**Linear Regression**

Linear regression assumes a linear relationship between input x and a single input y.

Y = B0 + B1\*X

Y = B0 + B1\*X1 + …. + Bn\*Xn

B0 : Intercept

B1, … , Bn : Coefficient

**\*\* You should work on Lab7\_Linear\_Regression.ipynb**

**\*\* Other files are for your review, but you can simply ignore them.**

To read a text file as a pandas dataframe use the following code:

>> df\_housing = pd.read\_csv("AmesHousing.txt", sep='\t')

**Ex-1 )** Refer to **Warm-Up: A Model with One Feature**

* Find the number of rows of your dataste
* Compute 80% or 90% of it, let's say “i”. So, the i’th row will be your train set and the next 20% or 10%, respectively will be the test set.
* Another way is to use “train\_test\_split” from sklearn shown in [this link](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html).

**Ex-2 )** Extract the intercept and coefficients and plot line by those. (e.g. plt.plot(x, B0 + B1\*X) or any other solution that works.)

**Ex-3)** Same as #1 but you should select multiple columns as your input features.

**Ex-4)** To fit a linear regression model to an input data set, all values must be numerical. So, if you have a categorical variable, replace it’s value by a numerical value.

E.g.

Assume column X of my df is categorical. So, I’ll follow these steps to replace them with numerical values:

* To know what categories I have: print(df[‘X’].unique())
* Assume I have 2 categorical values ‘A’ and ‘B’. I’m going to replace them with 0 and 1.
  + new = {'X': {‘A': 0, 'B': 1}}
  + df.replace(new)
* Follow these steps for all categorical columns

**Ex-5)** We’d like to have an equation like this:

Y = B1\*X1 + … + Bn\*Xn

And we don’t want to have an intercept.

When defining a linear regression model we use:

>> model = LinearRegression()

A <fit\_intercept> argument is set to True by default. Now that we don’t need any intercept we have to set this argument to False.

>> model = LinearRegression(fit\_intercept=False)