In [23]:

```
from sklearn.datasets import load_iris
import pandas as pd
import matplotlib.pyplot as plt
import mglearn
import numpy as np
from IPython.display import display

iris_dataset = load_iris()
print("Keys of iris dataset: \n{}".format(iris_dataset.keys()))
```

```
Keys of iris dataset:
dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'fil
ename', 'data_module'])
```

In [3]:

```
print(iris_dataset['DESCR'][:193] + '\n...')
```

```
.. _iris_dataset:

Iris plants dataset
--------------------

**Data Set Characteristics:**

:Number of Instances: 150 (50 in each of three classes)
:Number of Attributes: 4 numeric, predictive
...
```

In [4]:

```
print("Target Name: {}".format(iris_dataset['target_names']))
```

```
Target Name: ['setosa' 'versicolor' 'virginica']
```

In [5]:

```
print("Feature Names: {}".format(iris_dataset['feature_names']))
```

```
Feature Names: ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal
width (cm)']
```

In [6]:

```
print("Type of data: {}".format(type(iris_dataset['data'])))
```

```
Type of data: <class 'numpy.ndarray'>
```

In [7]:

```
print("Shape of data: {}".format(iris_dataset['data'].shape))
```

```
Shape of data: (150, 4)
```

In [8]:

```
print("First five rows of data: \n{}".format(iris_dataset['data'][:5]))
```

```
First five rows of data:
[[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]]
```

In [9]:

```
print("Type of Target: {}".format(type(iris_dataset['target'])))
```

```
Type of Target: <class 'numpy.ndarray'>
```

In [10]:

```
print("Shape of Target: {}".format(iris_dataset['target'].shape))
```

```
Shape of Target: (150,)
```

In [11]:

```
print("Target: \n{}".format(iris_dataset['target']))
```

```
Target:
[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 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]
```

In [14]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(iris_dataset['data'], iris_data
print("X_train shape: {}".format(X_train.shape))
print("Y_train shape: {}".format(Y_train.shape))
print("X_test shape: {}".format(X_test.shape))
print("Y_test shape: {}".format(Y_test.shape))
```

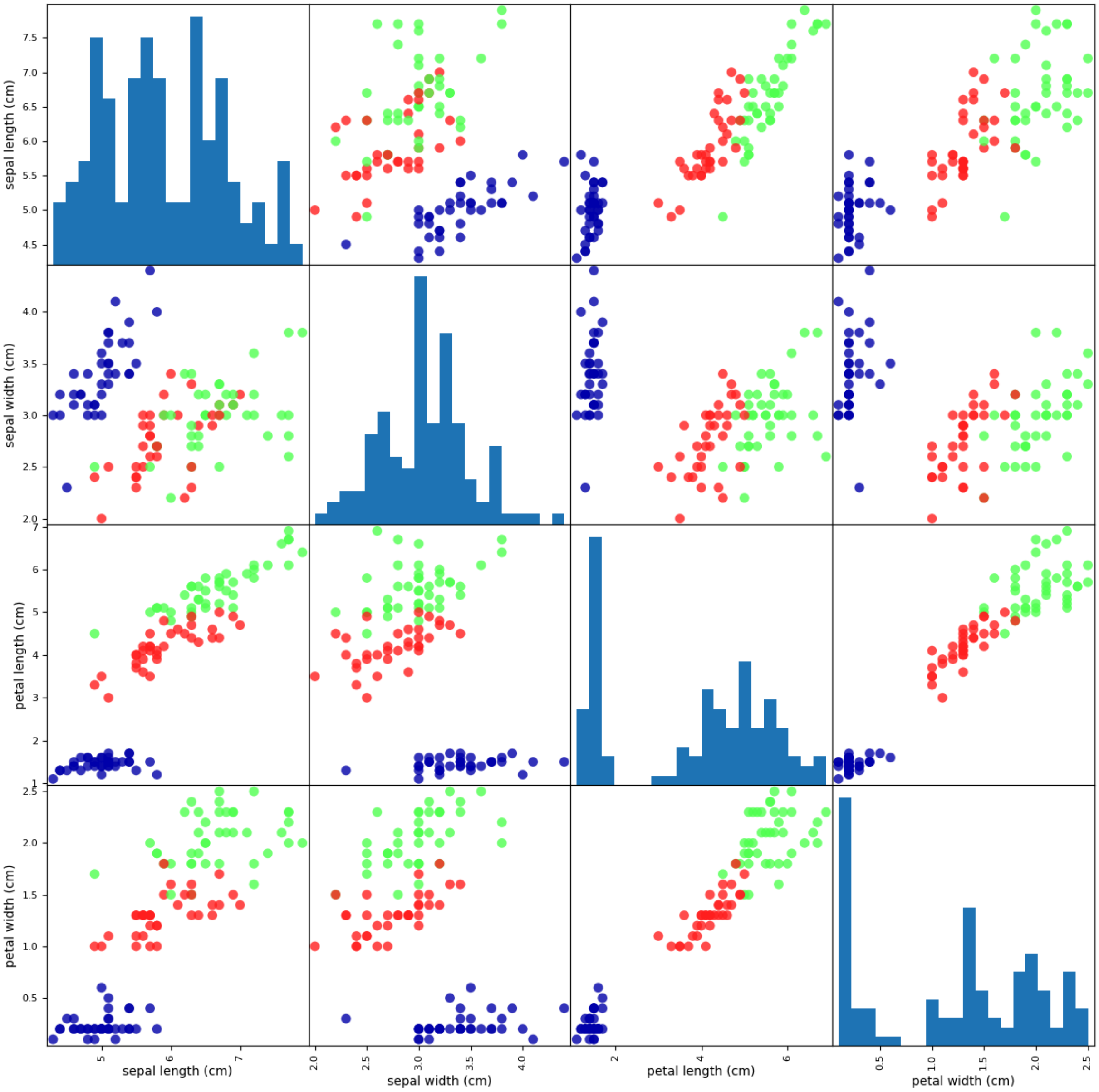
```
X_train shape: (112, 4)
Y_train shape: (112,)
X_test shape: (38, 4)
Y_test shape: (38,)
```

In [19]:

```
iris_dataFrame = pd.DataFrame(X_train, columns=iris_dataset.feature_names)
pd.plotting.scatter_matrix(iris_dataFrame, c=Y_train, figsize=(15, 15), marker='o',
```

Out[19]:

```
array([[<Axes: xlabel='sepal length (cm)', ylabel='sepal length (cm)'>,
        <Axes: xlabel='sepal width (cm)', ylabel='sepal length (cm)'>,
        <Axes: xlabel='petal length (cm)', ylabel='sepal length (cm)'>,
        <Axes: xlabel='petal width (cm)', ylabel='sepal length (cm)'>],
       [<Axes: xlabel='sepal length (cm)', ylabel='sepal width (cm)'>,
        <Axes: xlabel='sepal width (cm)', ylabel='sepal width (cm)'>,
        <Axes: xlabel='petal length (cm)', ylabel='sepal width (cm)'>,
        <Axes: xlabel='petal width (cm)', ylabel='sepal width (cm)'>],
       [<Axes: xlabel='sepal length (cm)', ylabel='petal length (cm)'>,
        <Axes: xlabel='sepal width (cm)', ylabel='petal length (cm)'>,
        <Axes: xlabel='petal length (cm)', ylabel='petal length (cm)'>,
        <Axes: xlabel='petal width (cm)', ylabel='petal length (cm)'>],
       [<Axes: xlabel='sepal length (cm)', ylabel='petal width (cm)'>,
        <Axes: xlabel='sepal width (cm)', ylabel='petal width (cm)'>,
        <Axes: xlabel='petal length (cm)', ylabel='petal width (cm)'>,
        <Axes: xlabel='petal width (cm)', ylabel='petal width (cm)'>]],
      dtype=object)
```



Building Model: K-Nearest Neighbour (KNN)

In [21]:
```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=1)

knn.fit(X_train, Y_train)
```

Out[21]:

▾ KNeighborsClassifier

```
KNeighborsClassifier(n_neighbors=1)
```

In [24]:
```
x_new = np.array([[5, 2.9, 1, 0.2]])
print("X_new.shape: {}".format(x_new.shape))
```

```
X_new.shape: (1, 4)
```

In [26]:

```
prediction = knn.predict(x_new)
print("Prediction: {}".format(prediction))
print("Predicted Target Name: {}".format(iris_dataset['target_names'][prediction]))
```

```
Prediction: [0]
Predicted Target Name: ['setosa']
```

In [27]:

```
y_pred = knn.predict(X_test)
print("Test set Prediction: \n{}".format(y_pred))
```

```
Test set Prediction:
[2 1 0 2 0 2 0 1 1 1 2 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0 2 1 0 2 2 1 0
 2]
```

In [28]:

```
print("Test set score: {:.2f}".format(np.mean(y_pred == Y_test)))
```

```
Test set score: 0.97
```

In [30]:

```
print("Test set score: {:.2f}".format(knn.score(X_test, Y_test)))
```

```
Test set score: 0.97
```