SOC ASSIGNMENT-2

GROUP:

Sai charan Teja	: IIT2016039
Anil Gupta Somisetty	: IIT2016131
Nitin Raj Singh	:IIT2016132
Utpal	:IIT2016133
Kiran	:ITM2016009

GIVEN:

Housing price dataset.

PREDICTION:

We need to predict HOUSE PRICE.

CASE-I:
DIVISION OF DATASET:

To conclude that our output is not overfitted and is the best fit we divide the data into parts for training and testing.

Training-400 samples

Testing-146 samples

ANALYSIS:

1.As the given data consists of features with classification problem i.e data with “yes” or “no” we remove them as we cannot use them in linear regression and we continue with the remaining data.

2.As we have irregular data meaning one feature dominates the other we NORMALISE the data with data=(data-data.mean())/data.std() .

3.We will be having 6 features including bias and weights would be 6.W=[W0,W1,W2,W3,W4,W5]

Done on Testing data:
PREDICTION ERROR=sum((predicted price-actual price)/actual price)/number of samples In testing data

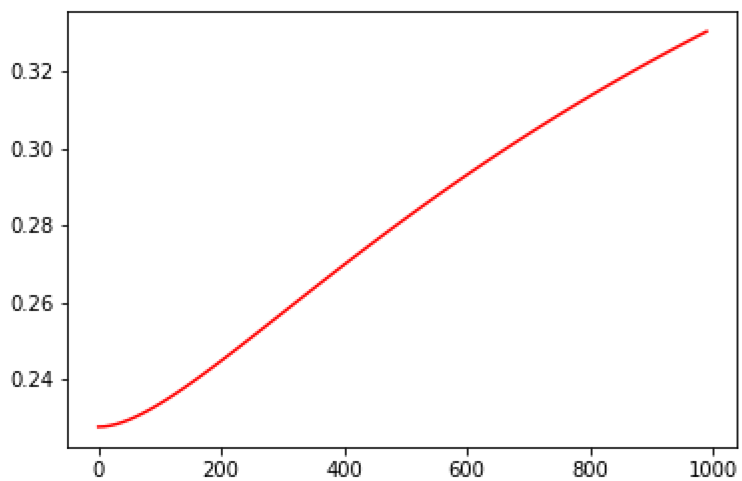
CONCLUSION:

NORMAL EQUATION:

Calculated the weights using the equation W=((X.T*X)^-1)*(X.T*Y)
Y=ORIGINAL HOUSING PRICE
X=FEATURE MATRIX
W=WEIGHTS

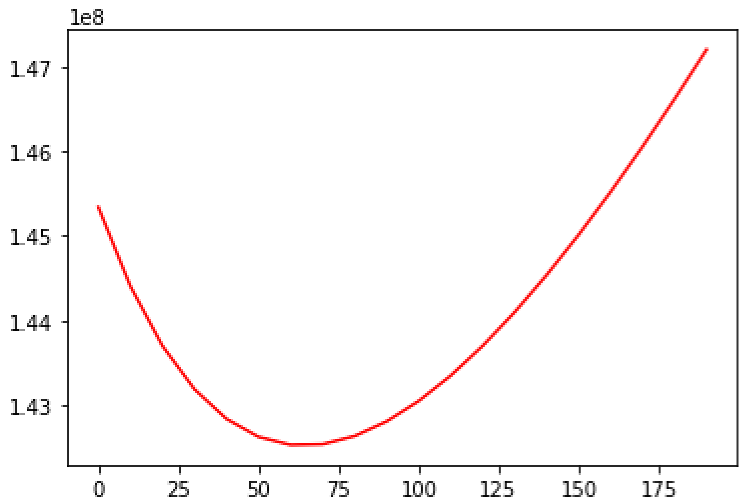
We take for different values of lambda and we calculate the cost in training and predicting and we take the lambda with less prediction error.

Lambda vs cost:



As you could observe the final cost of the normal equation goes on increasing on increasing lambda.

Lambda vs prediction cost:



At lambda=55 we get the lesser prediction error So we choose lambda=55

At lambda=55

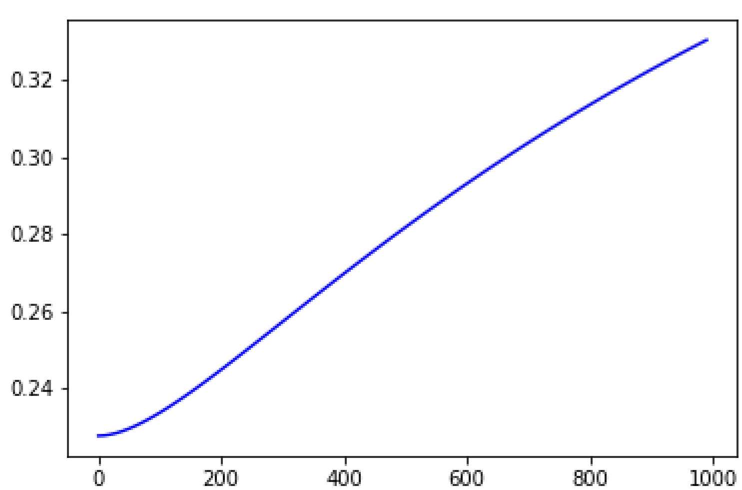
Cost training=0.2435303532374692

GRADIENT DESCENT:

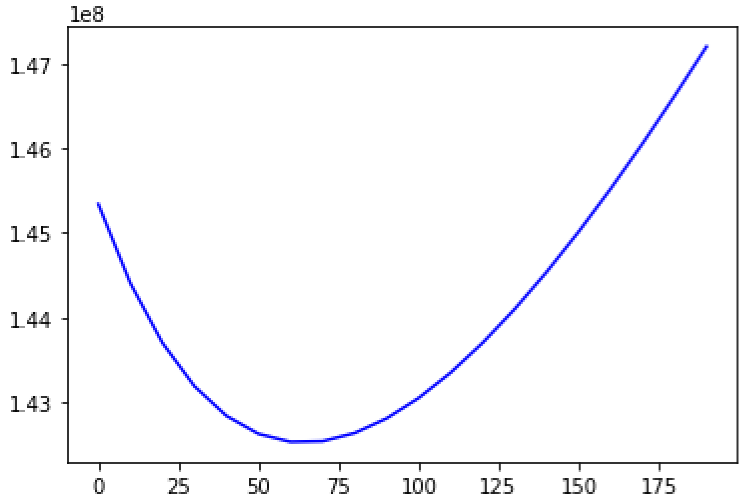
Initial W would be [0,0,0,0,0,0] and we would be changing W after every epoch and decreasing the cost.

RESULTS:

Lambda vs cost:



Lambda vs Prediction error



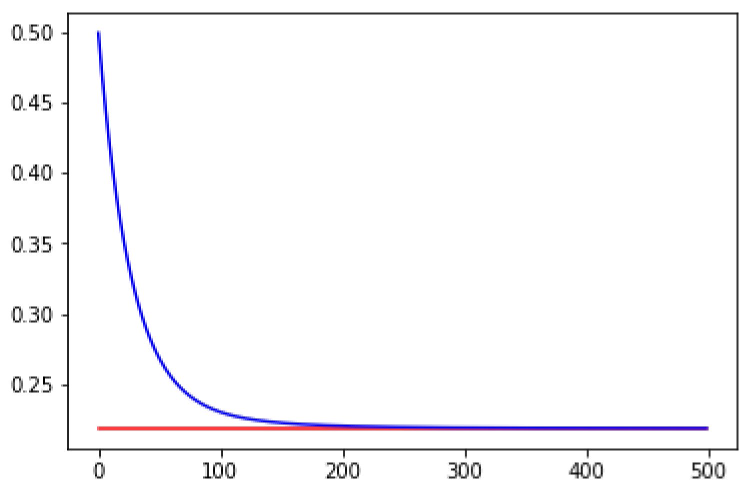
For lambda=55 the cost of prediction is less So we choose our lambda = 55.

For different values of learning rate the output graph convergence would be different.

Blue curve=Gradient Descent,Red curve=Normal Equation

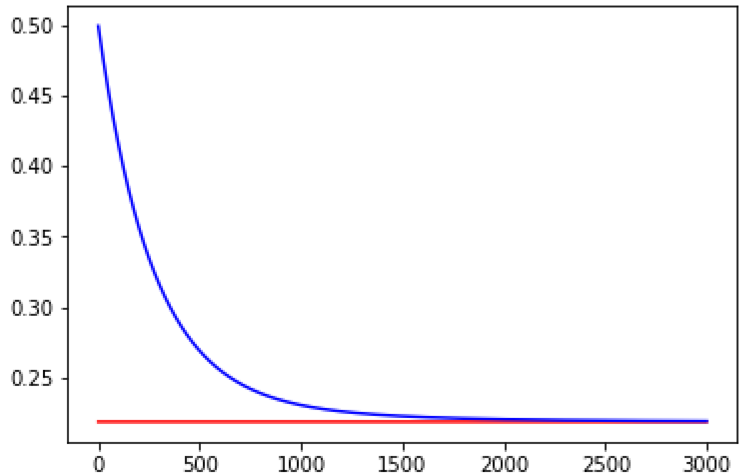
Learning rate=0.01

Cost vs epoch graph:



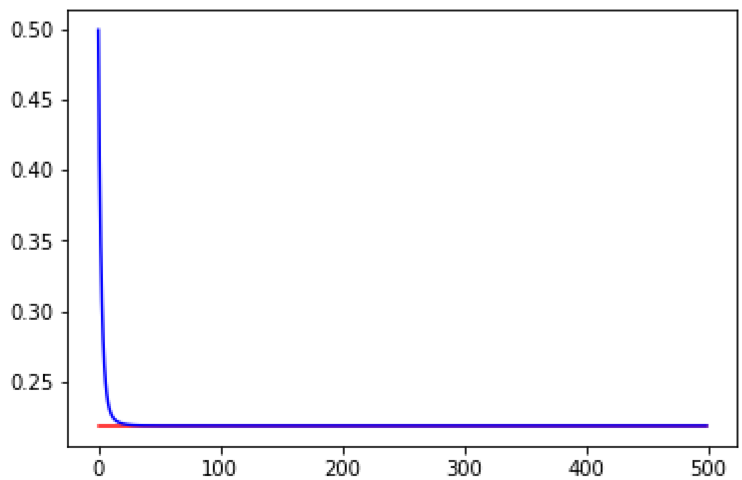
Learning rate=0.001

Cost vs epoch graph:



Learning rate=0.1

Cost vs epoch graph:



COMPARISION:

NORMAL EQUATION:

Final cost is 162279265.55137438

Gradient Descent:
Learning rate=0.1 took 30 epochs to converge the cost.

Learning rate=0.01 took 300 epochs to converge the cost.

Learning rate=0.001 took 3000 epochs to converge the cost.

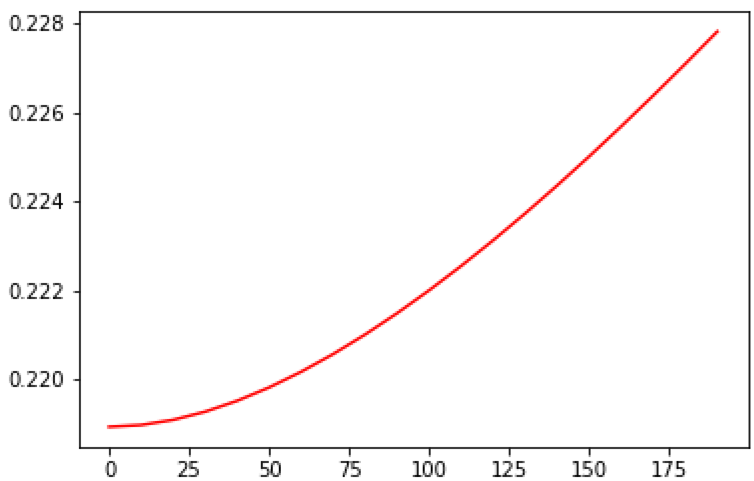
We choose the convergence learning rate 0.1 for our algorithm with more number of epochs after observing the prediction error and final cost of convergence.
If we observe the of 3 learning rates 0.1 fastly converges 0.01 moderately converges and 0.001 converges very lately.

CASE-II:

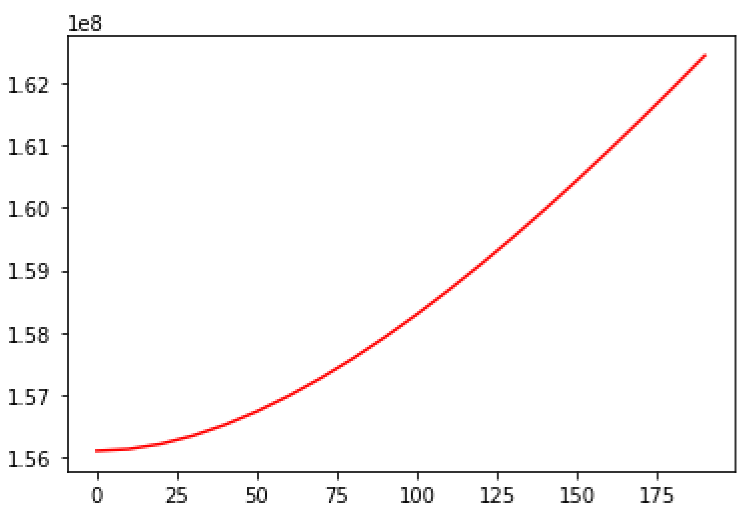
Without Dividing the dataset i.e Training and testing on same data.

NORMAL EQUATION:

Lambda vs cost:



Normal Equation lambda vs prediction error:



So we choose lambda = 0 for better results if we don't divide the data.

Its obvious because we use regularisation for removing overfitting of the data.Hence we are

predicting and training on same data without using regulariser the accuracy would be high.

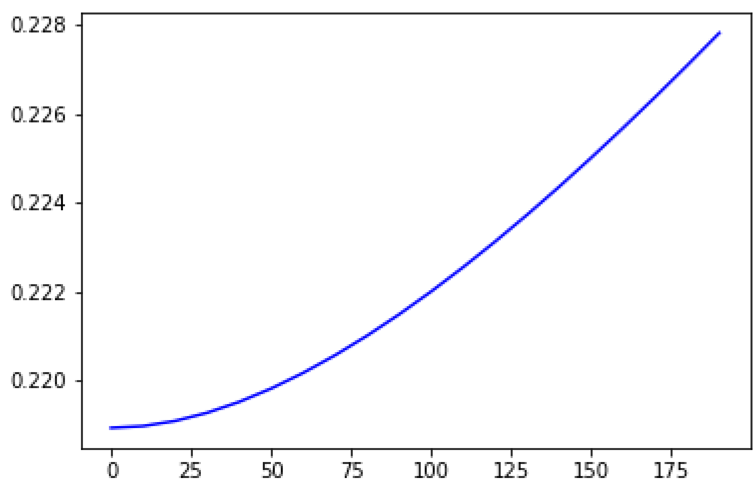
Final cost of Normal Equation = 0.22781060228136119

GRADIENT DESCENT:

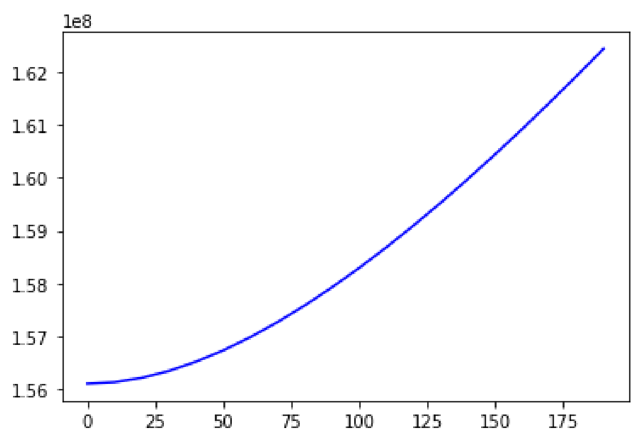
Initial W would be [0,0,0,0,0,0] and we would be changing W after every epoch and decreasing the cost.

We define learning rate=0.1 epoch = 500

Gradient Descent lambda vs cost:



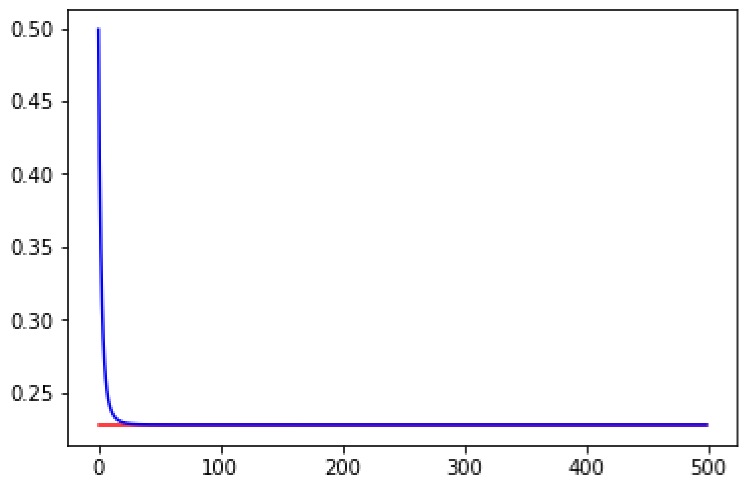
Gradient Descent lambda vs prediction error:



Its obvious because we use regularisation for removing overfitting of the data.Hence we are predicting and training on same data without using regulariser the accuracy would be high.

So we choose lambda = 0 for better results if we don’t divide the data.

Epoch vs Cost:



Final cost of Gradient Descent = 0.22781060228136127