ONLINE MESSAGE MODERATION WITH NATURAL LANGUAGE PROCESSING TECHNIQUES

A system design in the domain of online safety for the course Natural Language Processing(IC4253)

Undertaken By

Aneesh Poduval

(IC-A-5, GR No. 1710045)

Johnathan Fernandes

(IC-A-35, GR No. 1710168)

Pranjal Chaudhari

(ET-A 18 GR No. 1710167)

Sarthak Chudgar

(IC-A-15, GR No. 1710202)

Siddhi Jadhav

(ET-A 53 GR No. 1710015)

Under the Guidance of Prof.(Dr.) Siddharth Bhorge

Contents

Preface

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. It is broadly defined as the automatic manipulation of natural language, like speech and text, by software.

The study of natural language processing has been around for more than 50 years and grew out of the field of linguistics with the rise of computers.

Through this project, we aim to design and develop a system which utilizes Natural Language Processing classification algorithms on textual data to detect and classify abusive language, and integrate this system into a messaging service.

Case Study

Of over 5,000 students 12-17 years old in the US, 17.4% of students said they were a target of cyberbullying in 2019, compared to 16.5% in 2016 [10].

Targets of cyberbullying are at a greater risk than others of both self-harm and suicidal behaviors [11].

About 37% of young people between the ages of 12 and 17 have been bullied online. 30% have had it happen more than once [14].

23% of students reported that they’ve said or done something mean or cruel to another person online. 27% reported that they’ve experienced the same from someone else [2].

Girls are more likely than boys to be both victims and perpetrators of cyber bullying. 15% of teen girls have been the target of at least four different kinds of abusive online behaviors, compared with 6% of boys [13].

About half of LGBTQ+ students experience online harassment -- a rate higher than average [1].

Instagram is the social media site where most young people report experiencing cyberbullying, with 42% of those surveyed experiencing harassment on the platform [7].

Young people who experience cyberbullying are at a greater risk than those who don’t for both self-harm and suicidal behaviors [12].

83% of young people believe social media companies should be doing more to tackle cyberbullying on their platforms [4].

60% of young people have witnessed online bullying. Most do not intervene [5].

Only 1 in 10 teen victims will inform a parent or trusted adult of their abuse [6].

4 out of 5 students (81%) say they would be more likely to intervene in instances of cyberbullying if they could do it anonymously [3].

Programming tools

We chose to implement our system using the python language, due to its increasing popularity and range of packages that focus on data analysis, text processing, and machine learning. In particular, we have utilized nltk and sklearn packages for natural language processing and machine learning, respectively.

- The Natural Language Toolkit (NLTK) is a Python package for natural language processing. It is a tool for teaching, and working in, computational linguistics using Python.
- Scikit-learn is probably the most useful library for machine learning in Python. It contains a multitude of efficient tools for machine learning and statistical modeling.
- Tkinter is Python's de-facto, and the most popular standard GUI (Graphical User Interface) package. It is a thin object-oriented layer on top of Tcl/Tk.

Methodology

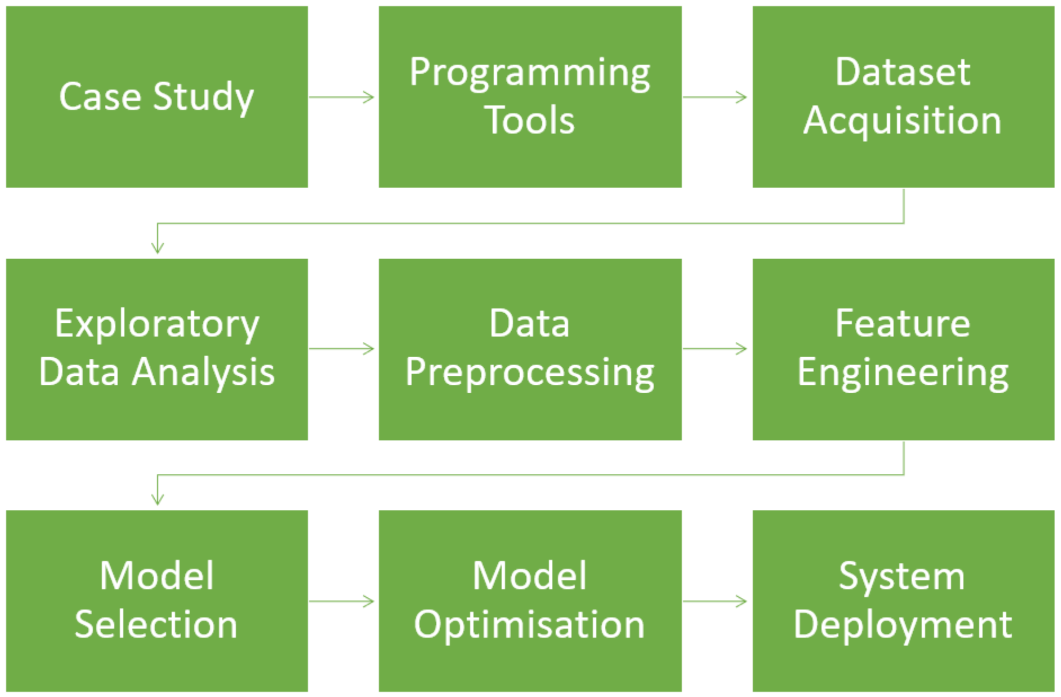


Figure 1: Block diagram indicating general flow of design.

Dataset acquisition

The dataset used was compiled by Davidson et. al. [9] by scraping public tweets, and consisted of tweets categorized into “hate”, “neither” and “offensive”. We simplified these classes into “hate speech” and “not hate speech”, and then added our own samples. In total, there were 24783 tweets out of which 20620 were abusive and 4163 were non abusive words.

Exploratory data analysis

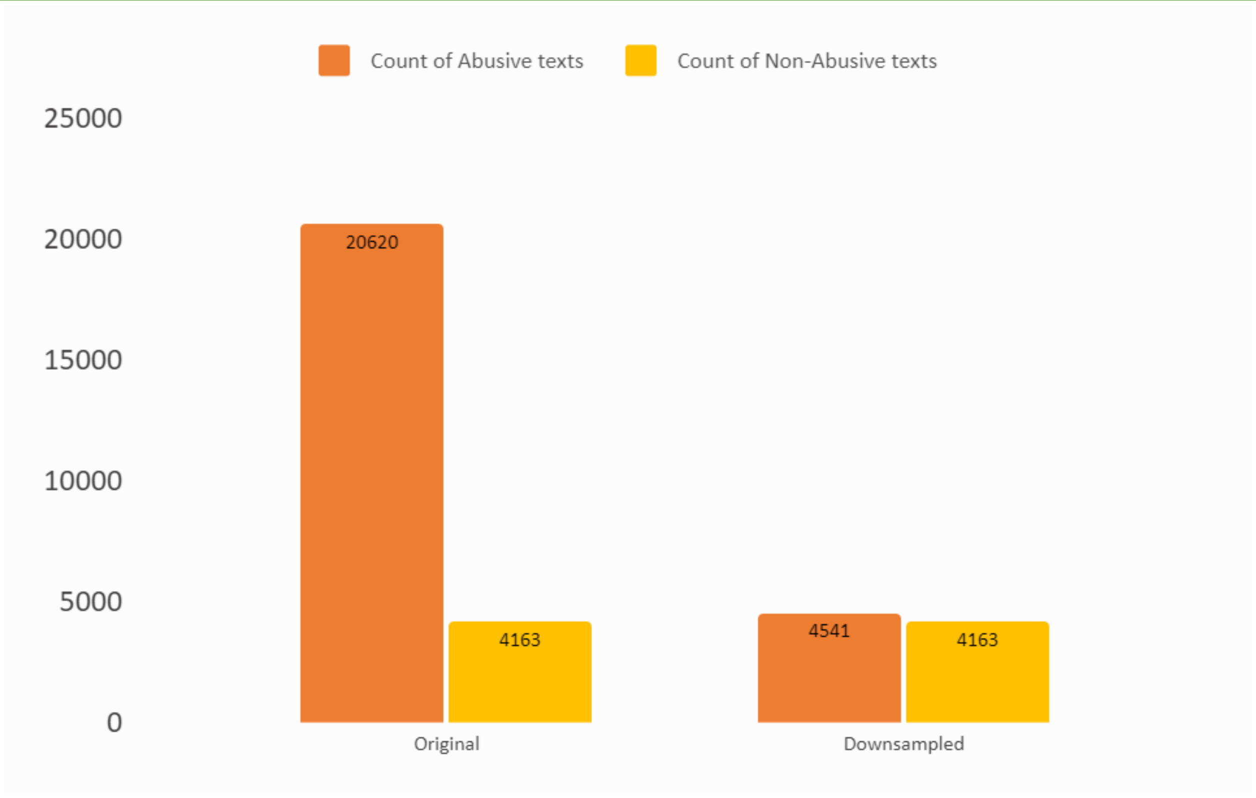


Figure 2: Number of samples per class

As shown in fig. 2, the training data is highly imbalanced, with the abusive text class having 4.95 times the number of samples as the non-abusive texts class. This can be remedied by up or down sampling the data, of which we do the latter in the data preprocessing step.

Data preprocessing

As observed during the exploratory data analysis phase, our training classes have an uneven number of samples, i.e. they are “unbalanced”. Training a model on such a dataset can lead to over-sensitivity of the model towards one class. In this case, our model would be overly sensitive to abusive text, and would run a high chance of misclassification. In order to remedy this, we randomly exclude samples until we achieve a mostly balanced set of classes as shown in fig. 2.

Since the dataset was collected from twitter via web scraping, we have to extract certain twitter-exclusive elements from it, along with applying normal NLP preprocessing techniques to the text.

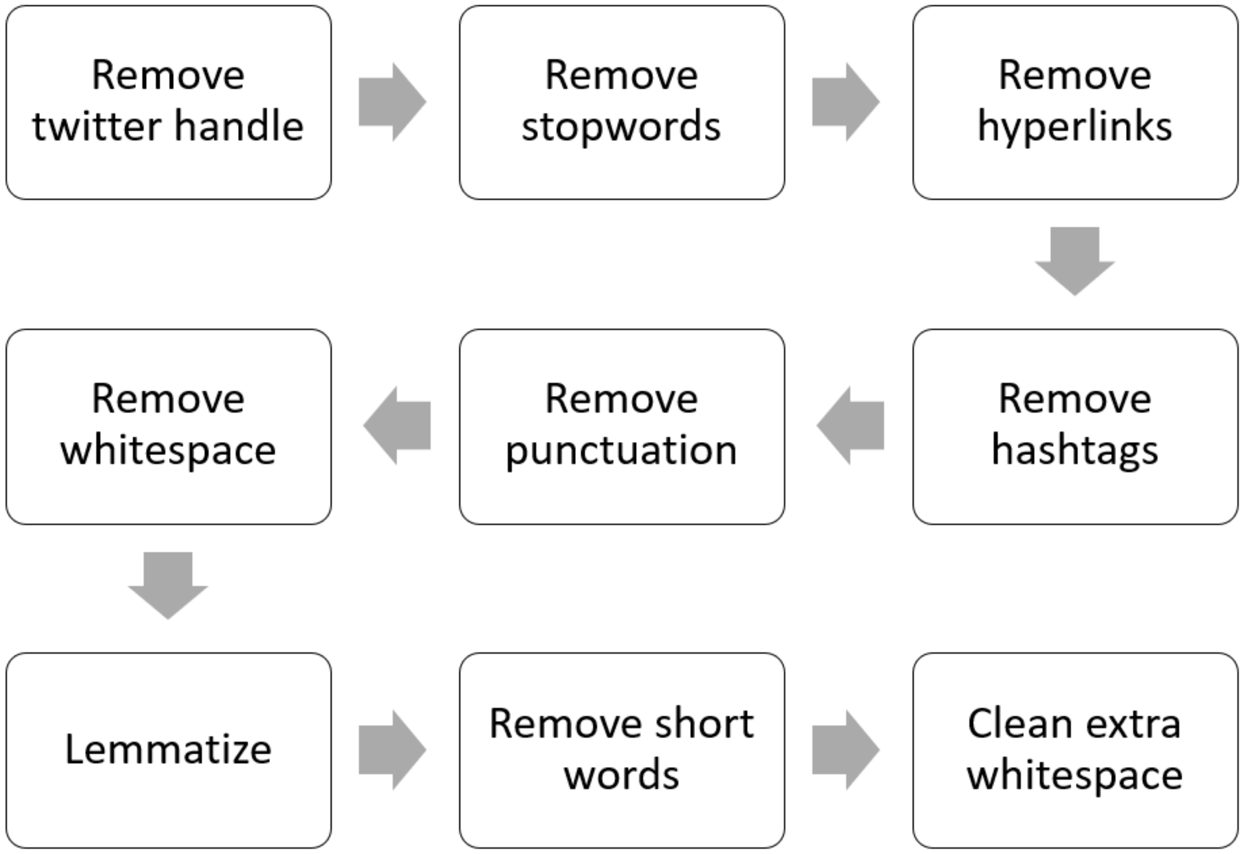
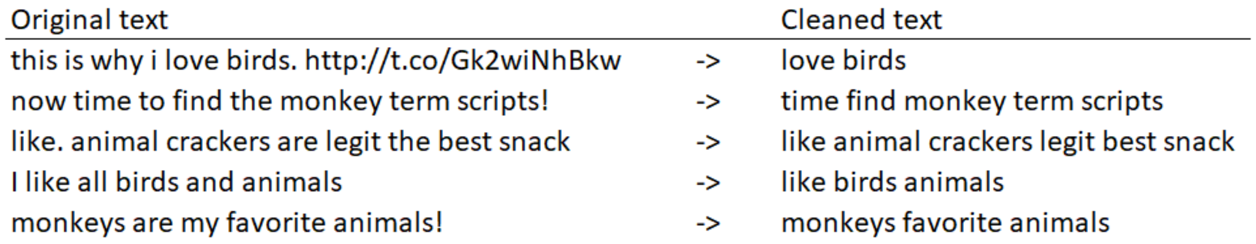


Figure 3: Text cleaning process



Original text		Cleaned text
this is why i love birds. http://t.co/Gk2wiNhBkw	->	love birds
now time to find the monkey term scripts!	->	time find monkey term scripts
like. animal crackers are legit the best snack	->	like animal crackers legit best snack
I like all birds and animals	->	like birds animals
monkeys are my favorite animals!	->	monkeys favorite animals

Figure 4: Sample text before and after cleaning

The cleaning process is documented in fig. 3, and the results in Fig. 4.

Applying these cleaning operations will improve model performance and overall efficiency during both the training and classification phases.

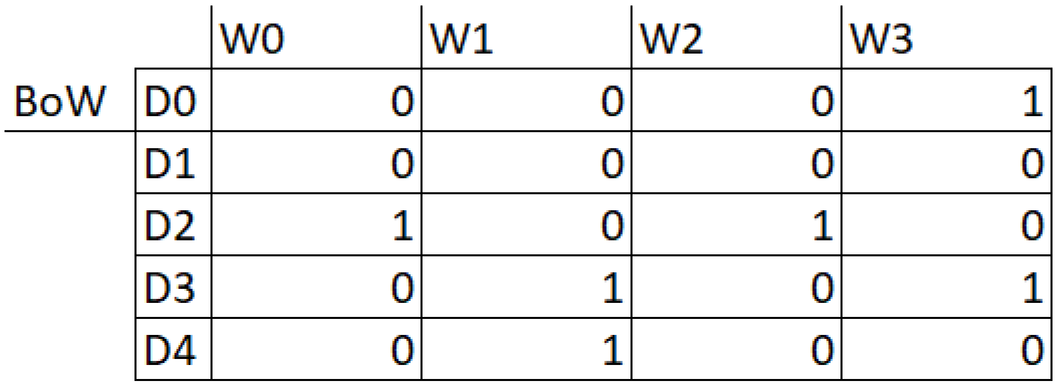
As such, the same process will be applied to new text for classification later.

Feature Engineering

Machine learning algorithms work with numerical data, and as such, we must associate our textual data with certain numerical values through a process known as “vectorization”. For this project, we explore two common vectorization techniques:

Bag of Words (BoW)

The BoW vectorization technique associates each word with its frequency in each document. It is extremely primitive, results in a sparse matrix, and assigns values based solely on frequency of words. BoW based models are usually tuned to the most common words due to the nature of the algorithm.



BoW		W0	W1	W2	W3
	D0	0	0	0	1
	D1	0	0	0	0
	D2	1	0	1	0
	D3	0	1	0	1
	D4	0	1	0	0

Figure 5: Bag of Words Feature Matrix

Term Frequency - Inverse Document Frequency (TF-IDF)

TF-IDF is a vectorization technique developed to overcome the previously stated shortcoming of the BoW model. It utilizes “term frequency”, which is essentially the same as BoW, but then also integrates an “inverse document frequency” term, which penalizes extremely common words. Thus, TF-IDF produces a good measure of word importance, rather than simple word frequency.

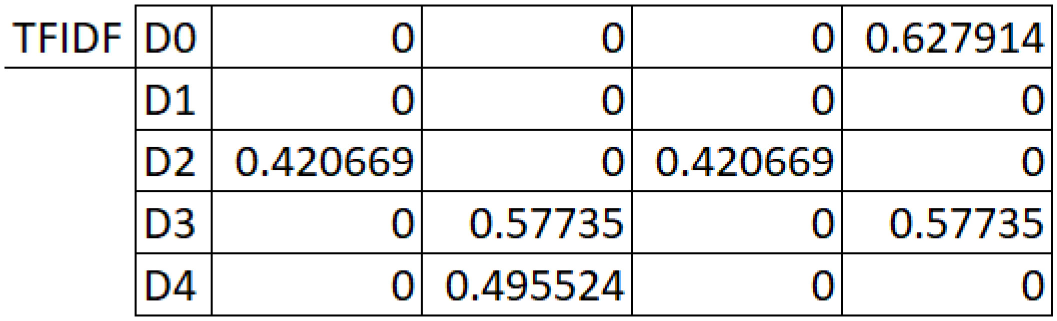
Equation 1: TF-IDF formula

$$tfidf(w) = \ tf(w) x \ idf(w)$$

$$tf(w) \ = \frac{(No.times \ of \ \text{“}w\text{”} \ appears \ in \ document)}{(total \ number \ of \ words \ in \ a \ document)}$$

$$idf(w) \ = log(\frac{No.of \ total \ documents}{No.of \ a \ document \ with \ \text{“}w\text{”} \ in \ it})$$

The formula for TF-IDF if given in eq.1



TFIDF	D0	0	0	0	0.627914
	D1	0	0	0	0
	D2	0.420669	0	0.420669	0
	D3	0	0.57735	0	0.57735
	D4	0	0.495524	0	0

Figure 6: TF-IDF Feature Matrix

Model Selection

We built and compared the performance of 3 models:

Random Forest

Random forest algorithm builds multiple decision trees and merges them together to get a more accurate and stable prediction. One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems. Owning to their versatility, they are the industry go-to for multiple classification problems.

Naive Bayes

Naive Bayes model is an easy to build probability based model which is particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Support Vector Machines

Support vector machine algorithm works by calculating a hyperplane in an N-dimensional space(with N being the number of features) that distinctly classifies the data points. SVM’s utilize a “kernel function” which can map data to alternate dimensions, leading to vastly superior performance on datasets of varying dimensions. Maximizing the support vector margin distance provides additional reinforcement so that future data points can be classified with more confidence.

Preliminary testing results

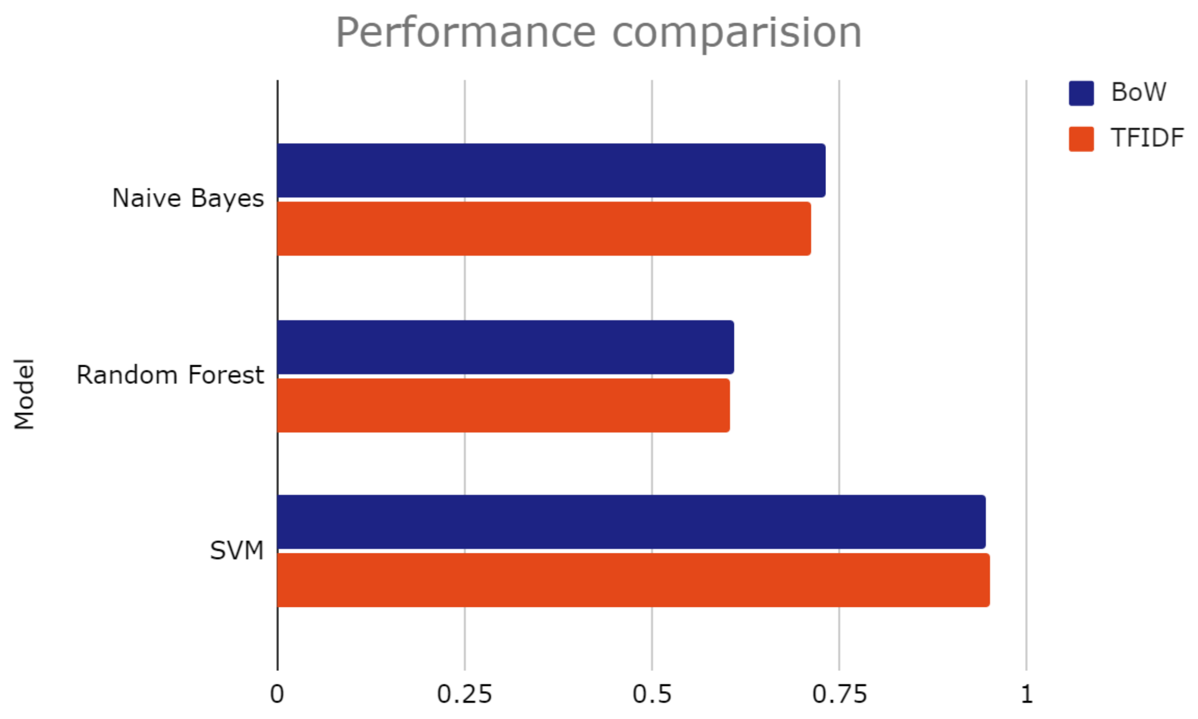


Figure 7: Performance of models with vectorizing techniques

As shown in fig. 4, SVM performed exceptionally well when compared to the other classifier models. This can be attributed to SVM's utilization of "kernel functions", which map data to alternate dimensions in order to improve classification.

Within SVM, TF-IDF shows marginally better performance due to the reasons discussed earlier. Given the nature of the vectorizing algorithms, this small advante displayed by TF-IDF over BoW in testing will likely translate into a big performance gain in real world use cases.

Model Optimization

The performance of our selected model can further be optimized by adjusting the various hyper-parameters such as number of features to consider, regularization parameter and the kernel function (all of which have their own parameters).

Due to the degree of computational complexity of the network, we cannot apply grid search techniques. We instead modify the parameters manually and measure performance.

Results

As with all classification systems, we can quantify the performance of our model using a confusion matrix:

Table 1: Confusion matrix

		Predicted	
		Non hate speech	Hate speech
Actual	Non hate speech	815	17
	Hate speech	68	841

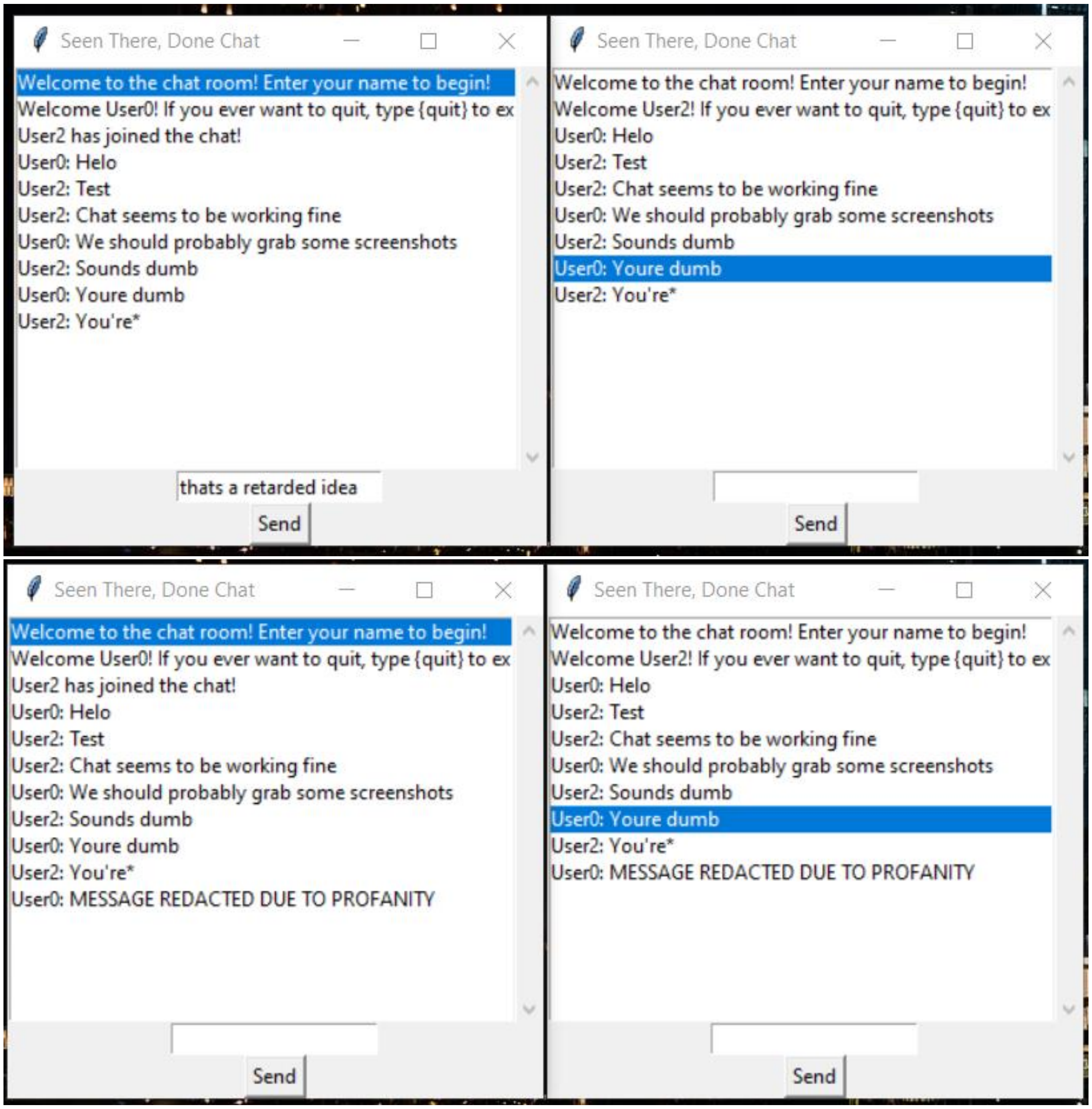
Table 2: Performance Metrics

Accuracy	0.951177
F1 Score	**0.9519**
Sensitivity	0.925193
Specificity	0.979567
Precision	0.979567

The classification data from the confusion matrix in table 1 can be used to gain further insights of the classification model, document in table 2.

Our model achieves an F-1 score of 95.19%

System Deployment


Seen There, Done Chat

```
Welcome to the chat room! Enter your name to begin!
Welcome User0! If you ever want to quit, type {quit} to ex
User2 has joined the chat!
User0: Helo
User2: Test
User2: Chat seems to be working fine
User0: We should probably grab some screenshots
User2: Sounds dumb
User0: Youre dumb
User2: You're*
```

thats a retarded idea

Send

Seen There, Done Chat

```
Welcome to the chat room! Enter your name to begin!
Welcome User2! If you ever want to quit, type {quit} to ex
User0: Helo
User2: Test
User2: Chat seems to be working fine
User0: We should probably grab some screenshots
User2: Sounds dumb
User0: Youre dumb
User2: You're*
```

Send

Seen There, Done Chat

```
Welcome to the chat room! Enter your name to begin!
Welcome User0! If you ever want to quit, type {quit} to ex
User2 has joined the chat!
User0: Helo
User2: Test
User2: Chat seems to be working fine
User0: We should probably grab some screenshots
User2: Sounds dumb
User0: Youre dumb
User2: You're*
User0: MESSAGE REDACTED DUE TO PROFANITY
```

Send

Seen There, Done Chat

```
Welcome to the chat room! Enter your name to begin!
Welcome User2! If you ever want to quit, type {quit} to ex
User0: Helo
User2: Test
User2: Chat seems to be working fine
User0: We should probably grab some screenshots
User2: Sounds dumb
User0: Youre dumb
User2: You're*
User0: MESSAGE REDACTED DUE TO PROFANITY
```

Send

Figure 8: GUI Application

In order to demonstrate the applications of our system in a practical setting, we integrated our model into an existing python-based chat application, [8] as demonstrated in figure 8. The application is based on the tkinter package and utilizes web socket technology to create a local chat application over network. Our model was integrated into the client, allowing us to censor the message before sending, or prevent it from being delivered at all.

Future Scope

The performance of the model can be improved by using a larger, more varied dataset. We can also improve performance by further tuning the model hyperparameters.

Conclusion

Through this system design, we proposed a solution to detect and moderate abusive words and hate language using Machine Learning and Natural Language Processing. After comparing multiple vectorization techniques and machine learning models, we noted that TF-IDF along with SVM performed the best, with an accuracy of 95.11%. We then integrated the model into a chat system, which upon detecting hate speech, censors the message before sending.

Acknowledgement

We would like to Prof.(Dr.) Siddharth Bhorge (class professor), Prof.(Dr.) Shripad Bhatlawande (Head of Department, Electronics and Telecommunication) Prof. Dr. (Mrs.) Shilpa Y. Sondkar (Head of Department, Instrumentation and Control) and Dr Rajesh Jalnekar (Director) for their constant guidance, and for giving us the opportunity and resources to perform this project.

Bibliography

1. "2015 National School Climate Survey." GLSEN. Accessed July 30, 2019. https://www.glsen.org/sites/default/files/2015%20National%20GLSEN%20 2015%20National%20School%20Climate%20Survey%20%28NSCS%29% 20-%20Full%20Report_0.pdf.
2. "Cyberbullying: Dealing with Online Meanness, Cruelty, and Threats." MediaSmarts. Accessed July 30, 2019, http://mediasmarts.ca/sites/mediasmarts/files/pdfs/publication-report/summary/YCWWIII_Cyberbullying_ExecutiveSummary.pdf.
3. "Cyberbullying: The Role of the Witness." Media Smarts. Accessed July 30, 2019. http://mediasmarts.ca/digital-media-literacy/digital-issues/cyberbulling/cyberbullying-role-witnesses.]
4. "Safety Net: Cyberbullying's Impact on Young People's Mental Health: Inquiry Report Summary." The Children's Society. Accessed July 30, 2019.

https://www.childrenssociety.org.uk/sites/default/files/social-media-cyberbullying-inquiry-summary-report.pdf.

5. "Safety Net: Cyberbullying's Impact on Young People's Mental Health: Inquiry Report." The Children's Society. Accessed July 30, 2019. https://www.childrenssociety.org.uk/sites/default/files/social-media-cyberbullying-inquiry-full-report_0.pdf.
6. "Stop Cyberbullying Before It Starts." National Crime Prevention Council. Accessed July 30, 2019. http://archive.ncpc.org/resources/files/pdf/bullying/cyberbullying.pdf.
7. "The Annual Bullying Survey 2017." Ditch the Label. Accessed July 30, 2019. "https://www.ditchthelabel.org/wp-content/uploads/2017/07/The-Annual-Bullying-Survey-2017-1.pdf."
8. Chaturvedi, S. (2017, Nov 23). Let's Write a Chat App in Python. Medium. https://medium.com/swlh/lets-write-a-chat-app-in-python-f6783a9ac170
9. Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. arXiv preprint arXiv:1703.04009.
10. Hinduja, Sameer, and Justin W. Patchin. "Cyberbullying: Neither an epidemic nor a rarity." European Journal of Developmental Psychology 9.5 (2012): 539-543.]
11. John, Ann, et al. "Self-harm, suicidal behaviours, and cyberbullying in children and young people: Systematic review." Journal of medical internet research 20.4 (2018): e129.
12. John, FFPH, Ann. "Self-Harm, Suicidal Behaviours, and Cyberbullying in Children and Young People: Systematic Review." Journal of Medical Internet Research. Accessed July 30, 2019. https://www.jmir.org/2018/4/e129/.
13. Marcum, Catherine D. et al. "Battle of the Sexes: An Examination of Male and Female Cyberbullying." International Journal of Cyber Criminology. http://www.cybercrimejournal.com/marcumetal2012janijcc.pdf. Anderson, Monica. "A Majority of Teens Have Experienced Some Form of Cyberbullying." Pew Research Center. Accessed July 30, 2019. https://www.pewinternet.org/2018/09/27/a-majority-of-teens-have-experienced-some-form-of-cyberbullying/.
14. Patchin, Ph.D, Justin. "2019 Cyberbullying Data." Accessed July 30, 2019, https://cyberbullying.org/2019-cyberbullying-data.

Appendix

Figures

Tables

Equations