Data Glacier

Your Deep Learning Partner

# Exploratory Data Analysis

## Hate Speech Detection using Transformers

**Yash Parmar: yparmar2024@gmail.com**

**June 30th, 2025**

# Agenda

Executive Summary

Problem Statement

Approach

EDA

EDA Summary

Recommendations

# Executive Summary

Social media platforms face growing pressure to control hate speech, but manual moderation is slow, costly, and difficult to scale. Existing automated tools often misinterpret the intent behind posts, especially when hate is expressed subtly or through coded language. These inaccuracies can lead to harmful content slipping through or benign posts being wrongly flagged, both of which damage user trust and the platform's reputation. Our project addresses this challenge by developing a deep learning model using transformer-based NLP to improve the accuracy and efficiency of hate speech detection. By reducing the need for manual review and minimizing false decisions, the solution aims to help platforms maintain safer, more respectful communities at scale.

# Problem Statement

Hate speech remains a persistent issue on social media, often spreading quickly and impacting users' safety and well-being. Traditional moderation methods rely heavily on human effort, which is time-consuming, inconsistent, and difficult to scale with the volume of user-generated content. To improve efficiency, platforms can leverage machine learning and NLP to detect hate speech automatically—particularly on high-traffic platforms like Twitter. By training transformer-based models on labeled tweet data, these systems can recognize subtle linguistic patterns and context that signal harmful intent. The challenge lies in balancing speed and scalability with the need for high accuracy, ensuring that real hate speech is flagged while avoiding the misclassification of harmless content.

# Approach

To tackle the problem effectively, I first focused on preparing the dataset and understanding its composition through exploratory data analysis (EDA). The following steps were taken:

1. Checked for missing or null values to ensure data completeness.
2. Built a custom cleaning function to remove unwanted symbols, links, emojis, and patterns that could interfere with model performance.
3. Standardized text by converting all tweets to lowercase for consistent tokenization.
4. Removed special characters and extra whitespace to reduce token noise and improve text clarity.
5. Stored the cleaned tweets in a separate column to preserve the original raw data.
6. Identified a class imbalance in the dataset, which informed later modeling and evaluation strategies.

# Exploratory Data Analysis

To gain deeper insights into the structure and content of the dataset, I performed several targeted text-based analyses:

- **POS Tagging:** Helped identify common parts of speech in hate vs. non-hate tweets to detect linguistic patterns.
- **Average Word Count:** Measured tweet lengths before and after cleaning to assess text noise and redundancy.
- **N-gram Analysis:** Revealed frequent word combinations that may signal hate speech across different tweet types.
- **Word Clouds:** Visualized dominant words in various tweet categories to highlight thematic differences.

# Exploratory Data Analysis Summary

To summarize key findings from the exploratory data analysis, I've highlighted what each method revealed about the dataset's structure, tone, and challenges:

- **POS Tagging:** Cleaning enhanced the presence of key parts of speech like adjectives and nouns, which are essential for detecting sentiment and hate speech.
- **Average Word Count:** Cleaning reduced noise (e.g., mentions, links) without removing meaningful content, resulting in more focused text.
- **N-gram Analysis:** After cleaning, hate tweets showed more aggressive or insulting phrases, while non-hate tweets reflected personal or emotional language.
- **Word Clouds:** Visualizations revealed class imbalance and showed that sentiment-heavy words like "love" appear in both hate and non-hate contexts, sometimes sarcastically.

# Recommendations (Technical Summary)

**Classical Machine Learning Models** *(Logistic Regression, Random Forest, XGBoost)***:**

Strengths:

- Fast training and low computational cost, making them easy to experiment with.
- Interpretable decision boundaries, especially for Logistic Regression.
- XGBoost and Random Forest handle tabular formats and class imbalance relatively well.

Limitations:

- Do not fully leverage the contextual richness of BERT embeddings.
- Struggle with non-linear and subtle semantic patterns like sarcasm or coded hate.
- Can underperform or overfit when applied to short, noisy text with limited context.

**Deep Learning Model** *(Multilayer Perceptron – MLP)*:

Strengths:

- Captures complex, non-linear relationships embedded in BERT vectors.
- Better generalization on short, informal, and ambiguous tweets.
- Adapts to subtle semantic cues revealed during EDA (e.g., emotional tone shifts).

Limitations:

- Requires more compute and longer training times.
- Less interpretable compared to traditional ML models.
- Sensitive to hyperparameters and may overfit with limited data.

# Recommendations (Business Summary)

Tweets containing hate speech are often short, indirect, and written in informal or coded language, which makes them difficult to classify accurately. Classical machine learning models like Logistic Regression or XGBoost, while fast and interpretable, struggle to capture the complex patterns embedded in BERT's contextual embeddings. These models tend to oversimplify nuanced language, leading to lower accuracy and more false negatives.

To address this, we recommend using a **Multilayer Perceptron (MLP)**, a deep learning model that is better equipped to recognize subtle differences in tone, structure, and semantic meaning. MLPs can learn non-linear relationships in the data, which is especially important when identifying hate speech that is sarcastic, emotionally charged, or indirectly phrased.

By leveraging the strengths of BERT and MLP together, our approach improves detection performance in real-world settings—reducing harmful content more effectively while minimizing false positives that could frustrate users.

# Thank You