A gauge with different faces

Description automatically generated

**Sentiment Analysis on Twitter**

Business Analytics & Insights Capstone, Centennial College

Report presented by: Sudish Basnet

Table of Contents

[Abstract 4](#_Toc143373161)

[1. Executive Summary 4](#_Toc143373162)

[1.1 Executive Introduction 4](#_Toc143373163)

[1.2 Executive Objective 4](#_Toc143373164)

[1.3 Executive Model Description 4](#_Toc143373165)

[1.4 Executive Recommendations 5](#_Toc143373166)

[2. Introduction 5](#_Toc143373167)

[2.1 Background 5](#_Toc143373168)

[2.2 Problem Statement 6](#_Toc143373169)

[2.3 Objectives 6](#_Toc143373170)

[2.4 Challenges 7](#_Toc143373171)

[2.5 Measurement Metrics 7](#_Toc143373172)

[3. Methodology 8](#_Toc143373173)

[4. Dataset 9](#_Toc143373174)

[4.1 Data Set Introduction 9](#_Toc143373175)

[4.2 Exclusions 10](#_Toc143373176)

[4.3 Data Exploration 10](#_Toc143373177)

[4.4 Data Cleansing and Tweets Pre-Processing 15](#_Toc143373178)

[5. Data Preparation and Feature Engineering 18](#_Toc143373179)

[5.1 Data Preparation Needs 18](#_Toc143373180)

[5.2 Feature Engineering 18](#_Toc143373181)

[5.2.1 Extracting Sentiments 18](#_Toc143373182)

[5.2.2 Sampling 21](#_Toc143373183)

[5.2.3 Label Encoding 21](#_Toc143373184)

[5.2.4 Extracting Features from Tweets 21](#_Toc143373185)

[6. Modelling 22](#_Toc143373186)

[6.1 Modeling Approach/Introduction 22](#_Toc143373187)

[6.2 Models 23](#_Toc143373188)

[6.3 Model Comparison and Conclusion 24](#_Toc143373189)

[6.4 Logistic Regression: Using all meta data features (Optional) 27](#_Toc143373190)

[7. Inference 28](#_Toc143373191)

[7.1 Optimal Features 28](#_Toc143373192)

[7.1.1 Optimal Features for Ideal Model 28](#_Toc143373193)

[7.1.2 Optimal Features for Experimental Model with all meta data features 28](#_Toc143373194)

[7.2 Application 29](#_Toc143373195)

[7.3 Recommendation 31](#_Toc143373196)

[7. Conclusion 31](#_Toc143373197)

[8. References 32](#_Toc143373198)

[Figure 1. Sequential Phases for Sentiment Analysis 9](#_Toc143369968)

[Figure 2. Data Dictionary 10](#_Toc143369969)

[Figure 3. Correlation Matrix of Dataset Attributes 11](#_Toc143369970)

[Figure 4. Data Distribution for Majority Target 12](#_Toc143369971)

[Figure 5. Distribution of majority target based on total credit for tweets. 12](#_Toc143369972)

[Figure 6. Relation between majority target and replies 13](#_Toc143369973)

[Figure 7. Relation between favorites and mentions. 13](#_Toc143369974)

[Figure 8. Relation between tweets a user has and number of followers a user has. 14](#_Toc143369975)

[Figure 9. Relation between friends a user has and followers. 15](#_Toc143369976)

[Figure 10. Raw Tweet vs Clean Tweet 17](#_Toc143369977)

[Figure 11. Word Cloud of Tweets with Noise 17](#_Toc143369978)

[Figure 12. Word Cloud for Clean Tweets 18](#_Toc143369979)

[Figure 13. Polarity Score 19](#_Toc143369980)

[Figure 14. Positive Tweets Word Cloud for Trending Topics 20](#_Toc143369981)

[Figure 15. Negative Tweets Word Cloud for Trending Topic 20](#_Toc143369982)

[Figure 16. Neutral Tweets Word Cloud for Trending Topic 21](#_Toc143369983)

[Figure 17. Sentiment Distribution across Dataset 22](#_Toc143369984)

[Figure 18: Accuracy Comparison for Feature Extraction Methods 23](#_Toc143369985)

[Figure 19. ML Model Comparison using Accuracy. 25](#_Toc143369986)

[Figure 20. ML Model Comparison using F1-Score 26](#_Toc143369987)

[Figure 21. ML Model Comparison using ROC-AUC SCORE 27](#_Toc143369988)

[Figure 22. Ideal Model Confusion Matrix and ROC Curve 28](#_Toc143369989)

[Figure 23. Confusion Matrix and ROC Curve for Experimented Model 28](#_Toc143369990)

[Figure 24. Optimal Features from Ideal Model 29](#_Toc143369991)

[Figure 25. Optimal Features from Experimented Model 30](#_Toc143369992)

[Figure 26. Application of Sentiment Analysis for Trend Analysis (Trend Line and Distribution of Sentiments) 30](#_Toc143369993)

[Figure 27. Application of Sentiment Analysis for Trend Analysis (Extracting Tweets) 31](#_Toc143369994)

[Figure 28. Application of Sentiment Analysis for Trend Analysis (Word Cloud) 32](#_Toc143369995)

# Abstract

This project conducts sentiment analysis on Twitter data, specifically targeting tweets to explore public sentiments towards diverse topics, events, brands, and more. Employing natural language processing techniques and machine learning algorithms, insights are extracted from a substantial dataset comprising over 134,198 meta data. The investigation highlights the significance of processed text compared to raw text when training machine learning models for sentiment analysis. Various feature engineering methods are explored, along with diverse statistical evaluation metrics. The report outlines the methodology, presents results, and emphasizes the role of sentiment analysis in real-time comprehension of public opinions.

**Keywords** – Tweets, *Sentiment Analysis, Machine Learning, Natural Language Processing, NumPy, Pandas, NLTK, NEATEXT, Regex, Seaborn, Sklearn, Vectorizer, Bag of Words, TF-IDF, Hashing, Tokenization, VADER Sentiment Analyzer, Polarity Score, Grid Search CV, Logistic Regression, Multinomial Naïve Bayes, Linear SVC, Random Forest, and K-Nearest Neighbor.*

# 1. Executive Summary

## 1.1 Executive Introduction

In an era where social media has become an integral part of modern communication, understanding public sentiments is of paramount importance. This executive summary provides a concise overview of our project that focuses on sentiment analysis using Twitter data. By leveraging natural language processing techniques and machine learning algorithms, we aimed to uncover insights from the vast ocean of tweets to gain valuable understanding about public perceptions, emotions, and reactions.

## 1.2 Executive Objective

The primary objective of this project was to perform sentiment analysis on Twitter data and decipher the underlying sentiments expressed by users. Our goal was to create a model that could accurately classify tweets into positive, negative, or neutral sentiments, allowing us to gain insights into how various topics, events, and brands are perceived by the Twitter community.

## 1.3 Executive Model Description

To achieve our objective, we employed a combination of data collection, preprocessing, and sentiment analysis techniques. Our approach included:

* **Data Collection:** We gathered a diverse dataset of tweets from CIC Truth Seeker, spanning different topics.
* **Preprocessing:** We cleaned and prepared the text data by removing noise, stop words, special characters, and many more.
* **Sentiment Analysis Techniques:** We explored both lexicon-based approaches and machine learning models to classify tweets' sentiments. Lexicon-based methods utilized sentiment dictionaries, while machine learning models utilized features extracted from the text data and other numerical features as applicable.

## 1.4 Executive Recommendations

Based on our analysis, we offer the following recommendations:

* **Real-time Tracking:** Utilize the developed sentiment analysis model to monitor public sentiments in real-time during major events, product launches, or public discussions. This can provide valuable insights into immediate reactions and trends.
* **Brand Perception Analysis:** Apply sentiment analysis to understand how your brand or product is being discussed on Twitter. This can help in adapting marketing strategies, addressing concerns, and enhancing customer engagement.
* **Sentiment-Driven Decision Making:** Incorporate sentiment analysis results into decision-making processes. Understanding public sentiments can guide policy adjustments, communication strategies, and resource allocation.

In conclusion, our sentiment analysis project on Twitter provides a novel and effective approach to understanding the dynamic landscape of public opinions. By harnessing the power of NLP and machine learning, we shed light on the sentiments surrounding various subjects. This executive summary offers a glimpse into the full project's methodologies, findings, and recommendations, which we explore in detail throughout this report.

# 2. Introduction

## 2.1 Background

Social media platforms like Twitter have become a significant source of real-time data, providing insights into public sentiment, opinions, and trends. Twitter being a microblogging platform where users post short messages (tweets) about various topics in real time. This makes it a valuable source of up-to-date information and opinions, making it useful for tracking current trends and events.

Sentiment analysis, on the other hand, also referred to as opinion mining, encompasses the utilization of techniques from natural language processing and machine learning to analyze and determine the emotional tone or sentiment expressed in textual data, such as tweets.

## 2.2 Problem Statement

In the era of digital communication and social media, understanding public sentiment and opinions is crucial for businesses, organizations, and researchers alike. The vast volume of user-generated content on platforms like Twitter provides a unique opportunity to gain insights into people's emotions and perceptions, which is not possible to deal with manually by segregating each individual piece of text and labelling their sentiment with the human mind. Thereby, our project aims to tackle the challenge of sentiment analysis on Twitter, specifically focusing on classifying tweets into three categories: positive, neutral, and negative, using natural language processing complemented by machine learning techniques.

Below are the applications that can be enhanced by our project:

* **Brand Management**: Companies need to gauge how their products or services are perceived by customers. Understanding sentiment helps them respond to feedback, identify areas for improvement, and enhance customer satisfaction.
* **Political and Social Trends**: Monitoring public sentiment towards political events, social issues, or policies can provide insights into prevailing opinions and potential shifts in attitudes.
* **Market Research**: Analyzing sentiment across product categories helps businesses anticipate market trends and design strategies accordingly.
* **Public Opinion Analysis**: Researchers and policymakers can benefit from understanding the sentiment of the public on various topics, guiding decision-making processes.

## 2.3 Objectives

The main goal of conducting sentiment analysis on Twitter is to extract and comprehend the emotional tone conveyed in tweets. It requires categorizing tweets into positive, negative, or neutral sentiments, aiming to gather insights into public sentiment, customer feedback, brand image, and evolving patterns. To achieve those goals, below are the steps we will follow:

* Develop and fine-tune a sentiment analysis model capable of accurately categorizing tweets into positive, neutral, and negative sentiment classes.
* Preprocess and clean the tweet data to mitigate noise and enhance the quality of the input features for the model.
* Address the challenges posed by contextual understanding, noise, and multilingual data through appropriate preprocessing and feature engineering techniques.
* Assess the model's effectiveness in sentiment classification by measuring its performance using different evaluation metrics.
* Provide insights into sentiment distribution, trends, and correlations within the analyzed tweet dataset.

## 2.4 Challenges

The problem of sentiment analysis on Twitter presents several challenges:

* **Textual Noise**: Tweets are often riddled with slang, abbreviations, emojis, and grammatical errors, making accurate sentiment analysis more difficult.
* **Contextual Understanding**: Understanding sarcasm, irony, and nuanced emotions within the limited context of a tweet is a complex task.
* **Ambiguity**: Some tweets might be inherently neutral, while others could be interpreted as either positive or negative, depending on the reader's perspective.
* **Data Imbalance**: The distribution of positive, neutral, and negative tweets might not be balanced, potentially leading to biased model performance.
* **Topic Variability**: Different topics and hashtags can alter the sentimental interpretation of certain words or phrases.
* **Memory Allocation and Computational Power**: Training models with tweet features (Features extracted from text for machine learning models) and additional features (attributes of the user profile like likes, favorites, and many more) sums up the data size of more than 80k features. With the increase in data size, the need for computational power and memory capacity increases.

## 2.5 Measurement Metrics

To assess the performance and effectiveness of the sentiment analysis model applied to Twitter data, several measurement metrics will be employed. These metrics will quantify the model's ability to accurately classify tweets into positive, neutral, and negative sentiments.

Initially, we will evaluate our models with an accuracy score, and later followed by F1-score (Precision and Recall included), Compound Matrix, and AUC-ROC Curve in different thresholds.

1. **Accuracy Score:** Accuracy is a commonly used metric in classification tasks. It measures the proportion of correctly classified instances out of the total instances in a dataset. Mathematically, accuracy is calculated as:

Accuracy = (Number of Correct Predictions / Total Number of Predictions) ×100%

While accuracy is straightforward to understand, it might not be the best metric to use when dealing with imbalanced datasets where one class is much more prevalent than the other, so we will use other metrics too.

1. **Precision, Recall, and F1-Score:** Precision and recall are metrics used in binary classification tasks to assess the quality of a model's predictions. Precision measures the proportion of true positive predictions (correctly predicted positive instances) out of all positive predictions made by the model. Recall measures the proportion of true positive predictions out of all actual positive instances in the dataset. However, F1-score is the harmonic mean of precision and recall. It provides a balanced measure that considers both false positives and false negatives. F1-score is especially useful when the class distribution is imbalanced.

F1-Score = 2 × (Precision + Recall) / (Precision × Recall)

1. **Confusion Matrix:** A confusion matrix is a table used to evaluate the performance of a classification algorithm. It summarizes the number of correct and incorrect predictions made by the model on each class. A typical confusion matrix for binary classification consists of four values:
   1. True Positives (TP): Correctly predicted positive instances.
   2. True Negatives (TN): Correctly predicted negative instances.
   3. False Positives (FP): Incorrectly predicted positive instances (Type I error).
   4. False Negatives (FN): Incorrectly predicted negative instances (Type II error).
2. **AUC-ROC Curve:** The AUC-ROC curve (Area Under the Receiver Operating Characteristic Curve) is a graphical representation that shows the performance of a binary classification model at various thresholds. It plots the True Positive Rate (Recall) against the False Positive Rate as the threshold for classification changes. AUC-ROC provides a measure of the model's ability to distinguish between the two classes. A higher AUC indicates better discrimination ability, with a value of 1 representing a perfect classifier and 0.5 indicating random guessing.

# 3. Methodology

Before starting the implementation of sentiment analysis on Twitter, we set out some of the processes that we should align with for the effectiveness of the project. After a thorough analysis and planning, we divided our implementation processes into the following linear sequential phases, as illustrated below, and the last two process of extracting features from tweets and modelling include several iterative processes to get the best feature extraction method and the best model to predict.

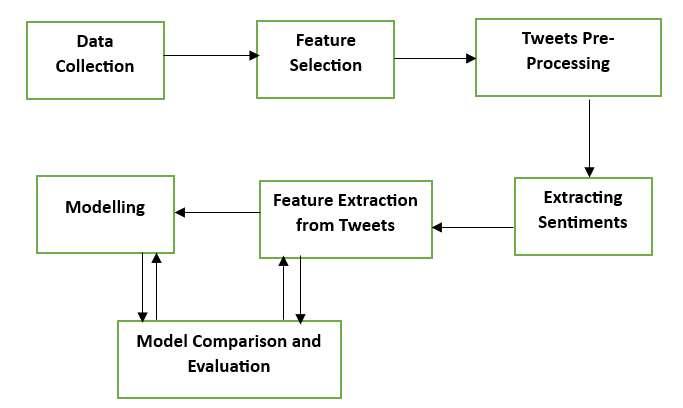


Figure 1. Sequential Phases for Sentiment Analysis

# 4. Dataset

## 4.1 Data Set Introduction

We sourced our dataset from CIC Truth Seeker Dataset 2023 (CIC, 2023). The dataset was previously analyzed by the Canadian Institute of Technology with the scope of finding the truthiness of the tweets based on several models and practices. We will further advance the analysis by contemporizing it with sentiment analysis.

Our dataset in total has 1,34,198 of records and 64 features. The feature attributes consist of 4 categories: Text Features, Lexical Features, Meta Data Features, and CIC populated features. Below is the list of attributes from our dataset:

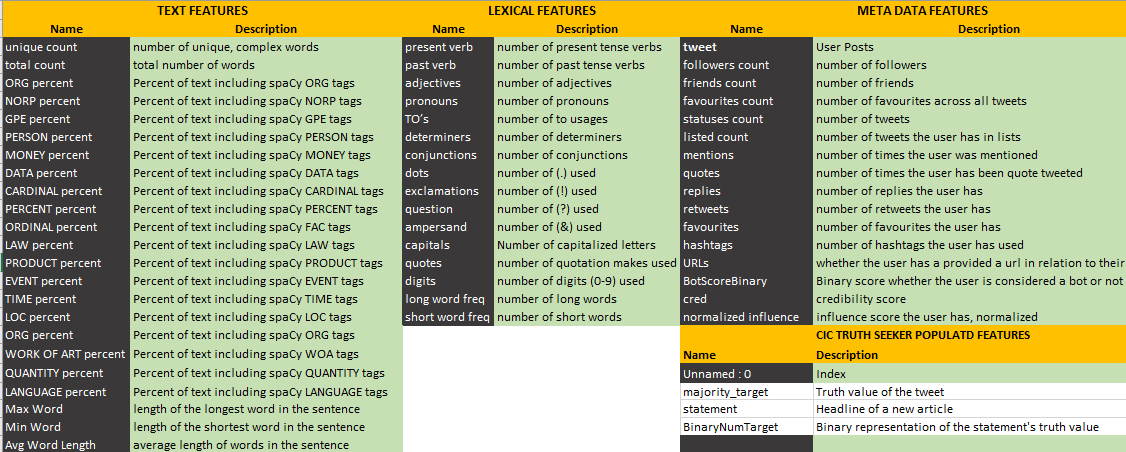


Figure 2. Data Dictionary

## 4.2 Exclusions

Out of these 4 categories of attributes in our dataset, we will be excluding the first 2 category attributes completely, which will be later discovered in our analysis process, and we will be working on the meta data features, especially tweets for the ideal model and the rest of the features for future work. From the metadata features, we will be excluding URLs, Hashtags, and Quotes due to their low importance for our modelling. Although the presence of URLs, hashtags, and quotes in tweets matters to know the sentiment but we don’t need the numbers of these features in tweets. Besides this, we will ignore unnamed: 0, statement, and BinaryNumTarget from the CIC populated features, as sequence does not matter in our dataset and the other two attributes don’t play a significant role for our model.

## 4.3 Data Exploration

In this section, we will explore the attributes that we are left with after the first phase of exclusion. As we choose only the applicable attributes for our modelling, we are left with 134198 rows × 11 columns. The 11 columns are: majority target, tweet, favorites, friends count, statuses count, listed count, mentions, replies, retweets, followers count, and cred.

1. **Uniqueness of every attributes**

In consideration of uniqueness in our dataset, as we explored below is the result for our dataset. As we are more focused towards tweets, having every tweet unique is very helpful for our model.

|  |  |  |  |
| --- | --- | --- | --- |
| majority\_target | 2 | listed\_count | 1979 |
| tweet | 134198 | mentions | 13 |
| favourites | 936 | replies | 273 |
| friends\_count | 10058 | retweets | 540 |
| statuses\_count | 49323 | cred | 81684 |
| Followers\_count | 14444 | | |

1. **Correlation Map of all attributes**

Below is the correlation matrix for our dataset, and it is clearly visible that our 4 features (retweets and favorites, followers\_count, and listed\_count) are highly corelated. We can get rid of retweets and listed\_count as they are less unique, and we can save some memory with less computation power to be used. Other than these other features are not much correlated.

A screenshot of a computer

Description automatically generated

Figure 3. Correlation Matrix of Dataset Attributes

1. **Data distribution for Majority target**

The majority target was the variable populated by CIC to know whether the tweet was true or not. The distribution looks to be normally distributed (68985 true and 65213 false), which is a good sign for future modelling.

Figure 4. Data Distribution for Majority Target

1. **Distribution of majority target based on total credit for tweets.**

The pie chart below shows that the credit for true tweets is higher than the credit for false tweets. This may be due to a small difference in the number of true tweets compared to false tweets.

A green and red circle with black text

Description automatically generated

Figure 5. Distribution of majority target based on total credit for tweets.

1. **Relation between majority target and replies:**

The relationship below shows the total number of replies for true tweets and false tweets. It clearly shows that people are replying to false tweets more than true tweets. This proves that people get more engaged in false news than the true news.

A red and green squares

Description automatically generated

Figure 6. Relation between majority target and replies

1. **Relation between favorites and mentions.**

From the scatter plot below, we can see that most users are mentioned many times, but they have the fewest number of favorites tweets in their account. The relationship shows that, in most cases, the higher a person is mentioned, the lesser the chance that the user will have some favorites.

A graph with blue dots

Description automatically generated

Figure 7. Relation between favorites and mentions.

1. **Relation between tweets a user has and number of followers a user has.**

From the scatter plot, we can see that most of the users have fewer followers and fewer tweets. But as in previous relationships, in this one too we can see for most of the cases, the higher the tweets a user has the lesser the follower a user has and vice versa.

A graph with blue dots

Description automatically generated

Figure 8. Relation between tweets a user has and number of followers a user has.

1. **Relation between friends a user has and followers.**

Beside one outlier and a few groups, we can visualize that for most of the cases, the number of friends is inversely proportional to the number of followers a user has. To point this out, most of the users have more friends than followers.

A graph with numbers and lines

Description automatically generated

Figure . Relation between friends a user has and followers.

## 4.4 Data Cleansing and Tweets Pre-Processing

Data cleansing, also known as data cleaning or data scrubbing, is the process of identifying and correcting or removing errors, inconsistencies, inaccuracies, and discrepancies in a dataset. This process is crucial for ensuring that the data used for analysis, modelling, or any other purpose is accurate, reliable, and consistent. Data cleansing involves a series of tasks like handling missing values, removing duplicates, correcting inaccuracies, handling outliers, data validation, and many more. Through EDA, we were able to see some outliers, but for the time being, we won’t be handling those due to time restrictions, but it will be the next step after modelling our ideal model. We don’t have any missing values, so we don’t have to worry about that.

Tweet pre-processing refers to the steps taken to clean, transform, and prepare raw tweet data for analysis. Due to the unique characteristics of Twitter text data, such as brevity, slang, hashtags, and mentions, pre-processing is essential to make the text suitable for natural language processing (NLP) tasks like sentiment analysis, topic modelling, and more. Within the range of 100000+ words that we are assuming for our tweets, it can be difficult to focus on only popular or trending words, so tweet preprocessing steps will help us maintain visual appearance and can even be efficient for later modelling.

We will be using the Neattext library for our pre-processing steps. Neattext is a Python library designed to simplify and enhance the preprocessing of text data. It provides a collection of functions that allow you to clean, normalize, and preprocess text in a streamlined and efficient manner.

Common tweet pre-processing steps that we followed for our tweets include:

1. **Lowercasing:** Converting all text to lowercase to ensure consistent treatment of words.
2. **Tokenization:** Splitting text into individual words or tokens for analysis.
3. **Removing Punctuation:** Removing punctuation marks (e.g., ‘’,?!;:`) that do not contribute to the analysis.
4. **Removing Stop Words:** Filtering out common words (e.g., and the, is) that do not carry significant meaning. We have 133590 stop words in our tweets.
5. **Removing Short Words:** Short words are not always useful so depending upon the project, we can exclude them. This can also help to remove some typo errors or some other words from another language and maybe slang. We have 99717 short words in our tweets.
6. **Handling Mentions and Hashtags:** Extracting mentions (@user) and hashtags (#topic) to understand their significance and removing it as necessary. We have 15349 hashtags and 97630 user handles.
7. **Handling URLs:** Removing or replacing URLs with a placeholder so that it won’t make more mess. Mostly the URLs are not very useful in sentiment analysis, however hashtags and mentions can be so we will be considering them as per the necessity in our analysis process. We just have 10 URLs in our tweets.
8. **Emoticon and Emoji Handling:** Converting emoticons and emojis to textual representations. Although, to ensure the integrity we have established the code for extracting and removing emojis, but we currently don’t have any in our tweets.
9. **Removing Numbers:** Depending on the analysis, numbers might be removed or retained. We are very concerned with the data size and memory our dataset uses, thereby we will not be using numbers as our features.
10. **Dealing with Special Characters:** Handling special characters that are not part of standard language. Special characters includes: %,#,\n,\* etc. We have 47814 special characters in our tweets.

After this pre-processing process, our clean tweets look like this:

A screenshot of a computer

Description automatically generated

Figure 10. Raw Tweet vs Clean Tweet

Below is the word cloud of our tweet’s words before cleaning. You can see words like amp, one, go, still, and many more. Because of such words, it will overlap with the other important words.

A close-up of words

Description automatically generated

Figure 11. Word Cloud of Tweets with Noise

Our word cloud for clean tweets is shown below. This will help us to focus more on trending topics.

A close-up of words

Description automatically generated

Figure . Word Cloud for Clean Tweets

# 5. Data Preparation and Feature Engineering

## 5.1 Data Preparation Needs

Effective data preparation helps us create a structured and feature-rich dataset that can be used to train machine learning models for sentiment analysis on Twitter data. The quality of our data and preprocessing steps directly impacts the model's ability to capture sentiment patterns and make accurate predictions. Until now, aside from meta data features and CIC-populated data features, we needed to derive our target variable (sentiments) and further use tweets to get the features out of it. So, the data preparation needs are a must, and we will do this with feature engineering.

## 5.2 Feature Engineering

Feature engineering is the process of creating, selecting, transforming, and enhancing features (input variables) from raw data to improve the performance and interpretability of machine learning models. Features are the attributes or characteristics of the data that the model uses to make predictions or classifications. Effective feature engineering involves making the most relevant and informative features available to the model, enabling it to learn meaningful patterns and relationships from the data.

### 5.2.1 Extracting Sentiments

We used the VADER sentiment analyzer to measure the level of sentiment through polarity and compound scores. VADER (Valence-Aware Dictionary and Sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool specifically designed for analyzing sentiments expressed in text. It's particularly effective for social media text due to its ability to handle informal language, slang, and context-specific expressions. VADER assigns sentiment scores to text, indicating the intensity of positive, negative, and neutral sentiments. We will pass raw tweets as an argument so that the sentiment will be accurately analyzed. We decided on a cutoff score of +-0.3, where tweets with a score less than -0.3 will be labelled negative, tweets with a score higher than 0.3 will be labelled positive, and others will be labelled neutral. Below is the output generated from VADER.

A screen shot of a computer code

Description automatically generated

Figure . Polarity Score

After setting the sentiment, we have created a 3-word cloud denoting positive, negative, and neutral tweets. Below is a word cloud of positive tweets with trending words. Biden, Trump, Vaccine, Wage, Support, and many other trending topics can be seen.

A close-up of words

Description automatically generated

Figure 14. Positive Tweets Word Cloud for Trending Topics

Below is the word cloud for negative sentiment tweets. The most popular topics again include Biden, Trump, and Vaccine. Additionally, we can see Terrorist, Death, Overdose, Suicide, and many more.

A black background with red and white text

Description automatically generated

Figure 15. Negative Tweets Word Cloud for Trending Topic

To look at the neutral sentiments word cloud, we can filter the most trending topics like American, Wage, Time, State, and many more. However, we can see the same 3 trending topics here too like in the other sentiment word cloud.

A close-up of words

Description automatically generated

Figure . Neutral Tweets Word Cloud for Trending Topic

### 5.2.2 Sampling

As we are done with our derived variable (sentiment), to measure the distribution of our target class, we created a pie chart, which is demonstrated below. As you can see, the negative class has a higher number of presences. To avoid future bias, we resampled our dataset based on the basic target variable. We can resample using different available functions, but we did it manually by setting the percentage of negative sentiment data from 41% to 30%.

Figure 17. Sentiment Distribution across Dataset

### 5.2.3 Label Encoding

Machine learning algorithms work best with numerical data, so label encoding is used to convert categorical labels into numerical values so that algorithms can process them effectively. As our target variable has ordinal data, label encoding is best for that. We labelled our target variable as 0 (negative), 1 (neutral), and 2 (positive).

### 5.2.4 Extracting Features from Tweets

After doing all the necessary steps, the most important step in the implementation is to extract features from the tweets. Although we can predict the sentiments with the other meta data features beside tweets, we will consider tweets for now and in the future. We can include other attributes as well for better analysis.

To extract the features, we consider comparing three methods and choose one of the best methods after evaluating the validation data. At this phase, we have considered two strategies:

1. Separated 4000 data as validation and testing for experimenting with different methods and later using it for the ML model’s hyperparameter tuning.
2. Use raw and clean data with all the methods and finalize one dataset to be carried forward.

The three methods are Bag of Words (BoW), TF-IDF, and Hashing. These are techniques used for text representation in natural language processing. BoW represents a document as a frequency vector of its constituent words, ignoring word order. TF-IDF (Term Frequency-Inverse Document Frequency) considers not only word frequencies but also the importance of words in a document relative to their prevalence across all documents. Hashing uses hash functions to map words to fixed-size vectors, enabling efficient storage and processing of text data. BoW simplifies text to word counts, TF-IDF captures word relevance, and Hashing provides dimensionality reduction, each catering to different needs in text analysis and machine learning. The result of the evaluation is illustrated below.

Figure 18: Accuracy Comparison for Feature Extraction Methods

**Conclusion**: The bag of words method generated a higher accuracy of 61.50%, and on top of that, a raw dataset was used for that. From now on, we need to use a clean dataset to visualize the scenarios and a raw dataset to test and train models. If we ever need to change the method for feature extraction, then we can consider TF-IDF, as the accuracy was 59.75% with the raw dataset. However, hashing didn’t perform well on both datasets. Nevertheless, some noises like numbers, special characters, and URLs are excluded from raw tweets to reduce some dimension and help the models predict better targets.

# 6. Modelling

## 6.1 Modeling Approach/Introduction

The dataset is finally ready to be modelled with five different classification algorithms to predict the target variable. Until now, the dataset has been divided into two categories.

1. Dataset with just tweets feature and target variable (sentiment)
2. Dataset with excluded meta-data features, tweets feature, and target variable.

For the ideal model, the dataset with only tweets will be used because there will be a memory allocation problem to train every dataset with additional features with different properties. However, to provide a starting point for further work, one model will be trained and evaluated with every attribute after knowing the ideal model first.

For hyper tuning the parameters of different machine learning models, the grid search technique is used. The first five model techniques are compared to get the ideal model, and the last model technique is an experiment for further work.

## 6.2 Models

The five models that will be trained for sentiment analysis are:

1. **Logistic Regression**: Logistic Regression is a linear classification algorithm that models the probability of a data point belonging to a particular class. It's suitable for binary and multi-class classification tasks. We extended Logistic Regression to handle multi-class classification problems through Softmax Regression technique (also known as "Multinomial Logistic Regression"). **With the help of grid search, the best regularization parameter “C” for the model in our dataset is 100.**
2. **Multinomial Naive Bayes**: Multinomial Naive Bayes is a probabilistic algorithm that's well-suited for text classification and other categorical data. The algorithm estimates the probabilities of features occurring in each class and calculates the likelihood of a class given the features, making the final prediction based on the class with the highest likelihood. With the help of grid search, the best value for alpha is 1.
3. **Linear Support Vector Classifier (Linear SVC)**: Linear SVC is a linear classification algorithm that seeks to find the hyperplane that best separates different classes while maximizing the margin between the classes. The algorithm transforms the data into a higher-dimensional space to find a linear hyperplane that separates the classes as well as possible. With the help of grid search, the best regularization parameter “C” is found to be 0.05.
4. **Random Forest**: Random Forest is an ensemble learning algorithm that combines multiple decision trees to make predictions. For multi-class classification, it constructs a forest of decision trees, each trained on a random subset of the data and a random subset of features. The final prediction is the majority vote from the individual trees. With the help of grid search, the best value for maximum depth is found to be 30 and the number of estimators to be 150.
5. **k-Nearest Neighbors (KNN)**: KNN is a non-parametric and instance-based classification algorithm. Given a new data point, it finds the k closest training data points in terms of feature similarity and assigns the most common class among these k neighbors to the new point. With the help of grid search, the best value for number of neighbors is found to be 2, p to be 1 (Manhattan Distance), and weights to be distance.

## 6.3 Model Comparison and Conclusion

After training models with 80% of the training data and testing them with 20% of the testing data, models are initially compared by accuracy score to determine the ideal model out of all. As accuracy score cannot alone decide the best model, all models are compared on the basis of F1 score, confusion matrix, and ROC curve.

Below is the result for the comparison of models based on accuracy score. It is justifiable that logistic regression is the best, followed by linear SVC. This means that if we need to use any other model than logistic regression in the future, we can consider linear SVC. However, KNN is doing worse than all of them.

Figure 19. ML Model Comparison using Accuracy.

To compare the models based on F1-Score, below is the bar chart for the outcomes after evaluation at a fixed threshold for all 3 classes. As expected, logistic regression and linear SVC stand out. Both models are equally accurate at predicting negative classes; however, logistic regression can predict neutral and positive classes by 1% more accurately. Our target variables are least favoured by the KNN model, but multinomial Nave Bayes and random forests can be considered if we ever wonder to experiment.

Figure 20. ML Model Comparison using F1-Score

Finally, with the evaluation of the models by ROC-AUC score at different thresholds, logistic regression is the ideal model, with a score of 0.92 for the negative class, 0.83 for the neutral class, and 0.91 for the positive class. Nevertheless, linear SVC is also giving outstanding performance with an AUC core of 0.92 for negative classes, 0.81 for neutral classes, and 0.90 for positive classes. Surprisingly, Random Forest’s performance is better predicted by the curve, with a 0.87 score for the negative class, 0.76 for the neutral class, and 0.86 for the positive class. But the best options for modelling are either logistic regression or multinomial Nave Bayes for our case.

Figure 21. ML Model Comparison using ROC-AUC SCORE

**Conclusion**: With the evaluation technique using three metrics, it is justifiable that logistic regression and linear SVC are predicting the target variable more accurately. Beside this, KNN is the worst model; however, multinomial Nave Bayes and Random Forest are decently predicting the sentiments.

Generally, logistic regression is computationally less expensive, making it suitable for larger datasets. It's a simpler algorithm compared to SVC. Logistic regression provides clear and interpretable results. The coefficients of the features directly show the impact and direction of each feature on the outcome. And importantly, it has always acquired higher accuracy by some percent than SVC; our ideal model is logistic regression.

Below are the ROC curve and confusion matrix of our ideal model. Our model is best at predicting negative classes, followed by positive classes and neutral classes. In a fixed threshold, the ideal model predicted 26.23% target true for the negative class out of 33%, 23.90% for neutral, and 24.64% for positive out of 33%. Beside this, the ROC curve shows the predictions are way more accurate than the random ones, with 92% accuracy for negative classes, 83% for neutral classes, and 91% for positive classes at different thresholds.

A graph of a function

Description automatically generated with medium confidenceA graph of confusion matrix

Description automatically generated

Figure 22. Ideal Model Confusion Matrix and ROC Curve

## 6.4 Logistic Regression: Using all meta data features (Optional)

This model is optionally built to initiate further work and improvements for our models while using all the other meta-data features like followers, retweets, and others. Training this logistic regression model with the ideal hyperparameter and 80% of all available data was not possible due to memory allocation problems and computational power. But we trained this model with just 10,000 pieces of available data. The accuracy and ROC curve were worse than our worst model. The model generated an accuracy of 33.1%, and the AUC score was parallel to the random, which is not good.

A graph of a line graph

Description automatically generatedA graph of confusion matrix

Description automatically generated

Figure 23. Confusion Matrix and ROC Curve for Experimented Model

# 7. Inference

## 7.1 Optimal Features

### 7.1.1 Optimal Features for Ideal Model

Below is the bar graph for the top 15 optimal features from our ideal model. Selecting the right set of features is crucial for achieving high performance and efficiency in these tasks. Died, Death, Great, and Like are the top 4 features from the tweets to predict the right sentiments.

A graph of different colored bars

Description automatically generated

Figure 24. Optimal Features from Ideal Model

### 7.1.2 Optimal Features for Experimental Model with all meta data features

As we explored more of the additional features with all the meta data features, the optimal feature list is different than the ideal model one. Although the dataset was not as huge as the dataset for the ideal model, this experiment gave an idea of the influence of the meta-data features on predictions. Died, Death, and Like are the topmost optimal features; however, it can be clearly seen that features like friends count, followers count, cred, and status count are in the list of the top 15 optimal features.

A graph of different colored bars

Description automatically generated with medium confidence

Figure 25. Optimal Features from Experimented Model

## 7.2 Application

Out of many ways, one of the ways a company can use this sentiment analysis is to watch the trend for particular popular topics. For example, Biden, Trump, Vaccine, and Marijuana are the three most popular topics. From the trend below, Trump has the most tweets with negative sentiment. If he wants to re-elect or wants to improve his following, he can dig more into this trend, as shown in the pie chart. After this, he can pinpoint those segments of 44.9% of tweet users to advertise for good things or take care of hot topics in between the negative tweets, or alternatively, he can go through each negative tweet.

A pie chart with numbers and a red circle

Description automatically generatedA graph with different colored lines

Description automatically generated

Figure 26. Application of Sentiment Analysis for Trend Analysis (Trend Line and Distribution of Sentiments)

If he wants to consider some tweets, then he can manually view as shown below:

A screenshot of a computer screen

Description automatically generated

Figure 27. Application of Sentiment Analysis for Trend Analysis (Extracting Tweets)

If he wants to see the popular topics, he can use this sentiment analysis for negative tweets using a word cloud, as shown below. The trending topics for negative tweets about Trump include words like Americans, eviction, moratorium, head, reconciliation, and many more.

A close up of words

Description automatically generated

Figure . Application of Sentiment Analysis for Trend Analysis (Word Cloud)

## 7.3 Recommendation

There are several recommendations for future work and potential avenues to explore. These recommendations can help build on the insights gained from the analysis and lead to a more comprehensive understanding and impactful actions.

1. **Fine-Tuning Models:** Investigate the possibility of fine-tuning sentiment analysis models specifically with the meta-data features. One of the insights we found during the experimental model training was the influence of the majority target, mentions, and cred in determining the sentiment. Further work on this project should include those features and include some fine-tuning for more accuracy and insights. Exclude data with nan’s if encountered during feature engineering or model trainings.
2. **Multilingual Sentiment Analysis:** Extend the sentiment analysis to include multiple languages, especially if the brand or topic has an international audience. Analyzing sentiments in different languages can provide a broader perspective.
3. **Aspect-Based Sentiment Analysis:** Move beyond overall sentiment and delve into aspect-based sentiment analysis. This involves analyzing sentiments towards specific features or attributes of a product, helping you identify which aspects are driving positive or negative sentiments.
4. **Temporal Analysis:** Explore how sentiments change over time. Analyzing sentiment trends can reveal patterns related to specific events, product launches, or marketing campaigns, helping you understand the impact of time on public perception.
5. **Computation Power and Memory Allocation:** Increase computational power and allocate memory to train our best model with the user data beside tweets, like the experimental model with all available data. This can save time for future work due to the device's efficient performance.

# 7. Conclusion

Twitter sentiment analysis is a challenging yet valuable task. Advancements in NLP and AI improve accuracy over time. In summary, the sentiment analysis conducted on Twitter tweets yielded valuable insights into public perceptions and sentiments. Both raw and clean data were tested, with raw data outperforming in terms of accuracy. However, clean data was chosen for visualization purposes while using raw data for model analysis. Among the feature engineering techniques applied to the tweets, the bag of words approach demonstrated superior performance.

The ideal model for sentiment classification was found to be logistic regression, achieving an accuracy of approximately 75%. This model excelled at predicting negative sentiments, followed by positive and neutral sentiments. This achievement, particularly in predicting negative sentiments accurately, aligned with the project's goals of enhancing brand image, gauging social and political trends, and extracting insights from popular topics.

Although the project achieved success with the ideal model utilizing tweets and sentiment alone, attempts were made to incorporate additional available metadata. However, the accuracy of these extended models was found to be limited, with an accuracy score of only 33.1%. This discrepancy suggests the complexity of integrating metadata effectively. Nevertheless, the potential for better accuracy through meticulous fine-tuning remains a promising avenue for future exploration.

Overall, the sentiment analysis endeavor not only provided a lens into the multifaceted sentiments within Twitter conversations but also demonstrated the importance of model selection, feature engineering, and data preprocessing in achieving accurate predictions. The insights gleaned from this analysis can contribute to brand image enhancement and a deeper understanding of prevailing social and political trends while illustrating the continuous journey of refining analytical methods for more comprehensive results.

# 8. References

1. **University of New Brunswick. (2023). *TruthSeeker 2023 dataset*.** Retrieved from <https://www.unb.ca/cic/datasets/truthseeker-2023.html>
2. **Toward Data Science. (n.d.). *Sentiment analysis: Concept, analysis, and applications*.** Retrieved from <https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17>
3. **Qualtrics. (n.d.). *Sentiment analysis: What it is and why it matters*.** Retrieved from <https://www.qualtrics.com/experience-management/research/sentiment-analysis/>
4. **Hootsuite. (n.d.). *Social media sentiment analysis tools: The ultimate guide*.** Retrieved from <https://blog.hootsuite.com/social-media-sentiment-analysis-tools/>
5. **Toward Data Science. (n.d.). *Logistic regression: Detailed overview*.** Retrieved from <https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc>
6. **DataTechNotes. (2020, July 6). *Classification example with LinearSVM in Python*.** Retrieved from <https://www.datatechnotes.com/2020/07/classification-example-with-linearsvm-in-python.html>
7. **Toward Data Science. (n.d.). *Grid search for model tuning*.** Retrieved from <https://towardsdatascience.com/grid-search-for-model-tuning-3319b259367e>
8. **Toward Data Science. (n.d.). *Understanding random forest*.** Retrieved from <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>
9. **UpGrad. (n.d*.). Multinomial Naive Bayes explained*.** Retrieved from <https://www.upgrad.com/blog/multinomial-naive-bayes-explained/>
10. **Analytics Vidhya. (2018, March 21). *Introduction to k-nearest neighbors algorithm & clustering*.** Retrieved from <https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/>