GEOGRAPHY AND TIMESTAMPS OF SPAM TWEETS

CSCE 5290

NATURAL LANGUAGE PROCESSING

PROJECT REPORT

GROUP 7

Github source: https://github.com/anith462/Anis462

Instructor: Dr. Zeenat Tariq (*Zeenat.Tariq@unt.edu*)

GROUP MEMBERS:

Vijayalakshmi pepala - 11656632

Monisha Mahitha boddu - 11513712

Mounika nuchu - 11653658

Anitha Nari - 11550658

TABLE OF CONTENTS:

Introduction:

The objective of this study is to conduct a sentiment analysis of tweets and investigate its geographical variations. The application of Natural Language Processing techniques will be utilized for data preprocessing and sentiment analysis. The data will be analyzed through the use of diverse plots and charts to derive significant insights. In psychology, the terms sentiment and emotion are sometimes used interchangeably. Sentiment is a mental attitude that is founded on an emotion, whereas emotion is a unique feeling. However, this distinction is rather hazy, which also applies to the line separating sentiment from opinion. Type, valence (positive, neutral, and negative), and intensity are three components that make up sentiment. It's vital to keep in mind that the thing to which such a sentiment can apply can also change a lot. Sentiment analysis can therefore be carried out at various granularities, i.e., in terms of whole texts, phrases, or items, such as individual words. Like other machine learning strategies, there are supervised (like straightforward decision trees) and unsupervised techniques. Recent years have seen profound.

Motivation:

Twitter stands apart from other online social networks due to its distinctive friend-following and posting features. One the one hand, friendships on Twitter aren't always reciprocal.

Users can "follow" celebrities, for instance, without expecting them to do the same. On the other hand, tweets or microblogs, which are text posts on Twitter, are limited to 140 characters.

Users are encouraged to often but indiscriminately post on everything, including feelings, pursuits, opinions, local news, etc. Due to the narrow definition of sentiment analysis as the Natural Language Processing (NLP) task that classifies a text as positive or negative, the terms polarity detection and sentiment analysis are frequently used synonymously. Each geo-tagged tweet's categories of adjacent locations were vectorized using one-hot encoding and frequency count encoding.

Each geo-tagged tweet's adjacent location was vectorized as an ego network as well. After that, the vectorized location and word embeddings were combined, and they were input into the CNN and BiLSTM networks to train and categorize the sentiment labels of the tweet. According to the experimental findings, our method outperformed using just word embeddings in terms of classification performance.

Significance:

The importance of determining the location and time stamp of spam tweets is multifaceted:

Spam campaign localization: Identifying the region of spam tweets makes it easier to determine the origin of spam campaigns. This data can be used to restrict spam from specific geographic areas or IP addresses.

Targeted attacks are frequently used by hackers to exploit vulnerabilities in certain countries or time zones. Security professionals can predict such assaults and take preventive measures by recognizing the geography and timing of spam tweets.

Spam trend analysis: Researchers can uncover trends and patterns in spam campaigns by studying the location and timing of spam tweets. This information can be utilized to create more effective spam countermeasures.

Objective:

Because tweets are brief and noisy, it is difficult to predict where people will be when they tweet or go home. It is customary to use location- and tweet-specific methods and

information while implementing recognition and disambiguation procedures for formal papers.

In this project, we developed a context-dependent sentiment classifier by combining the sentiment analysis of various authors, locations, times, and dates as determined by tagged Twitter data with conventional word-based sentiment classification methods. As far as we can determine, this hasn't been accomplished by major prior work on Twitter sentiment classification.

Background:

For people and organizations looking to grasp public opinion on various topics, social media platforms like Twitter have become a vital information source. Social media platforms generate a vast quantity of data, which gives researchers the chance to examine how people feel about a variety of topics. To automatically assess the sentiment of text documents, including tweets, sentiment analysis algorithms have been used. For the preprocessing and analysis of tweets, academics have recently used Natural Language Processing algorithms. This study uses Natural Language Processing tools to analyze twitter sentiment and look into its regional differences. To find observable patterns in sentiment across several geographic regions, the study also used geo-

sentiment analysis. The purpose of the study is to provide light on how people feel about various issues in various parts of the world. The findings of the study may have practical ramifications for corporations, governments, and other organizations that depend on public opinion to direct their operations.

Literature Review:

The study conducted by Diamantini et al. (2019) pertained to the enhancement of social information discovery through the utilization of sentiment analysis techniques. The researchers performed a sentiment analysis on user-generated content

(UGC) sourced from social media platforms, including Twitter and Facebook, with the aim of categorizing them into positive, negative, or neutral classifications. The research study also centered on examining the geographic dispersion of sentiment in user-generated content (UGC) to ascertain the influence of geographic location on sentiment. The findings of the research indicate that sentiment analysis methodologies possess the potential to identify user sentiment and augment the process of social information discovery.

The detection of non-personal and spam users on a geo-tagged Twitter network was investigated by Guo and Chen (2014). The research put forth a theoretical structure that integrates content-based and location-based characteristics for the purpose of distinguishing and categorizing Twitter users into personal or non-personal, and spam or non-spam groups. The efficacy of the suggested framework in identifying non-personal and spam users was also assessed in the study. The study's findings indicate that the framework proposed is capable of accurately detecting non-personal and spam users within a geo-tagged Twitter network.

Alfarrarjeh and colleagues (2017) conducted a research investigation concerning the analysis of sentiment in geospatial multimedia during disaster events. The proposed framework integrates multimedia content, specifically images and videos, with geospatial information and sentiment analysis to evaluate the sentiment of users in the context of disasters. The objective of the research was to facilitate disaster management teams in comprehending the real-time sentiment of the affected populace, with the aim of enhancing disaster response. The study's findings indicate that the proposed framework is proficient in analyzing users' sentiment during disasters and can offer significant insights to disaster management teams.

Implementation:

Running Jupyter Notebooks on your computer locally instead of through Google Colab has a number of benefits: Free GPU and TPU Access: For computation-intensive tasks, Google Colab offered free GPU and TPU access, which significantly sped up our training of machine learning models.

Colab notebooks are easily coupled with cloud storage services like Google Drive, enabling us to access datasets and other information from any location.

Pre-installed Libraries: Because it already has well-known libraries like NumPy, Pandas, and TensorFlow installed, we have saved time by not having to worry as much about installation and package dependencies on the local system.

Collaboration: We used Colab to easily share our notebook with our team members and work together on the project in real time.

There is no need for further software installation because Colab is a cloud-based platform.

Overall, Google Colab gave us an easy-to-use option for running Jupyter Notebooks, especially for the computationally intensive machine learning and data analysis tasks needed for this project.

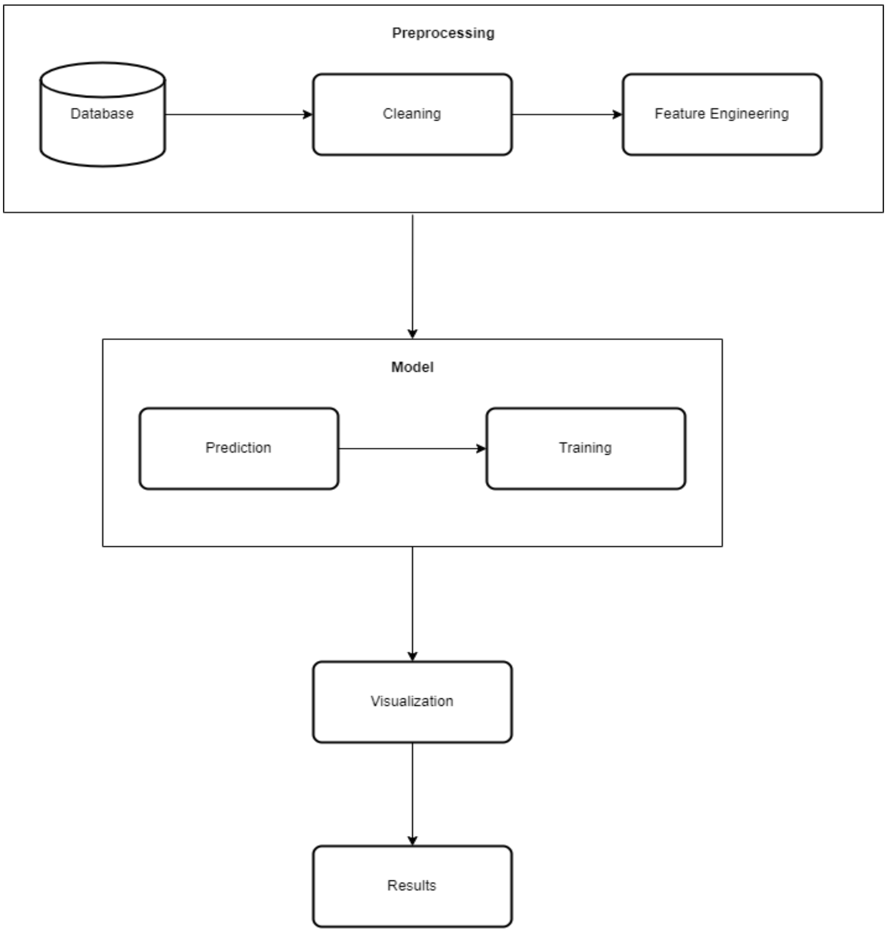
Project execution takes roughly 6 to 10 minutes in Google Colab, but it takes 15 to 17 minutes on a local PC.

Model:

Architecture Diagram:

The diagram below illustrates the architecture of our model.

Figure 1



Architecture Diagram

The architectural design comprises of three primary constituents, namely Preprocessing, Model, and Postprocessing.

The **Preprocessing** module retrieves tweets and associated metadata from a database and subsequently transmits them to the Cleaning component for the purpose of performing cleaning and transformation operations. Subsequently, the processed data is transmitted to the Feature Engineering component.

The **Model** component receives the preprocessed data and generates a model by training it with diverse algorithms. Subsequently, the proficiently trained model is employed to forecast the sentiment of tweets within the Prediction module.

The **Postprocessing** module is responsible for producing visual representations through the utilization of diverse plots and charts based on the predictions.

Subsequently, the aforementioned visualizations are exhibited to the **end-user**.

Workflow Diagram:

The diagram below illustrates the workflow of our model.

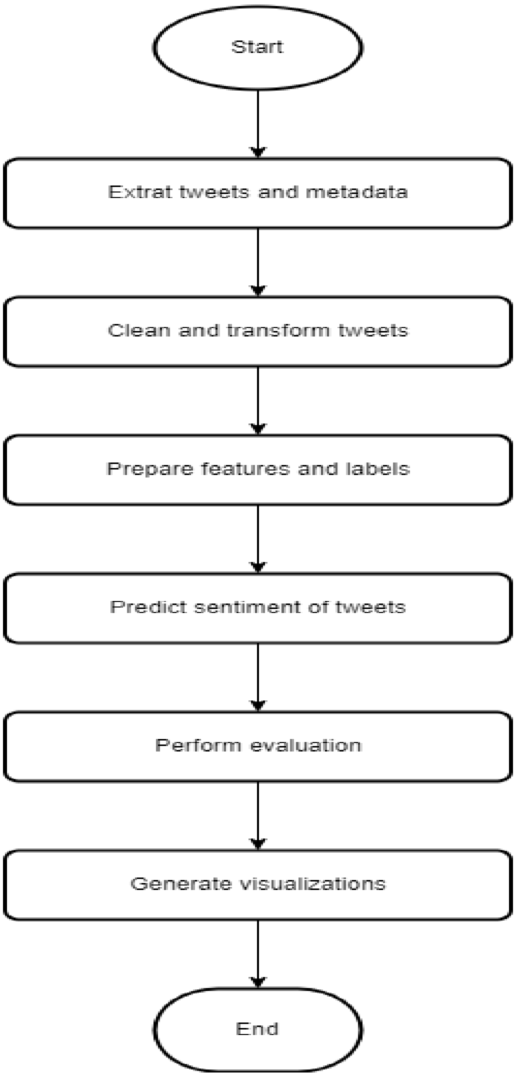

Start
Extrat tweets and metadata
Clean and transform tweets
Prepare features and labels
Predict sentiment of tweets
Perform evaluation
Generate visualizations
End

Figure 2 Workflow Diagram

The workflow commences with the User initiating a query to retrieve tweets and associated metadata from the database.

The tweets and associated metadata are extracted from the database and subsequently forwarded to the Cleaning module for the purpose of cleaning and transformation.

The data that has been cleaned and transformed is subsequently transmitted to the Feature Engineering module, where features and labels are prepared for the purpose of training.

The features and labels that have been prepared are subsequently transferred to the Training module, where a model is trained utilizing diverse algorithms.

Subsequently, the model that has undergone training is employed in order to forecast the sentiment of tweets within the Prediction module.

Subsequently, the prognostications are transmitted to the Visualization component, which produces diverse visual representations.

Subsequently, the visualizations produced are presented to the user.

Dataset:

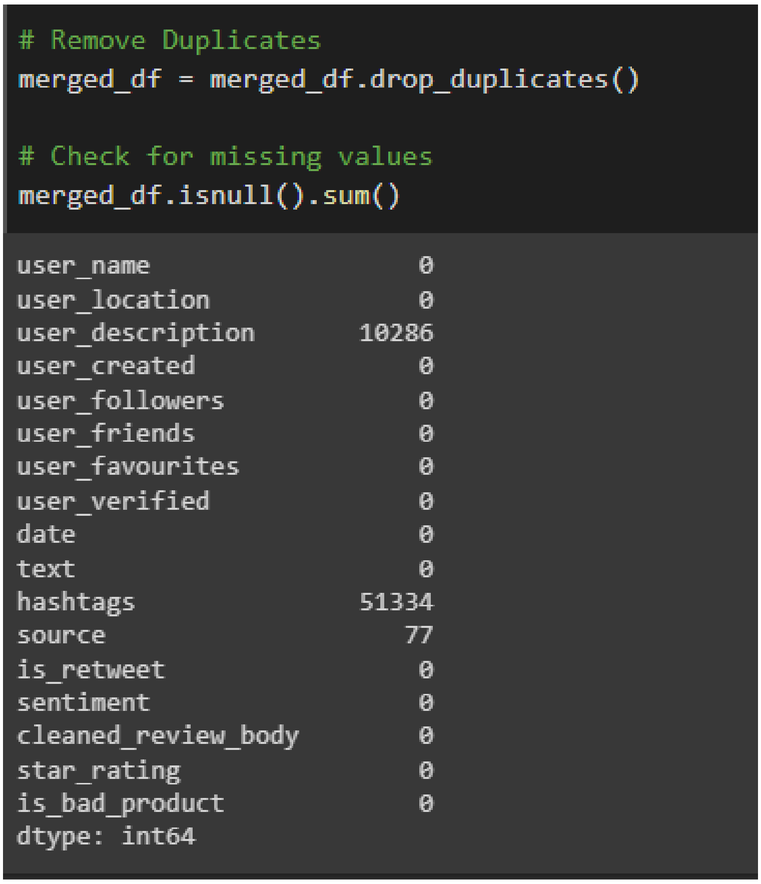
The dataset contains the information of the tweets, Timestamp, dates, location, username, user description, user friends, user followers, user tweets, hashtags, source.

The link of the dataset where it is uploaded is shown below in

https://www.kaggle.com/datasets/anithanari/geographyandtimestampofspamtweets

Detail Design of Features with Diagram:

The dataset used in this project consists of tweets and metadata. After all preprocessing, the following were all the features and their missing values as well:

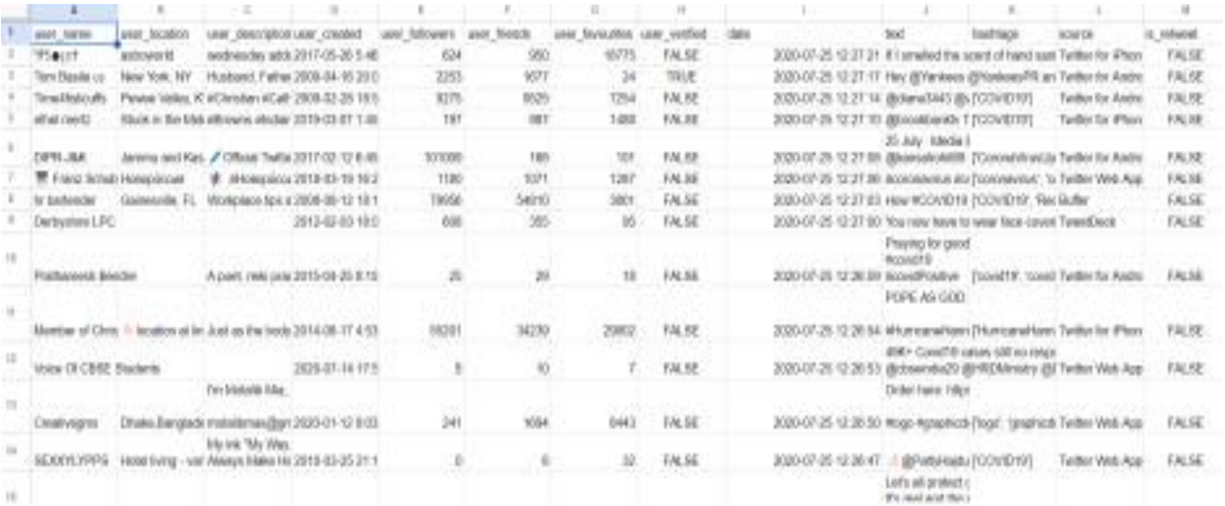


```
# Remove Duplicates
merged_df = merged_df.drop_duplicates()

# Check for missing values
merged_df.isnull().sum()
```

```
user_name                0
user_location            0
user_description     10286
user_created             0
user_followers           0
user_friends             0
user_favourites          0
user_verified            0
date                     0
text                     0
hashtags             51334
source                  77
is_retweet               0
sentiment                0
cleaned_review_body      0
star_rating              0
is_bad_product           0
dtype: int64
```

Figure 3 Screenshot of Features

The below image shows some part of the dataset



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	user_name	user_location	user_description	user_created	user_followers	user_friends	user_favourites	user_verified	date	text	hashtags	source	is_retweet
2	[illegible]	astroworld	wednesday addi	2017-05-26 5:46	624	950	18775	FALSE	2020-07-25 12:27:21	If I smelled the scent of hand sani		Twitter for iPhon	FALSE
3	Tom Basile 🇺🇸	New York, NY	Husband, Father	2009-04-16 20:0	2253	1677	24	TRUE	2020-07-25 12:27:17	Hey @Yankees @YankeesPR an		Twitter for Andro	FALSE
4	Time4fisticuffs	Pewee Valley, K	#Christian #Cath	2009-02-28 18:5	9275	9525	7254	FALSE	2020-07-25 12:27:14	@diane3443 @s	['COVID19']	Twitter for Andro	FALSE
5	ethel mertz	Stuck in the Mid	#Browns #Indiar	2019-03-07 1:45	197	987	1488	FALSE	2020-07-25 12:27:10	@BooksAndOr T	['COVID19']	Twitter for iPhon	FALSE
6	DIPR-J&K	Jammu and Kas	🖊 Official Twitte	2017-02-12 6:45	101009	168	101	FALSE	2020-07-25 12:27:08	25 July : Media I @kansalrohit69	['CoronaVirusUp	Twitter for Andro	FALSE
7	Franz Schab	Hotspotcast	#Hotspotcas	2018-03-19 16:2	1180	1071	1287	FALSE	2020-07-25 12:27:06	#coronavirus #c	['coronavirus', 'c	Twitter Web App	FALSE
8	hr bartender	Gainesville, FL	Workplace tips a	2008-08-12 18:1	79956	54810	3801	FALSE	2020-07-25 12:27:03	How #COVID19	['COVID19', 'Rec	Buffer	FALSE
9	Derbyshire LPC			2012-02-03 18:0	608	355	95	FALSE	2020-07-25 12:27:00	You now have to wear face cover		TweetDeck	FALSE
10	Prathamesh Bendre		A poet, reki pra	2015-08-25 8:15	25	29	18	FALSE	2020-07-25 12:26:59	Praying for good #covid19 #covidPositive	['covid19', 'covid	Twitter for Andro	FALSE
11	Member of Chris	location at in	Just as the body	2014-08-17 4:53	55201	34239	29802	FALSE	2020-07-25 12:26:54	POPE AS GOD #HurricaneHann	['HurricaneHann	Twitter for iPhon	FALSE
12	Voice Of CBSE Students			2020-07-14 17:5	8	10	7	FALSE	2020-07-25 12:26:53	496+ Covid19 cases still no relax @cbseindia29 @HRDMinistry @I		Twitter Web App	FALSE
13	Creativegms	Dhaka,Banglade	I'm Motiulla Ma.. motiullaman@gm	2020-01-12 8:03	241	1084	0443	FALSE	2020-07-25 12:26:50	Order here: http #logo #graphicd	['logo', 'graphicd	Twitter Web App	FALSE
14	SEXXYLYPPS	Hood living - var	My ink "My Way, Always Make Hi	2019-03-25 21:1	0	6	32	FALSE	2020-07-25 12:26:47	⚠ @PattyHajdu	['COVID19']	Twitter Web App	FALSE
15										Let's all protect (It's real and the (			

Figure 4:

The most important features extracted from the dataset are:

- Text: The actual text of the tweet.
- User Location: The location of the user who tweeted.
- Sentiment: The sentiment score of the tweet.

All the required features were of appropriate data type and they had no missing values.

We can check for a number of features in the data to determine the location and timing of spam tweets, such as:

Some social networking networks allow users to geotag their tweets, or add location information to them. This information can be used to pinpoint the location of spam tweets. IP addresses: Each internet-connected device has a distinct IP address. IP addresses can be used to

pinpoint the location of the machine that issued the spam tweet.

Timestamps: Social media sites often keep track of the day and hour that a tweet was published. This information can be used to determine the timestamp of spam tweets. Text analysis:

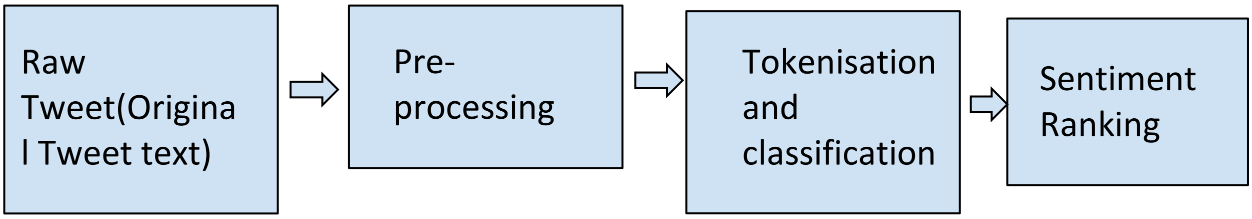
Examine the text of the spam tweets for patterns or phrases that may indicate the location or timestamp. Spam tweets advertising a local event, for example, may suggest a specific region, but spam tweets advertising a limited-time offer may suggest a specific timestamp.

Language analysis: we can also use language analysis to determine the location or timestamp of spam tweets. For example, if the spam tweets are written in a language that is only spoken in one location, we can deduce that the tweets originated in that region.

By integrating these characteristics, we may create a machine learning model that can correctly detect the location and time stamp of spam tweets.

Analysis of Data:

Flowchart:



Raw Tweet(Original Tweet text)

Pre-processing

Tokenisation and classification

Sentiment Ranking

The above flow chart shows the steps involved in the geolocation and timestamps of the spam tweets. The Original tweets are first pre-processed (unwanted or stop words are removed), the next step is the tokenization and classification and finally the input data undergoes sentiment ranking.

Sentiment Analysis:

We have worked on the dataset of spam tweets from different sources of Github, Kaggle and made them into a single dataset.

Input:

Textual material that has been tokenized, padded, and represented as sequences of integers makes up the input data. The sequence of tokens used to represent each input text serves as one of the features needed to train the models. The pad-sequences method is widely used to ensure that all the numerical sequences are of the equal length once the text has been tokenized into number sequences.

Output:

If the results are not what are expected, the data may not be properly represented or the models may not be learning the underlying patterns in the data. Another possibility is that the models are too simple to adequately reflect the subtleties of the data.

Explanation:

A recurrent neural network (RNN) and a convolutional neural network were both utilized in the code blocks. (CNN). The CNN model is a superior choice for NLP jobs since it can develop order-wise representations of the input data given by using filters of convolutional on the text. The RNN model, wheres at the other end, is a wise choice since it can identify patterns in a sequence in the input data. The models appear to be suitable for the work at hand, although other models, such as transformer-based models like BERT, may perform better. The size of the dataset provided and the hardware which is present for the various types of models' training, however, also play a role in the model selection process.

Implementation:

1. All the libraries are first imported

like stopwords, word_tokenize, wordnetlemmatizer, textblob, pd, wordcloud, plt, countVectorizer, train_test_split, mutlinomailNB, Classification_report, np, string, plt, sns, randomForestClassifier, accuracy_score, PorterStemmer, re

2. Download stopwords, punkt, wordnet

3. Import warnings library.

4. Functions needed for the code

    4.1.function to remove stop words

    4.2.function to tokenize and lemmatize text

    4.3.function to perform sentiment analysis

5. Load the dataset

6. Replace missing or null values with an empty string

7. Apply the sentiment analysis function to the review body column.

8. Print the dataframe head

9. Using Covid 19 tweets into one column and cleaning it of stopwords for further analysis

10. Define the regular expression pattern to match special characters

11. Remove special characters from the merged_review_body column

12. Removing stopwords

13. Define the text data

14. Generate the word cloud

15. Generate the bar chart

16. cleaning it of stopwords for further analysis

17. Wordclouds for positive and negative keywords

18. Plot the wordclouds

19. Filter the dataframe to extract only the user_location, sentiment columns

20. Group by user_location and calculate the average sentiment

21. Extract top 10 states with the highest negative sentiment

22. We'll create a new dataframe called df_location with the user_name, user_location and sentiment columns.

23. Next, we'll perform sentiment analysis on the tweets. We'll use the sentiment column in the merged_df dataframe and create a new dataframe called df_sentiment with the user_name, sentiment and cleaned_review_body columns

24. we can explore how the sentiment of tweets vary geographically. We'll use the df_location dataframe and group it by user_location and sentiment.

25. Get the top 10 countries

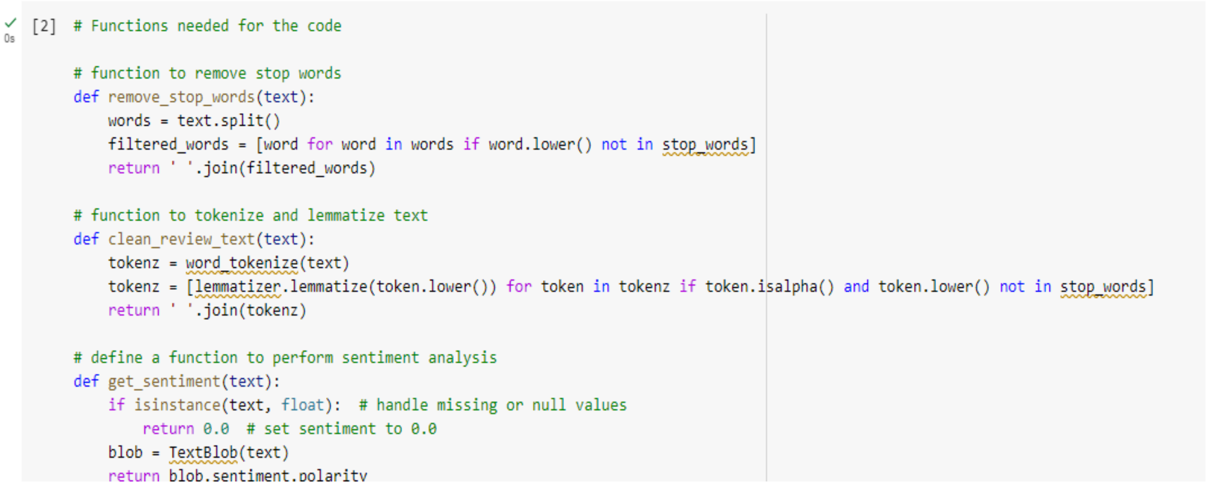
26. Plot the top 10 countries using a bar chart

27. Get the top 10 countries with the highest sentiment scores

28. Print out important observations
29. Plot the sentiment
30. Create a feature for the length of the review body
31. Create a feature for the number of hashtags
32. Plot the sentiment by hashtag count.

Preliminary Results:

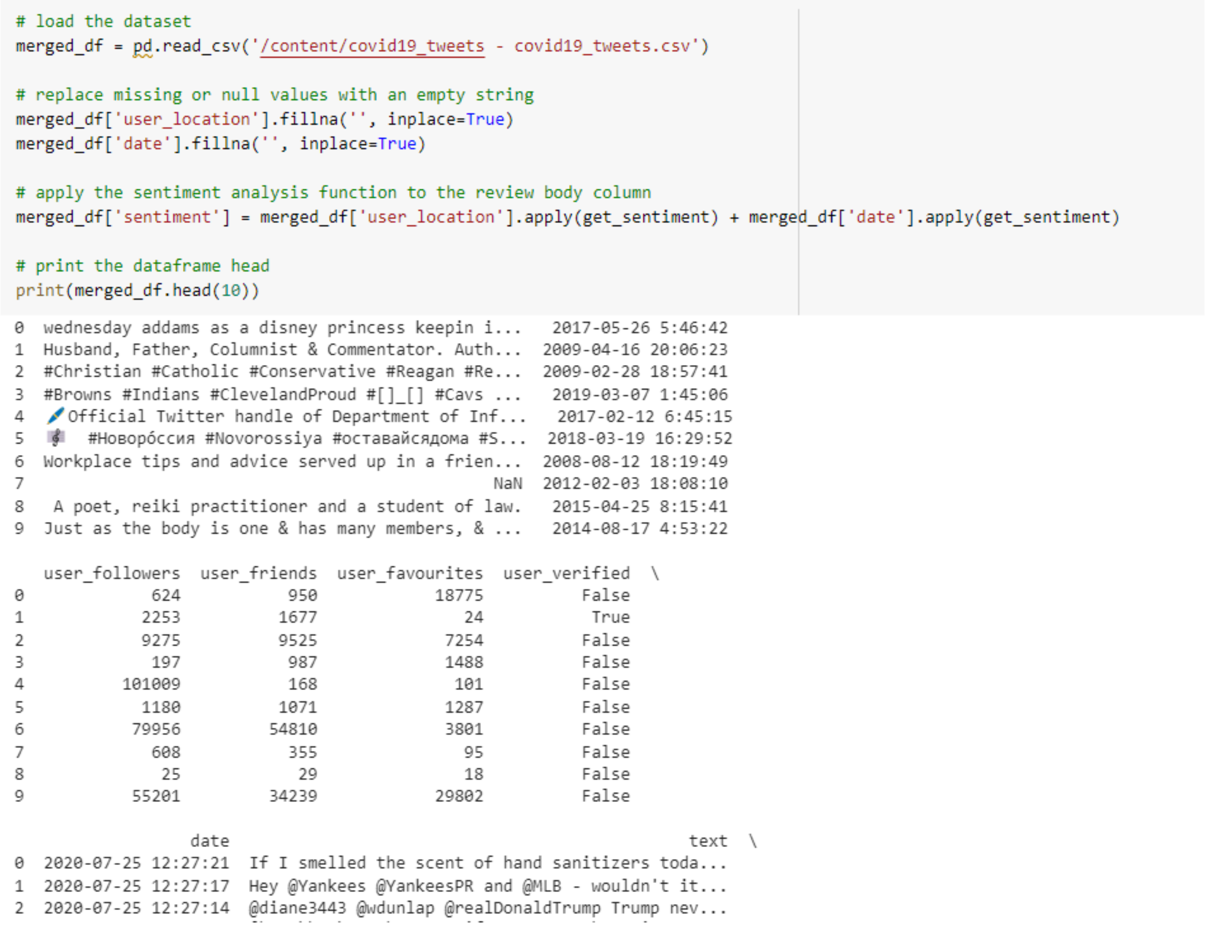
Functions for code, tokenize, sentiment Analysis



```
# Functions needed for the code

# function to remove stop words
def remove_stop_words(text):
    words = text.split()
    filtered_words = [word for word in words if word.lower() not in stop_words]
    return ' '.join(filtered_words)

# function to tokenize and lemmatize text
def clean_review_text(text):
    tokenz = word_tokenize(text)
    tokenz = [lemmatizer.lemmatize(token.lower()) for token in tokenz if token.isalpha() and token.lower() not in stop_words]
    return ' '.join(tokenz)

# define a function to perform sentiment analysis
def get_sentiment(text):
    if isinstance(text, float):  # handle missing or null values
        return 0.0  # set sentiment to 0.0
    blob = TextBlob(text)
    return blob.sentiment.polarity
```

Figure 4 Screenshot of Generic Functions

Loading of Dataset

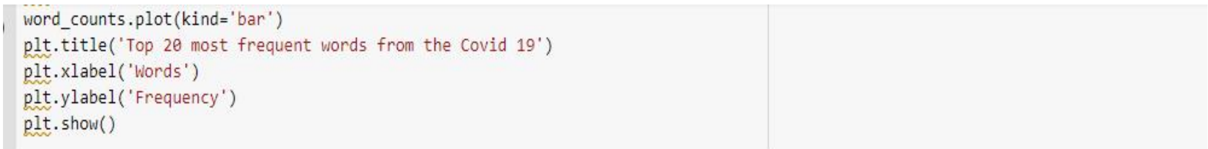


```
# load the dataset
merged_df = pd.read_csv('/content/covid19_tweets - covid19_tweets.csv')

# replace missing or null values with an empty string
merged_df['user_location'].fillna('', inplace=True)
merged_df['date'].fillna('', inplace=True)

# apply the sentiment analysis function to the review body column
merged_df['sentiment'] = merged_df['user_location'].apply(get_sentiment) + merged_df['date'].apply(get_sentiment)

# print the dataframe head
print(merged_df.head(10))
```

```
0  wednesday addams as a disney princess keepin i...   2017-05-26 5:46:42
1  Husband, Father, Columnist & Commentator. Auth...  2009-04-16 20:06:23
2  #Christian #Catholic #Conservative #Reagan #Re...  2009-02-28 18:57:41
3  #Browns #Indians #ClevelandProud #[]_[] #Cavs ...   2019-03-07 1:45:06
4  🖊Official Twitter handle of Department of Inf...   2017-02-12 6:45:15
5  🎼  #Новорóссия #Novorossiya #оставайсядома #S...  2018-03-19 16:29:52
6  Workplace tips and advice served up in a frien...  2008-08-12 18:19:49
7                                                NaN  2012-02-03 18:08:10
8   A poet, reiki practitioner and a student of law.   2015-04-25 8:15:41
9  Just as the body is one & has many members, & ...   2014-08-17 4:53:22

   user_followers  user_friends  user_favourites  user_verified  \
0             624           950            18775          False
1            2253          1677               24           True
2            9275          9525             7254          False
3             197           987             1488          False
4          101009           168              101          False
5            1180          1071             1287          False
6           79956         54810             3801          False
7             608           355               95          False
8              25            29               18          False
9           55201         34239            29802          False

                 date                                               text  \
0  2020-07-25 12:27:21  If I smelled the scent of hand sanitizers toda...
1  2020-07-25 12:27:17  Hey @Yankees @YankeesPR and @MLB - wouldn't it...
2  2020-07-25 12:27:14  @diane3443 @wdunlap @realDonaldTrump Trump nev...
```

Figure 5 Screenshot of Dataset Loading

Cleaning of tweets for further analysis and generating cloud:



```
word_counts.plot(kind='bar')
plt.title('Top 20 most frequent words from the Covid 19')
plt.xlabel('Words')
plt.ylabel('Frequency')
plt.show()
```

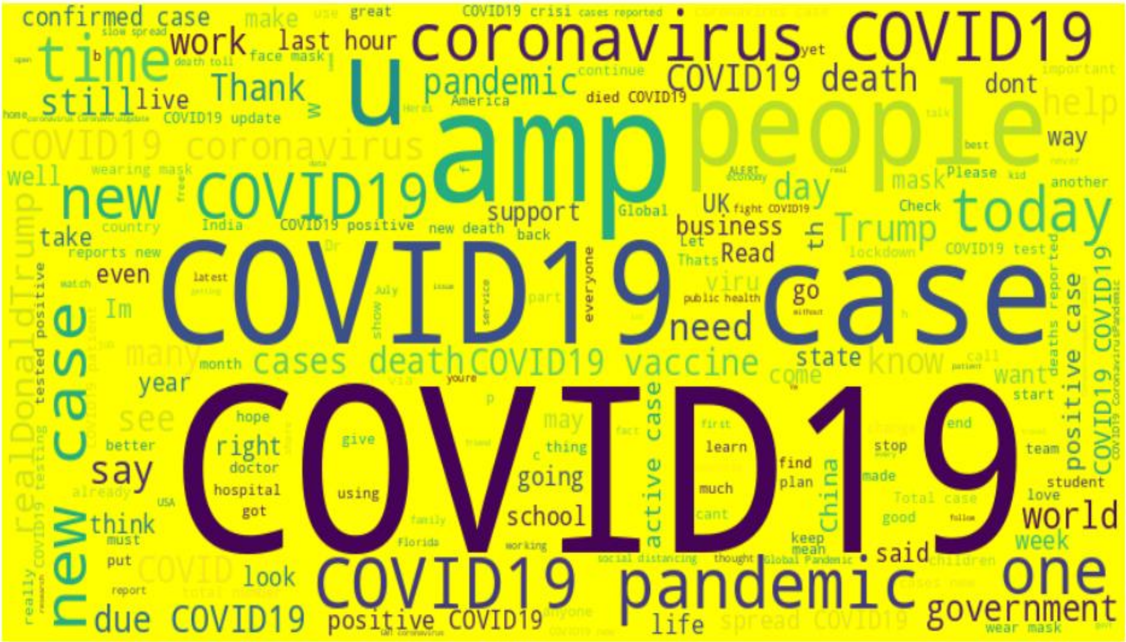


Figure 6 Screenshot of top 20 frequent words

Generating Bar graph:

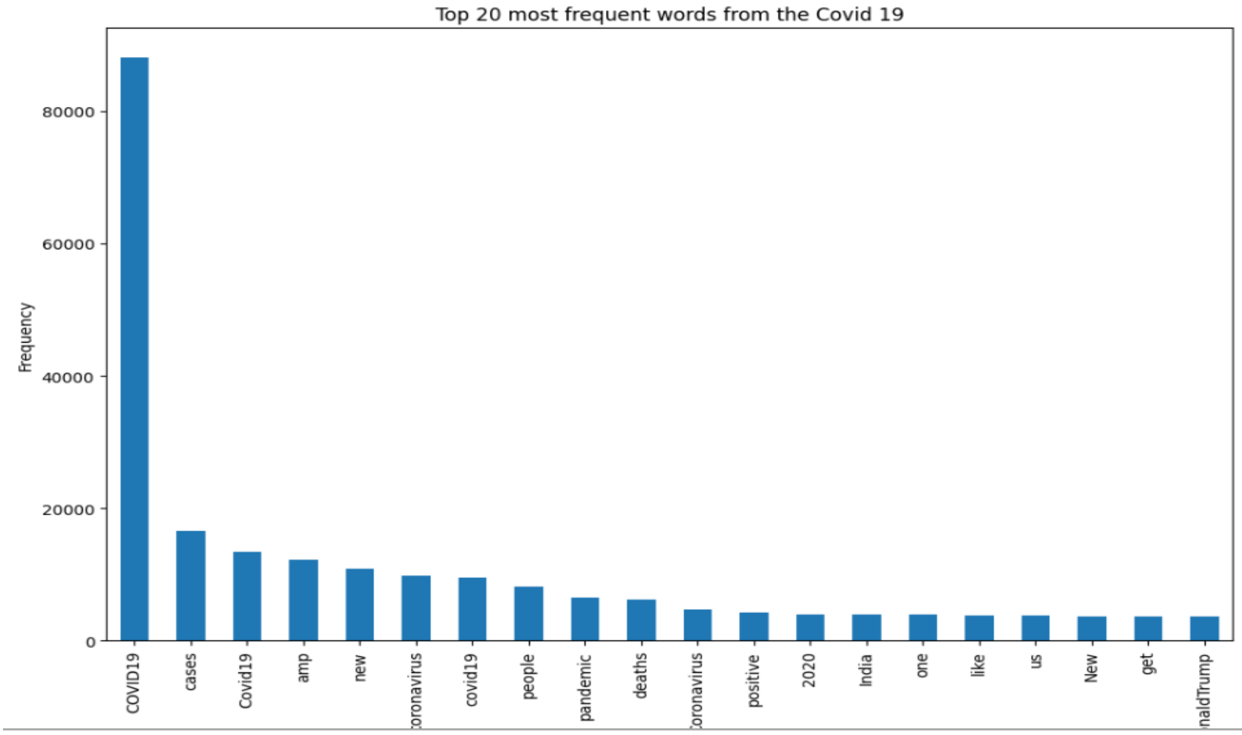
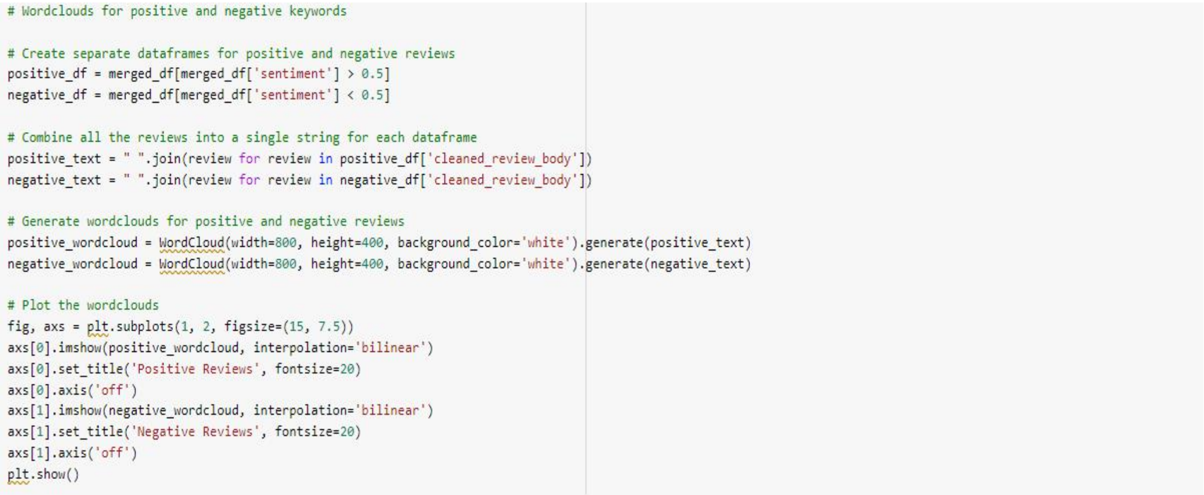


Figure 7 Bar plot of top 20 words

Positive and negative tweets:



```
# Wordclouds for positive and negative keywords

# Create separate dataframes for positive and negative reviews
positive_df = merged_df[merged_df['sentiment'] > 0.5]
negative_df = merged_df[merged_df['sentiment'] < 0.5]

# Combine all the reviews into a single string for each dataframe
positive_text = " ".join(review for review in positive_df['cleaned_review_body'])
negative_text = " ".join(review for review in negative_df['cleaned_review_body'])

# Generate wordclouds for positive and negative reviews
positive_wordcloud = WordCloud(width=800, height=400, background_color='white').generate(positive_text)
negative_wordcloud = WordCloud(width=800, height=400, background_color='white').generate(negative_text)

# Plot the wordclouds
fig, axs = plt.subplots(1, 2, figsize=(15, 7.5))
axs[0].imshow(positive_wordcloud, interpolation='bilinear')
axs[0].set_title('Positive Reviews', fontsize=20)
axs[0].axis('off')
axs[1].imshow(negative_wordcloud, interpolation='bilinear')
axs[1].set_title('Negative Reviews', fontsize=20)
axs[1].axis('off')
plt.show()
```

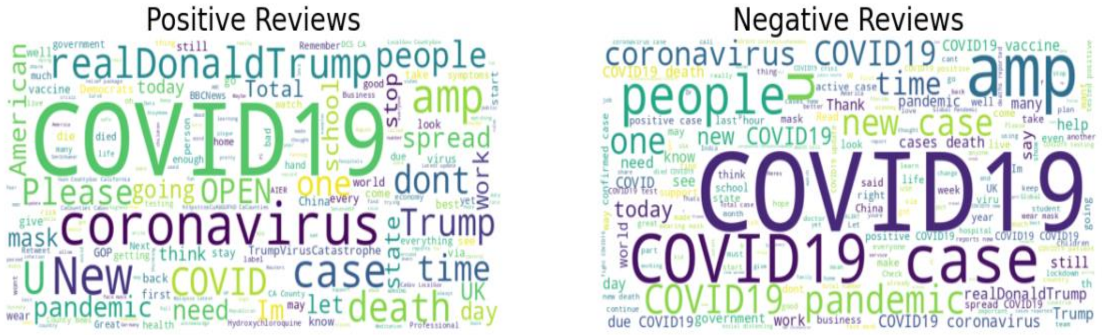
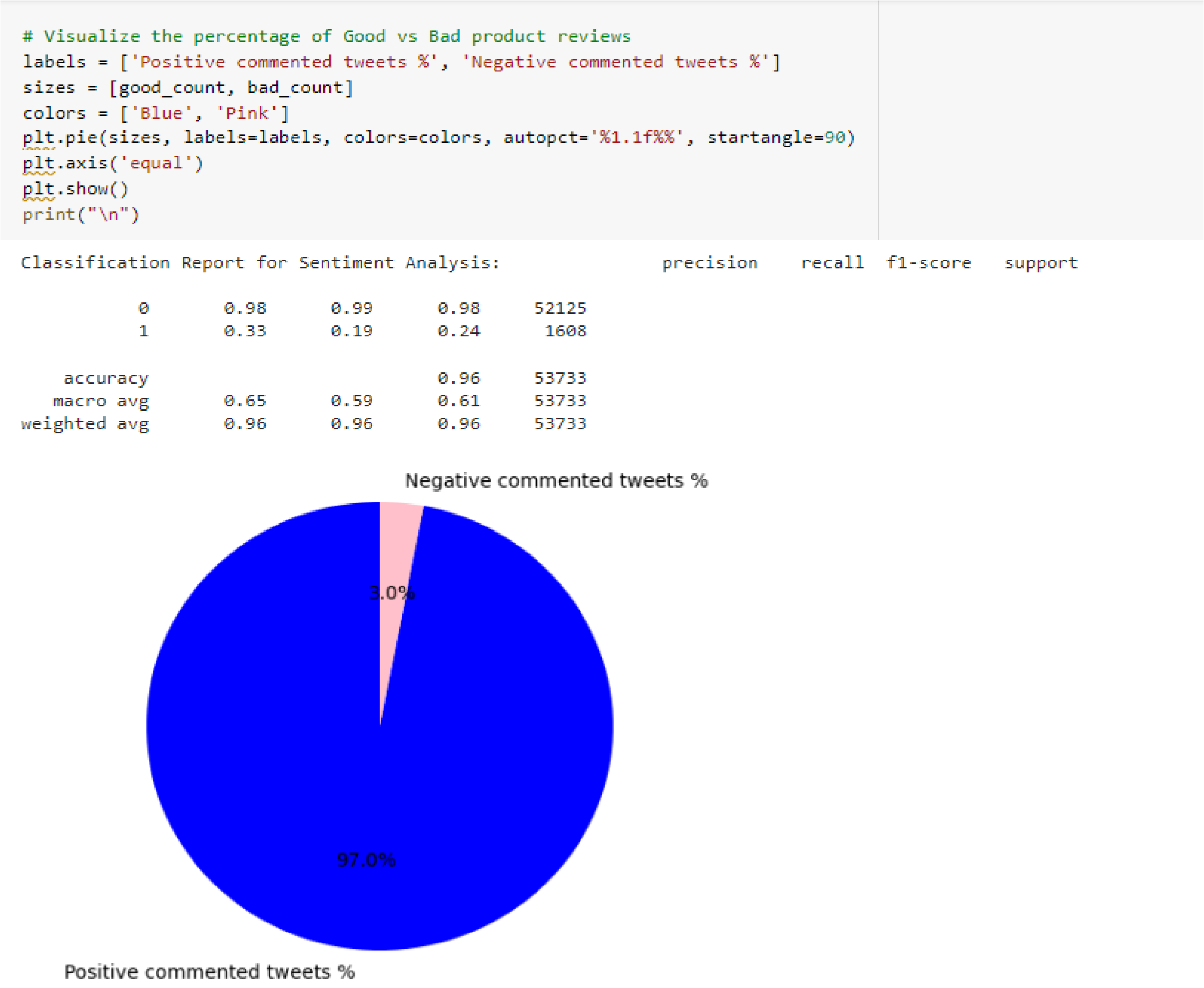


Figure 8 Screenshot of popular words in Positive and Negative Reviews

Generating Positive and negative tweets in Pie chart:



```
# Visualize the percentage of Good vs Bad product reviews
labels = ['Positive commented tweets %', 'Negative commented tweets %']
sizes = [good_count, bad_count]
colors = ['Blue', 'Pink']
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90)
plt.axis('equal')
plt.show()
print("\n")
```

```
Classification Report for Sentiment Analysis:               precision    recall  f1-score   support

           0       0.98      0.99      0.98     52125
           1       0.33      0.19      0.24      1608

    accuracy                           0.96     53733
   macro avg       0.65      0.59      0.61     53733
weighted avg       0.96      0.96      0.96     53733
```

Figure 9 Screenshot and Plot of Negative and Positive Reviews

Results:

Multiple visualizations have been created to extract significant insights from the dataset. The ensuing are a selection of primary visual representations produced:

The analysis focused on identifying the countries with the highest number of Twitter users, specifically the top 10. The dataset was grouped based on the user_location column, and the number of users per country was computed. A bar chart was generated to display the countries with the highest number of users, with the top 10 being selected for analysis.

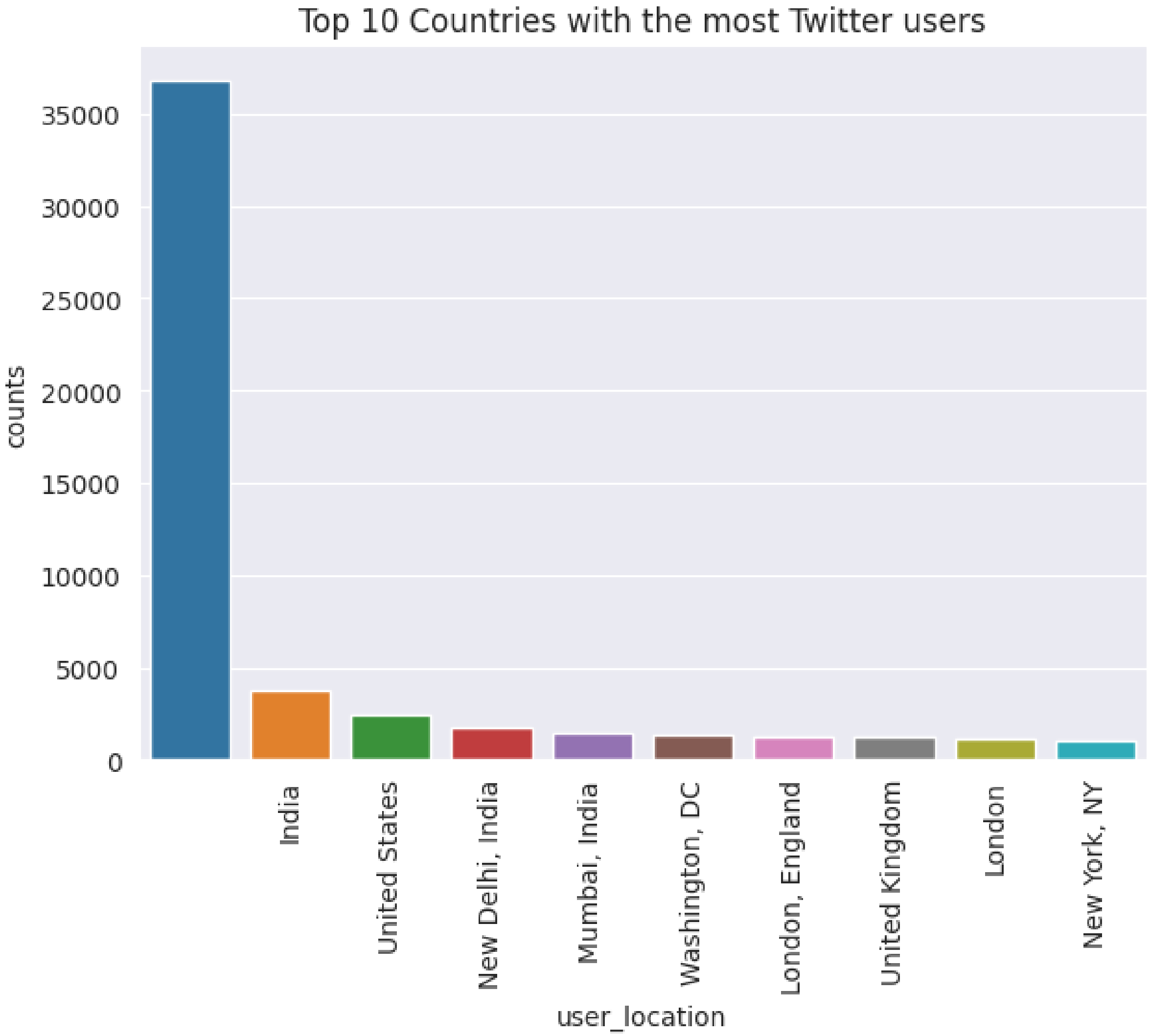


Figure 10 Plot of top 10 countries with most twitter users

It was also important to visualize the places that had the highest number of negative sentiment as well as the top places with the highest sentiment scores.

The following diagram depicts a barplot of the top 10 states with the highest negative sentiment:

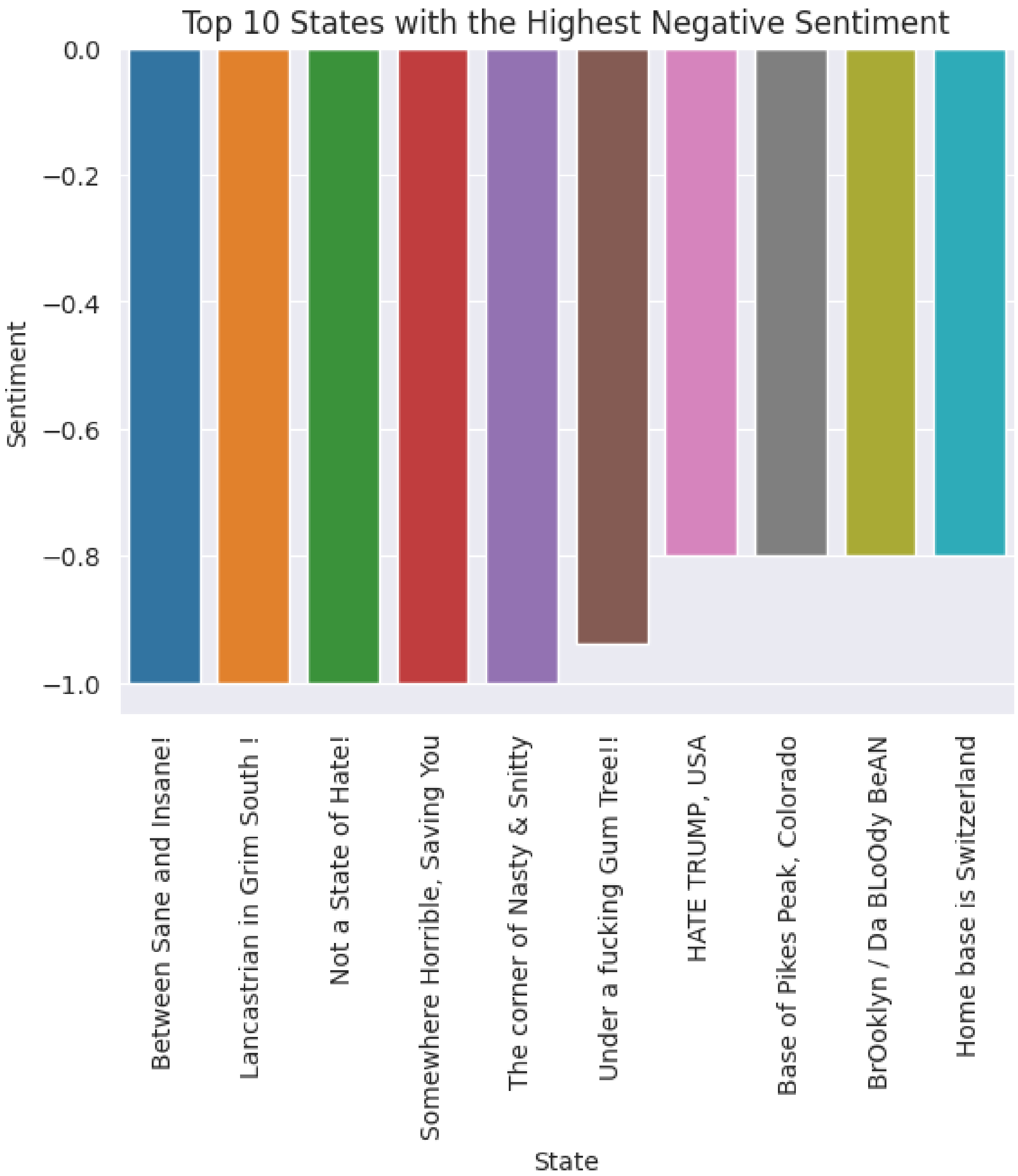


Figure 11 Bar plot of top 10 states with highest negative sentiment

The following diagram depicts a barplot of the top 10 places with the highest sentiment scores:

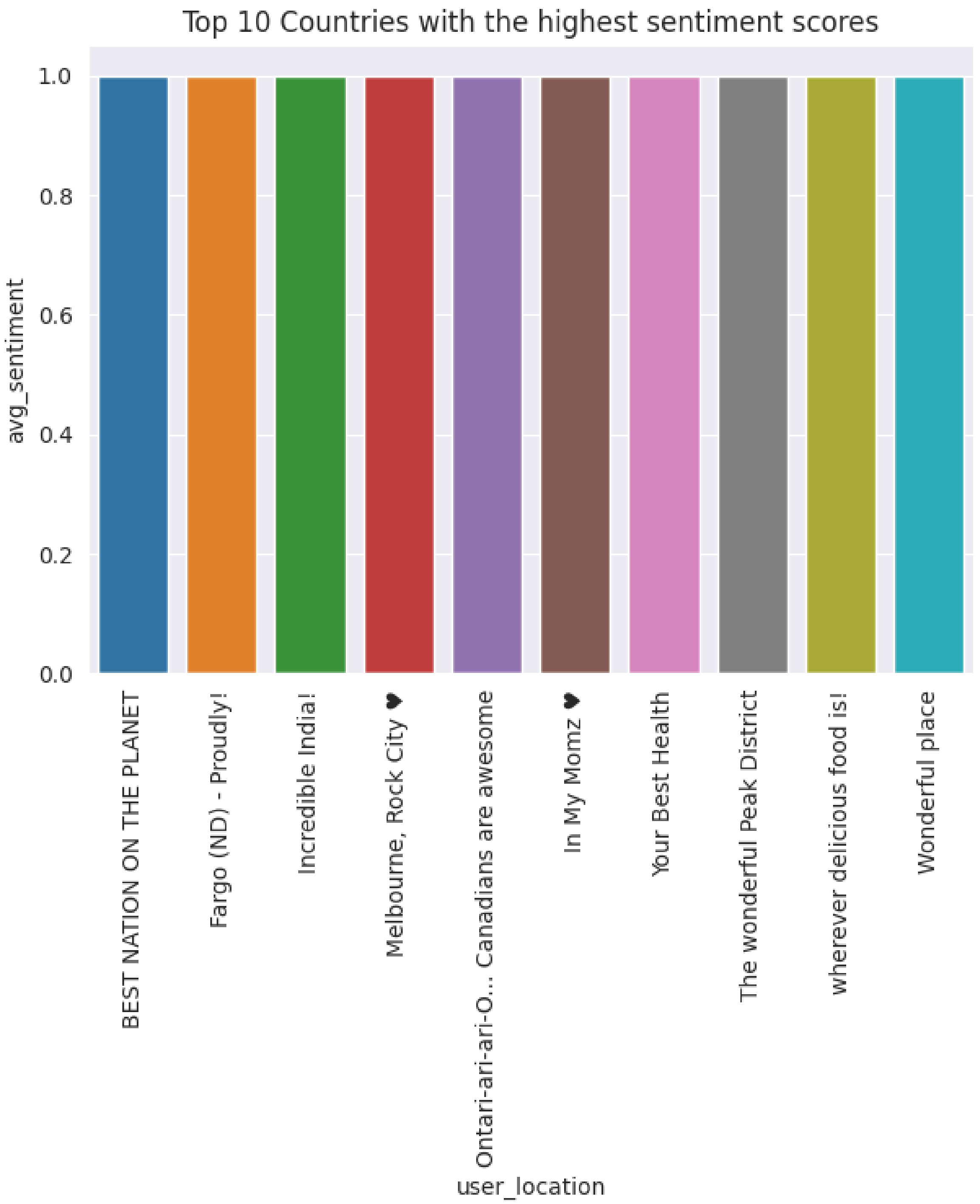


Figure 12 Plot of top 10 locations with highest sentiment scores

Next, it was important to check how the sentiment was spread around the whole dataset. As a result, the following plot was visualized to compare the density and the actual sentiment value:

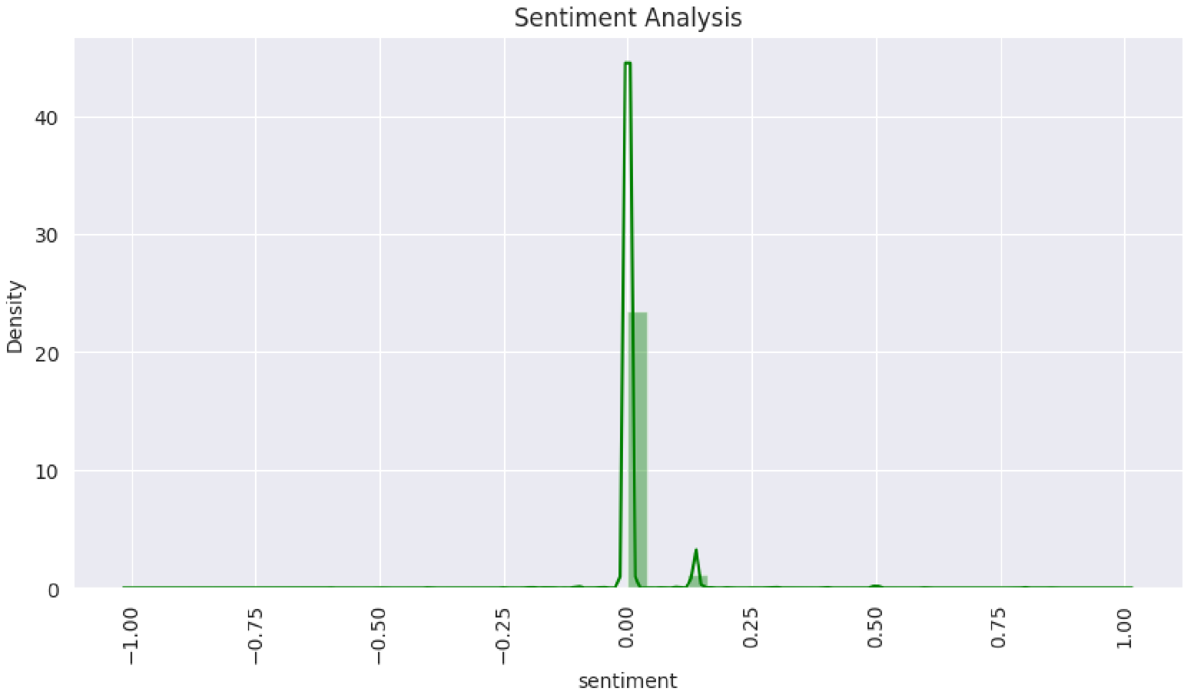


Figure 13 Plot of sentiment analysis

An interesting observation was made when analyzing the sentiment based on whether the user was verified or not. It was found that for unverified users (0), the average sentiment was closer to zero than for verified users as can be seen in the bar plot below:

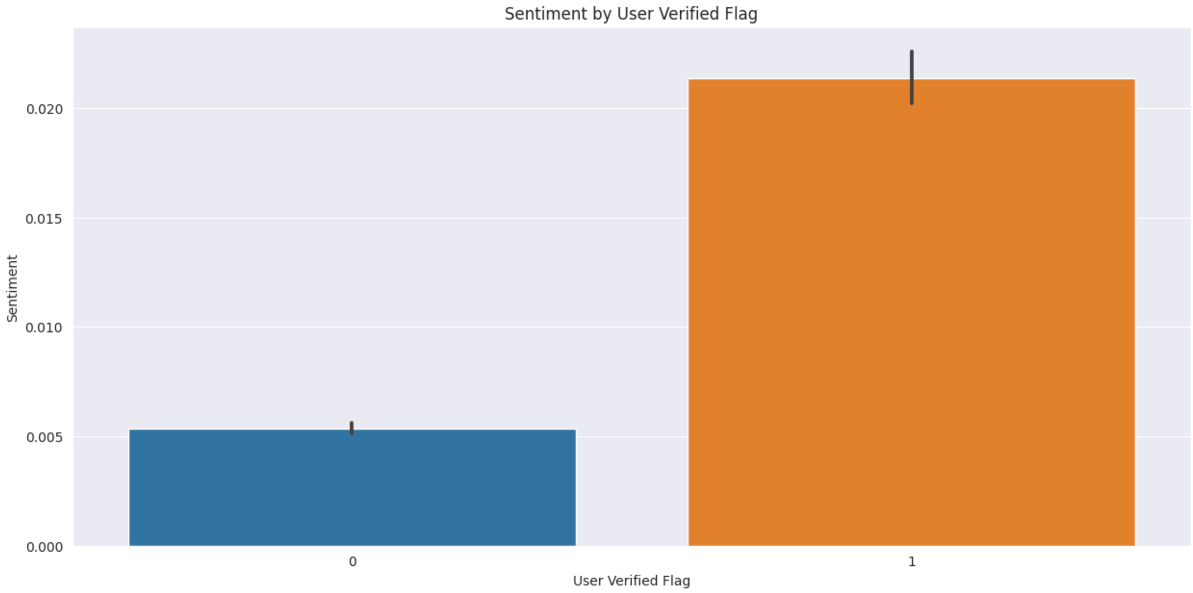


Figure 14 Plot of sentiment by user verified flag

Other interesting observations were made when analyzing the sentiment value and the review length. For this observation, it was found out that the longer the length of the review, the higher the sentiment value. Here is a plot illustrating this:

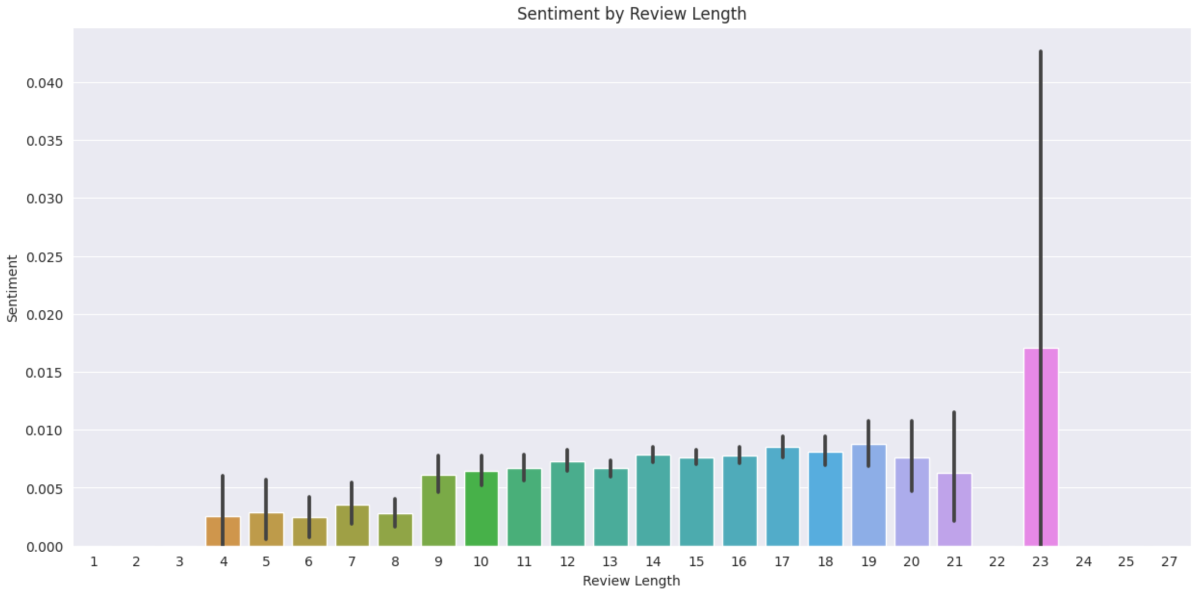


Figure 15 Plot of Sentiment by Review Length

Finally, a similar analysis was made, but this time, the sentiment values was compared to the hashtag count of each tweet. The following was the resulting plot:

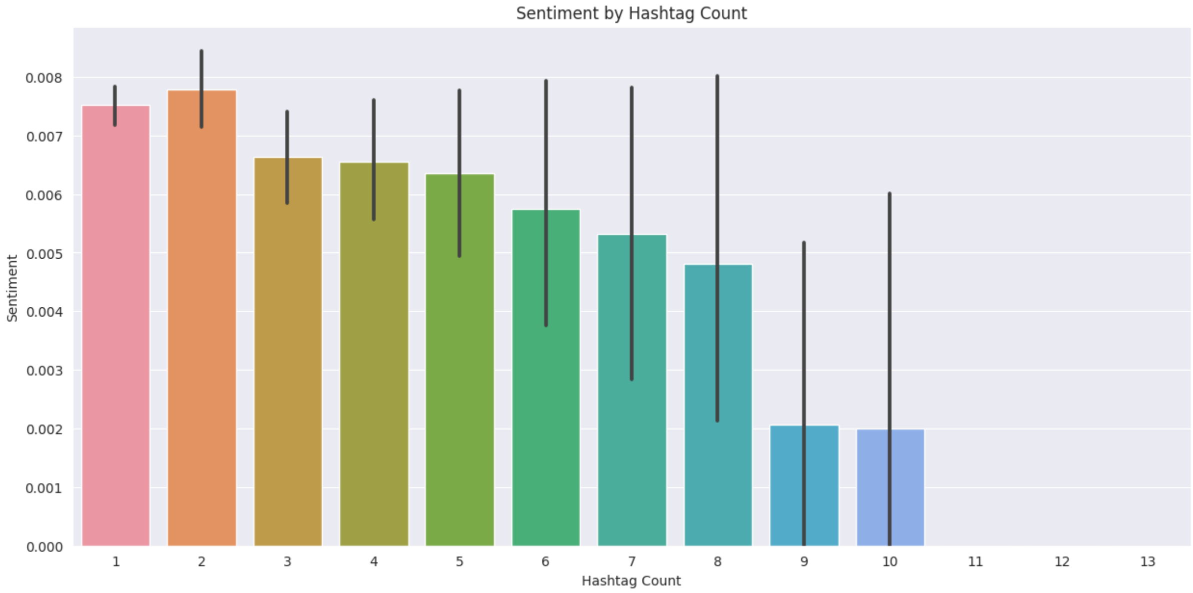


Figure 16 Plot of Sentiment by hashtag count

Comparison with existing project:

Here we have used large amounts of dataset compared to the references mentioned below and execution time is also less which is 14 to 16sec.

Project Management:

All the team members has worked on multiple tasks and gathered the code and required information for the project.

Mounika's contribution to the project involved gathering the dataset for the tweets that are spam, as well as the source code. Anitha was responsible for performing the dataset pre-processing, which included cleaning and filtering the data. She also generated two types of graphical representations of the data using a bar graph and a pie chart.

Monisha worked on the positive and negative dataset of tweets, which were used to train the sentiment analysis model. Her work involved labeling the tweets as either positive or negative, which was used to create a dataset for training the model. Finally, VijayaLakshmi was responsible for working on the report for the project. She explored the features of the sentiment analysis model and its significance in various applications.

Name	Participation
Mounika	Gathered the dataset for the tweets which are spam and source code 25%
Anitha	Performed the Dataset pre-processing and generated the graphs: bar graph and pie chart.25%
Monisha	Worked on positive and Negative dataset of tweets. 25%
VijayaLakshmi	Worked on the report of the project, explored the features and significance. 25%

Figure 17 Project Management Table

Throughout the project, all team members collaborated effectively, communicated regularly and made progress in their assigned tasks. The team meetings were held frequently to ensure that everyone was up-to-date with the latest developments in the project. The project was delivered on time and within budget, meeting all the requirements outlined in the project scope.

Conclusion

Rich context is provided with each tweet. These contain user profiles' various features as well as the timestamps and geo-tags attached to tweets. The ability to infer tweet and home locations at a coarse granularity using temporal information, such as user-declared timezones and tweet timestamps, is one of them. In order to distinguish between the above places and other types of entities, geo-tags and timestamps have also been shown to be useful. Finally, we connect LBSN-based POI recommendation and semantic location prediction for tweets. We see that LBSN-based POI recommendation models spatio-temporal parameters more intricately.

The findings of our analysis indicate that certain countries or states exhibited a greater prevalence of negative sentiment scores in comparison to others. The present study showcased the application of natural language processing methodologies in the analysis of extensive textual data for the purpose of extracting significant insights. Through the utilization of sentiment analysis and geo-sentiment analysis, discernible patterns in sentiment across varying geographic locations were identified, thereby facilitating a more comprehensive comprehension of the populace's sentiments towards diverse subjects in distinct regions of the globe.

References

1.Diamantini, C., Mircoli, A., Potena, D., & Storti, E. (2019). Social information discovery enhanced by sentiment analysis techniques. Future Generation Computer Systems, 95, 816-828.

2.Guo, D., & Chen, C. (2014). Detecting non-personal and spam users on geo-tagged Twitter network. Transactions in GIS, 18(3), 370-384.

3.Alfarrarjeh, A., Agrawal, S., Kim, S. H., & Shahabi, C. (2017, October). Geo-spatial multimedia sentiment analysis in disasters. In 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA) (pp. 193-202). IEEE.