Experiment No. 7

Expt No. 7	Implementation of K-Means Clustering Algorithm
Date :	

Aim:

To implement and understand the working of K-Means clustering algorithm, an unsupervised learning technique used for grouping data point into cluster based on their similarity in Machine Learning.

System Requirements:

- Operating System: Windows 8 or above / Linux / macOS
- Memory (RAM): Minimum 4 GB
- Processor: Minimum 2.33 GHz (Dual Core or higher)

Software/Tools Required:

Jupyter Notebook / Anaconda Navigator / Google Colaboratory / Spyder
Python 3.x with the following libraries:

- NumPy
- Pandas
- Matplotlib
- Seaborn
- Scikit-learn

Purpose of the Experiment:

- To implement the K-Means clustering algorithm on a dataset for unsupervised learning.

- To explore the process of clustering data into groups based on similarity using K-Means.
- To evaluate the performance and cluster separation by visualizing the clusters and their centroids.
- To understand the Elbow method for determining the optimal number of clusters.

Expected Outcomes:

- Understand the K-Means clustering algorithm and its application to data.
- Implement the K-Means algorithm and visualize the resulting clusters.
- Use the Elbow technique to determine the optimal number of clusters.
- Analyze the clustering results using scatter plots and centroids.

Theory:

K-Means Clustering:
K-Means is a popular unsupervised machine learning algorithm used for clustering tasks. It groups data points into K clusters based on similarity, where each cluster is represented by a centroid (mean of the points in the cluster). The K-Means algorithm minimizes the Sum of Squared Errors (SSE) between the points and their corresponding centroids.

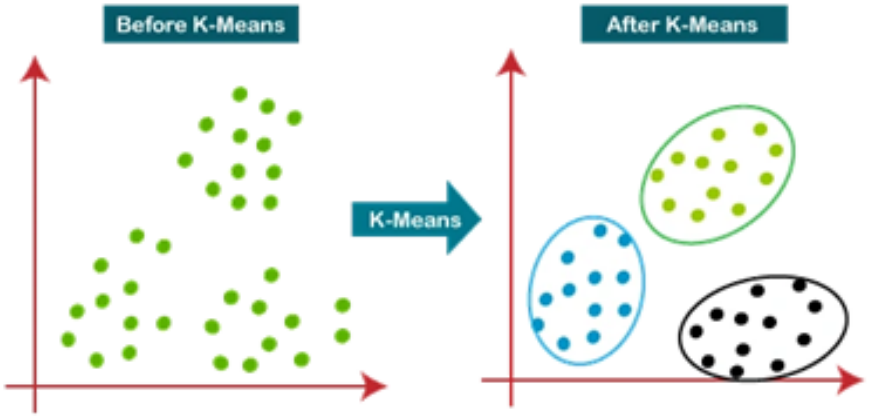
Working of K-Means:

1. Initialization: Choose K random centroids.
2. Assignment Step: Assign each data point to the nearest centroid based on Euclidean distance.
3. Update Step: Recalculate the centroid of each cluster by averaging the points assigned to it.
4. Convergence: Repeat the assignment and update steps until the centroids no longer change significantly.

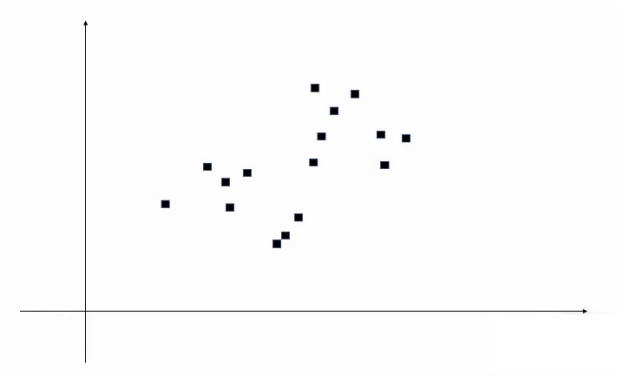
The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

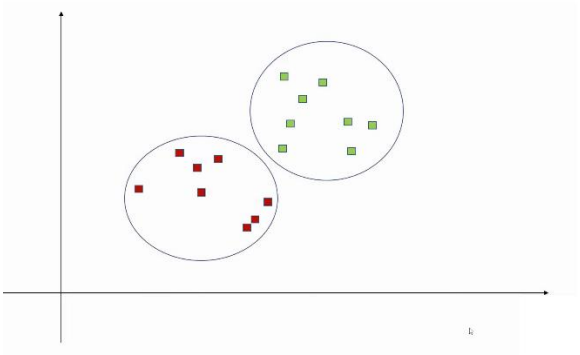
The k-means clustering algorithm mainly performs two tasks:

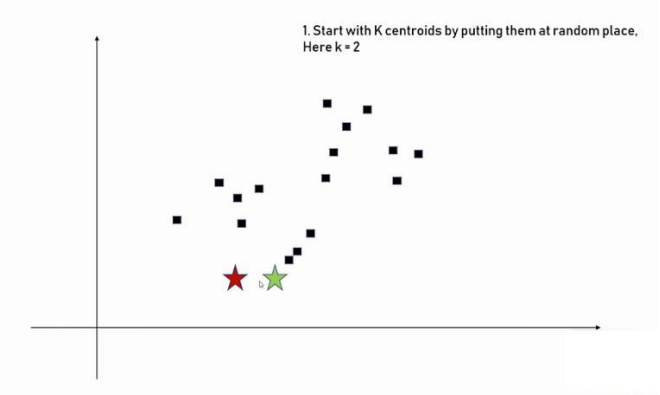
- Determines the best value for K center points or centroids by an iterative process
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

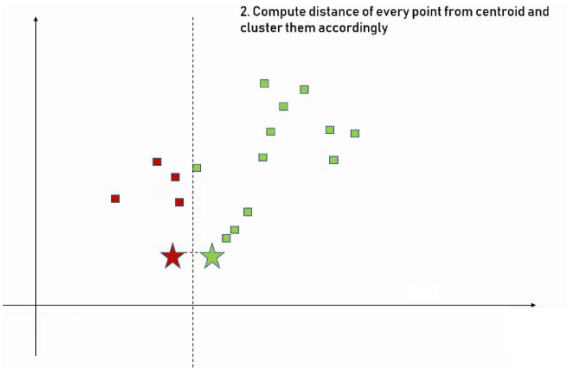


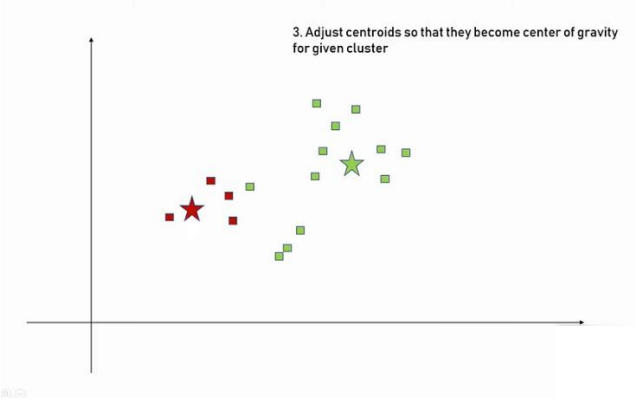
Below is the step-by-step procedure:

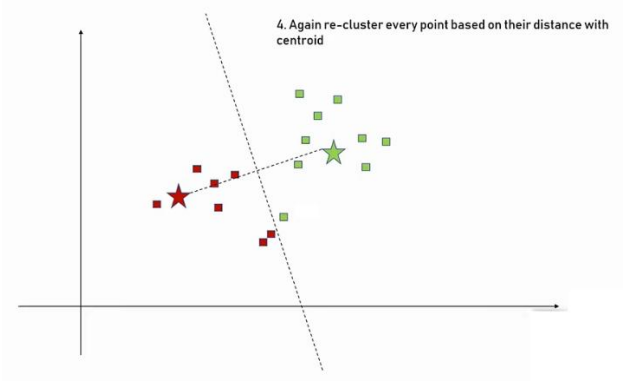


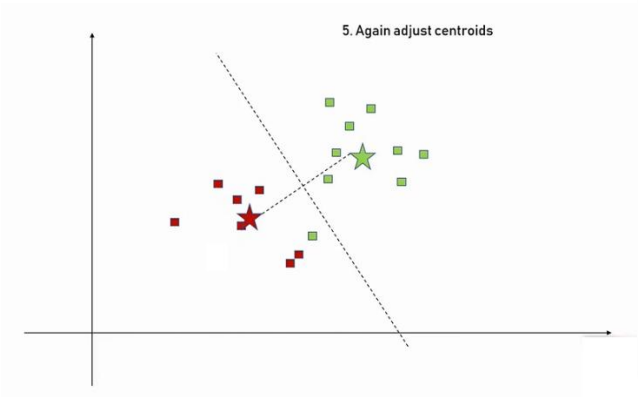


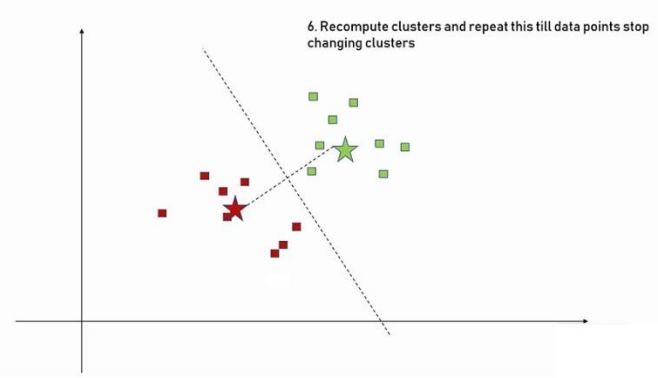


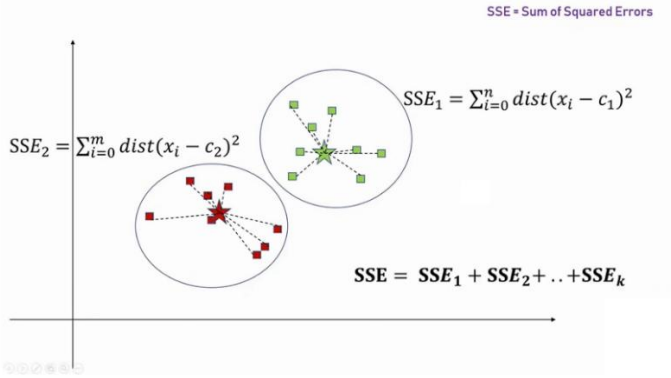


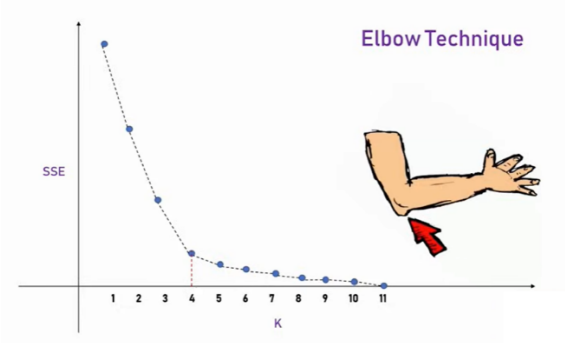












Elbow Technique for Optimal K:

Choosing the number of clusters K is crucial for the effectiveness of the K-Means algorithm. The Elbow method helps to find an optimal value of K by plotting the Sum of Squared Errors (SSE) for different values of K and looking for the "elbow" point on the plot. The elbow point corresponds to the K where increasing the number of clusters does not significantly reduce the SSE, suggesting a balance between model complexity and accuracy.

Real-World Applications of K-Means Clustering Algorithm:

1. Customer segmentation in marketing
2. Image compression
3. Anomaly detection
4. Data compression and pre-processing.

1. Problem Statement

The objective of this experiment is to apply the K-Means clustering algorithm to the income.csv dataset, which contains data on Age and Income($). The experiment involves:

1. Preprocessing the data by scaling the features.
2. Fitting the K-Means model to the data for clustering.
3. Visualizing the clusters and their centroids.
4. Using the Elbow method to determine the optimal number of clusters.

You can download the dataset from the following link:

https://github.com/25-Madhuri/Dataset/blob/main/income.csv

Sample Code

```
# Importing necessary libraries for data manipulation, visualization, and modeling

import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt

#Load the Income dataset from a CSV file and display first few rows of dataset

df = pd.read_csv("income.csv")
df.head()

# Import necessary library for plotting

import matplotlib.pyplot as plt
```

```
# Scatter plot to visualize Age vs Income

plt.scatter(df.Age, df['Income($)'])
plt.xlabel('Age')
plt.ylabel('Income($)')
plt.title('Age vs Income Scatter Plot')
plt.show()

# Import KMeans from sklearn

from sklearn.cluster import KMeans

# Initialize KMeans with 3 clusters

km = KMeans(n_clusters=3)

# Fit the model and predict the clusters for 'Age' and 'Income($)'

y_predicted = km.fit_predict(df[['Age', 'Income($)']])

# Display the predicted clusters

y_predicted

# Add the predicted cluster labels to the DataFrame

df['cluster'] = y_predicted

# Display the first few rows of the updated DataFrame

df.head()

km.cluster_centers_

#Visualizing the Clusters:
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
```

```
df3 = df[df.cluster==2]
plt.scatter(df1.Age,df1['Income($)'],color='green')
plt.scatter(df2.Age,df2['Income($)'],color='red')
plt.scatter(df3.Age,df3['Income($)'],color='black')
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purple',marker='*',
label='centroid')
plt.xlabel('Age')
plt.ylabel('Income ($)')
plt.legend()

# Import MinMaxScaler
from sklearn.preprocessing import MinMaxScaler

# Initialize MinMaxScaler
scaler = MinMaxScaler()

# Scale the 'Income($)' column
scaler.fit(df[['Income($)']])
df['Income($)'] = scaler.transform(df[['Income($)']])

# Scale the 'Age' column
scaler.fit(df[['Age']])
df['Age'] = scaler.transform(df[['Age']])

df
plt.scatter(df.Age,df['Income($)'])
km = KMeans(n_clusters=3)
y_predicted = km.fit_predict(df[['Age','Income($)']])
y_predicted

df['cluster']=y_predicted
df.head()

df['cluster']=y_predicted
df.head()

df1 = df[df.cluster==0]
```

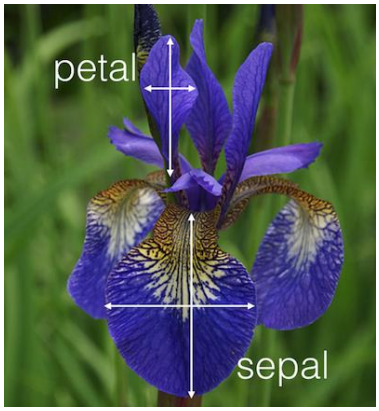
```
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
plt.scatter(df1.Age,df1['Income($)'],color='green')
plt.scatter(df2.Age,df2['Income($)'],color='red')
plt.scatter(df3.Age,df3['Income($)'],color='black')
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purple',marker='*',l
abel='centroid')
plt.legend()

sse = []
k_rng = range(1,10)
for k in k_rng:
    km = KMeans(n_clusters=k)
    km.fit(df[['Age','Income($)']])
    sse.append(km.inertia_)

plt.xlabel('K')
plt.ylabel('Sum of squared error')
plt.plot(k_rng,sse)
```

Observation

I. Record the following in observation table:

Sr.No	Metric	Description	Observations
1.	Shape of Dataset	Number of rows and columns in the dataset (df.shape)	
2.	Cluster Centers	Coordinates of the K centroids	
3.	Model Accuracy	Inertia score for K-Means	
4.	Elbow Method	Determine the optimal value of K based on SSE	

Exercise:



1. Use iris flower dataset from sklearn library and try to form clusters of flowers using petal width and length features. Drop other two features for simplicity.
2. Figure out if any preprocessing such as scaling would help here
3. Draw elbow plot and from that figure out optimal value of k
4. Dataset link: https://github.com/25-Madhuri/Dataset/blob/main/Iris.csv

Conclusion –

The K-Means algorithm successfully clustered the data points into groups based on similarity. The Elbow method was used to determine the optimal number of clusters by analyzing the SSE. The final clusters were visualized, and the centroids were plotted to illustrate how K-Means grouped the data.

Viva Questions

Sr.No	Question	CO
1.	What is the K-Means clustering algorithm, and how does it work?	1ICPC406_4
2.	How does K-Means reduce the error between data points and centroids?	1ICPC406_4

3.	Explain the Elbow method for determining the optimal number of clusters.	**1ICPC406_4**
4.	What is inertia in K-Means, and how does it help in clustering?	**1ICPC406_4**
5.	How do you interpret the centroids in K-Means clustering?	**1ICPC406_4**
6.	What are some limitations of the K-Means algorithm?	**1ICPC406_4**

References –

a. **Textbook –**
   i. Machine Learning with Python – An approach to Applied ML – Abhishek Vijayvargiya, BPB Publications, 1st Edition 2018
   ii. Machine Learning, Tom Mitchell, McGraw Hill Education, 1st Edition 1997

b. Online references –

i. https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/
ii. https://medium.com/@sayahfares19/k-means-clustering-algorithm-for-unsupervised-learning-tasks-f761ed7f37c0
iii. https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a/