SocialMediaDataAnalysis

September 12, 2025

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.

[2]:
```
# Import libraries.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import random
```

[ ]:
```
Task 2 - Generate random data for the social media data
```

[3]:
```
# Define list of categories.
# Generate dictionary with fields 'Date, category, number of likes' with random␣
 ↪data.

categories = ['Food', 'Travel', 'Fashion', 'Fitness', 'Music', 'Culture',␣
 ↪'Family', 'Health']
data = {'Date': pd.date_range('2021-01-01', periods=500),
       'Category': [random.choice(categories)for _ in range(500)],
       'Likes': np.random.randint(0, 10000, size=500)}
```

[ ]:
```
Task 3 - Load the data into a Pandas DataFrame, explore, and clean the data.
```

[4]:
```
# Load the data into a dataframe.

df = pd.DataFrame(data)
```

[5]:
```
# Print dataframe head.

df.head(10)
```

[5]:

	Date	Category	Likes
0	2021-01-01	Music	5112
1	2021-01-02	Music	8913
2	2021-01-03	Family	95
3	2021-01-04	Food	2809
4	2021-01-05	Travel	1056
5	2021-01-06	Family	8101
6	2021-01-07	Food	2936
7	2021-01-08	Health	5962
8	2021-01-09	Health	7924

```
9 2021-01-10  Fashion   9073
```

[6]:
```
# Check for size of the data.

df.shape
```

[6]:
```
(500, 3)
```

[7]:
```
df.info()
```

```
<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
```

[8]:
```
# Change 'Date' dtype to datetime format.

df['Date'] = pd.to_datetime(df['Date'])
```

[8]:
```
# Check to ensure that the 'Date' dtype was changed.
df.info()
```

```
<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
```

[9]:
```
# Print the dataframe description.

df.describe()
```

[9]:
```
              Likes
count   500.000000
mean   5059.998000
std    2864.260119
```

```
min      10.000000
25%    2799.750000
50%    5108.500000
75%    7481.750000
max    9998.000000
```

[ ]:
```
Based on the descriptive statistics the count value for the Likes is 500.
This means that there are 500 likes measurements represented in the dataset.

The 25th percentile for the Likes column is 2799.75 Which means that 25% of
the data is below 2799.75.

The 75th percentile for the Likes column is 7481. This means that 75% of
the likes values are below 7481
```

[10]:
```
# Pint the count of each 'Category' element

df['Category'].value_counts()
```

[10]:
```
Travel     76
Culture    66
Fitness    64
Health     62
Food       61
Music      61
Family     56
Fashion    54
Name: Category, dtype: int64
```

[ ]:
```
This returns a series object containing the count of
unique values and is sorted in descending order.
Travel Category tweets has the most content with 76 posts.
Followed by Culture with 66. The Category
that publishes the least number of content is Fashion with 66 posts.
```

[10]:
```
df['Likes'] = df['Likes'].astype(int)
```

[11]:
```
# Remove duplicates.
df.drop_duplicates(inplace = True)
```

[12]:
```
print(df)
```

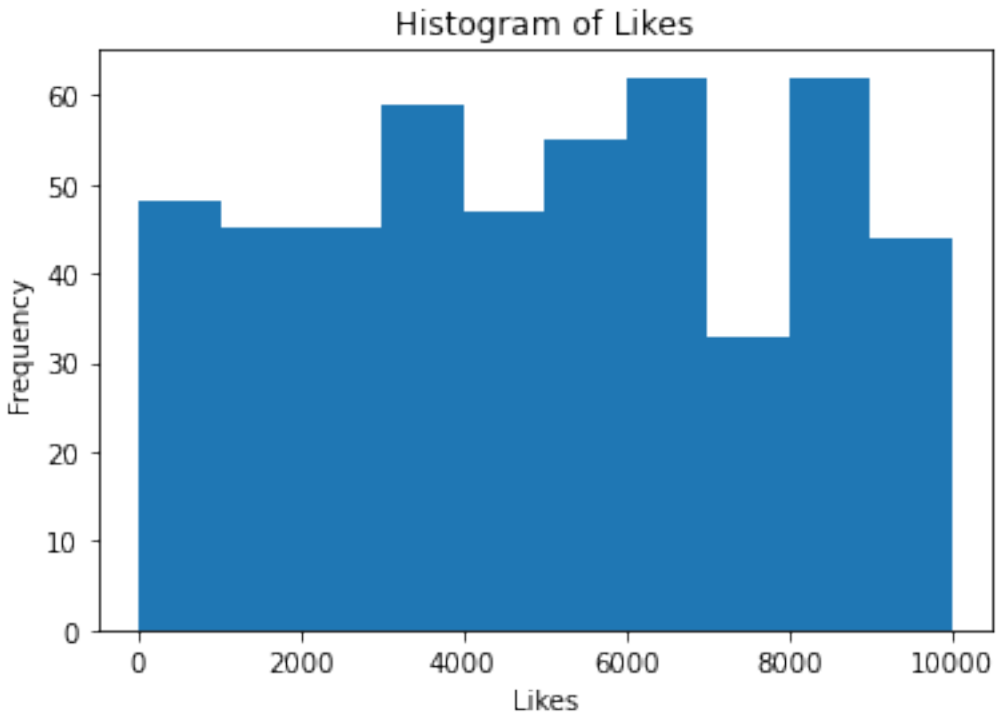
```
         Date Category  Likes
0  2021-01-01    Music   5112
1  2021-01-02    Music   8913
2  2021-01-03   Family     95
3  2021-01-04     Food   2809
```

```
4   2021-01-05   Travel   1056
..         …      …    …
495 2022-05-11  Fitness   3304
496 2022-05-12   Family   8816
497 2022-05-13    Music   7998
498 2022-05-14     Food   6148
499 2022-05-15     Food   8408

[500 rows x 3 columns]
```

[ ]:
```
Visualize and Analyze the data
```

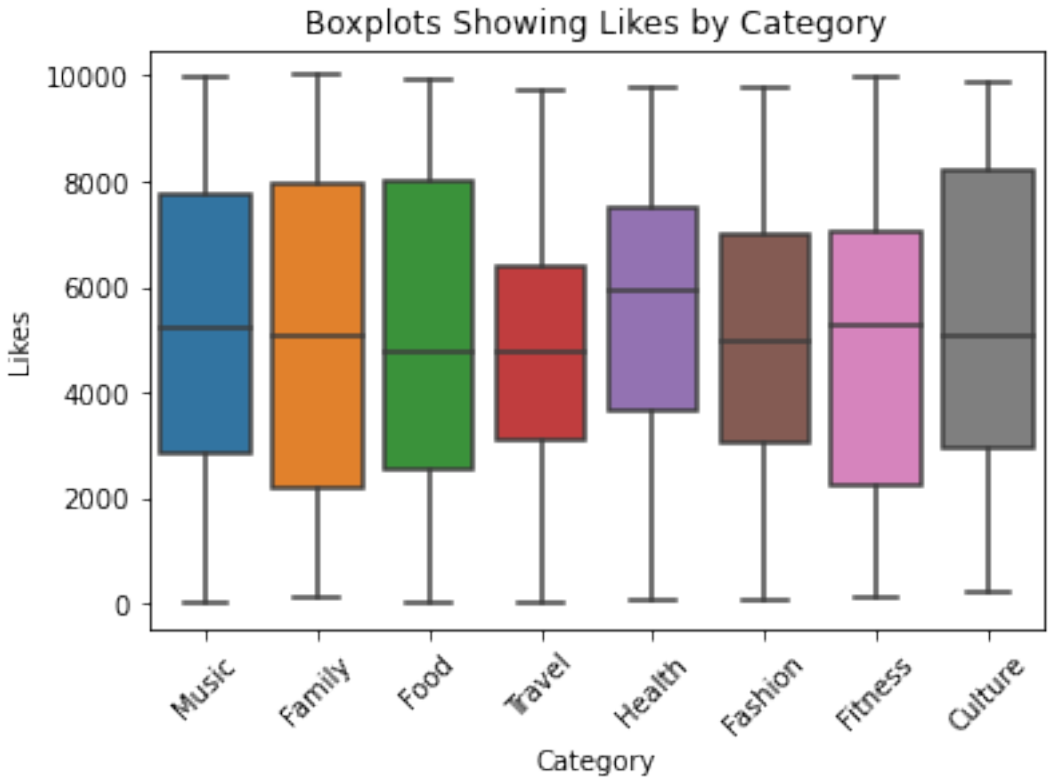
[13]:
```
plt.hist(data=df, x='Likes')
plt.title("Histogram of Likes")
plt.xlabel('Likes')
plt.ylabel('Frequency')
plt.show()
```



[ ]:
```
The histogram shows the distribution of Likes.
Generally, Likes are equally distributed between 0 to 25000.
```

[14]:
```
sns.boxplot(data=df, x='Category', y='Likes')
plt.title('Boxplots Showing Likes by Category')
plt.xticks(rotation=45)
```

```
plt.show()
```



[15]:
```
# Compute the mean value from the Likes column.

np.mean(df["Likes"])
```

[15]: 5059.998

[ ]:
```
This measure is important because it tells the average likes based on the data.
The mean value for the Likes column is approximately 5060
This means that the average likes from
the data is 5060
```

[16]:
```
# Compute the median value from the Likes column.
np.median(df['Likes'])
```

[16]: 5108.5

[ ]:
```
This measurement is important. This means that half the Likes values in the data
are below 5108.
```

[17]:
```
# Identify the minimum the value from the Likes column.
np.min(df['Likes'])
```

[17]:
```
10
```

[ ]:
```
The minimum value for the Likes is 10.
This means that the smallest value in the data is 10.
```

[18]:
```
# Identify the maximum value from the Likes column.
np.max(df['Likes'])
```

[18]:
```
9998
```

[ ]:
```
This value tells me which value is the largest likes value in the data.
The maximum value in the Likes column is 9998. This means that the largest value
in the data is 9998.
```

[19]:
```
# Compute the standard deviation for the Likes column.
np.std(df['Likes'], ddof=1)
```

[19]:
```
2864.260119133057
```

[ ]:
```
The standard deviation for the Likes column is 2864.26 (rounded to 2 decimal␣
↪places).
This measure how spread the out the Likes values are in the data.
```

[20]:
```
# Use the groupby method to print the mean for each category 'likes'.

category_mean_likes=df.groupby('Category')['Likes'].mean()
print(category_mean_likes)
```

```
Category
Culture    5183.511111
Family     5207.476190
Fashion    5034.625000
Fitness    4869.589041
Food       4985.155844
Health     5351.078125
Music      5306.519231
Travel     4656.596774
Name: Likes, dtype: float64
```

[ ]:
```
These measures are important because they tell the average for all of the
different categories.
Travel has the highest average - 4656.60 (rounded to 2 decimal)
```

[ ]:
```
Summary

Imported the required libraries which were:
pandas - for creating the data frame.
numpy - for forming a random number from a range.
Matplotlib.pyplot - for displaying graphs
seaborn - for plotting the data
random - for making a choice from a list of items.

Generated random tweet data to analyze.

Loaded the data into a pandas dataframe and explored it.

Exploratory Data Analysis(EDA) was performed on the data.

The data was cleaned: removed null data and duplicated data .

Converted Date field to datetime format so that it can be properly display the␣
 ↪data.

Converted Likes data type to int.


Conclusion

-The Health category had the highest median value. Which means, the contents␣
 ↪are read the most and has the most likes.

-Unfortunately, Health category does not have the most published articles, it␣
 ↪is fourth on the list.

-The Travel category publishes the most contents and has one of the smallest␣
 ↪medians, which means that it received the least
amount of likes, and there is a small number of users that like the Travel␣
 ↪contents.

-The second most engaging category that received the second highest median is␣
 ↪Music. It is the third to last on the list for the
number of published contents.



Recommendations:
1. Increase the number of published contents for the Health and Music␣
 ↪categories.
```

```
2. Publish less articles for categories that receive small numbers of likes.

3. Conduct a survey from a sample of the online users to learn about the␣
 ↪categories they like and ways to improve the categories
that receive a small number of likes.

4. More analyzing is recommended to be performed on the data. Models can be␣
 ↪built to better understand the relationship between
the different categories and likes.
```