

▼ Knowledge Discovery & Data Mining Lab-09

Name: Gurvinder Kaur Matharu


PRN: 20190802077

▼ AIM:

To implement K-Means clustering using python.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score
```


```
data = pd.read_csv('IRIS.csv')
data.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species	
0	5.1	3.5	1.4	0.2	Iris-setosa	
1	4.9	3.0	1.4	0.2	Iris-setosa	
2	4.7	3.2	1.3	0.2	Iris-setosa	
3	4.6	3.1	1.5	0.2	Iris-setosa	
4	5.0	3.6	1.4	0.2	Iris-setosa	

```
data.shape
```

```
(150, 5)
```

```
data.describe()
```

	sepal_length	sepal_width	petal_length	petal_width	
count	150.000000	150.000000	150.000000	150.000000	
mean	5.843333	3.054000	3.758667	1.198667	
std	0.828066	0.433594	1.764420	0.763161	
min	4.300000	2.000000	1.000000	0.100000	
25%	5.100000	2.800000	1.600000	0.300000	
50%	5.800000	3.000000	4.350000	1.300000	
75%	6.400000	3.300000	5.100000	1.800000	
max	7.900000	4.400000	6.900000	2.500000	

```
data.info()
```

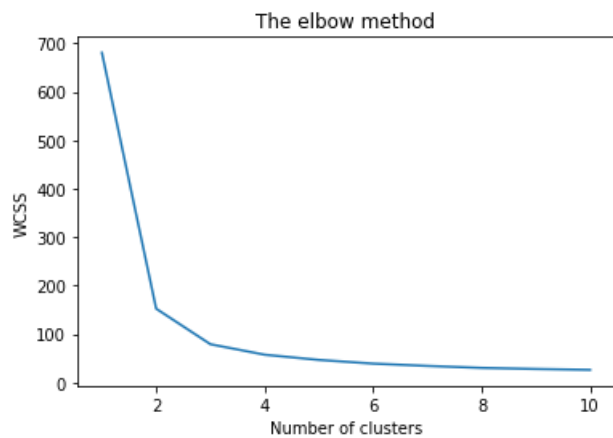
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   sepal_length    150 non-null   float64
1   sepal_width     150 non-null   float64
2   petal_length    150 non-null   float64
```

```
3   petal_width    150 non-null    float64
4   species        150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
x = data.iloc[:, [0,1,2,3]].values
y = data.iloc[:,4].values
```

```
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
    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()
```

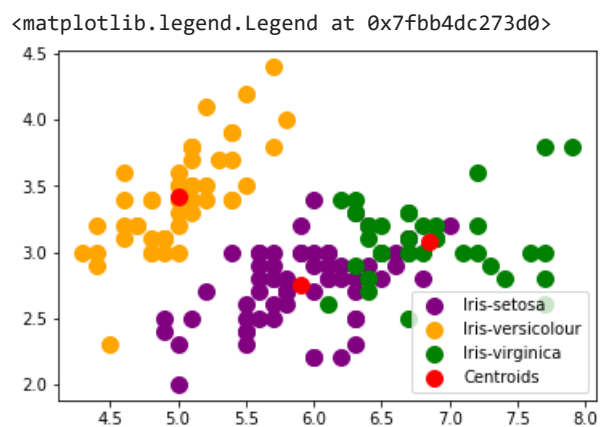


```
kmeans = KMeans(n_clusters = 3, init = 'k-means++', random_state = 0)
y_kmeans = kmeans.fit_predict(x)
```

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

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

plt.legend()
```



✓ 0s completed at 8:26 PM

