**8.1.3- Linear regression Mini project:**

Linear regression equation Y = Beta (B0) +B1\* X + €

X is known as independent variable, **explanatory variable, features, predictors**

Y is dependent variable, response variable

B0 is intercept and represents average of Y when all independent variables X are set to 0

B1 is the slope of the line, and represents average effect of one unit increase of X on YE

Assumptions made in Linear regressions: Y = Beta (B0) +B1\* X + €

1. Epsilon € : it is unobservable random variable that adds noise to the linear relation, € is assumed to be normally distributed with mean of 0
2. The residuals € are also assumed to be independently and identically distributed, i.e the residuals from one prediction do not have effect on another prediction

Our main focus in linear regression is to estimate B0 and B1, we will mostly use LEAST SQUARED method to estimate them, once we estimate B0 and B1, then we can calculate Y (estimated Y is called Y CAP), based on new values of X

LEAST SQUARED METHOD:

*ℒ*=∑*i*=1*Nϵ*2*i*

=∑*i*=1*N*(*yi*−*y*̂*i*)2

=∑*i*=1*N*(*yi*−(*β*0+*β*1*xi*))2

We want to find B0 and B1, that minimizes the squared error, so first we will do partial derivative w.r.t B0

If you do that you will get B0 = Y MEAN + B1\* X MEAN

Similarly, if take partial derivative w.r.t B1 and substituting above value, you will get following matrix notation

*β*̂=(*XTX*)−1*XTY*

**Linear regression using STATSMODEL:**

import statsmodels.api as sm

from statsmodels.formula.api import ols

m=ols(‘PRICE’ ~ ’RM’, bos).fit()

print(m.summary())

for statsmodels (ols or logit) calls you have a pandas data frame with column names that you will add to the formula.

You can force statsmodels to treat variables as categorical with the `C()` function, call numpy functions to transform data such as `np.log` for extremely-skewed data, or fit a model without an intercept by including `- 1` in the formula. For a quick run-down of further uses see the `statsmodels` [help page](http://statsmodels.sourceforge.net/devel/example\_formulas.html).

**Linear regression using SCIKIT LEARN:**

from sklearn.linear\_model import LinearRegression

X = bos.drop('PRICE', axis = 1)

# This creates a LinearRegression object

lm = LinearRegression()

lm

**8.1.4 – Logistic regression behind the scenes**

<http://www.stat.yale.edu/Courses/1997-98/101/ranvar.htm> -> Random variables, discrete random variable, continuous random variable, cumulative distribution function