Toxic Comments Classification

Course: Advance Math of Machine Learning

Group Members: Shanka Attanayake, Andres Lojan Yepez, Leandra Lai, Princy Chahal, Reza

Date: Dec 5, 2021

TABLE OF CONTENTS Page

LIST OF FIGURES

LIST OF FIGURES AND TABLES

1.0 INTRODUCTION

The internet and social media are one of the best innovations in the past century. However, there are a lot of problem they created at the same time. One of the significant issues is abuse of comments which leads to cyberbully and destroy the way we communicate. This project will use a publicly available dataset that contains toxic and non-toxic comments and try to identify them at the end after we train the models.

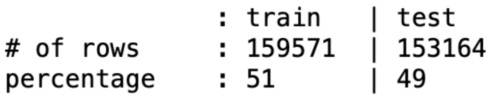
1.1 Problem statement

We want to identify if a comment is toxic or not on social media to enable a better and healthier communication environment.

1.2 Data

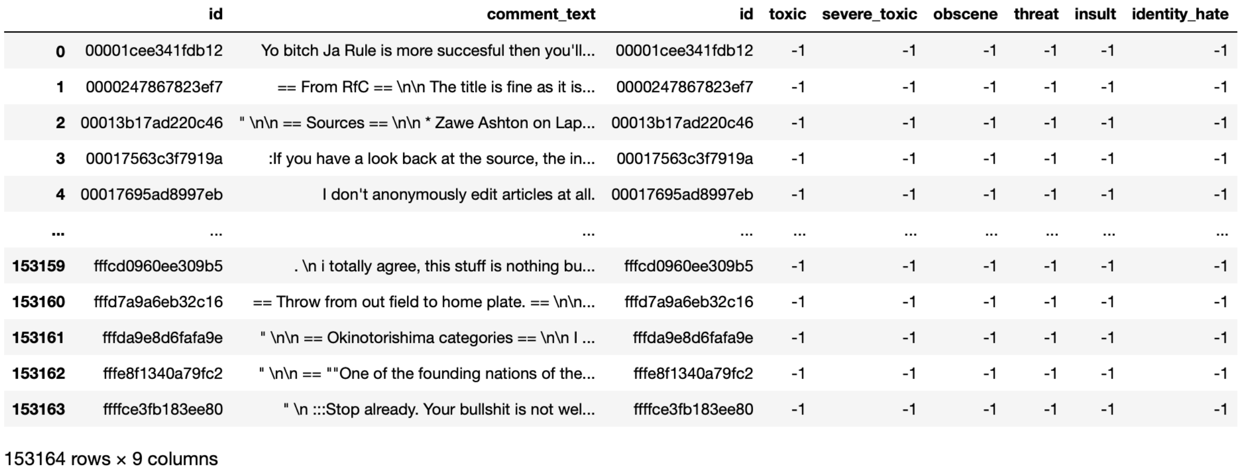
1.2.1 Dataset

The dataset we chose for this project was found on Kaggle. It consist of comments from the Wikipedia's talk page edits and it have already spited into 2 separate set: train and test. In both set, there are around 10% records are labeled as toxic. Both dataset each has over 150 thousand records and contains 9 features including: id, comment_text, id, toxic, severe_toxic, obscene, threat, insult, and identity_hate.



```
               : train   | test
# of rows      : 159571  | 153164
percentage     : 51      | 49
```

Figure 1: Train vs Test Dataset



	id	comment_text	id	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	00001cee341fdb12	Yo bitch Ja Rule is more succesful then you'll...	00001cee341fdb12	-1	-1	-1	-1	-1	-1
1	0000247867823ef7	== From RfC == \n\n The title is fine as it is...	0000247867823ef7	-1	-1	-1	-1	-1	-1
2	00013b17ad220c46	" \n\n == Sources == \n\n * Zawe Ashton on Lap...	00013b17ad220c46	-1	-1	-1	-1	-1	-1
3	00017563c3f7919a	:If you have a look back at the source, the in...	00017563c3f7919a	-1	-1	-1	-1	-1	-1
4	00017695ad8997eb	I don't anonymously edit articles at all.	00017695ad8997eb	-1	-1	-1	-1	-1	-1
...	...	...	...	...	...	...	...	...	...
153159	fffcd0960ee309b5	. \n i totally agree, this stuff is nothing bu...	fffcd0960ee309b5	-1	-1	-1	-1	-1	-1
153160	fffd7a9a6eb32c16	== Throw from out field to home plate. == \n\n...	fffd7a9a6eb32c16	-1	-1	-1	-1	-1	-1
153161	fffda9e8d6fafa9e	" \n\n == Okinotorishima categories == \n\n I ...	fffda9e8d6fafa9e	-1	-1	-1	-1	-1	-1
153162	fffe8f1340a79fc2	" \n\n == ""One of the founding nations of the...	fffe8f1340a79fc2	-1	-1	-1	-1	-1	-1
153163	ffffce3fb183ee80	" \n :::Stop already. Your bullshit is not wel...	ffffce3fb183ee80	-1	-1	-1	-1	-1	-1

153164 rows × 9 columns

Figure 2: Shape of Test Data

Figure x shows for the distribution of tags in the train dataset. Note that the toxicity is not evenly spread out. Therefore, we must be careful about imbalance issues.

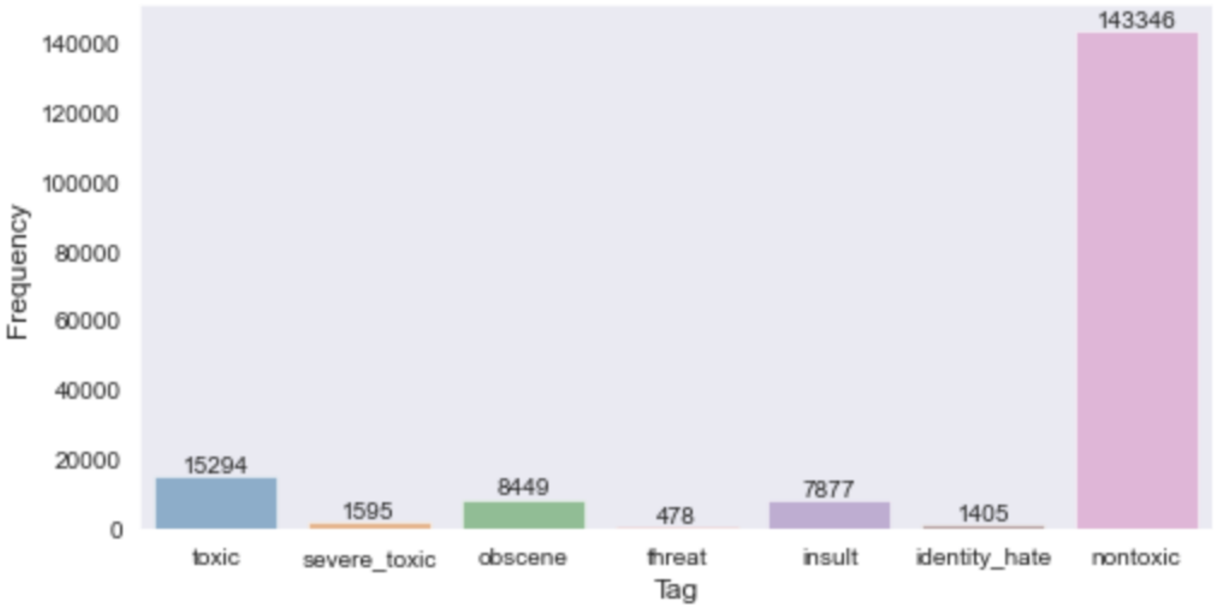


Figure 3: Distribution of tags in Train Data

The fig below shows how many comments have multiple tags. Some comments are tagged with more than one toxic tag.

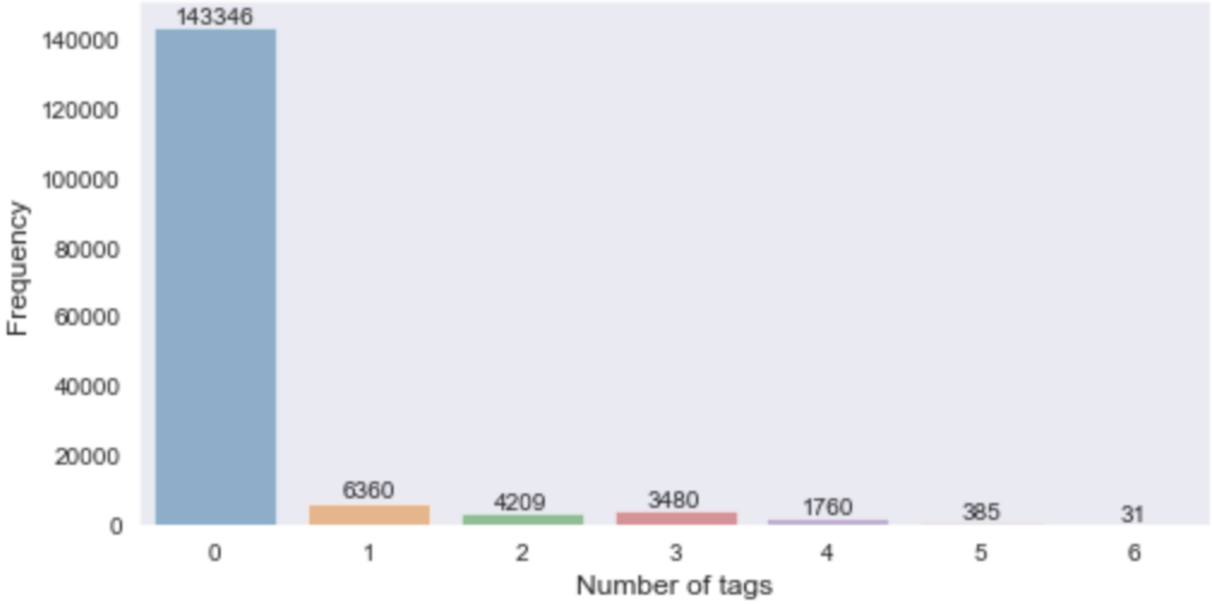


Figure 4: Number of Comments with Multiple Tags

This figure below shows the correlation of different tags.

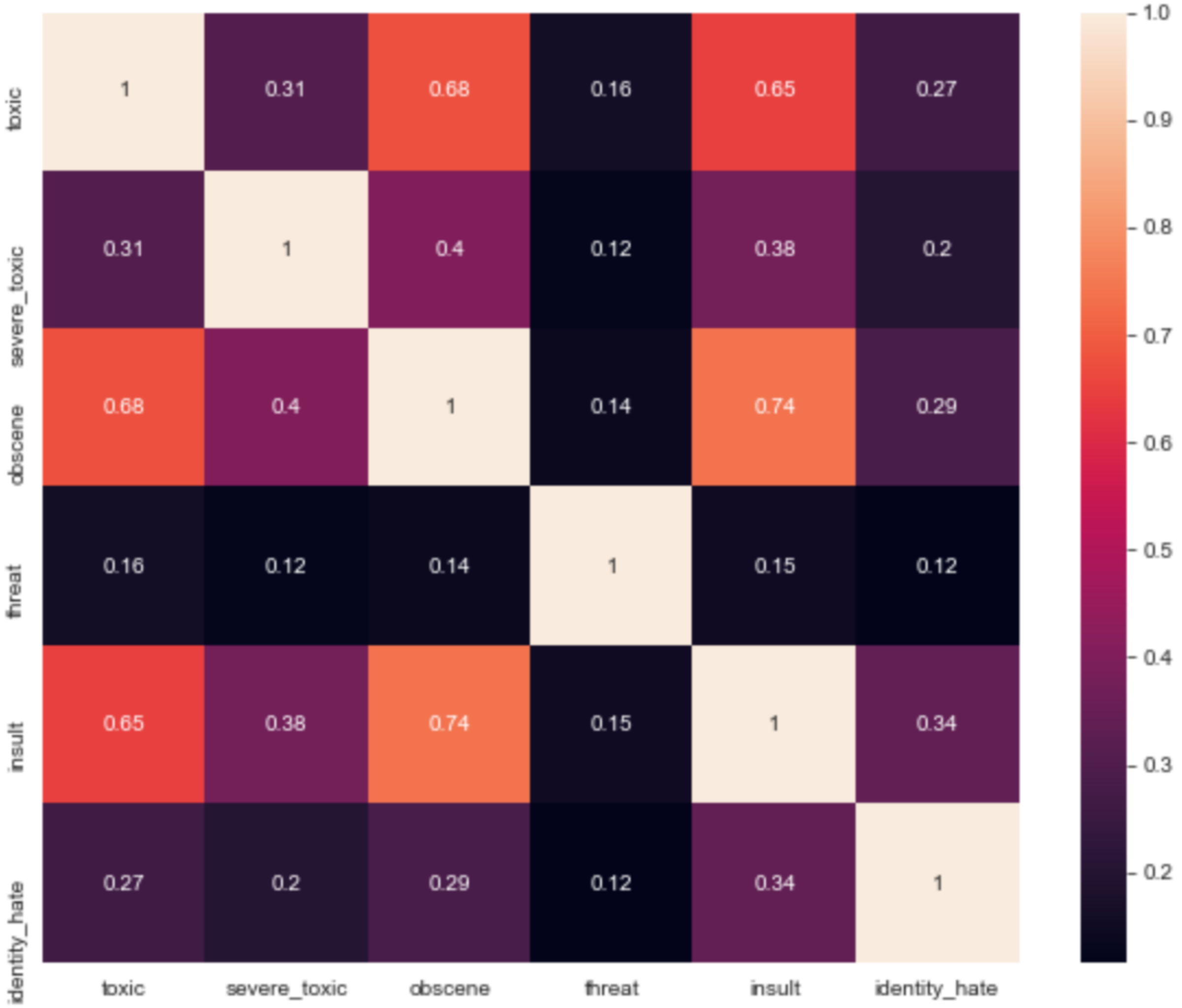
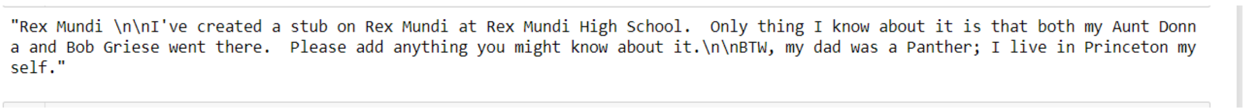


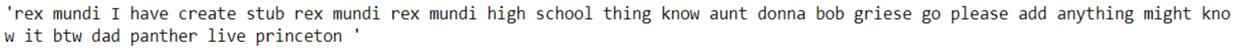
Figure 5: Correlation of Different Tags

1.2.2 Data Cleaning



```
"Rex Mundi \n\nI've created a stub on Rex Mundi at Rex Mundi High School.  Only thing I know about it is that both my Aunt Donn
a and Bob Griese went there.  Please add anything you might know about it.\n\nBTW, my dad was a Panther; I live in Princeton my
self."
```

Figure 6: Raw Data Before Cleanup



```
'rex mundi I have create stub rex mundi rex mundi high school thing know aunt donna bob griese go please add anything might kno
w it btw dad panther live princeton '
```

Figure 7: Data After Cleanup

In Figure: Raw Data Before Cleanup, it can be seen what the raw data was before the cleanup process. In Figure: Data After Cleanup, it can be seen what the data looks like after the cleanup process. This was obtained by completing the following processes.

- All the letters were converted to lowercase.
- All newlines were removed.
- Elements such as IP and username were removed.
- Words with apostrophes were converted to two individual words. (Ex: “you’re” to “you are”)
- Common words were taken out, such as “as”, “a”, “and”.
- All non-alphanumeric and digit characters were removed.

2.0 METHODOLOGY/MODEL USED

We tried to use machine learning classifiers and neural network in this project. Here are the details in below subsections.

2.1 Machine Learning Classifiers

The model used for this project is a pipeline with Countvectoriezer, Tf_IDF transformer, and a Machine learning Classifier. There were multiple Machine Learning Classifiers used for this project, which were Random Forest Classifier, Logistic Regression, Decision Tree, and Multinomial Naive Bayes.

2.2 Neural Networks

To explore the application of modern machine learning techniques we also train a simple neural network model to predict if a comment is toxic. An architecture for the 3 layer MLP (Multi Layer Perceptron) that we train is shown in Fig-1.

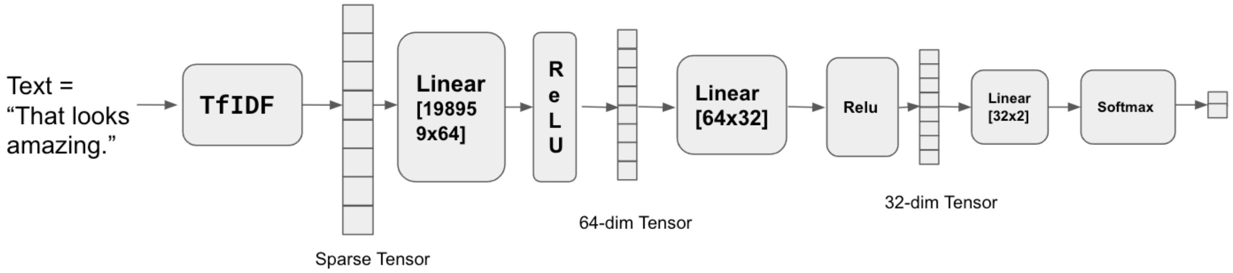


Figure 8: Neural Network Architecture

2.2.1 Neural Network Featurization

We tried two different techniques to featurize the text comments:

1. Tf-IDF: in this method we calculated the standard Tf-IDF weights for words in a text. This method treats each sentence as a bag of words. Then for each word it calculates a term frequency and multiplies it with the inverse document frequency. The term frequency measures how often the word occurred in the particular comment, the inverse document frequency measures the number of unique documents in which the word occurred. This method results in a large feature space of approximately 200,000.
2. BERT sentence encoder: BERT is a modern transformer based neural network which can generate embedding for sentences. Using this technique for each sentence we created a 384-dimensional embedding which was then fed into our MLP model.

2.2.2 Reweighing of Loss

The number of non-toxic comments in our train dataset is 9 times the number of toxic comments. This skew in the dataset increases the difficulty of the neural network learning process. To get around this we use a weight term in our CrossEntropy loss. We experimented with different values of the weight term, a weight penalty of 1 for the non-toxic class and 4 for the toxic class gives us the best F1-score.

3.0 RESULTS

3.1 Machine Learning

ML Classifier	Label	Precision	Recall	F1-Score	Support	Accuracy
Random Forest	Non-Toxic	0.94	0.96	0.95	57888	0.91
	Toxic	0.55	0.46	0.50	6090	
Logistic Regression	Non-Toxic	0.98	0.93	0.95	57888	0.92
	Toxic	0.55	0.79	0.65	6090	
Decision Tree	Non-Toxic	0.97	0.89	0.93	57888	0.88
	Toxic	0.42	0.76	0.54	6090	
Multinomial NB	Non-Toxic	0.92	1.00	0.96	57888	0.92
	Toxic	0.81	0.18	0.29	6090	

Table 1: Model Performance with Raw Data Training

Initially, all the machine learning algorithms with Countvectorizer and TF-IDF Transformer were trained with the original raw data without any preprocessing cleaning performed on the dataset to obtain a benchmark accuracy. The performance of each machine learning classifier can be seen in Table 1: Model Performance With Raw Data. The highest accuracy obtained was 92% with logistic regression, but the Precision, Recall and F1-Scores obtained by the Toxic comments were far from optimal. Most were below or within the 50% mark. The reason for this could be because of the imbalance of the training data as there are 10 times as many Non-toxic comments as toxic comments.

ML Classifier	Label	Precision	Recall	F1-Score	Support	Accuracy
Random Forest	Non-Toxic	0.96	0.97	0.97	57888	0.94
	Toxic	0.70	0.63	0.66	6090	
Logistic Regression	Non-Toxic	0.98	0.93	0.95	57888	0.92
	Toxic	0.54	0.78	0.64	6090	
Decision Tree	Non-Toxic	0.98	0.91	0.94	57888	0.90
	Toxic	0.48	0.80	0.60	6090	
Multinomial NB	Non-Toxic	0.92	1.00	0.96	57888	0.92
	Toxic	0.89	0.19	0.32	6090	

Table 2: Model Performance with Cleaned Data

The Performance of the machine learning classifier model with cleaned data can be seen in Table 2: Model Performance With Cleaned Data. The highest accuracy increased by 2% when the machine learning classifiers were trained with the cleaned data, where the highest accuracy was

94% with Random Forest Classifier. The Precision, Recall, and F1-Score for the Toxic comments increased with the cleaned data training , where most scores increased by 15%.

3.2 Confusion Matrix

As we have seen throughout the project, we have been experimenting with different methods such as classical algorithms namely Decision Trees, Random Forest, Multinomial Naive Bayes, and Logistic Regression, also Neural Networks were applied to the dataset to compare performances among all of them and see the highest accuracy reached for toxic and non-toxic comments

Countvectorizer and TF-IDF were utilized to allow the text to be processed, and after running the different alternatives we came to the conclusion that the highest accuracies were achieved by the Random Forest algorithm, therefore, We will be explaining in detail the confusion matrix for each case to get a deeper understanding about the scores and its meanings

A confusion matrix is a tool widely used to measure the performance of a model, and from it, we can derive some other measurements such as recall, accuracy, and precision. Based on the circumstances of this project, that matrix is made up of two labels (toxic, non-toxic), two types of values defined as predicted values and actual values, finally, it has 4 possible results which are: true positive, false positive, false negative, true negative

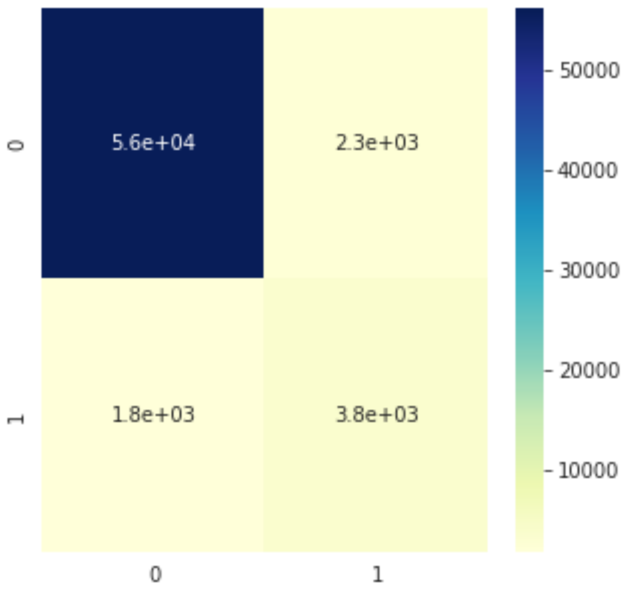


Figure 9: Confusion Matrix - Random Forest with CountVectorizer

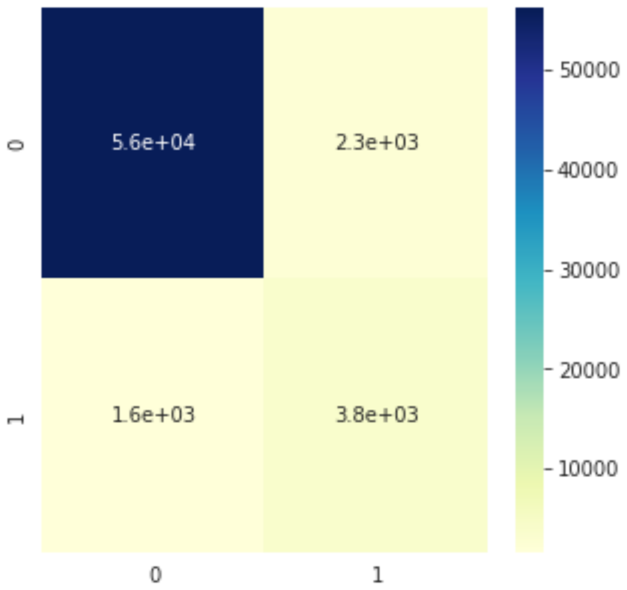


Figure 10: Confusion Matrix - Random Forest with CountVectorizer and Tf-idf Transformer

For both matrices, results are very similar in terms of true positives and true negatives: the majority of the results are allocated right in these segments respectively, nevertheless, there are two measurements that present a slight variation in comparison to the previous scenario, namely, F1 score went from 0.65 to 0.66 and Precision from 0.68 to 0.70, meaning that by readjusting the type of vectorizer we enhanced the model accuracy, which is 0.94 for both, for toxic comments

3.3 Experiments: Neural Network

S.No	Neural Network	Feature	Label	Precision	Recall	F1-Score	Support	Accuracy
1	2 layer MLP	Tf-IDF	Non-Toxic	0.93	0.91	0.92	57888	0.88
		Sent-encoder	Toxic	0.49	0.62	0.54	6090	
2	3 layer MLP	Tf-IDF	Non-Toxic	0.94	0.91	0.92	57888	0.90
		Sent-encoder	Toxic	0.52	0.63	0.57	6090	
3	2 layer MLP + weighted loss	Tf-IDF	Non-Toxic	0.97	0.94	0.95	57888	0.93
		Sent-encoder	Toxic	0.57	9,71	0.63	6090	
4	3 layer MLP + weighted loss	Tf-IDF	Non-Toxic	0.97	0.95	0.96	57888	0.94
		Sent-encoder	Toxic	0.62	0.76	0.65	6090	
5	2 layer MLP	Tf-IDF	Non-	0.96	0.94	0.95	57888	0.91

			Toxic					
		Sent-encoder	Toxic	0.54	0.67	0.60	6090	
6	3 layer MLP	Tf-IDF	Non-Toxic	0.95	0.97	0.96	57888	0.93
		Sent-encoder	Toxic	0.63	0.55	0.59	6090	
7	2 layer MLP + weighted loss	Tf-IDF	Non-Toxic	0.97	0.91	0.94	57888	0.90
		Sent-encoder	Toxic	0.47	0.77	0.59	6090	
8	3 layer MLP + weighted loss	Tf-IDF	Non-Toxic	0.97	0.92	0.95	57888	0.91
		Sent-encoder	Toxic	0.51	0.75	0.61	6090	

Table 3: Performance of Different Neural Network Settings

Our best performing neural network configuration achieved an accuracy of 0.94 with an F1-Score of 0.65. The primary improvement in our model configurations can be measured by focusing on the improvement in F1-Score when we reweigh the classes. Given an increased penalty for the Toxic class results in an increased F1-Score for the toxic class, for example for the 3-layer MLP the F1-Score for toxic class improved from 0.57 to 0.65 when we used the weighted loss term.

We experimented with a larger penalty term for the Toxic class but that led to a significant hit in the Precision measurement of the model. If we give a large penalty to the model for getting the toxic class wrong, then the model predictions would have a large number of false positive which would lead to a significant hit on the accuracy.

Using the BERT sentence encoder did not lead to improvements in the performance, this was likely because the dataset used for pre-training of the BERT encoder is significantly different from our toxic comments dataset. In our experiments the Tf-IDF features gave the best performance.

4.0 CONCLUSIONS

The purpose of this project was to identify toxic and non-toxic comments on social media, to address that, we started off with trying to see how the model perform with raw data and we used several algorithms, then after the data cleaning process we fit the model and made some predictions to create a benchmark analysis and make comparisons among all of them and also see how Neural Networks work for a task like this.

Based on all the previous procedures and after all the research we have done so far, here are some conclusions we have come to:

1. Clean data VS Raw data makes the difference in terms of accuracy:

   Feature engineering, data preprocessing, and cleaning lead us to better results, and for the algorithm, with the best accuracy, we went from 0.91 to 0.94 by applying these techniques to raw data.

2. Based on the simplicity of a binary classifier task, Classic Algorithms turn out a better option rather than NN:

   Considering the ranking we created to compare these classifiers, the best results were reached by Neural Networks and Random Forest, nonetheless, the last one outperformed the rest on this task since the simplicity of this case, that is, a binary classification problem doesn't require necessarily using a more sophisticated tool.

3. We are looking forward to seeing new results by trying some other variations of this methodology:

   We challenged our model by putting it into a test with some toxic and non-toxic phrases, some of them provided by and professor some classmates and it turned out that It worked out pretty well, however, we will be working on it to tackle the problems inherent to this topic like detecting sarcasm, for instance, and improve the model to make social media a place with a better and healthier communication environment

REFERENCES

[1] Narkhede, S. (2021, June 15). Understanding confusion matrix. Medium. Retrieved December 5, 2021, from https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62.

[2] Toxic comment classification challenge. Kaggle. Retrieved December 5, 2021, from https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data.