TASK 2: PREDICTION USING UNSUPERVISED ML

AUTHOR: VANIPENTA BALAJI GRIP JUNE 2023 THE SPARKS FOUNDATIONS

The Task is, From the given 'Iris' dataset, predict the optimum no.of clusters & represent it visually

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import datasets
```

In [2]:

```
# Loading the Iris dataset
iris = datasets.load_iris()
iris_df = pd.DataFrame(iris.data, columns = iris.feature_names)
iris_df.head()
```

Out[2]:

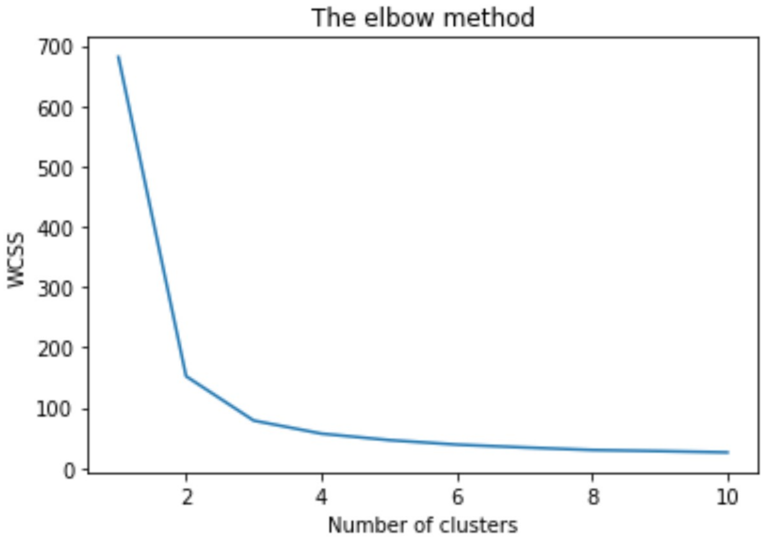
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

In [3]:

```
#Finding the optimum number of clusters for k-means classification

x = iris_df.iloc[:, [0, 1, 2, 3]].values

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, 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')

# Within cluster sum of squares
plt.ylabel('WCSS')
plt.show()
```

```
C:\Users\Vanipenta Balaji\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:8
81: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when t
```

```
here are less chunks than available threads. You can avoid it by setting the envir
onment variable OMP_NUM_THREADS=1.
  warnings.warn(
```



In [4]:

```
#From this we choose the no.of clusters as "3"

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

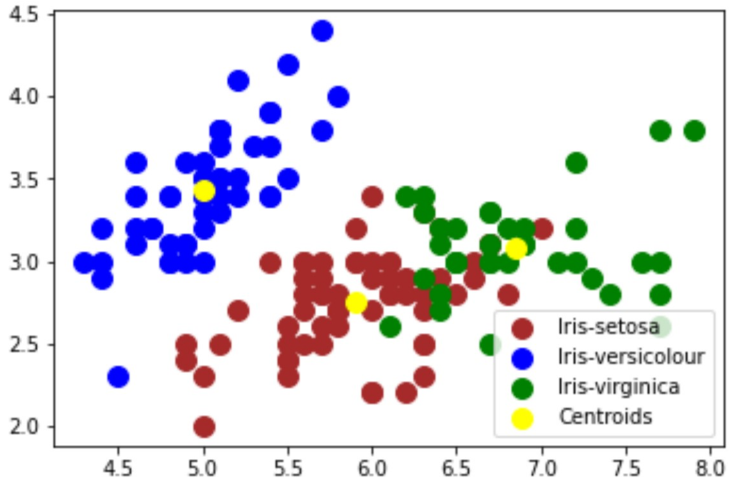
Visualising the clusters

In [6]:

```
# Visualising the clusters - On the first two columns

plt.scatter(x[y_kmeans == 0, 0], x[y_kmeans == 0, 1], s = 100, c = 'brown', label
plt.scatter(x[y_kmeans == 1, 0], x[y_kmeans == 1, 1], s = 100, c = 'blue', label =
plt.scatter(x[y_kmeans == 2, 0], x[y_kmeans == 2, 1], s = 100, c = 'green', label

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

plt.legend()
```

Out[6]:

```
<matplotlib.legend.Legend at 0x2c5b741c850>
```



```
In [ ]:
```