Batch: B2 **Experiment Number: 4**

Roll Number: 16010421073 **Name: Keyur Patel**

Aim of the Experiment:

To implement any four K-value selection algorithms along with K means clustering algorithm.

Program/ Steps:

1. Download unsupervised dataset.

2. Implement any two K value selection algorithms.

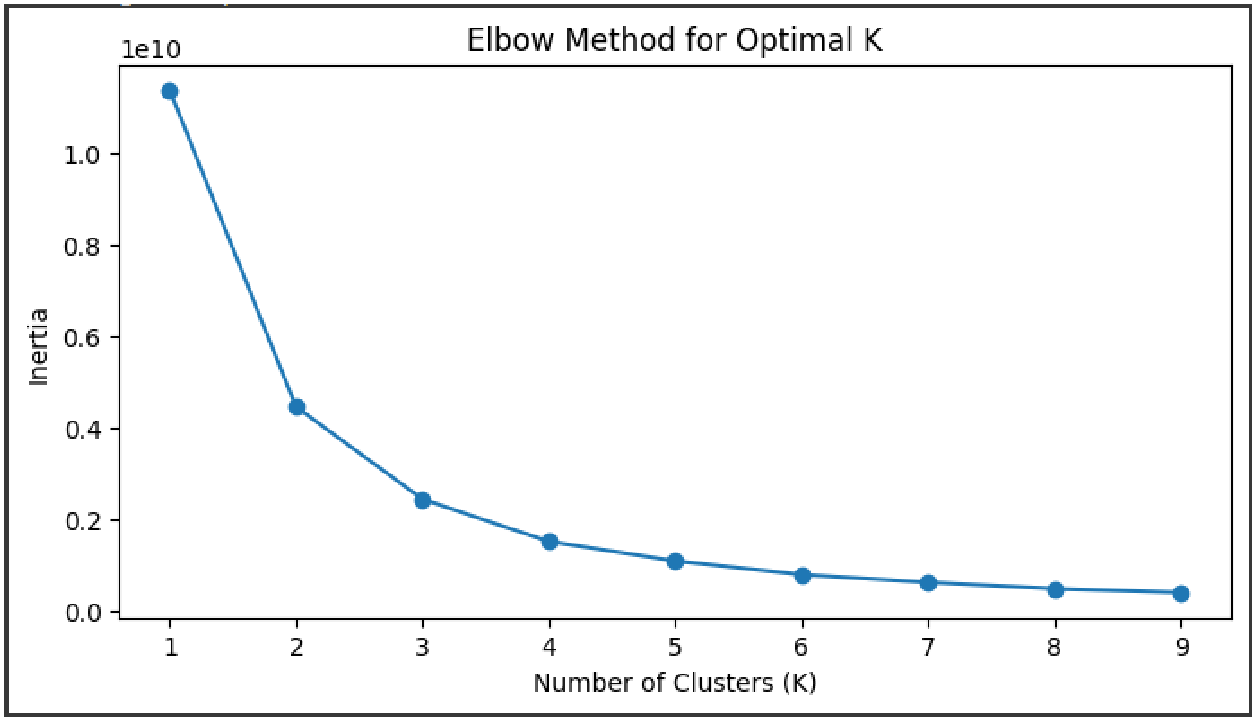
3. Using the selected value of K from selection algorithms, implement K means clustering algorithm.

4. Compare the results with different values of k and comment on same.

Output/Result:

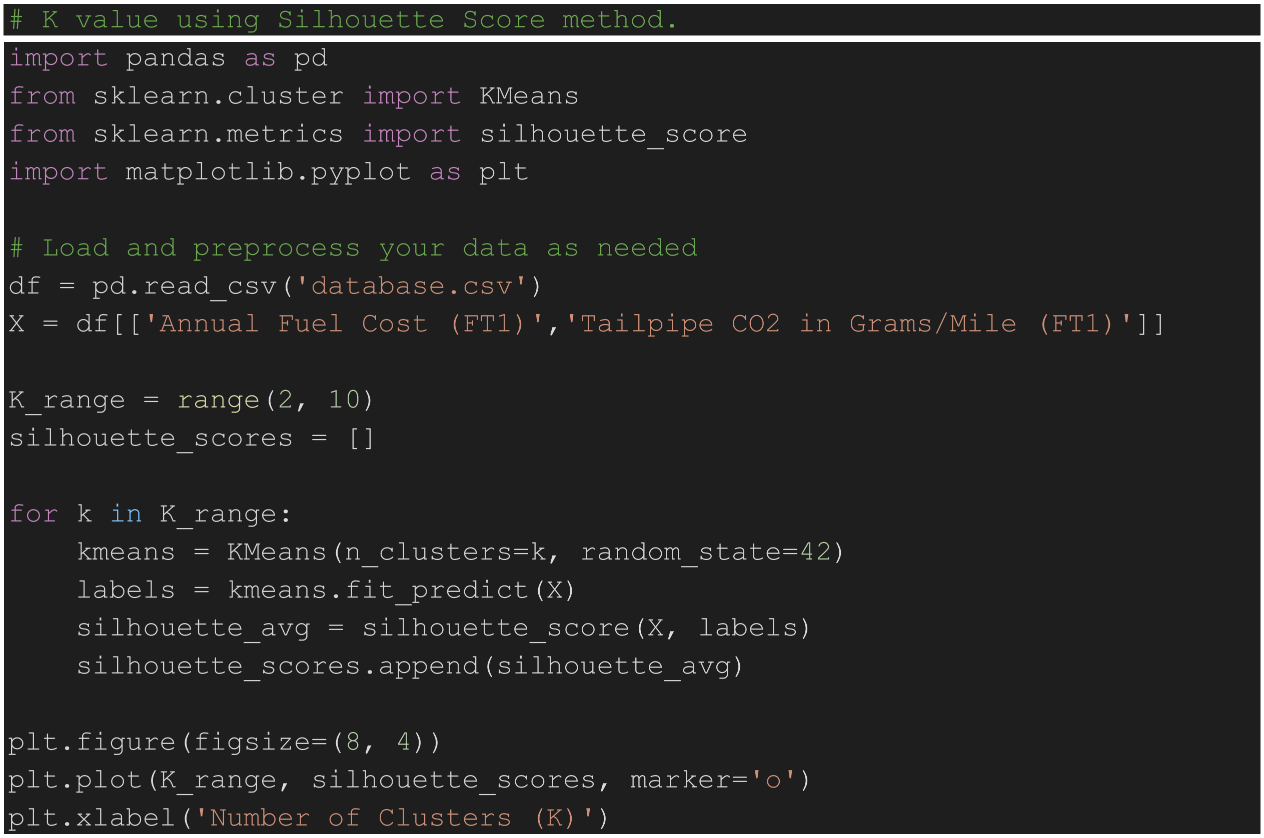
2. <u>Using Elbow Method for K value Selection.</u>

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

# Load and preprocess your data as needed
df = pd.read_csv('database.csv')
X = df[['Annual Fuel Cost (FT1)','Tailpipe CO2 in Grams/Mile (FT1)']]

K_range = range(1, 10)
inertias = []

for k in K_range:
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(X)
    inertias.append(kmeans.inertia_)

plt.figure(figsize=(8, 4))
plt.plot(K_range, inertias, marker='o')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('Inertia')
plt.title('Elbow Method for Optimal K')
plt.show()
```

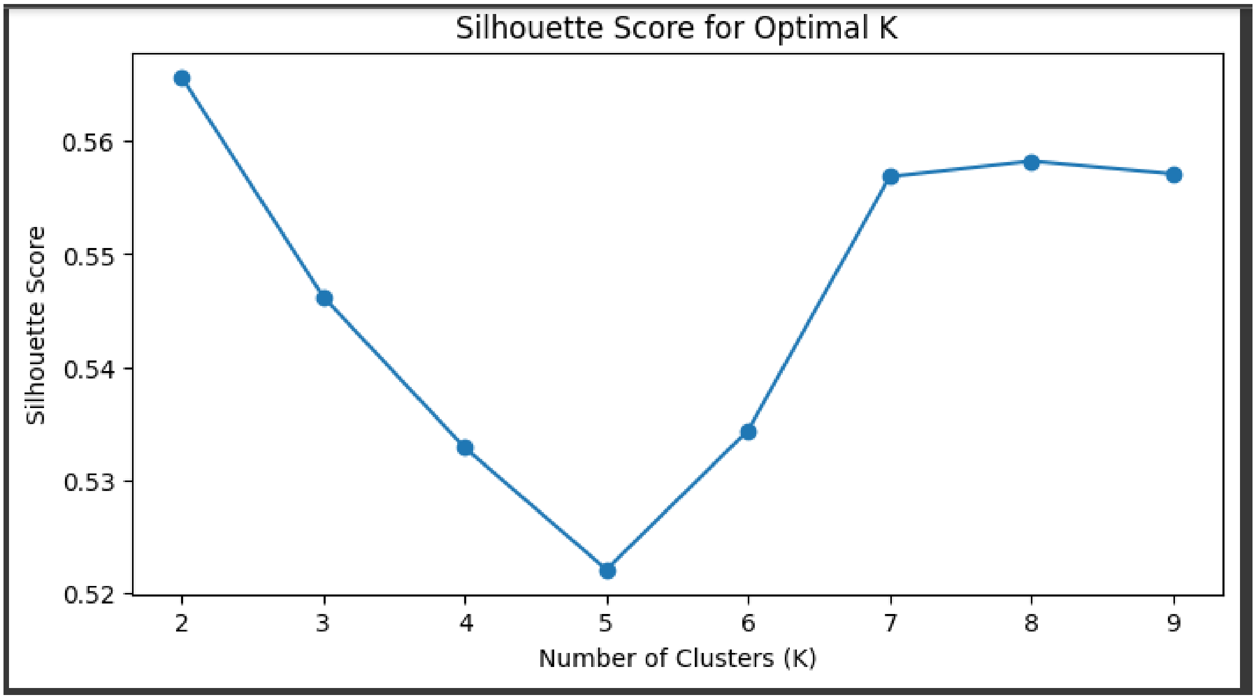


Therefor K value here will be 4.

Using Silhouette Score Method:



```
# K value using Silhouette Score method.
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt

# Load and preprocess your data as needed
df = pd.read_csv('database.csv')
X = df[['Annual Fuel Cost (FT1)','Tailpipe CO2 in Grams/Mile (FT1)']]

K_range = range(2, 10)
silhouette_scores = []

for k in K_range:
    kmeans = KMeans(n_clusters=k, random_state=42)
    labels = kmeans.fit_predict(X)
    silhouette_avg = silhouette_score(X, labels)
    silhouette_scores.append(silhouette_avg)

plt.figure(figsize=(8, 4))
plt.plot(K_range, silhouette_scores, marker='o')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Score for Optimal K')
plt.show()
```



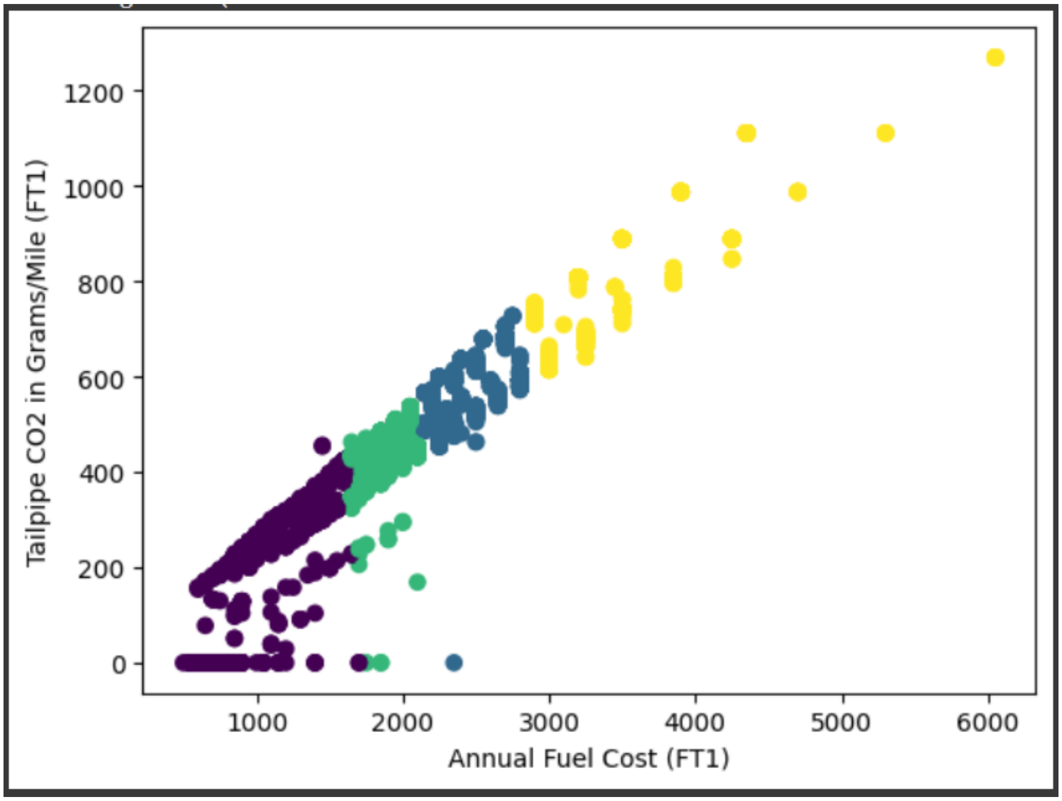
Therefor K value here will be 7.

3. K-means Clustering

a) Using Elbow method value: (K=4)

```
#K means Clusterring algorithm
import pandas as pd
from sklearn.cluster import KMeans
import numpy as np
import matplotlib.pyplot as plt

# Generate or load your dataset (replace this with your data)
df = pd.read_csv('database.csv')
x = df[['Annual Fuel Cost (FT1)','Tailpipe CO2 in Grams/Mile (FT1)']]

# Define the number of clusters (K)
K = 4


kmeans = KMeans(n_clusters=K, random_state=42)
kmeans.fit(x)
```

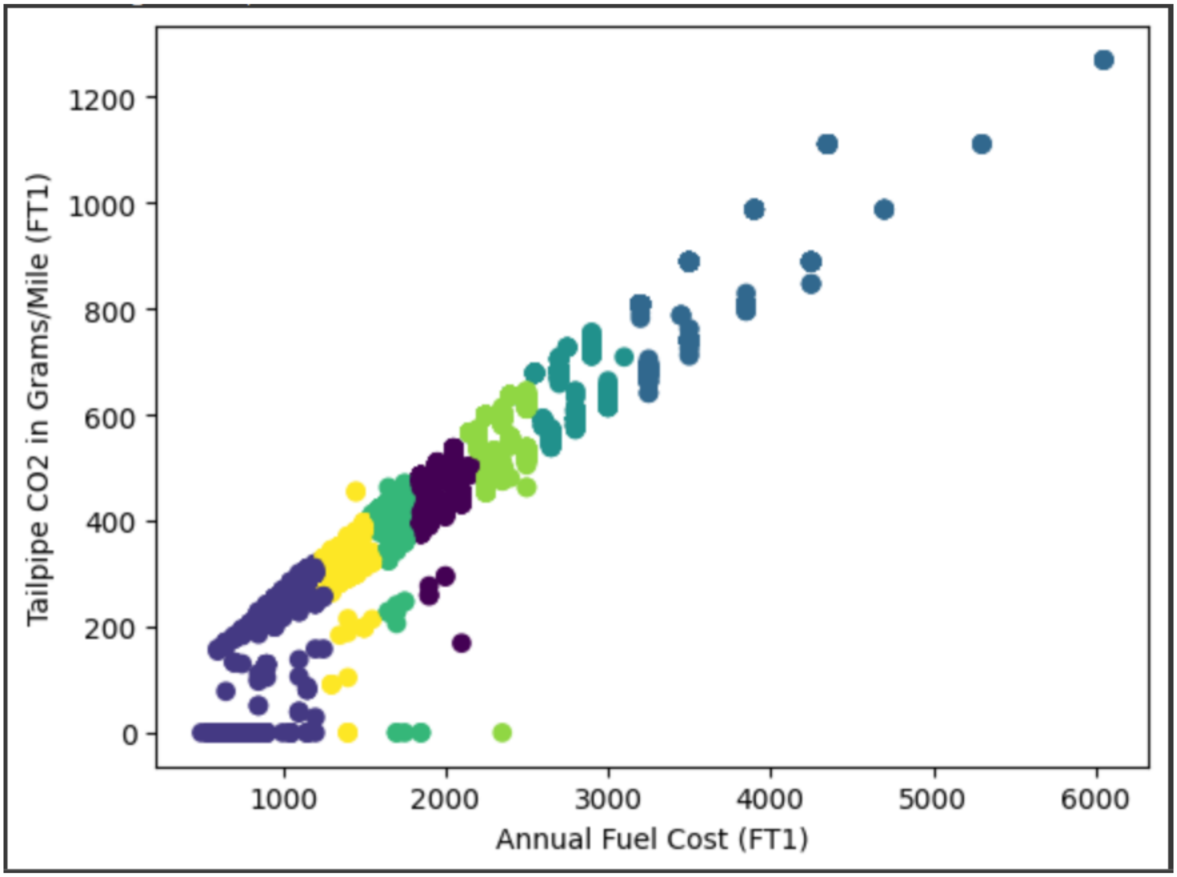
```
labels = kmeans.labels_

plt.scatter(x['Annual Fuel Cost (FT1)'], x['Tailpipe CO2 in Grams/Mile 
(FT1)'], c=labels)
plt.xlabel('Annual Fuel Cost (FT1)')
plt.ylabel('Tailpipe CO2 in Grams/Mile (FT1)')
plt.show()
```



b) Using Silhouette Score method value: (K=7)

```
#K means Clusterring algorithm
import pandas as pd
from sklearn.cluster import KMeans
import numpy as np
import matplotlib.pyplot as plt

# Generate or load your dataset (replace this with your data)
df = pd.read_csv('database.csv')
x = df[['Annual Fuel Cost (FT1)','Tailpipe CO2 in Grams/Mile (FT1)']]

# Define the number of clusters (K)
K = 7
```

```
kmeans = KMeans(n_clusters=K, random_state=42)
kmeans.fit(x)
labels = kmeans.labels_

plt.scatter(x['Annual Fuel Cost (FT1)'], x['Tailpipe CO2 in Grams/Mile
(FT1)'], c=labels)
plt.xlabel('Annual Fuel Cost (FT1)')
plt.ylabel('Tailpipe CO2 in Grams/Mile (FT1)')
plt.show()
```



Post Lab Question-Answers:

1. **What is the main difference between K-means and K-nearest neighbors?**

Here is a table summarizing the main differences between K-means and K-nearest neighbours (KNN):

Aspect	**K-means**	**K-nearest neighbors (KNN)**
Type	Clustering algorithm	Classification algorithm
Supervised/Unsupervised	Unsupervised	Supervised
Objective	Group data points into clusters	Predict the class of data points based on their neighbors
Usage	Clustering and unsupervised tasks	Classification and supervised tasks
Input	Features of data points	Features and labels of data points
Parameter (K)	Number of clusters (K) to be determined	Number of nearest neighbors (K) to consider
Training	Iteratively adjusts cluster centroids	Memorizes the entire training dataset
Decision boundary	Forms clusters with centroids	Forms decision boundaries based on data points
Distance Metric	Typically Euclidean distance	Various distance metrics can be used
Data Representation	Vector space model (numerical features)	Can handle various data types (e.g., categorical, numerical)
Interpretability	Clusters represent groupings of data	Decision boundaries and class labels provide interpretability
Example Applications	Customer segmentation, image compression	Image classification, recommendation systems
Scalability	Scales well with a large number of data points	Memory-intensive and may not scale well with a large dataset
Hyperparameter Tuning	Typically focuses on finding the optimal K value	Focuses on tuning the value of K and the choice of distance metric

2. What are some stopping criteria for k-means clustering?

Stopping criteria in K-means clustering are conditions that determine when to terminate the iterative process of assigning data points to clusters and updating cluster centroids. Using appropriate stopping criteria is essential to ensure that the K-means algorithm converges to a solution effectively. Here are some common stopping criteria for K-means clustering:

1. **Convergence:** The most common stopping criterion for K-means is based on convergence. You can stop the algorithm when the cluster assignments and cluster centroids no longer change significantly between iterations. This can be measured using a tolerance threshold for the change in cluster centroids or a maximum number of iterations.
2. **Maximum Iterations:** Set a predefined maximum number of iterations. If the algorithm reaches this limit without convergence, it will terminate. This is a safety net to prevent infinite loops or excessive computation.
3. **Minimum Cluster Membership:** You can set a minimum number of data points that a cluster must have. If a cluster has fewer data points than this threshold, consider the cluster to be too small and merge it with its closest neighbor or reassign the data points.
4. **Minimum Improvement in Inertia:** Inertia (within-cluster sum of squares) is a measure of the clustering quality. You can stop when the change in inertia from one iteration to the next is less than a certain threshold, indicating that further refinement is unlikely to significantly improve the result.
5. **Silhouette Score:** You can monitor the Silhouette Score, which measures the quality of clustering. If the Silhouette Score stabilizes or starts to decrease, it can be a sign to stop the algorithm. This method ensures that the clusters are well-separated.
6. **External Criteria:** If you have access to external criteria or ground truth labels for your data, you can use them to evaluate the quality of clustering. You can stop when the clustering result satisfies some external criterion like adjusted Rand index or normalized mutual information.
7. **Centroid Movement:** Set a threshold for the movement of cluster centroids between iterations. If the movement is below this threshold, consider the algorithm converged.
8. Runtime Limit: If you have constraints on the maximum time allowed for clustering, you can stop the algorithm when it reaches this runtime limit.
9. **Distance Threshold:** If you have domain-specific knowledge, you may want to stop the algorithm when the distance between cluster centroids becomes smaller than a certain threshold.

Outcomes:

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

Conclusion (based on the Results and outcomes achieved):

Understood the concepts of K value and K-mean algorithm and then implemented two k value algorithms on our database to find the appropriate value for K and then using this value implemented the K-mean clustering algorithm on our database.

References:

Books/ Journals/ Websites: