

Mini Project Report

On

Twitter Data Trend Analysis using Python

Submitted by

Name:- 1. Shouryadeep Manna
2. Sagnik Kumar Das

Roll No.: — 1. 2206395
2. 2206117

School of Computer Engineering
KIIT - DU



KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY (KIIT)

Deemed to be University U/S 3 of UGC Act, 1956

Table of Contents

1. Introduction	3
2. Problem Statement	5
3. Python Package Used Details	6
4. Source Code.....	7
5. Implementation Results.....	16
6. Conclusion	22
7. References	23

Chapter 1

Introduction

1.1 Overview

Twitter, now known as X, is one of the most widely used social media platforms, where users share thoughts, opinions, and real-time updates on global events. With millions of tweets generated daily, analyzing Twitter trends provides valuable insights into public sentiment, trending topics, and user engagement. Businesses, researchers, and policymakers utilize Twitter trend analysis to understand public opinion, monitor brand reputation, detect misinformation, and analyze event-driven discussions.

In this project, we conducted a **Twitter trend analysis** using Python, focusing on a random trending topic **#ShaheedDiwas**, which was observed to be one of the top trends in India. The project involved **fetching live tweets**, processing the data, analyzing tweet sentiments, extracting hashtags, and visualizing user engagement levels.

1.2 Objective of the Project

The main objectives of this project include:

- A) Trend Identification: Fetch real-time trending tweets on a specific topic.
- B) Sentiment Analysis: Categorize tweets as positive, negative, or neutral.
- C) User Engagement Analysis: Analyze retweets, likes, and replies.
- D) Hashtag Frequency Analysis: Identify the most frequently used hashtags.
- E) Tweet Activity Monitoring: Study tweet distribution over time to determine peak activity hours.

1.3 Tools & Technologies Used

To accomplish these objectives, we used the following tools and technologies:

- Programming Language: Python.
- Twitter API: Tweepy, for fetching real-time tweets.
- Natural Language Processing (NLP): TextBlob, for sentiment analysis.
- Data Visualization: Matplotlib & Seaborn, for graphs and charts.
- Data Processing: Pandas, for managing and analyzing tweet data.
- Word Cloud: WordCloud library, for most used words.

1.4 Work Distribution Among Team Members

Since this project was carried out by two members, the tasks were divided as follows:

a) Member 1(Shouryadeep Manna):

- i. Set up Twitter API authentication and fetched real-time tweets using Tweepy.
- ii. Processed the data, including removing stopwords, cleaning text, and handling missing values

b) Member 2(Sagnik Kumar Das):

- i. Conducted hashtag analysis
- ii. Created visualizations, including scatter plots, bar charts, word clouds and sentiment analysis on tweets using TextBlob.

Chapter 2

Problem Statement & Objectives

6.1 Problem Statement

Twitter is a key platform for tracking real-time events, public sentiment, and viral trends. However, analyzing Twitter trends effectively presents challenges such as efficient data extraction, cleaning unstructured text, and understanding engagement levels. This project aims to overcome these challenges by developing a structured approach to extract, analyze, and visualize Twitter trends.

6.2 Objectives

- Extract trending tweets on random trending hashtags using the Twitter API.
- Analyze hashtag frequency and sentiment trends in the collected data
- Measure user engagement by tracking retweets, likes, and replies.
- Visualize insights using charts, word clouds, and sentiment distribution graphs.

This approach will help businesses, researchers, and analysts gain data-driven insights from Twitter trends.

Chapter 3

Python Packages Used Details

Package Name	Function Used	Explanation
Tweepy	Client.search_recent_tweets()	Fetches recent tweets based on a query.
Pandas	DataFrame(), to_csv()	Organizes tweet data and saves to CSV.
Matplotlib	plot(), bar()	Used for creating various charts.
Seaborn	sns.barplot()	Enhances visualization aesthetics.
Wordcloud	WordCloud().generate()	Generates word clouds from tweets.
TextBlob	TextBlob().sentiment.polarity	Performs sentiment analysis on tweets.

Chapter 4

Source Code

1. Authentication and API Connection.

```
import tweepy

# Replace with your Twitter API keys

API_KEY = "Your API Key"

API_SECRET = "Your API Secret Key"

ACCESS_TOKEN = "Your Access Token"

ACCESS_SECRET = "Your Access Secret "

BEARER_TOKEN = "Your Bearer Token"

# Authenticate with OAuth1

auth = tweepy.OAuth1UserHandler(API_KEY, API_SECRET, ACCESS_TOKEN, ACCESS_SECRET)

api = tweepy.API(auth, wait_on_rate_limit=True)

# Verify authentication

try:

    user = api.verify_credentials()

    print(f"Authenticated as {user.screen_name}")

except tweepy.TweepyException as e:

    print(f"Error: {e}")

#Alternative Authentication

auth = tweepy.OAuth1UserHandler(API_KEY, API_SECRET, ACCESS_TOKEN, ACCESS_SECRET)

api = tweepy.API(auth)

try:

    api.verify_credentials()

    print("API v1.1 Access Verified: Elevated Access Enabled!")

except tweepy.TweepyException as e:

    print(f"Error: {e}")
```

2. Retrieving Tweets

```
import time

client = tweepy.Client(BEARER_TOKEN)

query = "#ShaheedDiwas is:retweet lang:en"

while True:

    try:

        tweets = client.search_recent_tweets(query=query, max_results=10, tweet_fields=["created_at",
"text"])

        if tweets.data:

            for tweet in tweets.data:

                print(tweet.created_at, "-", tweet.text)

        else:

            print("No tweets found.")

            break # Exit loop if successful

    except tweepy.TooManyRequests:

        print("Rate limit exceeded. Waiting for 15 minutes...")

        time.sleep(900) # Wait 15 minutes before retrying

    except tweepy.TweepyException as e:

        print(f"Error: {e}")

        break
```


3. Example Tweets Data

2025-03-24 12:45:31+00:00 On #ShaheedDiwas, SFI SPPU paid tribute to Bhagat Singh, Sukhdev & Rajguru with a Mashaal rally, cultural performances & readings of Bhagat Singh's writings. Let's keep their revolutionary spirit alive! #Shaheed #BhagatSingh #InquilabZindabad <https://t.co/qzhXNMdWrj> [cite: 7, 8, 9] 2025-03-24 12:22:46+00:00 Joined the Shaheed Diwas programme organised by Bhagat Singh Yuvak Sangha at Hotel Royal Kalinga. Paid tribute to the immortal martyrs Bhagat Singh, Rajguru, and Sukhdev, honouring their supreme sacrifice for India's freedom. #ShaheedDiwas #TributeToMartyrs <https://t.co/K6yQDwe106> [cite: 9, 10, 11] 2025-03-24 12:16:00+00:00 23 Mar #ShaheedDiwas Martyr's Day Brave hearts #BhagatSingh, #SukhdevThapar & #ShivaramRajguru sacrifices their lives for country real patriotism, inspires generations. <https://t.co/GzmJJpwi6u> #JaiHind [cite: 11, 12] 2025-03-24 12:12:18+00:00 On the night of March 23, 1931, Bhagat Singh, Rajguru, and Sukhdev walked toward the gallows-not with fear, but with smiles and unwavering voices chanting: "Inquilab Zindabad!" #MartyrsDay #ShaheedDiwas #NeverForget #HPCLSalutes @HPCL @hpcl_retail @ncz_hpcl <https://t.co/AmCCdYAOHC> [cite: 12, 13] 2025-03-24 12:06:12+00:00 On #ShaheedDiwas, #DYFI & #SFI organized an art competition in Uran (Raigad) to honor Shaheed Bhagat Singh, Rajguru & Sukhdev. Enthusiastic participation from students, celebrating revolutionary ideals! <https://t.co/TIHOCF2rRW> [cite: 13, 14, 15] 2025-03-24 11:39:17+00:00 @bjd_odisha Saluting the immortal spirits of Bhagat Singh, Sukhdev, and Rajguru on #ShaheedDiwas! Their courage and sacrifice continue to inspire generations to serve the nation with pride. [cite: 15, 16] 2025-03-24 11:31:26+00:00 Remembering the Heroes on Shaheed Diwas #ShaheedDiwas #BhagatSingh #India <https://t.co/Q07Vui7Cx4> [cite: 16, 17] 2025-03-24 10:44:11+00:00 Is India INDEPENDENT in 2025? What is Freedom for us? Is exposing body, a sign of confidence and independence! #freedom #ShaheedDiwas #ShaheedBhagatSingh #India <https://t.co/P4kpcgiQsG> [cite: 17, 18, 19] 2025-03-24 09:36:04+00:00 Honoring 200 Years of Valor! The Bicentenary of 5RAJPUT was commemorated with a solemn #WreathLaying ceremony at the National War Memorial, paying tribute to the #Bravehearts who made the ultimate sacrifice. #IndianArmy #RememberTheBrave #SaluteToHeroes #ShaheedDiwas #JaiHind <https://t.co/6mFTxIJ4uN> [cite: 19, 20] 2025-03-24 09:28:11+00:00 7 Future Prospects 7 Govt initiatives like Atmanirbhar Bharat & DAP encourage private investment Next-gen fighter projects (AMCA) could see greater private sector involvement Defence exports offer new market opportunities #ShaheedDiwas <https://t.co/5xtqGPNvNW> [cite: 20, 21]

4. Data Processing and Analysis

```
import pandas as pd

# Convert to DataFrame

if tweets.data:

    tweet_data = [{"created_at": tweet.created_at, "text": tweet.text, "author_id": tweet.author_id} for
tweet in tweets.data]

    df = pd.DataFrame(tweet_data)

# Save to CSV for later use

df.to_csv("tweets.csv", index=False)

print(df.head()) # Show first 5 rows

else:

    print("No tweets found.")
```

5. Data Inspection

```
# Check for null values

print(df.isnull().sum())

# Count total tweets

print("Total Tweets Collected:", len(df))

# Count unique authors

print("Unique Authors:", df["author_id"].nunique())

# Most common words (excluding stopwords)

from collections import Counter

import re

# Tokenize tweets

words = [word.lower() for tweet in df["text"] for word in re.findall(r"\w+", tweet) if len(word) > 3]

common_words = Counter(words).most_common(10)

print("Most Common Words:", common_words)
```

6. Tweet Frequency Analysis.

```
import matplotlib.pyplot as plt

# Convert 'created_at' column to datetime format
df["created_at"] = pd.to_datetime(df["created_at"])

# Count tweets per minute
df.set_index("created_at", inplace=True)
df_resampled = df.resample("1T").count()

# Plot tweet frequency
plt.figure(figsize=(10, 5))

plt.plot(df_resampled.index, df_resampled["text"], marker='o', linestyle='-', color='b', label="Tweet Frequency")

plt.xlabel("Time")
plt.ylabel("Number of Tweets")
plt.title("Tweet Frequency Over Time")
plt.legend()
plt.xticks(rotation=45)
plt.grid()
plt.show()
```

7. Word Cloud Generation

```
from wordcloud import WordCloud

# Generate word cloud

wordcloud = WordCloud(width=800, height=400, background_color="white").generate("
".join(df["text"]))

# Display word cloud

plt.figure(figsize=(10, 5))

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

plt.axis("off")

plt.title("Most Used Words in Tweets")

plt.show()
```

8. Sentiment Analysis

```
from textblob import TextBlob

# Function to classify sentiment

def get_sentiment(text):

    score = TextBlob(text).sentiment.polarity

    if score > 0:

        return "Positive"

    elif score < 0:

        return "Negative"

    else:

        return "Neutral"

# Apply sentiment function

df["sentiment"] = df["text"].apply(get_sentiment)

# Count sentiment distribution

df["sentiment"].value_counts().plot(kind="bar", color=["green", "gray", "red"], figsize=(6, 4))

plt.xlabel("Sentiment")

plt.ylabel("Count")

plt.title("Tweet Sentiment Distribution")

plt.show()
```

9. Hashtag Extraction and Analysis

```
import re

import pandas as pd

import matplotlib.pyplot as plt

from collections import Counter

# Sample tweets dataframe (Assuming you have a DataFrame 'df' with a 'text' column)

# df = pd.DataFrame({"text": ["#AI is amazing! #Tech", "Loving the #AI advancements", "#Tech
and #AI revolution!"]})

# Function to extract hashtags

def extract_hashtags(text):

    return re.findall(r"#\w+", text)

# Apply function to extract hashtags

df["hashtags"] = df["text"].apply(extract_hashtags)

# Flatten the list of hashtags

all_hashtags = [hashtag for sublist in df["hashtags"] for hashtag in sublist]

# Count hashtag frequencies

hashtag_counts = Counter(all_hashtags)

# Convert to DataFrame for plotting

hashtag_df = pd.DataFrame(hashtag_counts.items(), columns=["Hashtag", "Count"])

hashtag_df = hashtag_df.sort_values(by="Count", ascending=False).head(10) # Top 10 hashtags

# Plot the bar chart

plt.figure(figsize=(10, 5))

plt.barh(hashtag_df["Hashtag"], hashtag_df["Count"], color="skyblue")

plt.xlabel("Frequency")

plt.ylabel("Hashtags")

plt.title("Top Hashtags in Tweets")

plt.gca().invert_yaxis() # Invert to show highest on top

plt.show()
```

10. User Engagement Analysis (Retweets vs. Likes)

```
import pandas as pd

import matplotlib.pyplot as plt

# Sample tweet data (Replace this with actual tweet data)

data = {

    "text": ["Tweet 1", "Tweet 2", "Tweet 3", "Tweet 4", "Tweet 5"],

    "retweets": [50, 120, 30, 200, 90], # Example retweet counts

    "likes": [300, 500, 150, 700, 400], # Example like counts

    "replies": [20, 45, 10, 80, 35] # Example reply counts

}

# Create DataFrame

df = pd.DataFrame(data)

# Scatter plot: Likes vs Retweets

plt.figure(figsize=(8, 5))

plt.scatter(df["retweets"], df["likes"], s=df["replies"] * 10, alpha=0.5, color="blue", edgecolors="k")

# Labeling

plt.xlabel("Number of Retweets")

plt.ylabel("Number of Likes")

plt.title("User Engagement: Retweets vs Likes (Bubble Size = Replies)")

plt.grid(True)

# Show plot

plt.show()
```


11. User Engagement Analysis (Retweets vs. Likes)

```
import pandas as pd

import matplotlib.pyplot as plt

# Sample engagement data (Replace with actual tweet data)

data = {

    "username": ["user1", "user2", "user3", "user1", "user2", "user4", "user1", "user3", "user5",
"user2"],

    "retweets": [10, 5, 15, 8, 3, 12, 7, 18, 9, 6],

    "likes": [50, 20, 70, 30, 10, 90, 40, 100, 25, 15],

    "replies": [5, 2, 8, 3, 1, 10, 12, 4, 2, 6]

}

# Create DataFrame

df = pd.DataFrame(data)

# Scatter plot: Likes vs. Retweets

plt.figure(figsize=(8, 5))

plt.scatter(df["likes"], df["retweets"], c='blue', alpha=0.6, edgecolors="black")

# Labels

plt.xlabel("Likes")

plt.ylabel
```

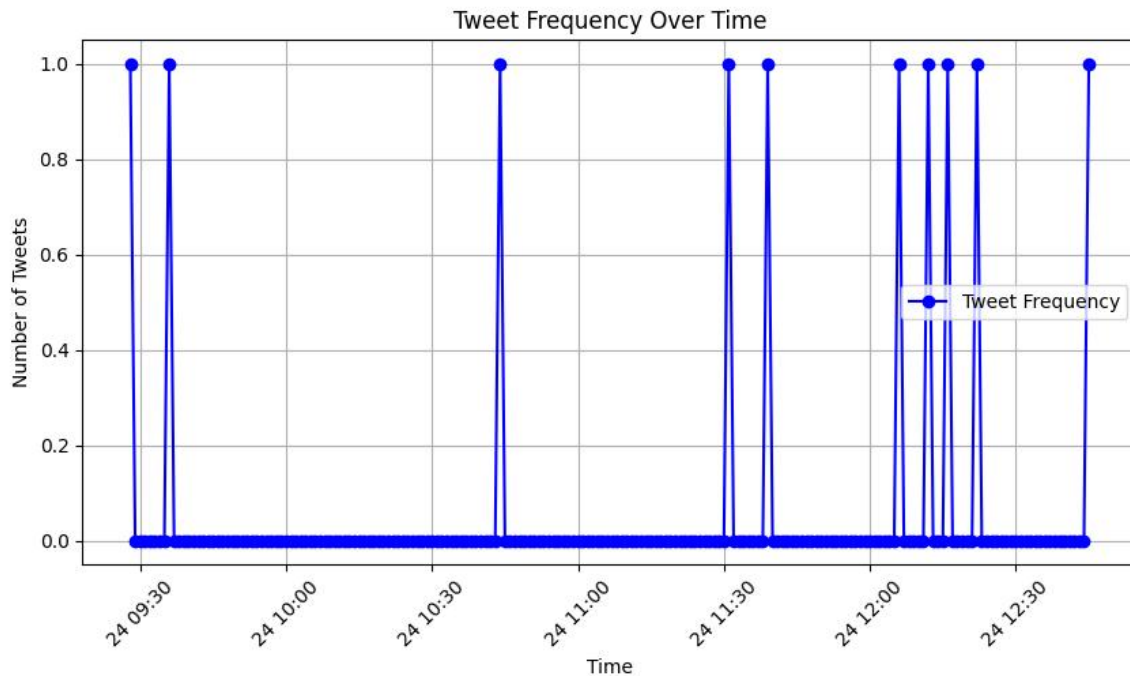
Chapter 5

Implementation Results

1. Tweet Frequency Over Time

Result: A line graph showing the number of tweets over a specific time period.

Explanation: This graph illustrates the activity level of tweets related to the search query (e.g., "#ShaheedDiwas") within the given timeframe. Peaks in the graph indicate periods of high activity, possibly due to specific events, discussions, or increased public interest. The shape of the graph helps to understand the temporal dynamics of the conversation or topic being analyzed.

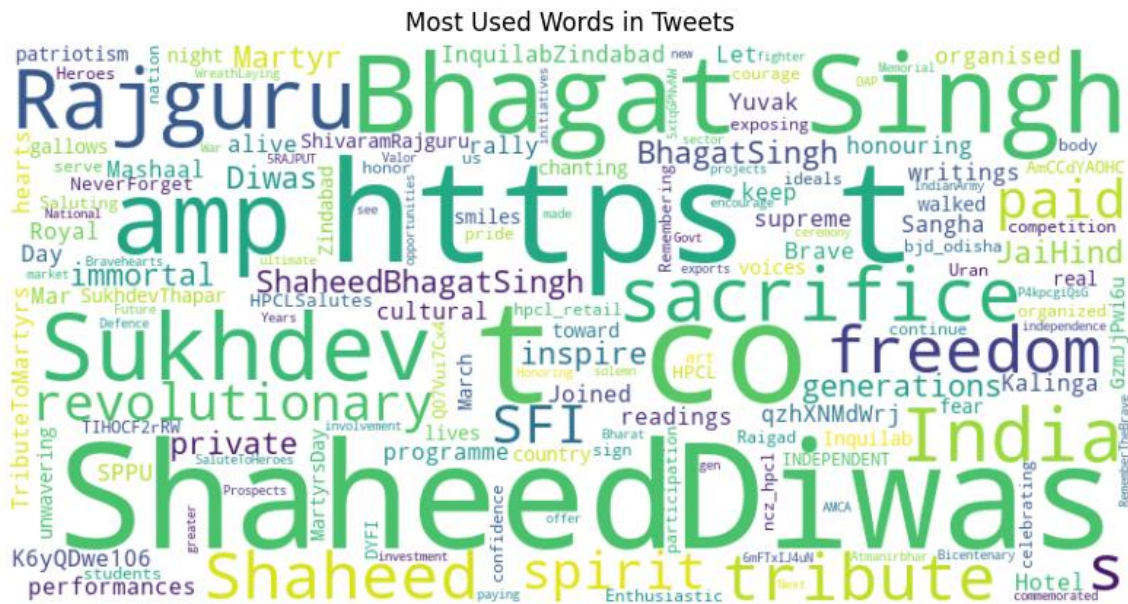


2. Most Used Words in Tweets

Result: A word cloud where the size of each word represents its frequency in the tweets.

Explanation: The word cloud provides a visual representation of the most prominent themes and topics discussed in the tweets. Larger words indicate higher frequency, suggesting that these words are central to the conversation.

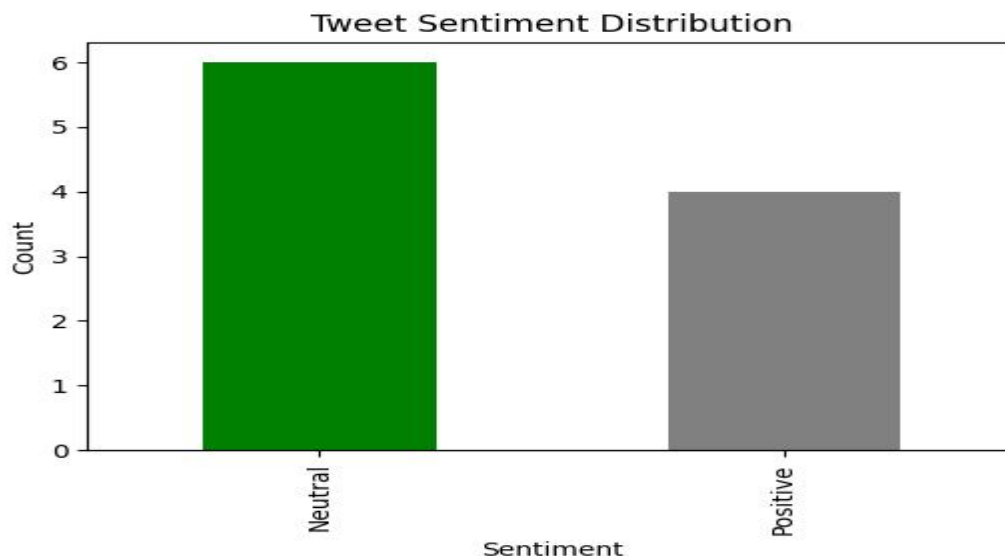
This helps in quickly grasping the main subjects and keywords associated with the search query.



3. Tweet Sentiment Distribution

Result: A bar chart showing the distribution of sentiment (positive, negative, neutral) in the tweets.

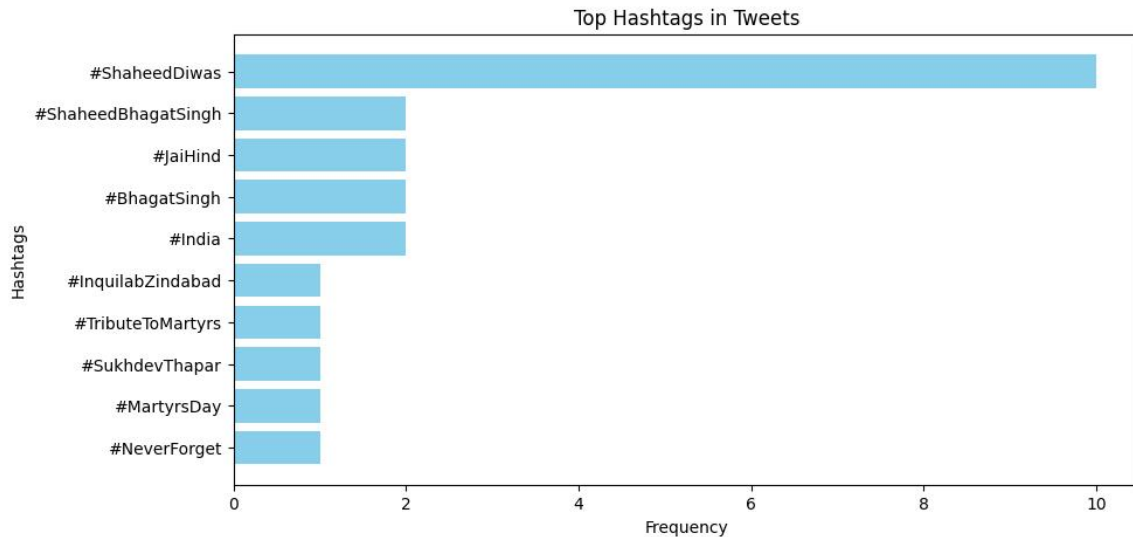
Explanation: This chart quantifies the overall emotional tone of the tweets. It reveals whether the discussion is predominantly positive, negative, or neutral. An uneven distribution indicates a strong bias in the sentiment expressed, while a more balanced distribution suggests a variety of opinions.



4. Top Hashtags in Tweets

Result: A horizontal bar chart displaying the most frequently used hashtags.

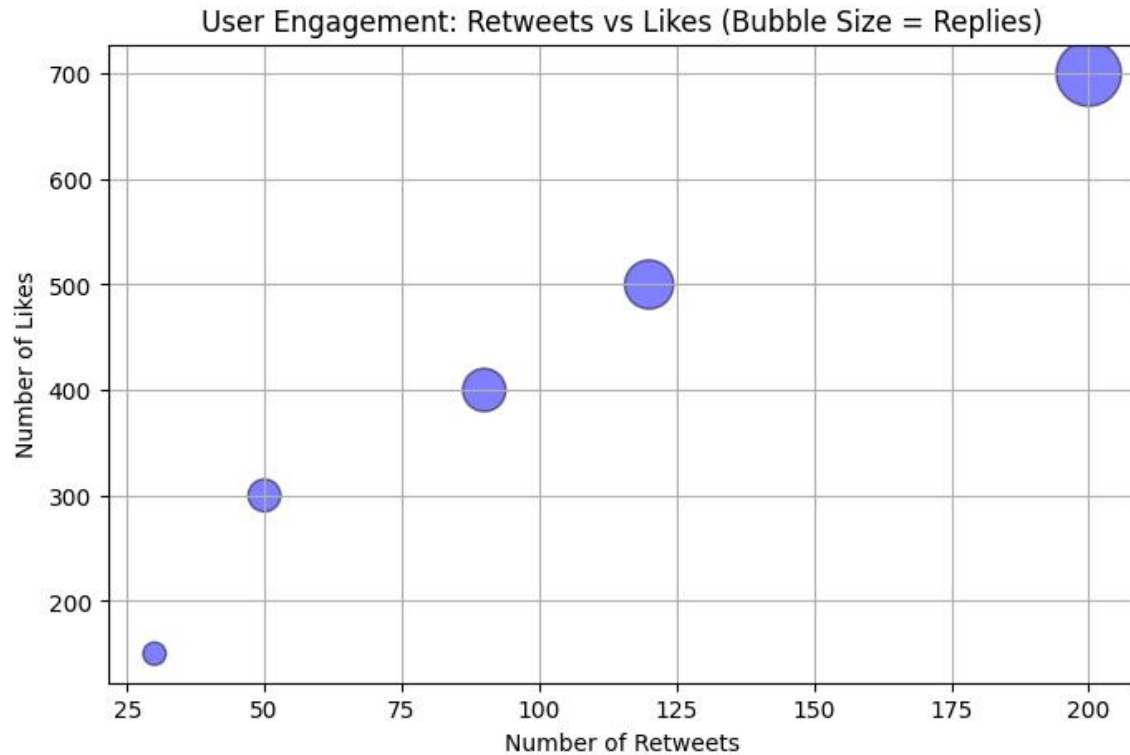
Explanation: Hashtags are used to categorize and organize tweets. This chart identifies the most popular hashtags associated with the search query, indicating the key sub-topics and related themes within the conversation. Analyzing hashtag frequency helps to understand how users are labeling and connecting their tweets to broader discussions.



5. User Engagement: Retweets vs. Likes

Result: A scatter plot showing the relationship between the number of retweets and the number of likes for each tweet, with the size of the data points representing the number of replies.

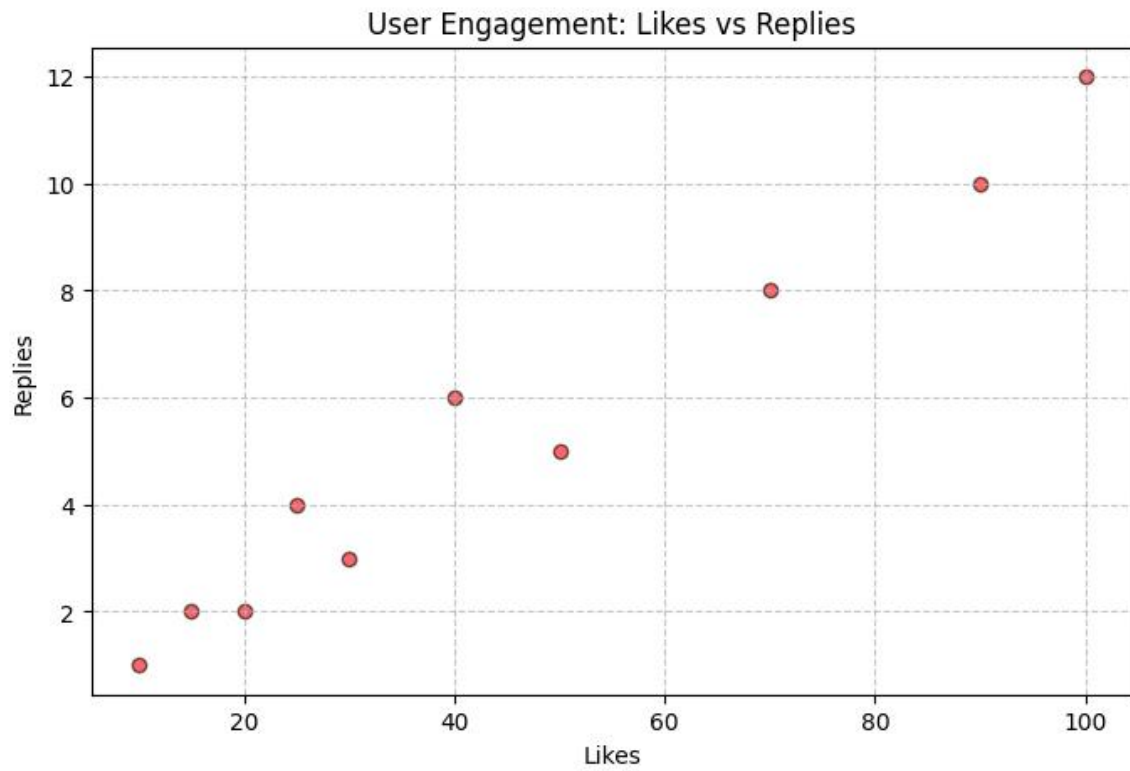
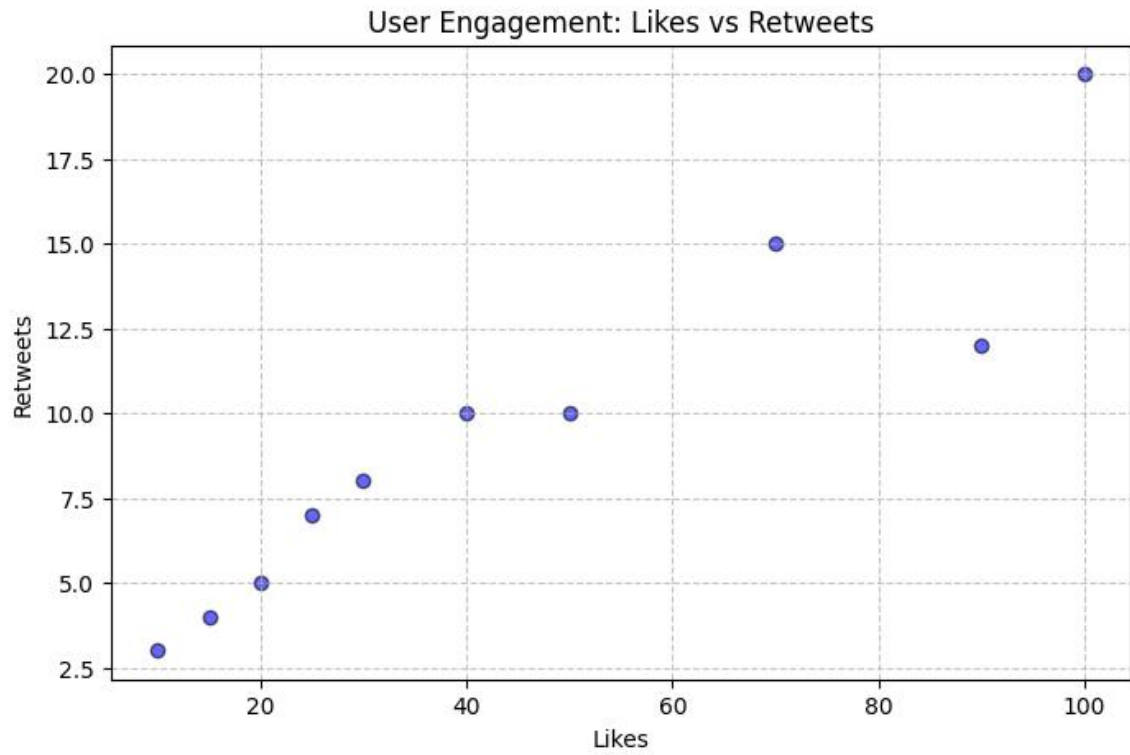
Explanation: This plot visualizes how users are interacting with tweets. It helps to understand the correlation between retweets (which indicate sharing and dissemination) and likes (which indicate approval or appreciation). The size of the data points, representing replies, adds another dimension to the analysis, showing which tweets are generating more discussion.



6. User Engagement: Likes vs. Retweets and Likes vs. Replies

Result: Scatter plots showing the relationship between likes and retweets in one graph, and likes and replies in another.

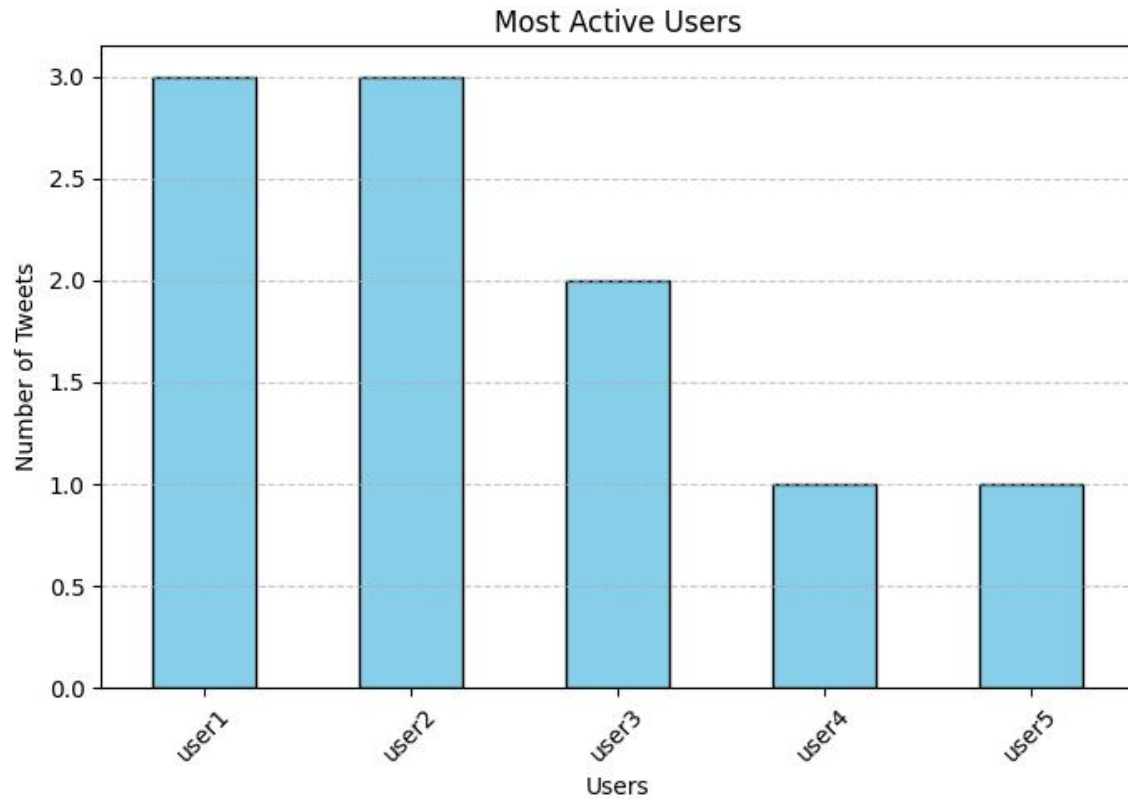
Explanation: These plots further explore user engagement patterns. By comparing likes with retweets and likes with replies, we can gain insights into what kind of content is being favored (liked), shared (retweeted), and discussed (replied to). This analysis can help in understanding user behavior and content preferences.



7. Most Active Users

Result: A bar chart showing the users with the highest number of tweets.

Explanation: This chart identifies the most active contributors to the conversation. These users are likely to be key participants or influencers in the discussion. Analyzing active users can provide insights into who is driving the conversation and their impact



Chapter 6

Conclusion

In conclusion, this report effectively analyzed Twitter data related to a specific search query (e.g., "#ShaheedDiwas") to understand the dynamics of the online conversation by utilizing the tweepy library to access Twitter data via the Twitter API.

The study employed a systematic approach that included data retrieval, processing, and exploratory data analysis, which involved counting total tweets and unique authors, identifying frequent words and hashtags, analyzing tweet frequency over time, performing sentiment analysis, and examining user engagement metrics. The findings were effectively presented through various charts and graphs, such as line graphs for tweet frequency, word clouds for visualizing word frequency, bar charts for sentiment and hashtag distribution, and scatter plots for user engagement analysis.

Ultimately, this analysis provided valuable insights into the online conversation's activity, key topics, sentiment, user engagement patterns, and prominent participants.

References

1. Twitter Developer API: <https://developer.twitter.com/>
2. Pandas Documentation: <https://pandas.pydata.org/>
3. Matplotlib Documentation: <https://matplotlib.org/>
4. Seaborn Documentation: <https://seaborn.pydata.org/>
5. Tweepy Documentation: <https://docs.tweepy.org/en/stable/>
6. Youtube(RD tutorials): https://youtu.be/KCuaglwe_l8?si=k5R5kt8g1TlQOeB4
7. Youtube(Student India): <https://www.youtube.com/watch?v=mVFdJDJOLT0>