Machine Learning Report

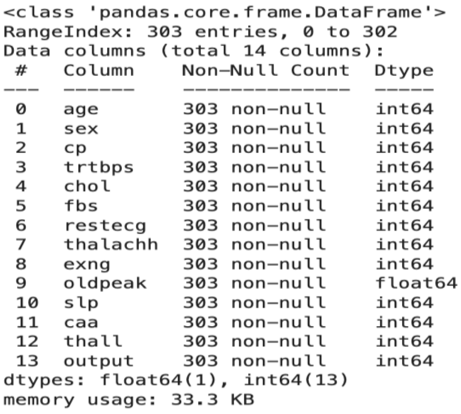
Siddharth Bahekar
College of Engineering
Northeastern University
Toronto,ON
bahekar.si@northeastern.edu

Abstract

In this report, I have applied and explored the performance of unsupervised learning algorithm on two datasets. This report outlines the exploratory data analysis (EDA) and modeling process undertaken to implement the clustering algorithms, PCA, ICA and several experiments on the algorithms. The two datasets included are **Heart Attack Analysis & Prediction** dataset and **Wine Quality** dataset. The learning curve, model complexity and clustering has been performed and analyzed on both the datasets.

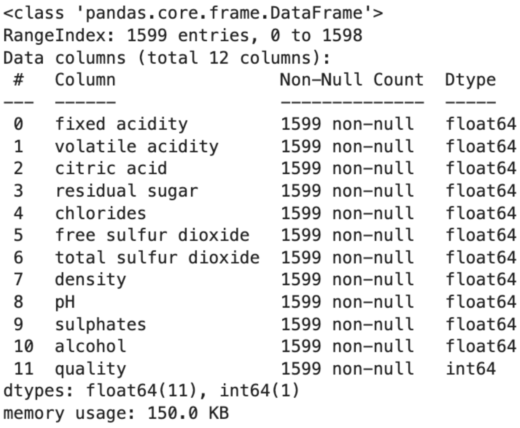
1. Dataset

The **heart attack analysis** dataset provides a comprehensive set of attributes for heart disease prediction, allowing for in-depth analysis of various factors contributing to heart health. The heart disease dataset contains 303 rows and 14 columns. The problem at hand is to classify whether a patient has heart disease or not based on various medical attributes. This is a binary classification problem using a dataset containing features and other relevant medical information.



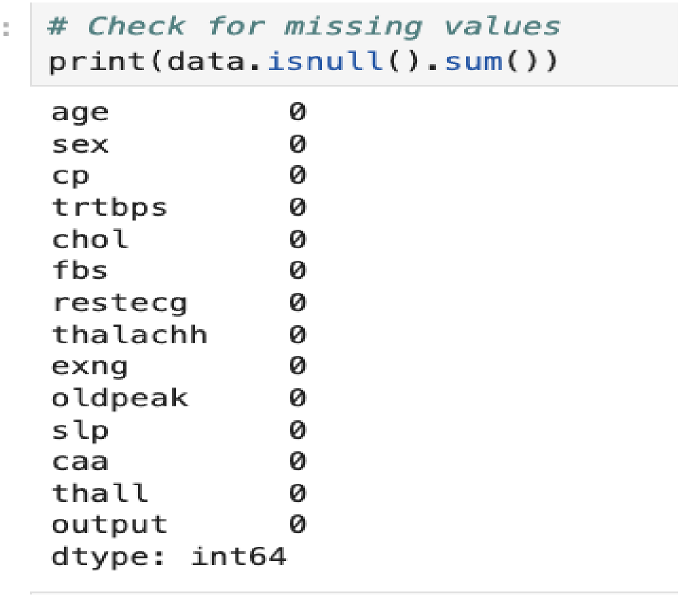
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trtbps    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
```

The **wine quality** dataset contains observations related to the physicochemical properties of red wine, including features and the target variable: wine quality. The dataset used in this analysis comprises 1,599 rows and 12 columns. Each row represents a unique wine sample with its respective measurements. The target variable, "quality," is an ordinal feature ranging from 3 to 8, indicating the wine's quality rating. This dataset provides a comprehensive view of various chemical attributes that potentially influence wine quality, allowing for in-depth exploratory data analysis and predictive modeling.



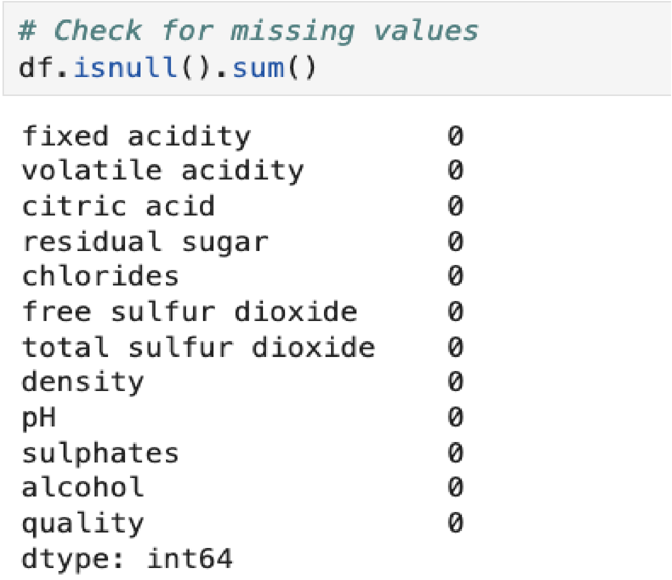
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

2. Null Values



```
# Check for missing values
print(data.isnull().sum())
```

```
age         0
sex         0
cp          0
trtbps      0
chol        0
fbs         0
restecg     0
thalachh    0
exng        0
oldpeak     0
slp         0
caa         0
thall       0
output      0
dtype: int64
```



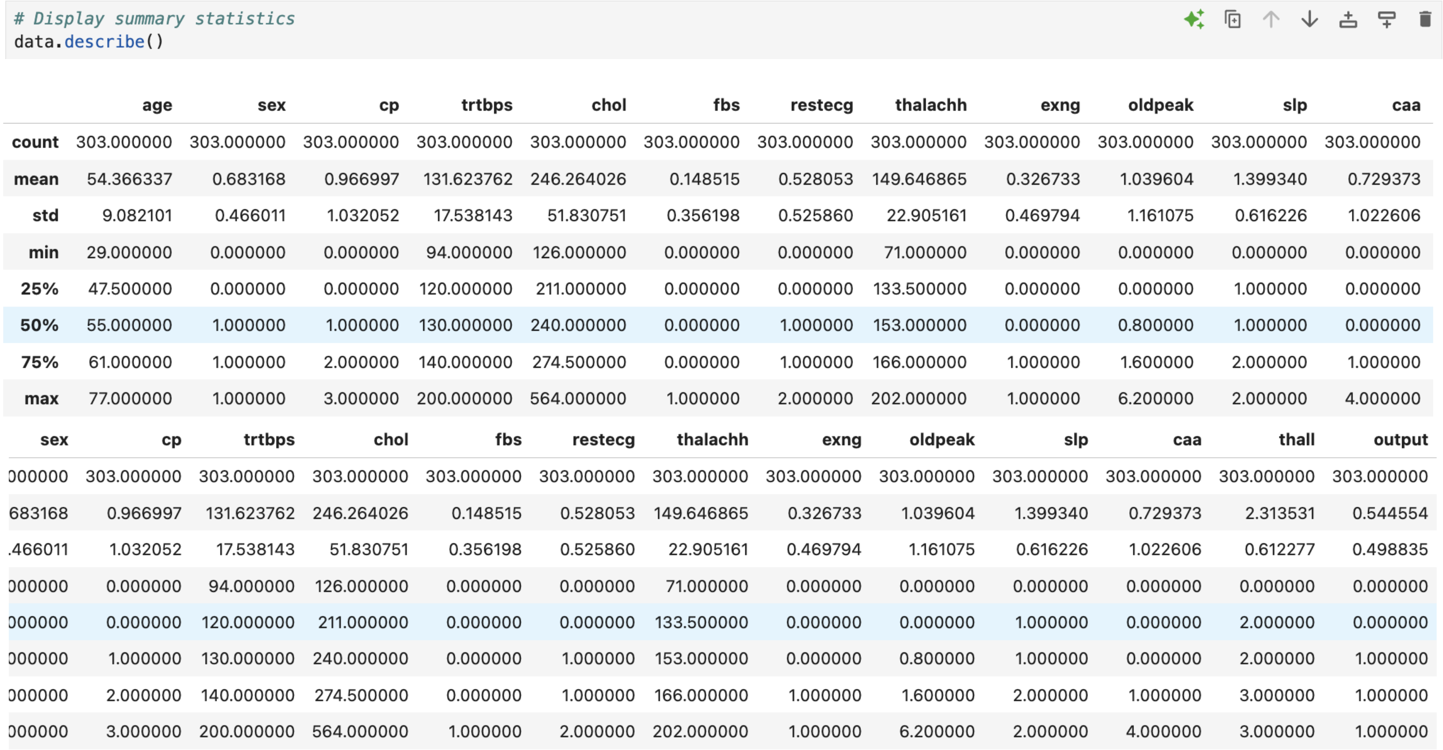
```
# Check for missing values
df.isnull().sum()
```

```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

Both the datasets do not contain any null or missing values, so there is no need to clean the data.

3. Dataset Description

Heart Attack dataset

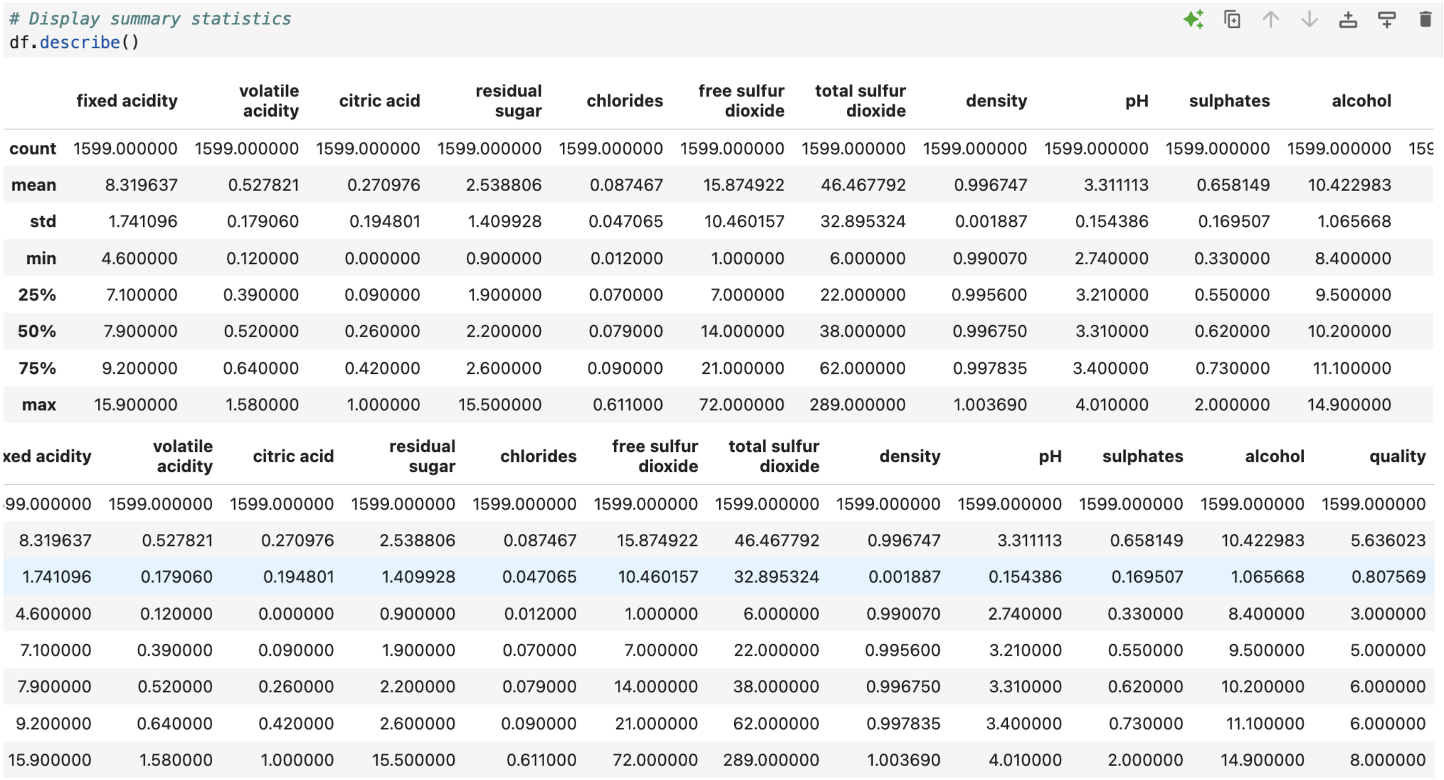


```
# Display summary statistics
data.describe()
```

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000

sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531	0.544554
.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277	0.498835
000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

Wine Quality dataset



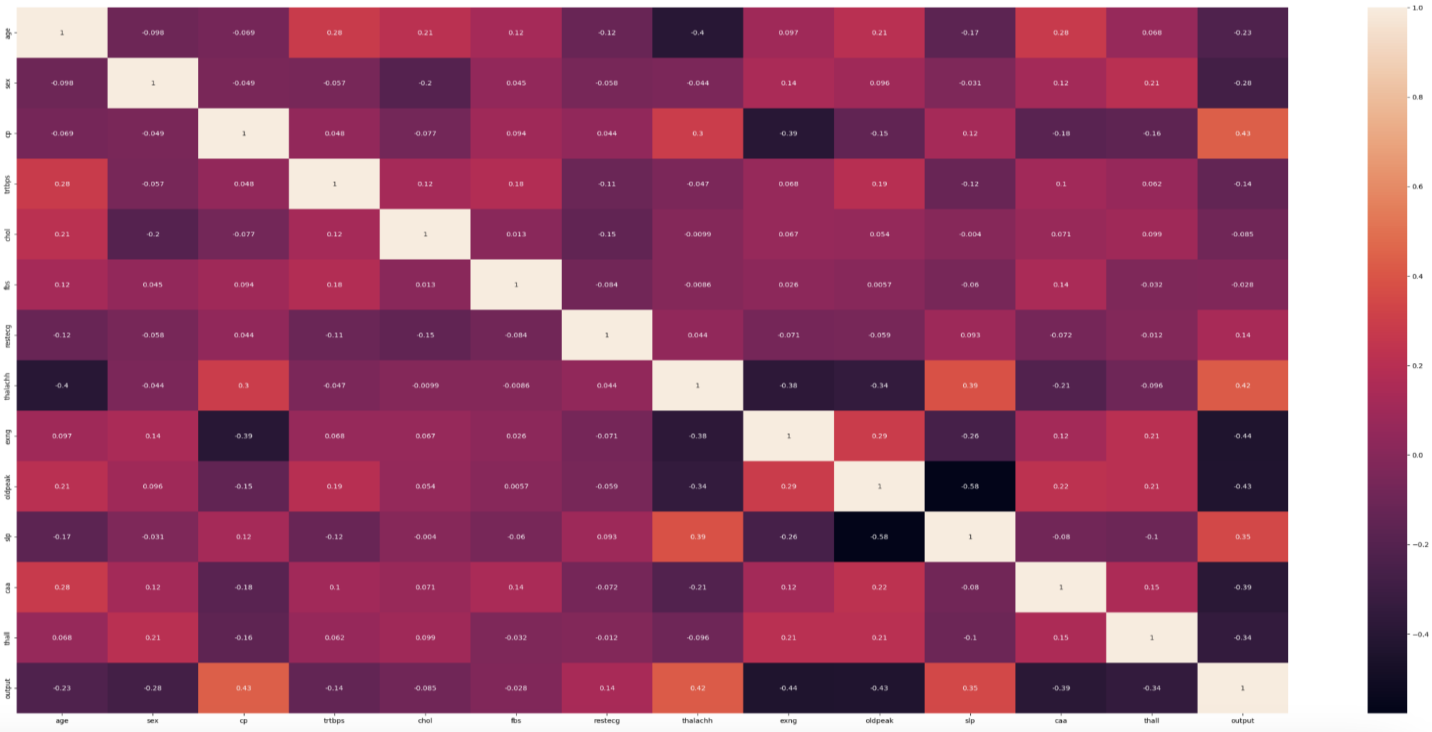
```
# Display summary statistics
df.describe()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	159
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	

xed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
99.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.636023
1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.807569
4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000000
7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000000
9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000000
15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000

4. Correlation Matrices

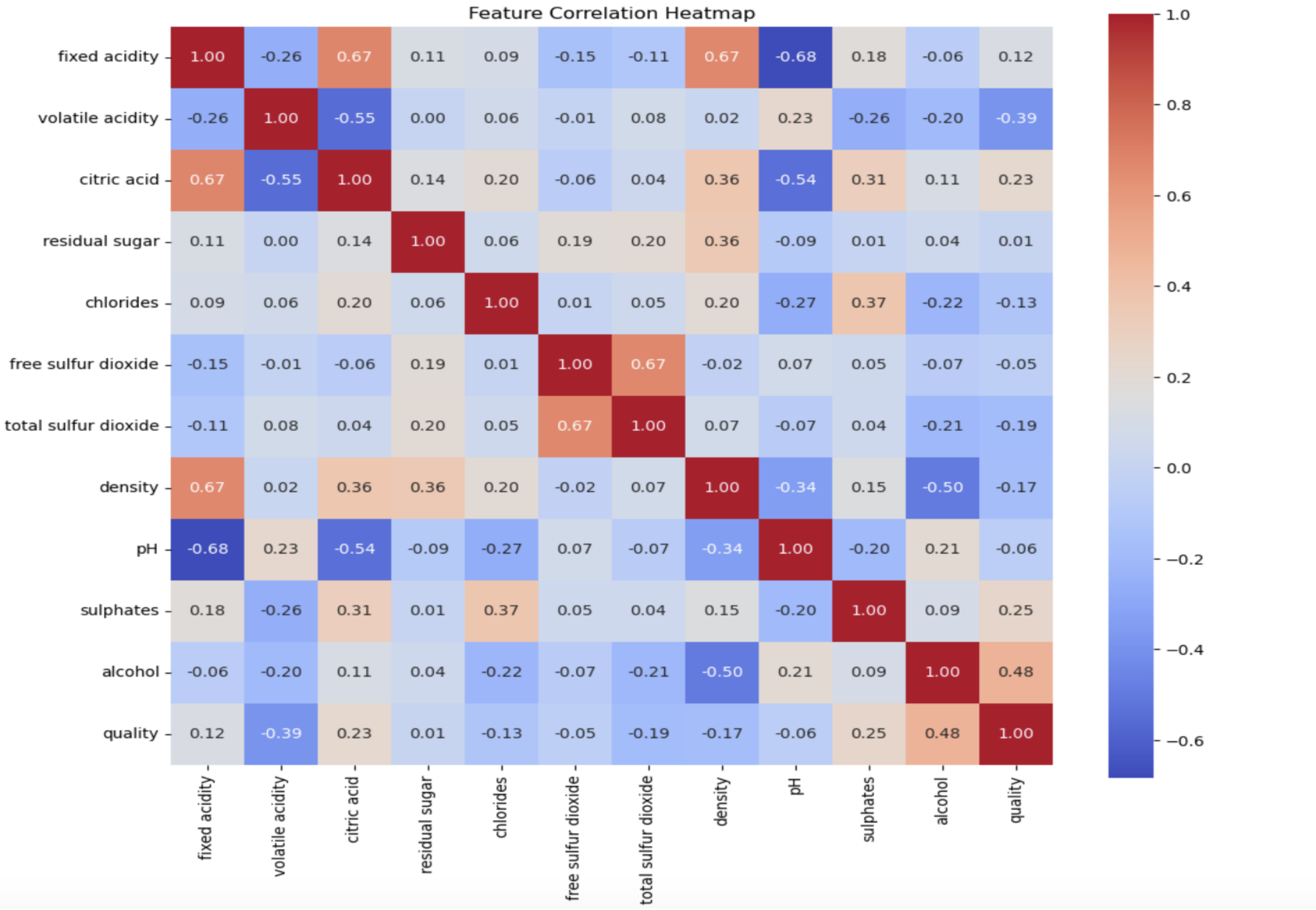
I generated a correlation heatmap to visualize the relationships between different features in our heart disease dataset. A strong positive correlation was observed between 'chest pain' and the target variable, indicating that certain types of chest pain are closely associated with the presence of heart disease. 'Max heart rate' showed a moderate negative correlation with the target variable, suggesting that lower maximum heart rates might be indicative of heart disease. 'ST depression' exhibited a positive correlation with the target, implying that higher ST depression values are associated with an increased likelihood of heart disease. These correlations provided valuable insights into which features might be most predictive of heart disease, guiding our feature selection process and helping us understand the underlying patterns in the data.



	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
age	1	-0.098	-0.069	0.28	0.21	0.12	-0.12	-0.4	0.097	0.21	-0.17	0.28	0.068	-0.23
sex	-0.098	1	-0.049	-0.057	-0.2	0.045	-0.058	-0.044	0.14	0.096	-0.031	0.12	0.21	-0.28
cp	-0.069	-0.049	1	0.048	-0.077	0.094	0.044	0.3	-0.39	-0.15	0.12	-0.18	-0.16	0.43
trtbps	0.28	-0.057	0.048	1	0.12	0.18	-0.11	-0.047	0.068	0.19	-0.12	0.1	0.062	-0.14
chol	0.21	-0.2	-0.077	0.12	1	0.013	-0.15	-0.0099	0.067	0.054	-0.004	0.071	0.099	-0.085
fbs	0.12	0.045	0.094	0.18	0.013	1	-0.084	-0.0086	0.026	0.0057	-0.06	0.14	-0.032	-0.028
restecg	-0.12	-0.058	0.044	-0.11	-0.15	-0.084	1	0.044	-0.071	-0.059	0.093	-0.072	-0.012	0.14
thalachh	-0.4	-0.044	0.3	-0.047	-0.0099	-0.0086	0.044	1	-0.38	-0.34	0.39	-0.21	-0.096	0.42
exng	0.097	0.14	-0.39	0.068	0.067	0.026	-0.071	-0.38	1	0.29	-0.26	0.12	0.21	-0.44
oldpeak	0.21	0.096	-0.15	0.19	0.054	0.0057	-0.059	-0.34	0.29	1	-0.58	0.22	0.21	-0.43
slp	-0.17	-0.031	0.12	-0.12	-0.004	-0.06	0.093	0.39	-0.26	-0.58	1	-0.08	-0.1	0.35
caa	0.28	0.12	-0.18	0.1	0.071	0.14	-0.072	-0.21	0.12	0.22	-0.08	1	0.15	-0.39
thall	0.068	0.21	-0.16	0.062	0.099	-0.032	-0.012	-0.096	0.21	0.21	-0.1	0.15	1	-0.34
output	-0.23	-0.28	0.43	-0.14	-0.085	-0.028	0.14	0.42	-0.44	-0.43	0.35	-0.39	-0.34	1

The correlation heatmap provides a visual representation of the relationships between different features in the dataset. It highlights how each feature correlates with wine quality and with each other. Notably, alcohol content shows a strong positive correlation with quality, suggesting its importance in predicting higher quality wines.

Volatile acidity has a negative correlation, indicating that higher acidity might detract from perceived quality. Features like residual sugar and chlorides exhibit low correlation with quality, suggesting they may have less impact on the model's predictive power.



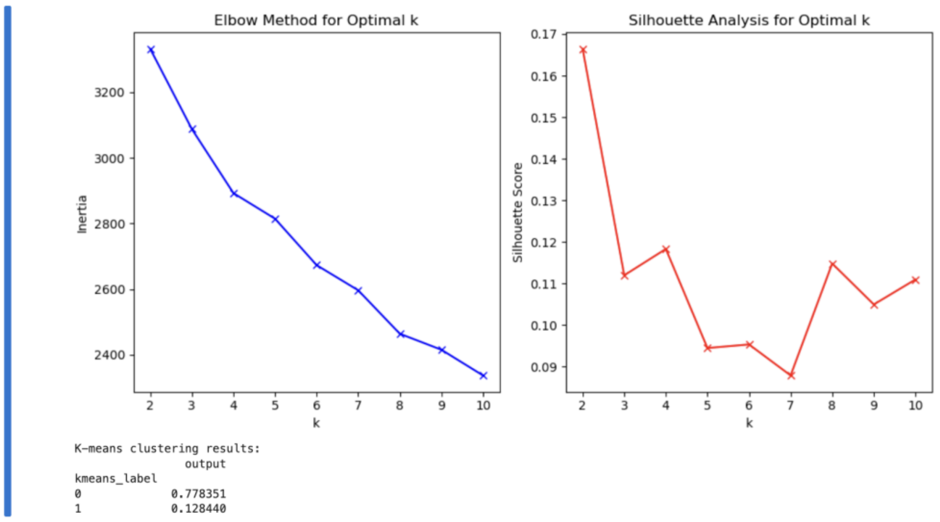
5. Why are these datasets interesting?

The dataset I have analyzed presents a compelling case study that bridges the gap between medical science and data analytics. Its real-world implications in potentially aiding early diagnosis of a leading cause of mortality worldwide make it particularly significant. The wine quality dataset is particularly interesting due to its practical application in the wine industry and its potential to reveal insights into the factors influencing wine quality. By analyzing physicochemical properties such as acidity, sugar content, and alcohol levels, I can identify key attributes that contribute to higher quality ratings. The challenge of addressing class imbalance and outliers provides a rich opportunity for applying advanced data preprocessing and machine learning techniques. This makes the dataset a valuable resource for both academic research and industry applications.

6. Clustering algorithms on raw datasets

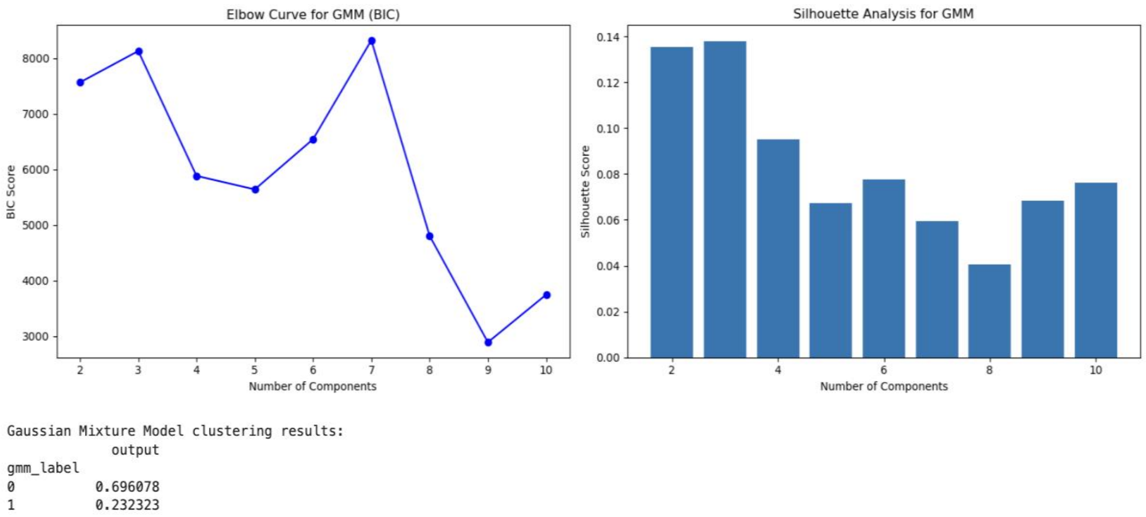
Dataset 1 -

K-means clustering analysis applied to a dataset, related to heart disease, with two main evaluation plots: the Elbow Method and Silhouette Analysis. The Elbow Method plot (left) shows how inertia (sum of squared distances between data points and their cluster centroids) decreases as the number of clusters, k, increases from 2 to 10. The curve suggests that the optimal number of clusters might be around k = 4, where the rate of inertia reduction slows down. The Silhouette Analysis plot (right) measures the silhouette score, which evaluates the separation between clusters. The highest silhouette score (~0.17) occurs at k = 2, indicating that two clusters provide the best separation in this case. Finally, the clustering results for k = 2 are displayed, showing that cluster 0 has an average 'output' value of 0.778 (likely indicating a higher probability of heart disease), while cluster 1 has an average 'output' of 0.128 (indicating a lower probability). This suggests that K-means with two clusters effectively distinguishes between individuals with higher and lower risks of heart disease.



```
K-means clustering results:
                 output
kmeans_label
0              0.778351
1              0.128440
```

Gaussian Mixture Model (GMM) clustering analysis on a dataset, related to heart disease, and evaluates the optimal number of clusters using the Bayesian Information Criterion (BIC) and Silhouette Analysis. The Elbow Curve (left plot) shows how the BIC score changes as the number of components (clusters) increases from 2 to 10. The curve suggests that the optimal number of components might be around 4, where the BIC score reaches its lowest point (~5500). The Silhouette Analysis (right plot) evaluates how well-separated and cohesive the clusters are, with higher silhouette scores indicating better-defined clusters. The highest silhouette scores (~0.14) are observed for 2 and 4 components, suggesting that these cluster configurations provide the best separation. After determining that 2 components offer a good balance between model simplicity and separation quality, GMM is applied with 2 components, and the clustering results show that cluster 0 has an average 'output' value of 0.696, while cluster 1 has an average 'output' of 0.232, indicating that cluster 0 likely represents individuals with a higher probability of heart disease. This analysis demonstrates how GMM can be used to identify distinct subgroups in medical data, with BIC and silhouette scores guiding the selection of the optimal number of clusters.

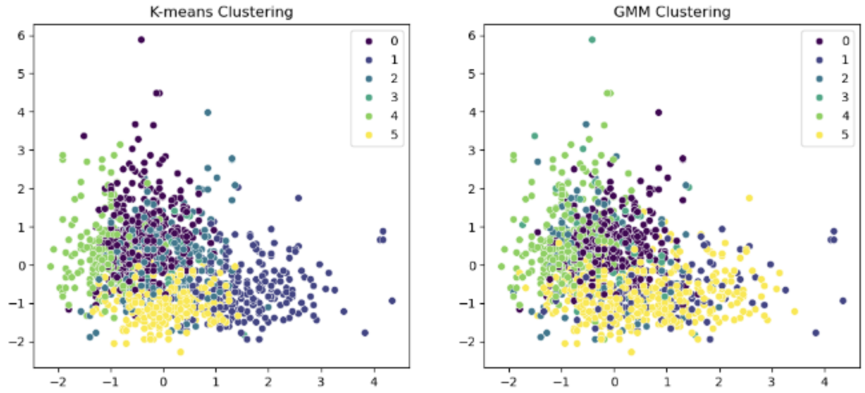


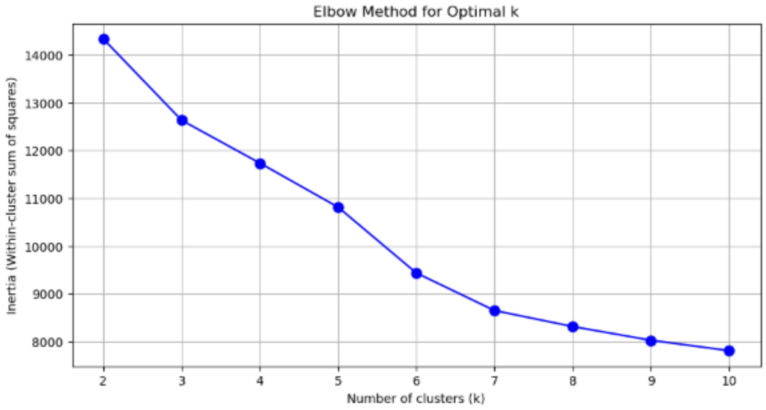
```
Gaussian Mixture Model clustering results:
              output
gmm_label
0           0.696078
1           0.232323
```

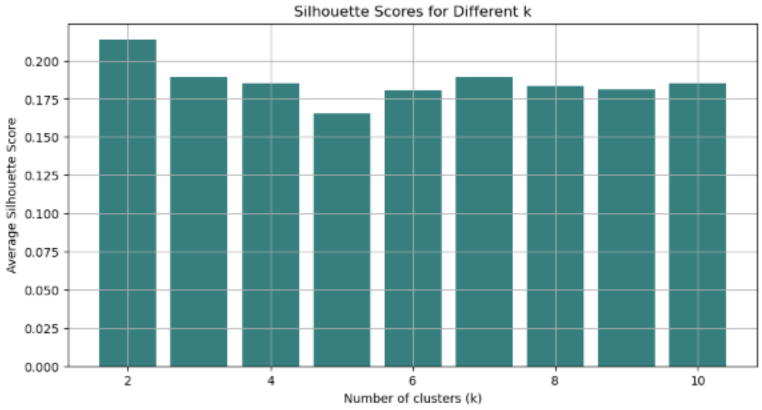
Dataset 2 -

The K-means clustering algorithm partitions the dataset into 6 clusters by minimizing the within-cluster sum of squares (WCSS). The scatter plot on the left shows how K-means assigns data points to different clusters, with each color representing a distinct cluster. K-means assumes spherical clusters due to its reliance on Euclidean distance, which may not always capture complex cluster shapes. The Elbow Method plot shows a gradual decrease in WCSS as the number of clusters increases, with a noticeable bend around k=4, indicating that 4 clusters might be optimal. However, the silhouette score for k=6k=6 is relatively low, suggesting that the clusters are not well-separated, and the highest silhouette score occurs at k=2, indicating better-defined clusters with fewer groups.

The Gaussian Mixture Model (GMM) clustering algorithm assumes that the data is generated from a mixture of Gaussian distributions and assigns probabilities to each data point belonging to a cluster. The scatter plot on the right shows GMM clustering results for 6 components, where the clusters appear more elliptical compared to K-means due to GMM's ability to model covariance structures. GMM can capture more complex shapes than K-means but may be more sensitive to initialization. While both methods were applied with 6 clusters for visualization, the evaluation through silhouette scores suggests that fewer clusters (e.g., k=2) might result in better-defined groupings. This is reflected in the silhouette score plot, where k=2 yields the highest score, indicating clearer separation between clusters compared to higher values of kk.



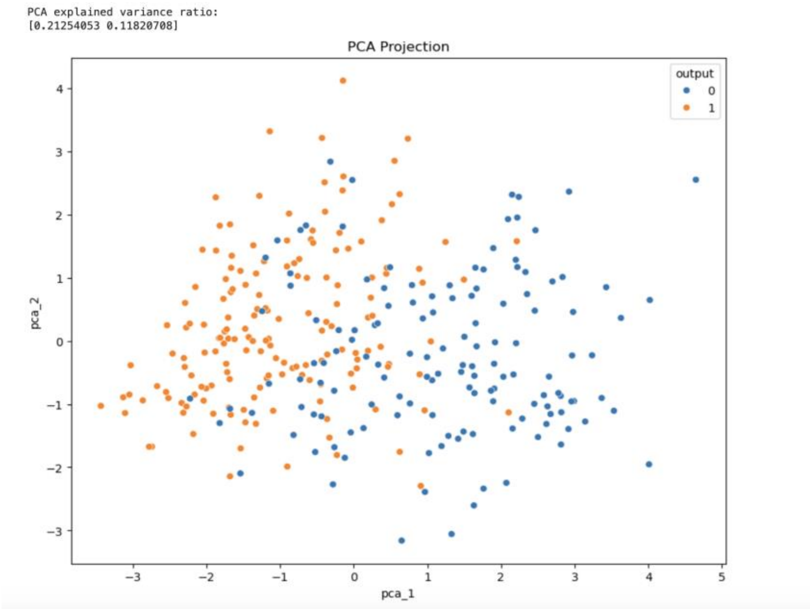




7. Dimensionality Reduction

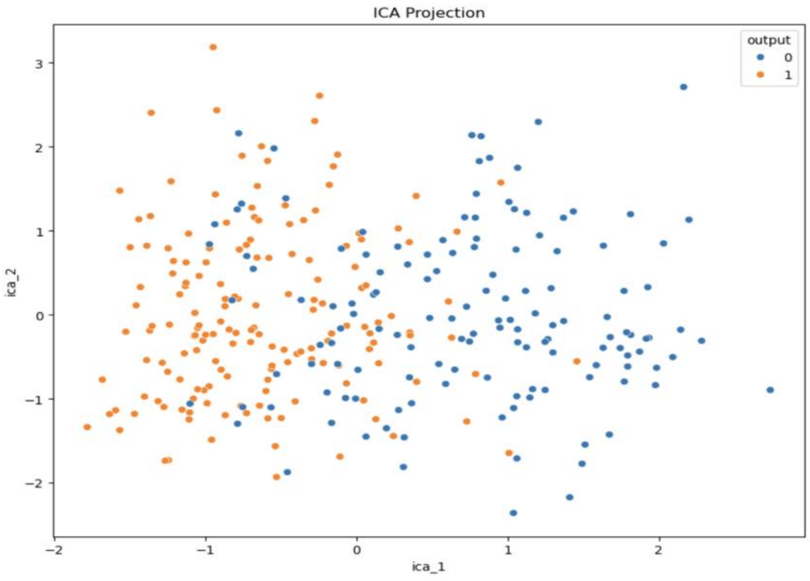
Dataset 1 -

PCA is a linear dimensionality reduction technique that transforms the data into a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first axis (principal component), the second greatest variance on the second axis, and so on. The goal is to reduce the dimensionality by projecting the data onto a smaller number of dimensions (in this case, 2), while preserving as much variance as possible. In the code, PCA is applied with n_components=2, meaning the data is projected onto two principal components. This output shows that the first principal component (pca_1) explains approximately 21.25% of the variance in the dataset, while the second component (pca_2) explains about 11.82%. Together, these two components capture around 33% of the total variance in the original dataset. The scatter plot titled "PCA Projection" visually represents this reduced 2D space, with points colored according to their class labels (output 0 or 1). The plot shows some separation between classes but also significant overlap, indicating that PCA alone may not fully separate the classes.



```
PCA explained variance ratio:
[0.21254053 0.11820708]
```

ICA is another dimensionality reduction technique but with a different goal than PCA. While PCA focuses on maximizing variance, ICA aims to find statistically independent components by minimizing mutual information between them. This makes ICA particularly useful when dealing with non-Gaussian data or when trying to separate mixed signals into independent sources. In this case, ICA is also applied with n_components=2, projecting the data onto two independent components (ica_1 and ica_2). Unlike PCA, ICA does not provide an "explained variance ratio" since it does not rely on maximizing variance but rather independence between components. The scatter plot titled "ICA Projection" shows how well ICA separates the classes in a 2D space. While there are clusters of points from different classes, there still appears to be some overlap between classes 0 and 1.

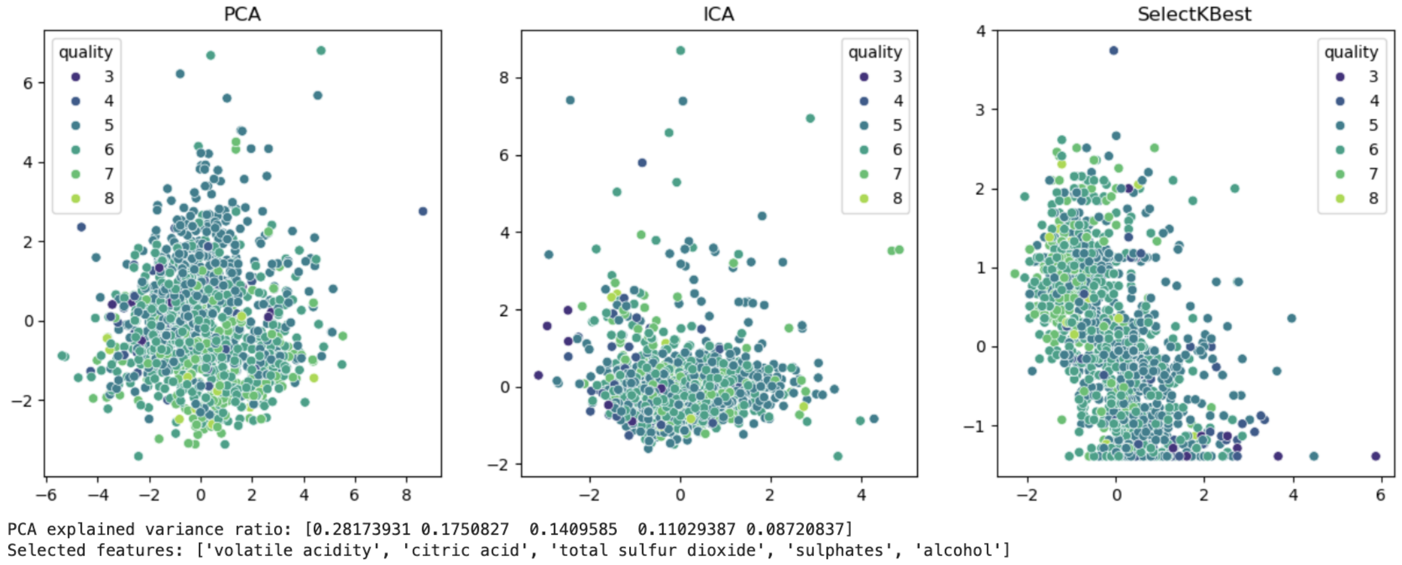


Dataset 2 -

PCA is a widely used technique for reducing the dimensionality of data while retaining as much variance as possible. In the code, PCA is applied with 5 components (n_components=5). The first two principal components are visualized in the scatter plot on the left. The PCA model outputs an explained variance ratio, which indicates how much variance each principal component captures from the original data. In this case, the first two components capture 28.17% and 17.50% of the total variance, respectively, with subsequent components capturing smaller proportions (14.10%, 11.03%, and 8.72%). The PCA plot shows that the data points are spread across a broad range in both dimensions, indicating that PCA has captured significant variance. However, it doesn't seem to clearly separate different quality classes (represented by colors), suggesting that linear combinations of features alone may not be sufficient to fully distinguish between wine qualities.

ICA is another dimensionality reduction technique but focuses on finding statistically independent components rather than maximizing variance as in PCA. In this case, ICA is also applied with 5 components (n_components=5), and the first two components are visualized in the middle plot. The ICA plot shows a more clustered structure compared to PCA, with many points concentrated near the origin. This could indicate that ICA has identified independent signals in the data but may not have captured as much spread or variance as PCA. The lack of clear separation between quality classes suggests that ICA may not be effective for this particular classification task.

SelectKBest is a feature selection method that selects features based on statistical tests. In this case, an ANOVA F-test (score_func=f_classif) is used to select the top 5 features most correlated with the target variable y (wine quality). The selected features are Volatile acidity, Citric acid, Total sulfur dioxide, Sulphates, Alcohol. The SelectKBest plot (right) shows data points projected onto two of the selected features. There is a more distinct clustering pattern compared to ICA and PCA, which might indicate that these selected features have stronger discriminative power for wine quality classification.

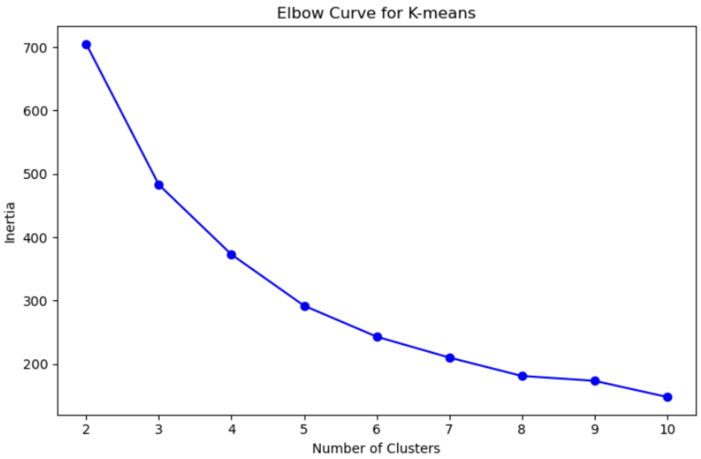


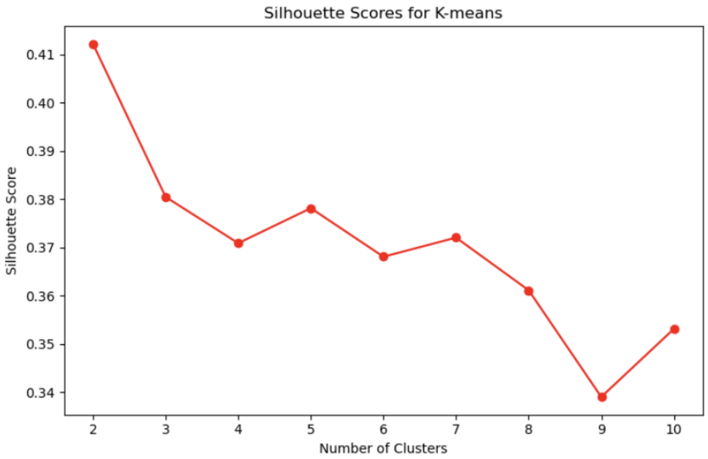
```
PCA explained variance ratio: [0.28173931 0.1750827  0.1409585  0.11029387 0.08720837]
Selected features: ['volatile acidity', 'citric acid', 'total sulfur dioxide', 'sulphates', 'alcohol']
```

8. Clustering on dimensionality reduced data

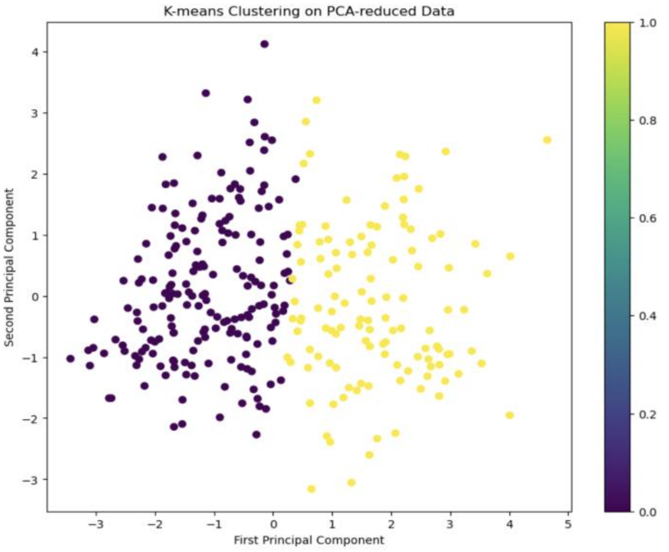
Dataset 1 -

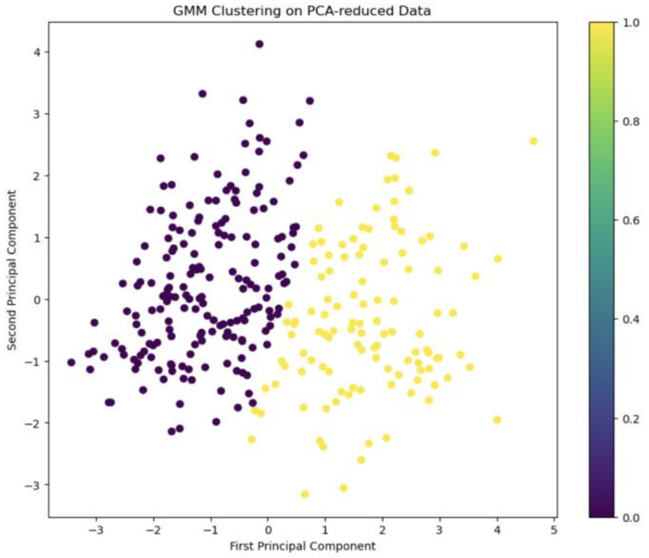
The application of K-means and Gaussian Mixture Model (GMM) clustering algorithms on data that has been reduced to two dimensions using Principal Component Analysis (PCA). The goal is to cluster the data into two distinct groups and evaluate the performance of both clustering methods. The first two images show scatter plots of the data clustered using GMM and K-means, respectively. Both plots display data points in a 2D space, where the x-axis represents the First Principal Component, and the y-axis represents the Second Principal Component. In both algorithms, two clusters are clearly visible, with one cluster primarily located around negative values of the first principal component and another around positive values.





The K-means clustering (second image) shows more rigid boundaries between clusters due to its centroid-based approach, while GMM clustering (first image) allows for softer boundaries due to its probabilistic nature. The elbow curve shows how the inertia (within-cluster sum of squares) decreases as the number of clusters increases. Inertia decreases sharply at first but begins to level off after 2 or 3 clusters, indicating that 2 clusters may be optimal. The silhouette score measures how well-separated the clusters are. A higher score indicates better-defined clusters. For 2 clusters, the silhouette score is around 0.41, which suggests moderately well-separated clusters. As the number of clusters increases beyond 2, the silhouette score decreases, further supporting that 2 is an optimal number of clusters.





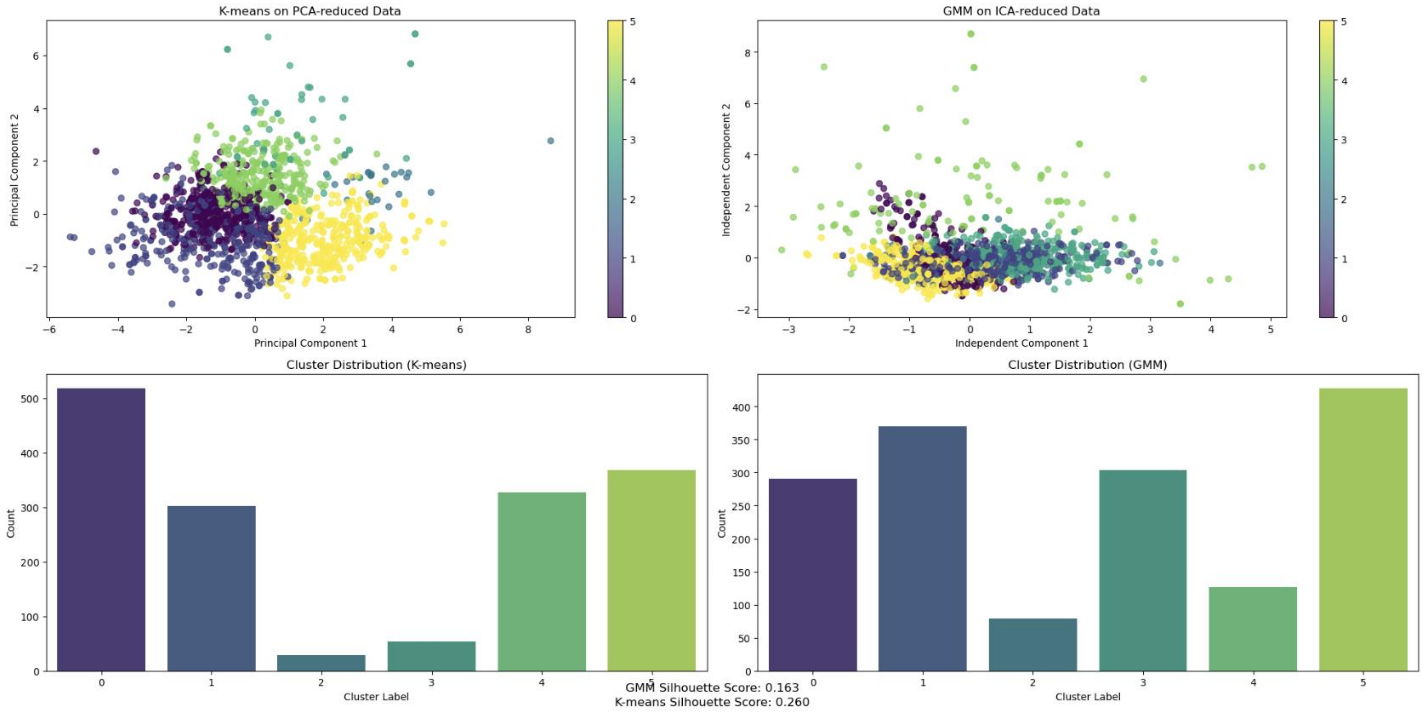
Dataset 2 -

The results of applying two clustering algorithms—K-means and Gaussian Mixture Model (GMM)—on datasets that have been reduced in dimensionality using Principal Component Analysis (PCA) and Independent Component Analysis (ICA).

The top two scatter plots show the spatial distribution of clusters generated by these algorithms, while the bottom bar charts display the distribution of data points across different clusters. In the top left plot, K-means clustering is applied to the PCA-reduced data. The scatter plot shows how data points are grouped into distinct clusters based on their principal components (PC1 and PC2). The color gradient represents different clusters, with some overlap between clusters, indicating that K-means struggles to perfectly separate the data in this reduced space. The silhouette score for K-means is 0.260, which suggests moderate cluster separation but indicates room for improvement.

In the top right plot, GMM clustering is applied to ICA-reduced data. Here, the independent components (IC1 and IC2) are used to visualize how GMM assigns points to different clusters. The clusters appear more spread out and less cohesive compared to K-means, which is reflected in GMM's lower silhouette score of 0.163. This lower score indicates weaker cluster separation and overlap between groups.

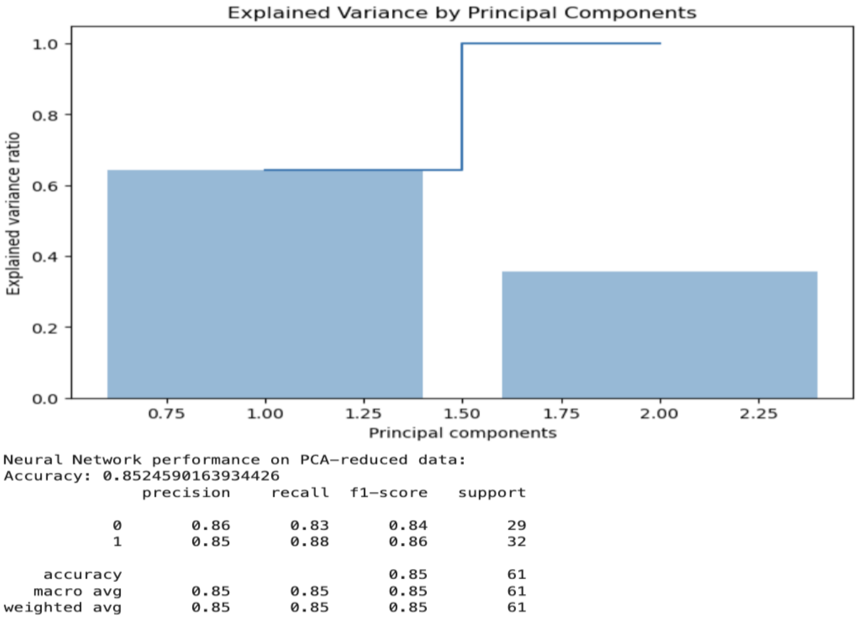
The bottom left bar chart shows the distribution of data points across different clusters for K-means. There is a significant imbalance, with one cluster containing over 500 points, while others have far fewer. This imbalance may affect the overall performance of K-means, as it suggests that some clusters dominate the dataset. The bottom right bar chart presents the cluster distribution for GMM. While there is still some imbalance, GMM shows a more even distribution across clusters compared to K-means. However, this does not necessarily translate into better performance, as evidenced by its lower silhouette score.



9. Neural Network on dimensionality reduced data

Dataset 1 -

The first plot showcases the explained variance ratio for the principal components derived from PCA. The bar chart indicates that the first principal component captures approximately 60% of the variance in the dataset, while the second principal component explains around 40%. Together, these two components account for nearly 100% of the variance, as shown by the cumulative variance line. This suggests that reducing the dataset to two dimensions retains most of the information, making it suitable for model training without significant loss of data variability.



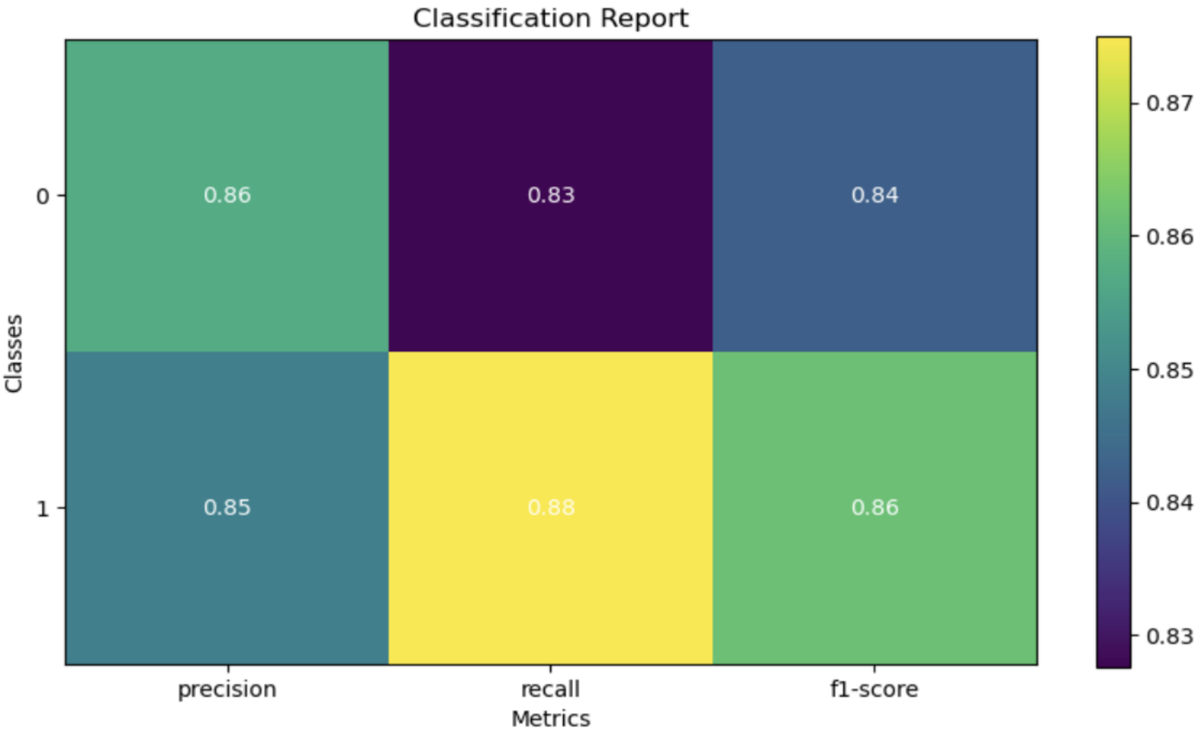
```
Neural Network performance on PCA-reduced data:
Accuracy: 0.8524590163934426
              precision    recall  f1-score   support

           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
```

The second image provides a detailed evaluation of the neural network's performance on PCA-reduced data. The neural network achieved an accuracy of approximately 85.25% on the test set. The classification report breaks down performance across two classes (0 and 1) using precision, recall, and F1-score metrics:

For Class 0: Precision = 0.86, Recall = 0.83, F1-Score = 0.84

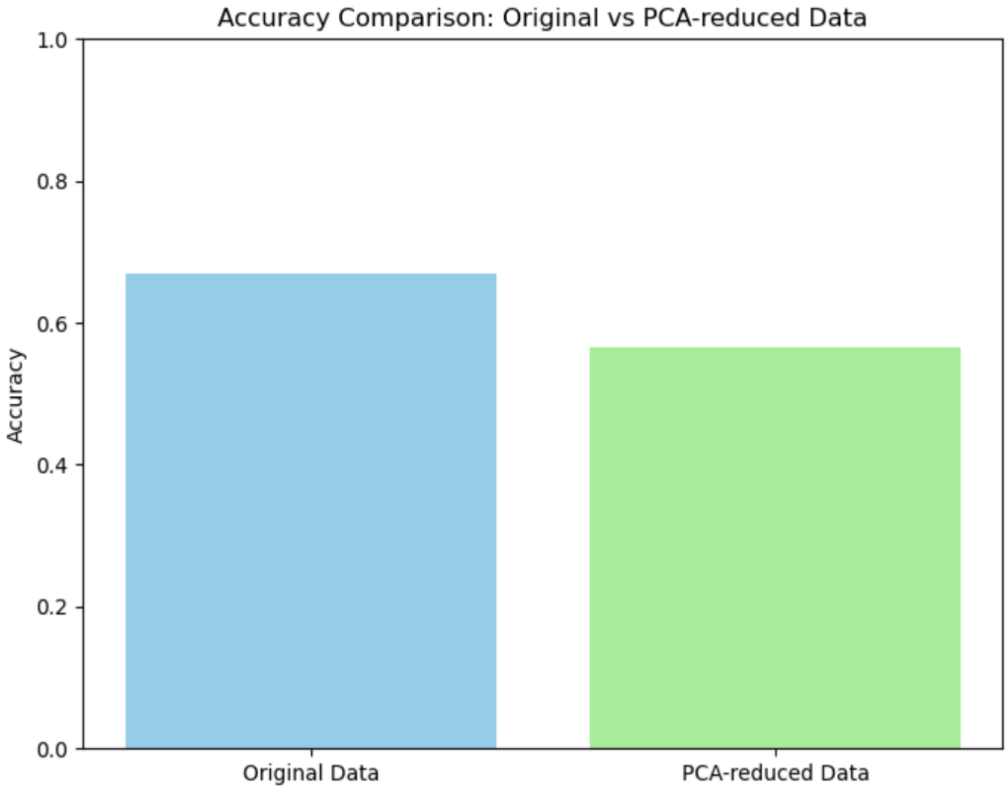
For Class 1: Precision = 0.85, Recall = 0.88, F1-Score = 0.86

These metrics indicate that the model performs slightly better in identifying Class 1 instances (higher recall), while maintaining balanced precision across both classes. The macro average and weighted average F1-scores are both 0.85, reflecting consistent performance across classes.

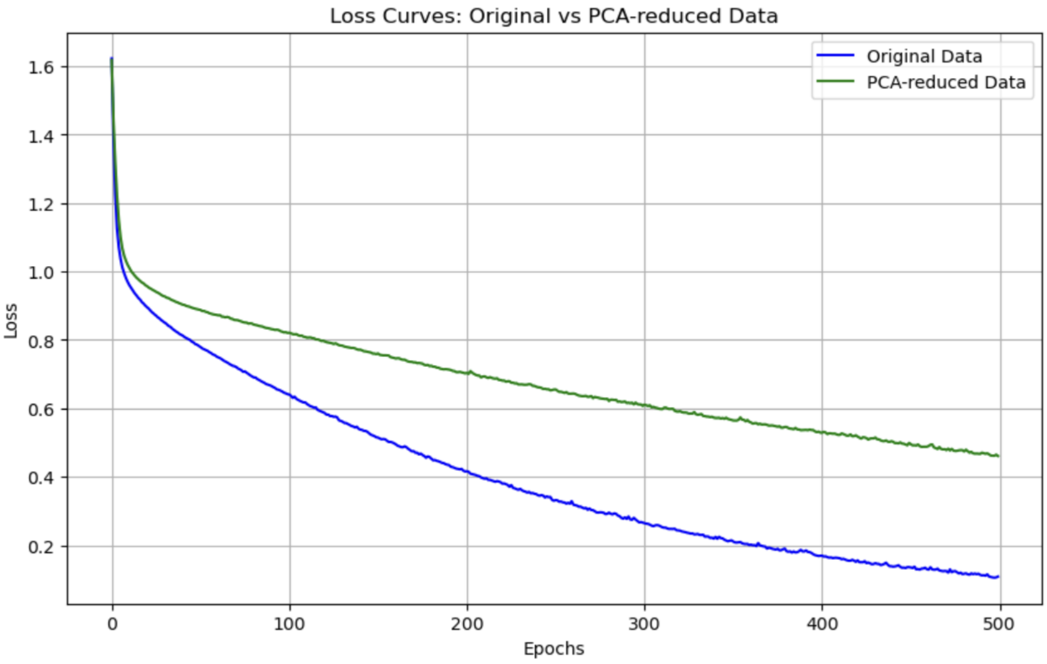


Dataset 2 -

The first image is a bar chart that compares the accuracy of the neural network models trained on the original data and PCA-reduced data. The model trained on the original data achieves an accuracy of approximately 0.65, while the model trained on PCA-reduced data has a lower accuracy of around 0.55. This suggests that while PCA reduces the dimensionality of the dataset, it may also lead to some loss of information, which negatively impacts the model's ability to classify correctly. The difference in accuracy highlights that dimensionality reduction via PCA can be a trade-off between reducing computational complexity and maintaining classification performance.



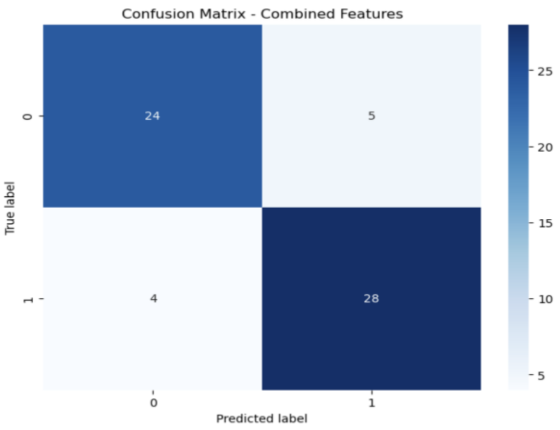
The second image shows the loss curves for both models over 500 epochs. The blue curve represents the loss for the model trained on the original data, and it demonstrates faster convergence, reaching a lower final loss value (around 0.2). In contrast, the green curve, representing the model trained on PCA-reduced data, converges more slowly and stabilizes at a higher loss value (around 0.5). This indicates that reducing dimensionality with PCA might have removed some critical information from the dataset, leading to slower learning and higher final loss during training. The lower loss for the original data model reflects its better ability to fit to the training data, while the higher loss for the PCA-reduced model suggests that it struggles more with optimization.

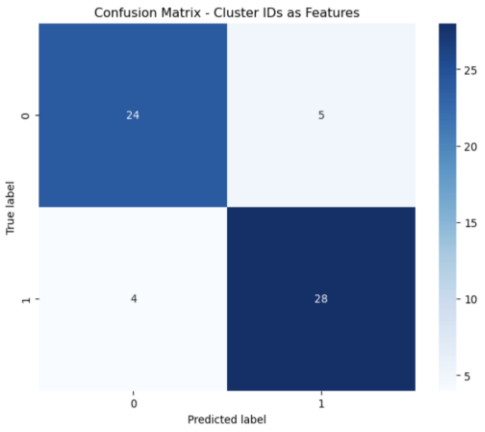


10. Neural Network with clustering as features

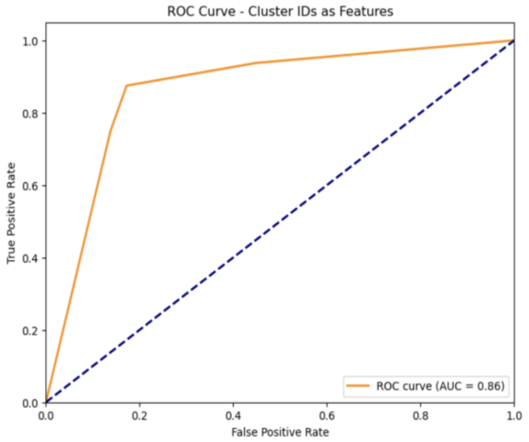
Dataset 1 -

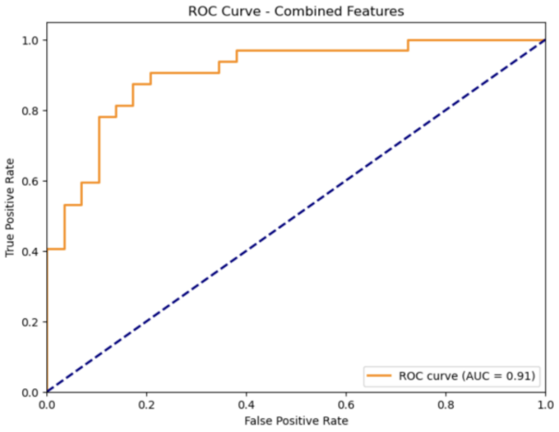
The confusion matrices for both models (using cluster IDs as features and using combined features) show similar results. In both cases, the model correctly classified 24 instances as negative (true negatives) and 28 instances as positive (true positives). However, there were 5 false positives and 4 false negatives in both models. This consistency suggests that both models have comparable performance in terms of classification accuracy. The confusion matrices provide a clear visual representation of the model's ability to differentiate between classes.



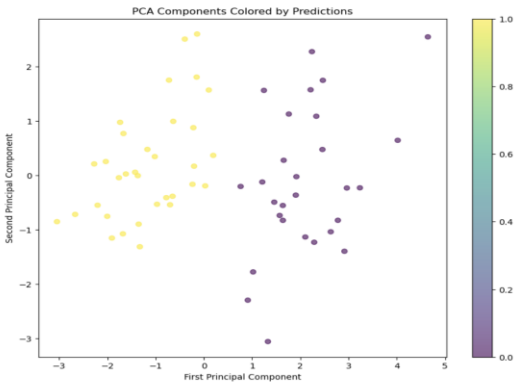


The ROC curve for the model using only cluster IDs as features shows an AUC (Area Under the Curve) of 0.86, indicating a relatively strong ability to distinguish between positive and negative classes. However, when PCA components are added to the feature set (combined features), the AUC increases to 0.91, demonstrating improved performance. The ROC curve visually represents the trade-off between true positive rate (sensitivity) and false positive rate, with a higher AUC indicating better classification performance. The increase in AUC when using combined features suggests that adding dimensionality reduction techniques (like PCA) enhances the model's predictive capabilities.





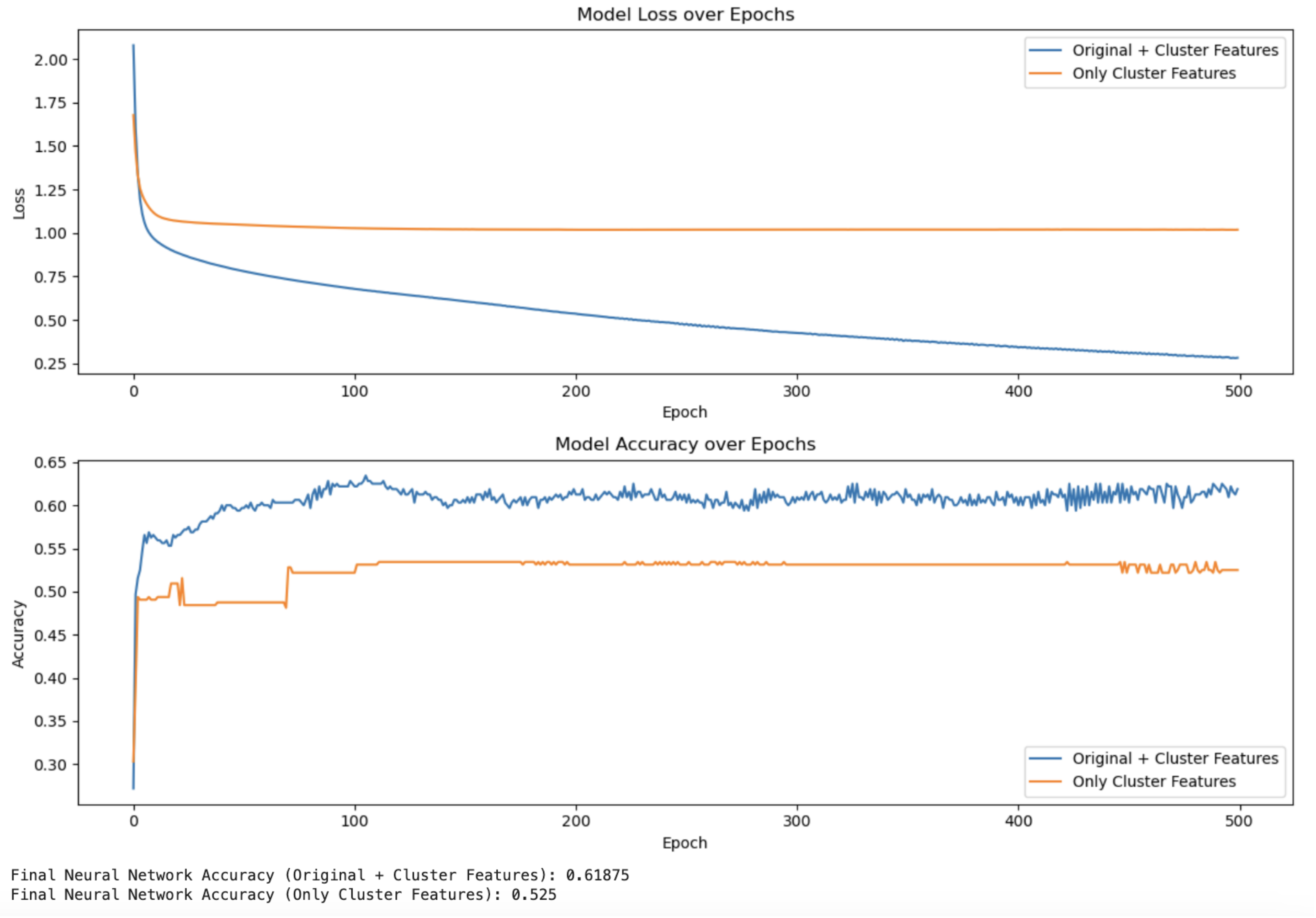
The scatter plot shows the first two principal components (PCA) of the dataset, with points colored based on the model's predictions. The yellow and purple points represent two different predicted classes. The plot demonstrates that the PCA transformation effectively separates the data into distinct clusters, which aids in classification. However, some overlap between classes is still visible, indicating that while PCA helps reduce dimensionality and visualize separability, it may not completely resolve class overlap.



Dataset 2 -

Both models use a Multi-Layer Perceptron (MLP) classifier implemented via MLPClassifier from scikit-learn. The architecture consists of two hidden layers with 100 and 50 neurons, respectively. The models are trained over 500 epochs using a warm-start approach, which allows the model to retain its learned weights between epochs, comparing their loss and accuracy.

The first model, trained on both the original features and cluster labels (from KMeans and Gaussian Mixture Models), shows a significant reduction in loss over time, as evidenced by the blue curve in the "Model Loss over Epochs" plot. This model's loss decreases consistently, indicating effective learning throughout training. Conversely, the second model, which is trained solely on cluster labels, exhibits a much slower decrease in loss (orange curve), plateauing early in training. This suggests that the model struggles to improve beyond a certain point due to the limited information provided by only the cluster labels.



```
Final Neural Network Accuracy (Original + Cluster Features): 0.61875
Final Neural Network Accuracy (Only Cluster Features): 0.525
```

In terms of accuracy, as shown in the "Model Accuracy over Epochs" plot, the model with both original and cluster features achieves higher accuracy, stabilizing around 0.62. This demonstrates that including original features alongside clustering information helps capture more relevant patterns for classification. On the other hand, the model trained only on cluster labels reaches a lower accuracy of approximately 0.52 and shows little improvement after initial epochs.

The final reported accuracies are 0.61875 for the model with original + cluster features and 0.525 for the model with only cluster features. These results highlight that while clustering labels can provide some useful information, they are insufficient on their own for optimal classification performance compared to when they are combined with the original feature set.

11. References

- https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset
- https://labelyourdata.com/articles/machine-learning-for-wine-quality-prediction#
- https://archive.ics.uci.edu/
- ChatGPT
- https://towardsdatascience.com/clustering-out-of-the-black-box-5e8285220717
- https://stats.stackexchange.com/questions/489459/
- https://stackoverflow.com/questions/34234209/machine-learning-clustering-algorithms-k-means-and-gaussian-mixtures