8.Problem : Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same dataset for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.

In [12]:
```
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.cluster import KMeans
import pandas as pd
import numpy as np
```

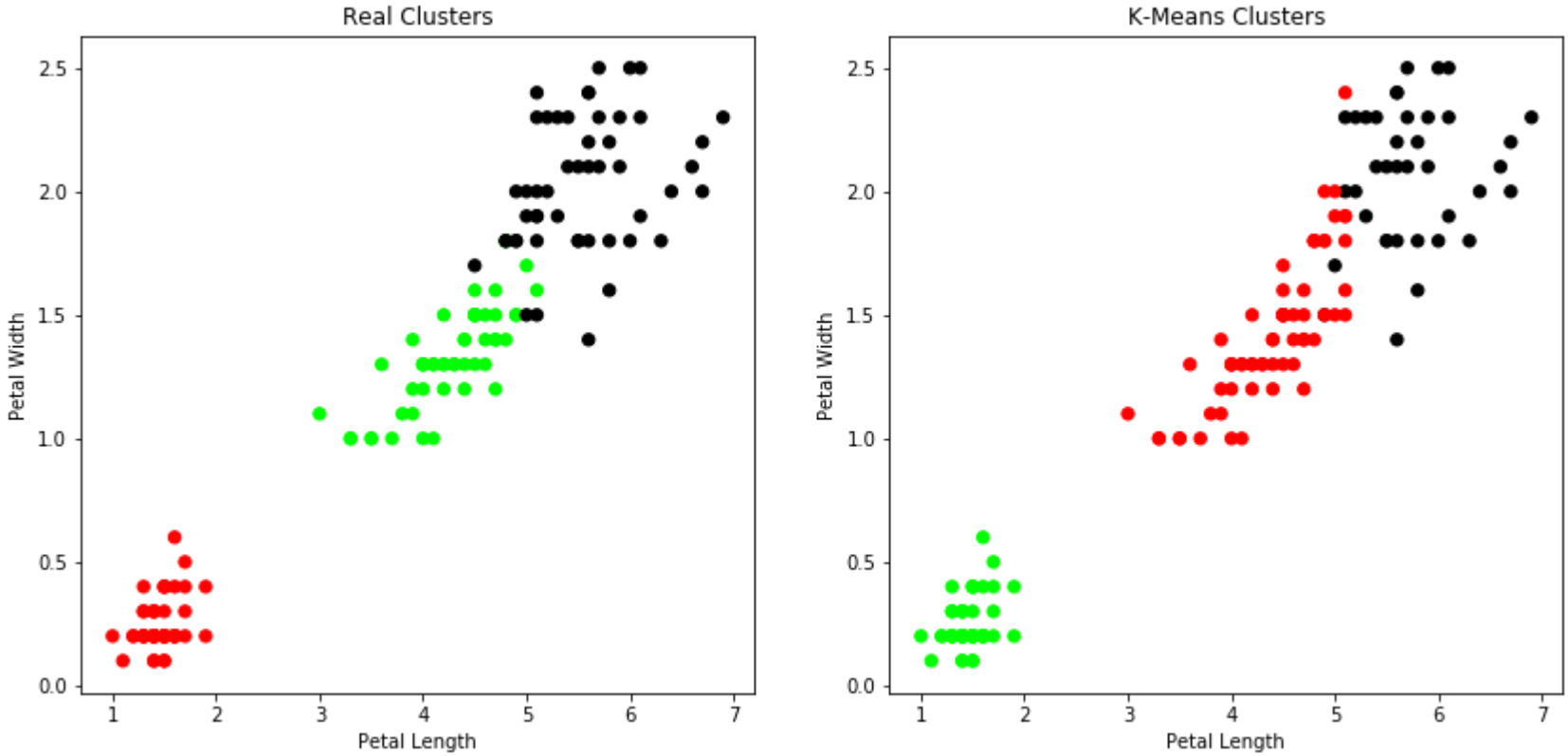
In [13]:
```
iris = datasets.load_iris()
X = pd.DataFrame(iris.data)
X.columns = ['Sepal_Length', 'Sepal_Width', 'Petal_Length', 'Petal_Width']
y = pd.DataFrame(iris.target)
y.columns = ['Targets']
```

In [14]:
```
model = KMeans(n_clusters=3)
model.fit(X)
```

Out[14]:
```
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 [15]:
```
plt.figure(figsize= (14,14))
colormap = np.array(['red', 'lime', 'black'])
plt.subplot(2,2,1)
plt.scatter(X.Petal_Length, X.Petal_Width, c=colormap[y.Targets], s=40)
plt.title("Real Clusters")
plt.xlabel("Petal Length")
plt.ylabel("Petal Width")
#Plot the model's classification
plt.subplot(2,2,2)
plt.scatter(X.Petal_Length, X.Petal_Width, c=colormap[model.labels_], s=40)
plt.title("K-Means Clusters")
plt.xlabel("Petal Length")
plt.ylabel("Petal Width")
```

Out[15]:
```
Text(0, 0.5, 'Petal Width')
```



EM Algorithm

In [16]:
```
from sklearn.mixture import GaussianMixture
scaler = preprocessing.StandardScaler()
scaler.fit(X)
xsa = scaler.transform(X)
xs = pd.DataFrame(xsa, columns=X.columns)
gmm = GaussianMixture(n_components=3)
gmm.fit(xs)
gmm_y = gmm.predict(xs)
plt.figure(figsize=(14,14))
plt.subplot(2,2,3)
plt.scatter(X.Petal_Length, X.Petal_Width, c=colormap[gmm_y], s=40)
plt.title("GMM Clustering")
plt.xlabel("Petal Length")
plt.ylabel("Petal Width")
print("The GMM using EM algo based clustering matched the true labels more closely than the KMeans")
```

```
The GMM using EM algo based clustering matched the true labels more closely than the KMeans
```