

# THE SPARKS FOUNDATION

## Data Science and Business Analytic

Name : URVI DALAL

## Task 2 : Prediction using Unsupervised ML

### Objective :-

From the given 'Iris' dataset, predict the optimum number of clusters and represent it visually.

### K-Means Clustering

#### Importing all libraries required in this notebook

```
In [3]: 1 # Importing the Libraries
        2 import numpy as np
        3 import matplotlib.pyplot as plt
        4 import seaborn as sas
        5 import pandas as pd
        6 from sklearn import datasets
```

#### Importing Dataset


```
In [4]: 1 iris = datasets.load_iris()
        2 df = pd.DataFrame(iris.data, columns = iris.feature_names)
```

In [5]:    
 1 *#Reading Dataset*  
 2 df


Out[5]:

	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
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns


In [6]:    
 1 *#checking the shape of the dataset*  
 2 df.shape

Out[6]: (150, 4)

In [7]:    
 1 *#Reading the first 10 observation*  
 2 df.head(10)

Out[7]:

	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
5	5.4	3.9	1.7	0.4
6	4.6	3.4	1.4	0.3
7	5.0	3.4	1.5	0.2
8	4.4	2.9	1.4	0.2
9	4.9	3.1	1.5	0.1

In [8]:    
 1 *#Reading the last 10 observation*  
 2 `df.tail(10)`


Out[8]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
140	6.7	3.1	5.6	2.4
141	6.9	3.1	5.1	2.3
142	5.8	2.7	5.1	1.9
143	6.8	3.2	5.9	2.3
144	6.7	3.3	5.7	2.5
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

In [9]:    
 1 *#Checking Numerical Data*  
 2 `df.describe()`

Out[9]:

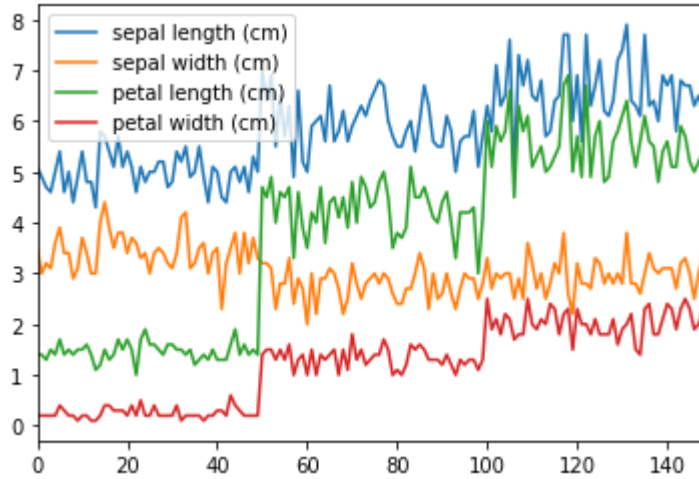
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [11]:    
 1 *#Checking the null value in the dataset*  
 2 `df.isnull().sum()`

Out[11]: sepal length (cm) 0  
 sepal width (cm) 0  
 petal length (cm) 0  
 petal width (cm) 0  
 dtype: int64

```
In [12]: 1 #plotting feature in line graph
        2 df.plot(kind = 'line')
```

Out[12]: <matplotlib.axes.\_subplots.AxesSubplot at 0x22590a95588>



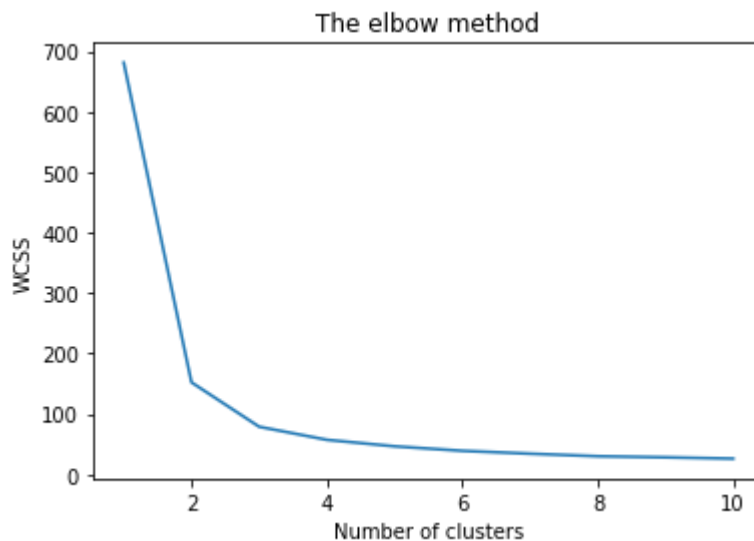
## Optimum number of cluster for k-means classification

```
In [13]: 1 x = df.iloc[:, [0, 1, 2, 3]].values
```

## Using the Elbow Method to find the optimum number of clusters

```
In [14]: 1 from sklearn.cluster import KMeans
```

```
In [15]: 1 wcss = []
2
3 for i in range(1, 11):
4     kmeans = KMeans(n_clusters = i, init = 'k-means++',
5                     max_iter = 300, n_init = 10, random_state = 0)
6     kmeans.fit(x)
7     wcss.append(kmeans.inertia_)
8
9 # Plotting the results onto a line graph,
10 # `allowing us to observe 'The elbow'
11 plt.plot(range(1, 11), wcss)
12 plt.title('The elbow method')
13 plt.xlabel('Number of clusters')
14 plt.ylabel('WCSS') # Within cluster sum of squares
15 plt.show()
```



You can clearly see why it is called 'The elbow method' from the above graph, the optimum clusters is where the elbow occurs. This is when the within cluster sum of squares (WCSS) doesn't decrease significantly with every iteration.

The maximum number of clusters that can be formed is 3 as observed from the Elbow Method

## Applying K-Means to the dataset

```
In [16]: 1 kmeans = KMeans(n_clusters = 3, init = 'k-means++',
2                     max_iter = 300, n_init = 10, random_state = 0)
3 y_kmeans = kmeans.fit_predict(x)
```



