GRIP-THE SPARKS FOUNDATION

Data Science and Bussiness Analytics

Task 2:Prediction using unsupervised ML ¶

Performed by: ATTIYIL AMMU PRASHANTH

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:

```
data = pd.read_csv("D:/Iris.csv")
data.head()
```

Out[2]:

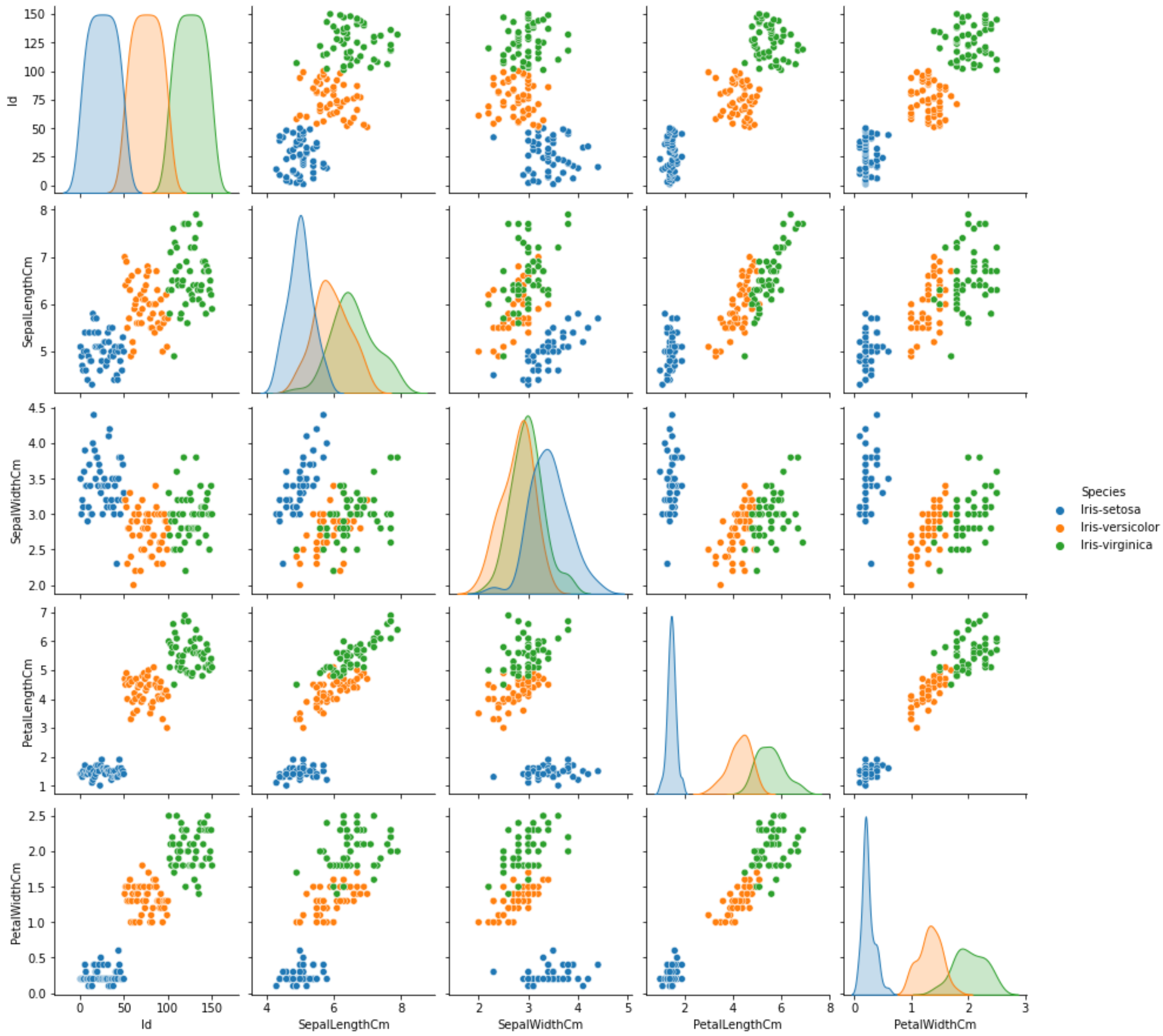
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [3]:

```
import seaborn as sns
sns.pairplot(data,hue='Species')
```

Out[3]:

<seaborn.axisgrid.PairGrid at 0x200a7efc730>



In [4]:

```
p = data.iloc[:,[0,1,2,3,4]].values
```

In [5]:

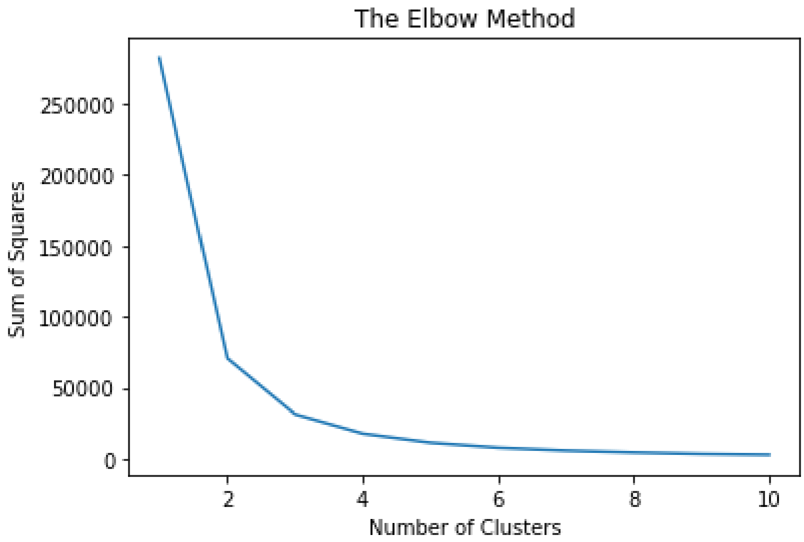
```
from sklearn.cluster import KMeans
```

In [6]:

```
w = []
for i in range(1,11):
    kmeans = KMeans(n_clusters = i,init = 'k-means++',max_iter = 300, n_init=10,random_stat
    kmeans.fit(p)
    w.append(kmeans.inertia_)
```

In [7]:

```
plt.plot(range(1,11),w)
plt.title('The Elbow Method')
plt.xlabel('Number of Clusters')
plt.ylabel('Sum of Squares')
plt.show()
```



In [8]:

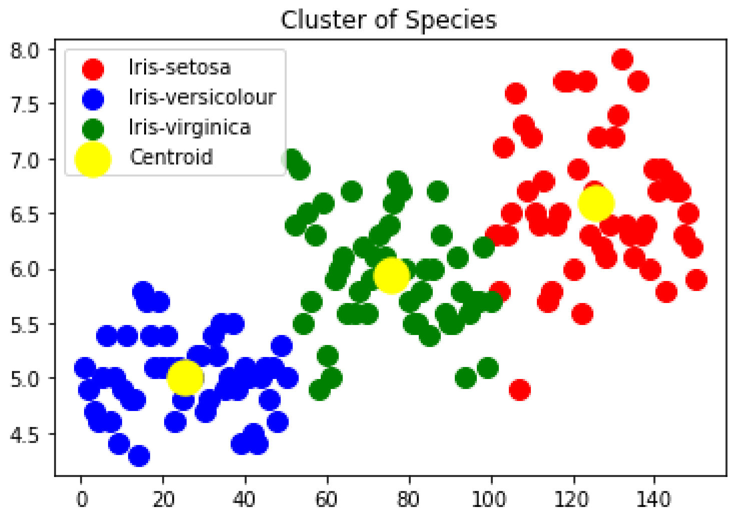
```
kmeans = KMeans(n_clusters = 3,init = 'k-means++',max_iter = 300, n_init=10,random_state=0)
y_kmeans = kmeans.fit_predict(p)
y_kmeans
```

Out[8]:

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 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])
```

In [9]:

```
plt.scatter(p[y_kmeans==0,0], p[y_kmeans==0,1],s = 100,c = 'red', label = 'Iris-setosa')
plt.scatter(p[y_kmeans==1,0], p[y_kmeans==1,1],s = 100,c = 'blue', label = 'Iris-versicolou
plt.scatter(p[y_kmeans==2,0], p[y_kmeans==2,1],s = 100,c = 'green', label = 'Iris-virginica
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], s = 300, c = 'yello
plt.title("Cluster of Species")
plt.legend()
plt.show()
```



In [10]:

```
KModel = kmeans.fit(p)
KModel
```

Out[10]:

```
KMeans(n_clusters=3, random_state=0)
```

In [11]:

```
KModel.labels_
```

Out[11]:

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 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])
```

In [12]:

```
KModel.cluster_centers_
```

Out[12]:

```
array([[125.5  ,   6.588,   2.974,   5.552,   2.026],
       [ 25.5  ,   5.006,   3.418,   1.464,   0.244],
       [ 75.5  ,   5.936,   2.77 ,   4.26 ,   1.326]])
```

In [ ]: