Multinomial Naive Bayes Classifier

Implementation Process :

Pre - Processing Data :

At first the data is filtered based on the selected set of valid categories (as mentioned in the problem statement).Further the dataset was column filtered as well because our dependent column was ‘category’ and independent column was ‘headline’.
This was followed by splitting the data into test - train sets
Preprocessing the data involved extracting the vocabulary from train set.For extracting vocabulary the following steps were taken :

- Removed Punctuation from every Headline.
- Removing Stop Words from every Headline.
- Converting each word to lowercase so as to prevent distinction between the same words of different case structure.

The Vocabulary set was made such that for every word a dictionary of the below form is maintained :

```
word_class_dict[word]={'business':0,'comedy':0,
'sports':0,'crime':0,'religion':0,
'healthy living':0,'politics':0}
```

So that whenever we are iterating over a particular headline(preprocessed) , for each word the class (category) value is incremented as per the category mentioned for the headline in the Y_Train set.Further we also maintain another dictionary called class_word, which gets incremented every time a word in that class(category) is counted.Thus after scanning all the headlines the frequency of each word for each category gets updated and class_words stores number words belonging to every class.Further, for words having $sum(count\_freq\_every\_class) < 2$ are deleted from the vocabulary set (as per problem statement).

Model Training :

This step mainly involved calculating the conditional probability values and prior probability values :
The Prior Probability for every category is calculated as :

$$P(c) = \frac{N_C}{N}$$

where N_C =Number of headlines per category we calculate directly from Y_Train and N is total number of headlines

The conditional probability is calculated as :

$$P(t|c) = \frac{T_{ct}+1}{(\sum_{t' \varepsilon V} T_{ct'})+B'}$$

where $t = term, c = category, V = Vocabulary \text{ and } B' = |V|$
T_{ct} is number of words(term) t in all the documents of class c
T_{ct} is calculated from word_class_dict[word][category]

$(\sum_{t' \varepsilon V} T_{ct'})$ is calculated from class_words[category]
The $+1$ *in numerator and* B' *in denominator* are used as part of Laplace smoothing Technique to handle zero probabilities.
B' is calculated len(word_class_dict.keys()) to vocabulary size.

Prediction :

Since we have already calculated and stored the conditional probability values for each term and each category, so from the test set for each headline we extract the vocabulary and calculate score for each category by taking log of the conditional and prior probabilities to prevent underflow.

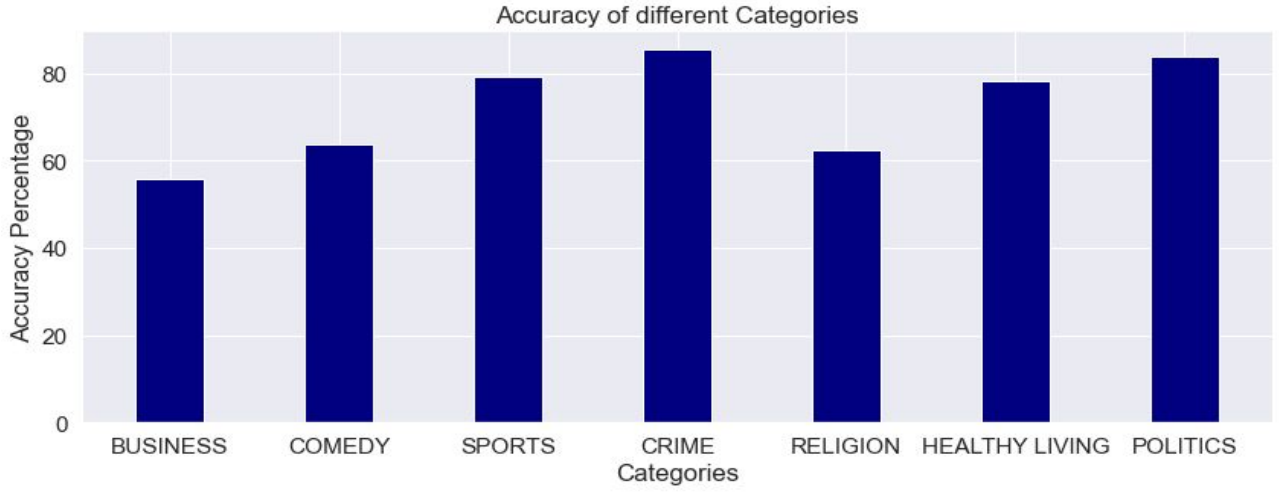
Model Accuracy Analysis :

Overall Testing Accuracy : 78.55%

Class wise Accuracy :

----------Class Wise Accuracies----------

BUSINESS Accuracy = 55.99%
COMEDY Accuracy = 63.65%
SPORTS Accuracy = 79.46%
CRIME Accuracy = 85.45%
RELIGION Accuracy = 62.47%
HEALTHY LIVING Accuracy = 78.47%
POLITICS Accuracy = 83.87%



Confusion Matrix

Model Training and Preprocessing Time : 1.54 seconds

Inferences :

- Thus we get the highest testing accuracy for ‘CRIME’ category and least for ‘Religion’ Category.
- The accuracy per class and overall accuracy is better as compared to multivariate model
- Due to using a dynamic programming approach to store and use the number of documents per class value the model preprocessing time got highly reduced.