Meet the data

In [1]:
```
from sklearn.datasets import load_iris
iris_dataset = load_iris()
```

In [2]:
```
import pandas as pd
import numpy as np
```

In [3]:
```
from pandas.plotting import scatter_matrix
import matplotlib.pyplot as plt
```

In [4]:
```
pip install mglearn
```

```
Requirement already satisfied: mglearn in c:\programdata\anaconda3\lib\site-packages (0.
2.0)
Requirement already satisfied: pillow in c:\programdata\anaconda3\lib\site-packages (fro
m mglearn) (8.2.0)
Requirement already satisfied: cycler in c:\programdata\anaconda3\lib\site-packages (fro
m mglearn) (0.10.0)
Requirement already satisfied: joblib in c:\programdata\anaconda3\lib\site-packages (fro
m mglearn) (1.0.1)
Requirement already satisfied: scikit-learn in c:\programdata\anaconda3\lib\site-package
s (from mglearn) (0.24.1)
Requirement already satisfied: imageio in c:\programdata\anaconda3\lib\site-packages (fr
om mglearn) (2.9.0)
Requirement already satisfied: pandas in c:\programdata\anaconda3\lib\site-packages (fro
m mglearn) (1.2.4)
Requirement already satisfied: matplotlib in c:\programdata\anaconda3\lib\site-packages
(from mglearn) (3.3.4)
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-packages (from
mglearn) (1.20.1)
Requirement already satisfied: six in c:\programdata\anaconda3\lib\site-packages (from c
ycler->mglearn) (1.15.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\programdat
a\anaconda3\lib\site-packages (from matplotlib->mglearn) (2.4.7)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\programdata\anaconda3\lib\site-pa
ckages (from matplotlib->mglearn) (1.3.1)
Requirement already satisfied: python-dateutil>=2.1 in c:\programdata\anaconda3\lib\site
-packages (from matplotlib->mglearn) (2.8.1)
Requirement already satisfied: pytz>=2017.3 in c:\programdata\anaconda3\lib\site-package
s (from pandas->mglearn) (2021.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\programdata\anaconda3\lib\site
-packages (from scikit-learn->mglearn) (2.1.0)
Requirement already satisfied: scipy>=0.19.1 in c:\programdata\anaconda3\lib\site-packag
es (from scikit-learn->mglearn) (1.6.2)
Note: you may need to restart the kernel to use updated packages.
```

In [5]:
```
import mglearn
```

In [6]:
```
iris_dataset
```

Out[6]:
```
{'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],
[5.1, 3.5, 1.4, 0.3],
[5.7, 3.8, 1.7, 0.3],
[5.1, 3.8, 1.5, 0.3],
[5.4, 3.4, 1.7, 0.2],
[5.1, 3.7, 1.5, 0.4],
[4.6, 3.6, 1. , 0.2],
[5.1, 3.3, 1.7, 0.5],
[4.8, 3.4, 1.9, 0.2],
[5. , 3. , 1.6, 0.2],
[5. , 3.4, 1.6, 0.4],
[5.2, 3.5, 1.5, 0.2],
[5.2, 3.4, 1.4, 0.2],
[4.7, 3.2, 1.6, 0.2],
[4.8, 3.1, 1.6, 0.2],
[5.4, 3.4, 1.5, 0.4],
[5.2, 4.1, 1.5, 0.1],
[5.5, 4.2, 1.4, 0.2],
[4.9, 3.1, 1.5, 0.2],
[5. , 3.2, 1.2, 0.2],
[5.5, 3.5, 1.3, 0.2],
[4.9, 3.6, 1.4, 0.1],
[4.4, 3. , 1.3, 0.2],
[5.1, 3.4, 1.5, 0.2],
[5. , 3.5, 1.3, 0.3],
[4.5, 2.3, 1.3, 0.3],
[4.4, 3.2, 1.3, 0.2],
[5. , 3.5, 1.6, 0.6],
[5.1, 3.8, 1.9, 0.4],
[4.8, 3. , 1.4, 0.3],
[5.1, 3.8, 1.6, 0.2],
[4.6, 3.2, 1.4, 0.2],
[5.3, 3.7, 1.5, 0.2],
[5. , 3.3, 1.4, 0.2],
[7. , 3.2, 4.7, 1.4],
[6.4, 3.2, 4.5, 1.5],
[6.9, 3.1, 4.9, 1.5],
[5.5, 2.3, 4. , 1.3],
[6.5, 2.8, 4.6, 1.5],
[5.7, 2.8, 4.5, 1.3],
[6.3, 3.3, 4.7, 1.6],
[4.9, 2.4, 3.3, 1. ],
[6.6, 2.9, 4.6, 1.3],
[5.2, 2.7, 3.9, 1.4],
[5. , 2. , 3.5, 1. ],
[5.9, 3. , 4.2, 1.5],
[6. , 2.2, 4. , 1. ],
[6.1, 2.9, 4.7, 1.4],
[5.6, 2.9, 3.6, 1.3],
[6.7, 3.1, 4.4, 1.4],
[5.6, 3. , 4.5, 1.5],
```

```
[5.8, 2.7, 4.1, 1. ],
[6.2, 2.2, 4.5, 1.5],
[5.6, 2.5, 3.9, 1.1],
[5.9, 3.2, 4.8, 1.8],
[6.1, 2.8, 4. , 1.3],
[6.3, 2.5, 4.9, 1.5],
[6.1, 2.8, 4.7, 1.2],
[6.4, 2.9, 4.3, 1.3],
[6.6, 3. , 4.4, 1.4],
[6.8, 2.8, 4.8, 1.4],
[6.7, 3. , 5. , 1.7],
[6. , 2.9, 4.5, 1.5],
[5.7, 2.6, 3.5, 1. ],
[5.5, 2.4, 3.8, 1.1],
[5.5, 2.4, 3.7, 1. ],
[5.8, 2.7, 3.9, 1.2],
[6. , 2.7, 5.1, 1.6],
[5.4, 3. , 4.5, 1.5],
[6. , 3.4, 4.5, 1.6],
[6.7, 3.1, 4.7, 1.5],
[6.3, 2.3, 4.4, 1.3],
[5.6, 3. , 4.1, 1.3],
[5.5, 2.5, 4. , 1.3],
[5.5, 2.6, 4.4, 1.2],
[6.1, 3. , 4.6, 1.4],
[5.8, 2.6, 4. , 1.2],
[5. , 2.3, 3.3, 1. ],
[5.6, 2.7, 4.2, 1.3],
[5.7, 3. , 4.2, 1.2],
[5.7, 2.9, 4.2, 1.3],
[6.2, 2.9, 4.3, 1.3],
[5.1, 2.5, 3. , 1.1],
[5.7, 2.8, 4.1, 1.3],
[6.3, 3.3, 6. , 2.5],
[5.8, 2.7, 5.1, 1.9],
[7.1, 3. , 5.9, 2.1],
[6.3, 2.9, 5.6, 1.8],
[6.5, 3. , 5.8, 2.2],
[7.6, 3. , 6.6, 2.1],
[4.9, 2.5, 4.5, 1.7],
[7.3, 2.9, 6.3, 1.8],
[6.7, 2.5, 5.8, 1.8],
[7.2, 3.6, 6.1, 2.5],
[6.5, 3.2, 5.1, 2. ],
[6.4, 2.7, 5.3, 1.9],
[6.8, 3. , 5.5, 2.1],
[5.7, 2.5, 5. , 2. ],
[5.8, 2.8, 5.1, 2.4],
[6.4, 3.2, 5.3, 2.3],
[6.5, 3. , 5.5, 1.8],
[7.7, 3.8, 6.7, 2.2],
[7.7, 2.6, 6.9, 2.3],
[6. , 2.2, 5. , 1.5],
[6.9, 3.2, 5.7, 2.3],
[5.6, 2.8, 4.9, 2. ],
[7.7, 2.8, 6.7, 2. ],
[6.3, 2.7, 4.9, 1.8],
[6.7, 3.3, 5.7, 2.1],
[7.2, 3.2, 6. , 1.8],
[6.2, 2.8, 4.8, 1.8],
[6.1, 3. , 4.9, 1.8],
[6.4, 2.8, 5.6, 2.1],
[7.2, 3. , 5.8, 1.6],
[7.4, 2.8, 6.1, 1.9],
[7.9, 3.8, 6.4, 2. ],
```

```
        [6.4, 2.8, 5.6, 2.2],
        [6.3, 2.8, 5.1, 1.5],
        [6.1, 2.6, 5.6, 1.4],
        [7.7, 3. , 6.1, 2.3],
        [6.3, 3.4, 5.6, 2.4],
        [6.4, 3.1, 5.5, 1.8],
        [6. , 3. , 4.8, 1.8],
        [6.9, 3.1, 5.4, 2.1],
        [6.7, 3.1, 5.6, 2.4],
        [6.9, 3.1, 5.1, 2.3],
        [5.8, 2.7, 5.1, 1.9],
        [6.8, 3.2, 5.9, 2.3],
        [6.7, 3.3, 5.7, 2.5],
        [6.7, 3. , 5.2, 2.3],
        [6.3, 2.5, 5. , 1.9],
        [6.5, 3. , 5.2, 2. ],
        [6.2, 3.4, 5.4, 2.3],
        [5.9, 3. , 5.1, 1.8]]),
 'target': 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, 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, 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, 2, 2, 2, 2]),
 'frame': None,
 'target_names': array(['setosa', 'versicolor', 'virginica'], dtype='<U10'),
 'DESCR': '.. _iris_dataset:\n\nIris plants dataset\n--------------------\n\n**Data Set
Characteristics:**\n\n    :Number of Instances: 150 (50 in each of three classes)\n    :
Number of Attributes: 4 numeric, predictive attributes and the class\n    :Attribute Inf
ormation:\n        - sepal length in cm\n        - sepal width in cm\n        - petal le
ngth in cm\n        - petal width in cm\n        - class:\n                - Iris-Setosa
\n                - Iris-Versicolour\n                - Iris-Virginica\n
\n    :Summary Statistics:\n\n    ============== ==== ==== ======= ===== ==============
=====\n                    Min  Max   Mean    SD   Class Correlation\n    ==============
==== ==== ======= ===== ====================\n    sepal length:   4.3  7.9   5.84   0.83
0.7826\n    sepal width:    2.0  4.4   3.05   0.43   -0.4194\n    petal length:   1.0
6.9   3.76   1.76    0.9490  (high!)\n    petal width:    0.1  2.5   1.20   0.76    0.95
65  (high!)\n    ============== ==== ==== ======= ===== ====================\n\n    :Mis
sing Attribute Values: None\n    :Class Distribution: 33.3% for each of 3 classes.\n
:Creator: R.A. Fisher\n    :Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)\n
:Date: July, 1988\n\nThe famous Iris database, first used by Sir R.A. Fisher. The datase
t is taken\nfrom Fisher\'s paper. Note that it\'s the same as in R, but not as in the UC
I\nMachine Learning Repository, which has two wrong data points.\n\nThis is perhaps the
best known database to be found in the\npattern recognition literature.  Fisher\'s paper
is a classic in the field and\nis referenced frequently to this day.  (See Duda & Hart,
for example.)  The\ndata set contains 3 classes of 50 instances each, where each class r
efers to a\ntype of iris plant.  One class is linearly separable from the other 2; the\n
latter are NOT linearly separable from each other.\n\n.. topic:: References\n\n   - Fish
er, R.A. "The use of multiple measurements in taxonomic problems"\n     Annual Eugenics,
7, Part II, 179-188 (1936); also in "Contributions to\n     Mathematical Statistics" (Jo
hn Wiley, NY, 1950).\n   - Duda, R.O., & Hart, P.E. (1973) Pattern Classification and Sc
ene Analysis.\n     (Q327.D83) John Wiley & Sons.  ISBN 0-471-22361-1.  See page 218.\n
- Dasarathy, B.V. (1980) "Nosing Around the Neighborhood: A New System\n     Structure a
nd Classification Rule for Recognition in Partially Exposed\n     Environments".  IEEE T
ransactions on Pattern Analysis and Machine\n     Intelligence, Vol. PAMI-2, No. 1, 67-7
1.\n   - Gates, G.W. (1972) "The Reduced Nearest Neighbor Rule".  IEEE Transactions\n
on Information Theory, May 1972, 431-433.\n   - See also: 1988 MLC Proceedings, 54-64.
Cheeseman et al"s AUTOCLASS II\n     conceptual clustering system finds 3 classes in the
data.\n   - Many, many more ...',
 'feature_names': ['sepal length (cm)',
  'sepal width (cm)',
  'petal length (cm)',
  'petal width (cm)'],
```

```
 'filename': 'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\sklearn\\datasets\\data\\iris.csv'}
```

In [7]:
```
print(iris_dataset.keys())
```

```
dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename'])
```

In [8]:
```
print(iris_dataset['DESCR'][:193])
```

```
.. _iris_dataset:

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

**Data Set Characteristics:**

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

In [9]:
```
print(iris_dataset['target_names'])
```

```
['setosa' 'versicolor' 'virginica']
```

In [10]:
```
print(iris_dataset['feature_names'])
```

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

In [11]:
```
type(iris_dataset['data'])
```

Out[11]:
```
numpy.ndarray
```

In [12]:
```
iris_dataset['data'].shape
```

Out[12]:
```
(150, 4)
```

In [13]:
```
iris_dataset['data'][:5]
```

Out[13]:
```
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]])
```

In [14]:
```
type(iris_dataset['target'])
```

Out[14]:
```
numpy.ndarray
```

In [15]:
```
iris_dataset['target'].shape
```

Out[15]:
```
(150,)
```

In [16]:
```
iris_dataset['target']
# 0 means setosa, 1 mean versicolor, and 2 means virginica
```

Out[16]:
```
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, 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])
```

Training and testing data

In [17]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    iris_dataset['data'], iris_dataset['target'], random_state=0)
```

In [18]:
```
print(X_train.shape)
print(y_train.shape)
```

```
(112, 4)
(112,)
```

In [19]:
```
print(X_test.shape)
print(y_test.shape)
```

```
(38, 4)
(38,)
```

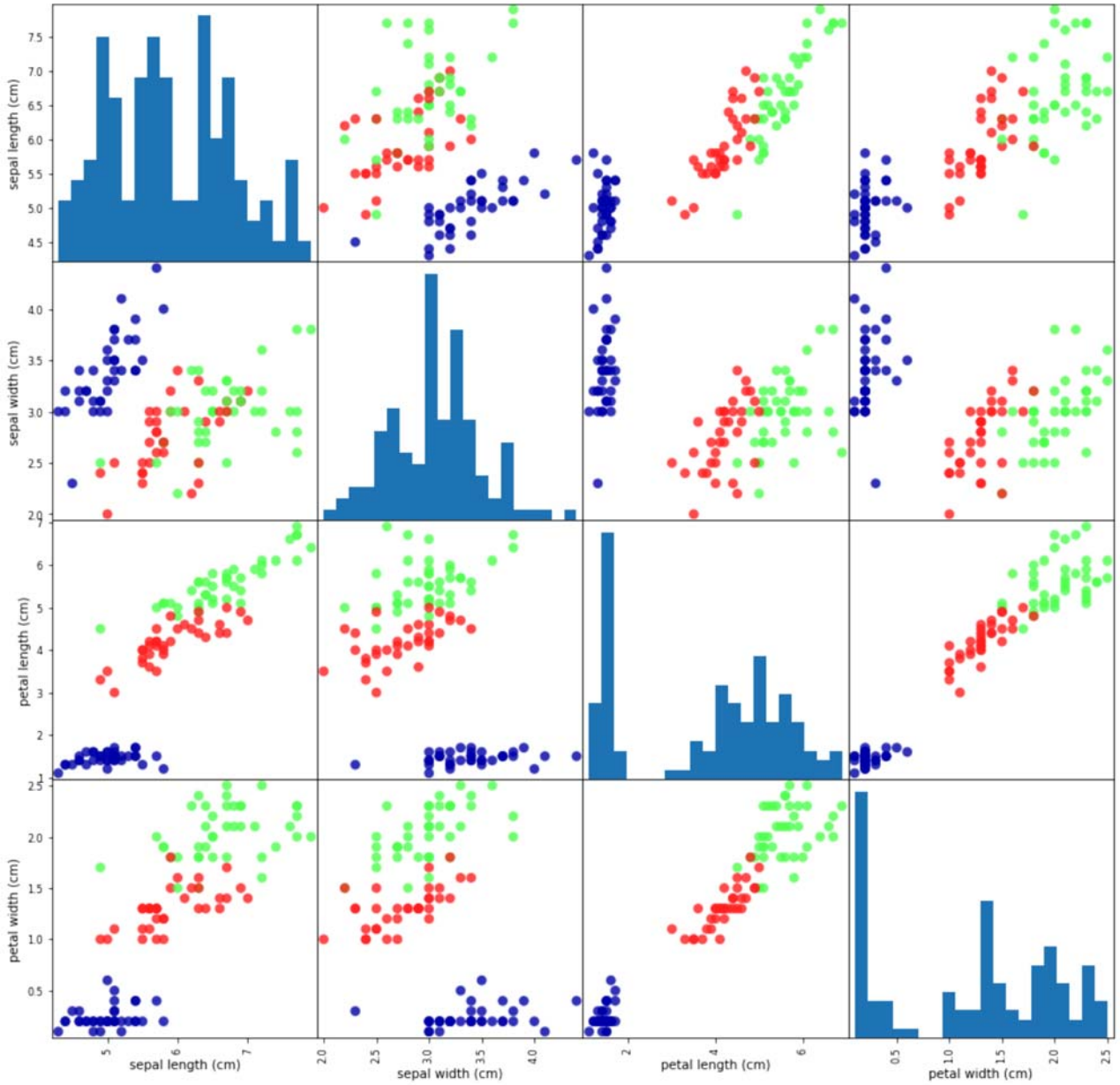
In [20]:
```
iris_dataframe = pd.DataFrame(X_train, columns = iris_dataset.feature_names)
iris_dataframe
```

Out[20]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.9	3.0	4.2	1.5
1	5.8	2.6	4.0	1.2
2	6.8	3.0	5.5	2.1
3	4.7	3.2	1.3	0.2
4	6.9	3.1	5.1	2.3
...	...	...	...	...
107	4.9	3.1	1.5	0.1
108	6.3	2.9	5.6	1.8
109	5.8	2.7	4.1	1.0
110	7.7	3.8	6.7	2.2
111	4.6	3.2	1.4	0.2

112 rows × 4 columns

In [21]:
```
#Pair plot
grr = scatter_matrix(iris_dataframe,
                     c=y_train,                #Specifies the color
                     figsize=(15,15),
                     marker='o',
                     s=60,                     #Size of marker
                     hist_kwds={'bins':20},
                     alpha=0.8,
                     cmap=mglearn.cm3)         #custom colormap from the mglearn library
```



Building Your First Model: k_Nearest Neighbors

In [22]:
```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=3)
```

In [23]:
```
knn.fit(X_train, y_train) #Train data itself
```

Out[23]:
```
KNeighborsClassifier(n_neighbors=3)
```

In [24]:
```
y_pred = knn.predict(X_test)
```

In [25]:
```
print("Test set score: {:.2f}".format(knn.score(X_test, y_test)))
```

```
Test set score: 0.97
```

In [26]:
```
from sklearn.metrics import accuracy_score

# Check the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print("Model accuracy: {:.2f}" .format(accuracy))
```

```
Model accuracy: 0.97
```