

[https://doi.org/ 10.33472/AFJBS.6.9.2024.950-955](https://doi.org/10.33472/AFJBS.6.9.2024.950-955)



## African Journal of Biological Sciences

Journal homepage: <http://www.afjbs.com>



Research Paper

Open Access

# Sentiment Analysis of Online Comments Using Vader - Valence Aware Dictionary for Sentiment Reasoning)

Nehaa Shree S A, Student, Bannari Amman Institute of Technology  
Sasikala D, Professor, Bannari Amman Institute of Technology  
Nivishna Shree , Student, Bannari Amman Institute of Technology

Article History

Volume 6, Issue 9, 2024

Received: 21 Mar 2024

Accepted : 28 Apr 2024

doi: 10.33472/AFJBS.6.9.2024.950-955

### Abstract

This study uses qualitative linguistics tools to examine online language perception. This file contains reviews from specialized online websites. Sentiment analysis was performed using the VADER (Valence-Aware Dictionary and Sentiment Reasoning) tool, which provides high scores for each analysis. This study investigates the distribution of sentiment scores and their relationship to the number of likes a message receives. It provides a user interface to access comments and get the desired results.

### Introduction

In today's technological era, online platforms serve as crucial hubs for communication and expression, fostering the exchange of opinions, feedback, and thoughts among users. Understanding the sentiment conveyed in online comments holds significant importance across various domains, including brand perception analysis, customer feedback evaluation, and public opinion monitoring. Sentiment analysis, a subset of Natural Language Processing (NLP), provides tools to analyze and quantify sentiment in textual data automatically.

This project focuses on sentiment analysis on a dataset of online comments using Python libraries NLTK and Pandas. The dataset comprises textual comments sourced from diverse online platforms such as social media, forums, and review websites. The project aims to preprocess the text data, calculate sentiment scores for each comment, conduct exploratory data analysis (EDA) to discern sentiment distribution trends and implement real-time sentiment analysis for user input.

The methodology involves loading the dataset into a Pandas DataFrame, where each comment is treated as a separate observation. Text preprocessing techniques, including lowercase

conversion, punctuation removal, and stopword elimination, are applied using NLTK's English stopwords corpus. Sentiment analysis is performed using the VADER (Valence Aware Dictionary and Sentiment Reasoner) tool, tailored for social media text sentiment analysis.

Exploratory data analysis techniques, such as calculating average sentiment scores and visualizing sentiment score distributions through histograms and boxplots, provide insights into sentiment distribution patterns within the dataset. Additionally, a real-time sentiment analysis function enables users to input comments interactively, with immediate feedback provided on whether the input comment appears positive, negative, or neutral.

In essence, this project showcases the practical application of sentiment analysis using NLTK and Pandas, offering valuable insights into the sentiment expressed on online platforms. It illustrates the potential of real-time sentiment analysis in diverse digital platforms and applications, facilitating informed decision-making and enhancing user engagement experiences.

## **Methodology**

### **1. Data Loading**

- The dataset containing online comments is loaded using the Pandas library. The dataset is assumed to be in CSV format and is read into a data frame for further processing.

### **2. Text Data Cleaning**

- The `clean_text` function is implemented to preprocess the comment text. This function performs the following steps:
  - Converts text to lowercase to ensure uniformity.
  - Removes punctuation and special characters using regular expressions.
  - Removes stopwords to eliminate common words that do not contribute significantly to the sentiment.

### **3. Stopwords Removal**

- Stopwords are common words in a language (e.g., "the," "is," "and") that do not carry much meaning and are often removed during text analysis. In this project, NLTK's English stopwords are loaded and used for removing stopwords from comment text.

### **4. Sentiment Analysis**

- The VADER (Valence Aware Dictionary and Sentiment Reasoner) sentiment analyzer from the NLTK library is utilized for sentiment analysis. VADER is specifically designed for sentiment analysis of social media text, including online comments. It provides a compound score that represents the overall sentiment polarity of the text.

### **5. Sentiment Score Calculation**

- The `get_sentiment` function is implemented to calculate the sentiment score for each comment. This function utilizes the VADER analyzer to obtain the compound score, which is then returned as the sentiment score.

### **6. Adding Sentiment Score to Dataset**

- The calculated sentiment scores are added as a new column (Sentiment Score) to the DataFrame containing the comment data.

## 7. Exploratory Data Analysis (EDA)

- EDA is performed to gain insights into the sentiment distribution within the dataset.
  - The average sentiment score across all comments is calculated to understand the overall sentiment tendency.
  - A histogram is plotted to visualize the distribution of sentiment scores, providing an overview of sentiment polarity.
  - Boxplots are generated to explore the sentiment distribution based on the number of likes received by comments, indicating potential relationships between sentiment and user engagement.

## 8. User Input and Sentiment Analysis

- A function named `analyze_user_input` is defined to allow users to input their comments for real-time sentiment analysis.
  - The user's input is cleaned using the same preprocessing steps as applied to the dataset.
  - Sentiment analysis is performed on the cleaned comment text using the `get_sentiment` function.
  - Feedback is provided to the user based on the sentiment score, indicating whether the comment appears positive, negative, or neutral.

## Results

1. **Average Sentiment Score:** Calculated the average sentiment score across all comments, providing an overall understanding of the sentiment tendency in the dataset.
2. **Distribution of Sentiment Scores:** Visualized sentiment score distribution via a histogram, offering insights into polarity and sentiment distribution within the dataset.
3. **Sentiment Distribution by Likes:** Examined sentiment score distribution based on comment likes using boxplots, helping understand sentiment engagement relationships.
4. **Real-time User Input Analysis:** Implemented interactive user input for real-time sentiment analysis, providing immediate feedback on the sentiment of input comments.

These results offer insights into online comment sentiment, showcasing NLTK and Pandas' application in sentiment analysis and aiding understanding of sentiment tendencies in online discussions. The real-time analysis feature allows instant feedback, beneficial for social media monitoring and customer feedback analysis.

## Discussion

### 1. Outcomes:

- Average Sentiment Score: Quantitative measure of overall sentiment tendency.
- Sentiment Score Distribution: Insights into polarity and prevalence of sentiments.
- Sentiment Distribution by Likes: Analysis of correlations between sentiment and user engagement.

### 2. Implications:

- Understanding Public Opinion: Valuable insights for gauging sentiment surrounding products or initiatives.

- Brand Perception and Reputation: Important for managing brand perception and addressing concerns.
- Customer Feedback Analysis: Efficient prioritization of customer issues and enhancement of satisfaction.

### 3. Limitations:

- Subjectivity and Context: Challenges with nuances and cultural references.
- Lexicon Dependency: Reliance on predefined lexicons may limit accuracy.
- Data Quality and Noise: Impact of data cleanliness and relevance on accuracy.

### 4. Future Directions:

- Advanced Techniques: Explore machine learning for improved accuracy.
- Domain-Specific Customization: Tailor models for specific contexts or domains.
- Integration with Feedback Systems: Enhance real-time monitoring for prompt responses.

## Observations

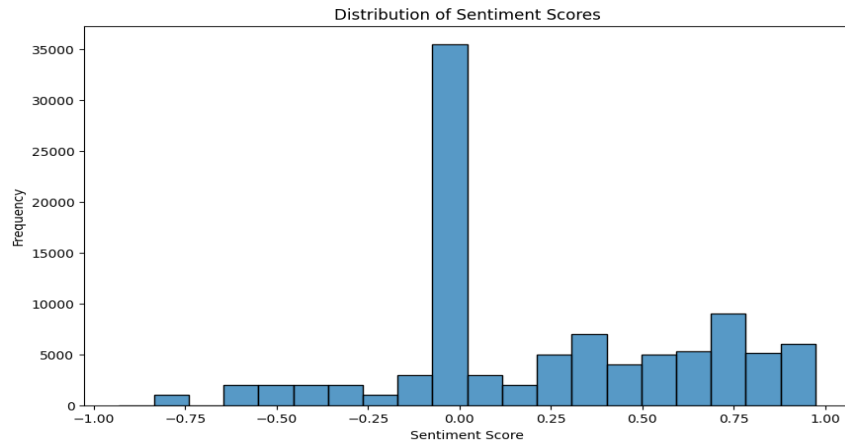
▶ data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 5 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Comment                100000 non-null object
1   Anonymized Author      100000 non-null object
2   Published At           100000 non-null object
3   Likes                  100000 non-null int64
4   Reply Count            100000 non-null int64
dtypes: int64(2), object(3)
memory usage: 3.8+ MB
```

▶ # Average sentiment score

```
avg_sentiment = data["Sentiment Score"].mean()
print(f"Average Sentiment Score: {avg_sentiment:.2f}")
```

➡ Average Sentiment Score: 0.23



```
[ ] # User Input and Sentiment Analysis
```

```
def analyze_user_input():  
    user_comment = input("Enter your comment: ")  
    cleaned_comment = clean_text(user_comment)  
    sentiment = get_sentiment(cleaned_comment)  
    if sentiment > 0.05:  
        print("Your comment seems positive!")  
    elif sentiment < -0.05:  
        print("Your comment seems negative.")  
    else:  
        print("Your comment seems neutral.")  
  
analyze_user_input()
```

```
Enter your comment: nice  
Your comment seems positive!
```



### # User Input and Sentiment Analysis

```
def analyze_user_input():
    user_comment = input("Enter your comment: ")
    cleaned_comment = clean_text(user_comment)
    sentiment = get_sentiment(cleaned_comment)
    if sentiment > 0.05:
        print("Your comment seems positive!")
    elif sentiment < -0.05:
        print("Your comment seems negative.")
    else:
        print("Your comment seems neutral.")

analyze_user_input()
```



```
Enter your comment: I don't think it looks good on you
Your comment seems negative.
```

## References

1. NLTK Library Documentation: <https://www.nltk.org/>
2. Pandas Library Documentation: <https://pandas.pydata.org/docs/>
3. VADER Sentiment Analysis Tool Documentation: <https://github.com/cjhutto/vaderSentiment>
4. Python Regular Expressions (re) Documentation: <https://docs.python.org/3/library/re.html>