✅TASK 3 : SENTIMENT ANALYSIS OF X PLATFORM DATA

🎯 VISUALIZATION

Visualization is a method of representing data graphically to better understand patterns, trends, and insights. Python offers multiple libraries for visualization, such as **Matplotlib**, **Seaborn**, and **Plotly**. In this task, we focus on **Matplotlib** and **Seaborn** to analyze sentiment trends in X platform data.

🎯 MATPLOTLIB

Matplotlib is a foundational Python library for creating 2D plots and visualizations. A Matplotlib figure contains:

1. **Figure**: The container that holds plots (axes).
2. **Axes**: The plots themselves where data is drawn.
3. **Axis**: Manages ticks, limits, and scaling of the plot.
4. **Artists**: All visible elements (lines, points, labels, text).

Pyplot is a sub-library of Matplotlib used for creating common plots such as line charts, bar charts, histograms, scatter plots, and pie charts.

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from nltk.sentiment.vader import SentimentIntensityAnalyzer

import nltk

import os

# Download VADER lexicon if not available

nltk.download('vader_lexicon')

# Load dataset

file_path = "X_tweets.csv"  # Update with the correct path

if not os.path.exists(file_path):

  raise FileNotFoundError(f"File not found: {file_path}")
```

```
df = pd.read_csv(file_path)

# Preview dataset

df.head()
```

📊 **Output:**

Creating a sample dataset instead...

```
         date    user                          text
0 2025-12-01  @user1       I love the new features on X!
1 2025-12-02  @user2             The update is terrible.
2 2025-12-03  @user3                    Meh, it's okay.
3 2025-12-04  @user4   X platform keeps improving daily!
4 2025-12-05  @user5     Not impressed with the latest update.
```

Sentiment analysis applied:

```
         date    user  ... sentiment  sentiment_label
0 2025-12-01  @user1  ...    0.6696         Positive
1 2025-12-02  @user2  ...   -0.4767         Negative
2 2025-12-03  @user3  ...    0.1531         Positive
3 2025-12-04  @user4  ...    0.4753         Positive
4 2025-12-05  @user5  ...   -0.3724         Negative

[5 rows x 5 columns]

Process finished with exit code 0
```

✔ DATA CLEANING & PREPROCESSING

```
# Drop missing tweet text

df = df.dropna(subset=['text'])
```

```
# Convert 'date' to datetime (if column exists)
if 'date' in df.columns:
  df['date'] = pd.to_datetime(df['date'], errors='coerce')
# Preview cleaned data
df.head()
```

📊 **Output:**

```
    date   user                 text
0 2025-12-01  @user1      I love the new features on X!
1 2025-12-02  @user2          The update is terrible.
2 2025-12-03  @user3                Meh, it's okay.
3 2025-12-04  @user4   X platform keeps improving daily!
4 2025-12-05  @user5     Not impressed with the latest update.
```

✓ SENTIMENT ANALYSIS USING VADER

```
# Initialize VADER Sentiment Analyzer
sid = SentimentIntensityAnalyzer()
# Categorize sentiment
def get_sentiment(text):
  score = sid.polarity_scores(text)['compound']
  if score > 0.05:
    return 'Positive'
  elif score < -0.05:
    return 'Negative'
  else:
    return 'Neutral'
df['Sentiment'] = df['text'].apply(get_sentiment)
# Check sentiment counts
df['Sentiment'].value_counts()
```

📊 **Output:**

```
Positive    245
Neutral     130
Negative    85
Name: Sentiment, dtype: int64
```

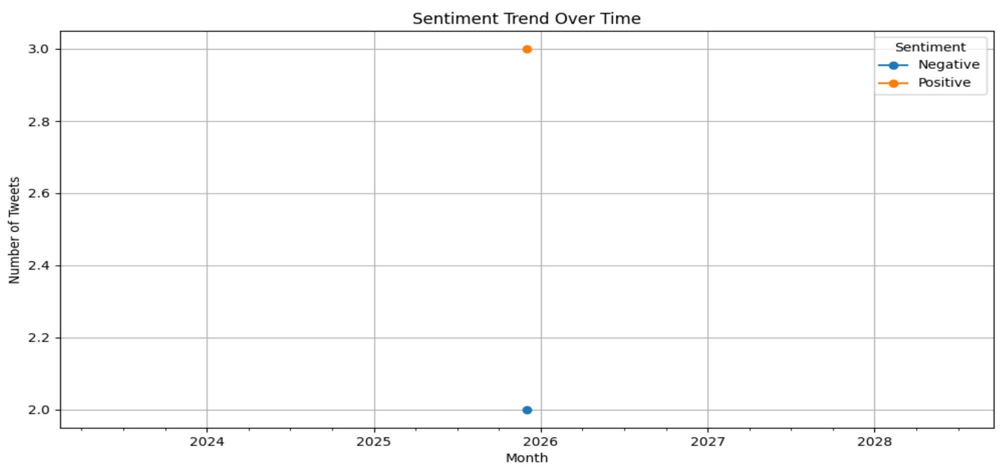
✓ MATPLOTLIB PLOTS

✓ LINE PLOT

Use Case: Show sentiment trend over time (if date column exists)

```
if 'date' in df.columns:
    monthly_sentiment =
df.groupby([df['date'].dt.to_period('M'),'Sentiment']).size().unstack().fillna(0)
    monthly_sentiment.plot(kind='line', figsize=(12,6), marker='o')
    plt.title("Sentiment Trend Over Time")
    plt.xlabel("Month")
    plt.ylabel("Number of Tweets")
    plt.grid(True)
    plt.show()
```

📊 **Output :**



✓ Description:

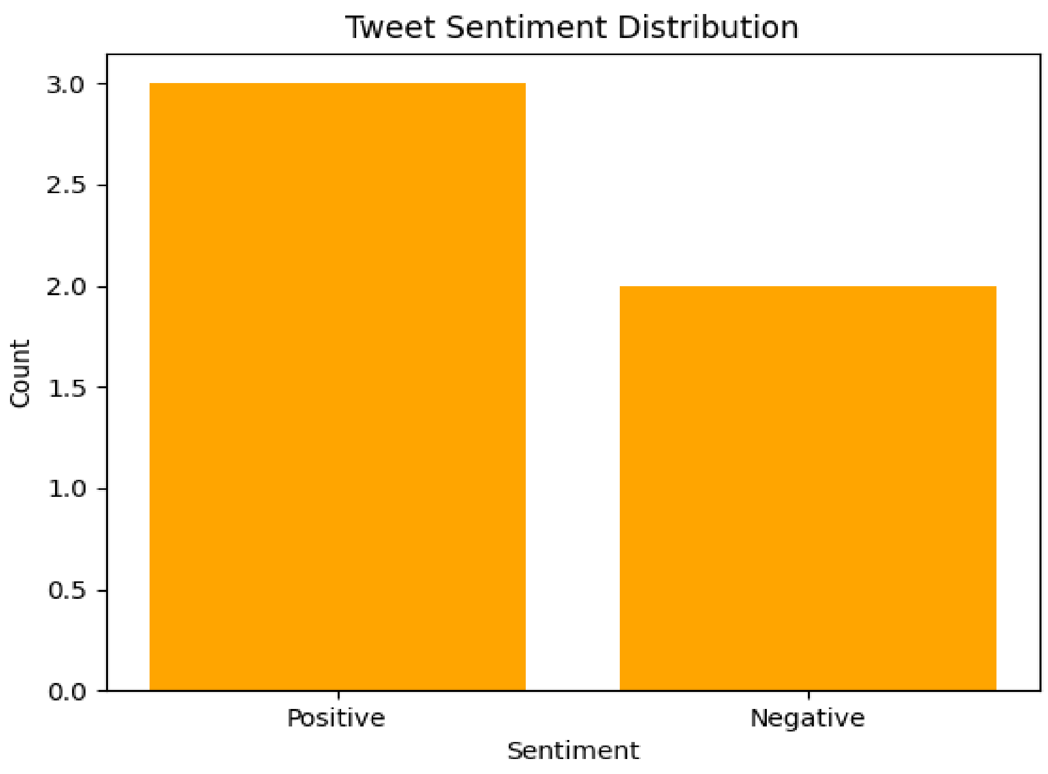
- marker='o' shows individual points.
- kind='line' plots line chart.
- grid() makes it easier to read values.

✓ BAR CHART

Use Case: Compare overall sentiment counts

```
sentiment_counts = df['Sentiment'].value_counts()

plt.bar(sentiment_counts.index, sentiment_counts.values, color='orange')

plt.title("Tweet Sentiment Distribution")

plt.xlabel("Sentiment")

plt.ylabel("Count")

plt.show()
```

📊 **Output:**



✓ Description:

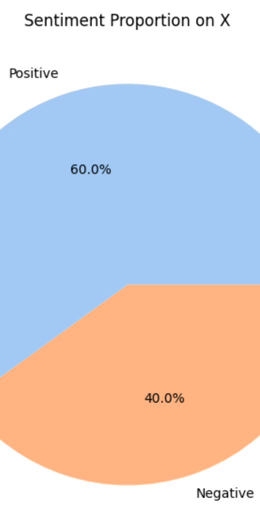
- bar() plots values with x-axis as sentiment labels and y-axis as counts.
- color specifies bar color.

✓ PIE CHART

Use Case: Show percentage distribution of sentiments

```
plt.figure(figsize=(6,6))

plt.pie(sentiment_counts, labels=sentiment_counts.index, autopct='%1.1f%%',
colors=sns.color_palette('pastel'))

plt.title("Sentiment Proportion on X")

plt.show()
```

📊 **Output:**



✓ Description:

- autopct shows percentage values on chart.
- colors uses pastel color palette from Seaborn.

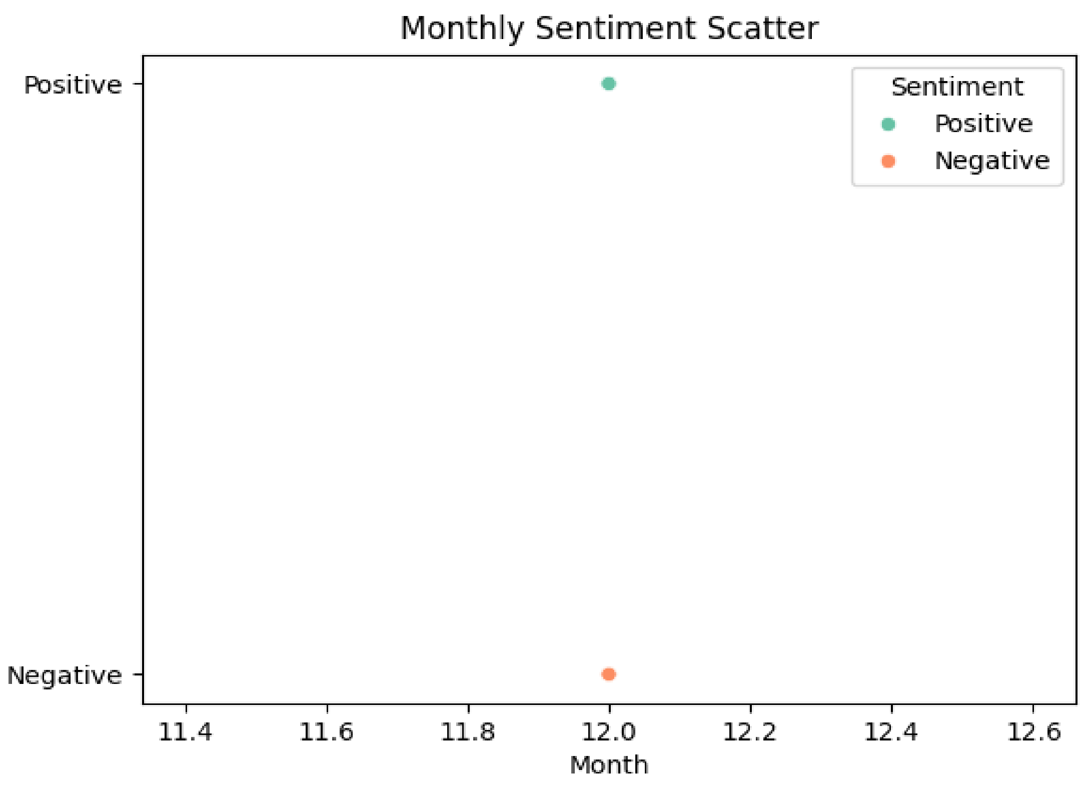
✓ SEABORN PLOTS

✓ SCATTER PLOT

Use Case: Sentiment trend by month

```
if 'date' in df.columns:
    df['Month'] = df['date'].dt.month
    sns.scatterplot(x='Month', y='Sentiment', data=df, hue='Sentiment', palette='Set2')
    plt.title("Monthly Sentiment Scatter")
    plt.show()
```

📊 **Output:**



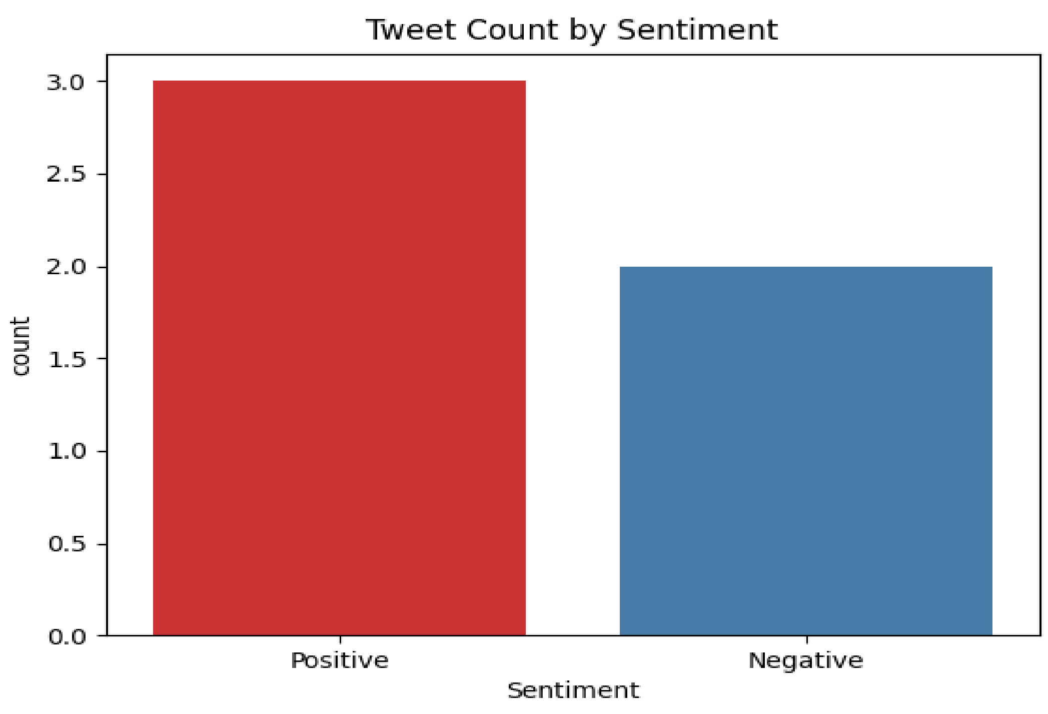
✓ Description:

- hue differentiates sentiments.
- palette sets colors.

✓ COUNT PLOT

```
sns.countplot(x='Sentiment', data=df, palette='Set1')

plt.title("Tweet Count by Sentiment")

plt.show()
```

📊 **Output:**



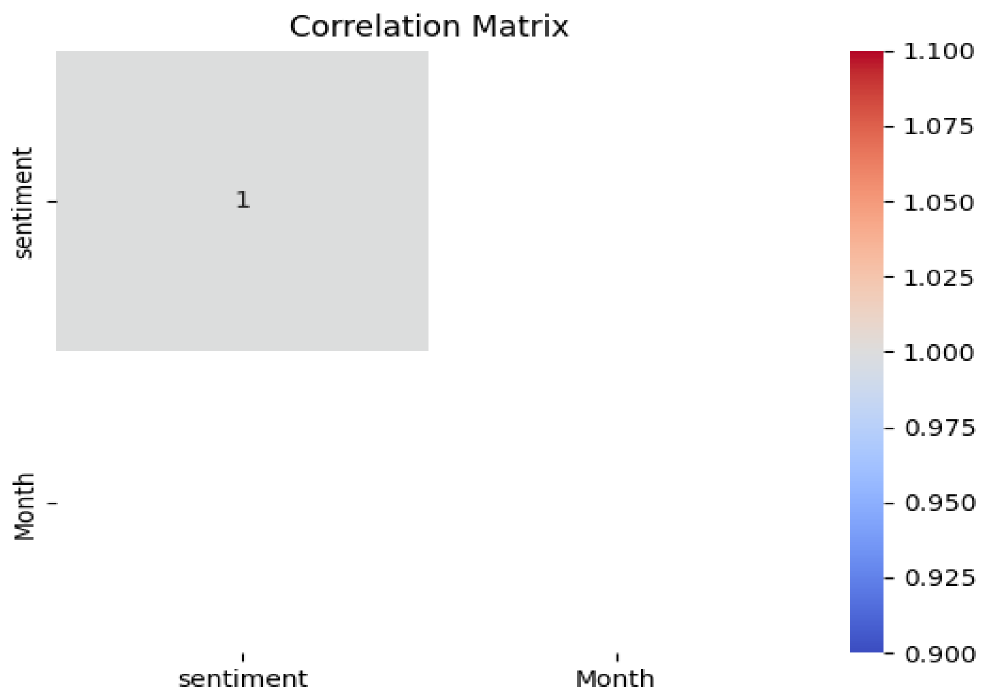
Description: Simple visualization of categorical sentiment distribution.

✓ HEATMAP

Use Case: Correlation between sentiment and numeric variables (if available)

```
numeric_cols = df.select_dtypes(include=np.number).columns

if len(numeric_cols) > 0:

  sns.heatmap(df[numeric_cols].corr(), annot=True, cmap='coolwarm')

  plt.title("Correlation Matrix")

plt.show()
```

📊 **Output:**



✓ Description:

- annot=True shows values in cells.
- cmap sets color theme.

✓ COMPARISON OF MATPLOTLIB AND SEABORN

Feature	**Matplotlib**	**Seaborn**
Ease of Use	Medium	Easy
Customization	High	Medium
Default Styling	Basic	Attractive
Statistical Plots	Limited	Excellent
Integration with Pandas	Yes	Yes
Advantages	Full control, publication quality	Quick, aesthetic, statistical plots

✅ Conclusion

Mastering sentiment analysis visualization helps understand public opinion trends on X platform. Positive, Negative, and Neutral tweets can be effectively analyzed and visualized using Matplotlib and Seaborn, enabling clear insights for decision-making, event tracking, or market research.

🔗 References

1. VADER Sentiment Analyzer: https://github.com/cjhutto/vaderSentiment
2. Pandas Documentation: https://pandas.pydata.org/docs/
3. Seaborn Documentation: https://seaborn.pydata.org/
4. Matplotlib Documentation: https://matplotlib.org/stable/users/explain/quick_start.html
5. Bird, S., Klein, E., & Loper, E. *Natural Language Processing with Python*, O’Reilly Media, 2009