THE SPARKS FOUNDATION

#GRIPAPR'22

AUTHOR : Devarapalli Ruthwik Reddy

TASK-2 Prediction using unsupervised ML

In this task by using "Iris" dataset, we need to predict the optimal number of clusters and represent it visually.

Importing the required libraries

In [79]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings as wg
wg.filterwarnings("ignore")
```

Import the dataset

In [80]:

```
Iris=pd.read_csv("C:/Users/ruthwik/Downloads/Iris.csv")
```

In [81]:

```
Iris
```

Out[81]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [82]:

```
Iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             150 non-null    int64  
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object 
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

In [83]:

```
Iris.describe()
```

Out[83]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

In [84]:

```
Iris.columns
```

Out[84]:

```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species'],
      dtype='object')
```

In [100…

```
#Removing Id and Species columns.
Iris_df=Iris.iloc[:,[1,4]].values
```

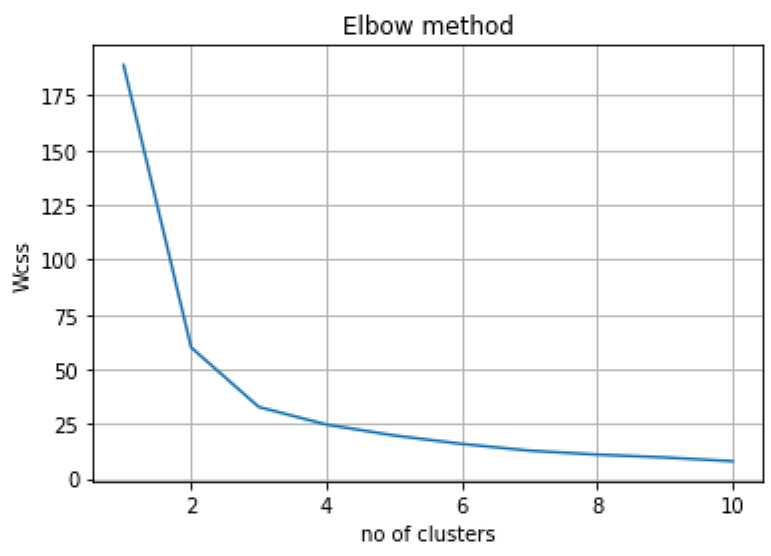
elbow method to calculate optimal number of clusters

In [101…

```
from sklearn.cluster import KMeans
wcss=[]
for i in range(1,11):
    kmeans=KMeans(n_clusters=i,init='k-means++',random_state=0)
    kmeans.fit(Iris_df)
    wcss.append(kmeans.inertia_)
```

In [102…

```
plt.plot(range(1,11),wcss)
plt.title("Elbow method")
plt.xlabel("no of clusters")
plt.ylabel("Wcss")
plt.grid()
plt.show()
```



from above plot we can consider taking no of clusters=3.

Kmeans clustering

In [103…

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

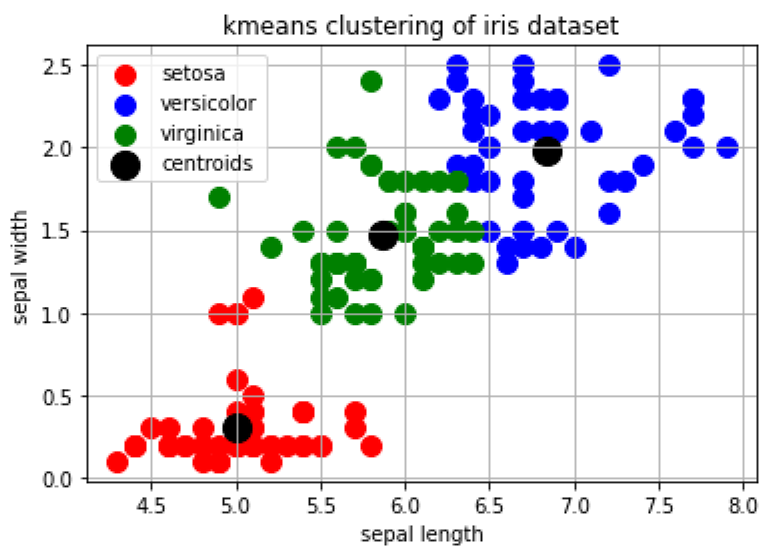
In [104…

```
print(y_kmeans)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 1 2 1 2 2 0 1 2 0 2 2 2 2 1 2 2 2 2 2 2 2 2
 2 1 1 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 0 2 2 2 2 0 2 1 2 1 2 1 1 2 1 1 1 1
 1 1 2 2 1 1 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 1 2 2 1 1 1 2 1 1 1 2 1 1 1 1 1
 1 2]
```

In [111…

```
plt.scatter(Iris_df[y_kmeans==0,0],Iris_df[y_kmeans==0,1],s=100,c='red',label='setosa')
plt.scatter(Iris_df[y_kmeans ==1,0],Iris_df[y_kmeans ==1,1], s=100, c='blue', label='versicolor')
plt.scatter(Iris_df[y_kmeans ==2,0],Iris_df[y_kmeans ==2,1], s=100, c='green', label='virginica')
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],s=200,c='black',label='centroids')
plt.title("kmeans clustering of iris dataset")
plt.xlabel("sepal length")
plt.ylabel("sepal width")
plt.grid()
plt.legend()
plt.show()
```



Conclusion:

The optimal number of clusters needed = 3.