SENTIMENT ANALYSIS OF SOCIAL MEDIA POSTS ABOUT APPLE AND GOOGLE PRODUCTS

JOY GITAU, DERRICK WAITITU, KNIGHT MBITHE, LEVIS KIMANI, COLLINS KWATA

24 January, 2025

TABLE OF CONTENTS

Contents

Business Understanding

Business Overview

Social media platforms have become key spaces for consumers to express opinions about products, brands, and services. Companies like Apple and Google are frequently mentioned in posts, where users share thoughts on their products and updates, generating valuable data on public perception.

However, extracting insights from this massive volume of unstructured data is a challenge. Traditional sentiment analysis methods struggle with the nuances of slang, sarcasm, and ambiguous language, making it difficult for companies to understand true sentiment.

A potential solution is building a Natural Language Processing (NLP) model to automatically classify the sentiment of posts related to Apple and Google products. This would allow businesses to assess sentiment in real-time and improve decision-making.

By leveraging NLP, Apple and Google could gain actionable insights into customer sentiment, leading to better product development and more effective marketing strategies.

Problem Statement

Consumers frequently share opinions about Apple and Google products on social media, creating a vast pool of unstructured data. However, due to the sheer volume of posts, analyzing these opinions manually is time-consuming and impractical.

Sentiment analysis on this data is challenging because social media posts often contain slang, sarcasm, and ambiguous language, making it difficult to determine public sentiment accurately. Without an automated system, companies like Apple and Google struggle to gauge consumer sentiment efficiently and make informed decisions based on real-time public opinion.

Objectives

Main Objective

- To develop a Natural Language Processing (NLP) model that can accurately classify the sentiment of social media posts related to Apple and Google products into positive, negative, and neutral categories, providing businesses with actionable insights to guide marketing strategies, product development, and customer engagement.

Specific Objectives

1) To examine the distribution of sentiment labels and individual variables in the dataset:
   - Understand the proportions of sentiment categories, identify key features in the text data, and ensure the dataset is balanced and ready for modeling.

2) To identify relationships between variables in the dataset:
   - Explore how text features like word usage or length relate to sentiment labels and uncover patterns that inform model design.

3) To build and evaluate baseline sentiment classification models:
   - Develop both binary and multiclass classifiers using Logistic Regression to establish a performance baseline for sentiment analysis.

4) To enhance sentiment classification models with advanced techniques:
   - Improve the performance of both binary and multiclass classifiers by implementing Support Vector Machines (SVM), Randomforest Classifier, XGBoost Classifier, and BERT Transformer.

Success Criteria

- Accuracy: The overall percentage of correctly classified Tweets (target: 80%).
- Precision: The proportion of true positive sentiment predictions for each class (positive, negative, neutral) (target: 75% for each class).
- Recall: The proportion of actual sentiments correctly predicted (target: 75% for each class).
- F1 Score: A balanced measure of precision and recall (target: 75% for each class).
- Confusion Matrix: Minimize misclassifications across sentiment classes.

Data Understanding

The dataset comes from CrowdFlower via [data.world]. The dataset consists of over 9,000 Tweets about Apple and Google products. Key columns include tweet_text, the content of the Tweet; emotion_in_tweet_is_directed_at, which identifies the targeted product or brand; and is_there_an_emotion_directed_at_a_brand_or_product, indicating the sentiment toward the brand.

The DataFrame contains 3 columns whose datatypes are all objects. While the tweet_text and is_there_an_emotion_directed_at_a_brand_or_product columns have no missing values, the emotion_in_tweet_is_directed_at column has 3,291 non-null entries, indicating a significant portion of missing data.

This dataset contains 9093 entries and 3 columns. Each entry represents recorded sentiments from users.

The DataFrame contains mostly unique tweet_text entries, with the most frequent being retweeted 5 times. The emotion_in_tweet_is_directed_at column is dominated by "iPad" (946 occurrences), and the majority of tweets (5,389) indicate "No emotion toward brand or product."

Data Preparation

Data Cleaning

The data cleaning and preprocessing steps involved:

a) Shortening column names for easier reference.

b) Handling missing values **by f**illing missing values in the brand_product column based on the tweet's content and dropping rows with missing values. Since the 'brand_product' column is crucial for the analysis and cannot be dropped, we will fill its missing values with relevant categories based on the content of the 'tweet' column.

c) Checking and removing duplicates to avoid skewed results.

d) Merging product categories into Apple and Google brands for a broader sentiment analysis. Since the `brand_product` column comprises products under either Apple or Google as a brand, merging all Apple products and Google products, respectively, is a necessary step to determine the customer base reactions towards the brands. We will do this by creating a `brand_category` column

e) Sorting the emotion column into positive, negative, and neutral categories.

Text Preprocessing

a) Text cleaning: Removing unwanted characters like URLs, mentions, hashtags, and special characters.

b) Text tokenization: Breaking down the cleaned text into individual words or tokens.

c) Lowercasing: Converting all tokens to lowercase for consistency.

d) Removing stop words: Removing common words that don't contribute significant meaning (e.g., "the," "is," "and").

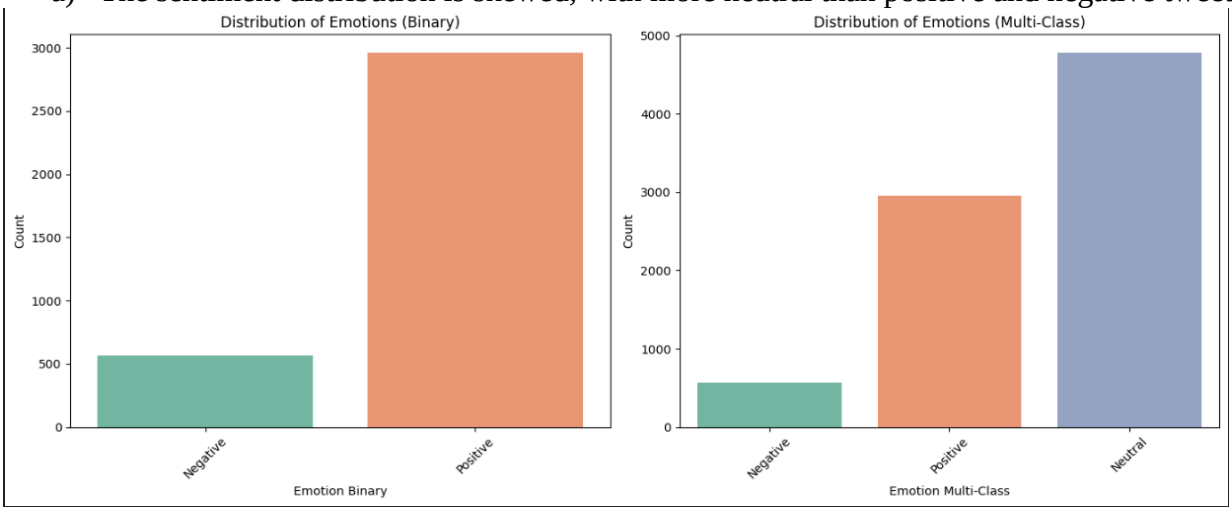
e) Lemmatization: Reducing words to their base form to improve accuracy.

f) Vectorization: Converting the cleaned text into numerical features using TF-IDF for the machine-learning model.

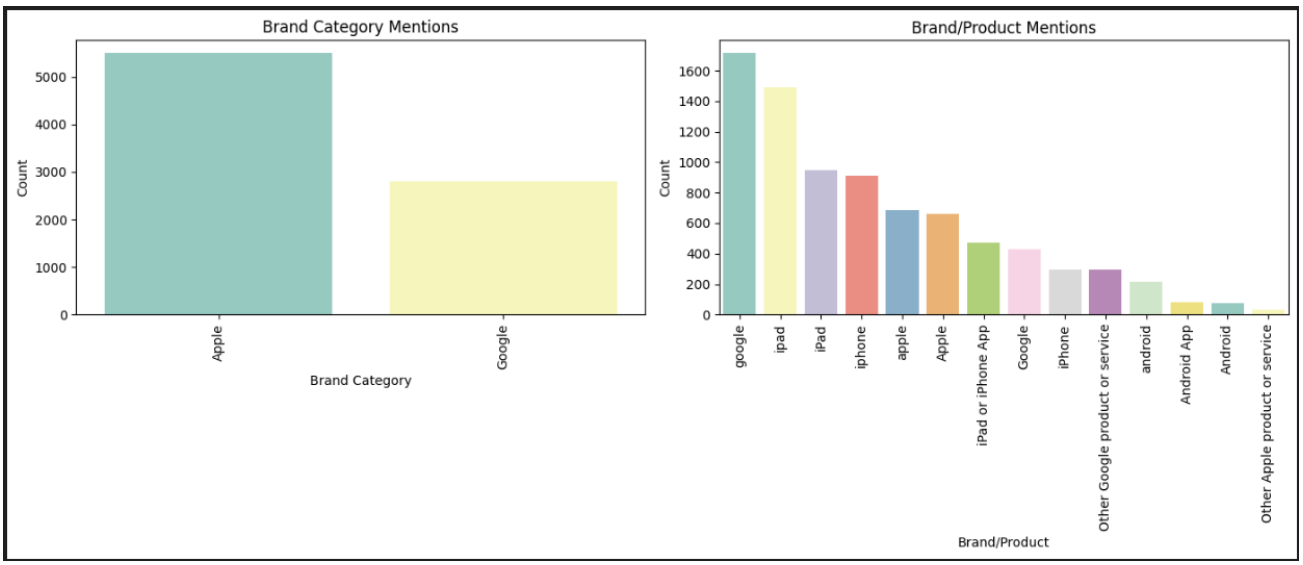
Exploratory Data Analysis

The EDA phase involved visualizing the distribution of sentiment, brand mentions, tweet length, and word frequency to understand the characteristics of the data. Here are some key findings:

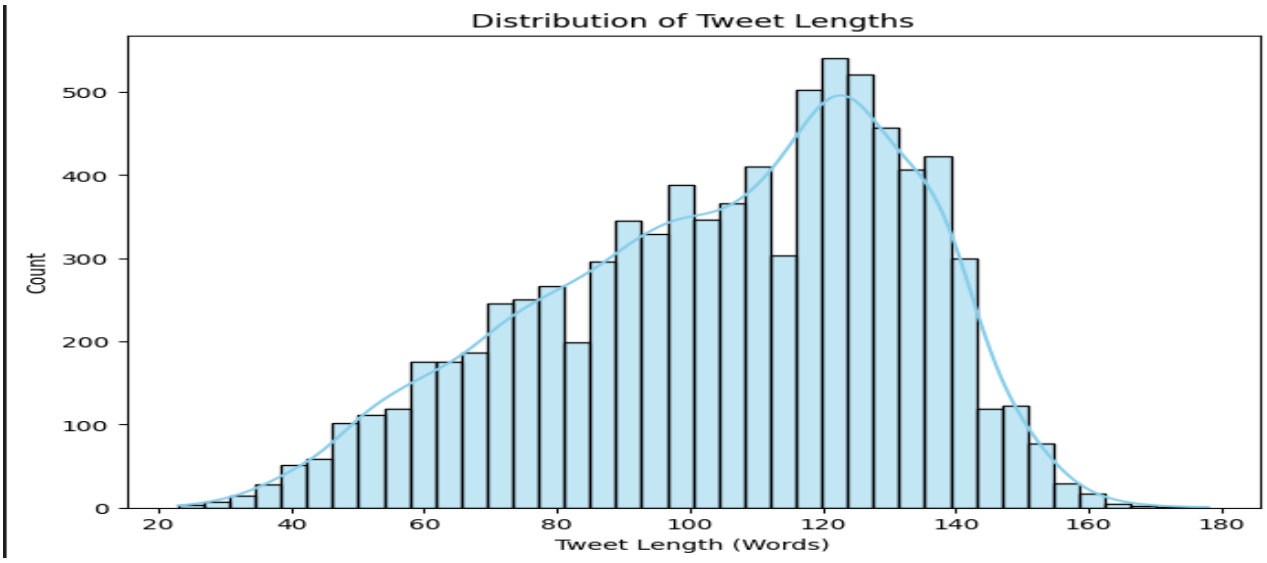
a) The sentiment distribution is skewed, with more neutral than positive and negative tweets.



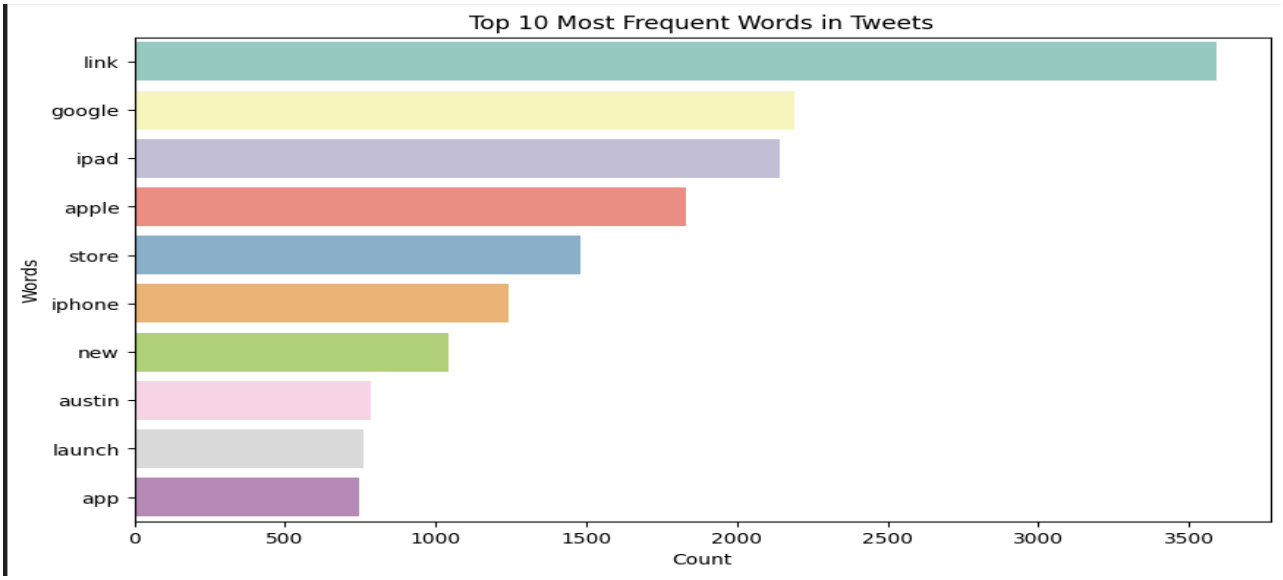
b) Apple products receive a higher volume of mentions than Google products.



c) The tweet length distribution peaks around 120-130 words.



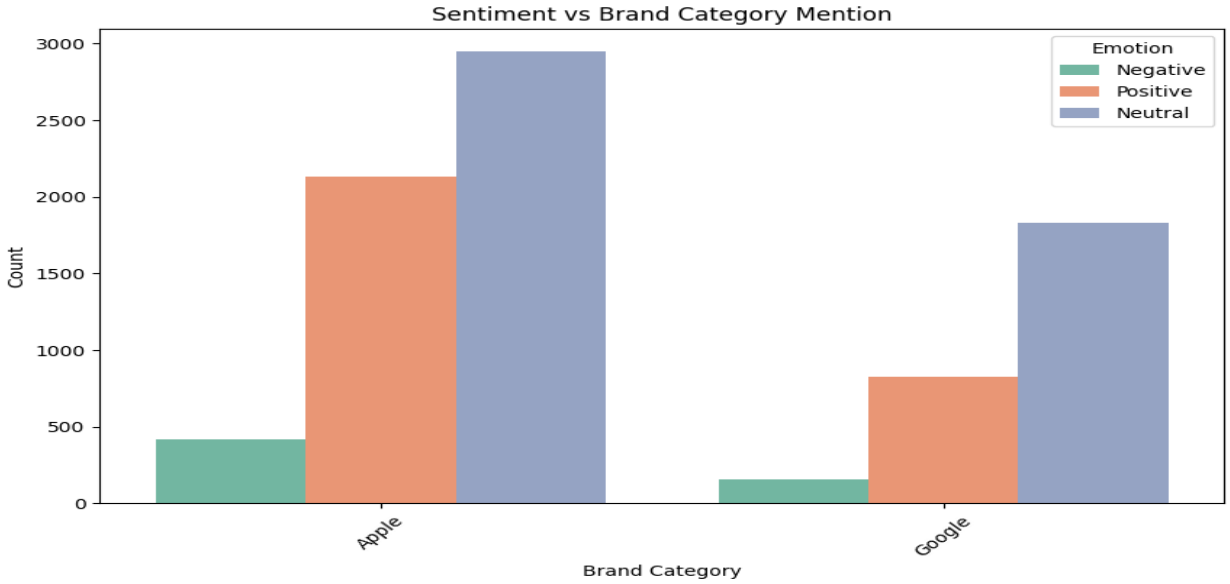
d) "Link," "Google," "iPad," "Apple," and "store" are among the most frequently used words.



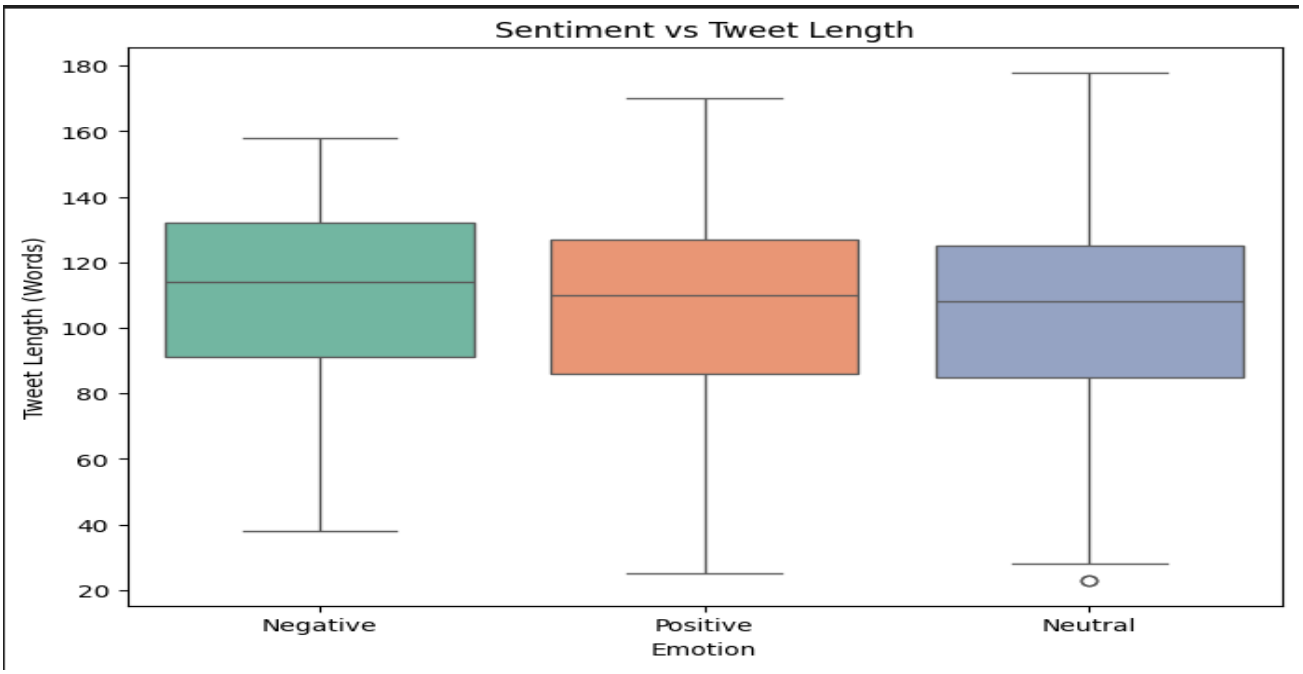
Bivariate and Multivariate Analysis

We further analyzed the relationships between sentiment and other variables:

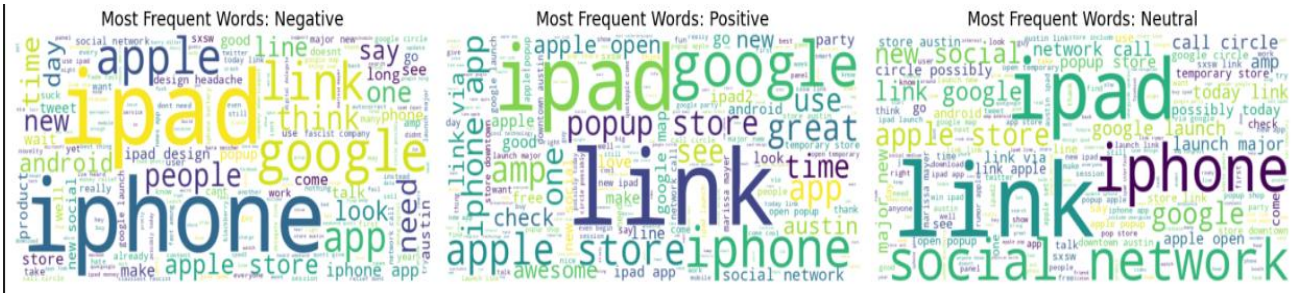
a) Sentiment vs. Brand Mention: The sentiment distribution appears similar for both Apple and Google brands, with a majority of neutral sentiment.



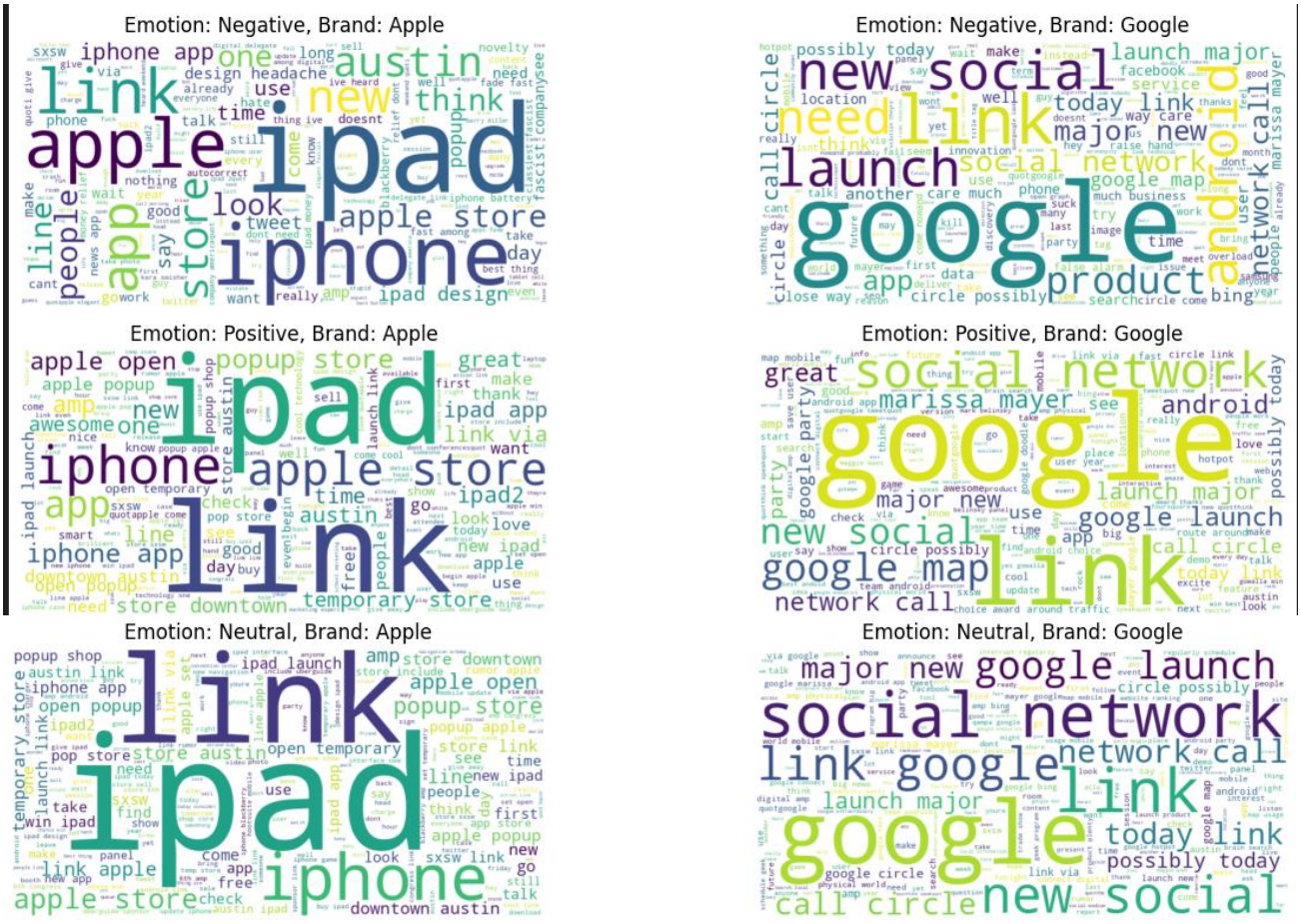
b) Sentiment vs. Tweet Length: Tweet length doesn't seem to affect sentiment distribution significantly.



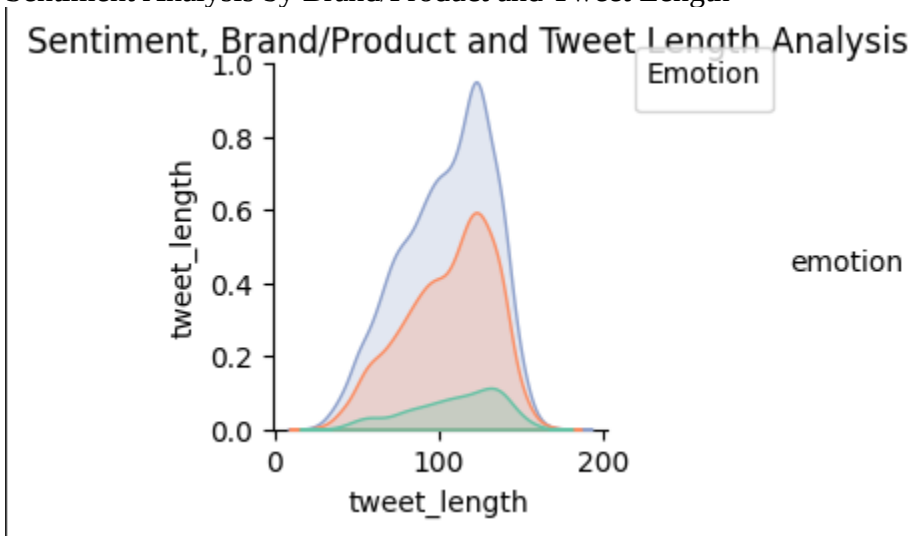
c) Sentiment vs. Specific Brands: Word clouds provided insights into the sentiment associated with specific brands (e.g., "love" and "happy" for positive sentiment towards Apple products).



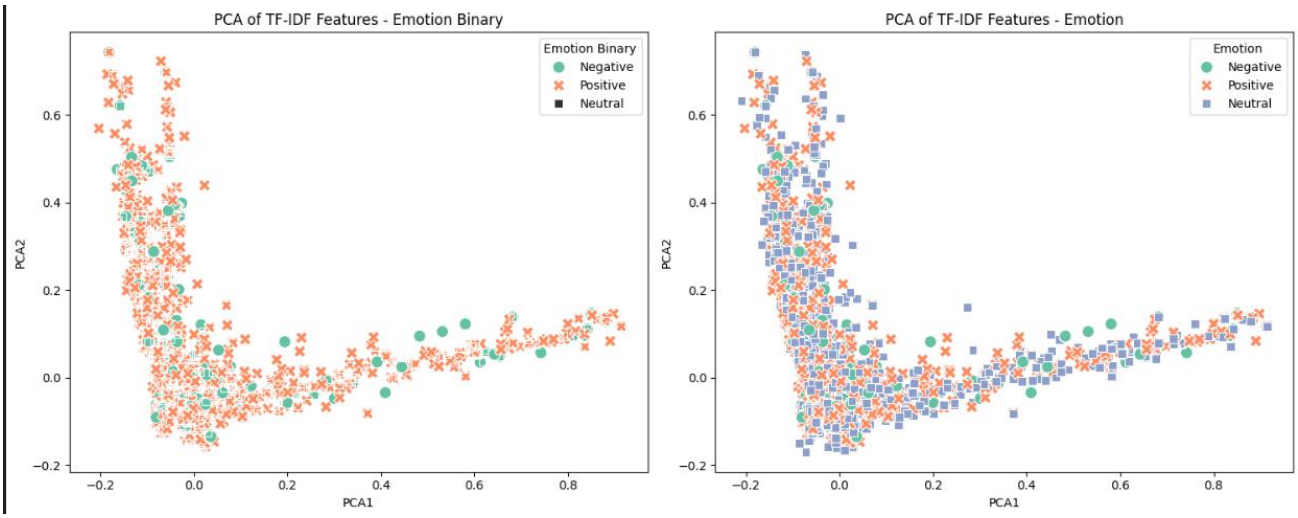
d) Sentiment vs. brand vs. frequency of words: Here, we still observe that the most frequent words are almost equally common and appear in most sentiment categories. However, as expected, apple products are mostly mentioned in the `Apple` category, and Google & products are mentioned in the `Google` category.



e) Sentiment Analysis by Brand/Product and Tweet Length



f) NLP Features (TF-IDF or Word Embeddings) and Sentiment



The left plot clearly separates Negative and Positive emotions in a binary classification, with some overlap in the middle. The right plot introduces a Neutral category, adding complexity and increasing overlap between classes, especially in the middle region. While PCA effectively reduces dimensionality, including the Neutral class, class boundaries are less defined, highlighting the challenge of distinguishing between closely related emotions. The binary classification achieves better separation due to reduced labeling complexity

Data encoding

- The categorical columns, emotion, and brand_category, were encoded into numerical labels for machine learning modeling.

- Encoding the `emotion` column: We assigned the `Negative` emotion to `0`, `Positive` to `1`, and `Neutral` to `2`. This will allow us to have this as a binary column when we create a subset of the data with only the positive and negative sentiments.

- Encoding the `brand_category` column: Since our `brand_category` column only has two categories, we automatically map it to be binary, with `Google` as `0` and `Apple` as `1`.

- We then created a dataframe with our encoded columns only. Checked the shape of our preprocessed text dataframe, `X-df` before concatenating to ensure that both dataframes have the same number of rows.

- Concatenated the processed dataframe by combining `X_df` and `encoded_df` to create `final_df`, a clean and fully preprocessed dataset.

- We then created a binary dataframe where we filtered out the tweets with only the `positive` and `negative` sentiments in the target `emotion` column.

Modeling.

Step 1. Importing and Splitting the two clean dataframes(df_binary and df).

We split the dataset into features and target variables, with the emotion column as the target in binary and multi-class classification tasks. The data is divided into training and testing sets for both datasets, typically with 80% of the data used for training and 20% for testing. A random state is set to ensure the split's reproducibility. This approach allows for a consistent evaluation of models on both binary and multi-class classification tasks, providing training and testing datasets for each.

Evaluate Models: LogisticRegressionModel, RandomForestClassifier, Xgboost Model, SVMModel Models.

Model training and tuning in the SentimentModel class involved several key steps.

1) The preprocess_data method scales the features using StandardScaler and optionally applies Principal Component Analysis (PCA) to reduce dimensionality, retaining 95% of the variance.
    - This ensures that the model works with well-conditioned data, improving performance.
2) The apply_smote method addresses class imbalance by applying the Synthetic Minority Over-sampling Technique (SMOTE), which generates synthetic samples for the minority class.
3) The train method, the class iterates through multiple machine learning models (Logistic Regression, SVM, Random Forest, and XGBoost), training each model on the preprocessed and balanced data. After training, the models make predictions on the test set, and their performance is evaluated using metrics such as accuracy, classification report, and confusion matrix. This workflow ensures the models are properly trained, tuned, and evaluated for binary and multi-class sentiment analysis tasks.

Binary Models

The Binary Classification results highlight:

- Logistic Regression:

Accuracy: 80.33%

Balanced performance with a weighted F1-score of 0.80.
Misclassifications are evident in the confusion matrix, with a few instances incorrectly classified between classes 0 and 1.

- SVM:

Accuracy: 92.33%

Strong precision and recall, resulting in a weighted F1-score of 0.92.

Minimal misclassifications, making it one of the best-performing models.

- Random Forest:

Accuracy: 87.33%

Good overall performance with a weighted F1-score of 0.87.

Slightly higher misclassification rates compared to SVM.

- XGBoost:

Accuracy: 89.11%

Strong performance with a weighted F1-score of 0.89.

Performs better than Random Forest but slightly below SVM.

Multi-Class classification

The results of the model training and evaluation highlight the performance of four classification models: Logistic Regression, Support Vector Machine (SVM), Random Forest, and XGBoost among the models:

- Logistic Regression achieved an accuracy of 88.4%. The confusion matrix shows it performs well across all three classes, with a weighted F1-score of 0.88, indicating balanced precision, recall, and F1 performance.

- SVM stands out with an accuracy of 94.0%, demonstrating strong classification performance across all classes. Its weighted F1-score of 0.94 highlights its precision and recall balance, with the confusion matrix revealing minimal misclassifications.

- Random Forest closely follows with an accuracy of 93.3%. It shows strong precision and recall for all classes, with a weighted F1-score of 0.93. The confusion matrix indicates a slightly higher misclassification rate compared to SVM and XGBoost.

- XGBoost performs best, achieving the highest accuracy of 94.13%. It provides the best balance of precision, recall, and F1-scores, with a weighted F1-score of 0.94. Its confusion matrix highlights minimal misclassifications, particularly for the first two classes.

While Logistic Regression serves as a solid baseline, SVM, Random Forest, and XGBoost demonstrate superior performance, with XGBoost slightly outperforming the others. These results suggest that SVM and XGBoost are the most suitable models for this classification task.

Outputting the findings in tabular form(below)

	Model	Classification Task	Accuracy	Precision	Recall	F1-Score
0	Logistic Regression	Binary	0.8033	0.80	0.80	0.80
1	Logistic Regression	Multi	0.8840	0.88	0.88	0.88
2	SVM	Binary	0.9233	0.92	0.92	0.92
3	SVM	Multi	0.9400	0.94	0.94	0.94
4	Random Forest	Binary	0.8733	0.87	0.87	0.87
5	Random Forest	Multi	0.9333	0.93	0.93	0.93
6	XGBoost	Binary	0.8911	0.89	0.89	0.89
7	XGBoost	Multi	0.9413	0.94	0.94	0.94

Results and Insights

Model Performance

- SVM and XGBoost consistently deliver superior performance in both binary and multi-class tasks, achieving the highest accuracy and F1-scores.
- Random Forest provides robust results but tends to have slightly more misclassifications than SVM and XGBoost.
- Logistic Regression serves as a strong baseline model, particularly excelling in interpretability. However, it lags behind in accuracy compared to the other models.

These results suggest that SVM and XGBoost are the most reliable options for tasks prioritizing accuracy and precision.

Key Insights

- Tweets containing strong adjectives (e.g., "amazing," "terrible") were easier to classify accurately.
- Neutral tweets posed the biggest challenge due to their ambiguous language.
- The preprocessing steps significantly improved model performance.

Conclusion

This project successfully developed an NLP model capable of analyzing Twitter sentiment related to Apple and Google products.

Based on social media data, this sentiment analysis report provides valuable insights into customer perception of Apple and Google products. The findings reveal a neutral sentiment bias, with a higher volume of tweets mentioning Apple products.

Further analysis using machine learning models can help quantify these observations and identify specific aspects influencing sentiment.

By continuously monitoring social media sentiment, Apple and Google can better understand customer satisfaction and use these insights to improve their products, marketing strategies, and overall customer experience.

Recommendations

- Consider using deep learning models such as LSTMs or transformers for better context understanding.
- Enhance feature engineering by incorporating sentiment lexicons and contextual embeddings.
- Regularly update the model with new data to keep it relevant.
- - Prioritize SVM and XGBoost for effective classification of tweet emotions due to their superior performance in sentiment analysis.
- Further, optimize BERT to better capture the complexities of tweet emotions and improve sentiment classification accuracy or combine BERT with some of the prioritized traditional models (SVM and XGBOOST).
- Explore Multi-Label Classification: Implement multi-label classification to capture tweets with mixed or overlapping emotions more accurately.
- Enhance Text Representation: Use more advanced text representation techniques like word embeddings to improve the model's understanding of tweet emotions.
- Consider using deep learning models such as LSTMs for better context
- understanding
- Performing aspect-based sentiment analysis to understand the specific features of products that evoke positive or negative sentiment.

Limitations

This report focuses on a basic sentiment analysis of social media posts. Here are some limitations to consider:

- The dataset might not be entirely representative of the global social media landscape.

- Sarcasm and other forms of nuanced language can be challenging to detect accurately using sentiment analysis techniques.

- The report primarily focuses on sentiment classification. Aspect-based sentiment analysis, which identifies the specific aspects of products that users have positive or negative feelings about, could provide even deeper insights.

Next Steps

- Optimize BERT's parameters to enhance sentiment analysis, especially for complex tweet emotions.
- Deploy the Model: Implement the best-performing model to analyze emotions in real-time tweets.
- Monitor and Retrain: Continuously monitor the model's performance and retrain it with new tweet data to maintain accuracy.