In [51]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearnex import patch_sklearn #for speed up CPU
#from daal4py.oneapi import sycl_context #for speed up GPU
patch_sklearn()
from sklearn.cluster import KMeans
from sklearn_extra.cluster import KMedoids
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_samples, silhouette_score
```

```
Intel(R) Extension for Scikit-learn* enabled (https://github.com/intel/sciki
t-learn-intelex)
```

In [52]:

```
'''iris = pd.read_csv('iris.csv')
iris.head()'''
iris=datasets.load_iris()
iris
```

Out[52]:

```
{'data': array([[5.1, 3.5, 1.4, 0.2],
        [4.9, 3. , 1.4, 0.2],
        [4.7, 3.2, 1.3, 0.2],
        [4.6, 3.1, 1.5, 0.2],
        [5. , 3.6, 1.4, 0.2],
        [5.4, 3.9, 1.7, 0.4],
        [4.6, 3.4, 1.4, 0.3],
        [5. , 3.4, 1.5, 0.2],
        [4.4, 2.9, 1.4, 0.2],
        [4.9, 3.1, 1.5, 0.1],
        [5.4, 3.7, 1.5, 0.2],
        [4.8, 3.4, 1.6, 0.2],
        [4.8, 3. , 1.4, 0.1],
        [4.3, 3. , 1.1, 0.1],
        [5.8, 4. , 1.2, 0.2],
        [5.7, 4.4, 1.5, 0.4],
        [5.4, 3.9, 1.3, 0.4],
```

In [53]:

```
x=pd.DataFrame(iris.data)
```

In [54]:

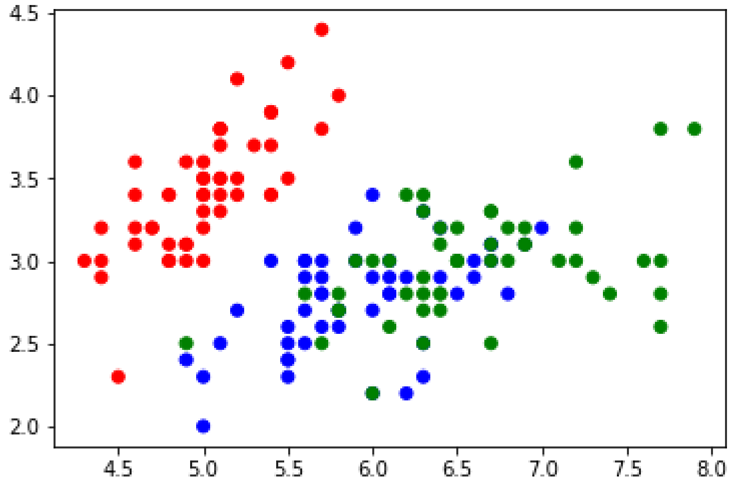
```
x.columns= ['sepal_length','sepal_width','petal_length','petal_width']
x.head()
```

Out[54]:

	sepal_length	sepal_width	petal_length	petal_width
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 [55]:

```
color= np.array(['red','blue','green'])
z=plt.scatter(x.sepal_length,x.sepal_width, c=color[iris.target])
```



In [56]:

```
sc = StandardScaler().fit(x)
x_new = sc.transform(x)
```

In [57]:

```
km = KMedoids(n_clusters = 3)
km.fit(x_new)
```

Out[57]:

```
KMedoids(n_clusters=3)
```

In [58]:

```
predict=km.fit_predict(x_new)
```

In [59]:

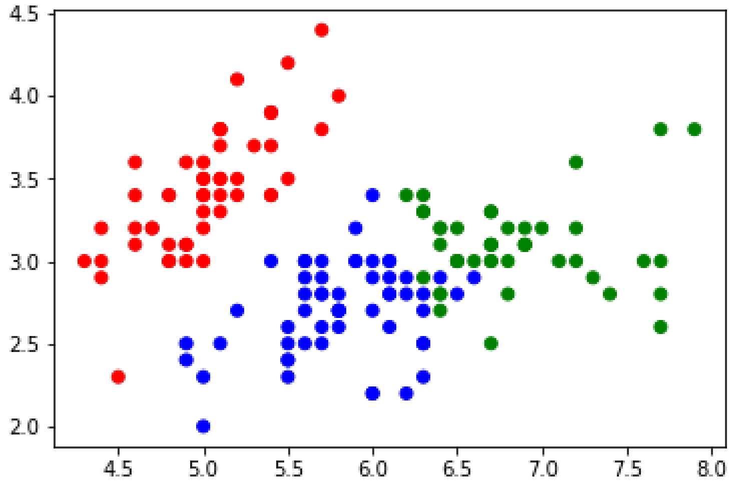
```
predict
```

Out[59]:

```
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, 2, 2, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2,
       2, 2, 2, 1, 1, 2, 2, 2, 2, 1, 2, 1, 2, 1, 2, 2, 1, 1, 2, 2, 2, 2,
       2, 1, 1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 1], dtype=int64)
```

In [60]:

```
color= np.array(['red','blue','green'])
z=plt.scatter(x.sepal_length,x.sepal_width, c=color[predict])
```



In [61]:

```
accuracy_score(iris.target,predict)
```

Out[61]:

```
0.84
```

In [62]:

```
km.inertia_
```

Out[62]:

```
131.87877332824286
```

In [63]:

```
silhouette_score(x_new,predict)
```

Out[63]:

```
0.4590416105554613
```

In [64]:

```
sample_silhouette_values = silhouette_samples(x_new,predict )
for i in range(3):
    ith_cluster_silhouette_values = sample_silhouette_values[predict == i]
    print(np.mean(ith_cluster_silhouette_values))
```

```
0.636330614585637
0.3768888598233938
0.36213487963471125
```

In [ ]: