**Assignment 3 Report**

Name: Prajwala Mugajji Shambulingappa

**Aim:**

Working with Naïve Bayes classifiers.

**Observation Criteria and Analysis:**

As part of the project, Dataset used is provided by Kaggle which has a large number of Wikipedia comments that have been labelled by toxic behaviour by human reader.

Packages used: From Naïve Bayes: GaussianNB, BernoulliNB, MultinomialNB, CategoricalNB. Other: pickle, CountVectorizer, TruncatedSVD, csv, rex

* Data was initially preprocessed to tokens by lowercasing, removing whitespaces, splitting and removing empty strings. While normalization, I choose not to exclude the stop words as that would impact the contextual training of the model.
* During training, I used the CountVectorizer to convert tokens created from previous normalization into a matrix of token counts. After this I passed the matrix to train our classifier model, however here the fit was very slow and due to insufficient RAM runtime kept crashing, thereby I had two options to either pass the datasets in batches or use TruncatedSVD. I chose to go with TruncatedSVD, which is a dimensionality reduction technique.

Also, I choose to train multiple Naïve Bayes models, to compare the accuracy and understand which model performs better.

* During testing, I again faced runtime kept crash, so I used TruncatedSVD. However, the testing was extremely slow and kept executing for over 30 mins so I divided the dataset into smaller chunks and passed the chunks one by one. After which I calculated the probability and classified the text as toxic or not. The classification was compared with the test label to evaluate the models accuracy.

**Summary:**

1. Classifier Models was successfully implemented to analyse a dataset which has a large number of Wikipedia comments that have been labelled by toxic behaviour.
2. Model accuracy:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | GaussianNB | GaussianNB with Grid\_search | BernoulliNB | MultinomialNB |
| Accuracy (in percentage) | 17.28% | 17.31% | 52.50% | 95.97% |

1. It can be observed that the Multinomial NB performed well.
2. With GaussianNB classifier, the model underperforms due to overfitting. Hence, I choose to reduce the over fitting using regularization using alpha parameter to control the strength of L2 regularization.

This increased the model’s performance to 17.32% which wasn’t much of help.

1. In attempt to improve GaussianNB classifier’s performance, I tried to implement feature engineering by adding a new column to the dataset “text\_len”, that has the length of each comment. Then I passed the updated dataset for preprocessing and the trained the model. This however didn’t help much.
2. I feel that using a neural net model like LSTM would be a better way to improve the model’s performance.