Prediction using Unsupervised ML

by : Nidisha Mandlik

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

Data Source : https://bit.ly/3kXTdox

Importing all the libraries that required for this project.

In [20]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from sklearn import datasets
```

Load the iris dataset

In [47]:

```
df = pd.read_csv(r'C:\Users\HP\Desktop\Iris.csv')
df.head(5)
```

Out[47]:

	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

In [48]:

```
df.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 [49]:

```
df.describe()
```

Out[49]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm

	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

Elbow Method

Finding the optimum number of clusters for k-means classification.

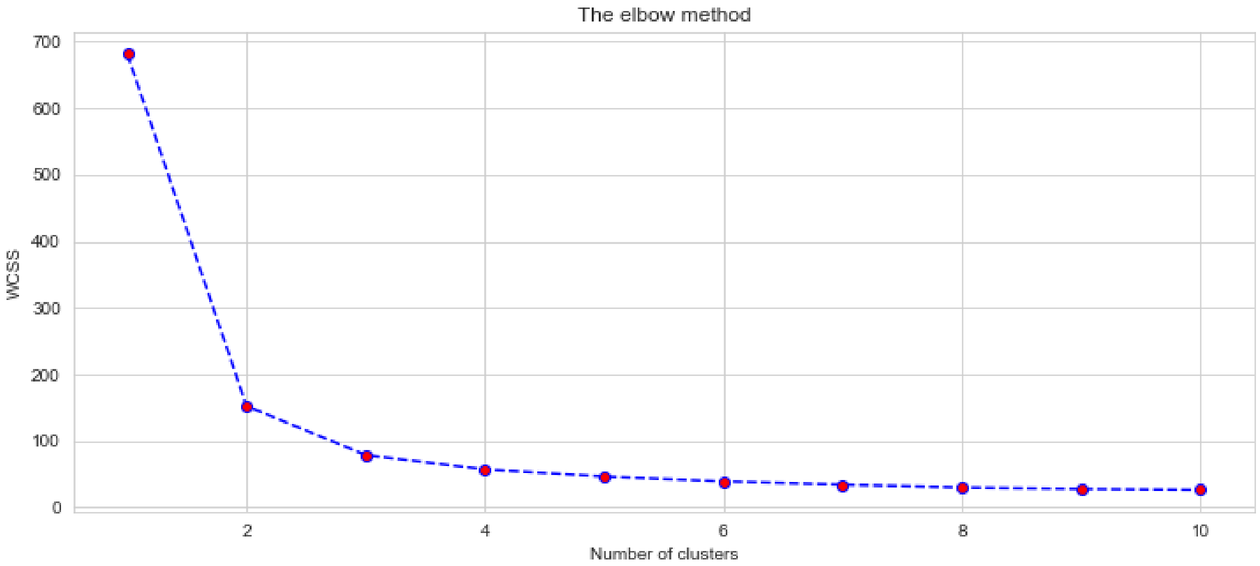
In [50]:

```
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_)
```

Plotting the graph of elbow method

In [51]:

```
plt.figure(figsize=(12,5))
plt.plot(range(1, 11), wcss,color='blue',linestyle='--',marker='o',
        markerfacecolor='red')
plt.title('The elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS') # Within cluster sum of squares
plt.show()
```



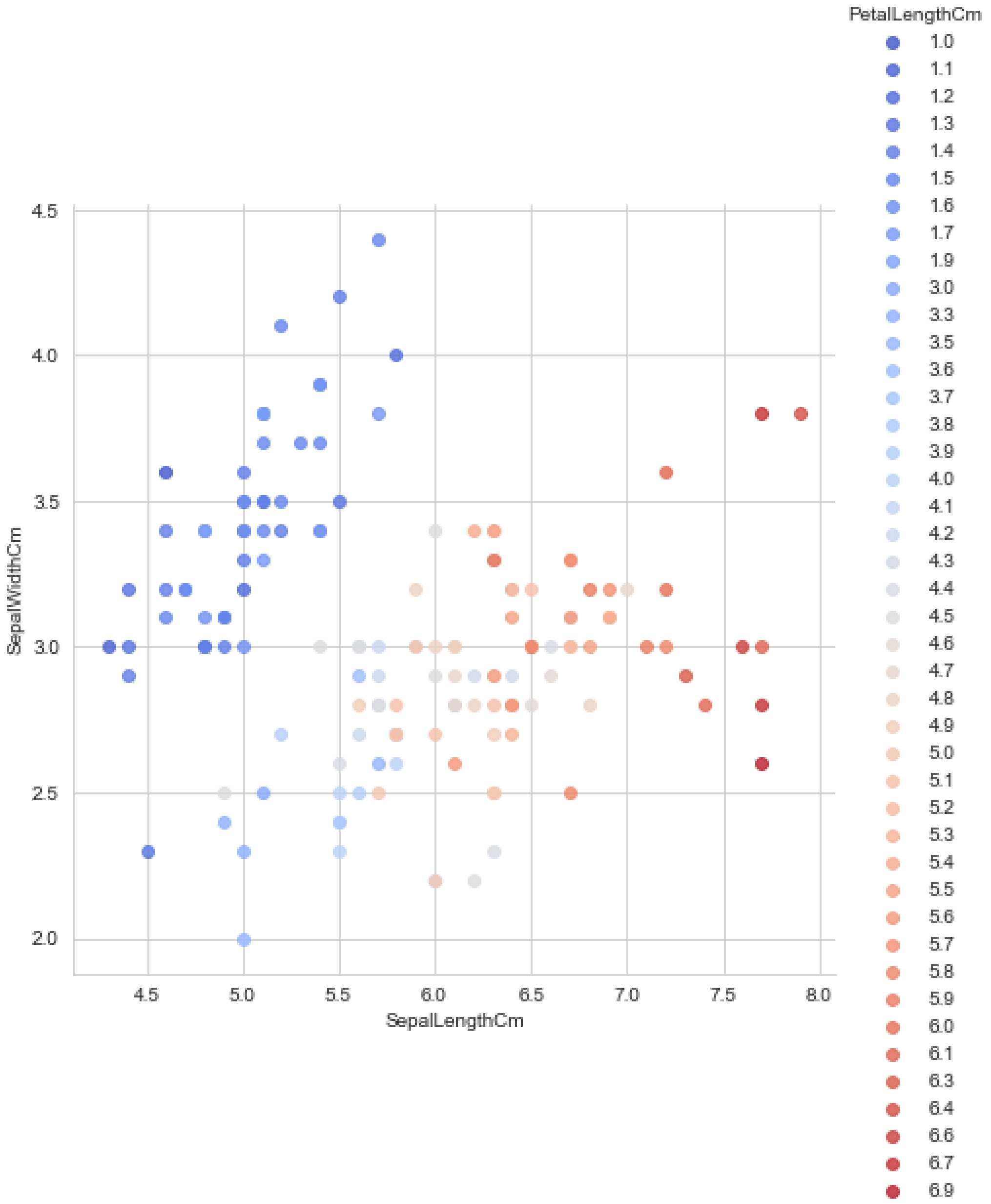
Visualising the clusters.

In [71]:

```
sns.set_style('whitegrid')
sns.lmplot(x='SepalLengthCm',y='SepalWidthCm',data=df, hue='PetalLengthCm',
           palette='coolwarm',height=6,aspect=1,fit_reg=False)
```

Out[71]:

```
<seaborn.axisgrid.FacetGrid at 0x1700905b788>
```



In [54]:

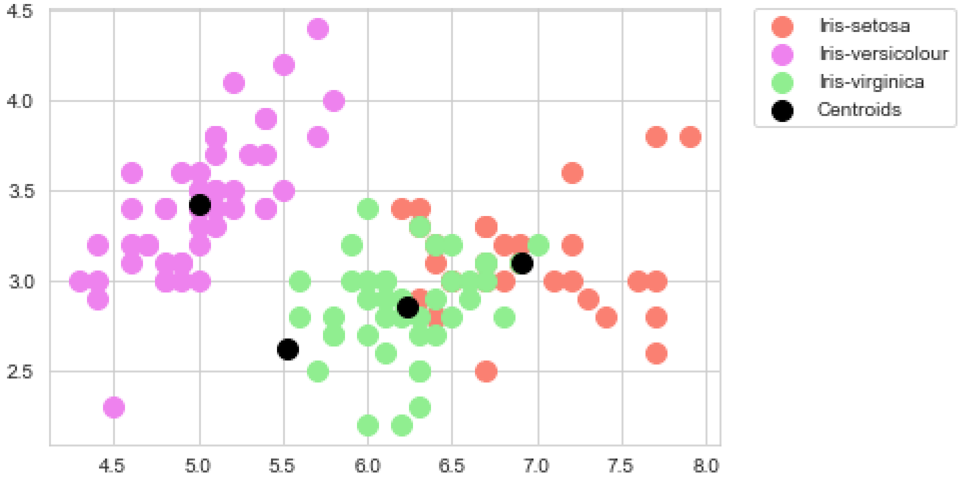
```
kmeans = KMeans(n_clusters = 4, init = 'k-means++',
                max_iter = 300, n_init = 10, random_state = 0)
y_kmeans = kmeans.fit_predict(x)
```

In [78]:

```
plt.scatter(x[y_kmeans == 0, 0], x[y_kmeans == 0, 1],
            s = 100, c = 'salmon', label = 'Iris-setosa')
plt.scatter(x[y_kmeans == 1, 0], x[y_kmeans == 1, 1],
            s = 100, c = 'violet', label = 'Iris-versicolour')
plt.scatter(x[y_kmeans == 2, 0], x[y_kmeans == 2, 1],
            s = 100, c = 'lightgreen', label = 'Iris-virginica')

# Plotting the centroids of the clusters
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:,1],
            s = 100, c = 'black', label = 'Centroids')

plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
```

```
<matplotlib.legend.Legend at 0x17009626888>
```

Out[78]:


4.5
4.0
3.5
3.0
2.5
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Iris-setosa
Iris-versicolour
Iris-virginica
Centroids

```
In [ ]:
```