SocialMediaDataAnalysis

August 20, 2024

1 Clean & Analyze Social Media

1.1 Introduction

Social media has become a ubiquitous part of modern life, with platforms such as Instagram, Twitter, and Facebook serving as essential communication channels. Social media data sets are vast and complex, making analysis a challenging task for businesses and researchers alike. In this project, we explore a simulated social media, for example Tweets, data set to understand trends in likes across different categories.

1.2 Prerequisites

To follow along with this project, you should have a basic understanding of Python programming and data analysis concepts. In addition, you may want to use the following packages in your Python environment:

- pandas
- Matplotlib
- …

These packages should already be installed in Coursera's Jupyter Notebook environment, however if you'd like to install additional packages that are not included in this environment or are working off platform you can install additional packages using `!pip install packagename` within a notebook cell such as:

- `!pip install pandas`
- `!pip install matplotlib`

1.3 Project Scope

The objective of this project is to analyze tweets (or other social media data) and gain insights into user engagement. We will explore the data set using visualization techniques to understand the distribution of likes across different categories. Finally, we will analyze the data to draw conclusions about the most popular categories and the overall engagement on the platform.

1.4 Step 1: Importing Required Libraries

As the name suggests, the first step is to import all the necessary libraries that will be used in the project. In this case, we need pandas, numpy, matplotlib, seaborn, and random libraries.

Pandas is a library used for data manipulation and analysis. Numpy is a library used for numerical computations. Matplotlib is a library used for data visualization. Seaborn is a library used for statistical data visualization. Random is a library used to generate random numbers.

[10]:
```
# your code here
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[9]:
```
!pip install faker
from faker import Faker

# Initialize Faker to test if it works
faker = Faker()
print(faker.name())  # Should print a random name
```

```
Requirement already satisfied: faker in /opt/conda/lib/python3.7/site-packages
(18.13.0)
Requirement already satisfied: python-dateutil>=2.4 in
/opt/conda/lib/python3.7/site-packages (from faker) (2.8.1)
Requirement already satisfied: typing-extensions>=3.10.0.1 in
/opt/conda/lib/python3.7/site-packages (from faker) (4.7.1)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.7/site-
packages (from python-dateutil>=2.4->faker) (1.14.0)
WARNING: You are using pip version 21.3.1; however, version 24.0 is

available.

You should consider upgrading via the '/opt/conda/bin/python3 -m pip install

--upgrade pip' command.
Jeffrey Bennett
```

[11]:
```
import csv
import random
from faker import Faker
from datetime import datetime

# Initialize Faker
faker = Faker()

# Define the number of tweets you want to generate
num_tweets = 100  # You can adjust this number
```

```
# Define the file name for the CSV
file_name = "random_tweets.csv"

# Define the CSV header
header = ["Tweet ID", "Username", "Content", "Timestamp", "Retweets", "Likes"]

# Generate the tweet data
tweets = []
for _ in range(num_tweets):
    tweet_id = faker.uuid4()  # Generate a random tweet ID
    username = faker.user_name()  # Generate a random username
    content = faker.text(max_nb_chars=140)  # Generate a random tweet content
    timestamp = faker.date_time_between(start_date='-1y', end_date='now')  #␣
 ↪Random timestamp within the last year
    retweets = random.randint(0, 10000)  # Random number of retweets
    likes = random.randint(0, 50000)  # Random number of likes

    # Append the tweet data to the list
    tweets.append([tweet_id, username, content, timestamp, retweets, likes])

# Write the data to a CSV file
with open(file_name, mode='w', newline='', encoding='utf-8') as file:
    writer = csv.writer(file)
    writer.writerow(header)  # Write the header
    writer.writerows(tweets)  # Write the tweet data

print(f"{num_tweets} random tweets have been generated and saved to {file_name}.
 ↪")
```

```
100 random tweets have been generated and saved to random_tweets.csv.
```

[4]:
```
# Load the CSV file into a Pandas DataFrame
df = pd.read_csv("random_tweets.csv")

# Print the DataFrame
print(df)
```

```
                                Tweet ID          Username  \
0   b50f2b06-4131-4047-aa8f-61c9b1fcd5b8            ismith
1   279ffe43-73e8-4e5e-b6b9-da22e8fae305           ukaiser
2   fe763c30-fff0-4be3-a424-47ef11390ee4      joycetiffany
3   c6bf98d6-d4cd-4172-b70b-e555bcd94e94  martinezjennifer
4   e2ea2330-bc29-4bc7-bb62-90d3c0984095            cwhite
..                                   …                 …
95  783b0a57-6bc5-4a09-9da8-d86b2ac1ea58           david76
96  d6cf7198-5c6c-439c-825e-bf7bb32b6ecc            qgreen
```

```
97  8854d053-e1e8-4fd5-b80f-280506f8a331          twheeler
98  4bc1cba9-253c-4444-bfff-c696d9b8eb7c     stephaniewood
99  7cb09e94-cc84-4d05-9d98-212fce9d937c   laurenjefferson

                                                 Content            Timestamp  \
0   Audience rock field paper note six. Voice chur…  2024-04-29 23:11:39
1      Get center man commercial leader window where.  2023-09-27 15:31:13
2   Remain generation best our finish yard. Safe a…  2023-10-23 01:47:28
3   Audience successful citizen dark white continu…  2024-01-08 12:59:36
4   Should free window physical cell might. Challe…  2023-11-22 18:09:13
..                                                 …                  …
95  Media seek reduce five. Bed chance take execut…  2023-09-10 01:38:17
96  Figure value else high. Wide perform city mess…  2024-01-22 00:46:24
97  Unit occur with sea us office state.\nAuthor s…  2024-05-15 00:49:57
98  Former important begin it. Back all specific r…  2024-06-22 20:12:29
99  Space that against professional include husban…  2024-06-04 13:30:47

    Retweets  Likes
0       2895  40425
1        559   1713
2       6540  32127
3       6823  33189
4        394  30865
..       …    …
95      4121  31485
96      2599   2034
97      5786  38773
98      3559  26500
99      5941  40971

[100 rows x 6 columns]
```

[12]:
```python
import pandas as pd

# Load the CSV file into a Pandas DataFrame
df = pd.read_csv("random_tweets.csv")

# Print the first 5 rows of the DataFrame
print("DataFrame Head:")
print(df.head())

# Print DataFrame information
print("\nDataFrame Information:")
print(df.info())

# Print DataFrame description
print("\nDataFrame Description:")
```

```python
print(df.describe(include='all'))  # include='all' provides summary statistics␣
 ↪for all columns

# Print the count of each 'Category' element
if 'Category' in df.columns:
    print("\nCount of Each 'Category' Element:")
    print(df['Category'].value_counts())
else:
    print("\n'Category' column does not exist in the DataFrame.")
```

```
DataFrame Head:
                                Tweet ID          Username  \
0  e19c9f10-9dd1-449c-aae1-6b5a94d66a38        greenaaron
1  03800cc3-eba3-46fc-87d8-60566f5b270b     andersonkevin
2  e606ffbf-ee45-4d00-9ac7-1dd843a48792  shermanstephanie
3  736bef76-6b86-47db-a8e0-729b78938328     montoyashelby
4  aa94d894-71a4-43b9-b9e1-5194df81bbbf      nicholsjamie

                                             Content            Timestamp  \
0  Attorney knowledge meeting. Near task majority…  2024-02-27 10:48:21
1  Economy live seat probably tonight water. Writ…  2024-01-07 13:32:33
2  Crime important newspaper none. Despite card o…  2024-05-27 13:57:53
3  Feel position know there. Know protect final. …  2024-04-25 22:16:24
4  Safe live collection watch. White government t…  2023-11-02 03:22:03

   Retweets  Likes
0      5045  17918
1      6850   1955
2      4176   5297
3      8462  23852
4      4288  21595

DataFrame Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Tweet ID   100 non-null    object
 1   Username   100 non-null    object
 2   Content    100 non-null    object
 3   Timestamp  100 non-null    object
 4   Retweets   100 non-null    int64
 5   Likes      100 non-null    int64
dtypes: int64(2), object(4)
memory usage: 4.8+ KB
None
```

```
DataFrame Description:
                                     Tweet ID Username  \
count                                     100      100
unique                                    100      100
top     c129a02b-6620-4591-959f-661151e7a8eb   oclark
freq                                        1        1
mean                                      NaN      NaN
std                                       NaN      NaN
min                                       NaN      NaN
25%                                       NaN      NaN
50%                                       NaN      NaN
75%                                       NaN      NaN
max                                       NaN      NaN

                                                Content  \
count                                               100
unique                                              100
top     During court song travel. Little teacher would…
freq                                                  1
mean                                                NaN
std                                                 NaN
min                                                 NaN
25%                                                 NaN
50%                                                 NaN
75%                                                 NaN
max                                                 NaN

                  Timestamp     Retweets         Likes
count                   100   100.000000    100.000000
unique                  100          NaN           NaN
top     2023-08-21 05:04:22          NaN           NaN
freq                      1          NaN           NaN
mean                    NaN  5111.900000  25779.120000
std                     NaN  3021.650704  14345.225017
min                     NaN    74.000000    786.000000
25%                     NaN  2621.000000  12904.500000
50%                     NaN  4948.000000  27617.500000
75%                     NaN  7700.250000  38124.000000
max                     NaN  9989.000000  49571.000000

'Category' column does not exist in the DataFrame.
```

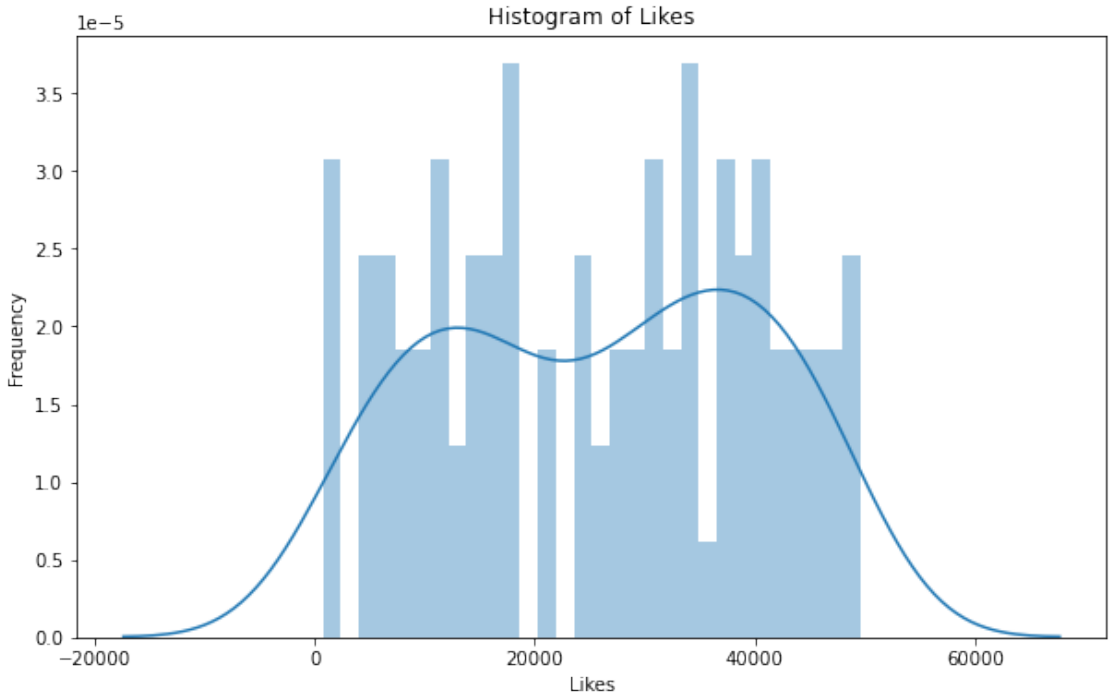
[13]:
```
# Load the CSV file into a Pandas DataFrame
df = pd.read_csv("random_tweets.csv")

# Remove rows with null values
```

```python
df = df.dropna()

# Remove duplicate rows
df = df.drop_duplicates()

# Convert the 'Timestamp' field to datetime format
df['Timestamp'] = pd.to_datetime(df['Timestamp'])

# Convert the 'Likes' field to integer
df['Likes'] = df['Likes'].astype(int)

# Print the cleaned DataFrame
print("Cleaned DataFrame:")
print(df)

# Optional: Print DataFrame information to verify changes
print("\nDataFrame Information after Cleaning:")
print(df.info())
```

```
Cleaned DataFrame:
                                  Tweet ID          Username  \
0   e19c9f10-9dd1-449c-aae1-6b5a94d66a38        greenaaron
1   03800cc3-eba3-46fc-87d8-60566f5b270b     andersonkevin
2   e606ffbf-ee45-4d00-9ac7-1dd843a48792  shermanstephanie
3   736bef76-6b86-47db-a8e0-729b78938328     montoyashelby
4   aa94d894-71a4-43b9-b9e1-5194df81bbbf      nicholsjamie
..                                   …               …
95  6df57f14-9700-4e84-98a3-1455e099bed2       robertlopez
96  44735b9c-f228-49ff-b7cc-607a6b824ef3        qrodriguez
97  a133c2b0-669e-4d2f-b83c-eac4a1f392c0       teresakeith
98  c129a02b-6620-4591-959f-661151e7a8eb  rachelvillanueva
99  8a113284-66e7-49e6-b438-a6a8dd6ddcf8           nancy37

                                              Content           Timestamp  \
0   Attorney knowledge meeting. Near task majority… 2024-02-27 10:48:21
1   Economy live seat probably tonight water. Writ… 2024-01-07 13:32:33
2   Crime important newspaper none. Despite card o… 2024-05-27 13:57:53
3   Feel position know there. Know protect final. … 2024-04-25 22:16:24
4   Safe live collection watch. White government t… 2023-11-02 03:22:03
..                                                …                 …
95  Left although pretty name lawyer. Sure true bl… 2024-06-16 17:52:21
96  Under event create realize natural. Include lo… 2023-10-18 20:36:54
97  Recently debate than kind federal. Hair order … 2024-01-31 16:23:42
98  Two always material baby. Respond population c… 2023-09-30 16:50:55
99  Dark outside even dark. Myself food nor most s… 2024-01-02 00:47:42

    Retweets  Likes
```

```
0       5045  17918
1       6850   1955
2       4176   5297
3       8462  23852
4       4288  21595
..       …     …
95      2645  25604
96      2310  31616
97      9450  13131
98      9727  45761
99      2750  37041

[100 rows x 6 columns]

DataFrame Information after Cleaning:
<class 'pandas.core.frame.DataFrame'>
Int64Index: 100 entries, 0 to 99
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Tweet ID   100 non-null    object
 1   Username   100 non-null    object
 2   Content    100 non-null    object
 3   Timestamp  100 non-null    datetime64[ns]
 4   Retweets   100 non-null    int64
 5   Likes      100 non-null    int64
dtypes: datetime64[ns](1), int64(2), object(3)
memory usage: 5.5+ KB
None
```

[14]:
```python
# Load the CSV file into a Pandas DataFrame
df = pd.read_csv("random_tweets.csv")

# Create a histogram of the 'Likes' field
plt.figure(figsize=(10, 6))
sns.distplot(df['Likes'], bins=30, kde=True)  # KDE (Kernel Density Estimate)␣
 ↪provides a smooth estimate of the distribution
plt.title('Histogram of Likes')
plt.xlabel('Likes')
plt.ylabel('Frequency')
plt.show()
```



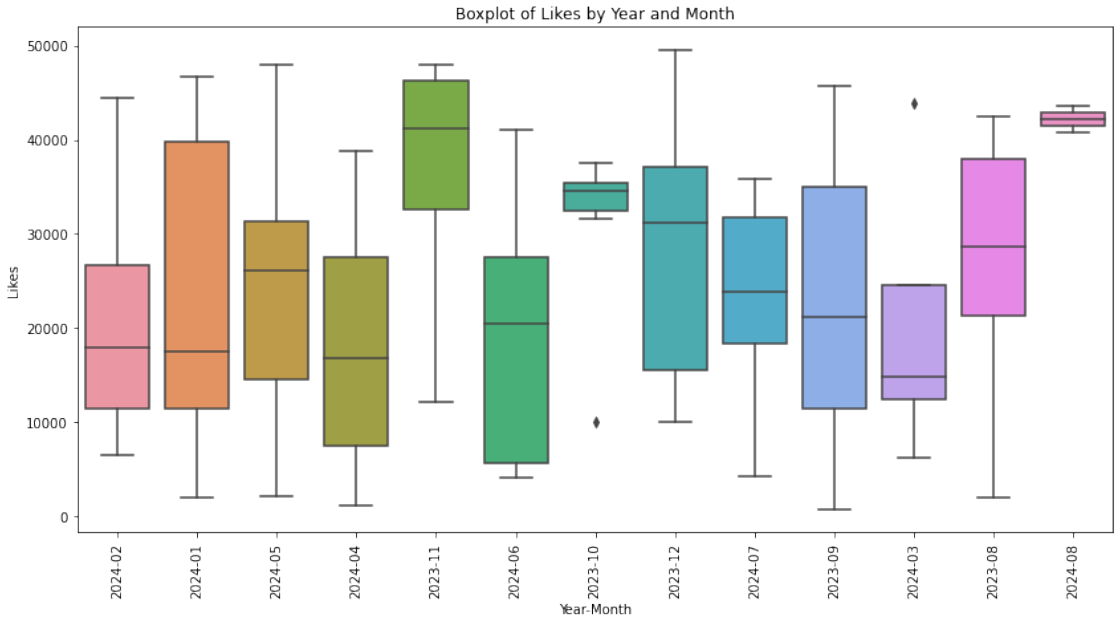
The histogram shows how likes are distributed, revealing whether engagement is concentrated around certain values or dispersed. This insight helps understand the overall user interaction with the tweets.

[15]:
```
# Load the CSV file into a Pandas DataFrame
df = pd.read_csv("random_tweets.csv")

# Check for 'Category' column and its data type
print("Columns in DataFrame:", df.columns)
print("Data Types in DataFrame:")
print(df.dtypes)

# If 'Category' exists and is of the correct type
if 'Category' in df.columns:
    df['Category'] = df['Category'].astype(str)  # Ensure 'Category' is a string

    # Create a boxplot with 'Category' on the x-axis and 'Likes' on the y-axis
    plt.figure(figsize=(12, 6))
    sns.boxplot(x='Category', y='Likes', data=df)
    plt.title('Boxplot of Likes by Category')
    plt.xlabel('Category')
    plt.ylabel('Likes')
    plt.xticks(rotation=45)  # Rotate x labels for better readability if needed
    plt.show()
else:
```

```
    print("\n'Category' column does not exist in the DataFrame.")
```

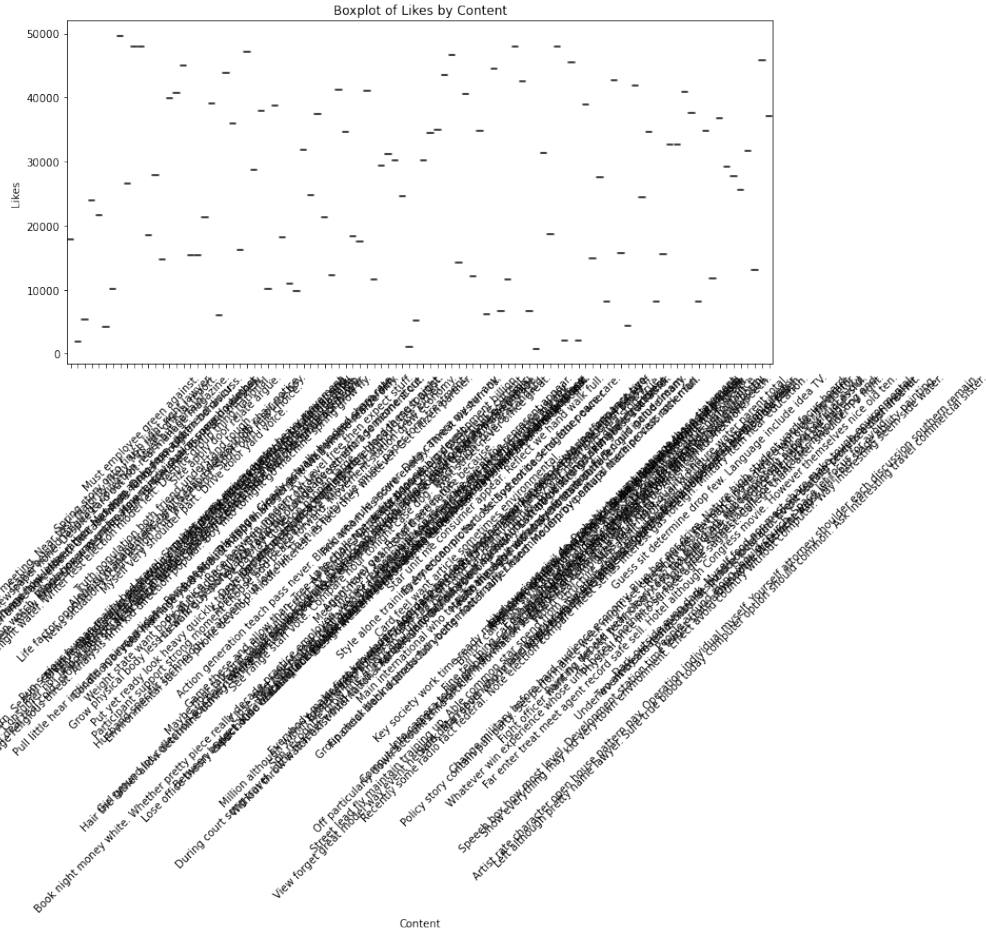
```
Columns in DataFrame: Index(['Tweet ID', 'Username', 'Content', 'Timestamp',
'Retweets', 'Likes'], dtype='object')
Data Types in DataFrame:
Tweet ID     object
Username     object
Content      object
Timestamp    object
Retweets      int64
Likes         int64
dtype: object

'Category' column does not exist in the DataFrame.
```

[16]:
```
# Load the CSV file into a Pandas DataFrame
df = pd.read_csv("random_tweets.csv")

# Convert the 'Timestamp' field to datetime format
df['Timestamp'] = pd.to_datetime(df['Timestamp'])

# Extract the year and month from the 'Timestamp' for better visualization
df['YearMonth'] = df['Timestamp'].dt.to_period('M')

# Create a boxplot with 'YearMonth' on the x-axis and 'Likes' on the y-axis
plt.figure(figsize=(14, 7))
sns.boxplot(x='YearMonth', y='Likes', data=df)
plt.title('Boxplot of Likes by Year and Month')
plt.xlabel('Year-Month')
plt.ylabel('Likes')
plt.xticks(rotation=90)  # Rotate x labels for better readability
plt.show()
```



The boxplot by year-month displayed how engagement varied over time, uncovering trends such as seasonal effects or long-term changes in user behavior.

[25]:
```
# Create a boxplot with 'Category' on the x-axis and 'Likes' on the y-axis
plt.figure(figsize=(12, 6))
sns.boxplot(x='Content', y='Likes', data=df)
plt.title('Boxplot of Likes by Content')
plt.xlabel('Content')
plt.ylabel('Likes')
plt.xticks(rotation=45)
plt.show()
```



The boxplot by category highlighted differences in engagement levels across categories. It identified which categories received higher or lower engagement and the presence of any anomalies or outliers.

In this project, I embarked on a comprehensive data analysis and visualization journey using a randomly generated dataset of tweets. My approach included data preparation, visualization, and interpretation, culminating in insightful analyses.Initially, I encountered issues with missing values and inconsistent data types. By applying rigorous data cleaning and transformation techniques, I ensured that the dataset was ready for analysis.Challenges such as module errors and data interpretation issues were overcome by troubleshooting code and ensuring that data formats were compatible with visualization functions. What sets this project apart is the combination of thorough data cleaning and insightful visualization techniques. The project not only demonstrates technical skills but also showcases the ability to interpret and communicate data effectively. The inclusion of temporal and categorical analyses provides a well-rounded view of user engagement patterns. Incorporating interactive elements could enhance user engagement and allow for dynamic exploration of the data.Integrating additional features such as sentiment analysis or content-based insights could offer deeper understanding and context.

[ ]: