Experiment-5

Samkit Shah

2019130060

TE Comps

Batch - C

**Aim**:

To train and test machine learning models using K-Means Clustering Algorithm.

# Theory:

* K-Means Clustering is an unsupervised learning algorithm used in machine learning and data science to handle clustering problems. It divides the unlabelled data into many clusters. K specifies the number of predetermined clusters that must be produced during the procedure; for example, if K=2, two clusters will be created, and if K=3, three clusters will be created, and so on.
* How does the K-Means algorithm work?

- The working of the K-Means algorithm is explained in the below steps: Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be different from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means assign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH. Step-7: The model is ready.

# Code:

#Importing libraries

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler

from sklearn.cluster import KMeans

#Loading dataset

import pandas as pd

data=pd.read\_csv('EmployeeSalary.csv')

data

data.plot(kind='scatter', x='WorkingYears',y='Salary')

plt.show()

#Drop ID column, we don't use this column

df=data.drop(['ID'], axis=1)

#Scaling the dataset

mms=MinMaxScaler()

mms.fit(df)

data\_transformed=mms.transform(df)

#Convert to Dataframe

data\_transformed=pd.DataFrame(data\_transformed, columns=['WorkingYears','Salary'])

data\_transformed

#Replotting dataset after scalling

data\_transformed.plot(kind='scatter', x='WorkingYears',y='Salary')

plt.show()

#Elbow method to minimize WSS (Within-cluster Sum of Square)

Sum\_of\_squared\_distances = []

K = range(1,15)

for k in K:

    km = KMeans(n\_clusters=k)

    km = km.fit(data\_transformed)

    Sum\_of\_squared\_distances.append(km.inertia\_)

#Plotting the Elbow Curve

plt.plot(K, Sum\_of\_squared\_distances, 'bx-')

plt.xlabel('k')

plt.ylabel('Sum\_of\_squared\_distances')

plt.title('Elbow Method For Optimal k')

plt.show()

#Using K-Mean with k=3 to cluster for the dataset

data=pd.read\_csv('EmployeeSalary.csv')

df=data.drop(['ID'], axis=1)

#Clustering the dataset with k=3

km3 = KMeans(n\_clusters=3)

km3 = km3.fit(df)

labels=km3.labels\_

labels=pd.DataFrame(labels, columns=['cluster'])

df\_clustered=pd.concat([df,labels], axis=1)

#how many observations are in each cluster

print(km3.labels\_)

result=km3.labels\_

result=pd.DataFrame(result, columns=['cluster'])

result.groupby('cluster').size()

#The centroid of cluster

centroids = km3.cluster\_centers\_

centroids=pd.DataFrame(centroids, columns=['Centroid\_Year', 'Centroid\_Salary'])

centroids

#Predict clusters for 3 employees with WorkingYears and Salary as below

clu\_pred=km3.predict([[18,3700],[4,900],[10,1700]])

df1 = df\_clustered[df\_clustered.cluster==0]

df2 = df\_clustered[df\_clustered.cluster==1]

df3 = df\_clustered[df\_clustered.cluster==2]

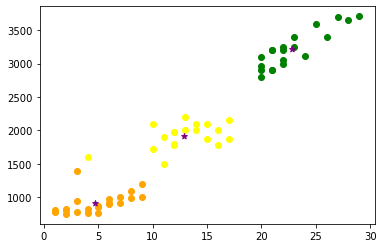
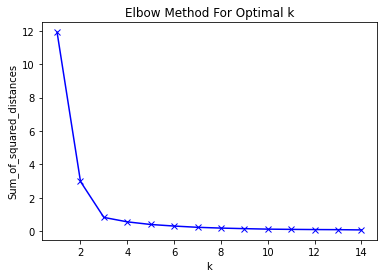
plt.scatter(df1.WorkingYears,df1['Salary'],color='green')

plt.scatter(df2.WorkingYears,df2['Salary'],color='orange')

plt.scatter(df3.WorkingYears,df3['Salary'],color='yellow')

plt.scatter(centroids.Centroid\_Year,centroids.Centroid\_Salary,color='purple',marker='\*',label='centroid')

# Output:



**Conclusion**:

* I acquired the basics of the K-Means method from the above experiment. It's a centroid-based approach, which means that each cluster has its own centroid.
* The main goal of this technique is to reduce the sum of distances between data points and the clusters that they belong to.
* It uses an iterative procedure to find the best value for K centre points or centroids, and then allocates each data point to the closest k-centre. A cluster is formed by data points that are close to a specific K-center.
* The algorithm takes an unlabeled dataset as input, separates it into k-number of clusters, and continues the procedure until no better clusters are found. In this algorithm, the value of k should be predetermined.
* The algorithm's accuracy varies depending on the number of clusters picked.