5/12/2025

TWITTER SENTIMENT ANALYSIS PROJECT

PROJECT:
Building an NLP Model to Classify Tweet Sentiment

PRESENTED BY:
Group 8

PRESENTED TO:
Stakeholders

Our team

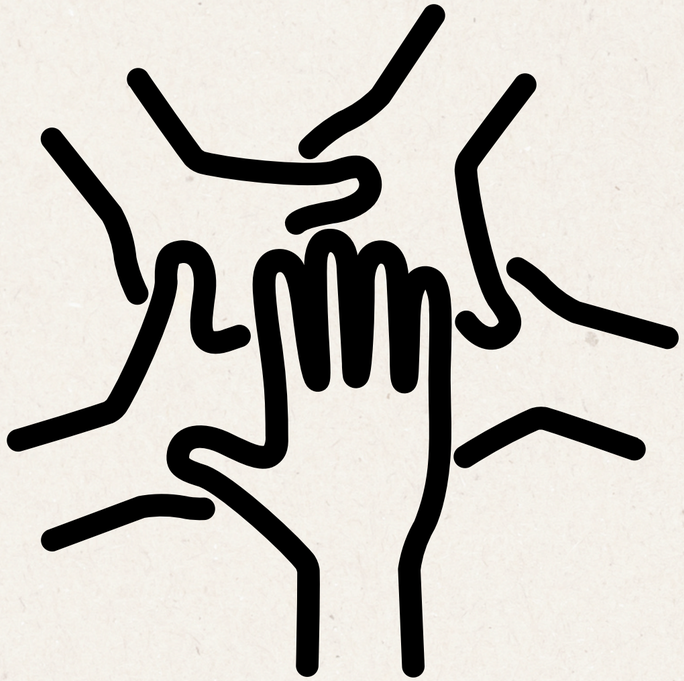
Priscillah Giriama

Sharon Nyakeya

Barclay koin

Michelle Maina

Fatuma Tari



Agenda

1	Project Overview
2	Business Understanding
3	Data overview
4	Data cleaning
5	Exploratory Data Analysis
6	Data preparation
7	Modeling
8	Model Evaluation
9	Error Analysis
10	Conclusion

Overview

This project involved developing a machine learning-based NLP model that analyzes Twitter data to automatically classify sentiment—positive, negative, or neutral—across more than 9,000 human-labeled tweets about Apple and Google products.

Why This Matters:

- Real-time brand monitoring and reputation management
- Customer feedback analysis at scale
- Automated social media listening system

Business Understanding

Goal

Build a comprehensive sentiment analysis system for tech brand mentions on social media, built using the CRISP-DM methodology. This project classifies emotions (Positive/Negative/Neutral) toward brands like iPhone, iPad, Google, and various apps.

Business Objectives

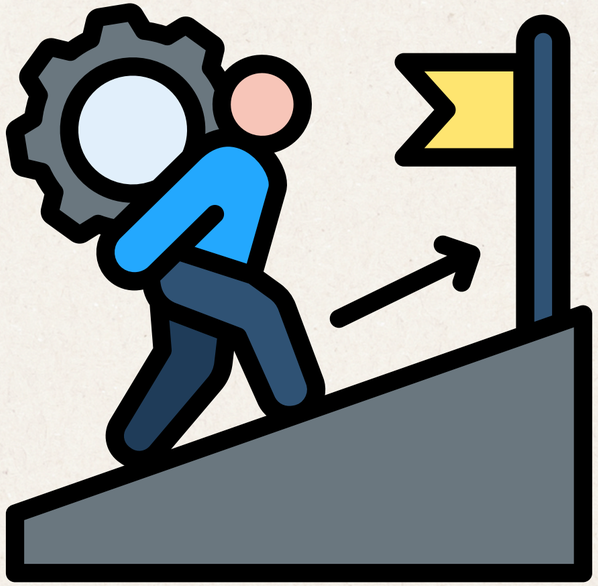
- Monitor brand sentiment
- Generate insights for marketing and product teams

Success Criteria

- Understand customer sentiment towards tech brands/products
- Enable real-time sentiment monitoring
- Provide actionable insights for brand management
- Achieve above 80% accuracy with F1-score above 0.80

Stakeholders

- Marketing – monitors brand trends
- Customer Support – identifies complaints
- Product Team – understands feature feedback



Constraints

- Strong class imbalance
- Potential computational constraints (classical ML vs Transformer)

Data Overview

Data Source

- Source: 9,093 real tweets rated by human experts
- Focus: Tweets about iPhones, iPads, Android, Google services
- Quality: Each tweet was reviewed and labeled by people for accuracy

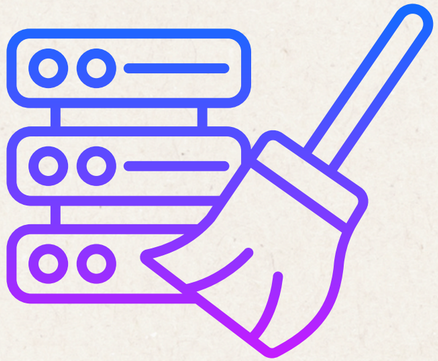
Data Understanding

- Concrete steps
- Inspect the CSV for columns and data types.
- Confirm the target column name and classes. Typical columns: tweet_id, text, sentiment (values like positive/neutral/negative).
- Run EDA: class distribution, length distribution, missing values, sample tweets.

Think of it as: A collection of digital conversations about technology that we're going to analyze.

Our Job: Take this human-reviewed data and teach a computer to do the same thing automatically

Data Cleaning



1.Cleaning the Data and Making It Usable

First Step: Removed 22 duplicate tweets

- These were identical or nearly identical tweets
- Examples: Same news story tweeted multiple times

Impact: Went from 9,093 to 9,071 unique conversations

Second Step: Simplified our focus

- Original data had 3 columns of information per tweet
- We discovered one column ("emotion_in_tweet_is_directed_at") was missing 64% of the time

Smart Decision: Removed this piece entirely since our main goal is understanding feelings, not tracking companies

2.Final Data Quality

After Cleaning:

- 9,070 perfectly clean tweets
- Every tweet has its text (no missing content)
- Every tweet has a clear sentiment label (no missing feelings)
- Simple structure: Text + Feeling = Ready for analysis

Think of it as: 9,070 examples where a human has already told us "this is what the tweet says" and "this is how the person feels

Sentiment Class Distribution Analysis

Analyzing 9,070 tweets revealed:

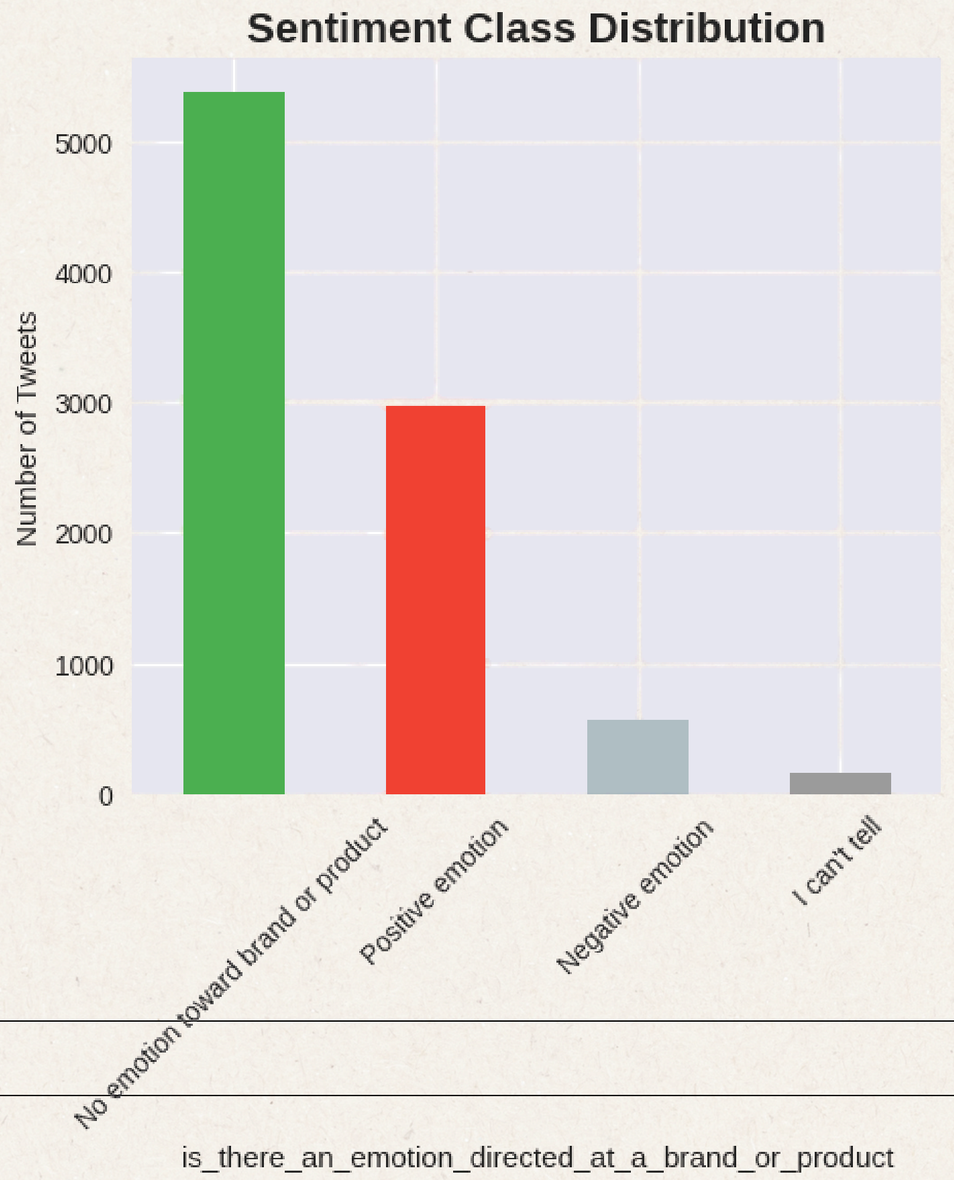
Most Conversations Are Neutral (59.3%)

- People are mostly sharing information
- Example: "Google announced a new product today"
- Not expressing strong feelings either way

When People Do Have Feelings:

- 6.3% are Negative ("My iPhone battery dies too fast!")
- 32.7% are Positive ("Loving my new iPad!")
- 1.7% are Hard to Tell (Ambiguous or sarcastic)

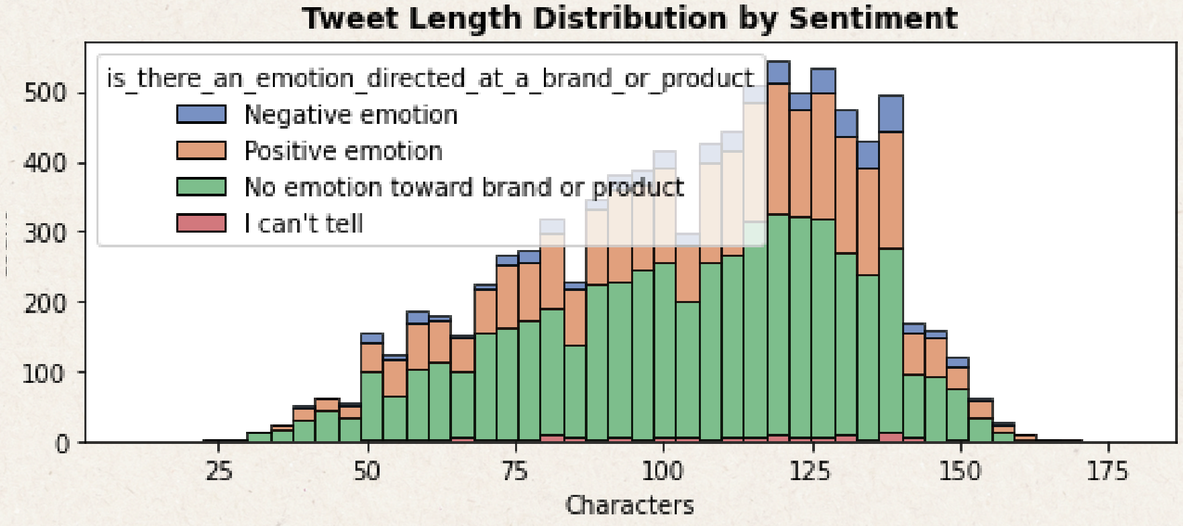
Key Insight: Most tech talk on Twitter is informational, not emotional

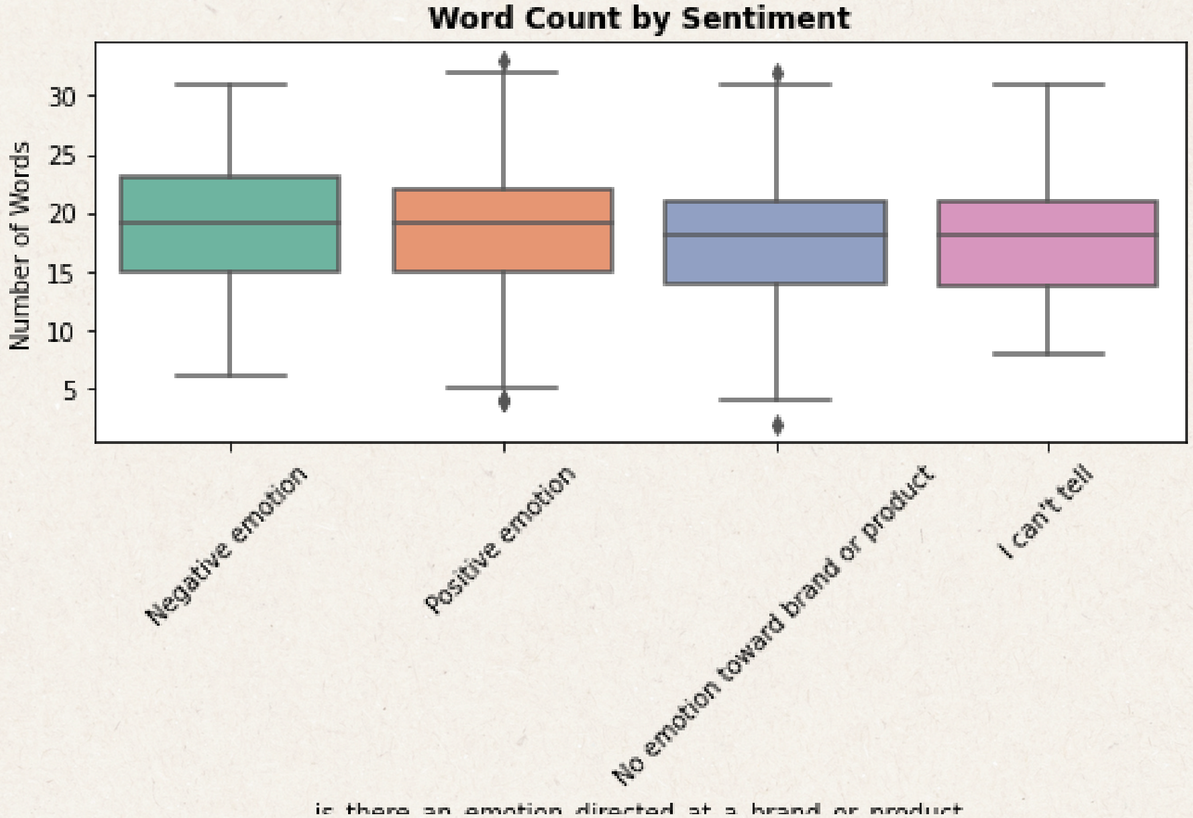


Tweet Length Distribution Analysis

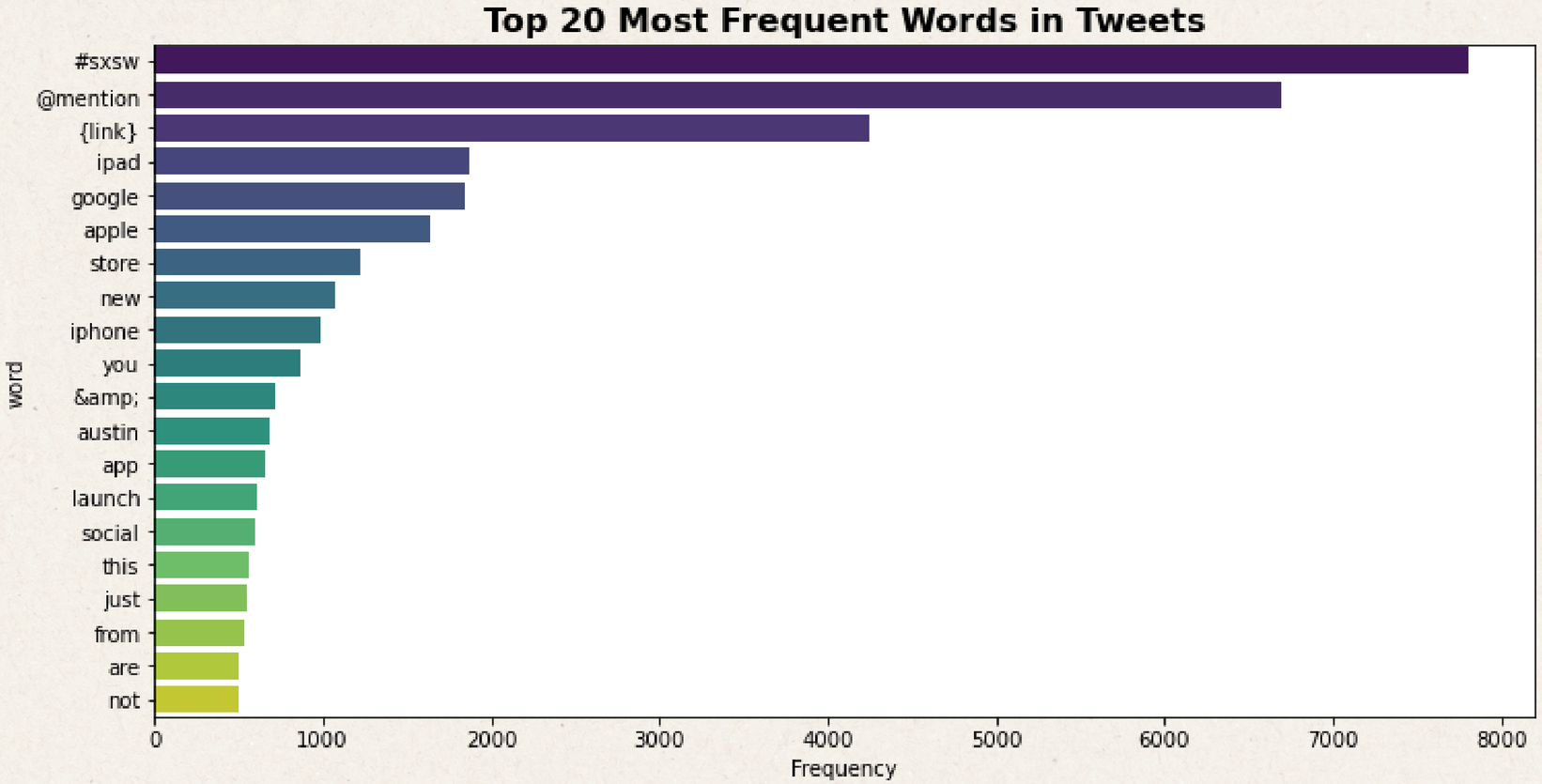
Method: Calculated character length for each tweet
Visualization: Histogram showing tweet length frequency
Insights:

- Most tweets: 100-150 characters
- Important for feature engineering decisions





Word Frequency Analysis



Top Words by Sentiment Category:

- **Positive:** "great", "awesome", "love", "good"
- **Negative:** "crappy", "hate", "problem", "bad"
- **Neutral/No emotion:** Product names, brand mentions

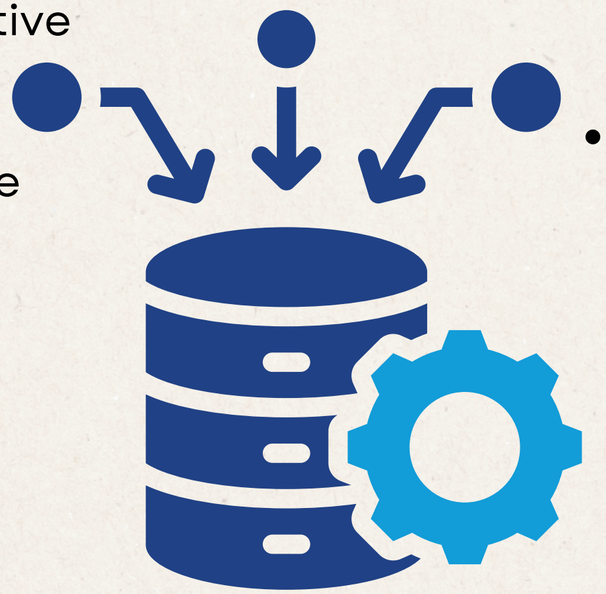
Data Preparation

Key preprocessing tailored for tweets

- Multi class classification - Correctly framed the tasks as 3 classes: positive, negative and neutral sentiments

- Advanced preprocessing - Two stage pipeline.

1. Stage 1 (advanced_clean_tweet): Normalisation + explicit sentiment-preserving token injection
2. Stage 2 (preprocess_text): Classic lemmatisation + stop-word removal



- Label Encoding - Mapped negative emotion with 0, neutral emotion as 1 and Positive emotion with 2

- Feature Engineering - The core insight we had is that on 140-character social text, punctuation and surface patterns are often stronger predictors than bag-of-words alone. So we deliberately engineered a parallel dense feature track from the raw tweet before any cleaning destroys the signal

Modeling

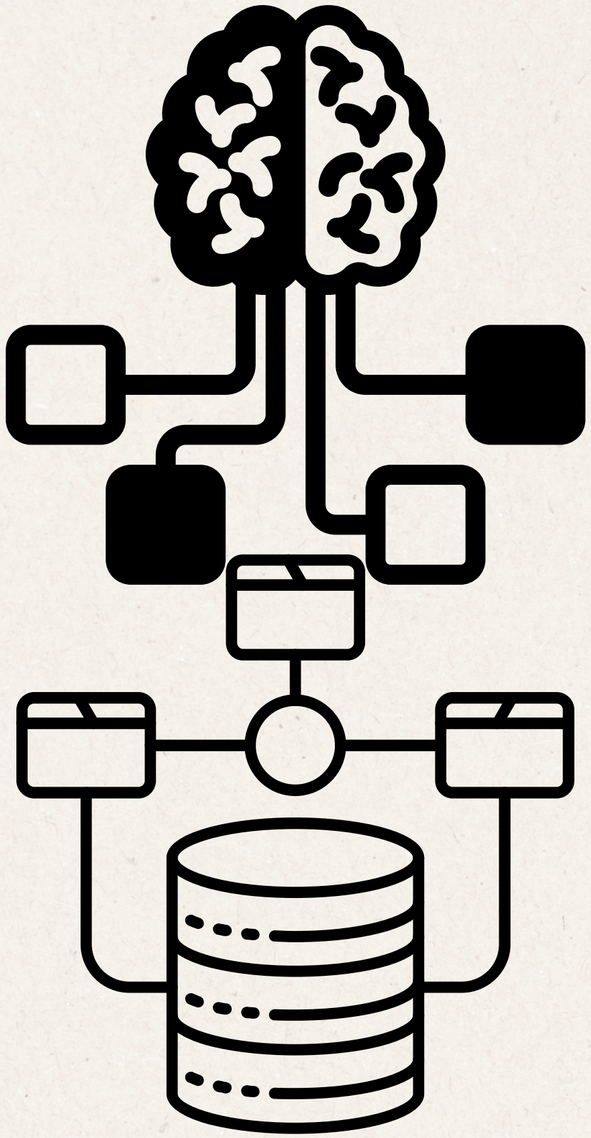
Model Evaluation & Interpretation We analyzed: Accuracy, Recall, F1 Cross-validation stability Confusion matrix Top predictive features

Feature Engineering & Modeling
We design features that help the model learn sentiment patterns:

- TF-IDF captures contextual words and phrases
- Numeric features capture expressions of emotion
- SMOTE corrects training imbalance
- A train/test split ensures fair evaluation

Training multiple models:

- Logistic Regression (Baseline model)
- Linear SVC - Random Forest
- Gradient Boosting
- XGBoost



Model Evaluation & Interpretation

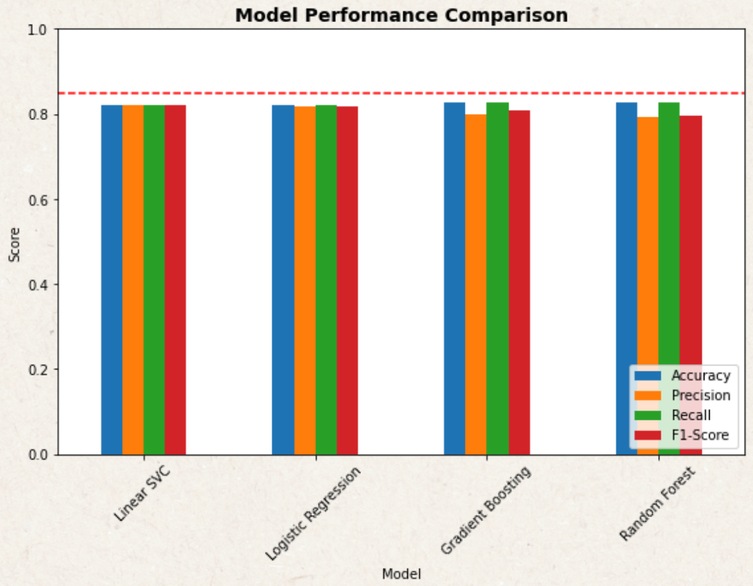
We analyzed:

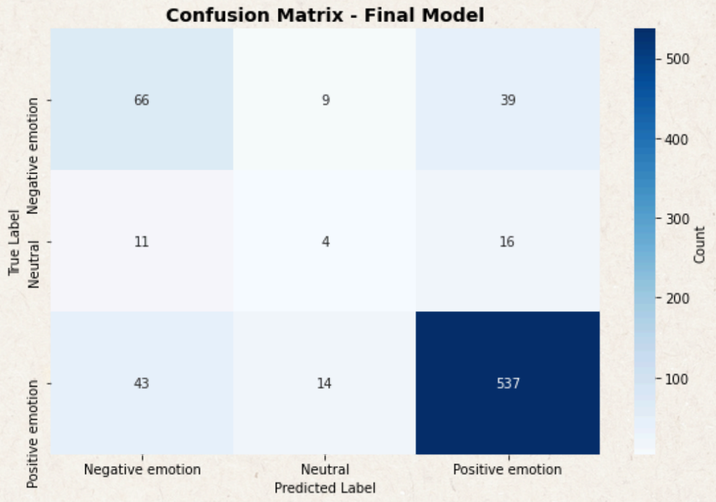
- Accuracy,
- Recall,
- F1 Cross-validation stability
- Confusion matrix
- Top predictive features

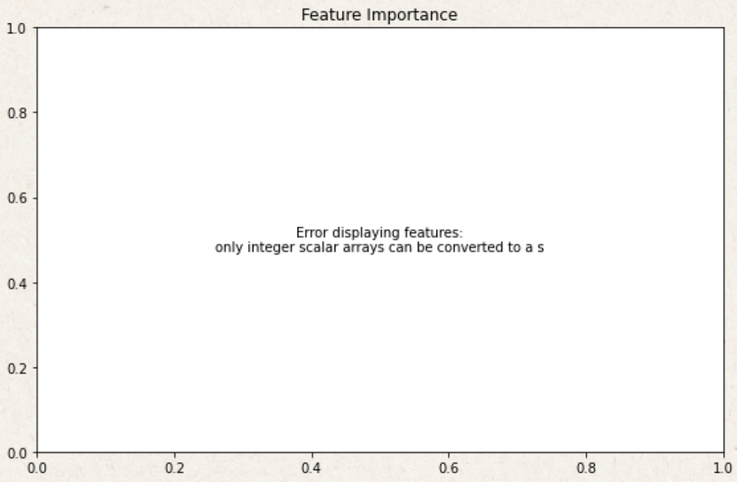
The model effectively classifies tweet sentiment with high reliability.

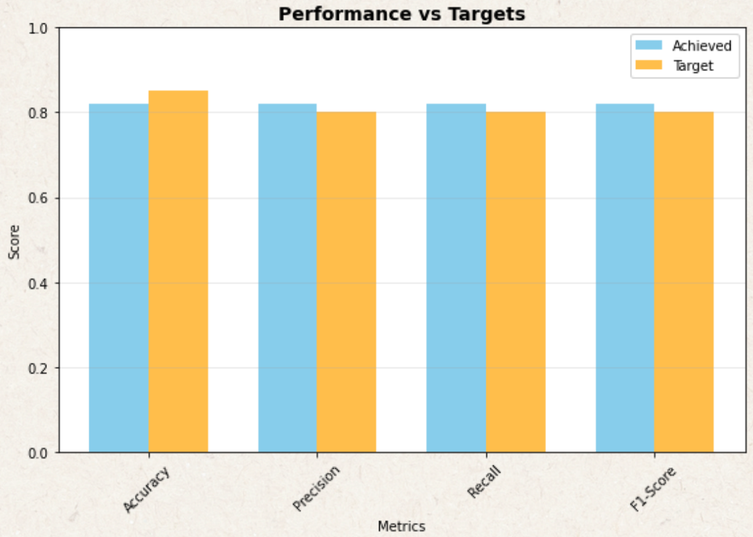
Engineered features from raw text, combined with TF-IDF representations of cleaned tweets, improved the model's predictive power.

This framework can be applied to brand monitoring, customer feedback analysis, or any application where sentiment classification of text is required.







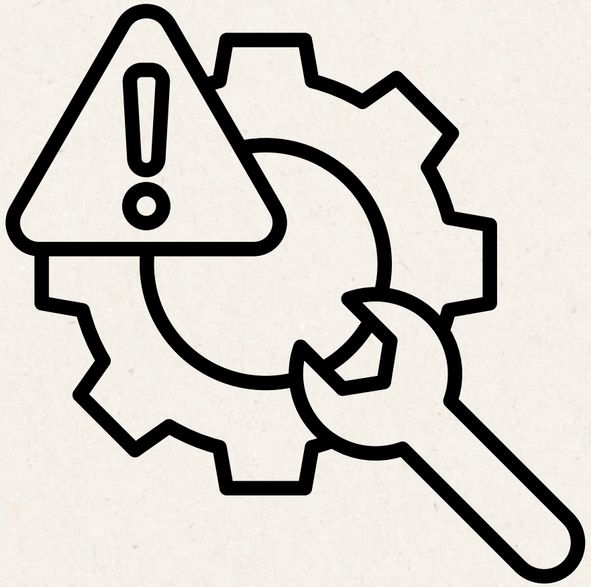


Error Analysis

Summary of Error Analysis

The majority of errors stem from: sarcasm mixed emotions vague statements slang/emoji-driven sentiment structural ambiguity

These limitations are common in classical NLP models and can be resolved using modern transformers and richer feature extraction.



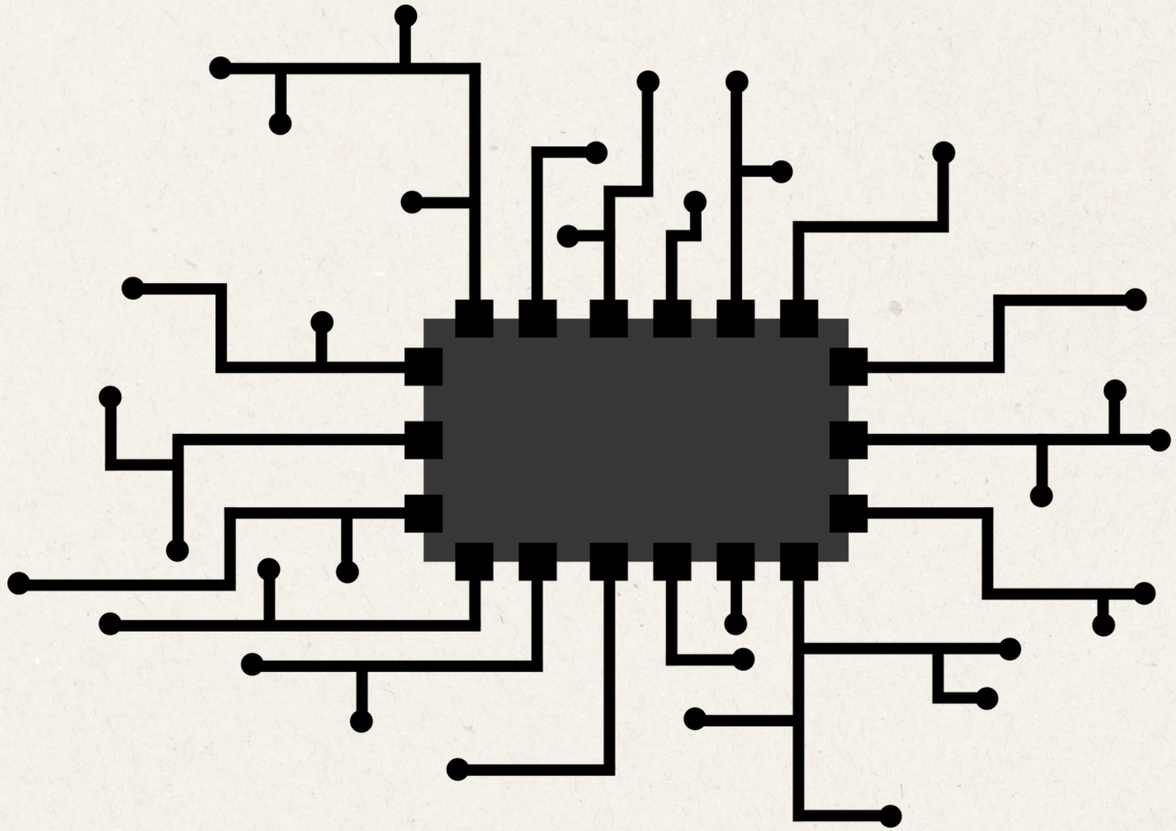
Examples from the System

What the System Sees & Understands:

Tweet: "Just got the new iPhone and the battery life is amazing! Best phone ever!"
➔ **System says:** POSITIVE ✓ (Keywords: amazing, best)

Tweet: "Google maps sent me to wrong location again. So frustrating!"
➔ **System says:** NEGATIVE ✓ (Keywords: wrong, frustrating)

Tweet: "Thinking about buying an iPad for school"
➔ **System says:** NO EMOTION ✓ (Just stating facts)



Why This Matters for the Business

Before Our System:

- Marketing guesses what customers want
- Support reacts to complaints (sometimes too late)
- Product teams rely on surveys (slow feedback)

After Our System:

- Real-time customer mood tracking
- Data-driven decisions
- Proactive problem solving
- Happy customers → Better products → More sales

THE BOTTOM LINE

We built a "customer feeling detector" that:

1. Listens to social media 24/7
2. Understands emotions in text
3. Alerts teams when needed
4. Provides insights for better decisions

For companies: It's like having a team of analysts reading every tweet and providing instant reports - but automated and scalable.

For customers: Their feedback gets heard faster and leads to better products.

Thank you for your Patience

Questions are Welcome

GITHUB https://github.com/Namuki-art/grp8-phase4-project