Assignment II - Kaggle Competition

Toxic Comment Classification Challenge

By Nishtha KHURANA and Irene SUNNY – Team Never Back Down, Never What!?

1. Introduction

In the digital age, online platforms have become central to our social interactions, information dissemination, and public discourse. However, this rise in digital communication brings with it the challenge of ensuring these spaces are respectful, inclusive, and free from harmful content. The Kaggle competition aimed to tackle one of the most pressing issues in this domain: **classifying the toxicity of comments**. This report outlines our approach leveraging state-of-the-art NLP models, such as BERT, Distill BERT, RoBERTa etc fine-tuned with a focus on reducing bias and improving accuracy across different demographic groups.

2. Dataset Description

The dataset comprises of training (2,69,037 rows), validation (45,179 rows), and testing (1,33,781 rows) files.

- The input files train_x, val_x_ and test_x containes index, string
- The target file provides metadata on demographic groups, toxicity categories, and the binary target variable 'y'. All these columns have binary values – 0 or 1. train_y, val_y contains 'male', 'female', 'LGBTQ', 'christian', 'muslim', 'other_religions', 'black', 'white', y (target variable).
- The other columns in target files - severe_toxicity, obscene, threat, insult, identity_attack, sexual_explicit, from_source_domain were dropped for this study due to lack of relevance.

3. Problem Statement

The goal is to classify comments as toxic (1) or non-toxic (0) while mitigating bias across different demographic groups. The traditional models often struggle with biases, inadvertently leading to the misclassification of comments related to specific identity groups. This issue is particularly acute in discussions involving sensitive demographic identities, where context and nuance play a crucial role in determining the nature of the content. The primary challenge is to build a robust classifier that ensures fairness and impartiality in comment classification.

4. Evaluation Metric

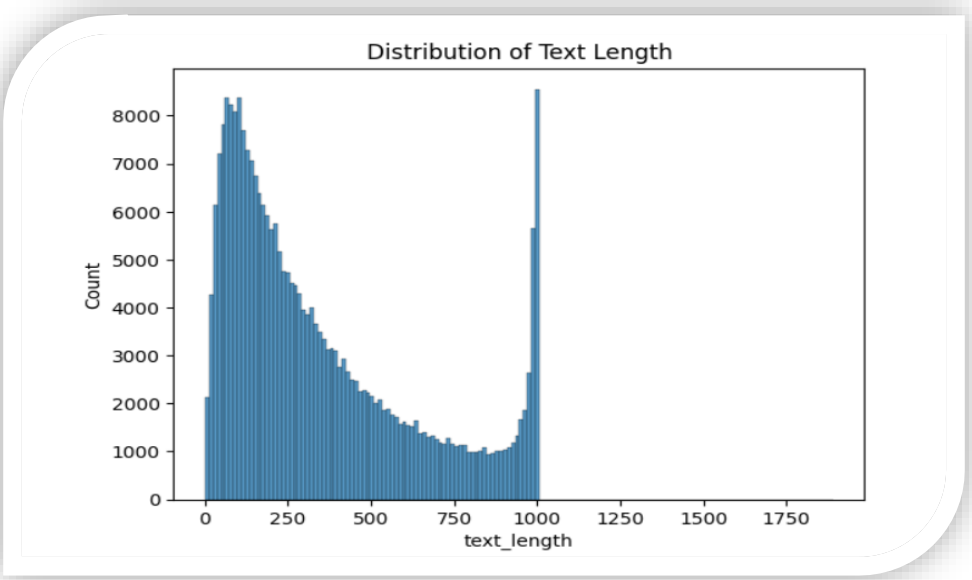
The primary metric for evaluating model performance in this competition was the **'Worst Group Accuracy' (WGA)**. This metric is designed to highlight and address potential biases in classification models. It operates on the principle that a model's performance should not only be judged by its overall accuracy but also by how well it performs on the most challenging subgroup - the "worst group."

How WGA works - The accuracy is calculated separately for each of the 16 subgroups (8 identity groups × 2 for identity being present (1) and identity being absent (0) classifications).

5. Exploratory Data Analysis (EDA)

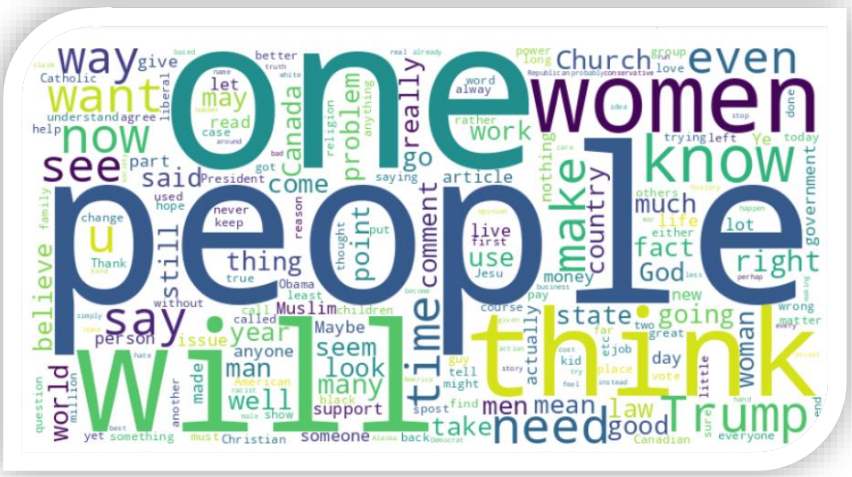
5.1 Comment Length Distribution

Upon examining the length distribution of comments, we noted a left-skewed trend featuring two distinct peaks, indicating the presence of a notable subset of comments with extended lengths.

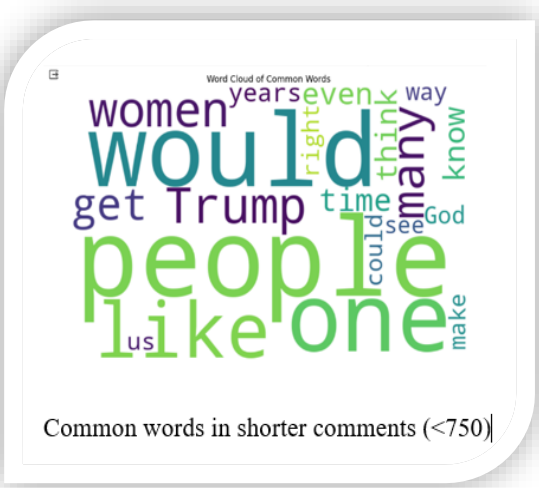


5.2 Common Words Analysis

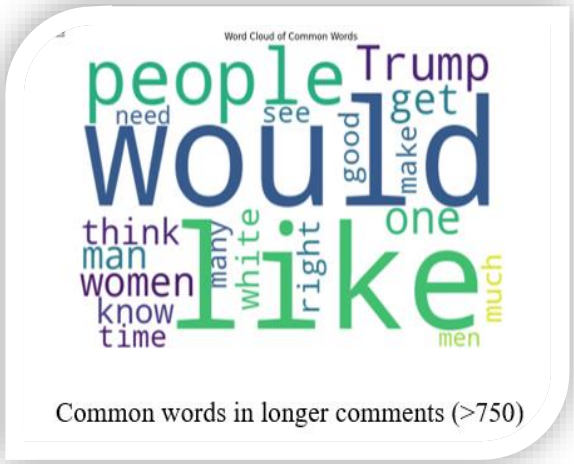
Common words in the dataset: Analyzing the dataset's frequently used words offered insights into the primary subjects of discussion. The noticeable occurrence of political terms hints at a possible bias within the dataset. Furthermore, the prevalent use of certain words like "support," "believe," and "hope" shed light on the overall sentiment or tone conveyed in the comments. This also indicated to us that transformer-based models like BERT or RoBERTa are known for their ability to understand context and sentiment, making them suitable for datasets with complex themes and varied tones.



Through further analysis for comments of different lengths, we identified common words in longer comments, but their similarity to shorter comments necessitated further analysis. By checking the correlation between the longer comments and target variable, we found no dependency. Similarly for shorter comments and target variable.



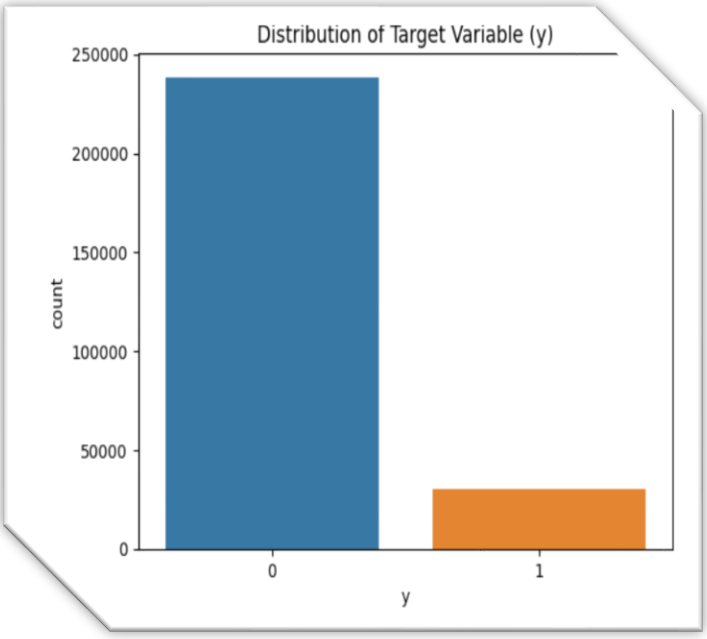
Common words in shorter comments (<750)



Common words in longer comments (>750)

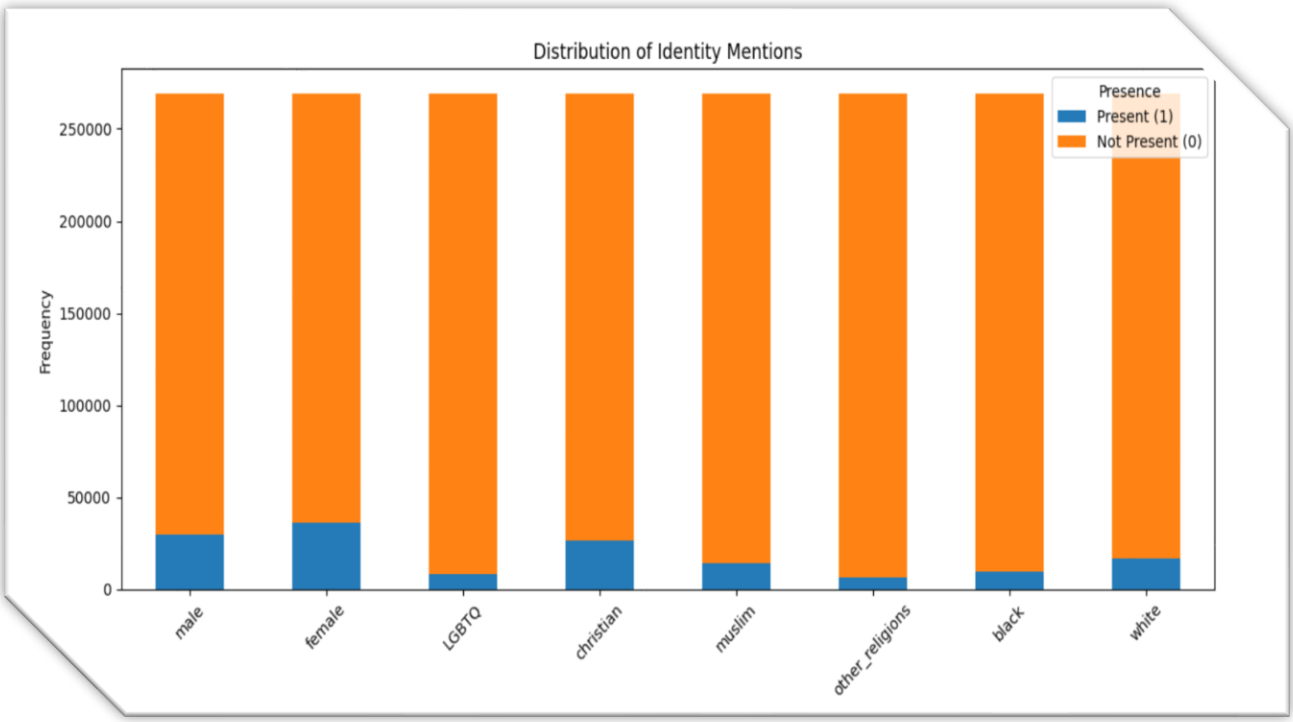
5.3 Imbalanced Dataset

The target (y) variable distribution revealed a highly imbalanced dataset. To address fairness, we examined 'y' distribution with respect to demographic variables, highlighting significant imbalances in specific groups. This will have to be addressed during modeling.



5.4 Distribution of Demographic Groups

We explored the relationship between 'y' and demographic columns. From the figure below, we confirm that proportion of presence of an identity is lower in proportion to identity not present. This creates a highly imbalanced dataset as evaluation criteria considers accuracy across the subpopulations.



5.5 Toxicity Levels vs Target Variable

We explored the relationship between 'y' and toxicity level columns, revealing a high probability of 'y' being 1 when toxicity columns are 1. By checking the probability of target variable when any of the toxic columns ('severe toxic', 'threat', 'obscene', 'insult' etc.) is present, there is a high probability of target variable 'y' to be 1 i.e. the comment is toxic. However, we decided to not consider them for our analysis as multi label classification of comments into various types can only make the model complex.

6. Modeling

At stage 1, we explored diverse models to cover a broad spectrum of NLP methodologies. While traditional methods like **Logistic Regression** offer baseline insights, advanced models like **BERT, RoBERTa, DistilBERT, and LSTM** bring sophisticated neural network capabilities to tackle the complexity of natural language in online comments. This diverse selection helped in identifying the most effective approach for the specific challenges presented by the dataset.

6.1. Data Preprocessing

6.1.1 Treatment of null values

NaN values were replaced with an empty string.

6.1.2. For BERT and Distil BERT models

- Lowercasing
- Tokenization.
- Creation of Tensor Datasets and Data Loaders, applying the AdamW optimizer.
- Adding class weights to account for the imbalanced dataset (toxic vs non-toxic comment distribution).

6.1.3. For LSTM

- Along with the data preprocessing done for BERT, for LSTM, additional preprocessing is required.
- As LSTM requires input sequences of equal length, we implemented text padding.
- Additionally, we implemented sequence encoding and created sequential input data (as LSTM requires integer encoding for each token.

6.1.4. Target Variable

Since the labels are already numerical values (0 for non-toxic and 1 for toxic), no additional preprocessing is needed for labels when using LSTM models.

6.2 Implementation

Model	Architecture Summary	Validation WGA	Kaggle WGA
Logistics Regression *(Traditional Machine Learning)*	**Feature:** Linear Classifier **Input representation:** TF-IDF Vectorisation, Stop words removed **Model Function:** Logistic Regression	0.75	0.75
BERT *(Transformer-based Deep Learning)*	**Feature:**Bidirectional Context, Uses word piece tokenisation **Input representation:** tokenised using BertTokenizer accompanied by attention masks, data loaded in batches using a Sequential Sampler **Loss Function:** BCEWithLogitsLoss **Optimiser:** AdamW **Learning Rate:** 1e-5 **Batch Size**: 64 **Epochs:** 4	0.68	0.76
Distill BERT *(Transformer-based Deep Learning)*	**Feature:** Distilled Architecture, Efficient version of BERT **Input representation:** tokenised using DistilBertTokenizer accompanied by attention masks, data loaded in batches using a Sequential Sampler **Loss Function:** BCEWithLogitsLoss **Optimiser:** AdamW **Learning Rate:** 1e-5 **Batch Size**: 64 **Epochs:** 4	0.64	0.73
LSTM *(Recurrent Neural Network)*	Feature: Long-Range Dependencies **Input representation:** tokenised comments using KerasTokeniser, created sequences and applyed padding **Loss Function:** Binary Cross Entropy **Optimiser:** AdamW **Batch Size**: 32 **Epochs:** 10	0.71	0.71
Roberta *(Transformer-based Deep Learning)*	**Feature:** Optimized Pre-training Version of BERT **Input representation:** tokenised comments using RobertaTokeniser, created sequences and applyed padding **Loss Function:** BCEWithLogitsLoss **Optimiser:** AdamW **Learning Rate:** 2e-5 **Batch Size**: 16 **Epochs:**4 **Speacial Features:** Assigning Custom Weights and custom layers, Explored Freezing Layers, Implemented Layer Wise Learning Decay	0.75	0.76

Evaluation Result of BERT model is below :

```
Validation Loss: 0.88
Worst Group Accuracy: 0.68
male_true: 0.89
male_false: 0.85
female_true: 0.89
female_false: 0.87
LGBTQ_true: 0.89
LGBTQ_false: 0.75
christian_true: 0.88
christian_false: 0.92
muslim_true: 0.89
muslim_false: 0.76
other_religions_true: 0.89
other_religions_false: 0.83
black_true: 0.90
black_false: 0.68
white_true: 0.90
white_false: 0.70
```

6.3. Model Selection and Enhancement

In selecting BERT From Stage 1 for further fine-tuning, we were guided by its superior performance in capturing bidirectional context, a crucial aspect for accurately classifying the nuanced and context-dependent nature of online comments. BERT's robust pre-training on a diverse language corpus made it a prime candidate for adaptation to our specific dataset.

6.3.1 Fine-Tuning BERT

We observed that certain identities (black, white, muslim and LGBTQ) were disproportionately represented in toxic categories due to biases in the data. The key aspect of our methodology was the calculation of custom weights based on the difference between toxic and non-toxic comments within each identity group inorder to mitigate the impact of this imbalance.

Calculating Difference Weights – Refer the results here in Appendix

The calculation involved the following steps:

1. For each demographic identity (e.g., 'male', 'female', 'LGBTQ', etc.), we counted the number of toxic and non-toxic comments in the training dataset.
2. We then calculated the difference between the counts of non-toxic and toxic comments for each group. This difference helps in understanding the imbalance in representation of each class within the group.
3. The difference_weight was calculated as the ratio of the total count of comments (toxic and non-toxic) to the difference between non-toxic and toxic counts. This weight reflects the degree of imbalance for each group.
4. A higher difference_weight indicates a greater imbalance, suggesting that the model needs to pay more attention to these groups during training to learn their characteristics better.

Assigning Weights in Training Data

1. With the difference_weights calculated, we proceeded to assign custom weights to the training data:
2. All examples in the training set were initially assigned a weight of 1.0, indicating equal importance.
3. We updated the weights for non-toxic comments in underrepresented groups using the calculated difference_weight. This adjustment was done by adding the difference_weight to the current weight of examples that belonged to each specific group and were labeled as non-toxic.
4. The focus on non-toxic comments in these groups was intentional. In cases where certain demographics are frequently mislabeled as toxic, increasing the weight of non-toxic comments in these groups helps the model learn not to overgeneralize or be biased towards predicting toxicity.
5. For specific groups like 'black' and 'white', we observed a need for even greater adjustment due to significant imbalances or misclassification rates. Therefore, we **doubled the weight** for non-toxic comments in these groups to further emphasize their importance in training.
6. For the next worse groups – muslim, LGBTQ and other religions, the **weights were multiplied by 1.5**.

Other Parameter Selection

- Additionally, we tried a learning rate of 2e-5 recommended by the original authors of BERT model, providing a balance between fast convergence and training stability.
- Reducing the batch size from 64 to 16 helped manage GPU memory constraints and improved model generalization, as smaller batches introduced more noise during training.

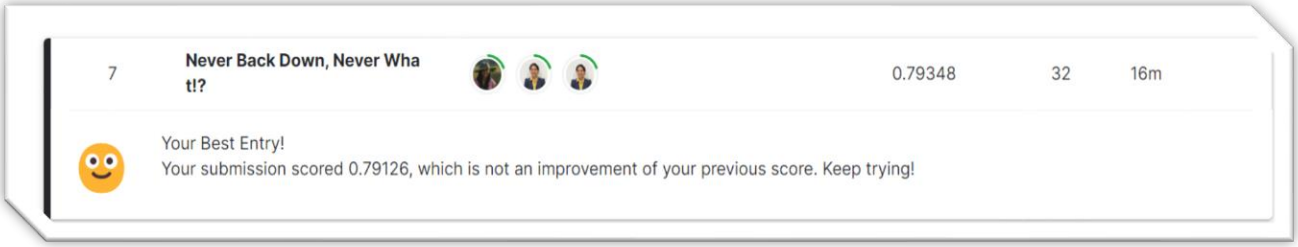
6.4 Winning Model - ENSEMBLE

In our final approach, we employed an ensemble strategy combining the initial BERT model, the fine-tuned BERT model, and a variant of the fine-tuned BERT with a raised threshold of 0.66 for positive label prediction each with differenr weightage.

This adjustment from the standard 0.5 threshold was implemented to specifically address a high rate of false positive and improve the precision metric. By being more lenient with the threshold, the model less readily classifies comments as toxic. The ensemble method leverages the strengths of each individual model, providing a more robust and accurate overall prediction system.

Model	Architecture Summary	Validation WGA	Kaggle WGA
Fine Tuned BERT	**Learning Rate:** 2e-5 **Batch Size**: 16 for train and val **Epochs:** 4 **Features**: Custom weights for each identity group	0.76	0.78
ENSEMBLE	We ensembles different combinations of different models implemented. BERT from Stage 1 (weightage = 1) + BERT Tuned (weightage = 2) + BERT Tuned variation with a positive class prediction threshold of 0.66 instead of default 0.5 (weightage = 3)	-	0.791

6.5 Leaderboard Standing



Note : *We had worked through multiple Kaggle accounts to access the Free GPU P100 and hence accidently once a submission was made through the second Kaggle Account of Irene SUNNY. Hence the team seems to have three accounts merged. This is just a two members team otherwise.*

The zipped code folder contains two files – implementation of BERT, Fine Tuned BERT and Ensemble and another file of just EDA. Both are reproducible and scalable

7. Discussion

1. Understanding and Implementing Complex Models: Navigating the complexities of advanced models like BERT and RoBERTa was another significant hurdle. These models have intricate architectures and numerous parameters, making them challenging to fine-tune and optimize.
2. GPU Allocation and Computational Constraints: One of the primary challenges faced was the limitation posed by GPU allocation. Training complex models like BERT and RoBERTa demands significant computational resources. This often required us to make strategic decisions, such as reducing batch sizes, to accommodate the extensive computational demands.
3. Dealing with a Strict Metric: Worst Group Accuracy (WGA): Adhering to the stringent requirements of the Worst Group Accuracy metric necessitated a careful and balanced approach to model training, involving the implementation of techniques like weighted loss functions to address class imbalances and potential biases within the data.
4. Experimentation and Threshold Adjustments: Moving beyond the standard 0.5 threshold to a higher value of 0.66 was a calculated decision to improve recall and reduce false negatives. This step highlighted the importance of fine-tuning not just the model parameters but also the decision boundaries based on the specific needs of the task.

5. Other attempts: Some of the implementation that did not yield expected result includes data augmentation techniques like SMOTE and synonym replacement due to memory and time constraints

8. Appendix

	Identity	Toxic_Count	Non_Toxic_Count	Total_Count	difference	\
0	male	4437	25373	29810	20936	
1	female	4962	31282	36244	26320	
2	LGBTQ	2265	6155	8420	3890	
3	christian	2446	24292	26738	21846	
4	muslim	3125	10829	13954	7704	
5	other_religions	1003	5541	6544	4538	
6	black	3111	6785	9896	3674	
7	white	4682	12016	16698	7334	

	difference_weight
0	1.423863
1	1.377052
2	2.164524
3	1.223931
4	1.811267
5	1.442045
6	2.693522
7	2.276793

Appendix 1. Assigning weights to identities as part of fine tuning

9. References

- *BERT: https://www.geeksforgeeks.org/toxic-comment-classification-using-bert/*
- *LSTM: https://github.com/simrann20/Toxic-Comment-Classification-LSTM-and-Bi-LSTM*
- *Distil BERT: https://huggingface.co/distilbert-base-uncased*
  https://huggingface.co/tillschwoerer/distilbert-uncased-finetuned-toxic-comments-detection
- *RoBERTa: https://huggingface.co/roberta-base*