

# Machine learning 1st project

Firstly going through the versions of all the python libraries in the PC required for the Machine Learning project

```
In [12]: import sys
print('Python: {}'.format(sys.version))
import scipy
print('Scipy: {}'.format(scipy.__version__))
import numpy
print('Numpy: {}'.format(numpy.__version__))
import matplotlib
print('matplotlib: {}'.format(matplotlib.__version__))
import pandas
print('pandas: {}'.format(pandas.__version__))
import sklearn
print('sklearn: {}'.format(sklearn.__version__))
```

```
Python: 3.7.6 (default, Jan 8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]
Scipy: 1.4.1
Numpy: 1.18.1
matplotlib: 3.1.3
pandas: 1.0.1
sklearn: 0.22.1
```

Importing all the library modules.

```
In [ ]: 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
```

Loading the dataset from the source directly, and also calling for the variety of attributes present in it.

```
In [19]: 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)
```

calling the print function for getting the information of shape of the dataset loaded.

```
In [20]: print(dataset.shape)
```

```
(150, 5)
```

printing the head of 20 values from the dataset.

```
In [21]: 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

Describing the dataset loaded (for the total dataset).

```
In [22]: 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

Grouping the dataset by the class and printing the size of each class.

```
In [23]: print(dataset.groupby('class').size())
```

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

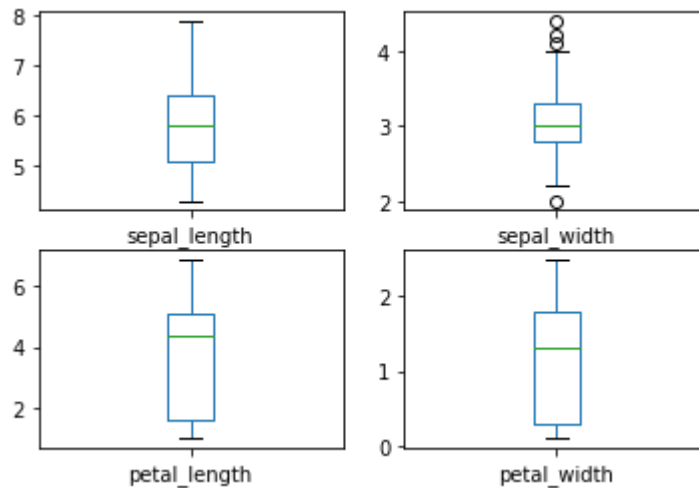
Calling the above cells all together for the general representation.

```
In [24]: from pandas import read_csv
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)
print(dataset.shape)
print(dataset.head(20))
print(dataset.describe())
print(dataset.groupby('class').size())
```

```
(150, 5)
   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
   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
class
Iris-setosa      50
Iris-versicolor  50
Iris-virginica   50
dtype: int64
```

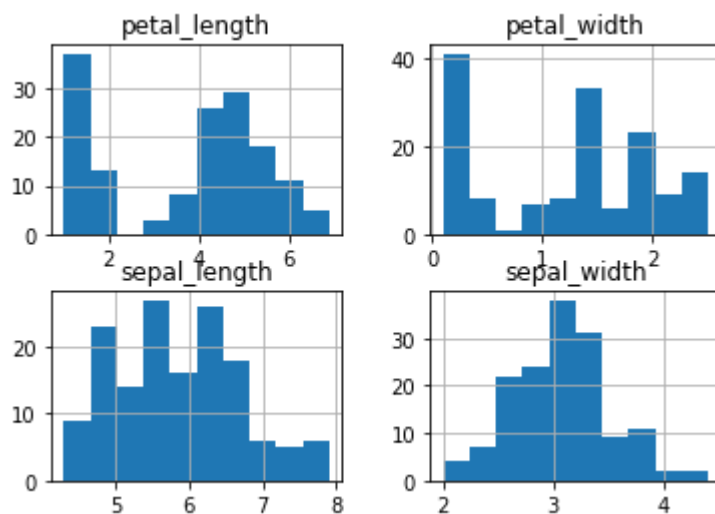
Plotting the dataset from the information. and using the matplotlib module.

```
In [25]: dataset.plot(kind='box',subplots=True,layout=(2,2),sharex=False, sharey=False)  
pyplot.show()
```



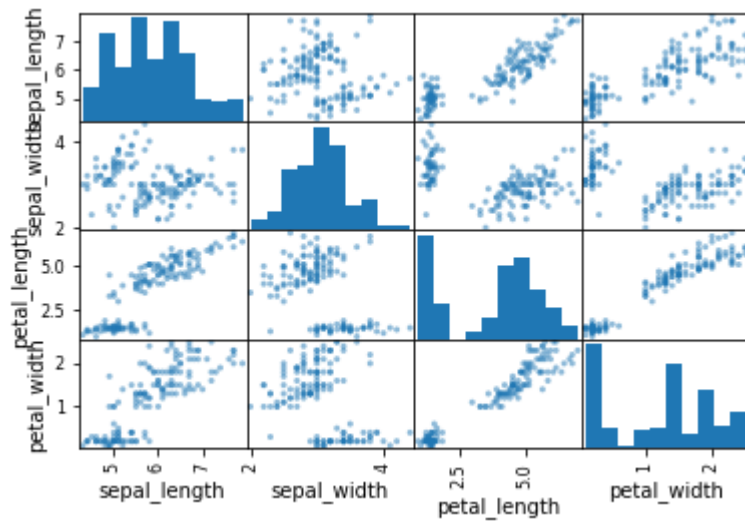
Also creating the histogram and plotting it on the graphs for the each attribute.

```
In [26]: dataset.hist()  
pyplot.show()
```



plotting the scatter matrix for the dataset using with the 4 attributes.

```
In [27]: scatter_matrix(dataset)
pyplot.show()
```



calling above called cells all together for general representation.

```
In [28]: from pandas import read_csv
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)
print(dataset.shape)
print(dataset.head(20))
print(dataset.describe())
print(dataset.groupby('class').size())
dataset.hist()
pyplot.show()
scatter_matrix(dataset)
pyplot.show()
```

(150, 5)

	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

	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

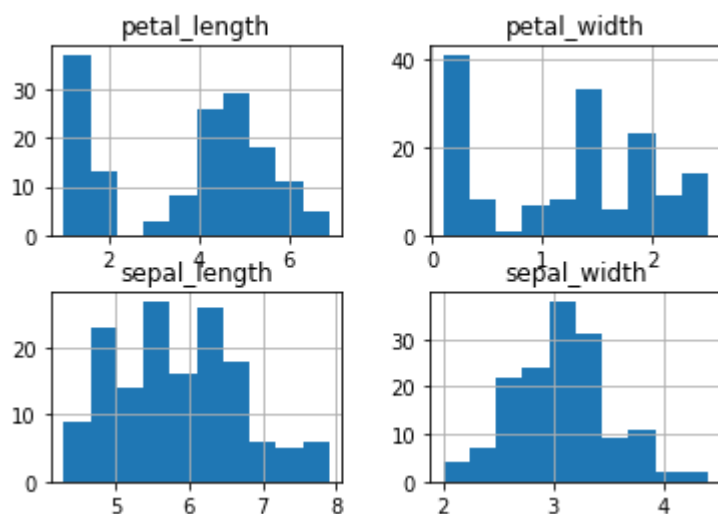
class

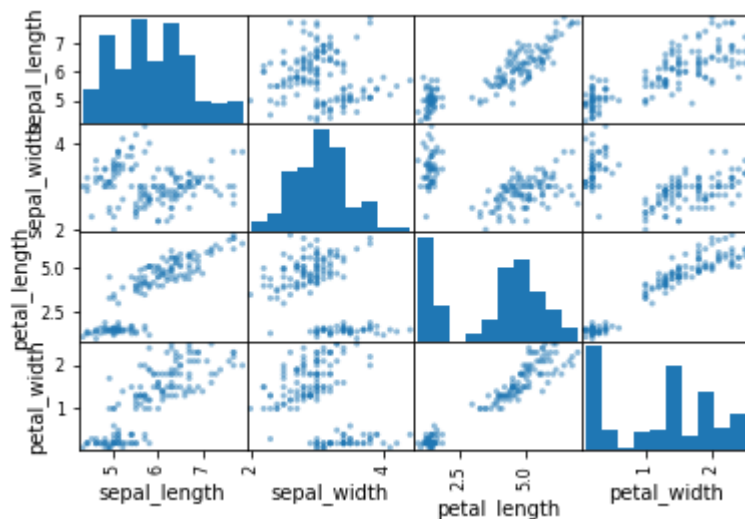
Iris-setosa 50

Iris-versicolor 50

Iris-virginica 50

dtype: int64





Creating the x,y training and validation data.

```
In [29]: 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=0.20,
, random_state=1)
```

using the 6 models (logistic regression, linear discriminant analysis, K neighbor Classifier, Decision Tree Classifier, Gaussian NB and Support Vector Machine.)

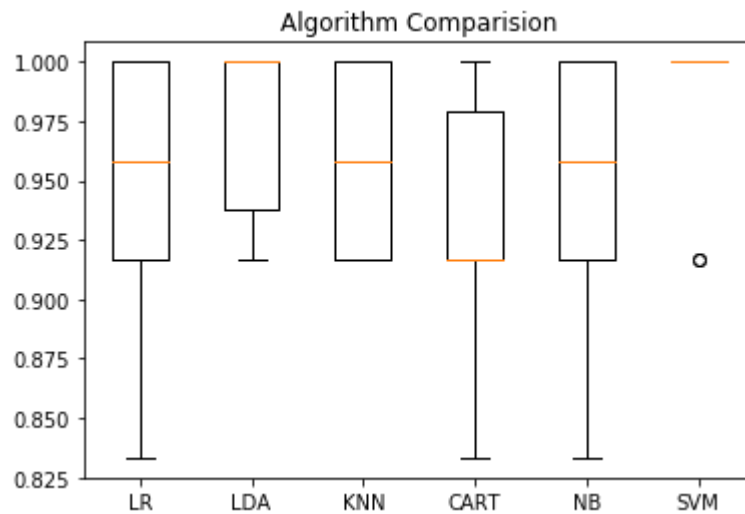
```
In [31]: models=[]
models.append(('LR', LogisticRegression(solver='liblinear',multi_class='ovr'
)))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN',KNeighborsClassifier()))
models.append(('CART',DecisionTreeClassifier()))
models.append(('NB',GaussianNB()))
models.append(('SVM',SVC(gamma='auto')))
results=[]
names=[]
for name,model in models:
    kfold=StratifiedKFold(n_splits=10,random_state=1,shuffle=True)
    cv_results = cross_val_score(model, X_train, Y_train, cv=kfold, scoring='a
ccuracy')
    results.append(cv_results)
    names.append(name)
    print('%s: %f (%f)' % (name, cv_results.mean(), cv_results.std()))

LR: 0.941667 (0.065085)
LDA: 0.975000 (0.038188)
KNN: 0.958333 (0.041667)
CART: 0.933333 (0.050000)
NB: 0.950000 (0.055277)
SVM: 0.983333 (0.033333)
```

plotting the 6 models in the plot.



```
In [32]: pyplot.boxplot(results,labels=names)
pyplot.title('Algorithm Comparision')
pyplot.show()
```



make predictions, and then print the accuracy score along with the confusion matrix and classification report.

```
In [34]: model=SVC(gamma='auto')
model.fit(X_train,Y_train)
predictions=model.predict(X_validation)
```

```
In [35]: print(accuracy_score(Y_validation,predictions))
print(confusion_matrix(Y_validation,predictions))
print(classification_report(Y_validation,predictions))
```

```
0.9666666666666667
```

```
[[11  0  0]
 [ 0 12  1]
 [ 0  0  6]]
```

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	11
Iris-versicolor	1.00	0.92	0.96	13
Iris-virginica	0.86	1.00	0.92	6
accuracy			0.97	30
macro avg	0.95	0.97	0.96	30
weighted avg	0.97	0.97	0.97	30

```
In [ ]:
```