**TWEET SENTIMENT PROJECT**

**1. Business Understanding**

**1.1. Overview/Background**

The landscape of traditional advertising has changed dramatically, with many companies now employing highly targeted strategies. By understanding customer demographics, businesses can communicate more directly and effectively. Social media platforms like Twitter and Facebook offer a direct channel for consumers to share their opinions about brands, products, and services. While this real-time feedback is invaluable, managing the sheer volume of messages can be challenging.

**1.2. Business Problem**

Consumers frequently use social media to share their thoughts, presenting companies with the challenge of extracting actionable insights from the overwhelming amount of data. For example, during SXSW 2011, Apple and Google introduced numerous new products and services, resulting in a flood of tweets. Sifting through thousands of unlabeled tweets to gain meaningful insights is a significant challenge for companies like Apple.

**1.3. Project Aim and Scope**

The project aims to assist companies such as Apple and Google by developing a predictive classification model that can analyze tweets. This model will categorize the sentiment of tweets as "Positive" or "Not positive" (including neutral or sentiment-lacking tweets). By doing so, companies can better organize and utilize the information embedded in the tweets they receive during events like SXSW or in their regular social media interactions.

**1.4. Stakeholders**

The primary stakeholders in this project are companies like Apple and Google, which will benefit from an enhanced ability to understand consumer sentiment. Secondary stakeholders include marketing teams, product managers, and customer service departments that can leverage these insights for better decision-making and strategy formulation.

**1.5. Success Criteria**

Aim: To realize a high accuracy rate, ideally above 70%.

**1.6. Conclusion**

This classification model offers numerous benefits, particularly by identifying "Positive" tweets about Apple and Google products:

1. Gauging public opinion
2. Obtaining direct consumer feedback
3. Retraining the model on custom datasets for specific products or regions
4. Understanding target demographics
   * Identifying individuals who show positive interest
   * Tailoring advertising strategies more effectively
   * Facilitating advertising within social media circles of existing fans, enhancing outreach and engagement

* The project has the potential to transform how companies interact with and respond to consumer feedback on social media, providing valuable insights and fostering better consumer relationships.

**2.0 DATA UNDERSTANDING**

To build an effective NLP model for analyzing Twitter sentiment about Apple and Google products, we need to thoroughly understand the dataset and its properties. The dataset in question comes from CrowdFlower via data.world and contains over 9,000 tweets that have been rated by human raters as positive, negative, or neither. Below is a detailed breakdown of the data understanding process:

**2.1 Dataset Overview**

The dataset consists of 9,093 tweets related to technology products and brands, with a focus on Apple and Google products. The data was collected during and after the 2011 South by Southwest (SXSW) Conference. Each tweet has been pre-labeled by human raters for sentiment analysis and product and brand identification.

**2.2 Source of Data and Suitability**

**Source**

The dataset used in this project originates from CrowdFlower, now known as Figure Eight, and was subsequently made available on [data.world](https://data.world/crowdflower/brands-and-product-emotions) .Kent Cavender-Bares contributed this valuable resource on August 30, 2013, sharing it with the data science community.

The data is contained in a CSV file named "judge-1377884607\_tweet\_product\_company.csv", which serves as the primary source for our analysis. This file contains a wealth of information about consumer sentiments towards technology products, particularly focusing on tweets related to Apple and Google during the 2011 South by Southwest (SXSW) Conference.

**Suitability**

* The dataset is highly relevant for the project as it specifically contains tweets about Apple and Google products, aligning perfectly with the project's aim.
* The manual sentiment ratings (positive, negative, or neither) provide a robust foundation for training a sentiment analysis model.
* Raters judged if the tweet's text expressed a positive, negative, or no emotion towards a brand . When an emotion was expressed, the rater identified the brand that was the target of that emotion.

**2.3 Data Size and Structures**

* **Data Size**

The dataset comprises over 9,000 tweets, which is a substantial amount for training an NLP model. This size is generally sufficient to capture a wide range of sentiment expressions and variations in language.

* **Data Structures**

The resulting data file contains three columns per row:

a) tweet\_text: The actual content of the tweet

b) emotion\_in\_tweet\_is\_directed\_at: The product or brand the emotion is directed at (if identifiable)

c) is\_there\_an\_emotion\_directed\_at\_a\_brand\_or\_product: The sentiment of the tweet (Positive, Negative, or No emotion)

**2.4. Feature Inclusion and Relevance**

### Features:

* **Tweet Text:** The primary feature for sentiment analysis. The content of the tweet will be tokenized and transformed into numerical representations (e.g., TF-IDF, word embeddings) for model training.
* **Emotion Expressed:** The target variable for the model, indicating whether the sentiment is positive, negative, or neither.
* **Target Product/Brand:** This feature can provide additional context and help in understanding the sentiment in relation to specific products or brands.

### Justification:

* The tweet text is directly relevant as it contains the information needed to determine sentiment.
* The emotion expressed is essential for supervised learning, providing the ground truth for model training and evaluation.
* The target product/brand feature can help in fine-tuning the model to understand sentiment in the context of specific products or brands, which is particularly useful for companies like Apple and Google.

**2.5. Data Limitations**

### Limitations:

* **Class Imbalance:** The dataset has a significant imbalance in sentiment classes (e.g., There was overwhelmingly neutral with very few examples of negative sentiment.), this could affect model performance. Techniques such as resampling or class weighting may be necessary.
* **Noise and Ambiguity:** Tweets often contain slang, abbreviations, and emojis, which can introduce noise and ambiguity. Preprocessing steps like text normalization and the use of advanced NLP techniques can help mitigate this.
* **Contextual Understanding:** Tweets are short and may lack context, making it challenging to accurately determine sentiment. Incorporating additional context or using models capable of understanding nuanced language (e.g., transformers) can improve performance.

By thoroughly understanding the dataset and addressing its limitations, we can build a robust NLP model to analyze Twitter sentiment about Apple and Google products. This model will help companies like Apple and Google gain valuable insights from social media feedback, enhancing their ability to respond to consumer sentiment effectively.

**3.0 DATA EXPLORATION**

Before diving into model building, let's get acquainted with our training data. By exploring its characteristics, we can uncover valuable insights. This exploration will focus on key aspects like data types, missing values (null values), the frequency of different values (value counts), and the spread of classes (for classification tasks).

## Data Types

Understanding the data types of each feature is crucial for preprocessing and model building. We'll check the data types of all columns in the dataset to ensure they are appropriate for the tasks ahead.

## Missing Values (Null Values)

Missing values can significantly impact model performance. We'll identify columns with missing values and determine the best strategies to handle them, such as imputation or removal.

## Value Counts

Analyzing the frequency of different values in categorical columns can provide insights into the distribution of the data. We'll perform value counts for key columns to understand their distributions better.

## Spread of Classes

For classification tasks, it's essential to understand the distribution of the target variable. We'll examine the spread of sentiment classes (positive, negative, neither) to check for class imbalances and plan appropriate strategies if needed.

## 3.1 Data Preparation

This process will include thorough cleaning of the tweet text by removing special characters, standardizing product names, and handling any missing or inconsistent values. We will also focus on preprocessing the text data through tokenization, removing stop words, and potentially applying stemming or lemmatization techniques. Finally, we'll encode the sentiment categories and balance the dataset if necessary, ensuring our model has a solid foundation for learning the patterns in consumer sentiment towards tech products.

The DataFrame consists of three columns with a total of 9093 entries. The datatypes of all columns are objects (strings). The column-specific observations are:

* **Tweet:** Contains 9092 non-null entries, indicating there is 1 missing value.
* **Brand/Product:** Contains 3291 non-null entries, indicating a significant number of missing values (5802). These missing data points will be filled in with an 'Uncategorized' label since they are not detrimental to the current study.
* **Sentiment:** Contains 9093 non-null entries, with no missing values.

## 3.2 Analyzing Key Features

## 3.2a Brand Data

The extract\_brand function is included to improve the accuracy and consistency of brand identification in the dataset. By leveraging both the Brand/Product and Tweet text fields, the function helps in reducing the number of "Uncategorized" entries and provides a clearer picture of the brand distribution, facilitating better analysis and decision-making.

Our extract\_brand function effectively categorized most tweets with their corresponding brands. Since our data includes hashtags, we'll add new columns: one for hashtags, another for word count, and a final one replicating the original tweet text for future modeling purposes.

## 3.2b Tweet Sentiment Data

We will streamline our sentiment analysis by creating a binary classification system. This involves examining the current sentiment labels and introducing a new column with simplified categories: "Positive" for positive sentiments and "Not Positive" for all others. This transformation from multi-class to binary sentiments will sharpen our classification model's focus, enabling it to more effectively distinguish positive tweets from the rest. By reducing complexity, we aim to enhance the model's accuracy and make results more easily interpretable.

## 3.3 Preparing Tweets for Modelling

We'll now focus on cleaning the tweet text. This process involves removing unnecessary characters like punctuation, URLs, and hashtags. Additionally, we'll convert all text to lowercase and eliminate common stop words. Following this cleaning step, we'll tokenize the text, splitting it into individual words. Finally, we'll lemmatize these tokens, reducing them to their base forms. This will result in a list of meaningful words (lemmas) for each tweet, ready for further analysis.

To ensure our analysis captures the emotional tone of tweets effectively, we'll customize the stop word list from NLTK. This involves removing generic words that might hold emotional significance in our context. Additionally, we'll remove punctuation to match the format of the cleaned tweets and prepare them for the clean\_and\_lemmatize\_tweet function.

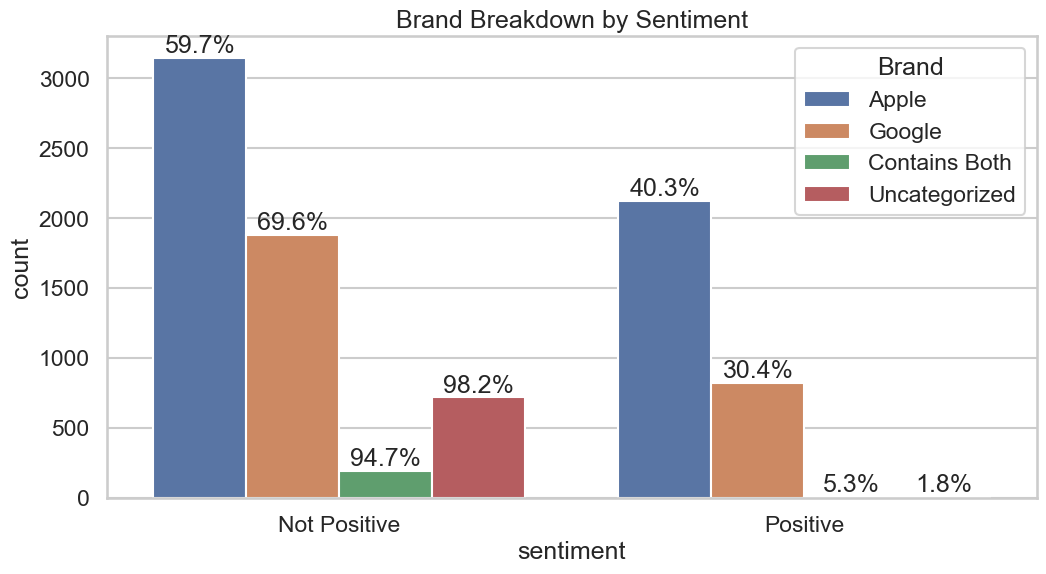
NLTK Stop Words: ['a', 'about', 'above', 'after', 'again', 'against', 'ain', 'all', 'am', 'an', 'and', 'any', 'are', 'aren', "aren't", 'as', 'at', 'be', 'because', 'been', 'before', 'being', 'below', 'between', 'both', 'but', 'by', 'can', 'couldn', "couldn't", 'd', 'did', 'didn', "didn't", 'do', 'does', 'doesn', "doesn't", 'doing', 'don', "don't", 'down', 'during', 'each', 'few', 'for', 'from', 'further', 'had', 'hadn', "hadn't", 'has', 'hasn', "hasn't", 'have', 'haven', "haven't", 'having', 'he', 'her', 'here', 'hers', 'herself', 'him', 'himself', 'his', 'how', 'i', 'if', 'in', 'into', 'is', 'isn', "isn't", 'it', "it's", 'its', 'itself', 'just', 'll', 'm', 'ma', 'me', 'mightn', "mightn't", 'more', 'most', 'mustn', "mustn't", 'my', 'myself', 'needn', "needn't", 'no', 'nor', 'not', 'now', 'o', 'of', 'off', 'on', 'once', 'only', 'or', 'other', 'our', 'ours', 'ourselves', 'out', 'over', 'own', 're', 's', 'same', 'shan', "shan't", 'she', "she's", 'should', "should've", 'shouldn', "shouldn't", 'so', 'some', 'such', 't', 'than', 'that', "that'll", 'the', 'their', 'theirs', 'them', 'themselves', 'then', 'there', 'these', 'they', 'this', 'those', 'through', 'to', 'too', 'under', 'until', 'up', 've', 'very', 'was', 'wasn', "wasn't", 'we', 'were', 'weren', "weren't", 'what', 'when', 'where', 'which', 'while', 'who', 'whom', 'why', 'will', 'with', 'won', "won't", 'wouldn', "wouldn't", 'y', 'you', "you'd", "you'll", "you're", "you've", 'your', 'yours', 'yourself', 'yourselves']

**4.0 Exploratory Data Analysis**

## 4.1 Sentiment proportion

## 

* The significant difference in the number of tweets between “Not Positive” and “Positive” suggests an imbalance.

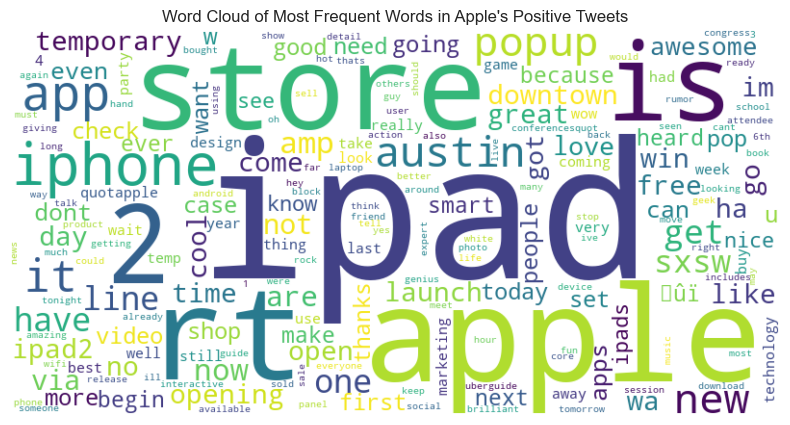


* Overall, not positive sentiments dominate across brands not just in count but in overall percentage of tweets which makes sense as there are about twice as many not positive tweets in the dataset

## 4.2 Brand Coverage

## 

## 4.2a Analyze positive tweets for Apple products



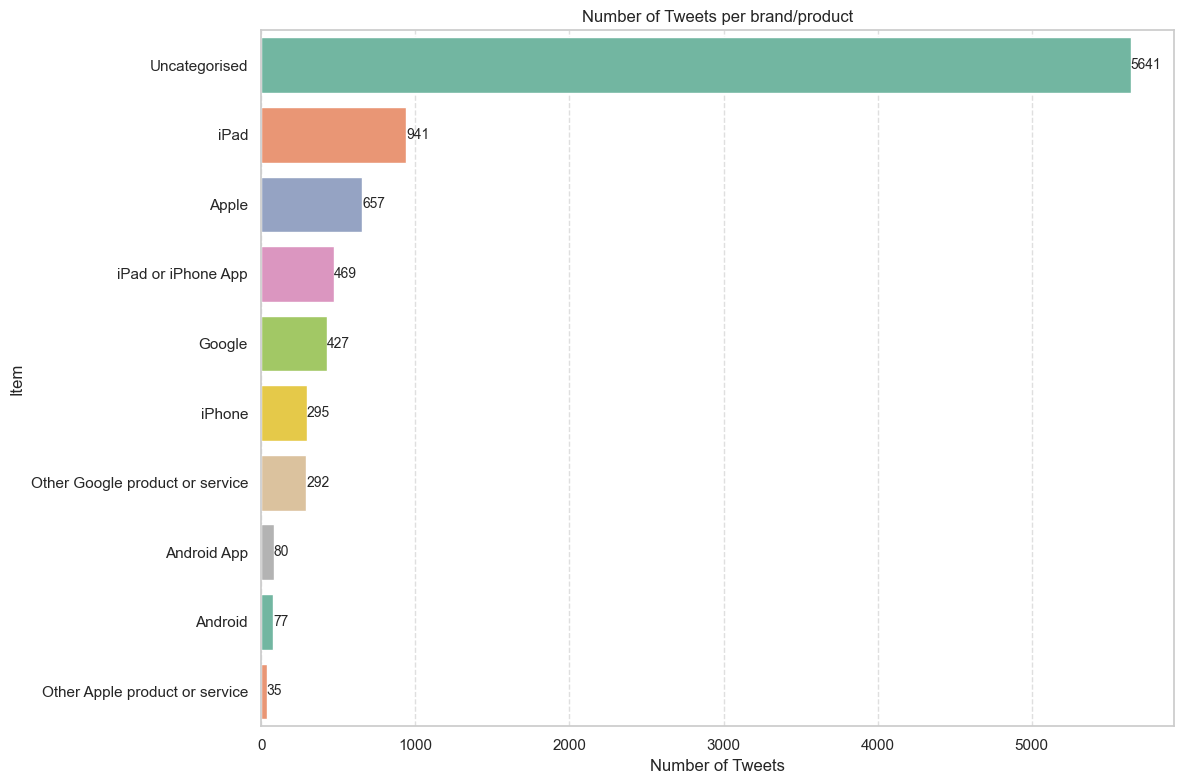
* The word cloud analysis indicates that Apple products, particularly the iPad 2 and iPhone, are frequently mentioned in positive tweets. The iPhone app and Apple Store also appear prominently, suggesting a strong association with positive sentiment.
* The inclusion of "free" implies potential giveaways or promotions that may have contributed to the overall positive tone. The word cloud also hints at a positive reception for the new Apple Store in downtown Austin.

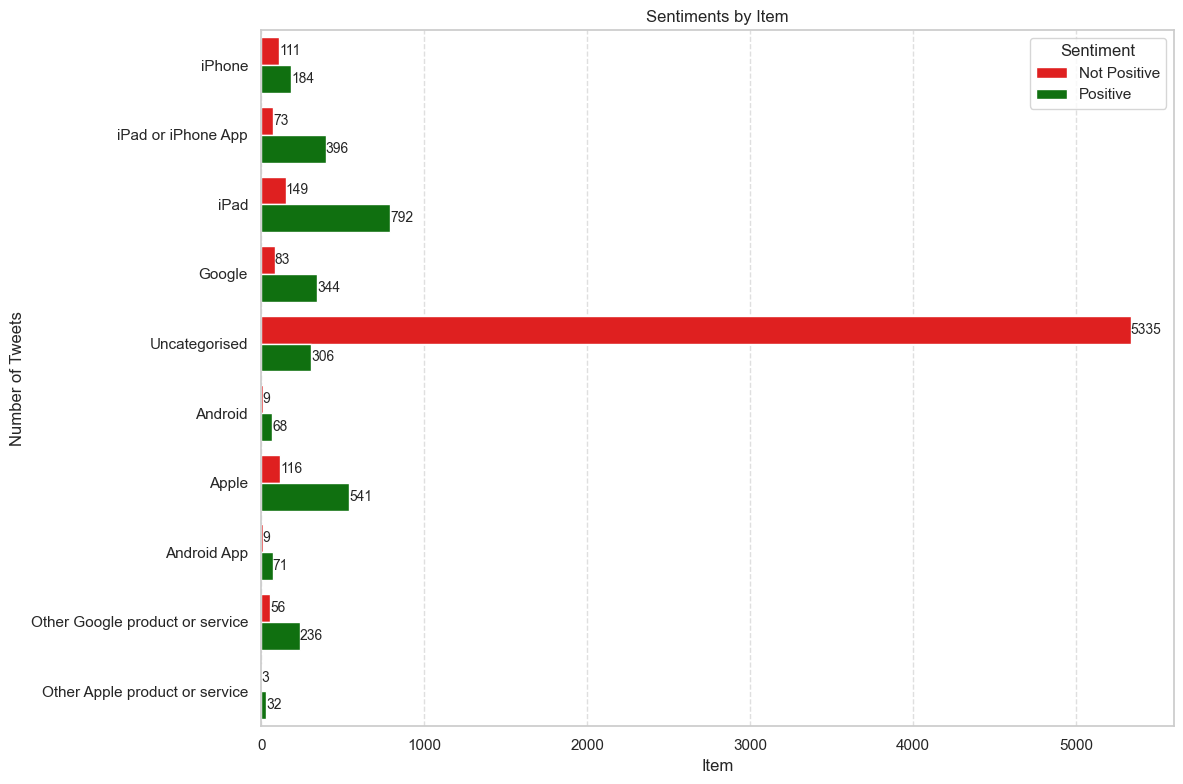
# **4.2b Analyze negative tweets for Apple products. Same process as for the positive tweets.**



* Negative sentiment towards the company is primarily focused on its applications, physical stores, and customer interactions. Battery life issues are frequently cited.
* There are complaints about long wait times at a specific store location. Additionally, the company's business practices and product design have been criticized, with terms like "fascist," "bad," and "fail" appearing in negative tweets. Some comparisons to a competitor's products are evident in the data.

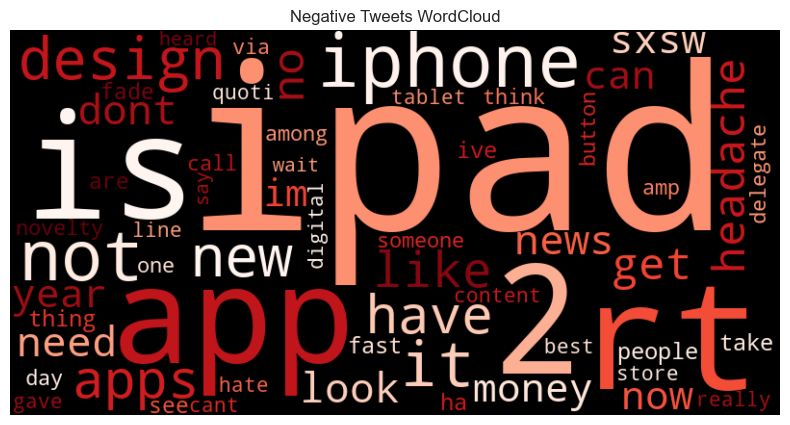
## Brand/Product distribution

****



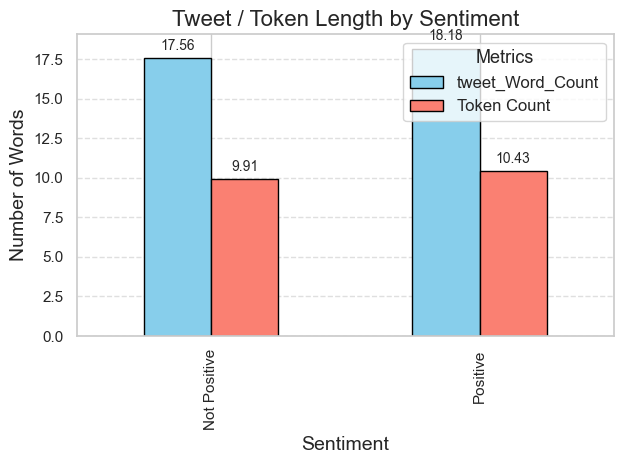
* The item with the highest number of positive sentiments is “iPad,” followed closely by “Apple.”
* It can be seen that all brand/products except the uncategorized have many more positive tweets than negative.

# **The negative tweets about iPad**

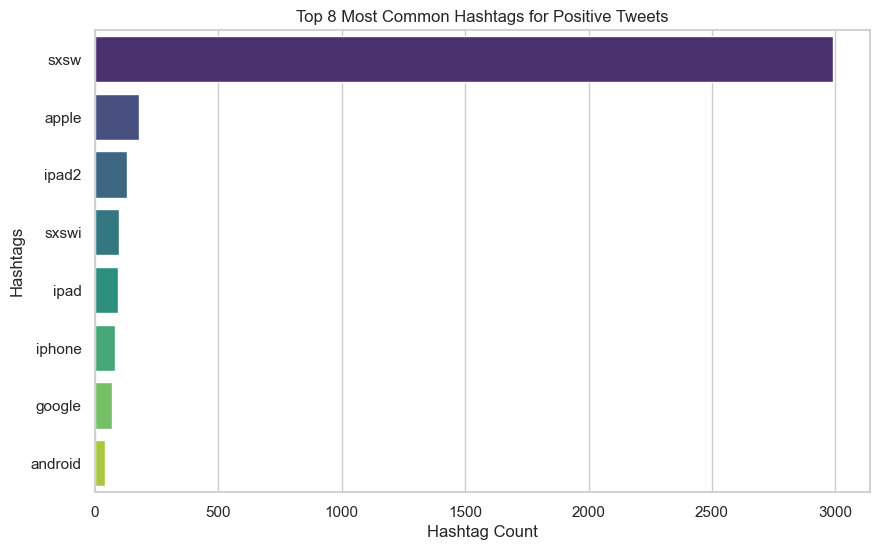


* Analysis of negative tweets about the iPhone indicates that battery life is the primary customer complaint. Additionally, comparisons to a competing smartphone brand are evident in the data.

# **Tweet word count/ Token Count by Sentiment**



# **Hashtag Exploration**



**Top 15 Hashtags for Not Positive are:** [('sxsw', 5951), ('google', 250), ('apple', 236), ('sxswi', 217), ('iphone', 181), ('ipad', 167), ('ipad2', 164), ('android', 88), ('circles', 80), ('austin', 70), ('tech', 59), ('japan', 51), ('ubersocial', 47), ('gsdm', 47), ('infektd', 46)]

* Preliminary analysis indicates that hashtag frequency is consistent across different sentiment categories. Consequently, hashtags are deemed unlikely to be effective predictors of sentiment.

# **NLP Modeling**

# Data Preprocessing for Modeling

A new DataFrame df\_model is created that contains only the columns relevant for NLP modeling: 'Sentiment', 'Processed\_Tweet', and 'Token Count'. This helps in focusing on the necessary data and reduces memory usage by excluding irrelevant columns

sentiment\_encoded

0 0.630596

1 0.369404

Name: proportion, dtype: float64

* we got class imbalance in our dataset so will use SMOTE to increase minority counts

# **DummyClassifier**

# TfidfVectorizer with DummyClassifier

Pipeline(steps=[('vectorizer',

TfidfVectorizer(stop\_words=['i', 'me', 'my', 'myself', 'we',

'our', 'ours', 'ourselves', 'you',

"you're", "you've", "you'll",

"you'd", 'your', 'yours',

'yourself', 'yourselves', 'he',

'him', 'his', 'himself', 'she',

"she's", 'her', 'hers', 'herself',

'it', "it's", 'its', 'itself', ...],

token\_pattern=None,

tokenizer=<bound method TweetTokenizer.tokenize of <nltk.tokenize.casual.TweetTokenizer object at 0x0000016412A47CB0>>)),

('smote', SMOTE(random\_state=42, sampling\_strategy=1)),

('classifier',

DummyClassifier(random\_state=42, strategy='most\_frequent'))])

precision recall f1-score support

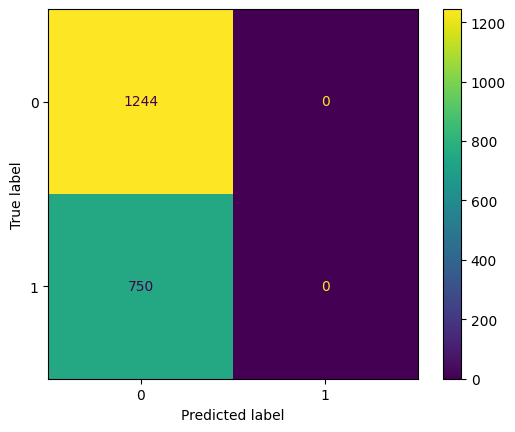
0 0.62 1.00 0.77 1244

1 0.00 0.00 0.00 750

accuracy 0.62 1994

macro avg 0.31 0.50 0.38 1994

weighted avg 0.39 0.62 0.48 1994



# Random Forest Model

## TfidfVectorizer with Random Forest:

Pipeline(steps=[('vectorizer',

TfidfVectorizer(stop\_words=['i', 'me', 'my', 'myself', 'we',

'our', 'ours', 'ourselves', 'you',

"you're", "you've", "you'll",

"you'd", 'your', 'yours',

'yourself', 'yourselves', 'he',

'him', 'his', 'himself', 'she',

"she's", 'her', 'hers', 'herself',

'it', "it's", 'its', 'itself', ...],

token\_pattern=None,

tokenizer=<bound method TweetTokenizer.tokenize of <nltk.tokenize.casual.TweetTokenizer object at 0x0000016412A47CB0>>)),

('smote', SMOTE(random\_state=42, sampling\_strategy=1)),

('rfc', RandomForestClassifier(random\_state=42))])

precision recall f1-score support

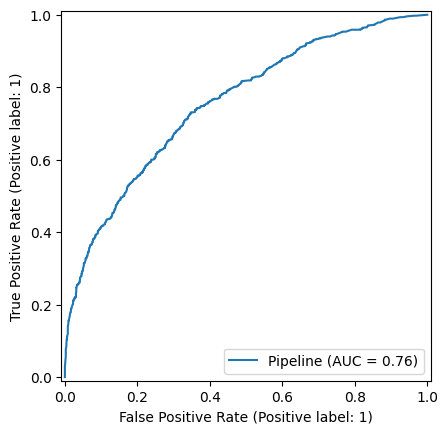
0 0.75 0.82 0.78 1244

1 0.64 0.54 0.58 750

accuracy 0.71 1994

macro avg 0.69 0.68 0.68 1994

weighted avg 0.70 0.71 0.71 1994



## CountVectorizer with Random Forest:

Pipeline(steps=[('vectorizer',

CountVectorizer(stop\_words=['i', 'me', 'my', 'myself', 'we',

'our', 'ours', 'ourselves', 'you',

"you're", "you've", "you'll",

"you'd", 'your', 'yours',

'yourself', 'yourselves', 'he',

'him', 'his', 'himself', 'she',

"she's", 'her', 'hers', 'herself',

'it', "it's", 'its', 'itself', ...],

token\_pattern=None,

tokenizer=<bound method TweetTokenizer.tokenize of <nltk.tokenize.casual.TweetTokenizer object at 0x0000016412A47CB0>>)),

('smote', SMOTE(random\_state=42, sampling\_strategy=1)),

('rfc', RandomForestClassifier(random\_state=42))])

Count Vectorizer:

precision recall f1-score support

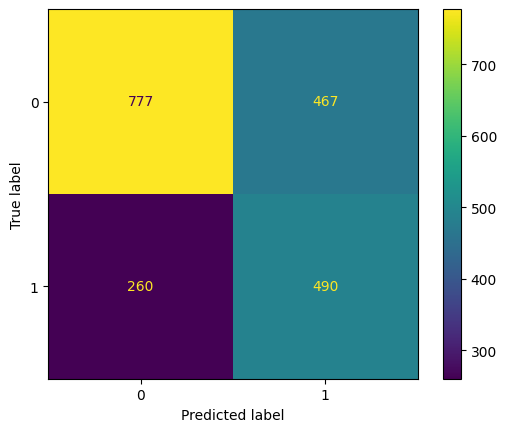
0 0.75 0.62 0.68 1244

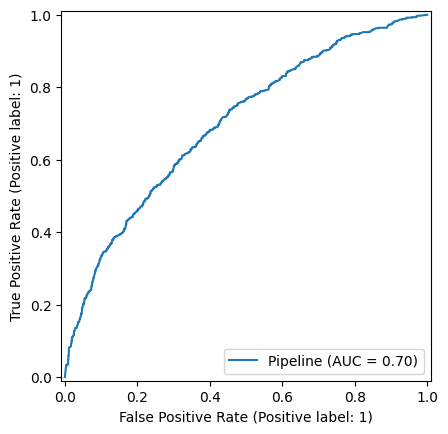
1 0.51 0.65 0.57 750

accuracy 0.64 1994

macro avg 0.63 0.64 0.63 1994

weighted avg 0.66 0.64 0.64 1994





## Combined Features with Random Forest

Classification Report:

precision recall f1-score support

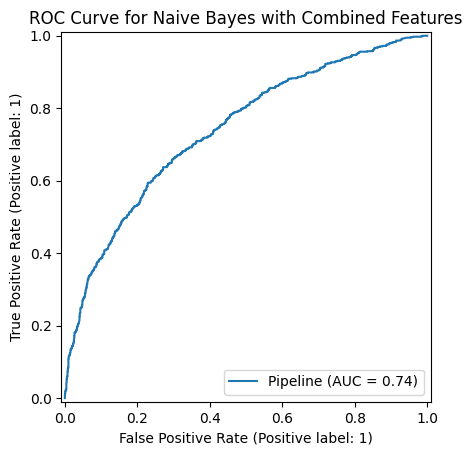
0 0.78 0.66 0.72 1244

1 0.55 0.69 0.61 750

accuracy 0.67 1994

macro avg 0.67 0.68 0.66 1994

weighted avg 0.69 0.67 0.68 1994



* Combined Features seems to be the best approach. It provides a balance between precision and recall for both classes and achieves a reasonable accuracy. This approach leverages the strengths of both TfidfVectorizer and CountVectorizer, resulting in a more robust model.
* Comparing this random forest model with combined features with dummy model, this one performs much better with an accuracy of 0.67 and an AUC score of 0.74

**GridSearch on random forest**

Classification Report:

precision recall f1-score support

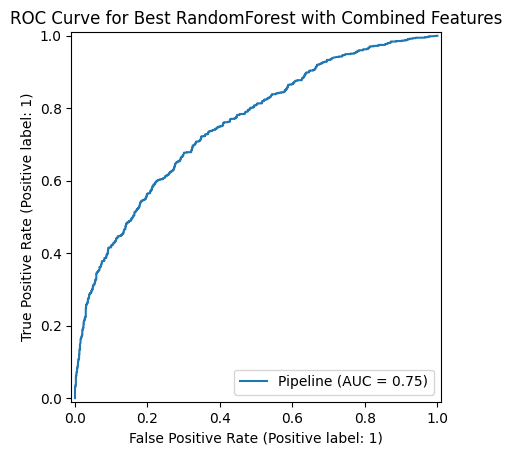
0 0.74 0.83 0.78 1244

1 0.65 0.52 0.57 750

accuracy 0.71 1994

macro avg 0.69 0.67 0.68 1994

weighted avg 0.71 0.71 0.70 1994

****

* The overall accuracy of the model is 71%, indicating that 71% of the tweets were correctly classified.
* The model correctly identified 83% of the negative sentiment tweets, meaning it has a strong ability to detect negative tweets
* The model correctly identified only 52% of the positive sentiment tweets, meaning it misses almost half of the positive tweets.

# 1st logistic regression model

## logistic regression model with TfidfVectorizer

Pipeline(steps=[('vectorizer',

TfidfVectorizer(stop\_words=['i', 'me', 'my', 'myself', 'we',

'our', 'ours', 'ourselves', 'you',

"you're", "you've", "you'll",

"you'd", 'your', 'yours',

'yourself', 'yourselves', 'he',

'him', 'his', 'himself', 'she',

"she's", 'her', 'hers', 'herself',

'it', "it's", 'its', 'itself', ...],

token\_pattern=None,

tokenizer=<bound method TweetTokenizer.tokenize of <nltk.tokenize.casual.TweetTokenizer object at 0x0000016412A47CB0>>)),

('smote', SMOTE(random\_state=42, sampling\_strategy=1)),

('lr', LogisticRegression(random\_state=42))])

precision recall f1-score support

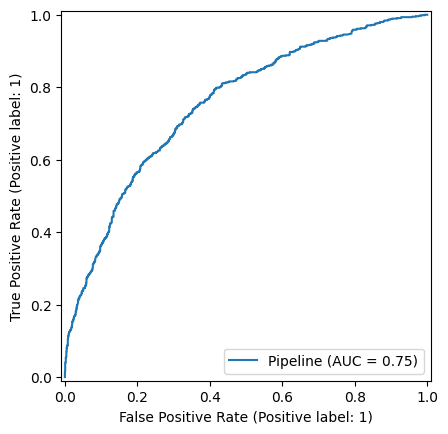
0 0.77 0.74 0.75 1244

1 0.59 0.63 0.61 750

accuracy 0.70 1994

macro avg 0.68 0.68 0.68 1994

weighted avg 0.70 0.70 0.70 1994



## Logistic regression model with CountVectorizer

Pipeline(steps=[('vectorizer',

CountVectorizer(stop\_words=['i', 'me', 'my', 'myself', 'we',

'our', 'ours', 'ourselves', 'you',

"you're", "you've", "you'll",

"you'd", 'your', 'yours',

'yourself', 'yourselves', 'he',

'him', 'his', 'himself', 'she',

"she's", 'her', 'hers', 'herself',

'it', "it's", 'its', 'itself', ...],

token\_pattern=None,

tokenizer=<bound method TweetTokenizer.tokenize of <nltk.tokenize.casual.TweetTokenizer object at 0x0000016412A47CB0>>)),

('smote', SMOTE(random\_state=42, sampling\_strategy=1)),

('lr', LogisticRegression(random\_state=42))])

Classification Report with CountVectorizer:

precision recall f1-score support

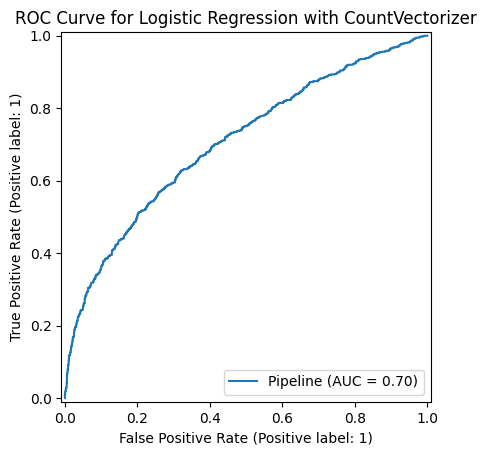
0 0.75 0.68 0.71 1244

1 0.54 0.62 0.58 750

accuracy 0.66 1994

macro avg 0.65 0.65 0.65 1994

weighted avg 0.67 0.66 0.66 1994



* TfidfVectorizer is the better choice in this scenario as it demonstrates higher accuracy, better precision, recall, and F1-score for both classes. The overall macro and weighted averages also favor the TfidfVectorizer, indicating that it provides better performance for the logistic regression model on this dataset.

## GridSearch on Logistic Regression

precision recall f1-score support

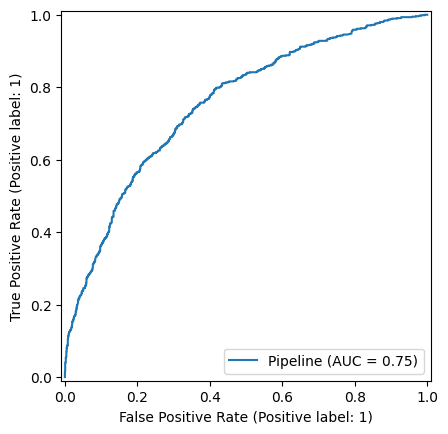
0 0.77 0.74 0.75 1244

1 0.59 0.63 0.61 750

accuracy 0.70 1994

macro avg 0.68 0.68 0.68 1994

weighted avg 0.70 0.70 0.70 1994



**Modeling Conclusion: random forest model with combined features has the best results. With an Accuracy: 0.71 Recall for Class 0: 0.83 ,AUC Score: 75**

* Based on the analysis, the model with Combined Features, SMOTE, and RandomForestClassifier generally performs the best, especially in terms of accuracy, precision, and recall for negative sentiment (Class 0). It also ties with the Tuned Logistic Regression in several other metrics, including AUC.
* Given our goal of extracting useful information from tweets on Apple and Google products and exploring the emotion distribution, the Pipeline with Combined Features, SMOTE, and RandomForestClassifier is the best model to use for your project. It provides a good balance of precision, recall, and overall accuracy, making it suitable for analyzing sentiment in tweets.

# Adaptation and Utilization of the Model for Apple and Google

## For Apple:

### Product Feedback:

* **Positive Tweets:** Focus on aspects like iPad2, iPhone, and Apple Store, especially noting mentions of "amazing" and "free products."
* **Negative Tweets:** Address issues related to applications, physical stores, and battery life. Mentions of "bad," "fail," "fascist," and long wait times at stores indicate specific pain points.

### Customer Interaction:

* **Positive Sentiment:** Highlight customer experiences with promotional giveaways and the opening of new retail locations.
* **Negative Sentiment:** Improve customer service at physical stores to reduce wait times. Address battery life concerns and enhance product design to avoid criticisms.

## For Google:

### Product Feedback:

* **Positive Tweets:** Highlight features of Google Maps, social network launches, and positive mentions of company events or leaders (e.g., Marissa Mayer).
* **Negative Tweets:** Address issues with the mapping product, and any controversies surrounding product launches, indicated by terms like "suck," "lost," and "deadly."

### Customer Interaction:

* **Positive Sentiment:** Promote successful events and product launches, emphasizing user engagement and satisfaction.
* **Negative Sentiment:** Improve the functionality of products like Google Maps and address any issues related to product launches.

## Findings from Wordcloud Visualization

### Apple:

* **Positive Sentiment:** iPad2, iPhone, Apple Store, amazing, free.
* **Negative Sentiment:** Applications, physical stores, battery life, fascist, bad, fail, long wait times, business practices, product design.

### Google:

* **Positive Sentiment:** Map, circle, party, launch, social network, new, mobile, Marissa Mayer, free.
* **Negative Sentiment:** Social, circle, launch, bing, network, suck, lost, deadly, mapping product.

## Recommendations

### For Apple:

1. **Address Battery Life Issues:** Prioritize enhancements in battery technology and customer education on battery management.
2. **Improve Store Experience:** Reduce wait times and enhance customer service.
3. **Promote Positive Experiences:** Continue with promotional giveaways and leverage positive feedback in marketing campaigns.
4. **Respond to Criticisms:** Be proactive in addressing criticisms about business practices and product design.

### For Google:

1. **Enhance Product Functionality:** Focus on improving the mapping product and resolving any issues with new product launches.
2. **Leverage Positive Feedback:** Use successful events and launches to strengthen marketing and customer engagement strategies.
3. **Address Negative Sentiment:** Actively respond to criticisms and controversies to maintain customer trust and satisfaction.

## Words Leading to Sentiment Classification

### Positive Words:

* **Apple:** iPad2, iPhone, Apple Store, amazing, free.
* **Google:** Map, circle, party, launch, social network, new, mobile, Marissa Mayer, free.

### Negative Words:

* **Apple:** Fascist, bad, fail, battery life, long wait times, business practices, product design.
* **Google:** Suck, lost, deadly, social, circle, bing, network, mapping product.

## Future Work

### Model Improvement:

1. **Hyperparameter Tuning:** Further refine models through hyperparameter tuning.
2. **Ensemble Methods:** Explore ensemble methods to improve prediction accuracy.
3. **Deep Learning Models:** Implement deep learning models like RNNs or BERT for better understanding of context.

### Data Expansion:

1. **Additional Data Sources:** Incorporate data from other social media platforms and review sites.
2. **Larger Dataset:** Expand the dataset to include more tweets for more robust analysis.

### Real-Time Analysis:

1. **Stream Processing:** Implement real-time data streaming to provide up-to-date sentiment analysis.
2. **Dashboard Development:** Create dashboards for Apple and Google to monitor sentiment trends and customer feedback in real-time.

### Customized Analysis:

1. **Product-Specific Models:** Develop separate models for different products to gain more granular insights.
2. **Regional Analysis:** Conduct regional sentiment analysis to understand geographical variations in customer feedback.

* By leveraging these findings and recommendations, Apple and Google can enhance their customer service, improve product offerings, and better understand their customers' sentiments, ultimately leading to increased customer satisfaction and loyalty.