Santa Clara University
COEN 240 Assignment Pr2
Syeda Gousia Sultana- W1587235

Rank: 21
F1 Score: 62%
Learning rate:0.5
Number of iterations: 18000

The following steps were taken for the sentiment analysis of the movie reviews- •
Text preprocessing
- Creating matrix
- Building the logistic regression model

Text Preprocessing:
Multiple approaches were implemented for preprocessing text without using any libraries and I have encountered multiple roadblocks.

Approach-1:
Text Cleaning :-
In the first step, I have converted given train.dat file into lower case, removed all the punctuations by using the module **string** and importing punctuations. Then, I removed stop words by filtering out the words with less than 4 characters. I have created a list of lists of all the words in the file. Imported the module counter and created a dictionary with the count of all the words in the file.

Sparse-Matrix :-
By using the csr_matrix I have created a sparse matrix where I got 25000 rows and 276043 columns for the train.dat file which was different from the dimensions of the test.dat file. The dimensions of test.dat file are 25000 rows and 277673 columns, due to which it was throwing the dimension error. When I tried to reshape the matrix, I was able to reshape it to 25000X500 as follows:
M = csr_matrix((data, indices, indptr), dtype=int)
mat=csr_matrix((M.data, M.indices, M.indptr), shape=(25000,500),dtype=double)

But when I tried to run this ,it showed the same error of “column index exceeds matrix dimensions”. Then, I tried running it changing the dimensions of the matrices by only taking the unique words in the document. The accuracy was very low when I trained the model by fitting this data. This approach was discarded.

Approach-2:
Text Cleaning :-
In this approach, the first step includes converting the train.dat file to lower case, removing all the punctuations and special characters using the **regex** module, followed by removing all the new line characters and combine all into one. Next I created a list of lists of words by splitting the text.

Encoding the data:
Later I encoded the data by creating a dictionary. The most common words in the file are given the lowest integer so that all the unique words are recorded. Since the most frequently

occurring words are the stop words, I removed all the stop words and truncated the lines with much longer length. This way I was able to scale the columns. I then padded the columns with zeros if the length was short and if it was long deleted a few words. The matrix that was obtained from slicing the data was converted to array and this data was used to fit in the model.

Model:

I used the model of Logistic regression to fit and predict the data. The sigmoid function is used to bound the hypothesis function between values 0 and 1. To calculate the cost function for minimizing the cost for every iteration I used the logistic cost function. The partial derivative of the cost function and calculated the gradient to find the next steps or the theta values**.** The given problem is a classification problem where the target variables or labels are dependent on the input data and is categorical. The associated parameters were obtained through a lot of trial and error. Implemented gradient descent for optimizing the cost function. If there are a lot of iterations it takes a lot of time and the cost starts increasing after it has reached a particular point. When I used the maximum likelihood cost function there are negative values produced for the cost function and the accuracy remained the same even after all the iterations. The learning rate which controls the steps it takes plays a very important role. If the learning rate is high, it is not converging and producing nan errors. When it is low, it is taking a lot of time to converge. The number of iterations is also equally important, if it is low it stops iterating which leads to ending the program before error is minimized which leads to poor accuracy.
Bias/Variance analysis: Managing to reduce both bias and variance will lead to a more accurate model. High bias and variance both are bad for a model to generalize. When the model had high bias it lead to underfitting, whereas high variance tends to overfitting which gave a high accuracy.