**Exercise Number 1:**

**Use Matplotlib libraries to plot this dataset. you can use a scatter plot to visualize the data, since it has only two properties to plot (profit and population). Plot population on x axis and profit on y- axis. Label the dataset and plot graph**

import numpy as np

import matplotlib.pyplot as plt

with open("ex1data1.txt") as f:

data = f.read()

data = data.split('\n')

x = [row.split(',')[0] for row in data]

y = [row.split(',')[1] for row in data]

axes=plt.axes()

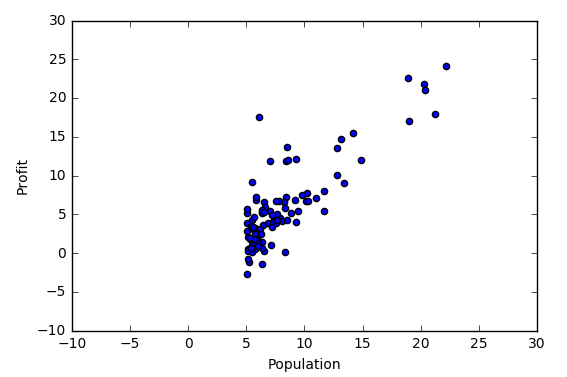
axes.set\_xlim([-10,30])

axes.set\_ylim([-10,30])

plt.xlabel('Population')

plt.ylabel('Profit')

plt.scatter(x,y)

plt.show()

**Exercise Number 2:**

**Use Scikit linear regression algorithm to predict profit as a function of population. The function used out here is y = Mx + C, where the parameter M & C will provide the equation for best fit line (linear regression). Further, y – is profit and X – is population. So purpose of this exercise is derive a simple equation which define profit as a function of population.**

***Exercise 3 :*use the parameters obtained for linear regression to plot the linear fit in the same scatter plot used to visualize the data. Purpose is the draw the line y = MX + C, obtained above. You will be able to visualize that straight line is the best fit possible given this data set. For a better representation we will need to fit through a quardratic function of the form y = AX2  + MX + C or an higher order polynomial.**

=============================================================

%matplotlib notebook

import numpy as np

import matplotlib.pyplot as plt

with open("ex1data1.txt") as f:

data = f.read()

data = data.split('\n')

X = [row.split(',')[0] for row in data]

Y = [row.split(',')[1] for row in data]

axes=plt.axes()

axes.set\_xlim([-10,30])

axes.set\_ylim([-10,30])

plt.xlabel('Population')

plt.ylabel('Profit')

plt.scatter(X,Y)

plt.show()

X=list(map(float,X))

Y=list(map(float,Y))

def slope\_intercept(x\_val,y\_val):

x=np.array(x\_val)

y=np.array(y\_val)

m=((np.mean(x)\*np.mean(y)) - np.mean(x\*y))/((np.mean(x)\*np.mean(x)) - np.mean(x\*x))

m=round(m,2)

b=(np.mean(y) - np.mean(x)\*m)

b=round(b,2)

return m,b

slope\_intercept(X,Y)

m,b=slope\_intercept(X,Y)

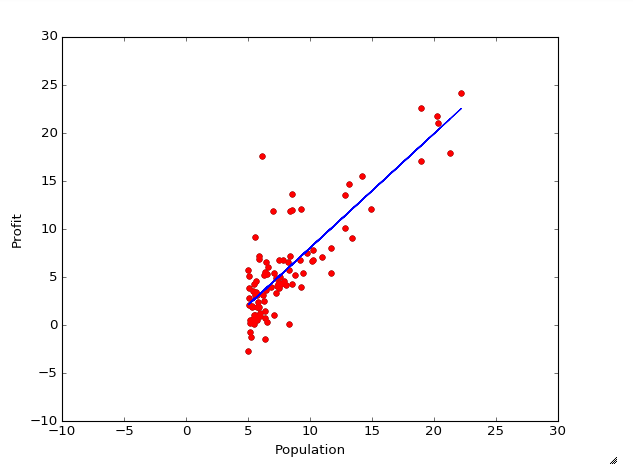
reg\_line=[(m\*x)+b for x in X]

plt.scatter(X,Y,color="red")

plt.plot(X,reg\_line)

plt.xlabel('Population')

plt.ylabel('Profit')

plt.show()