Implementation of the K-means Algorithm to a given dataset

Team YOLO - s19 (205, 355, 408, 420)

1. Introduction

Clustering is an essential technique in machine learning that enables the grouping of similar data points without prior knowledge of labels. This project focuses on implementing the k-means clustering algorithm to categorize iris flowers into three clusters based on their attributes. The aim is to analyze the effectiveness of k-means in classifying the iris dataset and explore methods to improve clustering performance.

2. Definitions

Notion of Similarity

Similarity is a measure of how close or related data points are in a given feature space. It is usually determined using distance metrics such as Euclidean distance, cosine similarity, or Manhattan distance. In clustering, similar data points are grouped together based on their proximity in the feature space.

Clustering

Clustering is an unsupervised machine-learning technique that groups data into clusters based on common traits or features. Unlike supervised learning, clustering does not use labeled data instead, it seeks to discover inherent structures in the dataset.

Types of Clustering

1. **Exclusive Clustering -** Each data point belongs to only one cluster. This type of clustering is also known as hard clustering, where a data point is assigned strictly to one group, such as in k-means clustering.
2. **Non-Exclusive Clustering -** Data points can belong to multiple clusters simultaneously. This is also called soft clustering or fuzzy clustering, as seen in techniques like Fuzzy C-Means (FCM).
3. **Hierarchical Clustering -** Clusters are organized in a tree-like structure. This approach can be divided into -
    - **Agglomerative (Bottom-Up) -** Starts with individual points and merges them iteratively.
    - **Divisive (Top-Down) -** Starts with a single cluster and recursively splits it into smaller ones.

4. **Partitional Clustering -** The dataset is divided into distinct clusters without any hierarchical structure. K-means is a common example of partitional clustering.

Properties of Clustering

- **Density-Based Clustering -** Forms clusters based on dense regions of data, making it useful for detecting arbitrarily shaped clusters and noise, as in DBSCAN.
- **Centroid-Based Clustering -** Uses central points to define clusters. K-means falls under this category as it optimizes the cluster centers iteratively.
- **Connectivity-Based Clustering -** Groups points based on their connectivity in a graph. Hierarchical clustering methods often use this approach.

Implementation Approaches

Clustering implementations can be categorized into different approaches as follows,

- **Agglomerative Clustering -** A bottom-up approach where each data point starts as its cluster, and clusters are merged iteratively.
- **Divisive Clustering -** A top-down approach where all data points start in one cluster and are progressively split.
- **Serial Clustering -** Clusters are built sequentially, processing data one point at a time.
- **Simultaneous Clustering -** All data points are processed in parallel to determine cluster assignments.
- **Graph-Theoretic Clustering -** Uses graph structures where nodes represent data points and edges represent similarities.
- **Algebraic Clustering -** Uses mathematical transformations, such as Singular Value Decomposition (SVD), to discover cluster structures.

The Sum of Squared Errors

K-means clustering uses the *Sum of Squared Errors (SSE)* as a convergence criterion. SSE measures the total variance within clusters and is computed using the formula,

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

where,

- yi is the actual value of the dependent variable for each of the n data points of the independent variable.
- ŷ is the predicted value of the dependent variable for each of the n data points of the independent variable.

Graphically, this formula can be displayed as,

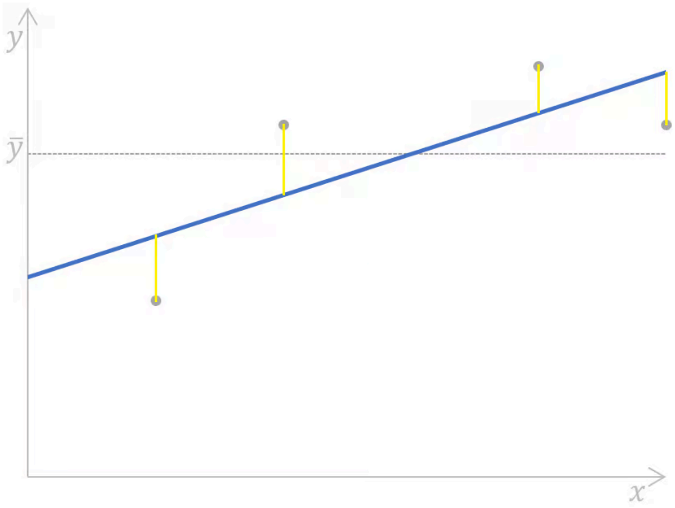


Figure 1 : Graphical representation of the SSE

Here, the distances (the yellow segments on the plot) for each pair of actual-predicted values of the dependent variable are measured, squared, and summed up. In regression analysis, minimizing the SSE is key to improving the accuracy of the regression model.

3. Project Steps

1. Identification of the K-Means Clustering and its properties
2. Problem Identification & Project Planning
3. Data Preparation with Iris dataset (iris.txt)
4. K- Mean Algorithm Development and optimizing
5. Testing with the dataset
6. Report Writing and Conclusion

4. Methodology

Implementation of K-Mean

- Data Preprocessing : Loaded the dataset (iris.txt), selected the 4 numerical features (sepal length, sepal width, petal length, petal width) and scaled them using StandardScaler to normalize the feature ranges.
- Choosing K : K is selected by understanding the dataset. (k=3)
- K-Means Clustering :
    - Applied K-Means using k = 3 clusters with random_state=42 and n_init=10 to improve initialization.
    - Cluster assignments were added to the data frame.
- Visualization :
    - Reduced dimensions from 4D to 2D using PCA for better visualization.

  - Plotted the clustered data points in 2D colored by cluster labels.
- Validation : Calculated the Silhouette Score to evaluate the clustering quality.

Convergence Criterion

In our implementation, K-Means stops iterating when either,

- The centroids stop changing significantly (centroid positions stabilize).
- Or after reaching the maximum number of iterations (default is usually 300 in sklearn).

Since we used scikit-learn's KMeans, the convergence criterion is based on the tolerance value (tol parameter), which checks if the movement of centroids between iterations is less than a small threshold (default tol=1e-4).

Handling Noise in Clustering

K-Means does not handle noise/outliers well since it tries to assign every data point to a cluster. However, in our case,

- The Iris dataset is clean and does not have significant noise or outliers, so standard K-Means works reasonably well.

5. Results

1. **K = 3 & with all 4 features.**

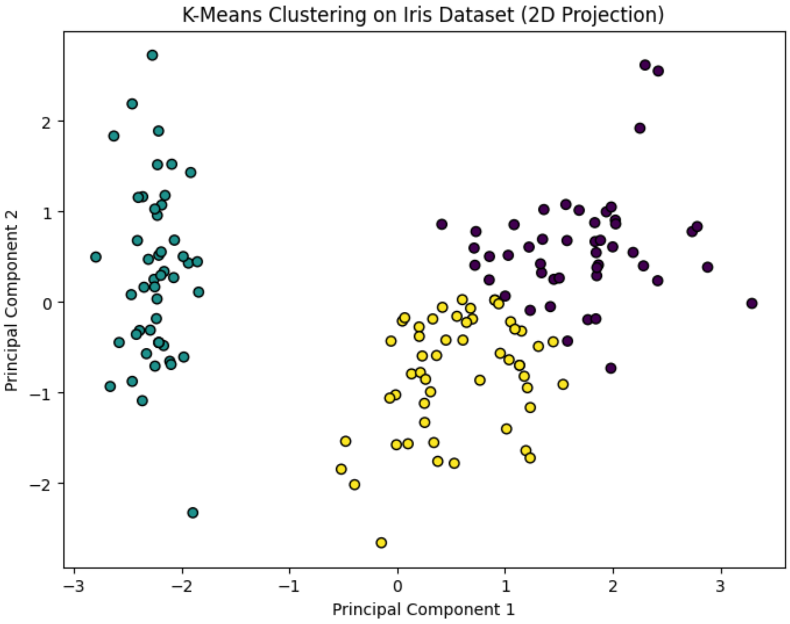


Figure 2 : K-Means Clustering when K=3 & with all features

In this scenario, K-Means clustering was applied to the Iris dataset using k=3, which matches the known number of classes. All four numerical features were used, and after standardizing the data, the clustering yielded an accuracy of 83.2%. The 2D PCA projection plot [Figure 2] shows three distinct clusters, though some overlap is visible, especially between two clusters on the right-hand side. This overlap suggests that while K-Means captures the general structure of the data, certain classes are not perfectly separable in feature space using K-Means, which is expected due to their natural similarity.

2. Find the K value using the Elbow method & perform clustering with all 4 features

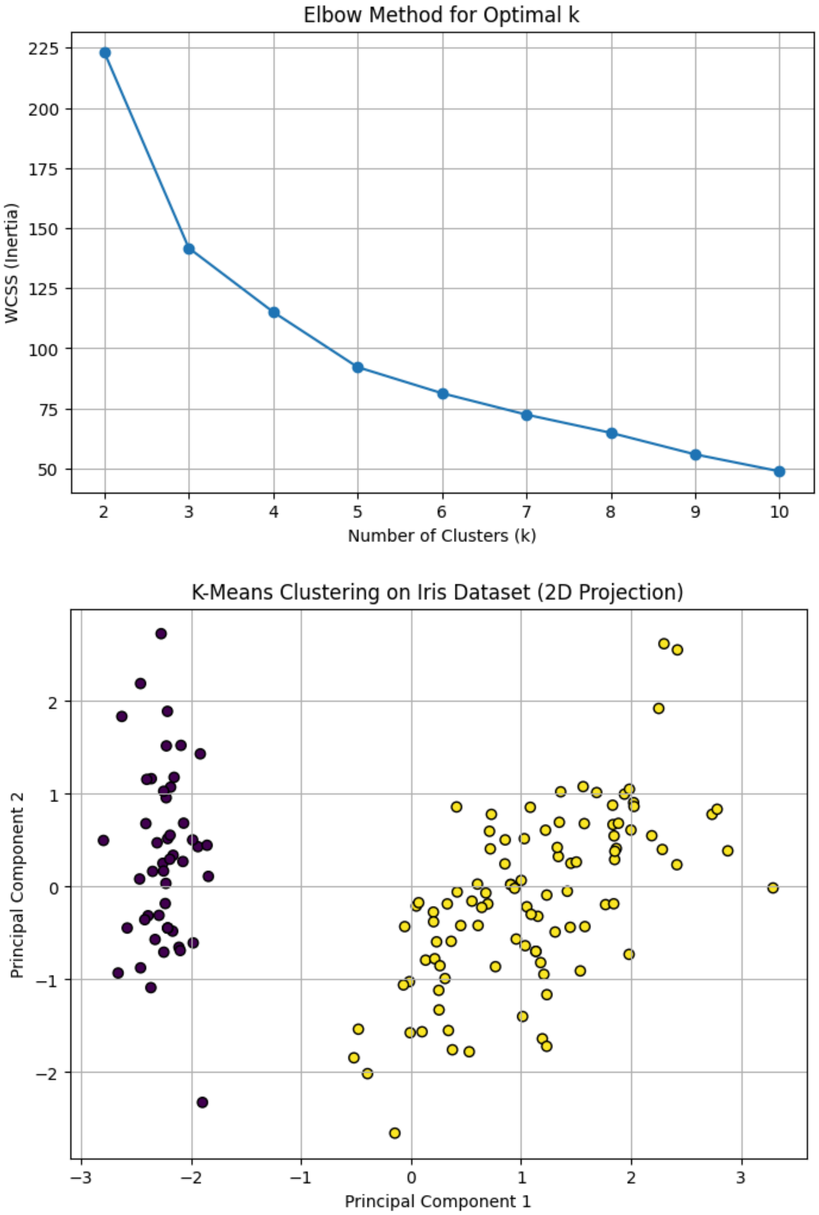
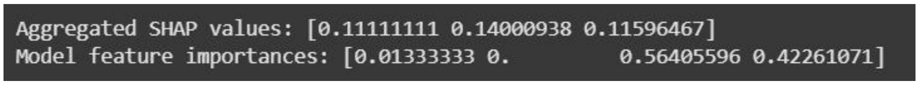


Figure 3 : Finding the K value itself & clustering with all features

In the second scenario, the program automatically selected the number of clusters using the Elbow Method and Silhouette Score. It chose k = 2, even though we know there are actually 3 types of Iris flowers. Because of this, the clustering accuracy (66.4%) was lower compared to the first scenario. The 2D plot shows that the data points were divided into two main groups, but some points that should belong to a third group were mixed into these two clusters. This shows that while the automatic method tried to find the best number of clusters based on the data, it couldn’t fully capture the real structure of the dataset. It also highlights that K-Means might struggle when two groups look very similar in their features.

3. Find features importance using SHAP



```
Aggregated SHAP values: [0.11111111 0.14000938 0.11596467]
Model feature importances: [0.01333333 0.         0.56405596 0.42261071]
```

Figure 4 : SHAP results

SHAP (SHapley Additive exPlanations) is an explainable AI framework used to assess feature importance in a dataset. When applied to the Iris dataset, only three features are assigned significant importance, with the remaining feature having minimal effect on the model's predictions. Specifically, using model feature importance, we can identify that the second feature, Sepal Width, has little to no contribution to the model’s outcome.

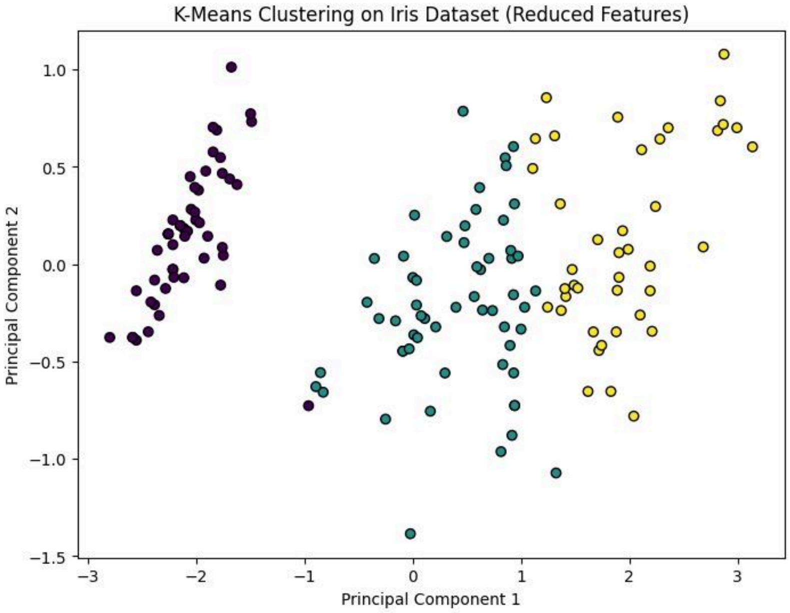


Figure 5 : Without 2nd feature (Sepal Width)

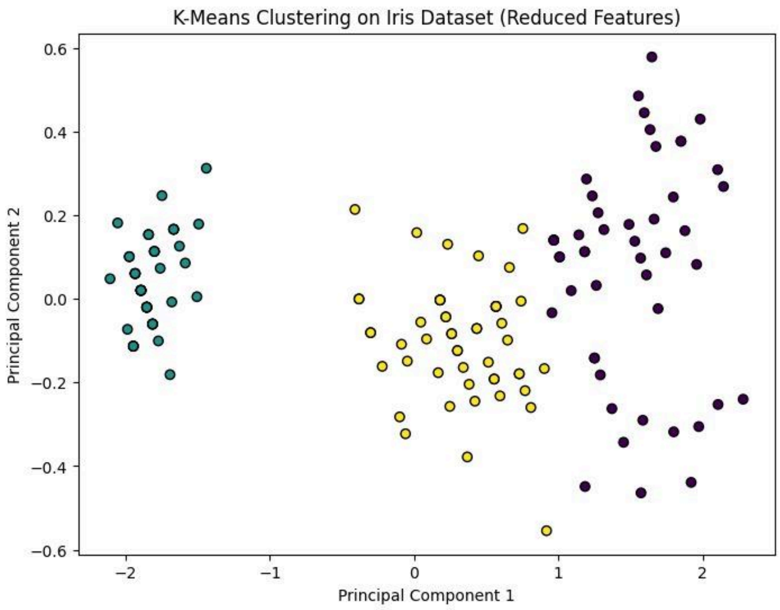


Figure 6 : Without 1st and 2nd features

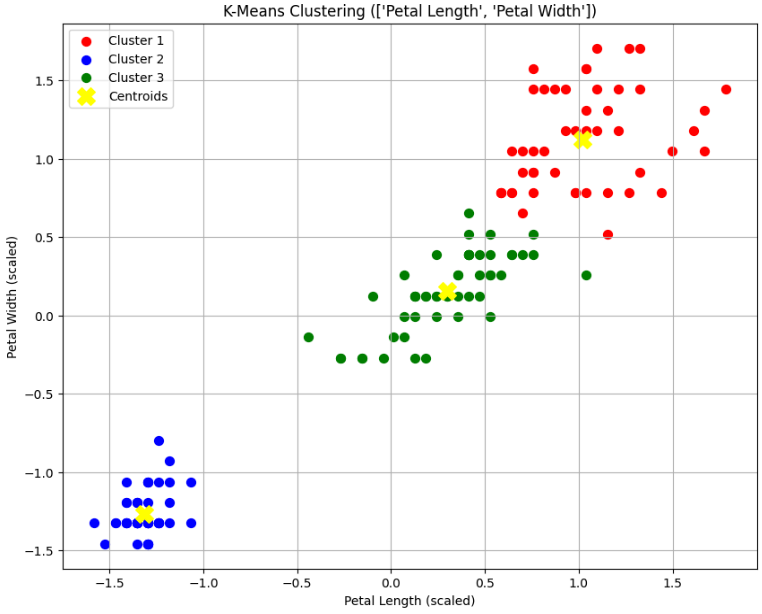


Figure 7 : With Petal Width and Petal Length

The clustering results on the Iris dataset reveal that using only Petal Length and Petal Width yields the highest performance, achieving an accuracy of 95.97% and an Adjusted Rand Index (ARI) of 0.8842, with very few misclassifications, as these two features are highly effective at separating the species. In contrast, including Sepal Length or Sepal Width alongside petal features reduces clustering quality, with accuracies of 86.58% and 85.91%, and ARIs of 0.6692 and 0.6508, respectively, due to the added noise and feature overlap.

Using all four features further degrades performance, resulting in the lowest accuracy of 83.22% and ARI of 0.6150, as Sepal Length and Sepal Width introduce variability that hampers cluster separation. Therefore, Petal Length and Petal Width are the optimal feature combinations for KMeans clustering on this dataset. To improve clustering with all features, applying dimensionality reduction techniques like PCA or experimenting with models such as Gaussian Mixture Models (GMM) is recommended.

6. Task Breakdown

The project was divided into two main tasks,

Task	Team Members
Algorithm Implementation & Testing	S19408, S19420
Project Reporting & Documentation	S19205, S19355

7. Conclusion

The analysis demonstrates that Petal Length and Petal Width are the most significant features for clustering the Iris dataset using K-Means, achieving the highest accuracy and ARI. The Elbow Method suggested k = 2, but the best results were obtained with k = 3, matching the actual number of iris species. SHAP analysis further confirmed that Sepal Width contributes the least to clustering performance. Including all four features reduces accuracy due to feature overlap and added noise. Thus, focusing on petal-based features leads to more effective and accurate clustering. To further enhance clustering when using all features, advanced techniques like PCA or Gaussian Mixture Models (GMM) could be applied.

References

- Jain, A. K. (2010). "Data Clustering: 50 Years Beyond K-Means."
- Iris Dataset (UCI Machine Learning Repository)
- DataCamp. (n.d.). "Regression Sum of Squares." Retrieved from https://www.datacamp.com/tutorial/regression-sum-of-squares