Content Moderation

Group 5: Valerie Chu, Allen Choun, Rohit Suresh, Hunter Jacobson, Vishal Saminathan

GitHub Repo: https://github.com/UConnCSE3000Group5/AIContentModerationProject

Methodology

Our process for this project started upon choosing which prompt we wanted to do. We chose Content Moderation because it would allow us to expand our skills in working with datasets. Our next goal was finding a suitable dataset that would give us efficient data to test. To that end we chose a dynamically generated hate speech dataset found on GitHub. One issue we had was certain datasets were missing a specific column necessary for the code. We bypassed this issue by modifying the code such that it focused on the right columns. Our goal then became to test whether or not the phrases marked in text files in the dataset were hate speech or not. We used DeepNote to conduct this analysis and used Python to create a script to do this. From there we kept tweaking values within the code to yield a much more satisfactory accuracy score.

Ethical Reflection

There are times when models will produce false positives, otherwise known as comments that are mistakenly flagged as harmful. For our model, we analyzed a dataset to detect if a comment is declared as hate speech. Although we achieved an accuracy of 78.17%, we found that our model is prone to having false positives as well. We also thought about how this can impact the user's experience, for instance, users who feel that their content has been unfairly censored might experience frustration, confusion, or even that they are being discriminated against. Over time, the user may also start to lose trust in the platform and therefore it decreases their engagement in the platform as well. When users no longer have that trust in the platform, the low engagement turns into leaving the platform behind altogether. This could also be damaging for specific communities or groups who often experience this as well, especially those who grew up using language that is unfamiliar to others. It is important to recognize bias in the

model when it comes to misinterpretations suggesting someone's comment is toxic. The dataset that we analyzed is prone to bias and to actively get rid of this bias, we can make adjustments such as diversifying or cleaning the training data and consider incorporating user feedback so that the model can be retrained to recognize what is really considered harmful and what is not.

If toxic comments from certain groups are more frequently flagged, there are certain steps that could mitigate this bias. As mentioned before, we could diversify and clean the dataset to ensure that there is more representation in different dialects and in making sure that our model is up to date on commonly used slang. There were some cases where culturally specific expressions were flagged as toxic which raises ethical concerns in unfairly targeting marginalized groups or those who use slang that the model is not familiar with. The steps we took to mitigate further bias was cleaning text through tokenization, stopword removal, and special character filtering including transforming the data with a TF-IDF vectorizer. Training a logistic regression model, we were able to conduct further analysis into the inconsistencies in bias that our model may encounter.

Proposed Improvements

During the process of modeling the data we ran into a few issues where the accuracy was not favorable. Originally, the overall accuracy of our model was about 67.6%, this was later boosted to 78.17% accuracy, almost 10% more than it was at the beginning. We first began with tuning the parameters within LogisticRegression by adding the following:

```
model = LogisticRegression(class_weight='balanced', penalty='l2', C=1,
solver='liblinear',random_state=42).
```

From there we looked for more methods to reduce bias. We tried adjusting C to increase and decrease the strength of regularization as well as introduce SMOTE but some of these changes made our model less accurate. The last method we tried was introducing feature selection by

using SelectKBest. By introducing feature selection, our accuracy jumped to ~73% and by adjusting some values such as C (regularization strength) and K (top-k most informative features), our accuracy ultimately became 78.17%. The classification report ended up with 76 false positives and 55 false negatives. From adjusting certain values and adjusting various parameters, we were able to substantially increase accuracy with ~22% inaccuracy. There may be better adjustments but from what our group was able to find this was the best increase we were able to get.

Due to this ~22% error rate, there is still much room for issues among users. For this system to detect hate speech, the misclassification of “not hate” as “hate” and missing actual hate speech, these can lead to many impacts. As this works off a small data set, the use of sarcasm, slang, and certain phrases can be misunderstood and classified as “hate.” For those who are activists, they may be silenced unintentionally and could be wrongly accused of hate speech. For groups that are being targeted by hate speech, by missing actual hate speech, exposes them to posts that directly target them, inducing distress and fear. Due to these false positives and negatives, this model may be unintentionally suppressing expression and reinforcing bias as well as destroying users' trust of the platform.

The idea that nuance should be taken into account, with regards to how the model flags toxic content, ties into our arguments about false positives and false negatives. Looking at just one part of the text, it may look offensive, but the true intention is entirely dependent on context. Therefore, bias mitigation strategies should also aim to preserve expression by incorporating more context-aware methods of analysis, in order to capture the nuance of the original content. Of course, this would be easier if the content we are working with was longer than just a

sentence or two, but oftentimes online, the messages are only this long too. In an ideal world, the model would be able to pick up on the nuance with only the minimal required context.