



# **Data Science Intern at Data Glacier**

**Project:** Hate Speech Detection Using Transformers

**Week 10:** Project Deliverable 4

**Team Name:** Fibo

**Team Members:** Mahyar Arani

**Email:** [arani.mahyar@gmail.com](mailto:arani.mahyar@gmail.com)

**Batch code:** LISUM19

**Submission date:** May 9, 2023

**Submitted to:** Data Glacier

## **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.

## 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	Model Evaluation
Weeks 12	May 23, 2022	Flask Development + Heroku
Weeks 13	May 30, 2022	Final Submission (Report + Code + Presentation)