```
#6B Clustering algorithms for unsupervised classification.
import pandas as pd
import matplotlib.pyplot as plt
#Jupyter Notebooks to show the plots
%matplotlib inline
```

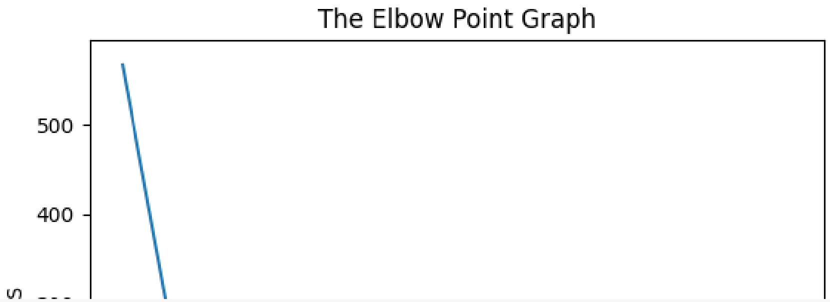
```
#Importing the dataset
iris = pd.read_csv("/content/drive/MyDrive/KRAI/iris_csv.csv")
```

```
iris_clustering = iris.drop(columns = ['class'])
X = iris_clustering.iloc[:, [0,2]].values
X
```

```
       [7.4, 6.1],
       [7.9, 6.4],
       [6.4, 5.6],
       [6.3, 5.1],
       [6.1, 5.6],
       [7.7, 6.1],
       [6.3, 5.6],
       [6.4, 5.5],
       [6. , 4.8],
       [6.9, 5.4],
       [6.7, 5.6],
       [6.9, 5.1],
       [5.8, 5.1],
       [6.8, 5.9],
       [6.7, 5.7],
       [6.7, 5.2],
       [6.3, 5. ],
       [6.5, 5.2],
       [6.2, 5.4],
       [5.9, 5.1]])
```

```
from sklearn.cluster import KMeans
import warnings

warnings.filterwarnings('ignore', category=FutureWarning)
wcss = []
for i in range(1, 11):
   kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
   kmeans.fit(X)
   wcss.append(kmeans.inertia_)
#Plotting The Elbow graph
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Point Graph')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



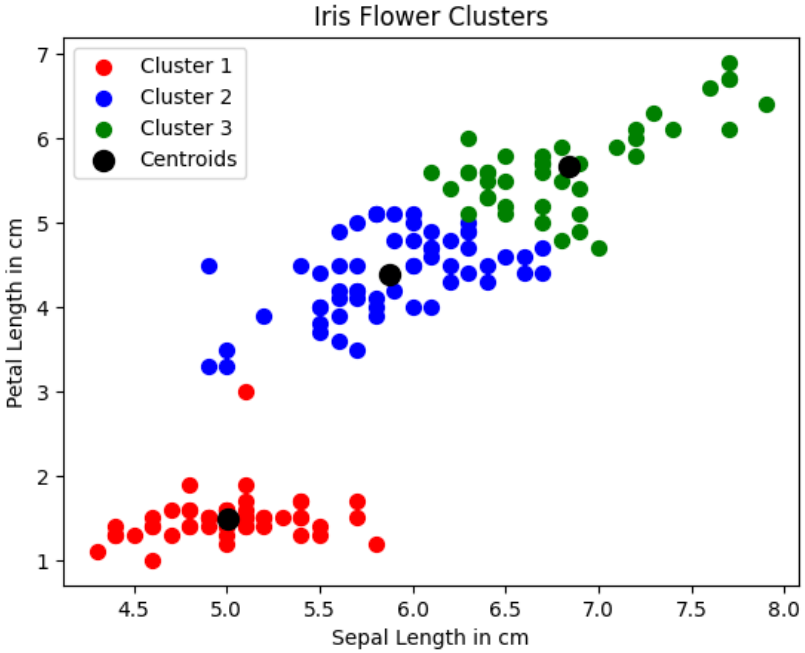

```
#Initialising K-Means With Optimum Number Of Clusters
kmeans = KMeans(n_clusters = 3, init = 'k-means++', random_state = 0)
y = kmeans.fit_predict(X)
print(y)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 2 1 2 2 2 2 1 2 2 2 2
 2 2 1 1 2 2 2 2 1 2 1 2 1 2 2 1 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 1 2
 2 1]
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_
  warnings.warn(
```

```
plt.scatter(X[y == 0, 0], X[y == 0, 1], s = 50, c = 'red', label = 'Cluster 1')
plt.scatter(X[y == 1, 0], X[y == 1, 1], s = 50, c = 'blue', label = 'Cluster 2')
plt.scatter(X[y == 2, 0], X[y == 2, 1], s = 50, c = 'green', label = 'Cluster 3')

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

plt.title('Iris Flower Clusters')
plt.xlabel('Sepal Length in cm')
plt.ylabel('Petal Length in cm')
plt.legend()
plt.show()
```


Iris Flower Clusters
Cluster 1
Cluster 2
Cluster 3
Centroids
Petal Length in cm
Sepal Length in cm
1
2
3
4
5
6
7
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0