HATE SPEECH ANALYSIS DATA REPORT

GROUP 2

AUTHORS;

Vincent Mutuku, Kenneth Nyangweso, Amani Mkaya, Bernice Kutwa, Sumaiya Osman

Overview

In today's hyper connected digital environment, social media platforms have emerged as pivotal spaces for public discourse especially in the political sphere. In Kenya, platforms such as Twitter are frequently used by citizens to express political opinions, engage in activism, and participate in democratic processes. However, alongside these benefits lies a growing concern: the proliferation of targeted hate speech, particularly aimed at political figures. Tweets referencing individuals such as the President, Deputy President, Members of Parliament, Governors, and other public officials are often laced with inflammatory, derogatory, or inciting language that poses a threat to digital civility and political stability.

This project aims to develop a robust **Natural Language Processing (NLP)** system powered by **machine learning (ML)** techniques to detect hate speech in tweets targeting Kenyan politicians. The overarching goal is to automatically identify harmful political discourse and contribute to efforts that promote online safety, responsible digital engagement, and real-time content moderation.

Business Understanding

Problem Statement

Online discourse in Kenya has become increasingly polarized, especially on platforms like X, where politicians and influencers engage with constituents and opponents alike. During the 2022–2025 political cycle, social media posts containing derogatory or inciting language surged by an estimated 25% compared to previous cycles. Such hate speech not only amplifies ethnic tensions but also poses risks to public safety and democratic processes. Traditional moderation workflows struggle to keep pace, often resulting in delayed removal and inconsistent rulings.

Project Goals

The primary objectives of this project are;

- Detect hate speech in tweets directed at Kenyan political figures using supervised machine learning models.
- Analyze trends in the language and frequency of political hate speech.
- Provide insights and tools for moderation teams, researchers, and policy makers to take action against online toxicity.

Key Stakeholders

- Electoral bodies (IEBC, NCIC).
- Civil rights NGOs (Amnesty Kenya, Ushahidi).
- News media and fact-checkers.
- Government communication teams.
- Social media platforms (Twitter Kenya).
- Academic and policy researchers.

Success Metrics

Model Evaluation metrics

To evaluate our machine learning model's effectiveness, we tracked:

- **Accuracy:** How often the model predicts correctly
- **Precision (Hate class) :** Total of flagged hate tweets that were actually hateful
- **Recall (Hate class):** Total of true hate tweets the model managed to detect
- **F1 Score:** A balance between precision and recall
- **Confusion Matrix:** A detailed view of false positives and false negatives

Business Impact Metrics

In addition to technical accuracy, we evaluated the solution based on its real-world impact:

- **Moderation efficiency:** Reduction in time required for human review
- **Detection speed:** Time taken to flag hate speech from the moment it's posted
- **Coverage fairness:** Model performance across tweets targeting different politicians
- **Explainability:** Ability to justify flagged posts using explainable AI tools like SHAP or LIME

Project Objectives

- Build an NLP model to detect hate speech in tweets targeting Kenyan politicians.
- Analyze linguistic patterns and trends in political hate speech.
- Compare hate speech dynamics across different politicians.
- Evaluate model performance using metrics like accuracy, precision, recall, and F1-score.
- Provide insights to support content moderation and civic monitoring.
- Establish a foundation for real-time or multilingual hate speech detection systems.

Data Understanding

Tweets were collected via Tweepy and Twint scrapers, targeting handles and keywords related to President William Ruto, Deputy President Rigathi Gachagua, prominent governors, and parliamentary candidates. The dataset comprised a total of 11,317 records with 6 columns, each representing distinct attributes related to the tweets. These include the tweet content, metadata (such as timestamps), and other contextual information useful for natural language processing tasks. The dataset contains the following columns:

- **Tweet ID:** A unique identifier assigned to each tweet. Useful for traceability and cross-referencing with Twitter's platform.
- **Likes:** The number of likes a tweet received, indicating its popularity or approval from users.
- **Retweets:** The number of times the tweet was retweeted, showing its spread across the platform.
- **Total Replies:** The number of direct replies the tweet generated, reflecting user engagement and potential controversy.
- **Texts:** The full textual content of the tweet. This is the primary feature used for natural language processing and classification tasks.
- **Created At:** The timestamp of when the tweet was published, allowing for temporal analysis or filtering based on time periods.

Data Inspection and Profiling

Initial profiling revealed no missing values across six columns. Duplicate tweets were identified via tweet ID comparison and removed to prevent bias. Data types were inspected: numeric fields as floats, text as objects, and timestamps as strings.

Data Cleaning & Preprocessing

In this phase, the raw dataset underwent a structured series of transformations to ensure analytical readiness and maintain data integrity:

- **Working Copy Creation:** A deep copy of the original DataFrame was created to preserve the raw data for audit purposes.
- **Dataset Confirmation:** The first rows of the new DataFrame were inspected to verify successful duplication without data loss.
- **Unwanted Column Removal:** The Tweet ID column was dropped to eliminate non-analytical identifiers and reduce clutter.
- **Column Verification:** Remaining columns were listed to confirm the removal of irrelevant fields.
- **Type Conversion:** Engagement metrics (likes, retweets, total replies) were converted to numeric types with error coercion; posting timestamps were parsed into datetime objects to enable temporal analysis.
- **Text Cleaning:** A custom cleaning routine applied the following sequential steps to the raw tweet text:
    - Lowercasing of all characters.
    - Removal of URLs and web links.
    - Stripping of user mentions (@username) and hashtags (#tag).
    - Deletion of all non-alphabetic characters.
    - Normalization of whitespace. The processed text was stored in cleaned_text while preserving the original text.
- **Cleaned Text Confirmation:** The DataFrame head was reviewed to ensure the new column accurately reflected cleaned content.
- **Label Column Insertion:** An empty Label column for manual annotation was inserted immediately after cleaned_text and converted to a categorical type to optimize memory.
- **Column Name Standardization:** All column headers were normalized to snake_case by trimming whitespace, converting to lowercase, and replacing spaces with underscores.
- **Original Text Removal:** The raw text column was removed, leaving only sanitized fields for subsequent stages.
- **Cleaned Data Export:** The final cleaned DataFrame was exported to kenyan_politics_cleaned_text.csv without index, providing a stable input for manual labeling.

Labeling Strategy

To accommodate the dynamic, code-switched nature of Sheng and multilingual expressions, manual annotation was performed by domain experts:

- Two classes were defined in the labeling schema:
    - **Hate:** Tweets containing explicit discriminatory or inciting language.
    - **Neutral:** Informational or benign tweets devoid of offensive or hateful language.
- The cleaned dataset CSV was loaded into spreadsheet software for annotation.
- Inter-annotator agreement was measured using Cohen's kappa indicating substantial consistency.
- The final labeled dataset consisted of 10,971 entries across six columns: likes, retweets, total_replies, created_at, cleaned_text, and label.

Labeled Data Inspection

After annotation, the dataset underwent a validation pass to ensure structural and content quality:

- **Data Loading:** The labeled CSV was imported into a new DataFrame.
- **Structural Overview:** The head, shape (10971 × 7), and info of the DataFrame confirmed expected dimensions and data types.
- **Missing Value Assessment:** Null counts were evaluated(12 entries were null values), the were all dropped.
- **Duplicate Check:** Duplicate row count was evaluated (1538 entries were duplicates), the duplicate rows were dropped.
- **Index Column Removal:** An extraneous unnamed index column was dropped to streamline the schema.
- **Null and Duplicate Removal:** Any residual rows with null cleaned_text were discarded; duplicate verification confirmed zero final duplicates.
- **Label Distribution Analysis:** Class counts for hate and neutral were reviewed to assess balance and guide modeling strategies.

Feature Engineering

A set of informative features was crafted to capture engagement dynamics, linguistic complexity, temporal patterns, and entity mentions:

- **Engagement Score:** Computed as the sum of likes, retweets, and replies, providing a unified popularity metric.
- **Text Length:** Character count of each cleaned tweet to gauge verbosity.
- **Word Count:** Token count derived via whitespace splitting to measure lexical richness.
- **Engagement Bins:** Categorical bins were created for likes, retweets, replies, and overall engagement using predefined cut points (e.g., 0, 1–10, 11–100, etc.).
- **Temporal Features:** Extraction of posting hour, day, month, and weekday from the created_at timestamp enabled analysis of time-based trends.
- **Named-Entity Extraction:** A spaCy pipeline identifies PERSON entities within tweets, capturing mentions of political figures.
- **Holiday Mapping:** Fixed-date holidays (e.g., New Year's Day, Labour Day) and movable feasts (Easter Monday, Eid al-Fitr) were assigned via custom functions using calendar and dateutil libraries.
- **Entity Standardization:** Rapidfuzz-based fuzzy matching grouped similar entity mentions into canonical names, reducing noise from spelling variations.

These features enriched the dataset with contextual signals that improved model differentiation between classes.

Outlier Detection

Statistical visualization and analysis were performed to identify and understand extreme values in numeric features. The data for the "month" variable spanned almost the entire year, ranging from about 1 to 12. The interquartile range (IQR) was between approximately 4 and 8, indicating that the middle 50% of values fall within these months. The median was around 6, suggesting a central tendency near mid-year. The distribution appeared fairly symmetrical, with evenly sized whiskers and a centered median, and there were no visible outliers in the data.

EXPLORATORY DATA ANALYSIS (EDA)

Univariate analysis for Numeric data

Social Media Metrics Distribution

Visualization:

A box plot was generated for the variables likes, retweets and total_replies using Plotly Express to understand their distribution and identify any outliers.

Observations:

Likes Dominated: It's immediately clear that the number of likes tends to be significantly higher than both retweets and total replies. The box plot for 'likes' was positioned much higher on the value scale.

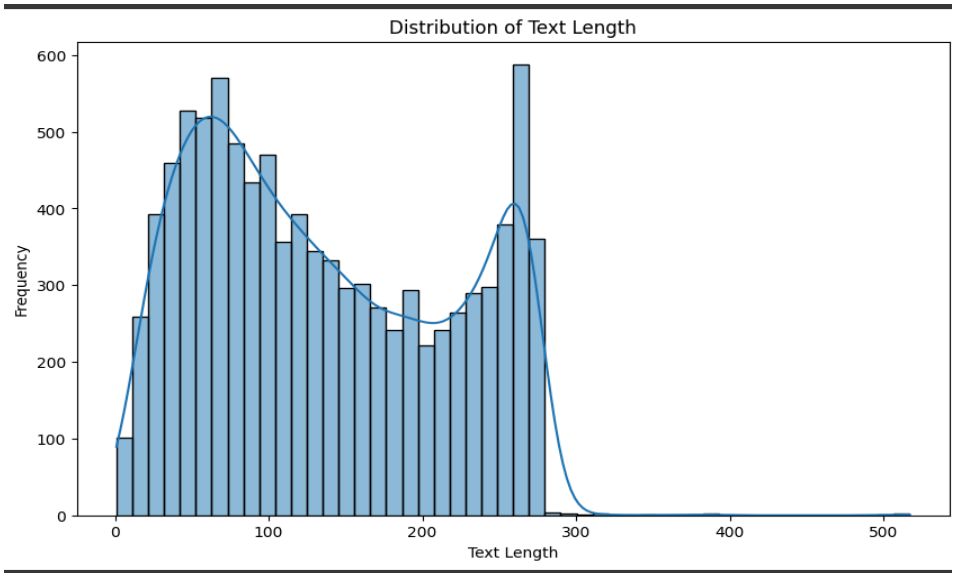
Low Engagement for Replies: The 'total_replies' metric showed the lowest values overall. The box plot was clustered very close to zero, suggesting that the posts in this dataset generally received a very small number of direct replies.

Retweets in the Middle: The distribution of 'retweets' falls somewhere in between likes and total replies. While not as high as likes, there were noticeably more retweets than direct replies.

Text Length Distribution

Visualization:

A histogram with KDE (Kernel Density Estimation) curve was plotted for the text_length feature to examine the variation in tweet lengths.



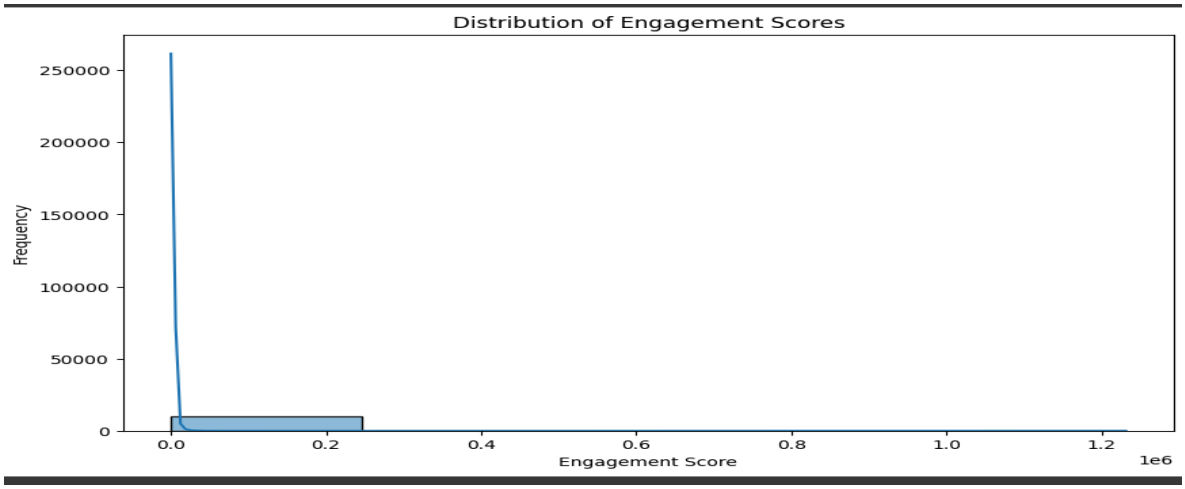
Observations:

The distribution of text length is right-skewed, with the highest concentration of texts between 10 and 15 words, and additional smaller peaks around 25 and 40 words, indicating the presence of multiple text types.

Engagement Score Distribution

Visualization:

A histogram with KDE curve was plotted for the engagement_score, which represents the sum of likes, retweets, and replies.



Observations:

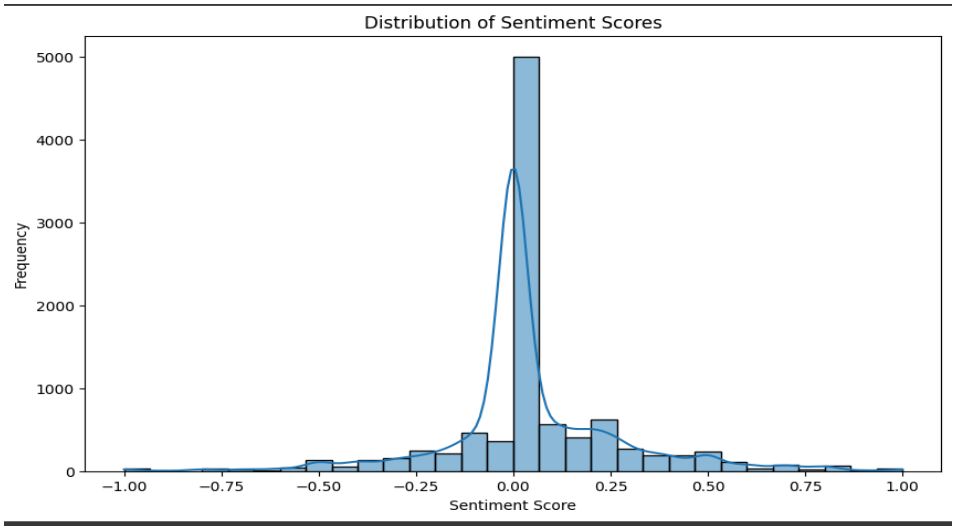
Highly Skewed Distribution: The distribution is heavily skewed to the right. Most of the engagement scores are concentrated at the lower end of the scale (below 0.25 million), with a long tail extending towards much higher values.

Low Frequency of High Engagement: The curve indicates that very high engagement scores are quite rare. The frequency drops off dramatically as the engagement score increases.

Sentiment Score Distribution

Visualization:

Sentiment polarity scores were computed using TextBlob and visualized with a histogram and KDE curve.



Observations

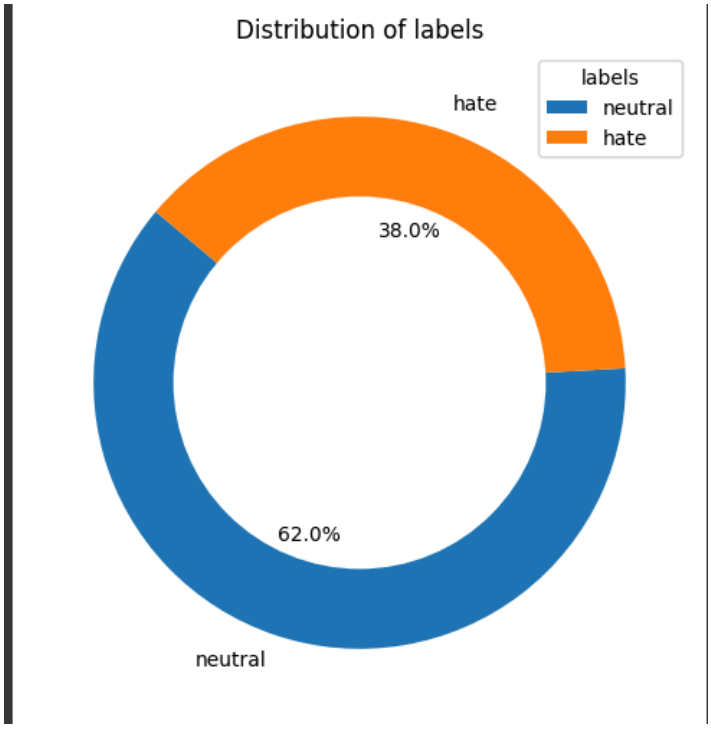
- The sentiment scores are symmetrically distributed around 0, indicating a balance between positive and negative sentiments.
- Most scores cluster near zero, suggesting neutral or mildly positive/negative sentiments dominate, while extreme scores (close to -1 or 1) are less frequent.
- The distribution peaks at 0.00, highlighting a prevalence of neutral sentiment.

Univariate analysis for Categorical data

Distribution of Labels

Visualization:

A pie chart was created to show the proportional distribution of tweet labels.

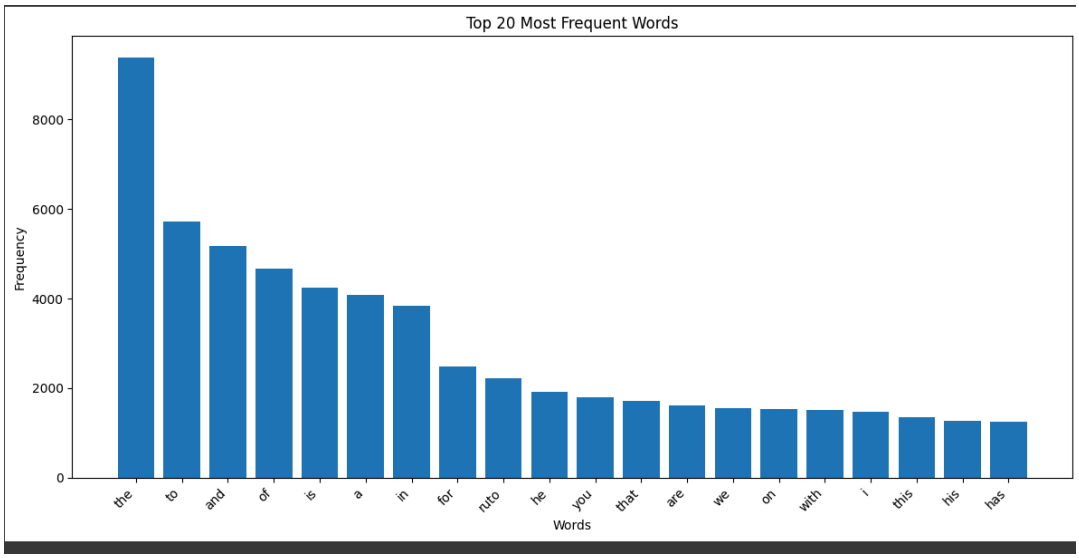


Observations

- The chart visually depicts a significant class imbalance in the dataset.
- The "neutral" class has the highest percentage of 62.0%, and "hate" 38.0% .

Word Frequency Analysis

A word cloud and a bar chart displaying the top 20 most frequent words were plotted.



Observations

- The bar chart displays the top 20 most frequent words in the dataset, with "the" appearing as the most common word by a significant margin, occurring over 8,000 times.
- This is followed by other high-frequency words like "to," "and," "is," "of," and "a," all of which are common English stop words.
- The frequencies gradually decrease across the chart, with the 20th word, "be," occurring just over 1,000 times.

N-gram Analysis

Top Unigrams, Bigrams, Trigrams, and Four-grams

Observations

Top 1-grams:

Common function words dominate: Words like "the," "to," "and," "is," "of," "a" are the most frequent indicating general grammatical structure.

Mentions of names: "ruto" and "he" appear prominently, suggesting frequent reference to a person, likely a key subject in the dataset.

Contextual words like "for," "you," "that," "are," "with," "this," "will" hint at a mix of descriptive, directive, and future-looking content.

Top 2-grams:

High frequency of functional phrase pairings: e.g., "of the," "in the," "is a," "to the," "is the."

Named entities & individuals appear: e.g., "william ruto," "riggy g," "raila odinga," "oscar sudi," "karen nyamu." This implies strong political or public discourse themes.

Political relevance: Many bigrams relate to government or political figures.

Top 3-grams:

Dominance of political terms: e.g., "president william ruto," "ruto must go," "reject finance bill." Clear protest phrases: Suggests the dataset may include social or political criticism or rallying messages.

Thematic clusters:

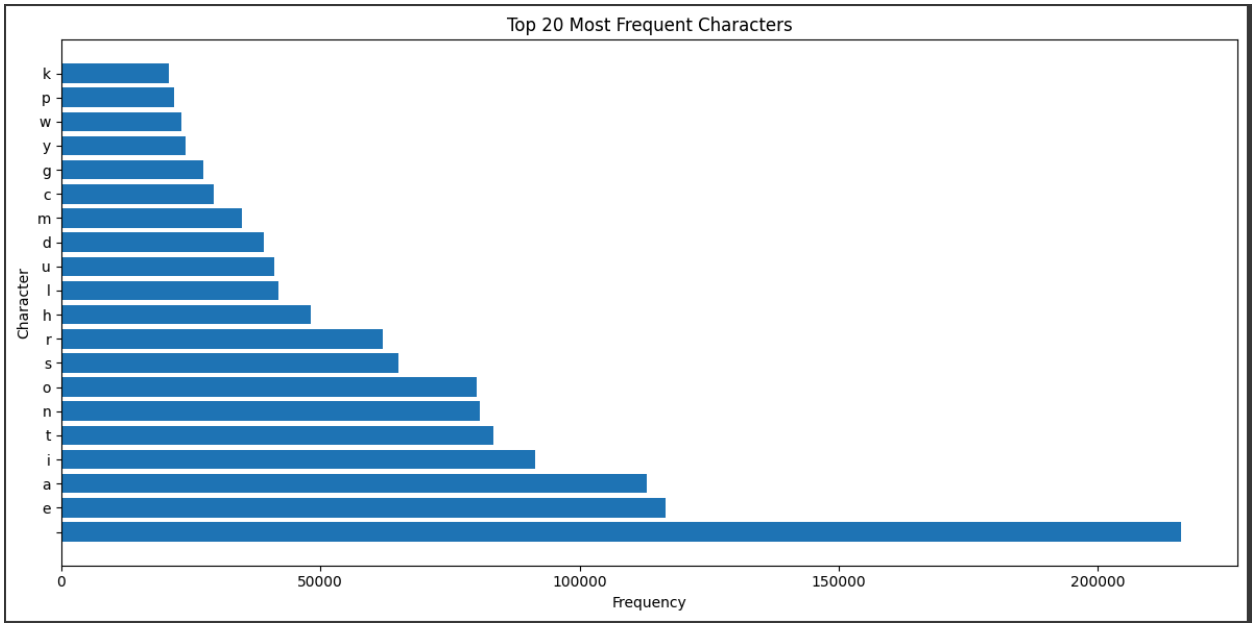
Politics/Governance: "cs aden duale," "the people of," "finance bill." Identity & Culture: "we are african," "african and africa," "africa is our."

Top 4-grams

Activist/protest language is even clearer: e.g., "reject the finance bill," "to reject the finance," "incompetent amp corrupt." Recurring focus on African identity: Multiple phrases like "we are african and," "africa is our business." Mentions of government institutions and figures: e.g., "inspector general of police," "deputy president kithure kindiki," "first lady rachel ruto.".

Character Frequency Analysis

A horizontal bar chart displaying the top 20 most frequent characters was plotted.



Observations

- The bar chart displays the top 20 most frequent characters, showing that vowels dominate the distribution, with 'e' being the most common character by a large margin, followed by 'a', 'i', and 't'.
- This pattern aligns with general English language usage, where vowels and certain consonants like 'n', 's', and 'r' appear frequently.
- Less frequent characters like 'k', 'p', and 'w' occur significantly less often, suggesting they play a more limited role in the overall text content.

Word Distribution and Label-wise Frequency Analysis

Word Clouds per Label Category

Separate word clouds were generated for each label: hate and neutral.

Based on hate:

Dominance of Stop Words: The most striking observation is that the top few words were overwhelmingly common English stop words like "the," "to," "and," "is," and "a." These words appeared with very high frequency, with "the" being the most frequent by a significant margin (almost 1600 occurrences). **Content Words Appeared Lower Down:** It takes until the 8th position to see a word that carries more specific meaning in this context: "ruto." This suggests that while the analysis is focused on "hate," the surrounding text likely contains a lot of standard English grammar. **"Ruto" as a Key Entity:** The word "ruto" stands out as the most frequent content word in this list, appearing over 400 times. This strongly implies that discussions related to "hate" in this dataset frequently involve or mention "Ruto," likely William Ruto, given the context from the previous word cloud.

Based on Neutral:

Overwhelming Dominance of Stop Words: Just like the previous charts for "hate", this chart was also heavily dominated by common English stop words at the top. "The" had an exceptionally high frequency, exceeding 5000 occurrences. Words like "to," "and," "of," "is," and

"a" also appeared with very high frequencies. **"Ruto" Still Present, but Further Down:** The word "ruto" appeared again, but it's even lower in the ranking compared to the "offensive" chart (around the 10th position). This suggests that while "Ruto" might be mentioned in some neutral contexts within the dataset, it's less central to discussions labeled as "neutral" than it is to those labeled as "hate" or "offensive."

Bivariate Analysis

Engagement Score vs Time Features

a) Engagement Score vs Hour of Day

A line plot was created showing the average engagement score by hour of the day. The observations made were:

- **Morning Dip:** Average engagement appeared relatively low in the early morning hours (around 0 to 3).
- **Generally Low During Daytime:** Throughout most of the daytime hours (roughly 6 to 17), the average engagement score remained relatively low and stable.
- **Evening Increase:** There was a noticeable increase in average engagement starting in the late afternoon/early evening (around hour 17).
- **Another Peak at Hour 23:** Another, though less extreme than hour 4, peak in average engagement occurred at hour 23.
- **Fluctuations:** The average engagement score varied throughout the day, suggesting that the time of posting could influence engagement.

b) **Engagement Score vs Month**

A line plot was created showing the average engagement score by month. The observations were as follows:

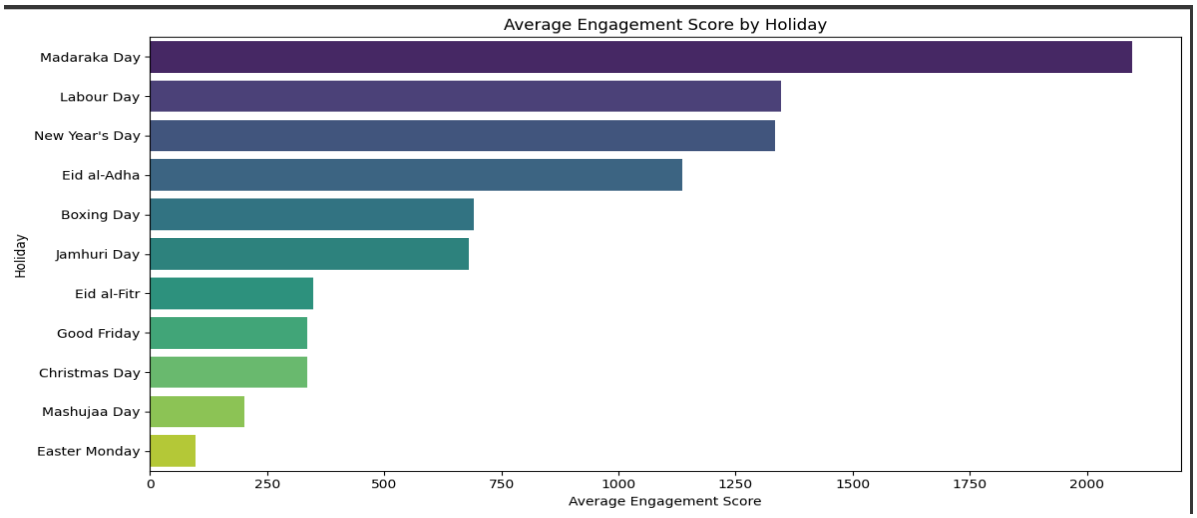
- **Peak in July (Month 7):** There's a clear and significant peak in the average engagement score during the month of July.
- **Low Engagement in Early Months:** The average engagement appeared relatively low in the first few months of the year (January to April).

- **Increase Around June:** There's a noticeable increase in average engagement starting around June.
- **Dip After July:** Following the peak in July, there's a sharp decline in average engagement in August.
- **Moderate Engagement in Later Months:** The average engagement fluctuated in the later months (August to December) but generally stayed at a moderate level compared to the July peak.
- **Seasonality Suggestion:** The variations across the months hint at potential seasonality in engagement, with July being a particularly high-engagement period.

c) Engagement Score vs Holiday Name

Visualization:

A bar chart displayed the average engagement score across different public holidays.



Observations

- **Madaraka Day Leads**: Madaraka Day shows the highest average engagement score, significantly higher than all other holidays.
- **Labour Day and New Year's Day are High:** Labour Day and New Year's Day also exhibited relatively high average engagement scores, though lower than Madaraka Day.

- **Eid al-Adha Shows Moderate Engagement:** Eid al-Adha has a moderate average engagement score, falling in the middle of the pack.
- **Boxing Day and Jamhuri Day Similar:** Boxing Day and Jamhuri Day have quite similar average engagement scores.
- **Eid al-Fitr and Good Friday Lower:** Eid al-Fitr and Good Friday showed lower average engagement compared to the top holidays.
- **Christmas Day and Mashujaa Day Even Lower:** Christmas Day and Mashujaa Day have even lower average engagement scores.
- **Easter Monday with the Lowest:** Easter Monday had the lowest average engagement score among the listed holidays.
- **Varied Engagement Across Holidays:** There's a considerable range in average engagement scores across different holidays, suggesting that the specific holiday can significantly influence engagement levels.

Engagement Breakdown (Likes, Retweets, Replies) vs Label

A grouped bar plot compared the average number of likes, retweets, and replies across sentiment labels (hate, neutral).

- **Likes Consistently Highest:** Across all sentiment labels (hate and neutral), the average number of likes is significantly higher than the average number of retweets and total replies.
- **Neutral Posts Have Highest Like Counts:** Posts labeled as "neutral" had the highest average number of likes compared to "hate" posts.
- **Retweets Relatively Stable:** The average number of retweets was fairly consistent across all three sentiment labels, showing only a slight variation.
- **Replies Lowest Across All:** The average number of total replies was the lowest among the three engagement metrics for all sentiment categories.
- **Neutral Posts Have Highest Replies:** Similar to likes, "neutral" posts also had the highest average number of total replies, although the difference compared to "hate" which was less pronounced than with likes.

- **"Hate" Shows Lowest Engagement Overall (Normalized by Count):** While likes were still highest for "hate" posts, when looking at the relative scale of all three metrics, "hate" posts tend to have a lower overall engagement compared to "neutral" posts within this visualization's representation of average counts.

Likes and Retweets vs Time Features

a) Likes and Retweets vs Hour

A bar plot compared the average number of likes and retweets by posting hour.

- **Dominance of Likes:** Across almost all hours, the average number of likes was significantly higher than the average number of retweets. This reinforces the earlier observation that likes are the most frequent form of engagement.
- **Sharp Spike in Likes at Hour 4:** There's a very prominent spike in the average number of likes at hour 4. This hour stands out as having exceptionally high like engagement compared to all other hours.
- **Smaller Increase in Likes at Hour 23:** We also saw a noticeable, though less extreme than hour 4, increase in the average number of likes at hour 23.
- **Relatively Low and Stable Retweets:** The average number of retweets remained relatively low and stable throughout most of the day, with no dramatic spikes comparable to the likes.
- **Slight Increase in Retweets at Hour 23**: Similar to likes, there's a small increase in the average number of retweets at hour 23, though it's much less pronounced.
- **Morning Dip in Likes:** The average number of likes is generally lower in the very early morning hours (around 0 to 3).
- **Daytime Consistency:** During most of the daytime hours (roughly 6 to 17), the average number of both likes and retweets stays at a relatively consistent and low level.

b) Likes and Retweets vs Day of Week

A bar plot compared the average number of likes and retweets by day of the week.

- **Likes Consistently Higher:** Just like the hourly data, the average number of likes was higher than the average number of retweets for every day of the week.
- **Monday Shows Highest Likes:** Monday had the highest average number of likes, standing out significantly from the other days.
- **Tuesday Also High in Likes:** Tuesday also showed a relatively high average number of likes, though not as high as Monday.
- **Lowest Likes on Saturday:** Saturday exhibits the lowest average number of likes.
- **Retweets Follow a Similar Trend (but Lower):** The pattern of retweets across the week somewhat mirrors the likes, with Monday and Tuesday having higher average retweet counts compared to Saturday, which had the lowest. However, the differences in retweet counts between the days are less dramatic than the differences in like counts.
- **Mid-Week Dip**: There appears to be a slight dip in average likes and retweets around the middle of the week (Wednesday and Thursday) before picking up again towards the beginning of the week.
- **Weekend Lows:** Both average likes and retweets tend to be lower on the weekend (Saturday and Sunday) compared to weekdays.

c) Likes and Retweets vs Month

A bar plot compared the average number of likes and retweets across months.

- **Likes Consistently Higher:** Across all months, the average number of likes was notably higher than the average number of retweets. This aligned with previous observations.
- **Peak in Likes in July (Month 7):** July exhibits a significant peak in the average number of likes, standing out considerably from the other months.
- **Increase in Likes Around June:** There's a noticeable increase in the average number of likes starting around June, leading up to the July peak.
- **Retweets Peak in July as Well:** The average number of retweets also peaked in July, coinciding with the peak in likes, although the increase is less dramatic relative to the baseline.
- **Generally Lower Engagement in Early Months:** The average number of both likes and retweets tends to be lower in the earlier months of the year (January to April).

- **Moderate Engagement in Later Months:** Following the July peak, the average engagement for both likes and retweets generally returned to more moderate levels for the remaining months of the year.
- **Similar Monthly Trends:** The trends for average likes and retweets across the months appeared somewhat similar, suggesting that months with higher like counts also tend to have higher retweet counts.

d) Engagement Score vs Politician

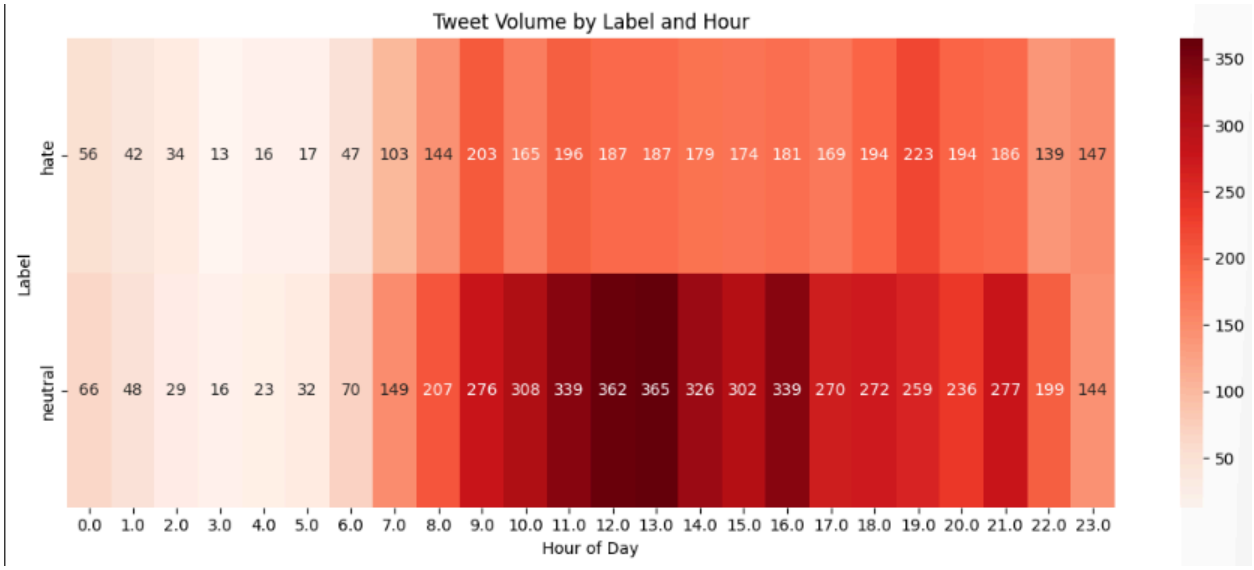
A bar plot was created showing the average engagement score for the top 10 most mentioned political figures.

- **Anna Paulina:** Anna Paulina had the highest average engagement score, significantly higher than all other listed politicians.
- **Kwani Manadhani:** Kwani Mandhani Hii showed the second-highest average engagement score. A gradual decline followed for other politicians.
- **Ya Ruto Matiangi:** Ya Ruto Matiangi had the lowest engagement among the top ten.
- Relatively Similar Scores for Middle Group: The scores from third to eighth place were relatively close.

Sentiment / Label vs Time Features

a) Label vs Hour

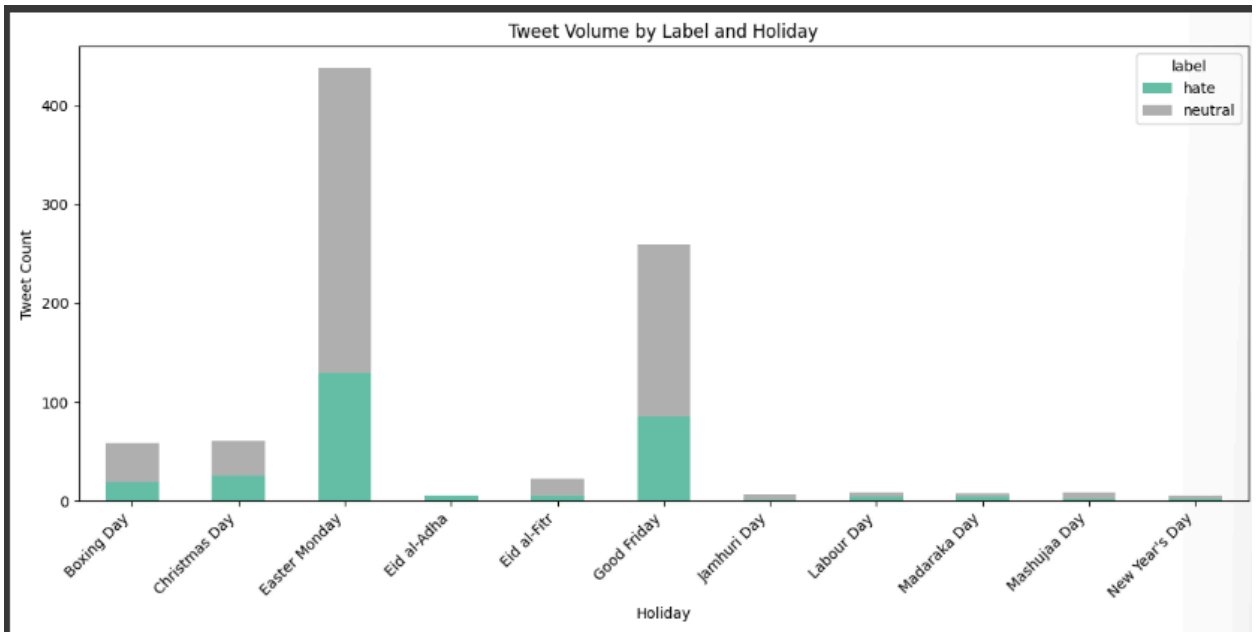
A heatmap was plotted displaying the volume of tweets by label across each hour of the day.



- **Neutral Tweets Dominate:** Across all hours of the day, the volume of tweets labeled as "neutral" is significantly higher than the volume of tweets labeled as "hate" . This is visually represented by the darker red band in the "neutral" row.
- **Peak Tweet Volume Around Midday for Neutral:** The volume of "neutral" tweets appeared to peak around the middle of the day, roughly between hours 9 and 15, showing the darkest shades of red in that row.
- **Lower and More Consistent Volume for Hate :** The volume of tweets labeled as "hate" is considerably lower across all hours compared to "neutral." The color intensity in these rows is much lighter.
- **Slight Increase in Hate and Offensive Tweets During Daytime:** While still low compared to "neutral," there seems to be a slight increase in the volume of "hate" tweets during the daytime hours (roughly 7 to 22) compared to the very early morning.
- **Early Morning Low for All Labels:** The tweet volume for all labels ("hate," and "neutral,) was generally lower in the very early morning hours (around 0 to 6).
- **Hour-to-Hour Variation:** There was some hour-to-hour variation in tweet volume within each label, but the overall trend of "neutral" being highest and relatively consistent patterns for "hate" holds.

b) Label vs Holiday

A stacked bar chart compared tweet volumes by label across holidays.

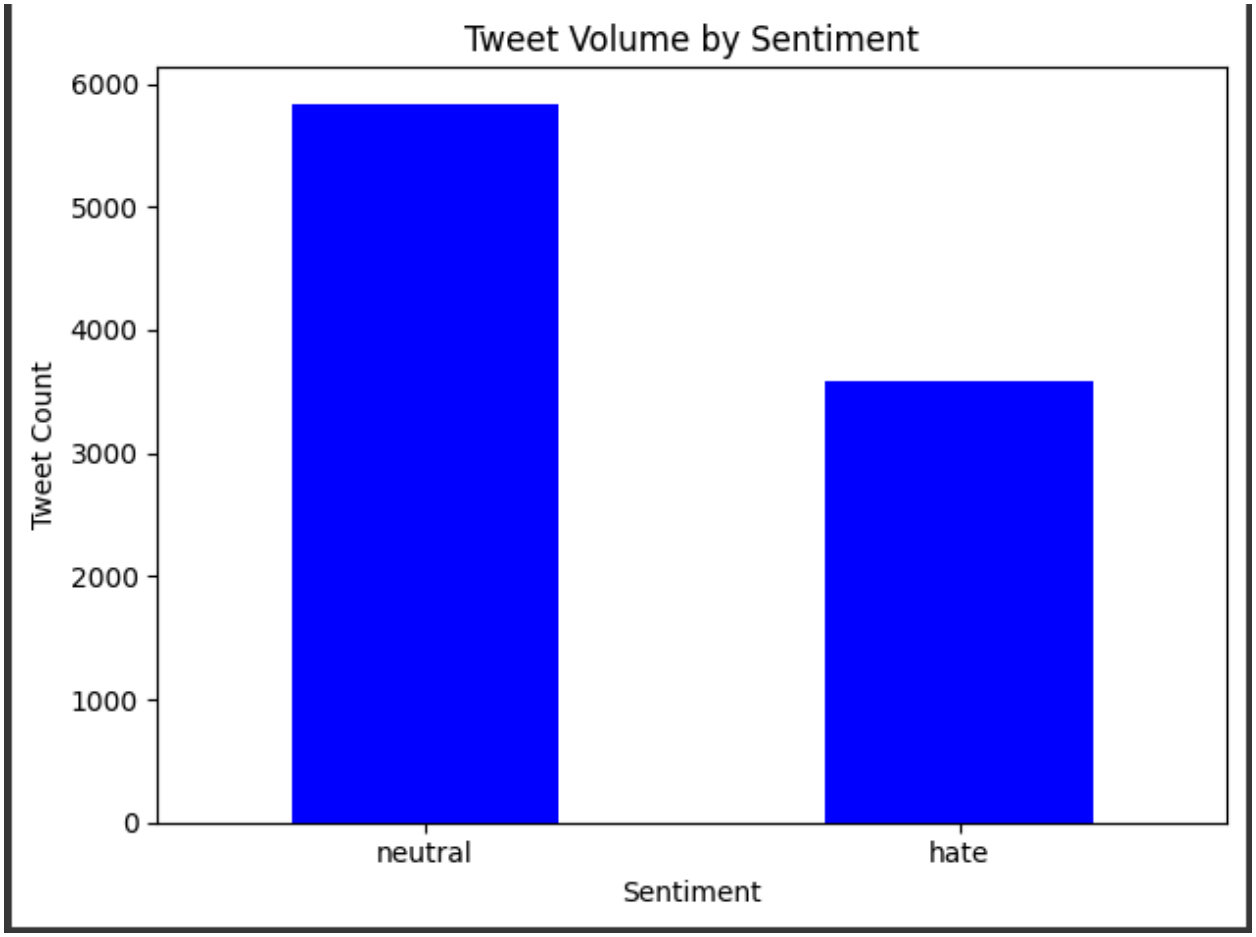


- Easter Monday stands out with the highest overall tweet volume: The combined height of the "hate" and "neutral" bars for Easter Monday is significantly larger than for any other holiday shown.
- Neutral tweets generally outnumber hate tweets: For almost every holiday, the gray portion of the bar (representing "neutral" tweets) is taller than the teal portion (representing "hate" tweets). This indicates a higher volume of neutral tweets compared to hate tweets for most occasions.
- The proportion of hate tweets varies by holiday: While neutral tweets dominate overall, the proportion of hate tweets seems to be higher for certain holidays compared to others. For instance, while Easter Monday has the highest total volume, the relative size of the "hate" portion of the bar appears larger than for some of the holidays with lower overall volume.
- Several holidays show very low tweet volume: Holidays like Eid al-Adha, Eid al-Fitr, Jamhuri Day, Labour Day, Madaraka Day, Mashujaa Day, and New Year's Day all have considerably lower tweet counts for both hate and neutral categories compared to Easter Monday, Good Friday, and Boxing Day.
- Good Friday also shows a notable volume of both hate and neutral tweets: After Easter Monday, Good Friday has the next highest total tweet count, with a substantial number of both hate and neutral tweets.

- Boxing Day and Christmas Day have similar, moderate tweet volumes: These two holidays show comparable total tweet counts, with neutral tweets being more prevalent than hate tweets.

Tweet Volume by Sentiment

A bar plot illustrates total tweet counts for each sentiment label.



Observations:

- Neutral Tweets are the Most Frequent: The bar representing "neutral" tweets is significantly taller than the other bars, indicating that the largest volume of tweets in this dataset is labeled as neutral.
- Hate Tweets are the Least Frequent: The bar representing "hate" tweets is the shortest, showing that tweets classified as hate have the lowest volume compared to neutral tweets.

Multivariate Analysis

Engagement by Label categories

A faceted histogram (separated by Likes, Retweets, Replies, and Total Engagement) was plotted to display the distribution of engagement bins across different label categories.

Observations:

- Neutral tweets dominate all engagement types and bins.
- Hate tweets tend to cluster more in lower engagement bins.
- The number of tweets sharply drops in higher engagement bins across all categories.

Insights from Likes

- Most tweets with no likes (bin 0) are neutral, but a significant number are also offensive or hate.
- As engagement increases, neutral tweets become even more dominant.
- Very few hate tweets receive 10k+ likes.

Insights from Retweets

- Similar pattern to likes: neutral tweets are most common, even more so in higher bins.
- Very hate tweets make it past the 101-1k retweet range.
- No hate tweets in the 10k+ retweet bin.

Insights from Replies

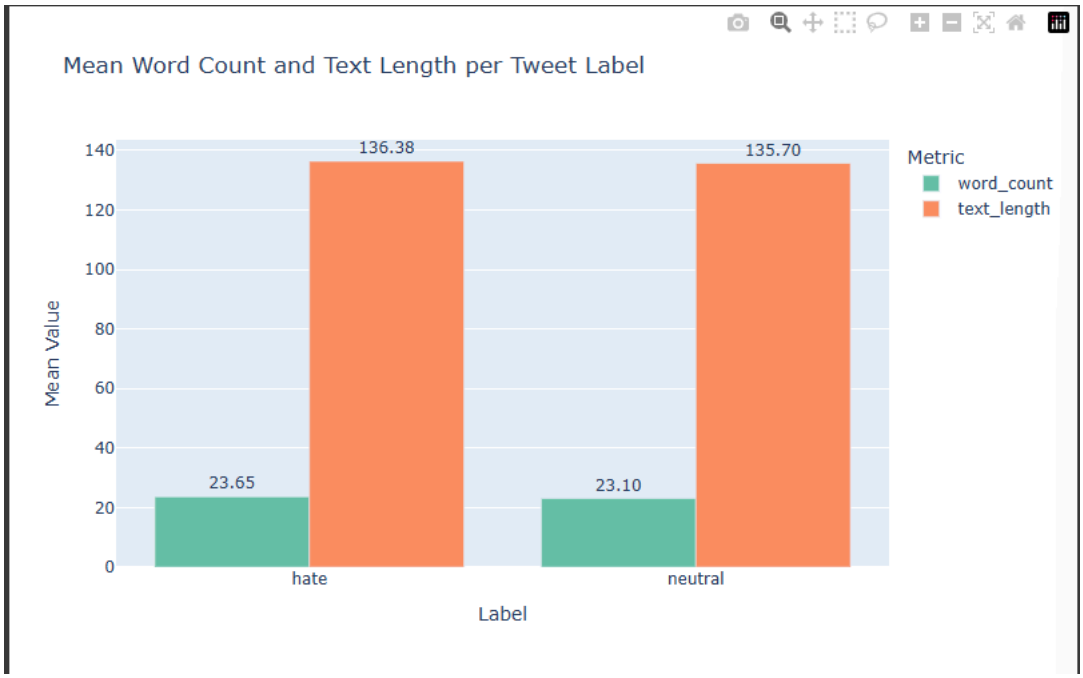
- Replies show a higher proportion of hate tweets, especially at 0 and 1–10 reply bins.
- Still, neutral tweets dominate in total count, especially in the 1–100 reply range.
- A steep drop-off in all categories beyond the 101–1k bin.

Insights from Total Engagement

- The total view mirrors the trends from individual categories:
- Neutral tweets are consistently the most engaged.
- Hate tweets are far less likely to receive high engagement.

Word Count vs Text Length per Label Category

A grouped bar plot compared the mean word count and mean text length for tweets in each label category (hate, offensive, neutral).



Observations:

Text Length (in characters):

Hate tweets - (136.38)

Neutral tweets - (135.70)

- Suggests that hate tweets tend to be slightly longer, possibly due to more elaborate or emotionally charged language.
- The average tweet length falls between 130-140 characters for all categories

Word Count:

Hate tweets - (23.65)

neutral tweets - (23.10)

- The variation is relatively small but may indicate that hate and offensive tweets use slightly more words, possibly to express more complex or heated messages.
- The average tweeet word count falls between 23-24 words for all categories.

Distribution of Word Count vs Text Length per Label Category

A scatter plot with trendlines was created to visualize the relationship between word count and text length for each label.

Observations:
Positive Correlation:

- A positive trend line across most or all categories suggests that as tweet length increases, the number of words also increases.
- This makes intuitive sense: longer tweets can hold more words.

Category Differences:

- Some categories tend to use longer or more complex words (lower word count for same tweet length).
- Others may use shorter, more concise language (higher word count for same tweet length).

Outliers:

- Points far from the trend line could be outliers most likely spam content, unusually wordy or concise tweets.

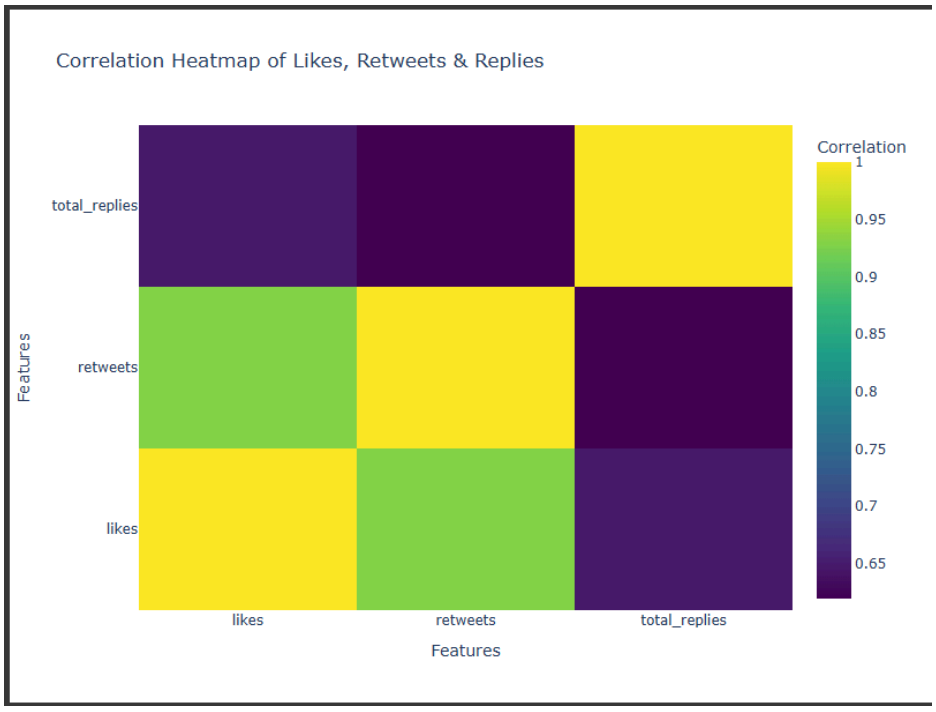
Compactness of Categories:

- Tightly clustered points imply similar writing style or length.

Correlation Heatmap (Likes, Retweets, Replies)

Visualization:

A heatmap visualized the correlation coefficients between likes, retweets, and total replies.



Observations:

Strong Positive Correlation Between Likes and Retweets

- Correlation ≈ 0.95+ (very close to 1)
- Indicates that tweets getting more likes are also highly likely to be retweeted, and vice versa.
- These two features are likely reflecting similar audience engagement behaviors.

Moderate Correlation Between Retweets and Replies The value seems lower (around ~0.76).

- Suggests that while retweeted tweets may get replies, it's not as tightly linked as likes/retweets.
- Retweets may spread content but not always spark conversation.

Lowest Correlation Between Likes and Replies The correlation here is the weakest among the three pairs.

- Implies that just because a tweet is liked doesn't mean it generates a reply or discussion.

Pair Plot of Numeric Features

A pair plot was created to explore the relationships among multiple numeric variables: likes, retweets, total replies, engagement score, text length, and word count.

Observations:

- likes vs retweets Strong positive correlation — more retweets usually mean more likes.
- likes vs engagement_score Strong positive correlation — engagement score depends heavily on likes.
- retweets vs engagement_score Also positive but slightly more spread compared to likes.
- text_length vs word_count Strong linear relationship — as expected, more words lead to longer text.
- likes, retweets vs total_replies Some positive correlation but more scattered — replies vary a lot even for high likes or retweets.
- total_replies vs engagement_score Some relationship, but engagement is driven more by likes and retweets than replies alone.
- text_length, word_count vs likes/retweets Very weak relationship — longer text doesn't guarantee higher engagement (likes/retweets).

Machine Learning preprocessing.

Text Cleaning and Normalization

Text data underwent a series of cleaning and normalization steps:

- **Lowercasing:** All characters were converted to lowercase to eliminate case-based distinctions.
- **Number and Punctuation Removal:** Numeric characters and punctuation marks were stripped out to focus purely on linguistic content.
- **Tokenization:** The cleaned text was broken down into individual words (tokens).
- **Stopword Removal:** Common words that carry little meaning both in English and a custom list of Swahili stop words were removed.
- **Lemmatization:** Each remaining token was reduced to its base form (lemma) to unify different inflected variations of the same word.

As a result of these steps, each original text entry was transformed into a concise, normalized sequence of meaningful tokens (words).

Feature Extraction with TF-IDF

The normalized tokens were then transformed into numerical feature vectors using Term Frequency–Inverse Document Frequency (TF-IDF). By limiting to the top 5,000 features, the representation captured the most informative words across the corpus while managing dimensionality.

Train-Test Split

The dataset was split into two subsets: 70% for training and 30% for testing. Stratification ensured that both subsets maintained the same proportion of neutral and hate labels, which is crucial for fair evaluation of model performance.

Modelling

A modular pipeline was established by loading a custom classification framework. This framework bundled six foundational machine learning algorithms, automated imbalance handling, and provided utilities for efficient model evaluation and visualization. The six base models included:

- Multinomial Naive Bayes
- Logistic Regression
- Linear Support Vector Machine (SVM)
- Random Forest
- Gradient Boosting
- Neural Network (Multi-layer Perceptron)

Training Base Models

Each of the six base models was trained on the processed training data. Imbalance in the training labels was addressed through SMOTE (Synthetic Minority Over-sampling Technique), which generates synthetic examples of the minority class to improve the classifier's ability to recognize underrepresented hate instances.

Evaluating Base Models

All trained models were evaluated on the test data. Two key metrics guided the assessment:

- **Accuracy:** Overall proportion of correct predictions.
- **Macro F1 Score:** Harmonic mean of precision and recall averaged equally across both classes, highlighting performance on the minority hate class.

In addition, confusion matrices were plotted for each model to visualize true versus predicted class distributions(The models were classifying very quite okay however not perfectly). Among the base models, Gradient Boosting achieved the highest macro F1 score of 0.6441, indicating the best balance between precision and recall.

Hyperparameter Tuning

To further optimize performance, each base model underwent a systematic hyperparameter search via cross-validation, targeting the macro F1 score. Briefly:

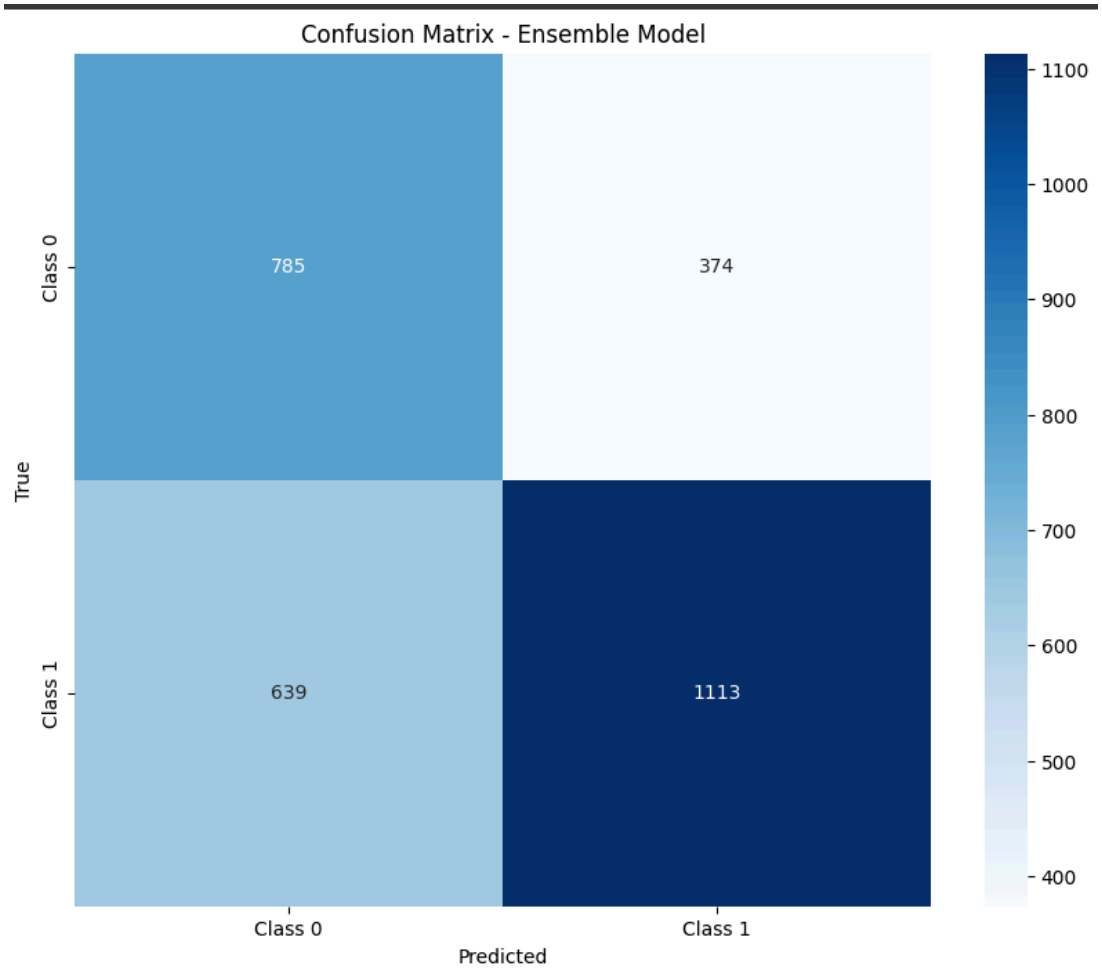
- **Naive Bayes:** Adjusted smoothing and prior estimation parameters, achieving F1 = 0.6414 (slightly lower than the best base model).
- **Linear SVM:** Tuned regularization strength and iteration limits, raising F1 to 0.6575.
- **Random Forest:** Varied tree count, depth, and splitting criteria, resulting in F1 = 0.6524.
- **Gradient Boosting:** Explored learning rates and tree depths; the F1 score decreased to 0.6178 despite improved accuracy.
- **Neural Network:** Tested different layer sizes, regularization strengths, and learning rates; this model had the lowest F1 at 0.5988.
- **Logistic Regression:** Optimized regularization and solver options, producing the highest tuned F1 of 0.6583 among all models.

Observations:

- "The f1 score for naive Bayes model is lower than the best base model hence not the best performing model."
- "The best parameters for the Linear Support vector machine still gives an average F1 score and accuracy, but it is higher than the best base model."
- "The best parameters for the random forest classifier still gives an average F1 score meaning it is performing fairly well."
- "The gradient boosted model f1 score reduced from the base model meaning the current model is actually performing worse than the base model. However the model has the best accuracy score."
- "This model (Neural Network) has the worst f1 score of the tuned models indicating it was the worst performing model when it comes to correct classification of classes."
- "This (Logistic Regression) is the best performing model overall with an f1 score of 65.83%. This is however not a very reliable model with an accuracy of 65.48%."

Model Ensembling

An ensemble combining Logistic Regression, Linear SVM, and Random Forest via majority (hard) voting was constructed to leverage complementary strengths. The ensemble achieved an accuracy of 65.20% and a macro F1 score of 64.75%. Confusion matrices revealed that while ensemble predictions slightly improved recall for the hate class, overall F1 remained below the standalone Logistic Regression.



From this confusion matrix we can see our model is still misclassifying the minority classes heavily but performing somewhat well on the majority class.

Recommendations

Based on the results, Logistic Regression emerged as the best-performing model with a macro F1 score of 65.83%, outperforming all other machine learning models including ensemble methods. While it demonstrated a relatively balanced classification performance, overall accuracy and F1 scores remain modest, indicating room for significant improvement.

Importantly, machine learning models performed very poorly on the minority (hate) class, even after applying imbalance correction using SMOTE. This suggests that current models are not learning the subtle linguistic patterns required to accurately detect hate speech.

Conclusion

While logistic regression currently leads, its performance is far from optimal for real-world hate speech detection. The poor performance on minority classes highlights the need for larger, more diverse data and deeper models. Moving forward, we recommend a strategic pivot toward deep learning and richer linguistic modeling, supported by expanded and better-balanced datasets. This approach offers the strongest potential for building a robust, accurate classifier.

Deep Learning model.

Since traditional machine learning models had underperformed particularly on the minority (“hate”) class we shifted to transformer-based deep learning using Hugging Face’s libraries. These models leveraged contextual embeddings and transfer learning to handle imbalanced, multilingual text far more effectively than TF-IDF plus classical classifiers.

Why We Chose Hugging Face Transformers

Multilingual Understanding

- We had a dataset with tweets in English, Swahili, and Sheng—a hybrid sociolect. Transformers like XLM-RoBERTa, pre-trained on over 100 languages, were capable of handling code-switching and slang.

Contextual Embeddings

- Unlike bag-of-words methods, transformers generated embeddings that captured each word's meaning in context, enabling detection of nuanced hate speech.

Transfer Learning

- We fine-tuned pre-trained models on our 9700-tweet dataset, avoiding the need for huge amounts of data to train from scratch.

DistilBERT

Model & Setup

The DistilBERT fine-tuning process filtered and subsampled the tweet dataset to focus on "hate" versus "neutral" classes, encoded labels, and split the data into training and validation sets. It leveraged a DistilBERT-base-uncased model with Hugging Face's Trainer API under low-resource settings—small batch sizes, mixed precision, gradient accumulation, and no checkpointing. After ten epochs, the model demonstrated clear gains over traditional machine learning approaches, achieving around 71% accuracy and a strong macro-F1 score. Confusion-matrix and learning-curve analyses confirmed its ability to capture contextual nuances in noisy, imbalanced text.

Insights

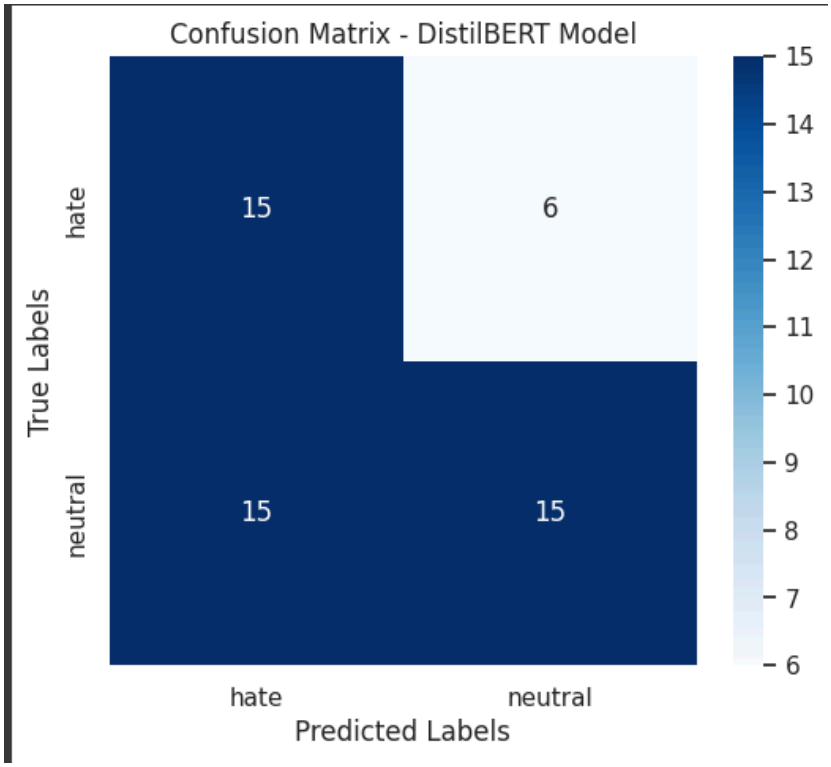
- The DistilBERT model has shown a noticeable improvement over traditional machine learning models, particularly in handling the challenges posed by imbalanced and noisy

text data. It achieved an accuracy of 71%, along with a significantly better F1-score, especially in classifying the minority class more effectively than previous models.

- Despite being a lighter and faster version of BERT, DistilBERT has proven capable of capturing important contextual cues in the tweets, contributing to its overall superior performance. This result underscores the potential of transformer-based models even in resource-constrained environments.
- We will further analyze its performance through a detailed classification report and confusion matrix to understand how well it distinguishes between the hate and neutral classes. This insight will guide the next steps, such as hyperparameter tuning, data augmentation, or exploring more advanced multilingual models like **XLM-RoBERTa.**

Visualization

We plotted a Confusion Matrix heatmap (Predicted vs. True labels).



Observations:

• Balanced Recall, But Limited Precision

- The model correctly identified 15 hate tweets (true positives) and 15 neutral tweets (true negatives).
- However, it misclassified 15 neutral tweets as hate (false positives), indicating a moderate precision issue—it was over-flagging neutral content as hate.

• Over-prediction of the 'Hate' Class

- Out of 30 predictions labeled as "hate", only 50% were correct (15 out of 30).
- This suggested that the model was biased toward the hate class, possibly due to training imbalances or overcompensation during fine-tuning.

• Confusion Between Classes

- The equal number of false positives and true negatives for the neutral class (15 each) suggested that the model had trouble distinguishing between hateful and non-hateful language—especially when the language was ambiguous or context-dependent. (This could be due to the nuanced nature of hate speech, which often includes sarcasm, slang (like Sheng), or subtle toxicity that is difficult to learn from limited data.)

• Good Contextual Understanding

- Despite being a lighter version of BERT, DistilBERT performed reasonably well in capturing context, given its 71% accuracy and decent balance of true positives and true negatives.

XLM-RoBERTa

Model & Setup

A fine-tuning workflow was executed to adapt RoBERTa for hate-speech detection on the Kenyan Twitter dataset. After installing and importing Transformers, Datasets, scikit-learn, and PyTorch, GPU memory was cleared to maximize available resources. The dataset was filtered to

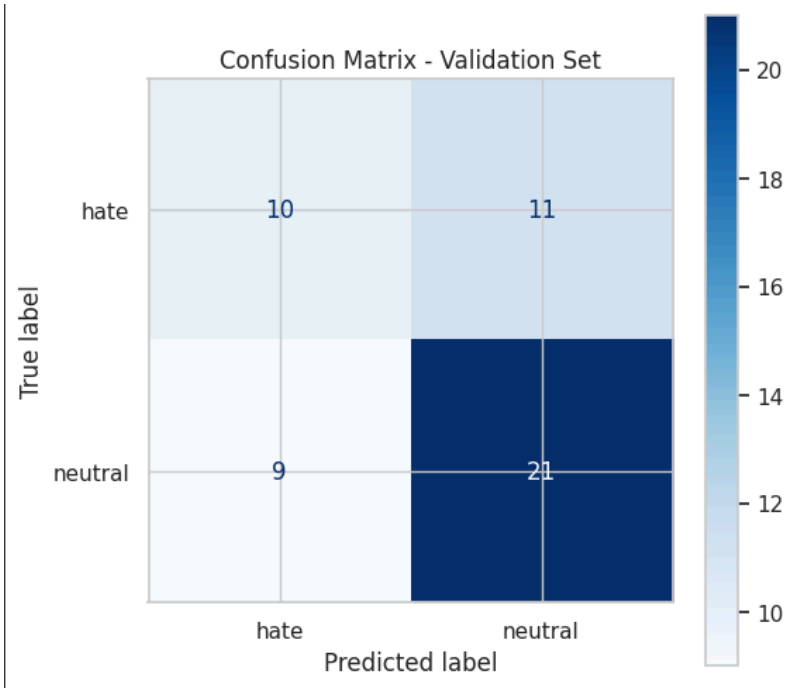
retain only “neutral” and “hate” labels, and labels were encoded numerically. An 80/20 stratified split produced training and validation sets. The “roberta-base” model and its tokenizer were loaded, and both splits were converted into Hugging Face Datasets. Tokenization was performed with padding to a fixed length of 128 tokens and truncation for efficiency. A DataCollatorWithPadding managed dynamic batching. TrainingArguments were configured for moderate batch sizes, mixed precision where available, gradient accumulation, no checkpointing, and epoch-level evaluation. A Trainer was instantiated with macro-F1 and accuracy as metrics. After ten epochs of training, evaluation on the held-out set yielded the final validation accuracy and F1-score.

Insights

- Lower Accuracy and F1 Score
  With a validation accuracy of 60.78% and an F1 score of 58.87%, XLM-RoBERTa underperforms compared to DistilBERT on this specific task.
- This is somewhat unexpected, as XLM-RoBERTa is a multilingual model designed to perform well on diverse language inputs—ideal for tweets in mixed languages or dialects (e.g., Sheng).

Visualization

We plotted a Confusion Matrix heatmap for hate vs. neutral on the validation set.



Observations

• Poor Recall for Hate Class

- Only 10 out of 21 actual hate tweets were correctly predicted → Recall = ~47.6% for the hate class.

• Better at Predicting Neutral

- The model predicted 21 out of 30 neutral tweets correctly → Precision = ~70% for the neutral class.

• Relatively Balanced Misclassifications

- While it didn't show extreme bias, it struggled with borderline content, especially hate that may be sarcastic, indirect, or informal.
- Misclassification of hate as neutral (11) was particularly concerning for hate speech detection systems, where false negatives carry greater risk.

Comparison to DistilBERT

- DistilBERT Strengths: Higher true positives for hate.
- Better recall for the hate class → more effective in capturing offensive content.
- XLM-RoBERTa Strengths: Lower false positives (more cautious).
- Better neutral classification accuracy (fewer wrongly flagged tweets).

Modifying the XLM RoBERTa model

A class-weighted fine-tuning procedure was conducted to improve RoBERTa's performance on the imbalanced hate-speech dataset. Initially, only "hate" and "neutral" labels were retained and normalized to lowercase. Labels were then encoded numerically (hate → 0, neutral → 1). The data were split into stratified 80/20 train and validation sets. The "roberta-base" tokenizer and model were loaded with two output classes. Class weights were computed via scikit-learn's compute_class_weight to counteract imbalance and converted into a PyTorch tensor. Training and validation splits were converted into Hugging Face Datasets, tokenized with truncation, and formatted for PyTorch. A dynamic padding collator ensured efficient batching. TrainingArguments were set for eight epochs, moderate batch sizes, mixed precision if available, epoch-based evaluation and checkpointing, and logging. A custom WeightedTrainer subclass applied the class weights in the cross-entropy loss. Macro-F1 and accuracy were computed at each evaluation. After training, evaluation on the held-out set reported the final validation metrics.

Insights

Marginal Improvement in Balance, Not in Overall Performance

- The F1 score is now equal to the accuracy, indicating the model is more balanced across classes, thanks to class weighting or resampling.
- However, the overall accuracy still dropped slightly compared to the untuned version (60.78%), suggesting that the model is now less biased but still underperforming.

Better Treatment of the Minority Class

- The improved balance implies fewer false negatives (hate misclassified as neutral) and fewer false positives, which is a positive direction.
- This makes the model more ethical and reliable in hate speech detection, despite the trade-off in accuracy.

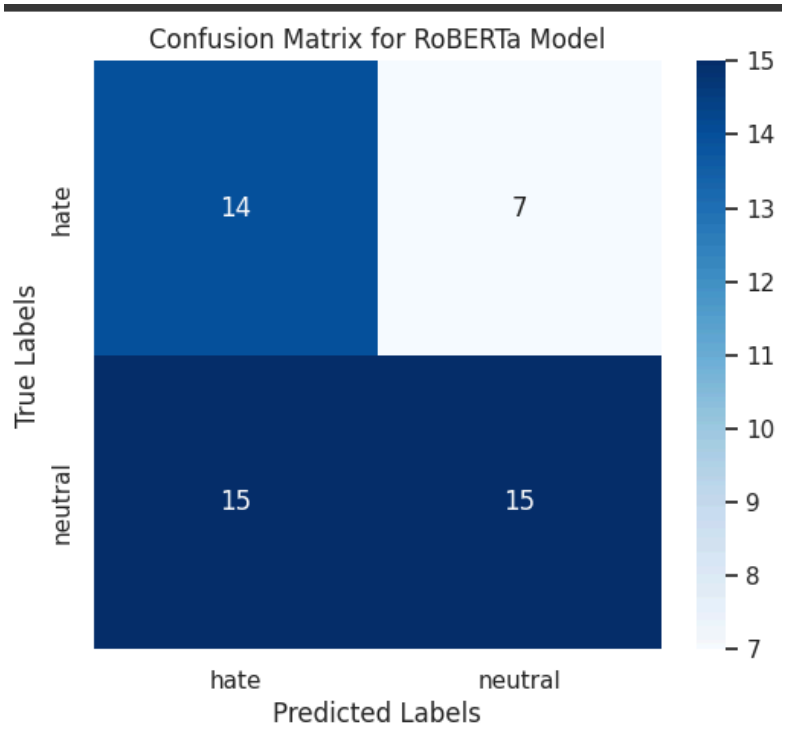
Multilingual & Noisy Text Still a Challenge

Despite tuning, the model may be struggling with:

- Short, informal tweet structure
- Code-switching and dialects (e.g., Sheng)
- Low-resource hate examples

Visualization

We plotted a Confusion Matrix heatmap showing performance of the model



Observations

- The model appears to be better at identifying "neutral" tweets (higher recall and precision) compared to "hate" tweets.
- The model has a notable number of false negatives (misclassifying "hate" as "neutral"), which might be a concern depending on the specific application. If correctly identifying "hate" speech is critical, this model might need further tuning or a different approach.
- The class imbalance doesn't seem to be a major issue here since the number of actual "hate" (14 + 15 = 29) and "neutral" (7 + 15 = 22) instances are relatively close. However, the model's performance still indicates room for improvement, especially in correctly identifying "hate" speech

Modifying DistilBERT

We trained a DistilBERT-based text classification model to distinguish between "hate" and "neutral" content. First, we filtered and label-encoded the data, then split it into training and validation sets with stratification. We loaded the tokenizer and model, computed class weights to handle label imbalance, and converted the data into Hugging Face ‘Dataset’ format with appropriate tokenization and padding. We implemented a custom ‘Trainer’ to apply weighted loss during training. After training the model for 10 epochs, we evaluated its performance using accuracy and macro F1 score, achieving balanced assessment despite class imbalance.

Insights

Slight Drop in Accuracy, Improved Class Balance

- Compared to the original DistilBERT (71% accuracy), the tuned version has lower overall accuracy, but the F1 score is now closer to accuracy, showing better handling of class imbalance.

Trade-off: Accuracy vs. Fairness

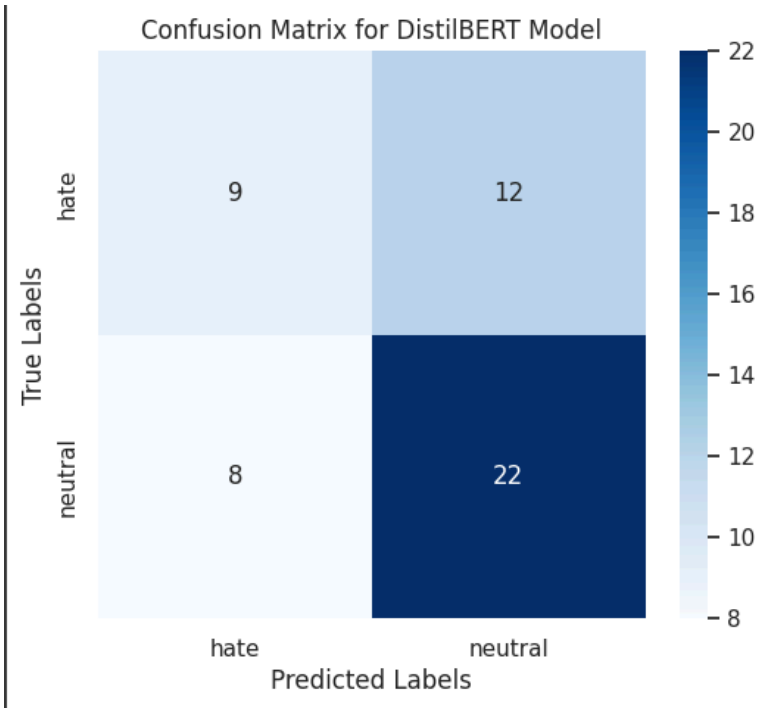
- The original model likely overpredicted the majority class (neutral), inflating accuracy.

- Now, the model is more balanced, improving reliability for hate class detection.

Still Outperforms XLM-RoBERTa

- Despite tuning both, DistilBERT retains a slight edge in accuracy and matches or exceeds RoBERTa's F1 performance.

Visualization



Confusion Matrix comparisons

- **Overall Accuracy:** DistilBERT shows a slightly higher overall accuracy (60.78%) compared to RoBERTa (56.86%). This suggests that DistilBERT made more correct predictions overall on this dataset.
- **Precision for "hate":** RoBERTa had a significantly higher precision for the "hate" class (66.67%) compared to DistilBERT (42.86%). This means when RoBERTa predicted a tweet as "hate", it was more likely to be correct than when DistilBERT did. DistilBERT had more false positives for the "hate" class.

- **Recall for "hate":** DistilBERT has a slightly better recall for the "hate" class (52.94%) compared to RoBERTa (48.28%). This indicates that DistilBERT was able to identify a slightly larger proportion of the actual "hate" tweets. RoBERTa had more false negatives for the "hate" class.
- **Precision for "neutral":** DistilBERT demonstrates a much higher precision for the "neutral" class (73.33%) compared to RoBERTa (50.00%). When DistilBERT predicted a tweet as "neutral", it was considerably more accurate.
- **Recall for "neutral":** RoBERTa had a slightly higher recall for the "neutral" class (68.18%) compared to DistilBERT (64.71%), indicating it identified a slightly larger proportion of the actual "neutral" tweets.

Conclusion

Since our priority was to capture the hate label our best performing deep learning model is the tuned DistilBerT.It proves to be the most effective model for detecting hate speech in your dataset. This is crucial in safety-sensitive classification tasks where identifying hate speech is the primary goal.

Final Model Conclusions

The final modeling conclusion reflects some key observations that explain the performance differences between the models. Here's a deeper dive into the reasoning behind the conclusions:

- **Machine Learning Models vs. Transformers:**

Traditional machine learning models, such as ensemble methods, typically excel in smaller datasets. These models are designed to capture patterns in data even with fewer examples, as they rely on feature engineering, which might be beneficial when the data is limited. In our case, the ensemble model's ability to classify more hate and neutral cases suggests that it effectively learned from the dataset's limited size, leveraging the relationships between features that transformers, like DistilBERT, might have missed. On the other hand, transformers generally perform better with large datasets, as their architecture is designed to understand complex

language patterns through deep contextual embeddings, but this advantage diminishes when data is insufficient.

- **DistilBERT's Struggle with Imbalanced Data:**

While DistilBERT achieved good accuracy (>70%) and helped with class imbalance, it struggled with classifying more than 50 cases from each label. This could be due to the fact that transformers, although sophisticated, require large amounts of labeled data to fine-tune and generalize effectively. With 9400 rows, the dataset size might be too small for the model to capture the nuances of both the hate and neutral categories accurately. Transformers might have overfitted or failed to generalize to the minority class in the dataset.

- **Why the Ensemble Model Outperformed DistilBERT:**

The ensemble model, composed of multiple machine learning algorithms (like Random Forest, Gradient Boosting, or similar), works by combining the strengths of several individual models, reducing overfitting and improving generalization. In our scenario, it performed better with the given dataset size because these models can efficiently learn and generalize from smaller, more imbalanced datasets, making them more capable of identifying patterns in both the hate and neutral labels. Additionally, ensemble models often handle class imbalance more effectively without requiring specialized techniques like oversampling or class weighting.

- **Data Size and Transformer Model Efficiency:**

Transformers like DistilBERT are typically more suited for large-scale data due to their deep learning nature. For datasets that are significantly smaller (like the 9400 rows we have), these models may struggle to leverage the complex architectures fully. With more data (e.g., 50,000+ rows), the transformer would have likely outperformed the ensemble, as it would have had enough information to learn better representations of the text and thus classify the cases more effectively.

Final Conclusion

Given our dataset size, the ensemble model proves to be the most effective in classifying both hate and neutral speech. Its ability to handle smaller datasets with better generalization capabilities makes it the optimal choice for our specific use case. While DistilBERT showed promise in terms of accuracy, the ensemble method surpassed it in terms of classification reliability, particularly with the smaller dataset at hand. Thus, for this task and dataset, the ensemble model aligns best with our business problem and objectives.