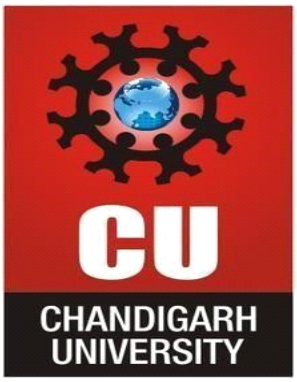
CHANDIGARH UNIVERSITY

UNIVERSITY INSTITUTE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



Submitted By: Rajiv Paul	**Submitted To:** Parveen Badoni
Subject Name	Machine Learning Lab
Subject Code	20CSP-317
Branch	BE-CSE
Semester	5th

LAB INDEX

Sr. No	Program	Date	Evaluation				Sign
			LW (12)	VV (8)	FW (10)	Total (30)	
1.1							
1.2							
1.3							
2.1							
2.2							
2.3							
2.4							
3.1	To Implement K-means clustering algorithm (cluster some sample data set into disjoint clusters using K-means).	26/10/22					
3.2							

3.3							

EXPERIMENT – 8

Student Name:Rajiv Paul **UID: 20BCS1812**
Branch:CSE **Section/Group: 20BCS_WM-702A**
Semester: 5th **Date of Performance: 26/10/2022**
Subject Name: Machine Learning Lab **Subject Code: 20CSP-317**

1. AIM OF THE EXPERIMENT:

To Implement K-means clustering algorithm (cluster some sample data set into disjoint clusters using K-means).

2. TASK TO BE DONE:

We will train a model using K-means and check its accuracy by plotting its graph.

3. PROGRAM CODE & OUTPUT:

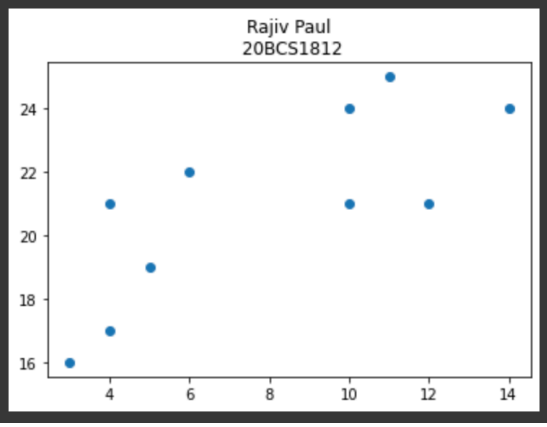
i) Importing libraries

- import numpy as np
- import pandas as pd
- import matplotlib.pyplot as plt
- from sklearn.cluster import KMeans
- import warnings
- warnings.filterwarnings('ignore')

ii) Creating our own data points to get a dataset.

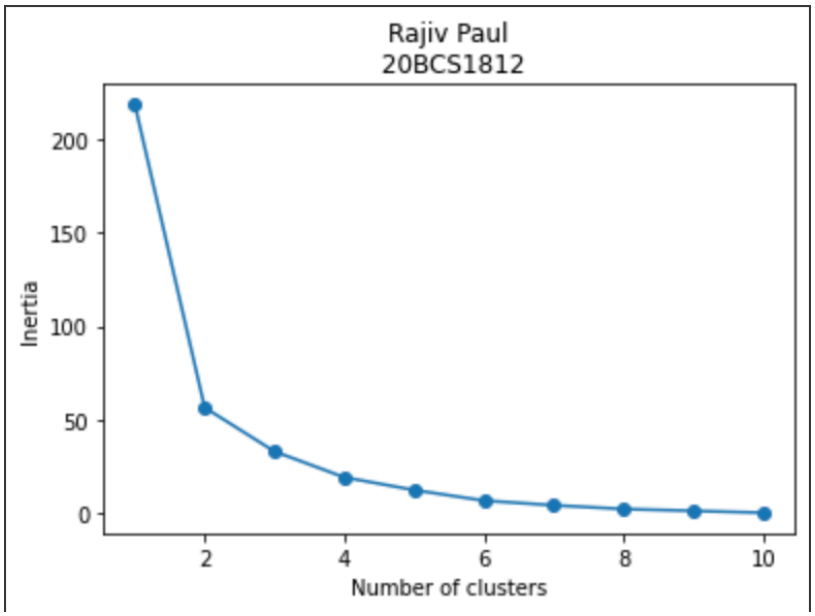
- x = [4, 5, 10, 40, 30, 11, 4 , 6, 1, 12,21,41,22,32,19]
- y = [21, 19, 24, 17, 16, 25, 24, 2, 21, 21,28,35,41,20,18]
- plt.scatter(x, y)
- plt.show()

```
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
x = [4, 5, 10, 4, 3, 11, 14 , 6, 10, 12]
y = [21, 19, 24, 17, 16, 25, 24, 22, 21, 21]
plt.scatter(x, y)
plt.title('Rajiv Paul\n 20BCS1812')
plt.show()
```



iii) Visualizing our dataset by plotting a graph.

- data = list(zip(x, y))
- inertias = []
- for i in range(1,11):
- kmeans = KMeans(n_clusters=i)
- kmeans.fit(data)
- inertias.append(kmeans.inertia_)
-
- plt.plot(range(1,11), inertias, marker='o')
- plt.title('Elbow method')
- plt.xlabel('Number of clusters')
- plt.ylabel('Inertia')
- plt.show()



iv) Fitting our training data in Kmeans to train the model and generating predictions.

- kmeans = KMeans(n_clusters=3)
- kmeans.fit(data)
- print(kmeans.labels_)

```
[ ]  kmeans = KMeans(n_clusters=2)
     kmeans.fit(data)
     print(kmeans.labels_)

     [0 0 1 0 0 1 1 0 1 1]
```

v) Score values of Kmeans.

- print(kmeans.cluster_centers_)

```
print(kmeans.cluster_centers_)

[[ 4.4 19. ]
 [11.4 23. ]]
```

vi) Visualizing predictions by plotting a graph.

- plt.scatter(x, y, c=kmeans.labels_)

- plt.show()