SocialMediaDataAnalysis

July 23, 2023

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.

[50]:

```
# your code here
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
import random
```

[51]:

```
categories = ['Food', 'Travel', 'Fashion', 'Fitness','Music','Culture',␣
 ↪'Family','Health']
data = {'Date': pd.date_range('2003-08-12', periods=500),'Category': [random.
 ↪choice(categories) for _ in range(500)],'Likes': np.random.randint(0, 10000,␣
 ↪size=500)}
```

[52]:

```
df = pd.DataFrame(data)
print(df.head())
print('\n')
print(df.info())
print('\n')
print(df.describe())
```

```
        Date Category  Likes
0 2003-08-12     Food   2759
1 2003-08-13  Culture   7926
2 2003-08-14     Food   8695
3 2003-08-15   Health   5322
4 2003-08-16   Health   7497


<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   5009.286000
std    2891.100417
min      31.000000
25%    2469.000000
50%    5001.000000
75%    7495.500000
max    9997.000000
```

[53]:

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

[53]:

```
Health     72
Music      72
Travel     70
Fitness    64
Food       62
Fashion    57
Culture    52
Family     51
Name: Category, dtype: int64
```

[54]:

```
df = df.dropna()
df
```

[54]:

```
          Date Category  Likes
0   2003-08-12     Food   2759
1   2003-08-13  Culture   7926
2   2003-08-14     Food   8695
3   2003-08-15   Health   5322
4   2003-08-16   Health   7497
..         …        …     …
495 2004-12-19   Health   6583
496 2004-12-20  Fitness   9439
497 2004-12-21   Family   5566
498 2004-12-22    Music   2821
499 2004-12-23   Family    811

[500 rows x 3 columns]
```

[55]:

```
df.drop_duplicates(inplace=True)
df
```

[55]:

```
          Date Category  Likes
0   2003-08-12     Food   2759
```

```
1    2003-08-13  Culture   7926
2    2003-08-14     Food   8695
3    2003-08-15   Health   5322
4    2003-08-16   Health   7497
..          …        …     …
495  2004-12-19   Health   6583
496  2004-12-20  Fitness   9439
497  2004-12-21   Family   5566
498  2004-12-22    Music   2821
499  2004-12-23   Family    811

[500 rows x 3 columns]
```

[56]:
```python
df['Date'] = pd.to_datetime(df['Date'])
df
```

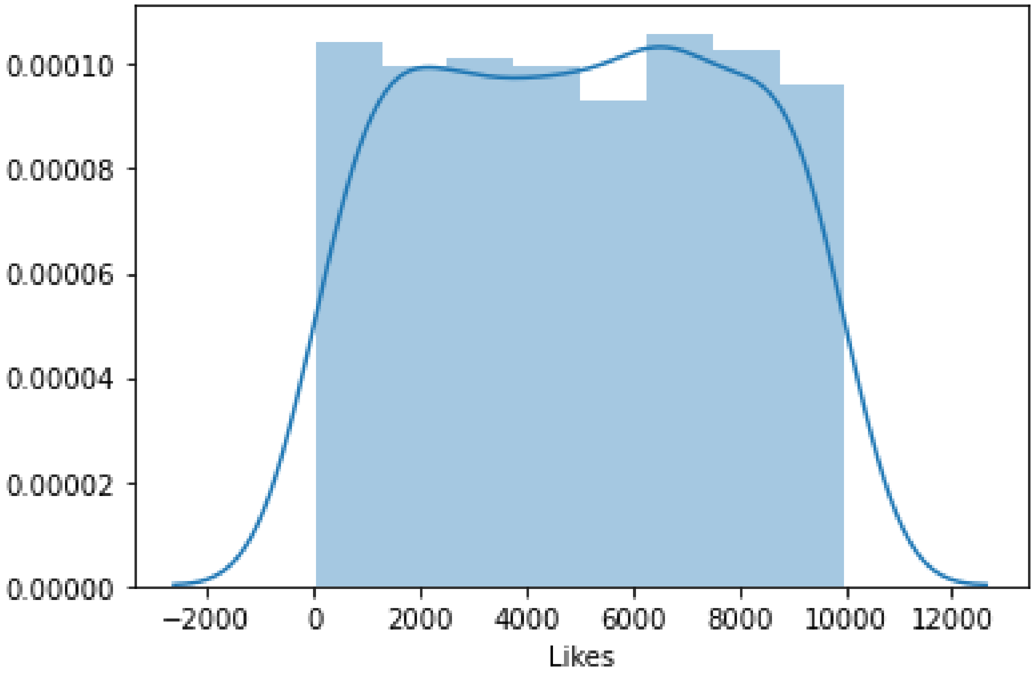
[56]:
```
           Date Category  Likes
0    2003-08-12     Food   2759
1    2003-08-13  Culture   7926
2    2003-08-14     Food   8695
3    2003-08-15   Health   5322
4    2003-08-16   Health   7497
..          …        …     …
495  2004-12-19   Health   6583
496  2004-12-20  Fitness   9439
497  2004-12-21   Family   5566
498  2004-12-22    Music   2821
499  2004-12-23   Family    811

[500 rows x 3 columns]
```

[57]:
```python
df['Likes'] = df['Likes'].astype(int)
df.dtypes
```

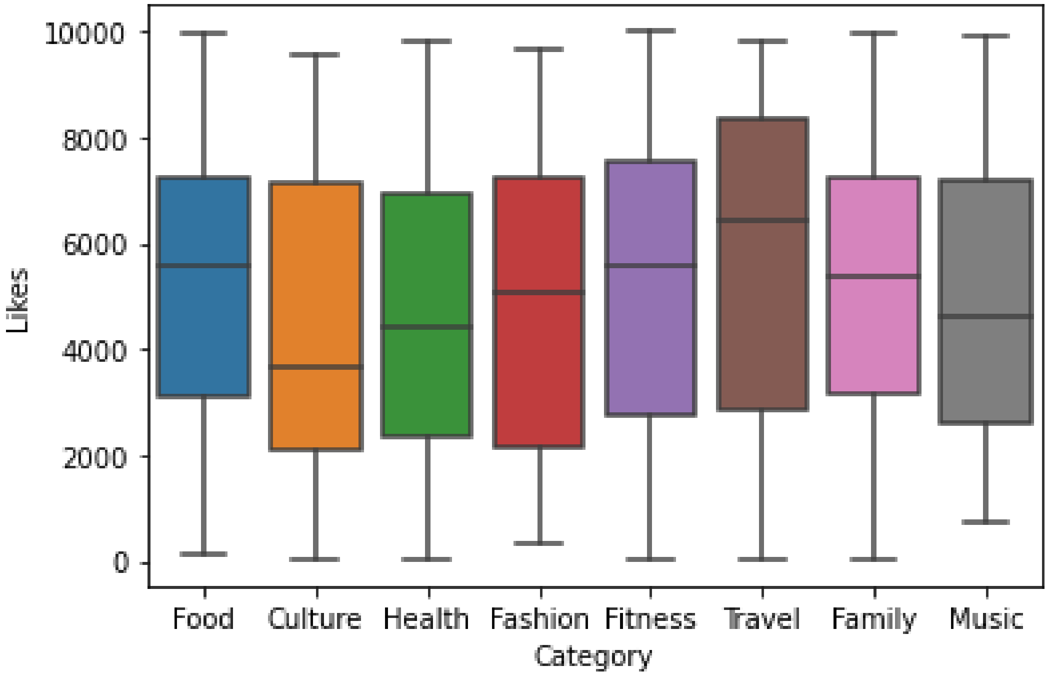
[57]:
```
Date        datetime64[ns]
Category            object
Likes                int64
dtype: object
```

[58]:
```python
sb.distplot(df['Likes'])
plt.show()
```



[59]:

```
sb.boxplot(x=df['Category'],y=df['Likes'])
plt.show()
```



[60]:

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

```
5009.286
```

[60]:

```
Category
Culture     4381.288462
Family      5281.313725
Fashion     4848.052632
Fitness     5180.093750
Food        5141.306452
Health      4624.638889
Music       4918.458333
Travel      5624.857143
Name: Likes, dtype: float64
```

[61]:

```
df.groupby('Category')['Likes'].agg(['mean','sum'])
```

[61]:

```
                 mean     sum
Category
Culture   4381.288462  227827
Family    5281.313725  269347
Fashion   4848.052632  276339
Fitness   5180.093750  331526
Food      5141.306452  318761
Health    4624.638889  332974
Music     4918.458333  354129
Travel    5624.857143  393740
```

1.5 Process

1. **Task 1: Importing Required Libraries**: In this step, I have imported the necessary libraries for data analysis. These include pandas for data manipulation, numpy for numerical computations, matplotlib and seaborn for data visualization, and random for generating random numbers.

2. **Task 2: Generating Random Data**: I have created a Random data Dictnoary containing 500 value, each representing a tweet. The data includes 'Date', 'Category', and 'Likes' Keys. The 'Date' Key's value is generated using `pd.date_range()`, the 'Category' Key's value is randomly chosen from the list of categories, and the 'Likes' Key's value contains random integers between 0 and 10000.

3. **Task 3: Converting Dictnoary to DataFrame and Explored it**: I have converted Dictnoary into DataFrame 'df' using `pd.DataFrame(data)`, I have performed some initial data inspection on the DataFrame 'df'. I used `df.head()` to display the first five rows of the data, `df.info()` to get information about the data types and non-null counts, and `df.describe()` to get summary statistics for the numeric columns.

4. **Task 4: Data Cleaning and Data Transformation**: I cleaned the data by dropping any rows with missing values using `df.dropna()`. Then, I removed duplicate rows using `df.drop_duplicates()`, I converted the ‘Date’ column to a datetime data type using `pd.to_datetime()`, and the ‘Likes’ column to integers using `.astype(int)`.

5. **Task 5: Data Visualization and Calculating Mean Likes across diffrent categories**: I created a distribution plot (histogram) using `sb.distplot()` to visualize the distribution of ‘Likes’ across all tweets. Additionally, I used a box plot with `sb.boxplot()` to compare the distribution of ‘Likes’ across different categories.I calculated the mean number of likes across all tweets using `df['Likes'].mean()` and also calculated the mean number of likes for each category using `df.groupby('Category')['Likes'].mean()`.

1.6 Key Findings

1. Through this `df.groupby('Category')['Likes'].agg(['mean','sum'])` functions I have found out that the topics related to `Travel` have most likes (‘393740’) and topics related to `Cluture` have least likes (‘227827’). so here we can say that the tiwtter user’s are more engaged towards `Travel` topics and less intrested in topics related to ‘Cl.

2. Through `sb.distplot(df['Likes'])` we can say the Likes are normal distributed for the above genrated data set.