

Toxic Comment Classification using LSTMs

Submitted by:

HARSHVARDHAN PALAWAT

**ACKNOWLEDGMENT**

I want to extend my sincere regards to the below mentioned sources and references who helped me a lot in completion of my Project:

Team FlipRobo

Team DataTrained

scikit-learn official documentation

<https://scikit-learn.org/stable/>

geeksforgeeks

https://www.geeksforgeeks.org/

programiz

[https://www.programiz.com](https://www.programiz.com/)

Machine Learning Mastery

<https://machinelearningmastery.com/>

Medium

[https://www.medium.com](https://www.programiz.com/)

**INTRODUCTION**

* Business Problem Framing

Social media has become an essential part of everyone's life now these days. Like any other technology, it also has its own curse. Some people have made this a habit to hide behind a screen and throw insults at others. Toxic comments has become an integral part of what was supposed to connect people across the globe. Sometimes, such hatred can turn into disasters. Therefore, it is utmost necessary to filter through millions and billions of these comments and remove the toxic ones.

* Conceptual Background of the Domain Problem

The domain of this project is Natural Language Processing. As all of us has witnessed it at some point of time, we can classify comments after reading them as toxic or non-toxic according to the language and words used. But, it is practically impossible to go through each and every comment. Therefore, we can leverage the power of Machine Learning to do the task for us. Here, we are given with labelled data classified into 6 categories. We can convert the comments into mathematical vectors using concepts like Bag of Words, TF-IDF or Word2Vec etc so that the problem can be portrayed as a Machine Learning Classification task.

* Review of Literature

As the problem at hand is related to Natural Language Processing, it is very important to study how can human laguage be converted into mathematical form that machines could understand. The word embeddings is the most important part of this project. We did a lot of research on methods of tokenization such as Bag of Words, TF-IDF, Word2Vec and the comparison between listed methods. Also, a deep research on how state of the art techniques (like LSTMs) can leverage the sequence information present in text was conducted.

* Motivation for the Problem Undertaken

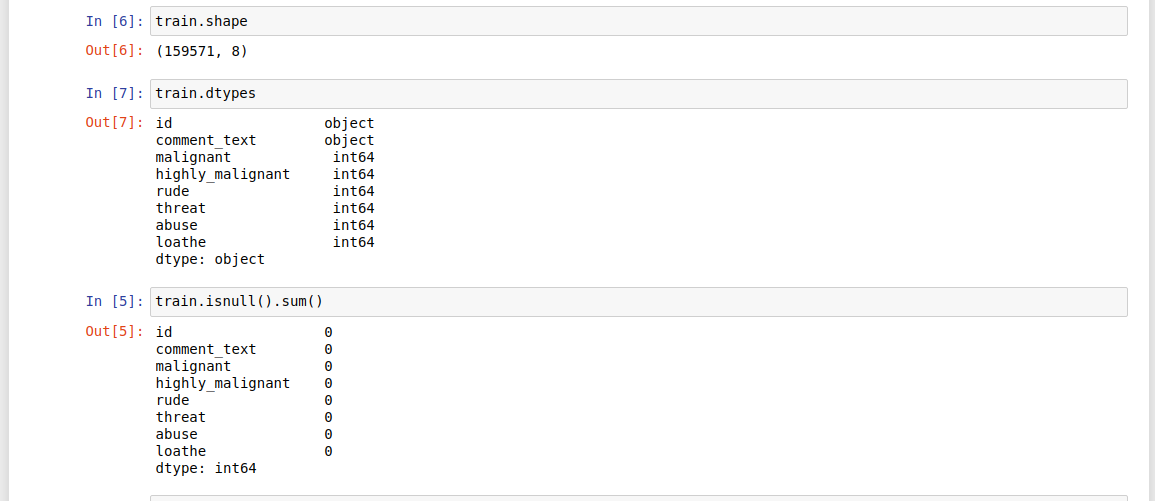
A lot of times, identity hatred, insults and threats on social media has very deep impact on the victim's life and may turn into a tragedy. There are a lot of reports of suicide becuase of such toxicity towards a person or a group of persons. Social Media was curated to help us connect with people across the globe and spread love, not hatred. A healthy criticism is always welcome, but sometimes people go far beyond that. The motive of this project is to take a step towards building systems for social media platforms that could identify such extremely harsh comments and automatically remove them leveraging the power of Machine Learning.

**Analytical Problem Framing**

* Mathematical/ Analytical Modeling of the Problem

First, we used mathematical techniques like Bag of Words, TF-IDF and Word2Vec to convert comment text into numerical vectors because we can not leverage text features directly in machine learning. Machine learning models like Multinomial Naive Bayes, Logistic Regression were used because they are good in dealing with high-dimensional data. Also, state-of-the-art LSTMs were used so that we can leverage the sequence information present in comments.

* Data Sources and their formats
* The client has provided data in two seperate CSV files namely train.csv and test.csv for training and testing respectively. There are 8 columns in train while only 2 columns in test file. The dtypes of each column are given in the snapshot of data.



* Data Preprocessing Done

There are no null values in the data. We used TF-IDF to convert the comment text into a 2000-dimensional vector. Also, we engineered a few features such as length of comments, the total number of words, the number of stop words etc to add more value to the data.

* Data Inputs- Logic- Output Relationships

This a classification task having 6 labels. Thus, we have decided to break the problem in 6 seperate classification tasks. The data is highly imbalanced. We have used Logistic Regression, Multinomial Naive bayes and Bi-directional LSTMs to logic the input-output relationships.

* State the set of assumptions (if any) related to the problem under consideration
* We have used the multinomial naive bayes for modelling and the assumptions associated with it are the independence of features and multinomial distribution of the data.
* Hardware and Software Requirements and Tools Used

The size of data is very small, therefore any system running on Windows 7 or higher, Mac or Linux based operating systems with 8 GB of RAM is more than sufficient for the given task. We can use any Python IDE or Jupyter notebooks or Google Colab for modelling.

Below is the list of tools used for the task:

sklearn for model building,

pandas for reading and manipulation of data,

numpy for numerical operations,

matplotlib and seaborn for data visualization

NLTK (TF-IDF) and gensim (Word2Vec) for tokenization and text pre-processing

tensorflow and keras for modelling the LSTM

joblib for saving the model

**Model/s Development and Evaluation**

* Identification of possible problem-solving approaches (methods)

We have used Bag of Words, TF-IDF and Word2Vec for tokenization, Logistic Regression, Multinomial Naive Bayes and Bi-directional LSTMs for modeling.

* Testing of Identified Approaches (Algorithms)

We used:

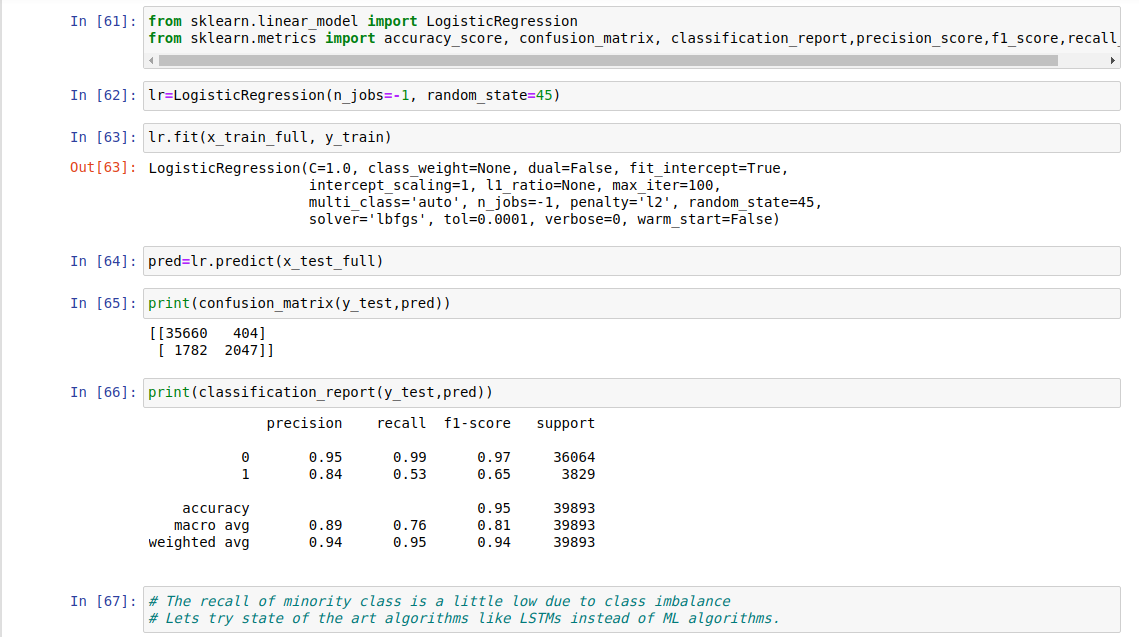
Logistic Regression,

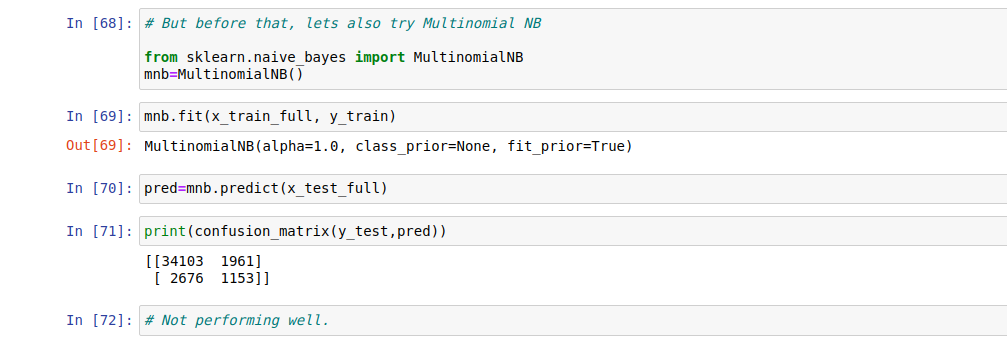
Multinomial Naive Bayes,

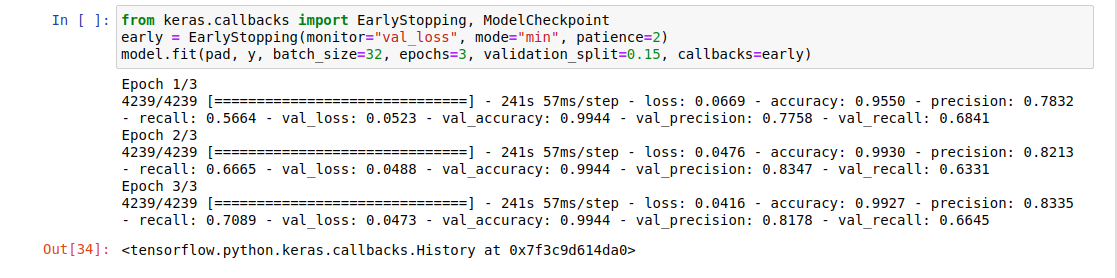
Bi-directional LSTMs

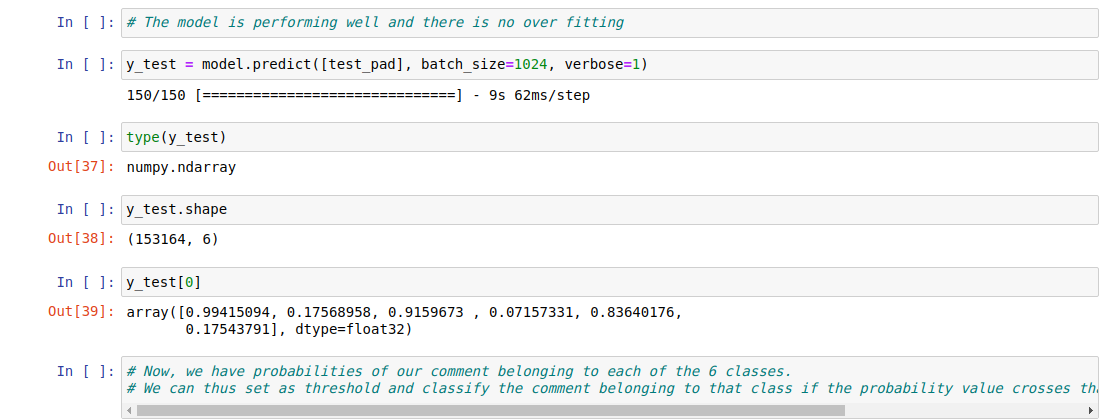
* Run and Evaluate selected models

Below are the snapshots of the code used for modeling:



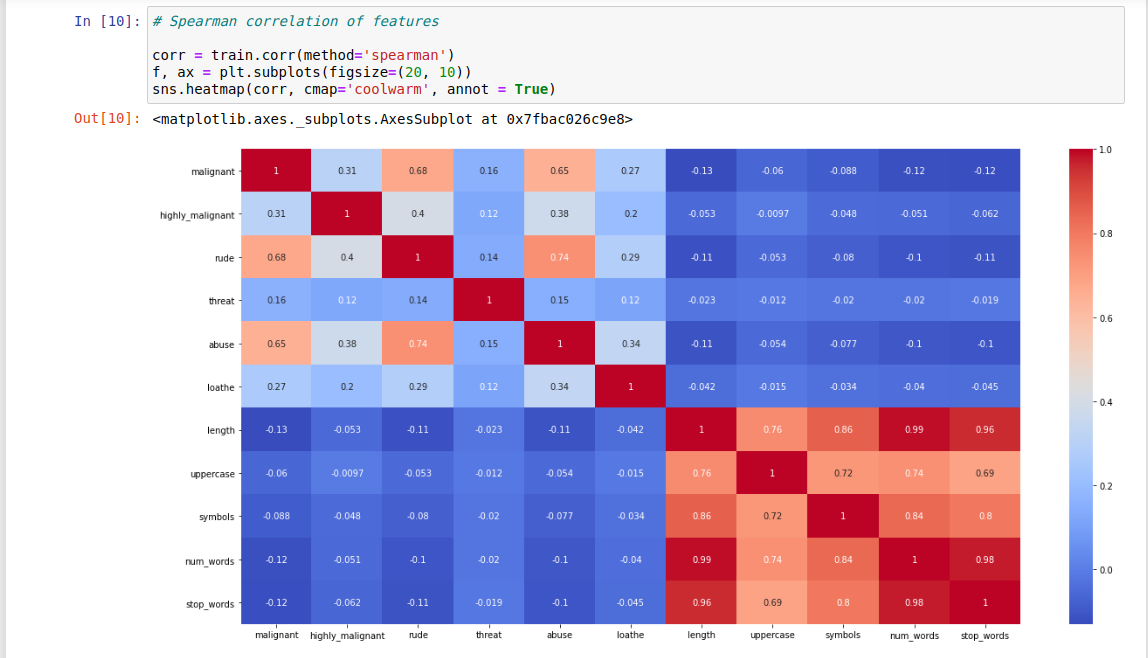


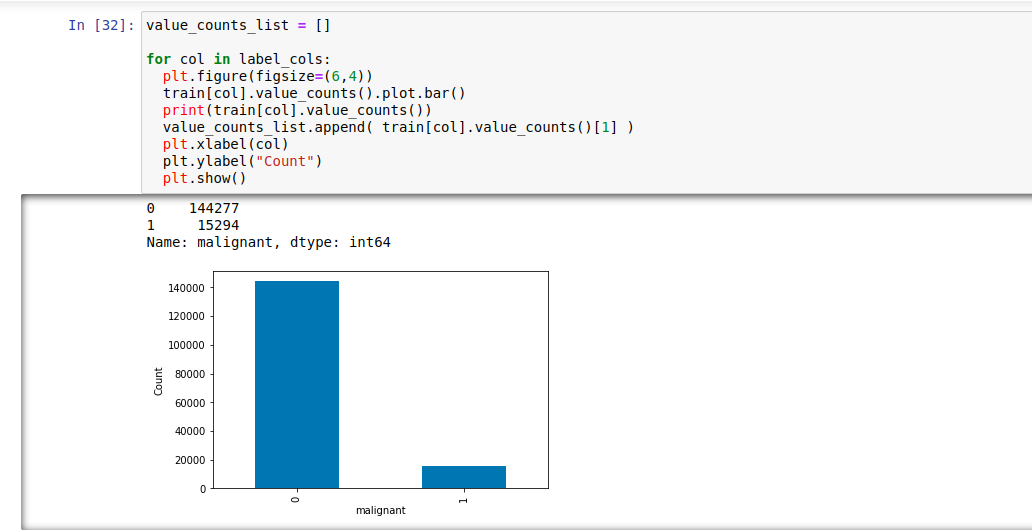


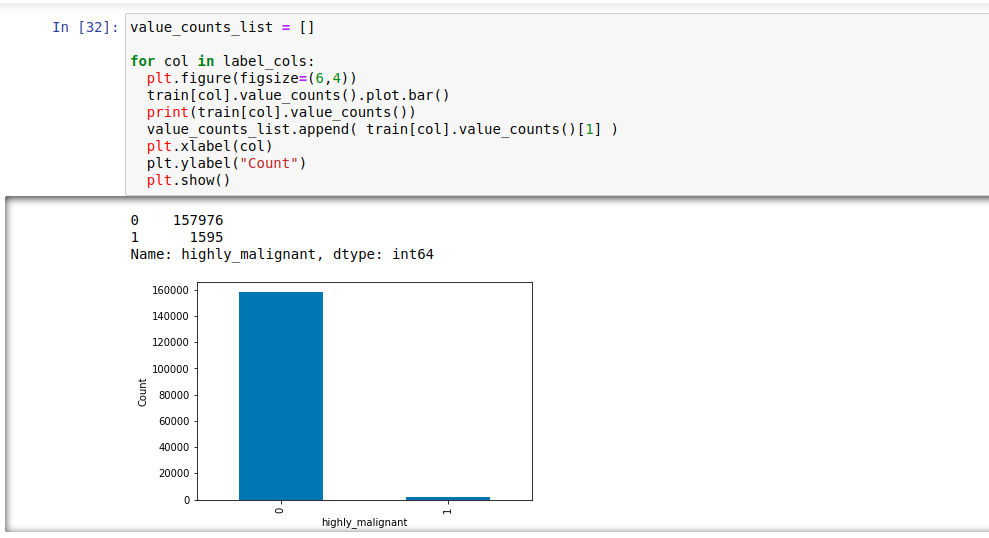


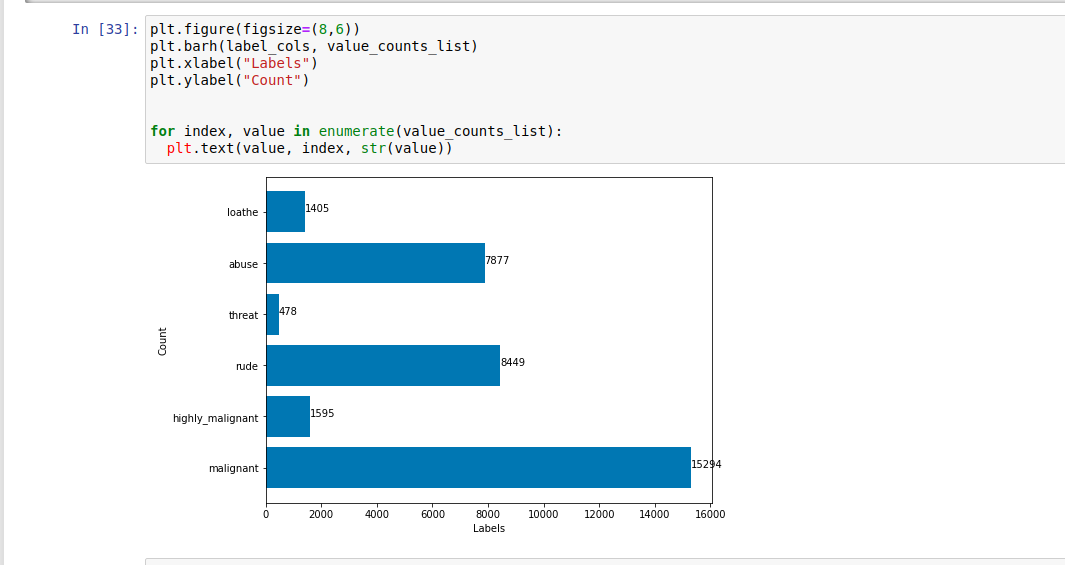
* Key Metrics for success in solving problem under consideration
* What were the key metrics used along with justification for using it? You may also include statistical metrics used if any. Since the data is imbalanced, classification\_matrix is an important indicator of the performance. We have used precision and recall as key metrics.
* Visualizations

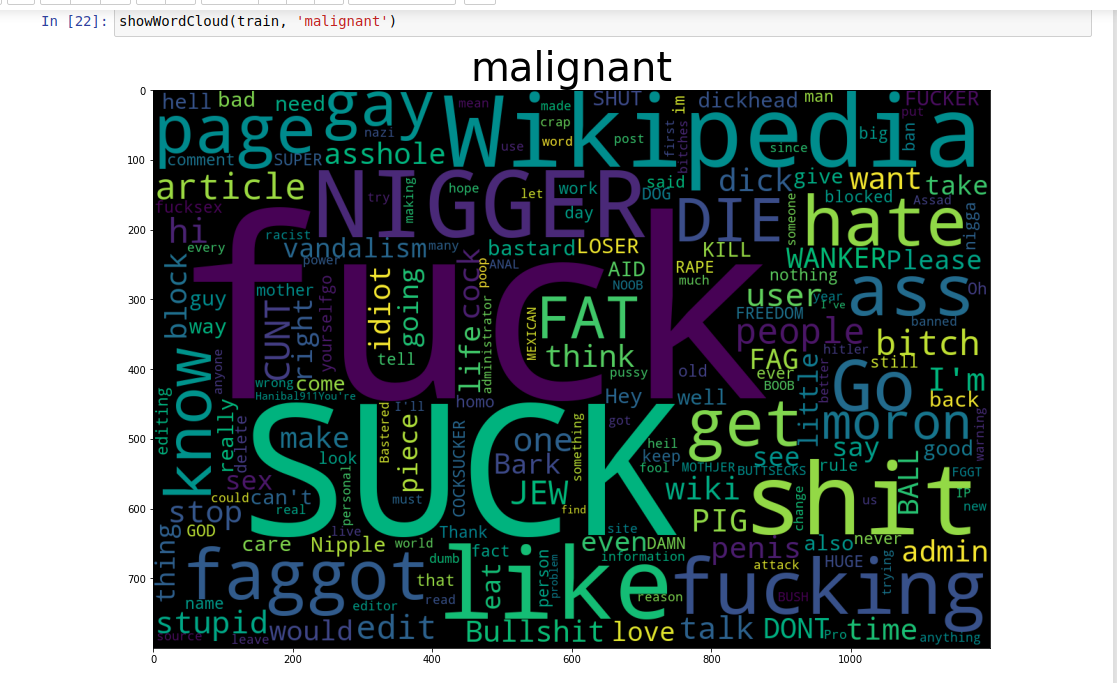
We have made the following plots:

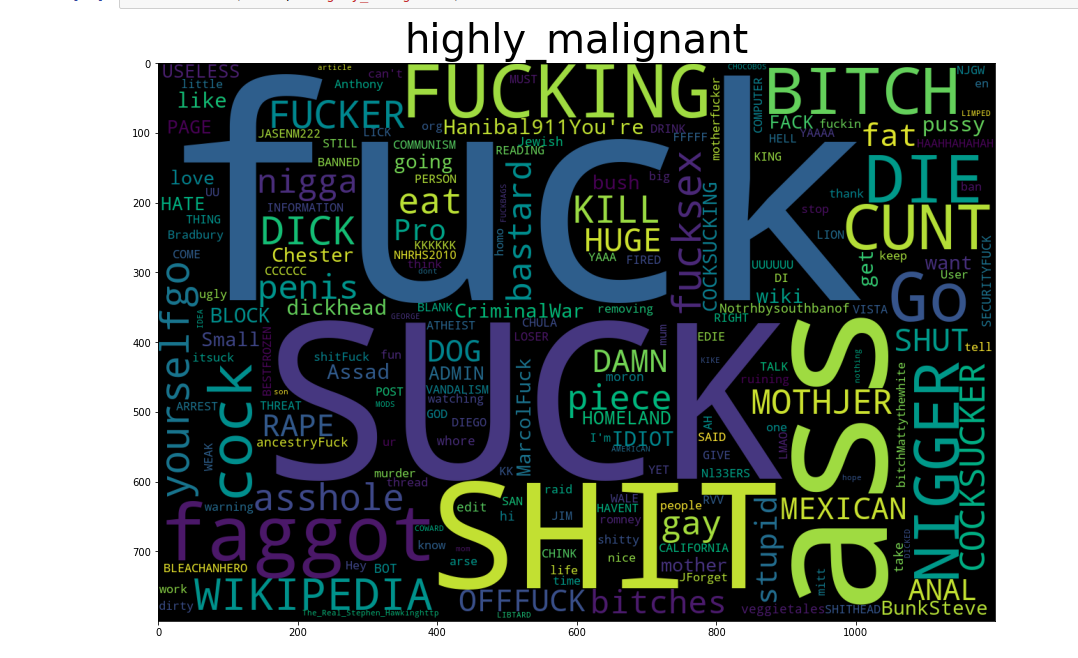




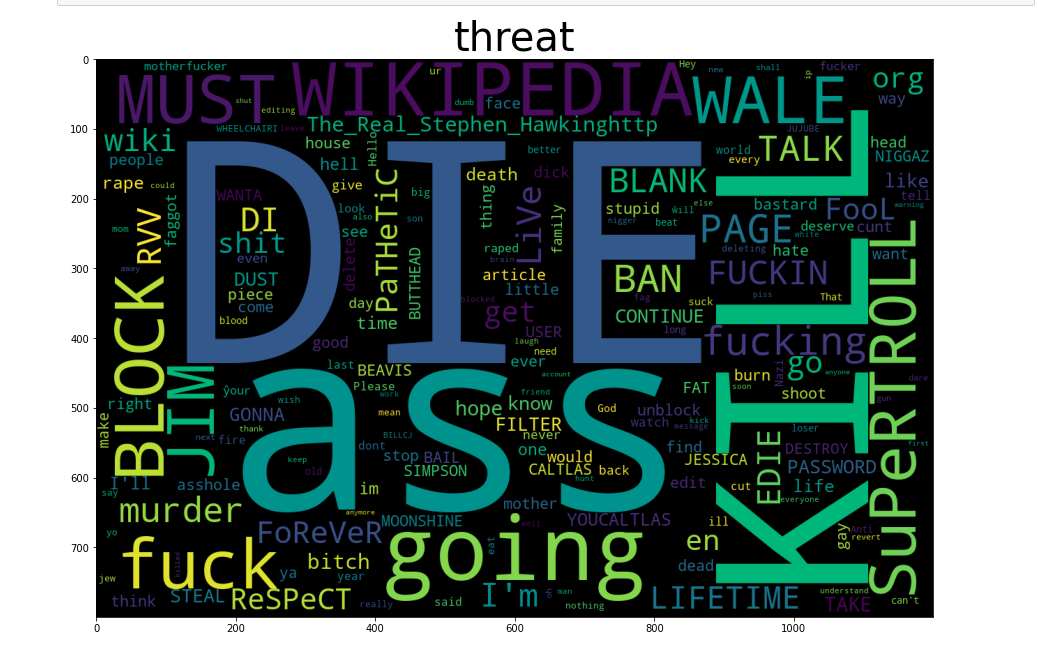




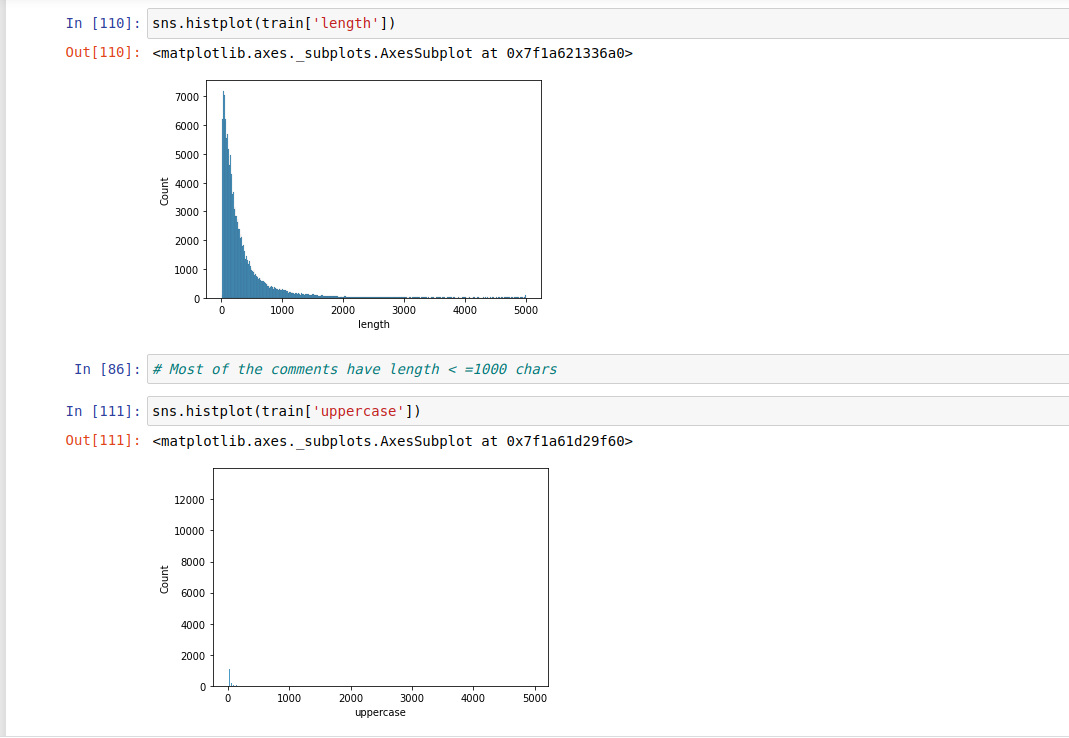


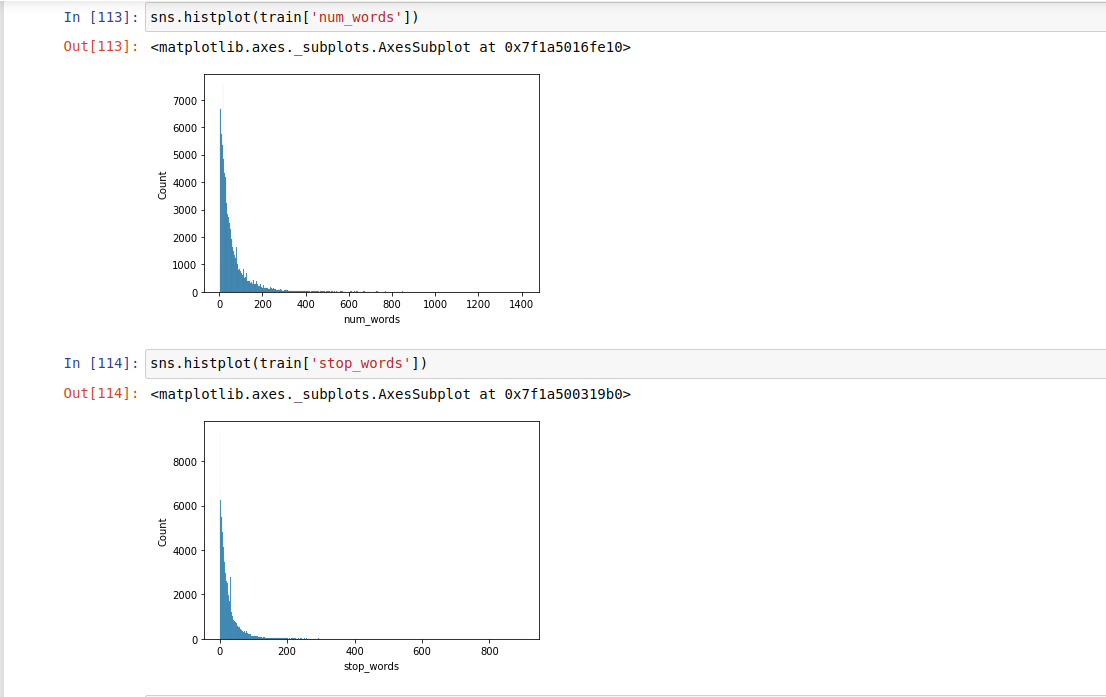












* Interpretation of the Results
* We have plotted wordclouds of various labels to understand the most occuring words in each class. Also, we have drawn plots to understand the distribution of each label class in the data. It shows that most of the comments are unlabelled and data is highly imbalanced with respect to classes. We have also drawn histograms to see the distribution of number of words, stop words etc in each comment.

**CONCLUSION**

* Key Findings and Conclusions of the Study

We have learnt that LSTMs work better than classical machine learning algorithms as they leverage the sequential information present in the data.

* Learning Outcomes of the Study in respect of Data Science

We learnt the distribution of various key words for each class of comments present in our data. We also learnt that LSTMs work better than classical machine learning algorithms as they leverage the sequential information present in the data. Due to the large size of data, we issued a lot of compute challenges. The colab notebook crashed due to memory issues while using word2vec for vectorization. We then used TF-IDF with only 2000 word features to avoid the issue.

* Limitations of this work and Scope for Future Work

We can use state of the art algorithms like Word2Vec or BERT to vectorize the data in future to improve the results. Also, we can collect more data to reduce the high imbalance in the data.