# Name: Sharad Parmar

## Task 2 : To explore UnSupervised Machine Learning.

From the given 'Iris' dataset, predict the optimum number of clusters(K-Means) and represent it visually.

## UnSupervised Learning:

Unsupervised learning is the training of machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. Here the task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data.

## Clustering:

A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.

## K-Means Clustering:

It is the simplest unsupervised learning algorithm that solves clustering problem.K-means algorithm partition n observations into k clusters where each observation belongs to the cluster with the nearest mean serving as a prototype of the cluster

# LOADING THE LIBRARIES

In [1]:

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.datasets import load_iris
```

In [2]:

```
iris=load_iris()
```

In [6]:

```
iris['feature_names']
```

Out[6]:

```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

In [7]:

```
data=pd.DataFrame(data=np.c_[iris['data'],iris['target']],columns=iris[
'feature_names']+['target'])
data.head()
```

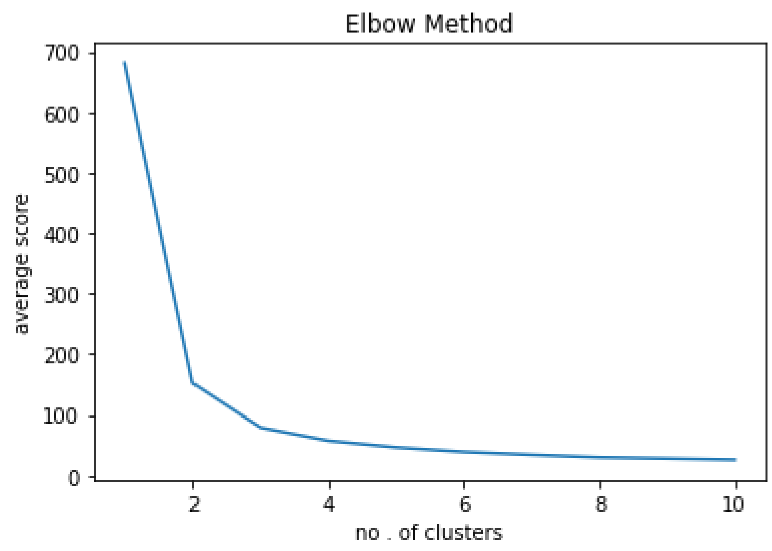
Out[7]:

| | sepal length (cm) | sepal width (cm) | petal length (cm) | petal width (cm) | target |
|---|---|---|---|---|---|
| 0 | 5.1 | 3.5 | 1.4 | 0.2 | 0.0 |
| 1 | 4.9 | 3.0 | 1.4 | 0.2 | 0.0 |
| 2 | 4.7 | 3.2 | 1.3 | 0.2 | 0.0 |
| 3 | 4.6 | 3.1 | 1.5 | 0.2 | 0.0 |
| 4 | 5.0 | 3.6 | 1.4 | 0.2 | 0.0 |

# USING ELBOW METHOD TO FIND NUMBER OF CLUSTERS FOR KMEANS

In [8]:

```
sse=[]
for n in range(1,11):
    kmean=KMeans(n_clusters=n)
    kmean.fit(iris['data'])
    sse.append(kmean.inertia_)
#Now plotiing the results on the line graph
from matplotlib import pyplot as plt
plt.plot(range(1,11),sse)
plt.title('Elbow Method')
plt.xlabel('no . of clusters')
plt.ylabel('average score')
plt.show()
```



In [9]:

```
#we willcreate k-mean classifier
model=KMeans(n_clusters=3)
#fitting iris data
model.fit(iris['data'])
```

Out[9]:

```
KMeans(n_clusters=3)
```

In [10]:

```
#prediction of datset
data.target=model.labels_
d1=data[data.target==0]
d2=data[data.target==1]
d3=data[data.target==2]
```

In [11]:

```
model.cluster_centers_
```

Out[11]:

```
array([[5.9016129 , 2.7483871 , 4.39354839, 1.43387097],
       [5.006     , 3.428     , 1.462     , 0.246     ],
       [6.85      , 3.07368421, 5.74210526, 2.07105263]])
```

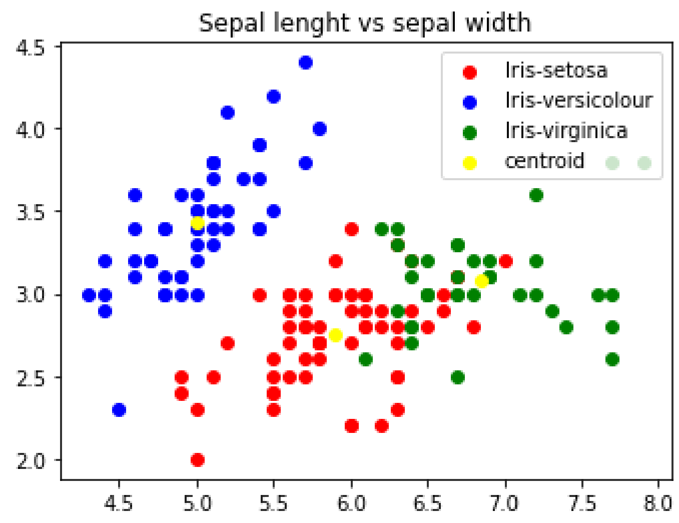
# plotting graph for iris data

In [12]:

```
plt.figure(figsize=(12,4))
plt.subplot(1,2,1)
plt.scatter(d1[['sepal length (cm)']],d1[['sepal width (cm)']], color =
'red', label = 'Iris-setosa')
plt.scatter(d2[['sepal length (cm)']], d2[['sepal width (cm)']], color =
'blue', label = 'Iris-versicolour')
plt.scatter(d3[['sepal length (cm)']], d3[['sepal width (cm)']], color =
'green', label = 'Iris-virginica')
# Plotting the centroids of the clusters
plt.scatter(model.cluster_centers_[:, [0]], model.cluster_centers_[:,[1
]], color = 'yellow', label = 'centroid')
plt.title('Sepal lenght vs sepal width')
plt.legend()
```

Out[12]:

```
<matplotlib.legend.Legend at 0x1d826a30>
```



**Thank you**