SENTIMENT ANALYSIS OF CUSTOMER REVIEWS ON SWIGGY

A project report submitted to ICT Academy of Kerala

in partial fulfillment of the requirements

for the certification of

CERTIFIED SPECIALIST

IN

Machine Learning and Artificial Intelligence

submitted by

ANAGHA V
ARJUN P KUMAR
LUBNAS S
NISHANA HAFSATH K T
SHILPA GRACE V
SUBITHA KUNNATH

ICT ACADEMY OF KERALA

THIRUVANANTHAPURAM, KERALA, INDIA
August 2023

List of Figures

List of Abbreviations

Table of Contents

ABSTRACT

Millions of people today use microblogging websites to voice their opinions. Thus, these microblogging websites offer a wealth of data. Knowing this kind of data might assist a company to identify areas that need improvement. Our objective is to analyze the reviews of customers on popular online food ordering services in India namely Swiggy and rank them in terms of highest positive and highest negative sentiments/reviews, by performing sentiment analysis on Twitter data.Sentiment analysis, a subfield of natural language processing, offers a powerful tool to extract insights from customer reviews by automatically determining the sentiment expressed within them, it can be a perception, appraisals, emotions, or attitude towards a topic, person, or entity. In this project, the dataset has been compiled by scraping public data on Twitter related to Swiggy.We have to calculate the subjectivity and polarity of a sentence to know whether the reviews are positive, negative, or neutral.Identifying the most commonly expressed sentiments, prevalent issues, and areas of customer satisfaction will allow Swiggy to make data-driven decisions to improve their service quality, address customer concerns, and enhance the overall customer experience.This calculation will help us to determine how customers accept swiggy in their daily life and valuable insights can be gained to drive continuous improvement and enhance customer satisfaction in the dynamic and competitive food delivery industry.

1. Problem Definition

1.1 Overview

The project focuses on sentiment analysis of customer reviews for Swiggy, utilizing Twitter data as our primary source. Through advanced natural language processing techniques, we will categorize sentiments as positive, negative, or neutral. The objective is to extract valuable insights to improve Swiggy's services, address customer feedback, and enhance overall user satisfaction. Our findings will offer data-driven recommendations to support Swiggy's growth and success in the food delivery industry.The project's methodology involves collecting a sizable dataset of customer reviews from Twitter, related to Swiggy, and subjecting the text to advanced sentiment analysis algorithms. The sentiment analysis process will classify each review based on the emotional tone, providing a comprehensive view of the collective sentiment around Swiggy's services.

1.2 Problem Statement

To perform sentiment analysis on customer reviews of Swiggy, extracted from Twitter data. We aim to automatically classify these reviews as positive, negative, or neutral sentiments. By doing so, we seek to identify trends, customer preferences, pain points, and areas of satisfaction related to Swiggy's food delivery services. The ultimate goal is to provide actionable insights to Swiggy, enabling them to make data-driven decisions for service optimization and improving overall customer experience.

2. Introduction

Sentiment analysis is increasingly used to gather and analyze data that includes a person's ideas, views, and feelings. Due to its emphasis on extracting valuable information from user feedback, this method is also known as "opinion mining." Intelligent techniques, natural language processing, and statistical models for characterizing qualities from massive amounts of data are used in sentiment analysis. Sentiment analysis is widely utilized in the business world, where it is used to gather consumer input on goods and services, evaluate it for clues about market trends, and predict future sales. Anyone with a Twitter account is able to write or read tweets, which are brief communications posted to the service. Twitter is being used to collect more unstructured, natural language data. Twitter sentiment analysis can also be used to probe further into underlying emotions by using lexical techniques, in addition to assessing whether tweets about a particular issue are good, negative, or neutral.

The main focus of this research is an analysis of Swiggy.Swiggy is a large and well-known food delivery business that primarily targets young adults (those between the ages of 18 and 35). Customers who are used to using apps on their smartphones are likewise catered to. Swiggy often incorporates the suggestions of its 1.98 million Twitter followers into the restaurant's operations.Numerous individuals find it helpful in gauging the attitudes of others.

As a result, several studies use lexicon-based methodologies for sentiment analysis. The manner that the acquired tweets' sentiment values are aggregated into positive and negative words is where it diverges from more conventional approaches. When computing emotions, the semantic orientation of words and sentences in a text is taken into account. It's more useful if you can accomplish it without setting aside a specific training batch of data. Therefore, the purpose of this research is to explore a sentiment analysis strategy for collecting consumer feedback on food delivery service apps like Swiggy.

3. Literature Survey

Sentiment Analysis is the task of classifying the polarity of a given text. For instance, text-based tweets can be categorized into either "positive", "negative", or "neutral"[1]. Given the text and accompanying labels, a model can be trained to predict the correct sentiment.Sentiment Analysis techniques can be categorized into machine learning approaches, lexicon-based approaches, and even hybrid methods[1]. Some subcategories of research in sentiment analysis include: multimodal sentiment analysis, aspect-based sentiment analysis, fine-grained opinion analysis, language specific sentiment analysis.More recently, deep learning techniques, such as RoBERTa and T5, are used to train high-performing sentiment classifiers that are evaluated using metrics like F1, recall, and precision.

Lexicon-based techniques were the first to be used for sentiment analysis.They are divided into two approaches:dictionary-based and corpus-based[2].In the former type,sentiment classification is performed by using a dictionary of terms,such as those found in SentiWordNet and WordNet. Nevertheless,corpus-based sentiment analysis does not rely on a predefined dictionary but on statistical analysis of the contents of a collection of documents,using techniques based on k-nearest neighbors(k-NN)[3], conditional random field(CRF)[4], andhiddenMarkovmodels (HMM) [5] , among others.

Machine-learning-based techniques[6]proposed for sentiment analysis problems can be divided into groups:(1) traditional models and (2) deep learning models.Traditional Models refer To classical machine learning techniques, such as the naïve Bayes classifier[7], maximum entropy classifier [8,9],or support vector machines (SVM) [10].The input to those algorithms include lexical features,sentiment lexicon-based features,parts of speech,or adjectives and verbs.The Accuracy of these systems depends on which features chosen.Deep Learning models can provide better results than traditional models.Different kinds of deep learning models can be used for sentiment analysis,including CNN,DNN,and RNN. Such approaches address classification problems at the document level,sentence level,oraspect level.These deep learning methods will be discussed in The following section.The Hybrid Approaches[11] combine lexicon-and machine-learning-based approaches.Sentiment lexicons commonly play a key role within majority of these strategies.

4. Dataset Description

Studies that perform sentiment analysis either generate their own data or use available datasets. Generating a new dataset makes it possible to use data that fits the problem the analysis is targeted at; moreover, the use of personal data ensures that no privacy laws are violated.However,the main drawback is having to label the dataset,which is a challenging task.Moreover,it is not always easy to generate a large volume of data.Our approach to selecting datasets was based on their availability and accessibility.Thus,we carefully chose datasets that are widely accepted by the research community.The dataset has been created by scraping public data on Twitter related to Swiggy. The features present in the dataset are 'date', 'favorite_count', 'followers_count', 'friends_count', 'full_text', 'retweet_count', 'retweeted', 'screen_name', 'tweet_id', 'user_id.The dataset is described below :-

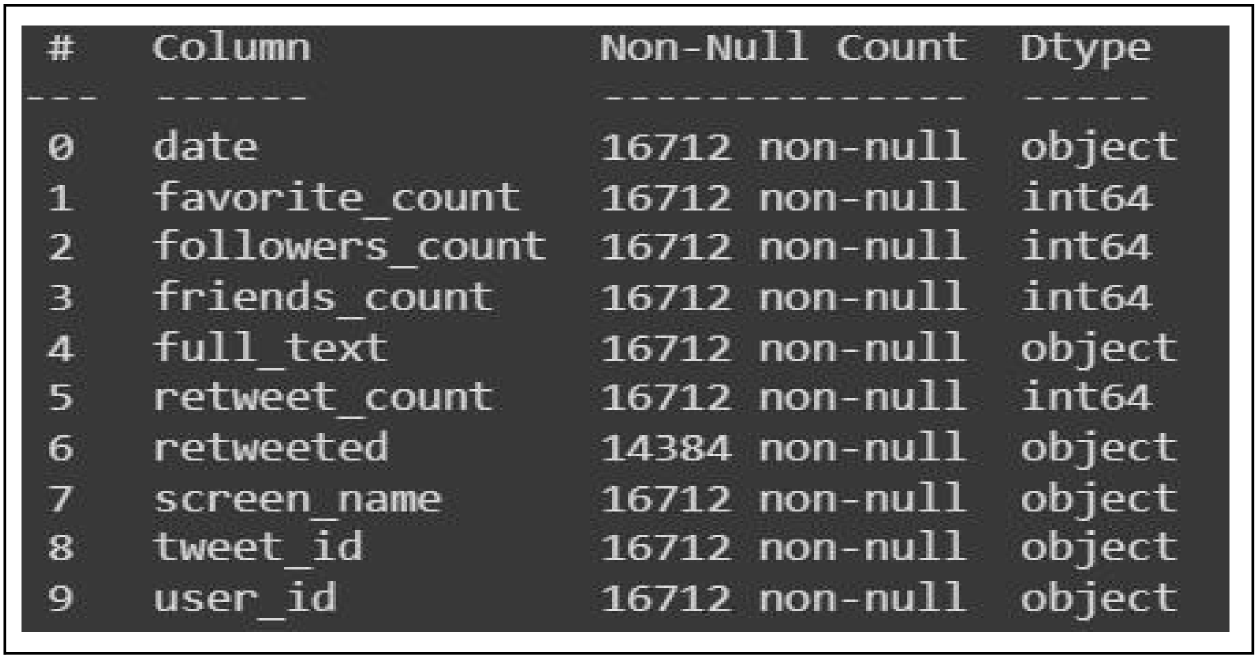
1. 'date': This feature refers to the date and time when the tweet was posted or recorded on Twitter.
2. 'favorite_count': It represents the number of times the tweet has been marked as a favorite (i.e., the "like" count).
3. 'followers_count': This feature indicates the number of followers the user who posted the tweet has on their Twitter account.
4. 'friends_count': It represents the number of users the Twitter account follows (i.e., the number of friends or accounts they follow).
5. 'full_text': This feature contains the actual text content of the tweet, which is the message the user posted on Twitter.
6. 'retweet_count': It represents the number of times the tweet has been retweeted by other users.
7. 'retweeted': This feature indicates whether the tweet is a retweet or not. It is usually a binary value, where 1 or "True" indicates a retweet, and 0 or "False" means it's an original tweet.
8. 'screen_name': This feature contains the Twitter username (handle) of the user who posted the tweet.

9. 'tweet_id': It is a unique identifier assigned to each tweet, allowing easy reference to a specific tweet.
10. 'user_id': This feature represents the unique identifier of the user account who posted the tweet. It helps link the tweet to a specific user in the dataset.

These features provide essential information about the tweets and the users who posted them, enabling various analyses and insights to be derived from the dataset.

We used the ‘full_text’ field for further processing.

The function df.info() provides a concise summary of the DataFrame's information.



```
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             16712 non-null  object
 1   favorite_count   16712 non-null  int64
 2   followers_count  16712 non-null  int64
 3   friends_count    16712 non-null  int64
 4   full_text        16712 non-null  object
 5   retweet_count    16712 non-null  int64
 6   retweeted        14384 non-null  object
 7   screen_name      16712 non-null  object
 8   tweet_id         16712 non-null  object
 9   user_id          16712 non-null  object
```

Figure 1: Dataset summary

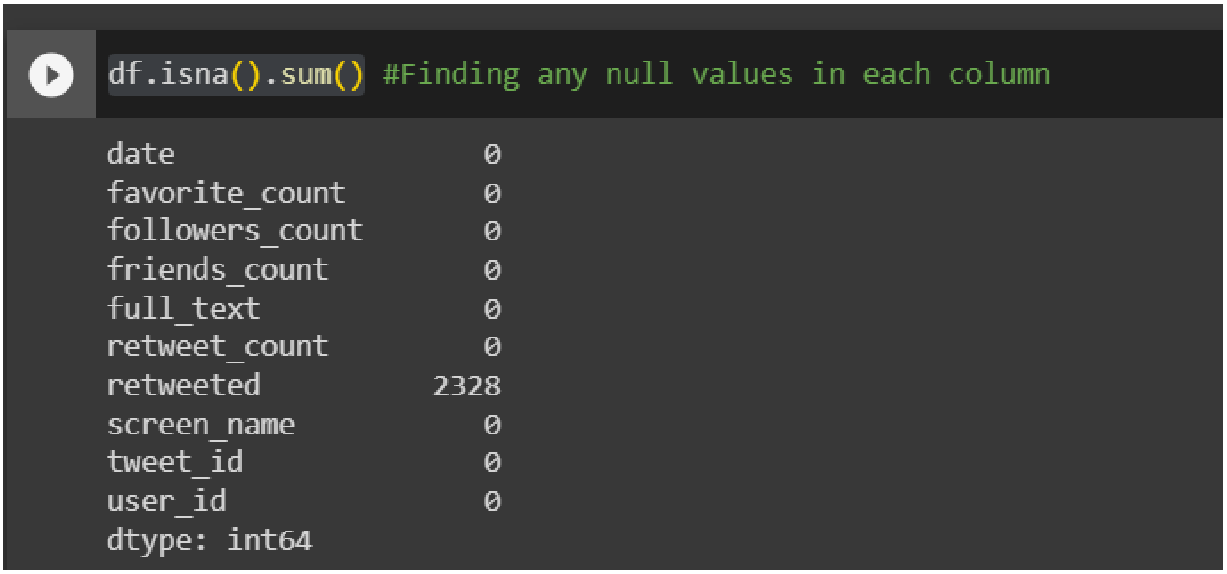
5. Data Preprocessing

Text cleaning is a preprocessing step aimed at removing words or components that lack relevant information and might consequently diminish the effectiveness of sentiment analysis. Text or sentence data typically contains whitespace, punctuation, and stopwords. The process of text cleaning involves several steps for sentence normalization. To ensure consistency across all datasets, the following steps were applied during the cleaning process: The dataset was cleaned using the following steps.

- Handling null values
- Time stamp splitting.
- Cleaning the texts by removing mentions such as @,#,and the links from the tweets.
- Text tokenization.
- Converting the text to lowercase.
- Text contractions are converted to their root form.
- Removing English stop words and punctuation.
- Stemming or lemmatization.

5.1 Handling null values

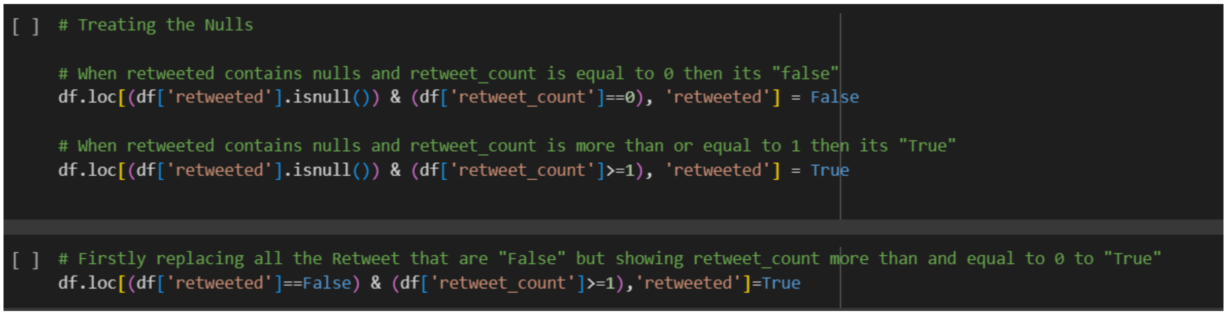
Missing Data can occur when no information is provided for one or more items or for a whole unit. Missing Data is a very big problem in real-life scenarios. Missing Data can also refer to as NA(Not Available) values in pandas. In DataFrame sometimes many datasets simply arrive with missing data, either because it exists and was not collected or it never existed.In Pandas missing data is represented by two value: None and NAN.In order to check missing values in Pandas DataFrame, we use a function isnull() this function return data frame of Boolean values which are True for NaN values.



```
df.isna().sum() #Finding any null values in each column

date                 0
favorite_count       0
followers_count      0
friends_count        0
full_text            0
retweet_count        0
retweeted         2328
screen_name          0
tweet_id             0
user_id              0
dtype: int64
```

Figure 2: Count of null values

Here the ‘retweeted’ column has 2328 null values.So we redefine that column based on ‘retweet_count’ column.If retweet_count greater than 1 retweeted will be True else False.



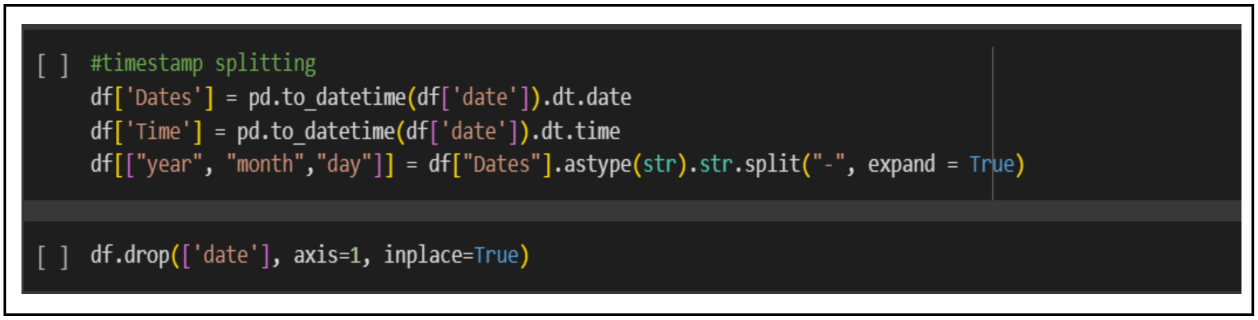
```
[ ]  # Treating the Nulls

     # When retweeted contains nulls and retweet_count is equal to 0 then its "false"
     df.loc[(df['retweeted'].isnull()) & (df['retweet_count']==0), 'retweeted'] = False

     # When retweeted contains nulls and retweet_count is more than or equal to 1 then its "True"
     df.loc[(df['retweeted'].isnull()) & (df['retweet_count']>=1), 'retweeted'] = True
```

```
[ ]  # Firstly replacing all the Retweet that are "False" but showing retweet_count more than and equal to 0 to "True"
     df.loc[(df['retweeted']==False) & (df['retweet_count']>=1),'retweeted']=True
```

Figure 3: Redefining the feature

5.2 Timestamp Splitting

Timestamp is the pandas equivalent of python’s Datetime and is interchangeable with it in most cases. It’s the type used for the entries that make up a DatetimeIndex, and other timeseries oriented data structures in pandas.For easy processing of datetime, splitted the timestamp to date,time,year,month,day.Also we dropped the original datetime column.



```
[ ]  #timestamp splitting
     df['Dates'] = pd.to_datetime(df['date']).dt.date
     df['Time'] = pd.to_datetime(df['date']).dt.time
     df[["year", "month","day"]] = df["Dates"].astype(str).str.split("-", expand = True)
```

```
[ ]  df.drop(['date'], axis=1, inplace=True)
```

Figure 4: Time-stamp Splitting

5.3 Cleansing

Removing unwanted characters like special symbols, emojis, and hashtags, which do not contribute to sentiment analysis.

5.4 Text Tokenization

Tokenization is the process of tokenizing or splitting a string, text into a list of tokens.word_tokenize() function is a wrapper function that calls tokenize() on an instance of the TreebankWordTokenizer class.

5.5 Normalization

Converts all uppercase characters into lowercase characters to maintain consistency.

5.6 Handling Contraction

Involves expanding the text to their full forms to ensure consistency and accuracy during analysis.

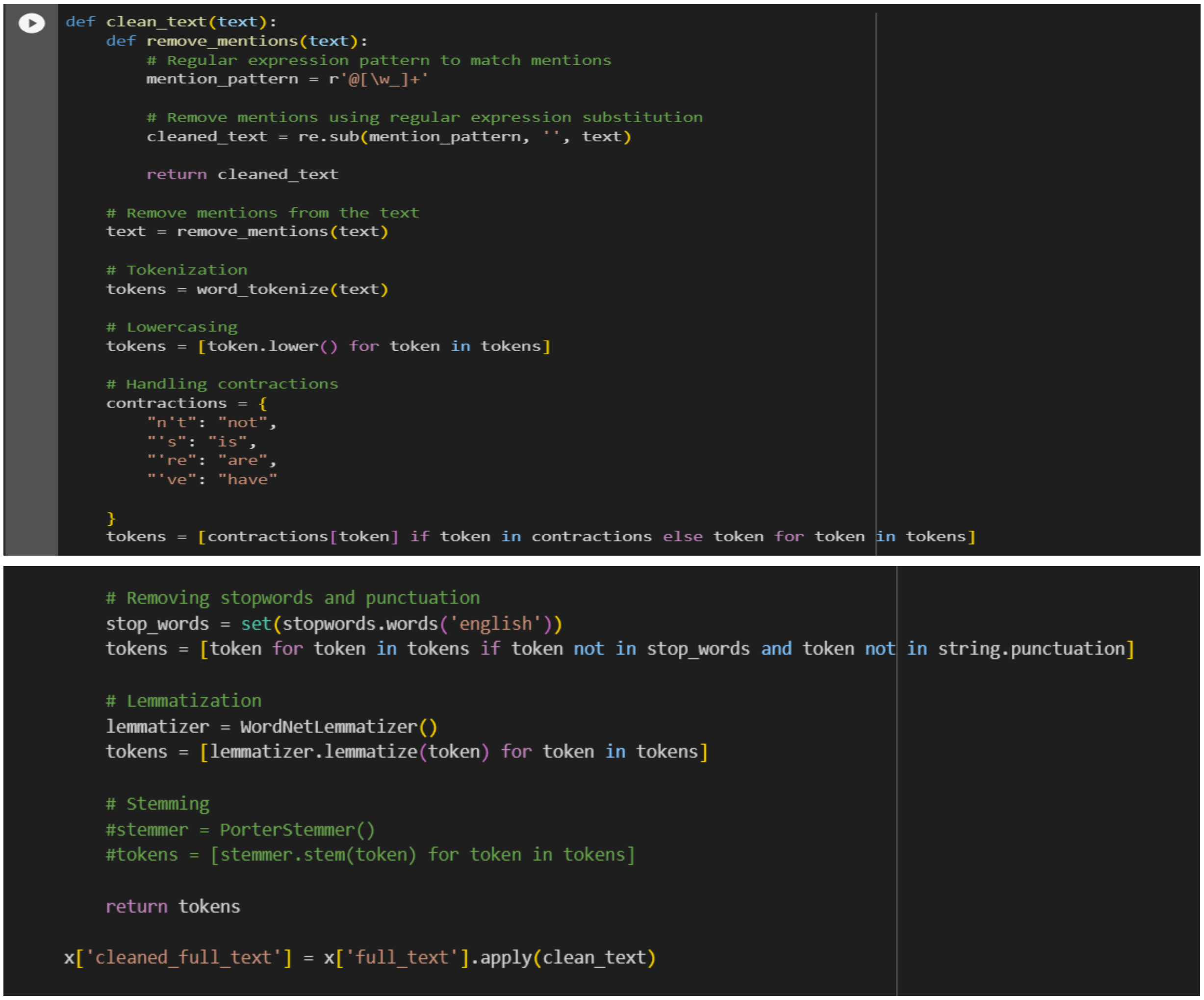
5.7 Removing stop words and punctuation

A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query. Import stopwords from nltk.corpus to remove stopwords and punctuations.

5.8 Lemmatization

In contrast to stemming, lemmatization is a lot more powerful. It looks beyond word reduction and considers a language's full vocabulary to apply a morphological analysis to words, aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma.There are 9 different approaches to Lemmatization but in this project wordnet is using.

Wordnet is a publicly available lexical database of over 200 languages that provides semantic relationships between its words. It is one of the earliest and most commonly used lemmatizer technique.It is downloaded from nltk package.



```
def clean_text(text):
    def remove_mentions(text):
        # Regular expression pattern to match mentions
        mention_pattern = r'@[\w_]+'

        # Remove mentions using regular expression substitution
        cleaned_text = re.sub(mention_pattern, '', text)

        return cleaned_text

    # Remove mentions from the text
    text = remove_mentions(text)

    # Tokenization
    tokens = word_tokenize(text)

    # Lowercasing
    tokens = [token.lower() for token in tokens]

    # Handling contractions
    contractions = {
        "n't": "not",
        "'s": "is",
        "'re": "are",
        "'ve": "have"

    }
    tokens = [contractions[token] if token in contractions else token for token in tokens]

    # Removing stopwords and punctuation
    stop_words = set(stopwords.words('english'))
    tokens = [token for token in tokens if token not in stop_words and token not in string.punctuation]

    # Lemmatization
    lemmatizer = WordNetLemmatizer()
    tokens = [lemmatizer.lemmatize(token) for token in tokens]

    # Stemming
    #stemmer = PorterStemmer()
    #tokens = [stemmer.stem(token) for token in tokens]

    return tokens

x['cleaned_full_text'] = x['full_text'].apply(clean_text)
```

Figure 5: Text Preprocessing

6. Exploratory Data Analysis

Let's suppose we want to create a data science project on the employee churn rate of a company. However, before building a model on this data, we need to analyze all the information present in the dataset. This analysis includes understanding the salary distribution of employees, the bonuses they receive, their starting time, and the teams they are assigned to. These steps of analyzing and modifying the data fall under the category of Exploratory Data Analysis (EDA).

Exploratory Data Analysis (EDA) is an approach that is used to analyze the data and discover trends, patterns, or check assumptions in data with the help of statistical summaries and graphical representations.

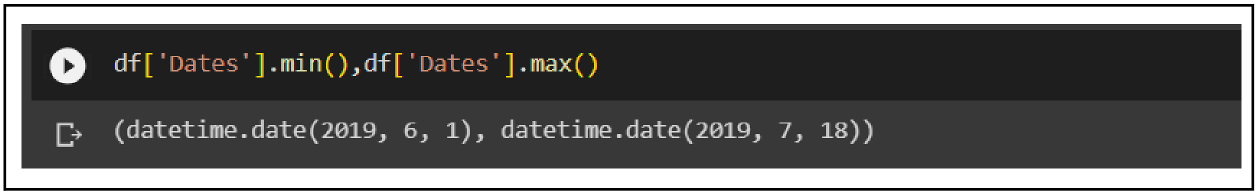
Types of EDA:Depending on the number of columns we are analyzing we can divide EDA into two types.

1. Univariate Analysis – In univariate analysis, we analyze or deal with only one variable at a time. The analysis of univariate data is thus the simplest form of analysis since the information deals with only one quantity that changes. It does not deal with causes or relationships and the main purpose of the analysis is to describe the data and find patterns that exist within it.
2. Bi-Variate analysis – This type of data involves two different variables. The analysis of this type of data deals with causes and relationships and the analysis is done to find out the relationship between the two variables.
3. Multivariate Analysis – When the data involves three or more variables, it is categorized under multivariate.

Depending on the type of analysis we can also subcategorize EDA into two parts.

1. Non-graphical Analysis – In non-graphical analysis, we analyze data using statistical tools like mean median or mode or skewness
2. Graphical Analysis – In graphical analysis, we use visualizations charts to visualize trends and patterns in the data

In this project, we are going to analyze how the swiggy is important in the daily life of a customer.First we find the min,max function for the ‘Dates’ column which gives the time period of tweets posted by customers.



```
df['Dates'].min(),df['Dates'].max()
```

```
(datetime.date(2019, 6, 1), datetime.date(2019, 7, 18))
```

Figure 6: Min-max for date

Next we group the tweets based on year and also find the count of tweets in corresponding year.In this dataset only one year present, i.e. from the year 2019 and have count of 16712 tweets.

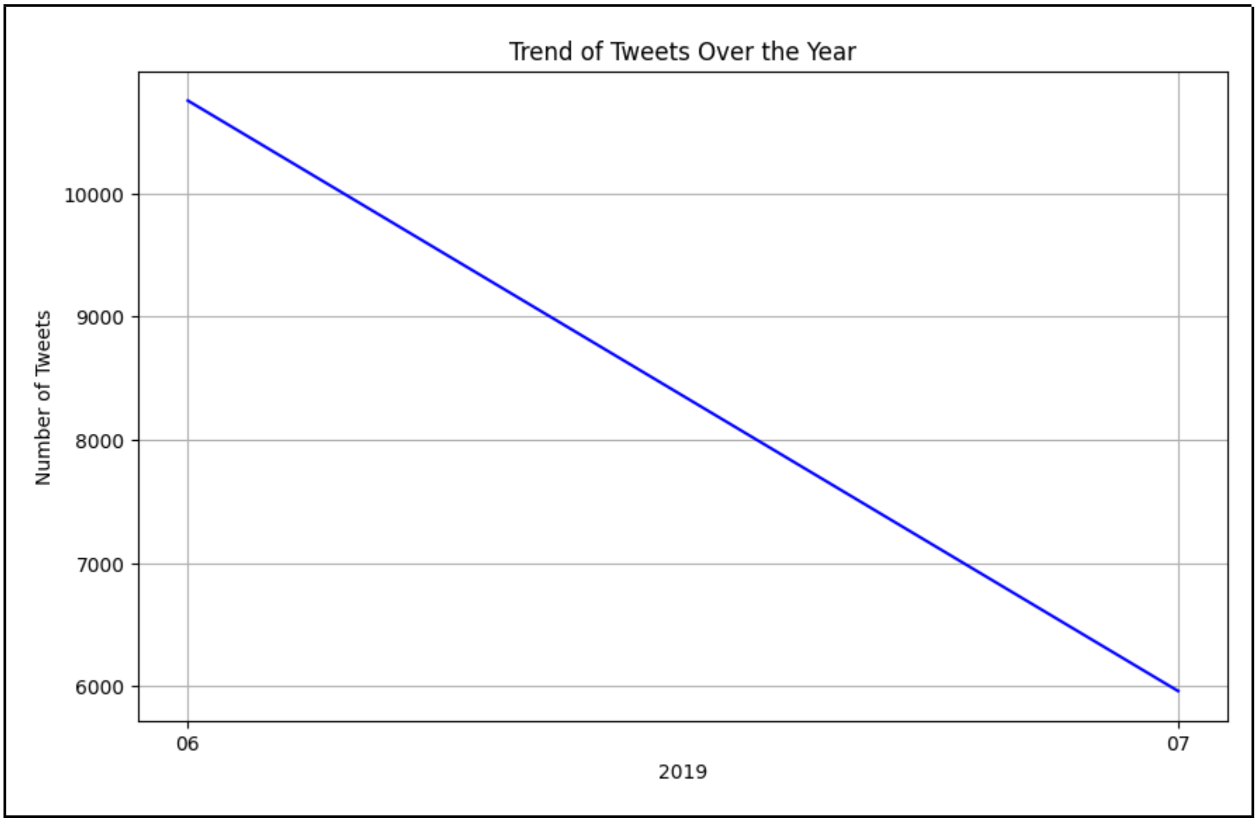


Figure 7: Trend of tweets over the year (2019)

Next, we grouped the tweets based on the ‘day’ and found the daily tweet count. Additionally, we print the maximum number of tweets and the date with the maximum number of tweets. According to the graph, it appears that the majority of the "Tweets" were posted on "2019-06-02".

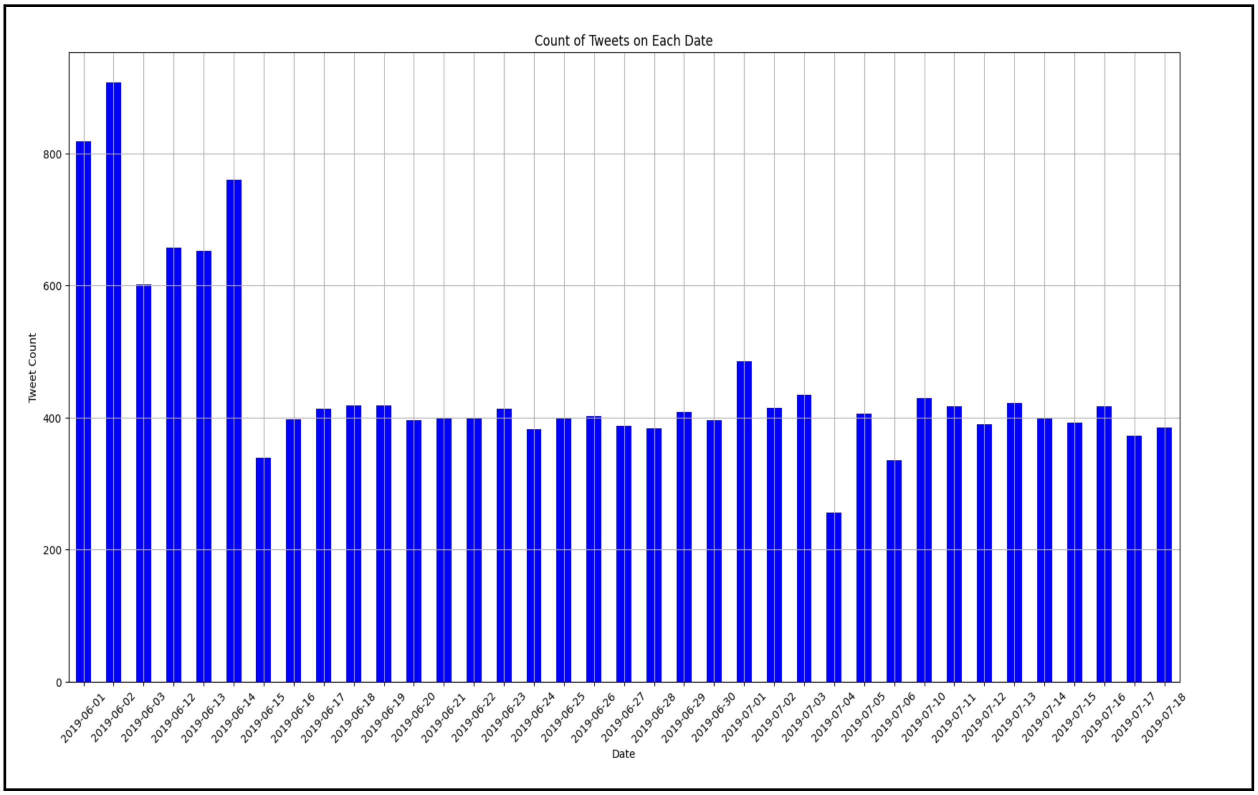


Figure 8: Daily tweet count throughout the year

A histogram is a graph showing frequency distributions, specifically the number of observations within each given interval .Plotting histogram for ‘followers_count’, ’friends_count’, ’retweet_count’ and ‘retweeted'.

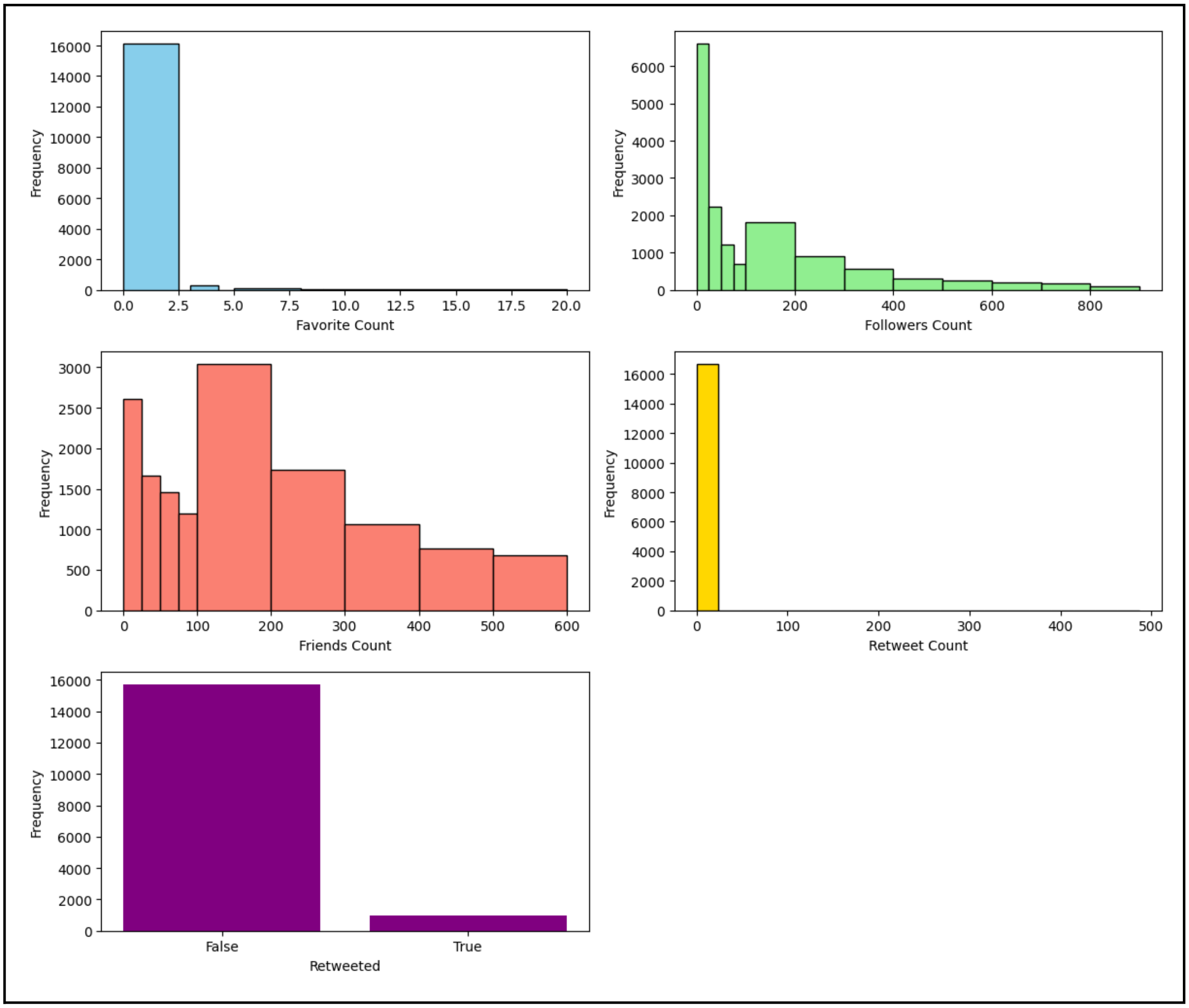


Figure 9 :Histogram

The next plot is a bar graph representing the feature 'month' based on value count. From this visualization, we can infer that this dataset may have been extracted between the months of July.

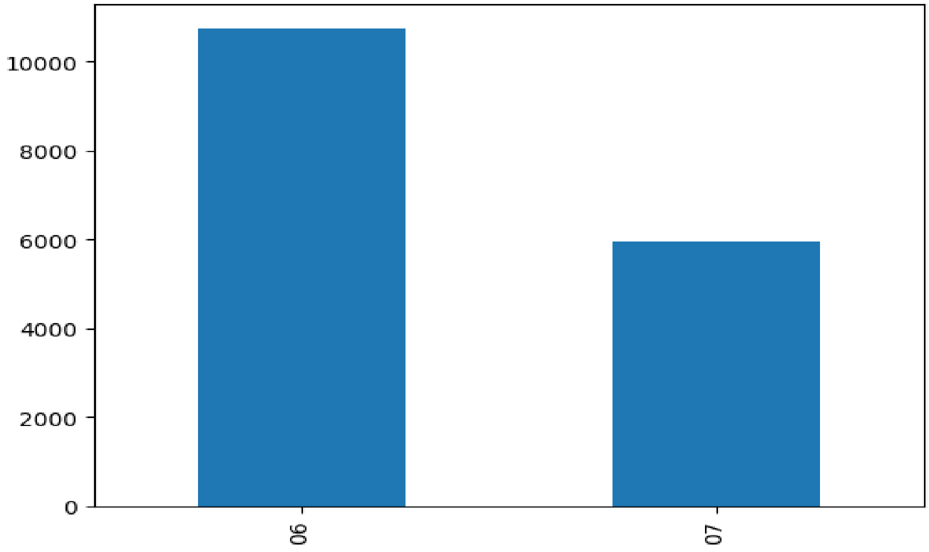
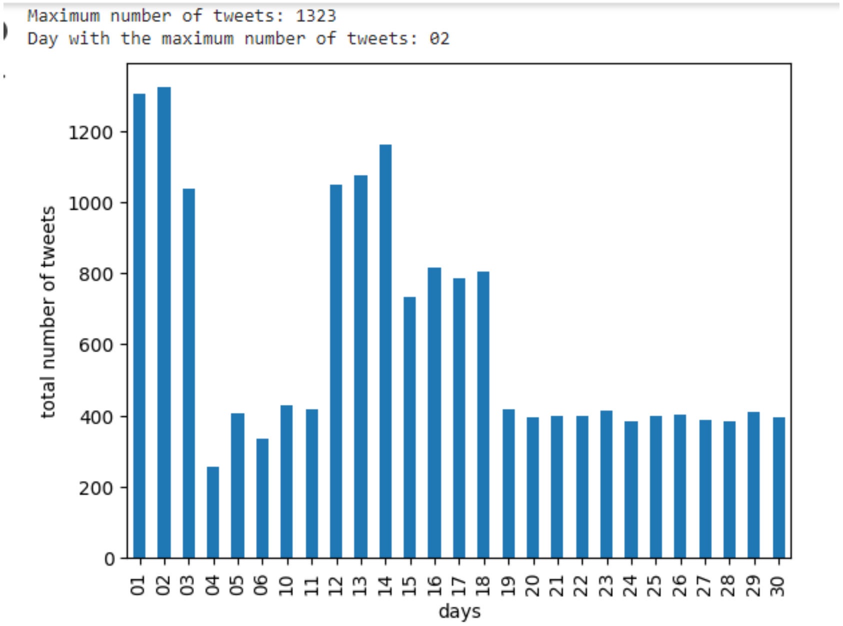


Figure 10: Tweets posted on each month

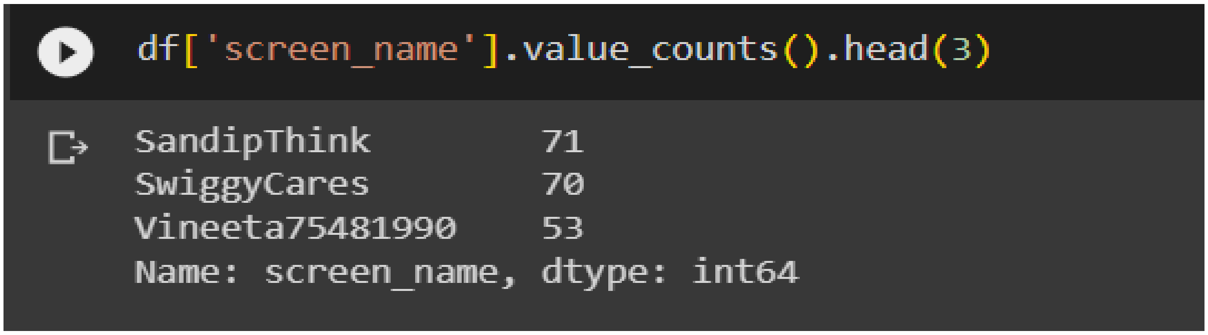
Now we can analyze Twitter's daily activity by plotting a bar graph based on the number of tweets posted each day. This will help the Media team identify the days with higher tweet volumes, enabling them to be more active strategically.



```
Maximum number of tweets: 1323
Day with the maximum number of tweets: 02
```

Figure 11: Daily tweet activity: A year in review

From the dataset, we can find the users who are more active on twitter based on their tweets.Also by using a bar graph, we can plot the top 10 users with respect to their tweets.After analyzing SandipThink is more active on twitter.



```
df['screen_name'].value_counts().head(3)

SandipThink        71
SwiggyCares        70
Vineeta75481990    53
Name: screen_name, dtype: int64
```

figure 12: Top 3 most active users

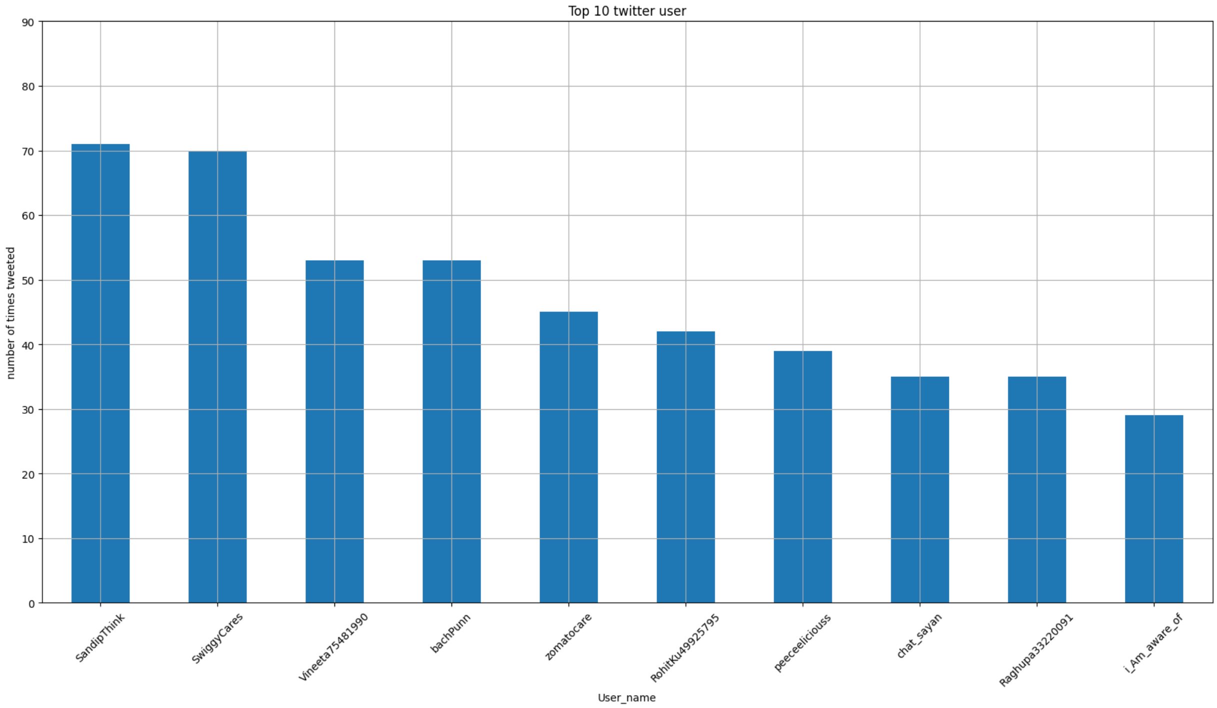
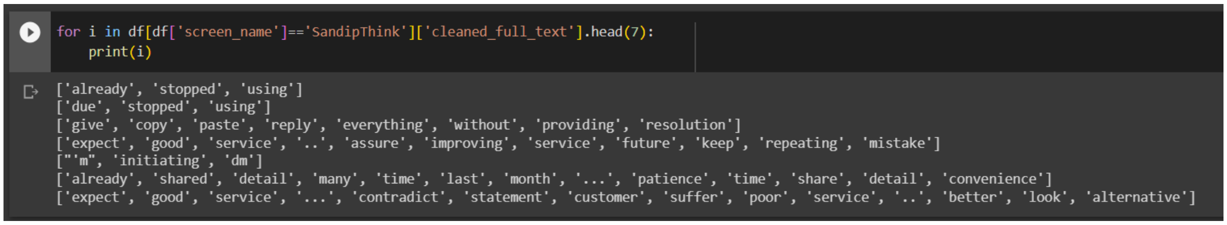


figure 13: Top 10 users based on tweet activity

‘SandipThink’ has tweeted the most regarding the performance of Swiggy.If there are any issues,the Swiggy team should promptly address these issues and ensure that other customers do not face similar problems in the future.



```
for i in df[df['screen_name']=='SandipThink']['cleaned_full_text'].head(7):
    print(i)
```

```
['already', 'stopped', 'using']
['due', 'stopped', 'using']
['give', 'copy', 'paste', 'reply', 'everything', 'without', 'providing', 'resolution']
['expect', 'good', 'service', '..', 'assure', 'improving', 'service', 'future', 'keep', 'repeating', 'mistake']
["'m", 'initiating', 'dm']
['already', 'shared', 'detail', 'many', 'time', 'last', 'month', '...', 'patience', 'time', 'share', 'detail', 'convenience']
['expect', 'good', 'service', '...', 'contradict', 'statement', 'customer', 'suffer', 'poor', 'service', '..', 'better', 'look', 'alternative']
```

Figure 14: Top 7 tweets from user ‘SandipThink’

For understanding trends in the data, we plotted a line graph between the dates of '2019-05-30' and '2019-07-20'. By analyzing the graph, we concluded that as Swiggy becomes more established, the initial rapid growth in tweet activity may naturally slow down due to market saturation. Tweet volumes may fluctuate based on seasonal factors, such as holidays or vacation periods, which can impact overall trends. Increased competition from other food delivery services may divert customer attention and reduce Swiggy-related tweet activity. Changes in service quality or occasional issues may lead to varying tweet volumes as customers express their experiences. Over time, loyal customers may become accustomed to Swiggy's service, resulting in fewer tweets as the novelty wears off.

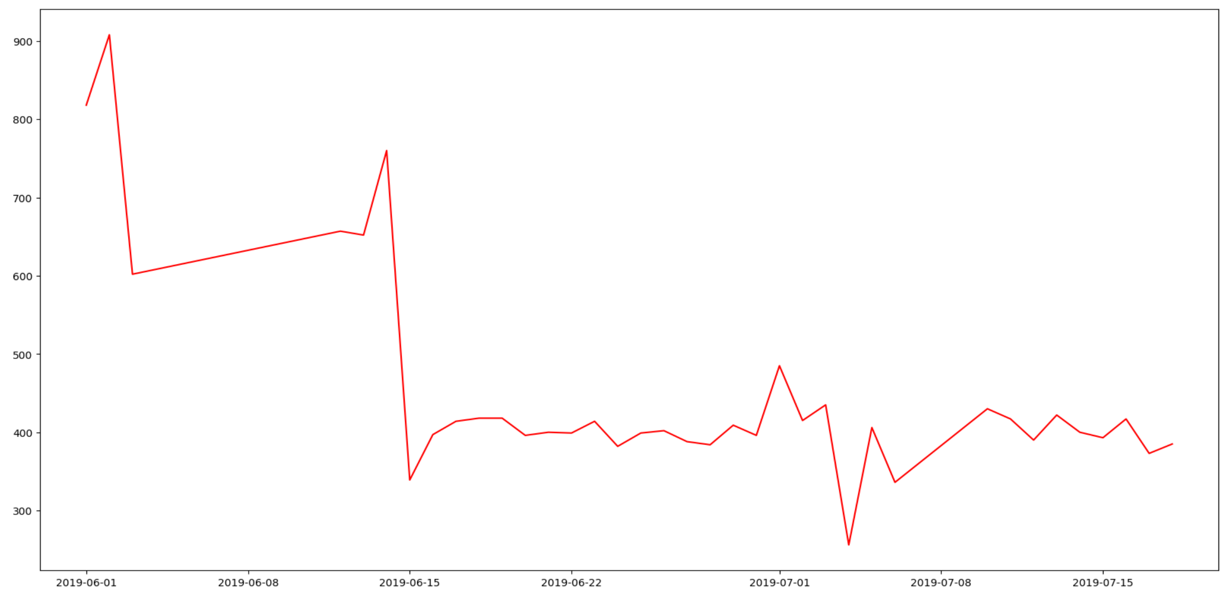
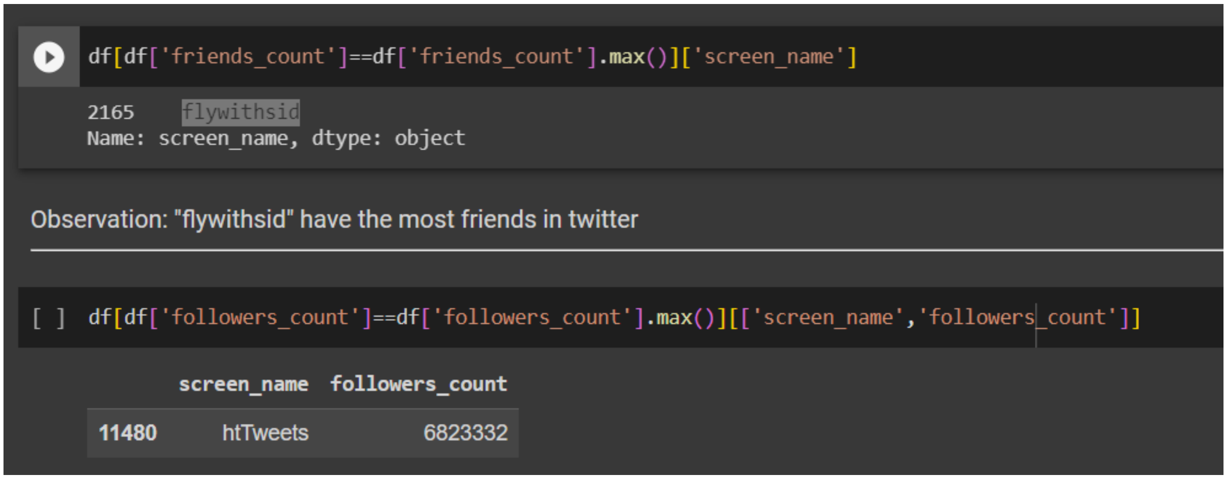


Figure 15: Monthly trends: Analyzing tweet activity over time

Now based on the ‘friend_count’ and ‘followers_count’,found out the top persons who have the most number of friends and followers on twitter using max() aggregation function.Based on the result, we observed that ‘flywithsid’ has the most friends on twitter and ‘htTweets’ has the most followers on twitter.



```
df[df['friends_count']==df['friends_count'].max()]['screen_name']

2165    flywithsid
Name: screen_name, dtype: object
```

Observation: "flywithsid" have the most friends in twitter

```
[ ]  df[df['followers_count']==df['followers_count'].max()][['screen_name','followers_count']]

         screen_name  followers_count
11480    htTweets             6823332
```

Figure 16: User with highest friends and followers

Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites.Using tweets present in dataset, word clouds are generated.

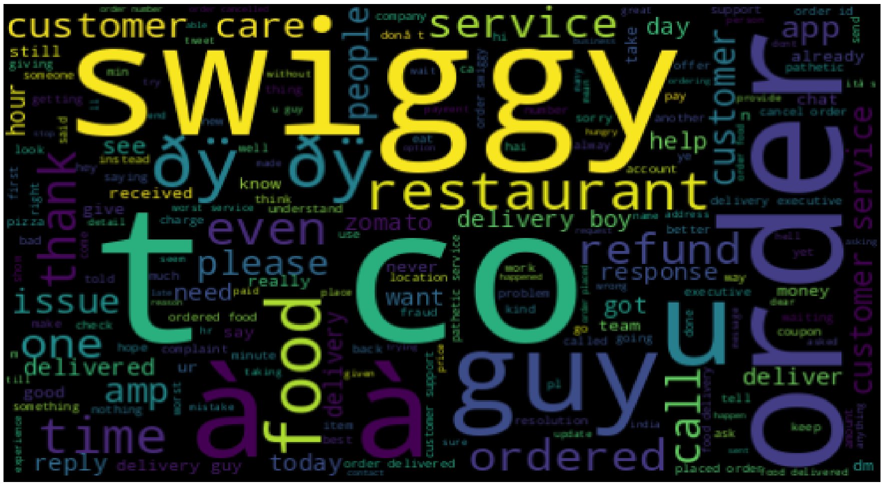


Figure 17: Word cloud : visual representation of trending words

7. Sentiment Analysis & Model Building

7.1 Sentiment Analysis

Sentiment analysis is the process of classifying whether a block of text is positive, negative, or neutral. The goal which Sentiment analysis tries to gain is to analyze people's opinions in a way that can help businesses expand. It focuses not only on polarity (positive, negative & neutral) but also on emotions (happy, sad, angry, etc.). It uses various Natural Language Processing algorithms such as Rule-based, Automatic, and Hybrid.

Sentiment analysis is the contextual meaning of words that indicates the social sentiment of a brand and also helps the business to determine whether the product they are manufacturing is going to make a demand in the market or not.

There are four approaches used:

1. Rule-based approach: Over here, the lexicon method, tokenization, and parsing come in the rule-based. The approach is to count the number of positive and negative words in the given dataset. If the number of positive words is greater than the number of negative words then the sentiment is positive else vice-versa.
2. Machine Learning Approach: This approach works on the machine learning technique. Firstly, the datasets are trained and predictive analysis is done. The next process is the extraction of words from the text. This text extraction can be done using different techniques such as Naive Bayes, Support Vector machines, hidden Markov model, and conditional random fields like this machine learning techniques are used.
3. Neural network Approach: In the last few years neural networks have evolved at a very rate. It involves using artificial neural networks, which are inspired by the structure of the human brain, to classify text into positive, negative, or neutral sentiments. It has Recurrent neural networks, Long short-term memory, Gated recurrent unit, etc to process sequential data like text.

4. Hybrid Approach: It is the combination of two or more approaches i.e. rule-based and Machine Learning approaches. The surplus is that the accuracy is high compared to the other two approaches.

For sentimental analysis for this dataset, only 2 columns named ‘Dates’ and ‘cleaned_full_text’ are taken.First find the polarity of text using TextBlob library.TextBlob module is a Python library and offers a simple API to access its methods and perform basic NLP tasks. It is built on the top of the NLTK module. Created a ‘polarity_score’ column using TextBlob and also defined sentiments for text based on values in polarity. If polarity is negative value sentiments are negative, if polarity is zero sentiments are neutral and if polarity is positive sentiments are positive.In sentiments values are redefined 0,1,2 respectively. We have plotted a pie chart for sentiments .

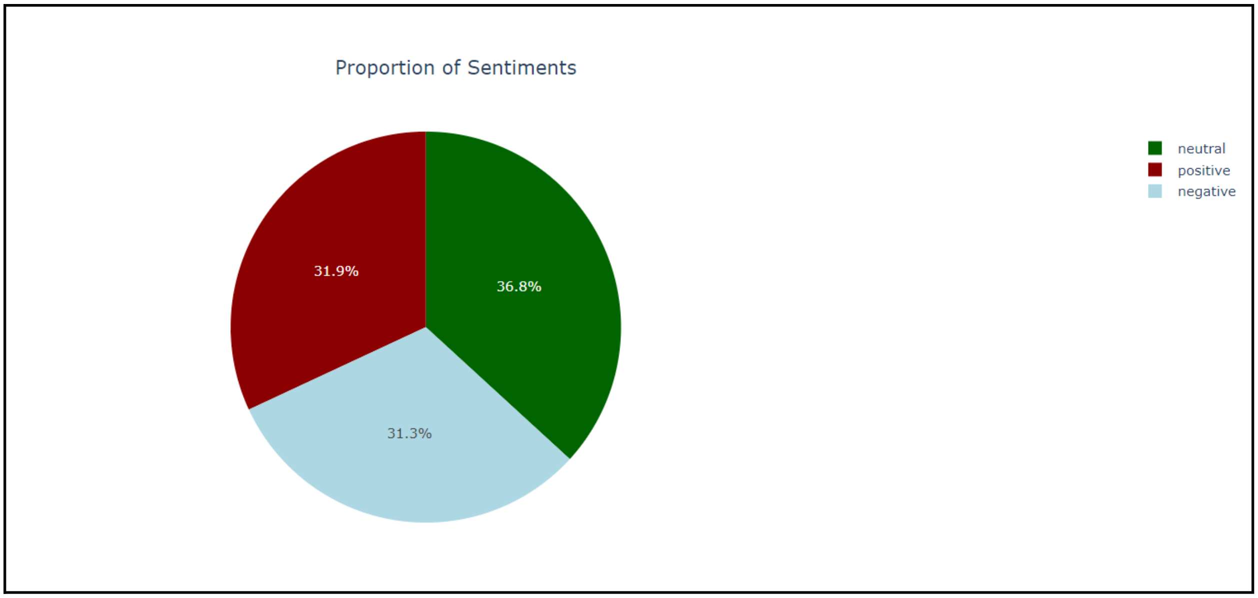


Figure 18: Pie chart :proportion of sentiments

7.2 Model Building

Train-Test split

The dataset is split into train and test using train test split. The training set is used to train the model and the testing set is used to test the model

Feature Extraction

Feature extraction refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. The process of feature extraction is useful when you need to reduce the number of resources needed for processing without losing important or relevant information. Feature extraction can also reduce the amount of redundant data for a given analysis. Also, the reduction of the data and the machine's efforts in building variable combinations (features) facilitate the speed of learning and generalization steps in the machine learning process.
Here we used two feature reduction methods - Bag of Words & TF-IDF

TF-IDF

TF-IDF stands for "Term Frequency — Inverse Document Frequency". This is a technique to quantify words in a set of documents. We generally compute a score for each word to signify its importance in the document and corpus. This method is a widely used technique in Information Retrieval and Text Mining.

Bag of Words/Count vectorization

The bag-of-words (BOW) model is a representation that turns arbitrary text into fixed-length vectors by counting how many times each word appears. This process is often referred to as vectorization. It's an algorithm that transforms the text into fixed-length vectors. This is possible by counting the number of times the word is present in a document. The word occurrences allow to compare different documents and evaluate their similarities for applications, such as search, document classification, and topic modeling.

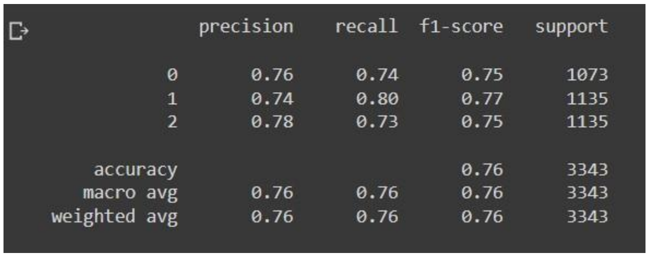
Machine learning algorithms used in both the models are

1. Naive Bayes
2. Random Forest
3. LinearSVC
4. Logistic Regression

Naive Bayes:

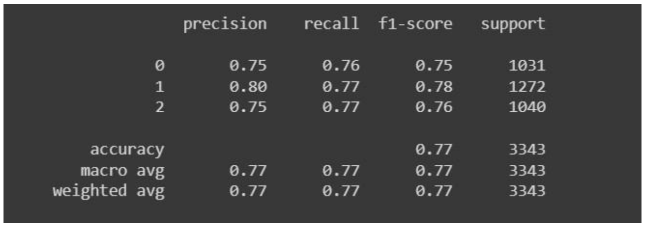
Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.It is mainly used in text classification that includes a high-dimensional training dataset.Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

Evaluation matrix



```
              precision    recall  f1-score   support

           0       0.76      0.74      0.75      1073
           1       0.74      0.80      0.77      1135
           2       0.78      0.73      0.75      1135

    accuracy                           0.76      3343
   macro avg       0.76      0.76      0.76      3343
weighted avg       0.76      0.76      0.76      3343
```

Fig 19: Naive Bayes algorithm using Bag of Words



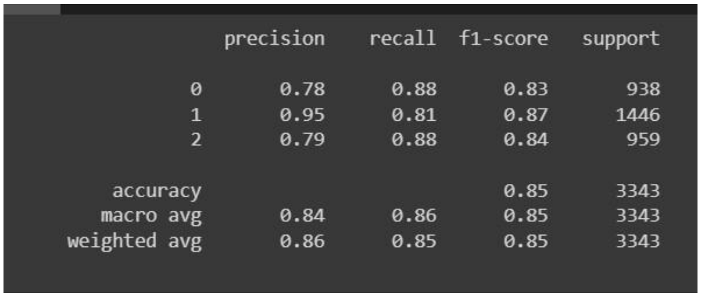
```
              precision    recall  f1-score   support

           0       0.75      0.76      0.75      1031
           1       0.80      0.77      0.78      1272
           2       0.75      0.77      0.76      1040

    accuracy                           0.77      3343
   macro avg       0.77      0.77      0.77      3343
weighted avg       0.77      0.77      0.77      3343
```

Fig 20:Naive Bayes algorithm using Tfi - df

Random Forest:

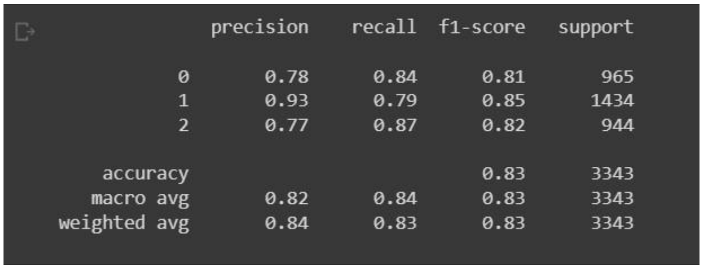
Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

Evaluation Matrix



```
              precision    recall  f1-score   support

           0       0.78      0.88      0.83       938
           1       0.95      0.81      0.87      1446
           2       0.79      0.88      0.84       959

    accuracy                           0.85      3343
   macro avg       0.84      0.86      0.85      3343
weighted avg       0.86      0.85      0.85      3343
```

Fig 21: Random Forest algorithm using Bag of Words



```
              precision    recall  f1-score   support

           0       0.78      0.84      0.81       965
           1       0.93      0.79      0.85      1434
           2       0.77      0.87      0.82       944

    accuracy                           0.83      3343
   macro avg       0.82      0.84      0.83      3343
weighted avg       0.84      0.83      0.83      3343
```

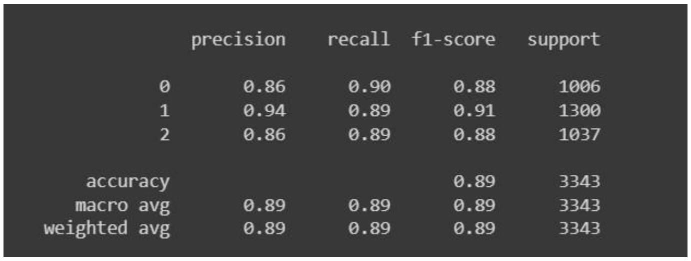
Fig 22: Random Forest algorithm using Tf - idf

LinearSVC:

Multi class text classification is one of the most common applications of NLP and machine learning. There are several ways to approach this problem and multiple machine learning algorithms perform relatively good depending on the quality of data. LinearSVC is one of the algorithms which performs quite well on a range of NLP based text classification tasks. However if the requirement is to have probability distribution over all the classes then LinearSVC in scikit-learn does not provide a function like predict_proba out of the box.

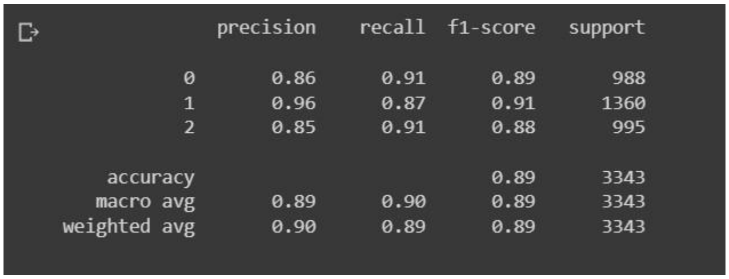
Linear SVC provides a decision_function method. The decision_function predicts the confidence scores for the samples. The confidence score for a sample is the signed distance of that sample to the hyperplane.

Evaluation Matrix



```
              precision    recall  f1-score   support

           0       0.86      0.90      0.88      1006
           1       0.94      0.89      0.91      1300
           2       0.86      0.89      0.88      1037

    accuracy                           0.89      3343
   macro avg       0.89      0.89      0.89      3343
weighted avg       0.89      0.89      0.89      3343
```

Fig 23: Linear SVC using Bag-of-Words



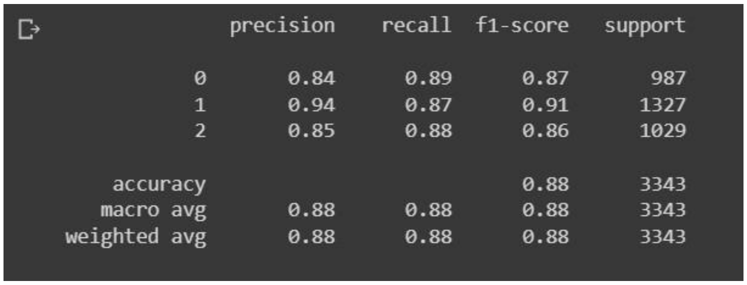
```
              precision    recall  f1-score   support

           0       0.86      0.91      0.89       988
           1       0.96      0.87      0.91      1360
           2       0.85      0.91      0.88       995

    accuracy                           0.89      3343
   macro avg       0.89      0.90      0.89      3343
weighted avg       0.90      0.89      0.89      3343
```

Fig 24: Linear SVC using Tf - idf

Logistic Regression:

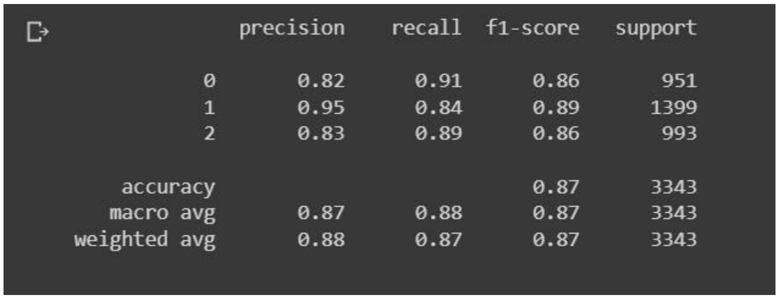
Logistic regression is a supervised machine learning algorithm mainly used for classification tasks where the goal is to predict the probability of an instance belonging to a given class. It is used for classification algorithms; its name is logistic regression. It's referred to as regression because it takes the output of the linear regression function as input and uses a sigmoid function to estimate the probability for the given class. The difference between linear regression and logistic regression is that linear regression output is the continuous value that can be anything while logistic regression predicts the probability that an instance belongs to a given class or not.

Evaluation Matrix



```
              precision    recall  f1-score   support

           0       0.84      0.89      0.87       987
           1       0.94      0.87      0.91      1327
           2       0.85      0.88      0.86      1029

    accuracy                           0.88      3343
   macro avg       0.88      0.88      0.88      3343
weighted avg       0.88      0.88      0.88      3343
```

Fig 25: Logistic Regression using Bag of Words



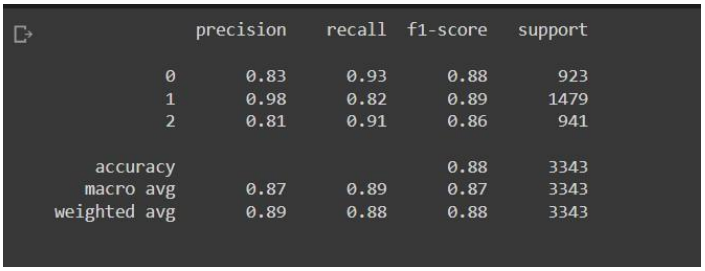
```
              precision    recall  f1-score   support

           0       0.82      0.91      0.86       951
           1       0.95      0.84      0.89      1399
           2       0.83      0.89      0.86       993

    accuracy                           0.87      3343
   macro avg       0.87      0.88      0.87      3343
weighted avg       0.88      0.87      0.87      3343
```

Fig 26: Logistic Regression using Tf - idf

XGBoost

XGBoost is an optimized distributed gradient boosting library designed for efficient and scalable training of machine learning models. It is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction. XGBoost stands for “Extreme Gradient Boosting” and it has become one of the most popular and widely used machine learning algorithms due to its ability to handle large datasets and its ability to achieve state-of-the-art performance in many machine learning tasks such as classification and regression.

Evaluation Matrix



```
              precision    recall  f1-score   support

           0       0.83      0.93      0.88       923
           1       0.98      0.82      0.89      1479
           2       0.81      0.91      0.86       941

    accuracy                           0.88      3343
   macro avg       0.87      0.89      0.87      3343
weighted avg       0.89      0.88      0.88      3343
```

Fig 27: XGBoost using Tf - idf

BERT

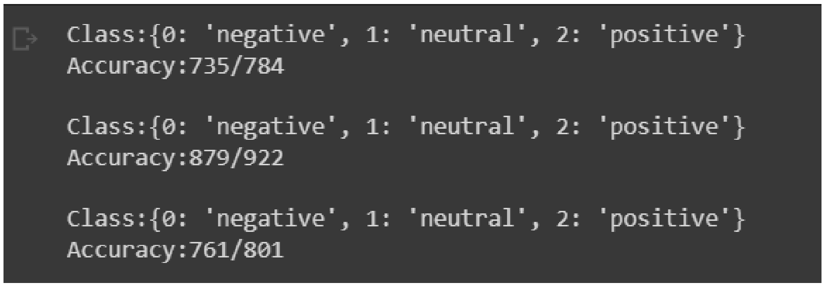
BERT (Bidirectional Encoder Representations from Transformers) is a Natural Language Processing Model proposed by researchers at Google Research in 2018. When it was proposed it achieve state-of-the-art accuracy on many NLP and NLU tasks such as:

- General Language Understanding Evaluation
- Stanford Q/A dataset SQuAD v1.1 and v2.0
- Situation With Adversarial Generations

BERT is basically an Encoder stack of transformer architecture. A transformer architecture is an encoder-decoder network that uses self-attention on the encoder side and attention on the decoder side. BERTBASE has 12 layers in the Encoder stack while BERTLARGE has 24 layers in the Encoder stack. These are more than the Transformer architecture described in the original paper (6 encoder layers). BERT architectures (BASE and LARGE) also have larger feedforward networks (768 and 1024 hidden units respectively), and more attention heads (12 and 16 respectively) than the Transformer architecture suggested in the original paper. It contains 512 hidden units and 8 attention heads. BERTBASE contains 110M parameters while BERTLARGE has 340M parameters.

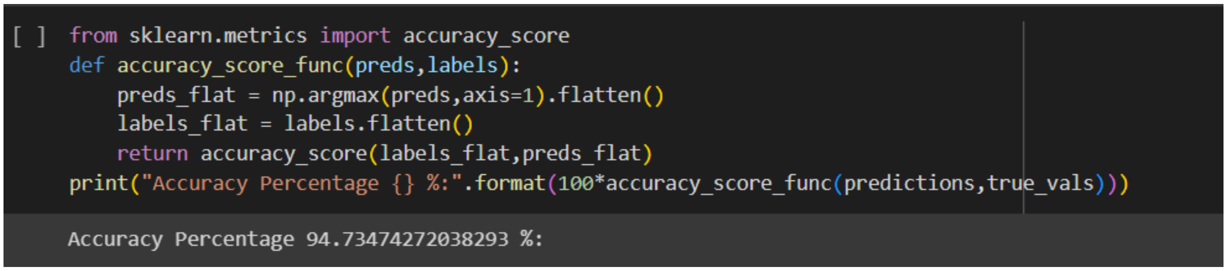
We have performed 6 epochs with 95% of f1_score and also calculated accuracy for each class.Using BERT model we got 94% of accuracy.

Evaluation Matrix



```
Class:{0: 'negative', 1: 'neutral', 2: 'positive'}
Accuracy:735/784

Class:{0: 'negative', 1: 'neutral', 2: 'positive'}
Accuracy:879/922

Class:{0: 'negative', 1: 'neutral', 2: 'positive'}
Accuracy:761/801
```

Fig 28: Accuracy per class



```
[ ] from sklearn.metrics import accuracy_score
    def accuracy_score_func(preds,labels):
        preds_flat = np.argmax(preds,axis=1).flatten()
        labels_flat = labels.flatten()
        return accuracy_score(labels_flat,preds_flat)
    print("Accuracy Percentage {} %:".format(100*accuracy_score_func(predictions,true_vals)))

    Accuracy Percentage 94.73474272038293 %:
```

Fig 29: Accuracy

Conclusion

In this project, we got a basic understanding of how Sentimental Analysis is used to understand public emotions behind people’s tweets. As you’ve read in this article, Twitter Sentimental Analysis helps us preprocess the data (tweets) using different methods and feed it into ML models to give the best accuracy. Twitter Sentimental Analysis is used to identify as well as classify the sentiments that are expressed in the text source. Logistic Regression, SVM, and Naive Bayes are some of the ML algorithms that can be used for Twitter Sentimental Analysis.Also we per BERT algorithm with 95% accuracy and selected as best model.

References

1. Nhan Cach Dang, María N. Moreno-García, Fernando De la Prieta(2020).Sentiment Analysis Based on Deep Learning: A Comparative Study. arXiv:2006.03541.
2. Salas-Zárate,M.P.;Medina-Moreira,J.;Lagos-Ortiz,K.;Luna-Aveiga,H.;Rodriguez-Garcia, M.A.;Valencia-García,R.J.C.Sentiment analysis on tweets about diabetes:Anaspect-level approach .Comput.Math. MethodsMed.2017,2017.
3. Huq,M.R.;Ali,A.;Rahman,A.Sentiment analysis on Twitter data using KNN and SVM.IJACSAInt.J.Adv. Comput.Sci.Appl.2017,8,19–25.
4. Pinto,D.;McCallum,A.;Wei,X.;Croft,W.B.Table extraction using conditional random fields.In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,Toronto,ON,Canada,28 July–1 August 2003;pp.235–242.
5. Soni,S.;Sharaff,A.Sentiment analysis of customer reviews based on hidden markov model.In Proceedings of the 2015 International Conference On Advanced Research in Computer Science Engineering & Technology (IRCSET 2015),Unnao,India,6 March 2015;pp.1–5.
6. Zhang,X.;Zheng,X.Comparison of Text Sentiment Analysis Based on Machine Learning.In Proceedings of the 2016 15th International Symposium Parallel and Distributed Computing(ISPDC),Fuzhou,China, 8–10 July 2016;pp.230–233.
7. Malik,V.;Kumar,A.Communication.Sentiment Analysis of Twitter Data Using Naive Bayes Algorithm. Int.J.RecentInnov.TrendsComput.Commun.2018,6,120–125.

8. Mehra,N.;Khandelwal,S.;Patel,P.Sentiment Identification Using Maximum Entropy Analysis Movie Reviews; StanfordUniversity:Stanford,CA,USA,2002.
9. Wu , H. ; Li ,J .;Xie, J .Maximum entropy-based sentiment analysis of online product reviews in Chinese. In Automotive, Mechanical and Electrical Engineering; CRC Press: Boca Raton , FL,USA, 2017; pp. 559 – 562.
10. FirminoAlves,A.L.;Baptista,C.d.S.;Firmino,A.;Oliveira,M.G.d.;Paiva,A.C.D.A Comparison of SVM versus naive-bayes techniques for sentiment analysis in tweets:A case study with the 2013FIFA confederations cup.In Proceedings of the 20th Brazilian Symposium on Multimedia and the Web ,JoãoPessoa, Brazil, 18–21 November 2014; pp.123 – 130.
11. Pandey,A.C.;Rajpoot,D.S.;Saraswat,M.Twitter sentiment analysis using hybrid cuckoo search method. Inf. Process.Manag.2017,53,764–779.