Aim:

program to implement K-means clustering technique using dataset world\_country\_and\_usa\_states\_latitude\_and\_longitude\_values.csv

Source Code:

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import sklearn

dataset = pd.read\_csv('world\_country\_and\_usa\_states\_latitude\_and\_longitude\_values.csv')

X = dataset.iloc[:, [1, 2]].values

print(X)

from sklearn.cluster import KMeans

wcss\_list = []

for i in range(1, 10):

kmeans = KMeans(n\_clusters = i, init = 'k-means++', random\_state = 42)

kmeans.fit(X)

wcss\_list.append(kmeans.inertia\_)

plt.plot(range(1, 10), wcss\_list)

plt.title('Elbow graph')

plt.xlabel('Number of clusters(k)')

plt.ylabel('wcss\_list')

plt.show()

kmeans = KMeans(n\_clusters = 5, init = "k-means++", random\_state = 42)

y\_predict = kmeans.fit\_predict(X)

print(y\_predict)

plt.scatter(X[y\_predict == 0, 0], X[y\_predict== 0, 1], s = 60, c = 'red', label = 'Cluster1')

plt.scatter(X[y\_predict == 1, 0], X[y\_predict == 1, 1], s = 60, c = 'blue', label = 'Cluster2')

plt.scatter(X[y\_predict == 2, 0], X[y\_predict == 2, 1], s = 60, c = 'green', label = 'Cluster3')

plt.scatter(X[y\_predict == 3, 0], X[y\_predict == 3, 1], s = 60, c = 'violet', label = 'Cluster4')

plt.scatter(X[y\_predict == 4, 0], X[y\_predict == 4, 1], s = 60, c = 'yellow', label = 'Cluster5')

plt.scatter(kmeans.cluster\_centers\_[:, 0], kmeans.cluster\_centers\_[:, 1], s = 100, c = 'black', label = 'Centroids')

plt.title('Country')

plt.xlabel('Longitude (k$)')

plt.ylabel('latitude (1-100)')

plt.legend()

plt.show()

Output:





