Computational Social Science: Tweet Hate Speech Detection

Master Program of Data Science
Ludwig-Maximilians-Universität München

Xiaohan Sun

Munich, Mar 12th, 2023

Supervised by Christoph Kern

Contents

1 Introduction

In recent years, social media platforms have significantly increased, allowing people to communicate and share information worldwide in real time. They provide an accessible and convenient way to connect with friends, colleagues, families, and even strangers from all over the world, regardless of geographic location. They also could serve as a valuable source of education and information. Many organizations and individuals use social media platforms to share educational resources, research, and news on various topics, from politics and economics to health and wellness. Many individuals are provided opportunities to learn new perspectives, ideas, and cultures, promoting understanding and tolerance across diverse communities. Another tremendous benefit that social media platforms bring is entertainment and leisure. By providing a vast array of content, from videos to music and art, social media platforms offer people a much-needed break from the stresses of daily life.

While social media platforms offer many benefits, there are also several disadvantages. First, they can be used to spread misleading and false information, which can have severe consequences for society, politics, public health, etc. Also, users may get addicted, spending excessive time on social media and reducing real-world social activities and interactions. Another noteworthy disadvantage is the prevalence of hate and provocative speech. They cause tremendously negative impacts on communities and individuals. Hate speech is the expression or communication employing discriminatory language against an individual or group based on religion, nationality, race, color, gender, and other identifying factors [12]. Detecting such speech in social media is vital not only to a legal requirement but also to a moral obligation for platform owners and society as a whole.

As the ninth most popular social media platform, Twitter demonstrates its significant presence in the social media landscape and its popularity among diverse populations. It has approximately 69 million adults in the United States and is used for various purposes, including social networking, entertainment, and news consumption [10]. In recent years, various researchers and organizations have organized shared competitions and tasks for tweet hate speech detection to promote developing and evaluating state-of-the-art methods. With the creation of many benchmark datasets and evaluation metrics, researchers are able to compare and evaluate different methods.

Developing accurate and effective methods for tweet hate speech detection is critical for promoting free speech, fostering a respectful online environment, and protecting communities and individuals from the harmful effects of hate speech. Therefore, in this report, I will explore the "Tweet Hate Speech Detection" task, starting by collecting different available online datasets with the help of Twitter API. Then, after processing data cleaning and visualizations, the study focuses on automatically identifying hate speech in tweets by training several machine learning models on one dataset and testing on others. Models are implemented, including Support Vector Classifier, Artificial Neural Network, Logistic Regression, and so on. In conclusion, the project provides a comprehensive overview of tweet hate speech detection, exploring its challenges, strengths, limitations, and future research directions.

2 Data Collection

Data collection is a fundamental aspect of developing any supervised machine-learning algorithm. For this project, I decided to train machine learning models on one dataset and test it on other datasets. It has several benefits, including:

1. **Better evaluation**: Testing a model on different datasets allows us to evaluate its performance more comprehensively and rigorously. It helps us identify the model's potential limitations or weaknesses and refine it accordingly.
2. **Improved generalization**: By training a model on one dataset and testing it on others, we can improve its ability to generalize to new and unseen data. It is essential because it ensures the model can perform well in real-world scenarios, where it may encounter data not present in the training set.
3. **Reduced overfitting**: Training a model on a single dataset can sometimes lead to overfitting. The model becomes too specialized to the training data and does not generalize well to new data. Testing the model on different datasets can help reduce overfitting by exposing it to a broader range of data and ensuring it becomes manageable for a specific dataset.

Based on my research, many online Tweet datasets are available. I chose four for this project: the Hugging Face Dataset for training, the University of Copenhagen Dataset, the Aristotle University Dataset, and the HASOC 2019 Dataset for testing. For each dataset, every row will contain a tweet and the corresponding label after the manipulation, as Figure 1 shown below. For the sake of simplicity, we say a tweet contains hate speech if it has a racist or sexist sentiment associated with it, and I labeled such tweet with the label "1", and the label '0' denotes the tweet is not racist/sexist.

	text	label
0	I hate er chase because if the Bitch that work...	1
1	RT @nyctophil3: Pineapples do not belong on pi...	1
2	Niggas keep talking about women wearing weave ...	1
3	@vappywave idiot that's not gonna work. you go...	1
4	RT @ayevonnn: bruh i fucking hate people like ...	1
...	...	...
56	@realDonaldTrump Just keep golfing, instead of...	0
57	D'banj conquered wen dia wasn't social media, ...	0
58	@JillStein4Prez Perj @LSteinRoeder @regstein m...	0
59	@hamboman Hi Hamish, amendments to orders can ...	0
60	@CNNPolitics Of course it's not....Tillerson, ...	0

Figure 1: Dataset Overview.

2.1 The Hugging Face Dataset

The Hugging Face dataset originally contained training data with 31692 entries in total, and it has already been modified only to have label and tweet as the data fields. They have been collected by crowdsourcing from tweets of users, and the tweets are primarily in English. This dataset is chosen for training the machine learning models later for two specific reasons: first, this dataset contains the most rows to be large enough to provide a presentative sample of the population being studied. Second, the dataset has high quality without errors, outliers, and missing values since it has been publicly available for a long time.

2.2 The University of Copenhagen Dataset

This dataset [11] was collected to investigate predictive features for hate speech detection on Twitter [5]. It originally contained three kinds of labels: none, sexism, and racism, and only the tweet ID was associated with the corresponding label, as Figure 2 is shown below.

	572342978255048705	racism
0	572341498827522049	racism
1	572340476503724032	racism
2	572334712804384768	racism
3	572332655397629952	racism
4	575949086055997440	racism
...	...	...
16901	576359685843861505	none
16902	576612926838046720	none
16903	576771329975664640	none
16904	560595245814267905	none
16905	569363477095174145	none

Figure 2: Original Copenhagen Dataset.

This is why Twitter API came in handy. Twitter API (Application Programming Interface) is a set of programming protocols that enable developers to interact with Twitter's vast data platform. It provides a way to programmatically access and analyze Twitter data, including user information, Tweets, and trends. Twitter API is also used for a wide range of applications, including sentiment analysis, content creation, social media monitoring, and customer service, and it is helpful to developers to build custom applications that leverage Twitter data or integrate Twitter functionality into existing applications.

The primary function I implemented for this project is to fetch the corresponding Tweet

contents based on the Twitter ID. To use the Twitter API, we will first need to create a Twitter Developer account and apply for access to the API. Once approved, we can use the API to access data from Twitter's platform using a programming language such as Python or R by authenticating with the access keys. Several libraries and tools are available to make it easier to work with the Twitter API, such as the Tweepy library for Python, which is also the one implemented in this project. Figure 3 shows the content fetched using Twitter API, and only the columns of 'text' and 'label' are kept to finalize the dataset.

	text	edit_history_tweet_ids	entities	created_at	geo	public_metrics	id	label
0	Drasko they didn't cook half a bird you idiot ...	[572341498827522049]	{'hashtags': [{'start': 46, 'end': 50, 'tag': ...	2015-03-02T10:23:41.000Z	{'place_id': '017453ae077eafd3', 'coordinates'...	{'retweet_count': 0, 'reply_count': 0, 'like_c...	572341498827522049	1
1	Hopefully someone cooks Drasko in the next ep ...	[572340476503724032]	{'hashtags': [{'start': 49, 'end': 53, 'tag': ...	2015-03-02T10:19:37.000Z	NaN	{'retweet_count': 0, 'reply_count': 0, 'like_c...	572340476503724032	1
2	of course you were born in serbia...you're as ...	[572334712804384768]	{'hashtags': [{'start': 71, 'end': 75, 'tag': ...	2015-03-02T09:56:43.000Z	NaN	{'retweet_count': 0, 'reply_count': 0, 'like_c...	572334712804384768	1
3	These girls are the equivalent of the irritati...	[572332655397629952]	{'hashtags': [{'start': 95, 'end': 99, 'tag': ...	2015-03-02T09:48:33.000Z	NaN	{'retweet_count': 0, 'reply_count': 0, 'like_c...	572332655397629952	1
0	RT @YesYoureRacist: At least you're only a tin...	[446460991396917248]	{'mentions': [{'start': 3, 'end': 18, 'usernam...	2014-03-20T01:39:29.000Z	NaN	{'retweet_count': 41, 'reply_count': 0, 'like_...	446460991396917248	1

Figure 3: Contents Fetched with Twitter API.

2.3 The Aristotle University Dataset

This dataset was collected to examine the widespread use of crowdsourcing and the characterization of abusive behavior on Twitter [5]. It initially contained four kinds of identified tweets: normal, spam, abusive, and hateful. The normal tweets are labeled as 0, and hateful tweets as 1, and they are only two types kept for later use. As the University of Copenhagen dataset, it also only contained tweet ID numbers to allow me to use Twitter API to fetch the tweet text.

2.4 The HASOC 2019 Dataset

The HASOC (Hate Speech and Offensive Content) dataset is a collection of tweets curated to train machine learning models to identify hate speech and offensive content on social media platforms, especially Twitter. The dataset was created as part of a multinational initiative to address the issue of online hate speech in different languages, as most existing work in this area has been done in English [8]. Multiple languages are applied to this dataset, including Hindi, German, and English, and the dataset covers a variety of themes and topics. Each tweet has been annotated with labels indicating whether it contains hate speech, offensive content, or neither. I only used those tweets containing normal text and hate speech. This dataset is a valuable resource for developers and researchers working on machine learning and natural language processing, as it provides a large and diverse set of annotated data for training and testing models. It also contributes to efforts to

address the issue of hate speech and offensive content online, by enabling the development of more accurate and effective detection methods.

3 Data Cleaning

Data cleaning is an essential process in machine learning and data analysis that involves identifying and correcting inconsistencies, errors, and outliers in the data. The quality of the data used for model training and analysis can tremendously impact the effectiveness and accuracy of the results. The datasets involved in this project are without missing values after the data collection process. Specifically for the Tweet hate speech detection, processing and understanding the meaning of natural language text is essential. Therefore, some preprocessing techniques from Natural Language Processing (NLP) are implemented, such as stemming, lemmatization, and removal of stopwords. They could reduce the complexity of natural language text and make it easier to analyze and understand. The following steps illustrate all cleaning procedures for extracting insights and meaning from the Tweet data.

- **Basic Cleaning**: Raw tweets contain much redundant information useless for training the models later, so I did some basic cleaning. First, I removed extra space, the username, HTTP links, Greek letters, and some useless words such as "&amp", "rt", and "mkr". Then, I changed some slang words such as 'lyk' to 'like', 'whateva' to 'whatever', 'ttyl' to 'talk to you later', and so on. After the hashtags had been fetched to another new column, I removed all punctuations.
- **Stemming**: Stemming is the process of reducing words to their root form, or stem, by removing suffixes and prefixes. This can help to reduce the number of unique words in a text corpus and simplify the analysis process. For example, the words "jumping", "jumped", and "jumps" would all be reduced to the stem "jump".
- **Lemmitization**: Lemmatization is a similar process to stemming, but instead of reducing words to their root form, it converts them to their base or dictionary form, or lemma [2]. This can help to preserve the meaning of the original words and reduce ambiguity. For example, the words "am", "are", and "is" would all be converted to the lemma "be".
- **Removal of Stopwords**: Stopwords are common words that are frequently used in natural language text but do not carry much meaning, such as "the", "a", and "and". Removing stopwords can help to reduce the noise in a text corpus and make it easier to focus on the more meaningful words.

Figure 4 and Figure 5 show some examples of the Twitter text before and after cleaning. Hashtag contents are kept in each sentence since they can be helpful for analysis, particularly for identifying popular trends, topics, or sentiments. Removing them can potentially remove important context that helps with analysis. They have also been fetched to another new "Hash Words" column for visualizing tweet data, particularly for highlighting the most common topics or themes in the tweet data. Figure 6 shows the hashtags extracted from the examples illustrated above, and texts without hashtags will be labeled

with “No hashtags”.

text
#abc2020 getting ready 2 remove the victums frm #pulseclub #prayfororlando
for her #bihday we got her a #nose #job @user √∞¬ü¬é¬àv∞¬ü¬ê¬ðv∞¬ü¬é¬åv∞¬ü¬é¬Çv∞¬ü¬é¬Å #bihday #petunia we love you v∞¬ü¬
off to concelebrate at the #albanpilgrimage for the first time. @user
@user let the scum-baggery begin.....
thank you! v∞¬ü¬ò¬çv∞¬ü¬ò¬Ü super love it! v¢¬ù¬§vØ¬∏¬è zpamdelacruz #wedding# @ dolores, capas tarlac.
a scourge on those playing baroque pieces on piano beyond belief
@user lets fight against #love #peace
happy fathervȼ¬Ã¬ôs day, mr. rayos #video #fathers #day #rayos #world #hotvideo #videos
@user ascot times with this babe v¢¬ù¬§vØ¬∏¬èv¢¬ù¬§vØ¬∏¬è #ascot #fashion #monochrome #style #instahappyday
the weekend..is here!!v∞¬ü¬ô¬åv∞¬ü¬ò¬âv∞¬ü¬ò¬òv∞¬ü¬í¬ôv∞¬ü¬ò¬ä #selfie #yolo #xoxo #like4like
happy at work conference: right' mindset leads to culture-of-development organizations #work #mindset
christina grimmie's last performance before being shot... via @user #christinarip #voice #christinagrimmie
we are ready to dance #roar #preschoolers #students #proudv¢¬Ä¬¶
you've really hu my feelings :(

Figure 4: Examples of the Data before Cleaning.

clean_tweet
abc2020 get readi 2 remov victum frm pulseclub prayfororlando
bihday got nose job ddpddd bihday petunia love
concelebr albanpilgrimag first time
let scumbaggeri begin
thank dd super love ai zpamdelacruz wed dolor capa tarlac
scourg play baroqu piec piano beyond belief
let u fight love peac
happi fathera day mr rayo video father day rayo world hotvideo video
ascot time babe aiai ascot fashion monochrom style instahappyday
weekendi hereddddd selfi yolo xoxo like4lik
happi work confer right mindset lead cultureofdevelop organ work mindset
christina grimmi last perform shot via christinarip voic christinagrimmi
readi danc roar preschool student prouda
realli hu feel

Figure 5: Examples of the Data after Cleaning.

Hash Words
#abc2020 #pulseclub #prayfororlando
#bihday #nose #job #bihday #petunia
#albanpilgrimage
No hashtags
#wedding#
No hashtags
#love #peace
#video #fathers #day #rayos #world #hotvideo #videos
#ascot #fashion #monochrome #style #instahappyday
#selfie #yolo #xoxo #like4like
#work #mindset
#christinarip #voice #christinagrimmie
#roar #preschoolers #students #prouda|
No hashtags

Figure 6: Examples of Hashtags Extracted.

4 Exploratory Data Analysis

EDA, or exploratory data analysis, is a widely used approach involving systematically examining and summarizing datasets to identify relationships, patterns, and other data characteristics, often with the aid of visualizations such as charts, plots, and graphs [3]. It involves implementing various techniques, such as statistical analysis and data transformation, to gain data-driven insights and understand its underlying structure. EDA could also help identify missing values, outliers, and other issues in order to inform further analysis, such as building predictive models. I analyzed the Hugging Face and HASOC datasets for this section based on their label distribution, the most popularly used hashtags, and the word frequency since they have some contrasting characteristics.

- **Hugging Face Dataset**: From the boxplot of the label distribution, as Figure 7 shows, the difference in the text between labels equal 1 or 0 is not apparent, as the average text length is about 50 for the label equals 0 and almost 60 for the label equals 1. However, there are more outliers for text length when the label equals 0, meaning that much non-hate speech tends to be longer.

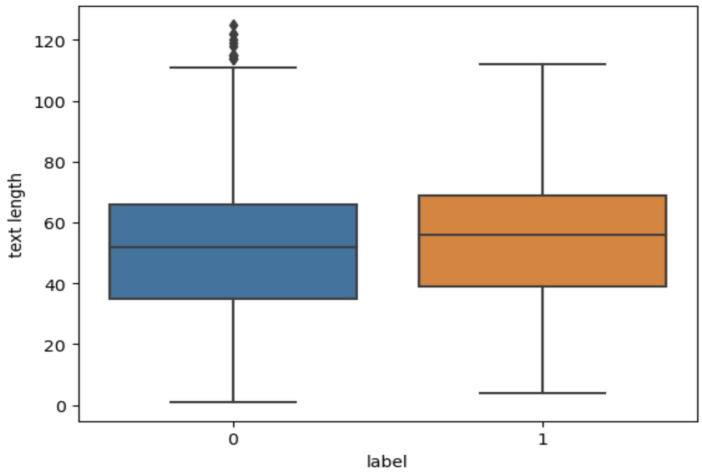


Figure 7: Hugging Face Dataset Label Distribution.

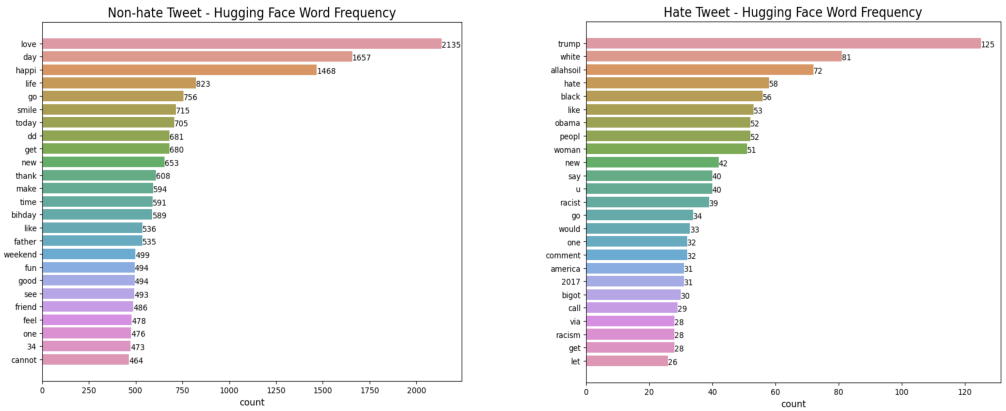
I also compared the difference between Wordcloud of hashtags. Based on Figure 8a shows, it is evident that for label 0, the most common hashtags are positive and neutral, like love, smile, life, friend, and so on, indicating these tweets are likely non-hate speech. For label 1, as Figure 8b shows, the most common hashtags are more controversial and have negative connotations with words like trump, Obama, allahsoil (a common hashtag associated with tweets involving Islamic references), bigot, hate, etc.

(a) Hashtags with Label 0.

(b) Hashtags with Label 1.

Figure 8: Wordcloud of Hashtags in Hugging Face Dataset.

Such distinction also appeared from the word frequency plots, as Figure 9 shows. For the Hugging Face dataset, we can see the most frequent words of non-hate tweets are love, day, happy, life, etc.; of hate tweets, words are trump, white, allahsiol, hate, etc. This information provides valuable insights into the characteristics of the texts in each class and can have several implications for model training. First, the presence of distinctive words in each class can help the models learn patterns that effectively differentiate between hate speech and non-hate speech. Models that can leverage these differences are more likely to achieve better performance in terms of accuracy, precision, recall, and F1 score. Second, the models are more likely to generalize well to unseen data and be robust if the word frequency differences between the two classes are significant. Third, understanding the word frequency can help to select appropriate models for the task. If the differences between classes are apparent, simpler models like Logistic Regression might perform well. On the other hand, if the differences are more subtle, more complex models like ANNs, RNNs, or transformer-based models might be better suited for capturing the patterns in the data. As the differences between classes are evident in the Hugging Face dataset, I chose it as the training dataset to train machine learning models later.



(a) Non-hate Tweets.

(b) Hate Tweets.

Figure 9: Word Frequency in Hugging Face Dataset.

- **HASOC Dataset**: Compared with the Hugging Face dataset, the texts in the HASOC dataset tend to be longer. As Figure 10 shows, the average text length is about 100, no matter for labels equal 0 or 1. For text classification tasks, longer texts usually contain more words, consequently, more features. This could lead to sparser feature vectors when conducting TF-IDF later, making it harder for some models to identify patterns in the data.

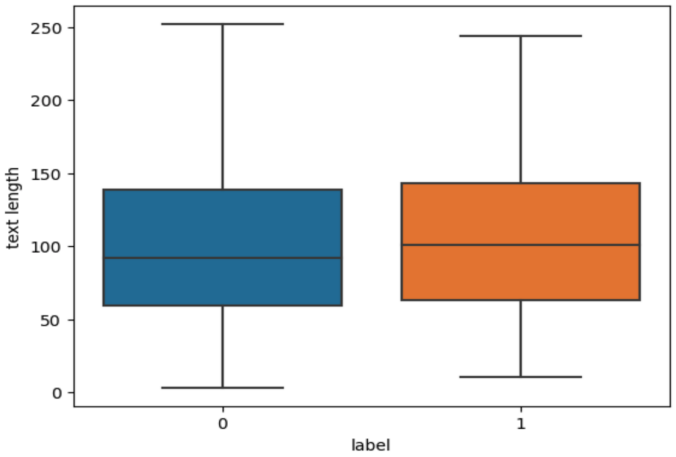
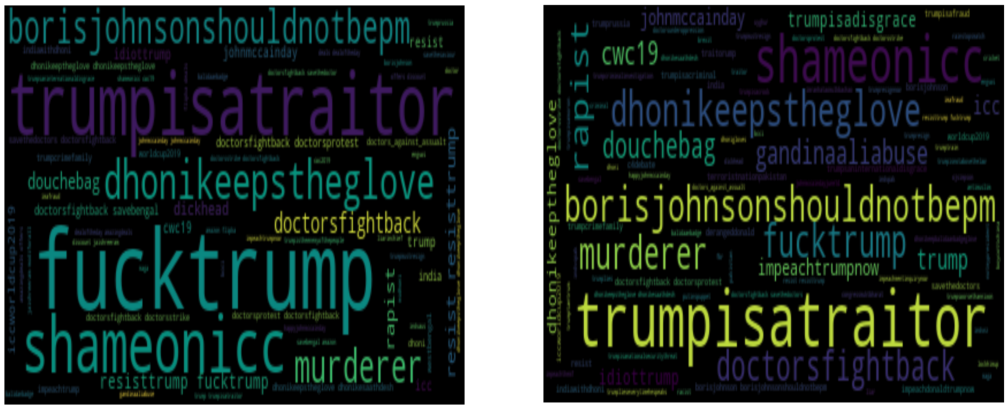


Figure 10: HASOC Dataset Label Distribution.

From the word clouds of hashtags used for both labels, they are very similar in the HASOC dataset. As Figure 11 shows, even hashtags for non-hate tweets are not positive and neutral. It indicates that the text features might not be distinctive enough to differentiate between hate and non-hate speech effectively. This could lead to challenges in training and evaluating machine learning models for hate speech detection.



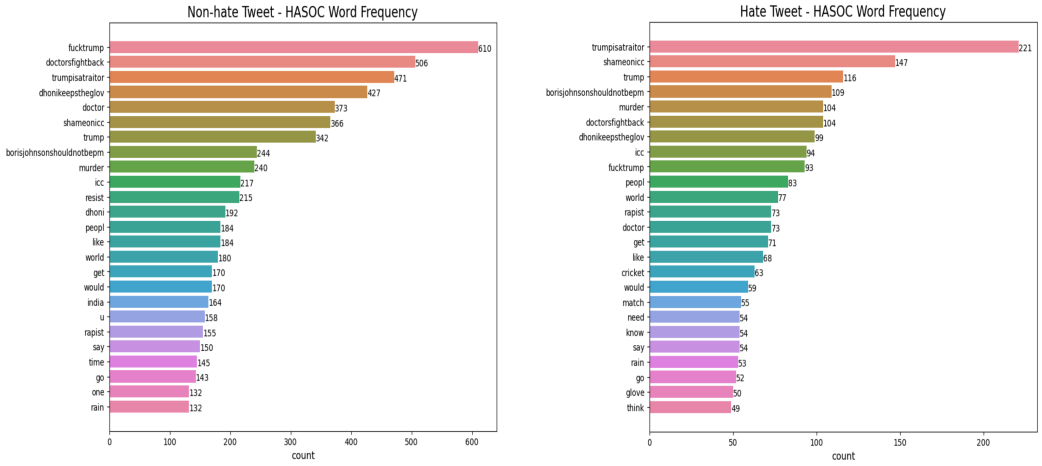
(a) Hashtags with Label 0.

(b) Hashtags with Label 1.

Figure 11: Wordcloud of Hashtags in HASOC Dataset.

There is considerable overlap between the most frequent words in each class for the

HASOC dataset, as Figure 12 shows. It might be more challenging for the models to differentiate between the classes, potentially leading to lower performance.



(a) Non-hate Tweets. (b) Hate Tweets.

Figure 12: Word Frequency in HASOC Dataset.

5 Models

I trained several machine-learning models on the Hugging Face dataset to detect tweet hate speech. The models include Artificial Neural Network, Random Forest Classifier, Support Vector Machine, Logistic Regression, and Gradient Boosting. These models were selected because they are commonly used in text classification tasks. I then tested the trained models on the HASOC, Aristotle, and Copenhagen datasets. These datasets were selected to evaluate the model's performance on different datasets and to assess their generalization ability. The following sections describe the models I used, including their architectures and training procedures. I also presented the results of the experiments and provided an analysis of the performance of each model on the test datasets.

5.1 TF-IDF

TF-IDF (term frequency-inverse document frequency) is implemented in conjunction with the machine learning models below. It is a widely used technique in natural language processing for extracting important features from textual data and implementing tasks such as sentiment analysis and text classification.

The working principle for TF-IDF is to assign weights to each word in a document based on its frequency and importance across a corpus of documents [9]. There are two values for each word in a document that are needed to be calculated: term frequency (TF) and inverse document frequency (IDF). TF quantifies the frequency of a word in a document and assigns a higher weight to words that occur more frequently. IDF quantifies the significance of a word across a collection of documents and gives a higher weight to rare words across the corpus. The TF-IDF score for each word is computed as the product

of its TF and IDF values. Here is one example of a sentence from the HASOC dataset: "relat doctor secur issu polit issu guess want make polit doctorsfightback didivsdoctor doctorsstrik". The TF-IDF score calculated for this sentence is shown in Figure 13. In this case, the words "issu", "polit", "doctor" have the highest TF-IDF scores, indicating that they are the most important words in the sentence. However, for words with scores of 0, they are insignificant in the sentence.

	tfidf
issu	0.558225
polit	0.511527
doctor	0.327857
secur	0.307218
relat	0.304153
guess	0.263079
want	0.183077
make	0.172092
ottbik	0.000000
ottawa	0.000000
ottawagiveaway	0.000000

Figure 13: TF-IDF Example.

When applied to the task of detecting hate speech in tweets, the use of TF-IDF has the potential to enhance the performance of machine learning models by extracting salient features from the tweets, which can aid in discriminating between hate speech and non-hate speech tweets.

5.2 Implemented Models

- **Artificial Neural Network**: ANN is a type of deep learning model designed to replicate the structure of the human brain [1]. It comprises multiple layers of interconnected neurons that transform and process the input data. For the hate speech detection task, the model architecture consists of three dense layers with activation functions and dropout regularization to prevent overfitting. As Figure 14 shows, the first dense layer has 128 units. It uses the ReLU activation function, commonly chosen for neural networks, since it can deal with non-linear relationships in the data. The input shape is decided by the number of features in the training data. After the first dense layer, there is a dropout layer with a dropout rate of 0.2. During training, it randomly drops out a fraction of the neurons in the layer to regularize. Finally, there is a third dense layer with 128 units and a softmax activation function, which outputs a probability distribution over the two possible classes: hate speech and non-hate speech.

```
# define and fit the model
def get_model(trainX,trainy):
    model = tf.keras.Sequential([
                            tf.keras.layers.Dense(128, activation='relu', input_shape=(trainX.shape[1],)),
                            tf.keras.layers.Dropout(0.2),
                            tf.keras.layers.Dense(32, activation='relu'),
                            tf.keras.layers.Dropout(0.2),
                            tf.keras.layers.Dense(128, activation='softmax')])
    model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    model.fit(trainX, trainy, epochs=10, verbose=2)
    return model
```

Figure 14: Artificial Neural Network Model.

- **Ranfom Forest Classifier**: Random Forest Classifier falls under the category of ensemble learning, which combines multiple decision trees to improve the robustness and overall accuracy of the model. Each tree is trained on a random subset of the data, and the final decision is determined by combining the predictions of all the trees through majority voting [7]. For this specific task, decision trees are set to 100.
- **Support Vector Classifier**: SVM belongs to binary classification models that separate the data into different classes. It works by identifying a hyperplane that can best separate two classes of data with maximum margin and uses this hyperplane to make predictions about the input new data. Specifically, the model predicts which category a new data point belongs to based on which side of the hyperplane the point falls on.
- **Logistic Regression**: Logistic Regression is a linear model that employs a logistic function to predict the probability of an input belonging to a specific class. A line is fitted to the input data, which separates the two classes, and makes predictions based on the estimated probability.
- **Gradient Boosting**: Gradient Boosting is also an ensemble learning model that combines multiple weak learners, such as decision trees, to become a strong learner. Each tree is built based on how it corrects the previous tree's errors. This is done by assigning weights to the data points, with higher weights given to the misclassified data points. In subsequent iterations, Gradient Boosting focuses more on those misclassified data points and continues to build trees that improve the model's accuracy [4]. For the tweet hate speech detection task, this model can be constructed to identify hate speech based on the text of a tweet. After the model is trained on a dataset of labeled tweets, it can be used to predict whether new tweets contain hate speech.

5.3 Results

Table 1 to Table 5 below show the results from the five models. I derived the analysis based on the average evaluation metrics across all five models. We can conclude the following:

1. The Aristotle dataset has the highest average accuracy, 0.780, among the three datasets, indicating that the models perform relatively better in classifying tweets as hate speech or non-hate speech on this dataset.
2. HASOC dataset has the lowest average performance across all evaluation metrics, showing that the models struggle more with the dataset than the other two. This is also the analysis result we achieved from the exploratory data analysis part. Because the hashtags and the most frequent words used in this dataset are not very distinctive, leading to poorer model results.
3. The Copenhagen dataset has the highest average precision, 0.7114, and an F1 score of 0.299. It suggests that the models achieve better positive predictive values and a balance between precision and recall on this dataset.

Based on these findings, choosing datasets would depend on the specific requirements. For example, if the focus is on maximizing the correct classification of tweets, Aristotle may be the preferred dataset. However, the Copenhagen dataset would be more appropriate if the focus is to achieve a better balance between precision and recall.

I also calculated the average performance of each model across all three datasets to determine the best model for the tweet hate speech detection task. We can conclude the following:

1. Artificial Neural Network has the highest average recall, 0.183, and F1 score, 0.255. It reflects that the best balance between identifying hate speech instances and minimizing false positives is reached from this model.
2. Logistic Regression and Gradient Boosting modes have the lowest average recall and F1 scores, indicating that they may not be as effective in identifying hate speech instances.
3. Random Forest Classifier has the highest average accuracy, 0.772, and the second-highest average precision, 0.575, showing strong performance in classifying tweets and demonstrating relatively good positive predictive values. It indicates that the model can effectively identify hate speech in tweets while maintaining reasonable precision in its predictions.

	Accuracy	**Precision**	**Recall**	**F1**
HASOC	0.680	0.229	0.125	0.161
Aristotle	0.776	0.495	0.125	0.199
Copenhagen	0.756	0.623	0.301	0.406

Table 1: Aritifical Neural Network Result.

	Accuracy	Precision	Recall	F1
HASOC	0.735	0.276	0.045	0.078
Aristotle	0.781	0.620	0.054	0.099
Copenhagen	0.800	0.829	0.352	0.494

Table 2: Random Forest Classifier Result.

	Accuracy	Precision	Recall	F1
HASOC	0.718	0.272	0.084	0.128
Aristotle	0.782	0.568	0.098	0.166
Copenhagen	0.768	0.719	0.267	0.390

Table 3: Support Vector Classifier Result.

	Accuracy	Precision	Recall	F1
HASOC	0.744	0.271	0.020	0.038
Aristotle	0.782	0.701	0.040	0.075
Copenhagen	0.734	0.710	0.070	0.128

Table 4: Logistic Regression Result.

	Accuracy	Precision	Recall	F1
HASOC	0.751	0.431	0.022	0.042
Aristotle	0.779	0.580	0.031	0.058
Copenhagen	0.729	0.675	0.042	0.079

Table 5: Gradient Boosting Result.

6 Conclusion

Tweet hate speech detection can significantly address social science problems. For example, by analyzing patterns in hate speech, researchers can gain insights into the intentions, motivations, and emotions of individuals engaging in harmful online behavior. Such identification contributes to developing interventions to reduce such behavior. Also, detecting and analyzing hate speech on social media platforms can make significant progress in developing evidence-based guidelines, recommendations, and policies for platform providers [6]. Thus, a safer online space that balances the need to protect users from harmful content and freedom of speech could be created.

This study aims to evaluate the performance of five machine learning models, including Artificial Neural Network (ANN), Random Forest Classifier, Support Vector Classifier, Logistic Regression, and Gradient Boosting. The analysis revealed that Random Forest Classifier demonstrates the best overall performance in terms of accuracy, precision, recall,

and F1 score across the three testing datasets: HASOC, Aristotle, and Copenhagen. Such a result could have important implications for the development of automated tools for hate speech detection. Additionally, the findings can guide practitioners and researchers in selecting appropriate models and techniques.

The cross-dataset evaluation conducted in this study highlights the importance of developing robust models that can handle the diverse nature of hate speech found on social media platforms and generalize well to different contexts. By training the models on the Hugging Face dataset and testing on others, the study enriched the models' ability to fit new data, which is fundamental and essential for real-world applications where data distributions may vary. I also used the Copenhagen dataset to train each model and compare the performances to address the importance better. However, the model performance still did not improve much (Please see the results from my GitHub). Therefore, making the models generalized is still a big challenging problem nowadays.

The methodology employed in this study, including data collection, data cleaning, exploratory data analysis, and model training, provided data-driven insights into the characteristics of the texts in each class. Such procedures could contribute to model selection and feature engineering. However, there may be limitations considering the choice of datasets and potential imbalances or biases in the data that could influence the results. Indeed, each dataset I chose for this study has fewer hate tweets, and the manual addition of such tweets to the dataset is needed for a better model performance in the future. Moreover, the traditional machine learning models may not fully capture the complexity of language patterns in hate speech compared to more advanced and state-of-the-art natural language processing techniques.

Several methods could be explored to improve no matter for the model performance or the model generalization. This may include developing new features for the models or employing more advanced techniques, such as deep learning-based approaches or transformers. Also, future studies could examine the performance of models in cultural contexts or in various languages to develop more versatile and effective hate speech detection platforms or tools.

In conclusion, this study adds to the expanding body of research on tweet hate speech detection and its potential societal impact. By investigating a range of machine learning models and evaluating their performance across multiple datasets, the findings provide valuable insights for developing efficient and reliable hate speech detection tools. The results emphasize the significance of model generalization and robustness, as well as the potential advantages of automated hate speech detection in fostering a safer and more inclusive online space. As natural language processing continues to progress, future research can build upon these findings to devise more advanced and effective methods for identifying and addressing hate speech on social media platforms.

7 Acknowledgment

I would like to express my sincere gratitude to Professor Kern, who provided invaluable support and guiudance throughout the course. His insights, expertise, and constructive

feedback has been instrumental in shaping the final outcome of my work. I am truly grateful for his unwavering support and dedication to helping me achieve my goals.

References

[1] Agatonovic-Kustrin, S. and Beresford, R. [2000]. Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research, *Journal of pharmaceutical and biomedical analysis* **22**(5): 717–727.

[2] Balakrishnan, V. and Lloyd-Yemoh, E. [2014]. Stemming and lemmatization: A comparison of retrieval performances.

[3] Behrens, J. T. [1997]. Principles and procedures of exploratory data analysis., *Psychological methods* **2**(2): 131.

[4] Bentéjac, C., Csörgő, A. and Martínez-Muñoz, G. [2021]. A comparative analysis of gradient boosting algorithms, *Artificial Intelligence Review* **54**: 1937–1967.

[5] Datascisteven [n.d.]. Datascisteven/automated-hate-tweet-detection: Developing a classification model to detect hate tweets ready for deployment using various nlp techniques.
URL: *https://github.com/datascisteven/Automated-Hate-Tweet-Detection*

[6] Davidson, T., Warmsley, D., Macy, M. and Weber, I. [2017]. Automated hate speech detection and the problem of offensive language, *Proceedings of the international AAAI conference on web and social media*, Vol. 11, pp. 512–515.

[7] Liu, Y., Wang, Y. and Zhang, J. [2012]. New machine learning algorithm: Random forest, *Information Computing and Applications: Third International Conference, ICICA 2012, Chengde, China, September 14-16, 2012. Proceedings 3*, Springer, pp. 246–252.

[8] Mandl, T., Modha, S., Majumder, P., Patel, D., Dave, M., Mandlia, C. and Patel, A. [2019]. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages, *Proceedings of the 11th forum for information retrieval evaluation*, pp. 14–17.

[9] Ramos, J. et al. [2003]. Using tf-idf to determine word relevance in document queries, *Proceedings of the first instructional conference on machine learning*, Vol. 242, Citeseer, pp. 29–48.

[10] Takhteyev, Y., Gruzd, A. and Wellman, B. [2012]. Geography of twitter networks, *Social networks* **34**(1): 73–81.

[11] Waseem, Z. and Hovy, D. [2016]. Hateful symbols or hateful people? predictive features for hate speech detection on twitter, *Proceedings of the NAACL Student Research Workshop*, Association for Computational Linguistics, San Diego, California, pp. 88–93.
URL: *http://www.aclweb.org/anthology/N16-2013*

[12] *What is hate speech?* [n.d.].
URL: *https://www.un.org/en/hate-speech/understanding-hate-speech/what-is-hate-speech*