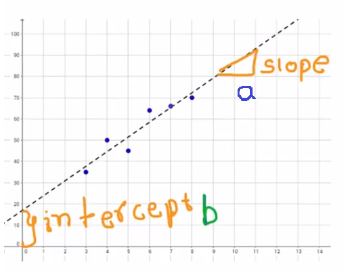
**Linear regression   
scikit learn - sklearn**



Y = ax + b   
a = slope - b = intercept

Our goal is to find the **best values** of slope (a) and intercept (b) to fit our data. The Linear Regression uses Ordinary Least Squares method to fit our data points,

height weight

4.0 -> 42  
4.5 -> 44  
5.0 -> 49  
5.2 -> 55  
5.4 -> 53  
5.8 -> 58  
6.1 -> 60  
6.2 -> 64  
6.4 -> 66  
6.8 -> 69

Jupyter

# I have some height and weight data of some people.   
# let’s use this data to do linear regression and try to predict weight of other people.

*from sklearn import linear\_model*

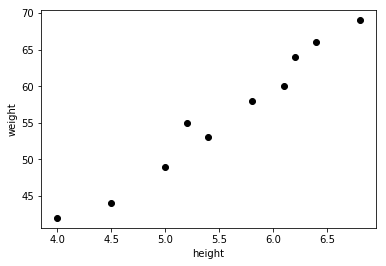
height=[[4.0],[4.5],[5.0],[5.2],[5.4],[5.8],[6.1],[6.2],[6.4],[6.8]]

weight=[ 42 , 44 , 49, 55 , 53 , 58 , 60 , 64 , 66 , 69]

print("height weight")

for row in **zip**(height, weight):

print (row[0][0],"->",row[1])



Import matplotlib.pyplot as plt

plt.scatter(height,weight,color='black')

plt.xlabel("height")

plt.ylabel("weight")

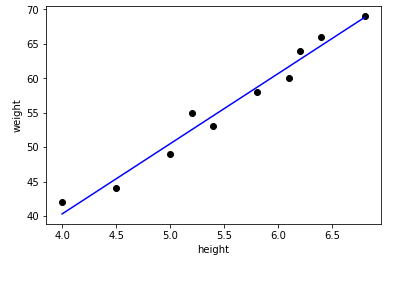
reg=linear\_model.LinearRegression()

reg.fit(height,weight)

m=reg.coef\_[0]

b=reg.intercept\_

print("slope=",m, "intercept=",b) >> slope= 10.1936218679 intercept= -0.4726651480



plt.scatter(height,weight,color='black')

predicted\_values = [reg.coef\_ \* i + reg.intercept\_ for i in height]

plt.plot(height, predicted\_values, 'b')

plt.xlabel("height")

plt.ylabel("weight")

*# Now we can go ahead and predict the weight of people whose data is not there with us:*

reg.predict(X=6.2) >> array([ 62.72779043])

reg.predict(X=8.0) >> array([ 81.07630979])