LA03_Ex3_DataUnderstanding

April 28, 2018

1 Data Understanding

Iris dataset

```
In [53]: import pandas as pd
         import sklearn
         import matplotlib.pyplot as plt
```

```
In [3]: import pandas as pd
        url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
        names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']
        dataset = pd.read_csv(url, names=names)
```

1.1 Task 1: Summary of the Dataset

- Dimensions of the dataset.
- Peek at the data itself.
- Statistical summary of all attributes.
- Breakdown of the data by the class variable.

```
In [ ]: # Ref: https://machinelearningmastery.com/machine-learning-in-python-step-by-step/
```

```
In [4]: # Dimensions of the dataset
        print(dataset.shape)
```

```
(150, 5)
```

```
In [7]: # Peek at the data itself
        print(dataset.head(10))
```

```
   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
```

```
In [8]: # Statistical summary of all attributes
        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
```

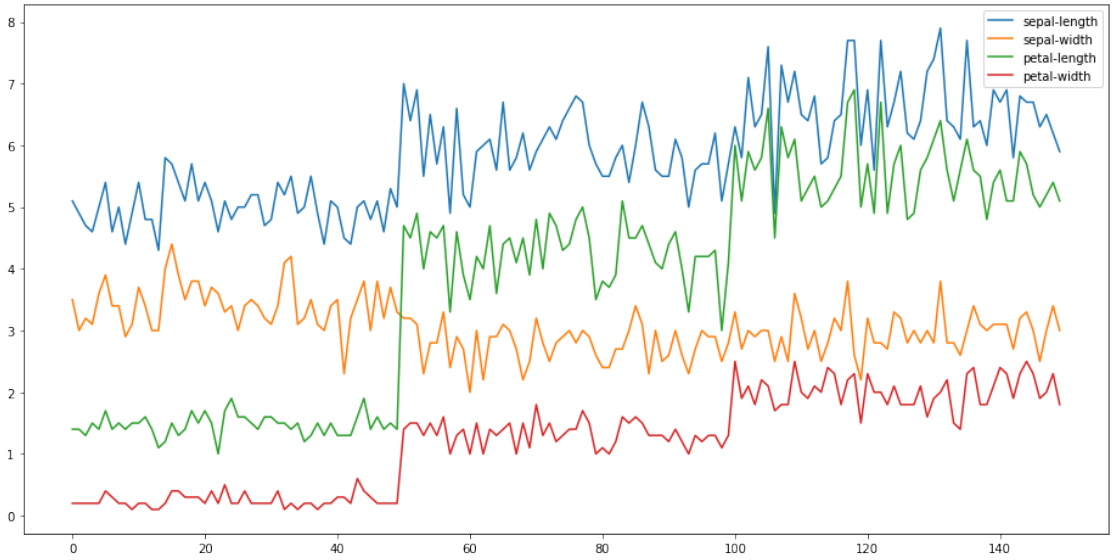
```
In [11]: # Breakdown of the data by the class variable
         print(dataset.groupby('class').size())
```

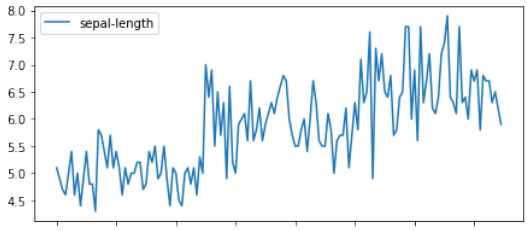
```
class
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
dtype: int64
```

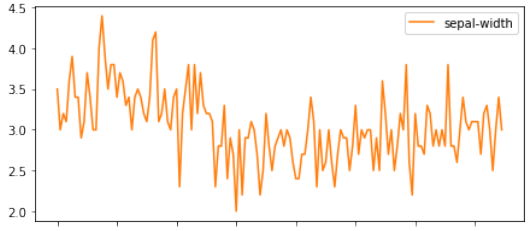
1.2 Task 2: Data Visualization

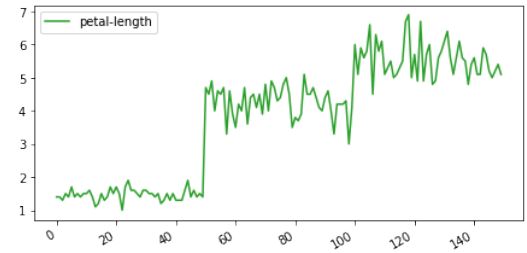
- Univariate plots, visualisation of each individual feature for better understand.
- Multivariate plots, visualisation relationships between attributes.

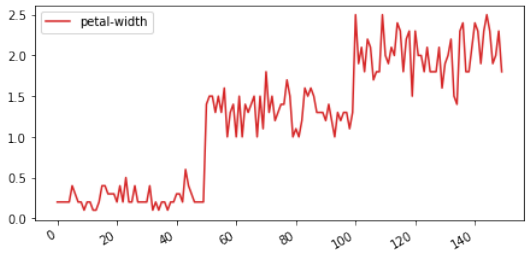
```
In [74]: # Univariate plots, visualisation of each individual feature for better understand
         dataset.plot(figsize=(16, 8))
         dataset.plot(subplots=True, layout=(2,2), figsize=(16, 8))
         dataset.hist(figsize=(16, 8))

         plt.show()
```

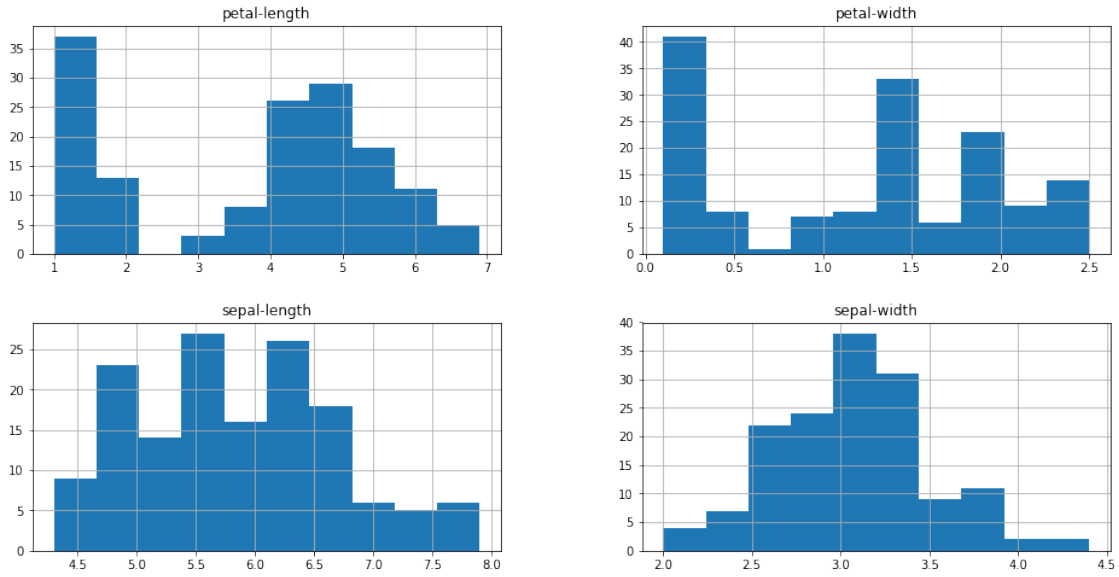

sepal-length
sepal-width
petal-length
petal-width


sepal-length

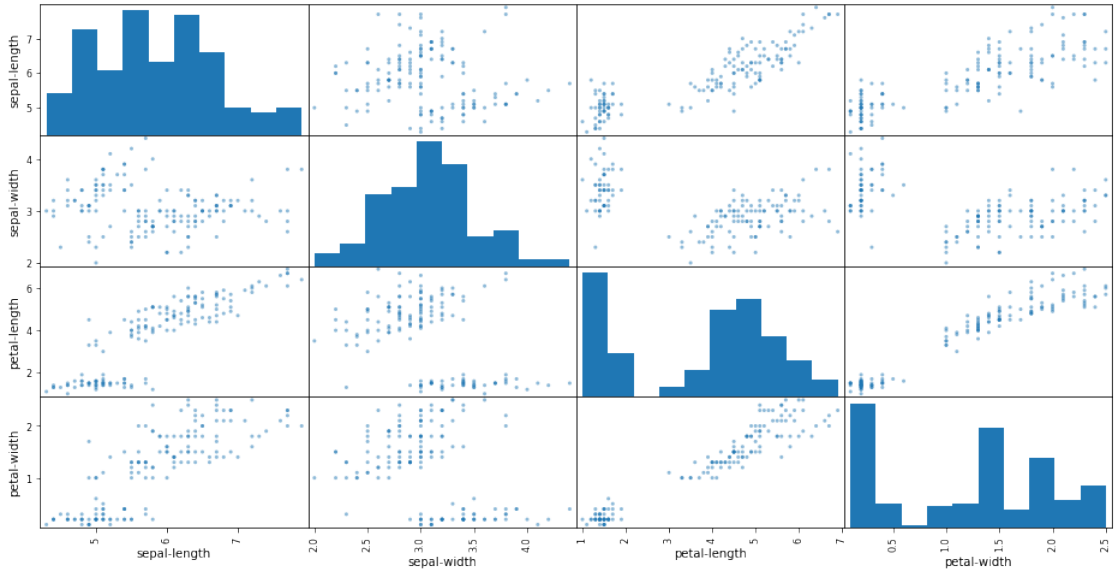

sepal-width


petal-length


petal-width



In [75]:
```
# Multivariate plots, visualisation relationships between attributes
pd.plotting.scatter_matrix(dataset, figsize=(16, 8))
plt.show()
```



In [38]:
```
dataset.corr()
```

Out[38]:

	sepal-length	sepal-width	petal-length	petal-width
sepal-length	1.000000	-0.109369	0.871754	0.817954

```
sepal-width      -0.109369     1.000000     -0.420516     -0.356544
petal-length      0.871754    -0.420516      1.000000      0.962757
petal-width       0.817954    -0.356544      0.962757      1.000000
```

1.3 Task 3: Validation set

We will split the loaded dataset into two, 80% of which we will use to train our models and 20% that we will hold back as a validation dataset.

```
In [56]: import sklearn.model_selection

In [70]: [training_set, test_set] = sklearn.model_selection.train_test_split(dataset, test_size=

         print 'Dataset shape: ', dataset.shape
         print 'Training set shape:', training_set.shape
         print 'Test set shape: ', test_set.shape

Dataset shape:  (150, 5)
Training set shape: (120, 5)
Test set shape:  (30, 5)
```