This post is inspired by Andrew Ng’s machine learning teaching. I have tried to implement linear regression using gradient descent in python without using libraries. Although after implementing the algorithm I have performed a relative comparison with sklearn libraries offered by python and it turned out to be fairly good enough to begin with and to build an intuitive understanding of linear regression. In this post I will pen down a brief about linear regression and share the code.

The things which are out of the scope are:

1. How to choose the learning rate alpha and the iteration count. It is an iterative process and via plotting the cost function one can choose the correct parameters.
2. Choosing the right features for your machine learning model. In this post I have taken all the features for prediction.
3. This post considers only continuous features for building the model.
4. This post does not attempt to describe about feature engineering.

**Background Linear Regression**

I will avoid using any mathematical terms in this post as I aim to share this information with programmers like me who found machine learning an alien beast to begin with. Once you get a good intuition about linear regression thru this article & code then I would highly recommend to read the blogs published by Analytics Vidhya & Machine Learning course taught by Andrew Ng’s on coursera for further mathematical details.

**Problem Statement**

Suppose we have a dataset which consists of following variables

* Size of the house
* Number of the Bedrooms
* Price of the house.

We are tasked to predict price of the houses

**Bit of background regarding Machine Learning:**

As stated out by experts machine learning problem can be of the following nature

* Supervised Methods: The supervised machine learning methods be of continuous or categorical in nature.
* Unsupervised Methods: The unsupervised machine learning methods cater to segmentation aka clustering, dimensional reduction etc.

Going by above theory our current problem statement falls under supervised machine method peculating down to a continuous prediction problem.

Hence, we will use Linear Regression in our case to solve this problem. Please note there are other methods to address this problem statement as well.

Few of the other pointers to consider when doing model building are:

1. While building a supervised machine learning model we need training and test dataset. I have added two text files on git link. The training dataset is used for getting the right equation which can used or applied on the test or out of sample dataset for predictions.
2. Machine Learning experts have prescribed methods via which one can perform model validations. We have different model validation methods for classification and regression problems. In our case we will be use Root Mean Squared Error to perform validation.
3. Choosing the right features. It is important in cases where not all the features in your dataset are useful and by choosing a handful features only you can build the model.
4. Feature engineer: Creating relevant features based on the business problem is an utmost important activity before applying any machine learning methods.

PS: The points 3 & 4 are not discussed in this post in detail because the focus of this post is to learn to implement linear regression using gradient descent.

**Linear Regression**

The main objective of linear regression is to figure an equation which can be used to predict future values. So say we have an equation to predict the price of the houses. The equation will typically be something like this:

* Price of House = X1 + X2\* Size of the house + X3 \* Number of the Bedrooms

Now suppose we have some prior information about the values of these X’s then we can easily predict the future house prices.

Hence the main objective /goal is to figure out what will be right values of X’s so that when we plug ininformation from out of sample data to our equation we get the correct prediction. This equation is also termed as hypothesis of our model. A high level brief: X1 is called the intercept (constitutes unaccounted variables and error part )and X2 till X3(it can be till Xn) are called slopes of the equation.

**Real Deal**

As we now know we need to find the value of X’s in order to predict the house prices we should try to do the following:

1. Assuming we have performed the data analysis on our dataset then the first thing one should look at is the scale of the features under consideration. In our example we can see size of the house and number of the bedrooms belongs two different scales. Hence it is highly recommended to perform feature scaling. A simple way to perform feature scaling is: find the average and standard deviation of the variables and then subtract each value of the feature with average and divide it by standard deviation.

Mathematical Formula: *Feature Scaling = (Xith – Average) / Standard Deviation*

1. Now we need to apply some logic in order to get the values of X’s as we can see we can plug in

any values for X’s. So, to obtain the optimized values of X’s we build :

* Cost Function: A cost function basically evaluate the error cost between the predicted values and actual values.

Say for example, we have determined some values for X’s and then we plug in these values to predict house price. The next step is to compare this predicted values with actual house price values we have in the training dataset. It is done through summation of squared difference between predicted and actual values. This summation is then divided by two times the number of observations. The lower the value of our cost functio, better our model is.

* Use Gradient Descent: Gradient Descent is used to determine the optimum values for yours X’s. In this we keep the initial values of X’s at zeroes along with iteration count and learning rate which is also called alpha. In each iteration with the usage of the alpha and cost function, gradient descent tries to converge to a local optimum value i.e. right values for X’s.

So, assume we have chosen iteration = 10000, alpha = .001 and initial values of X’s as zeroes. For each iteration we calculate the prediction by applying the X’s values to our equation and thereafter computing the error by subtracting it from the actual house price. This result is looped thru inside a nested loop for each value of X . At this juncture each value of X is multiplied by the error and then summation is applied to it. This aggregated value is then multiplied by alpha (learning rate) and divided by the number of observations which in turn is subtracted by the current value of X. This process is repeated for all the X’s. The obtained values of X’s is used to compute the cost function for a given iteration which is stored in an array form. One can plot the cost function to determine the error rates behavior for each iteration. Ideally it should reduce after each iteration.

1. In the next step we plot the cost function to see if the errors are decreasing with each iteration. This can give a good intuition about our choice of alpha and iteration count.
2. We perform model validation methods like in our case Root Mean Squared Error which is subtract the actual value from the predicted value and take a square and apply summation. Take a square root of the summation value while dividing it with the total observations. Please note the RSME of this model is not optimum
3. Once we have the parameters of our model then we can use it to predict the house prices on our testing dataset.

Here is the reference to my github: <https://github.com/sorghosh/Linear_Regression_UsingGradientDescent>