CSL2050: Pattern Recognition and Machine Learning
Project: Toxic Comment Classification
Instructor: Dr. Richa Singh

Ajay Nirmal(B20AI002), Saksham Singh(B20AI064), Siddharth Soni(B20AI041)

May 2, 2022

1 Introduction

The aim of the project is to correctly predict and classify the toxic comments on social media. Toxic comments on social media can be simply reported and removed if they are detected. In the long run, this would allow people to engage with each other more effectively in an increasingly digitised world. This project uses Natural Language processing (NLP).

2 Data analysis

2.1 Dataset used

Dataset: Jigsaw Unintended Bias in Toxicity Classification. The dataset contains 1971916 unique labelled examples of comment text that have been labelled for toxic behavior.

2.2 Toxic Level Count (Visualising class balance)

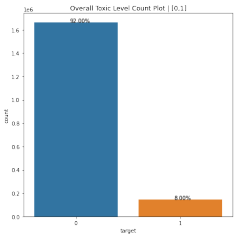
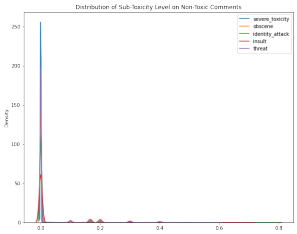


Figure 1: Toxic Level Count

2.3 Toxicity Based Features



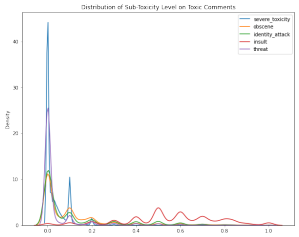


Figure 2: Sub-Toxicity Level on Non-Toxic and Toxic Comments

2.4 Correlation in Comment types

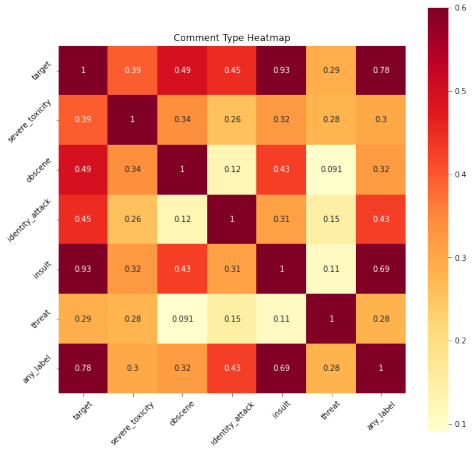


Figure 3: Heatmap for comment types

It is clearly visible from figure 3 that all the comment types are independent. Only target and insult show correlation.

2.5 Comment type counts

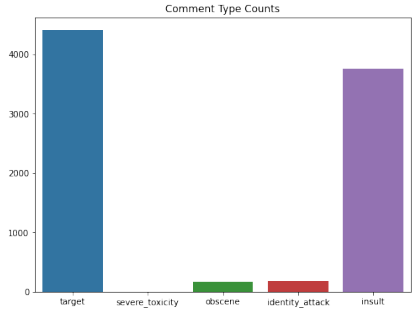


Figure 4: Count for comment types

2.6 Visualizing word counts

2.6.1 Word plot: clean comments only

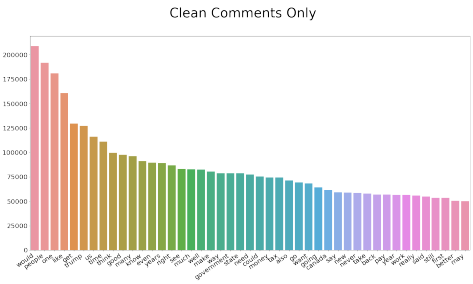


Figure 5: Clean comments only.

2.6.2 Word plots: Toxicity classes

Figure 6: Toxicity and severe-toxic.

Figure 7: obscene and threat.

Figure 8: insult and identity-hate.

3 Approach

A classification task with large dimensionality data is a natural language processing challenge. We'll vectorize the data and put many classification methods to the test.

3.1 Goals

- Predicting Overall Toxicity Level of a Comment
- Predicting Class Wise Toxicity Level of a Comment
- Build an API to classify toxic comments from text given by the user and also predicting the toxicity of any public twitter account.

3.2 Tasks

3.2.1 Binary classification

The binary classification goal was straightforward: given an input comment, our model had to determine whether or not the statement was harmful or non-toxic.

3.2.2 Multi-label classification

In multi-label classification: given an input comment, our model had to determine whether or not the statement was harmful or non-toxic. If found toxic,then determine what kind of toxicity this comment is(severe-toxicity, obscene, threat, insult or identity-attack).

4 Experiments

4.1 Data pre-processing

Pre-processing or cleaning of data leads to a more efficient model and higher accuracy. We have created a pipeline for the same including various functions which are listed below:

- Removal of html tags and URLs
- Removal of white space
- Lowercase all sentences: www.harvard.edu:80
- Contractions: isn't :- is not
- Replace starred toxic words: "sh*t": "shit
- Removal of punctuations: '!()*+,-./:;?@[]'— '
- Remove Stop-Words: "a", "the", "is", "are"
- Lemmatization:

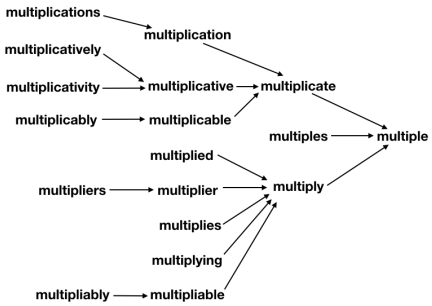


Figure 9: Example of Lemmatization.

4.2 Removing data imbalance

From the data analysis and figure 1 and figure 2, the high imbalance in the data is clearly visible. The data points in train data: 1804874.

It is necessary to remove this imbalance to improve the model. The following has been done for the same:

- Data Augmentation: We have class-wise upsampled the amount of data in the minority classes and also down sampled the majority class since the number of comments in the majority classes grew as a result of the multi-labeled comments.
- We have used Binary Relevance Method in which all the labels in the dataset are partitioned into single labels and each single label are performed as single label multi-class classification problem. This approch is valid since there is very less correlation between the sub-classes.

X	Y_1	Y_2	Y_3	Y_4
$x^{(1)}$	0	1	1	0
$x^{(2)}$	1	0	0	0
$x^{(3)}$	0	1	0	0
$x^{(4)}$	1	0	0	1
$x^{(5)}$	0	0	0	1

X	Y_1
$x^{(1)}$	0
$x^{(2)}$	1
$x^{(3)}$	0
$x^{(4)}$	1
$x^{(5)}$	0

X	Y_2
$x^{(1)}$	1
$x^{(2)}$	0
$x^{(3)}$	1
$x^{(4)}$	0
$x^{(5)}$	0

X	Y_3
$x^{(1)}$	1
$x^{(2)}$	0
$x^{(3)}$	0
$x^{(4)}$	0
$x^{(5)}$	0

X	Y_4
$x^{(1)}$	0
$x^{(2)}$	0
$x^{(3)}$	0
$x^{(4)}$	1
$x^{(5)}$	1

Figure 10: Binary Relevance Method

5 Evaluation metrics

We examined performance among the models for the binary classification task with word and character level granularity using accuracy, precision, recall, specificity, and F1-score. The confusion matrix obtained for each iteration of hyper parameter tweaking on the validation set was used to generate these measures. The confusion matrix summarises correct and incorrect predictions for each class using count values, which provides insight into the model's sorts of errors. In the event of imbalanced classes, the F1-score becomes a crucial metric for evaluating the classifier's performance.

Considering the multi-label classification task, accuracy measurement(label accuracy) reported label correctness across the entire train dataset.

6 Comparison of Models

6.1 Logistic Regression

It is a classification that serves to solve the binary classification problem. In NLP, logistic regression is the base- line supervised machine learning algorithm for classification.

6.2 Linear SVC

It performs well with a high number of data and uses a linear kernel function to perform classification.

6.3 SGD Classifier

It is similar to Linear SVC. It is an efficient approach to fit linear classifiers and regressors under convex loss functions.

6.4 LGBM Classifier

LightGBM is a fast and high performance gradient boosting framework based on decision tree algorithms.

Model	Accuracy	Precision	Recall	f1-score
Logistic Regression	0.84	0.77	0.81	91.4
Linear SVC	0.85	0.88	0.87	0.87
SGD Classifier	0.76	0.86	0.76	0.81
LGBM Classifier	0.77	0.76	0.83	0.79

Table 1: Comparison of models

The evaluation metrics on each of the model has been shown below in Table 1 in the form of the table. It is clearly visible that the metrics are better for the Logistic Regression model.

7 Results

7.1 Overall Quantitative Results

All the evaluation metrics have been calculated on the validation set. The values clearly show that the data imbalance in the dataset has been taken care of.

7.1.1 Binary Classification

In our binary classification job, Logistic Regression outperformed all other models, attaining the greatest accuracy, precision, and F1 scores. For this challenge, F1 scores did not differ significantly amongst models.

7.1.2 Multi-label classification

The comments are further also classified into classes of toxicity with their levels. All the values of log losses corresponding to labels are less. The corresponding values are for Logistic Regression model.

Labels	Neg log loss
toxicity	0.3638912698515414
severe-toxicity	0.6895414831086202
obscene	0.4307616328499047
threat	0.28885298039715335
insult	0.5801608055244084
sexual-explicit	0.40186228606213764

Table 2: Labels vs Neg log losses

7.2 Overall Qualitative Evaluation

Input	Judgement	Confidence	Tags
"I love you"	Non-toxic	0.08	None
"Sex is not a sacred act"	Toxic	0.96	Sexually explicit
"You suck"	Toxic	1.0	Obscene,Insult
"u r a piece of shiit!"	Toxic	0.96	Obscene
"I'll kill you"	Toxic	0.99	Severe-Toxicity, Threat

Table 3: Qualitative evaluation of Logistic Regression model.

By the above table, we can observe that our Logistic Regression model us able to fairly identify different forms of toxicity.

- The straightforward insults are identified with high confidence and tagged as insults.
- Our model is able to detect toxicity even in case of spelling mistakes and slang languagae..

8 Extra Effort: Application Programming Interface(API)

We have built an API as an extension of our project. API is capable of classifying the toxic comments from text entered by the user. The API also showcases the different levels of the toxic comment identified.

Further, API also correctly predicts whether a certain Twitter handle is toxic or not. We just have to enter the twitter account id and the API scraps the top 100 comments. The predictions are then made on the same. Screenshots of which have been shown below.

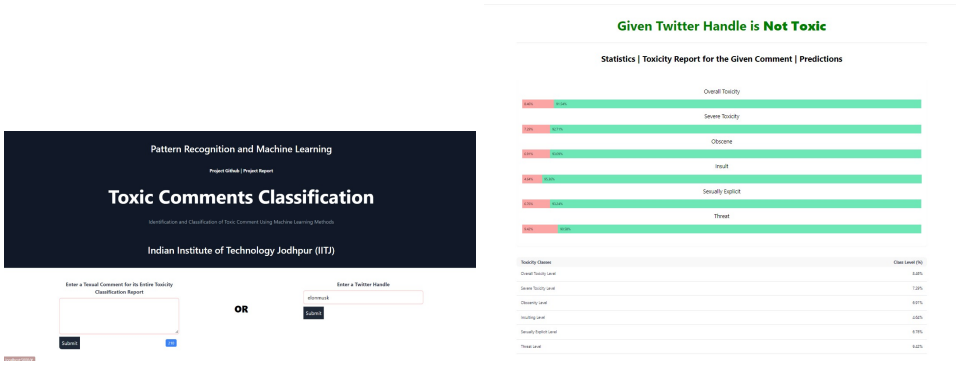


Figure 11: API interface

9 Conclusion

- The Logistic Regression model worked the best for our dataset and outperformed all the other models. This can be easily proven by the low log losses for each label shown in Table 1.
- Our model is able to detect toxicity and further levels of the same correctly on various type of inputs which can be verified through Table 3.

10 Contribution of each member

Ajay Nirmal

- Data pre-processing pipeline
- Hyper-parameterise ML models
- Report creation
- Implementing pipeline from data creation to ML models

Saksham Singh

- Data pre-processing pipeline
- Flask API
- Pipeline from testing models to final API

Siddharth Soni

- Data pre-processing pipeline
- Trying different ML pipelines
- Implementing Binary Relevance method

11 References

- https://towardsdatascience.com/tagged/tfidf-vectorizer.
- https://jayspeidell.github.io/portfolio/project05-toxic-comments/
- https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/reports/6837517.pdf
- https://medium.com/@nupurbaghel/toxic-comment-classification-f6e075c3487a