Experiment 2.3

Student's Name: Utkarsh Joshi **UID: 21BCS9158**
Branch: CSE **Section/Group: 802-A**
Semester: 5th **Date of Performance: 25-10-2023**
Subject Name: AI &ML with Lab **Subject Code: 21CSH-316**

1. **Aim:** To determine the optimal number of clusters (K) using K-means clustering algorithm

2. **Objective:** The objective is to assess how well the optimal number of clusters (K) using K-means clustering algorithm, and to analyze its effectiveness in comparison to other algorithms or approaches.

3. **Input/Apparatus Used**:

   Google collab and python libraries.

4. **Hardwire Requirements:**

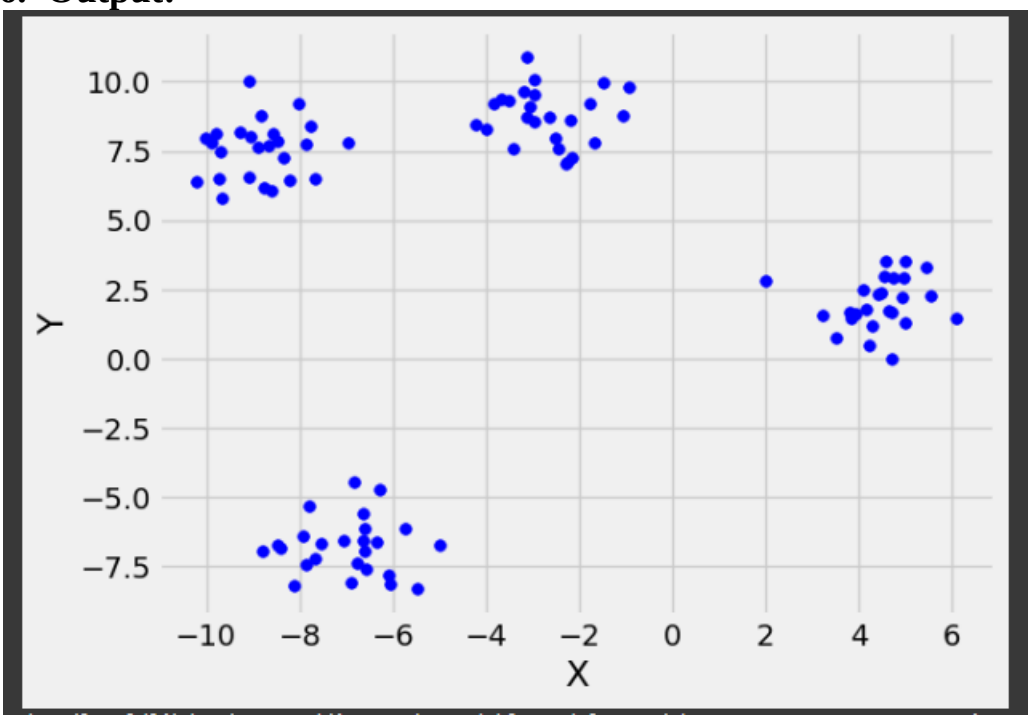
   Computer/Laptop minimum 4GB, windows and Power Supply

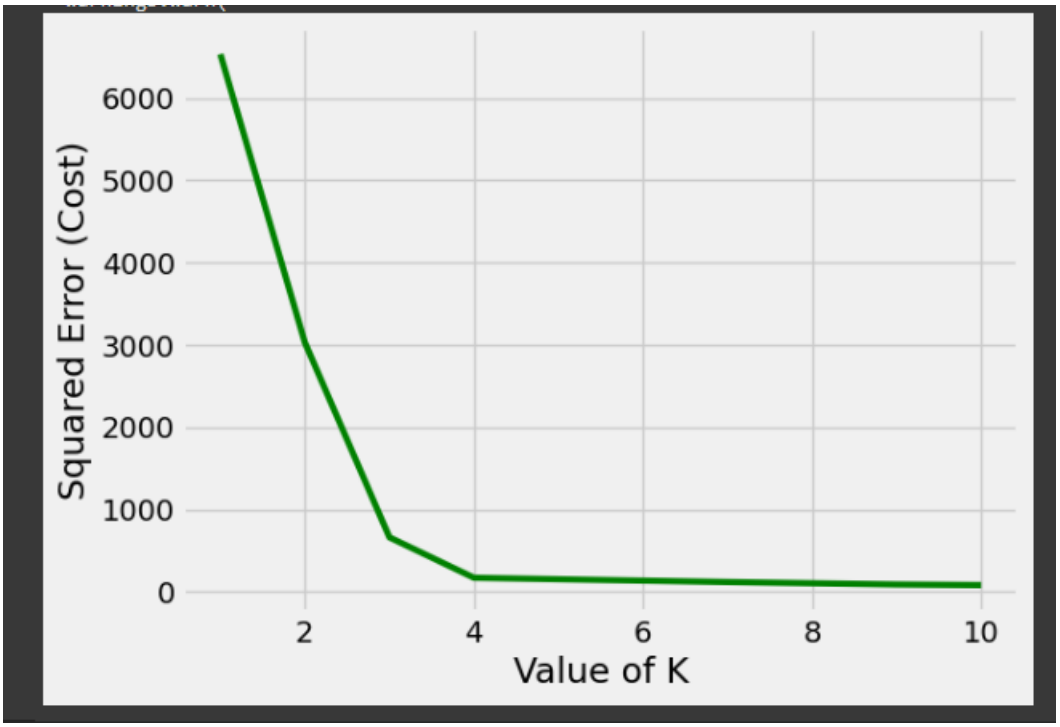
5. **Code:**

```
import matplotlib.pyplot as plt
from matplotlib import style
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs

style.use("fivethirtyeight")

# make_blobs() is used to generate sample points
# around c centers (randomly chosen)
X, y = make_blobs(n_samples=100, centers=4, cluster_std=1, n_features=2,
    random_state=42)

plt.scatter(X[:, 0], X[:, 1], s=30, color='b')

# label the axes
plt.xlabel('X')
plt.ylabel('Y')
```

```
plt.show()
plt.clf()  # clear the figure

cost = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, max_iter=500, random_state=42)
    kmeans.fit(X)

    # calculates squared error
    # for the clustered points
    cost.append(kmeans.inertia_)

# plot the cost against K values
plt.plot(range(1, 11), cost, color='g', linewidth=3)
plt.xlabel("Value of K")
plt.ylabel("Squared Error (Cost)")
plt.show()  # clear the plot

# The point of the elbow is the
    # most optimal value for choosing k
```

6. Output:





7. Learning Outcomes:

1. Understand how to implement the K-Means algorithm
2. Implement The optimal number of clusters (K) using K-means clustering algorithm