SocialMediaDataAnalysis-Copy1

October 30, 2024

0.1 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.

0.2 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`

0.3 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.

0.4 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
```

[2]:
```
import numpy as np
```

[3]:
```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

[4]:
```
import pandas as pd
import numpy as np
import random

# Step 1: Generate random tweet data
# Define a list of categories for social media data
categories = ['food', 'travel', 'fashion', 'fitness', 'music', 'culture',␣
 ↪'health','education']

# Generate random data
n = 500  # Number of entries
data = {
    'Date': pd.date_range(start='2021-01-01', periods=n),
    'Category': [random.choice(categories) for _ in range(n)],
    'Likes': np.random.randint(0, 10000, size=n)
}

# Create a DataFrame from the random data
df = pd.DataFrame(data)

# Display the first few entries of the DataFrame
print(df.head())

# Optional: Save the DataFrame to a CSV file
df.to_csv('random_tweet_data.csv', index=False)
```

```
        Date   Category  Likes
0 2021-01-01  education   5944
1 2021-01-02     travel   9988
2 2021-01-03       food    227
3 2021-01-04     health   9959
4 2021-01-05    fashion   7362
```

[5]:
```
# Step 2: Load data into a DataFrame
tweets_df = pd.DataFrame(data) # Create a DataFrame from the data dictionary

# Print the first few rows of the DataFrame to check the data
print(tweets_df.head())
```

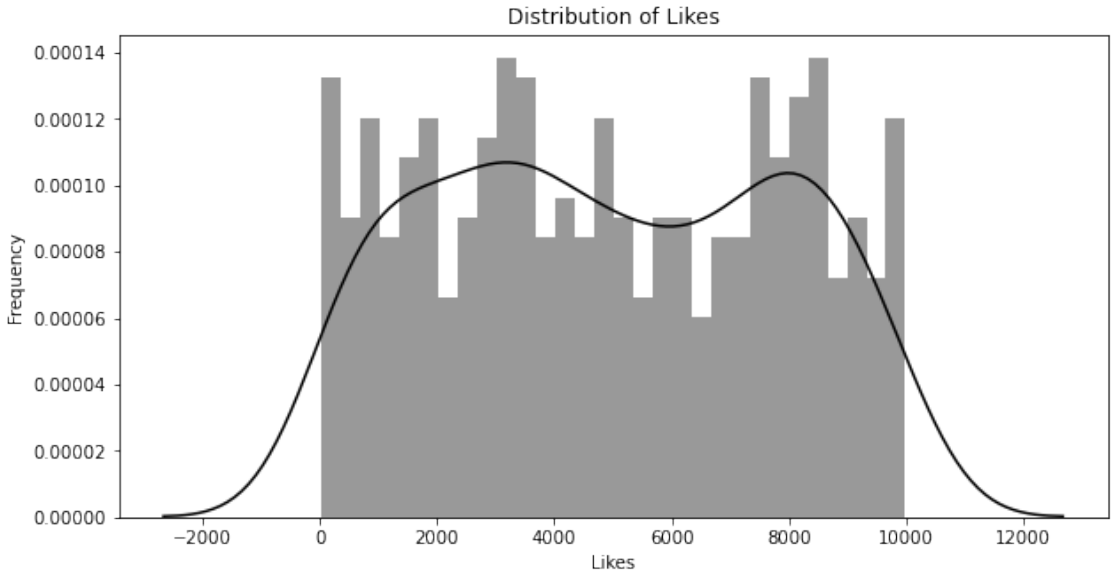
```
        Date   Category  Likes
0 2021-01-01  education   5944
1 2021-01-02     travel   9988
2 2021-01-03       food    227
3 2021-01-04     health   9959
4 2021-01-05    fashion   7362
```

[6]:
```
tweets_df.dropna(inplace=True) # Remove any rows with null values
tweets_df.drop_duplicates(inplace=True) # Remove duplicate entries
tweets_df['Date'] = pd.to_datetime(tweets_df['Date']) # Ensure 'Data' is in␣
 ↪datetime format
tweets_df['Likes'] = tweets_df['Likes'].astype(int) # Convert 'Likes' to␣
 ↪integer type


print(tweets_df.head())  # Print the first few rows of the cleaned data

print(tweets_df.info()) # Print a summary of the DataFrame
```

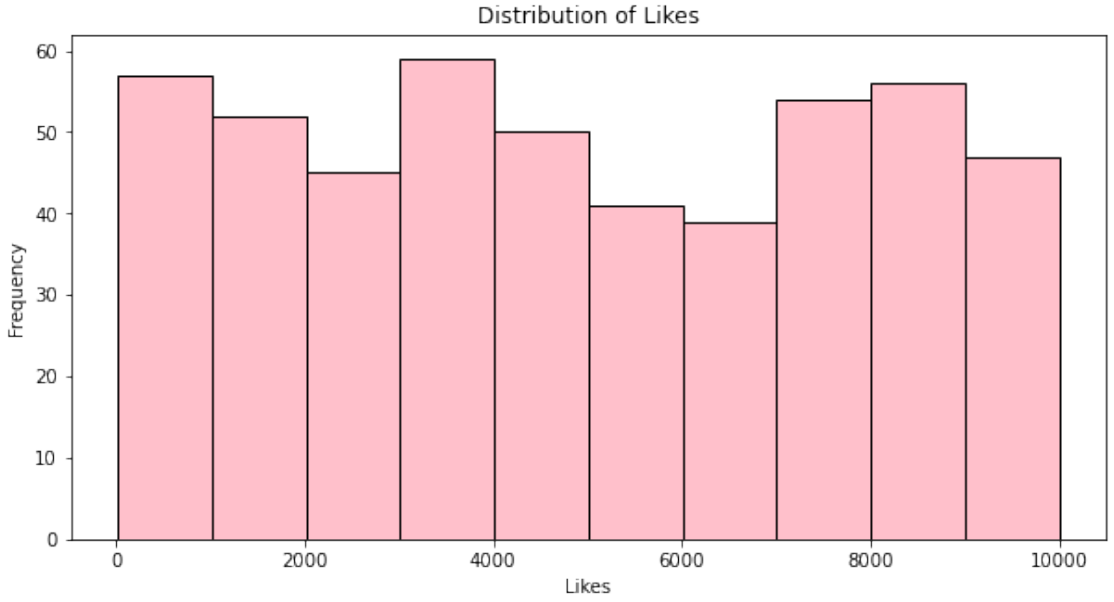
```
        Date   Category  Likes
0 2021-01-01  education   5944
1 2021-01-02     travel   9988
2 2021-01-03       food    227
3 2021-01-04     health   9959
4 2021-01-05    fashion   7362
<class 'pandas.core.frame.DataFrame'>
Int64Index: 500 entries, 0 to 499
Data columns (total 3 columns):
 #   Column    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
```

[7]:
```
import pandas as pd
import numpy as np
import random
import matplotlib.pyplot as plt
```

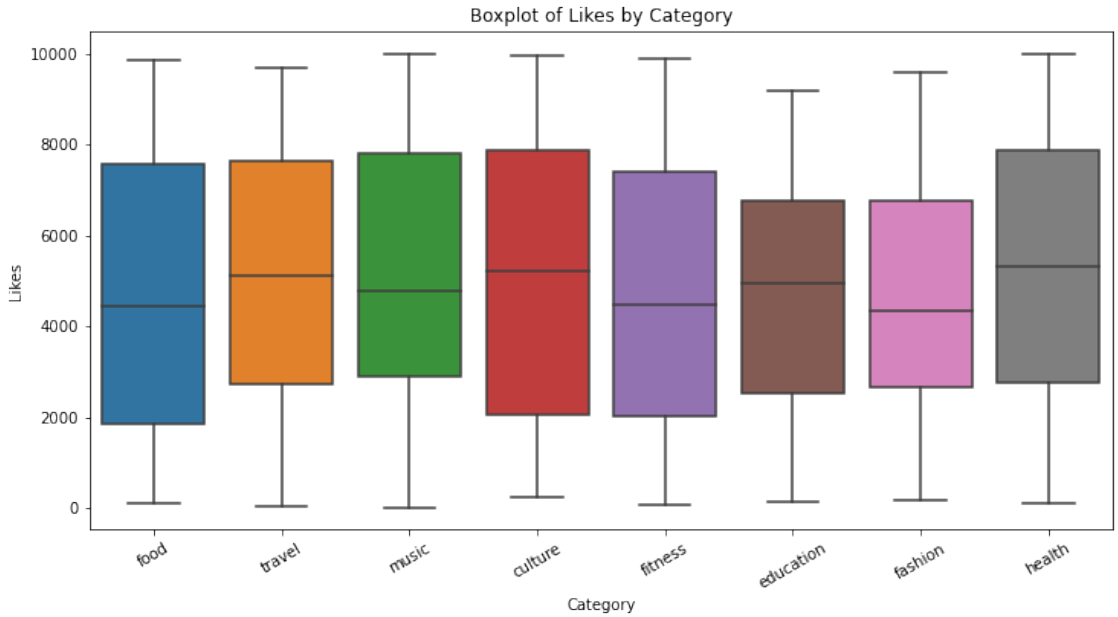
```
import seaborn as sns


categories = ['food', 'travel', 'fashion', 'fitness', 'music', 'culture',␣
↪'health','education']


n = 500  # Number of entries
data = {
    'Date': pd.date_range(start='2021-01-01', periods=n),
    'Category': [random.choice(categories) for _ in range(n)],
    'Likes': np.random.randint(0, 10000, size=n)
}

# Create a DataFrame from the random data
tweets_df = pd.DataFrame(data)



plt.figure(figsize=(10, 5))
sns.distplot(tweets_df['Likes'], bins=30,color='k')
plt.title('Distribution of Likes')
plt.xlabel('Likes')
plt.ylabel('Frequency')
plt.show()
```



[8]:
```
plt.figure(figsize=(10, 5))
plt.hist(tweets_df['Likes'],edgecolor='black',color='pink') # Plot the␣
 ↪distribution of Likes
plt.title('Distribution of Likes') # Add title to the histogram
plt.xlabel('Likes')
plt.ylabel('Frequency')
plt.show()
```



[9]:
```
plt.figure(figsize=(12, 6))
sns.boxplot(x='Category', y='Likes', data=tweets_df) # Plot Likes distribution␣
 ↪across categories
plt.title('Boxplot of Likes by Category') # Add title to the boxplot
plt.xlabel('Category')
plt.ylabel('Likes')
plt.xticks(rotation=30)
plt.show()
```



[10]:
```
mean_likes = tweets_df['Likes'].mean() # Calculate the mean of Likes
print(f"Mean Likes: {mean_likes:.2f}")
```

```
Mean Likes: 4925.55
```

[11]:
```
df_cleaned = df.dropna()# Remove rows with null values df_cleaned = df_cleaned.
 ↪drop_duplicates()# Remove duplicate rows
df_cleaned['Date'] = pd.to_datetime(df_cleaned['Date'])# Convert the 'Date'␣
 ↪field to datetime format
df_cleaned['Likes'] = df_cleaned['Likes'].astype(int)# Convert the 'Likes'␣
 ↪field to integer
print(df_cleaned.info())
```

```
<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
```

[12]:
```
import pandas as pd

# Assuming df is a pandas DataFrame that you are working with
df_cleaned = df.dropna()  # Remove rows with null values
df_cleaned = df_cleaned.drop_duplicates()  # Remove duplicate rows
df_cleaned['Date'] = pd.to_datetime(df_cleaned['Date'])  # Convert the 'Date'␣
↪field to datetime format
df_cleaned['Likes'] = df_cleaned['Likes'].astype(int)  # Convert the 'Likes'␣
↪field to integer
```

[ ]:
```

```