K Means Clustering Report

1. Introduction

K-Means clustering is a popular unsupervised machine learning algorithm used for partitioning a dataset into distinct groups based on similarities. It is widely used in various domains such as customer segmentation, image compression, and pattern recognition. This report outlines the implementation and evaluation of K-Means clustering for different numbers of clusters (K = 2, 4, 6, 7) using a custom Python implementation.

---

2. Methodology

2.1 Algorithm Implementation

The K-Means algorithm was implemented with the following steps:

1. **Initialization:** Randomly select K data points as initial centroids.
2. **Cluster Assignment:** Assign each data point to the nearest centroid based on the Euclidean distance.
3. **Centroid Update:** Calculate the mean of all data points in each cluster to update the centroids.
4. **Convergence Check:** Repeat steps 2 and 3 until the centroids do not change significantly or a maximum number of iterations is reached.

2.2 Data

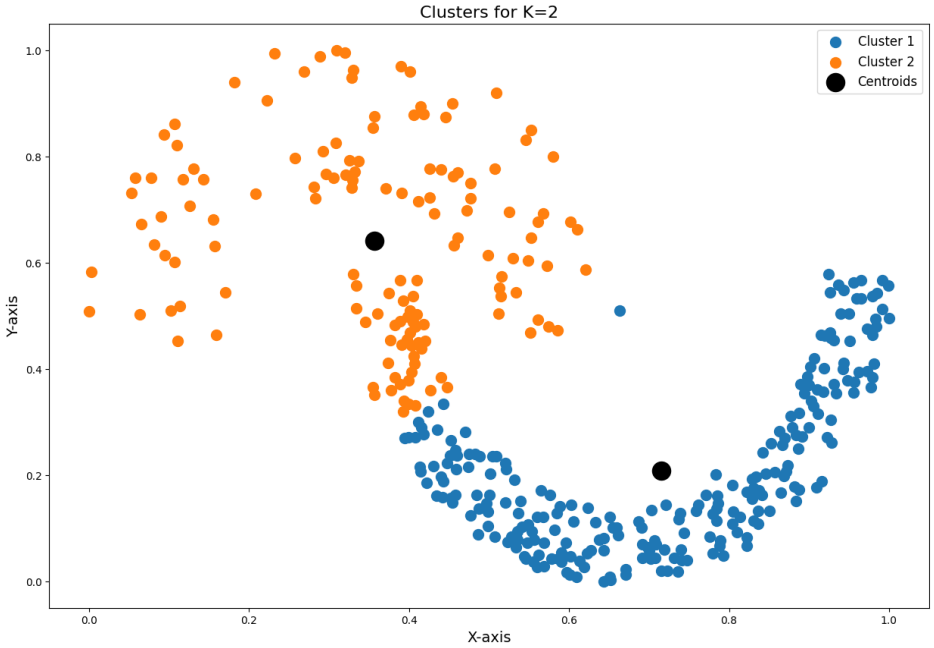
The dataset used for this analysis was loaded from a text file (dataset.txt) and processed into a 2D array format. Each row represents a data point, and each column corresponds to a feature.

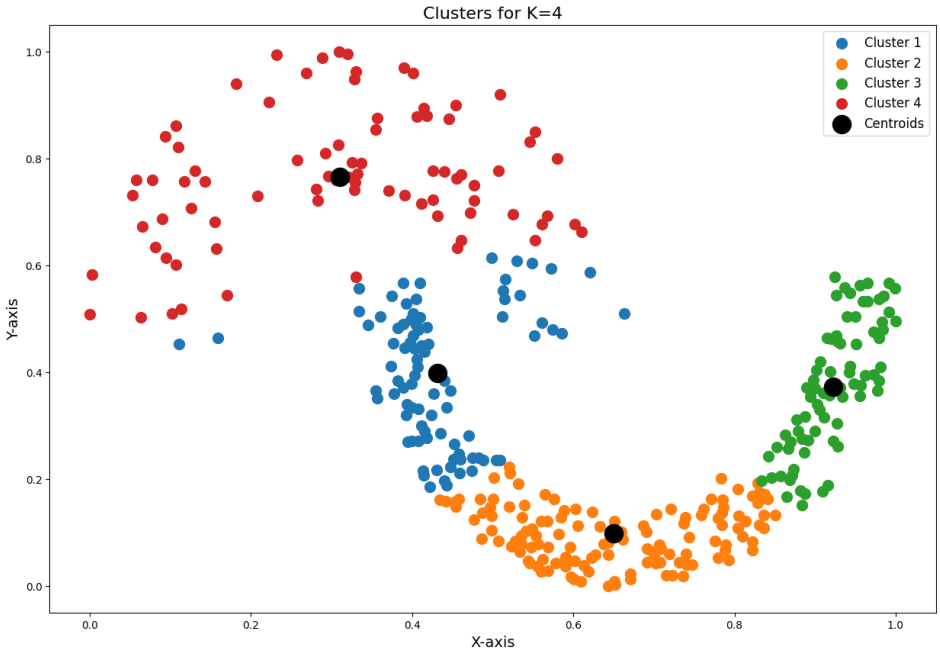
---

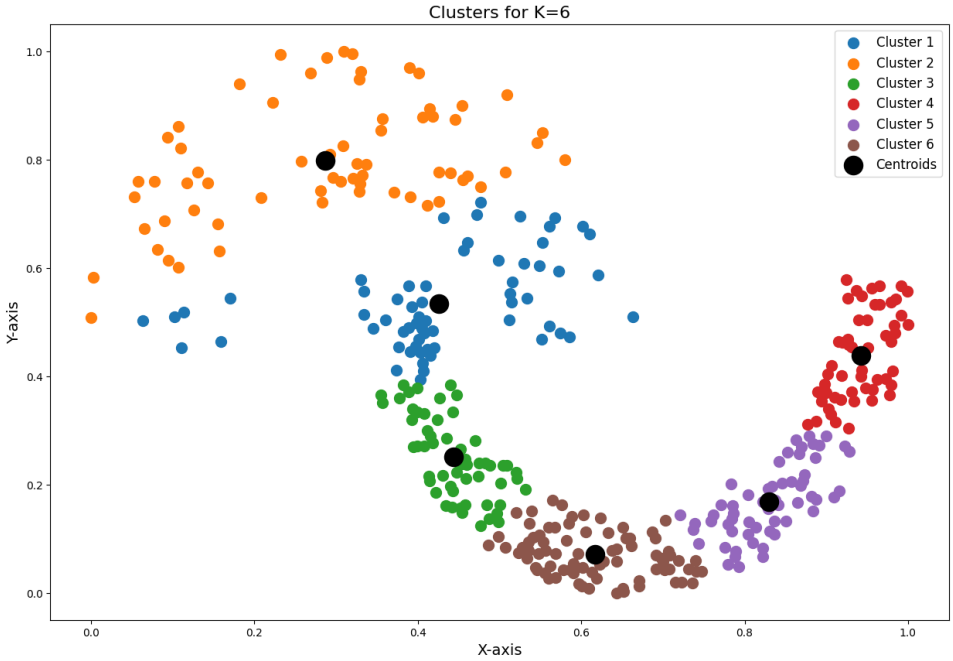
3. Results

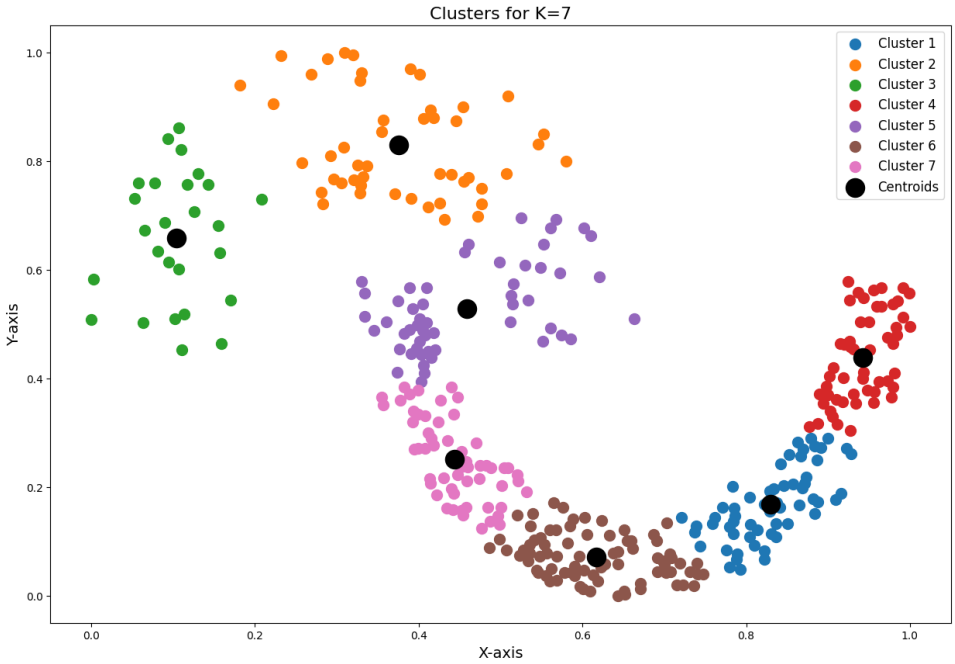
3.1 Visualization and Inertia

The algorithm was tested with K = 2, 4, 6, and 7. The clusters were visualized, and inertia (sum of squared distances between data points and their respective centroids) was calculated for each value of K.






Clusters for K=6
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Cluster 6
Centroids
Y-axis
X-axis
0.0
0.2
0.4
0.6
0.8
1.0


Clusters for K=7
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Cluster 6
Cluster 7
Centroids
Y-axis
X-axis
0.0
0.2
0.4
0.6
0.8
1.0

Results Summary

K	Inertia
2	**27.2246573429586**
4	**23.15086895183665**
6	**6.91467302652043**
7	**4.30498867876740**

3.2 Observations

- **K = 2:** The dataset was divided into two broad clusters, resulting in a higher inertia due to fewer centroids.
- **K = 4:** A more balanced division of clusters was observed, with significantly reduced inertia.
- **K = 6:** Further refinement in clustering, with distinct groups forming around local patterns in the data.
- **K = 7:** The smallest clusters began to emerge, with diminishing returns in reducing inertia.

---

4. Conclusion

The K-Means algorithm effectively segmented the dataset into meaningful clusters. The inertia values and visualizations for different K values provide insights into the optimal number of clusters for this dataset. While increasing K reduces inertia, the choice of K should balance complexity and interpretability.