Lab Experiment-2

K- Means Clustering

CS3103: Machine Learning
Department of Computer Science and Engineering

August 6, 2025

Experiment Name

Implement K-Means clustering by creating your own K-Means class

Aim

Implement the K-Means clustering algorithm to identify intrinsic groupings in the given dataset based on similar `status_type` behavior, where `status_type` represents different types of posts such as videos, photos, statuses, and links. Evaluate the model by different metrics.

Platform / Tools Used

- Python 3.x
- Jupyter Notebook / Google Colab / VS Code
- Libraries: `numpy`, `pandas`, `matplotlib`, `scikit-learn`, `seaborn`

Introduction

K-Means Clustering is an unsupervised machine learning algorithm that partitions a dataset into K distinct, non-overlapping clusters. Each cluster is represented by a central point called the **centroid**, which is the mean of all data points within that cluster.

The objective of K-Means is to minimize the **within-cluster sum of squares (WCSS)**, also known as **inertia**. Mathematically, the objective function is:

$$J = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{1}$$

Where:

- K is the number of clusters,
- C_i is the set of points belonging to cluster i,
- μ_i is the centroid of cluster C_i,
- $\|x - \mu_i\|^2$ is the squared Euclidean distance between point x and the centroid μ_i.

The algorithm proceeds iteratively with the following steps:

1. Initialize K centroids randomly.
2. Assign each data point to the nearest centroid based on Euclidean distance.
3. Recalculate the centroids as the mean of all points assigned to that cluster:

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x \tag{2}$$

4. Repeat steps 2 and 3 until convergence (i.e., the centroids do not change significantly).

K-Means is a widely used clustering method for discovering patterns or groupings in unlabeled data, with applications in market segmentation, document classification, and image compression.

Dataset Description

The dataset can be accessed from the following URL:

`https://archive.ics.uci.edu/ml/datasets/Facebook+Live+Sellers+in+Thailand`

This dataset consists of Facebook page data from 10 Thai fashion and cosmetics retail sellers. It includes various attributes related to post interactions and engagement. The `status_type` variable represents posts of different natures such as **videos**, **photos**, **statuses**, and **links**. Furthermore, the data set contains engagement metrics such as the number of **comments**, **shares**, and **reactions** for each post.

Tasks to be Performed

1. Check the shape of the dataset.
2. View a summary of the dataset.
3. Check for missing values in the dataset.
4. Drop redundant columns.
5. View the statistical summary of numerical variables.
6. Explore the `status_id` variable:
   - View the labels in the variable.
   - Count how many different types of values are present.
7. Explore the `status_published` variable:
   - View the labels in the variable.
   - Count how many different types of values are present.
8. Explore the `status_type` variable:
   - View the labels in the variable.
   - Count how many different types of values are present.
9. Drop the `status_id` and `status_published` variables from the dataset.
10. Declare the feature vector and target variable.
11. Convert categorical variables into integers.
12. Apply feature scaling using `MinMaxScaler()`.
13. Train your K-Means model with two clusters.
14. Evaluate the quality of the model's weak classification.
15. Use the elbow method to find the optimal number of clusters(show Graph).
16. Analyze your K-Means model with different numbers of clusters.
17. Visualize the different number of cluster.

18. Evaluate the performance of your model using different metrics.
19. Show how varying the number of clusters impacts the model's evaluation metrics in K-means.
20. Compare your model's performance with the `sklearn` K-Means model.

Expected Output

- Define your own K- mean Class.
- Train your K-Means model with two clusters..
- Elbow method to find the optimal number of clusters.
- Analyze your K-Means model with different numbers of clusters.
- Performance analysis of K - Means model.

Conclusion

The K-Means clustering experiment effectively demonstrated how the algorithm can be used to identify natural groupings within unlabeled data. By leveraging the concept of centroids, the model grouped similar data points based on their feature similarity. The experiment provided insights into the underlying patterns of the dataset, particularly in relation to the `status_type` variable. Additionally, the implementation of a custom K-Means algorithm reinforced understanding of the clustering process and its iterative refinement steps. The results showed that K-Means is a simple yet powerful technique for exploring the structure of complex datasets.

References

- `https://stanford.edu/~cpiech/cs221/handouts/kmeans.html`
- `https://neptune.ai/blog/k-means-clustering`
- `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html`
- Python Documentation - `https://www.w3schools.com/python/python_ml_getting_started.asp`