SocialMediaDataAnalysis

October 28, 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.

[1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import random
#Generate random data
categories = ['Food', 'Travel', 'Fashion', 'Fitness', 'Music', 'Culture',␣
 ↪'Family','Health']
n=500
data ={ 'Date':pd.date_range('2021-01-01',periods=n),
      'Category':[random.choice(categories) for i in range(n)],
      'Likes':np.random.randint(0,10000, size=n)}
#Loading
print("Results")
print('\n')
df=pd.DataFrame(data)
print(df.head())
print('\n')
print('#Output for head()of the dataframe')
print('\n')
print('\n')
print(df.info())
print('\n')
print('#Output for the info() of the dataframe')
print('\n')
print('\n')
print(df.describe())
print('\n')
print('#Output for describe() of the dataframe')
print('\n')
print('\n')
print(df['Category'].value_counts())
print('\n')
print('#Output for value_counts() of the dataframe')
print('\n')
print('\n')



#Cleaning
```

```
df_cleaned=df.dropna()
print('\n')
df_cleaned=df_cleaned.drop_duplicates()
df_cleaned['Date']=pd.to_datetime(df_cleaned['Date'])
df_cleaned['Likes']=df_cleaned['Likes'].astype(int)
print(df_cleaned.info())
print('\n')
print('#Output for the info() for the cleaned dataframe')
print('\n')
print('\n')

#visualising
sns.distplot(df_cleaned['Likes'],bins=30,kde=False)
plt.title("Distribution of Likes")
plt.xlabel("Likes")
plt.ylabel("Frequency")
plt.show()
print('\n')
print('#Output for the displot created for distribution of likes using seaborn')
print('\n')
print('\n')

#Analysing-Graphs2
sns.boxplot(x="Category",y='Likes',data=df_cleaned)
plt.title("Category")
plt.ylabel("Likes")
plt.xticks(rotation=45)
plt.show()
print('\n')
print("#Output for the boxplot created for the likes by the category using␣
 ↪seaborn")
print('\n')
print('\n')
print('\n')

#Statistics
mean_likes=df_cleaned['Likes'].mean()
print(f"Overall mean of likes : {mean_likes}")
mean_likes_by_category=df_cleaned.groupby("Category")['Likes'].mean()
print(mean_likes_by_category)
print('\n')
print("#Output for the overall mean of likes and man for the likes based on the␣
 ↪category in the DataFrame")
```

Results

```
        Date Category  Likes
0 2021-01-01   Travel   9241
1 2021-01-02    Music   7264
2 2021-01-03   Family   7362
3 2021-01-04    Music    148
4 2021-01-05   Health    610
```

```
#Output for head()of the dataframe
```

```
<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
```

```
#Output for the info() of the dataframe
```

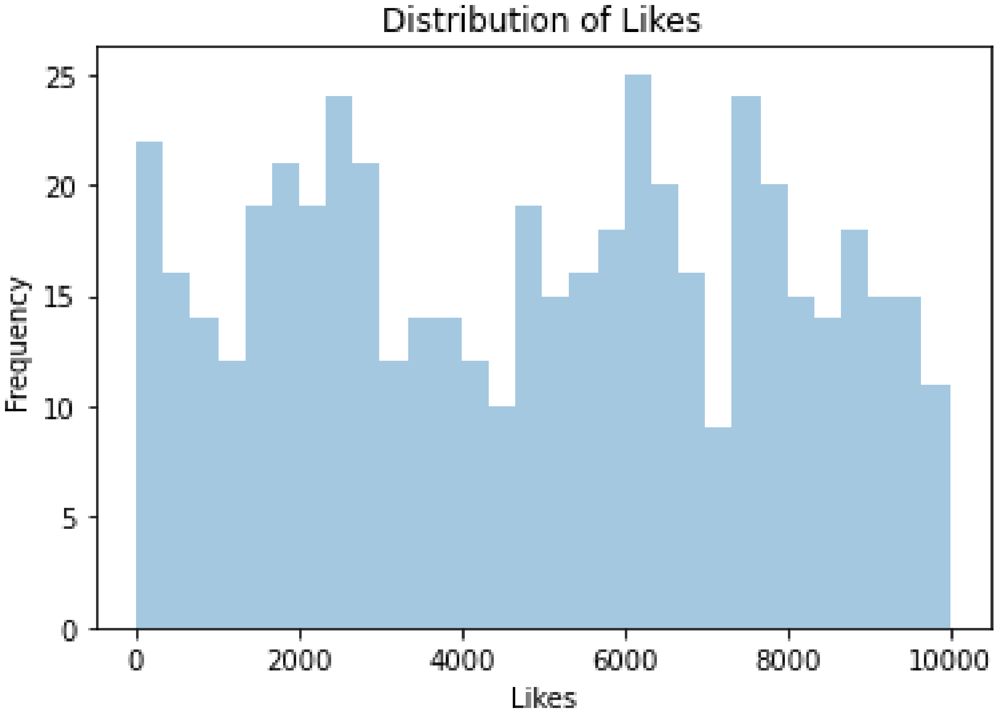
```
             Likes
count   500.000000
mean   4885.606000
std    2864.842143
min       4.000000
25%    2360.500000
50%    5041.500000
75%    7392.000000
max    9989.000000
```

```
#Output for describe() of the dataframe
```

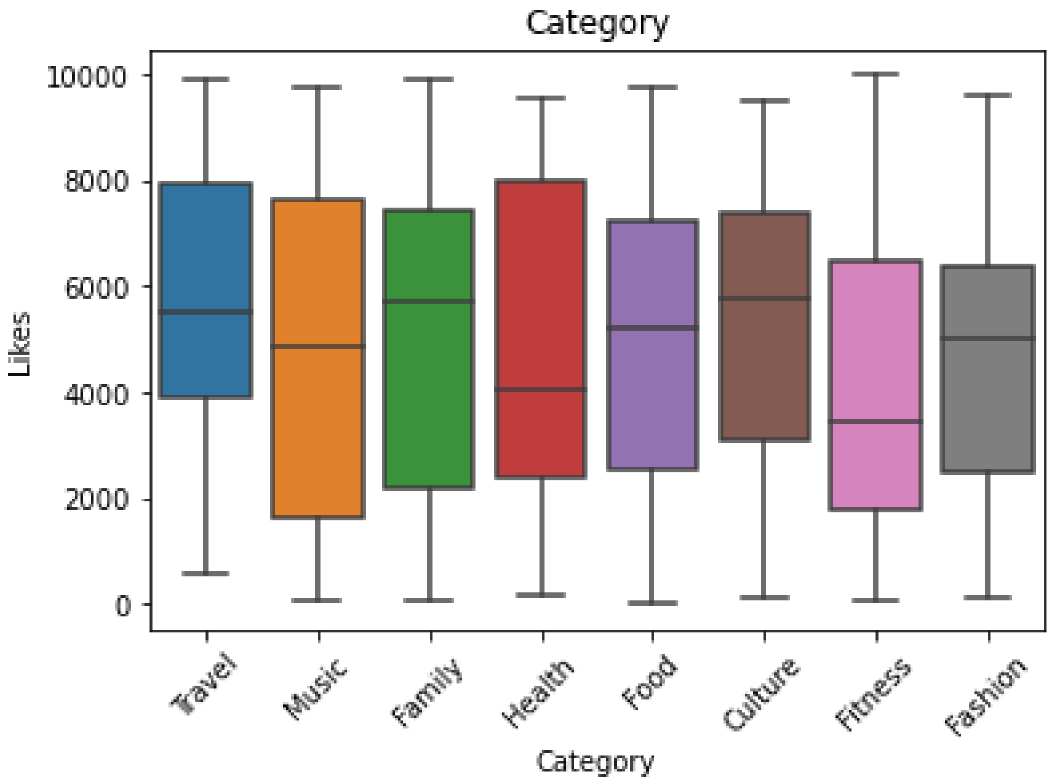
```
Music      75
```

```
Health     72
Food       71
Culture    65
Travel     60
Family     56
Fitness    53
Fashion    48
Name: Category, dtype: int64


#Output for value_counts() of the dataframe
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 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: 15.6+ KB
None


#Output for the info() for the cleaned dataframe
```



```
#Output for the displot created for distribution of likes using seaborn
```



```
#Output for the boxplot created for the likes by the category using seaborn
```

```
Overall mean of likes : 4885.606
Category
Culture    5172.200000
Family     4928.892857
Fashion    4622.125000
Fitness    4188.377358
Food       4994.718310
Health     4888.430556
Music      4545.906667
Travel     5653.516667
Name: Likes, dtype: float64
```

```
#Output for the overall mean of likes and man for the likes based on the
category in the DataFrame
```

[ ]:

```

```

[ ]:

```

```