**ASSIGNMENT 1: MACHINE LEARNING (CSE-6363**

**NAÏVE BAYES CLASSIFIER FOR TEXT CLASSIFICATAION**

**NAME: ABHISHEK JAIN**

**UTA ID:1001246087**

**Email Address: abhishek.jain2@mavs.uta.edu**

* **IMPLEMENTATION**

**The training and testing file is separated while file\_reading operation by limiting the input data for training from [:500] and then the rest of the file is taken testing to form [500: till eof]**

**TRAINING**

1. DATA PREPROCESSING:

* firstly, I extracted all the training data in one file
* Then using regular expression, to split the sentence into words.
* After that I removed all the English language stop words from the file
* Now for each word, I stemmed them to root words
* The above functions were implemented using the “nltk” standard python library

1. LOGICAL PART

* The naïve bayes classifier used in this project for classifying the news article in the newsgroup. The 20000 articles provided in the dataset are divided into 20 classes
* To implement this, I have taken 500 files as training dataset from each of the 20 classes
* In the part , I calculated the word count of the all the words received after preprocessing in the form of dictionary .
* Now we calculate all the unique words occurring in the all the classes and its count

**TESTING**

1. DATA PREPROCESSING

* It’s almost same as training with one change by taking each test file at a time for prediction.

1. LOGICAL PART

* In this part calculate probability for each token.
* Naïve bayes Implementation by using laplace correction

Probability = p(class) \* log(1+wordcount of token))/(total words in the respective class+ unique words in all classes)

* Once we calculate the probability of each word and adding them to given probability of the document it is belonging to the particular class.
* The class probability is stored in “Probabilitydict”, from which we take the maximum probability to label the class for that document/article.
* If article label matches the class label of that article,then the count is incremented by 1 for correctly predicting the class
* **RESULT**

The objective of the program is to predict the class label of the news articles in the test data using the naïve bayes classifier.

1. Initially, with 500 training and testing data, the accuracy of the function was 78.256%
2. When the number of training data is increased to 700 and testing dataset to 300, the accuracy of the function then increased to 82.421%

The accuracy of the program can be further increased by stemming and lemmatization of the tokens generated after preprocessing. Clean dataset results in increased accuracy of the function.