ITCS 6156: Machine Learning

Homework 2

Student Name: Rajdeep R Rao

For this homework assignment, I chose to work with python because I haven't worked with this language before, so it was a wonderful opportunity to learn. Also the numerous libraries that are available made it easier to do the project.

Steps Taken:

I first read, said data into a DataFrame using one of the libraries in python named 'Pandas'. I then converted the data into a binary representation of 1s and 0s instead of number of occurrences (Frequency) but the output was almost the same in either case. This data frame was then converted into an appropriate format required for the logistic regression classifier in the scipy library.

Logistic Regression:

Logistic regression is a categorization model used in machine learning. Logistic function is used to classify this data because of the two different asymptotes, it can be used to divide data into "yes/no" categories -- the low side being "no" and the high side being "yes." Logistic function is as follows:

$\sigma(t) = e^t / e^t + 1$

Logistic function is a sigmoid function by nature. We can categorize the data into binary or multiple categories(using one-vs rest method). If multiple categories are to be used, then it is called as multinomial logistic regression.

Logistic Regression Implementation:

For the implementation of Logistic Regression, I used a function called '**LogisticRegression'**. I also set the multi_class parameter to 'ovr' so that it becomes a multi class classifier as opposed to a binary classifer. I have also used the inverse regularization factor (C) as 1e5. This function performs the **Logistic Regression** algorithm on given data when inputted after transforming it into an appropriate format and then, a trained model is returned. This model is then tested against the test data. Finally, I matched the output with the labels in the test6.data using the accuracy_score function which displayed the **accuracy of this model to be around 32%**. Since the features are so much larger than the dataset, it is justifiable that the accuracy wouldn't be so good. After having trained the model using training data, having the model predict values for the same data yielded a good 99% using the score function, to prove that the model does indeed train well enough Also, Since I've truncated the value of the tables, I've hardcoded the column values of the frequency tables in the code. So, it's not flexible for all datasets.