Understanding the Dataset

The Modified_Heart_failure_data.csv dataset contains various features, including age, ejection fraction, presence of diabetes, etc., essential in assessing cardiac health.

Clustering is suitable for this dataset because it allows us to group patients with similar health characteristics, enabling us to identify patterns and assess cardiac risk levels without requiring labelled outcomes. Age, ejection fraction, and diabetes status are significant as they are primary indicators of cardiac risk.

Features that have more influence on cluster formation are: age, Anaemia, creatinine_phosphokinase, diabetes, ejection_fraction, high_blood_pressure, platelets, serum_creatinine, serum_sodium, sex, and smoking

```
COLUMN DATA TYPE:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 299 entries, 0 to 298
Data columns (total 13 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   age                       270 non-null    float64
 1   anaemia                   270 non-null    float64
 2   creatinine_phosphokinase  270 non-null    float64
 3   diabetes                  270 non-null    float64
 4   ejection_fraction         270 non-null    float64
 5   high_blood_pressure       270 non-null    float64
 6   platelets                 270 non-null    float64
 7   serum_creatinine          270 non-null    float64
 8   serum_sodium              270 non-null    float64
 9   sex                       270 non-null    float64
 10  smoking                   270 non-null    float64
 11  time                      270 non-null    float64
 12  patient_id                299 non-null    int64
dtypes: float64(12), int64(1)
memory usage: 30.5 KB
```

DATA SUMMARY:

	count	mean	std	min	25%	50%	75%	max
age	270.0	60.934570	11.838766	40.0	52.00	60.0	70.0	95.0
anaemia	270.0	0.437037	0.496941	0.0	0.00	0.0	1.0	1.0
creatinine_phosphokinase	270.0	574.266667	958.532760	23.0	119.50	250.0	582.0	7861.0
diabetes	270.0	0.411111	0.492949	0.0	0.00	0.0	1.0	1.0
ejection_fraction	270.0	37.622222	11.715885	14.0	30.00	38.0	45.0	70.0
high_blood_pressure	270.0	0.355556	0.479570	0.0	0.00	0.0	1.0	1.0
platelets	270.0	261522.410185	97633.259068	25100.0	210250.00	261000.0	302000.0	850000.0
serum_creatinine	270.0	1.355519	0.981258	0.5	0.90	1.1	1.4	9.4
serum_sodium	270.0	136.722222	4.226033	113.0	134.00	137.0	139.0	148.0
sex	270.0	0.640741	0.480674	0.0	0.00	1.0	1.0	1.0
smoking	270.0	0.300000	0.459109	0.0	0.00	0.0	1.0	1.0
time	270.0	131.292593	77.579169	4.0	73.25	120.0	201.0	285.0
patient_id	299.0	149.000000	86.458082	0.0	74.50	149.0	223.5	298.0

age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	patient_id
75.0	0.0	582.0	0.0	20.0	1.0	265000.00	1.9	130.0	1.0	0.0	4.0	0
55.0	0.0	7861.0	0.0	38.0	0.0	263358.03	1.1	136.0	1.0	0.0	6.0	1
65.0	NaN	146.0	0.0	20.0	0.0	162000.00	1.3	129.0	1.0	NaN	7.0	2
50.0	1.0	111.0	0.0	20.0	0.0	210000.00	NaN	137.0	1.0	0.0	7.0	3
65.0	1.0	160.0	1.0	20.0	0.0	327000.00	2.7	116.0	0.0	0.0	8.0	4

Data Pre-processing

- Handling missing values, scaling features, and removing outliers ensures that the data is clean and consistent, which is crucial for clustering algorithms that rely on distance metrics and data distribution.
- Without transformations like scaling, outlier removal, and dimensionality reduction, clustering algorithms might produce distorted or irrelevant clusters. Transformations ensure that the algorithm can accurately group similar data points.
- By ensuring that all features contribute equally (through scaling and outlier handling), the clustering results become more representative of the true patterns in the data, rather than being skewed by particular features or extreme values.
- Proper transformations improve clustering accuracy and make the results easier to interpret. Clean, scaled, and outlier-free data leads to clusters that have meaningful patterns, making them actionable for healthcare professionals and decision-makers.

While checking the dataset, identified 29 missing values in each column, which are handled by filling them out by mean or median values. Here, the Median is used for Numerical features, and Mode is used for binary features.

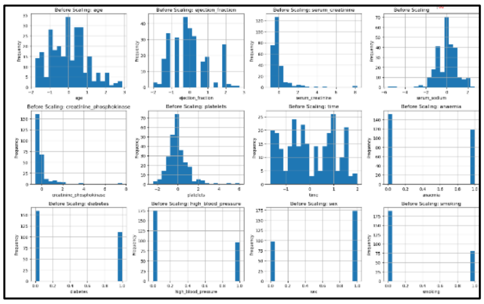
```
missing values in %

age                         9.698997
anaemia                     9.698997
creatinine_phosphokinase    9.698997
diabetes                    9.698997
ejection_fraction           9.698997
high_blood_pressure         9.698997
platelets                   9.698997
serum_creatinine            9.698997
serum_sodium                9.698997
sex                         9.698997
smoking                     9.698997
time                        9.698997
patient_id                  0.000000
dtype: float64
```

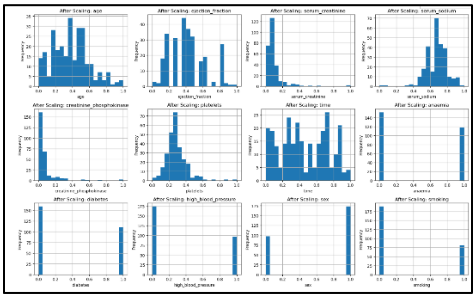
```
Missing values in each column:
 age                        29
anaemia                     29
creatinine_phosphokinase    29
diabetes                    29
ejection_fraction           29
high_blood_pressure         29
platelets                   29
serum_creatinine            29
serum_sodium                29
sex                         29
smoking                     29
time                        29
patient_id                   0
dtype: int64
```

```
Missing values after imputatio
 age                        0
anaemia                     0
creatinine_phosphokinase    0
diabetes                    0
ejection_fraction           0
high_blood_pressure         0
platelets                   0
serum_creatinine            0
serum_sodium                0
sex                         0
smoking                     0
time                        0
patient_id                  0
dtype: int64
```

Feature Scaling: Before Scaling: The histograms show the original range and distribution of each numerical feature, which may vary significantly. After Scaling: The histograms for scaled features now show values centered around 0 with a standard deviation close to 1. This adjustment is essential for Clustering since it ensures that features contribute equally to distance calculations, leading to more meaningful cluster formations.

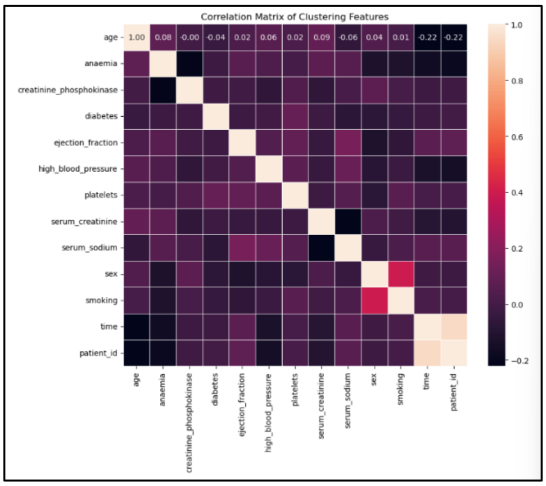
Without scaling, features with more extensive numerical ranges (like creatinine_phosphokinase or platelets) would dominate the clustering algorithm, leading to biased clusters. After scaling, each feature is normalized to the same range, making clustering algorithms more effective.



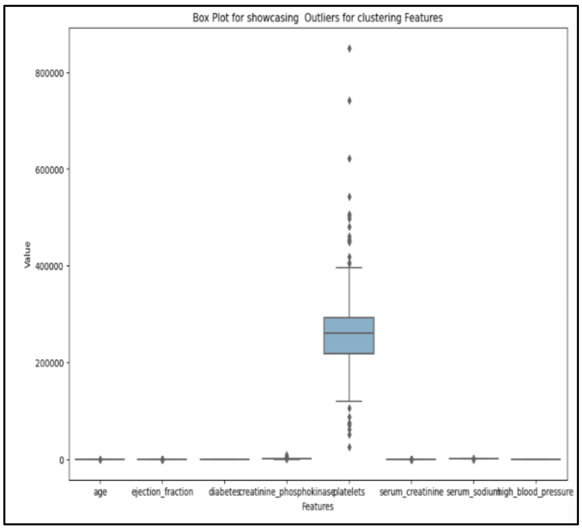


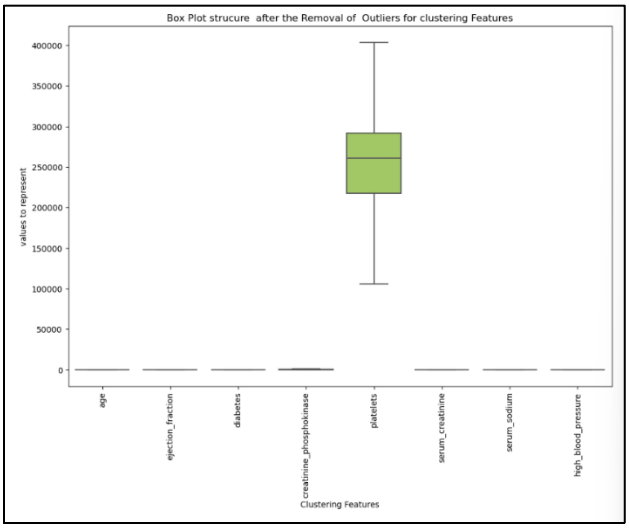
Correlation Matrix: It is a valuable tool in Clustering, providing insights into the relationships between variables that can help optimize the clustering process. By detecting redundant features, guiding feature selection, and improving distance calculations, the correlation matrix ensures that clustering algorithms work with meaningful and uncorrelated data, ultimately leading to more accurate and interpretable clusters. When preparing data for Clustering, a well-analyzed correlation matrix should be an essential step in the data preprocessing pipeline.

The weakly correlated fields are typically selected from the correlation matrix to evaluate the danger to cardiac health. While there is a weak link between age and ejection fraction (0.25) and high blood pressure (0.74), there is a negative correlation between serum sodium and serum creatinine (-0.22). Age,diabetes, ejection fraction, platelets, serum creatinine, and serum sodium are the clusters selected for this. Although there is a strong correlation between 0.46 and the other features—Patient_Id, Time, Sex, and Smoking—they have little bearing on the assessment of cardiac health risk.

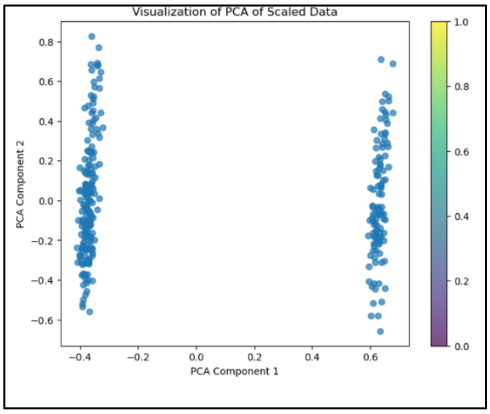


Outlier Detection: In medical datasets, some outliers may be significant (e.g., rare diseases), and removing them could lead to losing valuable insights. Thus, domain expertise is necessary to decide whether an outlier represents an error or an important medical condition. Here outlier is handled because it can distort statistical analyses or predictive models by overemphasizing rare cases or anomalies. By removing or transforming outliers, you can ensure the analysis more accurately reflects the population. Handling outliers ensures the data is clean, reducing noise and making it more suitable for machine learning algorithms and statistical analysis.





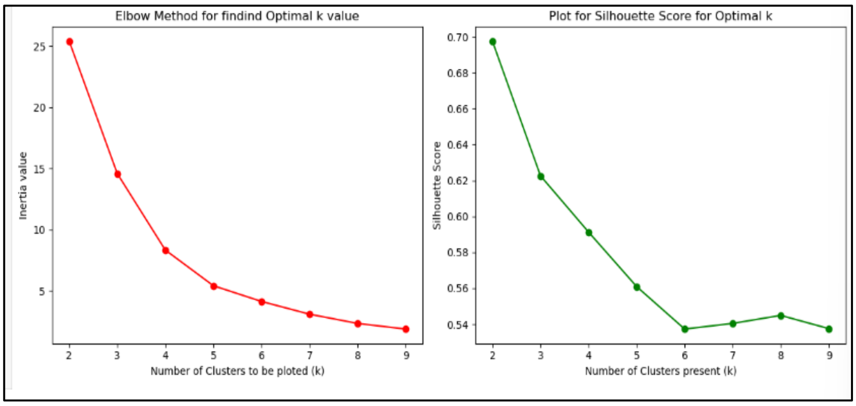
Principal component Analysis is an important technique in statistical analysis used for dimensionality reduction, which transforms a high-dimensional dataset into a lower-dimensional one while retaining most of the original data's variance. It is often used in data preprocessing, particularly when dealing with large datasets with many variables. PCA helps simplify the data, making it easier to visualize, analyse, and process while minimizing information loss. In this, the dimensionality is reduced to 2 components and its well suited because of its linear relationship.

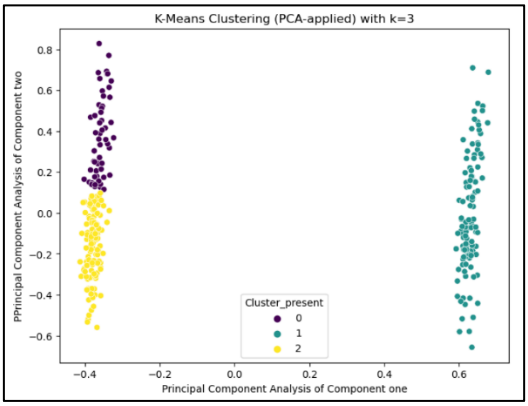


Applying Clustering Algorithms

K- Means Clustering is an unsupervised machine learning algorithm used for clustering datapoints into groups or clusters based on their similarity. The value of K, representing the number of clusters, must be determined beforehand. Common methods to determine the optimal value for K include: **Elbow Method:** The Elbow Method involves plotting the WCSS for different values of KKK and looking for an "elbow" point where the rate of decrease in WCSS slows down. The K at the elbow point is considered the optimal number of clusters. **Silhouette Score:** The Silhouette Score evaluates how similar each point is to its cluster compared to other clusters.A greater silhouette score suggests that the clusters are well-separated. The best value of K is the one that maximizes the silhouette score.

K-Means is a repetitive algorithm that partitions data into **K** separate, non-overlapping clusters. K-Means works best with **large datasets** that contain clearly distinguished clusters.

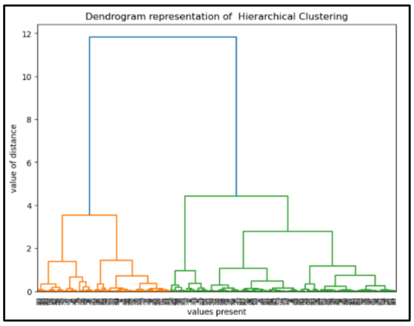


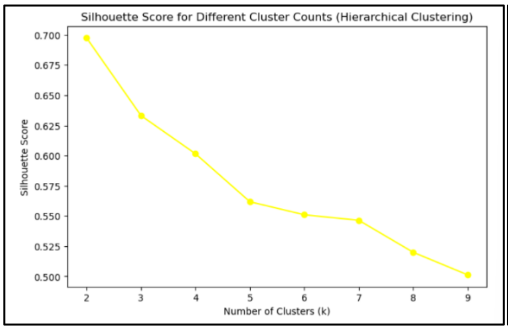


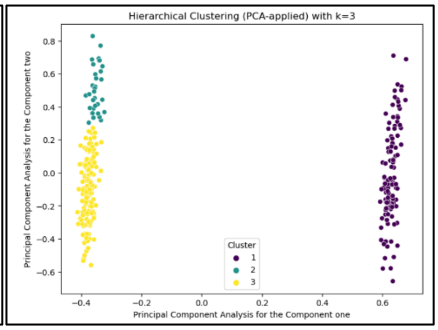
Hierarchial technique that builds a tree-like structure called a dendrogram, which shows the arrangement of clusters. In contrast to K-Means, hierarchical clustering does not necessitate that the user determine the number of clusters in advance. It works by either merging smaller clusters into larger ones or dividing larger clusters into smaller ones. The most common approach used is agglomerative hierarchical Clustering.

Hierarchical Clustering does not need the number of clusters to be specified ahead. It builds a tree of clusters (a dendrogram) by following two main approaches: **Agglomerative (bottom-up)**: Start with each data point as its cluster and iteratively merge the closest clusters. **Divisive (top-down)**: Start with all points in one cluster and iteratively split the clusters.

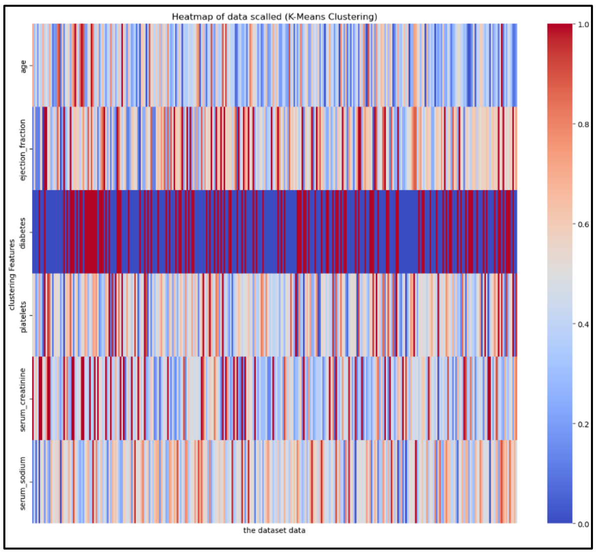
It's useful when you want to visualize the hierarchy of clusters (e.g., a dendrogram). It's beneficial for smaller datasets and when the exact number of clusters is unknown.

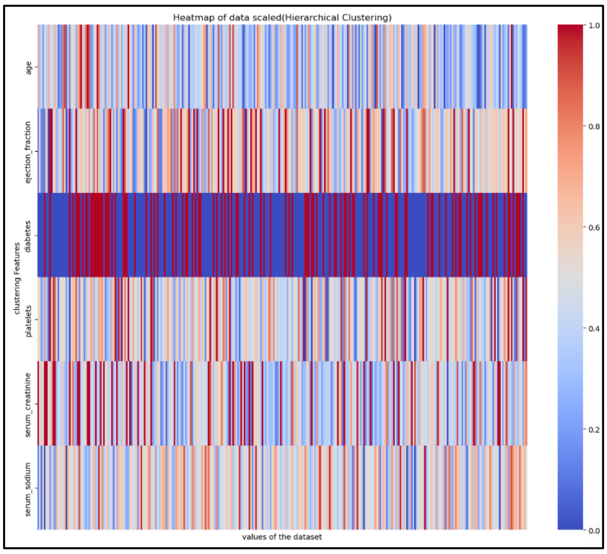






A **heatmap** is a great way to visualize cluster membership across the features. In the heatmap, each row represents a data point, and the columns represent the features. The clusters will be color-coded.





Comparative Analysis of the Clusters:

```
Final Silhouette Score of K means clustering: 0.62
Final Silhouette Score for Hierarchical Clustering: 0.63
```

K-Means Clustering: This method often performs well when the data is roughly spherical and evenly distributed across clusters. It is fast and efficient for large datasets. If the clusters are not well-separated or if the data is not spherical, K-Means might struggle

Hierarchical Clustering: Hierarchical Clustering can capture more complex structures and does not require the number of clusters to be specified upfront. However, it is computationally expensive for large datasets and might not perform well if the data has noise or outliers.

By comparing the Silhouette Scores, we can assess which algorithm gives better cluster cohesion and separation for this dataset. If the scores are similar, it suggests that both algorithms have found similar cluster structures. If they are significantly different, one algorithm may have performed better depending on the data's inherent structure.

Interpretation and Usefulness

When assessing cardiac health risks, the interpretability of clusters in healthcare hinges on how well the clusters differentiate between risk levels based on patient data. If clusters align with known health risk factors, such as age, blood pressure, cholesterol, and lifestyle, they can offer meaningful insights. For example, clusters reveal low, medium, and high cardiac risk groups, helping clinicians understand typical patient profiles within each risk category.

In real-world applications, Clustering can be highly useful for identifying high-risk patients, enabling proactive healthcare measures. For instance, hospitals could use Clustering to segment patients into groups based on their probability of experiencing cardiac events. This segmentation would allow healthcare providers to tailor interventions for each group such as prioritizing intensive monitoring for high-risk patients or developing specific wellness programs for moderate-risk groups. In this way, clustering aids in optimizing resources and targeting care to improve patient outcomes.