**PRACTICAL 7**

**Aim: Implementation and evaluation of Clustering Technique k-means in Python.from sklearn.cluster**

import KMeans

import pandas as pd

from sklearn.preprocessing import MinMaxScaler

from matplotlib import pyplot as plt

%matplotlib inline

df = pd.read\_csv("income.csv")

df.head()

plt.scatter(df['Age'],df['Income ($)'])

km = KMeans(n\_clusters=3)

km

y\_predicted = km.fit\_predict(df[['Age','Income ($)']])

y\_predicted

df['cluster'] = y\_predicted

df.head()

print(dir(km))

km.fit(df[['Age', 'Income ($)']])

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()

scaler = MinMaxScaler()

scaler.fit(df[['Income ($)']])

df['Income ($)'] = scaler.transform(df[['Income ($)']])

scaler.fit(df[['Age']])

df['Age'] = scaler.transform(df[['Age']])

df.head()

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()

km.cluster\_centers\_

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.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)