**StumbleUpon Evergreen Classification Problem**

Starting the problem with importing the data from kaggle directly to Colab notebook.

The data set had various columns like boilerplate, urls, ids, ratio of error rates , links , and other stuff as well, but by seeing the dataset , i found out that as our goal is to predict the sentiment or thinking of the user by which he or she would decided to mark it as evergreen or not. So,We just required the content of the page which we get via the boilerplate columns which was enough for us to get started.

**Data Preprocessing and Visualization**

Next , first started to analyze the data through some visualization using word cloud and barplots

In this I found that data contains unnecessary words like recipes, body, url words occurring very frequently so first tried to remove these words.

Next I moved on with further preprocessing the data I removed useless stopwords, as well as punctuation and symbols if it contains.

After that , I used lemmatization techniques to capture the root word from each words in the dataset which makes prediction easy.

**Preparing the dataset**

Splitted the dataset into a training and validation set (80% and 20% ) respectively.

Applying tf-idf vectorizer and getting the data ready to feed to the model.

**Model Building and Prediction**

I tested my dataset on two models Support Vector Machines and Multinomial Naive Bayes

Kernel used in the SVM model was ‘Linear’. And all the other properties were set to default.

**Conclusion**

The SVM model shows us most promising results with the following quantities as follows:

* "Evergreen" value will be the main parameter in calculating Recall and Precision, since we will analyze whether the site is worth marking it or not.
* From the result above, we get the value of **Accuracy** **around 0.82** which means that we have around **82%** level of confidence that the model was accurate.
* And we have the **Recall value around 0.77** which means that we have **77%** "Evergreen" review which predicted correctly.
* And we have the **Precision value around 0.85** which means that we have **85%** "Evergreen" review from predicted positive value which actually positive