Data Science And Bussiness Analytics Intern At TheSparksFoundation

GRIPJAN21

Author- Rajat Kumar

Task 2:-Prediction Using Unsupervised ML

Problem Statement:- From the given ‘Iris’ dataset, predict the optimum number of clusters and represent it Visually.

Step 1:- Importing all libraries

```
In [3]:  import pandas as pd
         import numpy as np
         import seaborn as sns
         import matplotlib.pyplot as plt
```

Step 2:-Reading Data-Set

```
In [30]: iris = pd.read_csv(r"E:\TSF\Task2\iris.csv")
         iris.head() # it will show first five row of data set
```

Out[30]:

	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 [29]: iris.tail() # It will show last five row of data set
```

Out[29]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
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

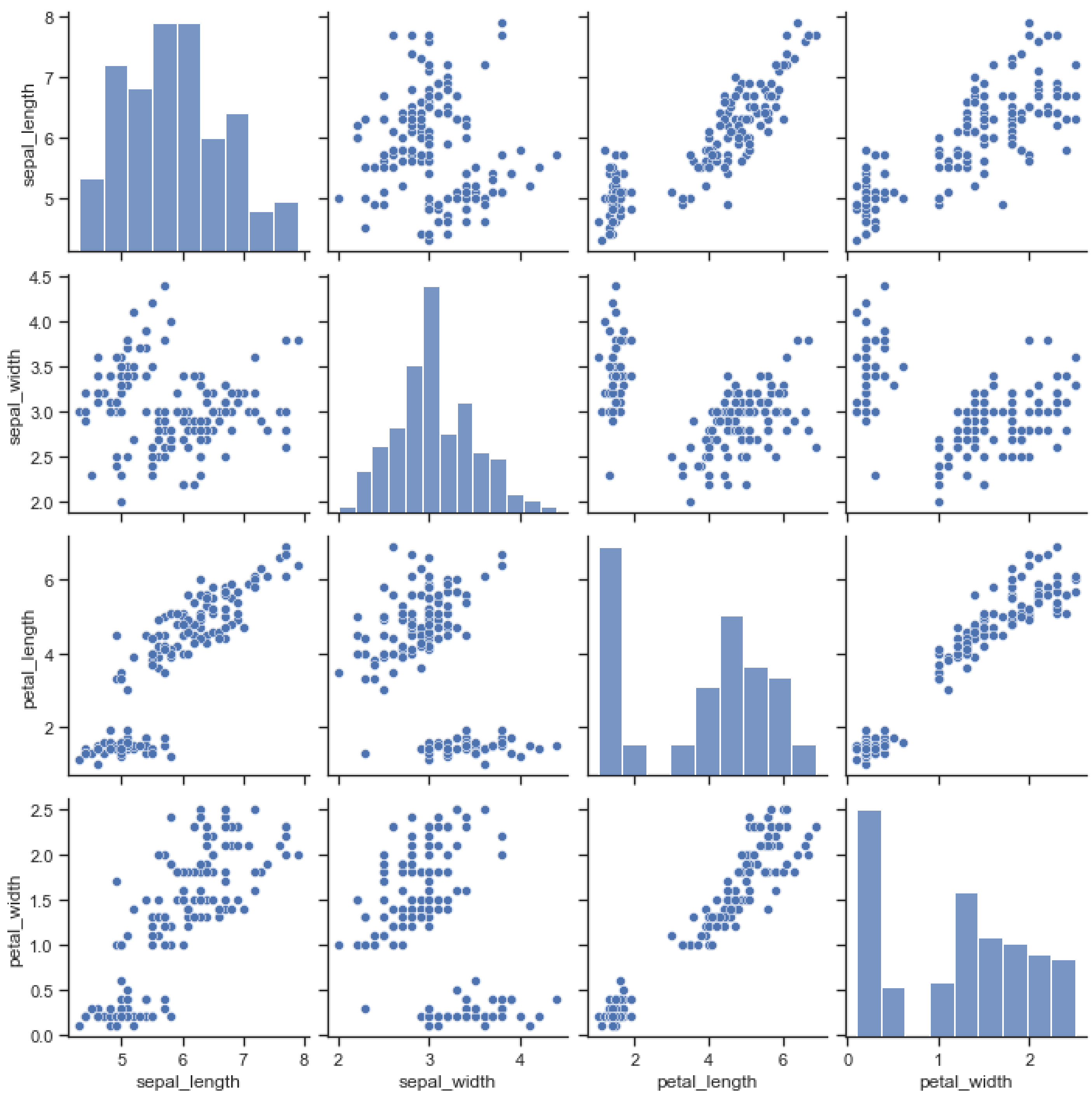
Step 3:-Discribing About Data.

```
In [21]: iris.describe().T
```

Out[21]:

	count	mean	std	min	25%	50%	75%	max
Id	150.0	75.500000	43.445368	1.0	38.25	75.50	112.75	150.0
SepalLengthCm	150.0	5.843333	0.828066	4.3	5.10	5.80	6.40	7.9
SepalWidthCm	150.0	3.054000	0.433594	2.0	2.80	3.00	3.30	4.4
PetalLengthCm	150.0	3.758667	1.764420	1.0	1.60	4.35	5.10	6.9
PetalWidthCm	150.0	1.198667	0.763161	0.1	0.30	1.30	1.80	2.5

```
In [31]: sns.set(style="ticks",color_codes=True)
         iris=sns.load_dataset("iris")
         g=sns.pairplot(iris)
```

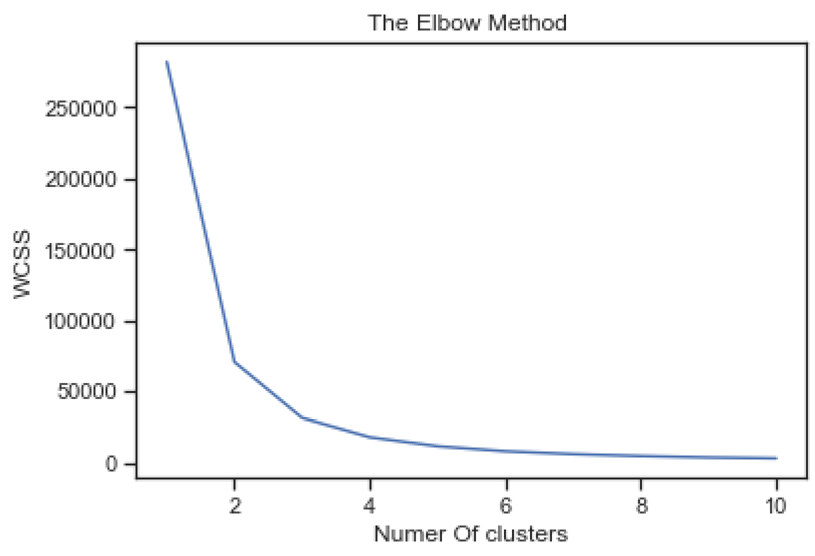


```
In [17]: x=iris.iloc[:,:-1].values
         x.shape

Out[17]: (150, 5)
```

Step 4:-Finnding Optimum number of clusters for KMean.

```
In [18]: from sklearn.cluster import KMeans
         WCSS=[]
         for i in range(1,11):
             kmeans = KMeans(n_clusters=i, init ="k-means++",random_state=42)
             kmeans.fit(x)
             WCSS.append(kmeans.inertia_)
         plt.plot(range(1, 11), WCSS)
         plt.title("The Elbow Method ")
         plt.xlabel("Numer Of clusters")
         plt.ylabel("WCSS")
         plt.show()
```



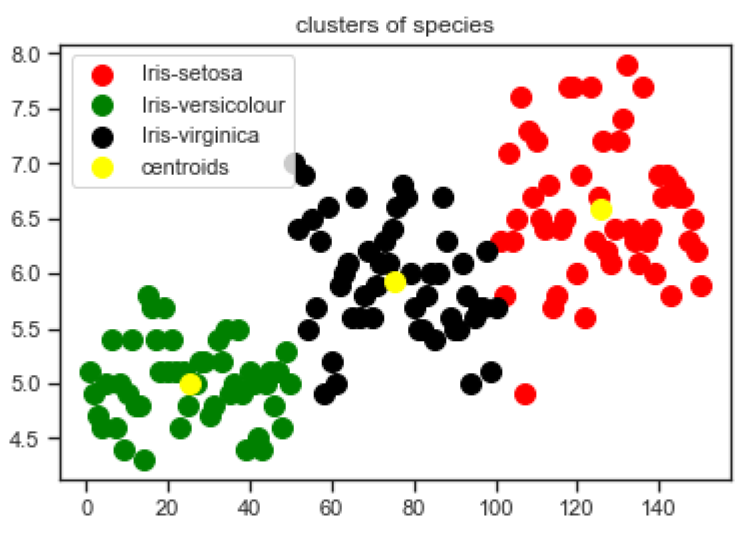
step 5:-Applying K-Means to the dataset

```
In [19]: kmeans=KMeans(n_clusters=3,init="k-means++",

                       max_iter=300,n_init=10,random_state=0)
         y_kmeans=kmeans.fit_predict(x)
```

```
In [23]: 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="green",label = "Iris-versicolour")
         plt.scatter(x[y_kmeans==2,0],x[y_kmeans == 2,1],
                     s=100,c="black",label = "Iris-virginica" )

         plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],
                     s=100,c="yellow",label="centroids")
         plt.title("clusters of species")
         plt.legend()
         plt.show()
```



Thank You

```
In [ ]:
```