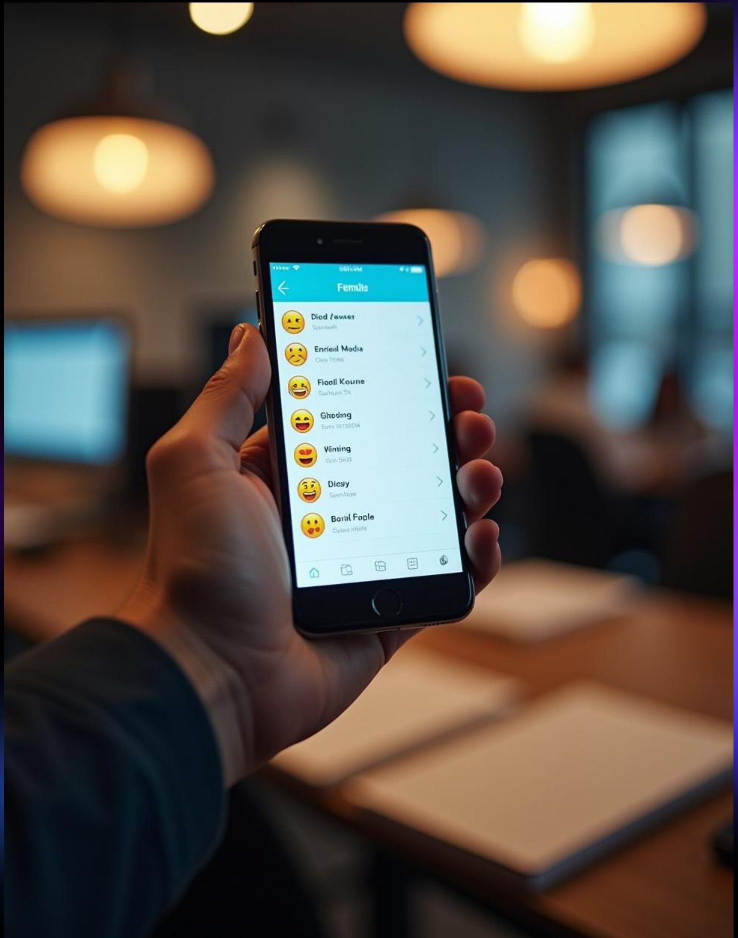
# Sentiment Analysis:

## Understanding Tweets



Group Members:
*Ryan Karimi*
*Harrison Kuria*
*Elizabeth Ogutu*
*Lewis Karanja*
*Rose Muthini*
*Hafsa M.Aden*

## Overview of Sentiment Analysis

01

- This project analyses thousands of tweets about Apple and Google to understand how people feel about their products.
- We built models that can detect whether a tweet is positive, negative, or neutral. These insights can help companies track public sentiment, identify areas for improvement, and benchmark against competitors.

Why this matters:

- Brand can understand customer perceptions in real time.
- Spot emerging issues before they escalate (e.g., product complaints).
- Measure the success of campaigns by monitoring sentiment shifts.
- Compare Apple vs. Google sentiment to assess competitive advantage.

## Business Understanding

01

- Twitter is a valuable source of real-time customer feedback, with over 611 million monthly active users worldwide as of 2025. Every day, users share their experiences, frustrations, and praises about a wide range of topics from politics and entertainment to lifestyle and technology products making it a rich platform for gauging public opinion.
- For global companies like Apple and Google, these conversations provide critical insights into customer satisfaction, product performance, and overall brand perception.
- However, manually reading and analyzing thousands of tweets is time-consuming and impractical. Automating this process through Natural Language Processing (NLP) makes it possible to quickly detect trends, identify areas for improvement, and support data-driven decision-making.
- In this project, we aim to develop a proof-of-concept sentiment analysis model to classify tweets about Apple and Google products into Positive, Negative, or Neutral sentiments. While the model will not immediately replace human analysts, it can serve as a foundation for tools that track customer opinions at scale and in real time.Why this matters:

## Stakeholders

- ➢ Apple & Google Marketing Teams will use insights to better understand public perception of their products and measure the success of campaigns.
- ➢ Product Development Teams can use the findings to understand customer pain points and and identify feature requests.
- ➢ Customer Support Teams can identify recurring complaints or issues that need immediate attention.
- ➢ Business Analysts can use the results to refine sentiment analysis methods and support data-driven decision-making.

## Objectives

- ➢ Perform exploratory data analysis (EDA) to understand sentiment distribution, brand mentions, and common text patterns.
- ➢ Preprocess tweet text by cleaning, tokenizing, removing stop words, applying lemmatization and feature engineering.
- ➢ Building an NLP model to classify sentiment in tweets directed at Apple and Google products.
- ➢ Deploy the final model for real-time predictions.

## Project Workflow:

1. Exploratory Data Analysis (EDA)

- ➢ Create visualizations (word clouds) to understand the nature of the text and most common words.
- ➢ Plot histograms to analyze the distribution of sentiment classes.

2. Data Preprocessing - Preparing the tweet text for sentiment analysis by

- ➢ Removing URLs, user mentions, hashtags, numbers, and special characters
- ➢ Perform tokenization, stop-word removal, lemmatization, and stemming on the text data.
- ➢ Encoded target variable and vectorized feature matrix for machine learning.
- ➢ Building pipeline to combine preprocessing, feature extraction, and modelling into a single reproducible process.

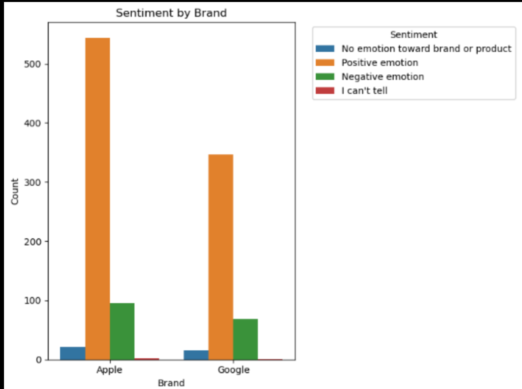
3. Modelling Selection & Deplyoment.

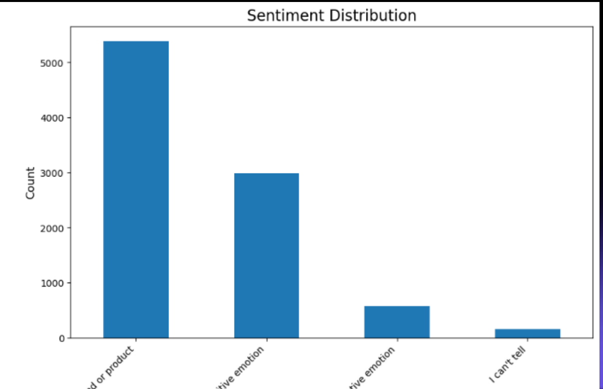
- ➢ Build an Natural Language Processing (NLP) model to analyze Twitter sentiment about Apple and Google products.
- ➢ Evaluate the model (accuracy, precision, recall, F1-score).
- ➢ Provide insights on how this model can be useful for businesses.

## Exploratory Data Analysis (EDA)

In order to better understand the dataset and prepare it for sentiment analysis, we focused on the following checks:

- -Preview the data: Inspect the first few rows to quickly grasp the dataset's structure.
- Handled any missing data that could introduce bias or cause issues during preprocessing and modelling.
- Removed duplicated tweets to prevent overrepresentation of certain entries.
- Reviewed the balance of sentiment categories, since skewed classes may bias the model toward majority classes. - Explored brand distribution and compared tweets related to Apple vs. Google to see if one brand dominates the dataset.





## Data Preprocessing:

To prepare the tweets for modelling, we applied several text-cleaning and transformation steps:

- ➢ Removed noise (URLs, hashtags, user mentions, numbers, symbols).
- ➢ Tokenized text (split into individual words).
- ➢ Removed stop words (common but unhelpful words like the, is, of).
- ➢ Lemmatized words (reduced to base form: e.g., ponies → poni).
- ➢ Vectorized text (converted words into numbers using TF-IDF).
- ➢ Encoded target labels (Positive, Negative, Neutral) into numeric values.

## Pipelines & Feature Engineering

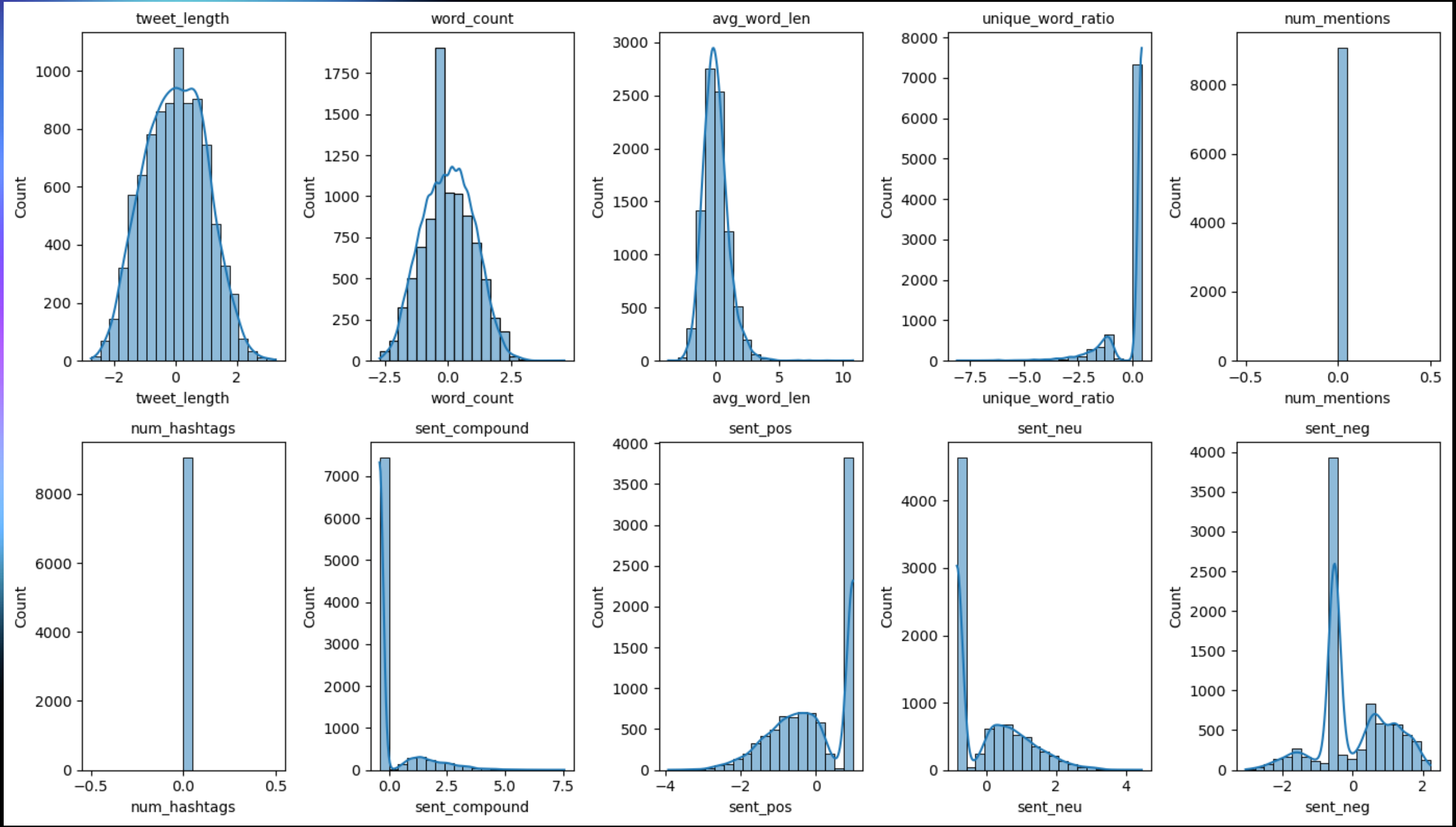
To convert textual data into machine-readable form, we:

- Captured word frequency within tweets.
- Assigned higher weight to distinctive words while reducing weight for common ones.
- Included bi-grams and tri-grams to capture short word sequences that add context beyond single words.

We also built pipelines to combine preprocessing, feature extraction, and modelling into a single reproducible process. This ensured:

- Consistency across training and testing
- Simplified experimentation with different models

# Pipelines & Feature Engineering

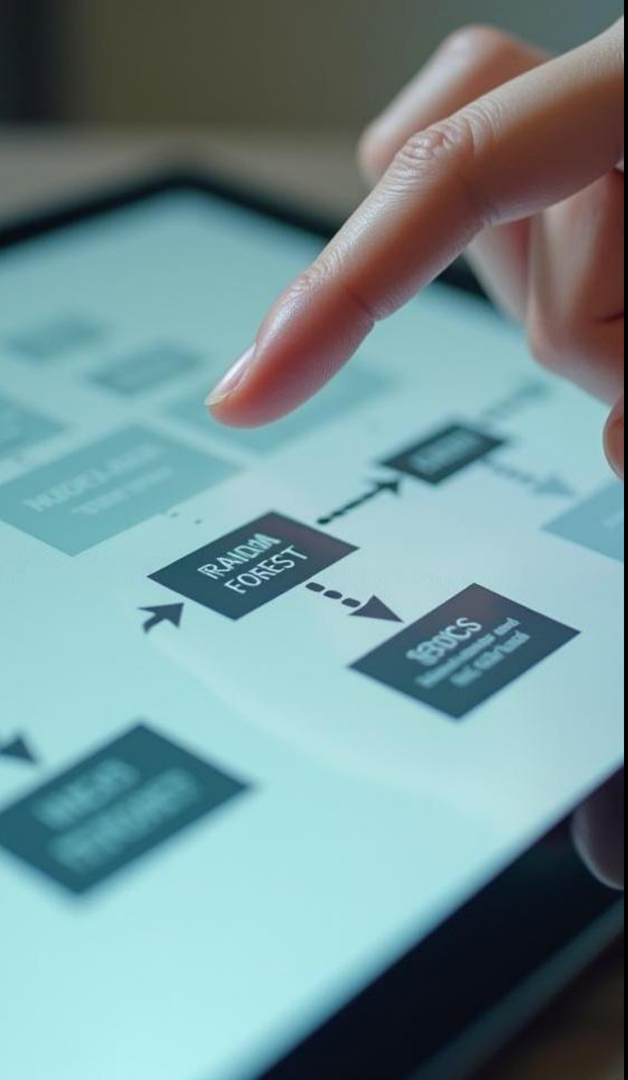


02

## Model Evaluation and Performance Insights

## Performance Metrics Analysis

- We used the tweet text as input and the sentiment label (positive, negative, neutral) as the target.
- The dataset was split into training (80%) to teach the model and testing (20%) to check performance.
- For the strongest models, we fine-tuned their settings using cross-validation to improve performance and reduce overfitting.



## Model Types and Their Characteristics

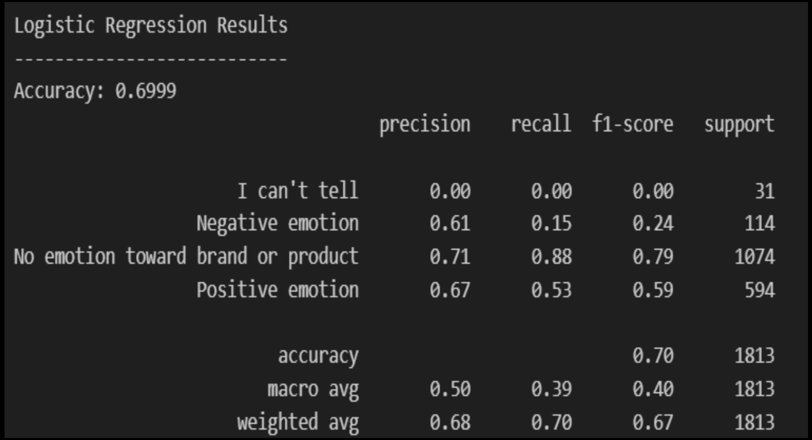
- To build a foundation for our sentiment analysis, we started with models commonly used for text classification that provide a good balance of simplicity and effectiveness.

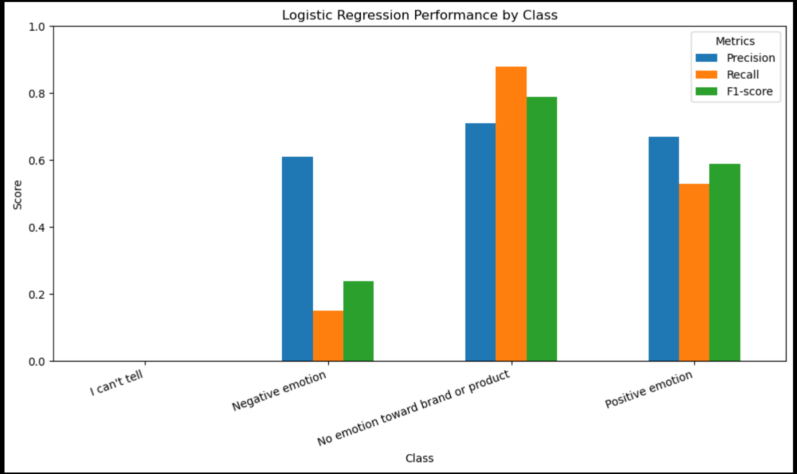
- The models we tested were:

1. Logistic Regression
2. Support Vector Machine (SVM)
3. Random Forest

- Each model was trained using the pre-processed training data and assessed on the test set, with key metrics used to provide a comprehensive evaluation of model performance, including:

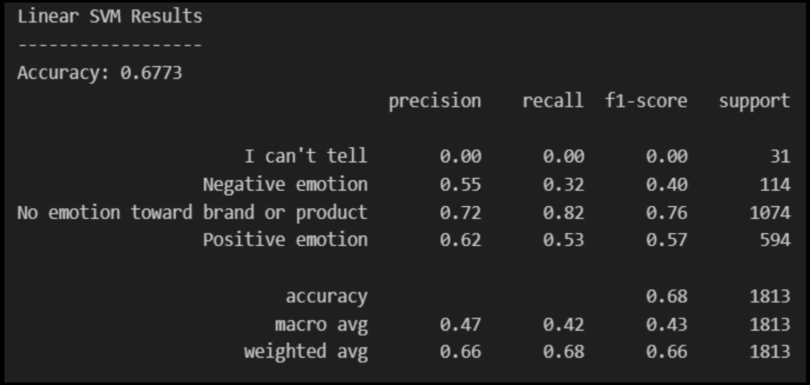
1. Accuracy
2. Precision, Recall, F1-score (per class)
3. Classification reports


```
Logistic Regression Results
--------------------------
Accuracy: 0.6999
                                  precision    recall  f1-score   support

                     I can't tell       0.00      0.00      0.00        31
                 Negative emotion       0.61      0.15      0.24       114
No emotion toward brand or product       0.71      0.88      0.79      1074
                 Positive emotion       0.67      0.53      0.59       594

                         accuracy                           0.70      1813
                        macro avg       0.50      0.39      0.40      1813
                     weighted avg       0.68      0.70      0.67      1813
```

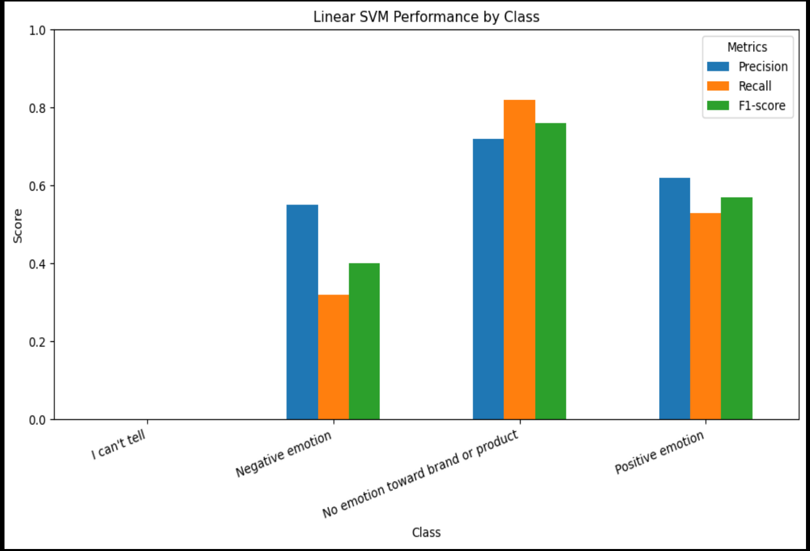


## a) Logistic Regression:

- Separates tweets into categories by finding patterns in the text.
- Estimates the likelihood that a tweet is positive, negative, or neutral.
- It was the best overall performance among the models.
- Strong on the No Emotion class (Recall: 0.88).
- Struggled with Negative and I Can’t Tell classes (low recall).
- Weighted F1-score: 0.67, showing balanced performance across classes.

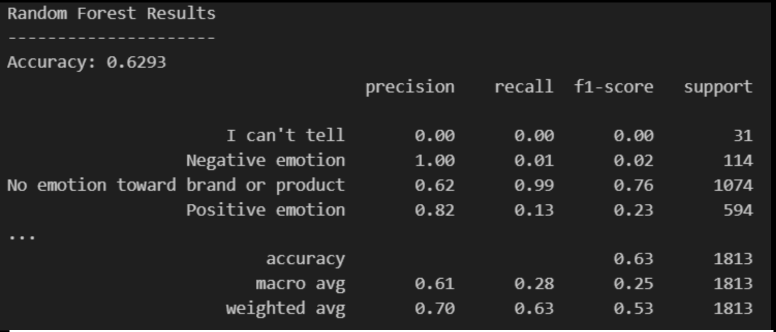


```
Linear SVM Results
------------------
Accuracy: 0.6773
                                  precision    recall  f1-score   support

                     I can't tell       0.00      0.00      0.00        31
                 Negative emotion       0.55      0.32      0.40       114
No emotion toward brand or product      0.72      0.82      0.76      1074
                 Positive emotion       0.62      0.53      0.57       594

                         accuracy                           0.68      1813
                        macro avg       0.47      0.42      0.43      1813
                     weighted avg       0.66      0.68      0.66      1813
```

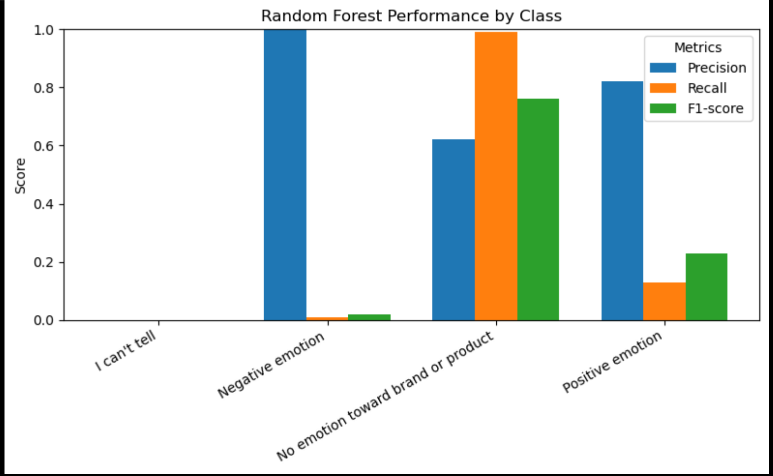


## b) Support Vector Machine

- The model finds the best dividing line to separate positive, negative, and neutral tweets.
- Accuracy: 68% (slightly below Logistic Regression).
- Strong on the No Emotion class (Recall: 0.82).
- Moderate on Positive tweets.
- Struggles to identify Negative sentiment reliably.



```
Random Forest Results
--------------------
Accuracy: 0.6293
                                  precision    recall  f1-score   support

                     I can't tell      0.00      0.00      0.00        31
                 Negative emotion      1.00      0.01      0.02       114
No emotion toward brand or product      0.62      0.99      0.76      1074
                 Positive emotion      0.82      0.13      0.23       594
...
                         accuracy                          0.63      1813
                        macro avg      0.61      0.28      0.25      1813
                     weighted avg      0.70      0.63      0.53      1813
```



## c) Random Forest

- This model combines many decision trees to make predictions.
- Each tree votes on the sentiment (happy vs. unhappy words), and the forest decides.
- Accuracy: 63% (lowest of the three models).
- Very strong on the No Emotion class (Recall: 0.99).
- Weak on both Positive and Negative tweets.
- Struggles with minority classes, heavily favors the majority class (No Emotion).

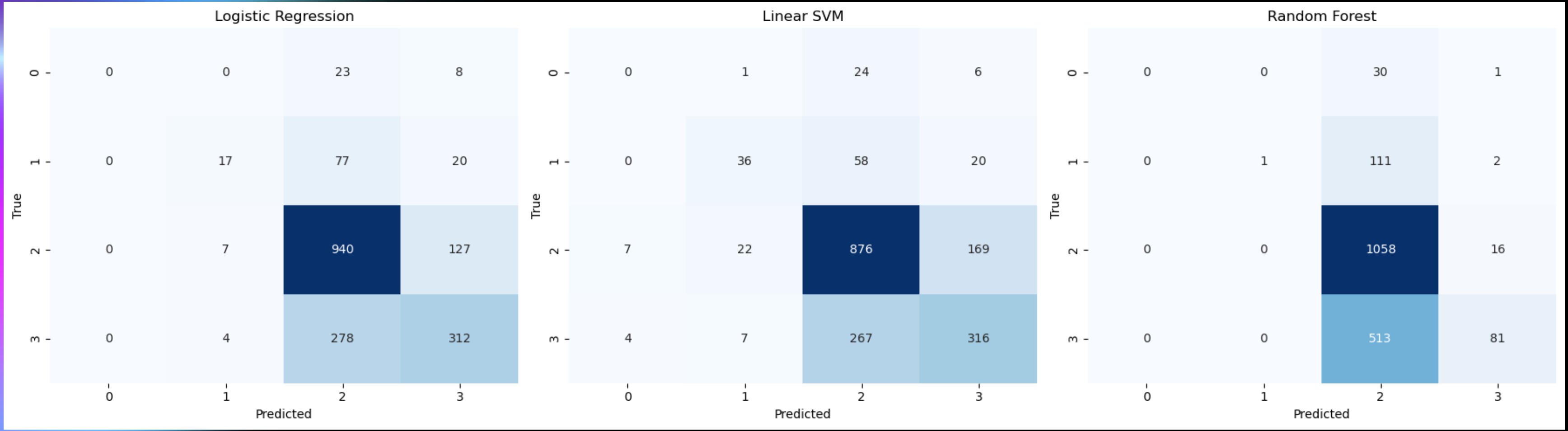
## Interpretation of Results and Insights

- Logistic Regression has the highest baseline accuracy (~70%) and balanced results across categories.
- Linear SVM performed slightly lower (~68%) but still strong and reliable.
- Random Forest struggled (63%), especially with smaller classes like Positive/Negative emotions, and heavily favored “No Emotion.”
- After tuning: Logistic Regression and SVM both stabilized at ~65% accuracy. While overall accuracy dropped slightly, recall for Negative emotions improved
- The “I can’t tell” class remained difficult for all models

## Conclusion:

- Logistic Regression is the most effective model for this dataset, but class imbalance (especially with “I can’t tell”) limits performance

## Interpretation of Results and Insights



Logistic Regression

| True \ Predicted | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 0 | 23 | 8 |
| 1 | 0 | 17 | 77 | 20 |
| 2 | 0 | 7 | 940 | 127 |
| 3 | 0 | 4 | 278 | 312 |

Linear SVM

| True \ Predicted | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 1 | 24 | 6 |
| 1 | 0 | 36 | 58 | 20 |
| 2 | 7 | 22 | 876 | 169 |
| 3 | 4 | 7 | 267 | 316 |

Random Forest

| True \ Predicted | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 0 | 30 | 1 |
| 1 | 0 | 1 | 111 | 2 |
| 2 | 0 | 0 | 1058 | 16 |
| 3 | 0 | 0 | 513 | 81 |

## Model Deployment

We deployed our sentiment analysis model so it can be used outside the notebook environment and accessed by different types of users.

- ➢ The trained model pipeline (TF-IDF + classifier + label encoder) was saved in a reusable format to ensure predictions remain consistent every time.
- ➢ The model logic was kept separate from the serving layer, making the system easier to manage and update.

We provided two ways to interact with the model:

- ➢ Flask Web App→ website where users can enter text and immediately see the sentiment prediction. This was designed for demonstrations and quick reviews.
- ➢ FastAPI → an API that allows developers to send text to the model and get results programmatically. It includes automatic documentation, making integration straightforward.

The model is hosted on Render which is connected to our GitHub repository. This means that whenever we push updates to GitHub, Render automatically rebuilds and redeploys the service. This avoids manual server setup and ensures the service is always up to date.

This setup allows:

- ➢ End-users → to interact with the model through the web interface.
- ➢ Developers → to integrate the model into their applications via the API.
- ➢ Future scaling → the system can grow as demand increases.

## Recommendations:

- The model can be applied to track customer emotions and opinions towards the company's products helping to capture market sentiment in real time.
- By connecting the model to Twitter's APIs the company can automatically filter tweets mentioning the brand or competitors and classify them into sentiment categories.
- Positive sentiment can be amplified in marketing campaigns to highlight brand strengths while the negative sentiment should be flagged for further analysis, enabling the company to identify pain points and potential areas for product or service improvement.
- Insights from competitor related tweets can help the company understand what customers value in rival products and adapt strategies accordingly.

## Future Improvement:

- Explore more advanced Modelling approaches like BERT for richer text representations beyond TF-IDF.
- Implement a feedback loop to retrain the model with newly collected tweets, ensuring it adapts to evolving language such as slang, new product references, and emojis.