Name:-Abhinav Jaiswal

Roll No:-27 Div:-TY3(B)

expt 4 DWM

Aim: Implementation of Clustering Algorithm (K-means / Agglomerative) using Python

Introduction:
Clustering is an unsupervised machine learning technique used to group similar data points. K-means is a popular clustering algorithm that partitions data into K clusters based on feature similarity. This experiment demonstrates K-means clustering using Python to classify data into four clusters.

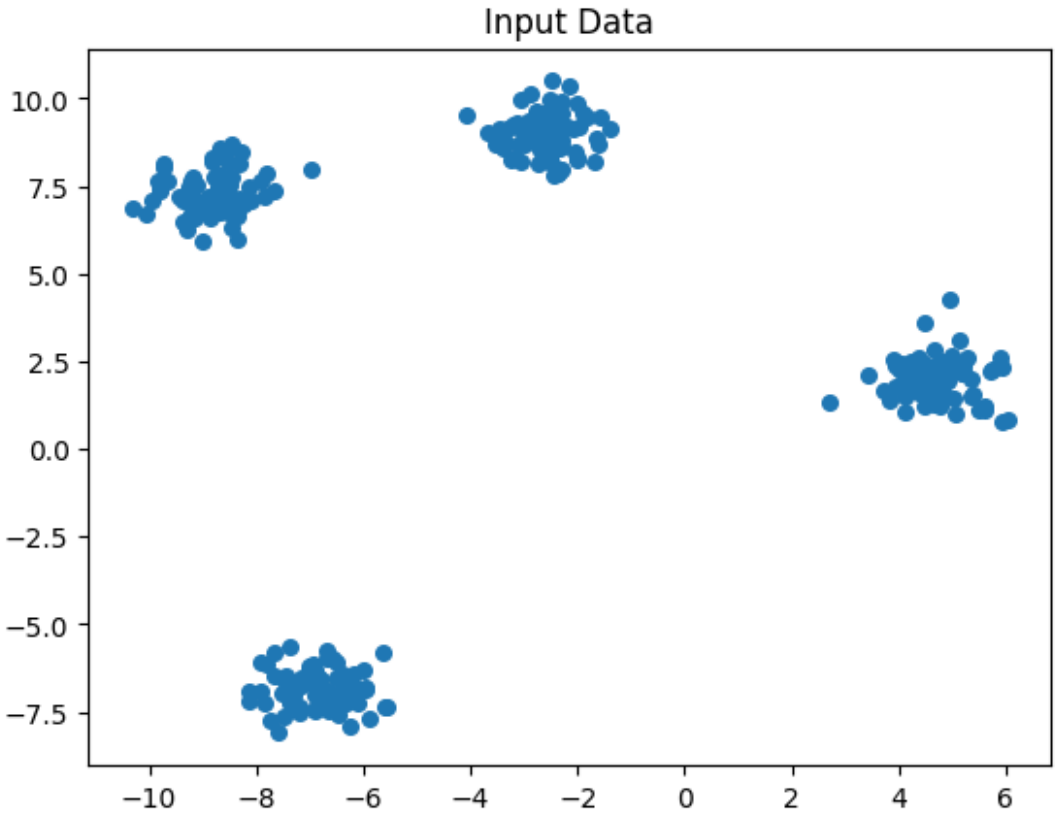
Procedure:

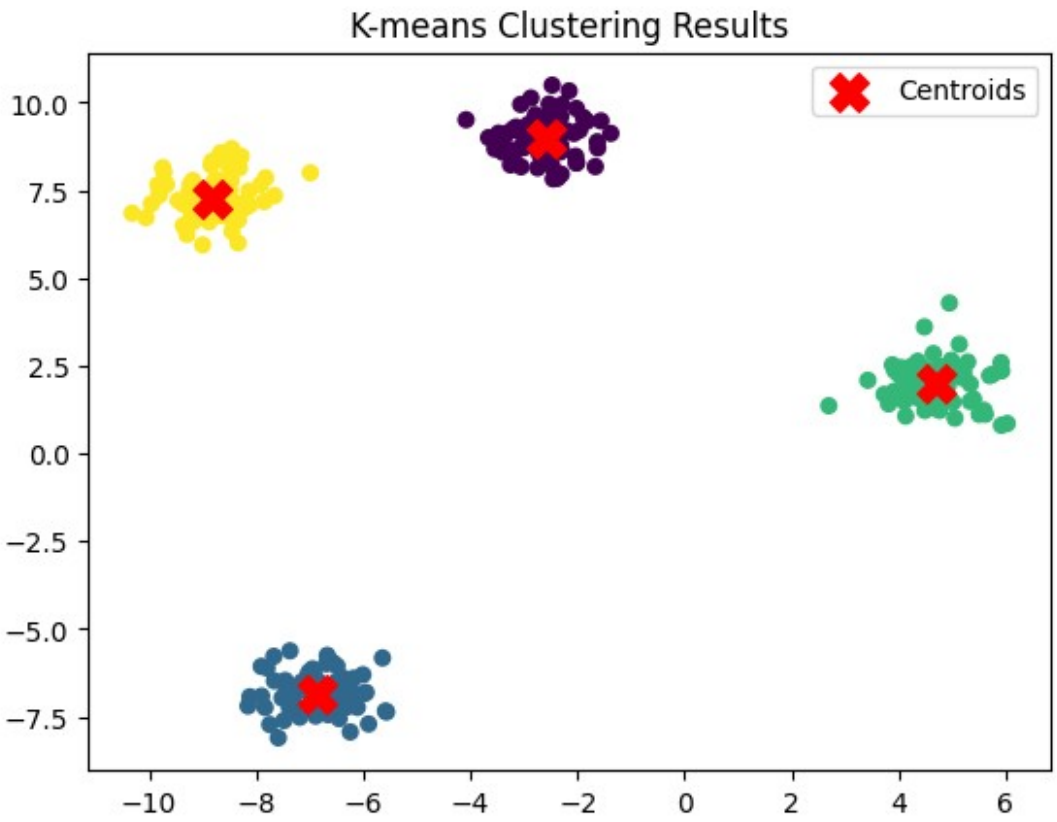
1. Generate synthetic data using the `make_blobs` function.
2. Visualize the input data using `matplotlib`.
3. Apply the K-means clustering algorithm with four clusters.
4. Obtain cluster centers and labels from the trained model.
5. Visualize the clustered data along with centroids.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

# Generate some sample data
from sklearn.datasets import make_blobs
data, _ = make_blobs(n_samples=300, centers=4, cluster_std=0.6,
random_state=42)

# Visualize the data
plt.scatter(data[:, 0], data[:, 1], s=30, cmap='viridis')
plt.title("Input Data")
plt.show()

# Apply K-means clustering
kmeans = KMeans(n_clusters=4, random_state=42)  # Set the number of
clusters (K)
kmeans.fit(data)

# Get the cluster centers and labels
centroids = kmeans.cluster_centers_
labels = kmeans.labels_

# Visualize the clustered data
plt.scatter(data[:, 0], data[:, 1], c=labels, s=30, cmap='viridis')
```

```
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', marker='X',
s=200, label='Centroids')
plt.title("K-means Clustering Results")
plt.legend()
plt.show()
```

```
<ipython-input-3-35c90386a3b9>:10: UserWarning: No data for
colormapping provided via 'c'. Parameters 'cmap' will be ignored
  plt.scatter(data[:, 0], data[:, 1], s=30, cmap='viridis')
```





Conclusion:
The K-means algorithm successfully grouped the data into four clusters, identifying patterns based on similarity. The centroids represent the cluster centers, and the visualization confirms the effectiveness of K-means in partitioning data.

Review Questions & Answers

1. **What is the K-means clustering algorithm, and how does it work?**
   K-means is an unsupervised clustering algorithm that partitions a dataset into K distinct clusters. It works as follows:
   - Select K initial cluster centroids (randomly or using specific techniques).
   - Assign each data point to the nearest centroid based on distance (typically Euclidean distance).
   - Compute new centroids by averaging the points in each cluster.
   - Repeat the assignment and centroid update steps until convergence (i.e., centroids no longer change significantly or a maximum number of iterations is reached).
2. **How do you determine the optimal number of clusters in K-means?**
   The optimal number of clusters (K) can be determined using:

- **Elbow Method**: Plot the Within-Cluster Sum of Squares (WCSS) against different K values and select the "elbow point," where the decrease in WCSS slows down.

- **Silhouette Score**: Measures the quality of clustering by evaluating how similar a point is to its own cluster versus others. A higher silhouette score suggests better clustering.

- **Gap Statistic**: Compares the performance of a clustering algorithm against randomly generated data to determine the optimal K.

3. **What are the common distance metrics used in Agglomerative Clustering?**
   Agglomerative Clustering is a hierarchical clustering method that uses various distance metrics, including:
   - **Euclidean Distance**: Measures the straight-line distance between points.

   - **Manhattan Distance**: Measures distance along grid-like paths (sum of absolute differences).

   - **Cosine Similarity**: Measures the cosine of the angle between vectors (used for high-dimensional data).

   - **Mahalanobis Distance**: Considers correlations between variables and is useful for multivariate data.
     The choice of metric affects how clusters are formed and their interpretability.