**ASSIGNMENT 5**

**Bernoulli and Multinomial Naïve Bayes in Scikit-learn**

**Introduction:**

Text data available in various forms both online and offline needs to be analyzed to be able to figure out the ground truth. To achieve this task, Naïve Bayes algorithm is used which assumes no bias towards the features in the data due to which it makes better predictions as there is equally likely probability for the classes to be selected. There are two types of Naïve Bayes algorithms used widely. Multinomial which uses term frequency or count of frequency of words as the input whereas Bernoulli uses Boolean representation meaning just looking at presence and absence of the word in the document. The goal of this assignment is to be able to build 3 models for each of the Naïve Bayes Algorithm for two different type of problems sentiment analysis and lie detection based on different vectorization methods, compare and select best one as per the requirements. We are using Scikit-learn package which is very fast and can be used to perform analysis of vast amounts of data which would be a tedious task to do in Weka.

The calculation of probability is made for the both the categories of the labels for the new data in each different problem and assign this unknown data a label correctly based on the highest probability of the class.

**Data Description:**

The dataset used here has 2 parts one for sentiment analysis and other for lie detector. The dataset is in .csv format. There are labels assigned to each of the review. For Sentiment Analysis, the labels are (‘p’,’n’) meaning positive and negative whereas for lie detector the same reviews have a label of (‘f’,’t’) fake and true. The Multinomial Naïve Bayes algorithm is applied to be able to predict and assign the negative, positive and the fake and true labels for the 2 datasets correctly based on training provided to it. There are in total 92 reviews in the .csv file with the ground truth already assigned to it as labels for us to confirm if the model on test data is labeling the reviews correctly or not. Both the Naïve Bayes Algorithms accuracy in assigning correct labels to the test data is to be compared.

**Methods:**

I used Scikit-learn package through python to perform the vectorization and applying the 2 Naïve Bayes Algorithms to be able to solve our problem in each problem case. For this I loaded the data from .csv into pandas. For the Sentiment Analysis, the sentiment column has labels p,n and the review column has reviews as data. For the Lie detector, lie column has labels f,t and the review column has reviews as data. So, for each problem 2 columns are used labels and their corresponding data values. The data is split into 0.4:0.6 ratio for the hold-out test where 40% of the data would be used for the testing purpose and 60% for the training purpose making it a total of 55 for training and 37 for testing data. Labels are checked in both training and testing data. Different Vectorizations methods including Boolean for the Bernoulli as the input and for the Multinomial term frequency or the count of the words as the input would be used along with different other parameters being changed. The model will be trained on the training data and then tested on the test data for calculating the accuracy, precision, recall, f1 score for the various vectorization techniques used to compare them. Also, error analysis would be helpful to find the false positives and false negatives being classified by the model used. For the cross-validation evaluation, we can use whole data and define the number of folds in which we want to do our evaluation to find the accuracy of the model as compared to hold-out test.

Various vectorization techniques used in this assignment. First for the Multinomial NB, I used frequency count as the input looking only for unigrams along with removal of stop words in English Language. This vectorization was selected to provide unigram term frequencies as the input to Multinomial NB. The default setting in the CountVectorizer for this technique was set to True to avoid double counting of words and removing stop words was important to not count frequency of words not important and occurring in abundance in the text. This helped to identify unigrams without same words being counted twice. In another technique used for Multinomial NB involved Unigrams along with Bigrams for the term frequency with lowercase letters and no stop words was used to find the bigrams which may not have been identified by previous technique. There can be instances where bigrams are able to give more meaningful words as compared to unigrams only. For the last technique, I used Unigram, Bigram and Trigrams all together, not using lowercase letters with removing stop words. This is useful to find the words which could be repeated due to both in lower and upper case along with trigrams to see if we find any more useful words taking 3 words at a time to be able to find repeated words due to case sensitivity. Also tried to use TFIDF technique with stop words removal and unigrams. This technique is useful to see the words which may not be frequent based on fractional frequencies being provided as input to Multinomial NB. TFIDF is not used widely with Multinomial NB because it works well on the discrete frequencies which are provided by Term Frequency. Removing Stop words would help in increasing accuracy. For Bernoulli binary parameter was changed to True to provide Boolean representation rather than word frequency along with changing the training function name from MultinomialNB() to BernoulliNB().

**Results:**

For Lie detector using Multinomial NB with Unigram frequency count without stop words as input using hold-out test, accuracy turned out to be 49% along with recall and precision being the same. Same number of false positives and false negatives were found out using the confusion matrix both being 9 each. The most informative features or the top words in each category based on probability were place, great, like, really, experience, best, good, ordered, went, food, restaurant starting with ones favoring the most negative(fake) to the end which favor other class(true). Error rate is very high, and it does not meet the baseline set as the metrics gives only 49% accuracy. For 2nd technique with Bigrams being included, the outcome did not change the accuracies and the most informative words based on their log difference was the same as unigrams. So, this technique did not produce any difference using only unigrams or both unigrams and bigrams gave the same result. For the 3rd technique, which involved trigrams, bigrams and unigrams without converting the text in lower case, accuracy and the other metrics such as precision, recall, f1 score didn’t change but there was change in the words bag. For the earlier 2 cases with lower case being used there were 11 words as in the bag but for this there are 14 words. This leads to conclusion that not using lower case did affect in the increase and new words being coming out for the classes. The number of false positives and false negatives changed to 10 and 8 instead of 9 each for the earlier two methods. The top features in each category based on the log difference changes with words such as This, It, we are being favored by one class which is a very surprising result as compared to 2 methods where lower case was used. For TFIDF method, the accuracy of the model reduced to 41% along with precision, recall and f1 score being the same. The top and bottom feature words remained the same but their co-efficients did change. Also, removal of stop words did increase the accuracy overall as compared to non-removal of stop words bit a very slight margin of around 1% but for TFIDF case this change in accuracy fluctuated by around 4 to 5% which is another result which was unique. So, using ngrams differently, with or without lower case and keeping or removing stop words also using TFIDF giving different accuracies and other metrices for the model evaluation.

For Sentimental Analysis using Multinomial NB, first with only unigrams with no stop words and lowercase, the accuracy was 73% with other metrices like Precision and Recall being 76% and 72% for the macro avg (taking all the categories equally weighted). Only 8 being false positives and 2 false negatives. The error rate overall is very low. The most informative features based on their ranking were ordered, place, went, great, really, great etc. For the sentiment analysis the words are not too different than for lie detector, but the ranking and coefficients differ making the model work on same reviews differently for different problems. For example, word such as best was favored for different class in lie detector and for this problem different class. For bigrams, there was no change in the results for the most informative features or the accuracy precision. The use of bigrams in the vectorization didn’t produce any change which is also unique result that came out for these 2 problems being solved using scikit-learn. For using all unigrams, bigrams and trigrams without lowercase and no stop words, the accuracy of the model reduced to 62% only as compared to 73% in the earlier case. The Precision, Recall also reduced to 65% and 63%. The number of false positives increased to 11 and false negatives to 3. The Error rate has increased overall. The most informative features included words such as we, the which I find intriguing if not using lowercase. For TFIDF technique using unigrams in lowercase without stop words, the accuracy was 70% better than previous one without lowercase and trigrams. The number of false positives reduced to 8 and false negatives to 3. The precision and recall increased to 72% and 71%. This means even if using TFIDF for Multinomial NB proved better at accuracy, informative features, error rate as compared to trigrams without lowercase.

For Bernoulli Naïve Bayes, the input is binary representation rather than term frequency. For the first case in Lie Detector problem, the accuracy with unigram Boolean removing stop words is 46% along with Precision and Recall both are 46%. The number of false positives were 9 and false negatives were 11. The number of false negatives detected by Bernoullis NB was more, but false positives same 9 as compared to same settings for Multinomial NB. For the 2nd case where bigrams were used for Boolean representation and removing stop words, it gave the same result for the accuracy, precision and recall. Also, the most important features top and bottom were the same for unigrams as well as the bigrams. The most informative features include great, place, like, ordered, went, food, restaurant. The order is changed as compared to Multinomial NB for some of the words based on the change in the coefficients value. For the 3rd case where unigrams, bigrams and trigrams without lowercase and no stop words used, the accuracy was the same, but precision and recall changed to 47%. The most important features match up to the ones with Multinomial NB including The, we etc. Also, one thing noticeable was that the position of restaurant word changed from 9 to 12 during the vectorization of training data, got shifted by 3 places for lowercase not used and the bag of words increased in size from 11 to 14. The number of false positives reduced to 6 but false negatives rose to 14. For TFIDF with lowercase and removing stop words, accuracy is same 46% but false positives were 9 with 11 false negative. No difference in Precision and Recall from unigram combination used in 1st technique.

For Bernoulli Naïve Bayes in sentiment Analysis, the accuracy with unigram Boolean removing stop words is 70% which is 3% less than Multinomial NB for the same technique. The Precision and Recall are 76% and 71% respectively. The number of false positives is 10 and false negatives just 1. Error has been reduced. The most informative features are the same as in case of Multinomial NB with the change in the coefficients being reduced. For bigrams without stop words and lowercase, the results are the same no new conclusion can be found out making the bigrams not useful for this case. For the final case, unigrams, bigrams and trigrams without lowercase and no stop words used, accuracy reduced to 68% and precision, recall to 75% and 68%. The position of restaurant word in training vectorization changed to 12 from 9 shifted by 3 places. The number of false positives increased to 11 and false negatives remained same at 1. For TFIDF with lowercase and removing stop words, accuracy is same 70% but false positives were 10 with 1 false negative. No difference in Precision and Recall from unigram combination used in 1st technique.

**Conclusion:**

For this case, both in Lie detection and Sentiment Analysis the accuracy didn’t vary much. There was difference of only 3% for both the cases when using the 2 Naïve Bayes Algorithm techniques. But the performance was better for Sentiment Analysis beating the baseline where for Lie detector both the algorithms didn’t meet the baseline to be called models. Cross fold validation with 3-fold was also tested on some cases which was better for evaluation than hold-out test which was done majorly. Multinomial NB is useful when the classes/features are discrete values and Bernoulli when the features are Boolean values (True/False or 0/1). Here the features were discrete and can also be treated as Boolean without making much of the difference so there was no much difference between the two. Also, the input passed to the two models is different Boolean to Bernoulli and word frequency to Multinomial NB. So, to use Boolean Multinomial Naïve Bayes should not be used. Removing stop words did make a small difference in the result. Not using lowercase showed visible difference by shifting the words position and increasing the bag size of the words thereby also adding the, we kind of words to the top informative features. The Precision and Recall of the two models can also be compared as an advantage to see if we can select one of them. TFIDF technique gives same accuracy for the different vectorization techniques in case of Bernoulli but for Multinomial NB using TFIDF reduces accuracy.