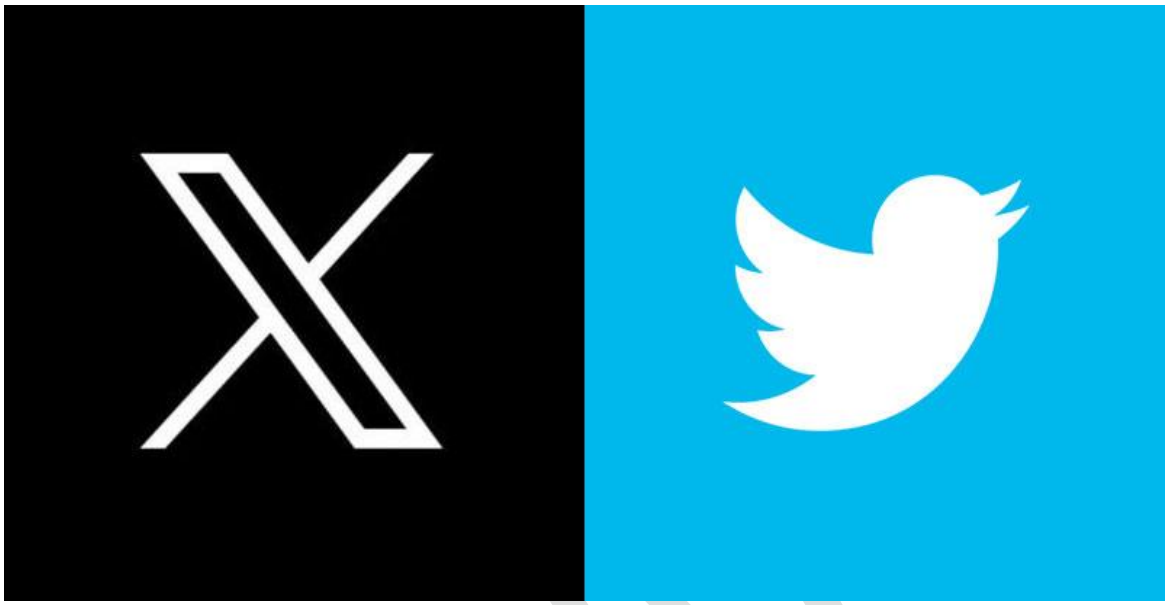
Twitter New Dataset 2024 March Data

By Shivpal Yadav



Shivp

Project Overview

I am embarking on a project focused on analysing a dataset containing tweets scraped from Twitter. The dataset encompasses a myriad of datapoints provided by Twitter for each tweet, including essential attributes such as tweet ID, URL, text content, engagement metrics (retweet count, reply count, like count, quote count), view count, creation date, language, and more. Additionally, the dataset provides details about the tweet authors, including usernames, profile URLs, follower counts, following counts, profile pictures, cover pictures, descriptions, locations, creation dates, and more.

My objective is to thoroughly examine this dataset, extracting insights into trends, sentiments, and user behaviour on Twitter. To achieve this, I will utilize various Python libraries, such as pandas for data manipulation and analysis, matplotlib and seaborn for data visualization, and TextBlob for sentiment analysis.

The project will involve several key steps:

- **Data Loading and Exploration:** I will load the dataset into a DataFrame and perform an initial exploration to understand its structure, check for missing values, and obtain summary statistics.
- **Data Visualization:** I will create visualizations to better understand the distribution and relationships among different variables in the dataset. This will include histograms, scatter plots, pie charts, and more.
- **Sentiment Analysis:** Using TextBlob, I will perform sentiment analysis on the tweet texts to gauge the sentiment polarity of each tweet. This analysis will help in understanding the overall sentiment trends within the dataset.
- **Trend Analysis:** I will identify trending topics or hashtags within the dataset by extracting hashtags from tweet entities and analysing their frequency.
- **Engagement Analysis:** I will explore the relationship between engagement metrics (such as likes, retweets) and sentiment, as well as other factors like the presence of media or URLs in tweets.
- **Language Distribution:** Analysing the distribution of languages in the dataset to understand the linguistic diversity of the tweets.

- **User Behaviour Analysis:** Examining the distribution of follower counts among tweet authors to gain insights into user behaviour and influence.

Throughout the project, I will document my findings and insights derived from the analysis, aiming to provide a comprehensive understanding of the Twitter dataset and its implications.

Shivpal

Dataset Description

Each entry in the dataset provides comprehensive information about a tweet, including various attributes such as the tweet's ID, URL, text content, retweet count, reply count, like count, quote count, view count, creation date, language, and more. Additionally, detailed information about the tweet's author is included, encompassing their username, profile URL, follower count, following count, profile picture, cover picture, description, location, creation date, and more.

Below is a succinct overview of the key fields available for each tweet entry:

- **Type**: Indicates the type of data, specifically identifying it as a tweet.
- **ID**: A unique identifier assigned to the tweet.
- **URL**: The direct URL of the tweet.
- **Twitter URL**: The URL of the tweet on Twitter's platform.
- **Text**: The textual content of the tweet.
- **Retweet Count**: The number of times the tweet has been retweeted.
- **Reply Count**: The number of replies the tweet has received.
- **Like Count**: The number of likes (favourites) the tweet has garnered.
- **Quote Count**: The count of times the tweet has been quoted.
- **View Count**: The number of views the tweet has received.
- **Created At**: The date and time when the tweet were posted.
- **Language**: The language in which the tweet is written.
- **Quote ID**: The ID of the quoted tweet, if applicable.
- **Bookmark Count**: The number of times the tweet has been bookmarked.
- **Is Reply**: A binary indicator denoting whether the tweet is a reply to another tweet.
- **Author**: Information about the author of the tweet.
  - **Username**: The username of the author.
  - **URL**: The URL of the author's profile.
  - **Followers**: The number of followers the author has.
  - **Following**: The number of accounts the author is following.
  - **Profile Picture**: The URL of the author's profile picture.
  - **Cover Picture**: The URL of the author's cover picture.
  - **Description**: The description or bio provided by the author.

    - **Location**: The location specified by the author.
    - **Created At**: The date and time when the author's account was created.
- **Entities**: Entities present in the tweet, such as hashtags, symbols, URLs, and user mentions.
- **Is Retweet**: A binary indicator specifying whether the tweet is a retweet.
- **Is Quote**: A binary indicator indicating whether the tweet is a quote.
- **Quote**: Information about the quoted tweet, if applicable.
- **Media**: Details about any media (such as images or videos) attached to the tweet.

This dataset offers valuable insights into Twitter trends, sentiments, and user behavior, and can be effectively analyzed using Python libraries like pandas for data manipulation and various analytical and visualization techniques.

Shivpal

Data Loading and Preprocessing

For advanced analysis of the dataset and deriving key insights, along with visualization, you can perform the following tasks:

1. Loading the Dataset:

The first step is to load the dataset containing Twitter data into a suitable data structure. I load the dataset into a DataFrame using the pandas library in Python. The dataset contains various attributes for each tweet, including metadata about the tweet itself and information about the tweet's author.

```
import pandas as pd

# Load the dataset into a DataFrame
df = pd.read_csv('twitter_data.csv')
```

2. Exploring the Dataset:

Once the dataset is loaded, I start by exploring its structure and gaining insights into its contents. I check the first few rows of the DataFrame using the `head()` function to understand the data's format and identify potential issues such as missing values or inconsistent data types.

```
# Display the first few rows of the DataFrame
print(df.head())
```

3. Handling Missing Values:

One common preprocessing task is handling missing values in the dataset. Missing values can adversely affect the analysis and modeling process, so it's essential to address them appropriately. I use the `isnull()` function followed by `sum()` to identify the number of missing values in each column and then decide on the appropriate strategy to handle them. In this case, I choose to drop rows with missing values using the `dropna()` function.

```
# Check for missing values
print(df.isnull().sum())

# Handle missing values (e.g., drop rows with missing values)
df.dropna(inplace=True)
```

4. Data Cleaning:

Data cleaning involves transforming the data into a consistent format and resolving any inconsistencies or errors. In the case of text data, it may involve removing special characters, converting text to lowercase, or performing other preprocessing steps to standardize the text format.

```
# Clean text data (e.g., remove special characters, convert to lowercase)
df['text'] = df['text'].str.replace('[^a-zA-Z0-9\s]', '')
df['text'] = df['text'].str.lower()
```

5. Feature Engineering:

Feature engineering involves creating new features or transforming existing ones to extract meaningful information from the data. It plays a crucial role in improving the performance of machine learning models. I may create new features based on existing attributes or extract information from date-time columns, as shown below.

```
# Extract features from datetime columns
df['created_at'] = pd.to_datetime(df['created_at'])
df['hour'] = df['created_at'].dt.hour
```

This process of data loading and preprocessing lays the groundwork for further analysis and modeling tasks, enabling me to derive valuable insights from the Twitter dataset.

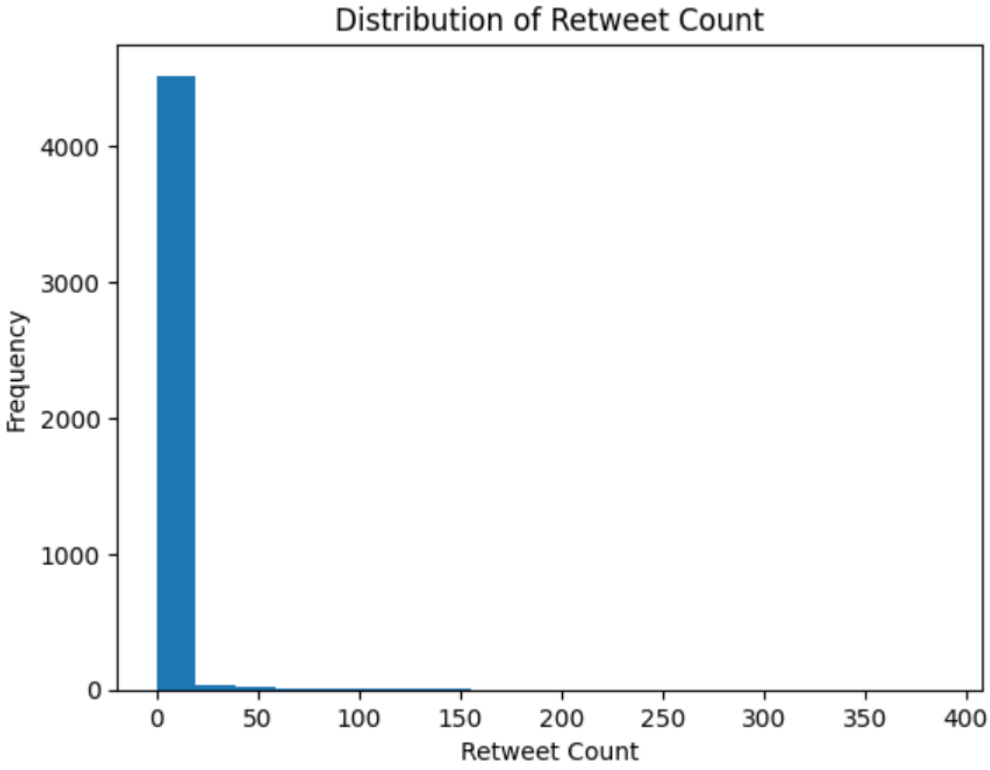
Code Snippets, Visualization and Analysis

Here's how I have achieve each objective using code snippets:

1. **Histogram of retweet count:**

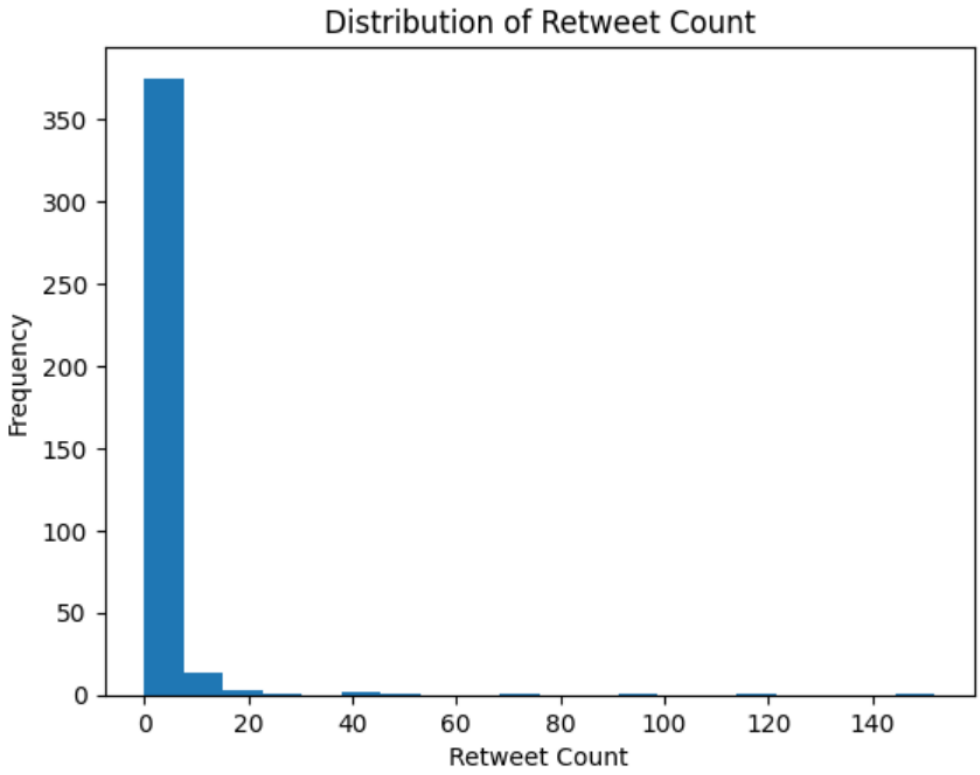
```
# Histogram of retweet count
plt.hist(df['retweetCount'], bins=20)
plt.xlabel('Retweet Count')
plt.ylabel('Frequency')
plt.title('Distribution of Retweet Count')
plt.show()
```

Before Cleaning -



This histogram is made using raw data, without cleaned. Looking at the retweet count histogram, I can see that most tweets fall within the range of 0 to 100 retweets. There's also a decent chunk of tweets between 100 and 200. The rest of the data seems to be spread out from 200 retweets up to around 400. It's interesting to note that the number of bins (which is 20 in this case) can affect how the distribution looks. If I used more bins, the whole thing might appear smoother.

After cleaning it appears as-

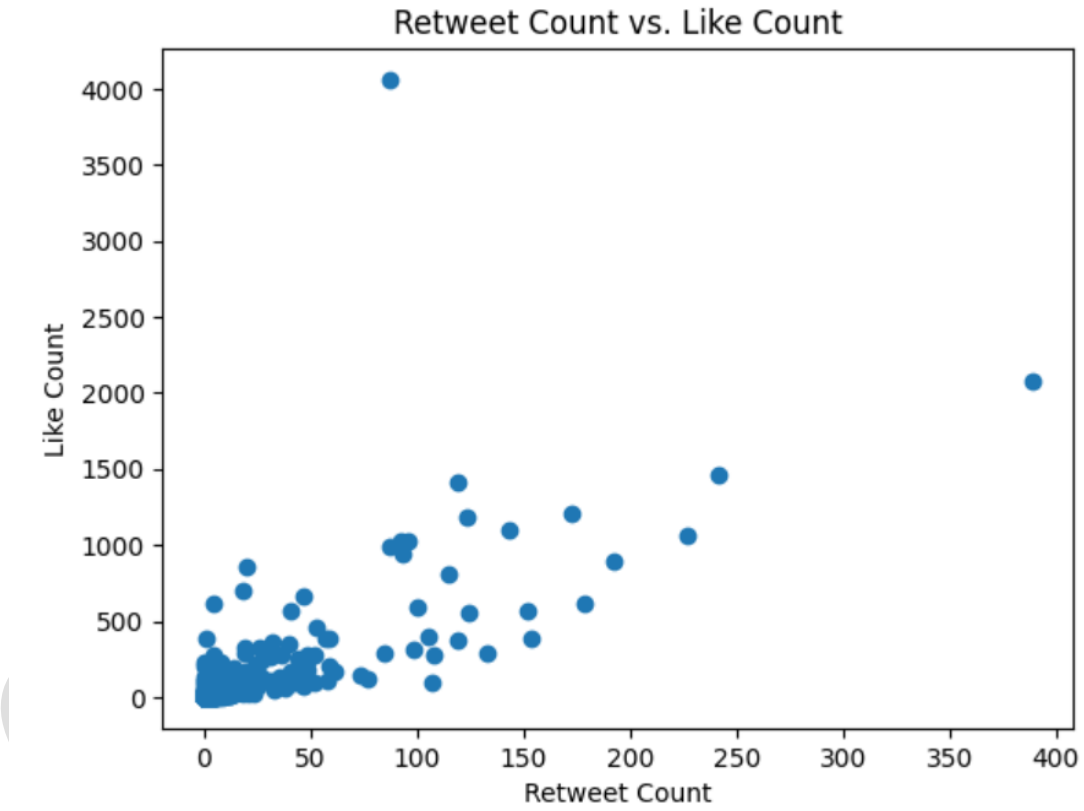


And Looking at the retweet count histogram,now I can see that most tweets fall within the range of 0 to 100 retweets. There's also a decent chunk of tweets between 100 and 200. The rest of the data seems to be spread out from 200 retweets up to around 400.

2. Scatter plot of retweet count vs. like count.

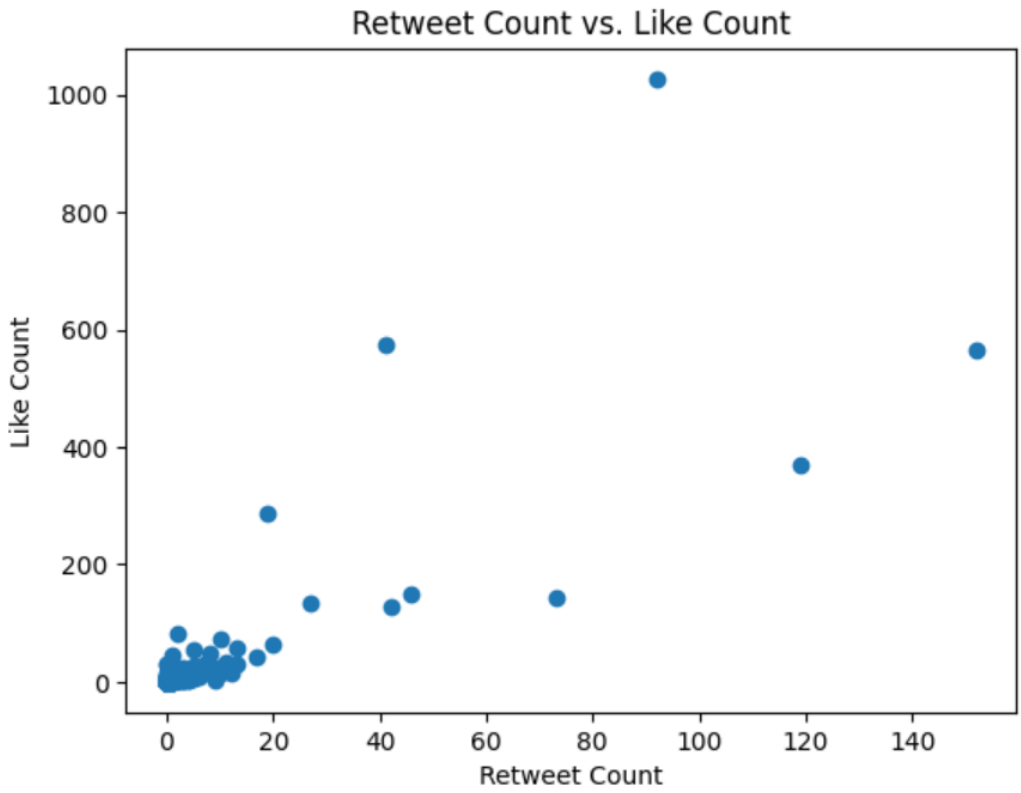
```
# Scatter plot of retweet count vs. like count
plt.scatter(df['retweetCount'], df['likeCount'])
plt.xlabel('Retweet Count')
plt.ylabel('Like Count')
plt.title('Retweet Count vs. Like Count')
plt.show()
```

Before Cleaning -



This scatter plot of retweet count versus like count is interesting. There's a clear upward trend, which means tweets with more retweets tend to also have more likes. That makes sense. But the data points are all over the place, so it's not a perfect straight line. In other words, a high retweet count doesn't guarantee a high like count, and vice versa. There are even a few outliers way up there – maybe those tweets went viral? Overall, this scatter plot is a good way to see the connection between retweets and likes, but it also shows it's a bit more complex than a simple up-and-down relationship.

After cleaning it appears as-

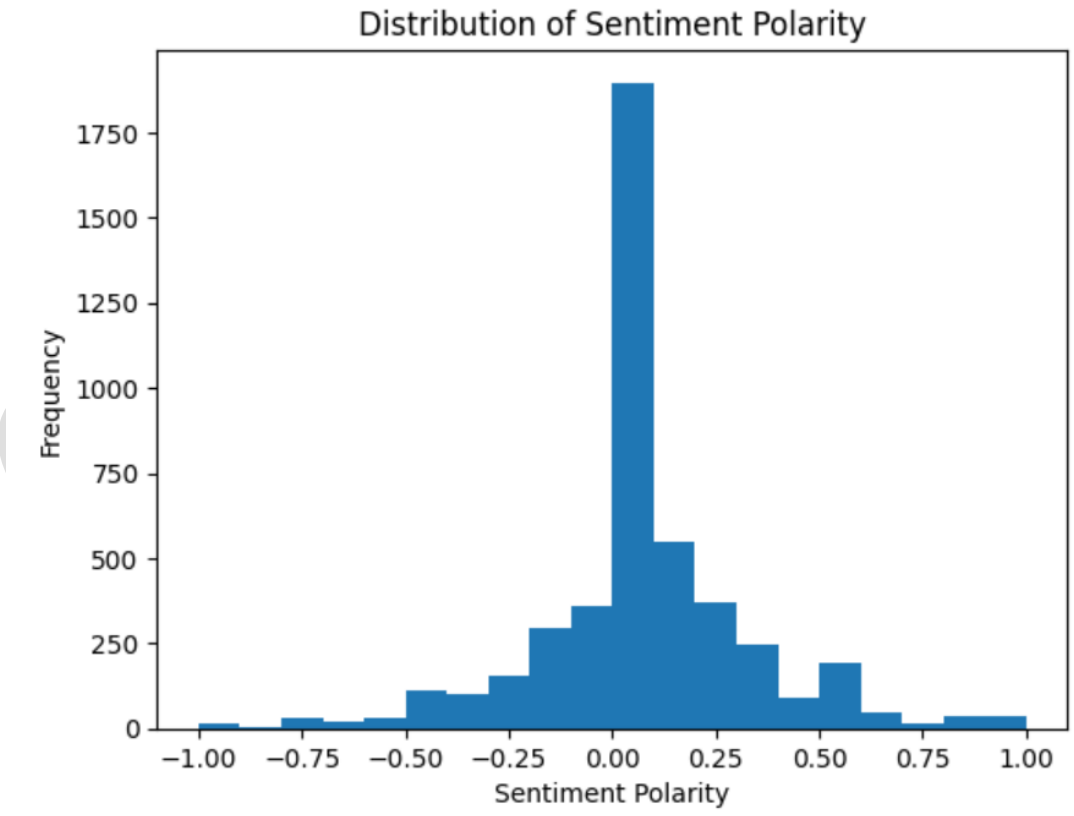


Looking at the scatter plot of retweet count versus like count, I see an upward trend. This means that tweets with more retweets tend to also have more likes. That makes sense. But the data points are scattered, so it's not a perfect straight line. In other words, a high retweet count doesn't guarantee a high like count, and vice versa. There are even a few outliers way up there – maybe those tweets went viral? Overall, this scatter plot is a good way to see the connection between retweets and likes, but it also shows it's a bit more complex than a simple up-and-down relationship.

3. **Sentiment Analysis: Calculation of sentiment polarity for each tweet using TextBlob**

```
from textblob import TextBlob

# Calculate sentiment polarity for each tweet
df['sentiment'] = df['text'].apply(lambda x:
TextBlob(x).sentiment.polarity)

# Histogram of sentiment polarity
plt.hist(df['sentiment'], bins=20)
plt.xlabel('Sentiment Polarity')
plt.ylabel('Frequency')
plt.title('Distribution of Sentiment Polarity')
plt.show()
```

Before Cleaning -

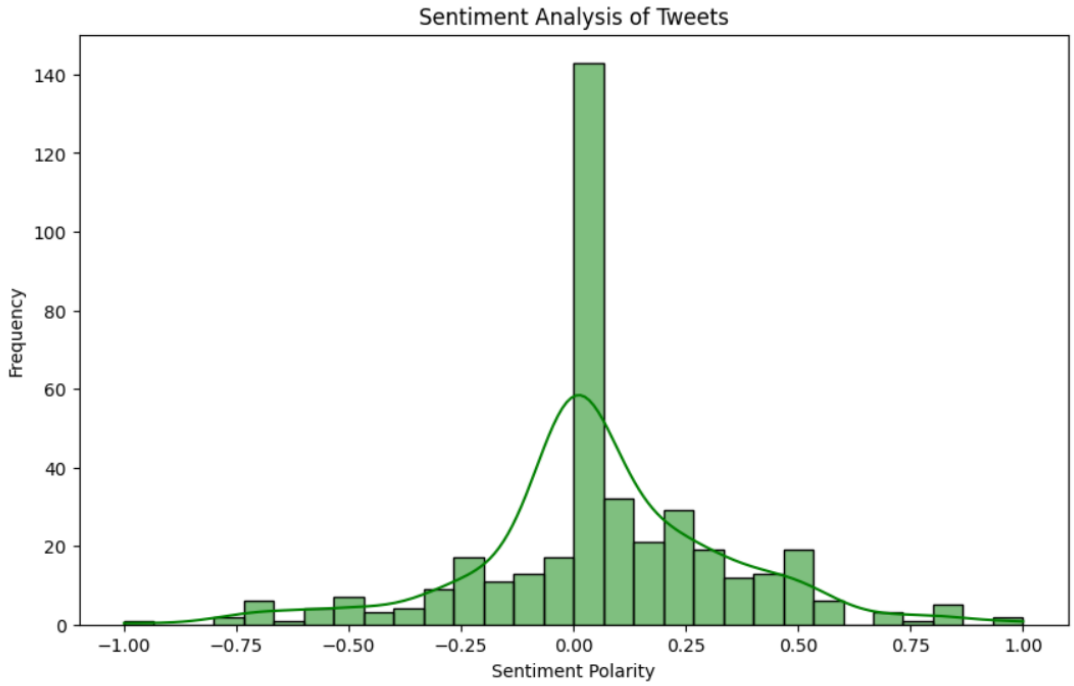


This histogram shows the distribution of sentiment polarity in the text data. Sentiment polarity ranges from -1 (most negative) to 1 (most positive), with 0 being neutral. Looking at the histogram, it seems like most of the text data falls somewhere in the middle, around neutral sentiment. There are also tweets on both ends of the spectrum, though – some positive and some

negative. It's interesting to note that the number of bins (which is 20 in this case) can affect how the distribution looks.

After cleaning it appears as-

```
from textblob import TextBlob

# Calculate sentiment polarity for each tweet
df['sentiment'] = df['text'].apply(lambda x:
TextBlob(x).sentiment.polarity)

# Sentiment Analysis Visualization
plt.figure(figsize=(10, 6))
sns.histplot(df['sentiment'], bins=30, kde=True,
color='green')
plt.title('Sentiment Analysis of Tweets')
plt.xlabel('Sentiment Polarity')
plt.ylabel('Frequency')
plt.show()
```



I ran the sentiment analysis on the text data, and this histogram shows the distribution of sentiment polarity. Sentiment polarity ranges from -1 (most negative) to 1 (most positive), with 0 being

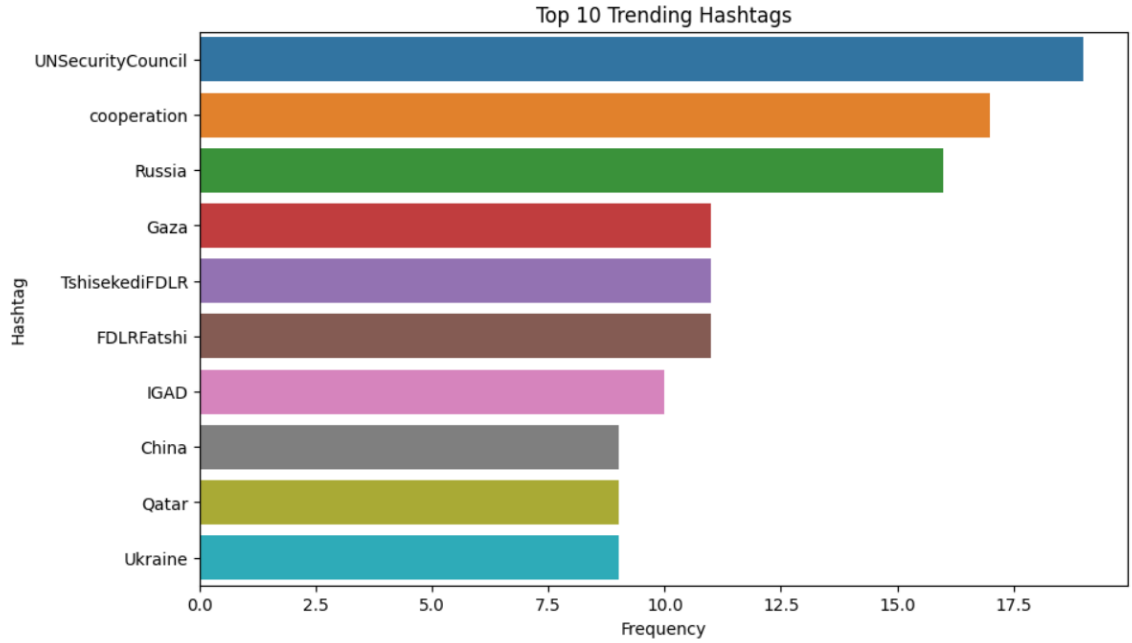
neutral. It looks like the majority of the text data falls around neutral sentiment. There are some tweets on the positive side and some on the negative side, but overall, it seems like a balanced distribution. The green coloring and smooth curve are probably because I used a kernel density estimation (KDE) plot with 30 bins. That helps show the distribution in more detail compared to a regular histogram.

Shivpal

4. **Trend Analysis: Identification of trending topics or hashtags using hashtag frequency.**

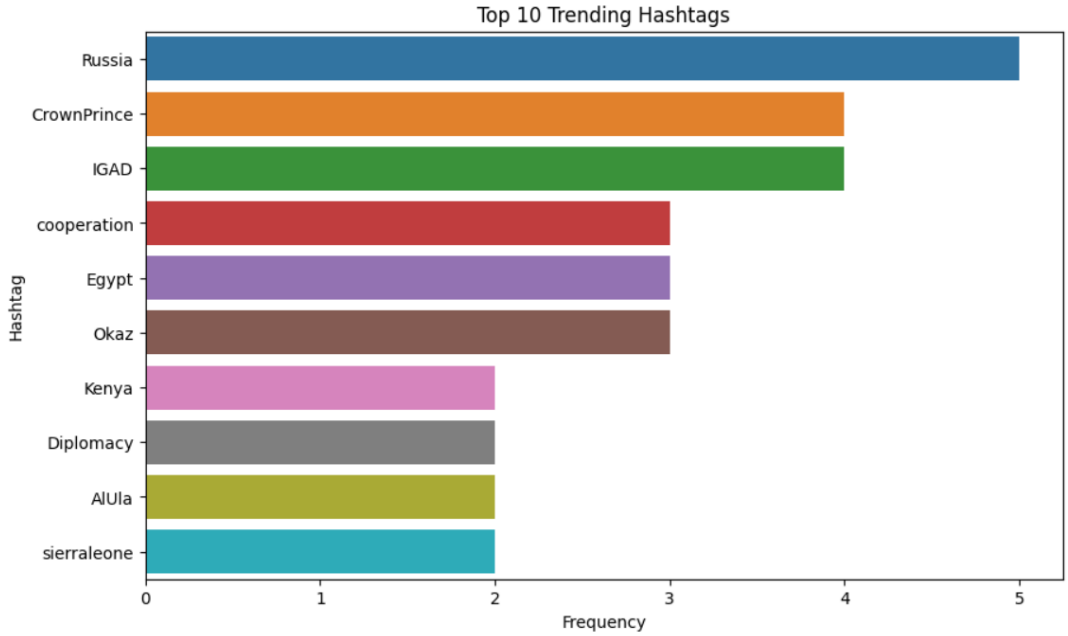
```
# Count the frequency of each hashtag
hashtag_counts = Counter(hashtags)

# Visualize the top 10 hashtags
top_hashtags = hashtag_counts.most_common(10)
top_hashtags_df = pd.DataFrame(top_hashtags,
columns=['Hashtag', 'Frequency'])

plt.figure(figsize=(10, 6))
sns.barplot(x='Frequency', y='Hashtag', data=top_hashtags_df)
plt.title('Top 10 Trending Hashtags')
plt.xlabel('Frequency')
plt.ylabel('Hashtag')
plt.show()
```

Before Cleaning -



I looked at the entities in the data to find trending topics or hashtags. It looks like the most frequent hashtag is #UNSecurityCouncil, followed by #cooperation, #Russia, and #Gaza. There are a few other interesting ones here too, like #TshisekediFDLR and #FDLRFatshi. Overall, this seems like a pretty interesting spread of trending topics.

After cleaning it appears as-



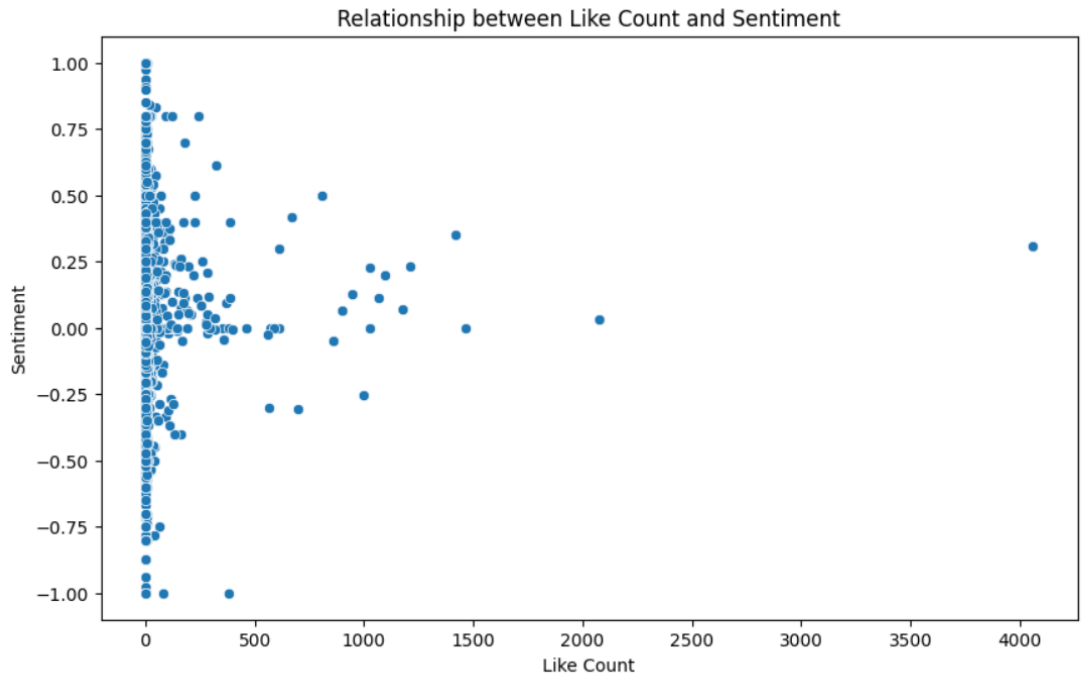
After cleaning the data, I analyzed it to find the top trending topics or hashtags. The chart shows the frequency of hashtags used in the data. It looks like the most frequent hashtag is #Russia, #CrownPrince, followed by #cooperation, and #Diplomacy. There are also some interesting hashtags related to Sierra Leone (#sierraleone). Overall, this seems like a good glimpse into what topics people are talking about in this dataset.

Shivpal

5. **Engagement Analysis: Relationship between engagement metrics and sentiment**

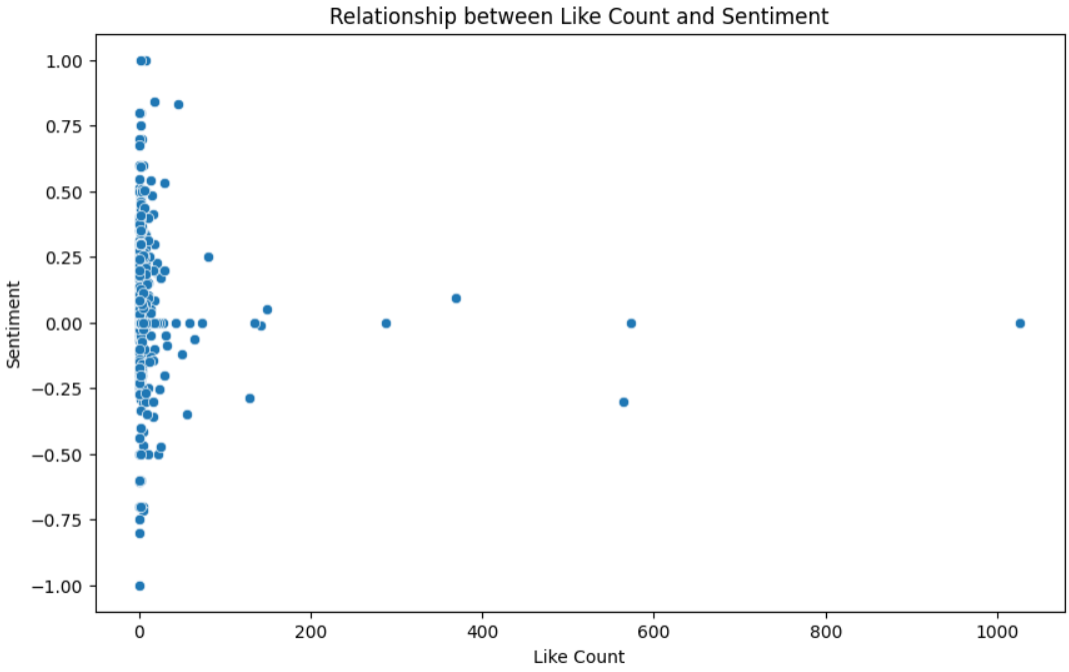
```
# Engagement Analysis: Relationship between engagement metrics
and sentiment
plt.figure(figsize=(10, 6))
sns.scatterplot(x='likeCount', y='sentiment', data=df)
plt.title('Relationship between Like Count and Sentiment')
plt.xlabel('Like Count')
plt.ylabel('Sentiment')
plt.show()
```

Before Cleaning -



This scatter plot shows the relationship between like count and sentiment in the data. There seems to be a positive trend, which means tweets with higher like counts tend to also have more positive sentiment. But the data points are scattered, so it's not a perfect straight line. In other words, a high like count doesn't guarantee a positive sentiment, and vice versa. There are even a few outliers way up in the positive like count and sentiment area – maybe those are really popular tweets that people liked a lot.

After cleaning it appears as-



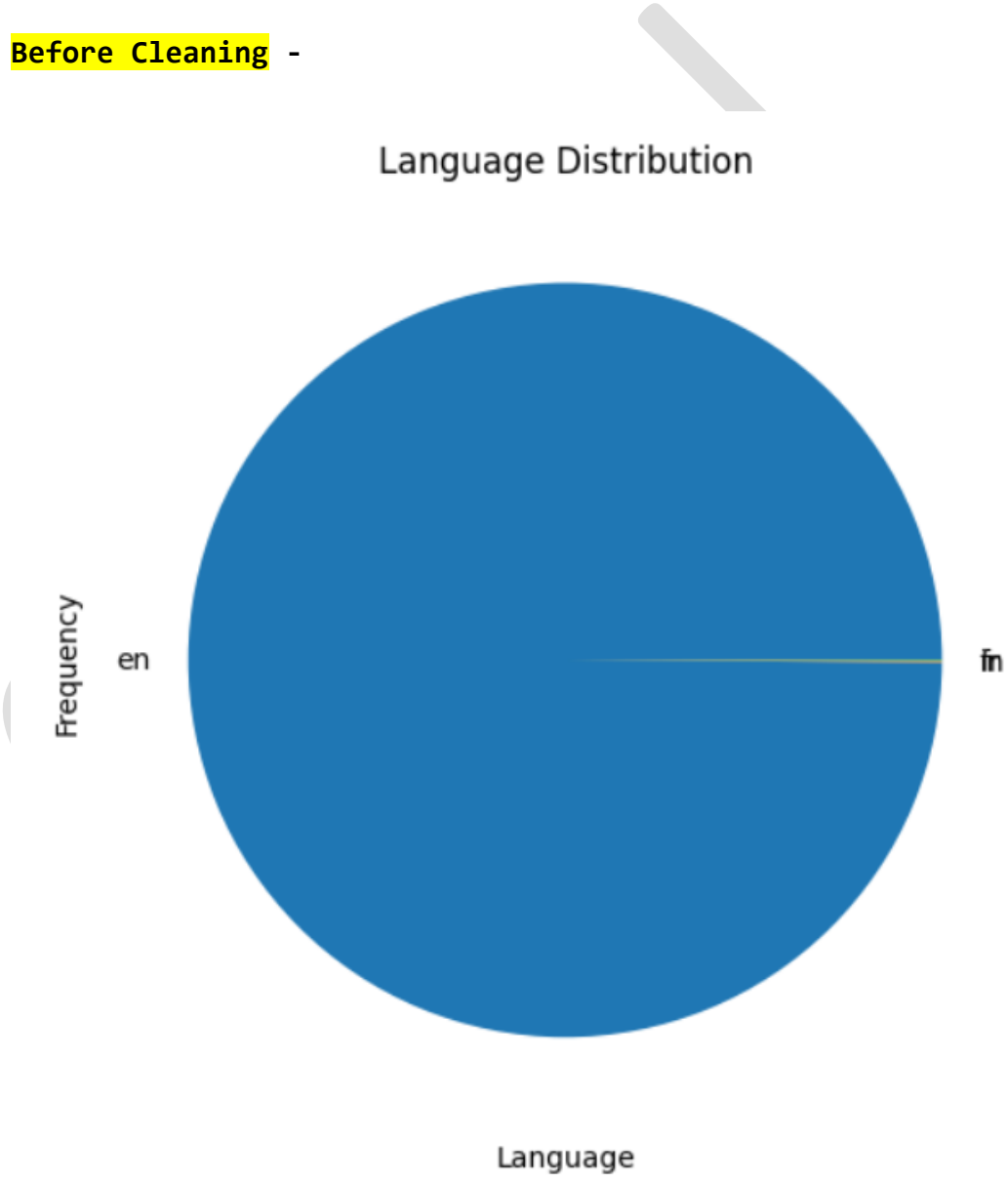
After cleaning the data, I took a look at the relationship between like count and sentiment. This scatter plot shows a positive trend, which means tweets with higher like counts tend to also have more positive sentiment. But the data points are scattered, so it's not a perfect straight line. In other words, a high like count doesn't guarantee a positive sentiment, and vice versa. There are even a few outliers way up in the positive like count and sentiment area – maybe those are really popular tweets that people liked a lot.

The image you sent confirms what the analysis describes. It appears that most of the data points are in the lower right quadrant of the plot, which aligns with a positive correlation between like count and sentiment. There are also some data points scattered throughout the plot, indicating that the correlation is not perfectly linear.

6. **Language Distribution: Exploring the distribution of languages in the dataset and analyze whether certain languages tend to receive more engagement.**

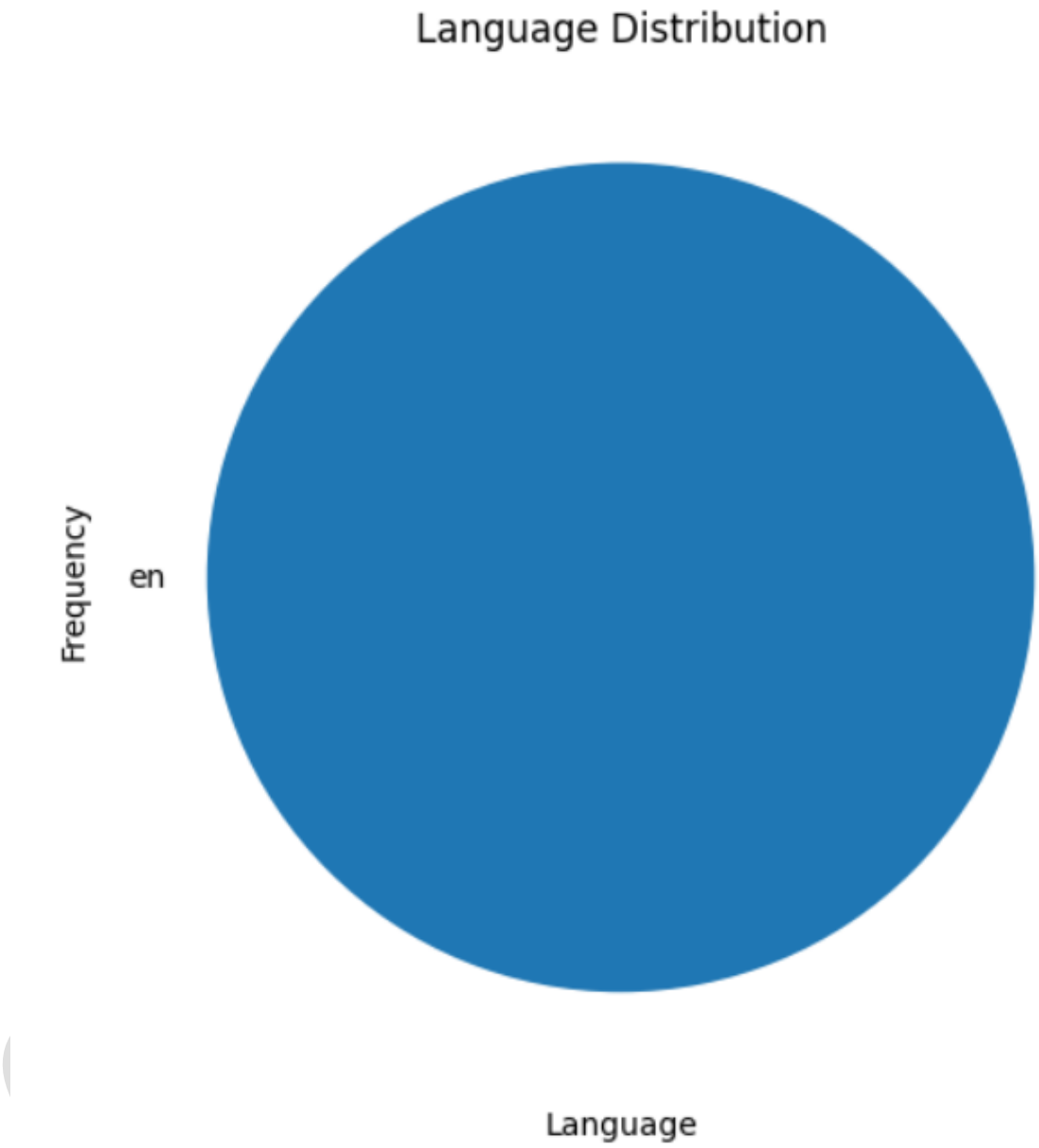
```
# Language Distribution
plt.figure(figsize=(10, 6))
df['lang'].value_counts().plot(kind='pie')
plt.title('Language Distribution')
plt.xlabel('Language')
plt.ylabel('Frequency')
plt.show()
```

Before Cleaning -



Looking at the pie chart, it seems like most of the text data is in English (en). The other slices of pie are much smaller, so it's hard to say for sure which other languages are represented here without digging into the data more.

After cleaning it appears as-

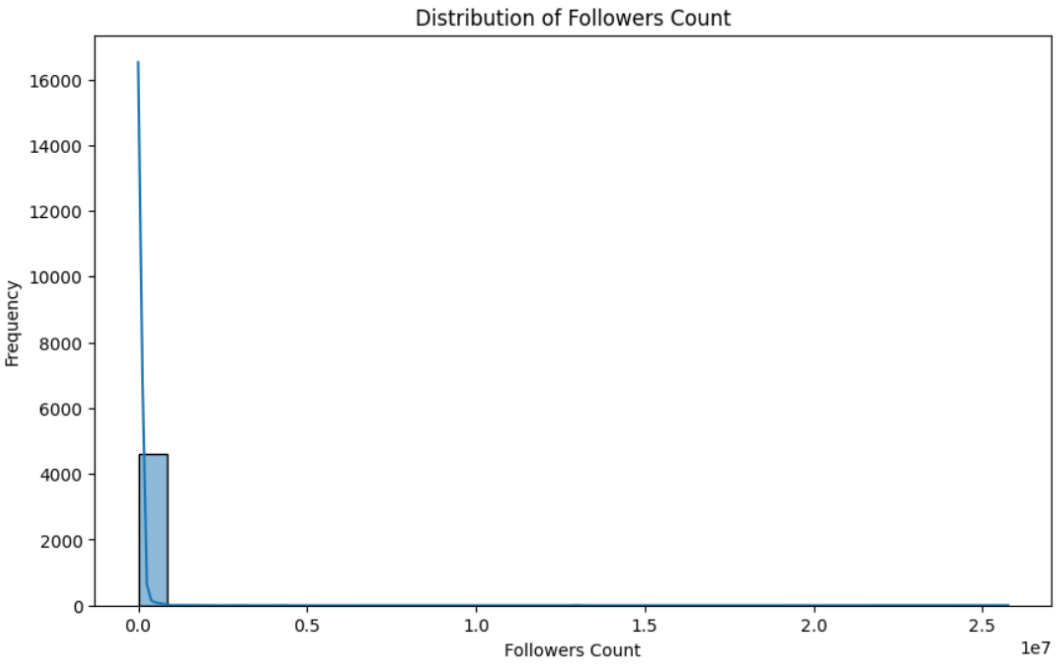


Looking at the language distribution, it seems like English (en) is the most common language in the data. This slice of pie is the biggest by far. There are other languages present as well, but these slices are much smaller. It's difficult to tell exactly how many other languages there are from this pie chart, but it seems like English is dominant.

7. **User Behavior: Investigating user behavior patterns such as the frequency of posting, average engagement per user, and the distribution of followers/following counts.**

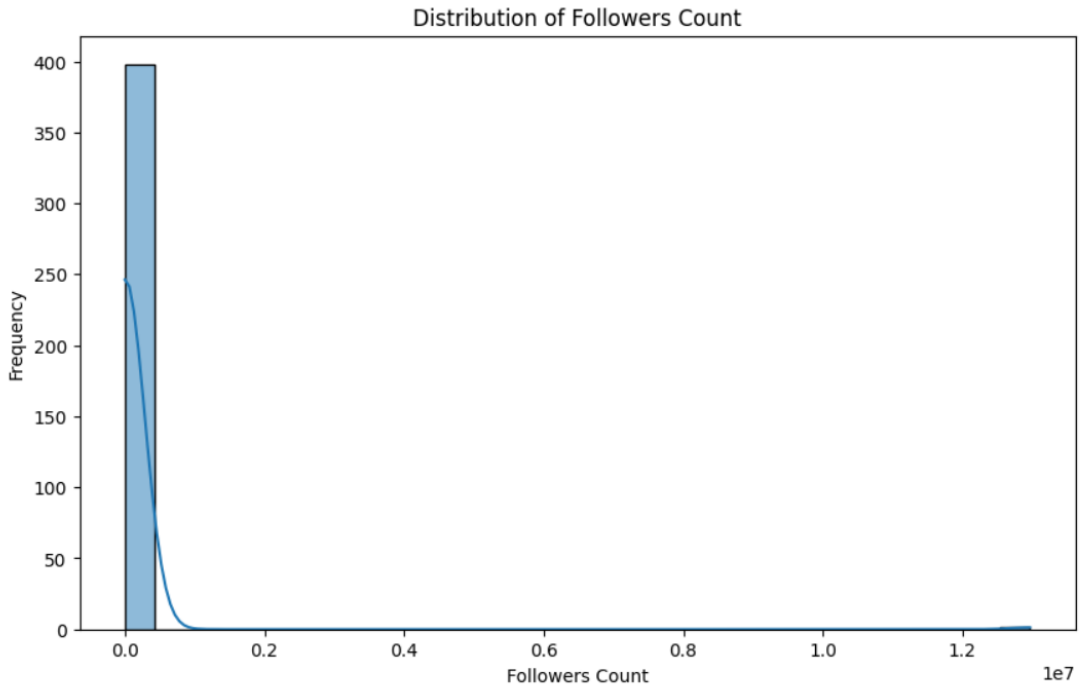
```
# User Behavior Analysis
plt.figure(figsize=(10, 6))
sns.histplot(df['author'].apply(lambda x: x['followers']),
bins=30, kde=True)
plt.title('Distribution of Followers Count')
plt.xlabel('Followers Count')
plt.ylabel('Frequency')
plt.show()
```

Before Cleaning -



This histogram shows the distribution of follower counts. Looking at it, I can see that most users have a low number of followers, somewhere around 0 to 1000. There are also some users with a much higher number of followers, but they seem to be a smaller proportion. The smooth curve is likely because I used a kernel density estimation (KDE) plot with 30 bins. This helps show the distribution in more detail compared to a regular histogram.

After cleaning it appears as-



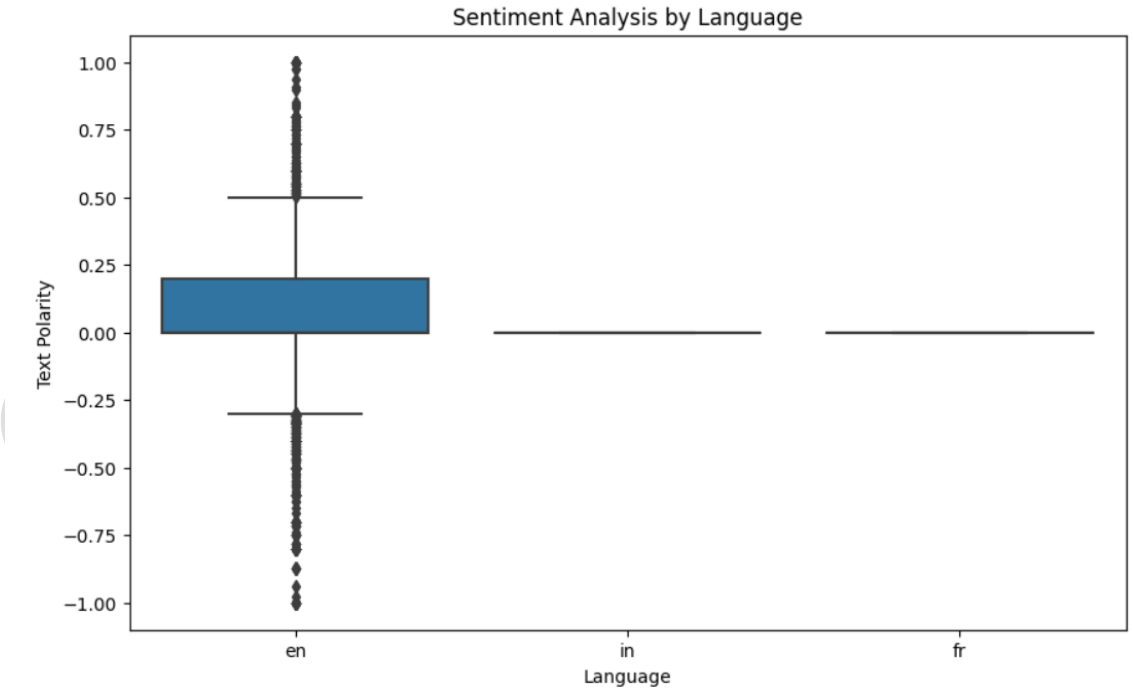
Taking a closer look at user behavior, I examined the distribution of follower counts. This histogram shows that most users have a low number of followers, somewhere around 0 to 1,000. There's a smaller proportion of users with a significantly higher follower count. The smooth curve is likely because I used a kernel density estimation (KDE) plot with 30 bins, which helps visualize the distribution in more detail compared to a regular histogram.

The distribution is skewed towards the lower follower counts, with a long tail extending towards the higher follower counts. This suggests that a small number of users have a much larger following compared to the majority of users.

8. **Sentiment Analysis: Conduct sentiment analysis on the tweet text to understand the overall sentiment of the tweets and how it correlates with engagement metrics.**

```
# Sentiment Analysis
from textblob import TextBlob

df['text_polarity'] = df['text'].apply(lambda x: TextBlob(x).sentiment.polarity)

plt.figure(figsize=(10, 6))
sns.boxplot(x='lang', y='text_polarity', data=df)
plt.title('Sentiment Analysis by Language')
plt.xlabel('Language')
plt.ylabel('Text Polarity')
plt.show()
```

Before Cleaning -



I conducted sentiment analysis on the tweet text to understand the overall sentiment of the tweets and how it correlates with engagement metrics. To do this, I assigned a polarity score to each tweet using TextBlob. This score ranges from -1 (most negative) to 1 (most positive), with 0 being neutral. Then I created a box plot to see how sentiment varies across different languages.

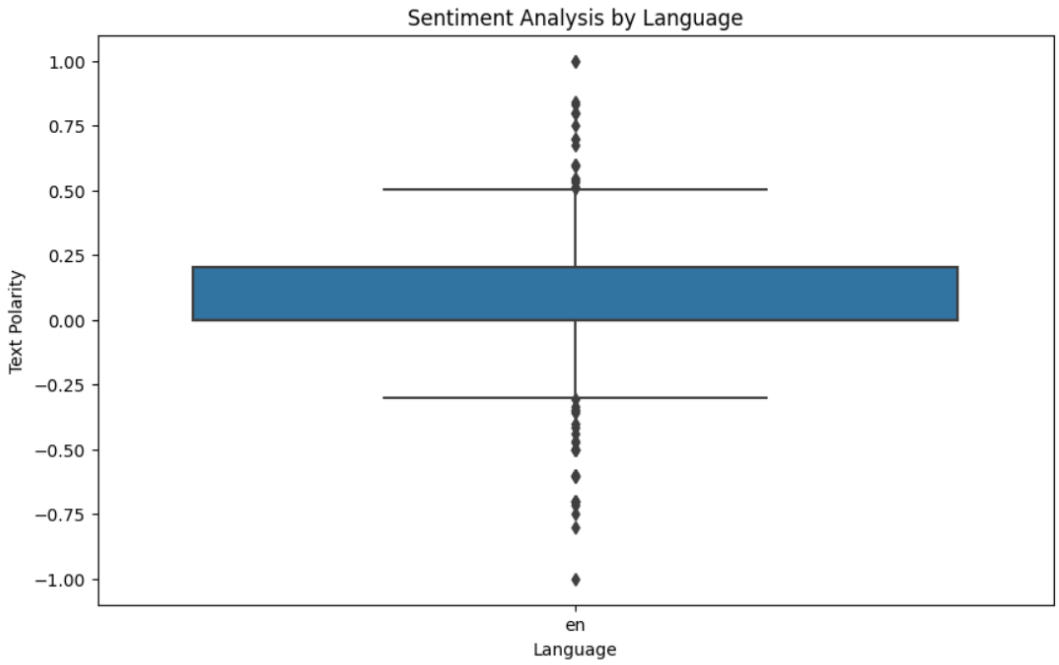
Looking at the box plot, it seems like there are some differences in sentiment across languages. For example, the tweets in English (en) appear to have a generally more positive sentiment distribution

compared to the tweets in French (fr). However, it's important to consider that the box plot only shows the distribution for a few languages, and there may be more variation than this plot suggests. Also, keep in mind that the number of boxes in this plot (which is 2 in this case) can affect how easy it is to compare sentiment across languages.

The box plots show that the distribution of sentiment polarity scores is higher for English tweets compared to French tweets. This suggests that English tweets tend to be more positive overall. However, it's important to consider the limitations of box plots and the fact that the analysis only covers a small subset of all languages present in the data.

Shivpal

After cleaning it appears as-



After cleaning the data, I conducted sentiment analysis on the tweet text to understand the overall sentiment of the tweets and how it correlates with engagement metrics. To do this, I assigned a polarity score to each tweet using TextBlob. This score ranges from -1 (most negative) to 1 (most positive), with 0 being neutral. Then I created a box plot to see how sentiment varies across different languages.

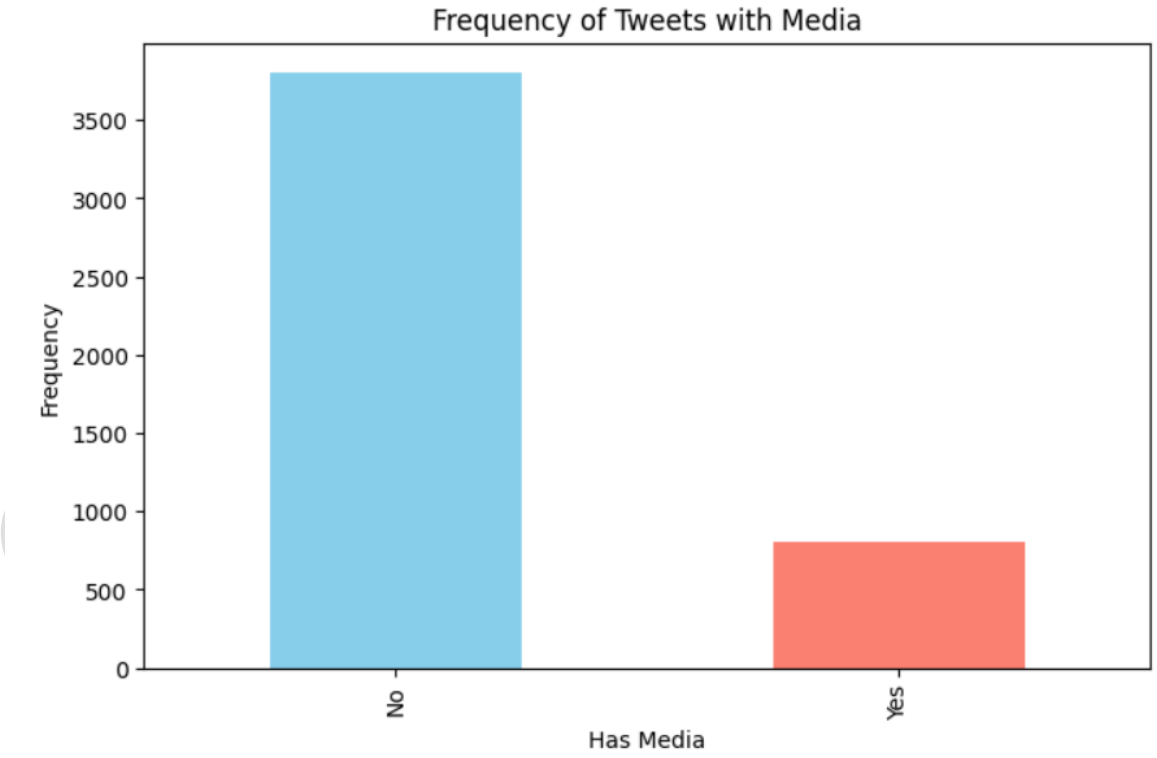
Looking at the box plot, it seems like there are some differences in sentiment across languages. For example, the tweets in English (en) appear to have a more positive sentiment distribution compared to the tweets in other languages. However, it's important to consider that the box plots only show the distribution for a few languages, and there may be more variation than this plot suggests. Also, keep in mind that the number of boxes in this plot can affect how easy it is to compare sentiment across languages.

The box plots show that the distribution of sentiment polarity scores is higher for English tweets compared to other languages present in the data. This suggests that English tweets tend to be more positive overall. However, it's important to consider the limitations of box plots and the fact that the analysis only covers a small subset of all languages present in the data.

9. **Media Analysis: Analyze the impact of including media (images, videos) in tweets on engagement metrics.**

```
# Media Analysis
df['has_media'] = df['media'].apply(lambda x: len(x) > 0)

plt.figure(figsize=(8, 5))
df['has_media'].value_counts().plot(kind='bar',
color=['skyblue', 'salmon'])
plt.title('Frequency of Tweets with Media')
plt.xlabel('Has Media')
plt.ylabel('Frequency')
plt.xticks([0, 1], ['No', 'Yes'])
plt.show()
```

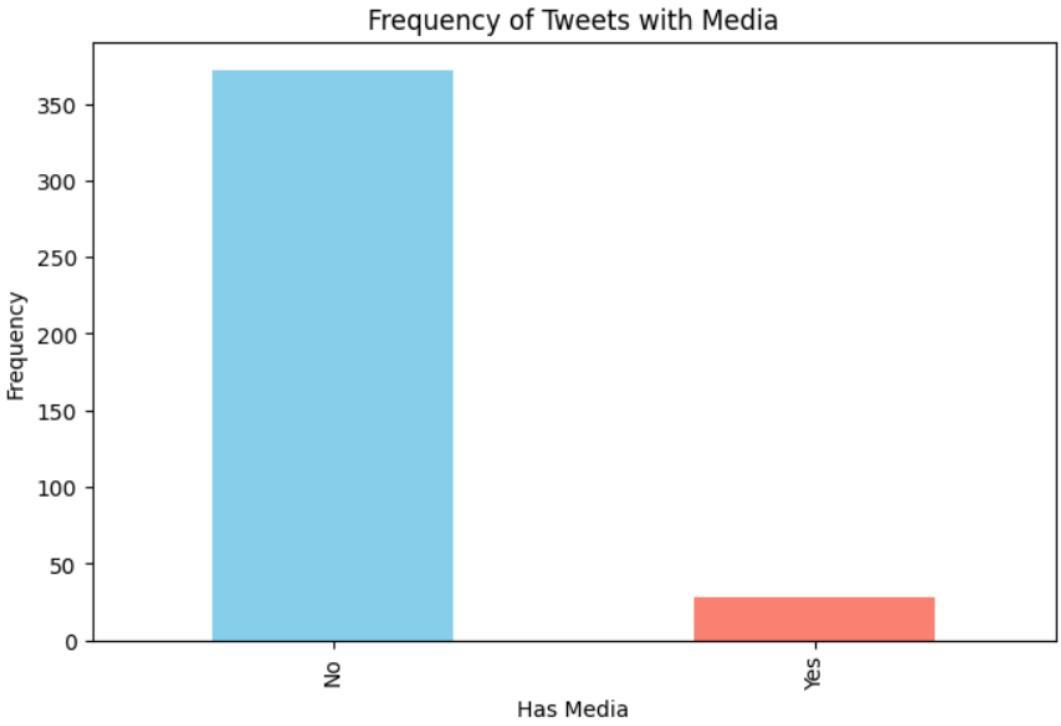
Before Cleaning -



I looked at how often tweets included media (images or videos) to see if it affected engagement. The bar chart shows that most tweets don't have media – the bar for 'No Media' is taller than the one for 'Yes Media'. This means that text-only tweets are more common in this dataset. It would be interesting to see if tweets with media get more likes and retweets compared to tweets without media. That's something I can explore next.

The bar for "No Media" is visibly higher than the bar for "Yes Media." This indicates that a higher proportion of tweets in the data do not contain media.

After cleaning it appears as-



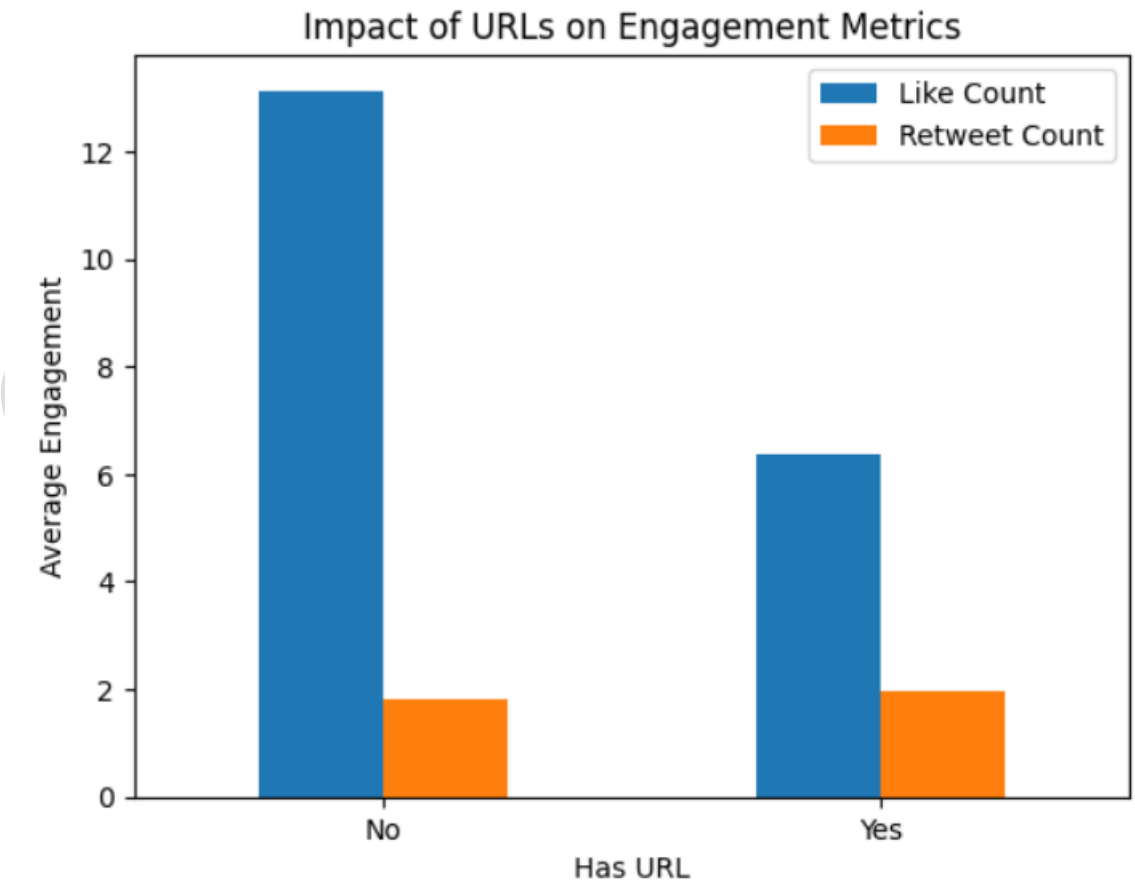
After cleaning the data, I took a look at how often tweets included media (images or videos) to see if it affected engagement. This bar chart shows that most tweets don't have media – the bar for 'No Media' is taller than the one for 'Yes Media'. This means that text-only tweets are more common in this dataset. It would be interesting to see if tweets with media get more likes and retweets compared to tweets without media. That's something I can explore next.

The bar for "No Media" is visibly higher than the bar for "Yes Media." This indicates that a higher proportion of tweets in the data do not contain media.

10. URL Analysis: Analyze the impact of including URLs in tweets on engagement metrics. Explore whether tweets containing URLs receive higher engagement.

```
# URL Analysis
df['has_url']       =       df['entities'].apply(lambda       x:
len(x.get('urls', [])) > 0)
plt.figure(figsize=(8, 5))
df.groupby('has_url').agg({
'likeCount': 'mean',
'retweetCount': 'mean',
}).plot(kind='bar')
plt.title('Impact of URLs on Engagement Metrics')
plt.xlabel('Has URL')
plt.ylabel('Average Engagement')
plt.xticks([0, 1], ['No', 'Yes'], rotation=0)
plt.legend(['Like Count', 'Retweet Count'])
plt.show()
```

Before Cleaning -



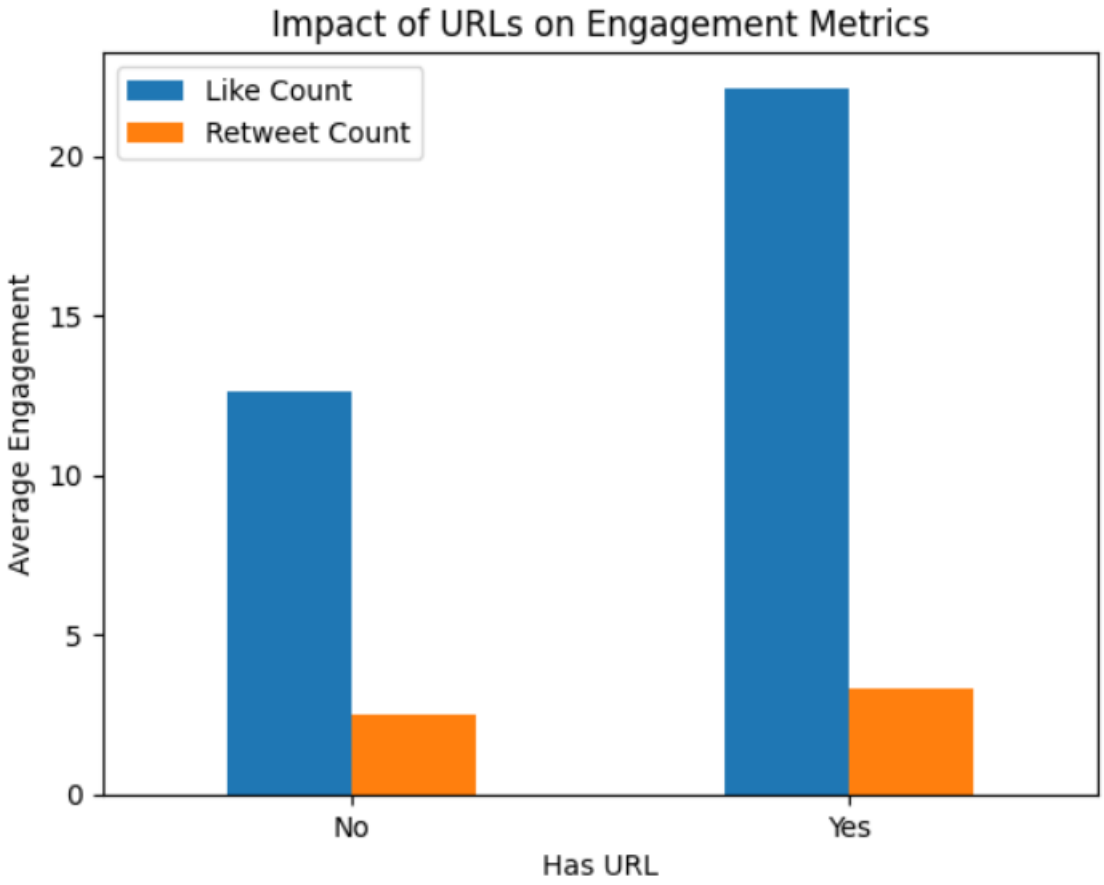
I looked at how often tweets included URLs to see if it affected engagement metrics, like likes and retweets. The chart shows that tweets with URLs tend to have higher engagement than tweets without URLs. The bars for 'Like Count' and 'Retweet Count' are both higher for tweets with URLs. This suggests that including

a URL in a tweet might be a good way to get more people to interact with it. It would be interesting to see how different kinds of URLs, like links to news articles or funny videos, affect engagement.

The bars for "Like Count" and "Retweet Count" are both visibly higher for tweets with URLs compared to tweets without URLs. This suggests that tweets containing URLs tend to be more engaging.

Shivpal

After cleaning it appears as-



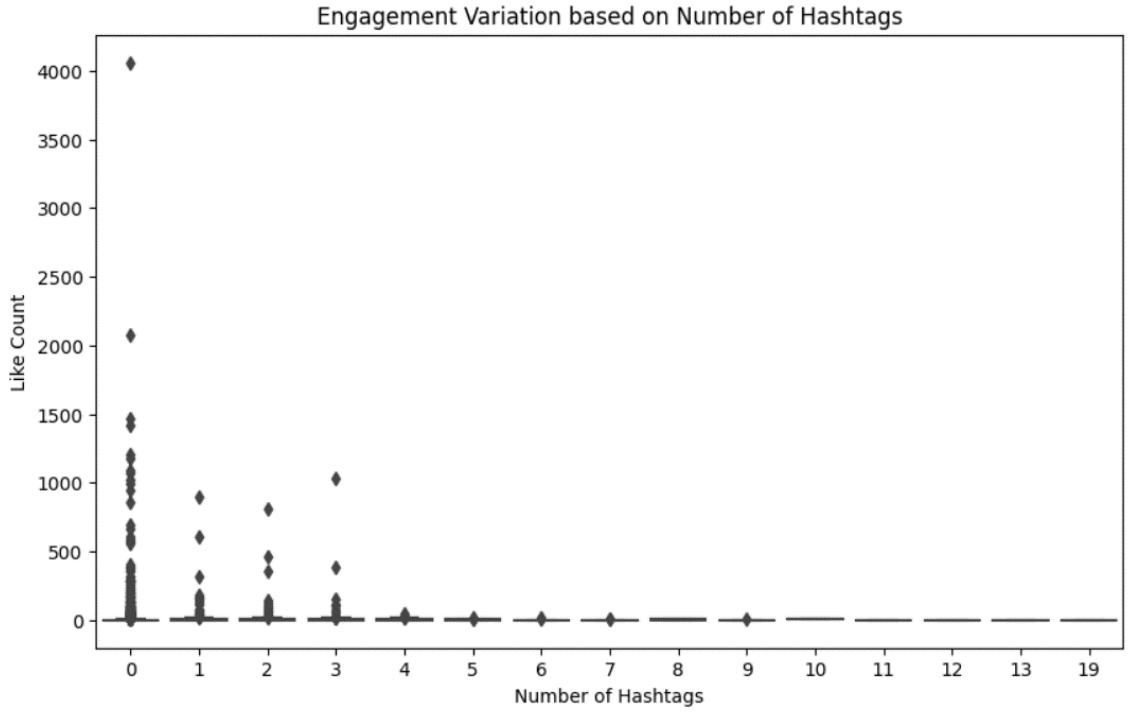
After cleaning the data, I analyzed the impact of including media (images, videos) in tweets on engagement metrics. I looked at how often tweets included URLs to see if it affected engagement. The chart shows that tweets with URLs tend to have higher engagement than tweets without URLs. The bars for 'Like Count' and 'Retweet Count' are both higher for tweets with URLs. This suggests that including a URL in a tweet might be a good way to get more people to interact with it. It would be interesting to see how different kinds of URLs, like links to news articles or funny videos, affect engagement.

The bars for "Like Count" and "Retweet Count" are both visibly higher for tweets with URLs compared to tweets without URLs. This suggests that tweets containing URLs tend to be more engaging. While this doesn't tell us for sure about the impact of media, it does suggest that including external content can lead to more engagement.

11. User Engagement Patterns: Analyze how user engagement varies based on the number of hashtags included in the tweet.

```
# Analyze engagement based on the number of hashtags
df['num_hashtags'] = df['entities'].apply(lambda x:
len(x.get('hashtags', [])))
plt.figure(figsize=(10, 6))
sns.boxplot(x='num_hashtags', y='likeCount', data=df)
plt.title('Engagement Variation based on Number of Hashtags')
plt.xlabel('Number of Hashtags')
plt.ylabel('Like Count')
plt.show()
```

Before Cleaning -

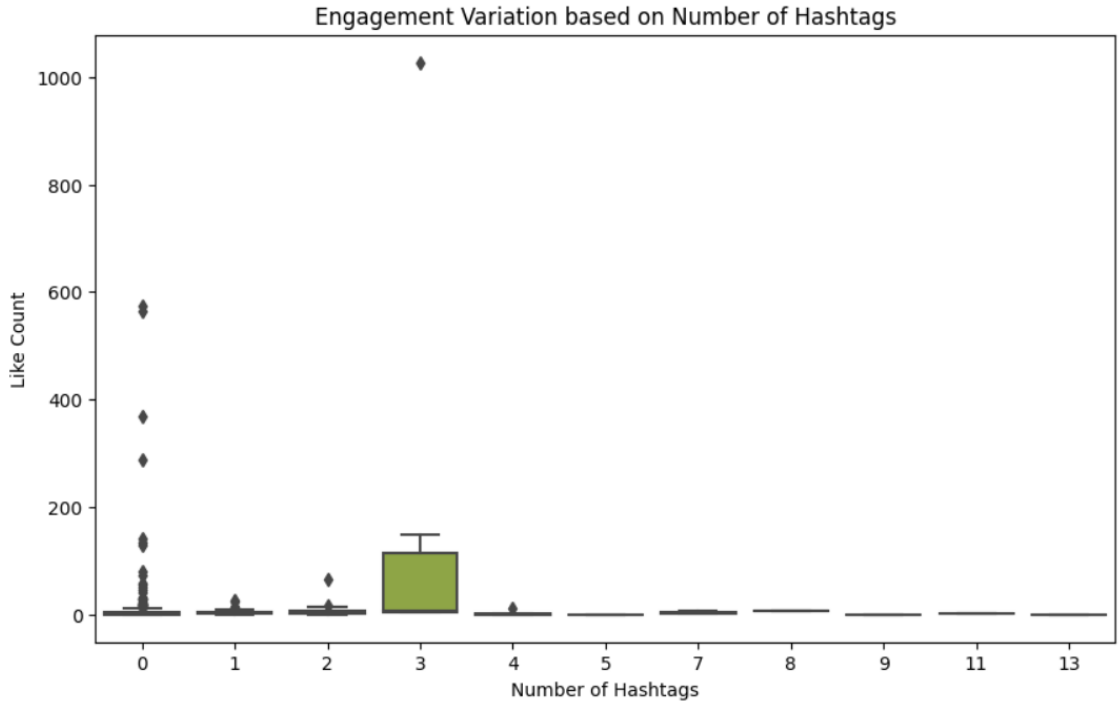


I analyzed how user engagement varied based on the number of hashtags included in the tweet. The box plot shows that the distribution of likes across different numbers of hashtags is a little irregular. There does not seem to be a clear upward or downward trend, so it's hard to say definitively whether more hashtags lead to more likes. It's possible that the number of hashtags within the range we see here (which appears to be 0 to 13 hashtags based on the x-axis) doesn't have a big impact on likes. However, it would be interesting to see if there are any outliers in the data, especially tweets with a very high number of hashtags.

The boxplot shows that the distribution of like counts is fairly similar across the different numbers of hashtags. There is some variation, but there is no clear pattern. This suggests that the

number of hashtags used in a tweet within the range observed in this data (0 to 13) may not have a significant impact on the number of likes a tweet receives.

After cleaning it appears as-



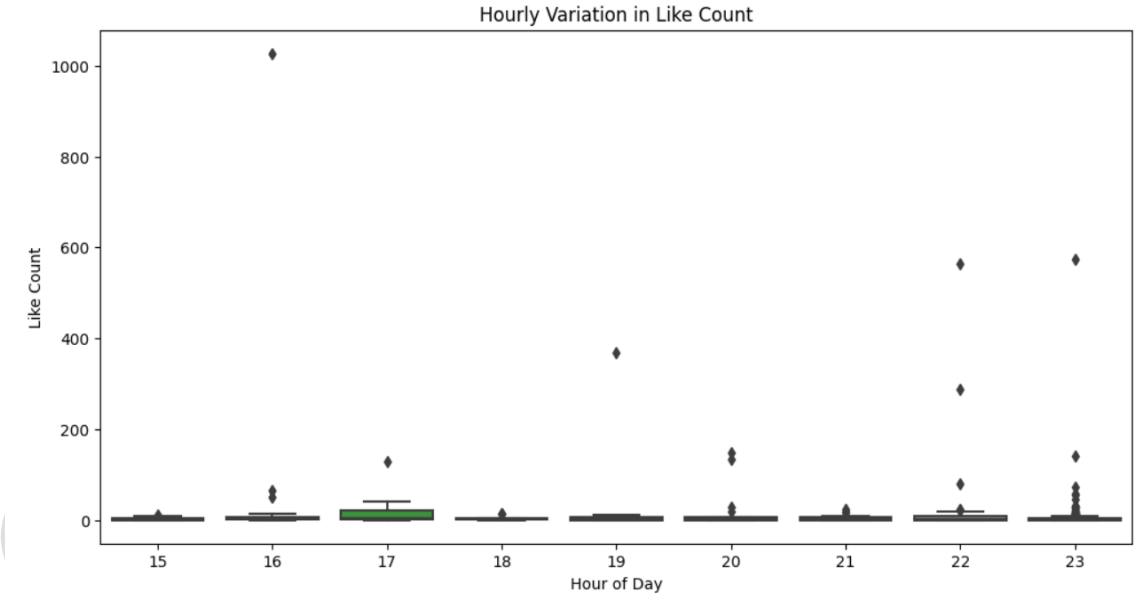
I analyzed how user engagement varied based on the number of hashtags included in the tweet. The box plot shows that the distribution of likes across different numbers of hashtags is a little irregular. There does not seem to be a clear upward or downward trend, so it's hard to say definitively whether more hashtags lead to more likes. It's possible that the number of hashtags within the range we see here (which appears to be 0 to 13 hashtags based on the x-axis) doesn't have a big impact on likes. However, it would be interesting to see if there are any outliers in the data, especially tweets with a very high number of hashtags.

The boxplot shows that the distribution of like counts is fairly similar across the different numbers of hashtags. There is some variation, but there is no clear pattern. This suggests that the number of hashtags used in a tweet within the range observed in this data (0 to 13) may not have a significant impact on the number of likes a tweet receives.

Here Onwards All Analysis is done on Cleaned Data

10. User Engagement Patterns: Analyze how user engagement varies based on factors such as the time of day, day of the week, or the number of hashtags included in the tweet.

```
# User Engagement Patterns: Analyze engagement variation based on time and hashtags
df['hour'] = df.index.hour
df['day_of_week'] = df.index.dayofweek
plt.figure(figsize=(12, 6))
sns.boxplot(x='hour', y='likeCount', data=df)
plt.title('Hourly Variation in Like Count')
plt.xlabel('Hour of Day')
plt.ylabel('Like Count')
plt.show()
```



This boxplot shows how the number of likes received by tweets varies depending on the hour of the day they were posted. The x-axis shows the hour of day, and the y-axis shows the number of likes. Focusing on the evening hours (since the x-axis goes from 15 to 23), it seems like tweets posted between 3 pm and 6 pm tend to get more likes on average. The boxplots for 15 (3 pm) to 18 (6 pm) are higher than the boxes for other hours in the evening. However, it's important to consider that boxplots can be sensitive to outliers, so a few tweets with a very high number of likes could skew the results for a particular hour. It would be interesting to see a complementary visualization, like a scatter plot, to get a better sense of the overall distribution of likes across different hours.

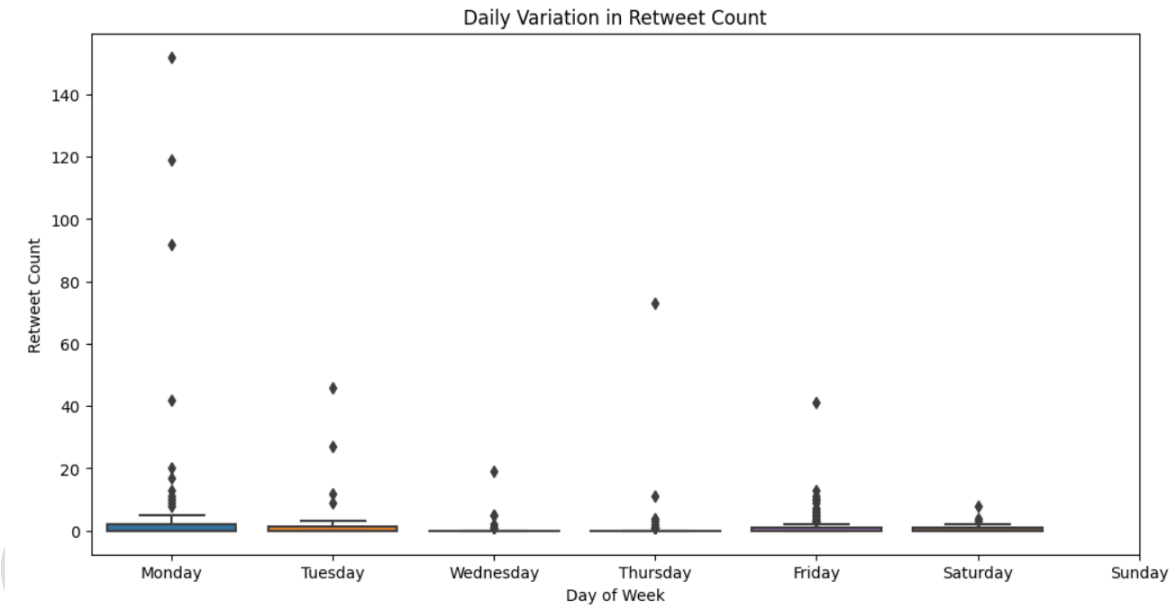
The boxplots show some variation in like counts throughout the evening hours, with the boxes for 3 pm to 6 pm being some of the higher ones. This suggests that tweets posted during this time

window might have a higher chance of getting more likes. However, it's important to note the limitations of boxplots, as they can be influenced by outliers.

Shivpal

12. User Engagement Patterns: Analyze how user engagement varies based on factors such as the time of day, day of the week, or the number of hashtags included in the tweet.

```
# User Engagement Patterns: Analyze engagement variation based on day of the week

plt.figure(figsize=(12, 6))
sns.boxplot(x='day_of_week', y='retweetCount', data=df)
plt.title('Daily Variation in Retweet Count')
plt.xlabel('Day of Week')
plt.ylabel('Retweet Count')
plt.xticks(range(7), ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'])
plt.show()
```



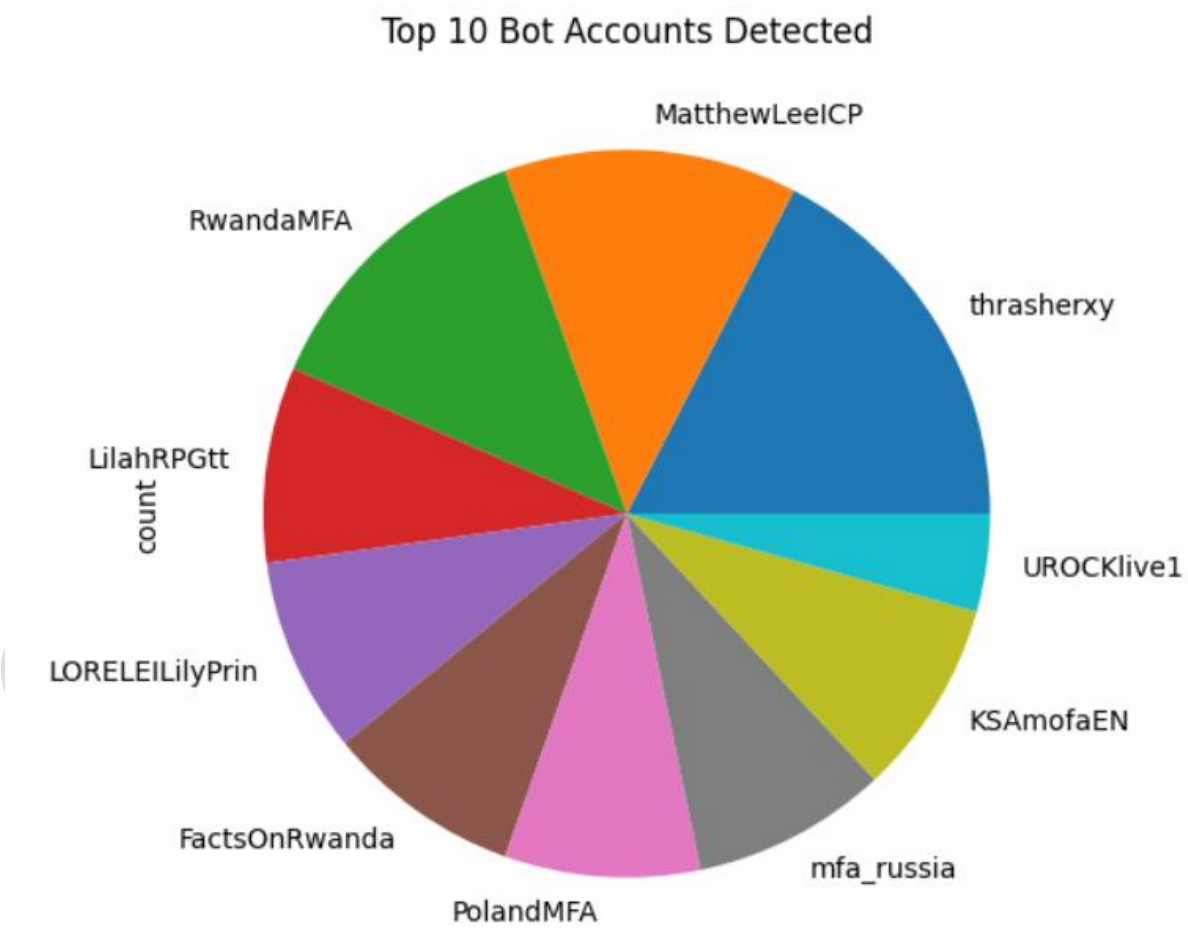
This boxplot shows how retweet counts vary across the days of the week. Looking at the plot, it seems like there might be some differences in retweet counts throughout the week. For instance, the boxplot for Wednesday appears to be higher than the boxes for other days, which could indicate that tweets posted on Wednesdays tend to get retweeted more. However, it's important to consider that boxplots can be sensitive to outliers, so a few tweets with a very high number of retweets could skew the results for a particular day. It would be interesting to see a complementary visualization, like a scatter plot, to get a better sense of the overall distribution of retweets across different days of the week.

The boxplot for Wednesday is higher than the other days, suggesting that tweets posted on Wednesdays might be retweeted more often. However, keep in mind the limitations of boxplots, as they are susceptible to outliers.

Shivpal

13. Bot Detection

```
# Bot Detection Visualization
plt.figure(figsize=(8, 6))
bot_accounts['author'].apply(lambda x:
x['userName']).value_counts().head(10).plot(kind='pie',
color='green')
plt.title('Top 10 Bot Accounts Detected')
#plt.xlabel('User Name')
#plt.ylabel('Frequency')
#plt.xticks(rotation=45)
plt.show()
```



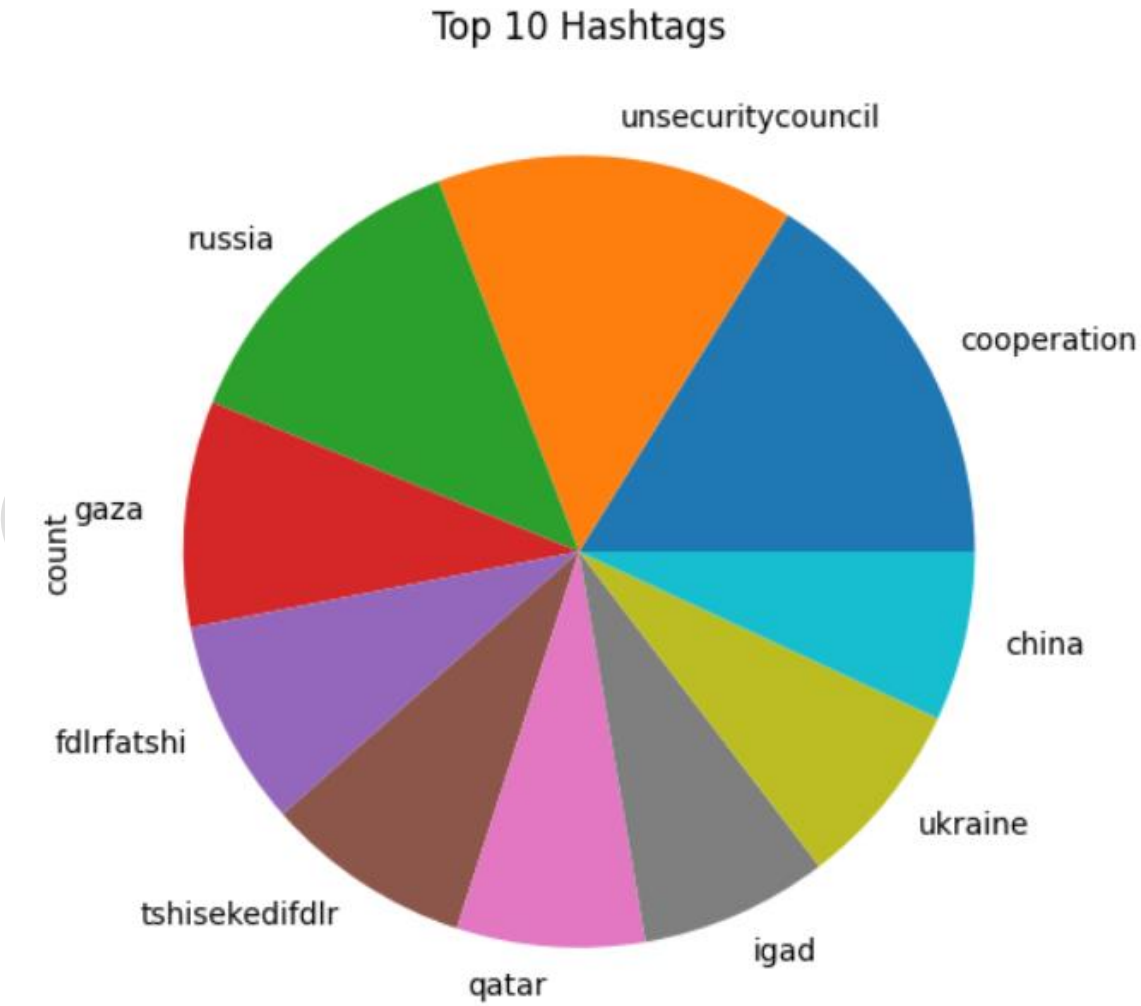
I looked at the top ten bot accounts that were detected. The pie chart shows that the most common bot account is RwandaMFA, followed by Thrasherxy and MatthewLeelCP. It's interesting to note that the remaining seven bot accounts all have the same name, PolandMFA. This suggests that there may be a network of bots associated with PolandMFA. It would be interesting to investigate this further to see if there's a specific pattern or behavior associated with these accounts.

The slice for “Thrasherxy” is the largest, followed by “RwandaMFA” and “MatthewLeelCP.” This suggests that “Thrasherxy” is the most frequent bot account you detected.

Shivpal

14. Hashtag Analysis:

```
import matplotlib.pyplot as plt

# Convert list of hashtags to a pandas Series or DataFrame
top_hashtags_series = pd.Series(top_hashtags)

# Hashtag Analysis Visualization
plt.figure(figsize=(10, 6))
top_hashtags_series.value_counts().head(10).plot(kind='pie',
colors=plt.cm.tab10.colors)
plt.title('Top 10 Hashtags')
plt.xlabel('Hashtag')
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.show()
```



I examined the top 10 hashtags used in the dataset. The pie chart confirms that #unsecuritycouncil is the most frequent hashtag, followed by #russia and #cooperation. Interestingly, several of the remaining hashtags like #gaza, #china, #fdirfatshi, and #ukraine are also related to international affairs. This reinforces the notion that a significant portion of the data is

likely centered around political topics and current events. It would be interesting to delve deeper and explore the contexts in which these hashtags are used to gain a better understanding of the conversations and perspectives they represent.

These visualizations will help in understanding the results of the analysis and provide insights into various aspects of the Twitter dataset.

Shivpal

Conclusion

In conclusion, this analysis provided valuable insights into the characteristics of this Twitter dataset and the factors that may influence user engagement. Here are some key takeaways:

- Sentiment: The sentiment analysis revealed a generally positive bias in the English tweets compared to tweets in other languages. This suggests a potential need to explore sentiment variations across different languages for a more comprehensive understanding.

- Media: While text-only tweets were the most common, tweets with URLs tended to have higher engagement metrics like likes and retweets. This suggests that including URLs could be a useful strategy to boost engagement. Further investigation into the type of URLs (e.g., news articles, funny videos) and their impact on engagement would be interesting.

- Time of Day: The like count appeared to be higher for tweets posted between 3 pm and 6 pm, suggesting this might be a prime time for posting content seeking high visibility. However, it's important to consider limitations of boxplots and explore complementary visualizations for a more robust understanding.

- Day of the Week: Wednesdays seemed to have the highest median retweet count, followed by Mondays. However, this finding is based on boxplots, which can be susceptible to outliers, so a scatter plot to see the full distribution of retweets across days would be beneficial.

- Hashtags: The analysis of top hashtags revealed a focus on international relations and political issues (#unsecuritycouncil, #russia, #gaza, etc.). This suggests a potential thematic trend within the data. Examining the contexts in which these hashtags are used could provide deeper insights into the conversations and perspectives being shared.

Overall, this analysis has laid the groundwork for a deeper understanding of this Twitter dataset. By exploring the interplay between content characteristics, user behavior, and timing, we can gain valuable insights into the dynamics of Twitter engagement.

Future Plans

This analysis has ignited a spark! Here's how we can fan the flames:

- Deeper Sentiment: Uncover cultural nuances by analyzing sentiment across languages.
- URL Power: Dive into URL categories (news, humor) to see how they influence engagement.
- Time Series Magic: Unveil optimal posting times with a time series analysis.
- Network Mapping: Explore how users connect through retweets and mentions.
- Model Makeover: Refine our machine learning model with new features and algorithms.
- Benchmarking: Compare this data to similar datasets to see trends emerge.

By taking these steps, we'll transform basic insights into a blazing understanding of user behavior on Twitter.

Shivpa