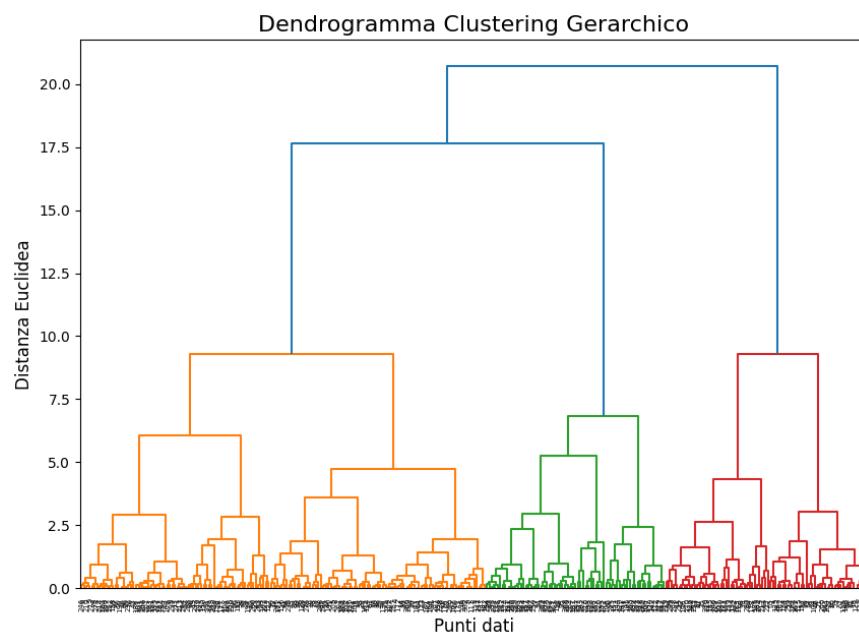
Dopodichè, per valutare la bontà dei risultati ottenuti, è stata disegnata la metrica della Silhouette, mostrata in Figura 2.4. Essa assegna un valore in un range compreso tra -1 e +1, fornendo un'indicazione di quanto un determinato punto appartiene correttamente al cluster a cui è stato classificato. Come si può osservare, si ottiene uno score medio vicino allo 0.4, il quale indica una buona clusterizzazione. Fanno eccezione alcuni elementi, seppur pochi, appartenenti al cluster 2, i quali presentano uno score negativo, a indicare che potevano essere rappresentati meglio da uno altro gruppo di cluster.

### 2.3.2 Clustering Gerarchico

Per avere un termine di paragone con il metodo K-Means, è stato applicato il clustering gerarchico sugli stessi attributi, per verificare se un diverso algoritmo di machine learning avrebbe prodotto risultati analoghi. A differenza del K-Means, il clustering gerarchico non richiede a priori la definizione del numero di cluster e organizza i gruppi in una struttura gerarchica ad albero. Questo metodo segue un approccio di tipo bottom-up: inizia considerando le singole unità statistiche e le aggrega progressivamente in gruppi più grandi, basandosi su criteri di similarità.

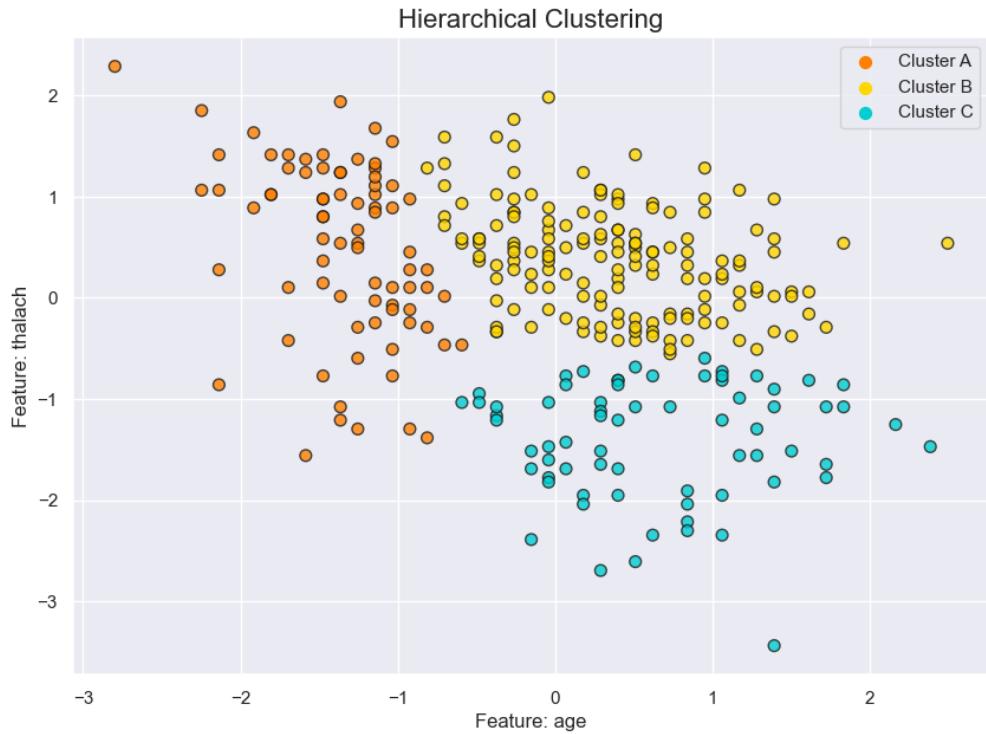
In Figura 2.5 è mostrato il dendrogramma ottenuto applicando il clustering gerarchico sui dati. Il dendrogramma rappresenta la gerarchia di raggruppamento, visualizzando le distanze alle quali i punti e i cluster vengono uniti.

Per identificare il numero ottimale di cluster, si possono osservare i "salti" nel dendrogramma, ossia le distanze verticali più ampie tra i rami. Questi salti indicano la fusione di cluster con una distanza considerevole, suggerendo che i cluster uniti a quel livello erano ben distinti. In questo caso, si nota un salto significativo intorno a una distanza di circa 10. Tagliando il dendrogramma a questa altezza, si ottengono tre rami principali, corrispondenti a tre cluster distinti. Pertanto, il numero ideale di cluster per questa suddivisione è 3.



**Figura 2.5:** dendrogramma

In Figura 2.6 è riportato il risultato del clustering ottenuto utilizzando il modello *AgglomerativeClustering* di **scikit-learn**, con il numero di cluster impostato a 3 e il criterio di fusione "ward". Quest'ultimo minimizza la devianza totale dal centroide del gruppo al momento della fusione, garantendo una migliore compattezza dei cluster.



**Figura 2.6:** cluster gerarchico

La suddivisione in cluster ottenuta in seguito all'applicazione del clustering gerarchico non è molto differente da quella individuata dal K-Means;

## 2.4 Clustering Multidimensionale - PCA

Oltre alle analisi svolte utilizzando solo due caratteristiche, è stato deciso di utilizzare tutte le colonne del dataset per ottenere un quadro generale. A tal fine, si è impiegata una tecnica di riduzione della dimensionalità che consente di conservare le informazioni più rilevanti del dataset: l'Analisi delle Componenti Principali (PCA). Grazie a questa tecnica, è possibile comprimere l'intero dataset in un nuovo dataframe con sole due colonne, che rappresentano le componenti principali e racchiudono la maggior parte delle informazioni originarie. Il risultato è un array bidimensionale, come mostrato in Figura 2.7.

```

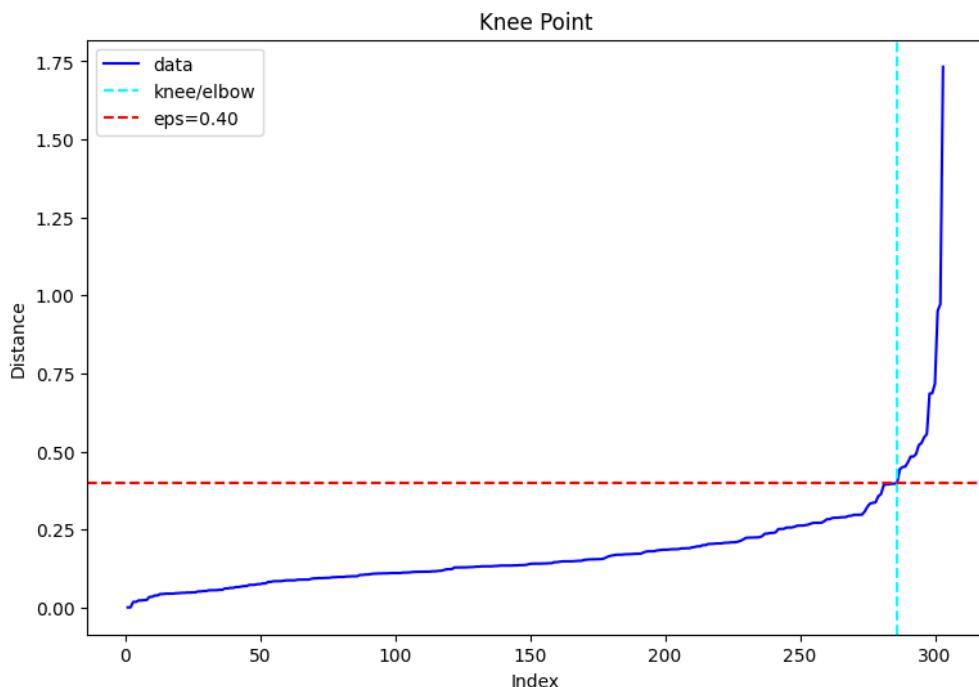
pca = PCA(n_components=2)
pca_X = pca.fit_transform(data) # effettuo la PCA a 2 comp sul dataset
data_reduced = pd.DataFrame(pca_X, columns=['PCA1', 'PCA2'])
print(data_reduced)
✓ 0.0s
      PCA1    PCA2
0  0.051739  2.624022
1  0.817441 -0.730375
2  2.057599 -0.039098
3  1.983843 -0.596701
4  0.768371  0.412545
...
298 -1.485287 -0.423483
299  0.149325 -1.215922
300 -2.681772  0.593019
301 -2.170858 -2.166992
302  0.365760  0.787916
[303 rows x 2 columns]

```

**Figura 2.7:** Dataset PCA

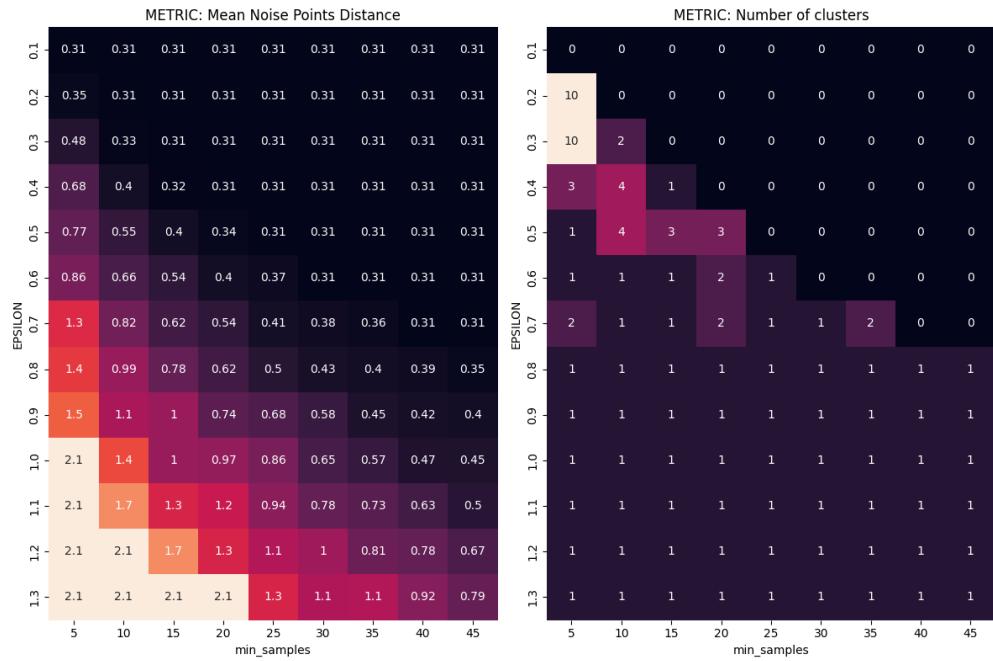
### 2.4.1 DBSCAN

Una volta aver eseguito la riduzione delle dimensioni tramite la tecnica del PCA, si è passati alla visualizzazione del clustering, costituita dall'utilizzo di un nuovo algoritmo di Machine Learning, il DBSCAN. Questo modello permette di fare clustering con una tecnica basata sulla densità che risulta differente rispetto al K-Means, il quale parte da un numero predefinito di cluster, e quindi di centroidi, ed evolve assegnano i punti al centroide più vicino. Non avendo la necessità di conoscere in anticipo il numero di cluster, il DBSCAN ha però bisogno di due parametri di input: **epsilon**, cioè la "distanza-soglia" sotto la quale due punti sono considerati vicini, ovvero appartenenti allo stesso cluster; **Min \_ Points**, che rappresenta il numero minimo di punti per formare una regione densa. Per stimare l'epsilon è stato utilizzato il NearestNeighbors: l'epsilon ideale è quello che corrisponde al punto della curva in cui la pendenza aumenta drasticamente, in maniera simile ad un andamento esponenziale. Dal grafico di Figura 2.8 si può osservare come il valore ricavato è pari a **epsilon = 0.4**



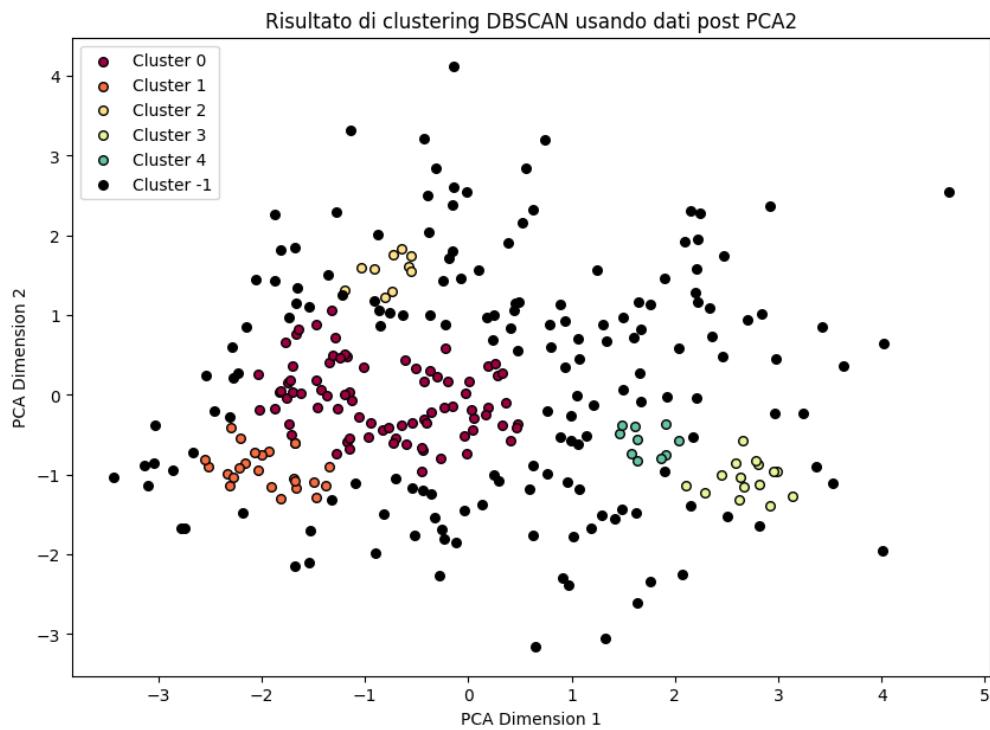
**Figura 2.8:** Knee Point DBSCAN

Per determinare i valori ottimali dei parametri di DBSCAN, in particolare **Epsilon** e **Min \_ Points**, è stata realizzata una heatmap per visualizzare le combinazioni più adatte. La funzione utilizzata genera una griglia di valori per **Epsilon** (nel range  $[eps-1, eps+1]$  con intervalli di 0.1) e per **Min \_ Points** (da 5 a 50 con incrementi di 5). Per ciascuna combinazione, vengono calcolate due metriche: la distanza media dei punti rumorosi (indicata nella heatmap di sinistra) e il numero di cluster rilevati (indicata nella heatmap di destra). Queste visualizzazioni permettono di identificare i parametri che ottimizzano il numero di cluster e minimizzano i punti rumorosi, come illustrato in Figura 2.9.



**Figura 2.9:** Heatmap Parametri DBSCAN

Andando a scegliere il valore di **Epsilon** ottenuto in Figura 2.8 ed il valore di **Min \_ Points** ottenuto per quell'**Epsilon**, come mostrato in Figura 2.9, che in questo caso si ottiene **Min \_ Points = 10**, il clustering ottenuto per mezzo dell'algoritmo DBSCAN è mostrato nella Figura 2.10

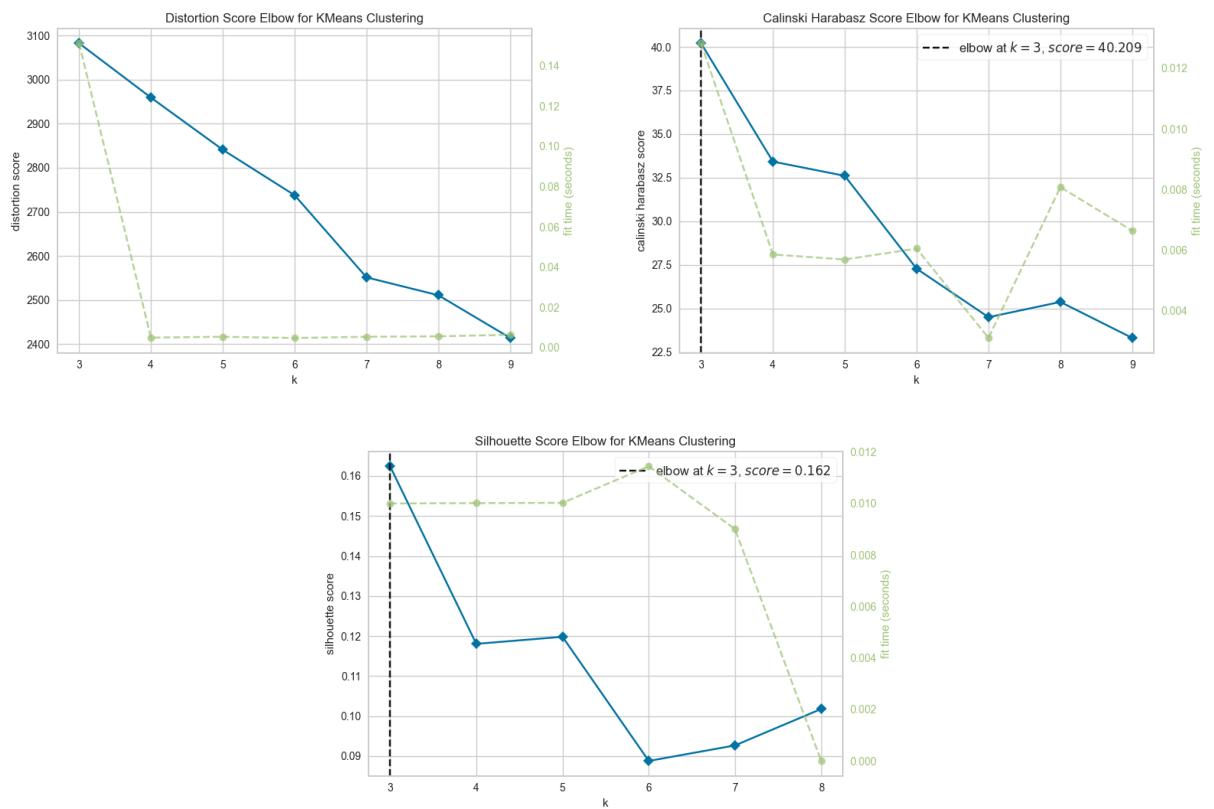


**Figura 2.10:** Risultato Clustering DBSCAN

Possiamo osservare quattro cluster principali (Cluster 0, Cluster 1, Cluster 2 e Cluster 3), indicati con colori distinti, oltre a un gruppo di punti etichettati come "rumore" (Cluster -1) rappresentati in nero. Il clustering appare sbilanciato: il Cluster 0 (in rosso scuro) è quello che contiene più punti mentre i cluster rimanenti (arancione, verde chiaro e azzurro) ne contengono pochissimi. I punti rimanenti, che non appartengono a nessun cluster, sono considerati rumore (punti neri). L'algoritmo sembra avere difficoltà con questo dataset, probabilmente a causa della distribuzione dei punti. DBSCAN ha formato un cluster principale che racchiude i punti più vicini e densi, mentre i cluster rimanenti si basano su gruppi molto più piccoli e isolati. Gli altri punti, troppo distanti, vengono etichettati come rumore.

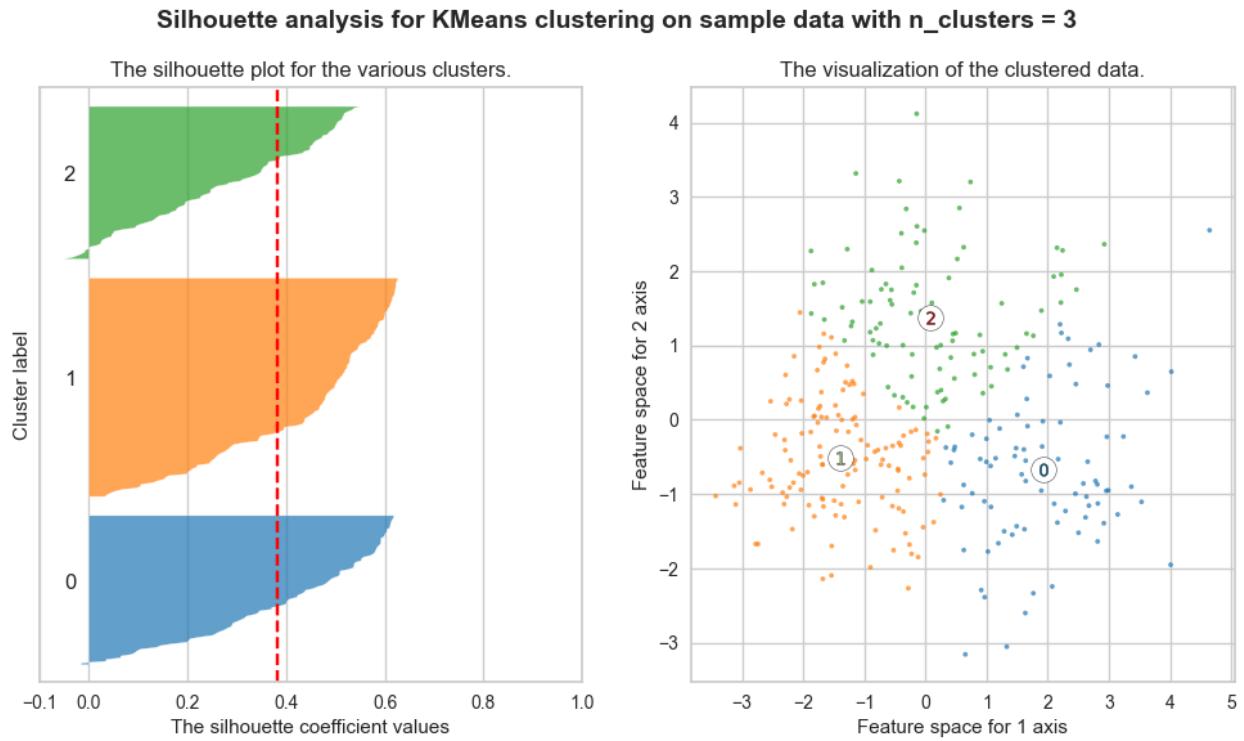
## 2.4.2 K-Means

A seguito dei risultati insoddisfacenti ottenuti dal DBSCAN, si è passati alla visualizzazione del clustering del dataset elaborato tramite la PCA costruita per mezzo dell'algoritmo K-Means. Nella Figura 2.11 è riportato l'Elbow Method, usando sempre le tre metriche Distorsione, Calinski e Silhouette, il quale suggerisce di considerare 3 cluster.



**Figura 2.11:** Metodi Elbow K-Means con PCA

Il risultato del clustering e le silhouette ottenute applicando K-Means sul dataset Post PCA a due dimensioni sono mostrate in Figura 2.12.



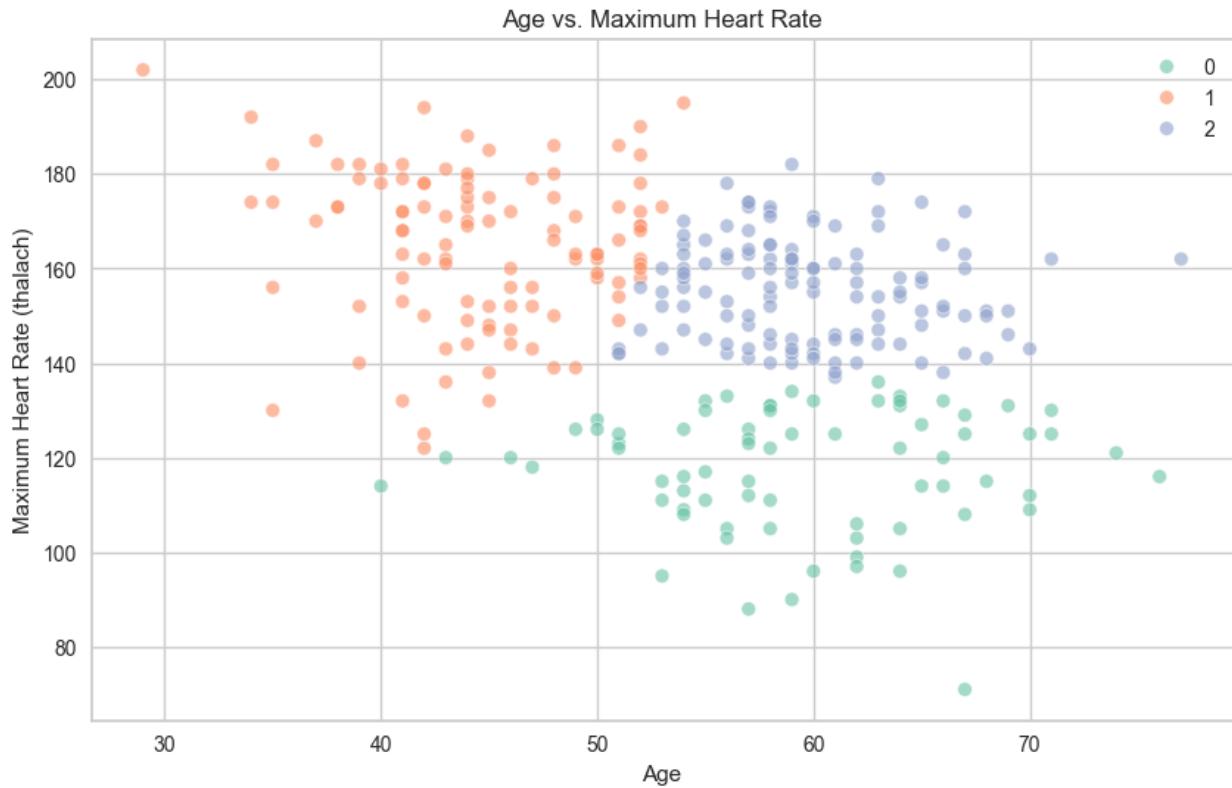
**Figura 2.12:** Silhouette (sinistra) e Clustering K-Means (destra)

Si noti che i cluster individuati dall'algoritmo appaiono densi e ben distinti tra loro, a prova del fatto che il K-Means sia riuscito a distribuire ciascun elemento all'interno dei cluster, fatta eccezione per alcuni outlier. Per confermare la bontà del modello, in Figura 2.12 è rappresentata la Silhouette. Come nel caso precedente del K-Means su due specifiche features del dataset, si ha uno score di circa 0.4, che risulta comunque accettabile. Inoltre, si può notare la presenza di elementi del cluster 2, seppur minimi, con uno score negativo, i quali potevano essere rappresentati meglio da un altro cluster.

## 2.5 Interpretazione dei risultati dei cluster- K-means

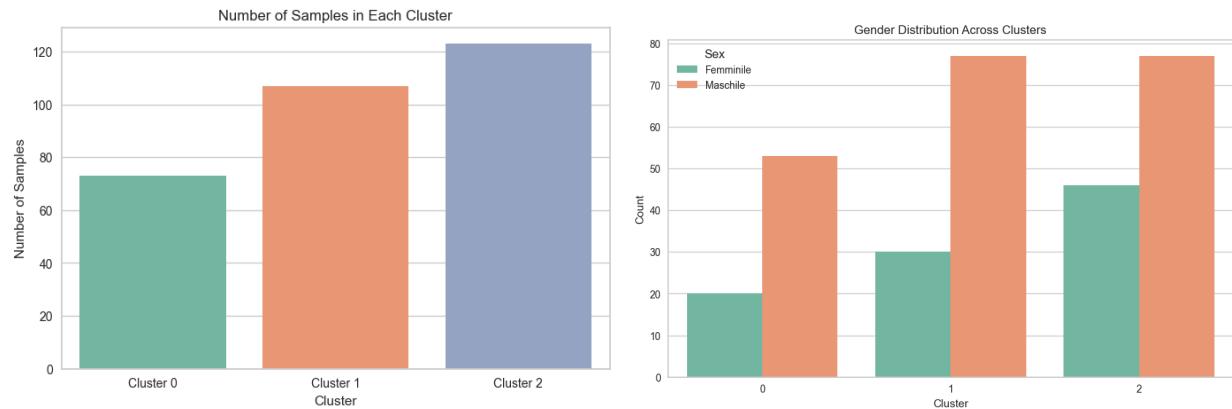
A questo punto, il lavoro si è concentrato sull'analisi delle caratteristiche dei singoli cluster, con l'obiettivo di profilare i pazienti in base a specifiche variabili. Questa procedura è stata svolta attraverso un'analisi esplorativa dei dati, utilizzando le feature "age" (età) e "thalach" (frequenza cardiaca massima) per esaminare la distribuzione dei cluster.

In Figura 2.13 è riportata la distribuzione bidimensionale dei cluster rispetto alle variabili "age" e "thalach". Questo grafico fornisce una panoramica delle relazioni tra età e frequenza cardiaca massima per i diversi cluster, mostrando come i gruppi di pazienti si distinguono per queste due variabili chiave.



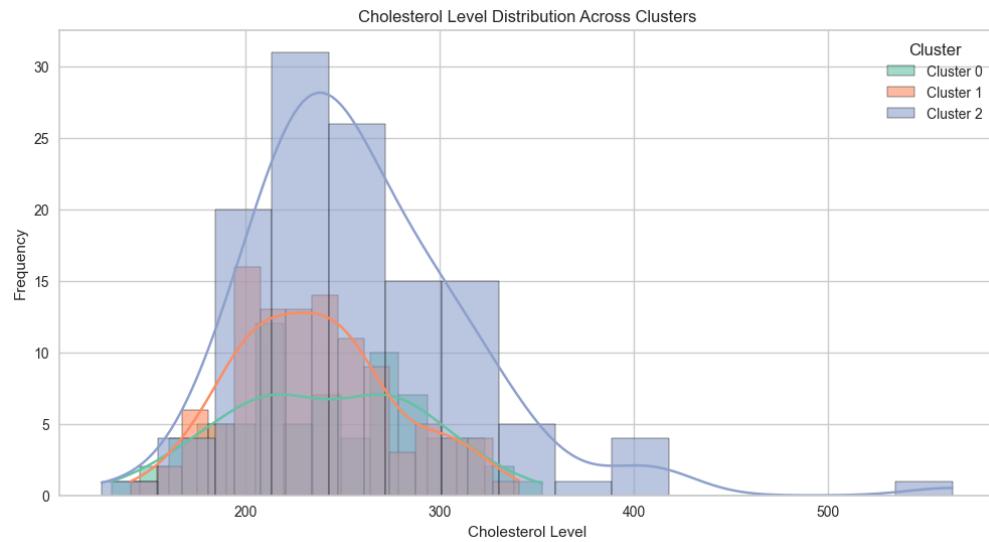
**Figura 2.13:** Confronto tra Età e Massima Frequenza Cardiaca

Successivamente, la Figura 2.14 illustra la distribuzione dei cluster sia in termini di numerosità complessiva (a sinistra) sia rispetto alla distribuzione di genere (a destra). Questo consente di comprendere non solo il bilanciamento quantitativo dei cluster, ma anche eventuali differenze tra uomini e donne all'interno di ciascun cluster.

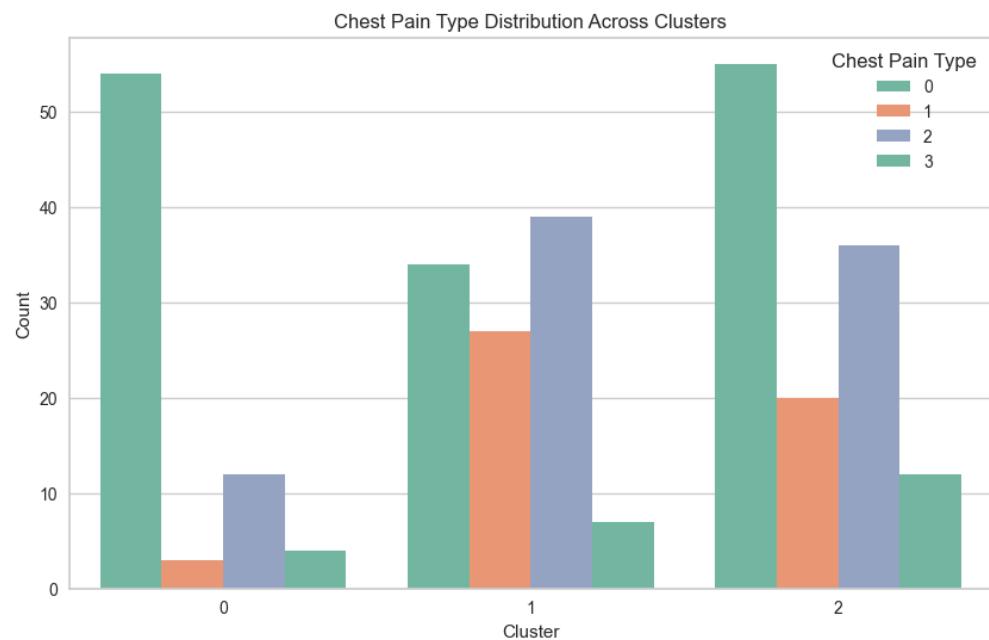


**Figura 2.14:** Distribuzione cluster per numerosità (sinistra) e per genere (destra)

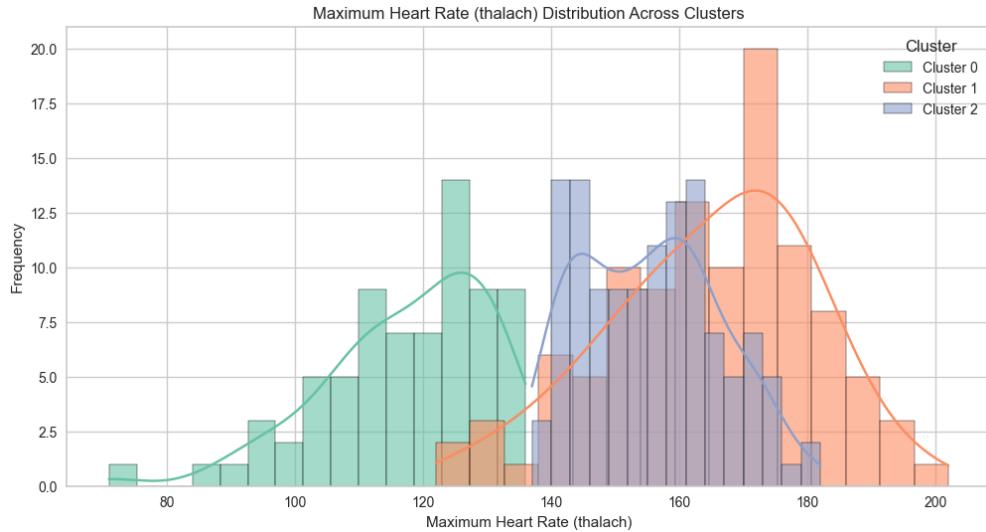
Le Figure 2.15, 2.16, 2.17, 2.18 mostrano rispettivamente la distribuzione dei valori di colesterolo (chol), dei tipi di dolore toracico (cp), della frequenza cardiaca massima (thalach) e della pressione arteriosa (trtbps) all'interno dei singoli cluster, attraverso istogrammi. Questi grafici permettono di identificare le tendenze delle variabili mediche rilevanti nei diversi cluster e di confrontarle tra loro.



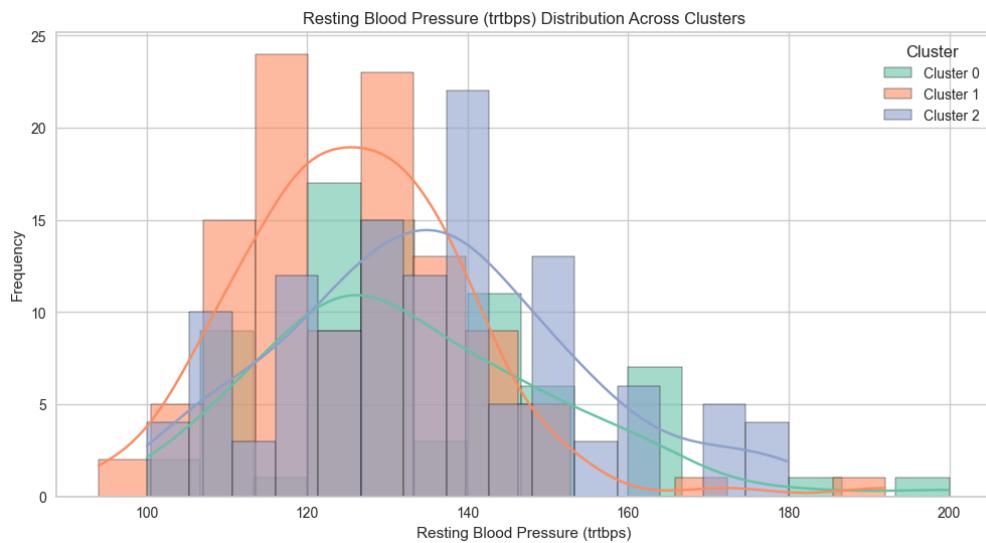
**Figura 2.15:** Distribuzione dei valori di colesterolo (chol) per cluster



**Figura 2.16:** Distribuzione dei tipi di dolore toracico (cp) per cluster



**Figura 2.17:** Distribuzione della frequenza cardiaca massima (thalach) per cluster



**Figura 2.18:** Distribuzione della pressione arteriosa (trtbps) per cluster

## 2.6 Profilazione dei Cluster

L'esplorazione dei dati ha permesso di comprendere come l'algoritmo K-Means abbia suddiviso i pazienti in diversi cluster, evidenziando differenze rilevanti tra i gruppi. In base alle caratteristiche cliniche e demografiche emerse, si è delineato un profilo per ciascun cluster, offrendo una visione più chiara delle condizioni dei pazienti e dei potenziali rischi.

### Cluster 0: Pazienti Sani

- l'età compresa dai 40 ai 80 anni
- frequenza cardiaca, colesterolo e pressione sanguigna sotto la media

- prevalenza di chest pain di tipo 0

#### **Cluster 1: Pazienti adulti a rischio**

- età compresa dai 30 ai 50 anni
- frequenza cardiaca e pressione sanguigna molto superiore rispetto alla media, mentre il colesterolo è poco sopra alla media
- alto numero di chest pain di tipo 1 e 2

#### **Cluster 1: Pazienti anziani a rischio**

- età superiore ai 50 anni
- frequenza cardiaca e pressione sanguigna sopra la media ma con colesterolo molto sopra alla media
- alto numero di chest pain di tipo 2 e 3

# 3 Classificazione

La classificazione è una tecnica di apprendimento supervisionato utilizzata per categorizzare i dati in classi predefinite, sfruttando le caratteristiche degli oggetti noti per identificare la classe di nuovi oggetti. L'obiettivo è costruire un modello capace di prevedere la classe di un oggetto in base alle sue peculiarità. In questo capitolo, verrà implementata questa tecnica per assegnare correttamente nuovi elementi alle categorie appropriate e confrontare le prestazioni di diversi modelli di classificazione in termini di accuratezza. I modelli sono addestrati su un sottoinsieme del dataset originale e testati sul restante sottoinsieme per valutare quale si adatti meglio ai dati.

## 3.1 Training e Risultati della classificazione

Per quanto riguarda le operazione di ETL, sono state effettuate le stesse operazioni viste nel capitolo del Clustering nella sezione 2.2.

La procedura di classificazione si articola in due fasi:

- Al modello vengono forniti dati di training etichettati, permettendogli di apprendere le caratteristiche delle diverse classi.
- Successivamente, al modello vengono forniti dati di test sconosciuti. Basandosi sull'apprendimento precedente, il modello classifica questi nuovi dati.

Per definire il modello, sono stati utilizzati algoritmi disponibili nella libreria Scikit-Learn di Python. Abbiamo deciso di testare otto diversi modelli di classificazione per confrontarne le prestazioni e determinare quale si adattasse meglio ai nostri dati. I modelli scelti sono:

- DecisionTreeClassifier
- RandomForestClassifier
- SVC (Support Vector Classifier)
- LogisticRegression
- XGBoost Classifier
- AdaBoostClassifier
- GradientBoostingClassifier
- LinearDiscriminantAnalysis

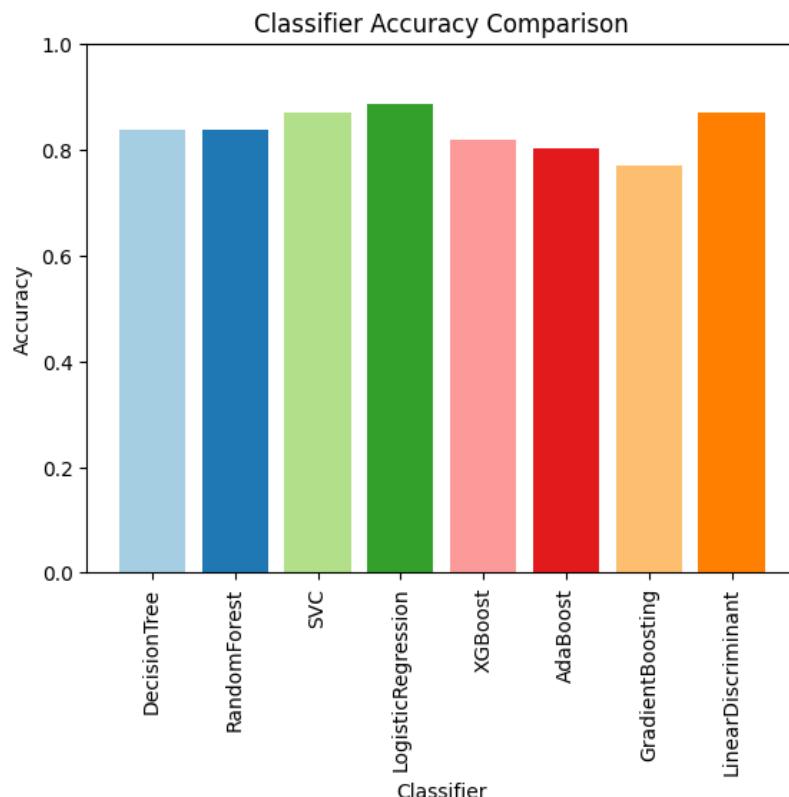
Per garantire la ripetibilità dell'esperimento, viene inizialmente impostato un random state fisso pari a 42, così da ottenere risultati coerenti a ogni esecuzione. Successivamente, il dataset viene diviso in Training Set e Test Set utilizzando la funzione `train_test_split()`, impostando `test_size=0.2` per assegnare l'80% dei dati al Training set e il 20% al Test set. Tra i modelli precedentemente elencati è possibile misurarne la qualità attraverso vari indici; in questo progetto è stata presa in considerazione l'accuratezza, ovvero il rapporto tra il numero di dati classificati correttamente e il numero di dati classificati. I risultati di accuratezza ottenuti dai diversi modelli sono mostrati in Figura 3.1:

```
<----- Test Accuracy ----->

Decision Tree Classifier: 0.8360655737704918
Random Forest Classifier: 0.8360655737704918
Support Vector Machine Classifier: 0.8688524590163934
Logistic Regression: 0.8852459016393442
XGBoost Classifier: 0.819672131147541
AdaBoost Classifier: 0.8032786885245902
Gradient Boosting Classifier: 0.7704918032786885
LinearDiscriminant: 0.8688524590163934
```

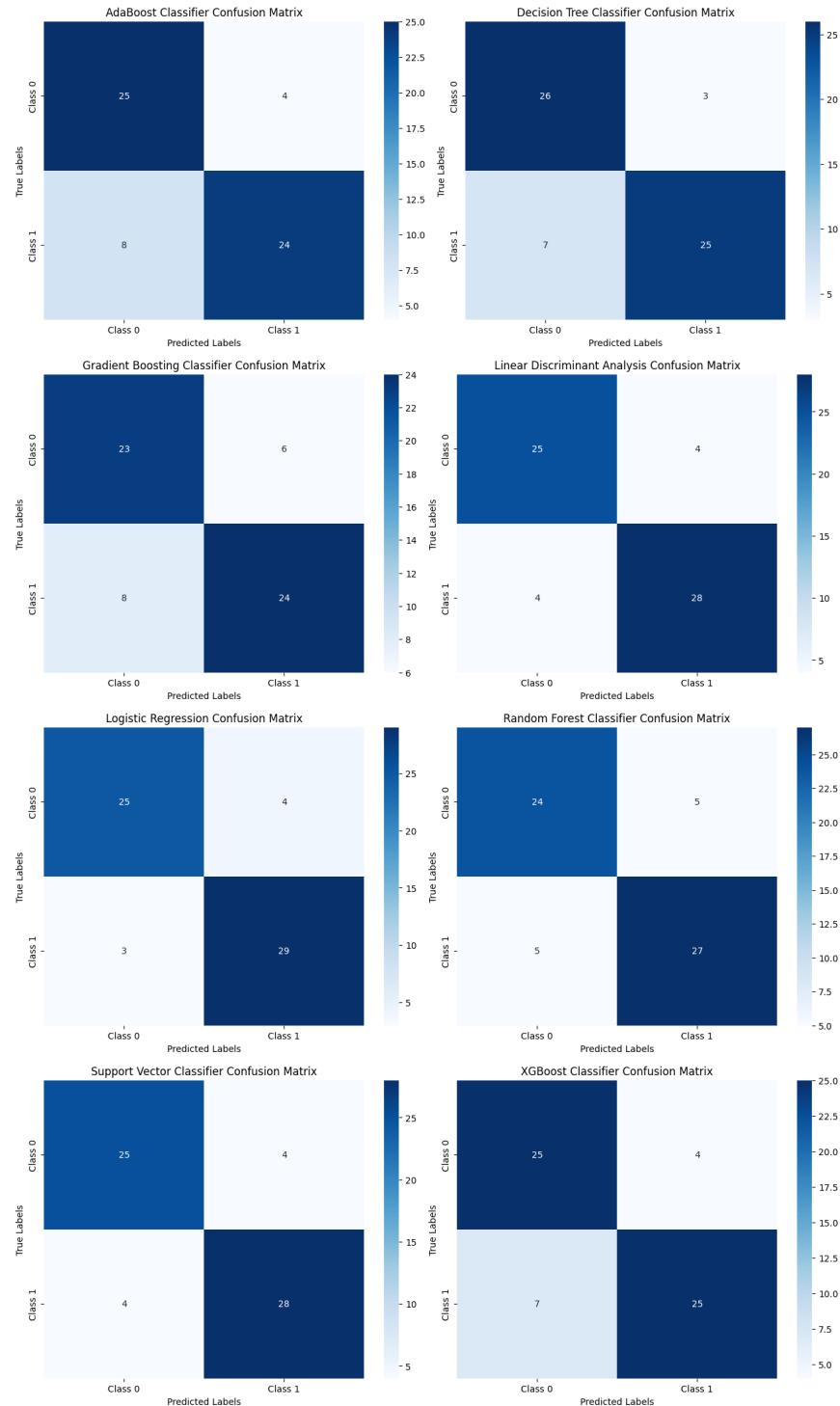
**Figura 3.1:** Accuratezza Classificatori

Per visualizzare meglio i risultati dell'accuratezza ottenuti per i diversi classificatori, viene riportata la rappresentazione grafica in Figura 3.2



**Figura 3.2:** Accuratezza Classificatori

Inoltre, in Figura 3.3 sono state visualizzate le matrici di confusione di ciascun classificatore, ovvero delle tabelle in cui sono riportate le previsioni in corrispondenza delle colonne e lo stato effettivo in corrispondenza delle righe; tramite questo strumento è possibile valutare le prestazioni dei modelli considerati. Tutte le previsioni corrette si trovano sulla diagonale della tabella, dove vi sono True Positive e True Negative, mentre le previsioni errate si trovano esternamente alla diagonale e sono i False Positive e i False Negative.



**Figura 3.3:** Matrici di Confusione Classificatori

Dalle matrici di confusione relative ai risultati di tale classificazione si evince come tutti gli algoritmi abbiano adoperato una buona classificazione, con alti valori di True Positive e True Negative. In supporto a tale strumento, è riportato il classification report, il quale mostra, per ogni classificatore, ulteriori metriche di valutazione delle performance.

#### Decision Tree Classifier:

	precision	recall	f1-score	support
0	0.79	0.90	0.84	29
1	0.89	0.78	0.83	32
accuracy			0.84	61
macro avg	0.84	0.84	0.84	61
weighted avg	0.84	0.84	0.84	61

#### Random Forest Classifier:

	precision	recall	f1-score	support
0	0.83	0.83	0.83	29
1	0.84	0.84	0.84	32
accuracy			0.84	61
macro avg	0.84	0.84	0.84	61
weighted avg	0.84	0.84	0.84	61

#### Support Vector Machine Classifier:

	precision	recall	f1-score	support
0	0.86	0.86	0.86	29
1	0.88	0.88	0.88	32
accuracy			0.87	61
macro avg	0.87	0.87	0.87	61
weighted avg	0.87	0.87	0.87	61

#### Logistic Regression:

	precision	recall	f1-score	support
0	0.89	0.86	0.88	29
1	0.88	0.91	0.89	32
accuracy			0.89	61
macro avg	0.89	0.88	0.88	61
weighted avg	0.89	0.89	0.89	61

## XGBoost Classifier:

	precision	recall	f1-score	support
0	0.78	0.86	0.82	29
1	0.86	0.78	0.82	32
accuracy			0.82	61
macro avg	0.82	0.82	0.82	61
weighted avg	0.82	0.82	0.82	61

## AdaBoost Classifier:

	precision	recall	f1-score	support
0	0.76	0.86	0.81	29
1	0.86	0.75	0.80	32
accuracy			0.80	61
macro avg	0.81	0.81	0.80	61
weighted avg	0.81	0.80	0.80	61

## Gradient Boosting Classifier:

	precision	recall	f1-score	support
0	0.74	0.79	0.77	29
1	0.80	0.75	0.77	32
accuracy			0.77	61
macro avg	0.77	0.77	0.77	61
weighted avg	0.77	0.77	0.77	61

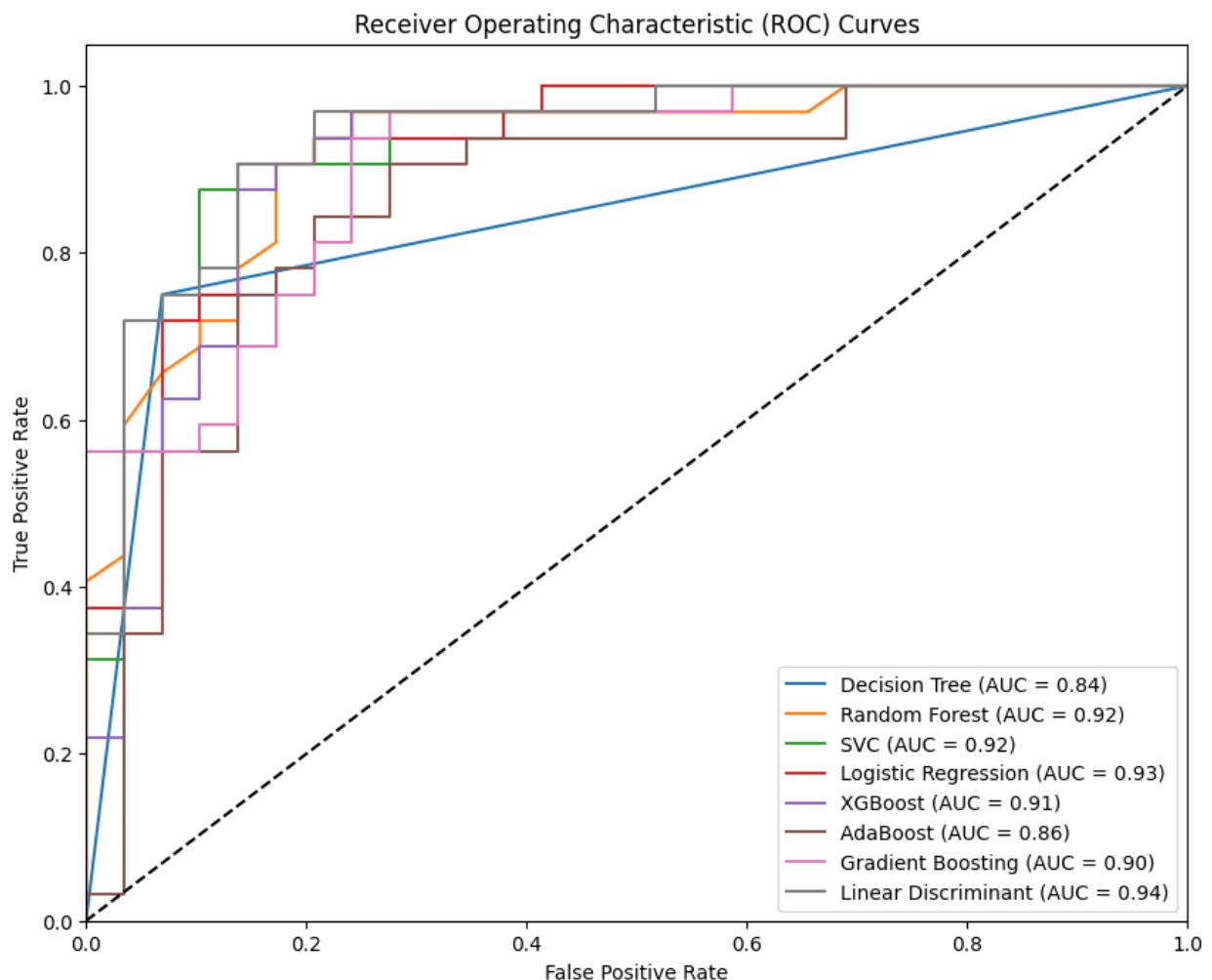
## Linear Discriminant Analysis:

	precision	recall	f1-score	support
0	0.86	0.86	0.86	29
1	0.88	0.88	0.88	32
accuracy			0.87	61
macro avg	0.87	0.87	0.87	61
weighted avg	0.87	0.87	0.87	61

Per completare l'analisi dei risultati si è ricorso all'utilizzo della **Curva ROC** (Receiver Operating Characteristics), una tecnica statistica che misura l'accuratezza di un test diagnostico lungo tutto il range dei valori possibili e utilizzabile solo in caso di classi binarie. Questo grafico mostra il trade-off tra Sensitività e Specificità mano a mano che si varia la soglia o la regola scelta per classificare un record. L'area sottostante alla curva ROC, chiamata AUC (Area Under the Curve), è una misura di accuratezza e in base a tale valore il test si definisce:

- non informativo, se  $AUC = 0.5$
- poco accurato, se  $0.5 < AUC \leq 0.7$
- moderatamente accurato, se  $0.7 < AUC < 0.9$
- altamente accurato, se  $0.9 \leq AUC < 1$
- perfetto, se  $AUC = 1$

La Figura 3.4 mostra le **Curve ROC** relative ai vari modelli. Non vengono delle linee spezzate molto probabilmente a causa di come è formato il dataset.

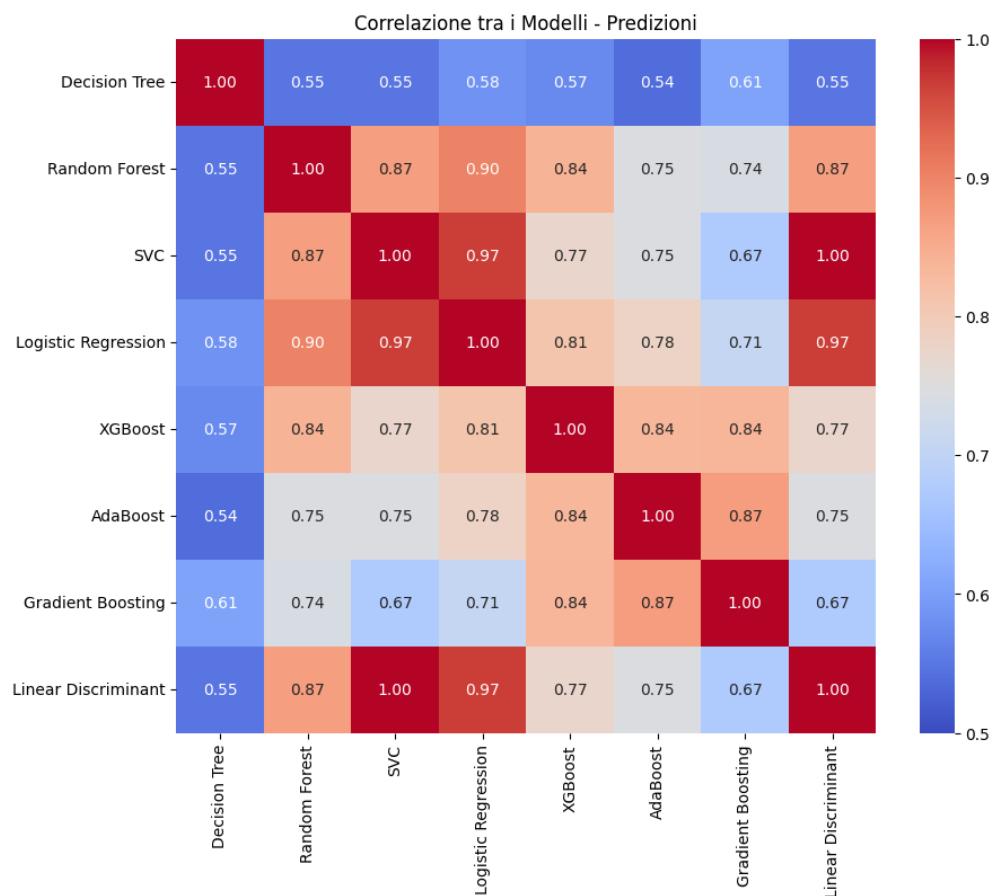


**Figura 3.4:** Curve ROC Classificatori

## 3.2 Grid Search

Un fattore importante nelle prestazioni di questi modelli sono i loro iperparametri, variabili che l'utente specifica in genere durante la creazione del modello stesso. La scelta dei valori appropriati per questi iperparametri può migliorare in modo significativo le prestazioni di un modello. Per ottimizzare gli iperparametri, si può utilizzare la Grid Search, una tecnica di ottimizzazione che esplora in modo esaustivo tutte le combinazioni di valori specificati per individuare quelli ottimali.

Per selezionare i classificatori sui quali applicare la Grid Search, è stata utilizzata una heatmap, mostrata in Figura 3.5 per valutare le correlazioni tra i diversi modelli.



**Figura 3.5:** Correlazioni tra modelli

I modelli con correlazione più alta tendono a fornire predizioni simili, quindi è stato scelto di applicare la Grid Search a un insieme di classificatori meno correlati tra loro per massimizzare la varietà delle predizioni. Sono stati scelti i seguenti modelli:

- Decision Tree Classifier
- Random Forest Classifier
- XGBoost Classifier
- AdaBoost Classifier

In Figura 3.6 sono riportati gli iperparametri utilizzati per i modelli di classificatori scelti:

```

def decision_tree():
    decision_tree = DecisionTreeClassifier()
    param_grid = {
        'max_depth': [3, 5, 7, 10, 15, 20],
        'min_samples_split': [2, 5, 10, 20],
        'min_samples_leaf': [1, 2, 4, 6, 8],
    }

def random_forest():
    random_forest = RandomForestClassifier(n_estimators=100, random_state=42)
    param_grid = {
        'n_estimators': [50, 100, 200],
        'max_features': ['auto', 'sqrt', 'log2'],
        'max_depth': [None, 10, 20, 30],
        'min_samples_split': [2, 5, 10],
        'min_samples_leaf': [1, 2, 4]
    }

def xgboost():
    xgboost = XGBClassifier(random_state=42, eval_metric='logloss')
    param_grid = {
        'n_estimators': [50, 100, 150],
        'learning_rate': [0.01, 0.1, 0.2],
        'max_depth': [3, 5, 7],
        'subsample': [0.6, 0.8, 1.0]
    }

def adaboost():
    adaboost = AdaBoostClassifier(random_state=42)
    param_grid = {
        'n_estimators': [50, 100, 200],
        'learning_rate': [0.01, 0.1, 1.0]
    }

```

**Figura 3.6:** Parametri GridSearch dei modelli scelti

Una volta eseguito l'algoritmo, i risultati ottenuti sono riportati in Figura 3.7.

```

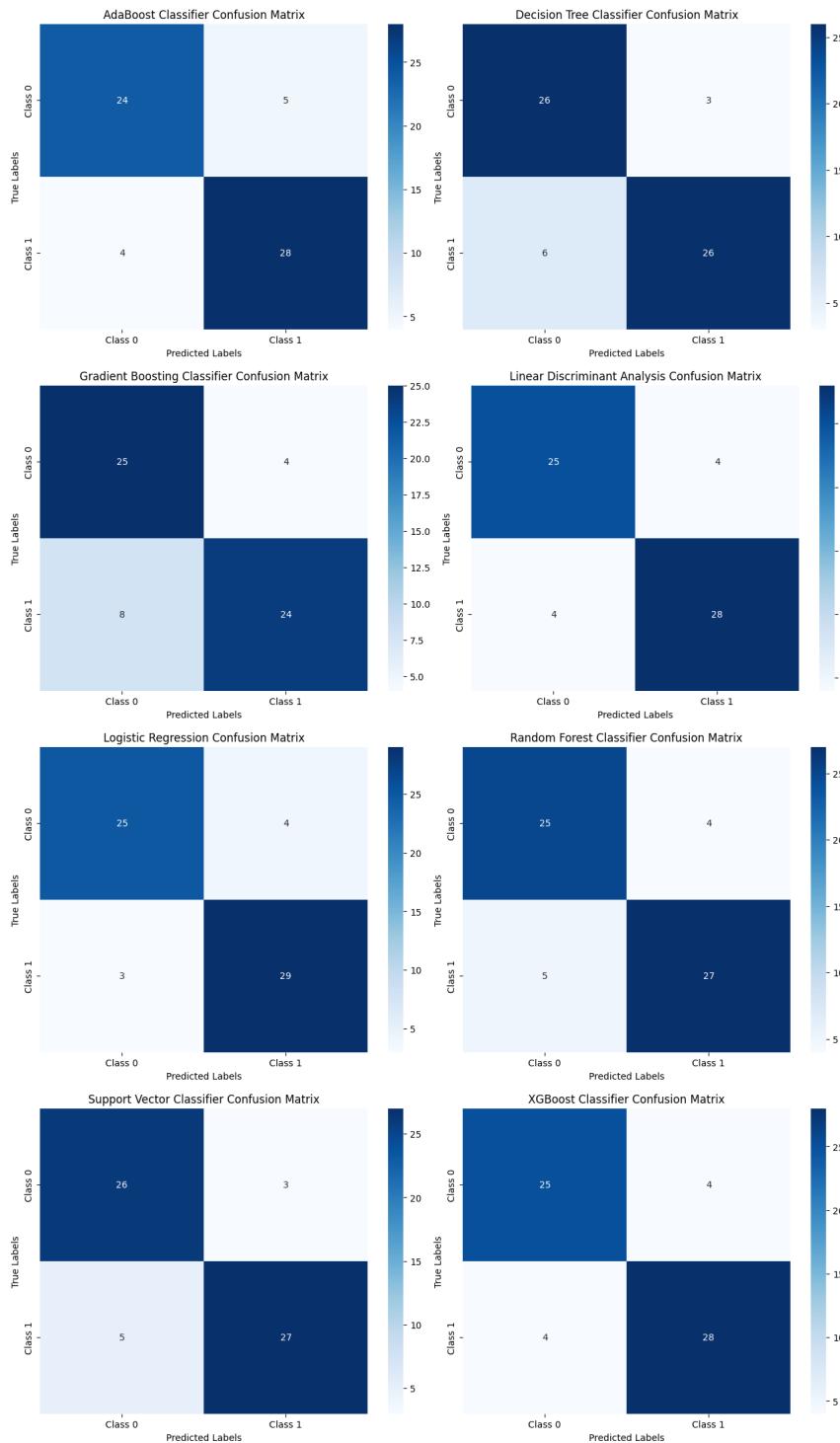
<----- Test Accuracy ----->

Decision Tree Classifier: 0.8524590163934426
Random Forest Classifier: 0.8524590163934426
Support Vector Machine Classifier: 0.8688524590163934
Logistic Regression: 0.8852459016393442
XGBoost Classifier: 0.8688524590163934
AdaBoost Classifier: 0.8524590163934426
Gradient Boosting Classifier: 0.8032786885245902
LinearDiscriminant: 0.8688524590163934

```

**Figura 3.7:** Accuratezza Classificatori con GridSearch

Si può notare che si ha un miglioramento notevole in tutti e quattro i classificatori. Inoltre, in Figura 3.8 sono state visualizzate le matrici di confusione di ciascun classificatore post GridSearch.



**Figura 3.8:** Matrici di Confusione Classificatori con GridSearch

Dalle matrici di confusione relative ai risultati di tale classificazione si evince come tutti gli algoritmi abbiano adoperato una buona classificazione, con alti valori di True Positive

e TrueNegative. In supporto a tale strumento, è riportato il classification report, il quale mostra, per ogni classificatore, ulteriori metriche di valutazione delle performance.

#### Decision Tree Classifier:

	precision	recall	f1-score	support
0	0.81	0.90	0.85	29
1	0.90	0.81	0.85	32
accuracy			0.85	61
macro avg	0.85	0.85	0.85	61
weighted avg	0.86	0.85	0.85	61

#### Random Forest Classifier:

	precision	recall	f1-score	support
0	0.83	0.86	0.85	29
1	0.87	0.84	0.86	32
accuracy			0.85	61
macro avg	0.85	0.85	0.85	61
weighted avg	0.85	0.85	0.85	61

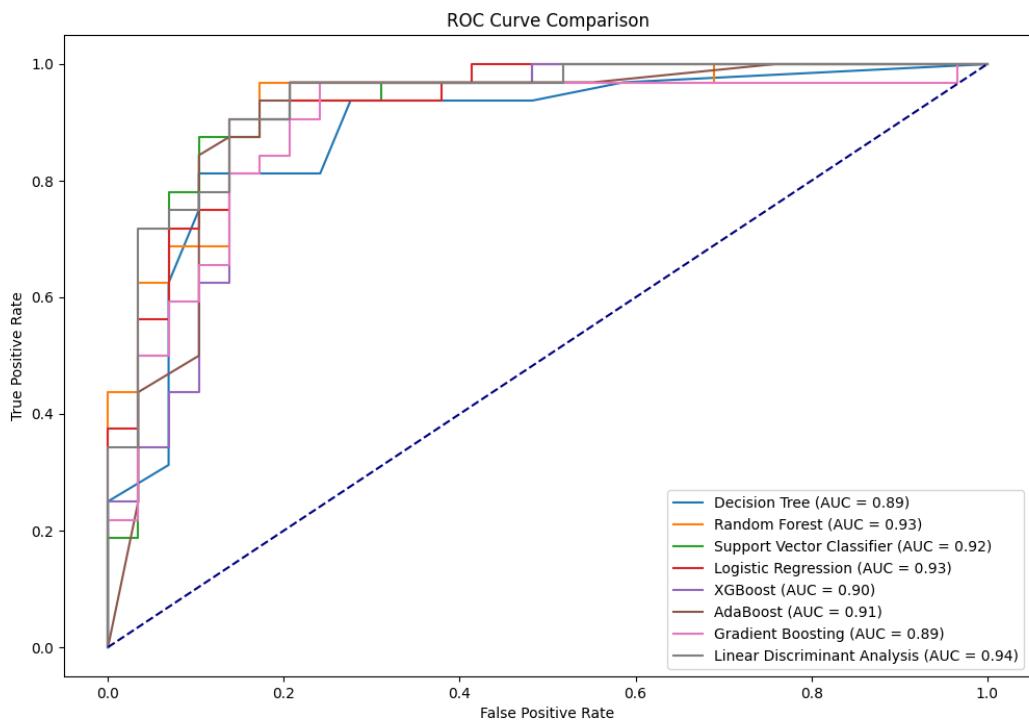
#### XGBoost Classifier:

	precision	recall	f1-score	support
0	0.86	0.86	0.86	29
1	0.88	0.88	0.88	32
accuracy			0.87	61
macro avg	0.87	0.87	0.87	61
weighted avg	0.87	0.87	0.87	61

#### AdaBoost Classifier:

	precision	recall	f1-score	support
0	0.86	0.83	0.84	29
1	0.85	0.88	0.86	32
accuracy			0.85	61
macro avg	0.85	0.85	0.85	61
weighted avg	0.85	0.85	0.85	61

Per completare l'analisi dei risultati, sono state calcolate le **Curve ROC** usando i classificatori con **GridSearch**, le quali sono mostrate in Figura 3.9.



**Figura 3.9:** Curve ROC Classificatori con GridSearc

# 4 Forecasting

In questo capitolo, proponiamo uno studio sull'andamento della temperatura nella città di Los Angeles nel periodo compreso tra il 2012 e il 2017. Il dataset utilizzato per questa analisi è stato ottenuto dal sito Kaggle ([link](#)) e contiene informazioni meteo storiche su diverse città del Nord America (vedi Sezione 4.1). Dopo una fase iniziale di ETL (Extract, Transform, Load), procederemo con lo sviluppo di modelli di previsione della temperatura per stimare l'andamento dei successivi 2-3 anni.

L'obiettivo di questo studio è mostrare come modelli di regressione possano essere applicati efficacemente per prevedere l'evoluzione di fenomeni climatici come la temperatura, utilizzando dati storici e approcci moderni di forecasting.

## 4.1 Dataset Weather North America

Di seguito sono riportati gli attributi presenti nei vari CSV che compongono il dataset Weather North America. Per il nostro studio andremo a concentrarci solo sul file *temperature.csv*.

Attributo	Descrizione
City	Città presa in analisi
Country	Paese di appartenza delle città analizzate
Latitude	Latitudine della Città
Longitude	Longitudine della Città

**Tabella 4.1:** city\_attributes.csv

Attributo	Descrizione
datetime	Data ed ora della misurazione
umidity	Umidità [%] misurata nella Città in quel DateTime

**Tabella 4.2:** humidity.csv

Attributo	Descrizione
datetime	Data ed ora della misurazione
pressure	Pressione Atmosferica [hPa] misurata nella Città in quel DateTime

**Tabella 4.3:** pressure.csv

Attributo	Descrizione
datetime	Data ed ora della misurazione
temperature	Temperatura [ $^{\circ}\text{C}$ ] misurata nella Città in quel DateTime

**Tabella 4.4:** temperature.csv

Attributo	Descrizione
datetime	Data ed ora della misurazione
weather_description	Descrizione Meteo nella Città in quel DateTime

**Tabella 4.5:** weather\_description.csv

Attributo	Descrizione
datetime	Data ed ora della misurazione
wind_direction	Direzione del vento [Gradi Meteorologici] misurata nella Città in quel DateTime

**Tabella 4.6:** wind\_direction.csv

Attributo	Descrizione
datetime	Data ed ora della misurazione
wind_speed	Velocità del vento [m/s] misurata nella Città in quel DateTime

**Tabella 4.7:** wind\_speed.csv

## 4.2 Librerie utilizzate

### statsmodels

La libreria **statsmodels** è uno strumento per l’analisi statistica in Python, progettato per consentire la stima di modelli statistici, l’esplorazione dei dati e la realizzazione di test statistici. **statsmodels** si distingue dalle altre librerie per la sua capacità di gestire modelli econometrici e statistici classici, fornendo strumenti dettagliati per l’analisi di regressioni, serie storiche, e modelli di probabilità, il tutto corredata da una ricca gamma di funzioni per la visualizzazione e la valutazione dei risultati.



**Figura 4.1:** Logo StatsModels

Nella relazione che segue, utilizzeremo le caratteristiche principali della libreria **statsmodels**, con particolare attenzione ai suoi componenti essenziali come i modelli di regressione lineare e non lineare, le serie storiche e i modelli ARIMA, le tecniche di stima e i test statistici integrati.

### pmdarima

La libreria **pmdarima** è uno strumento noto per la sua implementazione efficiente del modello ARIMA (AutoRegressive Integrated Moving Average). Estendendo le capacità della libreria **statsmodels**, **pmdarima** si concentra su un aspetto fondamentale dell’analisi delle serie temporali: l’automazione della selezione dei parametri del modello ARIMA attraverso la funzione `auto_arima`. Questo la rende estremamente utile per chiunque voglia applicare modelli ARIMA senza dover affrontare il processo manuale di identificazione degli ordini del modello ( $p$ ,  $d$ ,  $q$ ).

### Scikit-Learn

La libreria è descritta nella sezione 2.1.

## 4.3 ETL

Durante la fase di ETL, ci siamo concentrati sul raggruppamento dei dati, poiché i campioni originali sono registrati a cadenza oraria. Tuttavia, per il nostro scopo, non è necessario mantenere tale granularità temporale, e i campioni possono essere aggregati su base bisettimanale, trisettimanale o quadrisettimanale. A tal fine, abbiamo creato tre diversi file CSV di lavoro, ciascuno con un diverso intervallo di aggregazione, per confrontare quale tra essi fornisce le migliori prestazioni nel processo di forecasting.

Per l’aggregazione, è stata calcolata la media delle temperature all’interno di ciascun intervallo temporale scelto.

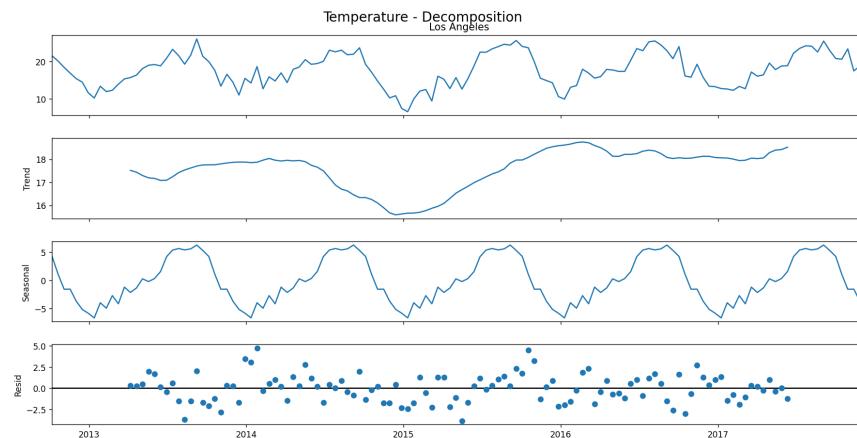
### 4.3.1 Analisi della serie temporale

Per valutare la stazionarietà delle serie temporali ottenute dai tre file CSV (con aggregazioni bisettimanali, trisettimanali e mensili), è stato applicato l'Augmented Dickey-Fuller (ADF) test. I risultati del test hanno mostrato, per tutte e tre le serie, un p-value inferiore a 0.05, indicando che le serie temporali sono stazionarie e quindi adatte per procedere con la modellazione. Successivamente, è stata eseguita la decomposizione della serie temporale

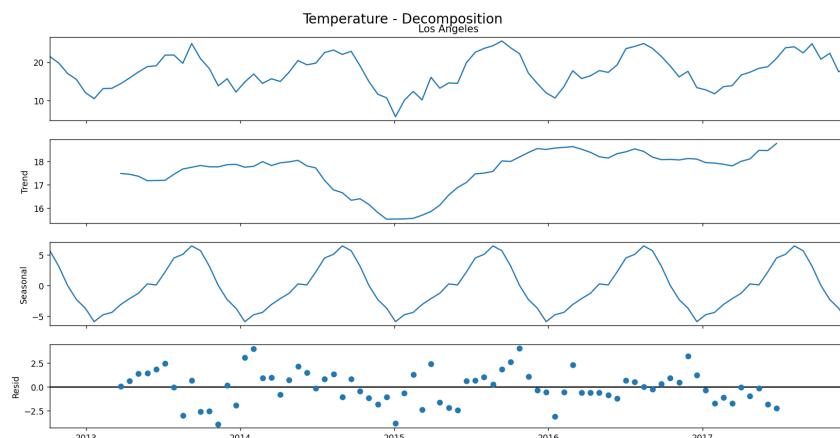
Aggregazione	ADF Statistic	p-value	Valori Critici (1%, 5%, 10%)
Bisettimanale	-5.6024	1.26e-06	-3.48, -2.88, -2.58
Trisettimanale	-5.9036	2.74e-07	-3.51, -2.90, -2.59
Mensile	-5.5551	1.59e-06	-3.55, -2.91, -2.59

**Tabella 4.8:** Risultati dell'ADF Test per la temperatura

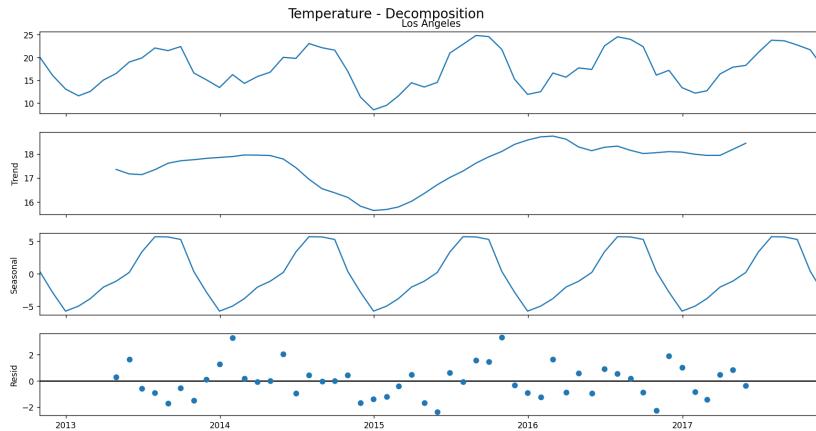
utilizzando la funzione *seasonal\_decompose*. Questa analisi ha evidenziato la presenza di una componente stagionale ben definita in ciascuna delle serie, confermando che la stagionalità è una caratteristica rilevante del dataset.



**Figura 4.2:** Decomposizione serie Bisettimanale



**Figura 4.3:** Decomposizione serie Trisettimanale



**Figura 4.4:** Decomposizione serie Mensile

Considerata la presenza di una componente stagionale, si è deciso di applicare il modello SARIMA (Seasonal AutoRegressive Integrated Moving Average), che è particolarmente adatto per gestire dati con stagionalità. La scelta del SARIMA si basa sulla capacità di questo modello di catturare sia le dipendenze temporali che le variazioni stagionali presenti nelle serie temporali.

## 4.4 SARIMA

Per la scelta dei parametri del modello SARIMA ci si è basati sulla libreria **pmdarima** che utilizza una procedura automatica per scegliere i parametri ottimali per il modello SARIMA attraverso la funzione *auto\_arima*. Questa funzione seleziona i parametri del modello ( $p$ ,  $d$ ,  $q$ ,  $P$ ,  $D$ ,  $Q$ ,  $s$ ) utilizzando tecniche di ottimizzazione e criteri statistici, seguendo i seguenti passaggi:

**pmdarima** effettua una ricerca a griglia (grid search) esplorando diverse combinazioni di parametri:

- $p$ : ordine dell'autoregressione (AR)
- $d$ : differenziazione (degree of differencing) per rendere la serie stazionaria
- $q$ : ordine della media mobile (MA)
- $P$ ,  $D$ ,  $Q$ : analoghi di  $p$ ,  $d$ ,  $q$  per la componente stagionale (SARIMA)
- $s$ : periodo stagionale (es. 12 per dati mensili, 26 per dati bisettimanali)

La funzione esplora diverse combinazioni di questi parametri per trovare il miglior set, basandosi su una ricerca combinatoria e su criteri statistici.

Per scegliere i parametri ottimali, **pmdarima** utilizza criteri di informazione come AIC (Akaike Information Criterion) e BIC (Bayesian Information Criterion). Entrambi i criteri bilanciano la qualità della previsione (massimizzare la verosimiglianza) con la complessità del modello (minimizzare il numero di parametri). L'obiettivo è ridurre al minimo questi criteri, selezionando quindi il modello che offre il miglior compromesso tra accuratezza e semplicità.

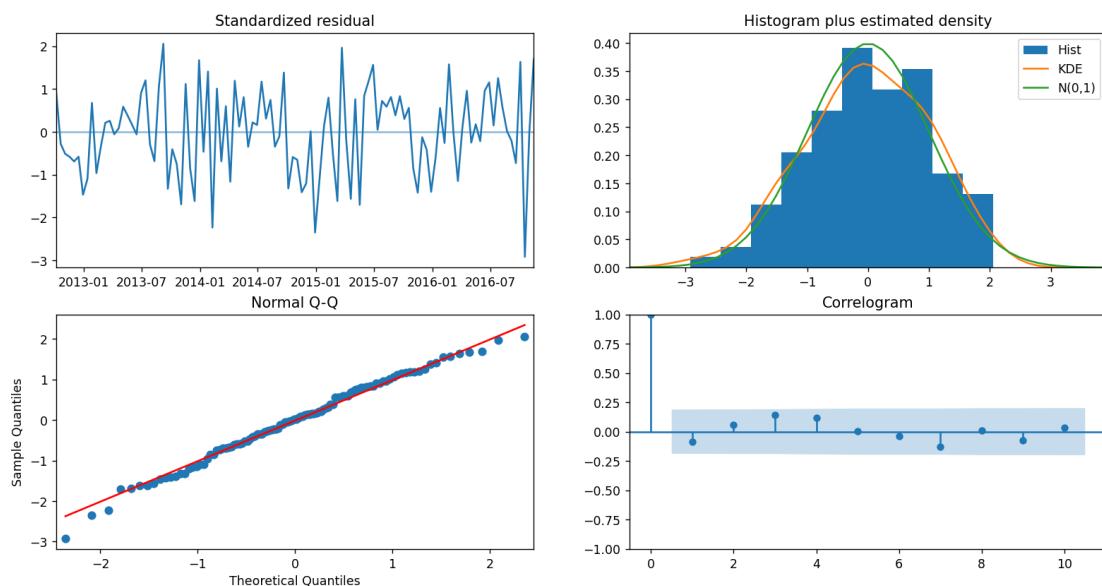
**pmdarima** effettua automaticamente test per valutare la stazionarietà della serie (ad esempio utilizzando l'ADF test) e, se necessario, applica differenziazioni (impostando i parametri  $d$  e  $D$ ) per rendere la serie stazionaria, sia per la componente normale ( $d$ ) che per quella stagionale ( $D$ ).

Durante il processo, **pmdarima** valuta anche la diagnostica dei residui e altri parametri per assicurarsi che il modello scelto sia valido e che non vi siano errori di specificazione significativi.

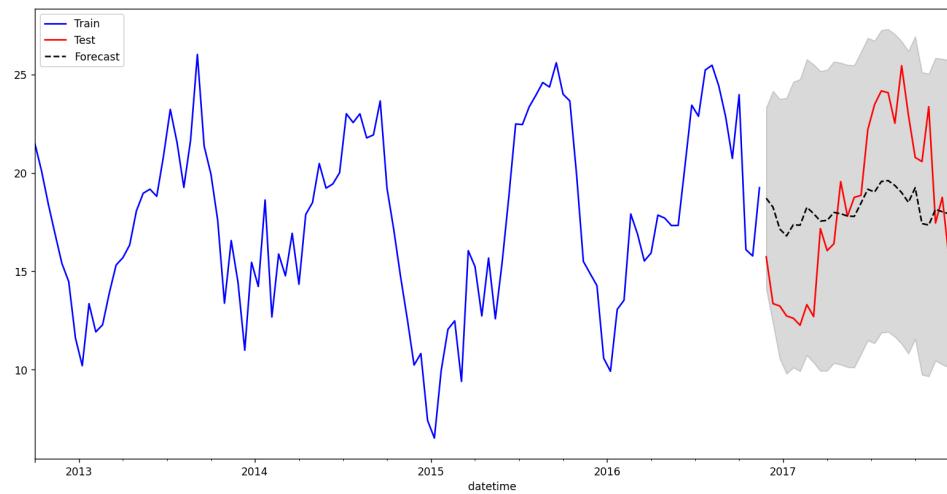
#### 4.4.1 Risultati con Aggregazione Bisettimanale

Best model	ARIMA(1,0,0)(1,0,0)[26] intercept				
Total fit time	9.653 seconds				
Parameter	coef	std err	z	P> z	[0.025, 0.975]
intercept	2.7285	0.839	3.252	0.001	[1.084, 4.373]
ar.L1	0.8008	0.059	13.669	0.000	[0.686, 0.916]
ar.S.L26	0.2311	0.106	2.181	0.029	[0.023, 0.439]
sigma2	5.5185	0.836	6.604	0.000	[3.881, 7.156]

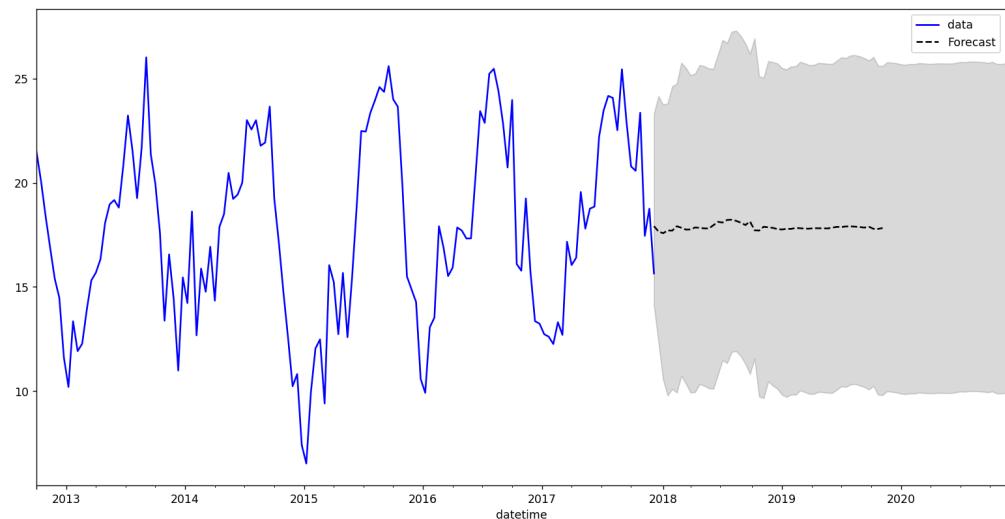
**Tabella 4.9:** Modello SARIMA per il CSV bisettimanale elaborato da pmdarima



**Figura 4.5:** Diagnostica Modello SARIMA - Aggregazione Bisettimanale



**Figura 4.6:** Forecast con dati di test - Aggregazione Bisettimanale

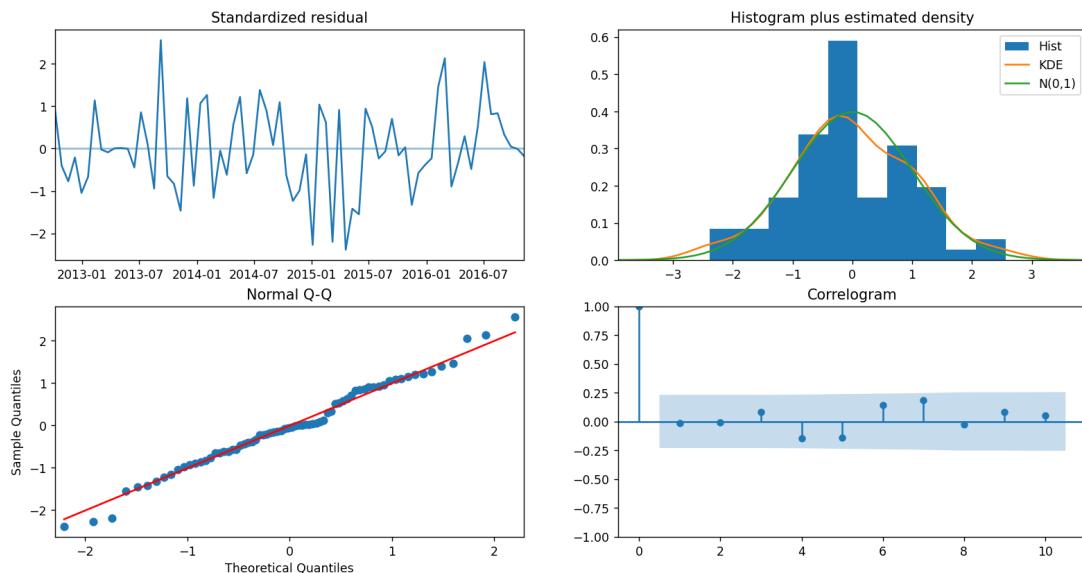


**Figura 4.7:** Forecast di 3 anni - Aggregazione Bisettimanale

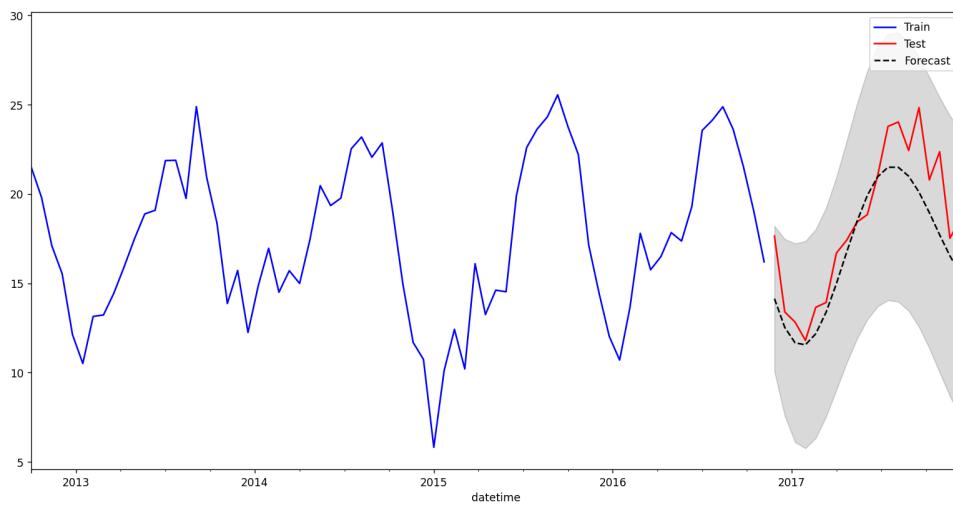
<b>Best model</b>	ARIMA(3,0,1)(0,0,0)[17] intercept				
<b>Total fit time</b>	5.102 seconds				
Parameter	coef	std err	z	P> z	[0.025, 0.975]
intercept	2.6822	0.706	3.797	0.000	[1.298, 4.067]
ar.L1	1.5309	0.157	9.743	0.000	[1.223, 1.839]
ar.L2	-0.4356	0.245	-1.776	0.076	[-0.916, 0.045]
ar.L3	-0.2493	0.127	-1.963	0.050	[-0.498, -0.000]
ma.L1	-0.8387	0.125	-6.691	0.000	[-1.084, -0.593]
sigma2	4.3029	0.798	5.393	0.000	[2.739, 5.867]

**Tabella 4.10:** Modello SARIMA per il CSV Trisettimanale elaborato da pmdarima

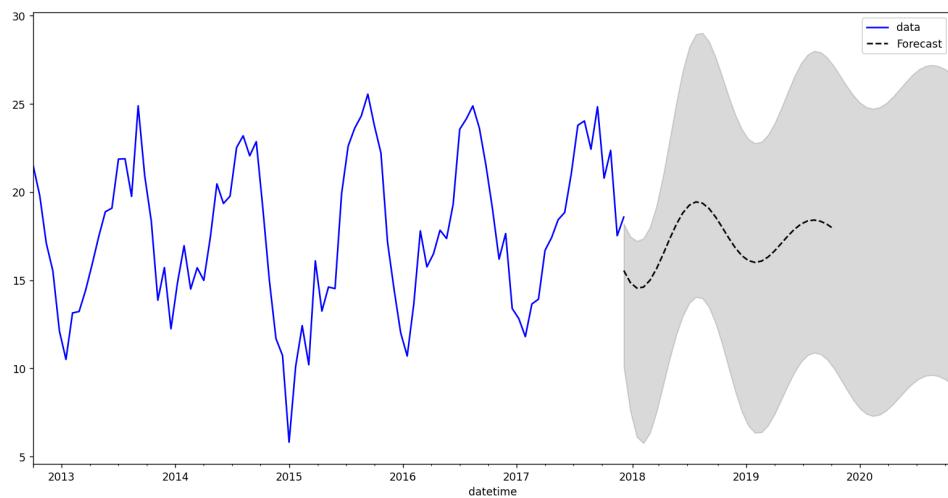
#### 4.4.2 Risultati con Aggregazione Trisettimanale



**Figura 4.8:** Diagnostica Modello SARIMA - Aggregazione Trisettimanale



**Figura 4.9:** Forecast con dati di test - Aggregazione Trisettimanale



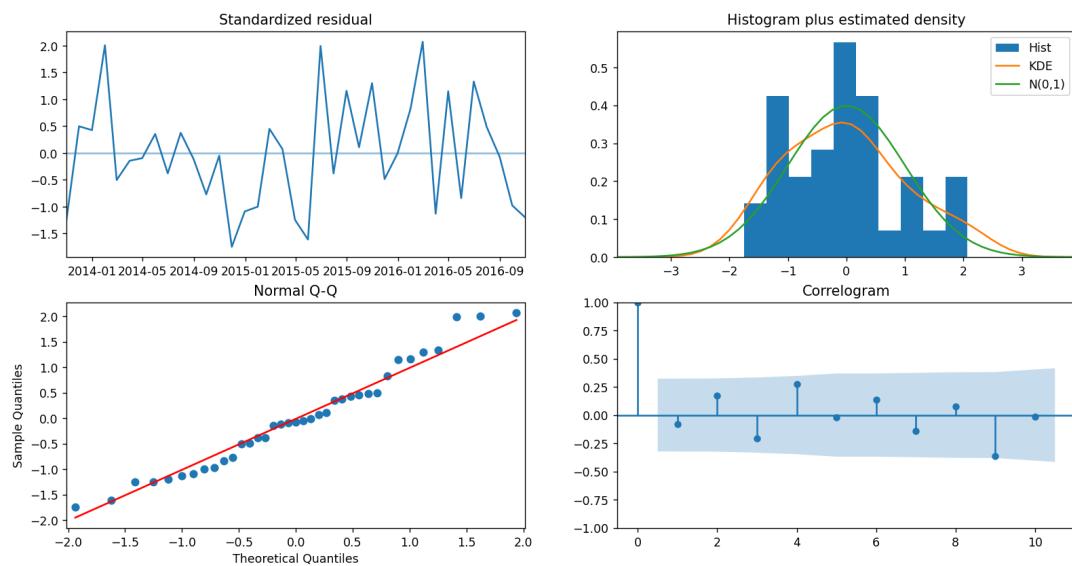
**Figura 4.10:** Forecast di 3 anni - Aggregazione Trisettimanale

#### 4.4.3 Risultati con Aggregazione Mensile

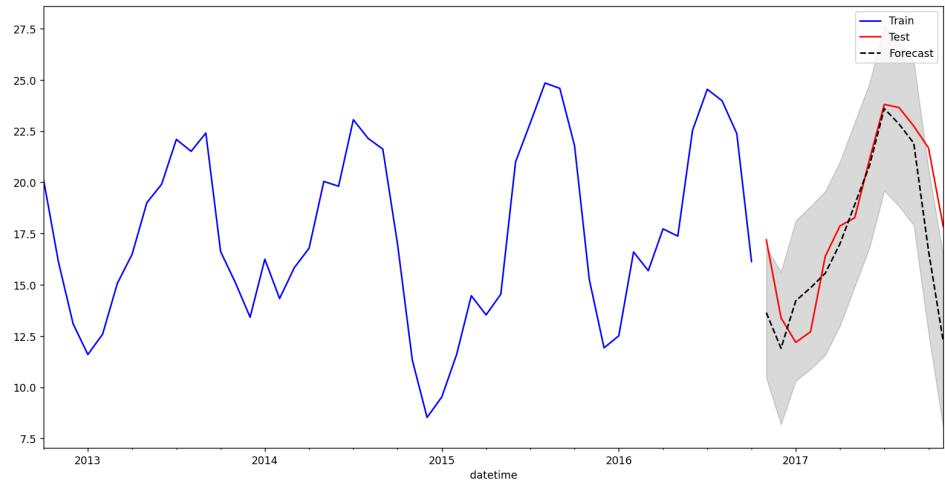
<b>Best model</b>	ARIMA(1,0,0)(2,1,0)[12] intercept				
<b>Total fit time</b>	4.897 seconds				

Parameter	coef	std err	z	P> z	[0.025, 0.975]
ar.L1	0.6104	0.143	4.280	0.000	[0.331, 0.890]
ar.S.L12	-0.6143	0.166	-3.693	0.000	[-0.940, -0.288]
ar.S.L24	-0.6109	0.316	-1.936	0.053	[-1.229, 0.008]
sigma2	2.6295	1.492	1.763	0.078	[-0.294, 5.553]

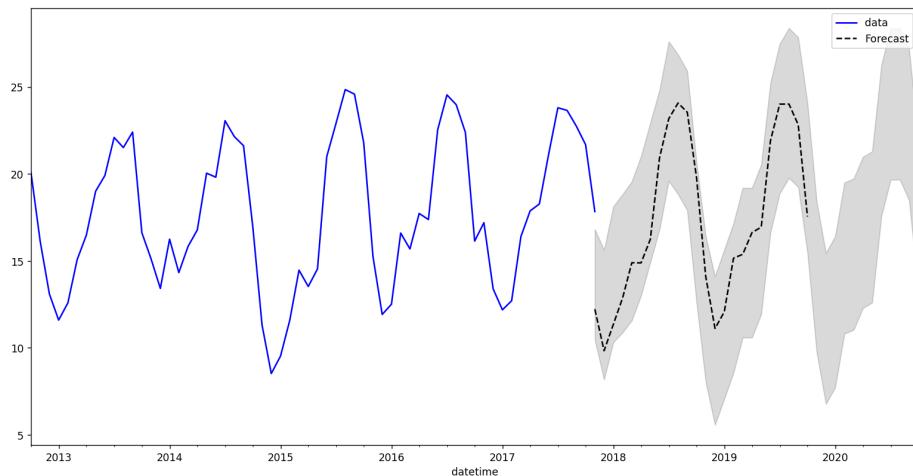
**Tabella 4.11:** Modello SARIMA per il CSV mensile elaborato da pmdarima



**Figura 4.11:** Diagnostica Modello SARIMA - Aggregazione Mensile



**Figura 4.12:** Forecast con dati di test - Aggregazione Mensile



**Figura 4.13:** Forecast di 3 anni - Aggregazione Mensile

## 4.5 Analisi dei risultati e conclusioni

Metrica	Bisettimanale	Trisettimanale	Mensile
<b>MAPE</b>	0.1826	0.0894	0.1098
<b>MSE</b>	13.1970	4.8830	6.4308
<b>MAE</b>	3.1200	1.7302	1.8717
<b>R<sup>2</sup></b>	0.2140	0.6847	0.5734

**Tabella 4.12:** Confronto dei risultati dei modelli SARIMA per le aggregazioni bisettimanale, tri-settimanale e mensile

- MAPE (Mean Absolute Percentage Error):

La trisettimanale ha il valore più basso (0.0894), indicando che questo modello ha la minore percentuale di errore medio assoluto rispetto ai valori reali. La bisettimanale ha la MAPE più alta (0.1826), suggerendo una minore accuratezza. L'aggregazione mensile si trova tra i due, ma è inferiore rispetto alla bisettimanale (0.1098).

- MSE (Mean Squared Error):

Anche in questo caso, il modello trisettimanale ha il valore più basso (4.8830), indicando che ha gli errori quadrati medi più piccoli. La mensile ha un MSE intermedio (6.4308), mentre la bisettimanale ha l'MSE più alto (13.1970), suggerendo errori più ampi nelle previsioni.

- MAE (Mean Absolute Error):

Il modello trisettimanale ha di nuovo il valore più basso (1.7302), confermando una performance migliore in termini di errore assoluto. L'aggregazione mensile ha un MAE leggermente superiore (1.8717), mentre la bisettimanale ha il MAE più alto (3.1200).

- $R^2$  (Coefficient of Determination):

Il modello trisettimanale ha il valore di più alto (0.6847), suggerendo che spiega una percentuale maggiore della variabilità dei dati rispetto agli altri modelli. Il modello mensile segue con =0.5734, mentre il bisettimanale ha il valore più basso (0.2140), indicando che spiega molto meno la variabilità dei dati rispetto agli altri due modelli.

Analizzando i grafici delle diagnostiche (Figura 4.5, 4.8, 4.11):

**Istogramma con stima KDE:** l'istogramma mostra la distribuzione osservata dei residui, mentre la linea arancione rappresenta la curva KDE (versione smussata dell'istogramma); la linea verde, invece, illustra una distribuzione normale. Per valutare un buon modello, la curva KDE (linea arancione) dovrebbe sovrapporsi o essere molto simile alla curva normale (linea verde). In tutti e tre i casi, le curve hanno una media di circa 0 e sono abbastanza simili, suggerendo che i residui seguono una distribuzione normale.

**Correlogramma:** tutti i lag si trovano all'interno della regione di ammissibilità, il che conferma la bontà del modello. Questo indica che non vi è autocorrelazione significativa nei residui.

**Q-Q Plot:** la maggior parte dei punti si dispone lungo la linea retta, indicando che i residui sono distribuiti normalmente. Tuttavia, nell'aggregazione mensile, alcuni punti si discostano maggiormente dalla linea rispetto alle altre due aggregazioni, segnalando una possibile deviazione dalla normalità in quel caso.

Osservando i grafici di previsione (forecast) con i dati di test (Figure 4.6, 4.10, 4.12), notiamo che solo nell'aggregazione mensile i dati reali si trovano al di fuori dell'intervallo di confidenza delle previsioni.

Per quanto riguarda il forecast sui tre anni successivi (Figure 4.7, 4.10, 4.13), l'aggregazione bisettimanale mostra un ampio intervallo di confidenza, con la linea di previsione che coincide principalmente con la temperatura media annuale di Los Angeles.

Nell'aggregazione trisettimanale, l'intervallo di confidenza varia seguendo l'andamento stagionale della temperatura, e la previsione rispecchia più fedelmente le fluttuazioni stagionali. Infine, l'aggregazione mensile mostra un intervallo di confidenza molto stretto, suggerendo una previsione precisa. Tuttavia, la linea di forecast, sebbene segua il trend reale, potrebbe portare a previsioni di temperature fuori dall'intervallo di confidenza.

Il modello SARIMA con aggregazione trisettimanale offre le migliori prestazioni complessive in tutte le metriche (MAPE, MSE, MAE,  $R^2$ ). Questo modello è più accurato sia in termini di errori percentuali che assoluti e riesce a spiegare meglio la variabilità dei dati. L'aggregazione mensile ha buone prestazioni ma leggermente inferiori al modello trisettimanale, mentre il modello bisettimanale risulta il meno performante in tutte le metriche. Alla luce di queste considerazioni, si conclude che il modello SARIMA (3,0,1)(0,0,0)[17], applicato ai dati aggregati trisettimanalmente, offre le prestazioni migliori in termini di accuratezza e affidabilità della previsione.

# Elenco delle figure

1.1	Dataset Heart Attack . . . . .	6
1.2	Iistogrammi . . . . .	7
1.3	Distribuzione dell'età dei campioni . . . . .	8
1.4	Percentuale di maschi e femmine . . . . .	8
1.5	Mappa ad albero per il tipo di dolore toracico (cp) . . . . .	9
1.6	Mappa ad albero per thall . . . . .	9
1.7	Andamento della pressione sanguigna a riposo . . . . .	10
1.8	Andamento dei valori del colesterolo . . . . .	11
1.9	Distribuzione dei livelli di zucchero nel sangue (fbs) . . . . .	11
1.10	Distribuzione dei risultati dell'ECG a riposo (restecg) . . . . .	12
1.11	Distribuzione della frequenza cardiaca massima (thalach) . . . . .	12
1.12	Distribuzione della variabile ‘exang’ . . . . .	13
1.13	Distribuzione dell’output . . . . .	13
1.14	Matrice di correlazione tra tutti i dati . . . . .	14
2.1	Logo Scikit Learn . . . . .	16
2.2	Standardizzazione Dataset . . . . .	17
2.3	Metodi Elbow K-Means . . . . .	18
2.4	K-Means Clustering . . . . .	19
2.5	dendrogramma . . . . .	20
2.6	cluster gerarchico . . . . .	21
2.7	Dataset PCA . . . . .	21
2.8	Knee Point DBSCAN . . . . .	22
2.9	Heatmap Parametri DBSCAN . . . . .	23
2.10	Risultato Clustering DBSCAN . . . . .	23
2.11	Metodi Elbow K-Means con PCA . . . . .	24
2.12	Silhouette (sinistra) e Clustering K-Means (destra) . . . . .	25
2.13	Confronto tra Età e Massima Frequenza Cardiaca . . . . .	26
2.14	Distribuzione cluster per numerosità (sinistra) e per genere (destra) . . . . .	26
2.15	Distribuzione dei valori di colesterolo (chol) per cluster . . . . .	27
2.16	Distribuzione dei tipi di dolore toracico (cp) per cluster . . . . .	27
2.17	Distribuzione della frequenza cardiaca massima (thalach) per cluster . . . . .	28
2.18	Distribuzione della pressione arteriosa (trtbps) per cluster . . . . .	28
3.1	Accuratezza Classificatori . . . . .	31
3.2	Accuratezza Classificatori . . . . .	31
3.3	Matrici di Confusione Classificatori . . . . .	32

3.4	Curve ROC Classificatori . . . . .	35
3.5	Correlazioni tra modelli . . . . .	36
3.6	Parametri GridSearch dei modelli scelti . . . . .	37
3.7	Accuratezza Classificatori con GridSearch . . . . .	37
3.8	Matrici di Confusione Classificatori con GridSearch . . . . .	38
3.9	Curve ROC Classificatori con GridSearc . . . . .	40
4.1	Logo StatsModels . . . . .	43
4.2	Decomposizione serie Bisettimanale . . . . .	44
4.3	Decomposizione serie Trisettimanale . . . . .	44
4.4	Decomposizione serie Mensile . . . . .	45
4.5	Diagnostica Modello SARIMA - Aggregazione Bisettimanale . . . . .	46
4.6	Forecast con dati di test - Aggregazione Bisettimanale . . . . .	47
4.7	Forecast di 3 anni - Aggregazione Bisettimanale . . . . .	47
4.8	Diagnostica Modello SARIMA - Aggregazione Trisettimanale . . . . .	48
4.9	Forecast con dati di test - Aggregazione Trisettimanale . . . . .	49
4.10	Forecast di 3 anni - Aggregazione Trisettimanale . . . . .	49
4.11	Diagnostica Modello SARIMA - Aggregazione Mensile . . . . .	50
4.12	Forecast con dati di test - Aggregazione Mensile . . . . .	50
4.13	Forecast di 3 anni - Aggregazione Mensile . . . . .	51