To explore unsupervised machine learning using iris dataset

In [1]:

```
# import standard libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [6]:

```
df = pd.read_csv('Iris.csv',)
df.head()
```

Out[6]:

	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 [7]:

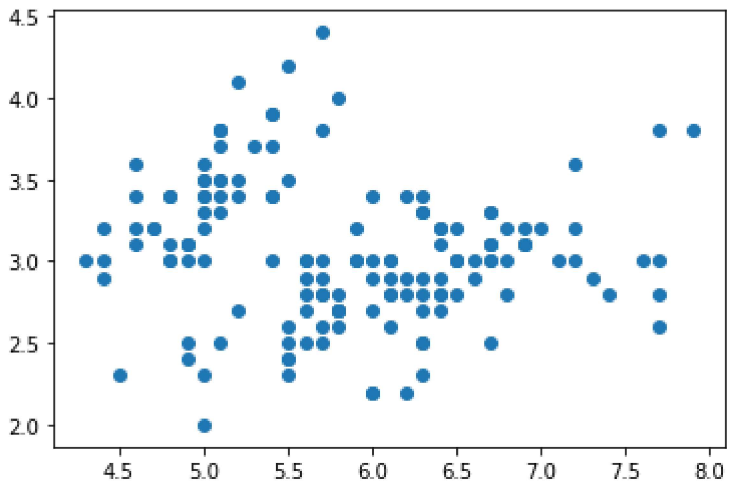
```
df1 = df.drop(['Id','PetalLengthCm','PetalWidthCm','Species'],axis = 1)
df1.head()
```

Out[7]:

	SepalLengthCm	SepalWidthCm
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2
3	4.6	3.1
4	5.0	3.6

In [8]:
```
plt.scatter(df1['SepalLengthCm'],df1['SepalWidthCm'])
```

Out[8]:
```
<matplotlib.collections.PathCollection at 0x1f7c9a80f08>
```



In [9]:
```
from sklearn.cluster import KMeans
```

In [10]:
```
km = KMeans(n_clusters = 3)
km
```

Out[10]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

In [11]:

```
y_pred = km.fit_predict(df1)
y_pred
```

Out[11]:

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

In [12]:

```
df1['Cluster'] = y_pred
df1.head()
```

Out[12]:

	SepalLengthCm	SepalWidthCm	Cluster
0	5.1	3.5	0
1	4.9	3.0	0
2	4.7	3.2	0
3	4.6	3.1	0
4	5.0	3.6	0

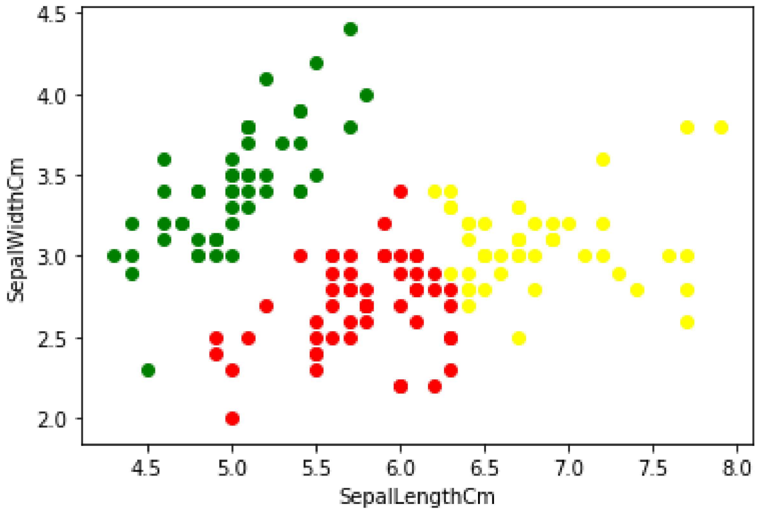
In [13]:

```
d1 = df1[df1.Cluster==0]
d2 = df1[df1.Cluster==1]
d3 = df1[df1.Cluster==2]
plt.scatter(d1['SepalLengthCm'],d1['SepalWidthCm'],color = 'green')
plt.scatter(d2['SepalLengthCm'],d2['SepalWidthCm'],color = 'yellow')
plt.scatter(d3['SepalLengthCm'],d3['SepalWidthCm'],color = 'red')

plt.xlabel('SepalLengthCm')
plt.ylabel('SepalWidthCm')
```

Out[13]:

```
Text(0, 0.5, 'SepalWidthCm')
```



In [14]:

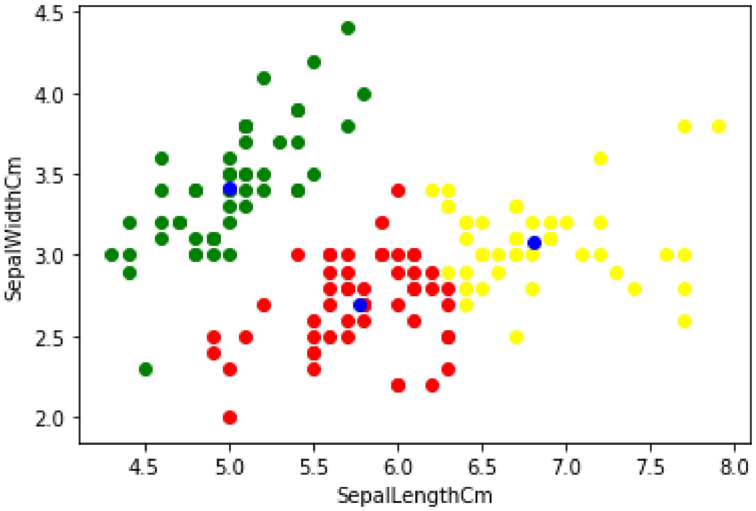
```
centroid = km.cluster_centers_
centroid
```

Out[14]:

```
array([[5.006     , 3.418     ],
       [6.81276596, 3.07446809],
       [5.77358491, 2.69245283]])
```

In [16]:
```
d1 = df1[df1.Cluster==0]
d2 = df1[df1.Cluster==1]
d3 = df1[df1.Cluster==2]
plt.scatter(d1['SepalLengthCm'],d1['SepalWidthCm'],color = 'green')
plt.scatter(d2['SepalLengthCm'],d2['SepalWidthCm'],color = 'yellow')
plt.scatter(d3['SepalLengthCm'],d3['SepalWidthCm'],color = 'red')
plt.scatter(centroid[:,0],centroid[:,1],color='blue')
plt.xlabel('SepalLengthCm')
plt.ylabel('SepalWidthCm')
```

Out[16]:
```
Text(0, 0.5, 'SepalWidthCm')
```



In [17]:

```
k_range = range(1,10)
wcss = []
for k in k_range:
    km = KMeans(n_clusters = k)
    km.fit(df1)
    wcss.append(km.inertia_)

wcss
```

Out[17]:

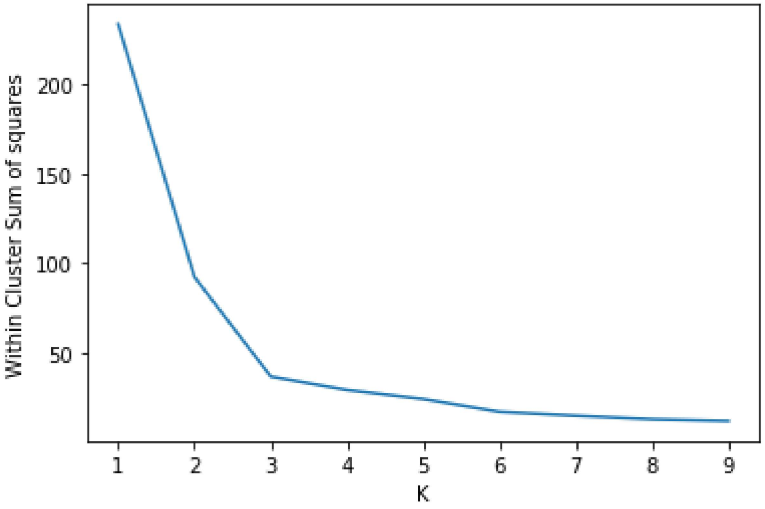
```
[233.12093333333337,
 92.5692,
 37.12370212765957,
 29.683368794326242,
 24.62265171269612,
 17.52604040072492,
 15.358431677018636,
 13.38927741361952,
 12.41260315853737]
```

In [18]:

```
plt.xlabel('K')
plt.ylabel('Within Cluster Sum of squares')
plt.plot(k_range,wcss)
```

Out[18]:

```
[<matplotlib.lines.Line2D at 0x1f7cc503508>]
```



In [20]:
```
df['Species'].value_counts()
```

Out[20]:
```
Iris-virginica     50
Iris-setosa        50
Iris-versicolor    50
Name: Species, dtype: int64
```

In [21]:
```
df = df.drop(['Id','Species'],axis=1)
df.head()
```

Out[21]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

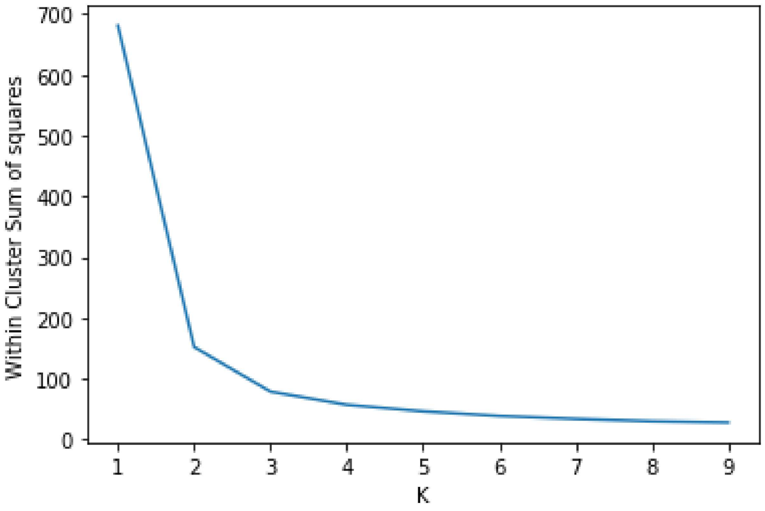
In [22]:
```
k_range = range(1,10)
wcss = []
for k in k_range:
    km = KMeans(n_clusters = k)
    km.fit(df)
    wcss.append(km.inertia_)

wcss
```

Out[22]:
```
[680.8244,
 152.36870647733906,
 78.94084142614602,
 57.31787321428571,
 46.56163015873016,
 38.964787851037855,
 34.1967910993998,
 30.209244428234708,
 28.24875]
```

```
In [23]: plt.xlabel('K')
         plt.ylabel('Within Cluster Sum of squares')
         plt.plot(k_range,wcss)

Out[23]: [<matplotlib.lines.Line2D at 0x1f7cca0adc8>]
```



```
In [24]: km = KMeans(n_clusters = 3)
         km

Out[24]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
                n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
                random_state=None, tol=0.0001, verbose=0)
```

```
In [25]: y_pred1 = km.fit_predict(df)
         y_pred1

Out[25]: 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, 0, 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, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 0, 0, 0, 0, 2, 0, 0, 0,
                0, 0, 0, 2, 2, 0, 0, 0, 0, 2, 0, 2, 0, 2, 0, 0, 2, 2, 0, 0, 0, 0,
                0, 2, 0, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 2])
```

In [26]:

```
df['Predicted'] = y_pred1
df.head()
```

Out[26]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Predicted
0	5.1	3.5	1.4	0.2	1
1	4.9	3.0	1.4	0.2	1
2	4.7	3.2	1.3	0.2	1
3	4.6	3.1	1.5	0.2	1
4	5.0	3.6	1.4	0.2	1

In [27]:

```
centroid = km.cluster_centers_
centroid
```

Out[27]:

```
array([[6.85      , 3.07368421, 5.74210526, 2.07105263],
       [5.006     , 3.418     , 1.464     , 0.244     ],
       [5.9016129 , 2.7483871 , 4.39354839, 1.43387097]])
```

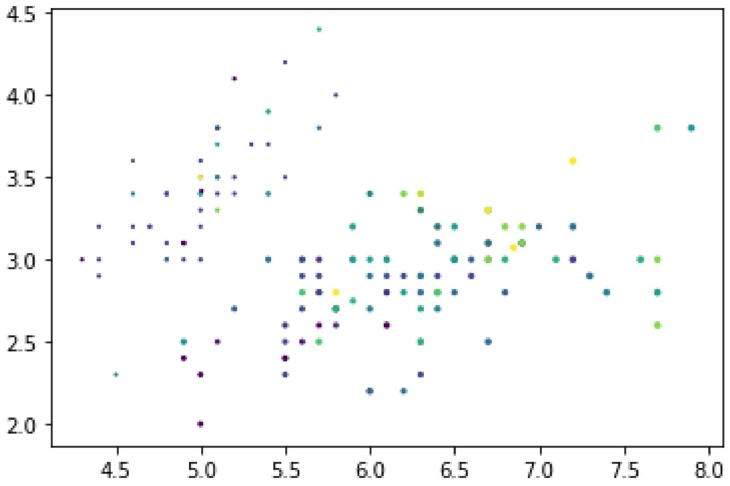
In [28]:

```
d4 = df[df.Predicted==0]
d5 = df[df.Predicted==1]
d6 = df[df.Predicted==2]
plt.scatter(d4['SepalLengthCm'],d4['SepalWidthCm'],d4['PetalLengthCm'],d4['PetalWidthCm'])
plt.scatter(d5['SepalLengthCm'],d5['SepalWidthCm'],d5['PetalLengthCm'],d5['PetalWidthCm'])
plt.scatter(d6['SepalLengthCm'],d6['SepalWidthCm'],d6['PetalLengthCm'],d6['PetalWidthCm'])

plt.scatter(centroid[:,0],centroid[:,1],centroid[:,2],centroid[:,3])
```

Out[28]:

```
<matplotlib.collections.PathCollection at 0x1f7ccb35a08>
```



In [29]:

```
new_data = pd.read_csv('Iris.csv')
new_data.head()
```

Out[29]:

	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 [30]:

```
new_data['Predicted'] = y_pred1
new_data.head()
```

Out[30]:

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

In [31]:

```
new_data['Species'] = new_data['Species'].map({'Iris-versicolor':0,'Iris-setosa':1,'Iris-virginica':2})
```

In [32]:

```
new_data
```

Out[32]:

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

150 rows × 7 columns

In [33]:

```
from sklearn.metrics import confusion_matrix
confusion_matrix(new_data['Species'],y_pred1)
```

Out[33]:

```
array([[ 2,  0, 48],
       [ 0, 50,  0],
       [36,  0, 14]], dtype=int64)
```

In [ ]:

```

```