**Twitter Sentiment Analysis using Natural Language Process (NLP).**

**Table of content**

1. **INTRODUCTION**
   1. Business Overview
   2. Problem statement
   3. Business objectives
   4. Success Criteria
2. **DATA**
   1. Data source
   2. Data Description
   3. Data Abnormalities
3. **ANALYSIS**
   1. Data Preparation
      1. Data Cleaning
      2. Text Preprocessing
      3. Feature Engineering
   2. Exploratory Data Analysis
      1. Univariate analysis
      2. Bivariate analysis
      3. Multivariate analysis
   3. Natural Language Process (NLP)Models
      1. Naïve Bayes, Logistics Regression, Supervised Vector Machine (SVM), XgBoost.
   4. Model Evaluation
4. **RESULTS & CONCLUSIONS**
   1. Findings
   2. Recommendations
   3. Conclusions

**INTRODUCTION**

**Business Overview**:

Sentiment analysis, also known as opinion mining, is a Natural Language Processing (NLP) technique used to automatically determine the emotional tone behind textual content. Companies leverage sentiment analysis of tweets to get a sense of how customers are talking about their products and services, get insights to drive business decisions and identify product issues and potential PR crises early on.

A 2023 systematic review published in [Information Processing & Management](https://www.sciencedirect.com/science/article/abs/pii/S0306457323001413?via%3Dihub) confirms that social media analytics have become a cornerstone of business intelligence, directly enabling data-driven decisions in product innovation and competitive positioning. This project applies sentiment analysis to tweets about Apple and Google products, aiming to classify and analyze public opinion. The insights will help technology companies identify emerging issues, leverage positive feedback, and make strategic decisions based on real-time customer sentiment.

**Problem Statement**

Apple and Google face the challenge of managing customer opinions expressed on social media. Negative sentiment about products or services can quickly harm brand perception, reduce customer loyalty, and impact sales. Without an automated solution, emerging product issues or trending complaints may go unnoticed, leading to reputational risks and missed market opportunities. This project aims to address this problem by using Natural Language Processing (NLP) to classify and analyze tweets, enabling Apple and Google to respond promptly, leverage positive feedback, and make data-driven strategic decisions.

**Business objectives**

The main objective

The objective of this project is to build an NLP model that can rate the sentiment of a Tweet based on its content.

Specific objectives

1. Develop machine learning models to classify tweets as positive or negative.
2. To clean and preprocess raw Twitter text data for analysis.
3. To explore and visualize sentiment distribution between Apple and Google products
4. To evaluate model performance using appropriate metrics such as accuracy and precision to ensure reliability.

**Success Criteria**.

The machine learning model should accurately classify tweets as positive, negative, or neutral, achieving a targetaccuracy of at least 75–80% on test data.

The model also demonstrates strong precision, recall, and F1-scores across all sentiment categories, ensuring balanced performance.

The model’s results should generalize well without overfitting and provide insights that are interpretable and actionable for understanding public sentiment.

**DATA**

**Data Source**

The dataset for this project comes from [CrowdFlower via data.world](https://data.world/crowdflower/brands-and-product-emotions) platform. Each record includes tweets related to Apple and google products and sentiment emotions towards the products offered by the companies in either positive, negative, or neutral.

**Data Description**

Most tweets are text-based and include user mentions, hashtags, and product names. It is designed to study and evaluate how Apple and Google users perceive their products through sentiment analysis of Twitter posts using NLP methods.

The dataset contains a total of 9,093 records and 3 features, and all features were object data types.

Key Features in the Dataset:

1. ***tweet\_text*** - The actual content of the tweet as written by the user. This serves as the main input for Natural Language Processing (NLP) to determine the expressed sentiment.
2. ***emotion\_in\_tweet\_is\_directed\_at***- Specifies the brand, company or product that the emotion is directed at (e.g., Apple, Google, iPhone, Android). This helps in comparing sentiment between brands.
3. ***is\_there\_an\_emotion\_directed\_at\_a\_brand\_or\_product***- The target variable indicating whether a tweet expresses a positive, negative, or neutral emotion toward a brand or product.

**Data Abnormalities**

The dataset exhibited a total of 5,803 missing values. Specifically, the *“emotion\_in\_tweet\_is\_directed\_at”* column accounted for 5,802 of these missing entries, while the *“tweet\_text”* column had one missing value. Furthermore, 22 duplicate records were detected within the dataset, indicating the need for data cleaning prior to analysis.

**ANALYSIS**

**Data Preparations**

Data Cleaning**:**

Since the *“emotion\_in\_tweet\_is\_directed\_at”* column contained 5,801 missing values, removing all of these rows would have eliminated nearly half of the dataset, resulting in fewer examples for the model to learn from. To preserve data volume, the missing values were therefore replaced with the category “Unknown.”

During the data inspection, duplicate records were also identified. Prior to their removal, the dataset consisted of 9,092 records and 3 features. After eliminating the duplicates, the dataset was reduced to 9,070 records while maintaining the same number of features, ensuring that each tweet in the dataset is unique.

Text Preprocessing**:**

In the initial stage, both brand and sentiment mappings were performed to structure the dataset. Brand and product names were categorized into three groups: *Apple*, *Google*, and *Unknown*. Similarly, sentiments were mapped into three classes: *positive*, *negative*, and *neutral*.

After the mapping process, text preprocessing was conducted to standardize and clean the dataset. All **texts were converted to lowercase** to ensure uniformity, and non-word elements such as URLs, user mentions, hashtags, special characters, and retweet tags were removed.

**Tokenization** was then performed using the Word Tokenizer to split texts into individual words, followed by the **removal of stopwords** to eliminate non-informative terms. **Lemmatization** was applied using the WordNet Lemmatizer to reduce words to their base forms, thereby enhancing semantic consistency across the corpus.

The cleaned texts were subsequently stored in a new column named *cleaned\_tweet* for vectorization and model training. This preprocessing pipeline improved model efficiency and interpretability by ensuring that only meaningful and standardized words were used as input features.

Feature engineering:

New features were engineered to enhance model understanding. Brand mentions were grouped into a *brand\_category* column like **Apple**, **Google**, or **Unknown**, while sentiments were mapped into a *sentiment* column as **positive**, **negative**, or **neutral**. The *sentiment* column, serving as the target variable, was labeled as **0 = negative**, **1 = positive**, and **2 = neutral**. A word frequency distribution was also conducted to identify the most common terms, with *“ipad,” “google,” “apple,”* and *“iphone”* appearing most frequently.

**Exploratory Data Analysis**

**Univariate analysis**

Distribution of the Target Variable (Sentiment):

* The bar chart shows that most tweets are neutral, followed by positive and negative sentiments.  
  This indicates a **class imbalance**, mainly caused by the many missing values categorized as neutral during preprocessing.

Distribution of tweet lengths:

* The distribution of tweet lengths is slightly right-skewed, showing that most tweets are short (about 60 – 85 characters). This suggests that users usually express their opinions briefly with a few, writing longer tweets to explain their experiences.

Distribution of brand categories:

* The bar chart shows that most tweets are labeled as Unknown, followed by Apple and Google.  
  This reflects the large number of missing values (about 5,801) in the brand category column, where tweets without clear brand mentions were classified as *Unknown*, while Apple was discussed more often than Google.

**Bivariate Analysis**

Sentiments vs brand category:

* The combined bar graph indicates that the Apple brand has more positive tweets than the rest of the categories. More so, the Unknown category has a high number of tweets with no category mentioned thus the neutral bar is high.

Sentiments vs tweet length:

* The box plots indicate that the median tweet length is similar across sentiment categories.

Brand category vs tweet length:

* Google tweets have a higher median tweet length compared to Apple and the unknown category.

**Multivariate Analysis**

Sentiment and brand category by tweet length:

* Positive tweets about Apple tend to be slightly longer than negative or positive neutral ones. Google tweets show similar patterns, but differences are minor.

**Natural language process (NLP)Models**

**Naïve Bayes model:**

The Navies Bayes is one of the most efficient models for text classification, which predicts the probability of a class assuming all features are independent. A pipeline combining **TF-IDF**, **SMOTE**, and **Multinomial Naïve Bayes** achieved **79% accuracy**.  
It performed well on **positive tweets** (precision = 0.93, F1 = 0.87) but struggled with **negative tweets** (precision = 0.42, F1 = 0.52) due to class imbalance.  
Overall, it shows good sentiment prediction with room for improvement in minority classes. Despite the class imbalance the model provided a good starting point for sentiment analysis, offering quick training time, easy implementation, and good accuracy

**Logistic Regression model**:

Logistic Regression, a supervised algorithm effective for high-dimensional text data, was trained using **TF-IDF features**.  
The model **achieved 82% accuracy**, with strong performance **on positive tweets** (precision = 0.93, recall = 0.85, F1 = 0.89) and moderate results on **negative tweets** (precision = 0.46, recall = 0.66, F1 = 0.52). This leads to improved predictions, particularly negative sentiments.  
Overall, it demonstrated balanced and reliable sentiment classification with reduced overfitting and good generalization**.**

**SVM (Supervised vector Machine):**

The **SVM model**, a robust supervised learning algorithm for text classification, performed effectively with **TF-IDF-generated features** and achieved an **accuracy of 82%**. It maintained strong performance on **positive tweets** (precision = 0.92, recall = 0.86, F1 = 0.89), showing high consistency in detecting positive sentiments.  
For **negative tweets**, the model achieved a (precision of 0.46, recall of 0.62, and F1-score of 0.53), indicating moderate performance but some difficulty in identifying minority classes.  
Overall, the SVM handled the class imbalance reasonably well and demonstrated stable, reliable results for sentiment analysis

**XGBoost model**:

The **XGBoost model**, a powerful gradient boosting algorithm, was trained using **TF-IDF features** and **SMOTE** to balance the sentiment classes.  
It achieved an **accuracy of 85%**, mainly influenced by its strong performance on **positive tweets** (precision = 0.88, recall = 0.95, F1 = 0.92).  
However, it struggled with **negative tweets** (precision = 0.57, recall = 0.35, F1 = 0.43), showing limited ability to interpret negative sentiments.  
Overall, despite its high accuracy, the model’s performance was skewed toward positive sentiments, indicating the need for better handling of minority classes.

**Model Evaluation**

| **Model** | **Accuracy** | **F1 (Weighted)** | **F1 (Negative)** | **F1 (Positive)** |
| --- | --- | --- | --- | --- |
| Naïve Bayes | 0.7938 | 0.8128 | 0.5229 | 0.8685 |
| Logistic Regression | 0.8220 | 0.8338 | 0.5435 | 0.8895 |
| SVM | 0.8234 | 0.8333 | 0.5318 | 0.8912 |
| XGBoost | 0.8531 | 0.8382 | 0.4348 | 0.9156 |

After training, each model was evaluated on the test dataset to measure its ability to generalize and accurately classify unseen tweets into positive or negative sentiments.  
The evaluation metrics included **accuracy** and **F1-scores** for both positive and negative classes, which together provide a balanced understanding of precision and recall performance.

**Model Comparison Using Metrics**

From the comparison, **XGBoost** achieved the highest overall **accuracy (85%)** and the best **F1-score for positive sentiments (0.91)**, indicating excellent performance in identifying positive tweets. This suggests that XGBoost effectively captured the dominant sentiment patterns in the dataset.

However, the model showed weaker results for **negative sentiments** (F1 = 0.43), meaning it often misclassified negative tweets as positive or neutral. This imbalance is likely due to the smaller number of negative samples in the dataset.

Both **Logistic Regression** and **SVM** provided more balanced performance, maintaining high accuracy (around 82%) while achieving relatively better scores on negative tweets (F1 ≈ 0.53).  
Meanwhile, **Naïve Bayes** demonstrated solid generalization but slightly lower precision across both classes.

Overall, while **XGBoost** emerged as the top-performing model in terms of accuracy, **Logistic Regression** and **SVM** handled class imbalance more evenly, making them valuable alternatives when balanced sentiment detection is required.

**Hyperparameter tuning of logistic regression**

Hyperparameter tuning was performed using **GridSearchCV** to improve model performance by optimizing both **TF-IDF** and **Logistic Regression** parameters.

**Tuned Parameters:**

* **TF-IDF:** max\_features, ngram\_range, min\_df, max\_df
* **Logistic Regression:** C, penalty, solver, class\_weight

The best combination was found as:  
C = 30.0, penalty = 'l2', solver = 'liblinear', class\_weight = 'balanced',  
max\_df = 0.8, max\_features = 2500, min\_df = 0.001, ngram\_range = (1, 2).

After tuning, the model achieved **84% accuracy** and an improved **weighted F1-score of 0.84**.  
The **positive class F1-score** slightly increased to **0.90**, and the **negative class F1-score** improved to **0.55**, showing better handling of class imbalance.  
Overall, the tuned model made more balanced predictions and reduced bias toward the majority class.

**Feature Importance**

The feature importance plot highlights the most influential words contributing to positive and negative tweet classifications.  
For **positive sentiments (Class 1)**, words such as *winning*, *free*, *ipad iphone*, *new android*, *awesome*, *fun*, *smart*, and *want ipad* had the highest positive coefficients, showing strong association with positive opinions.  
On the other hand, **negative sentiments (Class 0)** were driven by words like *fail*, *hate*, *long*, *launched*, *issue*, *sense*, *bit*, *ipad wait*, and *crash*, which indicate dissatisfaction or product-related problems.  
This analysis helps interpret how specific keywords influence sentiment prediction in the Logistic Regression model.

**Deep learning**

Although Logistic Regression achieved the best performance among traditional machine learning models, it is a linear classifier that may not capture complex, non-linear relationships in text data. Therefore, two neural network architectures were developed using TensorFlow and Keras to investigate whether deep learning could improve generalization and capture more intricate sentiment patterns.

Model 1: Simple Neural Network

| **Metric / Class** | **Precision** | **Recall** | **F1-Score** | **Support** |
| --- | --- | --- | --- | --- |
| **Negative (0)** | 0.52 | 0.47 | 0.50 | 114 |
| **Positive (1)** | 0.90 | 0.92 | 0.91 | 594 |
| **Accuracy** |  |  | 0.84 | 708 |
| **Macro Avg** | 0.71 | 0.69 | 0.70 | 708 |
| **Weighted Avg** | 0.84 | 0.84 | 0.84 | 708 |

The first neural network consisted of two hidden layers with 16 neurons each, activated by the ReLU function, and a sigmoid output layer for binary classification. The model used the Adam optimizer (learning rate = 0.001) and Binary Cross entropy as the loss function. To address class imbalance, class weights were applied (Negative class: 3.11, Positive class: 0.59). The model was trained for 50 epochs with a batch size of 32.

* Validation accuracy: 84.46%
* ROC-AUC score: 0.8238
* Precision & Recall: The model achieved strong recall for positive tweets (0.92) but struggled with negatives (recall = 0.47), indicating it favored the majority (positive) class.

This shows that while the model effectively recognized positive sentiments, it was less sensitive to negative expressions, similar to earlier traditional classifiers.

Model 2: Deeper Neural Network

| **Metric / Class** | **Precision** | **Recall** | **F1-Score** | **Support** |
| --- | --- | --- | --- | --- |
| **Negative (0)** | 0.49 | 0.59 | 0.53 | 114 |
| **Positive (1)** | 0.92 | 0.88 | 0.90 | 594 |
| **Accuracy** |  |  | **0.83** | 708 |
| **Macro Avg** | 0.70 | 0.73 | 0.72 | 708 |
| **Weighted Avg** | 0.85 | 0.83 | 0.84 | 708 |

To enhance learning capacity, a second model was designed with a deeper architecture—two hidden layers of 64 neurons each, also using ReLU activations and the same optimization settings. It was trained for 20 epochs, using the same TF-IDF input and class weighting strategy.

* Validation accuracy: 83%
* ROC-AUC score: 0.8364
* Precision & Recall: The deeper model showed a slight improvement in distinguishing negative tweets (recall = 0.59) but still displayed class imbalance, with positive recall remaining higher (0.88).

Comparison with Logistic Regression

Both neural networks achieved comparable accuracy to Logistic Regression (≈84%) but did not significantly outperform it. Moreover, the Logistic Regression model captured 61% of minority (negative) class samples, slightly better than the deeper Keras model’s 59%. This suggests that, given the limited dataset and TF-IDF representation, deep learning models did not provide a substantial performance gain over simpler linear methods.

We also did some testing on our own hand written data using logistic regression and the model did not seem to do badly off.

**RESULTS AND CONCLUSIONS**

**Findings**

All models performed well on positive sentiment detection, but **further improvement is needed for negative sentiment classification**, suggesting the need for more balanced data or advanced sampling methods.

Having the data imbalanced at most on the brand category and sentiment columns, the models had a difficult time to balance predictions between the negative sentiment and positive sentiments.

Using **TF-IDF with bigrams (1,2)** improved the representation of contextual meaning, enabling models like Logistic Regression and SVM to achieve consistent F1 scores above **0.83**.

The most influential **positive words** were *love, best, amazing, great*, while **negative words** included *bad, worst, hate, disappointed* — reflecting clear sentiment polarity captured by the models

**Conclusions**

After evaluating multiple models for sentiment analysis on tweets, Logistic Regression emerged as the best performing model. It achieved high accuracy, maintained a good balance between precision and recall, and generalized better to unseen real-world tweets compared to the deeper Keras neural network, which tended to predict the majority class.

Even with a slightly higher misclassification count, it handles both classes better, is more stable, and is easier to deploy real-world tweets. While some misclassifications occurred due to short, ambiguous, or slang-heavy tweets, Logistic Regression remains a reliable and interpretable choice for this dataset.

**Recommendations**

* Use Logistic Regression for deployment: Its simplicity, interpretability, and strong generalization make it ideal for production sentiment analysis on tweets.
* Consider feature improvements: Incorporating word embeddings (like Word2Vec or GloVe) or additional preprocessing to handle slang and abbreviations could reduce misclassifications.
* Monitor new data: Since social media language evolves rapidly, periodic retraining or updating the model with new tweets is recommended.

**Report By:**

**Royal Mbugua**

**Marion Mengech**

**Grace Wangui**

**Vincent Torotich**

**Zipporah Muriithi**