**EDINBURG FRUIT SURVEY**

The input dataset created by the Edinburg professor. The dataset download link finds below

<https://github.com/susanli2016/Machine-Learning-with-Python/blob/master/fruit_data_with_colors.txt>

**Looking into Data: -**

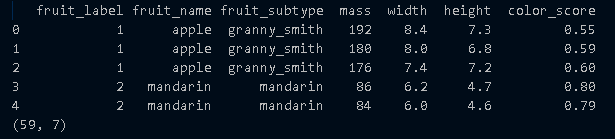
import pandas as pd

import matplotlib as plt

fruit = pd.read\_table(*r*"C:\Users\n0278588\GITHUB-Local\myML\Proj2-FruitSurvey-SimpleClassificationModels\InputDataSet.txt")

print(fruit.head())

print(fruit.shape)



The dataset has 59 rows and 7 feature vectors or columns.

Unique Fruits (instances) in dataset are

print(fruit['fruit\_name'].unique())

['apple' 'mandarin' 'orange' 'lemon']

Grouping the fruits by fruit name

print(fruit.groupby('fruit\_name').size())

fruit\_name

apple 19

lemon 16

mandarin 5

orange 19

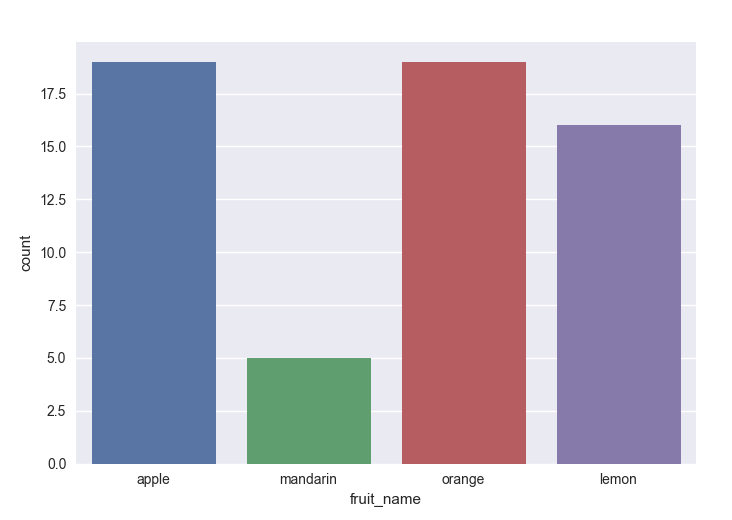
dtype: int64

Plotting the fruit name feature and see in graph for better data visualization. We will use Seaborn one of the most popular Python statistical data visualization tool

import seaborn as sns

sns.countplot(fruit['fruit\_name'],*label* = "Count")

plt.show()



**Visualizing the input variable distribution: -**

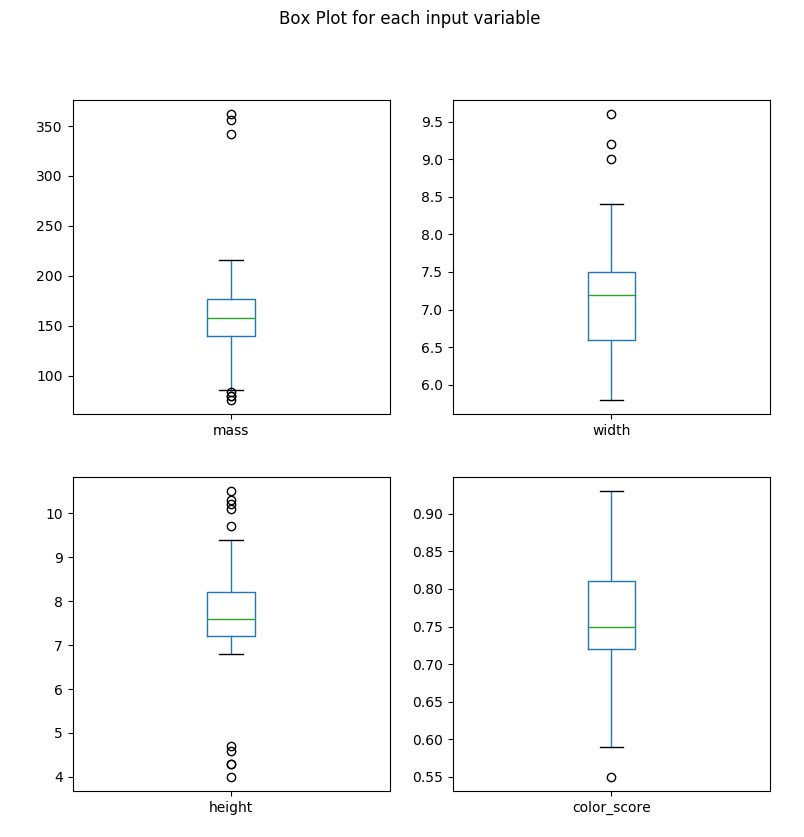
fruit.drop('fruit\_label', *axis*=1).plot(*kind*='box', *subplots*=True, *layout*=(2,2), *sharex*=False, *sharey*=False, *figsize*=(9,9),

*title*='Box Plot for each input variable')

#plt.savefig('fruits\_box')

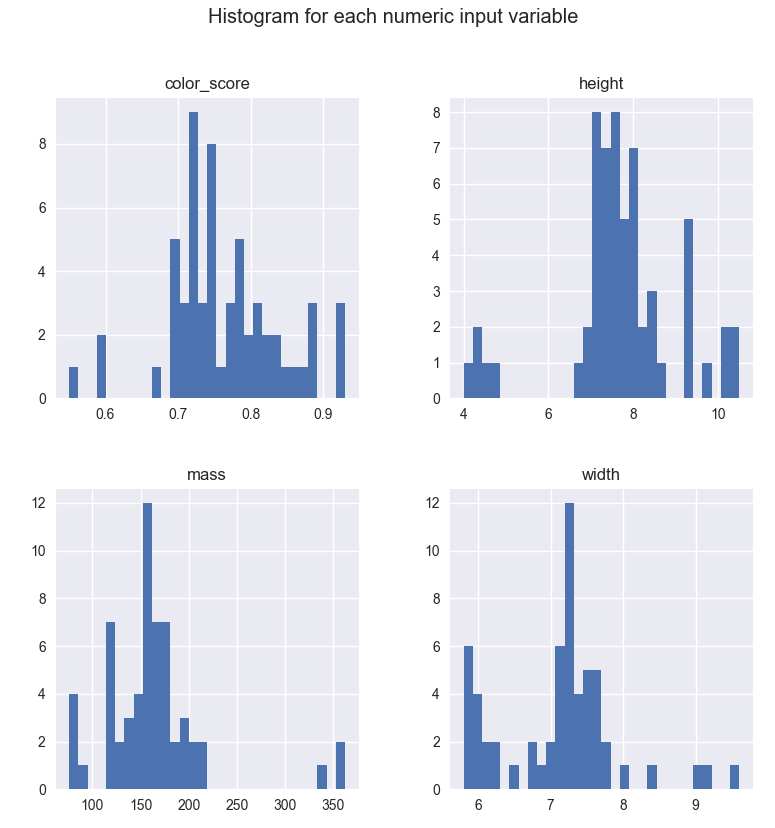
plt.show()

Only plotting happening for Features with numerical data.



**Visualizing in histogram: -**

The histogram shows there is some correlation between mass and width. Both the plots looks quite similar.



**SCATTER MATRIX FOR EACH INPUT VARIABLE: -**

from pandas.tools.plotting import scatter\_matrix

from matplotlib import cm

feature\_names = ['mass','width','height','color\_score']

X = fruit[feature\_names]

y = fruit['fruit\_label']

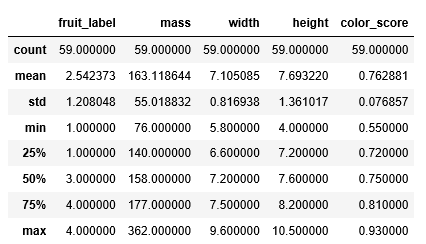
cmap = cm.get\_cmap('gnuplot')

scatter = pd.scatter\_matrix(X, *c*=y, *marker* = 'o', *s*=40, *hist\_kwds*=('bins':15),*figsize*=(9,9),*cmap*=cmap)

plt.suptitle('Scatter matrix for each input varible')

plt.savefig('fruits\_scatter\_matrix')

The numerical values are not in proper scale, we have to scale it for Train and test data



Creating the scaling for Train and Test data

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import MinMaxScaler

feature\_names = ['mass','width','height','color\_score']

X = fruit[feature\_names]

y = fruit['fruit\_label']

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,*random\_state* = 0)

scaler = MinMaxScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.fit\_transform(X\_test)

**BUILDING MODELS**