Name : Devkumar Biswas

Class: BE(AI&DS)

Div: B

Subject : ML(CL-I Lab)

Roll no. : BEAD21267

**Assignment No. - 4**

**Problem Statement** : Clustering Analysis (Any one)

1. Implement K-Means clustering on Iris.csv dataset. Determine the number of clusters using the elbow method. Dataset Link: <https://www.kaggle.com/datasets/uciml/iris>.

Code:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

import seaborn as sns

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

data

data.columns

for i,col in enumerate(data.columns):

print(f'Column number {1+i} is {col}')

data.dtypes

data.drop('Id', axis=1, inplace=True)

data.head()

data.isna().sum()

target\_data = data.iloc[:,4]

target\_data.unique()

clustering\_data = data.iloc[:,[0,1,2,3]]

clustering\_data.head()

fig, ax = plt.subplots(figsize=(15,7))

sns.set(font\_scale=1.5)

ax = sns.scatterplot(x=data['SepalLengthCm'],y=data['SepalWidthCm'], s=70, color='#f73434',

edgecolor='#f73434', linewidth=0.3)

ax.set\_ylabel('Sepal Width (in cm)')

ax.set\_xlabel('Sepal Length (in cm)')

plt.title('Sepal Length vs Width', fontsize = 20)

plt.show()

from sklearn.cluster import KMeans

wcss=[]

for i in range(1,11):

km = KMeans(i)

km.fit(clustering\_data)

wcss.append(km.inertia\_)

np.array(wcss)

kms = KMeans(n\_clusters=3, init='k-means++')

kms.fit(clustering\_data)

KMeans(n\_clusters=3)

clusters = clustering\_data.copy()

clusters['Cluster\_Prediction'] = kms.fit\_predict(clustering\_data)

clusters.head()

kms.cluster\_centers\_

import plotly.express as px

cluster0 = clusters[clusters['Cluster\_Prediction'] == 0]

cluster1 = clusters[clusters['Cluster\_Prediction'] == 1]

cluster2 = clusters[clusters['Cluster\_Prediction'] == 2]

fig = px.scatter(clusters, x='SepalLengthCm', y='SepalWidthCm', color='Cluster\_Prediction',

size\_max=30, opacity=0.7, title='Clusters', labels={'SepalLengthCm': 'Sepal Length (in cm)', 'SepalWidthCm': 'Sepal Width (in cm)'})

fig.add\_scatter(x=kms.cluster\_centers\_[:, 0], y=kms.cluster\_centers\_[:, 1],

mode='markers', marker=dict(size=20, color='yellow', line=dict(color='black', width=1)),

name='Centroids')

fig.for\_each\_trace(lambda t: t.update(name='Cluster ' + str(t.name)))

fig.update\_layout(legend\_title\_text='Clusters', xaxis\_range=[4, 8], yaxis\_range=[1.8, 4.5], xaxis\_title='Sepal Length (in cm)', yaxis\_title='Sepal Width (in cm)')

fig.show()

pip install nbformat>=4.2.0

Output:















