Machine Learning Fundamentals Lab-1

Name: Gaurav Prasanna
Reg No: 19BEC1315

Aim:

a) To read in dataset in format of csv file and visualize the data using Pandas and Matplotlib libraries.
b) To show and understand different Performance measures for knowing how well a model performs for the given dataset.

Software Required:

1) Anaconda Navigator
2) Jupyter Notebook

Libraries Required: Numpy, Sci-kit Learn, Pandas, Matplotlib

Code And Outputs: Part a)

In [3]:
```
# import sys
# Python version
import sys
print('Python: {}'.format(sys.version))
# scipy
import scipy
print('scipy: {}'.format(scipy.__version__))
# numpy
import numpy
print('numpy: {}'.format(numpy.__version__))
# matplotlib
import matplotlib
print('matplotlib: {}'.format(matplotlib.__version__))
# pandas
import pandas
print('pandas: {}'.format(pandas.__version__))
# scikit-learn
import sklearn
print('sklearn: {}'.format(sklearn.__version__))
```

```
Python: 3.8.10 (default, May 19 2021, 13:12:57) [MSC v.1916 64 bit (AMD64)]
scipy: 1.6.2
numpy: 1.21.1
matplotlib: 3.4.2
pandas: 1.2.5
sklearn: 0.24.2
```

In [4]:
```
# Load libraries
from pandas import read_csv
from pandas.plotting import scatter_matrix
from matplotlib import pyplot
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
```

In [5]:
```
# Load dataset
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/iris.csv"
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']
dataset = read_csv(url, names=names)
```

In [6]:
```
#dataset = read_csv("F:\\VITC\\Semesters\\Fall 20-21\\MLF Lab\Python files\\iris
```

In [7]:
```
# shape
print(dataset.shape)
```

```
(150, 5)
```

In [8]:
```
# head
print(dataset.head(20))
```

```
    sepal-length  sepal-width  petal-length  petal-width        class
0            5.1          3.5           1.4          0.2  Iris-setosa
1            4.9          3.0           1.4          0.2  Iris-setosa
2            4.7          3.2           1.3          0.2  Iris-setosa
3            4.6          3.1           1.5          0.2  Iris-setosa
4            5.0          3.6           1.4          0.2  Iris-setosa
5            5.4          3.9           1.7          0.4  Iris-setosa
6            4.6          3.4           1.4          0.3  Iris-setosa
7            5.0          3.4           1.5          0.2  Iris-setosa
8            4.4          2.9           1.4          0.2  Iris-setosa
9            4.9          3.1           1.5          0.1  Iris-setosa
10           5.4          3.7           1.5          0.2  Iris-setosa
11           4.8          3.4           1.6          0.2  Iris-setosa
12           4.8          3.0           1.4          0.1  Iris-setosa
13           4.3          3.0           1.1          0.1  Iris-setosa
14           5.8          4.0           1.2          0.2  Iris-setosa
15           5.7          4.4           1.5          0.4  Iris-setosa
16           5.4          3.9           1.3          0.4  Iris-setosa
17           5.1          3.5           1.4          0.3  Iris-setosa
18           5.7          3.8           1.7          0.3  Iris-setosa
19           5.1          3.8           1.5          0.3  Iris-setosa
```

In [13]:
```
dataset
```

Out[13]:

	sepal-length	sepal-width	petal-length	petal-width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

In [16]:
```
# descriptions
print(dataset.describe())
```

```
       sepal-length  sepal-width  petal-length  petal-width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.054000      3.758667     1.198667
std        0.828066     0.433594      1.764420     0.763161
min        4.300000     2.000000      1.000000     0.100000
25%        5.100000     2.800000      1.600000     0.300000
50%        5.800000     3.000000      4.350000     1.300000
75%        6.400000     3.300000      5.100000     1.800000
max        7.900000     4.400000      6.900000     2.500000
```

In [17]:
```
# class distribution
print(dataset.groupby('class').size())
```

```
class
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
dtype: int64
```

In [18]:
```
from sklearn import datasets
iris = datasets.load_iris()
```

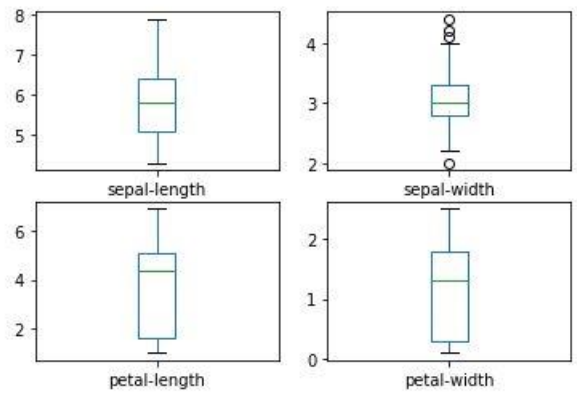
In [19]:
```
print(iris.data)
```

```
[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]
```

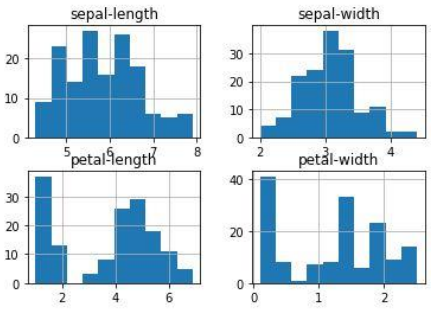
In [20]:
```
print(iris.target)
```

```
[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]
```

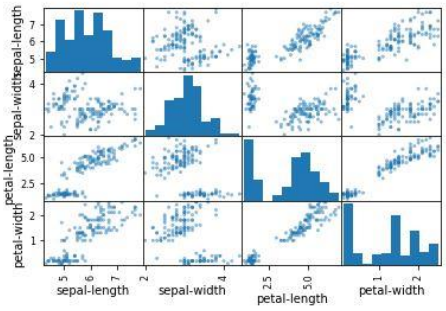
In [21]:
```
# box and whisker plots
dataset.plot(kind='box', subplots=True, layout=(2,2), sharex=False, sharey=False
pyplot.show()
```



In [22]:
```
# histograms
dataset.hist()
pyplot.show()
```



In [23]:
```
# scatter plot matrix
scatter_matrix(dataset)
pyplot.show()
```



In [24]:
```
# Split-out validation dataset
array = dataset.values
X = array[:,0:4]
y = array[:,4]
X_train, X_validation, Y_train, Y_validation = train_test_split(X, y, test_size=
```

In [25]:
```
X
```

Out[25]:
```
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3.0, 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5.0, 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5.0, 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, 3.0, 1.4, 0.1],
       [4.3, 3.0, 1.1, 0.1],
       [5.8, 4.0, 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],
```

Part b)

```
In [2]: import sklearn
        from sklearn.metrics import *

In [3]: X_actual = [1, 1, 0, 1, 0, 0, 1, 0, 0, 0]
        y_pred = [1, 0, 1, 1, 1, 0, 1, 1, 0, 0]

In [4]: results = confusion_matrix(X_actual, y_pred)
        print(results)

        [[3 3]
         [1 3]]

In [5]: accuracy = accuracy_score(X_actual, y_pred)
        print(accuracy)

        0.6

In [7]: print(classification_report(X_actual, y_pred))

                      precision    recall  f1-score   support

                   0       0.75      0.50      0.60         6
                   1       0.50      0.75      0.60         4

            accuracy                           0.60        10
           macro avg       0.62      0.62      0.60        10
        weighted avg       0.65      0.60      0.60        10

In [8]: print(roc_auc_score(X_actual, y_pred))

        0.625

In [9]: print(log_loss(X_actual, y_pred))

        13.815750437193334

In [10]: X_reg = [5, -1, 2, 10]
         y_reg_pred = [3.5, -0.9, 2, 9.9]

In [11]: print('R Squared =', r2_score(X_reg, y_reg_pred))

         R Squared = 0.9656060606060606

In [12]: print('MAE =', mean_absolute_error(X_reg, y_reg_pred))

         MAE = 0.42499999999999993

In [13]: print('MSE =', mean_squared_error(X_reg, y_reg_pred))

         MSE = 0.5674999999999999
```

Inference: So from the above code, we have imported the required libraries for performing various functionalities, for the first part we read in the iris dataset as csv file using pd.read_csv a function in pandas library

and using scatter plot function in pandas and pyplot function in matplotlib to visualize the dataset. In next part we have used sci-kit learn to understand about various performance measures, these measures helps us to understand how well a model works with different measures.

Result: All the code and output have been executed and shown for the respective cells in jupyter notebook and calculated performance measures are also shown.