KDDM Lab9

Name :- Kunal Sanjay Patil

PRN :- 20190802025

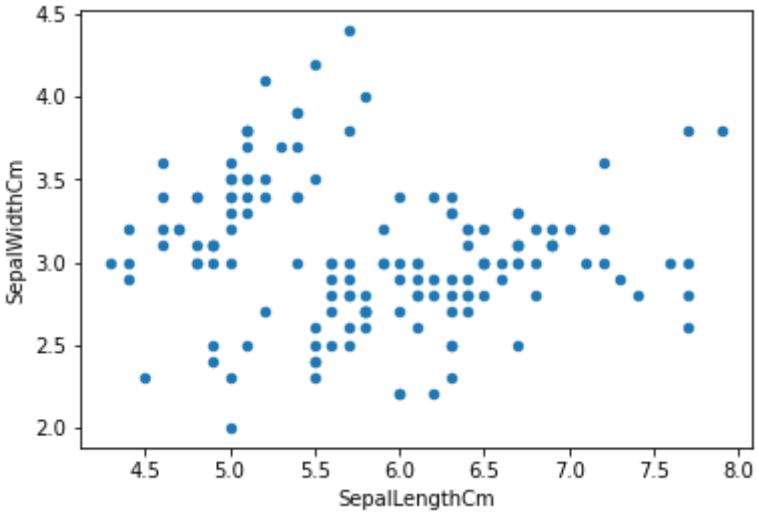
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
iris =   pd.read_csv("Iris.csv")
print(iris.head())
```

```
   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm      Species
0   1            5.1           3.5            1.4           0.2  Iris-setosa
1   2            4.9           3.0            1.4           0.2  Iris-setosa
2   3            4.7           3.2            1.3           0.2  Iris-setosa
3   4            4.6           3.1            1.5           0.2  Iris-setosa
4   5            5.0           3.6            1.4           0.2  Iris-setosa
```

```
print(iris.describe())
```

```
               Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
count  150.000000     150.000000    150.000000     150.000000    150.000000
mean    75.500000       5.843333      3.054000       3.758667      1.198667
std     43.445368       0.828066      0.433594       1.764420      0.763161
min      1.000000       4.300000      2.000000       1.000000      0.100000
25%     38.250000       5.100000      2.800000       1.600000      0.300000
50%     75.500000       5.800000      3.000000       4.350000      1.300000
75%    112.750000       6.400000      3.300000       5.100000      1.800000
max    150.000000       7.900000      4.400000       6.900000      2.500000
```

```
iris.plot(kind="scatter", x="SepalLengthCm",   y="SepalWidthCm")
plt.show()
```




```
from sklearn.datasets import load_iris
from sklearn.cluster import KMeans
```

```
iris_data=load_iris()   #loading iris dataset from sklearn.datasets
```

```
iris_df = pd.DataFrame(iris_data.data, columns = iris_data.feature_names) #creating datafr
```

```
kmeans = KMeans(n_clusters=3,init = 'k-means++',   max_iter = 100, n_init = 10, random_sta
```

```
x = iris.iloc[:, :-1].values #last column values excluded
```

```
y = iris.iloc[:,   -1].values #last column value
```

```
y_kmeans = kmeans.fit_predict(x)
```

```
print(kmeans.cluster_centers_) #display cluster centers
```

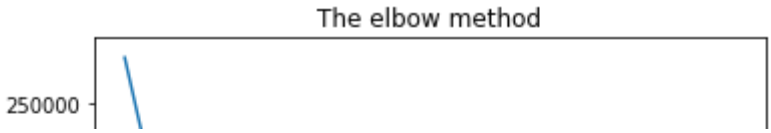
```
     [[ 25.5     5.006   3.418   1.464   0.244]
      [125.5     6.588   2.974   5.552   2.026]
      [ 75.5     5.936   2.77    4.26    1.326]]
```

```
kmeans = KMeans(n_clusters = 3, init = 'k-means++', max_iter = 300, n_init = 10, random_st
y_kmeans = kmeans.fit_predict(x)
```

```
#Finding the optimum number of clusters for k-means classification
from sklearn.cluster import KMeans
wcss = []

for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init = 10, randc
    kmeans.fit(x)
    wcss.append(kmeans.inertia_)


plt.plot(range(1, 11), wcss)
plt.title('The elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS') #within cluster sum of squares
plt.show()
```

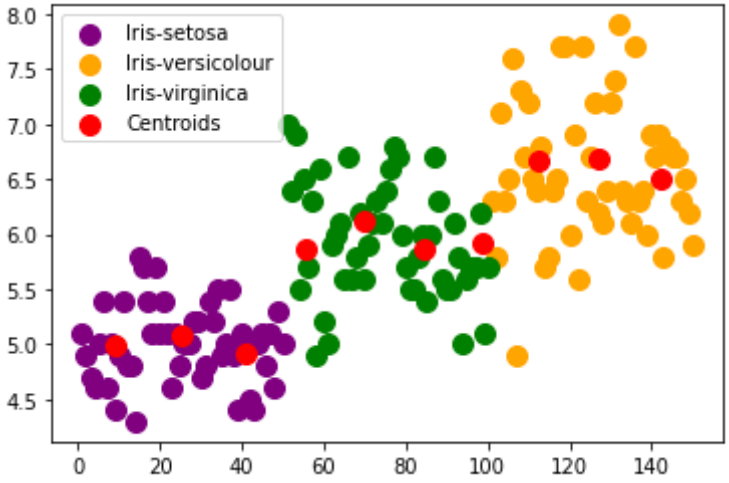



```
#Visualising the clusters
plt.scatter(x[y_kmeans == 0, 0], x[y_kmeans == 0, 1], s = 100, c = 'purple', label = 'Iris
plt.scatter(x[y_kmeans == 1, 0], x[y_kmeans == 1, 1], s = 100, c = 'orange', label = 'Iris
plt.scatter(x[y_kmeans == 2, 0], x[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Iris-

#Plotting the centroids of the clusters
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:,1], s = 100, c = 'red

plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f0730404d50>
```



✓ 0s completed at 10:41 PM