In [1]:

```
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.cluster import KMeans
import sklearn.metrics as sm
import pandas as pd
import numpy as np
#import matplotlib inline
```

In [2]:

```
iris = datasets.load_iris()
```

In [3]:

```
X = pd.DataFrame(iris.data)
X.columns = ['Sepal_Length','Sepal_Width','Petal_Length','Petal_Width']
y = pd.DataFrame(iris.target)
y.columns = ['Targets']
```

In [4]:

```
# K Means Cluster
model = KMeans(n_clusters=3)
model.fit(X)
# This is what KMeans thought
model.labels_
```

Out[4]:

```
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, 0, 0, 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, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 2, 2, 2, 0, 2, 2, 2,
       2, 2, 2, 0, 0, 2, 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 0, 0, 2, 2, 2, 2,
       2, 0, 2, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 0])
```

In [5]:

```
plt.figure(figsize=(14,7))
```

Out[5]:

```
<Figure size 1008x504 with 0 Axes>
```

```
<Figure size 1008x504 with 0 Axes>
```

In [6]:

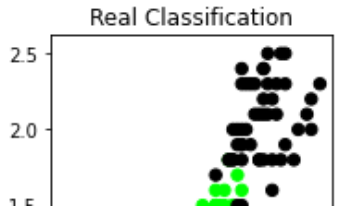
```
colormap = np.array(['red', 'lime', 'black'])
```

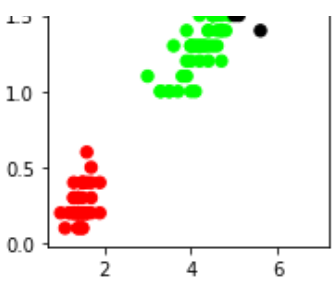
In [7]:

```
plt.subplot(1, 2, 1)
plt.scatter(X.Petal_Length, X.Petal_Width, c=colormap[y.Targets], s=40)
plt.title('Real Classification')
```

Out[7]:

```
Text(0.5, 1.0, 'Real Classification')
```



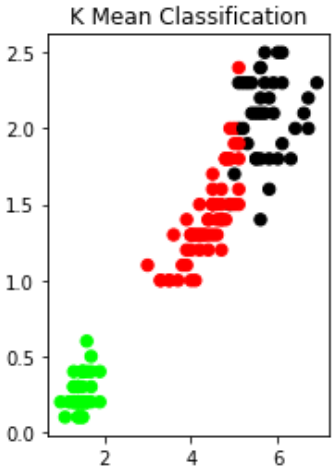


In [8]:

```
plt.subplot(1, 2, 2)
plt.scatter(X.Petal_Length, X.Petal_Width, c=colormap[model.labels_], s=40)
plt.title('K Mean Classification')
```

Out[8]:

```
Text(0.5, 1.0, 'K Mean Classification')
```



In [9]:

```
plt.figure(figsize=(14,7))
```

Out[9]:

```
<Figure size 1008x504 with 0 Axes>
```

```
<Figure size 1008x504 with 0 Axes>
```

In [10]:

```
print('The accuracy score : ',sm.accuracy_score(y, model.labels_))
sm.confusion_matrix(y, model.labels_)
```

```
The accuracy score :  0.24
```

Out[10]:

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

In [11]:

```
predY = np.choose(model.labels_, [0, 1, 2]).astype(np.int64)
```

In [12]:

```
print (predY)
```

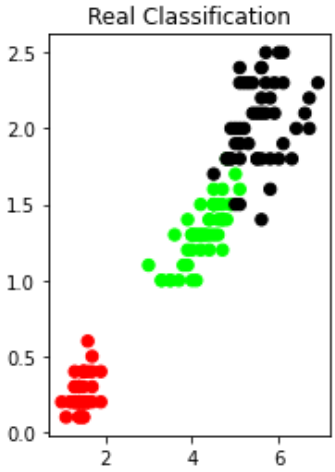
```
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 2 2 2 2 0 2 2 2 2 2
 2 2 0 0 2 2 2 2 0 2 0 2 0 2 2 0 0 2 2 2 2 2 0 2 2 2 0 2 2 2 0 2 2 2 0 2
 2 0]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

In [13]:

```
plt.subplot(1, 2, 1)
plt.scatter(X.Petal_Length, X.Petal_Width, c=colormap[y.Targets], s=40)
plt.title('Real Classification')
```

Out[13]:

```
Text(0.5, 1.0, 'Real Classification')
```

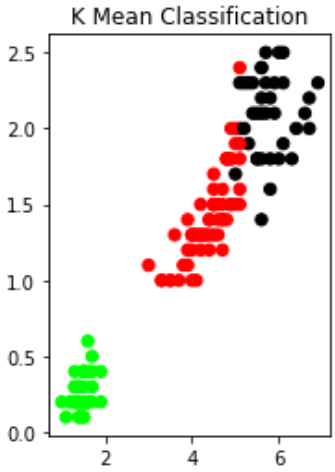


In [14]:

```
plt.subplot(1, 2, 2)
plt.scatter(X.Petal_Length,X.Petal_Width, c=colormap[predY], s=40)
plt.title('K Mean Classification')
```

Out[14]:

```
Text(0.5, 1.0, 'K Mean Classification')
```



In [15]:

```
print('The accuracy score of K-Mean: ',sm.accuracy_score(y, model.labels_))
print('The Confusion matrixof K-Mean: ',sm.confusion_matrix(y, model.labels_))
```

```
The accuracy score of K-Mean:  0.24
The Confusion matrixof K-Mean:  [[ 0 50  0]
 [48  0  2]
 [14  0 36]]
```

In [16]:

```
from sklearn import preprocessing
scaler = preprocessing.StandardScaler()
scaler.fit(X)
xsa = scaler.transform(X)
xs = pd.DataFrame(xsa, columns = X.columns)
```

In [17]:

```
from sklearn.mixture import GaussianMixture
gmm = GaussianMixture(n_components=3)
gmm.fit(xs)
```

Out[17]:

```
GaussianMixture(n_components=3)
```

In [18]:

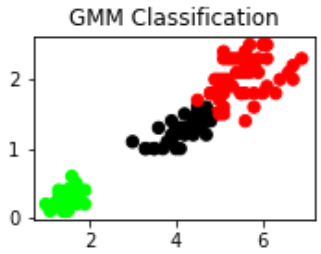
```
y_cluster_gmm = gmm.predict(xs)
```

In [19]:

```
plt.subplot(2, 2, 3)
plt.scatter(X.Petal_Length, X.Petal_Width, c=colormap[y_cluster_gmm], s=40)
plt.title('GMM Classification')
```

Out[19]:

```
Text(0.5, 1.0, 'GMM Classification')
```



In [20]:

```
print('The accuracy score of EM: ',sm.accuracy_score(y, y_cluster_gmm))
print('The Confusion matrix of EM: ',sm.confusion_matrix(y, y_cluster_gmm))
```

```
The accuracy score of EM:  0.0
The Confusion matrix of EM:  [[ 0 50  0]
 [ 5  0 45]
 [50  0  0]]
```

In [ ]:

```

```