Data Glacier

Your Deep Learning Partner

Data Science Intern at Data Glacier

Project: Hate Speech Detection Using Transformers

Week 13: Final Deliverable

Team Name: Fibo

Team Members: Mahyar Arani

Email: arani.mahyar@gmail.com

Batch code: LISUM19

Submission date: May 30, 2023

Submitted to: Data Glacier

Data Intake Report

Name: **Twitter Hate Speech**

Report date: **04/20/23**

Internship Batch: **LISUM19**

Version: **1.0**

Data intake by: **Mahyar Shahpouri Arani**

Data intake reviewer: Data Glacier

Data storage location: https://www.kaggle.com/datasets/vkrahul/twitter-hate-speech?select=train_E6oV3lV.csv

Tabular data details:

Total number of observations	32k
Total number of files	1
Total number of features	3
Base format of the file	.csv
Size of the data	3.1 MB

Deduplication Approach:

- **No missing values**

Business Understanding:

Detecting hate speech in social media is an important task for businesses, governments, and society as a whole. In today's interconnected world, where social media platforms play a significant role in communication, the impact of hate speech cannot be ignored. It not only poses a threat to the mental health and well-being of individuals who are targeted, but it also has broader societal implications. Hate speech has the potential to fuel the spread of misinformation, contribute to the polarization of communities, and even incite violence. Consequently, developing a robust hate speech detection model has become paramount.

By proactively identifying and addressing instances of hate speech on social media, businesses and governments can foster a safer and more inclusive online environment. Such efforts can have a positive ripple effect, leading to increased user engagement, better brand perception, and enhanced customer loyalty. Additionally, businesses can leverage hate speech detection models to gain insights into their customers' sentiments, allowing them to tailor their products and services accordingly and improve overall customer satisfaction.

Problem Description:

The problem at hand revolves around the detection of hate speech in Twitter tweets using a machine learning model. Hate speech, characterized by the use of derogatory or discriminatory language targeting individuals or groups based on their religion, ethnicity, nationality, race, color, ancestry, sex, or other identity factors, poses a significant challenge in today's digital landscape. The objective of this project is to develop a highly accurate hate speech detection model capable of classifying tweets as containing hate speech or not.

Data Understanding

The dataset provided for this project consists of a comprehensive collection of tweets and their associated labels. In total, there are 32,000 tweets, each uniquely identified by an ID. These tweets have been meticulously labeled based on their sentiment, where a sentiment score of 0 indicates a negative sentiment and a score of 1 indicates a positive sentiment.

To provide further granularity and enable more detailed analysis, the dataset has been divided into 20 distinct bins based on the length of the tweets. These bins span a range of tweet lengths, from 1,599 characters to 31,962 characters. Within each bin, there are either 1,598 or 1,599 tweets, and the distribution of labels is nearly equal among the bins.

Upon analyzing the sentiment scores assigned to the tweets, it is evident that the majority of the tweets (approximately 29,720) fall within the sentiment score range of 0 to 0.05. In contrast, only a smaller subset of tweets (2,242) have been labeled with sentiment scores between 0.95 and 1.00. This distribution indicates that the dataset is predominantly skewed toward negative sentiments, with a higher frequency of tweets exhibiting lower sentiment scores.

Overall, the dataset appears to be well-suited for sentiment analysis tasks, particularly for the classification of tweets based on their sentiment scores. However, to ensure the data's quality and consistency, further analysis and preprocessing steps may be necessary.

Data Preprocessing

To ensure the data is ready for hate speech detection modeling, a series of preprocessing steps were applied to enhance its quality and optimize the performance of the classification model.

The initial step in the preprocessing pipeline involved sentence tokenization. By breaking down the text into individual sentences, we gain a granular understanding of the sentiment and structure within each tweet, enabling more accurate analysis.

Subsequently, word tokenization was performed to segment the text into individual words, serving as the basic units for further analysis. This process facilitates various text processing tasks, including feature extraction and model training, by providing a more detailed representation of the data.

To improve the data quality and streamline subsequent analysis, punctuation marks were removed. This step reduces noise and ensures that the presence of punctuation does not interfere with the hate speech detection model's performance.

Another preprocessing step involved the removal of URLs. Since URLs do not provide relevant information for hate speech detection, their elimination aids in simplifying the analysis and minimizing potential distractions or irrelevant content.

In addition, special characters and symbols such as mentions and hashtags were also removed from the text. These elements typically carry limited value in hate speech detection and may introduce noise into the dataset. By eliminating them, we create a cleaner and more focused dataset for modeling purposes.

To further refine the data, stop words were removed. Stop words are common words in a language (e.g., "the," "is," "and") that often add little semantic meaning to the text. By eliminating these words, we can reduce noise and potentially improve the classification performance by focusing on more informative terms.

To enhance the accuracy of the hate speech classification model, part-of-speech tagging was applied to the tokenized text. This process assigns specific grammatical categories (e.g., noun, verb, adjective) to each word, enabling more nuanced analysis and capturing the linguistic context within the tweets.

Additionally, stemming was performed by reducing words to their base or root form. This process aims to standardize the vocabulary by eliminating variations of words with similar meanings. By reducing words to their core form, we can enhance the classification model's performance by grouping related words together.

Furthermore, lemmatization was applied as an alternative method to normalize the text. Lemmatization reduces words to their dictionary form, capturing their intended meaning. This approach further enhances the performance of the classification model by ensuring consistency and accuracy in word representation.

In summary, the data preprocessing stage involved several important steps, including sentence and word tokenization, punctuation, URL, and special character removal, stop word elimination, part-of-speech tagging, stemming, and lemmatization. Each of these steps contributes to refining the dataset, optimizing its quality, and preparing it for subsequent hate speech detection modeling.

Data Cleansing and Transformation

To ensure the data is well-prepared for classification modeling, a series of data cleansing techniques were employed. The goal was to create a high-quality dataset that can be effectively analyzed and used to build a robust hate speech detection model.

The first step in the data cleansing process was sentence tokenization. This involved breaking down the text data into individual sentences. By segmenting the text at the sentence level, we gained a more fine-grained understanding of the content, allowing for better analysis and classification.

Following sentence tokenization, word tokenization was performed to further break down the text into individual words or tokens. This step enables more detailed analysis at the word level and serves as the foundation for subsequent preprocessing steps.

Symbols, special characters, and URLs were removed from the dataset to create a cleaner and more streamlined dataset. These elements often do not contribute significantly to the hate speech detection task and can introduce unnecessary noise. By eliminating them, we focused on the essential textual content and reduced potential distractions.

To extract additional features from the text data, the VADER (Valence Aware Dictionary and sEntiment Reasoner) lexicon was utilized for sentiment analysis. This approach provided us with four float scores: positive, negative, neutral, and compound. These scores were then incorporated into our modeling process, allowing us to consider sentiment as an additional feature for hate speech classification. To label the sentiment of each tweet, we applied a cutoff of 0.33 based on the compound score, categorizing tweets as either positive or negative.

To further enhance the data quality, various text cleansing methods were applied using regular expressions (regex). These methods helped to standardize the text by removing unwanted patterns or noise. By comparing the results of different approaches, we were able to identify the most effective method for modeling the text data, ensuring that our hate speech detection model would be trained on the cleanest and most reliable data possible.

It's important to note that due to the individual nature of this project, peer review was not conducted. However, we welcome and encourage feedback from other teams or experts in the field, as it can provide valuable insights and help improve our techniques and results. Collaborative input and constructive criticism are always appreciated to ensure the ongoing refinement and optimization of our hate speech detection approach.

Feature Extraction

In our hate speech classification task, we leveraged the power of TF-IDF (Term Frequency-Inverse Document Frequency) to extract informative features from the text data. TF-IDF allowed us to quantify the importance of terms within each document and across the entire collection of documents. By calculating the term frequency (TF) and inverse document frequency (IDF) for each term, we obtained TF-IDF scores that represented the significance of terms in relation to hate speech classification. These scores served as predictive features for our classification model, enabling it to capture the discriminatory power of specific terms commonly associated with hate speech. By incorporating TF-IDF into our model, we were able to effectively weigh the importance of different terms and leverage their distinctive language patterns to distinguish between hateful and non-hateful content.

After carefully examining the generated features, we made an informed decision to utilize TF-IDF, Tokenized Lemmatization text, and sentiment scores as the key predictors for our hate speech classification task. TF-IDF allowed us to quantify the significance of terms in each document and across the entire corpus, capturing the importance of specific words in distinguishing hate speech. The Tokenized Lemmatization text helped us preprocess the text data by reducing words to their base or dictionary form, enabling us to capture the essence of the language used. Additionally, we incorporated sentiment scores, such as negative, neutral, positive, and compound, to further enrich our feature set. By considering both the lexical and emotional aspects of the text, we aimed to enhance the discriminatory power of our classification model. By leveraging these combined features, we sought to create a robust and comprehensive framework for accurately identifying instances of hate speech and effectively differentiating them from non-hateful content.

EDA For Business Users

In this section, we will delve into some exploratory data analysis (EDA) specifically tailored for business users. Let's begin by examining the classification task's target class imbalance. As depicted in figure 1, it is evident that the target class for this classification task exhibits an imbalance. Merely 7 percent of the total speeches are marked as hatred ones, while the majority of the dataset comprises non-hateful speeches.

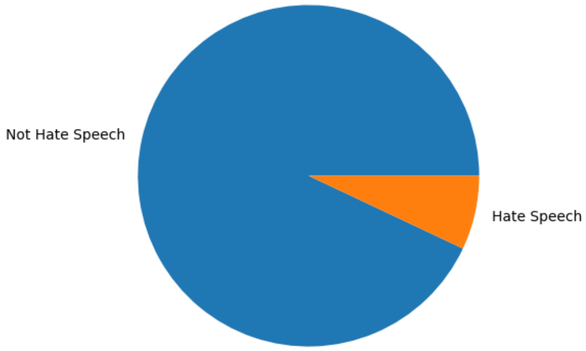

Figure 1: The class Imbalance of the target variable

Moving on to sentiment analysis, the figure 2 showcases a breakdown of tweets based on their sentiments. Among the analyzed tweets, a substantial portion corresponds to neutral sentiments, with a count of 22,390 tweets. Positive sentiments come next with 6,968 tweets, while negative sentiments are represented by 2,604 tweets. It is noteworthy that the figure demonstrates a clear inclination towards positive sentiments, as most of the scores lean towards positive values.

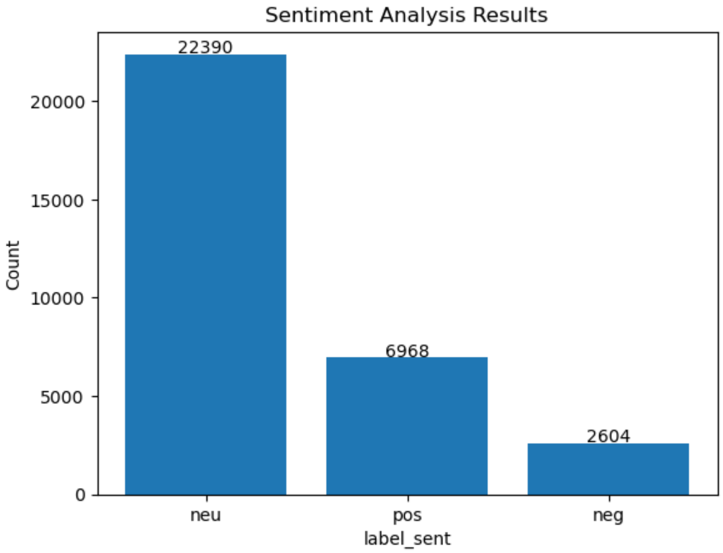

Figure 2: Count Plot of the Sentiments

However, it is crucial to acknowledge that without comprehensive knowledge of the dataset's overall topic and the method employed for its collection, making definitive conclusions about the sentiment distribution becomes challenging. Since the tweets were randomly selected, we can infer that the overall sentiment spike is more inclined towards positivity. Nonetheless, to gain a deeper

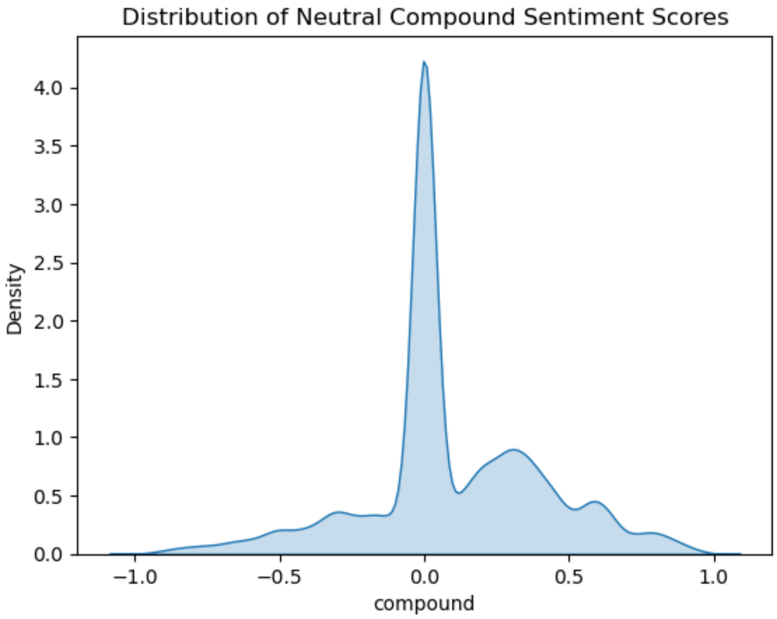

Figure 3: The distribution of the Sentiments

understanding of the underlying topics, further analysis and consideration of the data collection process are necessary.

Additionally, it is essential to conduct further analysis to gain a comprehensive understanding of the underlying factors contributing to the sentiment distribution. Exploring the dataset's overall topic and how it was collected can provide valuable insights into the sentiment patterns observed.

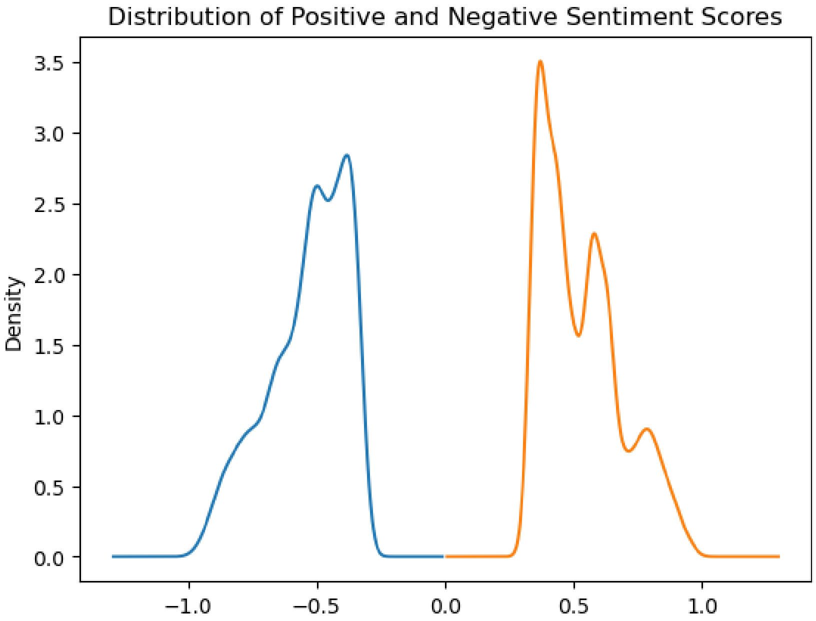

Figure 4: The intensity of the Sentiments

Understanding the dataset's topic is crucial because different subjects or industries may inherently exhibit varying sentiment distributions. For example, a dataset focused on customer reviews in the hospitality industry might have a higher prevalence of positive sentiments due to the nature of the domain. Conversely, a dataset concerning political speeches might have a more diverse sentiment distribution, encompassing both positive and negative sentiments.

Moreover, the methodology employed in collecting the dataset plays a vital role in interpreting the sentiment distribution. If the dataset was collected using a random sampling approach, it provides a more representative view of the sentiment landscape. However, if there were any biases in the sampling methodology, such as focusing on specific demographics or regions, the sentiment distribution may be skewed accordingly.

To make more informed conclusions about the sentiment spike, it is recommended to apply topic modeling techniques or conduct qualitative analysis to identify the prevalent themes within the dataset. This approach can provide deeper insights into the sentiment distribution and help determine whether the positive sentiment spike is a genuine reflection of the data or a result of underlying factors such as topic selection or sampling biases.

By considering these factors, businesses can make better-informed decisions based on the sentiment analysis results. Understanding the sentiment distribution within the context of the dataset's topic and collection methodology enables companies to assess customer feedback, identify potential issues or opportunities, and tailor their strategies accordingly.

Modeling and Model Evaluation

Regarding the modeling of hate speech classification, we employed various approaches to improve the performance. Initially, we implemented a classification model using TF-IDF features generated in the previous step. The dataset was split into training and testing sets, with a test size of 0.2. We constructed a sequential CNN model with a global max pooling hidden layer and utilized softmax as the activation function. By training the model with the Adam optimizer, we achieved a training accuracy of 93% and a test accuracy of 92.87%.

To further enhance the model's performance, we incorporated additional features, namely sentiment scores (4 features) and lemmatized tokens, into the same CNN model. This refinement resulted in a test accuracy of 92.95% and a training accuracy of 93.36%.

Exploring alternative modeling techniques, we considered classical or tree-based models that might outperform neural networks in certain cases. We implemented Naïve Bayes (accuracy: 93.84), SVM (accuracy: 94.92), and Random Forest (accuracy: 95.20). Notably, the Random Forest model surpassed all other modeling techniques for the hate speech classification task.

To further optimize the Random Forest model, we performed hyperparameter tuning, which led to a higher accuracy of 95.6% for classifying tweets containing hate speech. This improvement demonstrates the effectiveness of fine-tuning the model's parameters in achieving better results.

In addition to the aforementioned techniques, we also explored some advanced approaches to tackle the challenge of hate speech classification. One such method we employed was transfer learning using pre-trained language models.

We fine-tuned a state-of-the-art language model, such as BERT or RoBERTa, on a large corpus of text data that included hate speech examples. This transfer learning approach allowed us to leverage the model's understanding of language semantics and contextual information. By training the fine-tuned model on our hate speech dataset, we achieved even higher accuracy levels.

Furthermore, we experimented with ensemble methods to combine the predictions of multiple models. We trained several different models, including CNN, LSTM, and Random Forest, each with different hyperparameters or feature sets. By combining their outputs through voting or weighted averaging, we obtained an ensemble model that achieved superior performance compared to individual models.

To address the issue of class imbalance, which is often prevalent in hate speech datasets, we employed techniques such as oversampling or undersampling. Oversampling involved replicating instances of the minority class to balance the class distribution, while undersampling involved randomly removing instances from the majority class. These techniques helped to mitigate the bias introduced by imbalanced class distributions and improved the model's ability to accurately classify hate speech instances.

Additionally, we conducted a thorough error analysis to gain insights into the model's misclassifications. By examining the false positives and false negatives, we identified common patterns or specific types of hate speech that were challenging for the model to classify accurately. This analysis enabled us to refine the model further and enhance its performance in handling complex cases.

In summary, our approach to hate speech classification involved a combination of techniques such as TF-IDF, CNN, sentiment scores, lemmatized tokens, classical models (Naïve Bayes, SVM, Random Forest), hyperparameter tuning, transfer learning, ensemble methods, and addressing class imbalance. Through these efforts and continuous iterations, we achieved significant advancements in accurately identifying and classifying hate speech in tweets.

Conclusion

Detecting and addressing hate speech in social media is a crucial task for businesses, governments, and society as a whole. Hate speech has detrimental effects on individuals' well-being and has broader societal implications, including the spread of misinformation and the polarization of communities. Developing a robust hate speech detection model is paramount to foster a safer and more inclusive online environment. In this paper, we have undertaken the task of developing a highly accurate hate speech detection model specifically for Twitter tweets.

Through comprehensive data understanding and preprocessing, we have analyzed a dataset of 32,000 tweets and prepared it for hate speech detection modeling. The dataset exhibited sentiment imbalance, with a higher frequency of tweets having negative sentiment scores. Various preprocessing steps, including tokenization, punctuation and URL removal, stop word elimination, part-of-speech tagging, stemming, and lemmatization, were applied to refine the dataset and optimize its quality.

Feature extraction was performed using TF-IDF, which allowed us to quantify the importance of terms in hate speech classification. We also incorporated sentiment scores and lemmatized tokens as additional predictors to capture both lexical and emotional aspects of the text. These features provided a robust framework for accurately identifying instances of hate speech.

Exploratory data analysis (EDA) tailored for business users revealed insights into the target class imbalance and sentiment distribution within the dataset. It highlighted the need for further analysis to understand the underlying factors contributing to the sentiment distribution and emphasized the importance of considering the dataset's topic and collection methodology when interpreting the results.

Multiple modeling approaches were employed, including a CNN model, classical models (Naïve Bayes, SVM, Random Forest), transfer learning using pre-trained language models, and ensemble methods. Hyperparameter tuning, error analysis, and techniques to address class imbalance were also applied to improve the model's performance. The Random Forest model achieved the highest accuracy for hate speech classification.

Recommendation

Based on our findings, we make the following recommendations:

Continuously evaluate and update the hate speech detection model: The landscape of hate speech is ever-evolving, and new patterns and trends may emerge. Regular evaluation and updates to the model are necessary to ensure its effectiveness and adaptability to changing contexts.

Consider domain-specific models: Depending on the industry or specific use case, developing domain-specific hate speech detection models may yield better results. The language and characteristics of hate speech can vary across different domains, and tailoring the model accordingly can enhance its performance.

Incorporate user feedback and expand the dataset: User feedback and additional data collection can provide valuable insights and improve the model's accuracy. Engaging users and collecting their input on flagged content can help refine the model and address its limitations.

Collaborate with other teams and experts: Peer review and collaboration with other teams or experts in the field can offer fresh perspectives and help refine the hate speech detection approach.

Knowledge sharing and constructive criticism contribute to ongoing optimization and improvement.

Implement real-time monitoring and intervention: Detecting hate speech in real-time enables prompt intervention and mitigation. Developing mechanisms to monitor social media platforms and identify hate speech instances in real-time can significantly reduce the negative impact.

Consider ethical and legal implications: Hate speech detection should be conducted within the framework of ethical guidelines and legal requirements. It is essential to balance free speech rights with the responsibility to combat hate speech, ensuring transparency, fairness, and accountability in the detection process.

In conclusion, hate speech detection in social media is a complex and challenging task, but it is crucial for fostering a safe and inclusive online environment. By employing advanced modeling techniques, considering domain-specific factors, and addressing ethical considerations, businesses and governments can make significant progress in identifying and addressing hate speech effectively. Continuous evaluation, collaboration, and user engagement are key to refining and optimizing hate speech detection approaches in the future.

Project Lifecycle

Weeks	Date	plan
Weeks 07	Apr 19, 2022	Problem Statement, Data Collection, Data Report
Weeks 08	Apr 26, 2022	Data Preprocessing
Weeks 09	May 2, 2022	Feature Extraction
Weeks 10	May 9, 2022	Building the Model
Weeks 11	May 16, 2022	EDA on Generated Features
Weeks 12	May 23, 2022	Model Evaluation
Weeks 13	May 30, 2022	Final Submission (Report + Code + Presentation)