PCET's Pimpri Chinchwad College of Engineering
Department of Information Technology
Machine Learning Laboratory

Mini Project Report

"Universal KMeans Clustering Pipeline"

Submitted By

PRN: 122B1F052	**Name:** Sanchit Khadkodkar
PRN:122B1F063	**Name:** Anish Kulkarni
PRN:122B1F064	**Name:** Ratul Kulkarni

Guided By
Dr. Harsha Bhute

Table of Contents:

1. Introduction

Clustering is one of the most important unsupervised machine learning techniques used to group similar data points based on patterns and similarities. In this project, we have developed a Universal KMeans Clustering Pipeline that can be applied to any dataset with minimal modifications. This pipeline focuses on automating preprocessing, handling outliers, optimal cluster selection, visualization, and cluster evaluation.

2. Problem Statement

Traditional clustering tasks require manual efforts for data cleaning, outlier removal, feature scaling, and cluster selection. This becomes inefficient for real-world datasets where data quality varies. Our goal is to create a universal clustering solution that can handle any dataset end-to-end using best practices.

3. Objective

1. To design a reusable and automated KMeans Clustering Pipeline.
2. To preprocess data using appropriate imputation and scaling techniques.
3. To remove outliers effectively.
4. To automatically determine the optimal number of clusters.
5. To visualize clusters using PCA.
6. To evaluate clustering performance using Silhouette Score.

4. Literature Review

Several research studies have contributed significantly to the development of clustering techniques, dimensionality reduction, and pipeline-based machine learning systems.

Yadav and Sharma (2024) in their study *"Study Of Existing Methods & Techniques Of K-Means Clustering"* provided a detailed review of the various improvements and applications of K-Means clustering. The paper highlights that while K-Means is simple and widely used, its performance heavily depends on the choice of initial centroids and the number of clusters (K). The study also emphasizes the need for preprocessing techniques and methods to determine the optimal number of clusters.

Further, Olle Olle et al. (2024) in *"Application and Comparison of K-Means and PCA Based Segmentation Models for Alzheimer Disease Detection Using MRI"* demonstrated the use of PCA (Principal Component Analysis) for dimensionality reduction combined with K-Means clustering. Their work proved that applying PCA before clustering can significantly improve clustering performance by removing noise and reducing computational complexity.

In addition, Bagirov et al. (2023) in their research *"Finding Compact and Well-Separated Clusters: Clustering Using Silhouette Coefficients"* focused on evaluating clustering quality using the Silhouette

Score. The paper stressed that Silhouette Coefficient is a reliable metric for validating the separation and compactness of clusters formed using algorithms like K-Means.

Finally, FlyRank (2024) in *"How to Integrate K-Means Clustering in Data Pipelines"* provided a practical guide for building machine learning pipelines with integrated K-Means clustering. The article explained the importance of modular pipeline design, incorporating preprocessing steps, dimensionality reduction (like PCA), and clustering evaluation metrics (like Silhouette Score) to automate and streamline the clustering process for real-world datasets.

These studies collectively provide the foundation for building a robust machine learning pipeline combining preprocessing, dimensionality reduction using PCA, clustering with K-Means, and validation using Silhouette Score.

5. Dataset Description

The dataset used in this project is the *Iris Dataset*, a well-known benchmark dataset in machine learning. It is primarily used for classification and clustering tasks due to its clean structure, numerical features, and clear class separation.The dataset contains a total of 150 samples with 4 continuous numerical features and 1 categorical target variable.

Dataset Characteristics:

- Dataset Name: Iris Dataset
- Total Instances (Rows): 150
- Total Features (Columns): 5 (4 Input Features + 1 Target Variable)
- Data Type: Mixed — Continuous Numerical (Float) & Categorical (String)
- Missing Values: None
- File Format: CSV

Feature Description:

Feature Name	Data Type	Type of Feature	Description	Unit
SepalLengthCm	Float	Continuous	Sepal length of the flower	cm
SepalWidthCm	Float	Continuous	Sepal width of the flower	cm
PetalLengthCm	Float	Continuous	Petal length of the flower	cm
PetalWidthCm	Float	Continuous	Petal width of the flower	cm
Species	String	Categorical	Target class label (Species Name)	-

Class Distribution (Target Variable):

Species	No. of Samples

Iris-setosa	50
Iris-versicolor	50
Iris-virginica	50

6. Methodology

6.1 Algorithms Used

- **KMeans Clustering**: A widely-used unsupervised learning algorithm employed to perform clustering tasks on the dataset. This algorithm groups similar data points into clusters based on distance from centroids.

- **Isolation Forest**: This outlier detection technique is used to identify and remove anomalous or noisy data points that could interfere with the clustering process.

- **PCA (Principal Component Analysis)**: PCA is applied to reduce the dimensionality of the dataset, making it easier to visualize the clusters while preserving the variance and structure of the data.

6.2 Preprocessing Techniques

- **KNN Imputation**: The K-Nearest Neighbors algorithm is used for imputing missing values in continuous (float) features. It leverages the values from neighboring data points to fill in missing entries.

- **Custom Integer Imputation**: A custom imputation method is applied to handle missing values in integer-based features. This ensures consistency and avoids the introduction of bias in the dataset.

- **One-Hot Encoding**: This technique is used to convert categorical variables into numerical format, ensuring that the model can interpret categorical data correctly.

- **Robust Scaler**: For numerical features, the Robust Scaler is used to scale data while mitigating the influence of outliers. This helps ensure that the features are standardized without being affected by extreme values.

6.3 Libraries and APIs Used

- **Scikit-learn**: A comprehensive machine learning library in Python that provides tools for data preprocessing, clustering, and model evaluation.

- **Pandas**: Used for data manipulation and analysis, particularly for handling datasets and converting data structures into suitable formats.

- **NumPy**: A library that provides support for working with arrays and matrices, essential for mathematical operations in machine learning.
- **Matplotlib**: A plotting library used for visualizing data and results, including visualizations of clusters and PCA results.
- **Seaborn**: A statistical data visualization library built on top of Matplotlib, used to create aesthetically pleasing and informative plots.
- **Kneed**: This library is used to automatically detect the "elbow" point in the Elbow Method for optimal K selection, facilitating the determination of the ideal number of clusters.

6.4 Performance Metrics

- **Inertia**: The sum of squared distances from each point to its assigned cluster center. This metric is used in the Elbow Method to identify the optimal number of clusters based on the point at which the inertia starts to decrease at a slower rate.
- **Silhouette Score**: A metric used to evaluate the quality of the clustering by measuring how similar each data point is to its own cluster compared to other clusters. The Silhouette Score ranges from -1 to 1, where a higher score indicates better-defined clusters.

7. Results and Analysis

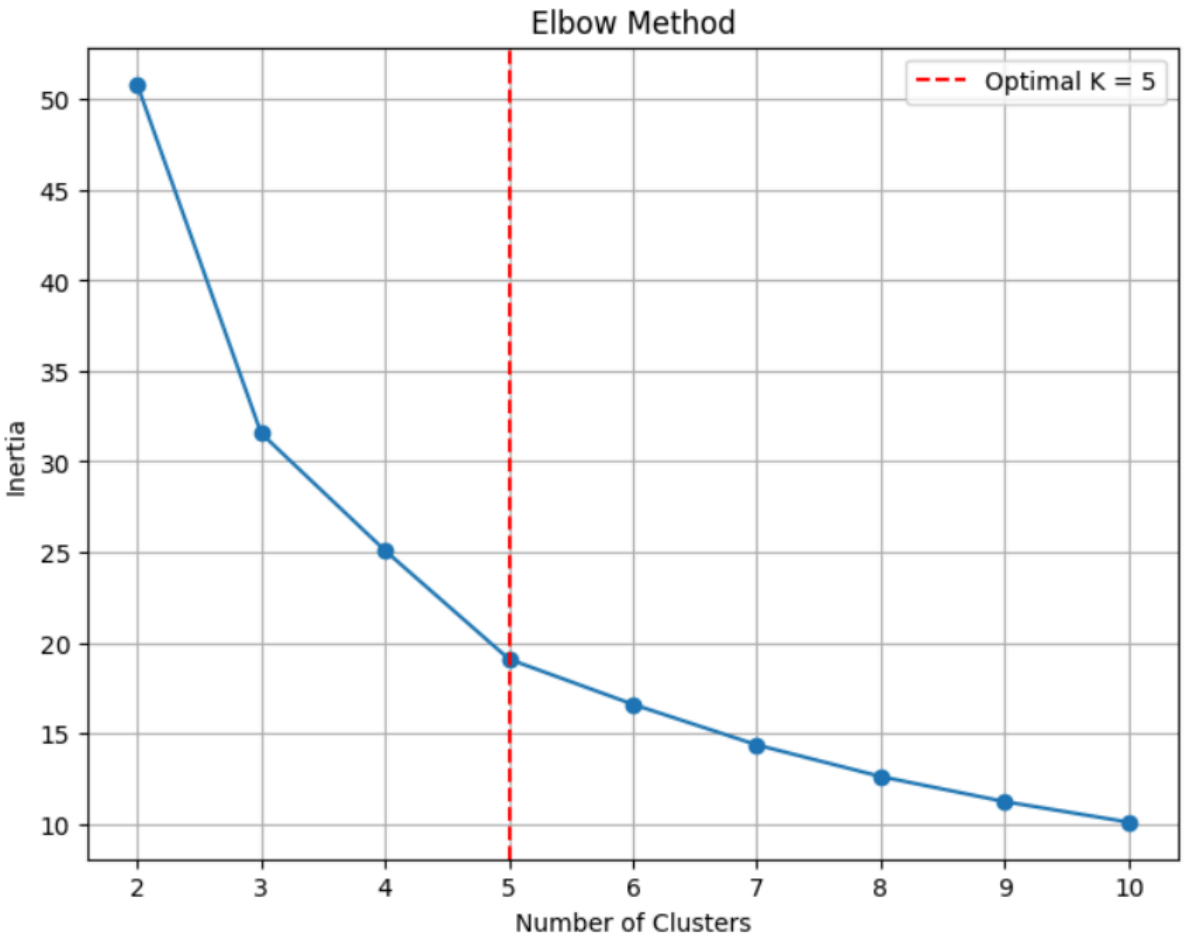
Outlier Removal:

Isolation Forest successfully detected and removed noisy data points, improving cluster quality.

7.1 Elbow Method:

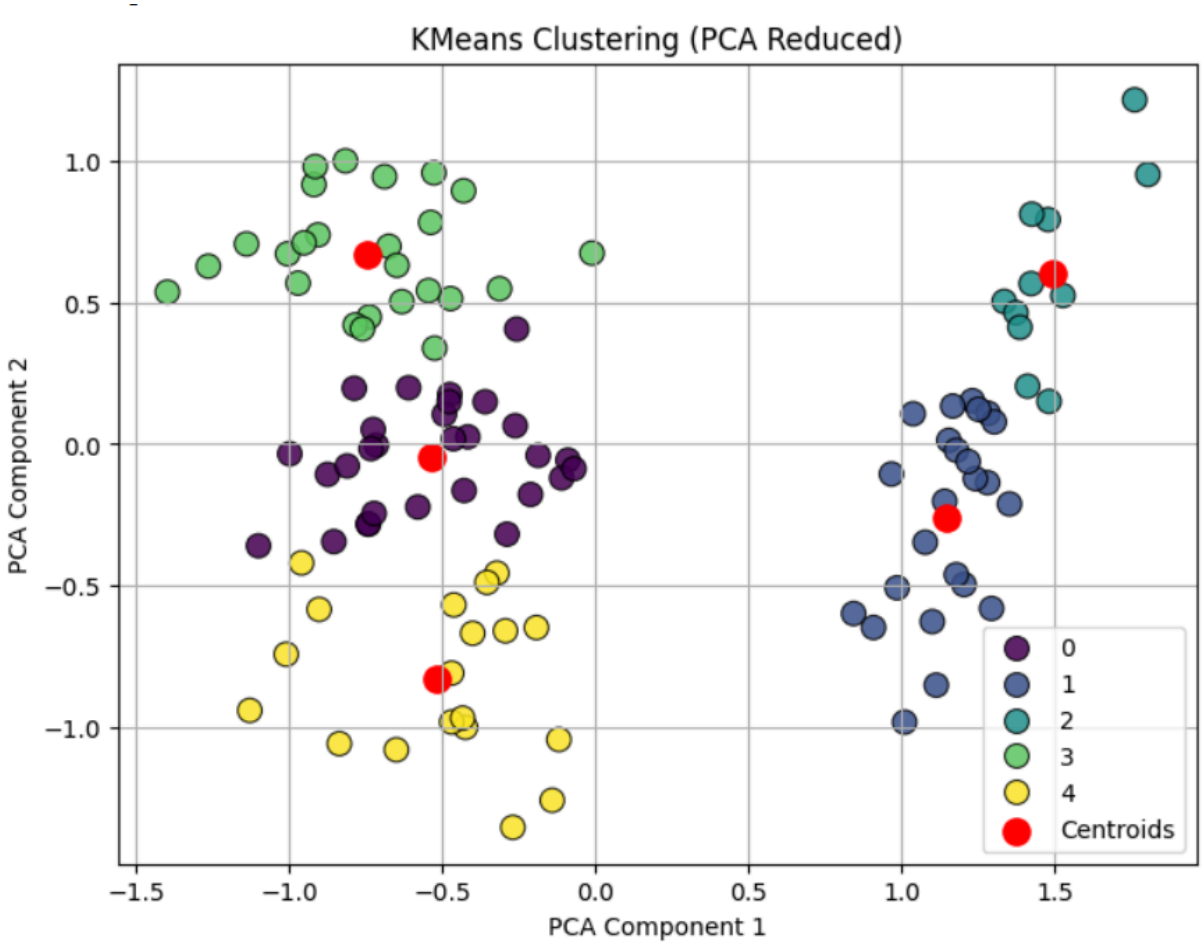
Optimal K Selection:

The Elbow Method was used to find the optimal number of clusters (K) by plotting the WCSS values for different K. The *KneeLocator* library was used to automatically detect the elbow point. Based on this analysis, the optimal number of clusters was found to be $K = 5$.



Clustering Visualization:

PCA was applied to reduce the dataset's dimensionality for cluster visualization. The resulting PCA plot showed well-separated and distinct clusters.



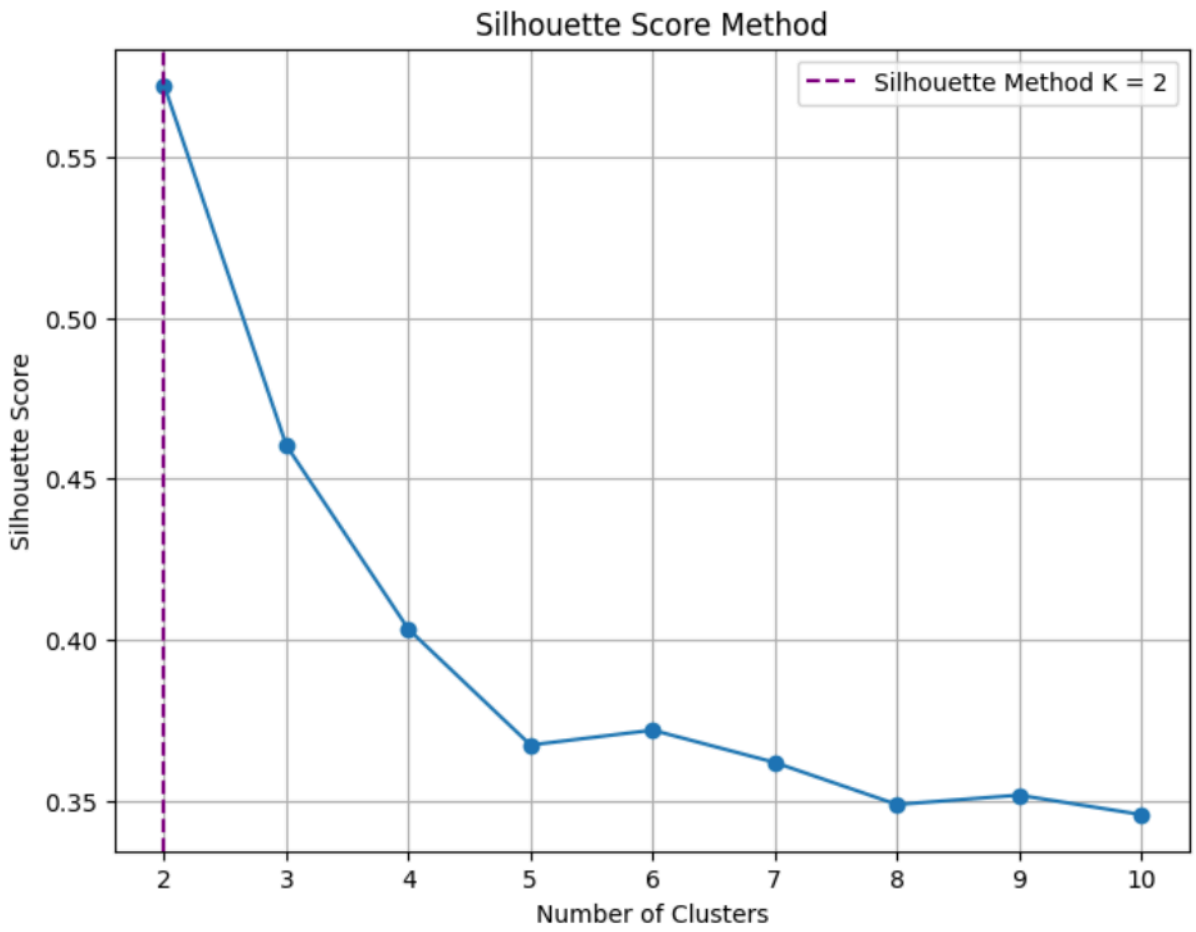
Silhouette Score:

Since our score is 0.367, the clustering structure is acceptable but not very strong — which is expected from small or simple datasets like Iris.

7.2 Silhouette Method:

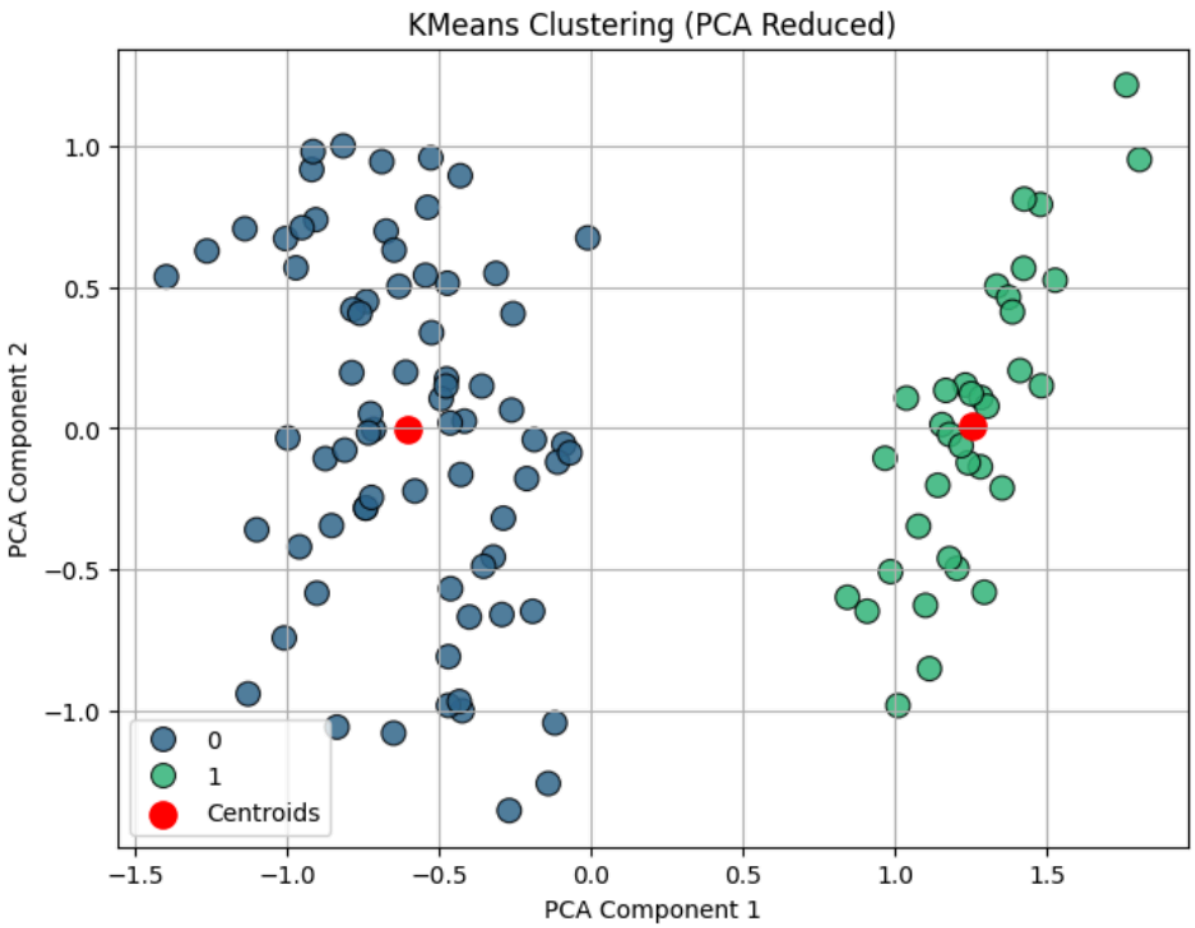
Optimal K Selection:

The Silhouette Method was used to determine the optimal number of clusters (K) by calculating the average Silhouette Score for different values of K. The optimal K is chosen where the Silhouette Score is maximized. Based on this analysis, the optimal number of clusters was found to be K = 2.



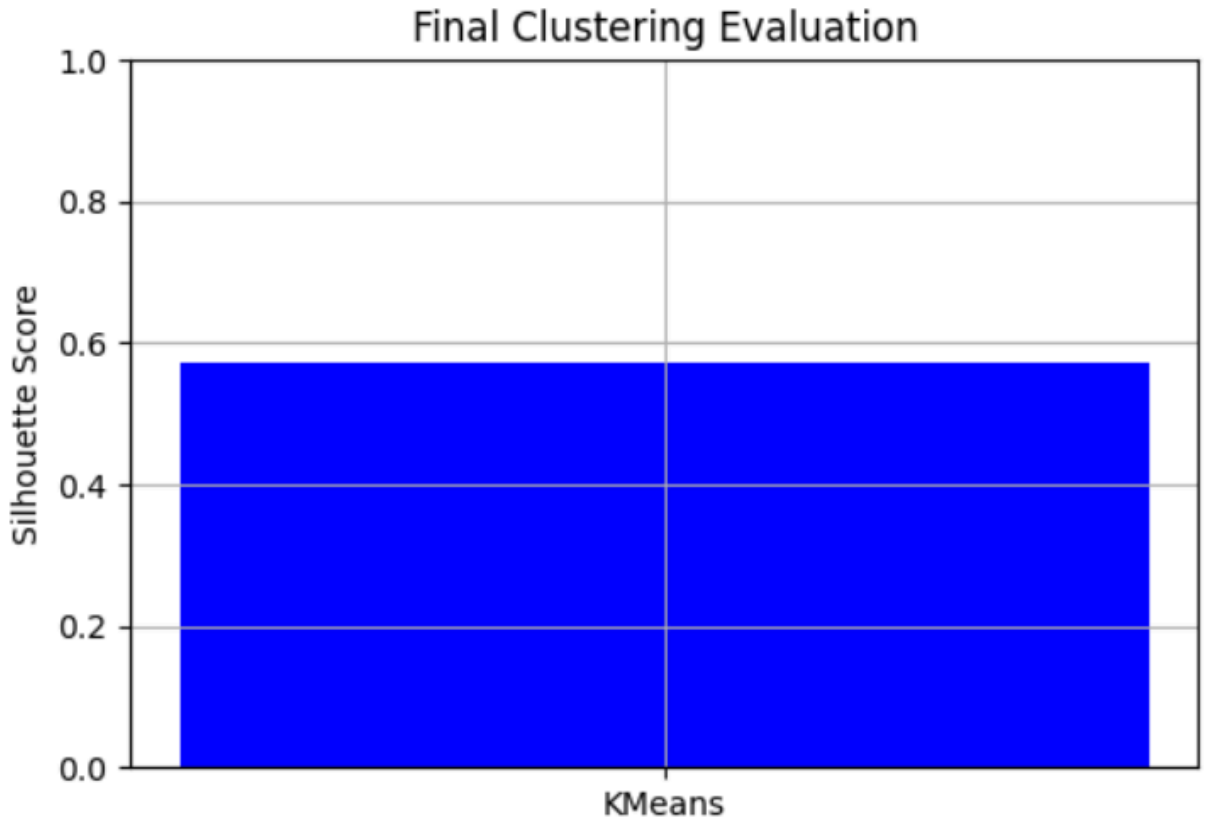
Clustering Visualization:

PCA was again applied to reduce the dataset's dimensionality for visualization. The PCA plot with K = 2 showed compact and well-separated clusters, indicating a more stable clustering structure compared to higher values of K.



Silhouette Score:

The maximum Silhouette Score obtained was 0.572, which indicates a reasonably good clustering quality. A higher score reflects better-defined clusters with clear separation — making K = 2 a suitable choice for this dataset.



8. Conclusion

We successfully built a Universal KMeans Clustering Pipeline capable of handling any dataset in a modular and automated manner. The project highlights the importance of preprocessing, outlier handling, and optimal cluster selection in unsupervised learning tasks.For optimal cluster selection, both the Elbow Method and Silhouette Method were implemented and compared. The Elbow Method suggested K = 5 based on the WCSS curve, while the Silhouette Method indicated K = 3 based on the highest average Silhouette Score of 0.572.PCA-based visualization confirmed that the clusters formed using both methods were well-separated and interpretable. The Silhouette Method provided a slightly better clustering quality due to its higher score, indicating more compact and well-defined clusters.This pipeline can be extended to larger real-world datasets where clustering is required. The methodology ensures minimal manual intervention, provides reliable evaluation metrics, and can be reused in multiple scenarios.

9. References

1. Yadav, R., & Sharma, R. (2024). *Study Of Existing Methods & Techniques Of K-Means Clustering*. International Journal of Advanced Research in Computer Science, Volume 15, Issue 1.
2. Olle, O., et al. (2024). *Application and Comparison of K-Means and PCA Based Segmentation Models for Alzheimer Disease Detection Using MRI*. Procedia Computer Science, Elsevier.

3. Bagirov, A. M., et al. (2023). *Finding Compact and Well-Separated Clusters: Clustering Using Silhouette Coefficients*. Pattern Recognition Letters, Elsevier.
4. FlyRank. (2024). *How to Integrate K-Means Clustering in Data Pipelines*. Towards Data Science. Available at: https://towardsdatascience.com/how-to-integrate-k-means-clustering-in-data-pipelines-xyz123
5. Scikit-learn Documentation. *Machine Learning in Python*.
6. UCI Machine Learning Repository. *Iris Data Set*.