NLP TWITTER ANALYSIS
FINAL GROUP PROJECT SUBMISSION

GROUP MEMBERS

1. Benson Kamau
2. Kevin Muchori
3. Nancy Chelangat
4. Sally Kinyanjui
5. Breden Mugambi

Student Pace: Full-Time

Instructor: Nikita Njoroge

Contents

NLP ANALYSIS ON TWITTER FEEDBACK ON APPLE AND GOOGLE PRODUCTS

Overview/background information

Sentiment analysis involves classifying opinions in text into categories like "positive" or "negative" or "neutral". It's also referred as subjectivity analysis, opinion mining, and appraisal extraction.

The aim is to analyze people's sentiments, attitudes, opinions emotions, etc. towards elements such as products, individuals, topics ,organizations, and services. Given Twitter's role as a major social media platform where users frequently express opinions, analyzing sentiment can provide valuable insights into public perception and consumer sentiment.

Businesses, marketers, and legislators need this information in order to improve engagement efforts and make data-driven decisions.

Problem statement

Accurately classifying the sentiments expressed in tweets about topics or brands into specific classes- positive, negative or neutral is a huge challenge for companies like Apple and Google. Given the diverse nature of informal data, with its use of slang, abbreviations, coming up with a reliable sentiment analysis model that can effectively interpret and classify the tweets can be a complex task. Getting this task right provides a wide variety of novel information for a company like Apple by providing insights and creating better understanding overall of how consumers interact with products/brands.

Objectives

The main objective is to build a model that can rate the sentiment of a tweet based on its content.

Challenges

Analyzing data on twitter comes with its own challenges. They include:

1. Data Variability.

   The texts/sentiments are highly variable i.e., users may use different levels of formality, slang, abbreviations or emojis.

   Tweets can also range from simple statements to complex ones, others may include sarcasm or irony.

   Something else to consider is the contextual differences, context in which a tweet is posted can influence its sentiment or emotion.

2. Sentiment imbalance

   Positive, negative, and neutral sentiment labels may be distributed unevenly, which could result in biased models that perform well in majority classes but badly in minority ones.

3. Scalability

   As more people access social media platforms like Twitter, the volume of tweets increases, leading to a challenge in efficiently processing and managing this growing data. Ensuring that the model can handle the large scale of data effectively and operates on a scalable infrastructure is essential.

Proposed solutions

1. Exploring various NLP techniques and models like the Logistic Regression, support vector machine(SVM) or advanced techniques like LSTM and CNN to handle sentiment classification. Choice of model depends entirely on specific requirements and resources available.

2. Data Preprocessing that will help in handling the data variability to ensure there is standardization of the input for the model.

3. Metrics of Success: Aim for at least 70% accuracy in sentiment classification. Evaluate using metrics such as Precision, Recall, F1 Score, and Confusion Matrix to ensure balanced performance across sentiment classes.

Conclusion

Challenges with data variability, sentiment imbalance, and scalability must be addressed by the model in order to assess Twitter sentiment properly. The sentiment analysis tool can offer insightful information about public opinion and enhance decision-making processes by emphasizing strong preprocessing, choosing relevant models, and striving for high accuracy.

DATA UNDERSTANDING

The dataset that was used contains 9093 data points. The dataset that will be used in this study comes from CrowdFlower via data.world through this link - https://data.world/crowdflower/brands-and-product-emotions.

Raters judged if the tweet's text expressed a positive, negative or no emotion towards a brand and/or product, any time an emotion was expressed the rater was then asked to identify the brand or product that was the target of that emotion.

The Tweets were in large part centered on Apple and Google products during/after the 2011 South by Southwest (SXSW) Conference. The resulting datafile contains three columns per row, one for the tweet's text, one for the emotion expressed and one for the target product/brand of that emotion, when identifiable.

The dataset contains the following columns:

1. 'tweet_text' column

   - Contains the text of the tweet.

2. 'emotion_in_tweet_is_directed_at' column

   - Contains the person or entity that the tweet is directed at.

3. 'is_there_an_emotion_directed_at_a_brand_or_product' column

   - Indicates the kind of emotion in the tweet directed at the brand or product

DATA CLEANING

Data cleaning is a crucial step in the preprocessing stage to ensure that text data is consistent and free of errors. For this project, we focus on handling missing values, removing duplicates, trimming white spaces, and managing capitalization. Below is a detailed explanation of each step involved in the data cleaning process.

Checking for missing values

To identify any missing values in the dataset, we use a command to check the sum of missing values for each column.

The results indicate:

- There is one missing value in the 'tweet' column.
- There are 5,802 missing values in the 'target_entity' column.
- There are no missing values in the 'emotion' column.

Since a tweet without text is not useful for analysis, the missing value in the 'tweet' column will be removed. The missing values in the 'target_entity' column are acceptable for this project as the focus is on overall tweet sentiment rather than specific items. These missing values will be replaced with the classification "Uncategorized."

Removing null tweets, duplicates, and filling missing target entity values

1. **Removing Null Tweets**: The single null entry in the 'tweet' column will be removed.
2. **Removing Duplicates**: Duplicate entries in the dataset will be removed to ensure there are no repeated records.
3. **Filling Missing Target Entity Values**: The null values in the 'target_entity' column will be replaced with "Uncategorized."

After performing these operations, the dataframe will have 9,070 entries, down from the original 9,092 due to the removal of duplicates and the null tweet. At this point, the dataset will be free of null values and duplicates.

Text cleaning

To ensure uniformity in the text data, the tweets will be cleaned by removing white spaces, converting to lowercase, and removing special characters.

The cleaning process involves:

- **Removing White Spaces**: Leading and trailing white spaces will be removed.
- **Converting to Lowercase**: The text will be converted to lowercase to ensure uniformity.
- **Removing Special Characters**: Special characters will be removed, retaining only alphanumeric characters and spaces.

The cleaned text will be stored in an additional column named `cleaned_tweet`. This column will contain the processed text data, ready for further analysis and modeling, ensuring consistency and accuracy in our project.

A preview of the cleaned data shows that each entry now has a corresponding cleaned version of the tweet, making the dataset ready for subsequent analytical steps.

DATA PREPARATION

To effectively analyze our data, a robust text preprocessing pipeline is essential. Our initial focus was on cleaning the raw tweets and transforming them into a suitable format for analysis.

Lemmatization

A critical step in our process was lemmatization. This linguistic technique reduces words to their base or root form, known as a lemma. By applying lemmatization to our cleaned tweets, we created a new column containing the lemmatized_tweets. This transformation is crucial as it groups together different forms of a word, enhancing the accuracy of subsequent analyses.

Stop Word Removal

Building on the lemmatized text, we identified and eliminated stop words. These are common words (like "the," "and," "is") that carry minimal semantic value in our analysis. Removing stop words helps to focus on the core content of the tweets and improves the efficiency of subsequent processes.

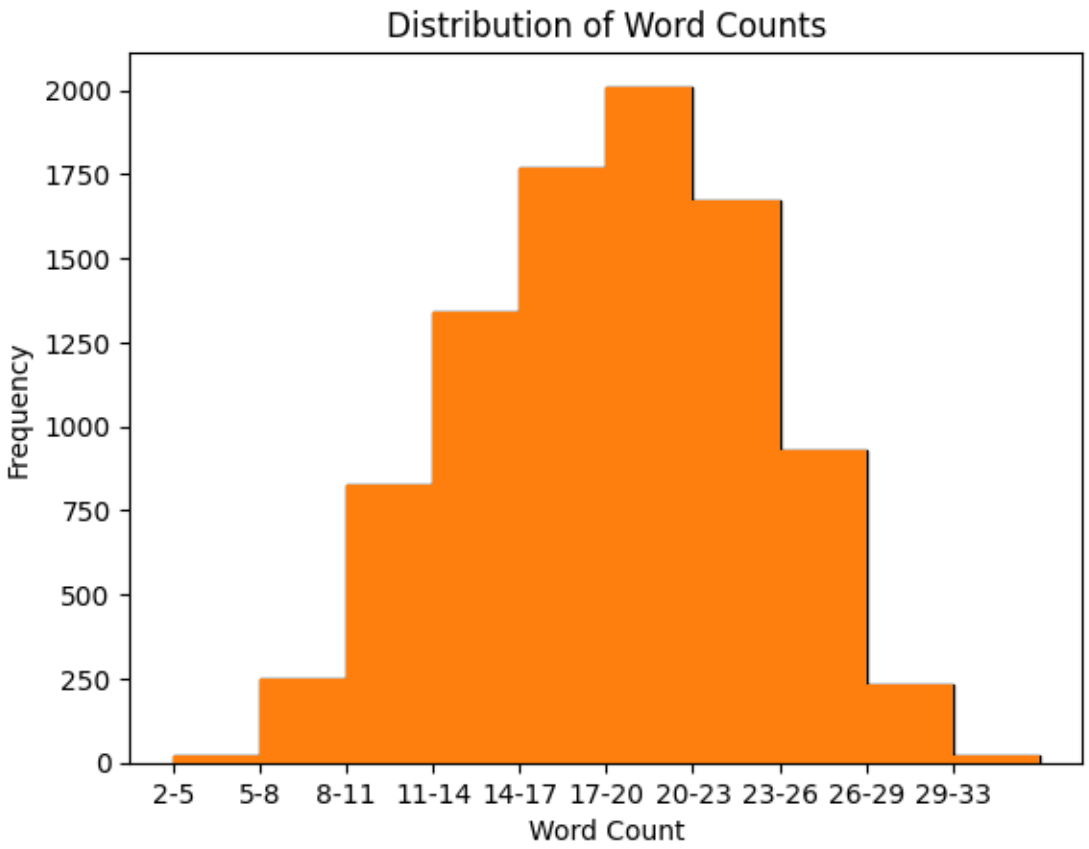
Tokenization

The final preprocessing step involved tokenization, which breaks down the text into individual words or tokens. This was formed in a column called tokenized_tweets

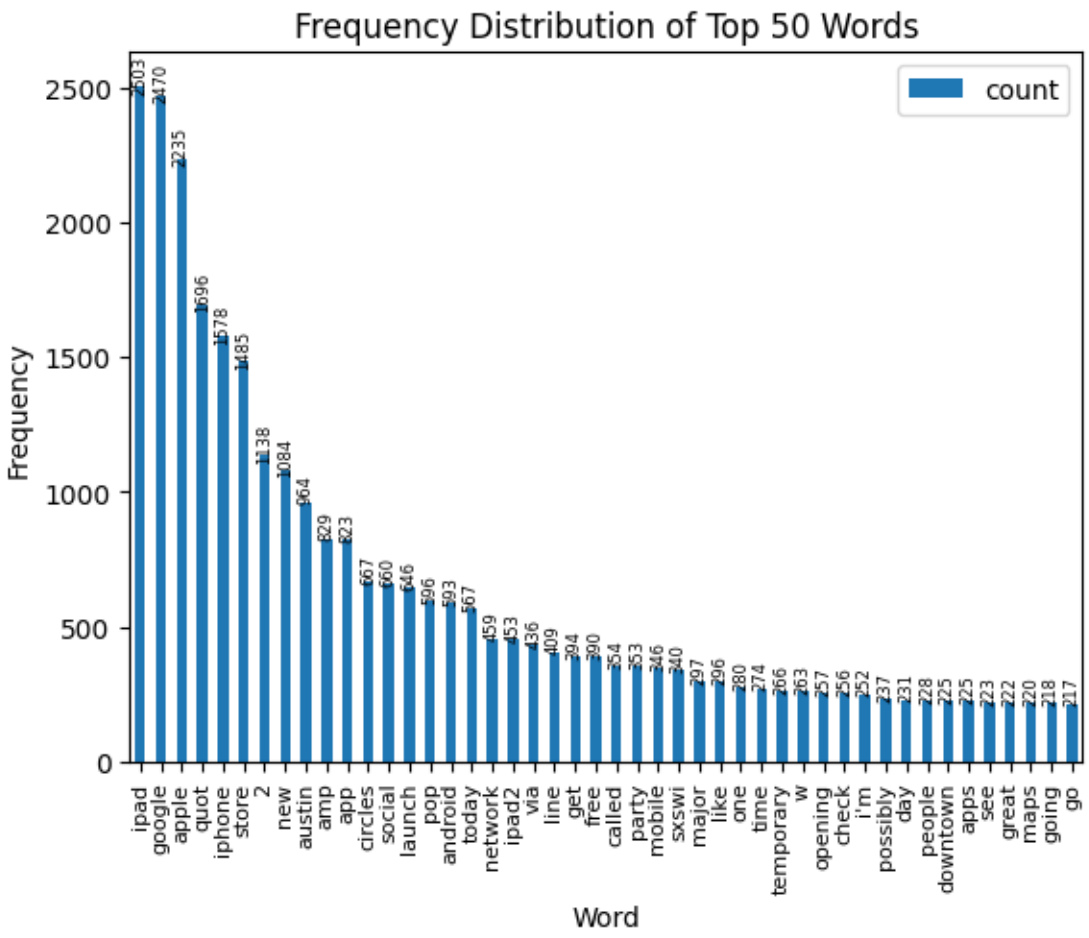
EXPLORATORY DATA ANALYSIS

This involves understanding the characteristics of the text data, identifying patterns, and uncovering potential challenges. By performing EDA, we gain valuable insights into the dataset, inform feature engineering decisions, and build more effective NLP models. This exploratory phase helps to prevent potential pitfalls and ensures that subsequent modeling efforts are grounded in a solid understanding of the data.

Firstly, a function named count_words is defined to calculate the number of words in a given text. This function splits the input text into individual words using the split() method and then returns the total count of these words. Subsequently, the apply method is used to apply the count_words function to every tweet in the df1['tweet'] column, creating a new column named word_count that stores the word count for each tweet.



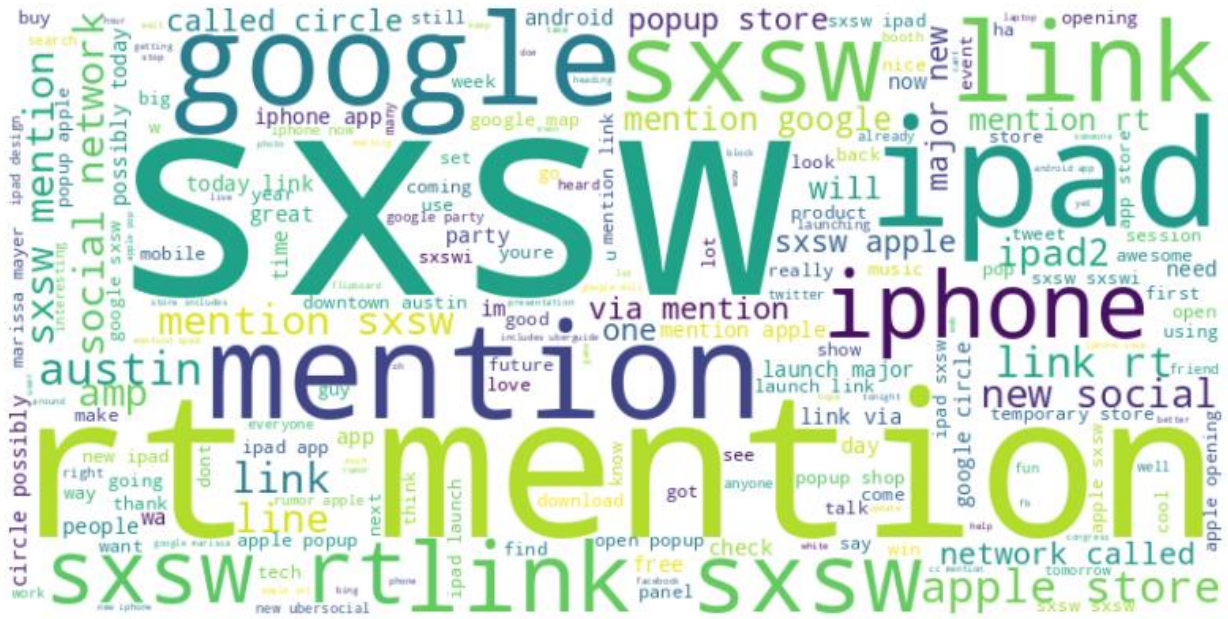
Visually, our data tells us that most of the words in the document lie between 20-23 words. But we need to find out the statistics of usage of each of the words in the document. For this, we can count each word in the "Tweet" column. This s easily accomplished by tokenizing the words in the "Tweet" column, and going through all the words in the column, the calculating word frequencies and visually displaying it



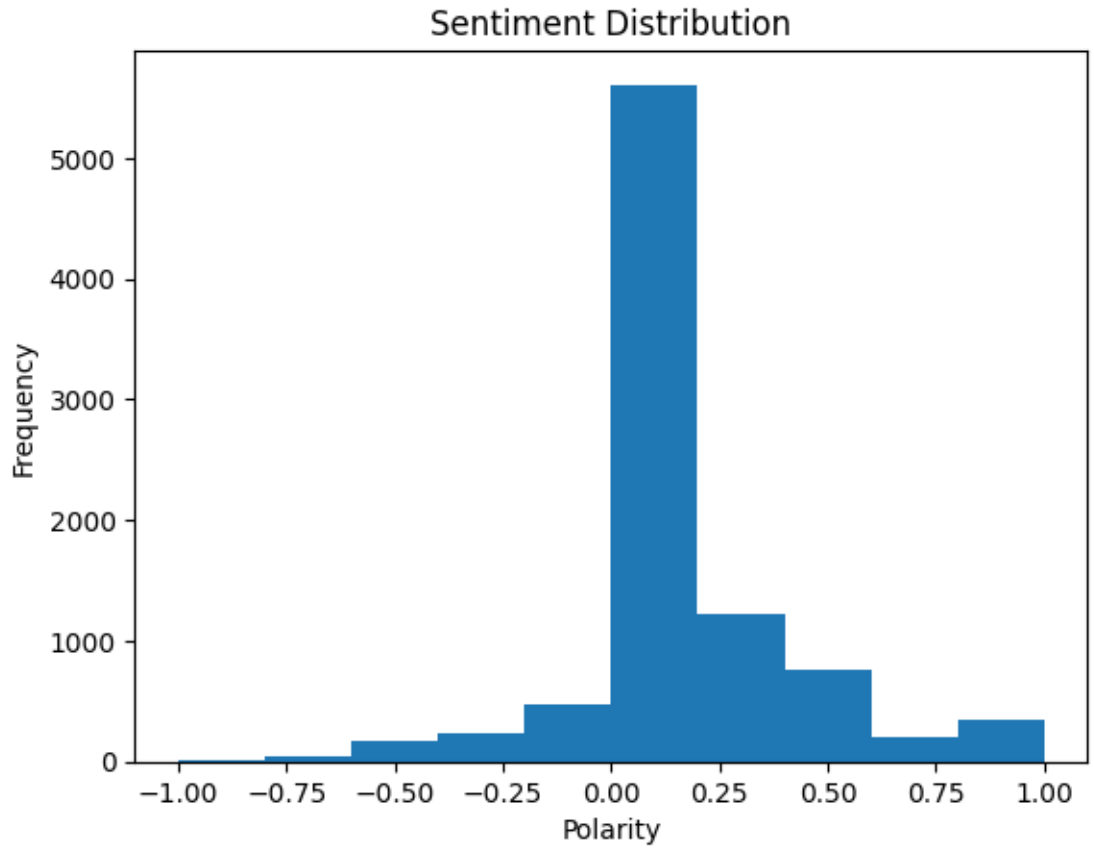
We can now see that the word 'ipad' frequently appears with 2503 appearances. For those who may need a better visualization medium, we can use a wordcloud

A word cloud is a visual representation of text data where words are sized proportionally to their frequency of occurrence in a given document or corpus. It provides a rapid overview of the most prominent terms, allowing for quick identification of key topics and themes. Word clouds are particularly effective when dealing with large volumes of text data, as they can condense information into a visually appealing format.

We can use the lemmatized_tweet column to generate a wordcloud



We can also use a bit of sentiment analysis to determine the positivity, negativity or neutrality of a tweet. We can use a library called TextBlob, as it returns a numerical value, ranging from -1 to 1, where negative 1 indicates negativity, while 1 indicates positivity. O indicates neutrality



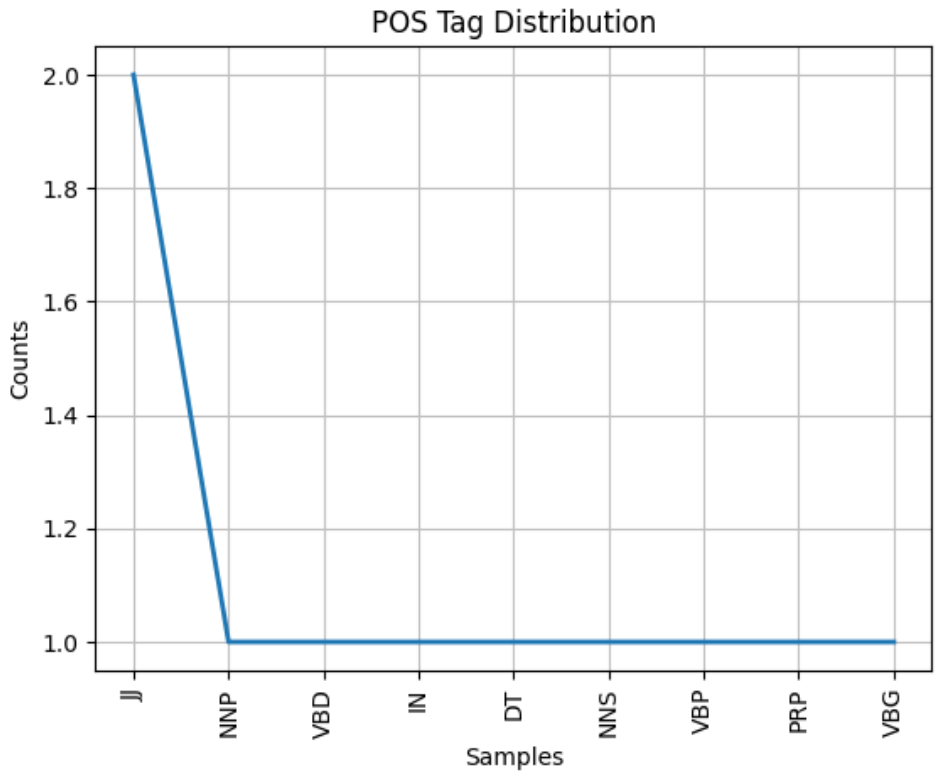
As seen, most of the words lie near the zero mark. But we can go even further. We can use tokens and their tags to find out where most of the words lie in the parts of speech. This is known as parts-of-speech tagging (POS tagging) These tags indicate the syntactic role of a word, such as whether it's a noun, verb, adjective, adverb, and so on.

How POS Tagging Works

- Tokenization: The text is broken down into individual words or tokens.
- Tagging: Each token is assigned a POS tag based on its context and the rules of the language.

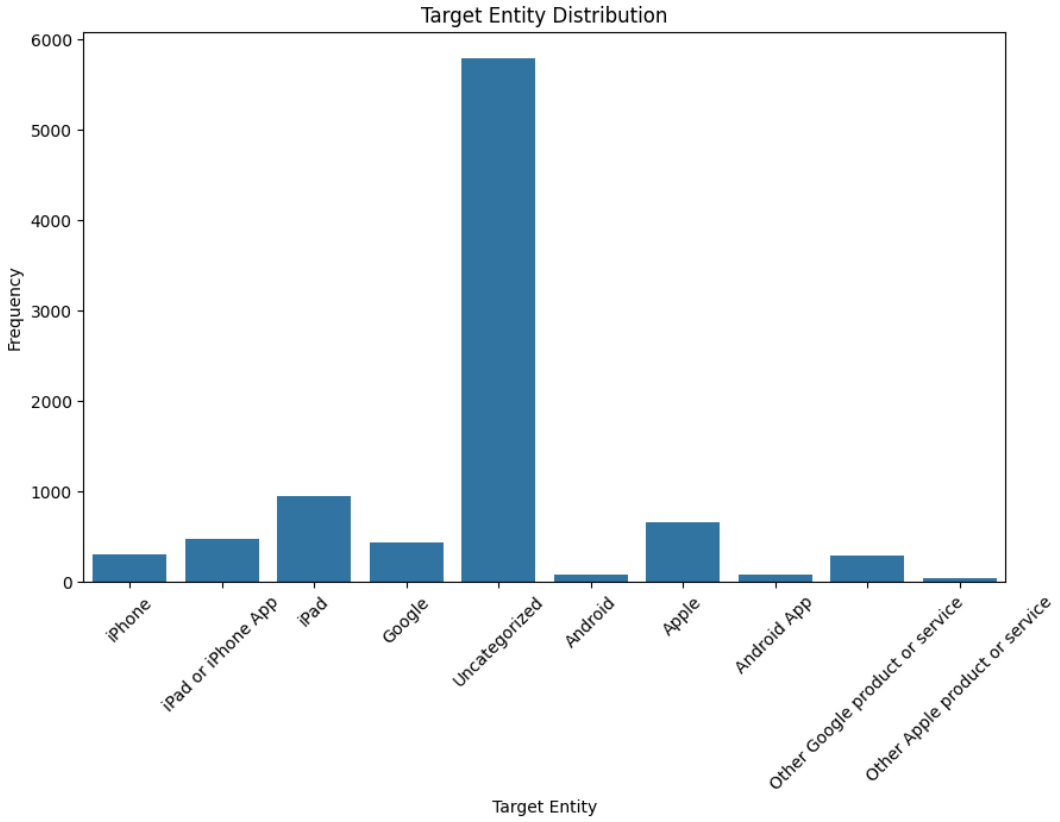
Common POS Tags

- Nouns (NN): person, place, thing, idea
- Verbs (VB): action words
- Adjectives (JJ): describe adjectives
- Adverbs (RB): describe averbs
- Adjectives, or other adverbs
- Prepositions (IN): show relationships between words
- Determiners (DT): articles (the, a, an), possessives (my, your)
- Conjunctions (CC): and, but, or

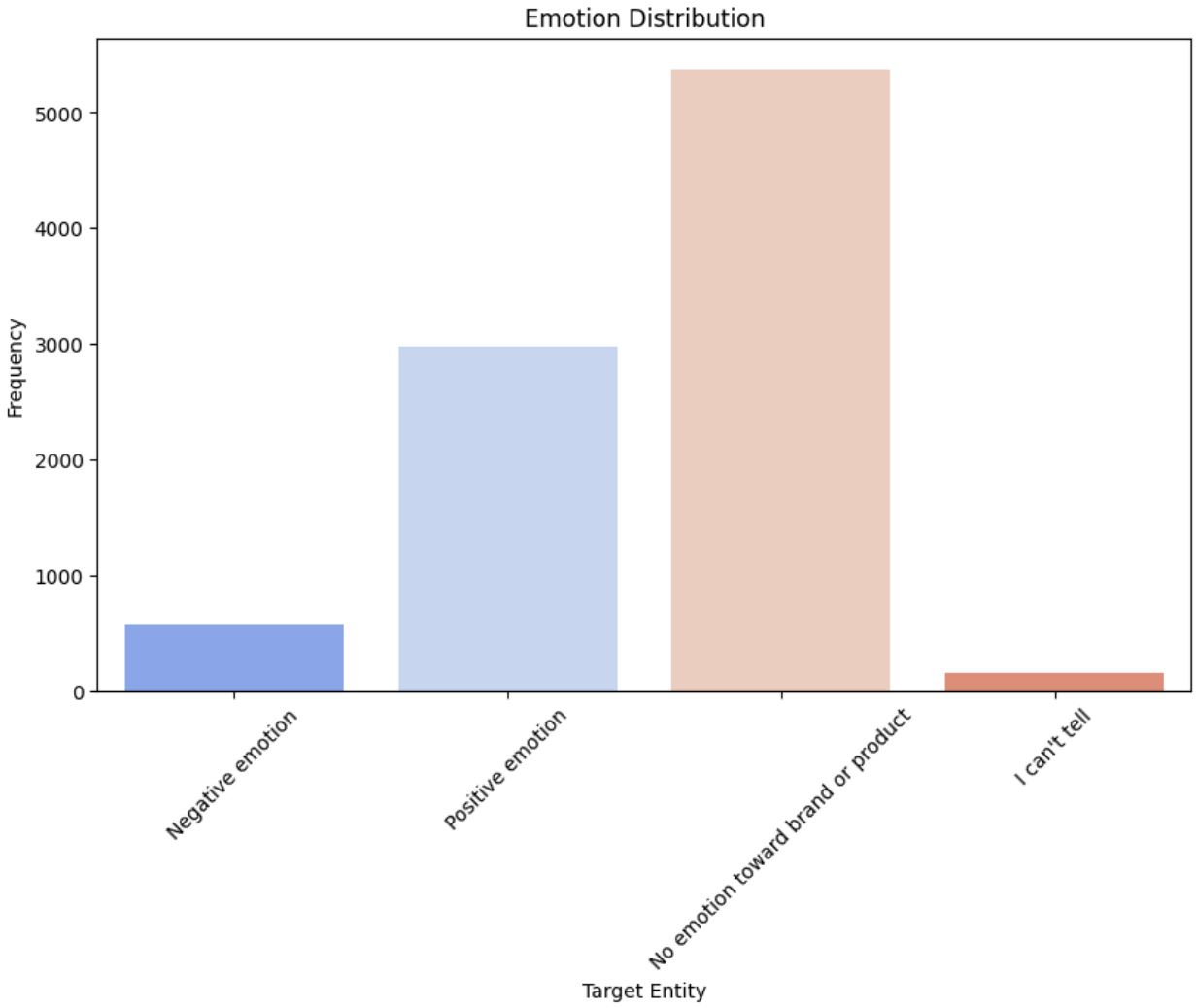


As seen, most of the words are mostly adjectives

in the target entities column, we focus on what services the tweets refer to. We can use the describe() function to find out on the info of the column



With the uncategorized, we remember that that was what we placed on all empty entries in the column. But apart from it, ipad, Apple and Google are the most frequent entered target entries. We then check on the emotion column to see the reception of the target entities based on the tweets


Emotion Distribution
Frequency
0
1000
2000
3000
4000
5000
Negative emotion
Positive emotion
No emotion toward brand or product
I can't tell
Target Entity

MODELLING

We will be using different types of models that is Naive Bayes, Logistic Regression, Decision Tree, and Deep Learning models to see how well they can predict the emotions in the tweets. To check how accurate they are, we will use the metric of accuracy to see how the model works. Accuracy shows how often they get things right.

Vectorization

Converting text data (which is unstructured) into a numerical format that machine learning models will understand and process. It will involve transforming text into numerical vectors that capture the semantic and syntactic information of the text.

Tfidf Vectorization (Term Frequency - Inverse Document Frequency)

Term Frequency-Inverse Document Frequency, is a technique that combines two metrics: TF (Term Frequency) and IDF (Inverse Document Frequency). It's particularly useful when working with multiple documents. TF-IDF helps us identify words that are unique and provide more meaningful insights within a document by giving higher importance to rare words and lowering the importance of common words found across all documents.

We will divide our dataset into two portions: training and testing. This helps us ensure that our models perform well on new, unseen data. We allocate 20% of the data for testing.

We start with the baseline model

1.Naive Bayes

Analysis

Overall accuracy: 0.67 - The model's predictions are correct for 67% of all tweets.

Class "Negative":

Precision: 1.00 - The model correctly identifies negative tweets 100% of the time among all predicted negative tweets.

Recall: 0.02 - The model captures only 20% of actual negative tweets among all actual negative tweets.

Class "No emotion":

Precision: 0.66 - The model correctly identifies no emotion tweets 66% of the time among all predicted no emotion tweets.

Recall: 0.96 - The model captures 96% of actual no emotion towards brand tweets among all no emotion tweets.

Class "positive":

Precision: 0.69 - The model correctly identifies positive tweets 69% of the time among all predicted positive tweets.

Recall: 0.26 - The model captures 26% of actual positive tweets among all actual positive tweets.

The model performs reasonably well for the "No Emotion" class, with a high recall score . This indicates that the model is effective at identifying tweets with no discernible sentiment. However, the model struggles with the "Positive" and "Negative" classes, as indicated by

low recall. This suggests that the model has difficulty distinguishing between these sentiment classes.

Also the overall model accuracy is quite low. We go ahead to look at other different models

2.Logistic regression

Analysis

Overall accuracy: 0.68 - The model's predictions are correct for 68% of all tweets.

Class "Negative":

Precision: 0.59 - The model correctly identifies negative tweets 59% of the time among all predicted negative tweets.

Recall: 0.11 - The model captures only 11% of actual negative tweets among all actual negative tweets.

Class "No emotion":

Precision: 0.72 - The model correctly identifies no emotion tweets 72% of the time among all predicted no emotion tweets.

Recall: 0.86 - The model captures 86% of actual no emotion towards brand tweets among all no emotion tweets.

Class "positive":

Precision: 0.59 - The model correctly identifies positive tweets 59% of the time among all predicted positive tweets.

Recall: 0.50 - The model captures 50% of actual positive tweets among all actual positive tweets.

The model performs reasonably well for the "No Emotion" class, with a high recall score and precision score.

However, the model struggles with the "Positive" and "Negative" classes, as indicated by

low recall but we have an improvement on class positive recall on this model as compared to the base model.

The overall model accuracy improved as compared to baseline model low.This model performed better.

3.Decision Tree

Analysis

Overall accuracy: 0.61 - The model's predictions are correct for 61% of all tweets.

Class "Negative":

Precision: 0.32 - The model correctly identifies negative tweets 32% of the time among all predicted negative tweets.

Recall: 0.28 - The model captures only 28% of actual negative tweets among all actual negative tweets.

Class "No emotion":

Precision: 0.72 - The model correctly identifies no emotion tweets 72% of the time among all predicted no emotion tweets.

Recall: 0.69 - The model captures 69% of actual no emotion towards brand tweets among all no emotion tweets.

Class "positive":

Precision: 0.48 - The model correctly identifies positive tweets 48% of the time among all predicted positive tweets.

Recall: 0.53 - The model captures 53% of actual positive tweets among all actual positive tweets.

The model performs reasonably well for the "No Emotion" class, with a high recall and precision score . This indicates that the model is

effective at identifying tweets with no emotion towards a brand.

However, the model struggles with the "Positive" and "Negative" classes, as indicated by low recall, but in this model the recall score improved to 53 % as compared to that of logistic regression which was 50%. This suggests that the model has difficulty distinguishing between these sentiment classes.

Also the overall model accuracy is low for this model as compared to that of baseline model(67%)and logistic regression model (68%).

4.Support vector machine

Analysis

Overall accuracy: 0.71 - The model's predictions are correct for 71% of all tweets.

Class "Negative":

Precision: 0.87 - The model correctly identifies negative tweets 87% of the time among all predicted negative tweets.

Recall: 0.17 - The model captures only 17% of actual negative tweets among all actual negative tweets.

Class "No emotion":

Precision: 0.72 - The model correctly identifies no emotion tweets 72% of the time among all predicted no emotion tweets.

Recall: 0.90 - The model captures 90% of actual no emotion towards brand tweets among all no emotion tweets.

Class "positive":

Precision: 0.65 - The model correctly identifies positive tweets 65% of the time among all predicted positive tweets.

Recall: 0.49 - The model captures 49% of actual positive tweets among all actual positive tweets.

The model performs reasonably well as compared to the other models

The overall model accuracy stands at 71% the highest compared to the other models.

DEEP LEARNING

We explored the deep learning models:

1. LSTM (Long Short Term Memory) - LSTM is capable of learning long term dependencies. Designed to remember information for long periods.

For this model, we will use the Bidirectional LSTM layer to capture both past and future context. We will also add a dropout layer to prevent overfitting. After that, we will add another bidirectional LSTM layer and another dropout layer. Finally, we will add a Dense layer with a softmax activation function to output the probabilities for each class.

2. CNN (Convolutional Neural Network) - This is effective in capturing local patterns in the data through convolutional filters.

For this model, we will use a 1D convolutional layer with a ReLU activation function followed by a global max pooling layer to capture the most important features. After that, we will add a Dense layer with a softmax activation function to output the probabilities for each class.

3. GRU (Gated Recurrent Unit) - It is effective in learning sequential data and capturing dependencies in the input sequence.

For this model, we will use a bidirectional GRU layer with a dropout layer to prevent overfitting. After that, we will add a Dense layer with a softmax activation function to output the probabilities for each class.

From the models the following observations were made:

1. Training Accuracy:
    - CNN achieved the highest training accuracy (0.9122), suggesting possible overfitting.
    - GRU (0.8700) and LSTM (0.8742) also performed well but lower than CNN.
2. Test Accuracy:
    - CNN (0.6544) outperformed LSTM (0.6466) and GRU (0.6422) in generalization to unseen data.
3. Precision

Class negative:

- GRU: 0.46 (highest)

- CNN: 0.43
- LSTM: 0.40 (lowest)

Class No emotion:

- All models showed similar precision (CNN: 0.73, LSTM: 0.74, GRU: 0.72), with LSTM marginally better.

Class Positive:

- CNN had the highest precision (0.54), followed by LSTM (0.53) and GRU (0.51).

4. Recall

Class positive :

- LSTM had the highest recall (0.34), followed by CNN (0.31) and GRU (0.22).

Class No emotion :

- CNN excelled in recall (0.76), followed by GRU (0.75) and LSTM (0.73).

Class positive:

- LSTM achieved the highest recall (0.58), compared to GRU (0.55) and CNN (0.55).

EVALUATION

In our study, we used the following models:

- Naive Bayes
- Support Vector Machine (SVM)
- Decision Tree
- Logistic Regression

- Long Short-Term Memory (LSTM)
- Convolutional Neural Network (CNN)
- Gated Recurrent Unit (GRU)

To evaluate the performance of these models, we utilized a variety of metrics:

- Accuracy: This metric quantifies the proportion of correctly predicted instances out of all instances. It is a straightforward measure and does not require any assumptions about the distribution of the classes.
- Precision: This metric measures the proportion of true positives out of all instances predicted as positive. It indicates the accuracy of the positive predictions.
- Recall: This metric measures the proportion of true positives out of all actual positive instances. It indicates the proportion of actual positive instances that were correctly predicted.

Based on the above metrics the following models had high metric values:

1. SVM with an overall accuracy of 71%.
   - Class negative: precision of 87%, recall of 19%
   - Class No emotion: Precision of 72% with a recall of 90%
   - Class Positive: precision of 65% with a recall of 49%
2. Logistic Regression with an overall accuracy of 68%.
   - Class negative: precision of 59%, recall of 11%
   - Class No emotion: Precision of 72% with a recall of 86%
   - Class Positive: precision of 59% with a recall of 50%

The following models performed poorly:

1. Decision Trees with overall accuracy of 61%
2. GRU with a overall accuracy of 64%

CONCLUSION, RECOMMENDATION AND NEXT STEPS

Conclusion

We were able to use various models in the study. We were able to integrate the concept of deep learning in this project.

The models performed well on the training data but not on the test data. This could be due to the imbalance in the dataset, which affects the model's performance.

Recommendations

Based on the accuracy of the models, we recommend that the SVM is the appropriate choice, with 71% accuracy, to use to rate the sentiment of a tweet based on its content.

Limitations

The model struggled to classify positive sentiment tweets as compared to other classes.

Identifying slang words and urban abbreviations to include in sentiment analysis. This will help to incorporate different generations especially the Gen Z and Gen Aplha who heavily use twitter.

Overfitting of our data.

Next Steps

1. Incorporate slang words and urban abbreviations to include to improve sentiment analysis.
2. Implement real-time sentiment monitoring for Twitter to detect shifts in public sentiment.
3. Regularly assessing the performance of our recommended model, incorporate new data as it becomes accessible, and fine-tune it in response to evolving business requirements and customer behaviors.