**ASSIGNMENT 4**

**MULTINOMIAL NAÏVE BAYES**

**Introduction:**

Textual Data in various forms is available online and offline. To be able to analysis the text correctly by machines based on the context as a human would think of is a big challenge. To predict the sentiment or do classification, the dataset containing text such as product reviews, public speeches, authors writing etc. various algorithms and techniques can be used. Naïve Bayes algorithm is one of them. Naïve means lacking any bias towards the features present so there is equally likely probability of classifying the data into the available classes/documents as the features are considered independent of each other. The classification into known classes is done for the unknown text as it already has labels assigned to the classes being a supervised classification model.

This model is one of the most widely used methods for text classification into the categories defined with the features being word frequency in the document. It calculates probability of the unknown document belonging to a specific category and the category with the highest probability gets assigned to the document to be classified.

**Data Description:**

The dataset used here has 2 parts one for sentiment and other for lie detector. The dataset is in arff format. There are 46 negative and 46 positive words in the sentiment text visible after loading the data. There are 46 words in lie detector as fake and true categories to be classified. The Multinomial Naïve Bayes algorithm has to applied to be able to predict and find the negative, positive and the fake and true words being classified using Weka.

**Methods:**

I used Weka to load the data in arff format on which the text classification is to be applied. The filter used here is StringToWordVector found inside unsupervised category under attribute section. For tokenization I have used all the default settings only 2 settings were changed. The outputWordCounts was set to True to be able to use Term Frequency of the words and not the Boolean representation because we are interested in their count and not only their presence/absence. Also, the lowerCaseTokens setting was set to true for the tokenizer to convert the strings into lower case. For Classification Multinomial Naïve Bayes with default settings was used. To validate the model, I have used 10-fold cross validation. The more the number of cross folds, the better is result but it is slow as compared to Hold out test. The number of decimal places in Multinomial Naïve Bayes was set to 10 instead of 2 to see the subtle difference between the probabilities of the word for the 2 categories. The Prior and conditional probability is calculated for each word and is multiplied to get Posterior probability which determines the which class the word/document should belong to. To find the 20 most positive and negative words in the text for the lie detectors and sentiment I have used python code. In the python code I loaded the data using .tsv file and features which are words in this case are created. 2 other categories in which the classification is to be done is created like fake and true for lie detector and positive, negative for sentiment analysis. The classification is based on the probabilities calculated under each category for each word and then assigning the highest probability category to the word. The words such as numerical values, symbols, junk strings such as 10, 10/2, $$, 10% were removed before being used to find out the most positive and negative words for the 2 documents lie detector and sentiment analysis which would not make sense and will cause errors in the model for classification.

**Results:**

For tokenization I changed the setting to lowercase letters True which changes the analysis and the numbers for the model. The top 20 words for sentiment analysis based on classification into positive and negative are calculated by taking the log difference between the conditional probabilities for each word. Th conditional probability is the probability of word in the class given a class which is obtained at the classifier output terminal in Weka further used to find most positive and negative words. The log difference is taken to avoid the very small probabilities such as 0.000 kind to underflow the calculation and may lead to misclassification.

The top 20 words for sentiment will be ranked based on the highest log difference of the conditional probabilities. The most positive words are customer, few, flavors, hot, liked, noodle, pick, rice, soft, town, atmosphere, friendly, ingredients, love, restaurants, sushi, best, makes, always, amazing. The most negative words are terrible, took, asked, no, said, worst, come, hour, bland, seated, came, drinks, found, half, minutes, our, pm, until, waited, bad.

The top 20 words for lie detector will be found out using the same criteria. The most positive words in the lie detector text are grocery, ignored, lunch, noisy, old, only, packed, pepper, shrimp, sister, sommalier, tables, thing, variety, what, been, calling, coffee, makes, tofu. The most negative words in this text are Plate, cold, tea, iced, Southern, could, ice, Comfort, chocolate, drinks, expensive, outstanding, All, Chipotle, Marshall, along.

I found that “few” is categorized as one of the most positive word in the sentiment set which is negative word. Using A few would result in the positive word. Other words such as amazing, friendly, liked, best all of them are positive words which are classified properly by the model. Hour, minutes, come, came, our, said are classified as negative words which are not negative and are misclassified. The most negative and positive words such as bland, worst, terrible, bad, best, amazing appear when we find out using the log difference and are not available using the highest conditional probabilities which makes it better classification model. The words whose highest conditional probabilities are seen are not necessarily negative or positive in nature. The word restaurant is classified as both positive and negative if using highest conditional probabilities which is a not correct. One more result which was noticeable is that in sentiment data restaurants was classified as one of the most positive words with a difference between the probabilities and the word restaurant singular as both positive and negative for higher conditional probability which makes it ambiguous and puzzling for anyone as to why such pattern is shown.

The models are good at certain points learning to find words such as amazing, worst etc. but didn’t perform well on words such as restaurant, said, hour, minutes. To evaluate model performance, we can look at the other aspects such as Accuracy, Precision, Recall and Confusion Matrix. For lie detector the performance of the model was 51% accuracy to classify correctly into the categories fake or true. The Precision and Recall values are also very less for this lie detector set which makes it not a good predictor in general. But for a lie detector to be able to predict correctly 60% of the time is very good as compared to achieving a 90% correct classification. There are equal number of observations in each category and there are 2 categories only, so the baseline is 50%, the dataset is balanced so getting accuracy of 60% beats the baseline to be able to say it is a good model, but the accuracy needs to be increased.

The model for sentiment analysis is achieving a high accuracy of 80% to classify correctly into negative and positive which is a very good performance. Also considering Precision and Recall which are 80% makes this a model a very good model. The difficulty in lie detector is clearly visible by looking at all the parameters including accuracy along with more words being misclassified.

The Computers are not properly able to detect lies as they can classify love, Plate, Chipotle etc. as negative. The percentage of times they are able to detect lies correctly are very less as compared to sentiment detection. The words such as Chipotle, Marshall etc. are the examples of Proper Names which cannot be classified into either category but in this case are fitted into one of them as there are only 2 categories.

**Conclusion:**

The model performs very well for the sentiment analysis problem but doesn’t perform well for the lie detectors if only taking numbers into account. Lie detection is more complex problem which would require better and more data to be able to correctly predict the fake and the true news. Machines are not able to classify the lies properly whereas they are able to do sentiment analysis well. The performance of the model being quantified by other parameters gives the same result as the baseline is the same and the dataset is balanced which makes it 50% chance of going into category so anything above this percentage is great but results in errors which cannot go unnoticeable. I would say the result can have neutral as a 3rd category which will set a new baseline and such misclassifications could be avoided by using other metrics for evaluation which have nothing much to say for this type of data.