CSE 574: PROJECT ONE REPORT

NAME: ANIKET DESHPANDE

UB PERSON NUMBER: 50133684

Introduction:

The project was based on application of Linear Regression. We have used two methods to showcase Linear Regression as follows:

1. Maximum Likelihood with Closed Form Solution.
2. Maximum Likelihood with Gradient Descent.

We were given a subset of Microsoft LETOR database to work on. The data consisted of around 69623 entries (which represented URLs) with each entry consisting of 46 features that were used to calculate the relevancy of that URL. The final task was to train our model based on this data and then predict the output of the test data.

The basic idea of linear regression is to choose optimal weights and basis functions so that the output of test data can be predicted with minimal error

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$

Approach:

1. The complete dataset was first divided as follows:
    a. Training Set: Consisted of 80% of data
    b. Validation Set: Consisted of 10% of data
    c. Testing Set: Consisted of remaining 10% of data

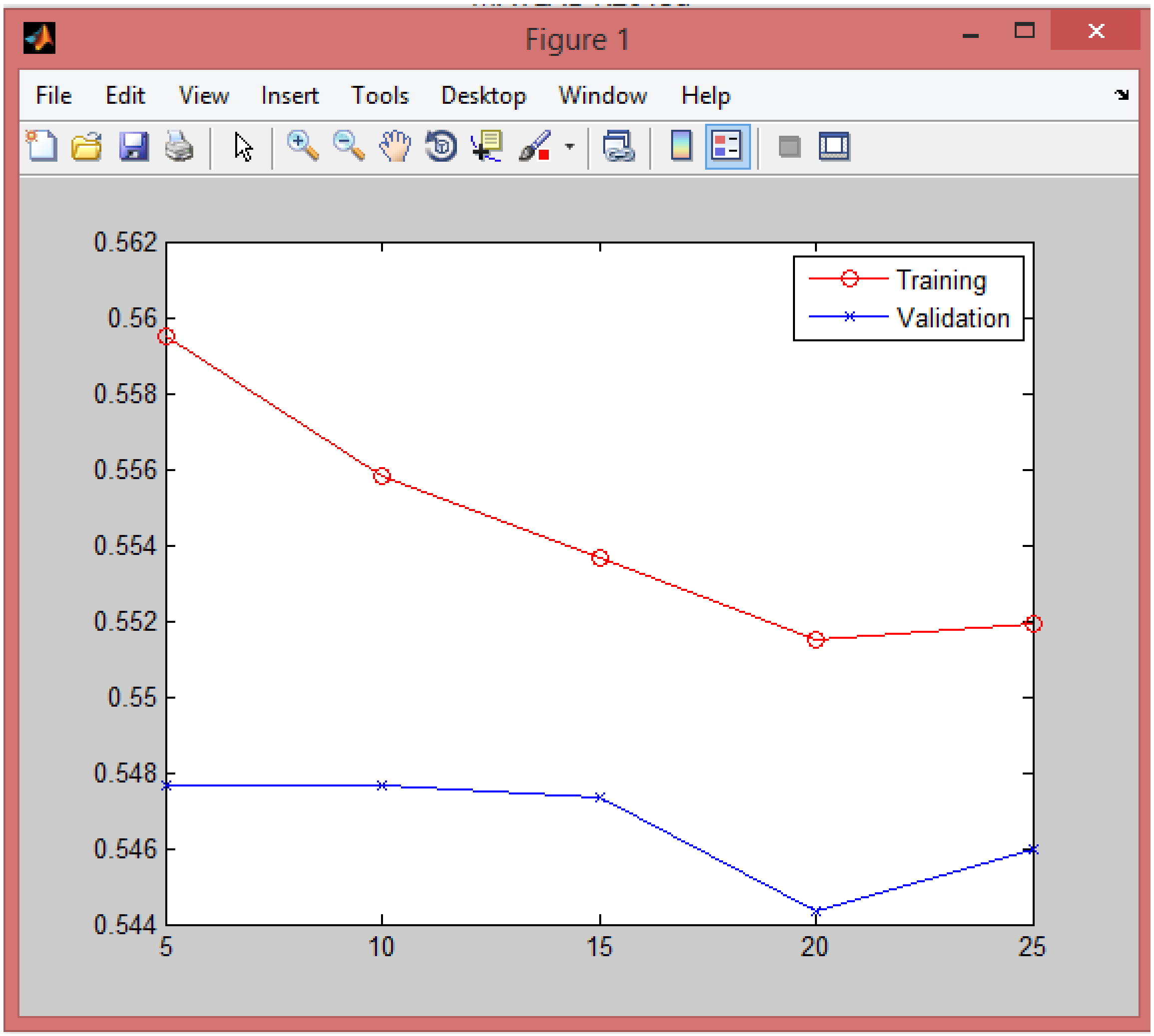
    The given “Querylevelnorm.txt” file was parsed using Matlab. A separate relevance label vector and input feature vector was created for each of the three sets namely Training, Validation and Testing.

2. Finding optimal values of following hyperparameters:
    a. Mu of the dataset which will be used in calculating basis function
    b. Sigma of the dataset which will be used in calculating basis function
    c. Order complexity M of the model
    d. Regularisation parameter Lambda (used only in closed form solution)
    e. Choice of basis function (Gaussian or sigmoidal)
    f. Learning rate eta (used only in Gradient Descent)

3. The optimal values was found out by running the model on training data set to calculate that weight vector and design matrix for which the Root Mean Square error for validation set is minimum.
4. After finding the optimal values, the model was run on training data to predict its output.

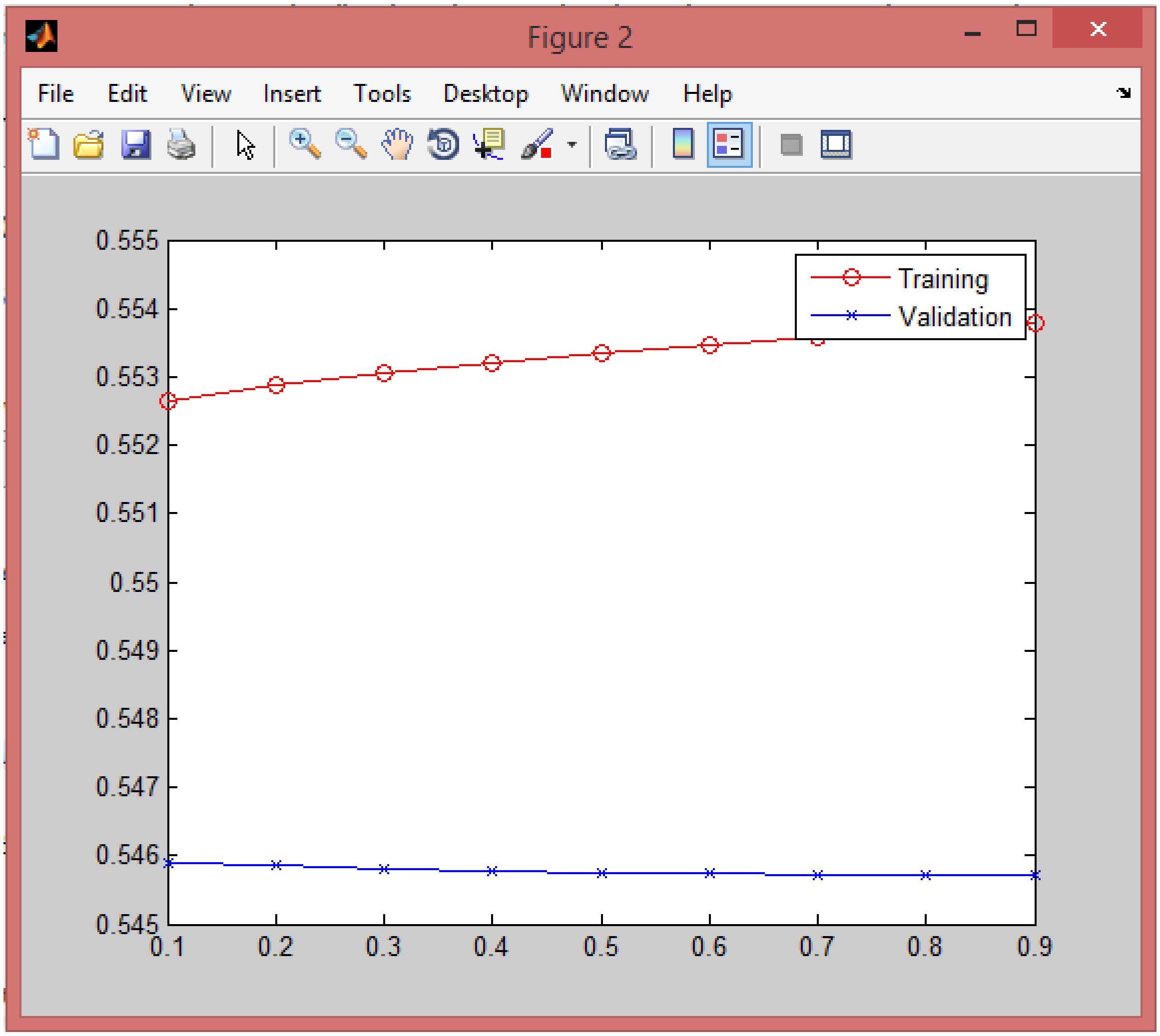
Closed Form Solution:

a. Finding mu:
   Mu is chosen as follows:
   Find the order of the model M. Then, total number of means will be (M-1) * 46. To get (M-1) * 46 means, we divide the dataset into (M-1) parts and calculate mean of the each column which can then be used inside the Basis function.
b. Finding variance (s):
   The value of s was taken to be a random value between 0 and 1. The value was then adjusted so as to make the RMS error less.
c. Finding order complexity M:
   First the value of lambda was chosen as 0 and then for each value of M from 5 to 25, weights were calculated and subsequently RMS error for both validation and testing set was calculated and plotted against M. The graph generated was as follows:



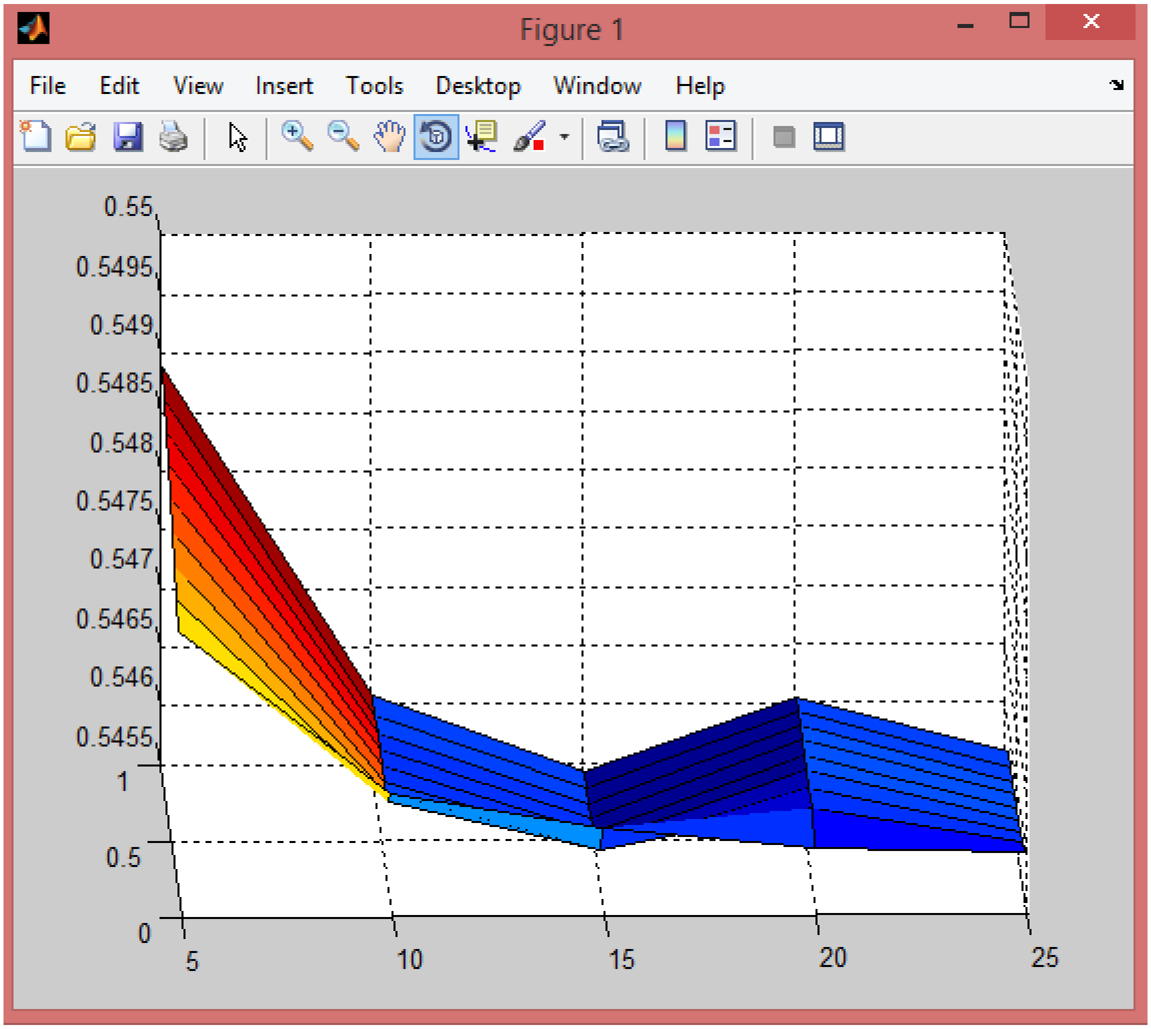
As we can see here the minimum RMS is at M=20. So we keep M fixed in our subsequent calculations for other hyperparameters. (Though this M may or may not be constant).

d. Finding regularization parameter lambda:
   After fixing M, we run our model with values of lambda as follows
   lambda = [0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9];
   The graph was as follows:



The value of lambda was chosen 0.7

Now, for combined values of M from 5 to 25 and lambda from 0.1 to 0.9 the model was run.



Thus M was chosen as 20 and lambda was taken as 0.5

e. After we fix the value of ‘M’, ‘lambda’ and calculate the ‘w’ vector, we find the PHI matrix on the TEST Set. Using that PHI and W matrix we calculate ‘y’ as y = PHI * W. And the difference between the predicted and actual value will be (PHI * W – t). We calculate the squared error function E(w) and use it in ERMS calculation given by ERMS = square root (2 * E(w) / N).
With the values of ‘M’ and ‘Lambda’ chosen as above (M = 20 and lambda = 0.5) , the ERMS we get on test project in my case has been 0.6290

Gradient Descent:

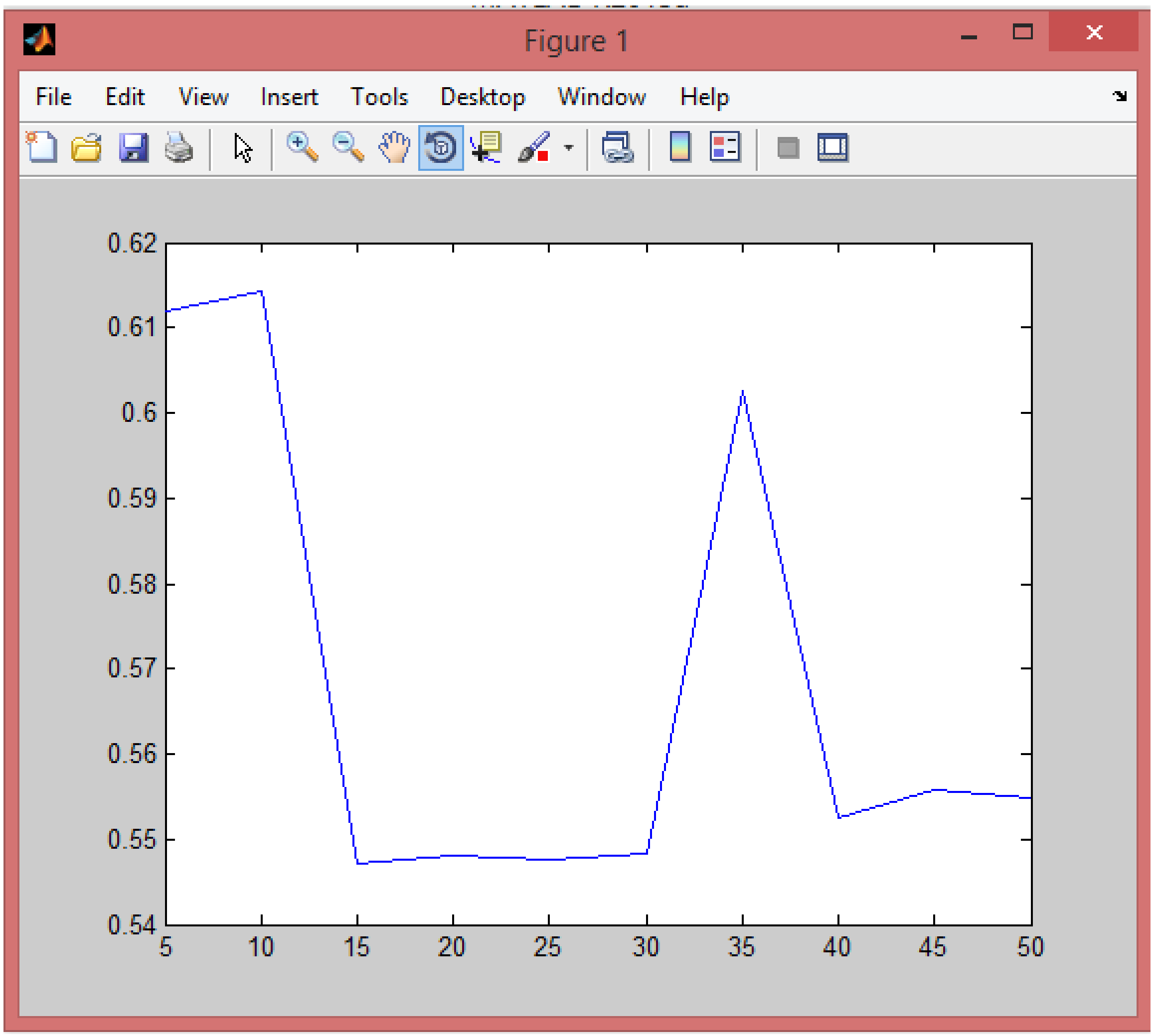
In this method the mu, design matrix phi was taken from closed form solution and then M was chosen as follows:

1. The formula for gradient descent is

   General Form:

   $$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n \qquad \mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta(t_n - \mathbf{w}^{(\tau)\mathrm{T}} \boldsymbol{\phi}_n)\boldsymbol{\phi}_n$$

   Here the heuristic applied for learning rate eta was that initially eta was chosen as 0.1 then for each iteration if the error increased then eta was further multiplied by 0.1.
2. Stopping condition was chosen as if two consecutive errors were less than 0.00001.
3. The graph obtained was as follows:



Thus the order complexity M was chosen as 15.

With this value of M gradient descent on test set gave an erms of 0.6328