Name- Rutuja Sohani

| The Sparks Foundation |

Data Science and Business Analytics

Task-2:Predict The Optimum Number Of Clusters And Represent Visually For The Iris Data Set.

Language:Python

Dataset Link:http://bit.ly/3kXTdox

In [37]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import datasets
from sklearn.cluster import KMeans
import warnings as wg
wg.filterwarnings('ignore')
```

In [38]:
```
Iris = datasets.load_iris()
Iris_df = pd.DataFrame(Iris.data, columns = Iris.feature_names)
Iris_df.head()
```

Out[38]:

	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

Exploring Data

In [39]:
```
Iris_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 4 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   sepal length (cm)  150 non-null    float64
 1   sepal width (cm)   150 non-null    float64
 2   petal length (cm)  150 non-null    float64
 3   petal width (cm)   150 non-null    float64
dtypes: float64(4)
memory usage: 4.8 KB
```

In [40]:
```
Iris_df.describe()
```

Out[40]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
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

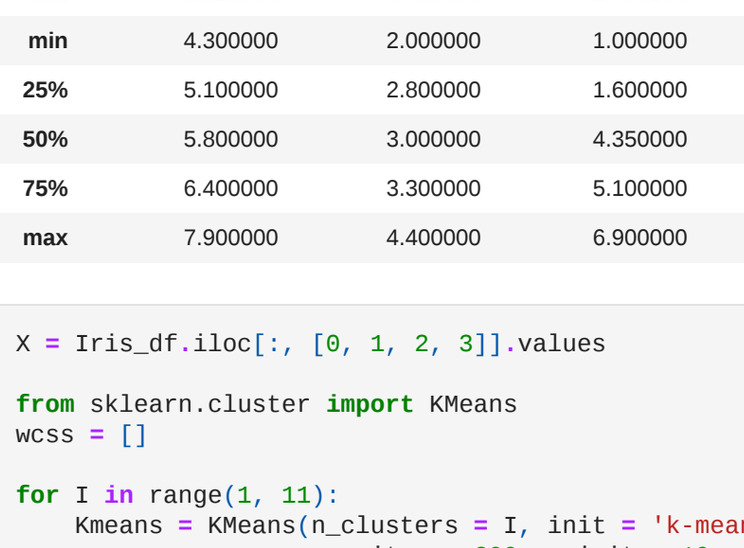
In [45]:
```
X = Iris_df.iloc[:, [0, 1, 2, 3]].values

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_)
print(wcss)
```

```
[681.3705999999996, 152.34795176035797, 78.851441426146, 57.25600931571815, 46.44618205128204, 39.03998724608725, 34.299712121212146, 30.014398496240617, 28.0
3690635345049, 26.534529220779234]
```

In [47]:
```
plt.plot(range(1, 11),wcss)
plt.title('The Elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

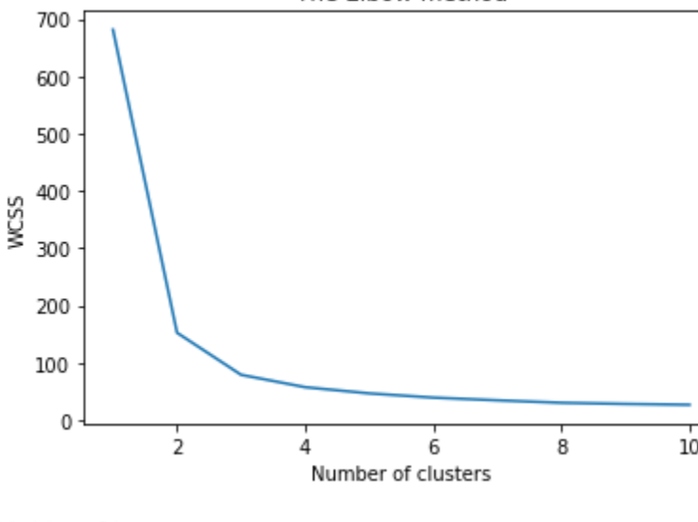


Making Clusters

In [48]:
```
Kmeans = KMeans(n_clusters = 3, init = 'k-means++',
max_iter = 300, n_init = 10, random_state = 0)
Y_Kmeans = Kmeans.fit_predict(X)
```

In [49]:
```
plt.scatter(X[Y_Kmeans == 0, 0], X[Y_Kmeans == 0, 1],
            s = 100, c = 'red', 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 = 'yellow', label = 'Iris-virginica')
```

Out[49]: <matplotlib.collections.PathCollection at 0x3c9a9d0>



Thank You