COS 424 Final Assignment: Unintended Bias in Online Comment Toxicity Classification

Jae Sohn
Economics Department
jaes@princeton.edu

Sally Hahn
Mathematics Department
yhahn@princeton.edu

Abstract

Providing a safe online environment for users to be able to meaningfully express themselves and their opinions is important for online communication service providers. A toxic conversation - such as verbal and psychological threats or attacks to identity, race, and culture, or completely shutting down negating view - can cause users to stop expressing themselves and furthermore stop them from seeking to understand others with different viewpoints. We seek to facilitate online conversations by developing unbiased machine learning models to detect toxicity in conversations while taking into account identity biases. Applying Logistic Regression, Linear SVC, and Random Forest models to both baseline (no identity) and extension (appended identity) datasets, we determined that while Random Forest did not produce significant improvements, both Logistic Regression and Linear SVC models benefited in having identity labels in classifying toxicity.

1 Introduction

Toxicity in online conversation is extremely commonplace on the web today. A brief skim in the comment sections of any common news sites provides abundance of evidence that systemic harassment and abuse is prevalent in the online community today. In an effort to improve online dialogue and evaluate the factors surrounding the toxicity of the conversations, researchers Aja Bogdanoff and Christa Mrgan built the Civil Comments plugin in 2015. Using independent news sites as its platform, the plugin would initiate a moderation process in which commentators must peer-review others' comments before they can post their own. After 2 years and nearly 2 million evaluations, the project came to a close in 2017[2].

1.1 Motivation

There is a need for a fair and scaleable moderation system that can more readily and efficiently (compared to human peer-reviews) evaluate comments on the online forum. In the optimal case, the supervised-learning model can immediately, upon seeing a new comment, be able to classify whether the string is needlessly toxic and take appropriate measures whether that is alerting the administrator, or even better, block the comment itself with assured certainty. The key here is to learn with enough certainty given unintended bias in such a model - nuances and the detection of context-based trigger words such as "gay" and "homosexual" may easily mislead prediction mechanisms. Furthermore, it is important to keep in mind the distinction between what is really considered "toxic", and merely disagreeable comments. The aim of this project is to explore different models that can begin to fulfill this moderation role and accurately classify purely toxic comments while regarding potential biases. Perhaps in the future, civil conversation can take place with minimal harassment, leaving room for more organic dialogue and worthwhile exchange of ideas.

2 Data Description

2.1 Feature Characteristics

The entire dataset, as sponsored by Google Jigsaw and distributed by CC0[1], contains 1.8 million comment entries and 45 features. The label is tagged as *target* and it is a continuous number between $[0, 1]$, which serves as the fraction of peer-reviews that have deemed the current comment toxic. This value is critical for classification purposes; for all *target* $>= 0.5$ (indicating that more than half of the peer-reviews deemed it toxic) the comment was labeled as toxic. The rest of the features include various characteristics about the comments. Many variables are related to identity, such as race, sexual orientation, and religion. These feature values are also a continuous number between $[0, 1]$, which again serves as the fraction of peer-reviews who thought the comment was about that feature. See below for an example comment entry:

Continue to stand strong LGBT community. Yes, indeed, you'll overcome and you have.

The above comment has the following label scores:

Toxicity Score: All 0.0
Identity Labels: homosexual_gay_or_lesbian: 0.8, bisexual: 0.6, transgender: 0.3 (all others 0.0)

These values generated from peer-reviews indicate the fraction of votes that has denoted the comment to be appropriately described by the label. For this example, everyone has agreed that the above comment is **not** toxic, wile 80% of the reviewers have tagged it to be related to homosexual_gay_or_lesbian, and so on.

2.2 Data Exploration

Organized by identity features, the following figure displays the number of comments broken down by toxicity:

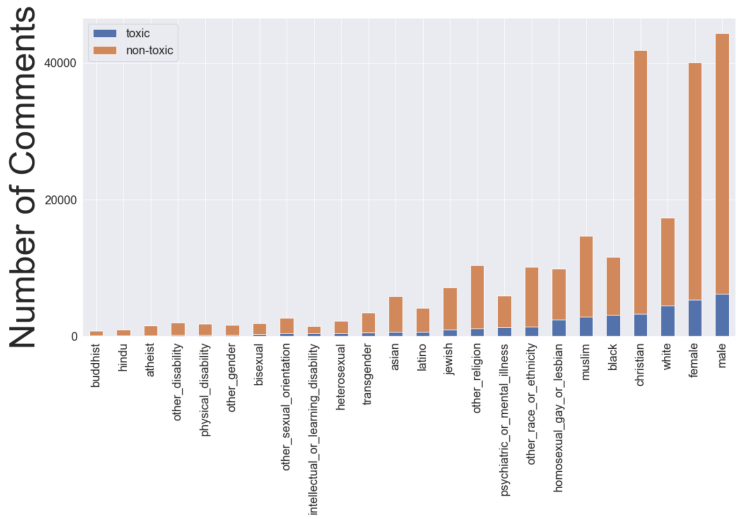


Figure 1: Number of toxic and non-toxic comments, organized by each identity label

Figure 1 shows that gender identity labels male and female, and race identity labels white and black are extremely prevalent in toxic comments. Overall, the number of comments including the identities are also shown by the stacked barchart including the non-toxic context; in this sense, male, female, and christian are the most commonly tagged identities.

This exploratory glimpse is quite revealing; we can see right from the start that gender-based topics (*male, female*) have the highest number of toxic occurrences, and that racial tension (*white, black*) is also without a doubt very present in the online community. To take into account inherently-common topics (e.g. *white* has less total toxic occurrences compared with *male, female* tagged comments,

although its ratio of toxicity is probably higher), we generate a chart (Figure 2) that displays the sorted weighted toxicity of each identity topic. Each occurrence of each topic is scaled by the respective toxicity rating, which is initially presented in the data set as a fraction.

The following figure weighs the identities by their toxic comment appearance:

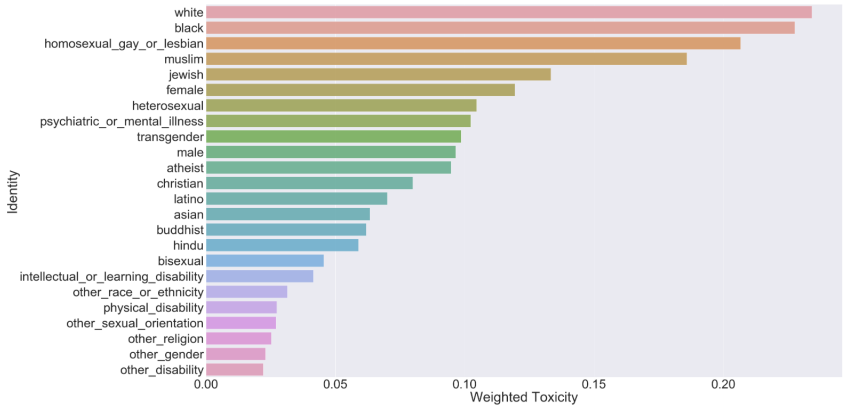
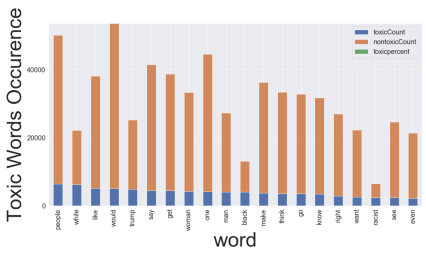


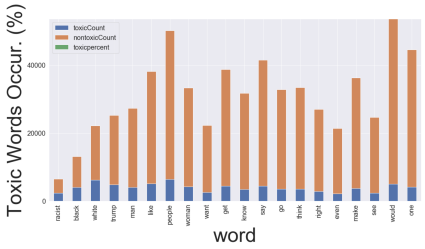
Figure 2: Weighted toxicity, organized by each identity label

Figure 2 shows that if we scale each appearance of identity label with the toxicity rating of each comment, race labels white and black were involved in the most toxic comments. Next, homoesexual_gay_or_lesbian, and religious labels muslim and jewish were also prevalent in toxic comments. These findings bring to light that indeed, many of the online communities' toxicity are centered towards racial, sexual orientation, and religious contexts.

Next, to dive deeper into the nature of the dataset, we clean up the comments into tokens and display prevalence results organized by individual words:



Number of time word appears in Toxic Comments



Number of time word appears in Toxic Comments, by percentage of total appearance

Figure (a) displays the words in descending order, organized by the number of times they appear in toxic comments. Some words, such as *people, like,* and *say*, clearly do not contain toxic elements. To combat this issue, we account for it in the Figure (b) by taking a percentage value instead. This methodology rules out words that are merely numerous in all comments and thus commonly found in toxic comments. In this new figure, the most commonly racist words comes out to be *racist, black, white, trump, man*, and so on.

3 Related Works

Google Jigsaw recently launched a toxicity recognition software called *Perspective* using machine learning: `https://www.perspectiveapi.com/#/`. By typing in any sentences in the website's provided textbox, the system would return in real-time the toxicity rating that is between $[0, 1]$. Using this experiment as the framework for their research study, Hosseini *et al.* from University of Washington sought to find adversarial conditions that would systematically undermine its usability. In their paper "Deceiving Google's Perspective API Built for Detecting Toxic Comments"[4], they first tested the effects of deliberately misspelling common offensive words, such as *idiiot* and *stu.pid*.

Compared to their correctly spelled counterparts, the typos produced significantly lesser scores on the toxicity scale.

One of the most successful approach used by previous attempts was to use a *bag of words* representation with the NB-SVM (Naive-Bayes Support Vector Machine) model[5]. This paper will also utilize this methodology, as we concluded that the bag of words representation is a logical choice given the free-response nature of the comments, as well its capability to tokenize the most commonly used words. Besides SVM, we will also fit with Logistic Regression and Bernoulli Naive Bayes models and compare for better performance.

Another interesting approach, employed by Dixon et al. from Google Jigsaw, is to balance the training set with synthetic comment entries using the following template [6]:

Template Examples	Label
I am <IDENTITY>	Non-Toxic
I am a <IDENTITY> person, ask me anything	Non-Toxic
<IDENTITY> people are just like everyone else	Non-Toxic
I hate all <IDENTITY>	Toxic
I am a <IDENTITY> person and I hate your guts and think you suck	Toxic
<IDENTITY> people are gross and universally terrible	Toxic

This trick cleverly allows biased identity labels to gain additional "correct" labeling such that the models can grant further weight to the correct interpretation of these labels. Emprically, the researchers were able to increase AUC by 1.4% using this method.

Finally, one paper from Google Jigsaw by Borkan *et al.* explores different metrics that can effectively utilize bias analysis in text classification [3]. Additionally from an overall AUC evaluation, Borkan *et al.* also sliced the data into 3 subgroups to take into account bias-detection performance. Out of these 3 generated metrics, a "generalized mean bias AUC" is calculated. We will also use this methodology to evaluate our models' bias detection performance, as specified in the following sections.

4 Methods

4.1 Data Engineering

The starting data set was of size (1,804,874, 46). We applied the following changes to clean and transform the data to a more workable format.

- **Missing Values:** Not all comments in X_train were evaluated by peers and the rows corresponding to these comments had missing values, and thus these rows were dropped. Dropping some data allowed us to not only delete NaN values, but also reduce the size of the data set and thus significantly decrease running time.
- **Irrelevant Data:** We dropped all unrelated metadata columns such as 'parent id', 'publication id', etc. We also dropped identity columns for identities that had no associated 'toxic' labeled comment for future ROC-AUC computation [1].
- **Train/Test Split:** We used sklearn tools to randomly split the data set into 80% train and 20% test sets. Then, the column 'toxicity' was extracted to be target value represented by y_train, y_test, respectively. Since we are attempting to predict toxicity based on comments, all other columns except 'comments' were dropped in X_test. Hence X_train included 'comments' (str values) as well as identity columns (float values). X_test only contained 'comments'. We did keep the identity column values of X_test stored somewhere else for evaluation purposes, called X_test_real.
- **Bag of Words (BoW):** Then we used the Bag of Words representation to vectorize the comments. Punctuation was removed and computer languages such as <href or \n were removed. Stop words were removed from the list of tokens, words were lemmatized, then tokenized. Finally, the comments were transformed into a bag of words representation with the 1,000 most frequented words. Because the data-set is of 0.2 million points even the 1,000th most freqeuented word - 'threaten' - had 1101 occurrences.

[1] sklearn roc-auc function cannot take arrays of size 0, and cannot take target vectors all of one value

- **Converting Float64 Values:** Standard classification-based sklearn models require target values to be in categories or discrete values. Because our identity *and* toxicity values are in fractions denoted in float64 data type, changed all values of 'toxicity' and identity columns to Boolean outcomes, by letting < 0.5 'False' and ≥ 0.5 'True'.

After all data clean up, we result with X_train of size (188,069, 1014) and X_test is of length 47018 where each entry is a comment in string.

4.2 Models

This project's goal is to calculate the toxicity levels of comments, taking into account that words referring to certain identities are correlated to toxicity. This required 2 separate steps. First, We needed to find a way to predict identities from comments, a.k.a approximate identity columns from only comments and append the estimated identity column values to X_test, which we refer to as X_test_estimated. Second, from this newly generated X_test_estimated, we have to be able to predict toxicity levels to test against y_test.

4.2.1 Predicting Identity Labels:

Here we are only dealing with the training set. We fitted models to predict *each* of the 17 remaining identity columns from only comments (BoW vectors).

- **Train/Test Split:** We split X_train again to 80-20 partitions so that we can use the 80-split to train and test against the 20-split.
- **Balancing:** Since the dataset is unbalanced, we re-weighted the 80 partition so that the weight of 'True' values of the identity column we were trying to predict was equal to that of 'False' values. This required re-weighting for each of the 17 columns.
- **Models**
  1. Naive Bayes: sklearn Bernoulli since the predictions are binary.
  2. Linear SVC: sklearn Linear SVC instead of SVM (kernel = linear) for speed.
  3. Logistic Regression: sklearn Logistic Regression. solver = 'lbfgs'.
- **Predicting Identity Labels:** Ultimately, logistic regression performed best, so we used logistic regression to approximate each identity column of X_test and append it, so that it had the same dimension as X_train. We call this newly approximated test set X_test_estimated. See Results section 5.1 for the ROC AUC scores of the models used in predicting identity labels.

4.2.2 Predicting Toxicity Scores

We then fitted the entire X_train - with both BoW and identities - to classify toxicity. We evaluated how X_test_estimated would perform in toxicity classification against y_test.

- **Random Forest:** Initially picked because the whole data set was imbalanced and Random Forests are more robust to imbalance bias. Used sklearn's Random Forest Classifier. We set n_estimators=1000 (but the results changed little from n_estimators = 50). Class weight was set to 'balanced' since only 10% of initial train set was 'toxic'.
- **Linear SVC:** Used sklearn's LinearSVC classifier as opposed to SVM kernel = 'linear' for computational speed. SVM kernel = 'linear' takes at least quadratic time and is almost impossible to run on big datasets. Class weight was also set to 'balanced' for the same reason.
- **Logistic Regression:** Used sklearn's Logistic Regression. Solver was set to 'lbfgs'. Penalty was set to 'l2' since the goal was highest accuracy, not extracting stronger vocabulary/features.

4.3 Evaluation Metric

We used a weighted combination of four AUC metrics to evaluate our models: Overall AUC and the Generalized Mean of 3 distinct Bias AUCs[2]. The evaluation metric was adjusted and coded from scratch accordingly, based on the description and formulas provided below.

[2]Metric taken from Borkan et al *Nuanced Metrics for Measuring Unintended Bias with Real Data for Text Classification [3]*

For **Overall AUC**, we used scikit learn's *roc_auc_score* library to calculate the ROC-AUC for the full evaluation set. This AUC_overall value is weighted with $w_0 = 0.25$ in the Final Metric.

For **Bias AUCs**, we took three different subgroups to more precisely evaluate bias:

- Subgroup AUC: The data set is reduced to include only the entries that contain the specific identity subgroup. This metric determines if the model can distinguish between toxic and non-toxic comments that mention the same identity.
- BPSN (Background Positive, Subgroup Negative) AUC: The data set is reduced to include only the non-toxic entries that contain the identity and toxic entries that do not. A high value in this metric means that the model can accurately differentiate between non-toxic entries that contain the identity and toxic entries that do not.
- BNSP (Background Negative, Subgroup Positive) AUC: The data set is reducted to include only the toxic entries that contain the identity and non-toxic entries that do not. A high value in this metric means that the model can accurately differentiate between toxic entries that contain the identity and non-toxic entries that do not.

Taking these three Bias AUCs, we calculate the generalized mean:

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

where:
M_p = the pth power-mean function
m_s = the bias metric m calculated for subgroup s
N = the number of identity subgroups.
In accordance with Borkan *et al.*'s methodology, we also applied p value of -5.

Weighting and combining the above AUCs, we get the **Final Metric**:

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

where:
A = number of submetrics (3)
$m_{s,a}$ = bias metric for identity subgroup s using submetric a
w_a, w_0 = are weights for the relative importance of each submetric; all four w values are set to 0.25.

5 Results

5.1 Baseline Results

We first present the baseline results for the three models predicting toxicity. By baseline we refer to what the results would have been if the dataset *only* contained comments and toxicity target values, and identities were not labeled. The comments in training set were converted to a Bag of Words representation, then fitted against toxicity values, then tested using the comments in the testing set converted into a Bag of Words representation using the same vocabulary in training. The results are as follows:

Baseline Measurements	Train Accuracy	Test Accuracy	Overall AUC	Subgroup AUC	BPSN AUC	BNSP AUC	Final Score
RandomForest	0.8927	0.8943	0.5690	0.5554	0.5725	0.5524	0.5582
LinearSVC	0.8988	0.8983	0.5455	0.5326	0.5360	0.5411	0.5388
LogisticReg	0.8989	0.8983	0.5569	0.5451	0.5433	0.5547	0.5500

Figure 4: Baseline results with Random Forest, Linear SVC, and Logistic Regression models.

The goal of this paper is to beat these figures so that the Final Score, which reflects Bias, is higher in our proposed model.

5.2 Identity Extension Results

Identity Extension	Train Accuracy	Test Accuracy	Overall AUC	Subgroup AUC	BPSN AUC	BNSP AUC	Final Score
RandomForest	0.9908	0.8849	0.5554	0.5310	0.5571	0.5300	0.5435
LinearSVC	0.7930	0.6645	0.6934	0.6527	0.5852	0.7592	**0.6726**
LogisticReg	0.7818	0.6589	0.6934	0.6560	0.5904	0.7579	**0.6801**

Figure 5: Identity extended results with Random Forest, Linear SVC, and Logistic Regression models. These models were fitted with identity labels pre-predicted.

We now present the results from our two-step identity inclusive model. From Final Score, it is immediately apparent that Linear SVC and Logistic Regression models improved drastically once we took into account identity labels by appending our predicted identity columns. Linear SVC saw a 24.8% increase, while Logistic Regression saw a 23.7% increase. On the other hand, Random Forest seems to have dropped in the Final Score metric by -2.63%.

Besides Final Score, the BNSP AUC values also saw meaningful increases for the Linear SVC and Logistic Regression models. As previously described in section 4.3, the BNSP AUC measures if the models can effectively differentiate between toxic entries that mention the identity and non-toxic entries that do not. In other words, a significant improvement in this score indicates that our identity-appended models were able to in fact account for some bias and correctly assign toxic comments referring to identities.

6 Discussion

6.1 Results Analysis

6.1.1 Underperformance of Random Forest

First, we notice that Random Forest has great test accuracy, and thus predictive properties, but has very low ROC-AUC and Bias Metric scores. Random Forest is known to overfit without proper hypertuning of the parameters [7]. Due to this tendency that ultimately traces its roots from the fundamental shortcoming of decision trees, we did not find it surprising that out of the three models studied, Random Forest not only overfitted severely (given Train and Test accuracy of 0.9908 and 0.8849, respectively) but also performed the worst (in terms of Final Score of just 0.5435). It probably overfitted the train data by separating the 'toxic' data points from the non-toxic ones, so the space that 'toxic' elements take up is probably very small, even when Random Forest Classifier was run with 'balanced'. It therefore also probably assigns most data points to be 'non toxic', giving a high Test Accuracy rate but low values for both bias and general AUC scores.

6.1.2 Predicting Identity vs. Toxicity

Second, we notice that AUC scores are significantly lower for predicting *toxicity* from a bag of words (see figure 4 and 5) when compared with AUC scores for predicting *identities* from a bag of words (see Appendix section 8.1), even when the same model was used (e.g. Logistic Regression). The general AUC scores are all below 70% when predicting toxicity, but all above 70% when predicting identities. Therefore, we conclude that toxicity is harder to predict than identities from a vector of words. This makes sense in that identities are referred to by words associated with the identity, and thus there is a direct correlation between the occurrence of a word with the subject topic of an identity. Toxicity, however, includes language complications such as sarcasm and dual meanings, meaning that although some words are correlated with toxicity, they are not necessarily always so. This explains the relatively poor performance of toxicity prediction in all models as opposed to identity predictions (see Appendix section 8.1).

6.1.3 Overall AUC vs. Bias AUCs

Third, we notice a big jump in Overall and Bias AUC scores from Figure 4 to Figure 5. We attribute this to the previous point, that toxicity is very hard to predict from a vector of words. Therefore including information on identity values, which contains a lot more context given the original vector,

can vastly improve toxicity prediction results. In particular, we see that all Subgroup AUC, BPSN, and BNSP increase with Identity Extensions, and that leads us to conclude that bias has been meaningfully mitigated - and plays a role in boosting overall AUC scores. We see the greatest jump in BNSP AUC, so there is improvement in correctly assigning toxic comments referring to identities; and some in BPSN AUC so there is also improvement in correctly assigning non-toxic comments referring to identities. This is reflected in the Subgroup AUC's improvement - adding clarity to the vector of words help understand the context in which an identity word was used.

6.2 Future Works

6.2.1 Computational Limitations

There are many sections in the project that can be significantly improved if we can bypass the computational limitations that prevented deeper learning. For example, partly due to run-time limits, we had to reduce our raw data set into a much smaller slice. Although we still ended up with around 0.2 million data entries, ideally we could also utilize the other 90% that was deleted. Another significant area that could be improved by stronger computation power includes having a larger word of bag structure. We limited our learning vocabulary to 1000 most common words, but ideally we could achieve better accuracy and performance in general if we can more comprehensively analyze comment structure. A larger bag of words would mean a better understanding in the way our sentences are nuanced, and therefore allow us to get a better grasp of biases in comments. Finally, tweaking some of our models (especially the under-performing Random Forest Classifier) and their hyperparameters (e.g. *n_estimators, max_depth*) could yield us better performance as well. This specific area of improvement also includes replacing LinearSVC with a canon SVC model, such that we can properly implement the model in terms of libsvm rather than liblinear for more consistent penalty and loss functions. Many of these potential improvements were easy to difficult to implement strictly due to computation limitations given the extremely large raw data set.

6.2.2 Latent Variable Models

Next, on a more fundamental side, we could perhaps seek underlying, latent structures within the feature set of our data. That is, by applying a variety of latent variable identification methods such as LDA or PCA, we could not only achieve a deeper understanding in the features themselves through their interactions, but also very likely achieve better performance overall. An added benefit with PCA then also includes a more complex feature selection process, albeit with an decreased interpretability trade-off. Through LDA we could discover the types of toxic and non-toxic comments, and have other indicator columns indicating the type of comment, just like the identity variable columns that we had, giving us even more context in toxicity prediction.

6.2.3 Neural Networks

As many research papers on this topic have utilized neural networks for the construction of identity frameworks, we would also like to pursue this approach and compare against our tested-models for additional insight and potential performance gain.

7 Conclusion

Predicting toxicity of a comment is important to ensure meaningful and organic exchange of ideas in online settings. However, due to the complicated nature of predicting toxicity from user inputted text strings, it is critical to consider other elements of the text (e.g. identities labels) to learn context. Otherwise, as we have shown in this paper, a naive machine learning model may be subject to biases and classify toxic labels incorrectly. As more data is harvested going forward and more people come online to engage in communications, it is ever-more essential to continue developing models that can effectively execute natural language processing, not just limited to predicting tox. Here, we showed that Logistic Regression and Linear SVC models are very viable and promising options looking ahead, especially with continual expansion of computational power.

8 Appendix

8.1 ROC AUC scores for Predicting Identity Labels

ROC_AUC_score	*Bernoulli NB*	*Linear SVC*	*Logistic Regression*
Asian	0.8801	0.8658	0.9001
Atheist	0.7253	0.6687	0.6904
Bisexual	0.7114	0.7154	0.7288
Black	0.8699	0.9367	0.9461
Buddhist	0.7791	0.7855	0.7756
Christian	0.8405	0.9226	0.9226
Female	0.8424	0.9498	0.9469
Heterosexual	0.8052	0.7613	0.8536
Hindu	0.8029	0.7822	0.8209
Homosexual_gay_or_lesbian	0.8622	0.9034	0.9167
Intellectual_or_learning_disabiliity	0.5564	0.7105	0.8237
Jewish	0.8723	0.9082	0.9265
Latino	0.7022	0.7052	0.7338
Male	0.8031	0.9338	0.9312
Muslim	0.8923	0.9312	0.9370
Transgender	0.7351	0.7301	0.7569
White	0.9237	0.9598	0.9577

Figure 6: The ROC_AUC_scores of the three models Bernoulli Naive Bayes, Linear SVC, and Logistic Regression in their ability to accurately predict identity labels. Based on this table, we conclude that Logistic Regression outperforms the other two models.

References

[1] https://creativecommons.org/share-your-work/public-domain/cc0/. *Creative Commons*. doi: https://creativecommons.org/share-your-work/public-domain/cc0/.

[2] A. Bogdanoff. Saying goodbye to civil comments. *Medium*, 2017. doi: https://medium.com/@aja_15265/saying-goodbye-to-civil-comments-41859d3a2b1d.

[3] J. S. N. T. L. V. Daniel Borkan, Lucas Dixcon. Nuanced metrics for measuring unintended bias with real data for text classification. *arXiv:1903.04561*, 2019. doi: https://arxiv.org/abs/1903.04561.

[4] B. Z. R. P. Hossein Hosseini, Sreeram Kannan. Deceiving googles perspective api built for detecting toxic comments. *Machine Learning (cs.LG)*, pages 1–4, 2017. doi: https://arxiv.org/pdf/1702.08138.pdf.

[5] J. Howard. Nb-svm strong linear baseline. *Kaggle*, 2018. doi: https://www.kaggle.com/jhoward/nb-svm-strong-linear-baseline.

[6] J. S. N. T. L. V. Lucas Dixon, John Li. Measuring and mitigating unintended bias in text classification. pages 1–7, 2017.

[7] P. Plonski. Does random forest overfit? 2019. doi: https://mljar.com/blog/random-forest-overfitting/.