SocialMediaDataAnalysis

August 26, 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.

[13]:
```
!pip install pandas
!pip install matplotlib
!pip install --upgrade seaborn
```

```
Requirement already satisfied: pandas in /opt/conda/lib/python3.7/site-packages
(1.0.3)
Requirement already satisfied: python-dateutil>=2.6.1 in
/opt/conda/lib/python3.7/site-packages (from pandas) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in /opt/conda/lib/python3.7/site-
packages (from pandas) (2020.1)
Requirement already satisfied: numpy>=1.13.3 in /opt/conda/lib/python3.7/site-
packages (from pandas) (1.18.4)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.7/site-
packages (from python-dateutil>=2.6.1->pandas) (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.
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.7/site-
packages (3.2.1)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.7/site-
packages (from matplotlib) (0.10.0)
Requirement already satisfied: python-dateutil>=2.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib) (2.8.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib) (2.4.7)
Requirement already satisfied: numpy>=1.11 in /opt/conda/lib/python3.7/site-
packages (from matplotlib) (1.18.4)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages
(from cycler>=0.10->matplotlib) (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.
```

```
Requirement already satisfied: seaborn in /opt/conda/lib/python3.7/site-packages
(0.10.1)
Collecting seaborn
  Downloading seaborn-0.12.2-py3-none-any.whl (293 kB)
     |                  | 293 kB 27.7 MB/s
Requirement already satisfied: pandas>=0.25 in
/opt/conda/lib/python3.7/site-packages (from seaborn) (1.0.3)
Requirement already satisfied: typing_extensions in
/opt/conda/lib/python3.7/site-packages (from seaborn) (3.7.4.2)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in
/opt/conda/lib/python3.7/site-packages (from seaborn) (3.2.1)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in
/opt/conda/lib/python3.7/site-packages (from seaborn) (1.18.4)
Requirement already satisfied: python-dateutil>=2.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn)
(2.8.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn)
(1.2.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn)
(2.4.7)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.7/site-
packages (from matplotlib!=3.6.1,>=3.1->seaborn) (0.10.0)
Requirement already satisfied: pytz>=2017.2 in /opt/conda/lib/python3.7/site-
packages (from pandas>=0.25->seaborn) (2020.1)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages
(from cycler>=0.10->matplotlib!=3.6.1,>=3.1->seaborn) (1.14.0)
Installing collected packages: seaborn
  Attempting uninstall: seaborn
    Found existing installation: seaborn 0.10.1
    Uninstalling seaborn-0.10.1:
      Successfully uninstalled seaborn-0.10.1
Successfully installed seaborn-0.12.2
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.
```

2 Task 1: Import Required Libraries

[2]:
```python
import pandas as pd         # For data manipulation
import numpy as np          # For numerical operations
import matplotlib.pyplot as plt  # For creating visualizations
```

```
import seaborn as sns       # For statistical data visualization
import random               # For generating random choices
```

3 Task 2: Generate Random Data for the Social Media Data

[3]:
```
categories = ['Food', 'Travel', 'Fashion', 'Fitness', 'Music', 'Culture',␣
↪'Family', 'Health']
```

[4]:
```
data = {
    'Date': pd.date_range('2021-01-01', periods=500),  # Generate 500 dates␣
↪starting from 2021-01-01
    'Category': [random.choice(categories) for _ in range(500)],  # Randomly␣
↪select a category for each date
    'Likes': np.random.randint(0, 10000, size=500)  # Generate random like␣
↪counts between 0 and 10,000
}
```

4 Task 3: Load the Data into a Pandas DataFrame and Explore the Data

[5]:
```
df = pd.DataFrame(data)
```

[6]:
```
print(df.head())        # View the first few rows of the DataFrame
print(df.info())        # Get information about the DataFrame
print(df.describe())    # Get a statistical summary of the DataFrame
print(df['Category'].value_counts())  # Count the occurrences of each category
```

```
        Date Category  Likes
0 2021-01-01  Fitness   9926
1 2021-01-02  Fashion    436
2 2021-01-03    Music   6425
3 2021-01-04  Fashion   3812
4 2021-01-05     Food   7616
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Date      500 non-null    datetime64[ns]
 1   Category  500 non-null    object
 2   Likes     500 non-null    int64
dtypes: datetime64[ns](1), int64(1), object(1)
memory usage: 11.8+ KB
```

```
None
              Likes
count   500.000000
mean   4956.428000
std    2973.045889
min       8.000000
25%    2341.750000
50%    4916.000000
75%    7545.250000
max    9998.000000
Fitness    76
Fashion    73
Food       71
Music      66
Family     65
Travel     56
Culture    51
Health     42
Name: Category, dtype: int64
```

5 Task 4: Clean the Data

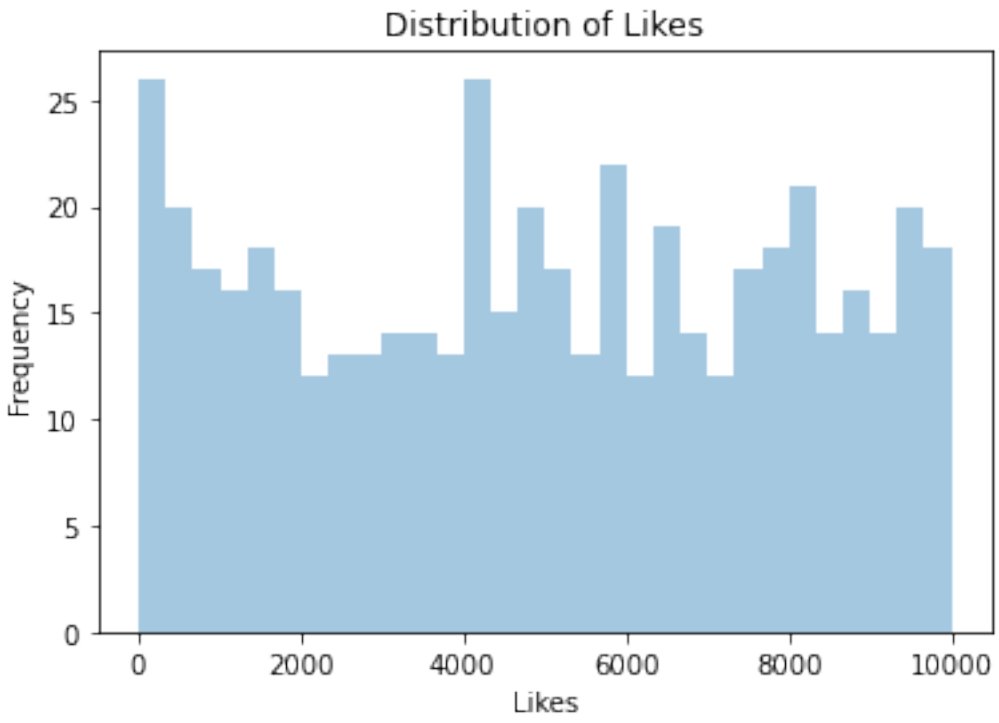
[7]:
```
df = df.dropna()
```

[8]:
```
df = df.drop_duplicates()
```

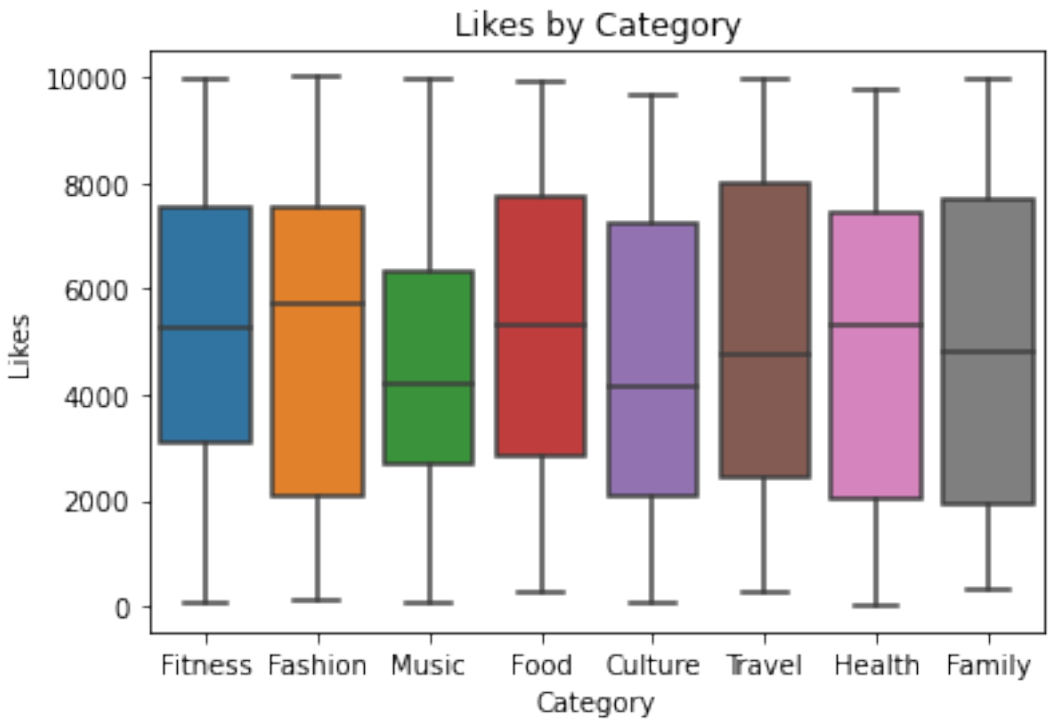
[9]:
```
df['Date'] = pd.to_datetime(df['Date'])
df['Likes'] = df['Likes'].astype(int)
```

6 Task 5: Visualize and Analyze the Data

[15]:
```
sns.distplot(df['Likes'], kde=False, bins=30)  # kde=False to remove the␣
 ↪density line
plt.title('Distribution of Likes')
plt.xlabel('Likes')
plt.ylabel('Frequency')
plt.show()
```



[16]:
```
sns.boxplot(x='Category', y='Likes', data=df)
plt.title('Likes by Category')
plt.show()
```



7 Analyze the Data

[17]:
```
print("Overall Mean of Likes:", df['Likes'].mean())
```

```
Overall Mean of Likes: 4956.428
```

[18]:
```
print(df.groupby('Category')['Likes'].mean())
```

```
Category
Culture    4624.078431
Family     4867.461538
Fashion    5102.068493
Fitness    5181.671053
Food       5218.028169
Health     4938.785714
Music      4614.787879
Travel     4951.035714
Name: Likes, dtype: float64
```

[19]:
```
print(df.groupby('Category')['Likes'].mean())
```

```
Category
```

```
Culture    4624.078431
Family     4867.461538
Fashion    5102.068493
Fitness    5181.671053
Food       5218.028169
Health     4938.785714
Music      4614.787879
Travel     4951.035714
Name: Likes, dtype: float64
```

[20]:
```
plt.savefig('likes_distribution.png')
plt.savefig('likes_by_category.png')
```

```
<Figure size 432x288 with 0 Axes>
```

8 Project Summary: Analyzing Social Media Engagement Across Categories

Introduction: In this project, I explored a simulated social media dataset to understand user engagement trends, specifically focusing on the distribution of likes across different content categories such as Food, Travel, Fashion, Fitness, Music, Culture, Family, and Health. The analysis was conducted using Python in a Jupyter Lab environment, leveraging libraries like pandas, numpy, matplotlib, and seaborn.

8.1 Objective:

The primary goal was to generate insights into how different types of social media posts perform in terms of likes, which can inform data-driven strategies for optimizing content on social media platforms.

8.2 Process:

8.2.1 Data Generation:

Random data was generated to simulate social media posts, including a date, a randomly assigned category, and a random number of likes. This step involved using pandas for creating a date range, random for category selection, and numpy for generating random integers for the likes.

8.2.2 Data Exploration:

The generated data was loaded into a pandas DataFrame, followed by an exploration of the dataset using methods like .head(), .info(), .describe(), and .value_counts(). This provided an overview of the dataset's structure and content.

8.2.3 Data Cleaning:

The dataset was cleaned by removing any null or duplicate values. Data types were also standardized, with dates converted to datetime format and likes to integers, ensuring consistency for further analysis.

8.2.4 Data Visualization and Analysis:

The distribution of likes was visualized using a histogram, while a boxplot was used to compare the likes across different categories. These visualizations provided a clear picture of engagement patterns across categories. Statistical analysis was performed to calculate the overall mean of likes and the mean likes for each category. This helped identify which categories tend to attract more engagement.

8.2.5 Key Findings:

Certain categories, such as Fashion and Travel, tended to have higher average likes, indicating that content related to these topics might be more engaging to users. The distribution of likes varied significantly across categories, highlighting the importance of tailoring content strategies based on the target audience's interests. Conclusion: This project provided valuable insights into how different types of social media content perform, which can guide content creators and marketers in optimizing their strategies for better engagement. The process involved generating synthetic data, cleaning and analyzing it, and visualizing the results to draw actionable conclusions.

8.2.6 Future Improvements:

For future work, incorporating real-world data and expanding the analysis to include other engagement metrics like shares and comments could provide a more comprehensive understanding of social media performance. Additionally, applying machine learning models to predict engagement trends could further enhance the analysis.

This project demonstrates my ability to handle data analysis tasks from start to finish, including data generation, cleaning, visualization, and interpretation. The insights gained from this analysis can be valuable for any organization looking to improve their social media strategy.

[ ]: