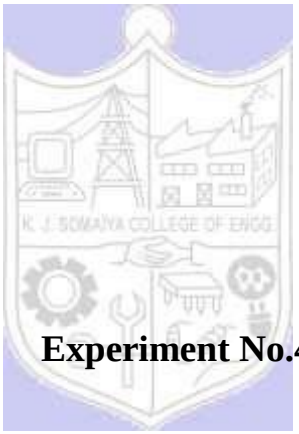


Experiment No.4

Title: Implementation of K- Means clustering algorithm for a given case study.

Batch: B2 **Roll No.: 16010420117** **Experiment No.: 4**

Aim: Implementation of K- Means clustering algorithm.

Resources needed: Any RDBMS, Java

Theory:

Cluster analysis or clustering is the process of partitioning a set of data objects (or observations) into subsets. Each subset is a cluster, such that objects in a cluster are similar to one another, yet dissimilar to objects in other clusters. The set of clusters resulting from a cluster analysis can be referred to as a clustering. **Types of clustering:**

Hierarchical algorithms:

Hierarchical algorithms find successive clusters using previously established clusters. These algorithms usually are either agglomerative ("bottom-up") or divisive ("topdown"). Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.

Partitioning algorithms:

Partitioning algorithms typically determine all clusters at once, but can also be used as divisive algorithms in the hierarchical clustering e.g K-mean, K-medoid.

Density-based clustering algorithms:

Density-based clustering algorithms are devised to discover arbitrary-shaped clusters. In this approach, a cluster is regarded as a region in which the density of data objects exceeds a threshold. DBSCAN and OPTICS are two typical algorithms of this kind.

K-Means clustering Algorithm:

The *k*-means algorithm assigns each point to the cluster whose center (also called centroid) is nearest. The center is the average of all the points in the cluster — that is, its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster.

The basic step of k-means clustering is simple. In the beginning determine number of cluster K and assume the centroid or center of these clusters. Take any random objects as the initial centroids or the first K objects in sequence can also serve as the initial centroids.

Then the K means algorithm will do the three steps below until convergence

Iterate until *stable* (= no object move group):

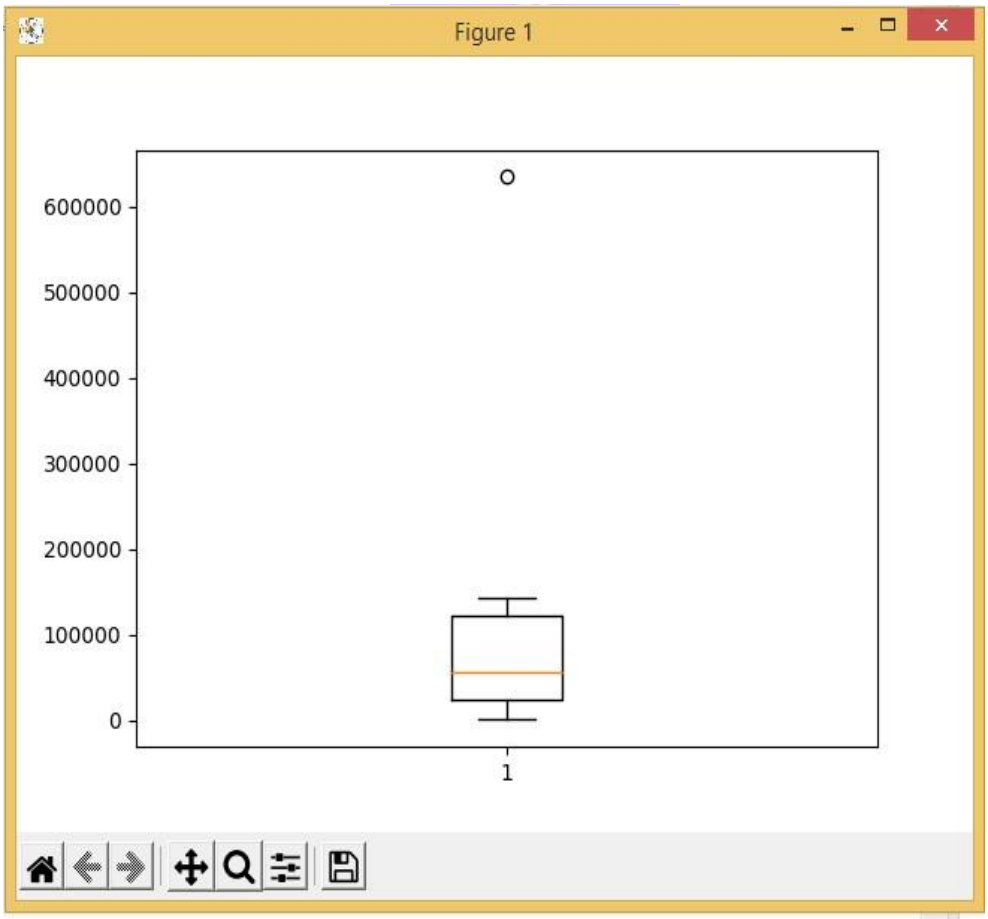
1. Determine the centroid coordinate
2. Determine the distance of each object to the centroids
3. Group the object based on minimum distance.

Procedure / Approach /Algorithm / Activity Diagram:

1. Download the dataset available at https://www.kaggle.com/code/vineetverma/clustering-bank-complaints/notebook. Identify attributes suitable for applying K-mean clustering
2. Apply some preprocessing techniques to drop some columns.
3. Draw a box plot to visualize the data.
4. Implement K-mean clustering on your dataset for clustering different types of complaints.
5. Evaluate the performance of your algorithm with suitable technique.

Results: (Program printout with output / Document printout as per the format)

3.) Boxplot

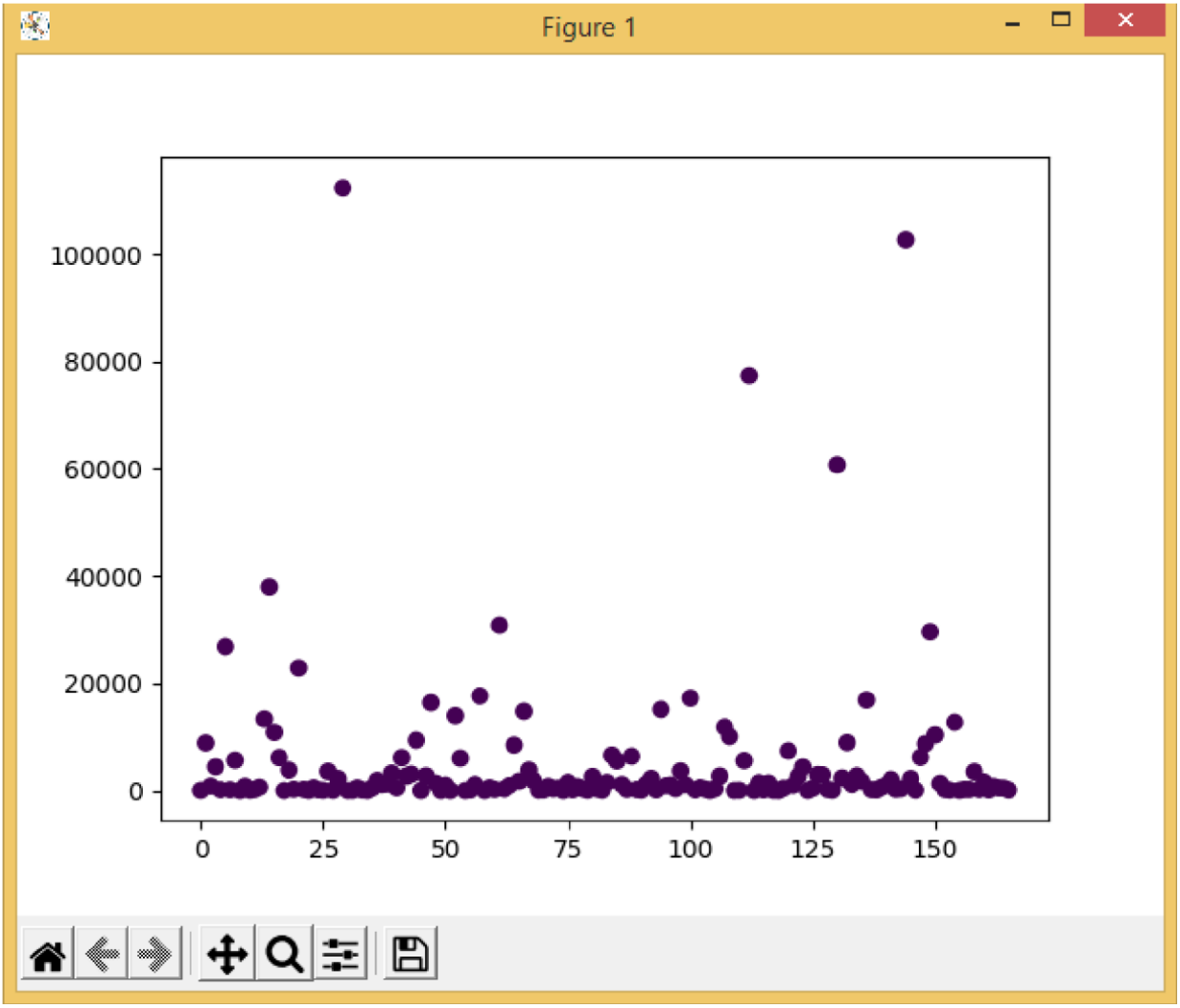


4.) Kmean clustering code:

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

df = pd.read_csv('Consumer_Complaints.csv') df.drop(['Sub-
issue','Consumer complaint narrative','Company public
response','Consumer consent provided?','Tags'],axis=1,inplace=True)

dicti={'Email':0,'Fax':0,'Phone':0,'Postal
mail':0,'Referral':0,'Web':0} for i in df['Submitted via']:
dicti[i]+=1

plt.boxplot(dicti.values())
plt.show()

s =
set(df["Issue"])
dicti2 = {} for
i in s:
dicti2[i] = 0

for i in df["Issue"]:

dicti2[i]+=1

print(dicti2)

dicti3 = [] for i in
range(len(dicti2)):
dicti3.append(i)



x = dicti3 y =

dicti2.values()

data = list(zip(x,

y))

kmeans = KMeans(n_clusters=1)
kmeans.fit(data)

plt.scatter(x, y, c=kmeans.labels_)
plt.show()
```

```
ytest = list(y)[100:] ypred =
kmeans.predict(list(zip(x[100:],list(y)[100:]))) print("Accuracy : " ,
(r2_score(ytest, ypred))*100 ,
```

Output:



5.) Evaluation:

```
Accuracy :  77.28418317343424 %
```

Questions:

1. What are advantages and disadvantages of K-means clustering algorithm?

Ans:

Advantages of k-means

- Relatively simple to implement.
- Scales to large data sets.
- Guarantees convergence.
- Can warm-start the positions of centroids.
- Easily adapts to new examples.
- Generalizes to clusters of different shapes and sizes, such as elliptical clusters.

Disadvantages of k-means

- **Choosing** k **manually.**
  - Use the "Loss vs. Clusters" plot to find the optimal (k)
- **Being dependent on initial values.**
  - For a low k, you can mitigate this dependence by running k-means several times with different initial values and picking the best result. As k increases, you need advanced versions of k-means to pick better values of the initial centroids (called **k-means seeding**).
- **Clustering data of varying sizes and density.**
  - k-means has trouble clustering data where clusters are of varying sizes and density.
- **Clustering outliers.**
  - Centroids can be dragged by outliers, or outliers might get their own cluster instead of being ignored. Consider removing or clipping outliers before clustering.

Outcomes: CO3: Comprehend radial-basis-function (RBF) networks and Kernel learning method

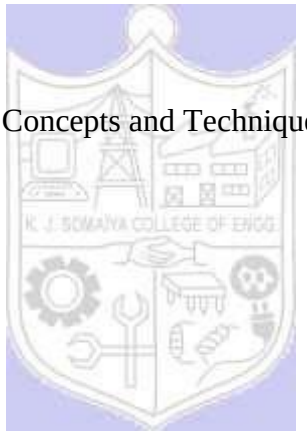
Conclusion: (Conclusion to be based on the objectives and outcomes achieved)
We conclude that we were able to implement the K-means clustering algorithm.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:



1. Han, Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann 3nd Edition