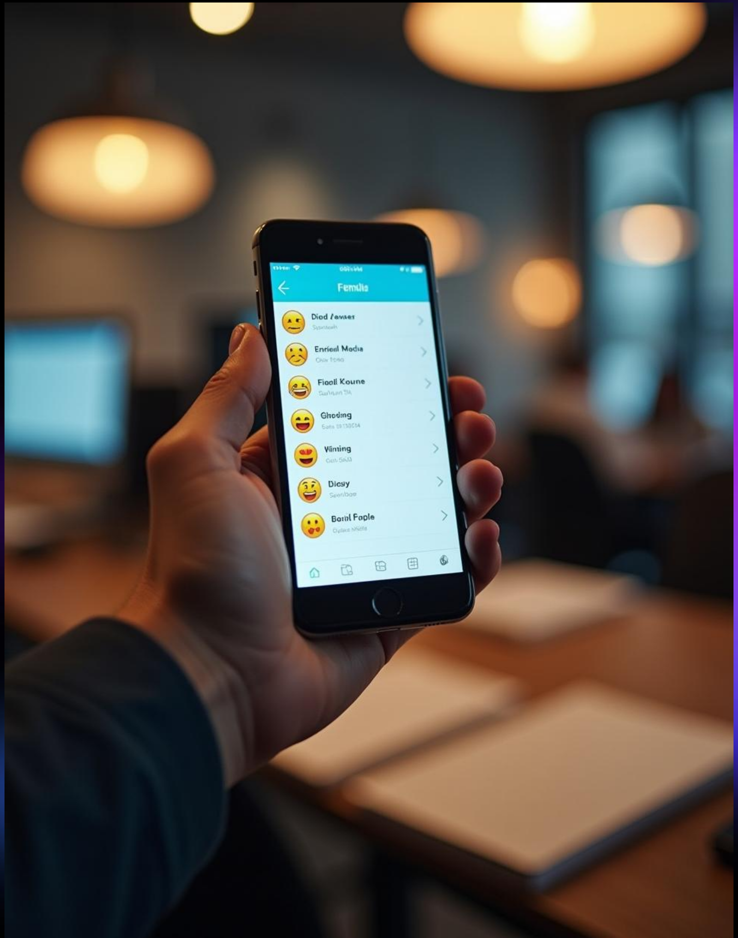
Sentiment Analysis:

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.

Project Workflow:

- The dataset used in this project is from CrowdFlower and contains over 9,000 tweets, each labelled by human raters as expressing a positive, negative, or neutral sentiment.

- We examined the dataset to understand its structure and quality before modelling:

a) Checked for missing values and cleaned them to avoid bias.

b) Removed duplicate tweets to prevent repetition in results.

c) Analysed the class balance of sentiment categories to see if there was imbalance

d) To assess brand representation, we Compared Apple vs. Google tweet volumes

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:

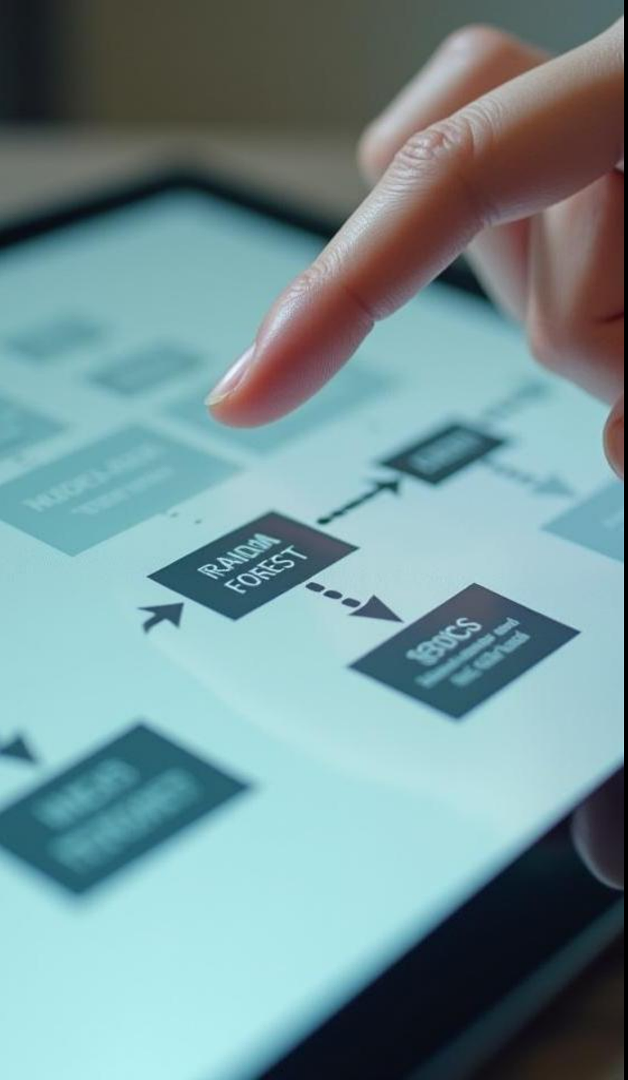
- Consistency across training and testing
- Simplified experimentation with different models

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

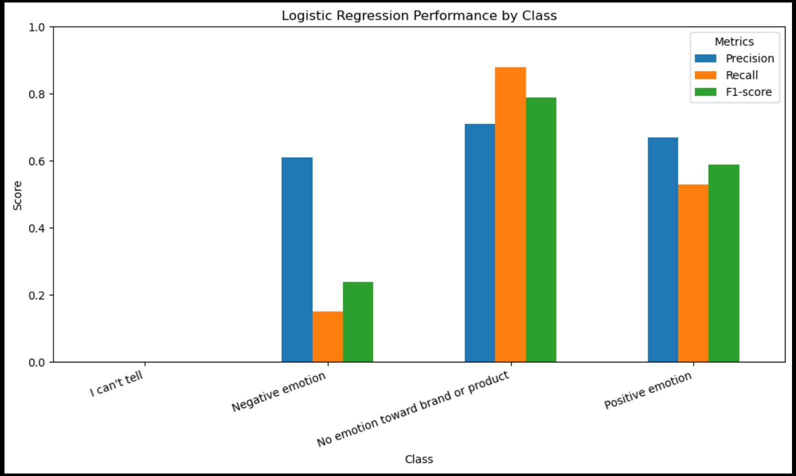
- Each model was trained using the pre-processed training data and assessed on the test set, with key metrics used to provide a comprehensive evaluation of model performance, including:

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
```



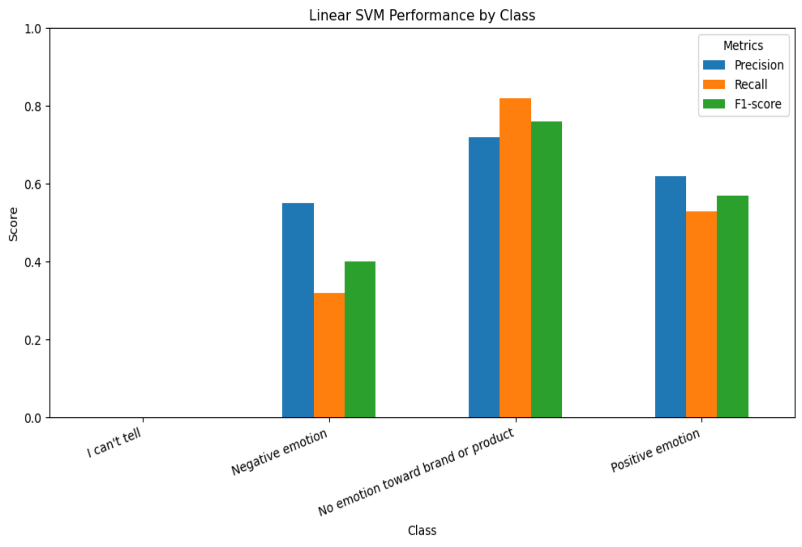
a) Logistic Regression:

- 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

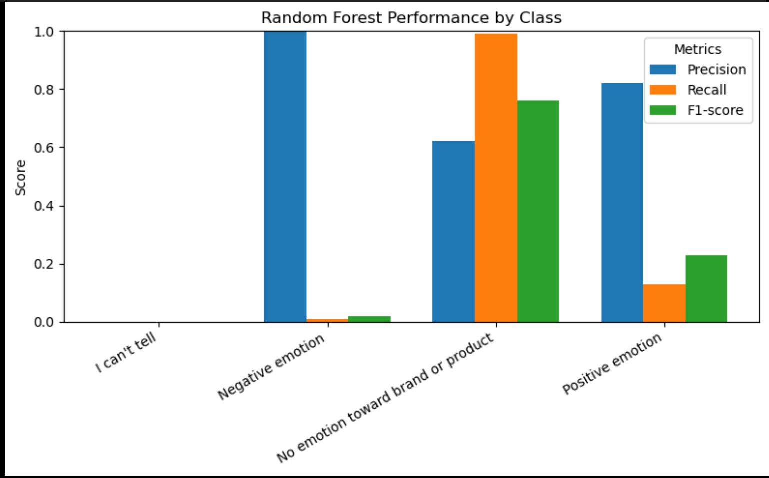


b) Support Vector Machine

- 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 favoured “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

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.