PowerText - Automatic content moderation for text in social media:
Detecting violation of terms of service and AI content.

IS4242 Group 8 Project Report

Name	Matriculation Number	NUS Email
Bikramjit Dasgupta	A0200600J	bikramjit@u.nus.edu
Lee Leonard	A0199526M	e0406507@u.nus.edu
Lin Yongqian	A0216803L	e0540360@u.nus.edu
Loh Hong Tak Edmund	A0199943H	e0406924@u.nus.edu
Tang Hanyang	A0211306Y	e0493637@u.nus.edu
Tay Zhi Sheng	A0201840W	e0415649@u.nus.edu
Wong Deshun	A0222482L	e0560014@u.nus.edu

GitHub Repository:
https://github.com/edologgerbird/is4242-group8

Table of contents

1 Problem Introduction

As of January 2023, more than half of the world's population (59%) use social media, with 137 million new users in the past 12 months (Chaffey, 2023). The rapid growth of social media usage has resulted in an increased demand for content moderation to maintain a safe environment for users (Ahmad, 2021). Traditionally, social media platforms rely on manual checks for content moderation, but with the ever-increasing volume of content generated daily, this approach has proven to be time-consuming, labor-intensive, and inconsistent.

At the same time, the explosion of generative AI agents like ChatGPT presents another challenge for social media. Social media platforms may soon be flooded with AI-generated content, which may also contain fake news or misleading information (Ghosh, 2023). Thus, an automated intelligent system may be critical to help maintain the quality and originality of the platform for long-term growth.

1.1 Project Objectives

The objective of this project is to develop an **automated content moderator** for text content in social media platforms. Our proposed system is designed to accurately identify and flag content that violates terms of service in various categories such as hate speech, impersonation, and advertisements, while also being capable of distinguishing between human-generated and AI-generated content. To achieve this, we will leverage several commonly used Natural Language Processing (NLP) algorithms, namely NaiveBayes, PassiveAggressive, XGBoost, CNN, LSTM, GRU, and Transformers.

For our models, we will utilize a manually tagged dataset scraped from Reddit posts (A global social media platform with diverse user-generated content), and an AI-generated dataset on social media content using GPT-3.5 and GPT-4. This combined dataset with a variety of real-world posts complements our extensive training and model evaluation, ensuring accuracy, effectiveness, and applicability across all domains of social media.

The system offers two distinct end products: an automated content collection and screening service for social media platforms, and a user side plug-in to check post/comments. Successful implementation will bring huge benefits to social media platforms, content creators and users, fostering a safer and healthier online environment.

1.2 Business Questions

This project aims to solve the following business problems for different stakeholders of social media platforms:

1. Currently, social media platforms rely on manual screening to filter out undesirable content violations, which can be a time-consuming and labor-intensive process due to their high volume of traffic. Can this process be partially or fully automated?
2. For users, is it possible to provide them with an automated content screening at the time of submission, warn them of any content that violates the platforms' terms of service, and prevent any unintentional violations before such content is published?
3. With the widespread use of generating AI agents such as ChatGPT and New Bing, it is crucial to protect the originality and quality of social media content. Although they may not violate any terms of service now, is it possible to detect such AI-generated content?
4. Can an intelligent system be developed to monitor the percentage of undesirable content by topics or threads, and possibly conduct further descriptive analytics to monitor public opinion and user engagement on social media platforms, for use by platform management or social media regulators?

1.3 Project Timeline

We follow a standard development pipeline for data analytics and information systems projects, which involves the following stages within one month:

1. Data collection: Scrape Reddit posts and comments from various topics, generate AI content from GPT-3.5 & GPT-4, and find additional high-quality datasets.
2. Data preprocessing: Manually tag the Reddit posts dataset, create a confounding undesirable content dataset, and conduct exploratory data analysis (EDA).
3. Model construction: Build up numerous models and train these models with different model structures.

4. Model comparison: Test the models, compare prediction outcome, and choose final models.
5. Intelligent system development: Develop and deploy the platform side & user side end products.

2 Requirements & Features

2.1 Violations Detection

By examining the terms of service from major social media platforms, we found that most of them share similar terms of violations. Here is a table of the common violations of terms of services by different platforms:

	Facebook	Twitter	Instagram	Reddit	YouTube
Harassment	X	X	X	X	X
Cyberbullying	X	X	X	X	X
Hate speech	X	X	X	X	X
Spam	X	X	X	X	X
Impersonation	X	X	X	X	X
Violence	X	X	X	X	X
Nudity/Pornography	X	✓ Restricted	X	✓ Restricted	X
Privacy Infringement	X	X	X	X	X
Terrorism	X	X	X	X	X
Intellectual Property	X	X	X	X	X
Drug use	X	X	X	X	X
Advertisements	✓	✓	✓	✓	✓

Table 1: TOS violations common across different social media platforms

Although advertisements are generally allowed on all platforms, we find it intriguing to monitor them because there are strict guidelines that must be followed, which many advertisements do not currently adhere to. Thus, we have consolidated seven commonly applicable metrics to test our models, namely:

(1) Hate/Harassment/Bullying; (2) Authenticity/Intellectual Property; (3) Privacy; (4) Sexual; (5) Impersonation; (6) Illegal actions; (7) Advertisements/promotions.

Given the challenge of detecting authenticity without additional information or cross-content access, we have decided to remove this specific violation and will address the remaining six.

2.2 AI Content Detection

From our exploration, we acknowledge that definitively differentiating between AI-generated content and human content is a challenging task. Indeed, we do observe certain patterns in AI-generated content, such as common language styles, specific phrasing and expressions associated with different topics. OpenAI has developed an AI classifier, but it has faced challenges in achieving complete reliability with the classifier (Kirchner et al., 2023). Therefore, as a baseline goal of our project, we aim to explore various models and techniques to further investigate the possibility of AI content recognition.

Due to the above-mentioned difficulty of spotting AI content manually, it is impractical for us to tag AI generated content from scraped social media texts. Instead, given the novelty of generative AI agents, we would treat all scraped social media content as human content. We then used GPT-3.5 and GPT-4 to generate text for our AI target classes. We would use the same approach to train and evaluate models on the AI dataset.

2.3 PowerText Intelligent System Design

Our team designed a comprehensive system to meet our business objectives. The design consists of 5 main layers as shown in figure 1. We will be elaborating on each step in the subsequent sections.

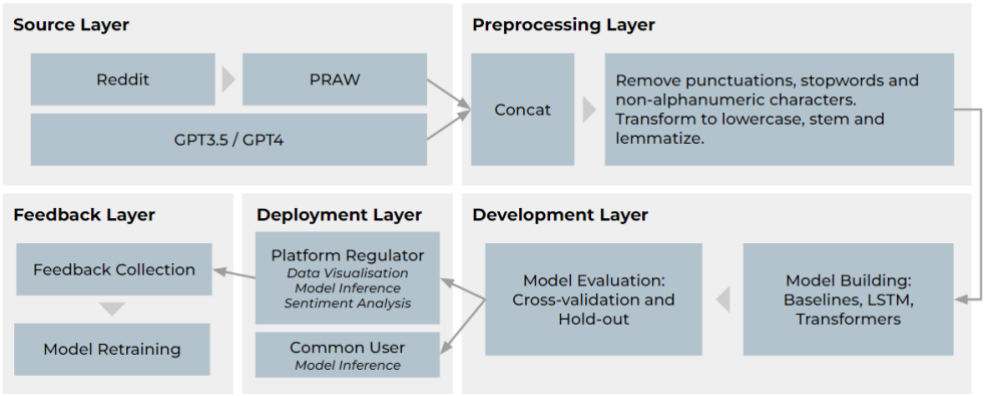

Figure 1: PowerText system design

3 Dataset

3.1 Data Collection & Preparation

Before scraping social media data, we checked the terms of service for content scraping. Reddit is the only major platform that tolerates automated web scraping. With high content similarity on Reddit and other text-heavy platforms (Twitter, Facebook, etc.), we decided to purely use Reddit posts and comments for our social media data.

We built a data scraper powered by the "praw" library and Reddit API, allowing posts and comments to be scraped from multiple subreddits to a structured data format. We used the scraper to collect Reddit raw data of 20+ topics from 60+ subreddits, which are then tagged manually for all six violations.

To add positive labels to certain violations and improve model robustness, we manually created a confounding dataset consisting of 260 randomly sampled "pure" social media posts, which were then paraphrased while keeping the structure for inclusion of violations. We also selected and fine-tuned high quality external datasets for hate content and advertisements, which are two most common violations.

For AI data, we primarily used GPT-3.5 to generate fake social media posts/comments and included several GPT-4 generated contents for wider future readiness.

Type	Entries	Description
Reddit Dataset	16828	Scrapped & Tagged Reddit posts / comments
AI Dataset	6043	Generated from GPT-3.5 & GPT-4
Additional Dataset	26676	Confounding dataset, additional hate & ads

Table 2: Summary of data collected

3.2 Data Processing

Data processing with Python comprised two stages. Technically, the first stage combined datasets by normalizing and concatenating text across categories, allocating target variables for each row, and treating Reddit data as non-AI content. The second stage centered on text processing, encompassing punctuation elimination, lower-case conversion, non-alphanumeric character removal, stopwords eradication, excess spaces, lines, and tabs reduction, alongside text lemmatization and stemming, ultimately discarding words with fewer than two characters.

3.3 Exploratory Data Analysis

To identify patterns and prepare for our model selection, we performed an in-depth Exploratory Data Analysis on the processed dataset, which included examining target class distribution, n-gram analysis, document and word-level clustering, polarity analysis, and assessing perplexity and burstiness. Key findings from this analysis are highlighted in the following sections:

Key Insight 1: Imbalanced Classes

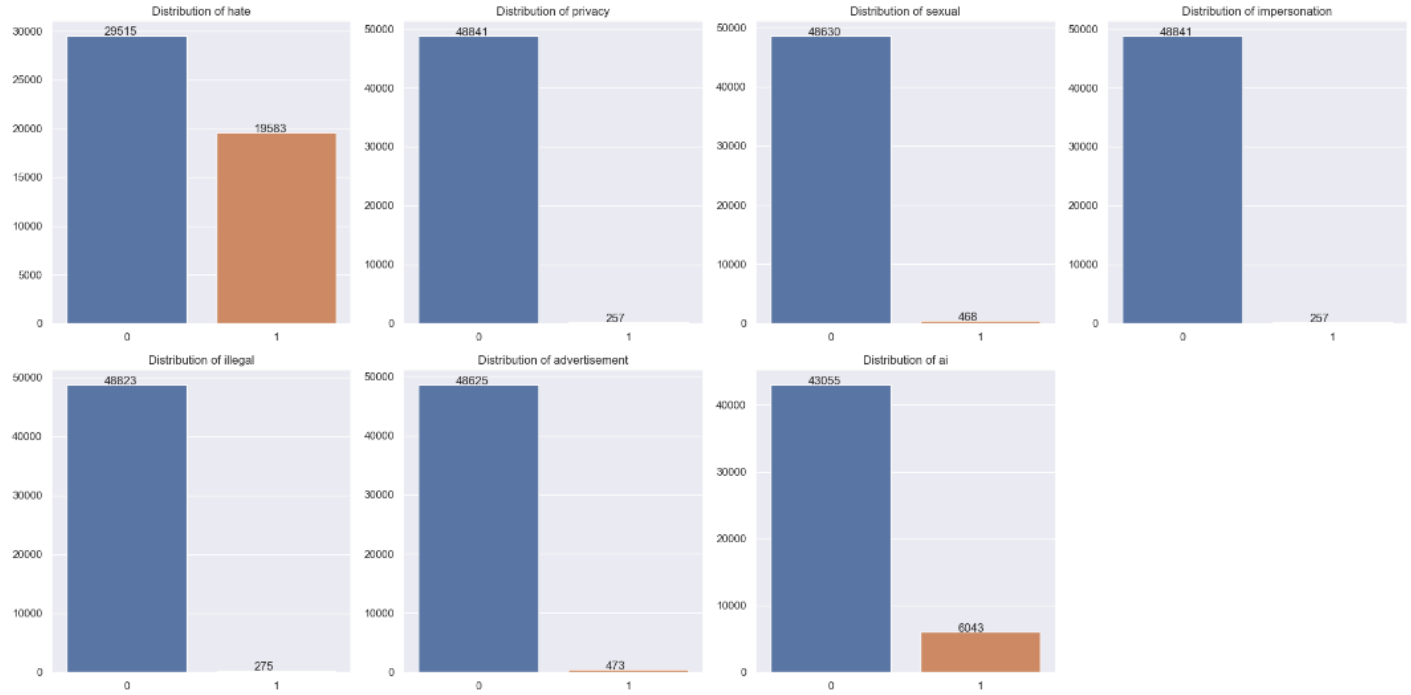

Figure 2: Class distribution of each violation

Our dataset classes demonstrate considerable imbalance, notably in targets besides "hate". This is due to Reddit's prior moderation of content, resulting in a reduced number of terms of service violations. To counteract this, we have incorporated text segments from positive classes into the corresponding target categories.

Key Insight 2: Each target violation is closely associated with a set of specific vocabulary.

Figure 3: Example word-cloud generated from "Hate" data (left) and "Sexual" data (right)

We found that individual target violations are closely linked to word usage patterns. For example, when differentiating "Hate" data from "Sexual" data, it becomes clear that each category has its own unique pattern of word selection.

Key Insight 3: Polarity of posts differ between targets violations.

The polarity analysis revealed a relationship between target violations and their polarity. For AI data, we found a significant bias towards positive sentiments, consistent with the tendency of common AI tools to produce more positive text. "Sexual", "impersonation", and "advertisement" violations showed predominantly positive sentiments, reflecting underlying intentions. In contrast, "hate" text displayed intermittent negative spikes, suggesting that while negative sentiments are not pervasive, when present, they are markedly intense.

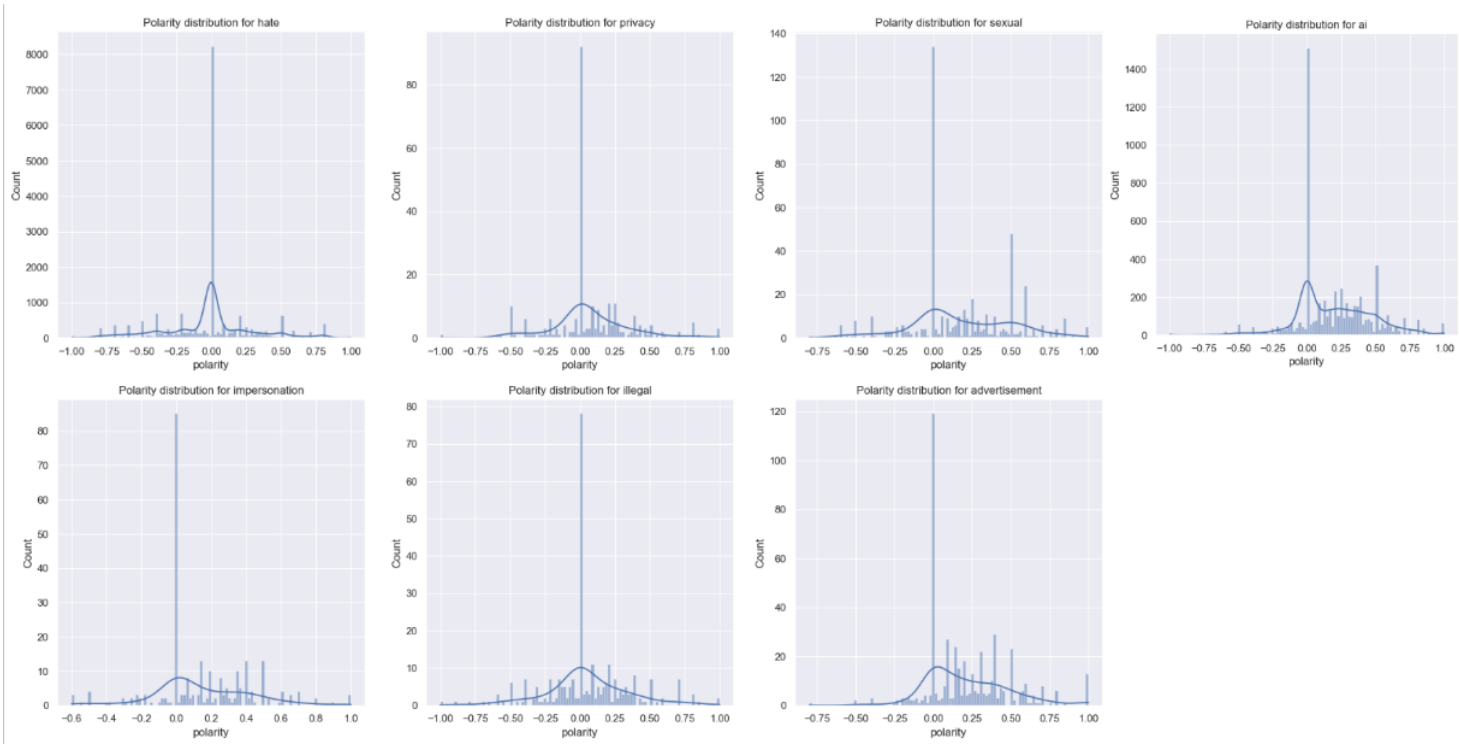

Figure 4: Distribution of polarity for each category of posts.

4 Methodology

4.1 Models Overview

Based on the insights gained from EDA, we outline several key considerations for selecting our models:

1. Imbalanced Dataset: Consider sampling methods or data augmentation to balance out the classes. However, due to the drastic differences in the label datapoints, this might not be viable. Therefore, consider an appropriate loss function, such as focal loss, that is specifically designed to handle imbalanced datasets and can down-weight easy examples while up-weighting difficult ones.
2. Model Architecture: opt for models that are less prone to overfitting and can be trained without taking up too much size or resources, such as those equipped with attention mechanisms or domain-specific knowledge. This is important as the model must undergo retraining to address evolving content trends.
3. Dataset Diversity: Given the distinct vocabulary patterns and polarity differences associated with each target violation, it is crucial to create relevant features that capture these patterns so that we can classify them into appropriate categories. In addition, it is essential for the models to be evaluated on diverse and unseen data to improve generalization abilities.

To meet those considerations, we considered a series of models and evaluated their performance based on precision, recall, and F1-score. These are the metrics we opted to use for evaluating the models as in our use case, both precision and recall are important as false positives and false negatives can dramatically impact whether potentially harmful violating content passes through.

4.2 Baseline Models

4.2.1 Models Chosen

To create baseline models, we developed the Multinomial Naive Bayes, Passive Aggressive Classifier, and XGBoost. These models offer strong out-of-the-box performance and quick training times. Naive Bayes is efficient in classification using conditional probabilities, making it suitable for sentiment analysis (Chaudhuri, 2022). The Passive Aggressive model is a linear, online learning model, effective in text classification tasks (Kumar, 2022). XGBoost, a gradient boosting algorithm, is known for high predictive accuracy and scalability, ideal for various classification tasks, including text classification (Tuychiev, 2021). We trained and evaluated them using 5-fold cross-validation. These models will serve as benchmark for the complex deep learning models.

	Macro Weighted			Micro Weighted		
Model	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
MultinomialNB	0.27	0.17	0.20	0.90	0.67	0.77
PassiveAggressive	0.67	0.55	0.60	0.86	0.85	0.85
XGBoost	0.79	0.42	0.53	0.94	0.81	0.87

Table 3: Macro and Micro Average scores of baseline models using 5-fold cross validation

4.3 GRU, CNN and LSTM

CNN (Convolutional Neural Network) is a type of deep learning architecture designed to automatically learn local patterns or features from input data through convolutional layers, which are specialized layers that perform convolution operations on the input data (Rana 2016). Through this method, local patterns can be split in our text, making them effective at identifying semantic features for our dataset – more than the bag of words approach.

LSTM (Long Short-Term Memory) is a type of RNN that is designed to process sequential data, including natural language sentences (Rana 2016), through a gating mechanism. This combats the vanishing gradient problem, especially for our long texts – and can be better than CNNs to capture long-term dependencies across time steps.

GRU (Gated Recurrent Unit) is a type of recurrent neural network (RNN) architecture that incorporates a simpler gating mechanism to combat the vanishing gradient problem (Rana 2016), allowing it to capture and retain relevant contextual information in our text data – with less computational power than the LSTM.

4.3.1 Model Setup

In our implementation, we built a n-gram based tokenizer from scratch, which generates a n-gram vocabulary from input text, tokenizing sequences into overlapping character groups. By capturing local patterns, this tokenizer enhances model understanding of sub-word information. We chose this due to its outperformance over a basic word-based tokenizer upon testing with all 3 models. All three architectures went through the same tokenized train-test data split, loss functions, training loops, training hyperparameters and evaluation methods.

4.3.2 Model Performance

In general, these three architectures have similar macro-weighted scores, outperforming MultinomialNB, but underperforming when compared to PassiveAggressive and XGBoost classifiers. GRU performed the best for macro-weighted scores, followed by LSTM and CNN.

4.4 Transformers

The Transformer is a deep learning model introduced by Google Brain in 2017. It has been highly successful in various NLP tasks, using self-attention mechanisms to capture the relationships between different words in a sentence, allowing it to learn representations of the input that are highly effective for our task of classification.

4.4.1 Model Setup

A basic transformer architecture was created with an embedding layer, a transformer encoder with 4 layers, and a fully connected linear layer for our classification task – for longer-range dependencies than the previous models.

As for transfer learning, several pre-trained variants and architectures were trained and evaluated. This included the BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) models, which were fine-tuned using our dataset. These models were used to take advantage of their contextual knowledge of the English language for our task – with more model complexity and expressiveness than our basic transformer.

Going beyond, GroNLP's hateBERT model (Caselli et al., 2021) was fine-tuned with our dataset through two architecturally different variants – both for a single-label classification task, due to the lack of overlap between labels. hateBERT has been pre-trained on-top of an underlying English BERT model with a corpus of hate-speech, offering more hate-speech-specific contextual understanding of our dataset, when compared to BERT and RoBERTa. The first variant uses a simple fully connected linear layer for classification, while the second variant includes a classification head in the form of a feed-forward neural network, placed on top of the pre-trained transformer encoder. The third variant added an additional attention mechanism, a position-wise feed-forward network and layer normalization layers, prior to an MLP classification head.

4.4.2 Model Performance

Due to the unbalanced nature of the dataset, evaluating the performance of model using the overall scores might be misleading. Taking minority classes into consideration, macro average scores were therefore used to compare the performance of sequence models (section 4.6). Performance across transformer models varied significantly, with the basic transformer scoring the worst macro average F1 score of 0.55, and the first variant of hateBERT scoring an F1 of 0.82. In terms of the overall sample size-weighted average scores, the transformer models performed very similarly, with the best ones hovering around 0.93 for F1 score.

4.5 Model Comparison for Sequence Models and CNN

Model	**Macro Weighted**			**Size-Weighted Average**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
CNN	0.33	0.31	0.32	0.84	0.85	0.84
GRU	0.33	0.34	0.33	0.85	0.88	0.86
LSTM	0.32	0.32	0.32	0.82	0.85	0.83
Basic Transformer	0.68	0.52	0.55	0.88	0.88	0.88
BERT	0.85	0.65	0.70	0.92	0.92	0.92
RoBERTa	0.80	0.63	0.68	0.93	0.93	0.93
hateBERT Variant 1	0.79	**0.79**	0.79	0.93	0.93	0.93
hateBERT Variant 2	**0.89**	0.77	**0.82**	0.93	**0.94**	0.93
hateBERT Variant 3	0.81	0.69	0.72	**0.94**	**0.94**	**0.94**

Table 4: Average scores of sequence models and CNN, with the same train-test split

Labels/Violations	**Precision**	**Recall**	**F1**	**Count**
Hate Speech	0.94	0.95	0.94	1958
Privacy	0.66	0.73	0.69	26
Sexual	0.67	0.57	0.61	46
Impersonation	0.60	0.46	0.52	26
Illegal	0.50	0.57	0.53	28
Advertisement	0.78	0.83	0.80	47
AI Content	0.98	0.98	0.98	684
Neutral	0.94	0.93	0.94	2175

Table 5: Cross Validation Scores for best performing hateBERT model, across all labels

After comparing the performance of various models and architectures, it became evident that the second hateBERT variant outperforms all models in detecting social media violations for macro-weighted scores. This model's superior performance, especially to other hateBERT variants, can be attributed to its moderate model complexity, and its underlying hateBERT model which was pre-trained on a corpus of hate-speech.

To further evaluate the robustness of the hateBERT variant, 10-fold cross-validation was performed. It was observed that the model's results for underrepresented classes were not as strong as those for more prevalent classes. This can be attributed to the limitations in the dataset's scale and diversity, as will be discussed later. Despite this shortcoming, the overall performance of the hateBERT variant remained superior to that of other models, cementing our decision to choose it as our main model for the end-product integration. The information provided by the F1, Precision and Recall scores were sufficient, so we did not include AUC-ROC metric.

5 End Product

5.1 Platform for Regulators: PowerText Analysis System

Our first developed intelligence system is designed for moderators and platform regulators. It automatically examines Reddit posts within a defined period, generating aggregate statistics, categorizing posts by potential violation using our fine-tuned hateBERT model, presenting statistics on flagged posts, and performing polarity and sentiment analysis by subreddit. This allows content moderators to assess Reddit posts for potential terms of

service breaches, monitor public opinion and engagement on social media, and feedback on classification accuracy which is used for model retraining. This ensures long-term model relevance and reduces model drifting.

The regulator platform has been deployed and can be accessed at: https://bit.ly/is4242_analysis_system .

5.2 Platform for Common Users: PowerText PowerCheck

We have also developed two user interfaces with distinctive designs for common users to easily check if a specific chunk of text input violates any guidelines. Users can simply submit their text and receive instant results indicating any detected violations in the form of messages or icon labels. The user side interface is built using Django in a modular structure, allowing for easy integration of enhanced models in the future. It can also serve as a simple plug-in on any social media platform, providing users with prior notice if their post submission is flagged for detected violations.

The user platform can be run via localhost and a demo video is available at: https://bit.ly/is4242_user_demo .

6 Discussions

6.1 Project Conclusions

In conclusion, our sizeable dataset and models have demonstrated the potential to effectively detect TOS violations and AI-generated content on social media platforms using a variety of techniques, such as linear, naive bayes, tree-based, and sequence models. Through an investigation into the performance of several different model types, we have also set the premise for the most effective models for this task of automated content moderation. Consequently, we have demonstrated a practical level of efficacy with the performance metrics of our best model, such that this project can be used in the real world.

Furthermore, our developed front-end platforms cater to both moderators and common users, streamlining the content moderation process and reducing the time and effort required.

6.2 Limitations

A limitation of this project is the restricted scale and diversity of the dataset used for training and evaluation, which hinders the ability for models to accurately detect some minority data collection. A contributing factor is the effective platform and content moderation effort by Reddit moderators, making violations scarce. As we have implemented various techniques including focal loss to down-weight easy samples and up-weight difficult minority categories, we believe that improving the dataset's scale and class balance would significantly improve model performance – since our existing model already prove that these classes can be detected.

Another limitation of this project is the time and resource requirement for extensive hyperparameter tuning and cross-validation for large sequence models. Given more time and resources, we can fine-tune our model hyperparameters, which can further improve the performance of larger models.

Furthermore, ethical concerns arise when regional diversity in policies is difficult to account for, through our models – due to our generalized dataset and target variables. False positives are also an issue, potentially unjustifiably limiting the speech of social media users – which might be a reason similar models have not been widespread in the industry. Projects like this may therefore assist human moderators instead of eliminating them.

6.3 Future Work

Our system offers potential for further advancements and refinements. For instance, we can automate the comprehensive scraping process and orchestrate execution using Apache Airflow. In addition, automating the model retraining phase would maintain model currency and avert drift. Implementing a cloud database would facilitate storage for scraped posts and collected feedback. Enabling users to choose specific subreddits for scrutiny would improve dashboard capabilities. Furthermore, by utilizing scraped posting timestamps, we can conduct time series analysis to track TOS violation and sentiment changes over time among various subreddits. Examining data at the user level could help identify individuals more likely to commit TOS violations. Overall, these proposals are designed to enhance the system's reliability and effectiveness.

7 References

Ahmad, B. (2021, September 17). *What is the importance of content moderation?* Retrieved from https://www.techmaish.com/what-is-the-importance-of-content-moderation/

Bex, T. (2023, April 8). *Beginner's Guide to XGBoost for Classification Problems | Towards Data Science.* Retrieved from https://towardsdatascience.com/beginners-guide-to-xgboost-for-classification-problems-50f75aac5390

Caselli, T., Basile, V., Mitrović, J., & Granitzer, M. (2020). Hatebert: Retraining bert for abusive language detection in English. *arXiv preprint arXiv:2010.12472*.

Chaffey, D. (2023, January 30). *Global Social Media Statistics Research Summary 2023*. Retrieved from https://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/

Chaudhuri, K. D. (2022, March 21). *Building Naive Bayes Classifier from Scratch to Perform Sentiment Analysis.* Retrieved from https://www.analyticsvidhya.com/blog/2022/03/building-naive-bayes-classifier-from-scratch-to-perform-sentiment-analysis/

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ghosh, S. (2023, April 5). *New Prediction: ChatGPT Content Would Flood Social Media Platforms.* Retrieved from https://aithority.com/representation-reasoning/information-fusion/chatgpt-content-would-flood-social-media-platforms/

Kirchner, J. H., Ahmad, L., Aaronson, S., & Leike, J. (2023, January 31). *New AI classifier for indicating AI-written text.* Retrieved from https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text/

Kumar, A. (2022, October 15). *Passive Aggressive Classifier: Concepts & Examples - Data Analytics*. Retrieved from https://vitalflux.com/passive-aggressive-classifier-concepts-examples/

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692.*

Rana, R. (2016). Gated recurrent unit (GRU) for emotion classification from noisy speech. *arXiv preprint arXiv:1612.07778.*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.