A Data Mining approach for Classification of Toxic Comments and Empowering Positive Discourse

Deeraj Nair
deenair@iu.edu

Malhar Dhopate
mdhopate@iu.edu

Rishit Puri
rispuri@iu.edu

Luddy School of Informatics, Computing, and Engineering
Indiana University Bloomington
Bloomington, IN, 47405

Table of Contents

Deeraj Nair *Malhar Dhopate* *Rishit Puri* *1*

1. Abstract And Introduction **3**

1.1 Abstract *3*

1.2 Introduction *3*

1.3 Background / Related work *3*

2. Methods **5**

2.1 Data Collection *5*

2.2 Data Preprocessing *5*

2.3 Data Visualization *5*

2.4 Logistic regression *10*

2.5 Random Forest *10*

2.6 SVM *11*

2.7 Neural Network *11*

2.8 Sarcastic Comment Classification *13*

3. Results and Discussion **13**

4. Future Scope **15**

5. Individual Contribution **15**

6. References **16**

1. Abstract And Introduction

1.1 Abstract

In an era of ubiquitous online communication, hate speech and toxic remarks have attracted a lot of attention. This project analyses this issue through data mining and machine learning algorithms. The primary focus of this project is to classify toxic comments with a high recall - as it is important to correctly identify toxic comments, and check the clean comments for sarcasm, as it can also cause a negative impact. We aim to identify the unfavorable content by utilizing natural language processing and sentimental analysis models. Additionally, based on these models we developed a recommendation system which promotes the users to opt for positive discourse. In addition to being a technical undertaking, this project is aimed towards fostering a more positive, respectful and inclusive online environment.

Keywords - Logistic Regression, Support Vector Machine (SVM), Sentimental Analysis, Classification, Recommendation, Neural Network, Tokenizer.

1.2 Introduction

The digital age has brought an unprecedented level of connectivity and communication through online platforms. In-spite of opening avenues for interactions, these platforms have become incubators for toxic and harmful comments. The presence of such comments disrupts meaningful conversations, threatens the emotional and psychological well-being of an individual and leads to a hostile online environment.

In response to this problem - instead of only identifying a toxic comment, our project aims to develop a reliable system to proactively recommend the users to revise their comments to foster a healthier and respectful online environment. This will be achieved by utilizing a variety of data analysis techniques, including sentiment analysis and machine learning models like Logistic Regression, Support Vector Machine (SVM), Random Forest, and Neural networks.

The project makes use of sentiment analysis to determine the emotional tone and purpose underlying the remarks. By integrating various machine learning model outputs, with sentiment analysis insights the study seeks to advance the creation of efficient comment moderation system for online platforms.

1.3 Background / Related work

There have been several studies conducted to classify the toxic comments with a variety of machine learning models, and there have been several studies to determine the effects of toxic comments on the online environment and an individual's mental and emotional health. The paper by Julian Risch, Ralf Krestel [6] talks about how crucial the comment sections are for online news platforms which is often misused by spammers and haters. Sentiment analysis can aid in both content moderation and understanding online discussion dynamics and help in toxicity detection.

There have been several studies conducted to determine the impact of different machine learning models, the study conducted [1] by Etibar Aliyev presents a practical implementation of a neural network-based model for toxic comment classification. The model is trained to classify

comments into six different categories of toxicity: toxic, severe toxic, obscene, threat, insult, and identity hate. The model is compiled with the Binary Cross entropy loss function and the Adam optimizer. The model predicts the probabilities of each toxicity category for the input text. A threshold of 0.5 is applied to convert the probabilities into binary predictions and have been evaluated using precision, recall and accuracy metrics.

Similar to [1], the research conducted by [2] aims to understand the effectiveness of deep learning models compared to machine learning models along with the most common models used by researchers in the last 5 years. The paper has also provided insight on the most common data sets utilized by researchers to detect toxic comments. To achieve this, they have compiled the datasets of research papers and analyse the algorithm used. The findings indicate that Long Term Short Memory is the most routinely mentioned deep learning model with 8 out of 26 research papers. There have been attempts to combine more than one deep learning algorithms, however these hybrid models might not result in a better accuracy than an original model. In conclusion LSTM has better accuracy with highest being at 97 percent compared to other models. This study determined that neural networks with TF-IDF pre-processing has the best accuracy. [4] has conducted studies using LSTM and Naive Bayes models to classify toxic comments and observed that the LSTM model had a higher true positive rate (20% more) compared to the Naive Bayes model. This paper also refers studies that have proven that a higher number of toxic comments are harmful to an individual's emotional and psychological health. The findings emphasize the potential of data science in creating a healthier online environment, achieving a promising accuracy of over 70% using LSTM.

[3] study identifies the types of online users' by analyzing their online activity in-terms of the comments posted. This research introduces two metrics, the F score and G score, to identify types of toxic online user behavior. Analyzing 4 million Reddit comments, the study classifies users into four categories: Steady Users, Fickle-Minded Users, Pacified Users, and Radicalized Users based on their toxicity scores. The paper segments the users based on their toxicity scores based on their comments. The paper concluded by saying that the most toxic behavior is when a user switched constantly between toxic and non-toxic commenting. The results suggests that a complex model could be developed to segment users based on activity, and the platform admins could use these results to warn the users who affect the online environment.

[5] the study focuses on automatic methods for discovering toxicity using machine learning models. The authors conducted a systematic review of 31 primary studies to understand the state of the art in this field. They analyzed various aspects, including publication trends, dataset usage, evaluation metrics, machine learning methods, classes of toxicity, and comment languages. The research reveals that this area gained significant attention from 2018, with most studies being conference papers, and it is still an evolving research topic. # The primary dataset used in many studies is Jigsaw's dataset, and deep neural networks are among the most effective machine learning methods.

Majority of the studies above have either trained models using neural networks to identify toxic comments or segment the toxic users based on their comment activity. Our study also tries to identify the performance differences between a variety of ML models and determine which model could be able to provide a relatively simple real-time application.

2. Methods

This project focuses on studying the effects of machine learning models like Logistic Regression, SVM, Random Forests, Neural Networks, etc. We are focusing on F1 and Recall Score, we detail our methods, along with the description of data used below.

Initially, the project began by gathering a substantial dataset of comments that are known to contain dangerous or toxic content. These remarks can be retrieved from social networking sites, forums, publicly accessible sources, pre-compiled datasets, and any other relevant source that contains such remarks. We will ensure that the dataset accurately reflects the harmful remarks that people could run into in everyday life.

2.1 Data Collection

The toxic comment dataset is available on kaggle. It's a common dataset used in challenges across kaggle. It consists of 8 columns and 159571 rows. The comments in this dataset have been classified into 6 different levels of toxicities: Toxic, severe toxic, obscene, threat, insult, identity hate.

Similar to the toxic comment dataset, we were able obtain a news headlines dataset on Kaggle to develop a model to identify Sarcasm. This dataset consists of 3 columns and 26709 rows, these rows are classified as either 0 – not sarcastic, or 1 – sarcastic.

2.2 Data Preprocessing

For both the datasets, we looked at we looked over the dataset for null values once the data was loaded. To clean the data all the links, special characters, and line spacing characters were removed. After cleaning the data, the stopwords were eliminated from the comments since they do not contribute to the accuracy of the model. Following that, we proceeded with tokenization, lemmatization, and stemming of the comments. The practice of reducing big words to smaller ones is called tokenization. Lemmatization is the process of deleting the inflectional ends from words to return them to their basic form. Stemming is an NLP method used to turn phrases into sequences so that domain vocabularies are obtained. This step allowed us to reduce the dimensions of the data as all the stopwords were removed, and the words were standardized to remove the tense eg – walking, walks, etc, became walk.

2.3 Data Visualization

The WordCloud, Seaborn, and Matplotlib packages were utilized for data visualization. We showed the percentages of the various categories of comments using a pie chart. We drew wordclouds for each sort of comment since there are six distinct levels of toxicity. A histogram was also generated to examine the average length of comments throughout the dataset. Additionally, a bar plot was created to examine the comments that fit into more than one classification.

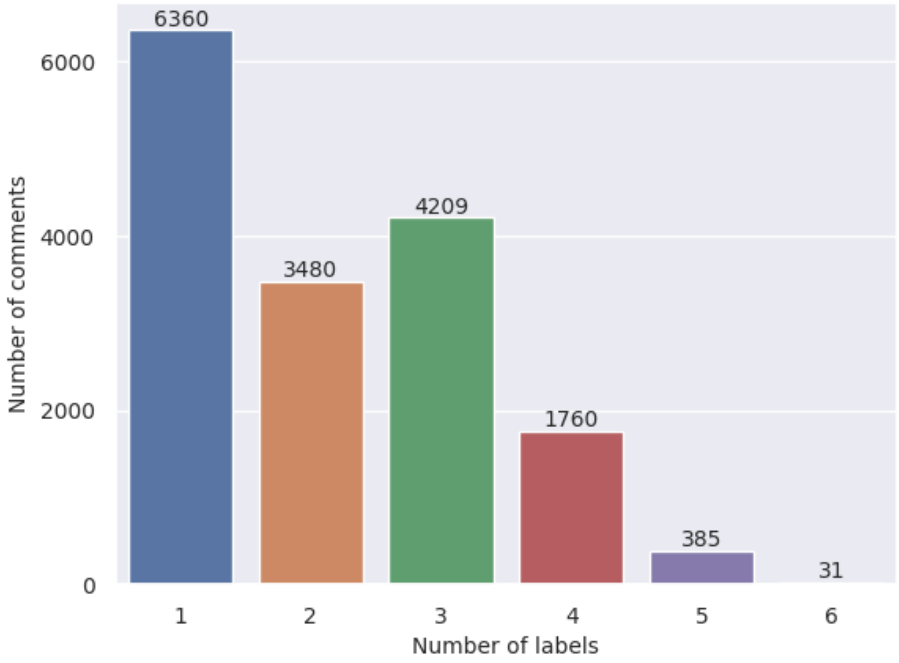


Fig 2.1 Bar plot to examine multi label classification.

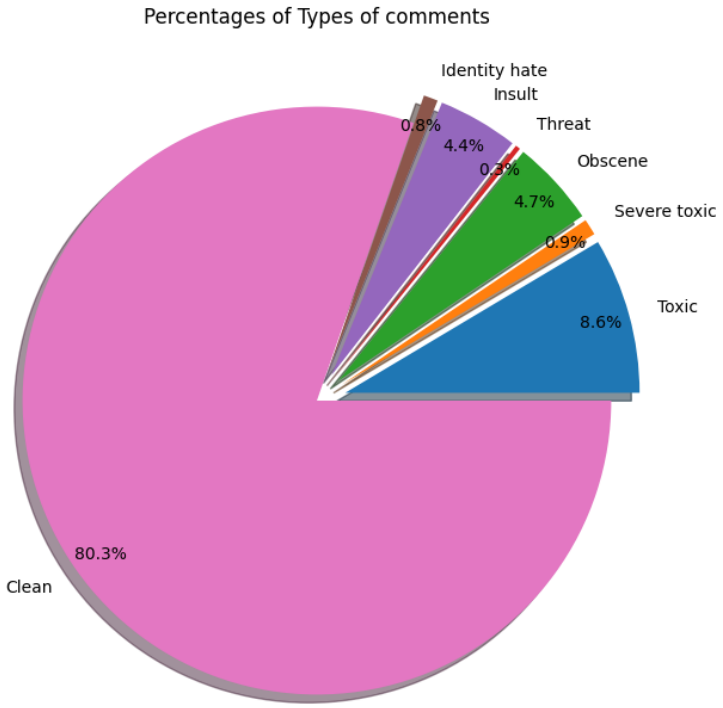


Fig 2.2 Pie chart to examine the percentages of types of comments

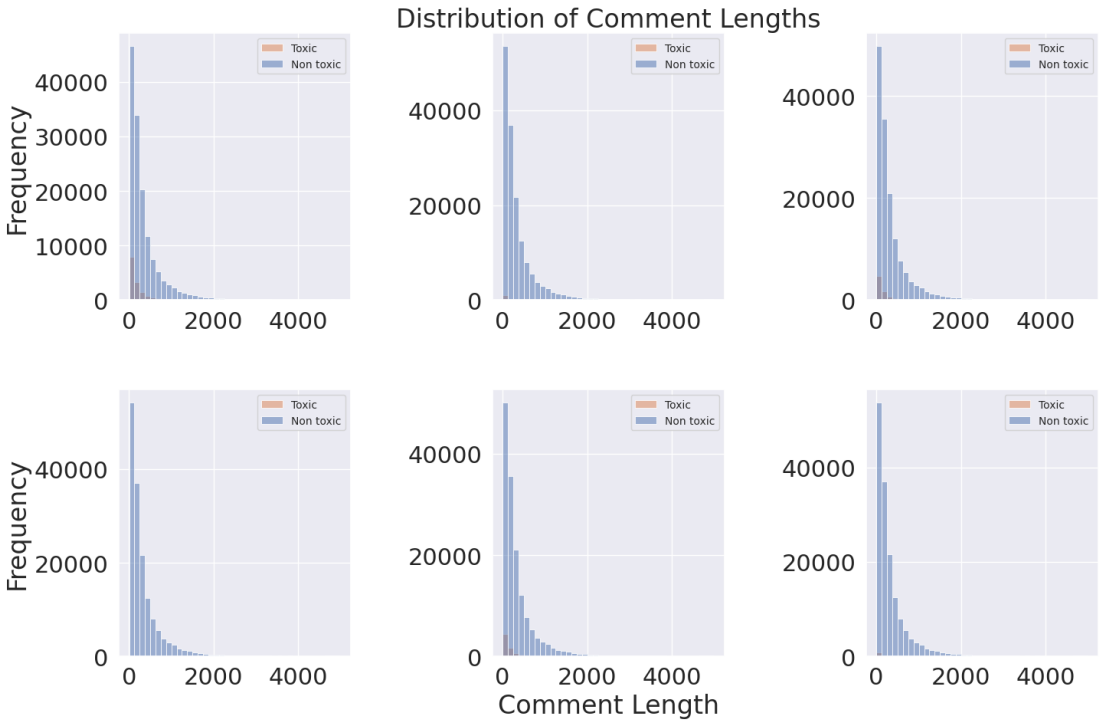


Fig 2.3 Histogram representing length of comments of different toxicities vs clean comments.

Fig 2.4 Severe toxic wordcloud

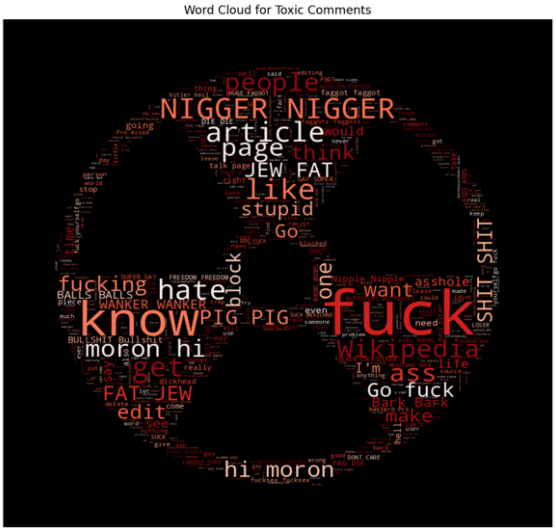


Fig 2.5 toxic wordcloud

Fig 2.6 Threat wordcloud

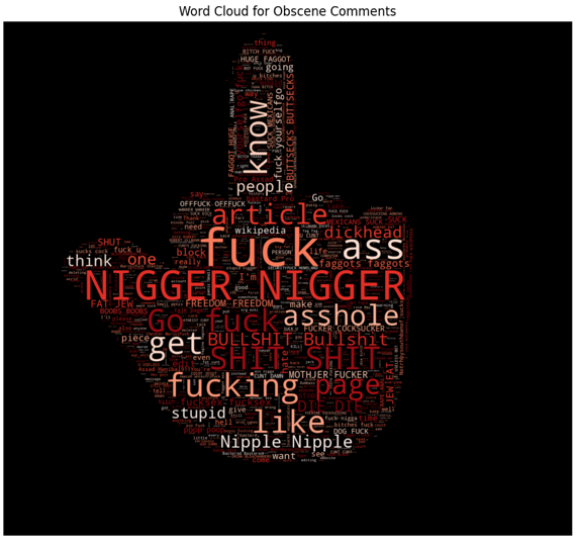


Fig 2.7 Obscene wordcloud

Fig 2.8 Identity hate wordcloud

Fig 2.9 Insult wordcloud

Fig 2.10 Sarcasm wordcloud

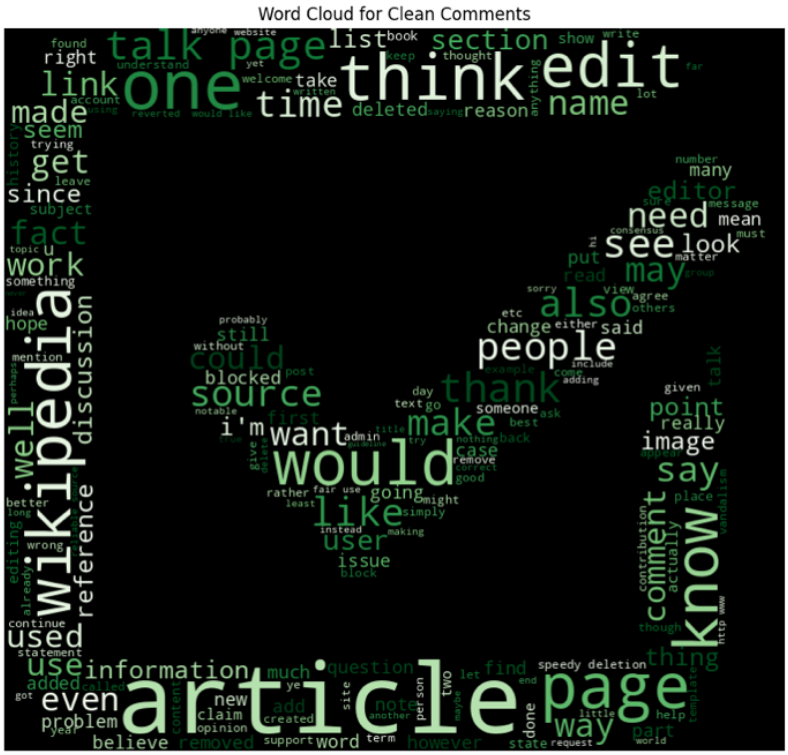


Fig 2.11 Clean wordcloud

2.4 Logistic regression

In cases when the dependent variable is categorical, logistic regression is employed. Additionally, we included all six classifiers into the "combined" attribute. After dividing the dataset into train and test halves, the model was trained, and the model performance was evaluated using the recall scores.

	Label	Accuracy	Precision	Recall	F1 Score
0	toxic	0.956415	0.954746	0.956415	0.952471
1	severe_toxic	0.990694	0.988483	0.990694	0.989017
2	obscene	0.976719	0.975659	0.976719	0.974597
3	threat	0.997807	0.997129	0.997807	0.997101
4	insult	0.969638	0.966786	0.969638	0.966007
5	identity_hate	0.991634	0.989695	0.991634	0.988865
6	Combined	0.919223	0.893610	0.919223	0.902498

Fig 2.12 Logistic Regression results

2.5 Random Forest

Random Forest is a machine learning approach which aggregates the output of several decision trees to produce a single outcome. We found that random forest took longer to compute than logistic regression when we were running the model. The result of random forest is given below,

	Label	Accuracy	Precision	Recall	F1 Score
0	toxic	0.958076	0.955908	0.958076	0.956163
1	severe_toxic	0.989879	0.985424	0.989879	0.986061
2	obscene	0.978881	0.977884	0.978881	0.978122
3	threat	0.997775	0.997101	0.997775	0.996905
4	insult	0.969732	0.967380	0.969732	0.968080
5	identity_hate	0.991634	0.989623	0.991634	0.988943
6	Combined	0.917938	0.893635	0.917938	0.901150

Fig 2.13 Random Forests results

The Recall and F1 scores obtained for Random Forest and Logistic Regression are very similar to each other, just the computational time for training the Random Forest Classifier is higher (30 mins) compared to Logistic Regression (5 mins).

2.6 SVM

SVM is a potent sci-kit-learn technique that makes use of supervised learning models to tackle challenging issues including regressions, classification, outlier identification, and so on. We noticed that this model requires a significant amount of time to train.

	Label	Accuracy	Precision	Recall	F1 Score
0	toxic	0.958953	0.957040	0.958953	0.956094
1	severe_toxic	0.990130	0.988277	0.990130	0.985517
2	obscene	0.977973	0.976777	0.977973	0.976508
3	threat	0.997713	0.997718	0.997713	0.996601
4	insult	0.969575	0.966656	0.969575	0.966981
5	identity_hate	0.991759	0.990327	0.991759	0.988873
6	Combined	0.919474	0.893647	0.919474	0.903212

Fig 2.14 SVM results

Though SVM is a very powerful classifier, it has a very high computational complexity, this model takes the longest to train (60+ mins). Though this model has very high Recall and F1 scores, the model is slightly imbalanced as the six training labels do not have equal representation.

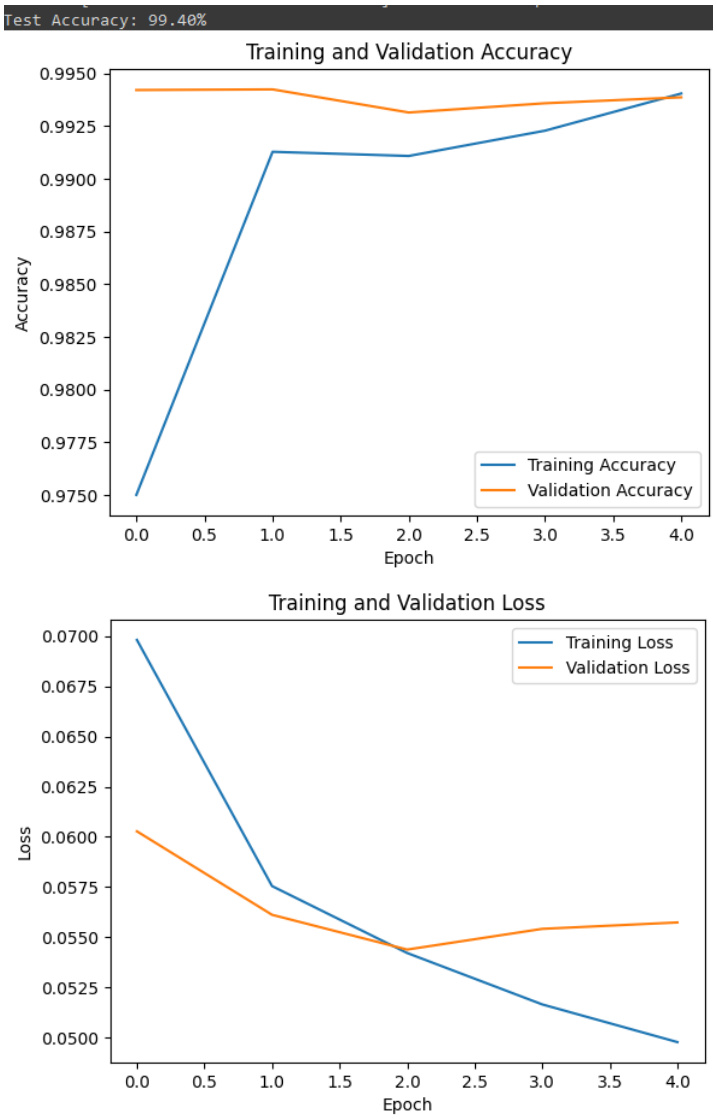
2.7 Neural Network

Neural networks are a form of machine learning technique that are inspired by the natural signals that are transmitted by neurons in the human brain. In some circumstances, they can perform as accurately as the human brain. One of the most active areas of research and application is neural network categorization. In order to categorize toxicity, and identify the sarcastic nature of remarks, we have developed a basic neural network and trained it on accessible data as part of this study.

Sequential neural networks for deep learning that enable information retention are called long short-term memory networks. It is a unique kind of recurrent neural network that can solve the RNN's vanishing gradient issue. Hochreiter and Schmidhuber created LSTM to address the issues with conventional RNNs and machine learning methods. The use of memory cells and a series of gates to regulate the information flow within the cell is the primary innovation of LSTM networks. Among the gates are Forget Gate: Chooses which state-related data from the cell should be retained or deleted. Input Gate: Adjusts the state of the cell to incorporate new data. Cell State: The cell's memory, which is updateable on a selective basis. Output Gate: Produces the output of the cell based on the modified cell state.

Text data and comments are sequential by nature. Important information is frequently conveyed by the arrangement of phrases in a comment or by the words in a statement. LSTMs work effectively for jobs where context is important because they can handle sequential input and

capture long-range dependencies. The vanishing gradient issue frequently affects traditional RNNs, making it difficult for the network to learn long-range relationships. In order to overcome this problem, LSTMs employ a more complex design that includes memory cells and gating mechanisms, allowing for more efficient training on sequential input.



- Text input is transformed into sequences of integers using the TextVectorization preprocessing layer in Keras.
- The maximum word count in the vocabulary is indicated by the variable MAX_FEATURES.
- The maximum length of the output sequences is determined by the parameter output_sequence_length.
- The vectorizer is adjusted to the dataset via vectorizer.adapt (X.values), which creates the vocabulary.
- The integer sequences that represent the comments are contained in vectorized_text.
- With thick layers, a bidirectional LSTM layer, and an embedding layer, the model is a sequential neural network.
- Embedding: Creates dense vectors of a predetermined size from integer sequences.
- Bidirectional (LSTM): The input sequence is processed both forward and backward by a

bidirectional LSTM layer.

- For further processing, dense layers with different activation functions are employed.
- For multi-label classification, the last layer employs sigmoid activation.

2.8 Sarcastic Comment Classification

After developing and implementing multiple machine learning models, we can identify toxicity precisely, with high Precision, but they are unable to classify the underlying emotional tone of the comments. Thus, we developed a model to identify sarcasm from comments as sarcasm is also one of the emotions that can be detrimental to a content creator. From the figures below, 2.15 we can observe that the Validation loss reduces as the training epochs increases, and the accuracy increases with each training epoch. The model is able to produce an accuracy of 99.7%

```
Epoch 1/20
585/585 - 13s - loss: 0.6057 - accuracy: 0.6546 - val_loss: 0.4654 - val_accuracy: 0.7807 - 13s/epoch - 22ms/step
Epoch 2/20
585/585 - 3s - loss: 0.3581 - accuracy: 0.8455 - val_loss: 0.4236 - val_accuracy: 0.8066 - 3s/epoch - 6ms/step
Epoch 3/20
585/585 - 3s - loss: 0.2493 - accuracy: 0.9012 - val_loss: 0.4530 - val_accuracy: 0.8028 - 3s/epoch - 5ms/step
Epoch 4/20
585/585 - 3s - loss: 0.1825 - accuracy: 0.9322 - val_loss: 0.5140 - val_accuracy: 0.7990 - 3s/epoch - 5ms/step
Epoch 5/20
585/585 - 3s - loss: 0.1396 - accuracy: 0.9483 - val_loss: 0.5771 - val_accuracy: 0.7922 - 3s/epoch - 6ms/step
Epoch 6/20
585/585 - 4s - loss: 0.1064 - accuracy: 0.9625 - val_loss: 0.6752 - val_accuracy: 0.7839 - 4s/epoch - 6ms/step
Epoch 7/20
585/585 - 3s - loss: 0.0858 - accuracy: 0.9711 - val_loss: 0.7574 - val_accuracy: 0.7797 - 3s/epoch - 5ms/step
Epoch 8/20
585/585 - 3s - loss: 0.0660 - accuracy: 0.9794 - val_loss: 0.8236 - val_accuracy: 0.7754 - 3s/epoch - 5ms/step
Epoch 9/20
585/585 - 2s - loss: 0.0531 - accuracy: 0.9837 - val_loss: 0.9142 - val_accuracy: 0.7747 - 2s/epoch - 4ms/step
Epoch 10/20
585/585 - 4s - loss: 0.0449 - accuracy: 0.9862 - val_loss: 1.0114 - val_accuracy: 0.7690 - 4s/epoch - 6ms/step
Epoch 11/20
585/585 - 4s - loss: 0.0368 - accuracy: 0.9883 - val_loss: 1.0889 - val_accuracy: 0.7649 - 4s/epoch - 7ms/step
Epoch 12/20
585/585 - 2s - loss: 0.0303 - accuracy: 0.9913 - val_loss: 1.2152 - val_accuracy: 0.7608 - 2s/epoch - 4ms/step
Epoch 13/20
585/585 - 2s - loss: 0.0256 - accuracy: 0.9929 - val_loss: 1.2665 - val_accuracy: 0.7621 - 2s/epoch - 4ms/step
Epoch 14/20
585/585 - 3s - loss: 0.0228 - accuracy: 0.9935 - val_loss: 1.3530 - val_accuracy: 0.7624 - 3s/epoch - 5ms/step
Epoch 15/20
585/585 - 3s - loss: 0.0191 - accuracy: 0.9944 - val_loss: 1.4365 - val_accuracy: 0.7606 - 3s/epoch - 6ms/step
Epoch 16/20
585/585 - 3s - loss: 0.0176 - accuracy: 0.9949 - val_loss: 1.5361 - val_accuracy: 0.7608 - 3s/epoch - 6ms/step
Epoch 17/20
585/585 - 2s - loss: 0.0170 - accuracy: 0.9944 - val_loss: 1.6285 - val_accuracy: 0.7571 - 2s/epoch - 4ms/step
Epoch 18/20
585/585 - 3s - loss: 0.0160 - accuracy: 0.9943 - val_loss: 1.6761 - val_accuracy: 0.7568 - 3s/epoch - 5ms/step
Epoch 19/20
585/585 - 3s - loss: 0.0107 - accuracy: 0.9966 - val_loss: 1.7899 - val_accuracy: 0.7561 - 3s/epoch - 5ms/step
Epoch 20/20
585/585 - 3s - loss: 0.0088 - accuracy: 0.9975 - val_loss: 1.8965 - val_accuracy: 0.7551 - 3s/epoch - 5ms/step
```

Fig 2.15 Sarcastic Comment Model Training

3. Results and Discussion

For this project we were able to successfully develop and implement 4 highly precise models to identify toxic comments. We can see the predictions made by the models below –

```
# Test Comment
new_comment = "You should keep fucking gay yourself."
```

Fig 3.1 Test Comment

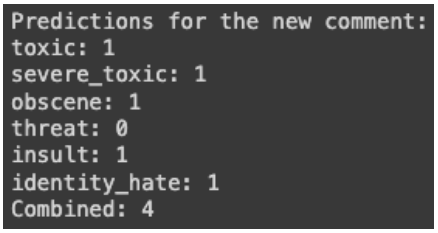


Fig 3.2 Logistic Regression Prediction

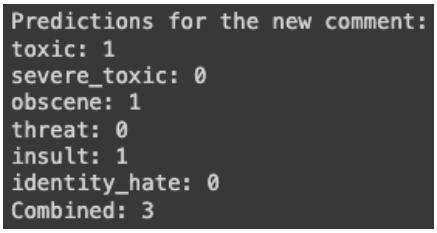


Fig 3.3 Random Forest Prediction

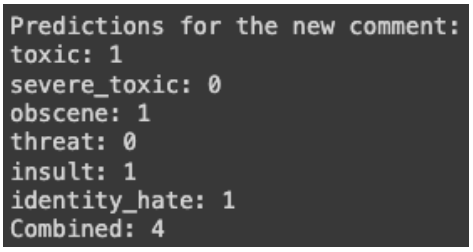


Fig 3.4 SVM Prediction

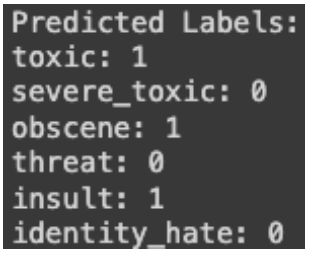


Fig 3.5 Simple Neural Net Prediction

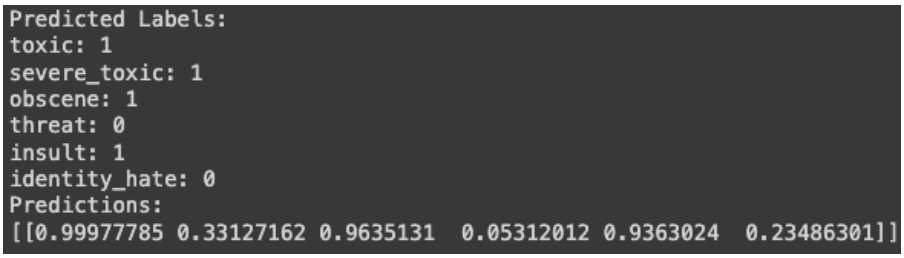


Fig 3.6 LSTM Neural Net Prediction

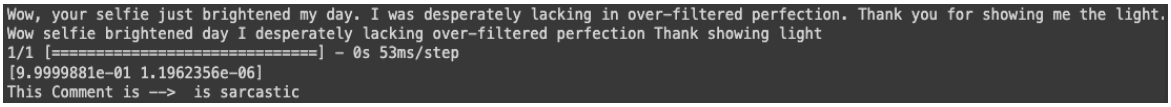


Fig 3.7 Sarcasm Input & Prediction

As seen from the above, different models predict different or same outcomes for the same comment, which can be explained below,

As seen in the Methods section, Logistic Regression, Random Forest and SVM have comparable Recall and F1 scores. Through the "Combined" metrics, we can see how the performance for multilabel (0- 6) is worse compared to the performance when the models are trained on individual labels. Though SVM had the higher Recall and F1 score for the 'Combined' label, we were unable to design an efficient model.

We realized that there are some instances where underfitting occurs, as some labels do not have sufficient training data. Since there aren't many data rows for 'identity_hate' and 'threat' comments, the models may be underfitting such labels. For example – out of the 150,000 + rows only 950 contribute to 'identity hate' due to which complex models like Random Forest, SVM, and Neural Network are unable to predict such labels accurately.

In conclusion, because the models can predict the comment toxicity with a high precision, we were able to achieve our project objective – which was to develop a model to identify toxic and sarcastic comments and can suggest to the users to choose positive discourse.

The model now is able to identify the toxicity of a comment, and also identify whether a comment is Sarcastic if **no** toxicity is identified.

4. Future Scope

The models developed through the course of this study are simple, in the sense where the comments are classified based on the presence of certain words which are used to determine Sarcasm and Toxicity. This model is not perfect and can be further improved in the below mentioned areas –

1. Ensemble models could be developed to identify/classify the comments.
2. The neural networks could be made more complex to analyze different languages for multilingual toxicity detection.
3. The pre-processing functions could be updated to incorporate emojis, as they can also be used to express toxicity to comments.
4. The model could be implemented with real-time monitoring capabilities to help the users identify and address the toxicity of their comments before they are posted.
5. Implement the model on various online platforms.
6. Integrate user feedback mechanisms to enhance the model's performance on real-world usage, this can allow users to manually identify the toxicity/emotion in a comment, which can help the model to be trained better.

5. Individual Contribution

1. Deeraj Nair

- Conducted EDA, Pre-Processing function, Data Visualization, Developed, trained, and implemented Logistic regression.
- Contributed to the compilation of the project report.

2. Malhar Dhopate

- Helped with Pre-Processing function, Developed, trained, and implemented Random Forest Model, and Sarcasm Detection neural network.
- Contributed to the design of the project report and code file.

3. Rishit Puri

- Developed, trained, and implemented SVM Model, and Toxic comment Classification neural networks.
- Contributed to the review, editing of the project report.

This breakdown reflects the individual contribution of authors. The authors have contributed to the best of their abilities in a very collaborative manner, while communicating efficiently, and helping each other when needed. The entire team contributed in data collection, brainstorming for the project.

6. References

[1] Etibar Aliyev. 2023. *Toxic Comment Classification: Identifying Harmful Content Using Recurrent Neural Networks*. medium.com

[2] Felix Museng, Adelia Jessica, Nicole Wijaya, Anderies Anderies, Irene Anindaputri Iswanto. 2022. *Systematic Literature Review: Toxic Comment Classification*. ieeexplore.ieee.org

[3] Raghvendra Mall, Mridul Nagpal, Joni Salminen, Hind Almerekhi, Soon-Gyo Jung, Bernard J. Jansen. *Four Types of Toxic People: Characterizing Online Users' Toxicity over Time*. https://dl.acm.org/doi/abs/10.1145/3419249.3420142?casa_token=O3sEJNOHJVcAAAAA:JAk37Wgzdn_2PZ1Nu0Ny802-eVUxnIJDSiowmjjbmdSaRfrBqa7NF81AChDvGz0ptegSyn_UgeFU

[4] David Stroud Sara Zaheri, Jeff Leath. 2020. *Toxic Comment Classification*. https://scholar.smu.edu/cgi/viewcontent.cgi?article=1134&context=datasciencereview

[5] Darko Androcec, *Machine learning methods for toxic comment classification: a systematic review,* researchgate.net, 2020

[6] Julian Risch, Ralf Krestel, *Toxic Comment Detection in Online Discussions*, link.springer.com, 2020

[7] Manjot Bedi, Shivani Kumar, Md Shad Akhtar, and Tanmoy Chakraborty. 2021, May 31. *Multi-modal Sarcasm Detection and Humor Classification in Code-mixed Conversations. https://arxiv.org/pdf/2105.09984.pdf*

[8] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[9] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

[10] https://www.kaggle.com/datasets/rmisra/news-headlines-dataset-for-sarcasm-detection/data

[11] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[12] https://github.com/ageron/handson-ml2