**Abstract:**

The categorization of data or text can be done through machine learning classifiers. This project discusses about a supervised learning model of Naïve Bayes theorem that approximates the category in which the data most probably falls under based on controlled or trained data which is already defined to a particular category. Based on this data, a filter is created that assigns probability of each word and assigns a category to it.

**Introduction:**

Data from various sensors incorporated with powerful machine learning algorithms and domain knowledge led to many practical oriented inventions like the search engines, barcode scanners, and speech recognition by Alexa and Siri.

Now the problem is, “Can we build a supervised learning model that actively learns from trained data which will later automatically predict the category of a text file.”

In the month of April 2020 only, an average of 18 million COVID-19 spam emails were sent per day via Gmail, according to Google. The machine learning model by Google continued to block 99.9% of spam, but that still meant 258,000 spam emails regarding COVID-19 still landed in people’s mailboxes.  
Another practical application would be news article categorization. But there is a problem, often news articles fall under multiple topic labels.   
For Example: An ownership transfer of a football team might be labelled under Sports and Business.

How do we as humans accurately classify an unknown content of an article and be fairly confident about it. Our goal is to cluster the way human interprets data and categorize it.

Therefore, a need arises for us to uncover patterns in news categorization for specific applications and research.

**Packages Imported:**

The packages are used only for i/o functions.

* os
* operator
* time

**Dataset and Pre-process of Data:**

The 20 newsgroups dataset comprises around 18000 newsgroups posts on 20 topics split in two subsets. The data is then pre-processed by removing the punctuations and stop-words and converting it to lowercase. The data is split into Training and Testing at 50-50 ratio.

**Code Implementation:**

* We extract and read data from the system for the analysis. Data is preprocessed as above.
* Data is split in the ratio of 50-50.
* The word count is calculated and the number of times the word appeared in each of the category is measured and a dictionary is created based on this data.
* Probabilities of each class and each word in each class is calculated.
* Based on the true positives, the number of maximum probabilities of a word in a category is computed and is divided by the total population to calculate the accuracy of the model.
* The execution time for training, testing and overall is calculated.

**Result:**

Accuracy achieved is 82.37%

**Room for improvement:**

* Laplace smoothing can be incorporated at a rate of 0.0001
* Assigning TF-IDF values to words in a category.
* For large datasets, we can remove samples from over-represented classes.