St. Francis Institute of Technology, Mumbai-400 103

Department Of Information Technology

A.Y. 2024-2025

Class: TE-ITA/B, Semester: VI

Subject: **Business Intelligence Lab**

Experiment – 6: To Implement K Means clustering algorithm using Java/Python

1. **Aim: :** To Implement K Means clustering algorithm using Java/Python

1. **Objectives:** After study of this experiment, the students will be able to implement K Means Algorithm

2. **Outcomes:** After study of this experiment, the students will be able to
   CO 4: Design and Implement various clustering data mining techniques such as Partitioning methods, Hierarchical Methods, Density - Based methods along with identification and analysis of outlier.

1. **Prerequisite:** Introduction to all the three clustering algorithms & Problem solving approach.

2. **Requirements:** Personal Computer, Windows XP operating system/Windows 7, Internet Connection, Microsoft Word, WEKA tool.

1. **Theory:**
   a. What is Clustering in Data Mining?
   b. Explain K-means clustering algorithm

7. **Laboratory Exercise:** Implementation of K means Classification Algorithm using Java/Python, Printout of implementation along with coding and snapshot

1. **Post-Experiments Exercise**

a. **Questions:**
   - In form of MCQ type test
   - K means solved numerical

a. **Conclusion:**
   - Summary of Experiment
   - Importance of Experiment
   - Application of Experiment

1. **Reference:** Data Mining: Concept & Techniques, 3rd Edition, Jiawei Han, Micheline Kamber, Jian Pei, Elsevier.

In [ ]:

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

In [ ]:

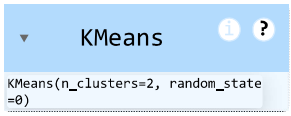
```
data = np.array([2, 15, 10, 15, 5, 20, 4, 40]).reshape(-1, 1)
```

In [ ]:

```
# Initialize the KMeans algorithm with k=2 clusters
kmeans = KMeans(n_clusters=2, random_state=0)

# Fit the model to the data
kmeans.fit(data)
```

Out[ ]:



In [ ]:

```
# Predict the cluster labels
labels = kmeans.predict(data)

# Get the cluster centers
centroids = kmeans.cluster_centers_
```

In [ ]:

```
# Assign data points to clusters
cluster_1 = data[labels == 0]
cluster_2 = data[labels == 1]

# Print the clusters
print(f"Cluster 1: {cluster_1.flatten()}")
print(f"Cluster 2: {cluster_2.flatten()}")
```

```
Cluster 1: [ 2 15 10 15  5 20  4]
Cluster 2: [40]
```

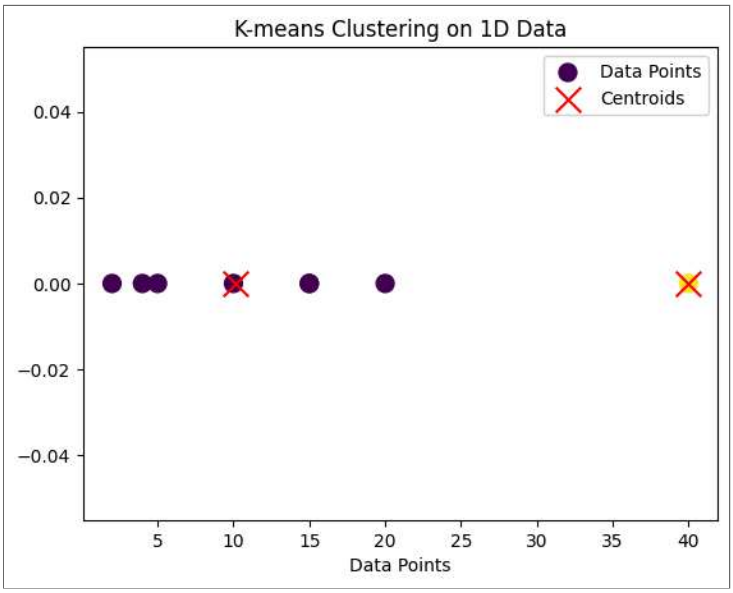
In [ ]:

```
# Plot the data points
plt.scatter(data, np.zeros_like(data), c=labels, cmap='viridis', marker='o', s=100, label='Data Points

# Plot the centroids
plt.scatter(centroids, np.zeros_like(centroids), c='red', marker='x', s=200, label='Centroids')

# Add labels and title
plt.xlabel('Data Points')
plt.title('K-means Clustering on 1D Data')
plt.legend()

# Show the plot
plt.show()
```


K-means Clustering on 1D Data
Data Points
Centroids
0.04
0.02
0.00
−0.02
−0.04
5
10
15
20
25
30
35
40
Data Points