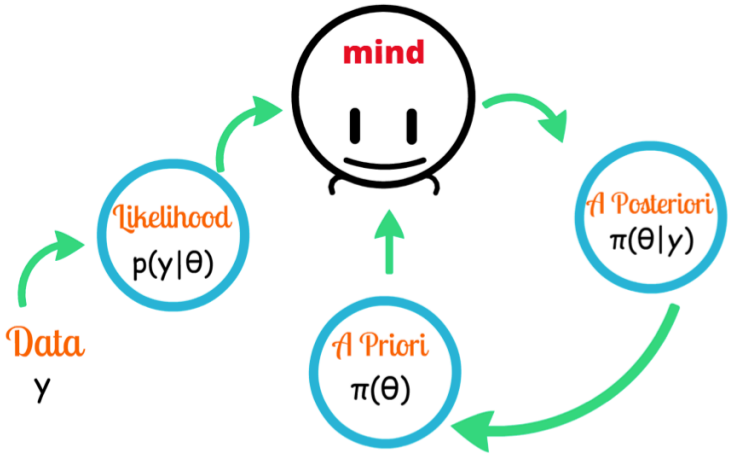
DIKSHA SHARMA
1001679176

PROJECT 2 REPORT
MACHINE LEARNING -6363-001

This project involves the implementation of **Multinomial Naive Bayes classification algorithm on 20 newsgroups data.**
The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups.
We need to predict the class/newsgroup in which the document will lie after performing probability calculations and data preprocessing.



CODE IMPLEMENTATION
Language Used - Python3
Environment- Anaconda(Jupyter Notebook & Spyder)
1.Import the libraries to read the path of the file and the current working directory.
2.After reaching the documents inside the folders, shuffle the data in order to train the model in the best possible way.
3.Split the data into training and testing set by applying slicing leaving 500 documents in each of the 20 folders for training and testing.
4.Perform preprocessing of the text in the documents by removing the metadata at the top of each document and removing stop words and block words which have the least significance while performing classification into categories.
5.Extract the vocabulary from the documents by using regular expressions and making the words consistent by converting all of them into lower case.
6.**Now, we start the implementation of Naive bayes algorithm-**

APPROACH-
Assumption: Features/words are independent given class/categories of the documents.
Typical additional assumption – Position in document doesn't matter: $P(X_i = x_i | Y=y) = P(X_k=x_i | Y=y)$
In order to calculate the **Prior P(Y) probability** ,count how many documents you have from each group (+ prior)
For each topic, count how many times you saw word in documents of this topic P(Xi |Y).
In order to test the model, we use Naive Bayes decision rule.
CODE FOR NAIVE BAYES:

1. After obtaining the preprocessed words list, we calculate the probability of each word.
2. As the count of some words will be zero, we will **apply Laplace smoothing** by adding 1(alpha) to the numerator.

DIKSHA SHARMA
1001679176

- Use naïve Bayes decision rule

$$h_{NB}(\mathbf{x}) = \arg\max_{y} P(y) \prod_{i=1}^{LengthDoc} P(x_i|y)$$

3. Now, we combine probability distribution of P for words with fraction of documents belonging to each class by mapping the probability of words with extracted vocabulary.
4. We have used **log probabilities instead of multiplication in order to avoid underflow** and in order to avoid the probability to go up if the word appears again, we take the log of the word count/frequency.
5. After calculating the total probability, we train the model by finding the maximum probability for the posterior calculation for the category.

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

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

6.Now ,the last step involves the testing of the probability on the test_Data by implementing a counter.We have performed predictions on each document of the test data.The accuracy of the model is 85%.

SUMMARY AND OBSERVATIONS:

Naive Bayes Classifiers are fast and easy to train. When we select the variables properly, Naive Bayes can perform as well as or even better than logistic regression and SVM. Naive Bayes requires a strong assumption of independent predictors, so when the model has a bad performance, the reason leading to that may be the dependence between predictors.

We have used Multinomial naive bayes classifier as the number of occurrences/multiple frequencies matter a lot.

We know that more training data = better model and more testing data = better accuracy on testing results.In order to get the true accuracy(if you have an infinite number of test examples, this is the accuracy you'll get if you test your model.),

With a larger training set, this true accuracy increases. With a larger test set, you get a better estimate of your model's accuracy. How you divide the data up depends on how much importance you put on these two things.

REFERENCES:

https://www.cs.cmu.edu/~epxing/Class/10701-08s/Lecture/lecture3-annotated.pdf
https://towardsdatascience.com/multinomial-naive-bayes-classifier-for-text-analysis-python-8dd6825ece67

RESULTS:

Here,we can see that the first line shows the predicted folder in which the test data /document has been classified according to the model.

The path for the document originally is the second line, which helps us to compare the result with the actual and the training data.

The last line show the probability/maximum probability that the document lies in the defined folder/category.

```
Folder in which the test data lie- alt.atheism
 Test data train_classify calculation-
path for the document- /Users/dikshasharma/Desktop/Machine Learning/20_newsgroups/talk.religion.misc/83559
probability- 0.6593668829471099
--------------------------------------------------------------------------------------------------
Folder in which the test data lie- talk.religion.misc
 Test data train_classify calculation-
path for the document- /Users/dikshasharma/Desktop/Machine Learning/20_newsgroups/talk.religion.misc/84331
probability- 1.0
--------------------------------------------------------------------------------------------------
Folder in which the test data lie- talk.religion.misc
 Test data train_classify calculation-
path for the document- /Users/dikshasharma/Desktop/Machine Learning/20_newsgroups/talk.religion.misc/82782
probability- 1.0
--------------------------------------------------------------------------------------------------
Folder in which the test data lie- talk.religion.misc
 Test data train_classify calculation-
path for the document- /Users/dikshasharma/Desktop/Machine Learning/20_newsgroups/talk.religion.misc/83504
probability- 0.5431260226201581
--------------------------------------------------------------------------------------------------
Folder in which the test data lie- soc.religion.christian
 Test data train_classify calculation-
path for the document- /Users/dikshasharma/Desktop/Machine Learning/20_newsgroups/talk.religion.misc/83758
probability- 1.0
--------------------------------------------------------------------------------------------------
Folder in which the test data lie- talk.religion.misc
 Test data train_classify calculation-
path for the document- /Users/dikshasharma/Desktop/Machine Learning/20_newsgroups/talk.religion.misc/83764
probability- 0.9999999999999998
--------------------------------------------------------------------------------------------------
Folder in which the test data lie- talk.politics.misc
 Test data train_classify calculation-
path for the document- /Users/dikshasharma/Desktop/Machine Learning/20_newsgroups/talk.religion.misc/83798
probability- 1.0
--------------------------------------------------------------------------------------------------
Folder in which the test data lie- talk.religion.misc
 Test data train_classify calculation-
path for the document- /Users/dikshasharma/Desktop/Machine Learning/20_newsgroups/talk.religion.misc/83459
probability- 1.0
--------------------------------------------------------------------------------------------------
Folder in which the test data lie- talk.politics.misc
 Test data train_classify calculation-
path for the document- /Users/dikshasharma/Desktop/Machine Learning/20_newsgroups/talk.religion.misc/83586
probability- 1.0
--------------------------------------------------------------------------------------------------
Folder in which the test data lie- talk.religion.misc
```