Final Report
Overcoming Biases in Toxicity Models for Inclusive Conversations

94-881 Managing Analytics Projects (Spring 2023)

Ruchi Bhatia, Caroline(Yuanmo) Zhu, Nikhil Reddy Pallepati

May 03, 2023

1. Executive Summary

The rise of cyberbullying among adolescents and teenagers has emerged as a serious concern, with a significant portion of this age group reporting to have experienced or witnessed such behavior. The problem is further compounded by the presence of political bullying in elections, which promotes harmful tactics and contributes to a rise in bullying in schools. Despite the existence of anti-cyberbullying laws, schools continue to face difficulties in responding to cyberbullying incidents due to a lack of clear guidance.

To address this issue, one potential solution is to develop a Machine Learning model that can effectively detect toxicity in online school forums. Manual monitoring by moderators is inefficient, prone to bias, and unsuitable for the increasing volume of comments. By utilizing contextual level machine learning models and natural language processing analytics, schools can automate the process of identifying and filtering out toxic comments, and gain valuable insights on the most common toxic topics to inform policy decisions. This approach can lead to the creation of a safer and more inclusive online environment for students.

2. Problem framing

Cyberbullying affects individuals of all ages, but it is particularly prevalent among adolescents and teenagers aged 12-17. Research reveals that 21% of tweens (9-12 years old) have experienced cyberbullying, cyberbullied others, or witnessed cyberbullying. Furthermore, 25% of students report experiencing cyberbullying through mean and hurtful comments, while 33% of adolescents report experiencing bias-based school bullying.[1]

The pervasive presence of political bullying in elections exacerbates this problem, negatively impacting children and leading to increased bullying at school, fear, anxiety, and prompting kids to mimic the behavior they witness from political leaders. Exposure to blame-shifting, name-calling, reputation-bashing, rumor spreading, and veiled threats by politicians teaches children that such behaviors are socially acceptable, ultimately normalizing these harmful tactics. Research indicates that children are more likely to engage in uncivil political discussions and harassment in school as a result of witnessing political bullying. Moreover, this exposure can foster social mistrust, diminishing a child's faith in people and society.[2]

Although many states have passed anti-cyberbullying laws in the past decade, there is a lack of specific guidance from states about how and when schools can take action in cyberbullying incidents. This dilemma has put many schools in a difficult position of having to respond to a mandate to have a cyberbullying policy without much guidance from the state about the circumstances under which they can or must respond.[3]

One of the solutions to this problem can be achieved with the help of technology. By developing a ML model that can detect toxicity in online school forums, educational institutions can proactively identify and respond to cyberbullying incidents, improving the overall safety and well-being of students.

Schools may have a manual process for monitoring the forums, which involves moderators manually reviewing each comment and deciding whether it violates the school's code of conduct. However, this process is time-consuming, inefficient, and often subject to the biases and interpretations of the moderators. Moreover, this approach does not scale well with the increasing volume of comments in the forums, making it impractical to maintain over the long term.

Therefore, by exploring data-driven solutions schools can automate the process of detecting and filtering out toxic comments while minimizing unintended bias towards certain groups. By leveraging the power of machine learning and natural language processing, schools can create a safer and more inclusive online environment for their students to learn and grow.

Unintended biases in machine learning models can lead to unfair treatment of certain groups or identities. Here are a few examples:
- **Racial Bias**: A model may associate negative sentiments with certain racial or ethnic groups, incorrectly classifying non-toxic comments as toxic. For example, a comment like "I am a proud African-American" might be flagged as toxic if the model has learned to associate "African-American" with negativity due to imbalanced training data.
- **Gender Bias**: Gender-related terms may also trigger unintended biases in the model. A statement like "As a successful female CEO, I strive to empower and mentor young women in the business world" could be incorrectly labeled as aggressive or self-promoting if the model has learned to associate the terms "female CEO" with negative connotations.

Similar issues can occur with **Political** and **Religious** statements as well. These unintended biases occur because the models are trained on real-world data, which often contains imbalanced representations of different groups or identities. Consequently, the models may learn to associate certain identities with toxicity, even when the comments are not inherently offensive.

To mitigate such biases, we will be using techniques that can balance the representation of different groups in the training data, as well as to employ evaluation metrics that specifically measure and penalize unintended biases. This will lead us to ultimately create a solution that fosters a healthier online environment for students.

1. **Decision to be improved**: How to accurately identify and filter toxic comments in online school forums while minimizing unintended bias towards certain identities, with a goal to automate the process.
2. **Decision-maker**: School administrators, forum moderators, and an AI system.
3. **Value of Improved Decision**: The value of improving this decision includes the following outcomes:
   a. **Quantitative**:
      i. Reduction in the number of toxic comments on the school forums, improving the overall discourse quality and student well-being.
      Ii. Reduction in the number of false positives/negatives, ensuring students with diverse backgrounds are treated fairly and not inadvertently censored.
      iii. Reduction in the time required to review and moderate comments, allowing moderators to focus on more strategic initiatives.
   b. **Qualitative**:
      i. Improved sense of safety and inclusion for all students in online discussions, irrespective of their identities.
      ii. Encouraging open and respectful conversations among students, promoting better learning outcomes and social interaction.
      iii. Enhanced trust and transparency in the moderation process, as the use of AI algorithms can provide a consistent and objective approach to comment review.

Business Requirements
- The model should minimize unintended biases while detecting and filtering toxic comments.
- The model should reduce the workload of humans & increase efficiency and productivity.
- The model should be scalable and able to handle a large volume of comments.

Technical Requirements
- The model training should have a labeled dataset of toxic comments for training the machine learning algorithms.
- The model training should use ML algorithms and NLP techniques to identify and classify toxic comments.

Assumptions
- **Data**: The school has access to a sufficient amount of labeled data for training the machine learning algorithms.
- **Staff**: The school has the technical expertise to integrate the system with its existing online forum platform.
- **Technology**: The school possesses the necessary resources to maintain and update the system provided by our team, ensuring optimal performance.

Hypothesis
- The implementation of the toxicity detection model will enable the school to identify comments that attack identities (based on their race, gender, religion, etc.) effectively
- The target toxicity score of comment texts is dependent on various features, such as text content, vocabulary, and contextual information.
- The toxicity target score of comment texts is determined by the severity of toxicity(obscene, identity_attack, insult, threat, race, gender, religion, etc.) and identity attack scores('muslim','black', 'psychiatric_or_mental_illness', etc.).

3. Initial choices for Version 1

3.1 Data Exploration

3.1.1 Data Source

The selected data source for this study is the **Jigsaw Unintended Bias in Toxicity Classification** dataset, obtained from the online platform Kaggle. The dataset is composed of 1,804,874 rows and 45 columns, featuring various attributes such as comment id, comment text content, toxicity score, toxicity category scores, identity targeted scores, and sentiment scores. The scores in this dataset range from 1 to 0, with larger values indicating a higher level of toxicity severity and smaller values indicating a lower level of toxicity severity.

In considering potential data sources for the model training, the **Toxic Comment Classification Challenge** dataset was also evaluated. This dataset comprises 159,572 rows and 8 columns, including attributes such as id, comment text, toxicity score, and toxicity category scores. However, it should be noted that all scores in this dataset are binary, with a score of 1 indicating toxicity and 0 indicating non-toxicity.

When evaluating data quality, it is important to consider potential biases that may be present in the dataset. For instance, in the Jigsaw dataset, some identity groups may have more data than others, which could potentially impact model performance on those groups. On the other hand, in the Toxic Comment dataset, since the data was labeled by human annotators, there may be variability in the labeling standards used, which may impact the quality of the dataset.

While both datasets are free to download, there may be additional costs associated with their use. For example, training a model on the dataset may require significant computational resources, resulting in high training costs, particularly for larger models or longer training times.

In our decision-making process, we prioritized data quality, recognizing that high-quality data is critical to achieving good model training results:

- Enough data records from one single dataset to train the model.

  The Jigsaw Unintended Bias in Toxicity Classification dataset has more than 1 million data records. We assume this amount will currently be enough for us to train our model.

- Less informational columns provided in the other dataset

  Compared with the Jigsaw Unintended Bias in Toxicity Classification dataset which includes identity attack scores, the other dataset only contains comment text column and toxicity score column. We believe that these identity columns will play an important role in further model training and thus we decided not to go with incorporating the other dataset.

Therefore, we decided to move forward with one single dataset: **Jigsaw Unintended Bias in Toxicity Classification**.

3.1.2 Data Preparation

To facilitate subsequent analysis and model training, the first step undertaken was the removal of extraneous data columns, specifically comment_id, publication_id, parent_id, and article_id.

Furthermore, it was observed that null values were present solely in the identity score columns. After reviewing a sample of these comments, it was determined that these null values were indicative of comments that did not target any particular identity. Consequently, a decision was made to substitute these null values with 0, to facilitate data analysis and model training.
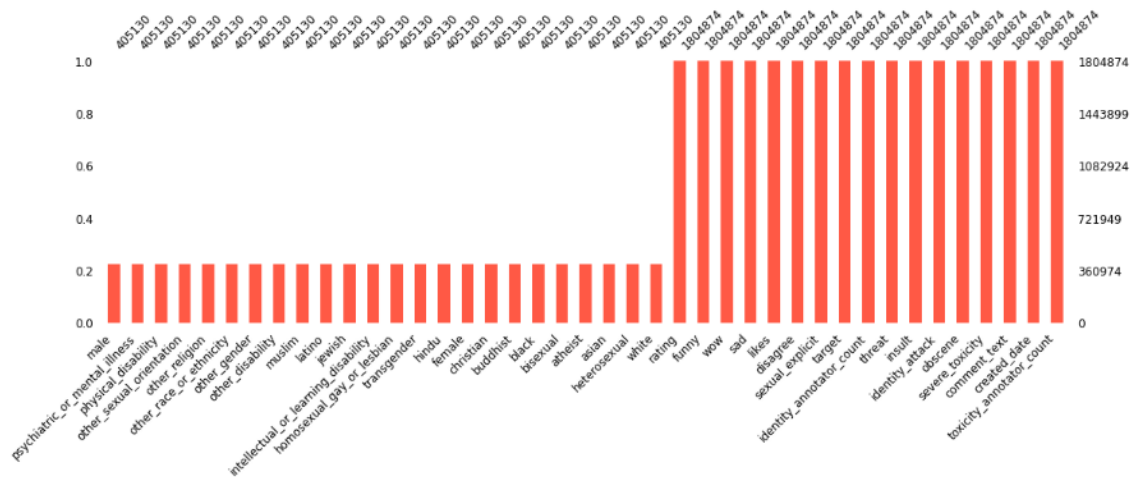
Figure 1

In order to identify outliers in the dataset, a series of statistical analyses were conducted, including summary statistics and univariate analysis. Two examples of univariate analysis are presented below.

It was observed that a large number of values in the dataset were recorded as 0. This can be attributed to the presence of numerous non-toxic comments. The prevalence of these 0 values can make other values in the dataset appear to be outliers. However, it was determined that each value represented a distinct evaluation of the toxicity of each comment text. As such, it was decided that all values should be retained rather than removing or replacing the outliers with column means or medians. This decision reflects a priority for data and training quality over any potential costs associated with a large dataset.
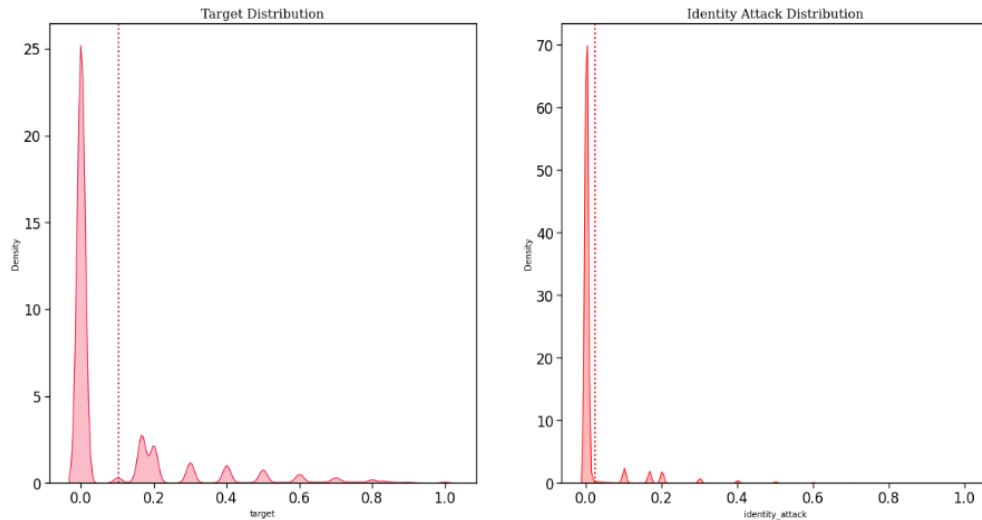
Figure 2

Preprocessing steps were also performed on comment text for further model training and comment analysis. Firstly, all text was converted to lowercase, followed by tokenization of all words. Subsequently, all stop words were removed, and lemmatization was performed. An example of the comment text before and after preprocessing is presented below:

Before preprocessing:

Canada is north of the USA border, its colder in Canada.

After preprocessing:

canada north usa border colder canada

Figure 3

3.2 Analysis Choices and Justification

The analytics problem type that matches this decision is a classification problem. The main goal is to classify comments as toxic or non-toxic in the online school forums while minimizing unintended bias towards certain identities. The problem involves creating an automated process to improve efficiency and reduce human biases in the moderation process.

Several analytics techniques can be considered for this classification problem, such as:

3.2.1 Word-Level Classification Techniques

By considering word frequencies and importance in the text, supervised learning methods can utilize word-level features, such as term frequency-inverse document frequency (TF-IDF) representations, to achieve accurate and coherent classification results.

In this study, we also evaluated the following classical machine learning algorithms for toxic text classification, as they exhibit various strengths in handling text data:

a. **Logistic Regression**: A straightforward and interpretable linear model that works effectively for binary classification tasks. It primarily takes into account individual words and their associations with the target variable, making it suitable for toxic text classification.

b. **Naive Bayes**: A probabilistic classifier based on Bayes' theorem, which assumes independence between features. Despite this assumption, Naive Bayes is known to perform well in text classification tasks due to its simplicity and efficiency.

c. **MLP (Multilayer Perceptron)**: An artificial neural network model that can capture nonlinear relationships between features and the target variable. MLPs are particularly useful for handling high-dimensional input data, such as text features.

d. **Linear SVC** (Support Vector Machines): A method that seeks the optimal separating hyperplane to classify data points in higher-dimensional spaces, providing good generalization performance. Linear SVC is especially effective in handling sparse data, which is often the case in text classification problems.

e. **Decision Tree**: A tree-like structure that recursively splits the data based on feature values, resulting in a series of decision rules. Decision trees can naturally handle both continuous and categorical data, making them suitable for text classification tasks.

f. **Random Forest**: An ensemble technique that combines multiple decision trees, making it more robust against overfitting and capable of handling complex relationships in the data. Random Forests can capture non-linear patterns and are less sensitive to noise, which is beneficial for toxic text classification.

g. **KNeighbors**: A non-parametric, instance-based learning algorithm that classifies new instances based on their similarity to the nearest neighbors in the training data. KNeighbors can be effective for text classification tasks, as it does not make strong assumptions about the data distribution and can adapt to local patterns.

The process we followed is described below:
- **Preprocess and vectorize text-data using TfidfVectorizer**: The raw text data was preprocessed by removing unnecessary characters, converting text to lowercase, tokenizing, and stemming or lemmatizing words. Then, the TfidfVectorizer was used to convert the text data into a numerical format. Term Frequency-Inverse Document Frequency (TF-IDF) is a popular technique for text vectorization that takes into account both the frequency of words in a document and their importance across the entire corpus.
- **Split data into training and validation sets**: The dataset was split into training and validation sets, with a ratio of 80:20, to ensure that the models could be evaluated on unseen data after training.
- **Model training**: The selected machine learning algorithms were trained on the preprocessed and vectorized text data from the training set.
- **Evaluate model on validation data**: After training, each model's performance was evaluated on the validation set to estimate its generalization capability on unseen data.

The performance of the models, as measured by the final metric, was relatively poor:

Models	Final Metric
Logistic Regression	0.5027
Naive Bayes	0.4879
MLP	0.4975
Linear SVC	0.5015
Decision Tree	0.4478
Random Forest	0.4341
KNeighbors	0.5028

The reasons for the subpar performance could be:
- **Insufficient feature representation**: TF-IDF vectorization may not capture the context or semantic meaning of words effectively, leading to a weak representation of the text data.
- **Inadequate hyperparameter tuning**: The models' performance might be limited due to suboptimal hyperparameter settings. A more comprehensive search for the best hyperparameters could improve the results.
- **Limited model capacity**: Classical machine learning models might not be suitable for capturing the intricacies of natural language data, as compared to more advanced models like BERT or RoBERTa.

To improve the performance, we could consider using more advanced text representations like word embeddings (e.g., Word2Vec or GloVe), experimenting with different preprocessing techniques, handling class imbalance, performing hyperparameter tuning, or even trying more sophisticated models like deep learning-based text classifiers.

3.2.2 Context-Level Classification Techniques

These models capture contextual information, taking into account not only individual words but also their relationships with surrounding words or phrases in a sentence or text. Preprocessing is not needed when using pre-trained language representation models like BERT. In particular, it uses all of the information in a sentence, even punctuation and stop-words, from a wide range of perspectives by leveraging a multi-head self attention mechanism.

Deep Learning
h. **CNN**: Convolutional Neural Networks can be trained using pre-trained GloVe word embeddings as input for text classification tasks, capturing some semantic and syntactic information about words.

The process we followed for training a convolutional neural network model is described below:
- **Preprocess and tokenize text data**: The raw text data was preprocessed and tokenized using a Keras Tokenizer. The tokenizer was fitted on the training dataset, and the texts were converted into sequences of word indices.
- **Pad text sequences**: All text sequences were padded or truncated to ensure they have the same length, which is necessary for input to the neural network.
- **Split data into training and validation sets**: The dataset was split into training and validation sets, with a ratio of 80:20, to ensure that the models could be evaluated on unseen data after training.
- **Load pre-trained word embeddings**: GloVe word embeddings were loaded from a file, and an embedding matrix was created to be used in the model, mapping each word index to its corresponding embedding vector.
- **Build and compile the model**: A convolutional neural network model was constructed with the input and output layers, using pre-trained word embeddings, convolutional and max-pooling layers, dropout, and dense layers. The model was compiled using categorical_crossentropy loss, and RMSprop optimizer.

```
Model: "model"
_____
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 250)]             0

 embedding (Embedding)       (None, 250, 100)          4333800

 conv1d (Conv1D)             (None, 250, 128)          25728

 max_pooling1d (MaxPooling1D  (None, 50, 128)          0
 )

 conv1d_1 (Conv1D)           (None, 50, 128)           49280

 max_pooling1d_1 (MaxPooling  (None, 10, 128)          0
 1D)

 conv1d_2 (Conv1D)           (None, 10, 128)           65664

 max_pooling1d_2 (MaxPooling  (None, 1, 128)           0
 1D)

 flatten (Flatten)           (None, 128)               0

 dropout (Dropout)           (None, 128)               0

 dense (Dense)               (None, 128)               16512

 dense_1 (Dense)             (None, 2)                 258

=================================================================
Total params: 4,491,242
Trainable params: 157,442
Non-trainable params: 4,333,800
_____
```

Figure 4

- **Model training**: The model was trained on the preprocessed, tokenized, and padded text data from the training set, with specified batch size and number of epochs.
- **Evaluate model on validation data**: After training, the model's performance was evaluated on the validation set to estimate its generalization capability on unseen data.

```
---------- Model Performance: CNN ----------

                         subgroup  subgroup_size  subgroup_auc  bpsn_auc  bnsp_auc
2       homosexual_gay_or_lesbian             52      0.653409  0.772581  0.765403
5                          muslim            124      0.688150  0.778919  0.792759
3                       christian            211      0.731935  0.889428  0.669406
4                          jewish             39      0.735714  0.679964  0.915166
6                           black             93      0.775776  0.693163  0.893339
1                          female            282      0.798221  0.796801  0.845759
7                           white            140      0.803355  0.683475  0.909801
8   psychiatric_or_mental_illness             22      0.835294  0.868578  0.804440
0                            male            209      0.861640  0.799534  0.887842
Final Metric: 0.7945709845269946
```

Figure 5

The performance of the CNN, as measured by the final metric (0.7945), was better than supervised models but still not optimal. GloVe embeddings are limited in their ability to capture the full context of a sentence due to their static nature, assigning a fixed vector to each word regardless of context. While GloVe can capture general relationships between words, it may struggle to represent contextual nuances, potentially leading to suboptimal performance in classification tasks.

i. **BERT, RoBERTa, DistilBERT**: These are pre-trained language models based on transformer architectures, which have achieved state-of-the-art performance on various natural language processing tasks, including text classification. They can capture deep contextual information and accurately understand language. Their strengths lie in bidirectional context processing, extensive pretraining, fine-tuning capabilities, the Transformer architecture, and scalability. These models excel at disambiguating word meanings, handling polysemy, and capturing complex patterns within text, resulting in improved performance on various text classification problems.

We fine-tuned various pre-trained models, including BERT, RoBERTa, and DistilBERT, for the task of toxic comment classification. The fine-tuning process involved the following steps:

- We loaded the pre-trained models from the Transformers library using the specified model name.
- A dropout layer with a rate of 0.3 and a linear layer were added to the models for binary classification (toxic or non-toxic).
- The entire models, including the pre-trained components and the added layers, were trained end-to-end on the toxic comment classification dataset.

For the training process, we employed the AdamW optimizer with weight decay and a learning rate of 3e-5. The weight decay was set at 0.001 for non-bias and non-LayerNorm parameters, while the bias and LayerNorm parameters used a weight decay of 0.0. Furthermore, we utilized a linear learning rate scheduler with warmup, which adjusted the learning rate throughout the training process to facilitate model convergence.
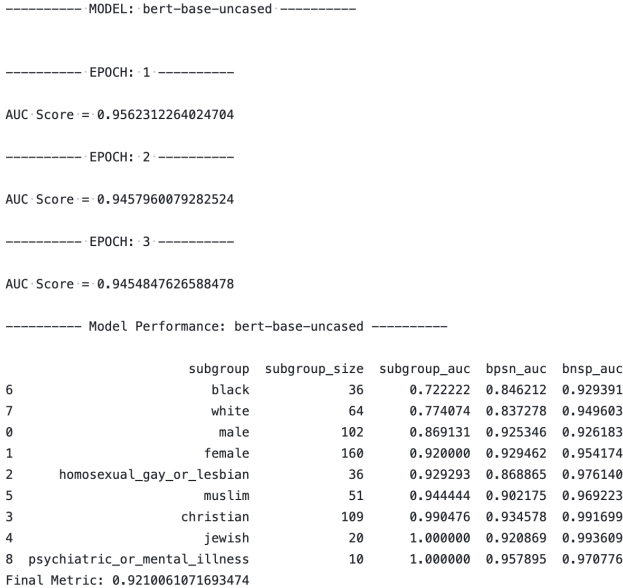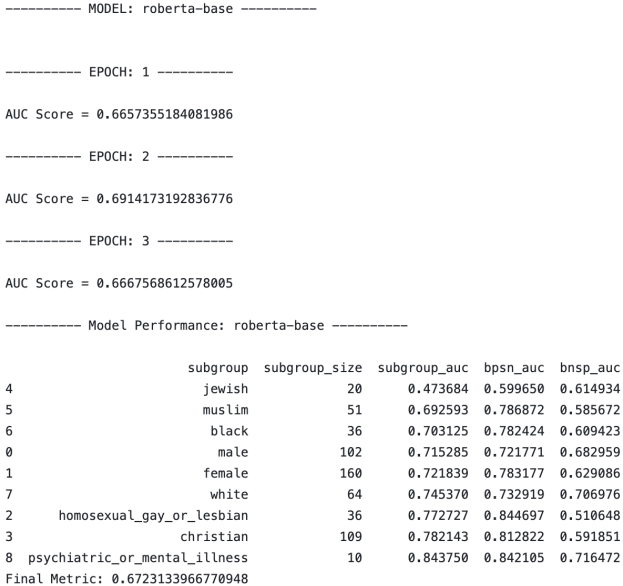
```
---------- MODEL: bert-base-uncased ----------


---------- EPOCH: 1 ----------

AUC Score = 0.9562312264024704

---------- EPOCH: 2 ----------

AUC Score = 0.9457960079282524

---------- EPOCH: 3 ----------

AUC Score = 0.9454847626588478

---------- Model Performance: bert-base-uncased ----------


                         subgroup  subgroup_size  subgroup_auc  bpsn_auc  bnsp_auc
6                           black             36      0.722222  0.846212  0.929391
7                           white             64      0.774074  0.837278  0.949603
0                            male            102      0.869131  0.925346  0.926183
1                          female            160      0.920000  0.929462  0.954174
2       homosexual_gay_or_lesbian             36      0.929293  0.868865  0.976140
5                          muslim             51      0.944444  0.902175  0.969223
3                       christian            109      0.990476  0.934578  0.991699
4                          jewish             20      1.000000  0.920869  0.993609
8   psychiatric_or_mental_illness             10      1.000000  0.957895  0.970776
Final Metric: 0.9210061071693474
```

Figure 6 BERT

```
---------- MODEL: roberta-base ----------


---------- EPOCH: 1 ----------

AUC Score = 0.6657355184081986

---------- EPOCH: 2 ----------

AUC Score = 0.6914173192836776

---------- EPOCH: 3 ----------

AUC Score = 0.6667568612578005

---------- Model Performance: roberta-base ----------


                         subgroup  subgroup_size  subgroup_auc  bpsn_auc  bnsp_auc
4                          jewish             20      0.473684  0.599650  0.614934
5                          muslim             51      0.692593  0.786872  0.585672
6                           black             36      0.703125  0.782424  0.609423
0                            male            102      0.715285  0.721771  0.682959
1                          female            160      0.721839  0.783177  0.629086
7                           white             64      0.745370  0.732919  0.706976
2       homosexual_gay_or_lesbian             36      0.772727  0.844697  0.510648
3                       christian            109      0.782143  0.812822  0.591851
8   psychiatric_or_mental_illness             10      0.843750  0.842105  0.716472
Final Metric: 0.6723133966770948
```

Figure 7 RoBERTa

```
---------- MODEL: distilbert-base-uncased ----------


---------- EPOCH: 1 ----------

AUC Score = 0.9562896310967292

---------- EPOCH: 2 ----------

AUC Score = 0.9449628169101552

---------- EPOCH: 3 ----------

AUC Score = 0.9356225016283083

---------- Model Performance: distilbert-base-uncased ----------

                           subgroup  subgroup_size  subgroup_auc  bpsn_auc  bnsp_auc
6                             black             36      0.812500  0.860606  0.950842
7                             white             64      0.842593  0.869568  0.953896
0                              male            102      0.906094  0.929049  0.945203
2          homosexual_gay_or_lesbian            36      0.909091  0.860329  0.976425
1                            female            160      0.923218  0.925913  0.960829
5                            muslim             51      0.944444  0.909846  0.969973
3                          christian            109      0.980952  0.945448  0.984158
4                            jewish             20      1.000000  0.918844  0.984448
8      psychiatric_or_mental_illness            10      1.000000  0.959649  0.981403
Final Metric: 0.9346879749831075

The best performing model is distilbert-base-uncased with a final metric of 0.9346879749831075
```

Figure 8 DistilBERT

It is important to note that the RoBERTa model underperformed in comparison to the other two models, BERT and DistilBERT. This outcome could be attributed to several factors, such as suboptimal hyperparameters, tokenization differences, or training duration. Consequently, future work could involve experimenting with various hyperparameters, or adjusting the fine-tuning strategy to improve the model's performance.

3.3 Visualizations and Communicating the Analytics

As the primary objective is to effectively classify comments as either toxic or non-toxic in the online school forums, while ensuring minimal unintended bias towards specific identities, utilizing analytics results that include top unigrams, bigrams and word cloud, fields relevant to the target score, and the best machine learning model with the highest final metric can significantly aid schools in making informed decisions on how to create conversation policies for the online forum, train their toxicity and identity attack detection model, and choose an accurate model for implementation.

a. Top keywords in comment text:
School stakeholders can effectively alleviate discrimination and toxicity situations by closely monitoring the top keywords in comment text that may generate biases or identity attacks. As mentioned above in the problem framing part, research indicates that children are more likely to engage in uncivil political discussions and harassment in school as a result of witnessing political bullying. In the current dataset, our comment analysis has revealed that "trump" and "donald trump" are frequently mentioned and associated with identity attacks and biases among the top unigrams and bigrams, aside from neutral words. With this information, the school's principal and other stakeholders can implement appropriate rules for conversations related to politics in the online forum.

Below are some sample toxic comments mentioning "trump":
```
comment: Trump:  "I'll build a wall on the Alaska border too.  It will be tall, beautiful, and long.  And it will k
eep all those Canadian beer drinkers out of our country, especially if they are Muslim."

"And Pierre's kid is gonna pay for it, too."
```
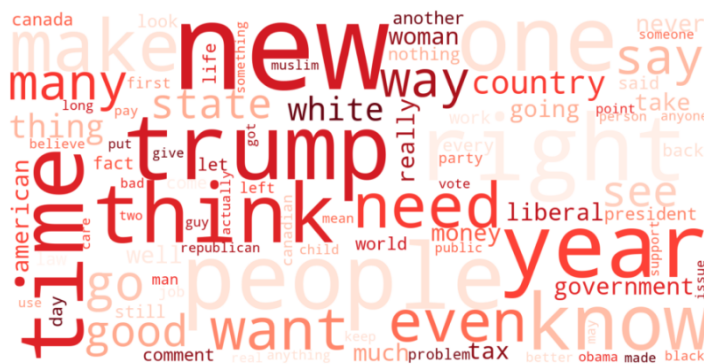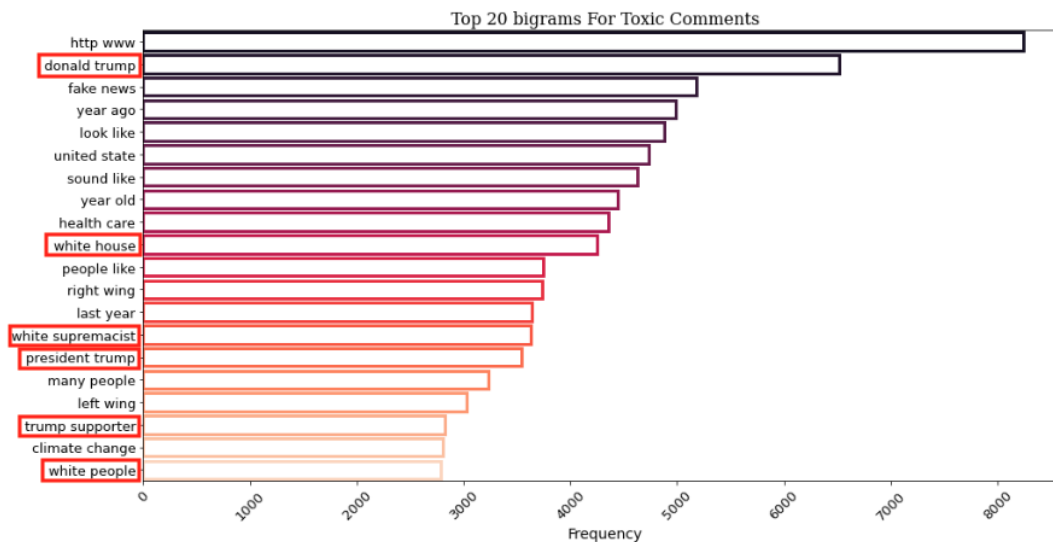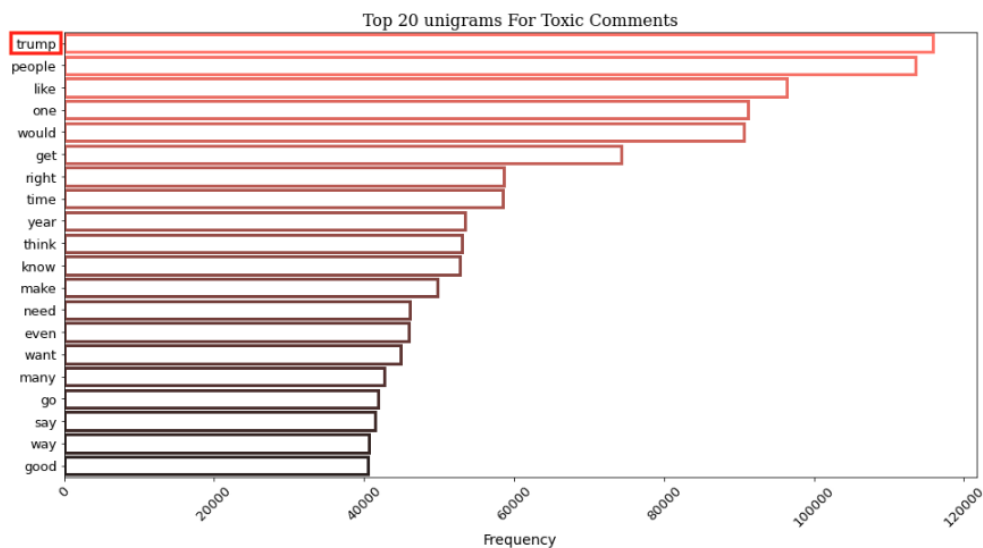**Figure 9**
```
comment: No, she was the one who said this, "Looking at a young black woman in the group, I told her: 'I would rath
er be a black man in America than a black woman in Africa.' "
```
Figure 10
```
comment: I guess half of Trump supporters don't deserve to vote. Bitch.
```
Figure 11

Figure 12

Figure 13

Figure 14

b. Columns highly relevant to the toxicity target score:

After conducting the bivariate analysis, we found out some attributes that were highly relevant to our target variable. By taking all sub groups columns of toxicity such as severe_toxicity, obscene, identity_attack, insult and threat into consideration when

training the model, we can achieve better model performance. This provides valuable guidance for school stakeholders when training their machine learning model for the school online forum on a larger dataset. By training the model with relevant input columns, stakeholders can increase model efficiency and performance.
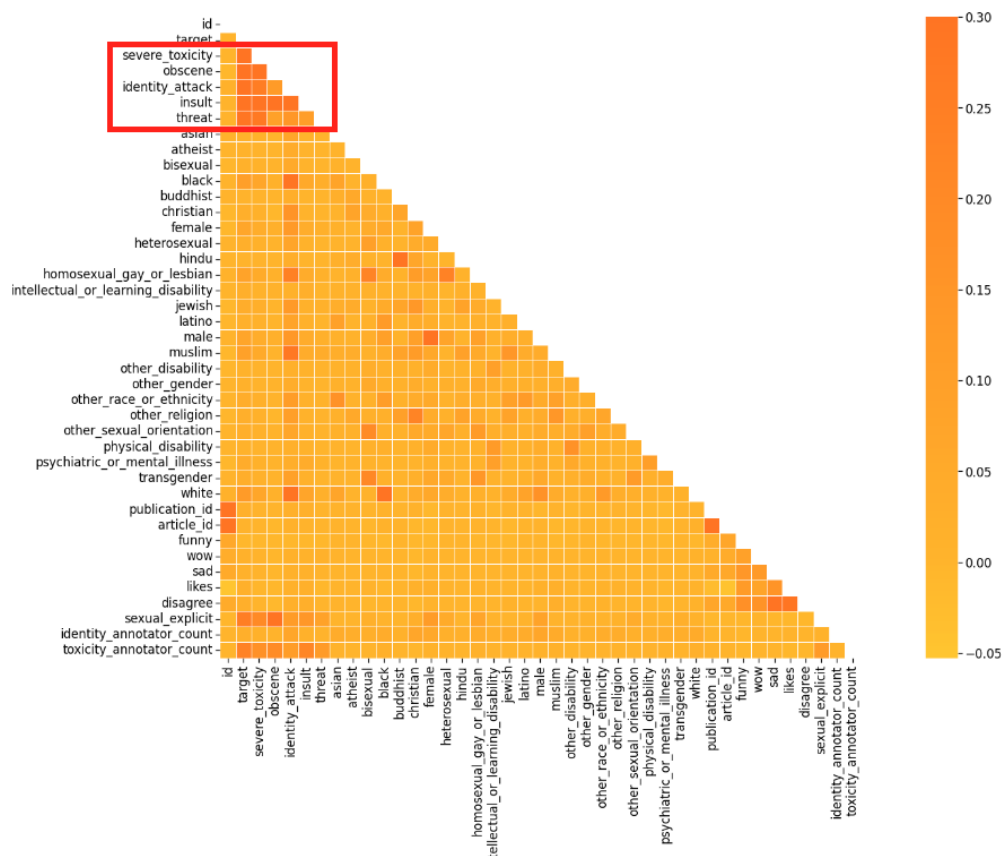
Figure 15

c. Best Model with highest final metric value:

The selection of an appropriate machine learning model is crucial for a toxicity classification problem, as many model choices exist. By training all relevant models and comparing their performance based on final metric values and AUC values for each subgroup (check the Appendix at the end of this document for more detail about the metrics), stakeholders can choose the most accurate model for further training on a larger dataset. Our analysis has identified that distilbert-base-uncased is the best model choice to continue the further work.

```
────────── MODEL: distilbert-base-uncased ──────────


────────── EPOCH: 1 ──────────

AUC Score = 0.9562896310967292

────────── EPOCH: 2 ──────────

AUC Score = 0.9449628169101552

────────── EPOCH: 3 ──────────

AUC Score = 0.9356225016283083

────────── Model Performance: distilbert-base-uncased ──────────

                      subgroup  subgroup_size  subgroup_auc  bpsn_auc  bnsp_auc
6                        black             36      0.812500  0.860606  0.950842
7                        white             64      0.842593  0.869568  0.953896
0                         male            102      0.906094  0.929049  0.945203
2    homosexual_gay_or_lesbian             36      0.909091  0.860329  0.976425
1                       female            160      0.923218  0.925913  0.960829
5                       muslim             51      0.944444  0.909846  0.969973
3                    christian            109      0.980952  0.945448  0.984158
4                       jewish             20      1.000000  0.918844  0.984448
8  psychiatric_or_mental_illness           10      1.000000  0.959649  0.981403
Final Metric: 0.9346879749831075

The best performing model is distilbert-base-uncased with a final metric of 0.9346879749831075
```

Figure 16

d. Other relevant information like cost and accuracy of the decision:
The cost can include:
- the financial cost of training the contextual models like transformer models on lots of training data (thereby utilizing lots of compute power),
- implementing policies or models, as well as
- the cost of potential legal action if discrimination or bias is not properly addressed.

The accuracy of the decision-making process here mainly involves model choices. As mentioned in part c, this can be measured by evaluating the model performance with final metric values (check the Appendix at the end of this document for more detail about the metrics), as well as analyzing the performance for different sub-groups to ensure that biases are minimized.

Stakeholders should aim to choose a model that balances both the accuracy and cost of implementation, while ensuring that the model is fair and does not discriminate against any specific group.

4. Results and recommendations

4.1 Results from our Analysis

The analysis has revealed several key findings, as outlined above. Notably, the most prevalent topic in toxic comments was found to be "trump." Furthermore, our dataset showed that the attributes most relevant to targeting toxicity scores are primarily related to toxicity category scores.

The highest performing model was found to be a contextual level model, specifically the distilbert-base-uncased model. This model demonstrated the strongest final metric values in our analysis.

4.2 Hypothesis Testing Results

These findings have confirmed our previous hypothesis regarding the importance of text content, vocabulary, and contextual information in determining the target toxicity score of comment texts. Our analysis supports the hypothesis that considering severity of toxicity(obscene, identity_attack, insult, threat, race, gender, religion, etc.) is an important factor in assessing toxicity.

Interestingly, we did not find a strong correlation between toxicity scores and identity attack scores('muslim','black', 'psychiatric_or_mental_illness', etc.) in the data.

4.3 Model Deployment

We also deployed our model using Streamlit so the toxicity of a sentence can be assessed using our distilbert-base-uncased model.

Figure 17

4.4 Recommendations

Firstly, due to the high prevalence of toxic conversations surrounding political topics, specific rules should be created to govern political discussions in the school's online forum. For instance, school administrators could allow political discussions but prohibit any form of identity attack in the content.

Secondly, we recommend incorporating positive placeholder text in the comment or post box. This approach could encourage students to post positive content and be mindful of their language choices.

Figure 18

Lastly, we suggest integrating the deployed Streamlit model with the forum backend to automatically detect toxicity and prevent students from posting by disabling the submit button. This could significantly reduce the number of toxic comments in the forum and promote healthy online interactions among students.

14

Figure 19

5. Project plan for Version 2

5.1 Stages, Milestones and Deliverables

	Stage 1: Question Formulation
Milestones	- Stakeholders agreed on a question that clearly (a) states the problem to be solved as a decision (choice among alternative actions) to be improved; (b) identifies the decision-maker; and (c) describes in measurable terms the value of improving the decision. - School receives and approves project proposal.
Deliverables	Project proposal for approval by the school.

	Stage 2: Data Exploration
Milestones	- Acquired relevant data sources for toxicity detection - Clean and preprocessed data to ensure quality and compatibility
Deliverables	- Relevant data sources - Cleaned and preprocessed dataset ready for model development - Data documentation (dataset descriptions, metadata, etc.) - Data brief with summary statistics - Data visualizations

	Stage 3: Data Modeling and Analytics
Milestones	- Train supervised models and transformer models for toxicity detection - Iteratively validate and refine models - Select final model based on final metric - Build deployment pipeline
Deliverables	- Final model - Model evaluation metrics - Deployment pipeline

	Stage 4: Interpretation
Milestones	- Translate Model's outputs into insights / recommendations that are understandable and actionable for program manager and other relevant stakeholders - Perform cost benefit analysis on models and proposed interventions - Build data visualizations - Meet with stakeholders - Determine recommendations
Deliverables	- Cost benefit analysis results - Data visualizations - Final recommendations

	Stage 5: Communication and Next Steps
Milestones	- Share deployable model with the school - Provide documentation for implementation - Deliver presentation to stakeholders - Provide final report
Deliverables	- Deployable model with documentation - Presentation - Final Report

5.2 Resources Required

5.2.1 Hardware and Software Resources

Hardware and Costs
a. Computers for staff: No additional cost (already available)
b. Cloud-based storage and compute: Approx. $500 per month (e.g., Google Cloud Platform or Amazon Web Services)
c. GPU-accelerated machine: Approx. $2,500 (one-time purchase)

Software and Costs
a. Machine learning frameworks and libraries (e.g., PyTorch, scikit-learn): Free and open-source
b. Natural language processing libraries (e.g., nltk, spaCy): Free and open-source
c. Data visualization libraries (e.g., Matplotlib, Seaborn, Plotly): Free and open-source
d. Integrated Development Environment (IDE) for Python (e.g., Visual Studio Code,): Free and open-source
e. Tools for model deployment (e.g., Streamlit): Free and open-source

The hardware and software resources for the toxicity model include essential tools for data processing, model training, evaluation, and deployment. The estimated costs are based on standard industry practices and may vary depending on the specific requirements and scale of the project.

5.2.2 Staffing Requirements

The following are the anticipated staffing requirements for the duration of the 6 months project:

Role	Skills Necessary	Years of experience Required	Time Needed
Data Analyst	Data cleaning, pre-processing, NLP techniques	2-4 years	10 weeks; 10 hours per week
ML Engineer	ML model development, evaluation, fine-tuning, debiasing	3-5 years	10 weeks; 20 hours per week
Project Manager	Project management, stakeholder engagement, communication	3+ years	10 weeks; 15 hours per week
DevOps/ MLOps Engineer	Cloud computing, system maintenance, backup, monitoring	3-5 years	10 weeks; 10 hours per week
NLP Specialist	Natural Language Processing, text analytics, sentiment analysis	4-6 years	10 weeks; 15 hours per week
Frontend Developer	Web development, HTML, CSS, JavaScript, responsive design	2-4 years	10 weeks; 15 hours per week
Backend Developer	Server-side development, API creation, database management	3-5 years	10 weeks; 20 hours per week

5.3 Potential Risks and Risk Mitigation Plan

Below are the anticipated risks associated with the project. The scale is as follows: Likelihood Level and Severity: 1 = High, 2 = Medium, 3 = Low.

Likelihood	Description of Risk	Severity	Response	Responsibility
1	Insufficient data quality or quantity	High	Implement data cleaning and preprocessing techniques to ensure data quality. Cross-reference user-reported data with validated sources.	Data Analyst
1	Lack of Stakeholder Buy-In	High	Engage stakeholders at all stages of the project, communicate the value of the project to the school's goals, and incorporate stakeholder feedback.	Project Manager
2	Model Performance and Accuracy	High	Test and fine-tune various machine learning models and evaluate model performance using appropriate metrics. Consider ensemble methods for improved accuracy.	ML Engineer, Data Analyst
3	Resource Constraints	Medium	Prioritize resources and focus on key aspects of the project. Utilize cloud computing and other cost-effective technologies.	Project Manager, DevOps Engineer

3	Technical Issues and Downtime	Medium	Have backup systems in place and perform regular maintenance and testing to prevent downtime. Utilize monitoring tools to detect issues early.	DevOps Engineer

5.4 Project Assumptions and Critical Success Factors

Our model aims to create a safer and more positive online environment within the school's forums and communication platforms. The success of this project relies on several key assumptions and factors that must be taken into account during the project's implementation.

Project Assumptions
- The school's online platforms have existing instances of toxic behavior that can be addressed and mitigated.
- The stakeholders, including administration, teachers, students, and parents, are open to adopting new technologies and solutions for creating a better online environment.
- The data used for training the toxicity detection model is representative of the types of toxic behavior occurring within the school's online platforms.

Critical Success Factors
- The ability to develop and implement a model with high accuracy, low bias, and the capability to detect various types of toxic behavior.
- The ability to seamlessly integrate the model into the school's existing online platforms, ensuring minimal disruption to users and easy adoption.
- The ability to continuously monitor and update the model, adapting to the evolving nature of toxic behavior and ensuring its long-term effectiveness.
- The ability to effectively communicate the benefits of the model to stakeholders, fostering trust and encouraging their participation in creating a safer and more supportive online community.
- The ability to measure the project's impact using relevant metrics, and to demonstrate tangible improvements in the online environment, user satisfaction, and stakeholder engagement.

By addressing these assumptions and focusing on the critical success factors, the project will have a higher likelihood of achieving its goals and creating a positive impact on the school's online environment.

5.5 Communicating project progress to stakeholders

In the context of the cyberbullying problem and the word level and contextual level models for detecting toxicity in online school forums, it is essential to communicate project progress to stakeholders at various stages. Here is a suggested timeline and approach to communicating project progress:

- **Project Initiation**: Begin by presenting the problem statement, objectives, and proposed solution to stakeholders, including school administrators, teachers, and IT staff. This will ensure that everyone understands the project's goals and potential benefits.

- **Model Training and Evaluation**: As the word level and contextual level models are trained, hold regular updates with stakeholders to discuss the progress, challenges, and any adjustments needed. This can be done through periodic meetings, written progress reports, or both.

- **Post-Model Training and Evaluation**: Upon completing the model training and evaluation, share the results of the training and validation with stakeholders. This should include an analysis of the models' performance, specifically highlighting the best performance model choice and its ability to minimize unintended biases and accurately detect toxic comments.

- **Integration**: During the integration phase, provide stakeholders with updates on the integration of the model into the school's online forum platform. This may include details on any technical requirements, staff training, and support needed for successful integration.

- **Post-Integration Review**: After the model has been integrated and used for a predetermined period, conduct a review with stakeholders to assess its effectiveness in addressing cyberbullying and minimizing unintended biases. This review should involve quantitative and qualitative analyses, focusing on the outcomes outlined in the "Value of Improved Decision" section.

- **Ongoing Updates**: Once the model is fully operational, continue to communicate with stakeholders regularly about any updates, improvements, or maintenance required to ensure the system remains effective and relevant. This may include sharing information on new techniques for bias mitigation or enhancements to the model's performance.

Effective communication with stakeholders is essential throughout the project lifecycle, from initiation to ongoing maintenance. Regular updates and discussions will help to ensure that the DistilBERT model for detecting cyberbullying and minimizing unintended biases is successful in achieving its goals and providing a safer and more inclusive online environment for students.

6. Division of labor

Ruchi
Trained 7 Supervised Models, Trained CNN Model, Trained 3 Transformer Models, Model Deployment, Report, Presentation

Caroline(Yuanmo)
Data Exploration, Word Level Analysis, Sentiment Analysis, Report, Presentation

Nikhil
Trained CNN Model, Report, Presentation

7. References

1. Pacer Center. (2020,November). Statistics on Bullying. https://www.pacer.org/bullying/info/stats.asp
2. Grose, J. (2020, June 21). 5 Bullying Tactics Politicians Use and How It Impacts Kids. Verywell Family. https://www.verywellfamily.com/5-bullying-tactics-politicians-use-and-how-it-impacts-kids-4080749
3. Hinduja, S. & Patchin, J. W. (2018). Cyberbullying Laws and School Policy: A Blessing or Curse? Cyberbullying Research Center. https://cyberbullying.org/cyberbullying-laws-and-school-policy-a-blessing-or-curse
4. National Center for Education Statistics (NCES). (2019). Student Reports of Bullying: Results From the 2017 School Crime Supplement to the National Crime Victimization Survey. Retrieved from https://nces.ed.gov/pubs2019/2019054.pdf

8. Appendix

Overall AUC:
This score tells us how well the model can tell the difference between toxic and non-toxic comments for all the examples.

Bias AUCs:
These scores tell us how well the model can tell the difference between toxic and non-toxic comments for specific groups of people. There are three types:
a. Subgroup AUC: This score is for comments that mention a specific group. A low score means the model struggles to tell if comments about that group are toxic or not.

b. BPSN AUC: This score is for non-toxic comments about a specific group and toxic comments not about that group. A low score means the model mistakes non-toxic comments about the group for toxic comments not about the group.

c. BNSP AUC: This score is for toxic comments about a specific group and non-toxic comments not about that group. A low score means the model mistakes toxic comments about the group for non-toxic comments not about the group.

Generalized Mean of Bias AUCs:
This is a single score that combines all the Bias AUCs for different groups. It gives more importance to groups with worse model performance.

$$M_p(m_s) = \left(\frac{1}{N} \sum_{s=1}^{N} m_s^p \right)^{\frac{1}{p}}$$

where:

M_p = the pth power-mean function

m_s = the bias metric m calulated for subgroup s

N = number of identity subgroups

We use a p value of -5 to improve the model for the identity subgroups with the lowest model performance.

Final Metric:
This is the final score for the model. It combines the Overall AUC with the Generalized Mean of Bias AUCs. Each score is equally important (25% weight). The goal is to improve the model for everyone and reduce unintended bias.

$$score = w_0 AUC_{overall} + \sum_{a=1}^{A} w_a M_p(m_{s,a})$$

where:

A = number of submetrics (3)

$m_{s,a}$ = bias metric for identity subgroup s using submetric a

w_a = a weighting for the relative importance of each submetric; all four w values set to 0.25