FINAL PROJECT REPORT

TOPIC: TOXIC COMMENT CHALLENGE

CS273P MACHINE LEARNING, SPRING 2019 (PROF. Xiaohui Xie)

Team Name: POTLUCK
Team Member : Mehul Kothari (53763452)

INTRODUCTION

Toxic Comment Challenge is a competition held by Conversation AI team, a research initiative founded by Jigsaw and Google. The aim of this project is to identify and classify toxic comments for eg on social media like Facebook, YouTube etc. So the challenge was to **build multi-headed model** that's capable of detecting different types of toxicity like threats, obscenity, insults and identity based hate.

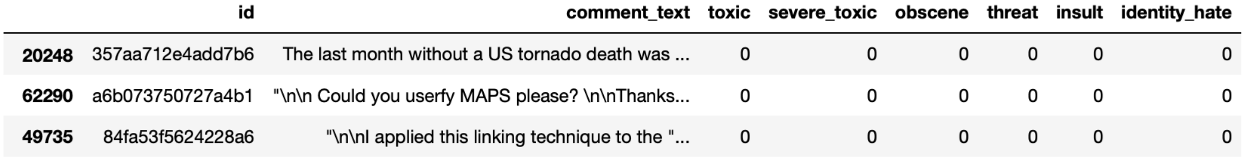
DATASET

Dataset of comments from Wikipedia's talk page edits. There are six types of toxicity toxic, severe toxic, obscene, threat, insult and identity hate.

DATA EXPLORATION

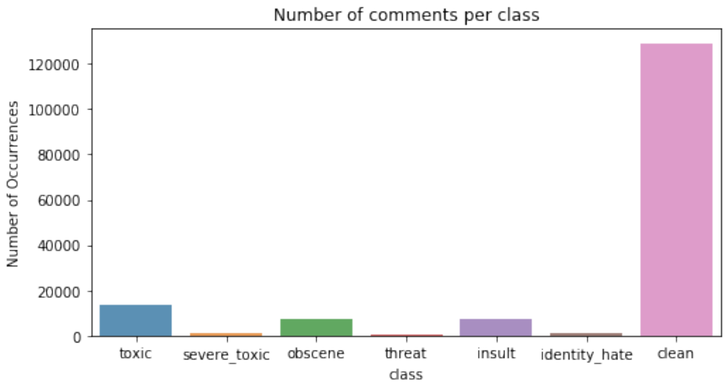
The first task in any machine learning project is to explore the data and understand the way data has been arranged. In this part we would look at some interesting conclusions that we are able to draw from the dataset given to us.

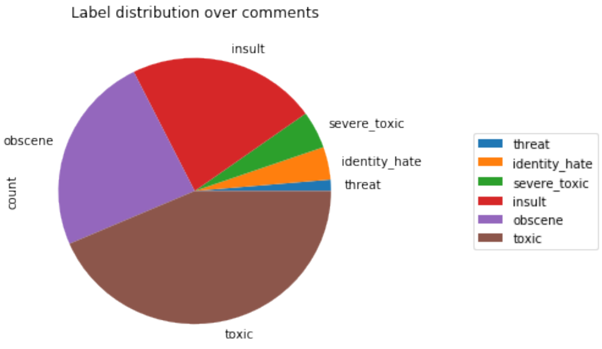
- The training set has around 143645 sample with 8 columns. One of the columns contain 'raw comment' and the other columns are 6 different types of toxic comments as shown in the figure.



	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
20248	357aa712e4add7b6	The last month without a US tornado death was ...	0	0	0	0	0	0
62290	a6b073750727a4b1	"\n\n Could you userfy MAPS please? \n\nThanks...	0	0	0	0	0	0
49735	84fa53f5624228a6	"\n\nI applied this linking technique to the "...	0	0	0	0	0	0

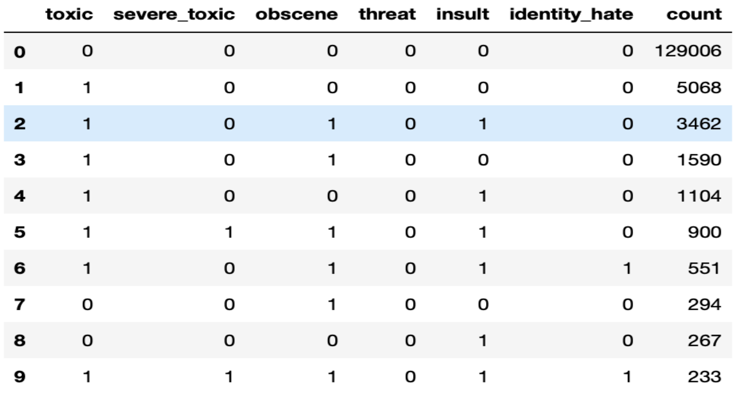
- There is an **imbalance** between the clean comments and toxic comments. As we can see from the image as well. Lets try to explore on this basis.
  - Total clean comments : 129006
  - Total tags : 31724





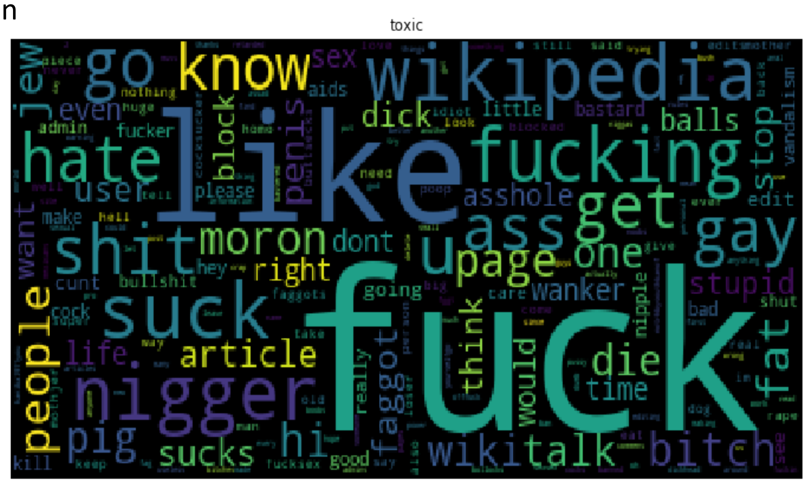
- We can see from the graphs above after clean comments **toxic, obscene and insult** are the most number of comments.

- Each comment can have multiple labels associated with it. Next I have tried to find the correlation between these labels and how many of them repeat and which pair is the most frequent.

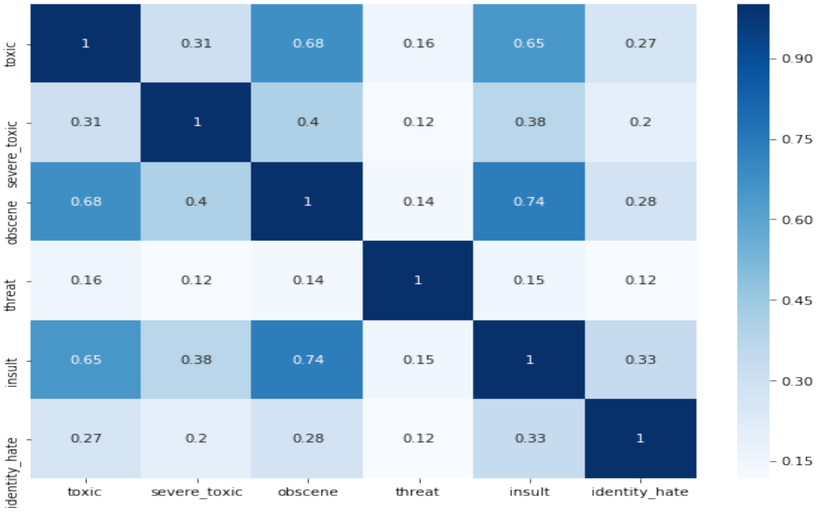


	toxic	severe_toxic	obscene	threat	insult	identity_hate	count
0	0	0	0	0	0	0	129006
1	1	0	0	0	0	0	5068
2	1	0	1	0	1	0	3462
3	1	0	1	0	0	0	1590
4	1	0	0	0	1	0	1104
5	1	1	1	0	1	0	900
6	1	0	1	0	1	1	551
7	0	0	1	0	0	0	294
8	0	0	0	0	1	0	267
9	1	1	1	0	1	1	233

- We can observe that 'none' has the most number of comments as expected. Followed by only Toxic with 5068. The combination of Toxic, obscene and insult occur 3462 times. **We can see that labels do overlap**.
- Next, I tried visualizing some common words which occur in comments having labeled as toxic, threats etc. Here is a word cloud for toxic.



- Finally I have tried plotting a **correlation matrix**, which shows the value of correlation between two labels.



DATA PRE-PROCESSING

- **Data Pre-processing is the second step in the process**. In this step I have tried to clean the dataset and **vectorized it using TF-IDF vectorizer.**
- I have tried preprocessing the comments by
    - **Changing it to lower case**
    - **Replacing \\n with “ ”**
    - **Converting sentences to tokens**
    - **Removal of stop words** – Stop words such as a,an,the etc were removed from the comments.
    - **Stemming** - stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form. I used **Porter Stemmer** for this purpose.
    - **Lemmatizing** - usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma. I am using **WordNet** to do the process.

- After all **the preprocessing I realized that it was not very impactful**. After some research, I came across a paper *“Is preprocess really worth your time for toxic comment classification”* – Fahim Mohammad , Intel Corporation. In this paper Fahim showed 35 transformation to the data and concluded that preprocessing does not affect accuracy a lot for toxic comment classification.
- The biggest revelation was to remove “non-alpha- words” which resulted in a big decrease in accuracy. The reason is, comments people write on social media contains all sorts of non-alpha characters for eg b!t(h , @ss etc.

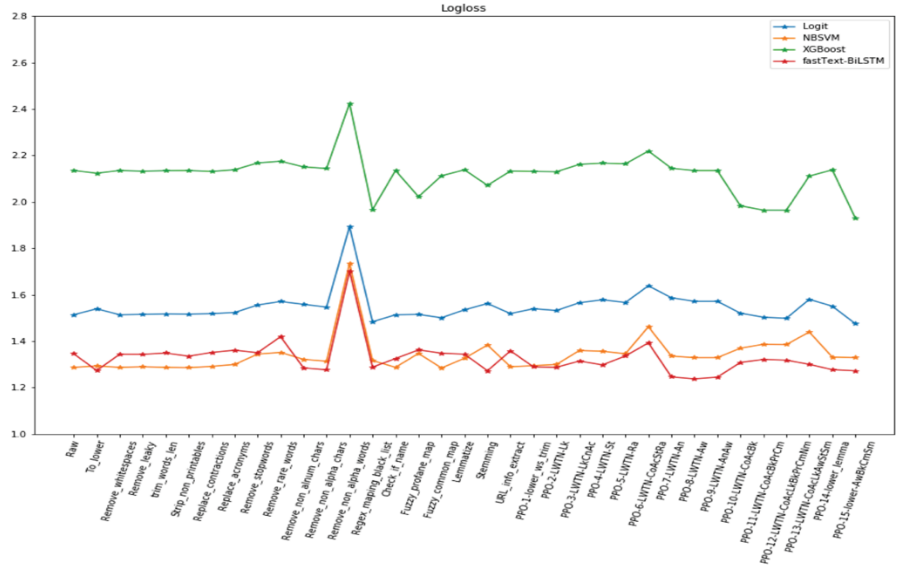


Fig. 3: Log loss plot for all four models on different transformations.

- It is visible from the graph that there is not a lot of difference between preprocessed data and raw data(First value on graph)

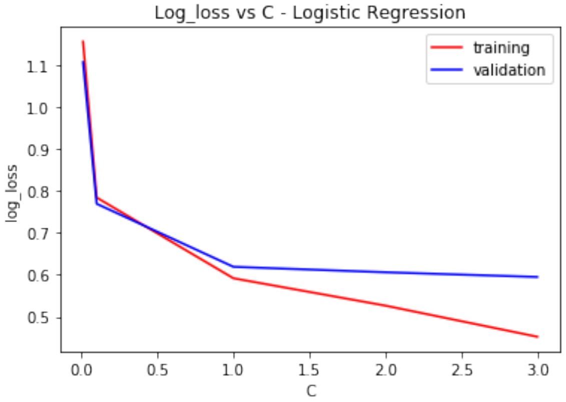
- Since the comments are in string format, I have converted them into vectors using TF-IDF vectorization technique. This is an acronym than stands for “Term Frequency – Inverse Document” Frequency which are the components of the resulting scores assigned to each word.
    - Term Frequency: This summarizes how often a given word appears within a document.
    - Inverse Document Frequency: This downscales words that appear a lot across documents.
- Without going into the math, TF-IDF are word frequency scores that try to highlight words that are more interesting, e.g. frequent in a document but not across documents.
- The other alternatives would be to use **word embeddings and the most popular ones are Glove, WordVec and FastText**

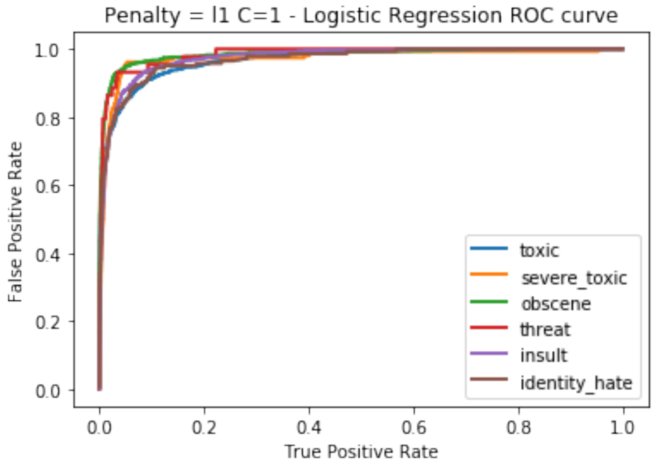
MODELS – TRAINING AND EVALUATION

I have experimented on four different algorithms. All the algorithms used are quite popular for multi-label classification. In most of the algorithms I have tuned the hyperparameters to get the best model possible. The **evaluation metric used is AUC** which is probably the best metric for multi-label classification. I have tuned hyper-parameters on the basis of validation dataset and calculated Final AUC using the Test-Data set.I even got an opportunity to find out that even log-loss give a pretty good picture about the model and I have used **log_loss** metric for hyperparameter tuning.

- **Logistic Regression** : I have used One vs Rest strategy. One-to-rest” strategy, could build multiple independent classifiers and, for an unseen instance, choose the class for which the confidence is maximized. The main assumption here is that the labels are *mutually exclusive*. You do not consider any underlying correlation between the classes in this method. For instance, it is more like asking simple questions, say, “*is the comment toxic or not*”, “*is the comment threatening or not?*”, etc. Also there might be an extensive case of overfitting here, since most of the comments are unlabeled, i,e., most of the comments are clean comments.

    I tweaked the hyper-parameters a little ie used l1 and l2 regularization and changed ‘C’ – inverse of regularization strength. **The best result was obtained using l1-regularisation and C=1.** The graph shows that model starts overfitting for increasing values of C.

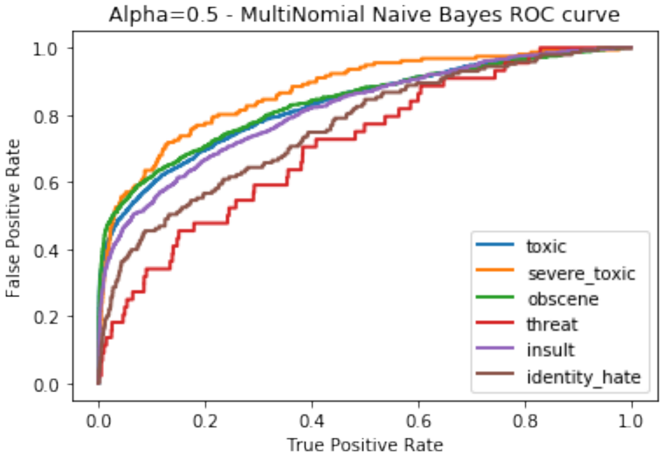




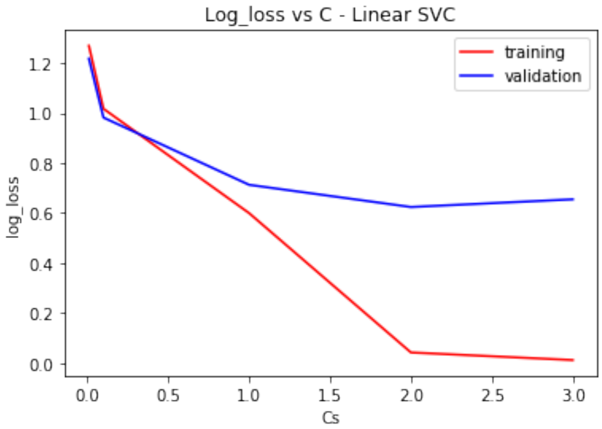
Mean AUC : 0.9755

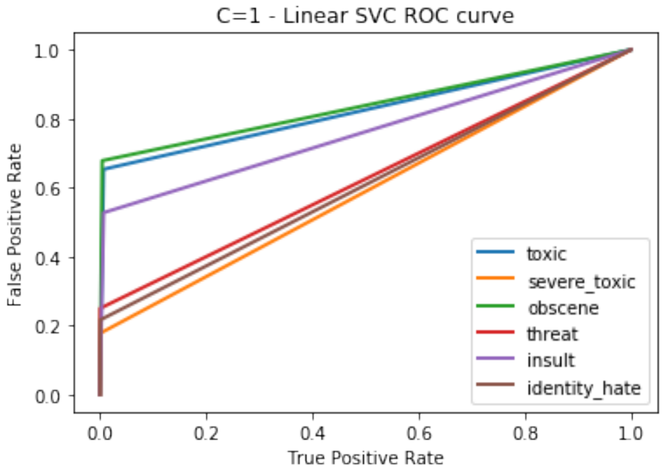
- **Multinomial Naïve Bayes** : This model is suitable for classification with discrete features. Alpha used is the additive smoothing Parameter. The best result was found for alpha=.5.

  Mean AUC : 0.8036 for alpha=0.5



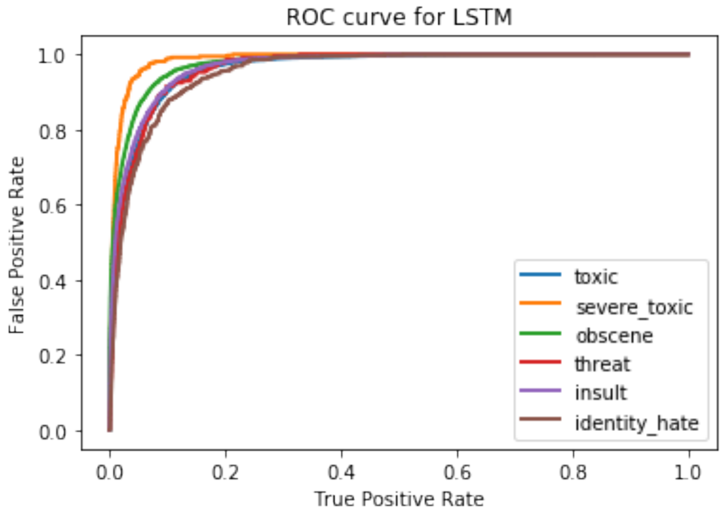
- **Linear SVC** :
  C is the penalty parameter used for Linear SVC. As we can observe from graph 1 as C goes on increasing there is overfitting. The optimal value for C is 1.0. Linear SVC is my base model. **Mean AUC : 0.7069**

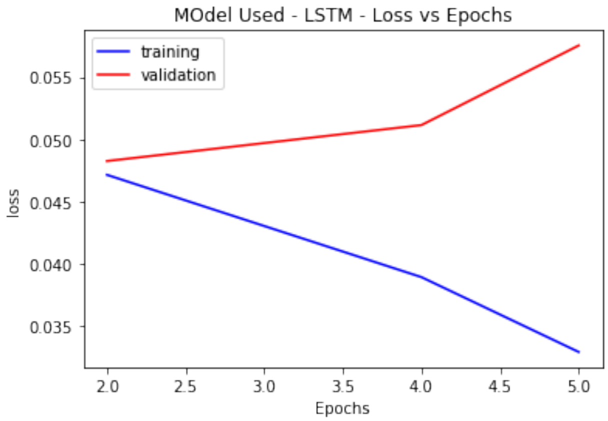




- **LSTM** :

In this model I have tuned the number of epochs and noticed that more than 2 epochs cause overfitting to the model. LSTM produces an output that has a dimension of 60 and returns the whole unrolled sequence of results. I have used Global max pooling layer to reduce the dimensionality of data. I have even used dropout layers as weel as Relu in the hidden layers and Sigmoid in the activation layer





Mean AUC : 0.97 for 2 epochs:

Conclusion and Future Work:

The accuracy provided by all models is quite good but they could definitely be improved using more hyperparameter tuning and more generalized dataset. The dataset has a lot of imbalance ie the ratio of clean to toxic is really high which somewhat affects the model as it tends to train mostly on clean data. One important conclusion from the dataset was how preprocessing did not have a lot on impact on the final model and raw data produced better results. Neural Network (LSTM) and Logistic regression had the best performance. I even noticed how hyperparameter tuning is affecting model performance and the model is overfitting after a point of time. The reason for this overfitting is dataset imbalance.

In the future I would like to implement FastText word embeddings with LSTM. I would like to try out RNN and bi-directional LSTM. I would love to stack and work on ensemble models and see how stacking of different models help in improving accuracy.

APPENDIX :
Jupyter Notebook – “Toxic_Comment_project.ipynb”

This file only needs “train.csv” and “test.csv” to be in the same folder.

The libraries to Install :

- Pandas
- Numpy
- Data_loader
- matplotlib.pyplot
- matplotlib.gridspec
- seaborn
- nltk
- sklearn
- wordcloud
- pillow

REFERENCES

- https://towardsdatascience.com/journey-to-the-center-of-multi-label-classification-384c40229bff
- https://medium.com/@zake7749/top-1-solution-to-toxic-comment-classification-challenge-ea28dbe75054
- https://csce.ucmss.com/cr/books/2018/LFS/CSREA2018/ICA4290.pdf
- http://scikit-learn.github.io/stable
- https://medium.com/@nupurbaghel/toxic-comment-classification-f6e075c3487a
- https://arxiv.org/abs/1809.07572