Parisha Joshi
parishamaheshj18@vt.edu

Assignment 1
Machine Learning in Communication

2019/10/18

1 Directions to use code files.

- Please Change the path names for accessing various datasets.
- There are .m files attached to the project as well. But the files were coded originally in .mlx format. Some functions might not work as intended due to structures of the files. For example, The plots will be shown sequentially in live script, But in .m file the figures will overlap.
- To run the live script: 1) Open File(number).mlx file 2) Change the path name of dataset 3) Click Run
- The report was typed in Latex and implemented in MATLAB-R2019a.

2 Question I

Regression. The objective of this question is to help you visualize the working of linear regression. Let us first consider a simple linear regression problem with one input and one output. You are given three different training data sets: P1-data1.txt, P1-data1a.txt and P1-data2.txt. In each of these files, the first column represents the input variable while the second column represents the output variable. Note that you can import the training data sets from the txt files into MATLAB using the load command. For this setup, answer the following questions.

2.1 Task a : File1.mlx

Visualize the training data set in file P1-data1.txt by plotting the output variable against the input variable.
Solution:

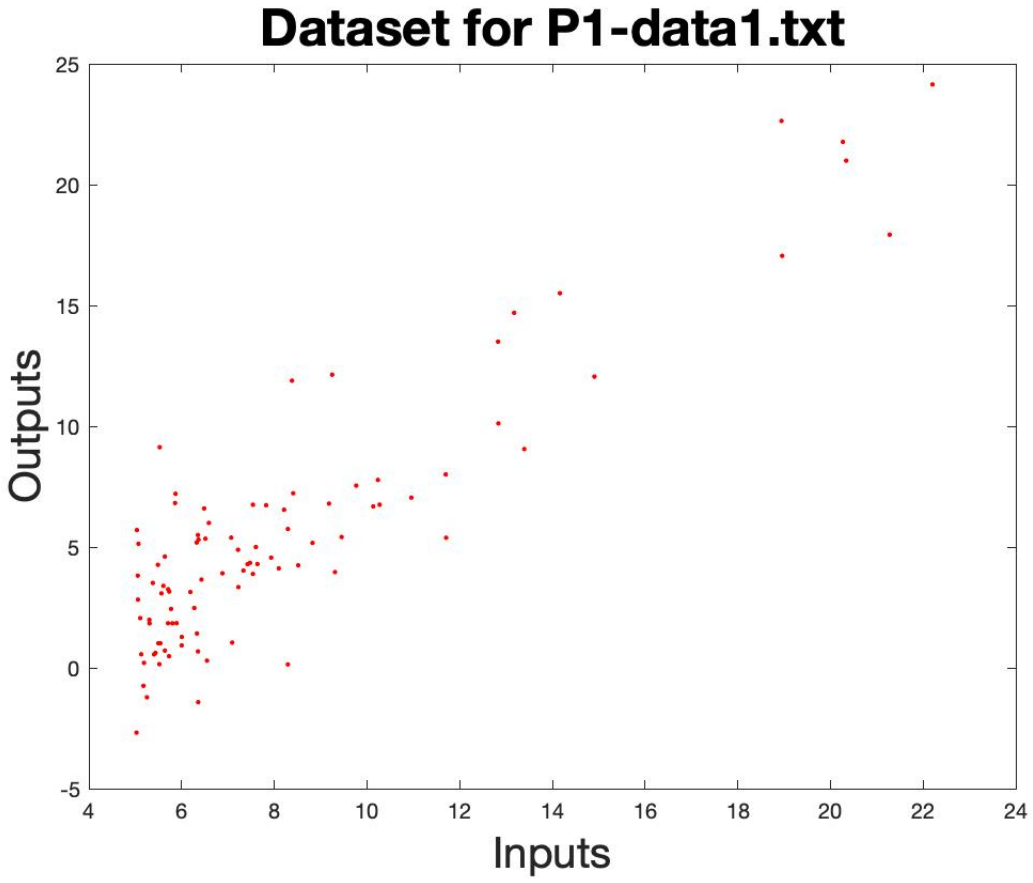

Figure 1: Dataset 1

2.2 Task b : File1.mlx

Consider the linear regression model $f(x) = \beta_0 + \beta 1 * x$ where x is the input variable. Here, f(x) is also called the hypothesis function. Assuming squared loss, plot the loss function for the following range of the learning parameters: $10 \leq \beta 0 \leq 10 and 2 \leq \beta 1 \leq 4$. From the plot that you get, can you infer if the loss function has a global minimum? Justify your answer for full credit.

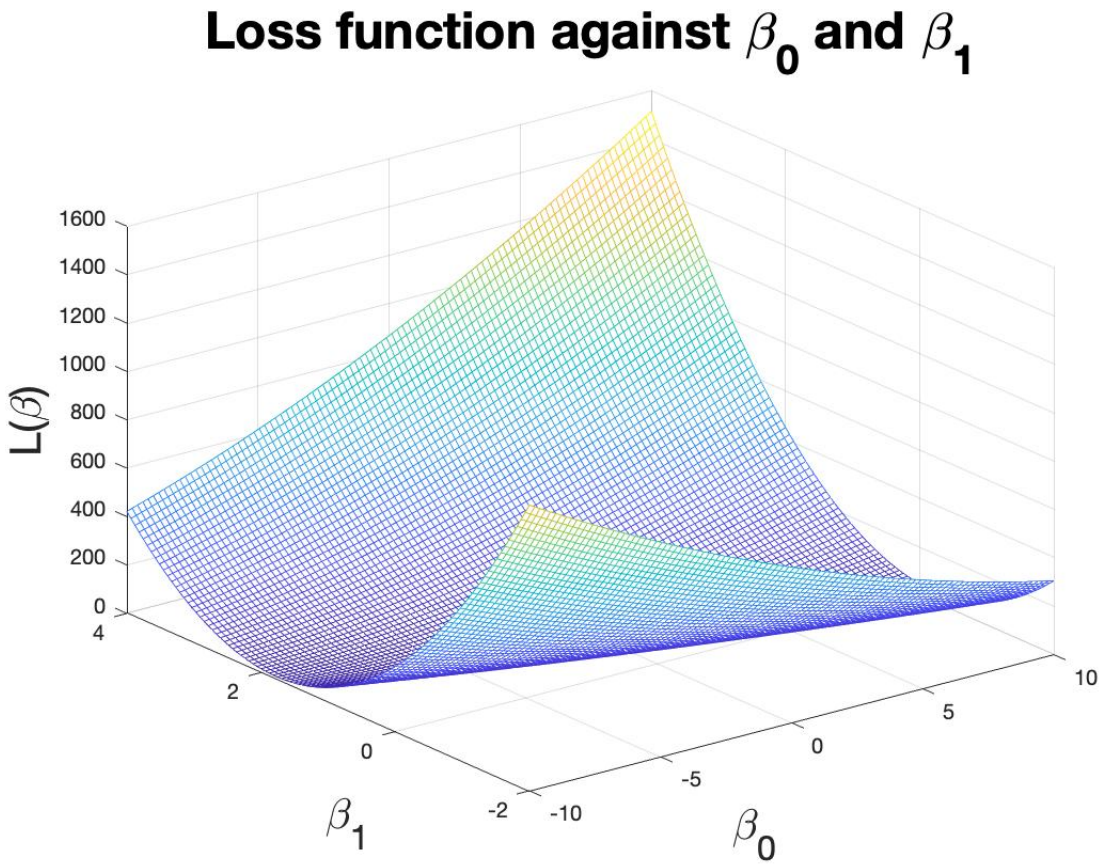

Figure 2: Change in Loss function over β

Solution: We cannot infer that the loss function has the global optimum in the given plot. The range of Beta is only limited to the [-10 10] and [-2 4]. It is the local optimum for this range. But we cannot say for sure if it is a global optimum.

2.3 Task c : File1.mlx

Fit the learning parameters β_0 and β_1 to the training data set in P1-data1.txt using the gradient descent algorithm. For the gradient decent algorithm, consider the learning rate to be 10^2 and the number of iterations to be 1500. Initialize both β_0 and β_1 to 0. Using the values of β_0 and β_1 obtained from the above procedure, plot the hypothesis function along with the training data set.
Solution: Figure 3

2.4 Task d : File1.mlx

Repeat (c) for the learning rate values of 10^{-6} and $3 * 10^{-2}$. Keep the same number of iterations.
Solution: Figure 4 for 10^{-6} and Figure 5 for $3 * 10^{-2}$

2.5 Task e : File1.mlx

From the plots that you obtained in (c) and (d), which hypothesis function would you use to fit the training data set? State clearly why the other hypothesis functions were not chosen.

Solution: To see how different learning rate converges to the optimum solution lets look at the graphs

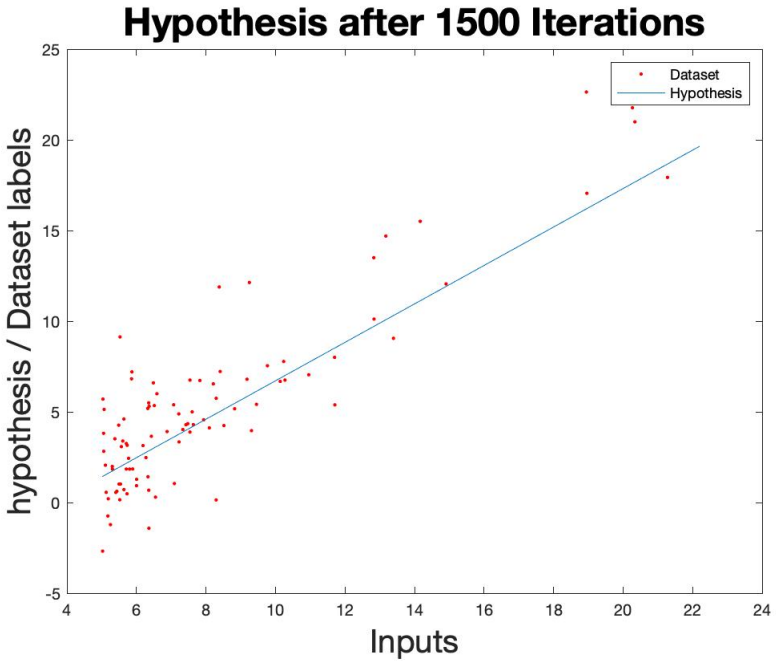

Figure 3: Hypothesis with learning rate 10^{-2}

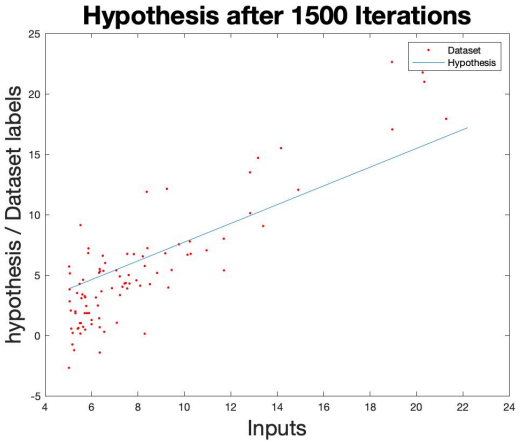

Figure 4: Hypothesis with learning rate 10^{-6}

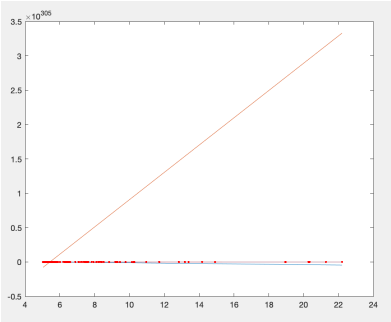

Figure 5: Hypothesis with learning rate $3 * 10^{-2}$

below. As shown in the last three Figures(6),(7),(8), The best learning rate would be 10^{-6}.From Figure(7), We can see the step size getting decreased so that there is a smooth convergence to the best hypothesis function.
1) learning rate = 10^{-2} has a tendency to oscillate while converging to the local minimum. The decrease in the step size is not small enough to reduce the convergence speed gently.

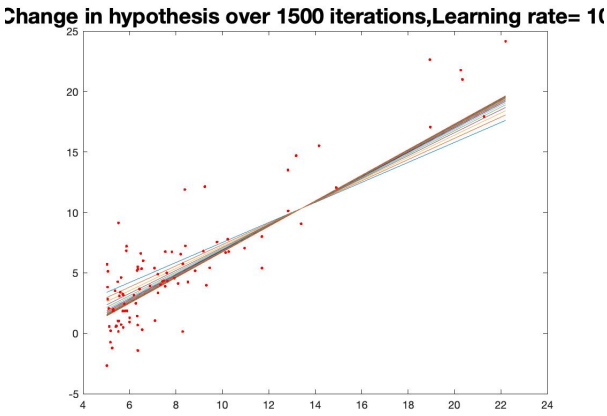

Figure 6: Convergence with learning rate 10^{-2}

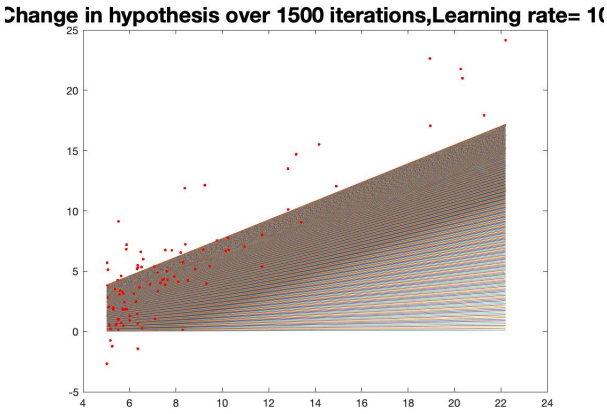

Figure 7: Convergence with learning rate 10^{-6}

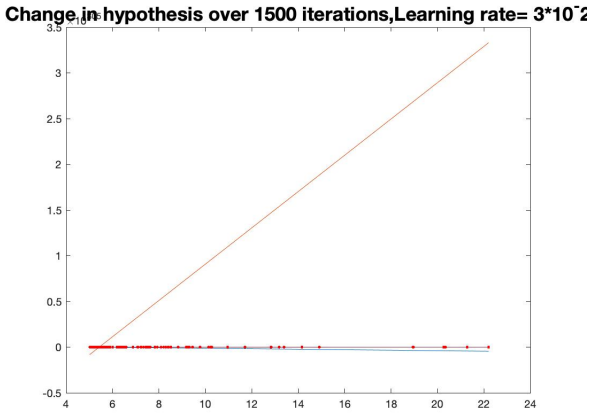

Figure 8: Convergence with learning rate 3*10^{-2}

2) $3 * 10^{-2}$ **gives overshoot results. As one can see from the Figure 8, the hypothesis function is at** 10^{305} **range on y axis.Hence, It is definitely not the best choice.**

2.6 Task f : File1.mlx

Compare the values of the learning parameters obtained by gradient descent algorithm in (c) with the ones obtained directly from the normal equation discussed in the class.
Solution:
The learning parameters after (c) are: $\beta 0 = -3.88697930021484$ **and** $\beta_1 = 1.05994353580051$ **The answer**

we get by applying equation discussed in class for both the values are: $\beta 0 = -4.4601$ **and** $\beta 0 = 1.2165$
The equation applied in MATLAB is

$$\beta = inverse(X * X^T) * X * Y$$

Here $\beta 0$ **achieved from equation is 0.5732 smaller than** $\beta 0$ **achieved from the iterative method is c. Similarly** $\beta 1$ **achieved from the equation is 0.1566 bigger than** $\beta 1$ **achieved from the iterative method.**

2.7 Task g : File2.mlx

Now assume that the training dataset has changed to P1-data1a.txt. Would you still choose a squared loss function? If not, which loss function is more appropriate for this? For this new loss function, train the linear regression model using the gradient descent method. Repeat the same for the squared loss function (using data set in P1-data1a.txt). Plot the two hypothesis functions in the same figure and explain your observations. Clearly state any assumptions that you make in this part.

Solution:
Squared error function is sensitive to outliers. Here the number of outliers are more. So, as a loss function using mean absolute error or Huber loss function would be efficient. But, Mean absolute error function cannot be used with gradient descent method. Reason being, we cannot take gradient at 0. for simplicity using Huber Loss function is more efficient.

Huber loss function acts as a least square function when closer to 0, and acts as a Mean Absolute square loss function when away from 0.
For example, Let delta be 0.02. For range of (-0.02, 0.02) the loss function will behave like square loss function. and for out of the range, The loss function acts like mean absolute error.

Assumptions:

-Delta is taken as close to 0 as possible hence, 0.02
-Number of iterations are taken as 1500.
-The learning rate is taken as 10^-6

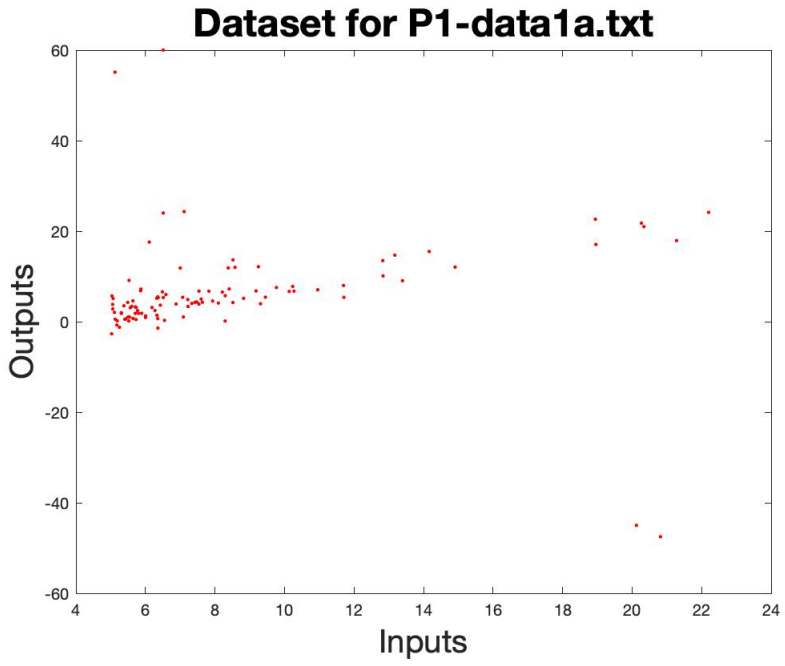

Figure 9: Dataset P1-data1a.txt

Observation:

Using only least square method with outliers causes the gradient descent method to overshoot. The

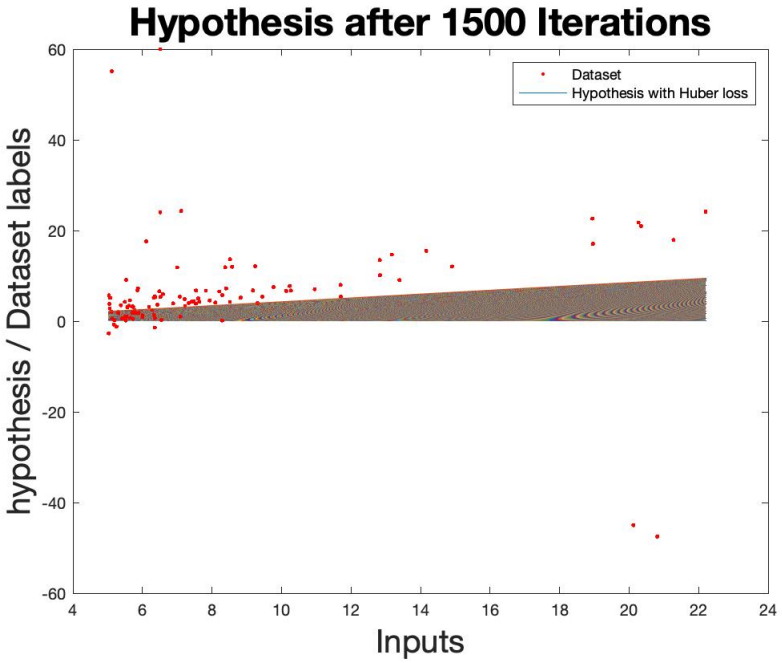

Figure 10: Huber Loss Hypothesis

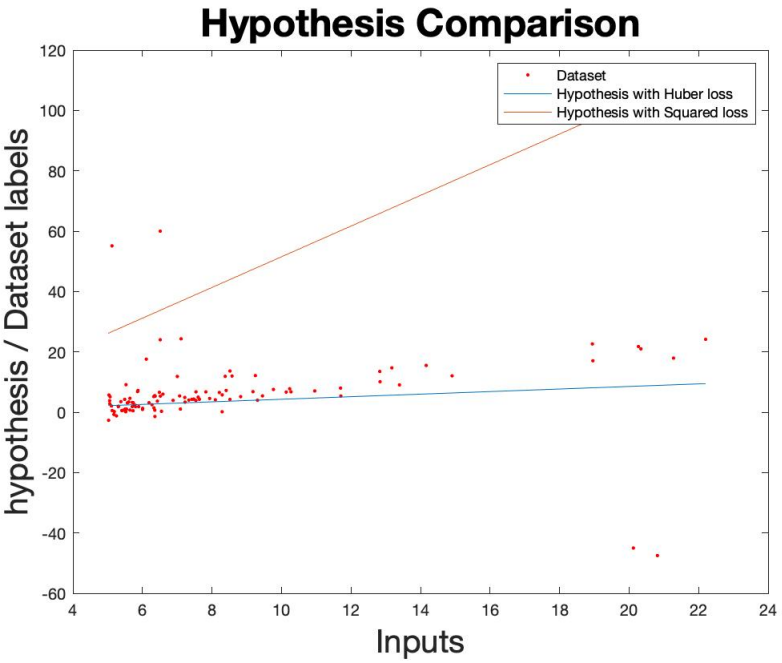

Figure 11: Hypothesis comparison

hypothesis is affected because of the outliers more in least square method than in Huber loss method. If the observation of outliers is necessary for the application then using least square method is efficient.Contrary to that, If we want to ignore the outliers and build a model robust to outliers, Using Huber method is efficient.

2.8 Task h : File3.mlx

Repeat (a) for the training data set given in P1-data2.txt.

Solution:

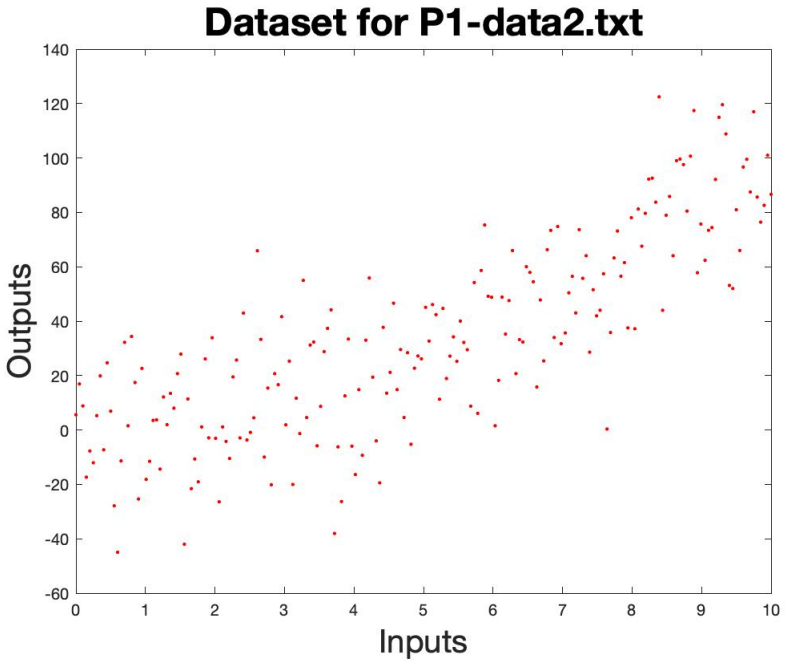

Figure 12: Dataset 2 : P1-data2.txt

2.9 Task i : File3.mlx

Fit the learning parameters (0, 1 and 2) to the training data set contained in the file P1- data2.txt using the following hypothesis function: $h(x) = \beta 0 + \beta 1 * x + \beta_2 * x^2$***, where x is the input variable. For the gradient decent algorithm, initialize all the learning parameters to 0, and use proper values for both the learning rate and the total number of iterations based on your observations from parts (c)-(e). Plot the hypothesis function along with the training data set. Clearly state your choices for the learning rate and the number of iterations.***

Solution:

The learning rate, from the previous example, is chosen to be 10^{-6}**, The hypothesis function seems to be converging in very few equations. So instead of taking 1500 iterations I have only taken 20 iterations. The Figure 13 below shows the final hypothesis plotted along with the data and the next one (Figure 14) shows the changes in hypothesis over 20 iterations.**

Observation: Even if the Hypothesis is non-linear in the feature space, It is linear in the β **space. Which means it is linear with respect to Weights of the feature.**

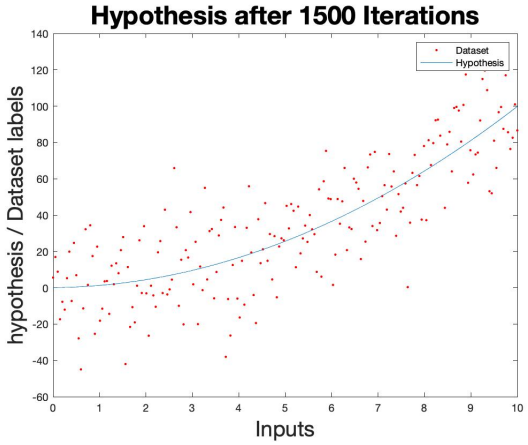

Figure 13: Hypothesis

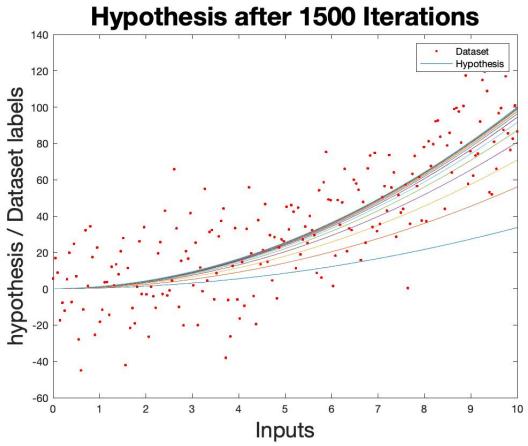

Figure 14: Hypothesis

2.10 Task j : File4.mlx, File5.mlx

Apply the k-NN algorithm to the datasets in P1-data1.txt and P1-data2.txt for different values of k (1 k 5). For each training data set, compare the plots obtained using k-NN algorithm with the hypothesis functions obtained using linear regression (i.e., the plot in (c) for P1-data1.txt, and the plot in (i) for P1-data2.txt).

Solution: The kNN plots are given below for k=1,2,3,4,5.

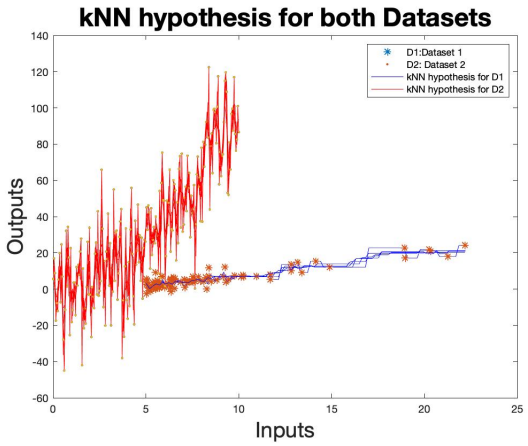

Figure 15: kNN hypothesis for both datasets,k=1,2,3,4,5

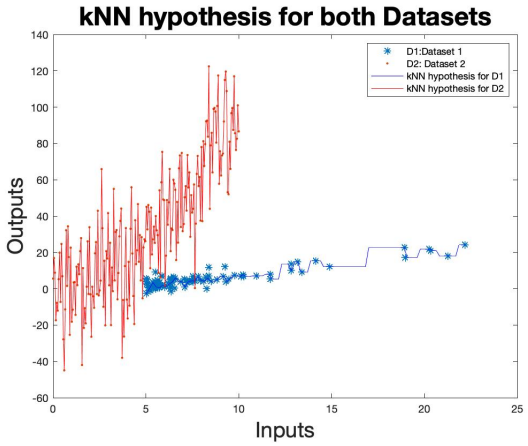

Figure 16: kNN hypothesis for both datasets,k=1

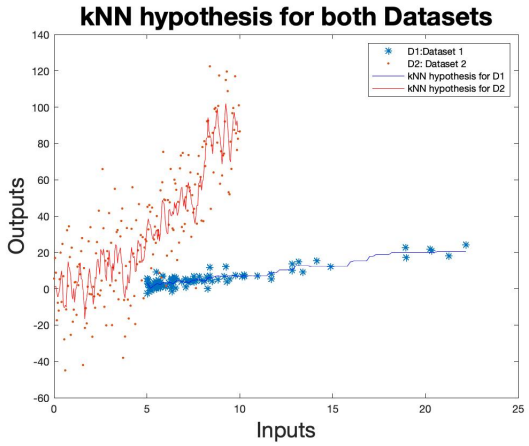

Figure 17: kNN hypothesis for both datasets,k=5

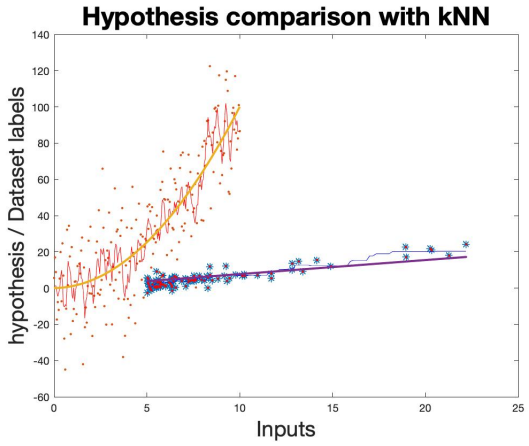

Figure 18: kNN and hypothesis from question (c) and (i).

3 Question II

Logistic Regression. In this problem, you will learn how to implement logistic regression. You are given: i) two different training data sets: P2-data1.txt and P2-data2.txt, and ii) two different validation data sets: P2-valdata1.txt and P2-valdata2.txt. As discussed in the class, the training datasets should be used to train your selected hypothesis classes while the validation datasets should be used to select the most appropriate hypothesis class. For each data set, there are two input variables, x1 and x2, and one output variable, y. In each file, the two inputs are represented by the first two columns, and the output variable is represented by the third column. Assume that the output variable is binary, i.e., y belongs to [0, 1]. For this setup, answer the following questions. In parts (a) and (c) below, please limit your choices to the polynomial hypothesis classes of degree 6 or less.

3.1 Task a : File6.mlx

Visualize the training data set in file P2-data1.txt by plotting the labeled data points as a function of the two features. Based on this, select the candidate hypothesis classes (equivalently, models). Train these models using P2-data1.txt. While you used the gradient decent algorithm to obtain the optimal linear regression parameters in Problem 1, you are going to use the fminunc built-in function in MATLAB to train your models in this problem. The function fminunc is an optimization solver that finds the minimum of an unconstrained optimization problem (which is the case for regression problems). For the function fminunc, please use the maximum number of iterations as 400, and initialize all the learning parameters to 0. Plot the decision boundary along with the training data set.

Solution:
After trying different models with degrees less than 6 for both the features X1 and X2, Model 1 with $B0 * x1 + B1 * X2$ **Fits the dataset perfectly. The training accuracy is 91 percent and the testing accuracy**

is 95 percent. It is pretty intuitive by looking at the dataset that the points are separable by a line. Please find the Matlab file for different models attached in the zip file.

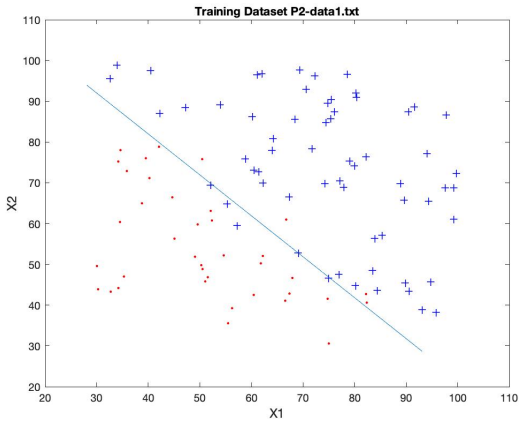

Figure 19: Dataset P2-data1.txt

Model 2: $B1 + B2 * x1 + B3 * x2 + B4 * x2^2$ **,Training accuracy : 62 percent , Testing accuracy : 85 percent**
Model 3 : $B1 + B2 * x1 + B3 * x2 + B4 * x2^2 + B5 * x1^3$**,Training accuracy : 62 percent , Testing accuracy : 85 percent**
Model 4 : $B1+B2*x1+B3*x2+B4*x2^2+B5*x1^6$ **,Training accuracy : 62 percent , Testing accuracy : 85**

Observation: The percentage accuracy values become stagnant after the first degree of polynomials for this dataset.It is obvious to happen because a simple visualization of data would tell us that the labels are separable by a line.

3.2 Task b : File6.mlx

Now use the validation dataset P2-valdata1.txt to select the most appropriate model from amongst the ones that you trained in part (a). You need to make sure that the validation error is at most 10 for your selected model. In other words, the percentage of prediction errors should not exceed 10 when your chosen model is applied to the validation dataset P2-valdata1.txt.

Solution:
Using the selected mode $\beta_0 + \beta_1 x1 + \beta_2 x2$ **is the best fit for the dataset. Validation error is 5 percent and the accuracy is 95 percent.**

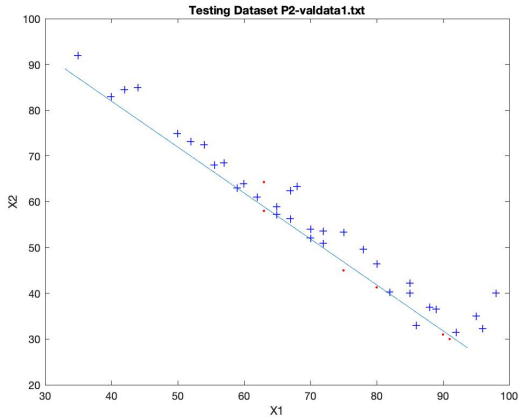

Figure 20: Dataset P2-valdata1.txt

3.3 Task c : File7.mlx

Repeat part (a) for the training data set in P2-data2.txt with the only difference being that you should use the regularized version of logistic regression in this part with a regularization parameter of 1. Similar to (b), consider maximum number of iterations to be 400 in the fminunc function and the initial values for all parameters to be zeros.
Solution: The following models were trained : Figures (21,22,23,24,25,26,27,28)

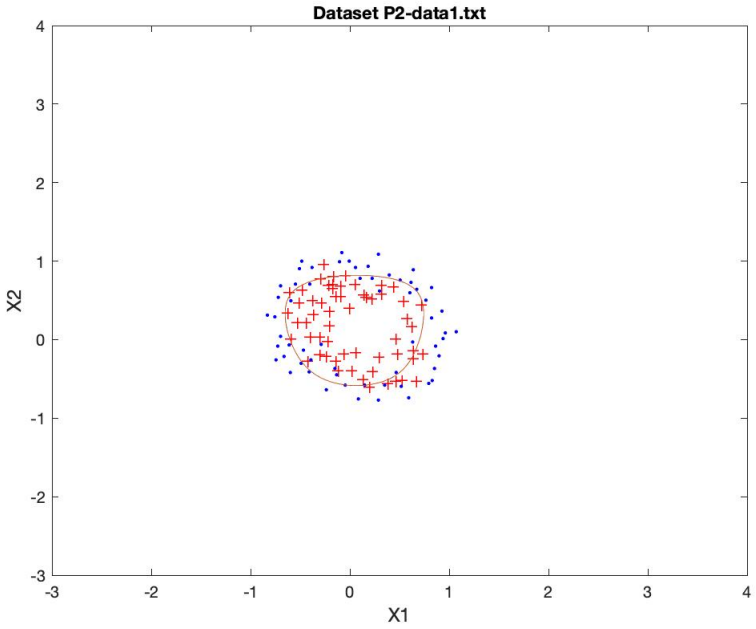

Figure 21: Best Model

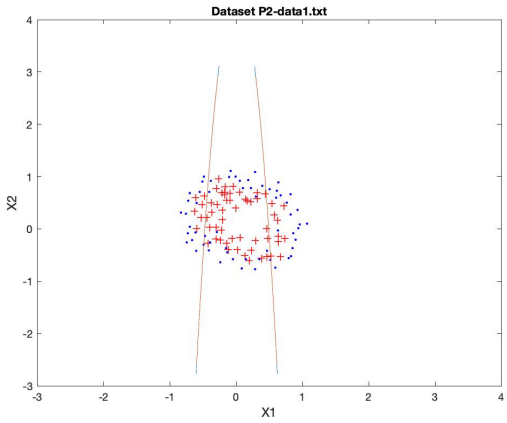

Figure 22: Model 2

The changes in the highest polynomial of the feature x1 and x2 affects the model errors as shown in Figure 29 and 30.

3.4 Task d : File7.mlx

Similar to (b), use the validation dataset P2-valdata2.txt to select the most appropriate model from amongst the ones that you trained in part (c). Make sure that the validation error is at most 15 percent for your selected model. **Solution:**
The Model ($\beta_1 + \beta_2 * x1 + \beta_3 * x2 + \beta_4 * x1.^2 + \beta_5 * x2^2 + \beta_6 * x2^3 + \beta_7 * x^6 + \beta_8 * x2^4 + \beta_9 * x2^6$) is the best fit for the dataset. Validation error is 15 percent and the accuracy is 85 percent.
In File7.mlx The Models are commented in the code section. To try a different model, We can uncomment the Code line for X and testX and see the Accuracy and error for a different model. To see the

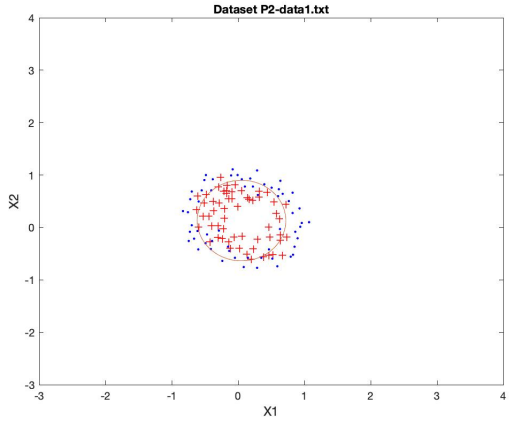

Figure 23: Model 3

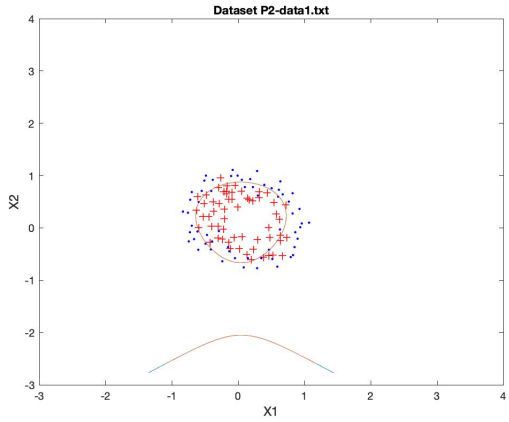

Figure 24: Model 4

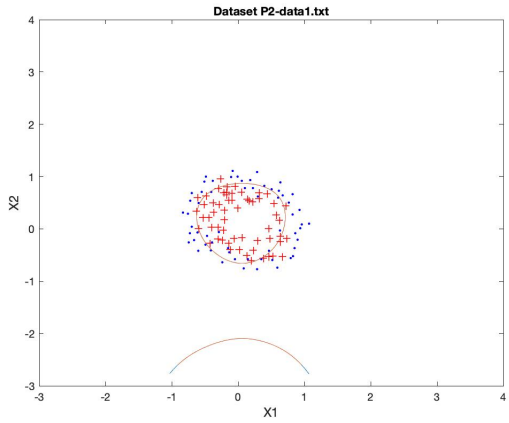

Figure 25: Model 5

plot, Changes need to be made in the plotDecisionBoundary() function.

3.5 Task e : File8.mlx

Repeat (c) and (d) with a regularization parameter of 0. Explain your observations.

Solution:
For lambda = 0 the training error= 20.33 and testing error = 12.5 percent. As shown in Figure 32.

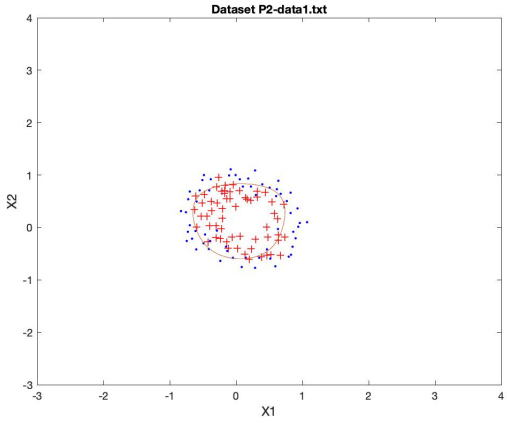

Figure 26: Model 6

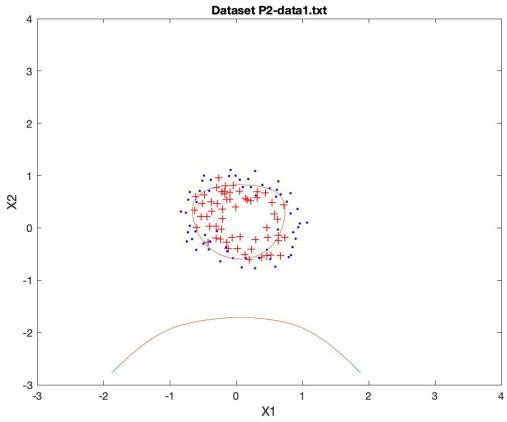

Figure 27: Model 7

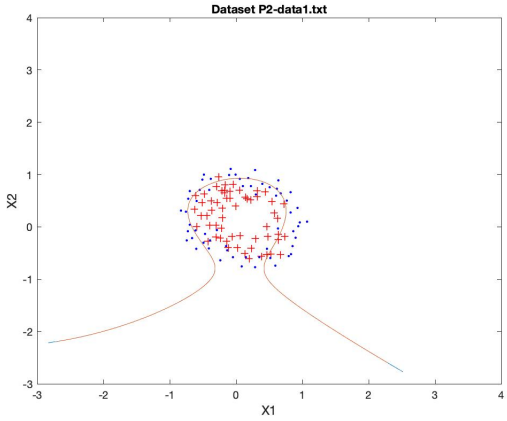

Figure 28: Model 8

Observation:
Interestingly, The two curves showup in the existing dicision boundary which is not at all related to the dataset.This happens because of the overfitting of out hypothesis function. Our selected best model has 6 degree polynomial to begin with If regularization is not there the 6th and 4th degree components are included as decision boundary as well. Reducing the regularization parameter increases overfitting.

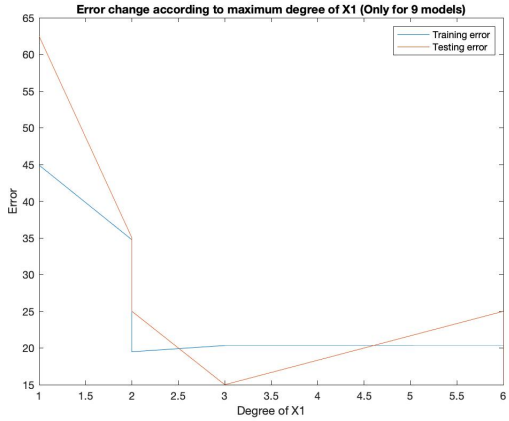

Figure 29: Degree of X1 with error percent value

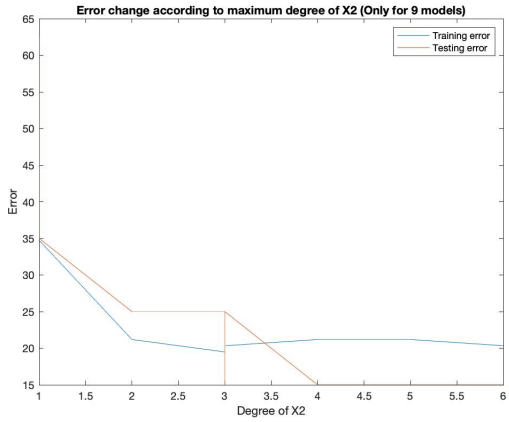

Figure 30: Degree of X2 with error percent value

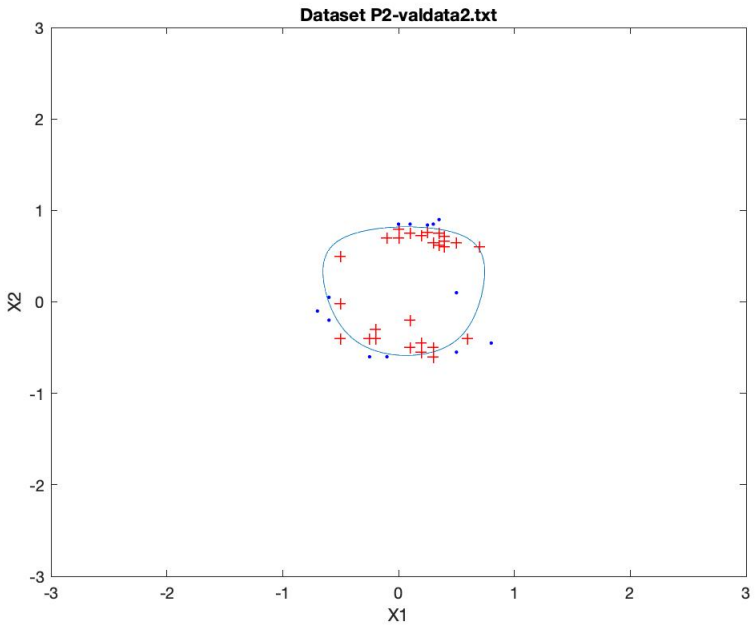

Figure 31: Dataset P2-valdata2.txt

3.6 Task f : File9.mlx

Repeat (c) and (d) with a regularization parameter of 100. Explain your observations. **For lambda =**

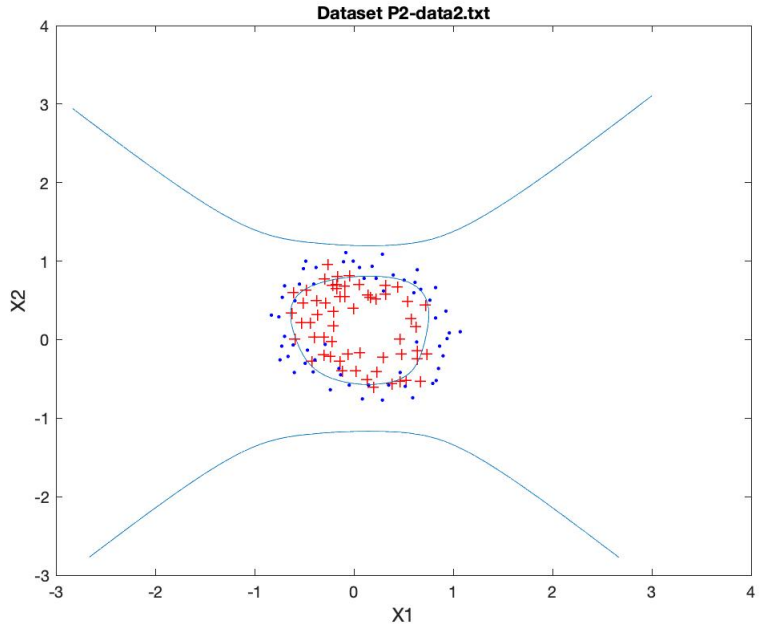

Figure 32: Regularization Parameter = 0

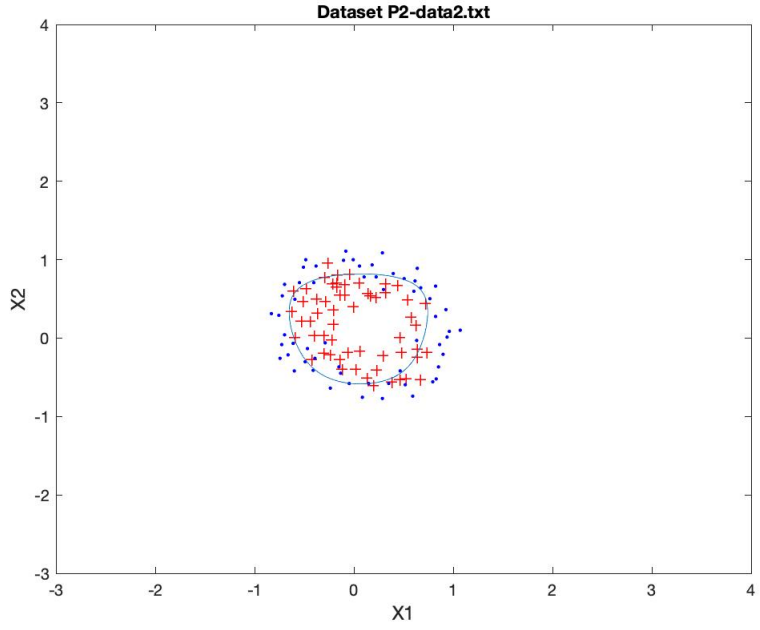

Figure 33: Regularization Parameter = 100

100 the training error= 20 percent and testing error = 15 percent.

Observation:
The regularization parameter reduces the over fitting in the logistic regression. Lamda affects the final accuracy only uptil some threshold value. Very large value of lambda does not make changes to the final result.