In [2]:
```
import pandas as pd
```

In [3]:
```
df3=pd.read_csv("C:/Users/sujit/Downloads/twitter_training.csv")
```

In [4]:
```
df3
```

Out[4]:

	Tweet ID	Entity	Sentiment	Tweet Content
0	2401	Borderlands	Positive	I am coming to the borders and I will kill you...
1	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...
2	2401	Borderlands	Positive	im coming on borderlands and i will murder you...
3	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder ...
4	2401	Borderlands	Positive	im getting into borderlands and i can murder y...
...	...	...	...	...
74676	9200	Nvidia	Positive	Just realized that the Windows partition of my...
74677	9200	Nvidia	Positive	Just realized that my Mac window partition is ...
74678	9200	Nvidia	Positive	Just realized the windows partition of my Mac ...
74679	9200	Nvidia	Positive	Just realized between the windows partition of...
74680	9200	Nvidia	Positive	Just like the windows partition of my Mac is l...

74681 rows × 4 columns

[8]:
```
df31
```

t[8]:

	Tweet ID	Entity	Sentiment	Tweet Content
0	352	Amazon	Neutral	BBC News - Amazon boss Jeff Bezos rejects clai...
1	8312	Microsoft	Negative	@Microsoft Why do I pay for WORD when it funct...
2	4371	CS-GO	Negative	CSGO matchmaking is so full of closet hacking,...
3	4433	Google	Neutral	Now the President is slapping Americans in the...
4	6273	FIFA	Negative	Hi @EAHelp I've had Madeleine McCann in my cel...
...	...	...	...	...
994	4891	GrandTheftAuto(GTA)	Irrelevant	⭐ Toronto is the arts and culture capital of ...
995	4359	CS-GO	Irrelevant	tHIS IS ACTUALLY A GOOD MOVE TOT BRING MORE VI...
996	2652	Borderlands	Positive	Today sucked so it's time to drink wine n play...
997	8069	Microsoft	Positive	Bought a fraction of Microsoft today. Small wins.
998	6960	johnson&johnson	Neutral	Johnson & Johnson to stop selling talc baby po...

999 rows × 4 columns

[9]:
```
df31.head()
```

t[9]:

	Tweet ID	Entity	Sentiment	Tweet Content
0	352	Amazon	Neutral	BBC News - Amazon boss Jeff Bezos rejects clai...
1	8312	Microsoft	Negative	@Microsoft Why do I pay for WORD when it funct...
2	4371	CS-GO	Negative	CSGO matchmaking is so full of closet hacking,...
3	4433	Google	Neutral	Now the President is slapping Americans in the...
4	6273	FIFA	Negative	Hi @EAHelp I've had Madeleine McCann in my cel...

In [10]:
```
vectorizer = TfidfVectorizer(max_features=10000)
X_train = vectorizer.fit_transform(df3["Sentiment"])
X_val = vectorizer.transform(df31["Sentiment"])


model = MultinomialNB(alpha=1.5)
model.fit(X_train, df3['Sentiment'])
```

Out[10]:
```
MultinomialNB
MultinomialNB(alpha=1.5)
```

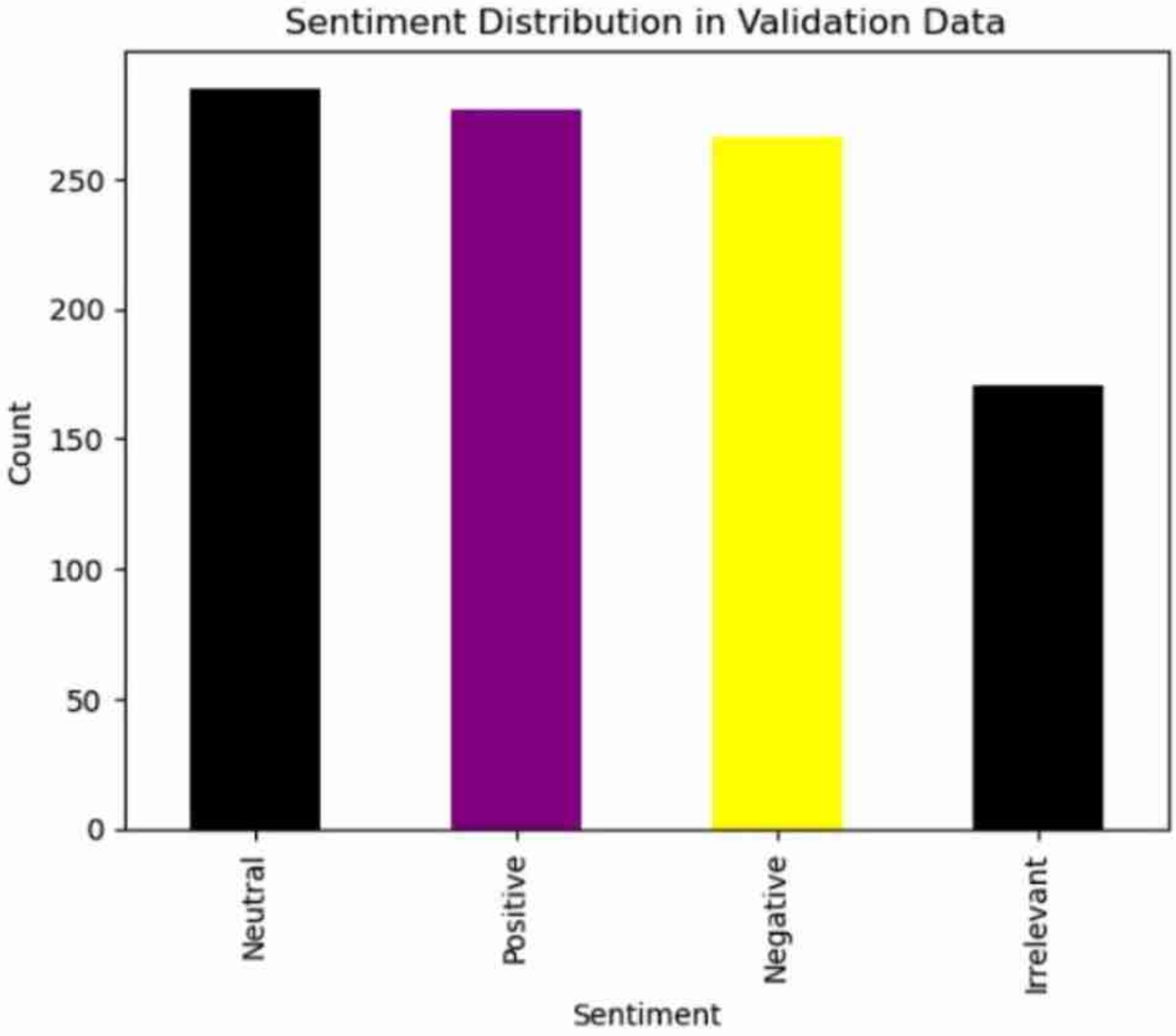
In [12]:
```
# Prediction on the validation set
predictions = model.predict(X_val)

from sklearn.metrics import f1_score

f1 = f1_score(df31['Sentiment'], predictions,average='micro')
print(f'F1 Score: {f1:.2f}')
```

```
F1 Score: 1.00
```

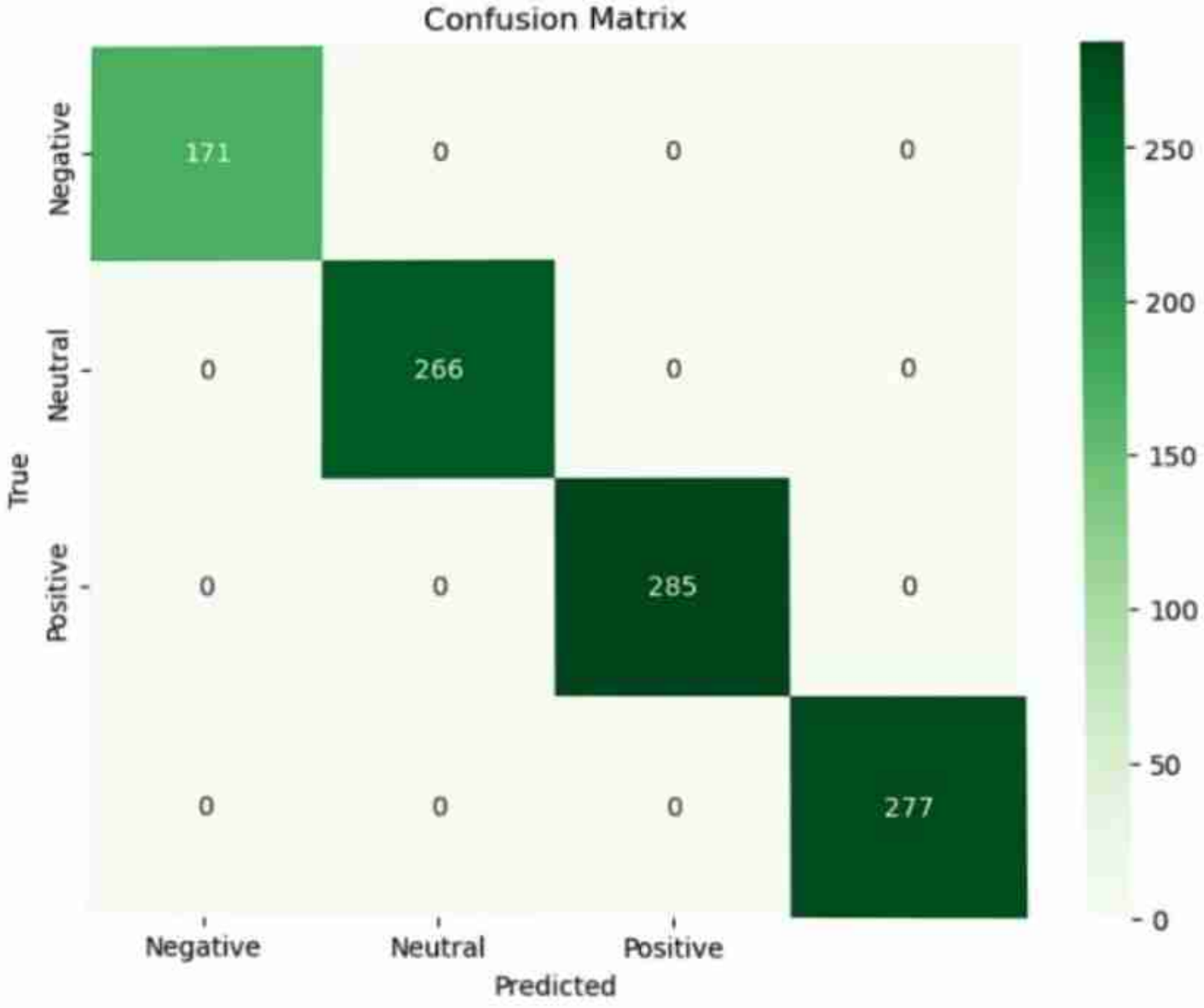
In [14]:
```
# Print classification report and confusion matrix
print('Classification Report:\n', classification_report(df31['Sentiment'], predictions))
print('Confusion Matrix:\n', confusion_matrix(df31['Sentiment'], predictions))
```

```
Classification Report:
               precision    recall  f1-score   support

  Irrelevant       1.00      1.00      1.00       171
    Negative       1.00      1.00      1.00       266
     Neutral       1.00      1.00      1.00       285
    Positive       1.00      1.00      1.00       277

    accuracy                           1.00       999
   macro avg       1.00      1.00      1.00       999
weighted avg       1.00      1.00      1.00       999

Confusion Matrix:
 [[171   0   0   0]
 [  0 266   0   0]
 [  0   0 285   0]
 [  0   0   0 277]]
```

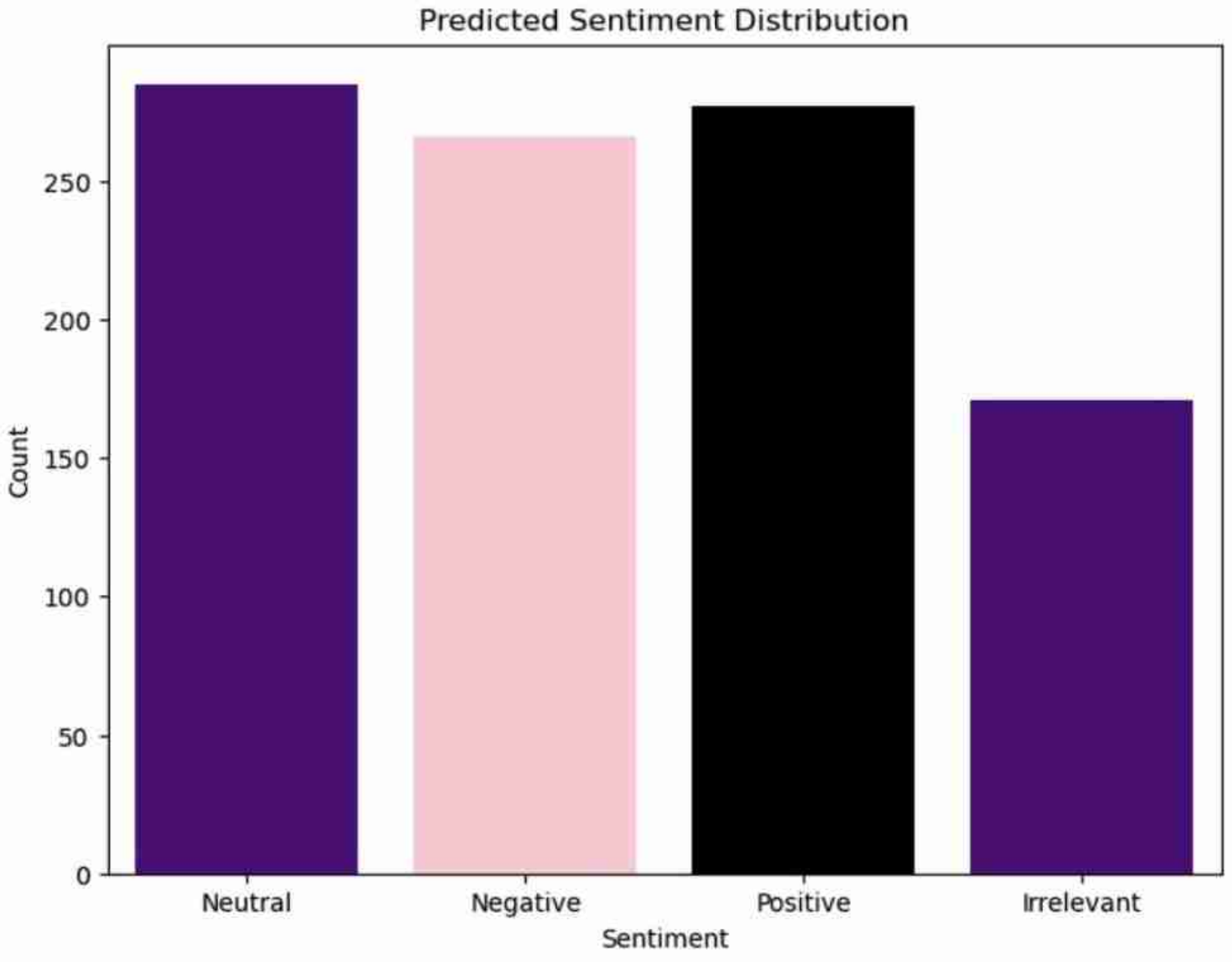
[16]:
```
# Visualization - sentiment distribution
sentiment_distribution = df31['Sentiment'].value_counts()
sentiment_distribution.plot(kind='bar', color=['black', 'purple', 'yellow'])
plt.title('Sentiment Distribution in Validation Data')
plt.xlabel('Sentiment')
plt.ylabel('Count')
plt.show()
```



[17]:
```
# Additional Analysis and Visualizations
# Visualize Confusion Matrix
plt.figure(figsize=(8, 6))
sns.heatmap(confusion_matrix(df31['Sentiment'], predictions), annot=True, fmt='d', cmap='Greens',
            xticklabels=['Negative', 'Neutral', 'Positive'], yticklabels=['Negative', 'Neutral', 'Positive'])
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.show()
```



In [18]:
```
plt.figure(figsize=(8, 6))
sns.countplot(x=predictions, palette=['indigo', 'pink', 'black'])
plt.title('Predicted Sentiment Distribution')
plt.xlabel('Sentiment')
plt.ylabel('Count')
plt.show()
```

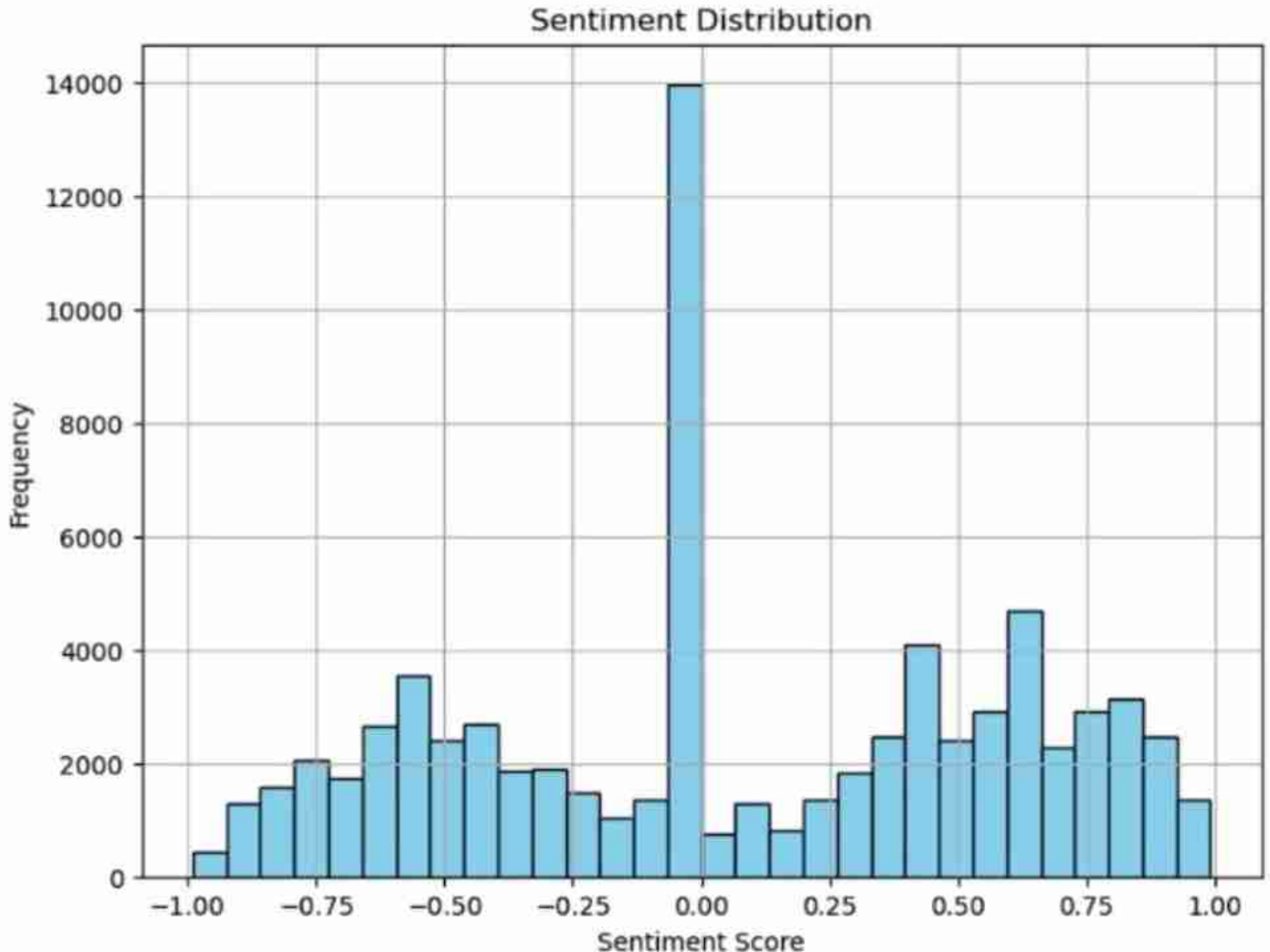


In [23]:
```
import pandas as pd
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import string
```

In [27]:
```
# Initialize Sentiment Intensity Analyzer
sid = SentimentIntensityAnalyzer()

# Function to preprocess text
def preprocess_text(text):
    if isinstance(text, str):  # Check if text is a string
        # Tokenize text
        tokens = word_tokenize(text)
        # Remove punctuation
        tokens = [word for word in tokens if word.isalpha()]
        # Convert to lowercase
        tokens = [word.lower() for word in tokens]
        # Remove stopwords
        stop_words = set(stopwords.words('english'))
        tokens = [word for word in tokens if not word in stop_words]
        return ' '.join(tokens)
    else:
        return ''  # Return empty string if text is not a string
```

```
df3['clean_text'] = df3['Tweet Content'].apply(preprocess_text)
df3['sentiment_score'] = df3['clean_text'].apply(lambda x: sid.polarity_scores(x)['compound'])
```

n [38]:
```
# Plot sentiment distribution
plt.figure(figsize=(8, 6))
plt.hist(df3['sentiment_score'], bins=30, color='skyblue', edgecolor='black')
plt.title('Sentiment Distribution')
plt.xlabel('Sentiment Score')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```



```
# Convert 'text' column to string type and handle missing values
df3['Tweet Content'] = df3['Tweet Content'].astype(str)

# Ensure 'text' column contains only string values
df3['Tweet Content'] = df3['Tweet Content'].apply(lambda x: x if isinstance(x, str) else "")
```

```
# Function to get sentiment scores
def get_sentiment_score(text):
    return sid.polarity_scores(text)['compound']

# Apply sentiment analysis
df3['sentiment_score'] = df3['Tweet Content'].apply(get_sentiment_score)

# Generate word cloud of positive and negative sentiments
positive_text = ' '.join(df3[df3['sentiment_score'] > 0]['Tweet Content'])
negative_text = ' '.join(df3[df3['sentiment_score'] < 0]['Tweet Content'])
```

```
plt.figure(figsize=(15, 10))

plt.subplot(1, 2, 1)
wordcloud_positive = WordCloud(width=800, height=400, background_color='white').generate(positive_text)
plt.imshow(wordcloud_positive, interpolation='bilinear')
plt.title('Positive Sentiment Word Cloud')
plt.axis('off')

plt.subplot(1, 2, 2)
wordcloud_negative = WordCloud(width=800, height=400, background_color='white').generate(negative_text)
plt.imshow(wordcloud_negative, interpolation='bilinear')
plt.title('Negative Sentiment Word Cloud')
plt.axis('off')

plt.show()
```