

```
import csv

# تحميل البيانات من ملف CSV
with open('sentimentdataset.csv', mode='r') as file:
    reader = csv.DictReader(file)
    data = list(reader)

# عرض أول 5 صفوف للتأكد من تحميل البيانات بشكل صحيح
for row in data[:5]:
    print(row)
```

{': '0', 'Unnamed: 0': '0', 'Text': ' Enjoying a beautiful day at the park! ', 'Sentiment': ' Positive ', 'Timestamp': '2023-01-15 12:30:00', 'User': ' User123 ', 'Platform': ' Twitter	
{': '1', 'Unnamed: 0': '1', 'Text': ' Traffic was terrible this morning. ', 'Sentiment': ' Negative ', 'Timestamp': '2023-01-15 08:45:00', 'User': ' CommuterX ', 'Platform': ' Twitter	
{': '2', 'Unnamed: 0': '2', 'Text': ' Just finished an amazing workout! 🏋️ ', 'Sentiment': ' Positive ', 'Timestamp': '2023-01-15 15:45:00', 'User': ' FitnessFan ', 'Platform': ' Instag	
{': '3', 'Unnamed: 0': '3', 'Text': ' Excited about the upcoming weekend getaway! ', 'Sentiment': ' Positive ', 'Timestamp': '2023-01-15 18:20:00', 'User': ' AdventureX ', 'Platform': ' Facebo	
{': '4', 'Unnamed: 0': '4', 'Text': ' Trying out a new recipe for dinner tonight. ', 'Sentiment': ' Neutral ', 'Timestamp': '2023-01-15 19:55:00', 'User': ' ChefCook ', 'Platform': ' Instagr	

```
columns_to_check = ['Text', 'Sentiment', 'User', 'Platform', 'Hashtags', 'Country', 'Year', 'Month', 'Day', 'Hour']
seen = set()
duplicates = []

# التحقق من الصفوف المكررة بناءً على الأعمدة المحددة
for row in data:
    # إنشاء مفتاح فريد مكون من القيم في الأعمدة المحددة
    row_key = tuple(row[column] for column in columns_to_check)

    if row_key in seen:
        duplicates.append(row)
    else:
        seen.add(row_key)

# عرض عدد الصفوف المكررة
print(f"عدد الصفوف المكررة: {len(duplicates)}")

# عرض أول 5 صفوف مكررة (إذا وجدت)
for row in duplicates[:5]:
    print(row)
```

عدد الصفوف المكررة: 20	
{': '280', 'Unnamed: 0': '284', 'Text': ' Proudly scaling the peaks of achievement, a mountaineer conquering challenges and planting the flag of success. ', 'Sentiment': ' Proud ', 'Timestamp': '2020-01	
{': '281', 'Unnamed: 0': '285', 'Text': ' Embraced by the hopeful dawn, a gardener sowing seeds of optimism, tending to the blooms of a brighter tomorrow. ', 'Sentiment': ' Hopeful ', 'Timestamp': '2022	
{': '282', 'Unnamed: 0': '286', 'Text': ' A playful escapade in the carnival of life, carousel laughter and cotton candy dreams swirling in the joyous atmosphere. ', 'Sentiment': ' Playful ', 'Timestamp	
{': '283', 'Unnamed: 0': '287', 'Text': ' Floating on clouds of inspiration, an artist painting the sky with strokes of creativity, creating a masterpiece of dreams. ', 'Sentiment': ' Inspired ', 'Timest	
{': '284', 'Unnamed: 0': '288', 'Text': ' Navigating the river of contentment, a serene boat cruise through the tranquil waters of inner peace and acceptance. ', 'Sentiment': ' Contentment ', 'Timestamp	

```
# عرض كل الصفوف المكررة
print(f"عدد الصفوف المكررة: {len(duplicates)}")

# عرض جميع الصفوف المكررة
for row in duplicates:
    print(row)
```

```
➡ عدد الصفوف المكررة: 20
{'': '280', 'Unnamed: 0': '284', 'Text': ' Proudly scaling the peaks of achievement, a mountaineer conquering challenges and planting the flag of success. ', 'Sentiment': ' Proud ', 'Timestamp': '2020-01
{'': '281', 'Unnamed: 0': '285', 'Text': ' Embraced by the hopeful dawn, a gardener sowing seeds of optimism, tending to the blooms of a brighter tomorrow. ', 'Sentiment': ' Hopeful ', 'Timestamp': '2022
{'': '282', 'Unnamed: 0': '286', 'Text': ' A playful escapade in the carnival of life, carousel laughter and cotton candy dreams swirling in the joyous atmosphere. ', 'Sentiment': ' Playful ', 'Timestamp
{'': '283', 'Unnamed: 0': '287', 'Text': ' Floating on clouds of inspiration, an artist painting the sky with strokes of creativity, creating a masterpiece of dreams. ', 'Sentiment': ' Inspired ', 'Timest
{'': '284', 'Unnamed: 0': '288', 'Text': ' Navigating the river of contentment, a serene boat cruise through the tranquil waters of inner peace and acceptance. ', 'Sentiment': ' Contentment ', 'Timestamp
{'': '285', 'Unnamed: 0': '289', 'Text': ' With empathy as a lantern, wandering through the dark alleys of sorrow, illuminating the path with compassion and care. ', 'Sentiment': ' Empathetic ', 'Timestan
{'': '286', 'Unnamed: 0': '290', 'Text': ' A free spirit soaring on the wings of dreams, leaving trails of independence in the azure sky of boundless possibilities. ', 'Sentiment': ' Free-spirited ', 'Tin
{'': '287', 'Unnamed: 0': '291', 'Text': ' Bathed in the golden hues of gratefulness, a sunset of appreciation casting its warm glow on the landscapes of the heart. ', 'Sentiment': ' Grateful ', 'Timestan
{'': '288', 'Unnamed: 0': '292', 'Text': ' Confident strides in the dance of life, a ballroom where self-assuredness leads, twirling through challenges with grace. ', 'Sentiment': ' Confident ', 'Timestan
{'': '289', 'Unnamed: 0': '293', 'Text': ' Hopeful whispers of wind, carrying the promises of a brighter tomorrow, a symphony of optimism in the air of possibilities. ', 'Sentiment': ' Hopeful ', 'Timesta
{'': '290', 'Unnamed: 0': '294', 'Text': ' Playfully juggling responsibilities, a circus performer balancing the acts of work and joy, tossing laughter into the air. ', 'Sentiment': ' Playful ', 'Timestan
{'': '291', 'Unnamed: 0': '295', 'Text': ' Whispering tales of inspiration to the stars, a storyteller crafting constellations from the threads of imagination. ', 'Sentiment': ' Inspired ', 'Timestamp':
{'': '292', 'Unnamed: 0': '296', 'Text': ' Charting a course through the waves of hopeful anticipation, a sailor steering towards the shores of dreams yet unexplored. ', 'Sentiment': ' Hopeful ', 'Timesta
{'': '293', 'Unnamed: 0': '297', 'Text': ' A compassionate rain, tears of empathy falling gently, nurturing the seeds of kindness in the garden of human connections. ', 'Sentiment': ' Compassionate ', 'Ti
{'': '294', 'Unnamed: 0': '298', 'Text': ' Proudly scaling the peaks of achievement, a mountaineer conquering challenges and planting the flag of success. ', 'Sentiment': ' Proud ', 'Timestamp': '2020-01
{'': '296', 'Unnamed: 0': '300', 'Text': ' A playful escapade in the carnival of life, carousel laughter and cotton candy dreams swirling in the joyous atmosphere. ', 'Sentiment': ' Playful ', 'Timestamp
{'': '332', 'Unnamed: 0': '336', 'Text': " Loneliness, a silent companion in the night, the only echo in the chamber of solitude, a heart's solitary nocturne. ", 'Sentiment': ' Loneliness ', 'Timestamp':
{'': '404', 'Unnamed: 0': '408', 'Text': 'Rejuvenated by the salty breeze and the sound of waves at the seaside. ', 'Sentiment': ' Rejuvenation ', 'Timestamp': '2021-09-20 14:30:00', 'User': ' SeaBreezeLo
{'': '406', 'Unnamed: 0': '410', 'Text': 'Savoring the warmth of a cup of cocoa on a chilly winter evening.  ', 'Sentiment': ' Coziness      ', 'Timestamp': '2020-01-12 18:00:00', 'User': ' WinterWarmth
{'': '407', 'Unnamed: 0': '411', 'Text': 'Heartfelt gratitude for the laughter shared during a family reunion. ', 'Sentiment': ' Gratitude   ', 'Timestamp': '2017-12-28 19:15:00', 'User': ' FamilyGatherer
```

```
# تحميل البيانات مع تجاهل العمود غير المطلوب
data = pd.read_csv('sentimentdataset.csv', index_col=0)

# عرض أول 5 صفوف للتحقق من البيانات
print(dUnnameddata.head())
```

	Unnamed: 0	Text	Sentiment	\
0	0	Enjoying a beautiful day at the park!	...	Positive
1	1	Traffic was terrible this morning.	...	Negative
2	2	Just finished an amazing workout! 🏋️	...	Positive
3	3	Excited about the upcoming weekend getaway!	...	Positive
4	4	Trying out a new recipe for dinner tonight.	...	Neutral

	Timestamp	User	Platform	\
0	2023-01-15 12:30:00	User123	Twitter	
1	2023-01-15 08:45:00	CommuterX	Twitter	
2	2023-01-15 15:45:00	FitnessFan	Instagram	
3	2023-01-15 18:20:00	AdventureX	Facebook	
4	2023-01-15 19:55:00	ChefCook	Instagram	

	Hashtags	Retweets	Likes	Country	\
0	#Nature #Park	15.0	30.0	USA	
1	#Traffic #Morning	5.0	10.0	Canada	
2	#Fitness #Workout	20.0	40.0	USA	
3	#Travel #Adventure	8.0	15.0	UK	
4	#Cooking #Food	12.0	25.0	Australia	

	Year	Month	Day	Hour
0	2023	1	15	12
1	2023	1	15	8
2	2023	1	15	15
3	2023	1	15	18
4	2023	1	15	19

```
# الأعمدة التي سيتم فحص القيم المكررة فيها
columns_to_check = ['Text', 'Sentiment', 'User', 'Platform', 'Hashtags', 'Country', 'Year', 'Month', 'Day', 'Hour']
```

```
# التحقق إذا كانت هناك قيم مكررة في الأعمدة المحددة
has_duplicates = data[columns_to_check].duplicated().any()
```

```
# عرض النتيجة
if has_duplicates:
    print("هناك قيم مكررة.")
else:
    print("لا توجد قيم مكررة.")
```

➡️ هناك قيم مكررة.

```
# إزالة الصفوف المكررة
data_cleaned = data.drop_duplicates(subset=columns_to_check)

# عرض البيانات بعد إزالة المكررات
print(data_cleaned.head())
```

➡️

Unnamed: 0	Text	Sentiment	\
0	Enjoying a beautiful day at the park!	...	Positive
1	Traffic was terrible this morning.	...	Negative
2	Just finished an amazing workout! 🏋️	...	Positive
3	Excited about the upcoming weekend getaway!	...	Positive
4	Trying out a new recipe for dinner tonight.	...	Neutral

	Timestamp	User	Platform	\
0	2023-01-15 12:30:00	User123	Twitter	
1	2023-01-15 08:45:00	CommuterX	Twitter	
2	2023-01-15 15:45:00	FitnessFan	Instagram	
3	2023-01-15 18:20:00	AdventureX	Facebook	
4	2023-01-15 19:55:00	ChefCook	Instagram	

	Hashtags	Retweets	Likes	Country	\
0	#Nature #Park	15.0	30.0	USA	
1	#Traffic #Morning	5.0	10.0	Canada	
2	#Fitness #Workout	20.0	40.0	USA	
3	#Travel #Adventure	8.0	15.0	UK	
4	#Cooking #Food	12.0	25.0	Australia	

	Year	Month	Day	Hour
0	2023	1	15	12
1	2023	1	15	8
2	2023	1	15	15
3	2023	1	15	18
4	2023	1	15	19

```
# التحقق إذا كانت هناك قيم مكررة بعد إزالة المكررات
has_duplicates_after_cleaning = data_cleaned[columns_to_check].duplicated().any()
```

```
# عرض النتيجة
if has_duplicates_after_cleaning:
    print("لا يزال هناك قيم مكررة.")
else:
    print("لا توجد قيم مكررة.")
```

⌕ لا توجد قيم مكررة

```
missing_values = df.isnull().sum()
print(missing_values)
```

⌕ Unnamed: 0.1 0
Unnamed: 0 0
Text 0
Sentiment 0
Timestamp 0
User 0
Platform 0
Hashtags 0
Retweets 0
Likes 0
Country 0
Year 0
Month 0
Day 0
Hour 0
dtype: int64

```
# التحقق من القيم المفقودة في الأعمدة
missing_values = data.isnull().sum()
```

```
# عرض الأعمدة التي تحتوي على قيم مفقودة
print(missing_values)
```

⌕ Unnamed: 0 0
Text 0
Sentiment 0
Timestamp 0
User 0
Platform 0
Hashtags 0
Retweets 0
Likes 0
Country 0
Year 0
Month 0
Day 0
Hour 0
dtype: int64

```
# التأكد من نوع البيانات في الأعمدة
print(data.dtypes)
```

⌕ Unnamed: 0 int64
Text object
Sentiment object
Timestamp object
User object
Platform object
Hashtags object
Retweets float64
Likes float64

```
Country      object
Year         int64
Month        int64
Day          int64
Hour         int64
dtype: object
```

```
from transformers import pipeline

# تحميل نموذج تحليل المشاعر بشكل صريح
sentiment_analysis = pipeline("sentiment-analysis", model="distilbert-base-uncased-finetuned-sst-2-english")

# تطبيق التحليل على النصوص
df['Sentiment'] = df['Text'].apply(lambda x: sentiment_analysis(x)[0]['label'])

# عرض النتائج
print(df[['Text', 'Sentiment']].head())
```

➡ Device set to use cpu

	Text	Sentiment
0	Enjoying a beautiful day at the park!	... POSITIVE
1	Traffic was terrible this morning.	... NEGATIVE
2	Just finished an amazing workout! 🏋️	... POSITIVE
3	Excited about the upcoming weekend getaway!	... POSITIVE
4	Trying out a new recipe for dinner tonight.	... NEGATIVE

```
# حساب توزيع المشاعر
sentiment_counts = df['Sentiment'].value_counts()

# طباعة النتائج
print("Sentiment Distribution:")
print(sentiment_counts)
```

➡ Sentiment Distribution:

Sentiment	
POSITIVE	570
NEGATIVE	162

Name: count, dtype: int64

```
# حساب توزيع المشاعر حسب المنصة
sentiment_by_platform = df.groupby('Platform')['Sentiment'].value_counts()

# طباعة النتائج
print("Sentiment by Platform:")
print(sentiment_by_platform)
```

➡ Sentiment by Platform:

Platform	Sentiment	
Facebook	POSITIVE	180
	NEGATIVE	51
Instagram	POSITIVE	201

```

        NEGATIVE      57
Twitter    POSITIVE    102
           NEGATIVE     26
Twitter    POSITIVE     87
           NEGATIVE     28
Name: count, dtype: int64
```

```
# حساب توزيع المشاعر حسب البلد
sentiment_by_country = df.groupby('Country')['Sentiment'].value_counts()
```

```
# طباعة النتائج
print("Sentiment by Country:")
print(sentiment_by_country)
```

```
🔗 Sentiment by Country:
Country      Sentiment
Australia    POSITIVE     30
              NEGATIVE     11
Australia    POSITIVE      2
Australia    POSITIVE      4
              NEGATIVE      1
              ..
USA          NEGATIVE      1
USA          POSITIVE      4
USA          POSITIVE      8
              NEGATIVE      1
USA          POSITIVE      9
Name: count, Length: 148, dtype: int64
```

```
# تحليل العلاقة بين الإعجابات والمشاعر
sentiment_vs_likes = df.groupby('Sentiment')['Likes'].mean()
```

```
# طباعة النتائج
print("Average Likes for Each Sentiment:")
print(sentiment_vs_likes)
```

```
🔗 Average Likes for Each Sentiment:
Sentiment
NEGATIVE    34.530864
POSITIVE    45.280702
Name: Likes, dtype: float64
```

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
```

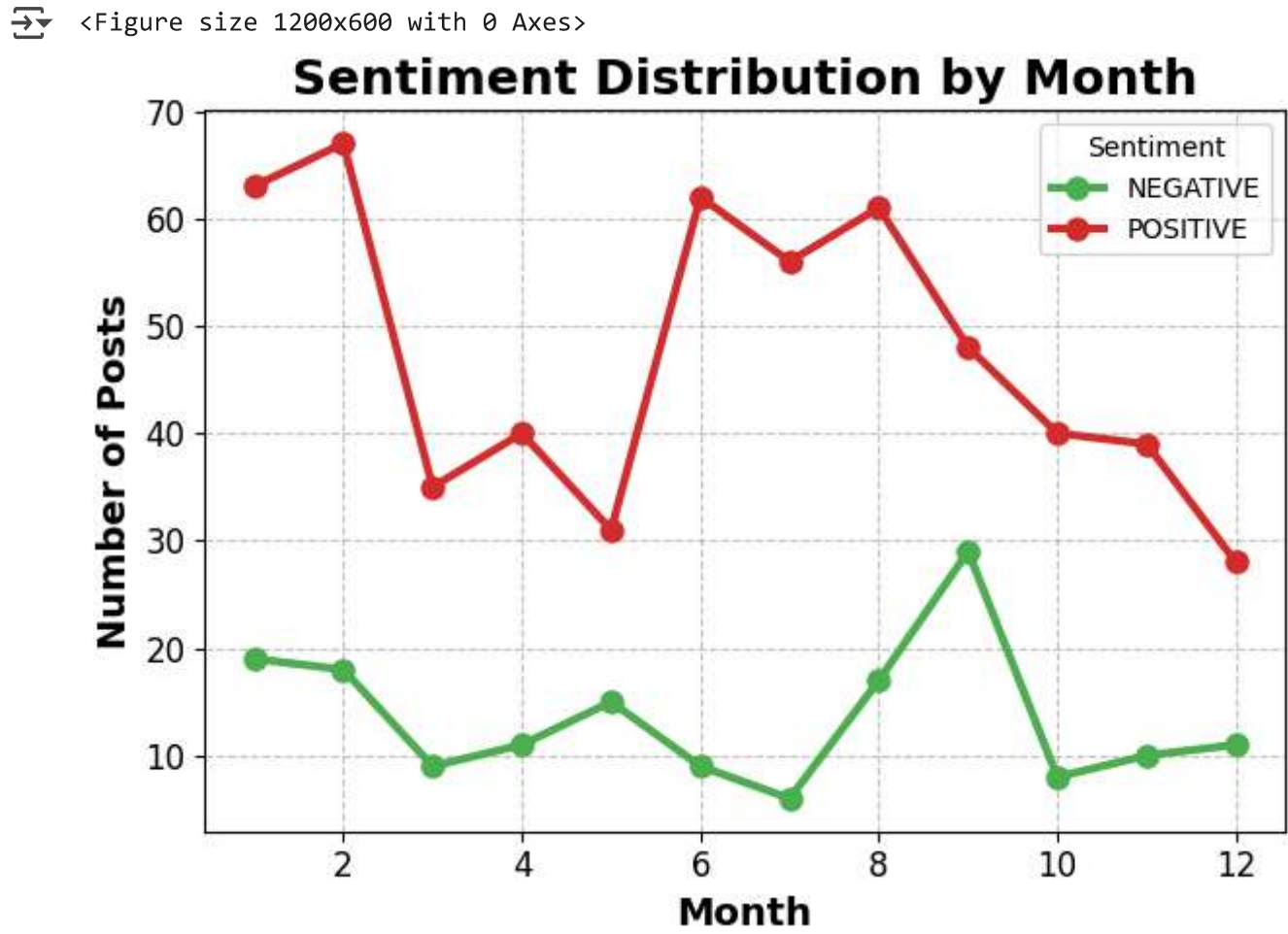
```
# كلمات من المنشورات الإيجابية
positive_text = ' '.join(df[df['Sentiment'] == 'POSITIVE']['Text'])
wordcloud_pos = WordCloud(width=800, height=200, background_color='white', colormap='Blues').generate(positive_text)
```

```
# عرض السحابة للكلمات الإيجابية
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 3))
```



```
plt.ylabel('Number of Posts', fontsize=14, fontweight='bold')
plt.xticks(rotation=0, fontsize=12)
plt.yticks(fontsize=12)
plt.grid(True, linestyle='--', alpha=0.7)

plt.tight_layout()
plt.show()
```



```
import numpy as np
import matplotlib.pyplot as plt

# التأكد من تنسيق القيم بشكل صحيح
df['Sentiment'] = df['Sentiment'].str.strip().str.capitalize()

# تجميع البيانات حسب الساعة ونوع المشاعر
sentiment_by_hour = df.groupby(['Hour', 'Sentiment']).size().unstack(fill_value=0)

# التأكد من تضمين جميع أنواع المشاعر حتى لو كانت غير موجودة
for sentiment in ['Positive', 'Negative', 'Neutral']:
    if sentiment not in sentiment_by_hour.columns:
        sentiment_by_hour[sentiment] = 0

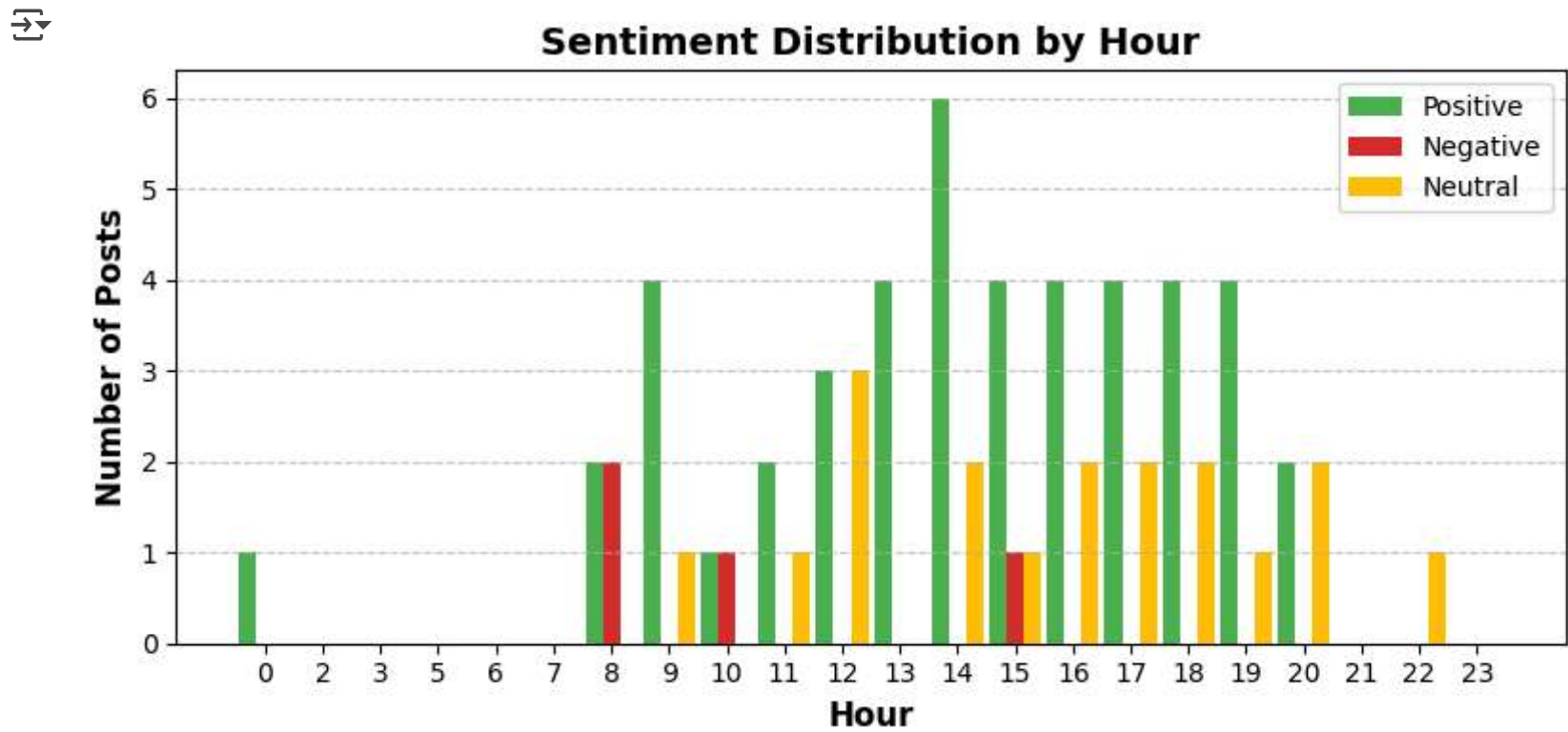
# ترتيب الأعمدة بشكل ثابت
sentiment_by_hour = sentiment_by_hour[['Positive', 'Negative', 'Neutral']]

# إعداد الرسم البياني بحجم أصغر
x = np.arange(len(sentiment_by_hour))
width = 0.3
```



```
plt.figure(figsize=(8, 4)) # تقليل حجم الرسم
plt.bar(x - width, sentiment_by_hour['Positive'], width, label='Positive', color='#4CAF50')
plt.bar(x, sentiment_by_hour['Negative'], width, label='Negative', color='#D32F2F')
plt.bar(x + width, sentiment_by_hour['Neutral'], width, label='Neutral', color='#FFC107')

# تخصيص المحاور والعناوين
plt.xlabel('Hour', fontsize=12, fontweight='bold')
plt.ylabel('Number of Posts', fontsize=12, fontweight='bold')
plt.title('Sentiment Distribution by Hour', fontsize=14, fontweight='bold', color='black')
plt.xticks(ticks=x, labels=sentiment_by_hour.index, rotation=0, fontsize=10)
plt.legend(fontsize=10)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



```
import matplotlib.pyplot as plt
import numpy as np

# حساب متوسط التفاعل (الإعجابات وإعادة التغريد) لكل نوع من المشاعر
sentiment_engagement = df.groupby('Sentiment')[['Likes', 'Retweets']].mean()

# التأكد من ترتيب المشاعر بشكل ثابت
sentiment_engagement = sentiment_engagement.reindex(['Positive', 'Negative', 'Neutral'])

# إعداد الألوان لكل نوع مشاعر
colors = ['#4CAF50', '#D32F2F', '#FFC107']

# إعداد الرسم البياني
x = np.arange(len(sentiment_engagement))
width = 0.3
```

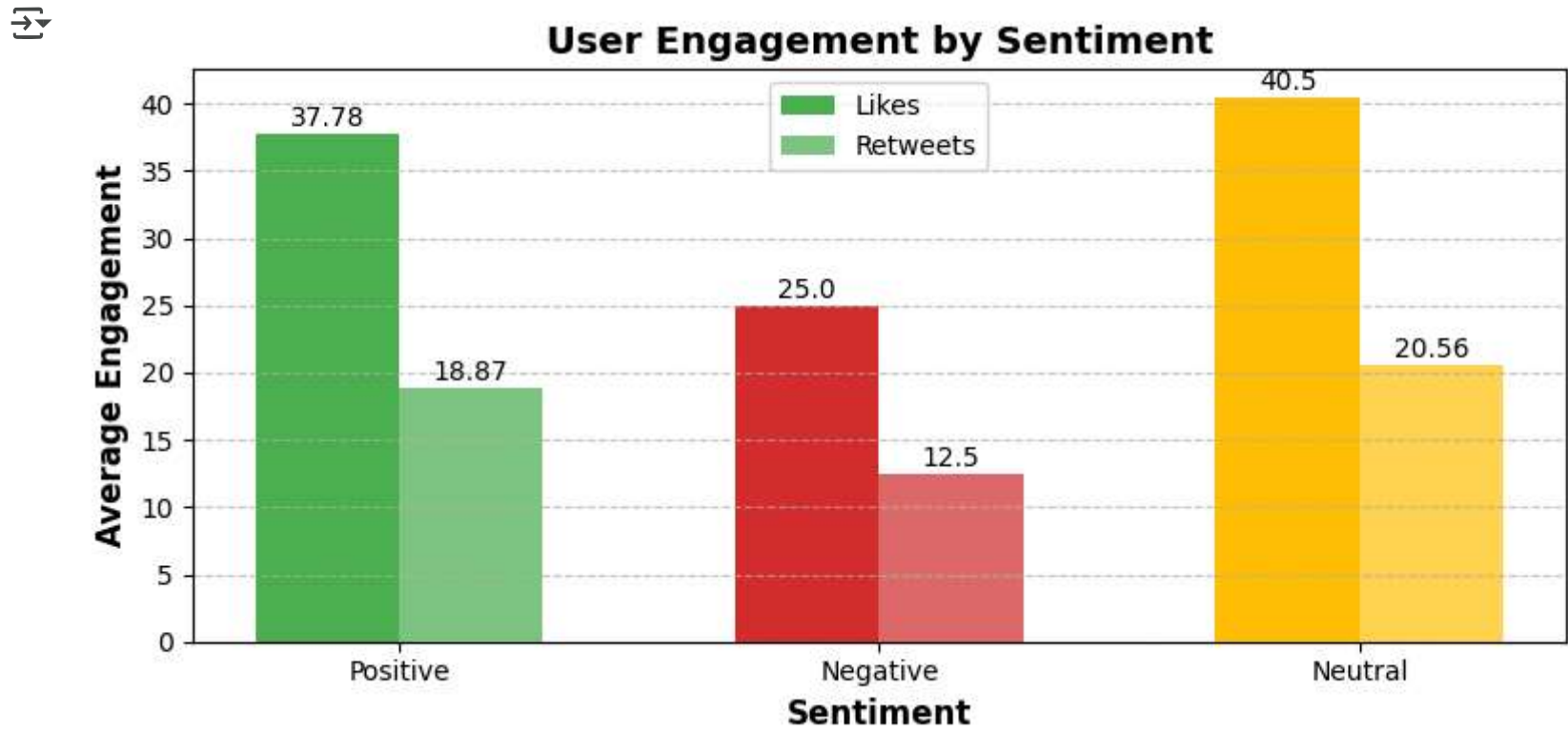
```
plt.figure(figsize=(8, 4)) # تصغير حجم الرسم
```

```
bars_likes = plt.bar(x - width/2, sentiment_engagement['Likes'], width, label='Likes', color=colors)
bars_retweets = plt.bar(x + width/2, sentiment_engagement['Retweets'], width, label='Retweets', color=colors, alpha=0.7)

# إضافة الأرقام فوق الأعمدة
for bar in bars_likes:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval + 0.1, round(yval, 2), ha='center', va='bottom', fontsize=10)

for bar in bars_retweets:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval + 0.1, round(yval, 2), ha='center', va='bottom', fontsize=10)

# تخصيص المحاور والعناوين
plt.xlabel('Sentiment', fontsize=12, fontweight='bold')
plt.ylabel('Average Engagement', fontsize=12, fontweight='bold')
plt.title('User Engagement by Sentiment', fontsize=14, fontweight='bold', color='black')
plt.xticks(ticks=x, labels=sentiment_engagement.index, fontsize=10)
plt.legend(fontsize=10)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



```
import matplotlib.pyplot as plt
from collections import Counter
import pandas as pd
import re

# اختر النصوص السلبية والإيجابية
negative_texts = data_cleaned[data_cleaned['Sentiment'] == 'Negative']['Text']
positive_texts = data_cleaned[data_cleaned['Sentiment'] == 'Positive']['Text']

# دالة لتنظيف النصوص
```

```
def clean_text(text):
    text = text.lower()
    text = re.sub(r'\s+', ' ', text) # إزالة المسافات الزائدة
    text = re.sub(r'^\w\s|$', '', text) # إزالة العلامات الخاصة
    return text

# تنظيف النصوص السلبية والإيجابية
clean_negative_texts = [clean_text(text) for text in negative_texts]
clean_positive_texts = [clean_text(text) for text in positive_texts]

# دمج النصوص السلبية والإيجابية
negative_words = ' '.join(clean_negative_texts).split()
positive_words = ' '.join(clean_positive_texts).split()

# حساب تكرار الكلمات
negative_word_counts = Counter(negative_words)
positive_word_counts = Counter(positive_words)

# الحصول على أكثر 10 كلمات تكرارًا
top_negative_words = negative_word_counts.most_common(10)
top_positive_words = positive_word_counts.most_common(10)

# رسم الكلمات الأكثر شيوعًا في النصوص السلبية (أفقي)
plt.figure(figsize=(6, 4)) # حجم أصغر
plt.barh([word[0] for word in top_negative_words],
         [word[1] for word in top_negative_words],
         color='darkred', edgecolor='black', height=0.7)

# إضافة النسب المئوية
total_negative = sum([word[1] for word in top_negative_words])
for i, word in enumerate(top_negative_words):
    plt.text(word[1] + 0.2, i, f' {(word[1]/total_negative)*100:.1f}%',
             va='center', fontweight='bold', fontsize=10, color='black')

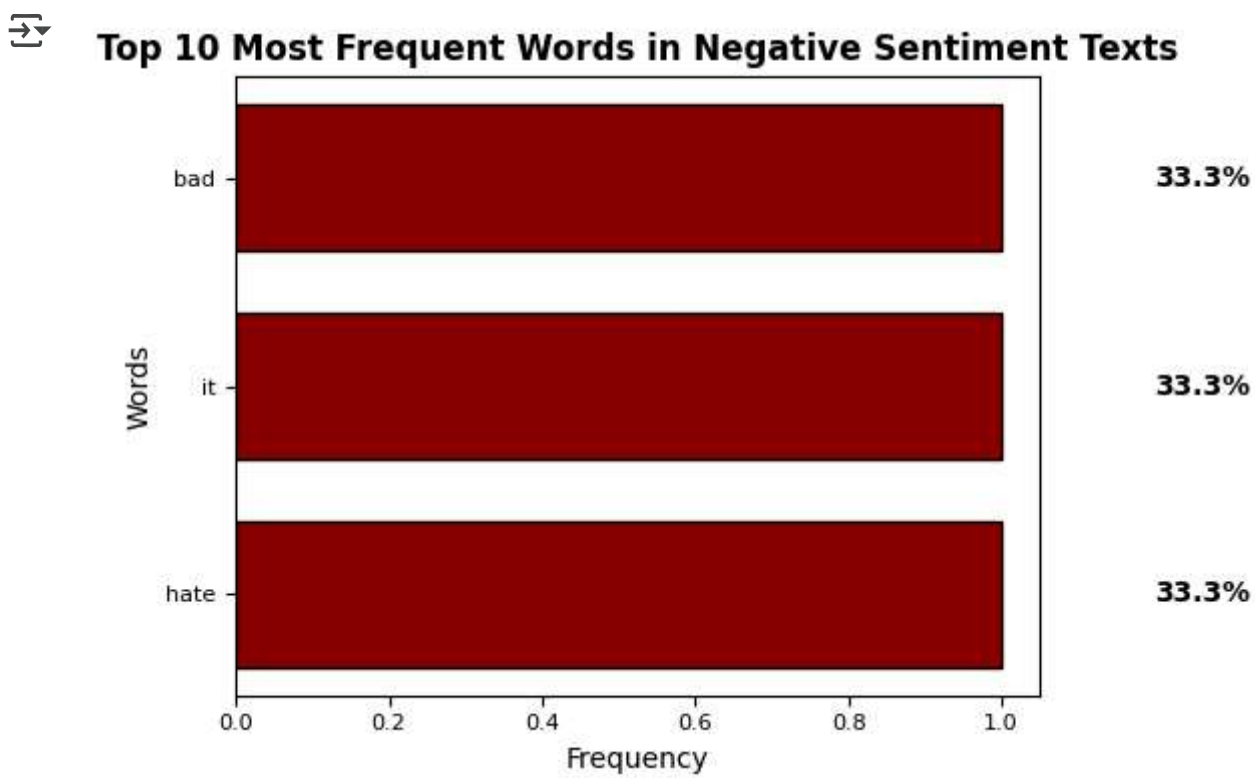
plt.title('Top 10 Most Frequent Words in Negative Sentiment Texts', fontsize=12, fontweight='bold')
plt.xlabel('Frequency', fontsize=10)
plt.ylabel('Words', fontsize=10)
plt.xticks(fontsize=8)
plt.yticks(fontsize=8)
plt.tight_layout() # لتحسين العرض
plt.show()

# رسم الكلمات الأكثر شيوعًا في النصوص الإيجابية (أفقي)
plt.figure(figsize=(6, 4)) # حجم أصغر
plt.barh([word[0] for word in top_positive_words],
         [word[1] for word in top_positive_words],
         color='darkgreen', edgecolor='black', height=0.7)

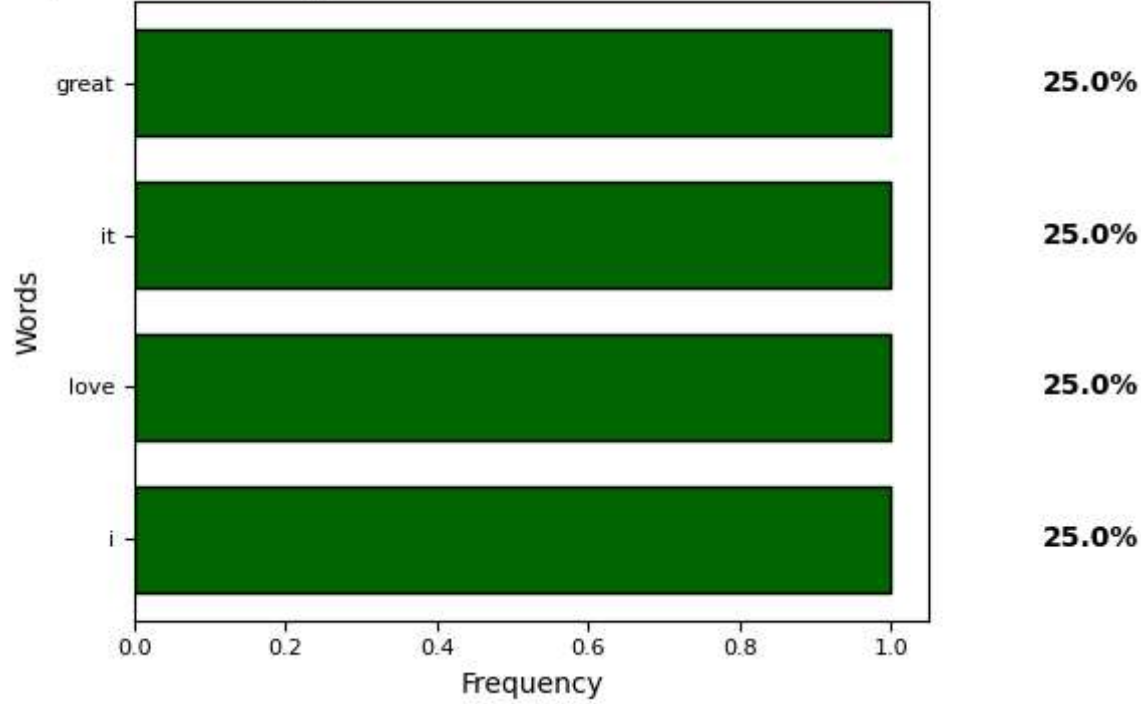
# إضافة النسب المئوية
total_positive = sum([word[1] for word in top_positive_words])
for i, word in enumerate(top_positive_words):
    plt.text(word[1] + 0.2, i, f' {(word[1]/total_positive)*100:.1f}%',
             va='center', fontweight='bold', fontsize=10, color='black')

plt.title('Top 10 Most Frequent Words in Positive Sentiment Texts', fontsize=12, fontweight='bold')
```

```
plt.xlabel('Frequency', fontsize=10)
plt.ylabel('Words', fontsize=10)
plt.xticks(fontsize=8)
plt.yticks(fontsize=8)
plt.tight_layout() # لتحسين العرض
plt.show()
```



Top 10 Most Frequent Words in Positive Sentiment Texts



```
import matplotlib.pyplot as plt
```

```
# تحديد الكلمات السلبية الأكثر شيوعًا
negative_words = ['Fear', 'Sadness', 'Anger', 'Frustration', 'Loneliness', 'Grief']
counts = [negative_words.count(word) for word in negative_words]
```

```
# إنشاء رسم بياني دائري
plt.figure(figsize=(6, 6)) # جعل الرسم البياني أكثر تناسُبًا
plt.pie(counts, labels=negative_words, autopct='%1.1f%%', startangle=140, colors=['#FF6666', '#FF6347', '#FF4500', '#FF1493', '#DC143C', '#B22222'])

# إضافة العنوان
plt.title('Sentiment Words Distribution in Negative Texts')

# عرض الرسم البياني
plt.axis('equal') # لضمان أن الشكل دائري
plt.show()
```

