Name: Bhavani Sai Shriya Anumala
ID: 1001870184

Project 2 - Report

MACHINE LEARNING

The main objective of the project is to classify the text documents of the 20_newsgroup dataset using the naive bayes classifier.
Here I have used the implementation of Multinomial Naive Bayes classification algorithm on 20 newsgroups data.
The 20 Newsgroups data set is a collection of about 20,000 newsgroup documents, divided evenly among 20 different newsgroups.
We need to predict the newsgroup to which the document will belong after splitting the data by 50% and after data preprocessing.

CODE :
DATA PREPROCESSING:

1.Import the libraries to read the path of the file.
2.Split the data into training and testing data by setting the test_size as 0.50. Where we use 50% of the data for training and the other 50% for testing.
3.Later data preprocessing of the text documents is done by removing the stop words and also converting the words to lowercase for consistency..
4.Extract the vocabulary from the documents by using regular expressions.

Implementation of Naive bayes algorithm:

Here we are using the naive bayes classifier to predict the class of the text document and here's how it's done.
Features are independent in each class document. They are unique in every newsgroup.

$$P(class|data) = (P(data|class) * P(class)) / P(data)$$

Depending on the features we can predict class of a document.
If we find the features extracted in document belonging to the features of a certain class then we conclude that the document is of that particular category

We have used log probabilities to avoid underflow and in order to avoid the difficulty of revisiting the words again.

$$Pr(j) \propto \log(\pi_j \prod_{i=1}^{|V|} Pr(i|j)^{f_i})$$

$$Pr(j) = \log \pi_j + \sum_{i=1}^{|V|} f_i \log(Pr(i|j))$$

Now ,the final step involves the testing of the accuracy on the test_data
We have performed predictions on each document of the test data.The accuracy of the model is 78%

SUMMARY :

Naive Bayes Classifiers are fast and easy to train.
Naive Bayes can perform same as or sometimes better than logistic regression and SVM.
We have used multinomial naive bayes classifier as there were so many files that we needed to track the frequencies of the features.
After applying multinomial naive bayes we observed that the accuracy was 0.78 and after developing the naive byes model from scratch we observed that it was again 0.78.
Hence we can say that the accuracies were the same.
Since, the folders have huge amount of data the runtime of the code tends to be very long.

REFERENCES:
https://towardsdatascience.com/multinomial-naive-bayes-classifier-for-text-analysis-python-8dd6825ece67
https://www.youtube.com/watch?v=60pqgfT5tZM&t=428s
https://stackoverflow.com

Result:

```
# Using Multinomial Naive Bayes
clf = MultinomialNB()
clf.fit(X_train_data,Y_train)
Y_test_pred = clf.predict(X_test_data)
sklearn_score_train = clf.score(X_train_data,Y_train)
print("Sklearn's score on training data :",sklearn_score_train)
sklearn_score_test = clf.score(X_test_data,Y_test)
print("Sklearn's score on testing data :",sklearn_score_test)
```

```
Sklearn's score on training data : 0.8200230034505176
Sklearn's score on testing data : 0.7872180827124069
```