

TABLE OF CONTENTS

Inner first page.....	(i)
PAC form.....	(ii)
Declaration.....	(iii)
Certificate.....	(iv)
Acknowledgement.....	(v)
Table of Contents.....	(vi)

1. INTRODUCTION	1
1.1. SENTIMENT ANALYSIS	1
1.2. HISTORY OF SENTIMENT ANALYSIS	4
1.3. APPROACHES OF SENTIMENT ANALYSIS	5
1.3.1. LEXICON-BASED APPROACHES	5
1.3.2. MACHINE LEARNING BASED APPROACHES	6
1.3.3. HYBRID APPROACHES	7
1.4. STEPS INVOLVED IN SENTIMENT ANALYSIS	7
1.4.1. DATA GATHERING	9
1.4.2. DATA PRE-PROCESSING	9
1.4.3. DATA ANALYSIS	10
1.5. CORPORATE REPUTATION	10
1.5.1. IMPORTANCE OF CORPORATE REPUTATION	10
1.5.2. BUILDING CORPORATE REPUTATION	11
1.5.3. PROTECTING CORPORATE REPUTATION	12
2. PROFILE OF PROBLEM	13
3. EXISTING SYSTEM	14
4. PROBLEM ANALYSIS	15
4.1. PRODUCT DESCRIPTION	15
4.2. FEASIBILITY ANALYSIS	16
4.3. PROJECT PLAN	16
5. SOFTWARE REQUIREMENT ANALYSIS	17

6. DESIGN	18
6.1. SYSTEM DESIGN	18
6.2. DETAILED DESIGN	19
6.3. FLOW CHART	20
6.3. PSEUDO CODE	21
7. IMPLEMENTATION	27
7.1. DATA GATHERING	27
7.2. MODEL TRAINING	29
7.3. PREDICTING SENTIMENTS AND CALCULATING CORPORATE REPUTATION	33
8. PROJECT LEGACY	36
8.1. CURRENT STATUS OF PROJECT	36
8.2. REMAINING AREAS OF CONCERN	36
8.3. TECHNICAL AND MANGERIAL LESSONS LEARNT	37
9. USER MANUAL	39
10. SOURCE CODE	43
10.1. DATA MINING FROM TWITTER	43
10.2. TRAINING THE ML MODEL	45
10.3. PREDICTING SENTIMENTS AND CALCULATING CORPORATE REPUTATION	57
11. BIBLIOGRAPHY	64

1. INTRODUCTION

Sentiment analysis has emerged as a popular tool for businesses to keep track of their brand reputation on social media. By analysing the emotions and opinions expressed by customers online, brand managers can gain valuable insights that help inform their marketing decisions. The primary benefit of using sentiment analysis for brand reputation monitoring is that it allows businesses to track changes in customer sentiment towards their brand over time. By doing so, they can evaluate the effectiveness of their marketing strategies and make the necessary adjustments to maintain a positive brand image. In summary, sentiment analysis is a valuable tool in the business domain that helps in assessing the brand reputation and making informed marketing decisions. This project aims to develop a comprehensive model that can accurately assess the reputation of a business based on data obtained from tweets. The model will have the ability to analyse and evaluate the content of tweets in order to determine a company's overall reputation. The primary objective of this project is to provide a reliable tool that businesses can use to assess their public image. The secondary objective of the research is to evaluate the effects of the recent layoffs on the perception of the company in the public eye. Through this examination, the study aims to determine how the company's image and reputation have been influenced by the downsizing.

1.1. Sentiment Analysis

Sentiment analysis, also known as opinion mining, is the process of using natural language processing and machine learning techniques to extract and identify the sentiment or emotion behind a piece of text. This technique has become increasingly important in the age of big data, as businesses and organizations seek to understand and respond to customer feedback and opinions. Sentiment analysis can be used to monitor social media for customer sentiment about a brand, product, or service, to analyze customer reviews and feedback, and to track changes in public opinion over time. The ability to accurately analyze and understand sentiment is a valuable tool for businesses looking to stay ahead of the curve and meet the needs of their customers. Depending on the underlying attitude of a text, it often assigns a positive, negative, or neutral label to it.



Figure 1: Example of Sentiments

However, it can also be used to analyze more complicated emotions like rage, sorrow, happiness, etc.

And sentiment analysis is not limited to one field its applications are spread across many fields.

Table 1: Example of Sentiment Analysis

Industry	Use Case	Positive Sentiment	Negative Sentiment
Business	Product reviews	"I love this product!"	"This product is terrible."
Healthcare	Patient feedback	"The hospital staff were so caring."	"The doctor was rude and dismissive."
Politics	Political campaigns	"I believe this candidate will make a great leader."	"This policy is unacceptable."
Entertainment	Film and TV reviews	"This movie was amazing!"	"This TV show was a waste of time."
Social media	Brand reputation management	"This movie was amazing!"	"This TV show was a waste of time."

This survey attempts to investigate how sentiment analysis can be used to gauge how layoffs affect a company's reputation. It will specifically look into the various techniques and methods used in sentiment analysis, the difficulties and restrictions of using sentiment analysis to gauge reputation, and the potential advantages of incorporating sentiment analysis into reputation management plans in the context of layoffs. The survey will include a summary of the research that has already been done on sentiment analysis and how it may be used to gauge how layoffs affect a company's reputation. The trends, gaps, and openings for additional study in this field will also be highlighted. The survey's results may be helpful for businesses, academics, and professionals who want to know how layoffs affect a company's reputation and how sentiment analysis might be used as a tool for reputation management in this situation. Such information can assist businesses in developing their communication plans, identifying possible problems, and making data-driven choices that will lessen the negative effects on their reputation.

However, there are significant difficulties in utilizing sentiment analysis to assess how layoffs affect a company's reputation. Sentiment analysis algorithms might not be able to fully capture the nuanced expressions of human emotions, sarcasm, or language used in certain contexts. The trustworthiness and bias of various data sources, including social media posts, news stories, and customer reviews, may differ, which might have an impact on the accuracy of sentiment analysis results. Additionally, utilizing sentiment analysis for reputation measurement may raise ethical questions around privacy, data security, and fairness.

Sentiment analysis provides a number of difficulties when attempting to measure the impact of layoffs on a company's reputation. The subtleties of human emotions, sarcasm, and language specific to a given situation may be difficult for sentiment analysis algorithms to effectively capture. The dependability and bias of various data sources, including news stories, customer reviews, and social network posts, might affect how accurate sentiment analysis conclusions are. Additionally, there can be moral issues with regard to privacy, data security, and justice when employing sentiment analysis to gauge reputation.

Despite these difficulties, sentiment analysis has the potential to be an important tool for comprehending and controlling the consequences. With a particular focus on the

consequences of layoffs on firm reputation, this study will, in the end, add to the increasing body of knowledge on sentiment analysis and reputation management. Researchers and practitioners interested in sentiment analysis as a tool for measuring reputation in the context of organizational downsizing, as well as companies looking to assess and manage the impact of layoffs on their reputation, may find the results of this survey to be useful.

1.2. History of Sentiment Analysis

Sentiment analysis dates back to the mid-20th century, but has gained tremendous momentum in recent years with advances in computing, machine learning, and big data.

1950s-1980s: Early Research

The origins of sentiment analysis can be traced back to the 1950s and 60s, when researchers began to explore ways to measure emotions in writing. In 1954, psychologist Charles Osgood proposed the "emotional differentiation" technique, which involves using a scale to measure the emotional state of words. In the 1960s, researchers such as Jacques Schenkein and Robert Plutchik developed theories of psychology and tried to categorize them. In the 1980s, theory research began to focus on using computational methods to analyse theory in texts. The researchers developed a rule-based system that uses manual rules and a dictionary to identify and classify emotions in text. These early systems relied on predefined lists of positive and negative words and written rules to determine sentiment.

1990s to 2000s: Dictionary-Based Approaches

In the 1990s, the study of psychology shifted to vocabulary-based approaches involving the use of vocabulary-based approaches or definitions. There is a list of notes to, positive, negative, neutral). Researchers have developed strategies for using these dictionaries to calculate text needs based on the frequency of positive and negative words in the text. In the early 2000s, researchers also began to explore machine learning-based methods for emotional analysis. This process involves using algorithms to recognize thought patterns from training data. Support Vector Machines (SVM), Naive Bayes, and Maximum Entropy are some of the popular machine learning algorithms used for real-time analysis.

From the 2010s to the present: Advanced Machine Learning Techniques

In recent years, the analysis of emotions has increased with the advent of deep learning and neural networks. Researchers have developed deep learning models such as convolutional neural networks (RNNs), convolutional neural networks (CNNs), and short-term neural networks (LSTMs) for sentiment analysis. These patterns are effective at capturing nuances of emotion in text content, including idioms, emotional content, and emotion in emoji or other multimedia content.

In addition to traditional supervised learning, researchers have explored other techniques such as unsupervised and semi-supervised learning, adaptive learning to enhance performance analysis thinking, especially when training data is scarce or very specific.

1.3. Approaches of Sentiment Analysis

In general, sentiment analysis techniques can be divided into three different types depending on their purpose and methodology.

1.3.1. Lexicon based approaches

Lexicon-based sentiment analysis techniques, which are also known as knowledge-based approaches, involve the use of pre-developed models that are manually created and analysed for syntactic and semantic patterns within the text. This type of analysis relies heavily on the establishment of patterns in grammatical syntax and seeks to understand the underlying meaning of the text.

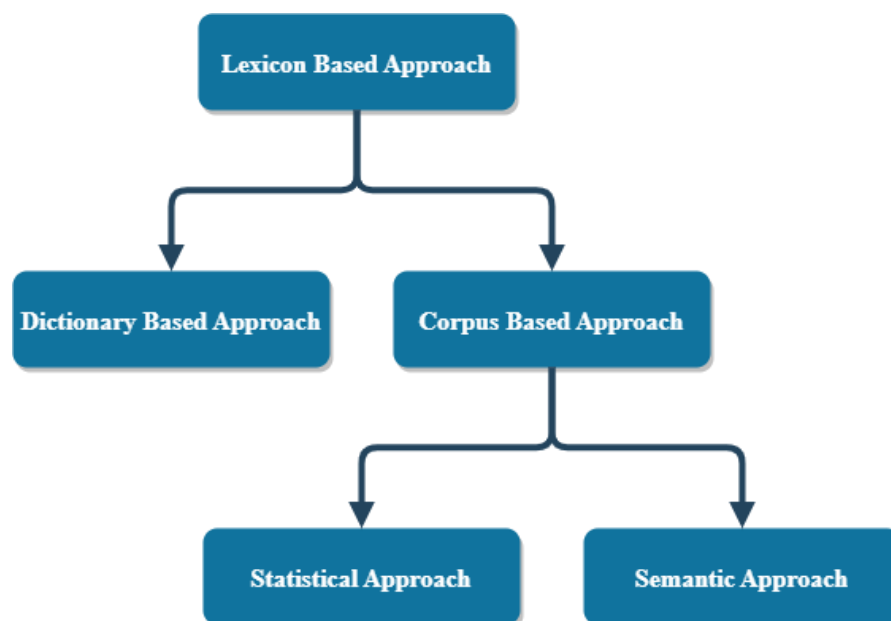


Figure 2: Lexicon-Based Methods

Dictionary based - The method of the dictionary-based approach involves selecting a small group of words to function as a reference point or dictionary. The goal is to establish a set of words that represent the concepts or ideas that are most relevant to the task at hand. This is done in order to facilitate further analysis and understanding of the text. By using a dictionary-based approach, one can quickly identify and categorize key terms and concepts.

Corpus based - A corpus-based approach is a technique used to analyse the sentiment or emotional orientation of particular words within a given context. This method involves analysing large collections of text data in order to gain insight into how certain words are used and the emotions they convey. By examining a broad range of textual sources, researchers can identify patterns and trends in language use and gain a more comprehensive understanding of the sentiment.

1.3.2. Machine learning based approaches

The process of determining sentiment based on statistical models involves vectorizing data, training the model, and then utilizing the model to predict sentiments. An advantage of this method is that the model can be tailored in any way desired, including being trained to detect sarcasm and irony. This flexibility allows for more accurate sentiment analysis.

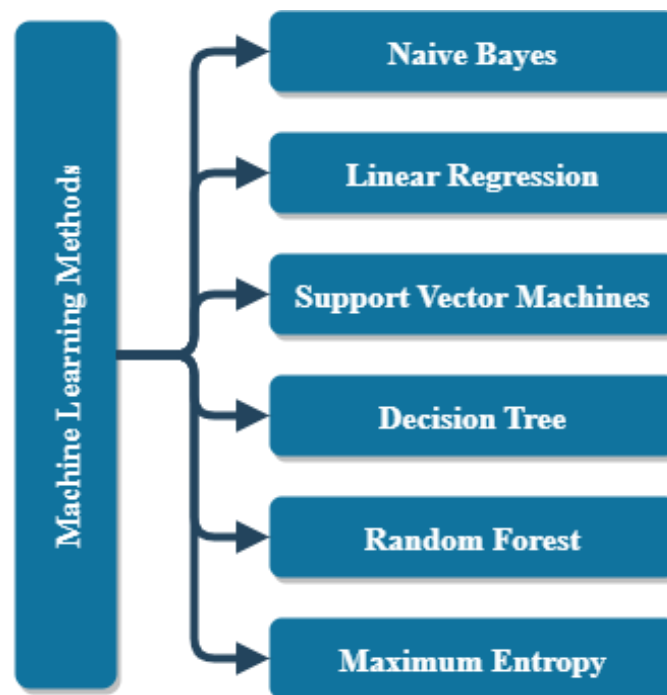


Figure 3: Machine Learning Methods

Machine learning, a subfield of artificial intelligence, is a technique that involves teaching machines to learn from data and improve their performance without being programmed explicitly. Machine learning can be divided into two main categories, which depend on the type of dataset being used. If the data is labelled, we use supervised learning algorithms, while unsupervised learning is used when the data is unlabelled. In our case, since we have a labelled dataset, we will be using supervised learning for our analysis. When working with supervised learning, the choice of the model depends on whether the data is continuous or discrete. Regression models are used for continuous data, while classifiers are used for discrete data. In our case, since the data is discrete, we will be using classifiers. Sentiment analysis is a common application of classifiers, and there are several classifiers that can be used for this purpose, including Naive Bayes, Support Vector Machines (SVM), and Random Forest Classifiers. Overall, the use of supervised learning and classifiers is an effective technique for sentiment analysis, and it can be critical for making informed decisions in various business domains.

1.3.3. Hybrid approaches

Both lexicon-based and ML (Machine Learning) techniques have their own advantages and disadvantages in the field of sentiment analysis. Therefore, businesses have started implementing hybrid methods that combine both techniques to overcome the limitations of each methodology. By combining both techniques, businesses are able to take advantage of the strengths of each approach in order to create a more powerful and accurate system. However, this approach requires a thorough understanding of both techniques to make the most of their potential in sentiment analysis applications.

1.4.Steps Involved in Sentiment Analysis

Sentiment analysis is a process that requires several sequential steps to be taken in order to obtain results that can be visualized. Initially, data gathering is needed in order to collect the data that will be analyzed later. Subsequently, data cleaning is performed to eliminate any errors or incorrect data that may interfere with the analysis process. Following that, data preprocessing is necessary to simplify the data and prepare it for analysis. Finally, data analysis is conducted, and the results are obtained and visualized. As can be seen, sentiment analysis is not a simple one-click process, but rather a series of necessary steps that need to be executed sequentially in order to get accurate and useful results.

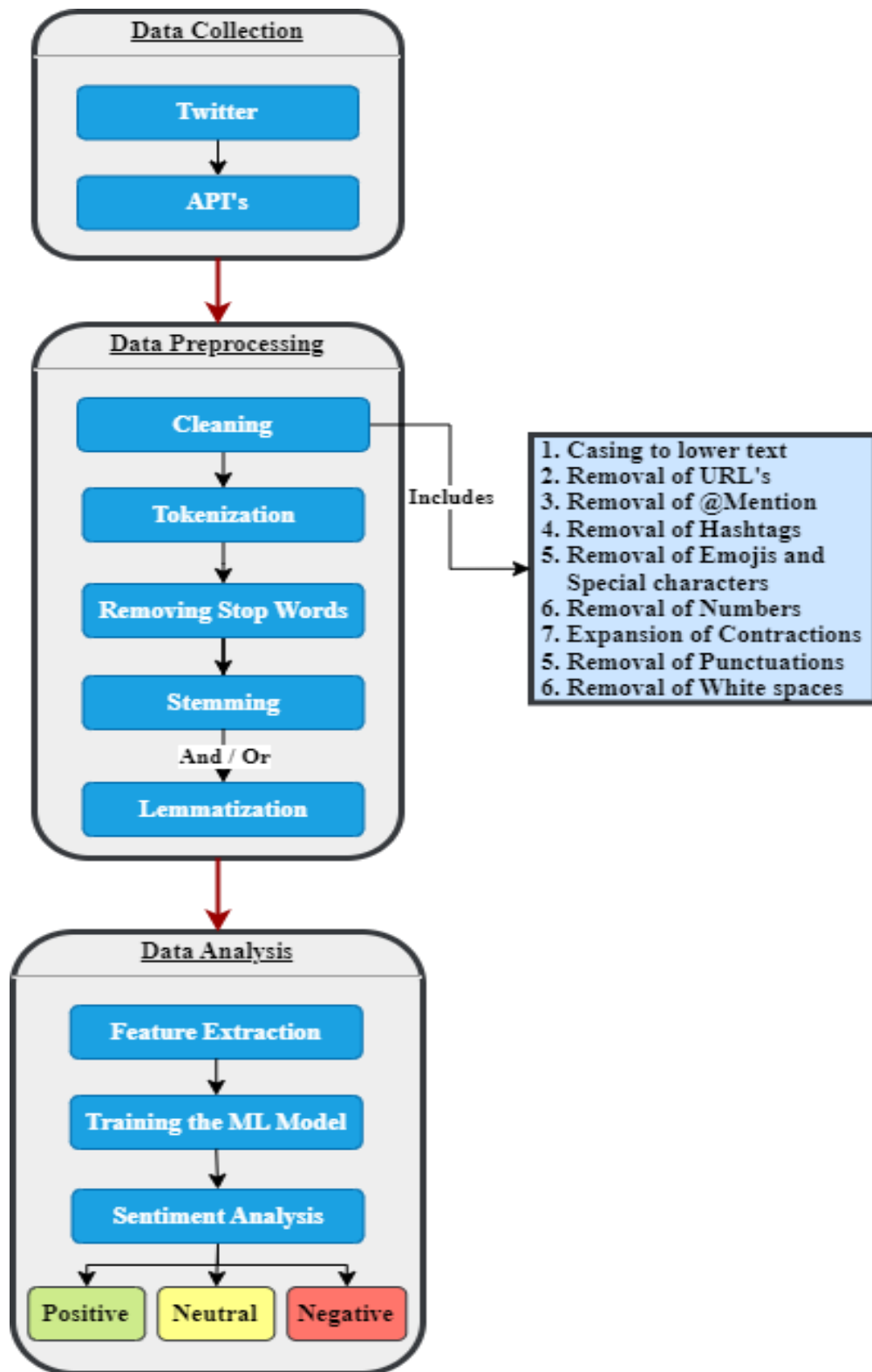


Figure 4: Process Chart of Sentiment Analysis Steps

1.4.1. Data Gathering

Twitter is a social media platform that generates an enormous amount of data, which can be used as a data source. By using the Tweepy API, users can easily connect to Twitter and extract its data in Python. The extracted data can then be stored in a CSV format using the popular Pandas module. Utilizing the Twitter API is a preferred method when compared to commercially available public datasets, as the latter has a smaller number and less diversity of datasets available for public use. This means that Twitter API provides users with the potential to access a wider range of data, which can lead to more substantial data analysis. The Twitter API has some notable limitations that users must be aware of. One of the most significant limitations is that tweets can only be extracted from the past week. However, this should not be a cause for concern since there are several third-party libraries that can be utilized to extract tweets as per their requirements. One of the most popular of these is Snsrape, a web scraper that can efficiently collect tweets from various users on Twitter. By making use of external tools, Twitter users can overcome the limitations of the Twitter API and gain access to all the tweets they require.

1.4.2. Data Pre-processing

The information retrieved from Twitter often includes a considerable amount of irrelevant data that is of no use. Therefore, it is important to eliminate any random characters or meaningless data from the content of tweets, which is accomplished through the use of a specific technology called natural language processing (NLP). NLP enables the extraction of valuable information from text data by identifying and removing irrelevant information such as random characters, grammatical errors, or meaningless words. By using NLP, the relevant information within the tweets can be analysed and extracted more effectively.

The process of pre-processing Twitter data begins by cleaning the data, which involves several steps. Firstly, all text is converted to lowercase, regardless of whether words have capital letters or not, to avoid treating them as different words. Secondly, mentions of users and numbers are removed as they don't provide any relevant context for sentiment analysis. Thirdly, URLs are removed, special characters and emojis are removed, contractions are expanded and punctuations are removed. Lastly, white spaces

are also removed. These pre-processing steps are essential for accurate sentiment analysis of tweets.

Then the process of tokenization is used to divide each statement into individual texts. Afterward, stop words such as "the," "but," "a," and "an" are excluded since they do not contribute to the meaning of the text. Lemmatization and stemming are two useful techniques used to bring words back to their original form or meaning. These techniques help to condense the text and increase the processing speed. Lemmatization involves reducing words to their base or dictionary form, while stemming involves removing prefixes and suffixes to get to the base word. By using these methods, the number of words is minimized, making it easier to process text quickly.

1.4.3. Data Analysis

As discussed in the approaches of sentiment analysis, there are several well-known tools available that employ lexicon-based approaches to analyse sentiment. Three of the tools that are commonly used by businesses are TextBlob, VADER, and SentiWordNet. TextBlob uses a pre-trained sentiment analysis model to assign sentiment scores to words and phrases, while VADER utilizes a rule-based system to analyse sentiment in text.

When it comes to analysing sentiments using machine learning, there are several models that are widely used in the industry. These models include naive Bayes, logistic regression, support vector machine, and random forest classifier. Each of these models has its own set of strengths and weaknesses that make them suitable for different scenarios. Naive Bayes is known for its simplicity and efficiency in text classification.

In comparison to Lexicon-based approaches, Machine learning methods require two different datasets - one for training the machine learning model and other for prediction. Although the complexity of the machine learning methods increases, they are still preferred for sentiment analysis of social media data. The reason for this preference is that machine learning models can be trained to identify subtleties such as irony and sarcasm in the text.

1.5. Corporate Reputation

Corporate reputation refers to the perception and image that an organization has among its stakeholders, including customers, investors, employees, suppliers, and the general public. It is a critical asset that can significantly impact a company's success, growth, and profitability. In today's hyper-connected world, where information spreads rapidly through social media and other digital platforms, managing corporate reputation has become more critical than ever.

1.5.1. Importance of Corporate Reputation

Corporate reputation is essential for many reasons. Firstly, it can influence consumer behavior. Customers are more likely to buy products or services from companies that they trust and have a positive reputation. Similarly, a negative reputation can lead to reduced sales, customer loyalty, and brand value.

Secondly, corporate reputation can affect investor decisions. Investors prefer to invest in companies that have a strong reputation and are perceived to be ethical and trustworthy. Conversely, a negative reputation can lead to a decrease in investment and a lower valuation.

Thirdly, corporate reputation is vital for attracting and retaining talented employees. People want to work for companies that have a positive reputation and are perceived to be socially responsible and ethical. A company with a negative reputation may find it challenging to attract and retain top talent.

Fourthly, corporate reputation can affect relationships with suppliers and partners. Suppliers and partners prefer to do business with companies that have a good reputation and are perceived to be reliable and trustworthy.

Finally, corporate reputation can impact regulatory and government relationships. Companies that have a positive reputation are more likely to have favorable regulatory and government relationships than those with a negative reputation.

1.5.2. Building Corporate Reputation

Building a strong corporate reputation requires a deliberate and sustained effort. Here are some strategies that companies can use to build a positive reputation:

1. **Be Transparent and Honest:** Companies should be transparent and honest in their dealings with customers, employees, investors, and the public. They should communicate openly about their operations, policies, and practices. They should also be honest about their successes and failures.
2. **Deliver High-Quality Products and Services:** Companies should focus on delivering high-quality products and services that meet the needs of their customers. This can help build customer loyalty and positive word-of-mouth recommendations.
3. **Foster a Positive Work Culture:** Companies should foster a positive work culture that values employee well-being, diversity, and inclusivity. This can help attract and retain top talent and improve employee morale and productivity.
4. **Engage with Customers and Stakeholders:** Companies should engage with their customers and stakeholders regularly to understand their needs and concerns. This can help build trust and improve relationships.

1.5.3. Protecting Corporate Reputation

Protecting corporate reputation is as critical as building it. Here are some strategies that companies can use to protect their reputation:

1. **Monitor social media:** Companies should monitor social media regularly to identify any negative comments or feedback. They should respond promptly and professionally to address any concerns and resolve any issues.
2. **Be Proactive:** Companies should be proactive in addressing potential reputation risks before they become significant issues. This can include implementing risk management strategies, crisis communication plans, and training employees to respond to negative situations.
3. **Maintain Ethical Standards:** Companies should maintain ethical standards and avoid engaging in any practices that could damage their reputation. This can include avoiding conflicts of interest, adhering to regulations and laws, and being transparent about any potential ethical issues.
4. **Be Prepared for Crisis:** Companies should have a crisis communication plan in place that outlines how to respond to any potential crisis that could damage their reputation.

2. PROFILE OF PROBLEM

Sentiment analysis can be applied to a number of approaches to determine corporate reputation. It's crucial to remember, though, that not every strategy will result in success. It's critical to consider the inherent weaknesses of different techniques when determining brand reputation. Some techniques rely on certain assumptions, limitations, or biases which may hinder the accuracy of the results. As a result, a thoughtful review of the various approaches is required for the effective measurement of corporate reputation using sentiment analysis. When it comes to sentiment analysis, selecting an appropriate approach hugely depends on the dataset being used. Therefore, it is essential to identify the best sentiment analysis approach that aligns with social media datasets such as the Twitter dataset.

To effectively analyse datasets in a business environment, it is essential to begin by cleaning the raw data. There are several steps in the data cleaning process, which can vary depending on the dataset being used and the chosen sentiment analysis approach. Some of the general steps that we can follow to clean our data include removing duplicate entries, filtering out emojis, URLs, and punctuations, addressing irrelevant data entries, and ensuring proper formatting of data. A robust data cleaning procedure helps ensure that the analysis is more accurate and reliable. Therefore, we must ascertain every step that will be required for cleaning our data. After doing the analysis, there are a number of actions that must be followed in order to determine the company's reputation.

Things like layoffs can have a great impact on any company's reputation, but exactly how much it has affected in recent times needs to be measured. It's crucial to assess the extent to which recent layoffs have impacted a company's reputation because the effects of layoffs might be severe. Companies can use this measurement to more accurately assess the damage and take the necessary action to protect their industry reputation. Various stakeholders, including employees, clients, and shareholders, may be impacted by layoffs, thus it is crucial for firms to take steps to minimise any unfavourable repercussions.

3. EXISTING SYSTEM

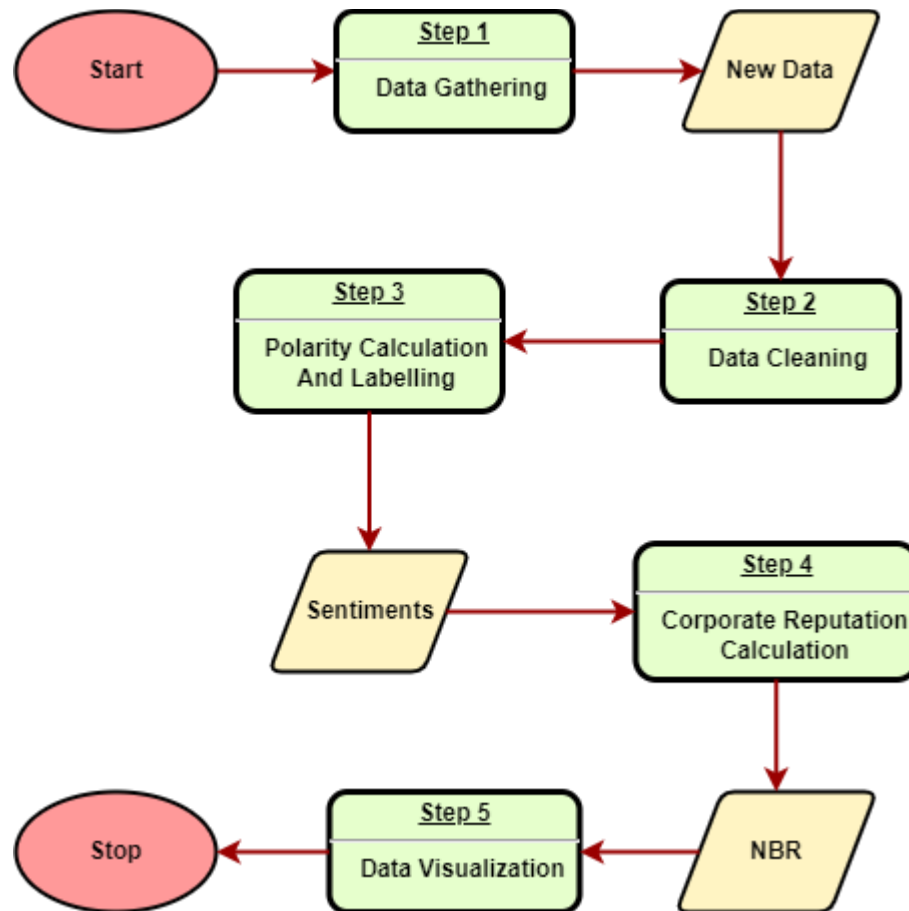


Figure 4: Existing System Flowchart

The current system relies on the use of a sentiment analyser called TextBlob, which leverages a lexicon-based approach to assign a polarity score to each text. The polarity score ranges from +1 to -1, indicating the degree of positivity or negativity of the text. This approach is widely used in analyzing the sentiment of textual data, but it lacks flexibility as it cannot be adjusted to meet specific needs. However, by implementing machine learning models, we can train them to operate according to our specific requirements, resulting in a significant boost in performance. By harnessing machine learning technology, it is possible to customize the system to meet the demands of our unique business needs for example, when we want to analyse sarcasm and irony machine learning models will have an upper hand over TextBlob.

4. PROBLEM ANALYSIS

4.1. Product Definition

The project is titled ‘Measuring effects of layoffs on company reputation using sentiment analysis’. As the project title suggests, the primary focus of the project is on sentiment analysis of Twitter data. The analysis is then used to calculate the reputation of corporate entities. The ultimate goal of this product is to provide a comprehensive model that can be utilized to evaluate the reputation of a company based on data from Twitter. It also hopes to provide a moderate, but comprehensive expansion of the impact that events such as layoffs can have on a company’s reputation. It will describe how such events affect the way a company is perceived by the public and the potential consequences that may result. The product consists of all the components that are required for gathering data, cleaning data, analysing data, calculating results, and presenting it in a graphical format. This includes the process of scraping Twitter, natural language processing, and machine learning, and then visualizing the results using charts and plots that can be easily comprehended. The aforementioned products not only assist in comprehending human opinions on different topics but also enable businesses to utilize this analysis to devise effective strategies for their growth and success. The product relies on a machine learning (ML) model that has been trained more than 100,000 tweets of span of 5 years, resulting in an impressive accuracy rate of 94%. This indicates the high success rates of the machine learning model and its ability to give accurate results. Moreover, the users have the flexibility to download specific data fragments and modify the model according to their preferences. This feature empowers the users to evaluate and calculate different types of reputation scores for the company. They can experiment with different models to evaluate a wide range of factors that may affect a company's reputation such as its financial strength, market position, customer satisfaction, and so on. By retraining the model with different data segments, users can get a better understanding of the reputation status of the company. One of the significant advantages of running this project is that you don't need any special software to run it smoothly. The project is developed on Google Colab, which is an online platform designed for NLP and ML, among others. The project can be readily uploaded to the platform and used right away. Google Colab offers a hassle-free solution for running the project, with no need to download and install specialized software.

4.2. Feasibility Analysis

The product has the capability to be used efficiently with almost every type of company data and can assist in analysing all sorts of events such as product launches. By using this product, businesses can gain insightful information on their public image without having to manually sift through large amounts of raw data. Such products also provide valuable insights into the preferences of potential customers, which can be leveraged to improve their overall experience with the company. Additionally, the analysis can help identify patterns and trends in the market, allowing for targeted marketing strategies that focus on the most promising potential customers. All in all, these products are a valuable tool for businesses looking to gain a better understanding of their target market and make informed decisions to foster their growth. After a few updates, this product has the ability to analyse the success of a movie or the popularity of celebrities through the use of sentiment analysis. Additionally, it can be applied in various other situations that require the analysis of sentiment. These analyses can help users understand how certain individuals or products are being perceived by the general public.

4.3. Project Plan

Project name: Measuring effects of layoffs on company reputation using sentiment analysis

Project mentor: Gagandeep Kaur

Start date: Feb. 10, 2023

End date: Apr. 12, 2023

Days required: 62

Team members: Rahul Kumar, Harshul and Sujal

Tasks: Getting a twitter developer account or finding an alternate to scrape the tweets, finalising the steps of data cleaning, experimenting with various ML models to determine the model best suited for this project, training the model with a huge dataset, calculating the net brand reputation, producing the result charts

Overall progress: Completed

5. SOFTWARE REQUIREMENT ANALYSIS

Pre-requisites:

- Internet Connectivity – A good and stable internet connectivity is required as we will be using an online IDE to run this project.
- Web Browser – We need a web browser to run the online IDE, any web browser will work but google chrome is preferred
- Google Account – As the IDE used to run this project is developed and managed by google, we need a google account to run all project.

After fulfilling the pre-requisites, we only need Google Colab to run this project.

Google Colab: Colab can be thought as a web-based version of the popular Jupyter Notebook. It allows users to create, edit, and share code using a web browser without the need for any installations or configurations. Moreover, Colab platform grants its users access to computing resources such as storage space, memory, processing power, GPUs, and TPUs. Through using Colab, users can perform various machine learning, data analysis, and algorithmic computations having these resources at their disposal. It also offers multiple pre-installed libraries that we can import and use such as Pandas, NumPy, Matplotlib, and the ones that are not pre-installed user can install it using !pip install command.

Most of the libraries used in this project are already available in Google Colab, they are NumPy, Pandas, NLTK, TextBlob, Matplotlib, Seaborn, WordCloud, and re.

We need to install just two external libraries.

Snsrape: It stands for social network scrape. It is an open-source python library that can be used to scrape data from several social media platforms like Twitter and Facebook.

Contractions: It is a python library that we need for expansion of contractions.

Contractions are words or combinations of words that are shortened by dropping letters and replacing them by an apostrophe.

6. DESIGN

6.1. System Design

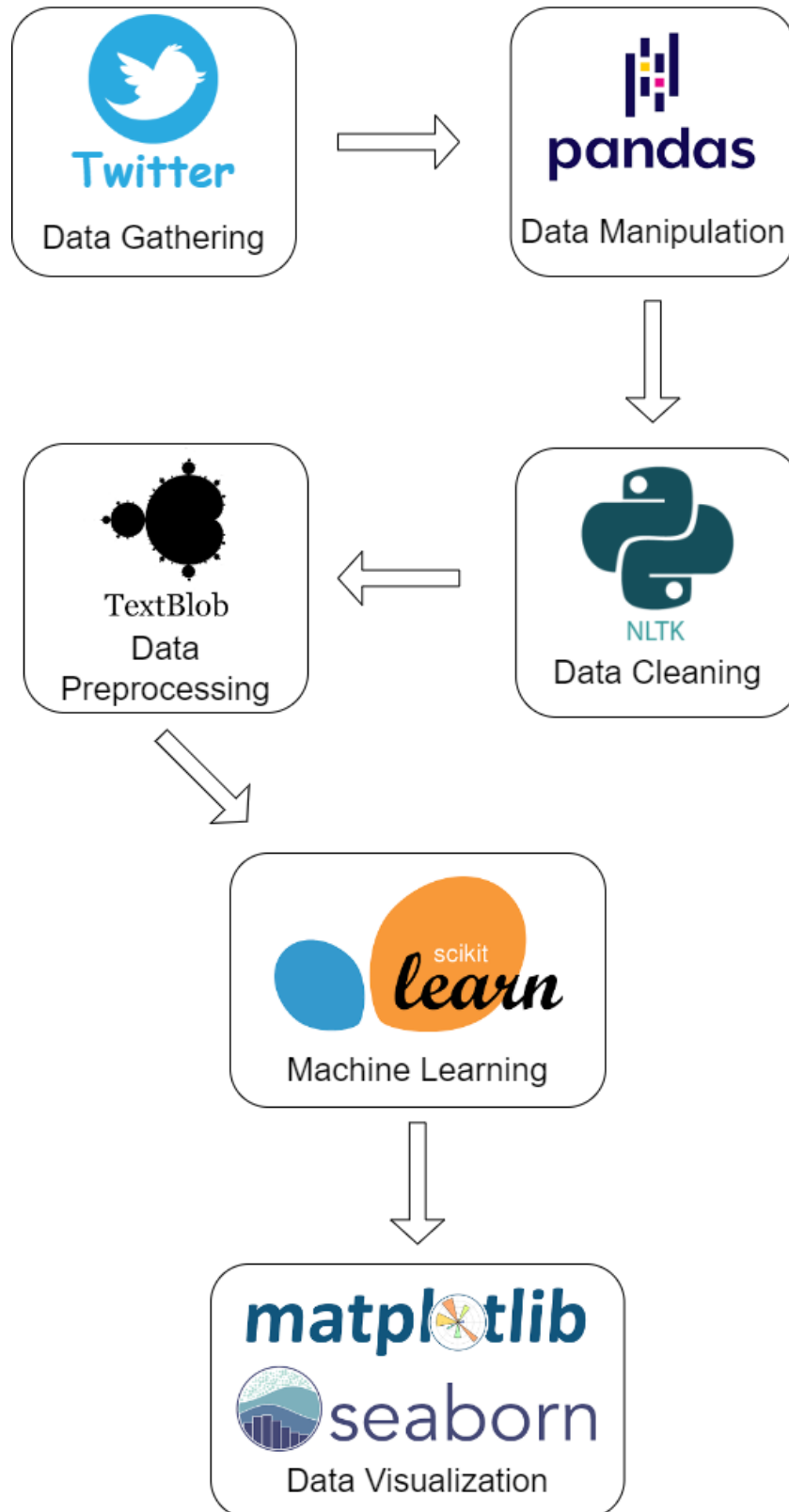


Figure 5: System Design

6.2. Detailed Design

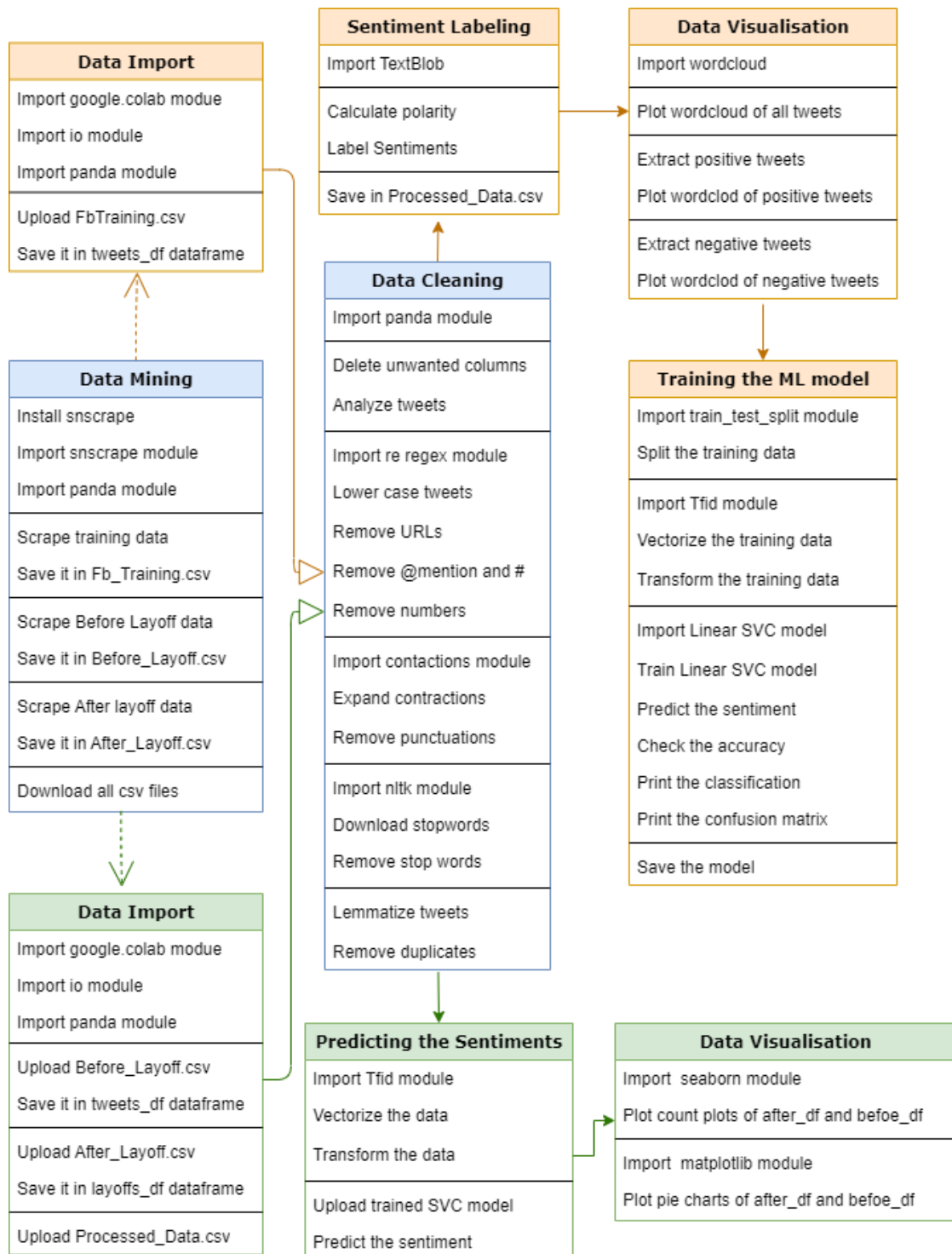


Figure 6: Detailed Design

6.3. Flow Chart

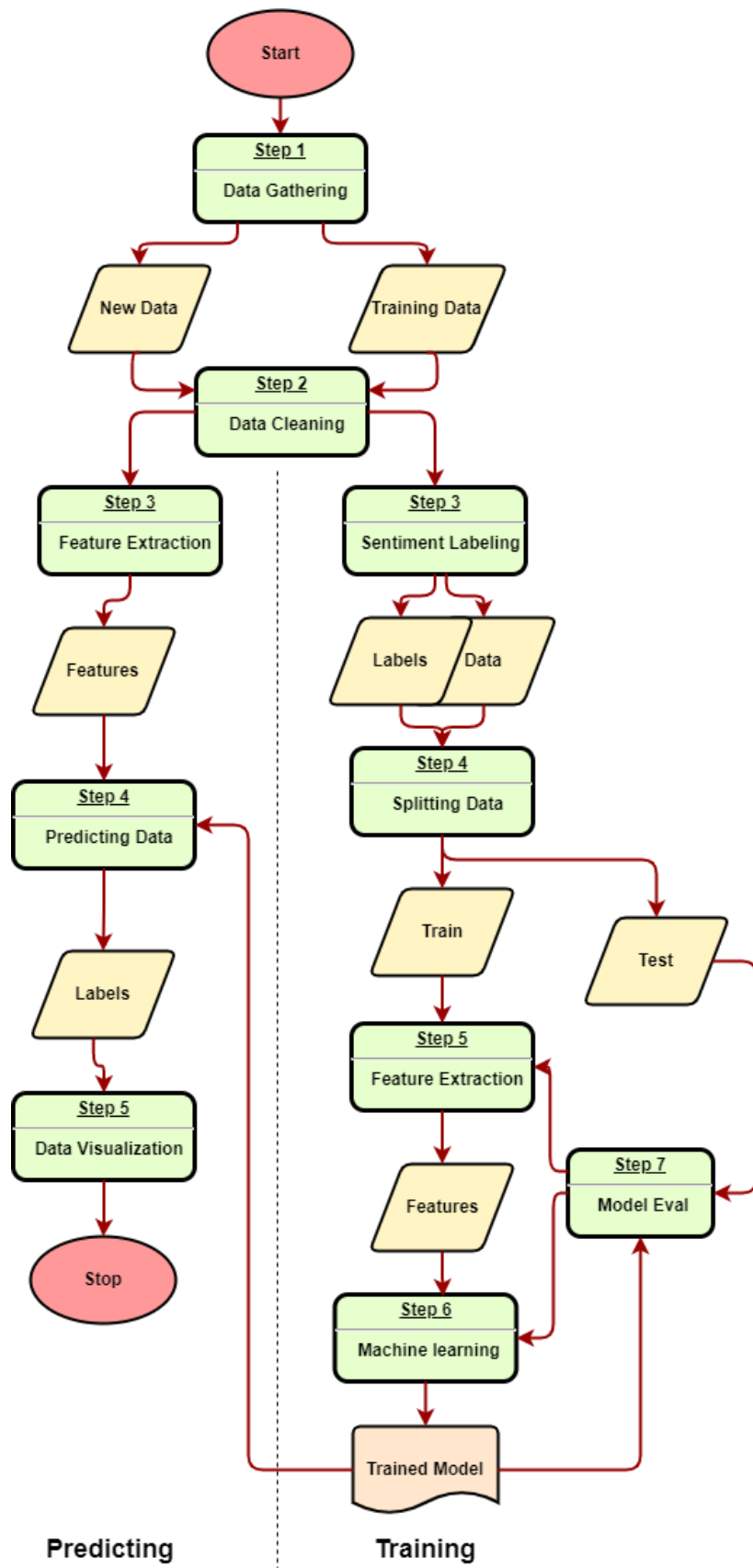


Figure 7: Flow Chart

6.4. Pseudo Code

ALGORITHM: Loading Data

START:

```
IMPORT files FROM google.colab
IMPORT io
IMPORT panda AS pd
FUNCTION upload()
    uploaded <- files.upload()
    filename <- next(iter(uploaded))
    df <- pd.read_csv(io.BytesIO(uploaded[filename]),lineterminator='\n')
    RETURN df
END FUNCTION
tweets_df <- CALLING upload
CALLING tweets_df.head()
layoffs_df <- CALLING upload
CALLING layoffs_df.head()
after_tweets_df <- CALLING upload
CALLING after_tweets_df.head()
```

STOP

ALGORITHM: Loading Trained ML Model

START:

```
IMPORT pickle
uploaded <- files.upload()
filename <- next(iter(uploaded))
model <- LOAD pickle.load(open(filename, 'rb'))
```

STOP

ALGORITHM: Importing Libraries

START:

```
IMPORT pandas AS pd
IMPORT numpy AS np
```

```

IMPORT re
INSTALL PACKAGE contractions
IMPORT contractions
IMPORT nltk
DOWNLOAD 'stopwords'
DOWNLOAD 'wordnet'
DOWNLOAD 'punkt'
IMPORT textblob FROM textblob
IMPORT word_tokenize FROM nltk.tokenize
IMPORT WordNetLemmatizer FROM nltk.stem
IMPORT stopwords FROM nltk.corpus
stop_words <- SET stopwords='english'
IMPORT TfidfVectorizer FROM sklearn.feature_extraction.text
IMPORT seaborn AS sns
IMPORT WordCloud FROM wordcloud
IMPORT matplotlib.pyplot AS plt
IMPORT style FROM matplotlib
USE style 'ggplot'
IMPORT warnings
CALL filterwarnings('ignore')

STOP

```

ALGORITHM: Deleting unwanted columns

START:

```

FUNCTION del_col(df)
  PRINT df.column
  RETURN df.drop(['User', 'Date Created'], axis=1)
END FUNCTION

```

```

tweets_df <- del_col(tweets_df)
layoffs_df <- del_col(layoffs_df)
after_tweets_df <- del_col(after_tweets_df)

```


STOP

ALGORITHM: Data cleaning

START:

```
FUNCTION data_cleaning(tweet)
    tweet <- CONVERT TEXT TO LOWERCASE
    tweet <- REMOVE ALL URLS
    tweet <- re.sub("@\w+|#", "", tweet)
    regex_pattern <- REMOVE EMOJIS
    tweet <- ".join(c for c in tweet if not c.isdigit())
    LIST expanded
    FOR word IN tweet.split()
        expanded.append(contractions.fix(word))
    tweet <- '.join(expanded)
    END FOR
    tweet <- re.sub('[^\w\s]', "", tweet)
    tweet_tokens <- word_tokenize(tweet)
    filtered_texts <- REMOVE STOP WORDS
    lemma <- WordNetLemmatizer()
    lemma_texts <- lemmatizing
    RETURN " ".join(lemma_texts)
END FUNCTION

tweets_df.Tweet <- tweets_df['Tweet'].apply(data_cleaning)
layoffs_df.Tweet <- layoffs_df['Tweet'].apply(data_cleaning)
after_tweets_df.Tweet <- after_tweets_df['Tweet'].apply(data_cleaning)
```

STOP

ALGORITHM: Checking for duplicate rows and deleting them

START:

```
FUNCTION drop_dupli(df)
```

```

        duplicate <- df[df.duplicated()]
        PRINT duplicate
        RETURN df.drop_duplicates('Tweet')
    END FUNCTION

    tweets_df <- drop_dupli(tweets_df)
    layoffs_df <- drop_dupli(layouts_df)
    PRINT tweets_df.shape, layoffs_df.shape
    after_tweets_df <- drop_dupli(after_tweets_df)
STOP

```

ALGORITHM: Predicting sentiments

START:

```

    FUNCTION pred_senti (df,proc_df)
        vect <- VECTORIZATION
        tweets <- vect.transform(df['Tweet'])
        sentiment <- model.predict(tweets)
        RETURN sentiment
    END FUNCTION

```

```

    processed_df <- upload()
    CALLING processed_df.head()
    tweets_df['Sentiment'] <- pred_senti(tweets_df, processed_df)
    layoffs_df['Sentiment'] <- pred_senti(layouts_df, processed_df)
    after_tweets_df['Sentiment'] <- pred_senti(after_tweets_df, processed_df)
STOP

```

ALGORITHM: Calculating Corporate Reputation

START:

```

    FUNCTION
        pos <- df[df.Sentiment == 'Positive']

```

```

pos_count <- len(pos.index)
neg <- df[df.Sentiment == 'Negative']
neg_count <- len(neg.index)
NBR <- ((pos_count-neg_count)/(pos_count+neg_count))*100
RETURN NBR
END FUNCTION

```

```

NBR_before_layoff <- Calculate_NBR(tweets_df)
NBR_only_layoff <- Calculate_NBR(layoffs_df)
NBR_after_layoff <- Calculate_NBR(after_tweets_df)
PRINT (NBR_before_layoff,NBR_after_layoff,NBR_only_layoff)
STOP

```

ALGORITHM: Data Visualization

START:

```

LIST Dataset <- ['Before Layoff', 'After Layoff', 'After (layoff tweets)']
values <- [NBR_before_layoff,NBR_after_layoff,NBR_only_layoff]
fig <- plt.figure(figsize = (5, 5))

```

CREATING THE BAR PLOT

```

plt.xlabel("Datasets")
plt.ylabel("Net Brand Reputation")
plt.title("Net Brand Reputation Comparisons")
plt.show()

```

```

fig, ax <- plt.subplots(1,3,figsize=(10,5))
fig.tight_layout(pad=4.0)
ax[0].title.set_text('Before Layoff')
ax[1].title.set_text('After Layoff')
ax[2].title.set_text('After Layoff (layoff tweets)')
sns.countplot(x = 'Sentiment', data = tweets_df, ax=ax[0])
sns.countplot(x = 'Sentiment', data = after_tweets_df, ax=ax[1])

```

```
sns.countplot(x = 'Sentiment', data = layoffs_df, ax=ax[2])  
fig.show()
```

STOP

ALGORITHM: Data Visualization

START:

```
fig, ax <- plt.subplots(1,3, figsize=(10,10))  
fig.tight_layout(pad=4.0)  
colors <- ("green","yellow","red")  
tags <- tweets_df['Sentiment'].value_counts()  
tags1 <- after_tweets_df['Sentiment'].value_counts()  
tags2 <- layoffs_df['Sentiment'].value_counts()  
explode <- (0.1,0.1,0.1)  
tags.plot OF INDEX 0  
tags.plot OF INDEX 1  
tags.plot OF INDEX 2
```

STOP

7. IMPLEMENTATION

The implementation of this project is broken down into three stages: first stage is data gathering or mining, the second stage is exploring different machine learning models, selecting the model that best fits our needs, and training it with a sizable dataset. And the final stage is utilizing the trained model to forecast the sentiments of tweets sent before and after a layoff, calculating the corporate reputation and comparing the results.

7.1. Data Gathering

Our first port of call for data collection is the Twitter API because we had decided to work on tweets for sentiment analysis. Twitter offers a platform and a Tweepy library via which we can access and use data from Twitter for reasons of our own. To get started, we first register a Twitter account, which is simple to do using a phone number or email address. Then we complete a brief form on the <https://developer.twitter.com> website explaining how we intend to use Twitter data to register for a developer account. Once our application has been processed, it will take a few days to be granted access. Meantime the developer's platform will look like below.

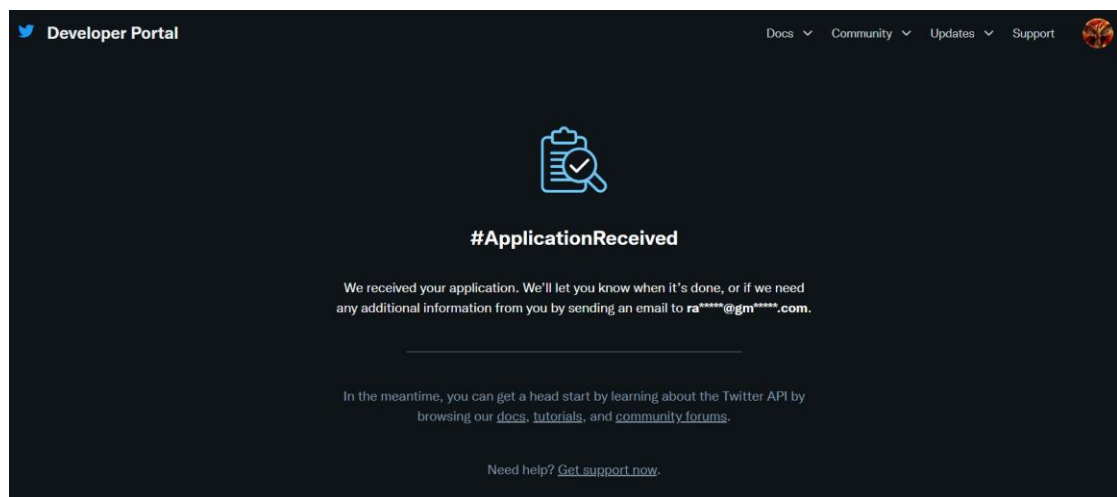


Figure 8: Developer Account

It took longer in our situation. So, we discover a different method for obtaining data from Twitter. The undocumented Python wrapper known as snsrape stood out among the libraries used to scrape tweets.

For IDE we use Google Colaboratory, also referred to as Colab, is a web IDE for Python that is convenient and easy to use. With cloud storage capabilities, Colab is a fantastic platform for data scientists to carry out Machine Learning and Deep Learning projects. We only need to perform a ‘Google Colab’ search, open the first result, and begin working on our project.

To gather data using snsrape, it is we create a script, which can be coded in Python. Python is a reliable language that is capable of delivering excellent results when used for data-gathering tasks. Its versatility and abundant external libraries allow programmers to manipulate different data types and perform machine learning effectively. As a result, it is a well-known language in the field of analytics and data science. The fact that it can be integrated with various tools and languages makes it a preferred option for developers.

We install snsrape with a straightforward pip install. The search keyword and the dates for which we require the tweets must be written in the script as a query. The Snsrape function returns a Twitter object with the following attributes.

Attributes in snsrape Tweet Object:

- url:
- date: Date tweet was created
- rawContent: Text content of tweet
- renderedContent:
- id: Id of the tweet
- user: Contains username, displayname, id, description, verified, etc.
- outlinks:
- replyCount: Count of replies
- retweetCount: Count of retweets
- likeCount: Count of likes
- quoteCount: Count of users that quoted the tweet
- conversationId:
- lang: Assumed language of tweet
- source: Device tweet was posted from, iPhone, Android, etc.
- media: Media object
- retweetedTweet: Id of original tweet if it's a retweet
- quotedTweet: Id of original tweet if it's a quoted tweet
- mentionedUsers: User object of mentioned users

Note: These are not all of the attributes and the description is left blank if purpose is unknown

Figure 9: Attributes of snsrape tweet object

Among all attributes we just pull-out user.Username, date and rawContent. In our project, four datasets are required. We train our machine learning model using a large dataset of 1 lakh tweets regarding 'Facebook' that were sent over a five-year period. Three datasets of 10,000 tweets each will be used to forecast the sentiments: a dataset of tweets about 'Facebook' from the year prior to their layoff, a dataset about their 'layoff', and a dataset from their post-layoff tweets. In order to store them on Colab, we must use pandas dataframes. A pandas dataframe is an efficient tool for data manipulation and is widely implemented in machine learning applications. With its advanced features and versatile capabilities, pandas offer a range of benefits to businesses interested in processing and analysing large datasets. After that, we convert them into CSV files and download them to our local device for later usage.

7.2. Model Training

We first upload the training data CSV file that was assembled in the previous phase. Twitter's raw data is unfit for use in modal training. The majority of tweets are text with at the rate mentions, emojis, hashtags, emoticons, URLs, timestamps, etc. As a result, we use NLTK functions to clean data as well as carry out various pre-processing operations. NLTK, or the Natural Language Toolkit, is widely recognized as an effective tool for data cleaning and pre-processing. It is especially helpful in processing language-related data, such as text. In data cleaning we first lower case the tweet, then we remove URL's, @mentions, hashtags, emojis, emoticons, numbers, then we expand contractions. Contractions, which are shortening of two words into one like can't (can + not), aren't (are + not), they'd (they + had/would), etc, have become a common practice in daily conversations, electronic communication, and online forums, particularly in social media platforms such as Facebook or Twitter. After that we remove punctuations. For the purpose of sentiment analysis, it is crucial to eliminate stopwords from the text as they do not convey any emotional sentiment. In other words, words like "the," "and," "a" have no emotional impact in a sentence. The final step of data cleaning involves the process of lemmatization. This method involves the reduction of words to their base form or lemma.

Table 2: Raw VS Clean tweet

Type	Tweet
Raw	#Facebook’s Customer #Chat Plugin Is Coming To A Website Near You ~ @villou 🖥️ 💬 https://t.co/CQXuz2I3V9 https://t.co/vwRinQa18I
Clean	facebook’s customer chat plugin coming website near

We then remove duplicate rows as they will bias the model in our case. To evaluate the sentiment of tweets, we employ the use of TextBlob which provides a polarity value. This value expresses a measure of how positive or negative the tweet is. +1 means highly positive and -1 means highly negative and 0 is treated as neutral. With help of polarity value, we label each tweet as positive, negative or neutral. And thus, our labelled data is ready for supervised learning. Before going to machine learning we create some wordclouds of positive tweets, negative tweets and neutral tweets. A wordcloud is a type of visual representation that displays the frequency of words, with the most frequently plotted biggest in the display.



Figure 10: Wordcloud of Tweets

In the process of initiating machine learning, one of the essential steps involves data splitting, which involves dividing the data into various subsets. This helps in developing an accurate model by training, testing, and validating the data. We divided train test data in 80:20 ratio.

Now before training the models, we do feature extraction. The process of converting data into numerical features is referred to as feature extraction. Basically, machine

learning doesn't work with text it needs numbers to work upon. We utilize the Tfidf vectorizer to choose the most relevant features for our analysis. Tfidf stands for Term Frequency-Inverse Document Frequency, that evaluates the significance of words based on their frequency in a given document.

In our analysis, since the data we have is labelled, we will be utilizing supervised models to aid in our analysis. These models will specifically be used for classification purposes since our data is discrete and not continuous. Classification in statistics involves the task of determining to which specific set of categories an observation might belong. It is a process of sorting and categorizing data into different groups based on certain parameters. Since they perform classification, they are called classifiers. We explore three classifiers named as support vector machine, logistic regression, and naïve bayes.

Support vector machines, or SVMs, are widely employed in supervised learning to address a variety of classification and regression issues. Machine learning professionals frequently utilise this technique to address classification issues. Its unique approach allows it to identify the boundaries between different classes, making it an efficient algorithm for classification tasks. The technique involves finding the best hyperplane to separate data points into different classes. In SVM we use LinearSVC.

Logistic Regression despite its name is a classifier that focuses on predicting the likelihood of an event occurring by examining the logarithm of the odds of that event happening. The major limitation of the approach is that it only performs well when the variable to be predicted is binary, when all predictors must be independent of each other, and when there are no missing values are anticipated in the data.

The likelihood that a specific data point would fall under a number of categories, or not, is determined using the probabilistic techniques of the Naive Bayes family. These algorithms make use of Bayes' theorem, which assumes that the probability of a hypothesis is directly proportional to the probability of the evidence given that hypothesis. This makes them a popular choice for classification tasks, such as spam filtering, sentiment analysis, and language identification.

After predicting the sentiments, we calculate accuracy of all three models and below is the result.

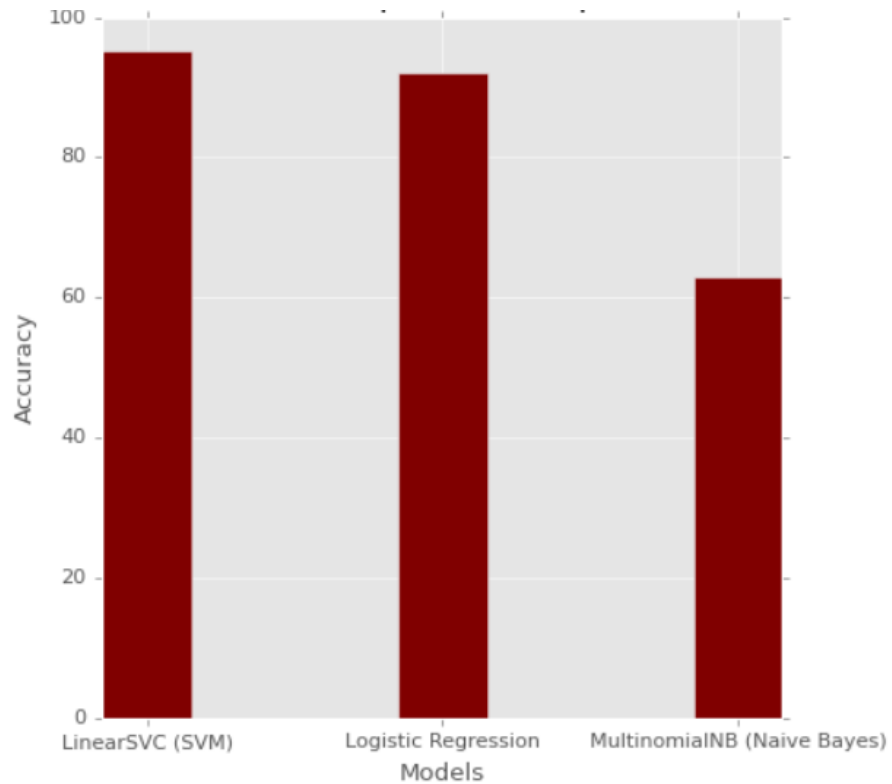


Figure 11: Accuracy Comparison of ML models

We see LinearSVC gave the best accuracy of 95.10% followed by Logistic Regression with 92.02% and last is MultinomialNB with 62.87%. Following is the confusion matrix of LinearSVC.

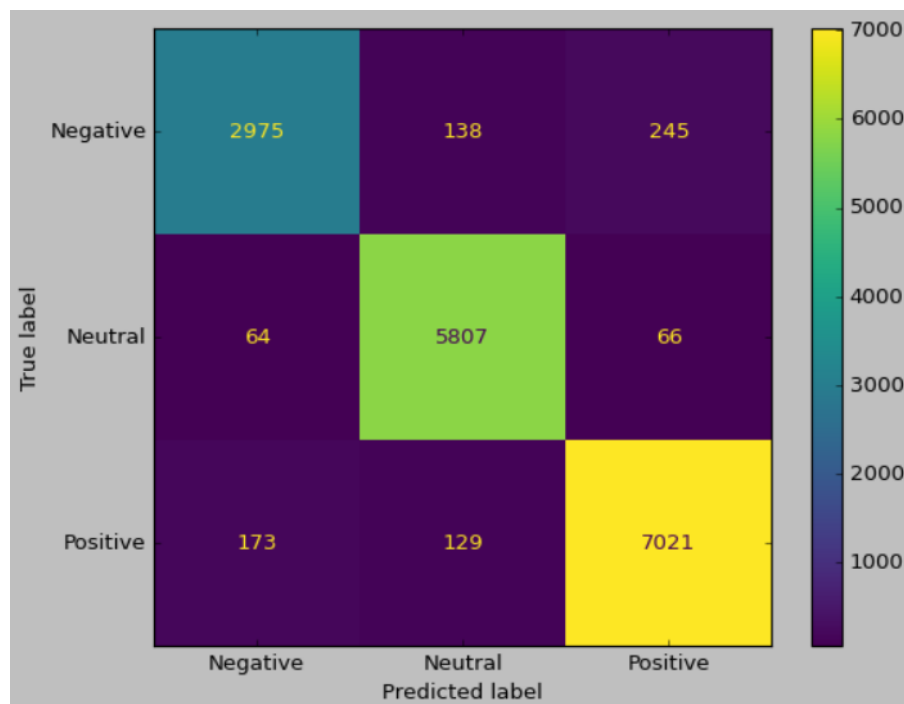


Figure 12: Confusion Matrix of LinearSVC

The confusion matrix is a commonly used tool to evaluate the performance of classification models, specifically for a given set of test data. It helps to visualize the accuracy and effectiveness of the model by capturing its true positives, true negatives, false positives, and false negatives. This matrix presents a detailed comparison between the predicted values and the actual values of the classification model, allowing for a clear and accurate understanding of model performance.

At last, with help of pickle module, we save this trained model in a .sav file, so that we can import it later and use for predictions.

7.3. Predicting Sentiments and Calculating Corporate Reputation

Following model selection and training. We can start with predictions of new data. We upload all three datasets that we had previously collected and saved. Next, we store them in data frames and name tweets_df that has before layoff tweets, layoffs_df which only contain layoff tweets and after_tweets_df that has tweets after layoffs, basically it contains tweets about Facebook after layoff time but but do not entirely focus on layoffs.

The next logical step is to eliminate unnecessary columns, clean up the data, and eliminate duplicate records. By sending the three datasets to the Python functions that were developed in the previous stage, we process all three datasets parallelly.

The trained model, a dataset that was used to vectorize the training dataset, is subsequently uploaded. These enable us to predict the sentiments of the three datasets.

Now, main step we calculate corporate reputation using the following formula:

$$NBR = \left[\frac{\text{Number of positive tweets} - \text{Number of negative tweets}}{\text{Number of positive tweets} + \text{Number of negative tweets}} \right] \times 100\%$$

Figure 13: NBR Formula

NBR stands for Net Brand Reputation and the higher the NBR score, the better the brand's reputation is thought to be. In our example, the NBR scores were 24.59 for the layoff data, 29.99 for the after-layoff data, and 35.10 for the before layoffs data.

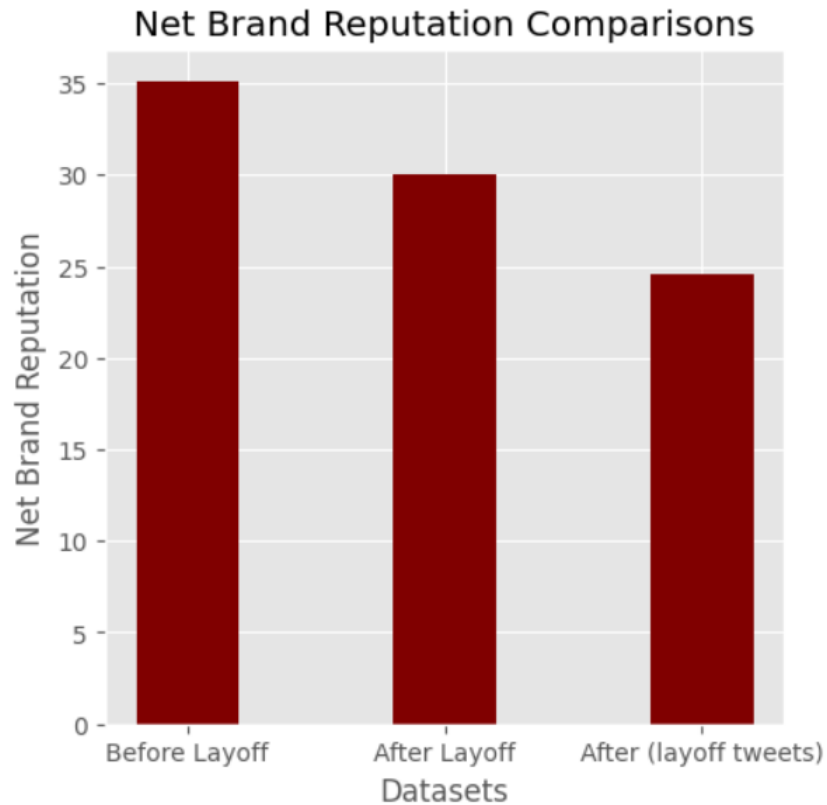


Figure 14: NBR Score Comparisons

It is evident that the implementation of layoffs has caused a decrease in the NBR score, ultimately affecting the company's reputation in a negative manner. In other words, the layoffs have had adverse effects on the company's image in the public eye. Statistics show that there was a 30% reduction in NBR score between tweets sent before and after the layoff.

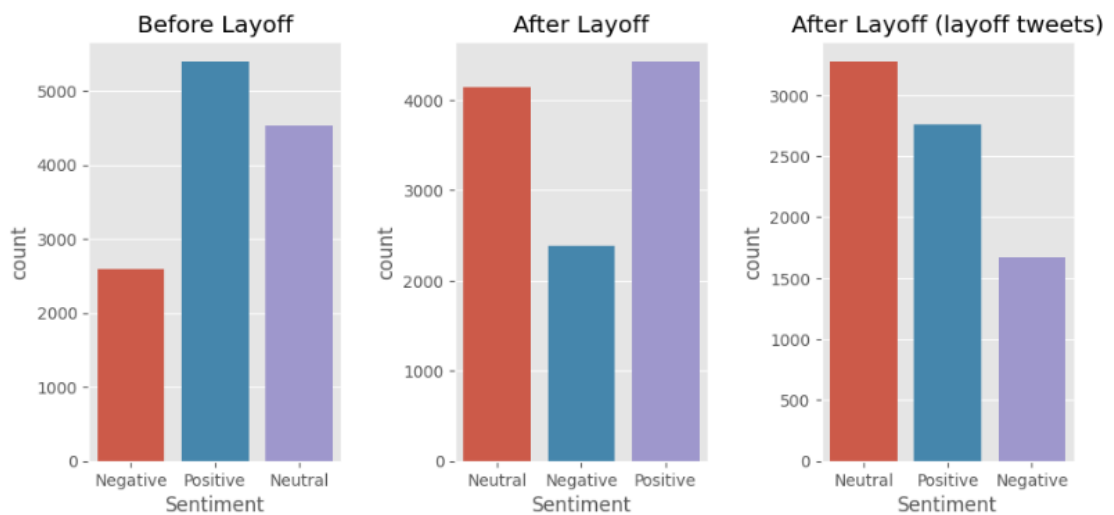


Figure 15: Count Plots of Sentiments

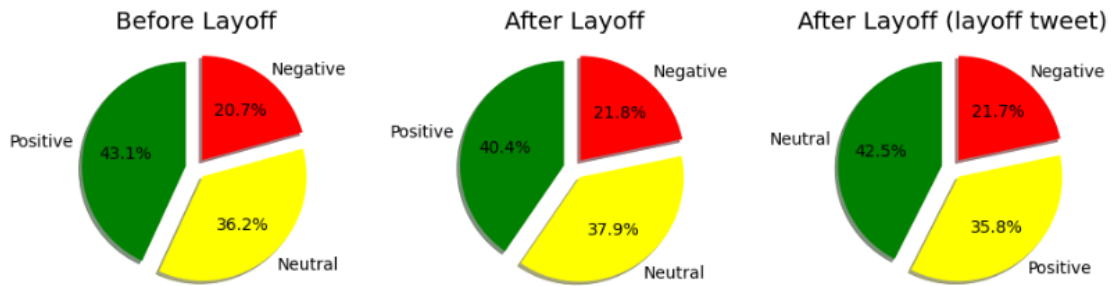


Figure 16: Pie Chart of Sentiments

The ratio of positive tweets has drastically decreased from 43.1% to 35.8%, while the percentage of negative tweets has only increased by 1%, as can be seen in Before Layoff and After Layoff (layoff tweet). We may now infer that most people had conflicting emotions regarding layoffs in Facebook based on the increase of neutral tweets, which went from 36.2% to 42.5%.

As previously mentioned, the timeline after layoffs occurred contains tweets regarding Facebook, but they were not filtered with the keyword "layoff" for which we created the middle pie chart, and the scraper we used determined which tweets it pulls and how many of them are regarding layoffs because no randomizer was used owing to snsrape's limitations. Currently, it also indicates a decline in NBR score, but if we take another dataset that is identical, it can show a better NBR score than the dataset from before the layoff, which is ideally inaccurate.

8. PROJECT LEGACY

8.1. Current Status of the Project

The team has achieved success in accomplishing the project by designing and building a comprehensive model that can execute a multitude of tasks ranging from extracting data from Twitter to executing machine learning algorithms to predict sentiments, assessing corporate reputation, and visualizing the results of the analysis. The project developed successfully can seamlessly encapsulate each step of the calculating corporate reputation from Twitter dataset, making it more efficient and effective.

Through the utilization of developed project, we successfully conducted a study to measure the effects of layoffs on the reputation of corporations. Our analysis enabled us to deduce the impact of such a decision on any company's image, which was then presented through visual aids such as graphs. As anticipated, the results indicated that layoffs have a negative impact on a company's reputation on social media. However, the effect observed was not as severe as expected. It can be inferred that most individuals are aware of the circumstances that led to the layoffs and have mixed emotions about it, which cannot be entirely characterized as positive or negative sentiments which play a crucial role in the calculation of corporate reputation.

8.2. Remaining Areas of Concern

One of the primary concerns associated with this project and Twitter sentiment analysis as a whole is the lack of appropriate datasets. When using the Twitter API, individuals are limited to retrieving tweets from only the past week. If they wish to obtain tweets from before this time period, they must target specific users. While third-party libraries such as snsrape claim to be able to retrieve historical tweets, they also have limitations of their own. To mitigate this issue, we attempted to write Python scripts to gather historical data, which were somewhat successful but not entirely as desired. Therefore, there is a need for an easy and convenient way to collect tweet data for the purpose of analysis.

When it comes to performing sentiment analysis on social media data, one of the major concerns is the informal way in which people express their views on these platforms.

There is significant use of sarcasm, irony, and the latest trend of relating things to memes, making it difficult to accurately determine their true sentiment. In order to address this challenge, it is necessary to develop models that are capable of identifying whether the text is straightforward or if it is conveying an irony or sarcasm, followed by an understanding of the intentions behind that irony. This will enable businesses to gain a more comprehensive understanding of how their customers truly feel about their products or services, which can be used to improve their offerings and customer experience.

Another issue with social media sentiment analysis is the potential lack of authenticity of opinions expressed on these platforms. As social media is widely accessible to everyone, people are free to share their opinions on anything, regardless of whether or not they have any legitimate reason to express their views. For example, an individual may criticize a specific product of a company without ever having used it, just because they have access to the platform and can share their thoughts. Therefore, it can be difficult to verify the credibility and accuracy of the opinions being expressed, especially when analysing consumer sentiment toward a certain product.

8.3. Technical and Managerial Lessons Learnt

Technical lessons:

Understanding technical requirements: Group projects require a deep understanding of the technical requirements of the project. This means that developers need to be able to identify the specific features that the project needs and understand how to implement them.

Learning technical skills: For developing a group project each team member has to learn some new technical skills in order to contribute to the project. Technical skills can be programming languages such as Python, or how to use an online IDE like google collab.

Testing and debugging: Group projects require testing and debugging to ensure that the final product is functional and free of errors. Developers should work collaboratively to identify and fix technical issues throughout the development process.

Documentation: Group projects require comprehensive documentation to ensure that all team members are on the same page and understand the project's technical requirements and progress.

Managerial Lessons:

Time management: Developing a group project requires time management skills to ensure that all tasks are completed on schedule.

Communication: Effective communication is essential for successful group project development. It is important to communicate clearly and effectively with team members to ensure that everyone is on the same.

Conflict resolution: Group projects may present conflicts between team members, such as disagreements over technical decisions or differences in work styles. Effective conflict resolution skills are essential to resolve these issues.

Collaboration: Group projects require effective collaboration and communication skills to ensure that all team members are working towards the same goals. It is important to collaborate effectively without plagiarizing each other's work.

Leadership: Developing a group project may require leadership skills to guide the team and ensure that everyone is working towards the same goals.

9. USER MANUAL

Pre-requisites:

- Internet Connectivity – A good and stable internet connectivity is required as we will be using an online IDE to run this project.
- Web Browser – We need a web browser to run the online IDE, any web browser will work but google chrome is preferred
- Google Account – As the IDE used to run this project is developed and managed by google, we need a google account to run all project.

After fulfilling the pre-requisites, we only need Google colab to run this project.

Note: As this project are nothing but python notebooks, you can use Jupyter notebook to run them or any other Python interpreter. (A few cells may need some changes based on the interpreter used)

Steps to run:

Step 1: Open any browser and search for google colab and open the first link (colab.research.google.com)

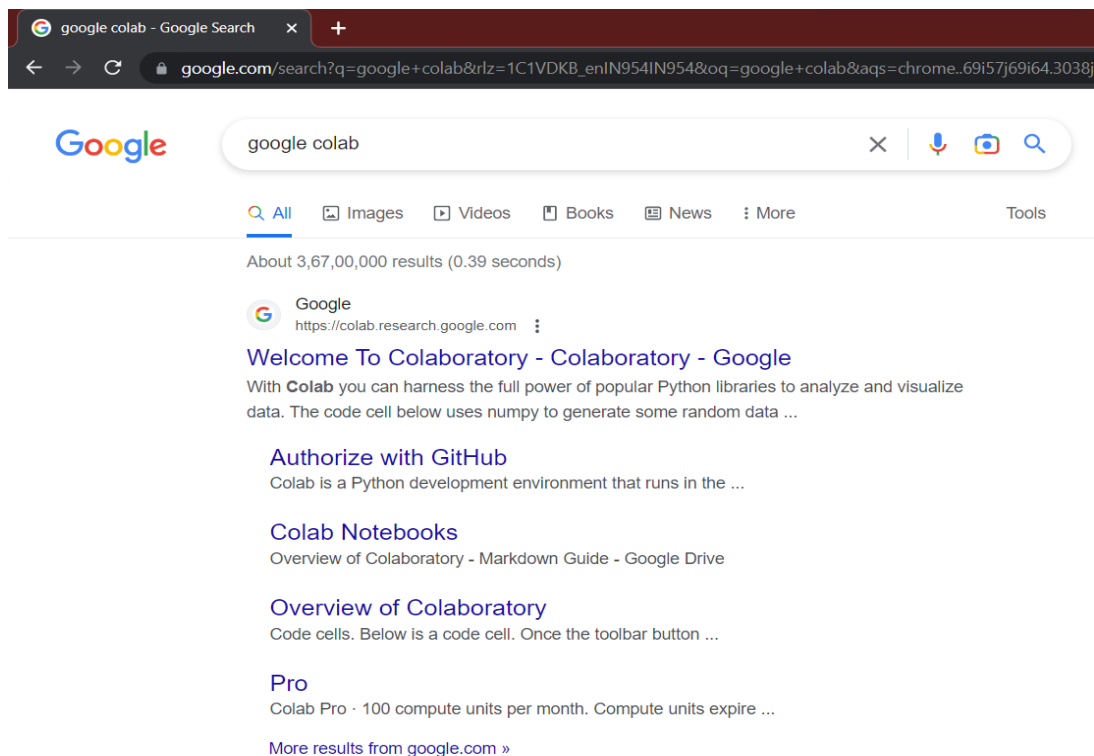


Figure 17: Step 1 – Open Google Collab

Step 2: Click on File and then Upload notebook (It will ask to sign in a google account if you haven't), then upload the notebook you want to run from your local device.

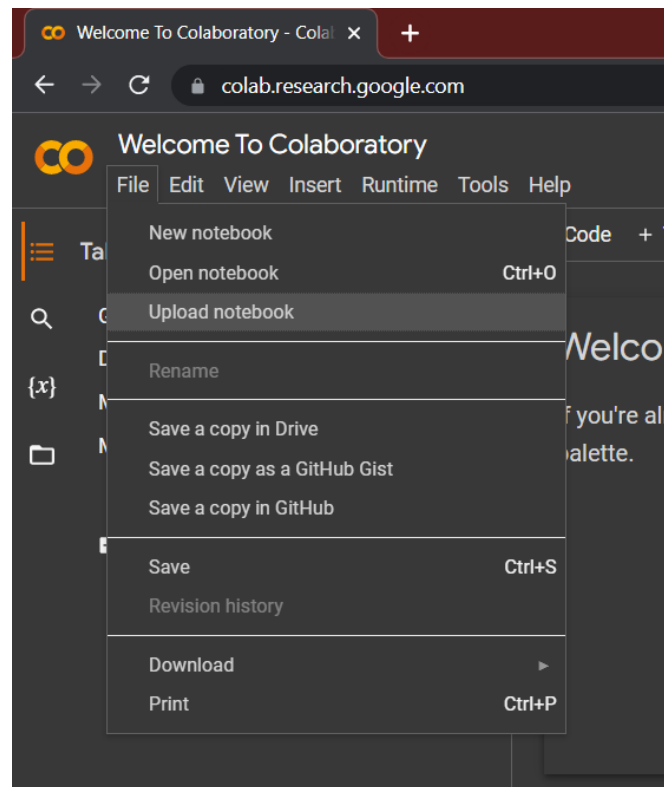


Figure 18: Step 2 – Upload notebook

Step 3: Before running cells, click on the Connect button on the left top corner so that google collab will connect to a runtime and allocate virtual space for variables and files.

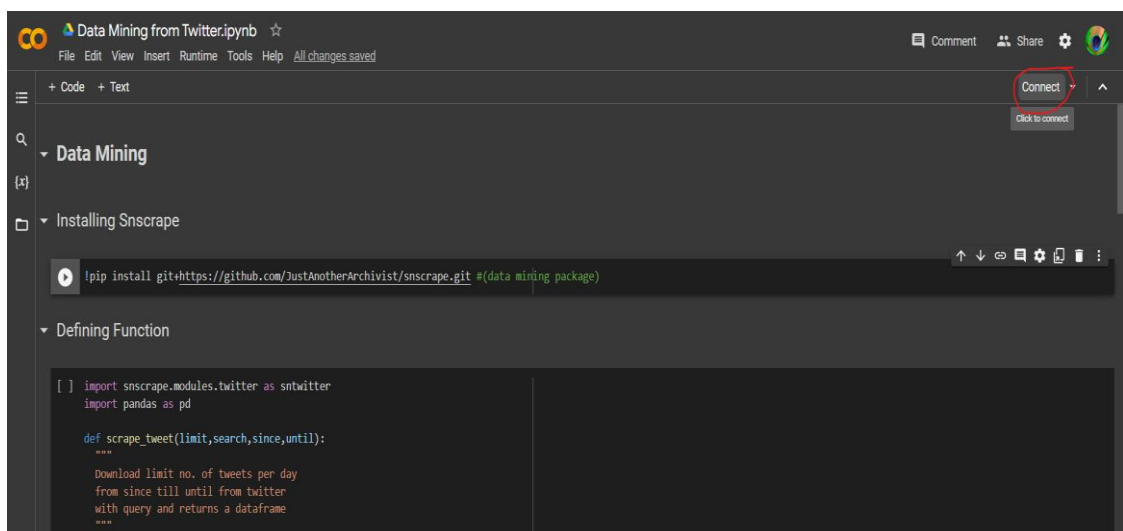
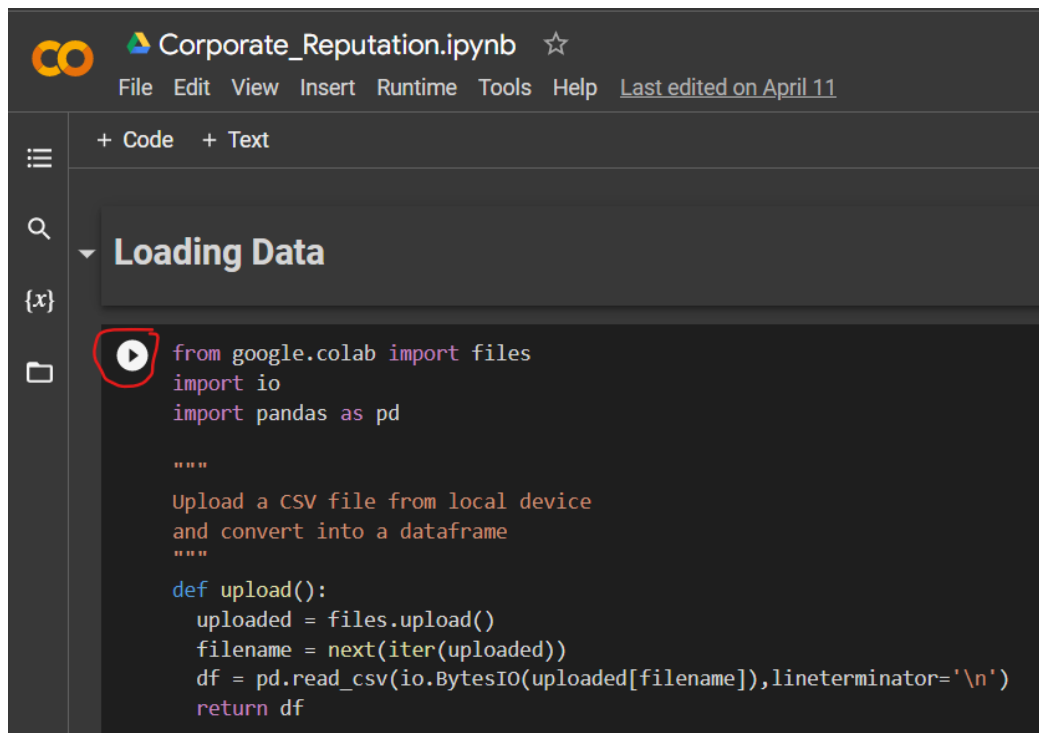


Figure 19: Step 3 – Connect Runtime

Step 4: To run a cell either click on the play button beside it or press Ctrl + Enter if you are inside that cell.



```

from google.colab import files
import io
import pandas as pd

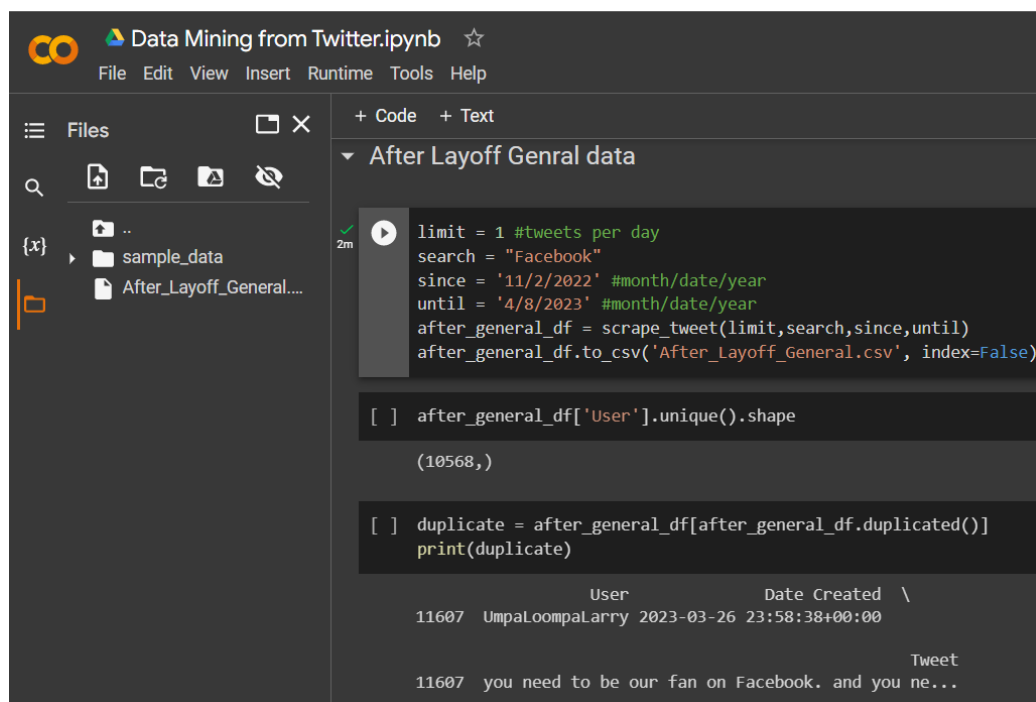
"""
Upload a CSV file from local device
and convert into a dataframe
"""

def upload():
    uploaded = files.upload()
    filename = next(iter(uploaded))
    df = pd.read_csv(io.BytesIO(uploaded[filename]), lineterminator='\n')
    return df

```

Figure 20: Step 4 – Run a Cell

Step 5: Saved CSV files after fetching the data can be found and downloaded from Files section of the page.



```

limit = 1 #tweets per day
search = "Facebook"
since = '11/2/2022' #month/date/year
until = '4/8/2023' #month/date/year
after_general_df = scrape_tweet(limit,search,since,until)
after_general_df.to_csv('After_Layoff_General.csv', index=False)

[ ] after_general_df['User'].unique().shape

(10568,)

[ ] duplicate = after_general_df[after_general_df.duplicated()]
print(duplicate)

      User      Date Created \
11607  UmpaLoompaLarry  2023-03-26 23:58:38+00:00

      Tweet
11607  you need to be our fan on Facebook. and you ne...

```

Figure 21: Step 5 – Saved Files

Step 5: Variables can be accessed in the Variable section {x}.

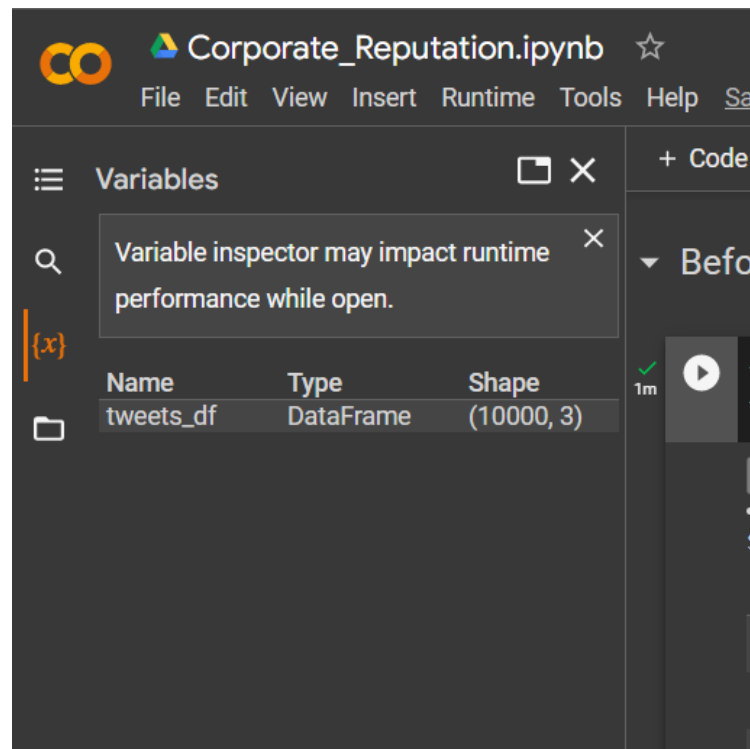


Figure 22: Step 6 – Variables

Step 5: To delete all the variables and files so that you can run the notebook from scratch you can Disconnect and delete runtime and the Connect again.

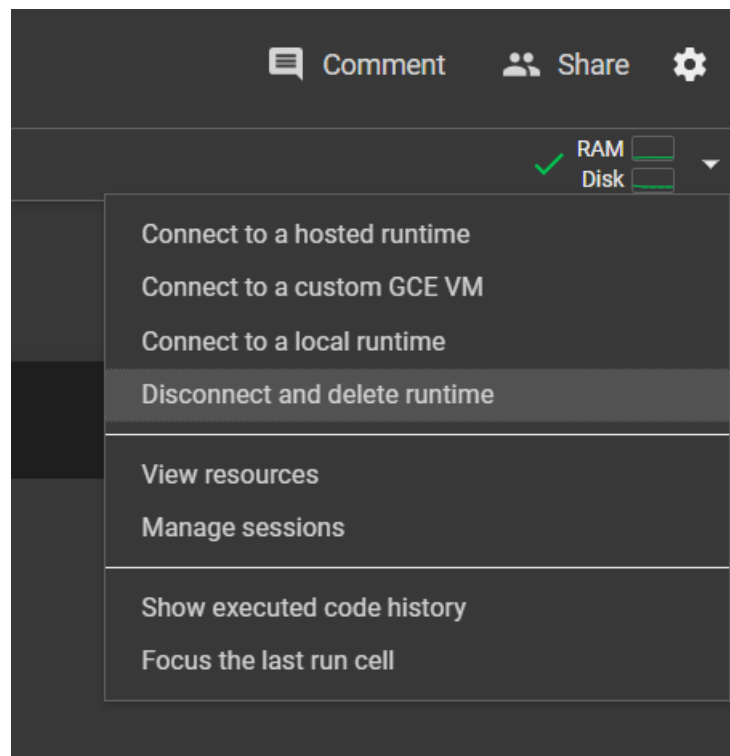


Figure 23: Step 7 – Disconnect and delete runtime

10. SOURCE CODE

10.1. Data Mining from Twitter

```
"""### Installing Snsrape"""

!pip install git+https://github.com/JustAnotherArchivist/snsrape.git


"""### Defining Function"""

import snsrape.modules.twitter as sntwitter

import pandas as pd

def scrape_tweet(limit, search, since, until):

    """

    Download limit no. of tweets per day

    from since till until from twitter

    with query and returns a dataframe

    """

    main_container = []

    dates = pd.date_range(start = since, end = until)

    for i in dates:

        query = "

        query += search

        date = i.date().strftime("%Y-%m-%d")

        query += " until:"

        query += date

        query += " exclude:retweets exclude:replies"

        mini_container = []
```

```

for tweet in sntwitter.TwitterSearchScraper(query).get_items():

    if len(mini_container) == limit:

        break

    else:

        if (tweet.lang == 'en'):

            mini_container.append([tweet.user.username, tweet.date, tweet.rawContent])

        main_container.extend(mini_container)

df = pd.DataFrame(main_container, columns = ["User", "Date Created", "Tweet"])

return df

"""### Training data"""

limit = 55 #tweets per day

search = "Facebook"

since = '1/2/2018' #month/date/year

until = '4/8/2023' #month/date/year

training_df = scrape_tweet(limit, search, since, until)

training_df.to_csv('Fb_Training.csv', index=False)

```

10.2. Training the ML Model

```

"""## **Data Importing**"""

from google.colab import files

import io

import pandas as pd

"""

```

Upload a CSV file from local device

and convert into a dataframe

```
"""
```

```
def upload():
```

```
    uploaded = files.upload()
```

```
    filename = next(iter(uploaded))
```

```
    df = pd.read_csv(io.BytesIO(uploaded[filename]),lineterminator='\n')
```

```
    return df
```

```
tweets_df = upload()
```

```
tweets_df.head()
```

```
"""## **Data Preprocessing**
```

```
### Importing Libraries
```

```
"""
```

```
## Data Manipulation
```

```
import pandas as pd
```

```
import numpy as np
```

```
## Text Preprocessing
```

```
!pip install contractions
```

```
import contractions
```

```
import nltk
```

```
nltk.download('stopwords') #for stopwords
```

```

nltk.download('wordnet') #for WordNetLemmatizer

nltk.download('punkt') #for word_tokenize

import re

from textblob import TextBlob

from nltk.tokenize import word_tokenize

from nltk.stem import WordNetLemmatizer

from nltk.corpus import stopwords

stop_words = set(stopwords.words('english'))

from sklearn.model_selection import train_test_split

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.feature_extraction.text import CountVectorizer

## Data Visualization

import seaborn as sns

from wordcloud import WordCloud

import matplotlib.pyplot as plt

from matplotlib import style

## Machine Learning Libraries

from sklearn.svm import SVC, LinearSVC

from sklearn.naive_bayes import MultinomialNB, GaussianNB

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix,
ConfusionMatrixDisplay

import warnings

warnings.filterwarnings("ignore")

```



```

"""### Deleting unwanted columns"""

tweets_df.columns

tweets_df = tweets_df.drop(['User', 'Date Created'], axis=1)

tweets_df.head()


"""### Data cleaning"""

def data_cleaning(tweet):

    # covert all text to lowercase

    tweet = tweet.lower()

    # remove all urls

    tweet = re.sub('http\S+|www\S+|https\S+', '', tweet, flags=re.MULTILINE)

    # remove @ user tags and #

    tweet = re.sub('@\w+|#', '', tweet)

    # remove emojis

    regex_pattern = re.compile(pattern = "["

        u"\U0001F600-\U0001F64F" # emoticons

        u"\U0001F300-\U0001F5FF" # symbols & pictographs

        u"\U0001F680-\U0001F6FF" # transport & map symbols

        u"\U0001F1E0-\U0001F1FF" # flags (iOS)

        u"\U00002702-\U000027B0"

        u"\U000024C2-\U0001F251"

        "]" + "", flags = re.UNICODE)

    regex_pattern.sub("", tweet)

    # remove numbers

```

```

tweet = ''.join(c for c in tweet if not c.isdigit())

# resolving contractions

expanded = []

for word in tweet.split():

    expanded.append(contractions.fix(word))

tweet = ' '.join(expanded)

# remove punctuations

tweet = re.sub('[^\w\s]', '', tweet)

# remove stop words

tweet_tokens = word_tokenize(tweet)

filtered_texts = [word for word in tweet_tokens if word not in stop_words]

# lemmatizing

lemma = WordNetLemmatizer()

lemma_texts = (lemma.lemmatize(text, pos='a') for text in filtered_texts)

return " ".join(lemma_texts)

tweets_df.Tweet = tweets_df['Tweet'].apply(data_cleaning)

"""### Checking for duplicate rows and deleting them"""

duplicate = tweets_df[tweets_df.duplicated()]

print(duplicate)

tweets_df = tweets_df.drop_duplicates('Tweet')

tweets_df.info()

"""### Calculating polarity"""

```

```

def polarity(Tweet):

    return TextBlob(Tweet).sentiment.polarity

tweets_df['Polarity'] = tweets_df['Tweet'].apply(polarity)

tweets_df.head(10)

"""### Labeling"""

def get_label(score):

    if score < 0:

        return 'Negative'

    elif score == 0:

        return 'Neutral'

    else:

        return 'Positive'

tweets_df['Sentiment'] = tweets_df['Polarity'].apply(get_label)

tweets_df.head()

"""### Saving the processed data"""

tweets_df.to_csv('Processed_tweets.csv',index=False)

"""### Plotting Word Cloud"""

words = ' '.join([tweets for tweets in tweets_df['Tweet']])

plt.figure(figsize = (20,15))

wordCloud = WordCloud(width = 1600, height = 800, random_state =
21).generate(words)

plt.imshow(wordCloud, interpolation = "bilinear")

```

```

plt.axis("off")

plt.show()

"""## **Machine learning**

### Data splitting

"""

x_train, x_test, y_train, y_test = train_test_split(tweets_df["Tweet"],
tweets_df["Sentiment"], test_size = 0.2, random_state = 50)

print(x_train,y_train)

"""### Feature extraction

#### Tfidf vectorizer

"""

vect = TfidfVectorizer(sublinear_tf=True).fit(x_train)

feature_names = vect.get_feature_names_out()

print("Number of features: {}".format(len(feature_names)))

print("First 50 features:\n {}".format(feature_names[:50]))

x_train = vect.transform(x_train)

"""### Linear Support Vector Classifier (SVM)"""

SVM_model = LinearSVC()

SVM_model.fit(x_train, y_train)

SVM_pred = SVM_model.predict(vect.transform(x_test))

SVM_accur = accuracy_score(SVM_pred, y_test)

```

```

print("Test accuracy: {:.2f}%".format(SVM_accur*100))

print(classification_report(y_test, SVM_pred))

style.use('classic')

conmat = confusion_matrix(y_test, SVM_pred, labels = np.unique(SVM_pred))

graph = ConfusionMatrixDisplay(confusion_matrix = conmat, display_labels =
np.unique(SVM_pred))

graph.plot()

"""### Saving the best performing model"""

import pickle

filename = 'final_trained_model.sav'

pickle.dump(SVM_model, open(filename, 'wb'))

```

10.3. Predicting Sentiments and Calculating Corporate Reputation

```

"""## **Loading Data**"""

from google.colab import files

import io

import pandas as pd

""" Upload a CSV file from local device and convert into a dataframe"""

def upload():
    uploaded = files.upload()
    filename = next(iter(uploaded))
    df = pd.read_csv(io.BytesIO(uploaded[filename]),lineterminator='\n')
    return df

"""###Loading Before Layoff Data"""

tweets_df = upload()

tweets_df.head()

```

```

"""## **Loading Trained ML Model**"""

import pickle
uploaded = files.upload()
filename = next(iter(uploaded))
model = pickle.load(open(filename, 'rb'))

"""### Deleting unwanted columns"""

def del_col(df):
    print(df.columns)
    return df.drop(['User', 'Date Created'], axis=1)

tweets_df = del_col(tweets_df)
layoffs_df = del_col(layoffs_df)
after_tweets_df = del_col(after_tweets_df)

"""### Data cleaning ( same as above section)"""

"""### Checking for duplicate rows and deleting them"""

def drop_dupli(df):
    duplicate = df[df.duplicated()]
    print(duplicate)
    return df.drop_duplicates('Tweet')

tweets_df = drop_dupli(tweets_df)
layoffs_df = drop_dupli(layoffs_df)
after_tweets_df = drop_dupli(after_tweets_df)

"""## **Predicting sentiments**"""

def pred_senti (df,proc_df):
    vect = TfidfVectorizer(sublinear_tf=True).fit(proc_df['Tweet'].values.astype('U'))
    tweets = vect.transform(df['Tweet'])
    sentiment = model.predict(tweets)

```

```

    return sentiment

processed_df = upload()
processed_df.head()

tweets_df['Sentiment'] = pred_senti(tweets_df, processed_df)
layoffs_df['Sentiment'] = pred_senti(layoffs_df, processed_df)
after_tweets_df['Sentiment'] = pred_senti(after_tweets_df, processed_df)

"""## **Calculating Corporate Reputation**"""

def Calculate_NBR(df):
    pos = df[df.Sentiment == 'Positive']
    pos_count = len(pos.index)
    neg = df[df.Sentiment == 'Negative']
    neg_count = len(neg.index)
    NBR = ((pos_count-neg_count)/(pos_count+neg_count))*100
    return NBR

NBR_before_layoff = Calculate_NBR(tweets_df)
NBR_only_layoff = Calculate_NBR(layoffs_df)
NBR_after_layoff = Calculate_NBR(after_tweets_df)
print(NBR_before_layoff,NBR_after_layoff,NBR_only_layoff)

"""## **Data Visualization**"""

"""### Bar Chart of NBR Comparison"""
Dataset = ['Before Layoff', 'After Layoff', 'After (layoff tweets)']
values = [NBR_before_layoff,NBR_after_layoff,NBR_only_layoff]
fig = plt.figure(figsize = (5, 5))
plt.bar(Dataset, values, color='maroon', width = 0.4)
plt.xlabel("Datasets")
plt.ylabel("Net Brand Reputation")
plt.title("Net Brand Reputation Comparisons")
plt.show()

```

11. BIBLIOGRAPHY

- [1] Kumar, R., Harshul., and Sujal. (2023). A Survey on Measuring Effects of Layoffs on Corporate Reputation. 7th International Joint Conference on Computing Sciences (ICCS-2023) [Accepted].
- [2] Syllaidopoulus, I., Skraparlis, A., and Ntalianis, K. (2022). Evaluating Corporate Online Reputation through Sentiment Analysis of News Articles: Threats, Maliciousness and Real Opinions. International Journal of Cultural Heritage (IJOCH).
- [3] Alsaeedi, A. and Khan, M. Z. (2019). A Study on Sentiment Analysis Techniques of Twitter Data. (IJACSA) International Journal of Advanced Computer Science and Applications. Vol. 10. No. 2.
- [4] Kaur, G. and Sharma, A. (2022). Comparison of Different Machine Learning Algorithms for Sentiment Analysis. (ICSCDS) International Conference on Sustainable Computing and Data Communication Systems.
- [5] Sharma, S. K., Daga, M., and Gemini, B. (2019). Twitter Sentiment Analysis for Brand Reputation of Smart Phone Companies in India. Proceedings of ICETIT 2019. Emerging Trends in Information Technology.