



IJIRCCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 3, March 2024

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.379



9940 572 462



6381 907 438



ijircce@gmail.com



www.ijircce.com

Sentiment Analysis on Comments

Santosh Y. Divekar, Krushna J. Gunjal, Pranay D. Jadhav, Mrunmayee V. Londhe,
Sayali N. Mundhe

Lecturer, Dept. of Computer Engineering, AISSMS Polytechnic, Pune, Maharashtra, India

Student, Dept. of Computer Engineering, AISSMS Polytechnic, Pune, Maharashtra, India

ABSTRACT: In this project, we present a sentiment analysis system aimed at analyzing customer reviews to gauge sentiment towards products or services. The system employs natural language processing techniques and machine learning algorithms to classify text-based reviews into positive, negative, or neutral sentiments. The sentiment analysis is facilitated through a web interface where users can input their reviews, and the system provides sentiment analysis results in real-time. The project integrates Flask, a Python web framework, with Scikit-learn for machine learning tasks and VADER Sentiment Analysis for sentiment scoring. Through this project, we aim to provide businesses with valuable insights into customer sentiment, which can inform decision-making and improve overall customer satisfaction.

KEYWORDS: VADER sentiment analysis tool, NLTK, stopwords, Scikit-learn.

I. INTRODUCTION

In today's digital era, businesses face an unprecedented influx of customer feedback through various channels such as social media, online reviews, and customer support interactions. Understanding the sentiments expressed in this vast amount of textual data is crucial for businesses to gauge customer satisfaction, identify areas for improvement. Customer feedback plays a crucial role in understanding market sentiment and improving product quality or service delivery. Sentiment analysis, a subfield of natural language processing, automates the process of extracting sentiment from text data, making it a valuable tool for businesses seeking to understand customer sentiment at scale.

Sentiment analysis, also known as opinion mining, is a branch of natural language processing (NLP) that focuses on automatically determining the sentiment expressed in text data. It involves analyzing the linguistic features of text to classify it as positive, negative, or neutral. By automating sentiment analysis, businesses can gain valuable insights into customer sentiment at scale, enabling them to respond promptly to feedback, improve customer satisfaction, and drive business growth.

II. RELATED WORK

- [1] Flask Imports: Flask is a web development framework for Python. The code starts by importing necessary modules from Flask, including `Flask`, `request`, and `render_template`.
- [2] Scikit-learn Imports: Scikit-learn is a machine learning library for Python. Here, it's used for text processing and analysis. Specifically, it imports `TfidfVectorizer` for converting text to numerical representations and `cosine_similarity` for calculating the similarity between texts.
- [3] VADER Sentiment Analysis Imports: VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool. It's specifically designed for social media text. The `vaderSentiment` module is imported to utilize its `SentimentIntensityAnalyzer`.
- [4] NLTK Imports: NLTK (Natural Language Toolkit) is a powerful library for natural language processing (NLP) in Python. It's used here for downloading stopwords, which are common words that are often filtered out during text preprocessing.
- [5] Flask App Initialization: An instance of the Flask application is created.
- [6] Route Definitions:
 - a. `/` route: This route renders a template called `form.html` when the user navigates to the root URL.

- b. `/` with POST method: This route is activated when the form in `form.html` is submitted. It retrieves the text data from the form, performs sentiment analysis, and then renders the template `form.html` again with the sentiment analysis results.
- [7] Text Preprocessing:
- a. The code converts the input text to lowercase.
 - b. It removes any digits from the text.
 - c. It removes stopwords (common words like "the", "is", "and", etc.) from the text using NLTK's stopwords corpus.
- [8] Sentiment Analysis:
- a. It uses VADER Sentiment Analysis (`SentimentIntensityAnalyzer`) to perform sentiment analysis on the processed text.
 - b. VADER returns a dictionary containing sentiment scores including positive, negative, neutral, and compound scores. The compound score is a metric that calculates the overall sentiment of the text.
 - c. The compound score is normalized to a scale of 0 to 1 and rounded to two decimal places.
- [9] Rendering Results: The sentiment analysis results (positive, negative, neutral, and compound scores) are passed to the template `form.html` for display.
- [10] Flask App Execution: The Flask application is run with debugging enabled on `127.0.0.1` at port `5002`.



Figure 1: Introduction

III. METHODOLOGY

VADER uses a list of lexical features (e.g. word) which are labeled as positive or negative according to their semantic orientation to calculate the text sentiment. It returns the probability of a given input sentence to be Positive- 99%, Negative- 1%, Neutral- 0%. All these probabilities will sum up to 100%. VADER stands for Valence Aware Dictionary and sEntiment Reasoner, which is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media, and works well on text from other domains. After the VADER, compound and sentiment type for each comment/description received.

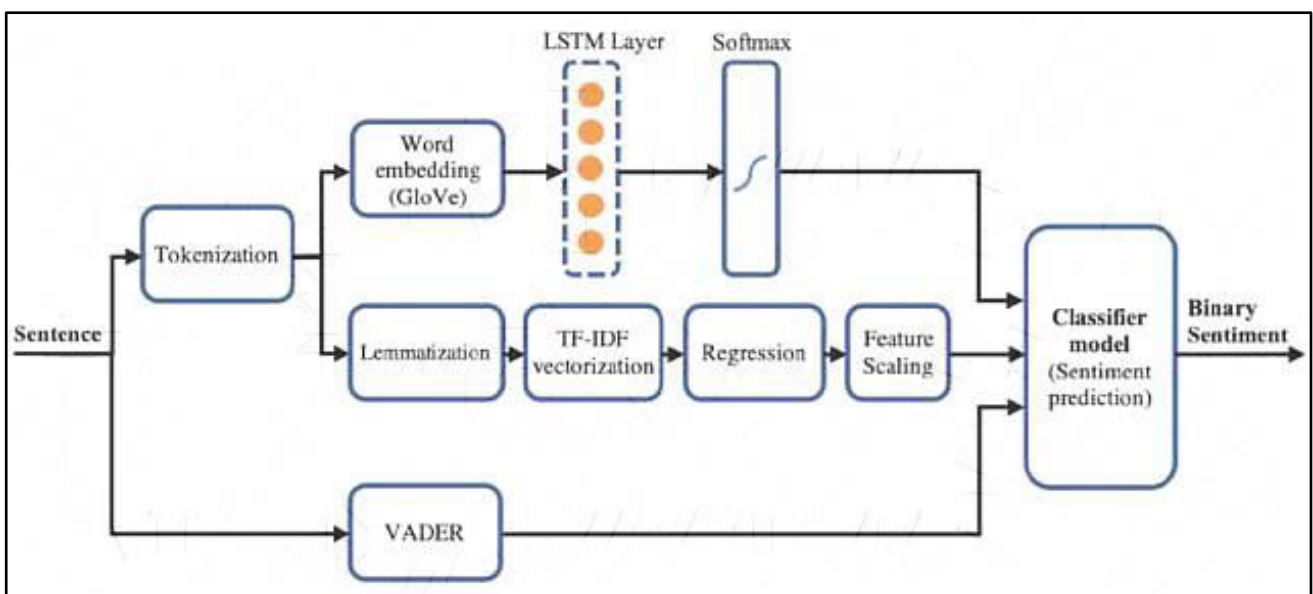
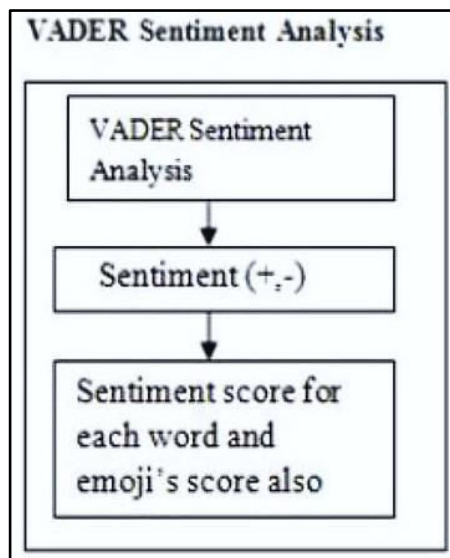


Figure 2: Sentiment Analysis with VADER compared with LSTM

IV. PROPOSED METHODOLOGY

Vader is optimized for social media data that can yield results when used with data from social media sources. As the above result shows the polarity of the word and their probabilities of being positive, negative, neutral, and compound.

Outline of the process:

1. Pre-processing of the input into its component sentences or words.
2. Identify and tag each token with a part-of-speech component (i.e., noun, verb, determiners, sentence subject, etc).
3. Assign a sentiment score from -1 to 1, Where -1 is for negative sentiment, 0 as neutral and +1 is a positive sentiment
4. Return score and optional scores such as compound score, subjectivity, etc. by using the two powerful python tools: VADER.

V. SIMULATION RESULTS

The provided Python code utilizes Flask, scikit-learn, VADER Sentiment Analysis, and NLTK to create a web application for sentiment analysis. It preprocesses user-provided text, calculates sentiment scores, and presents the results back to the user through a web interface. Understanding each component of the code is crucial for developing similar applications and gaining insight into how sentiment analysis can be integrated into web applications for various purposes.

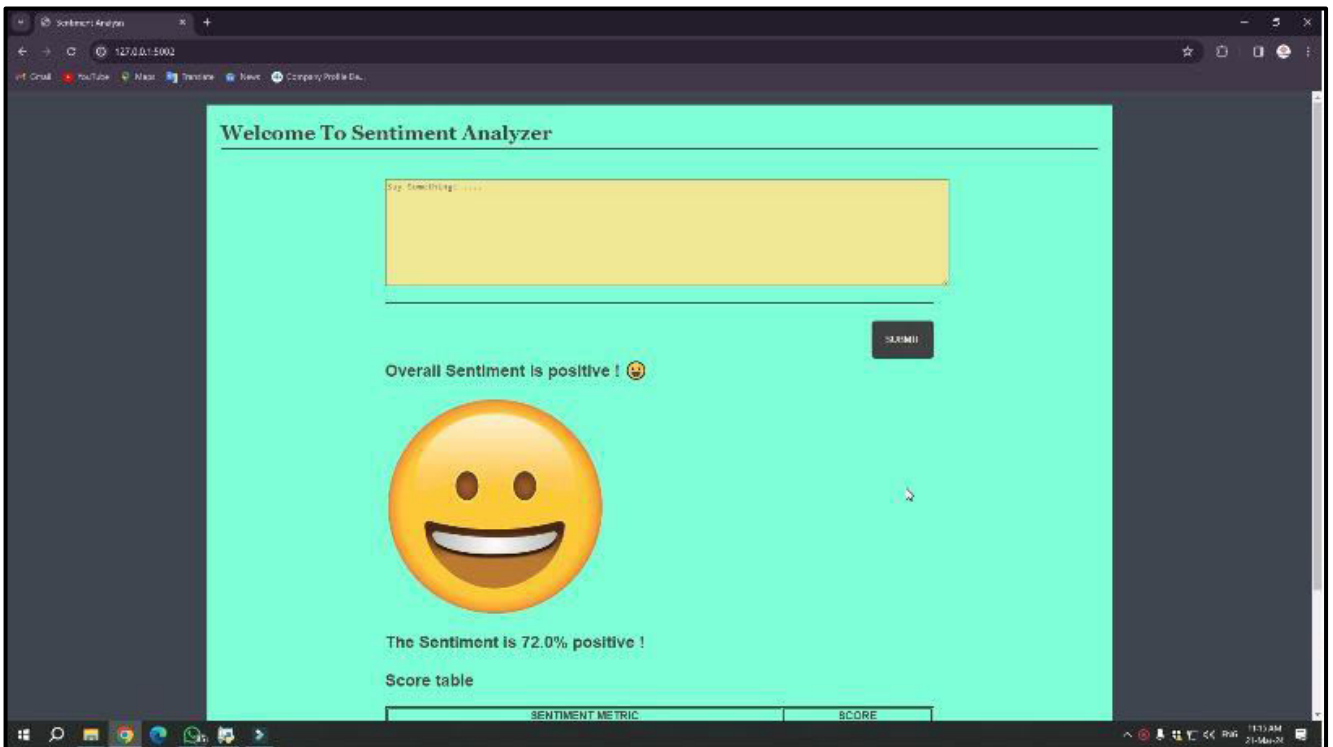


Figure 3: Simulation of sentiment type for a Positive comment/description received.

1. As we provide positive text “Very good product” to sentiment analysis model it will view result as follows:

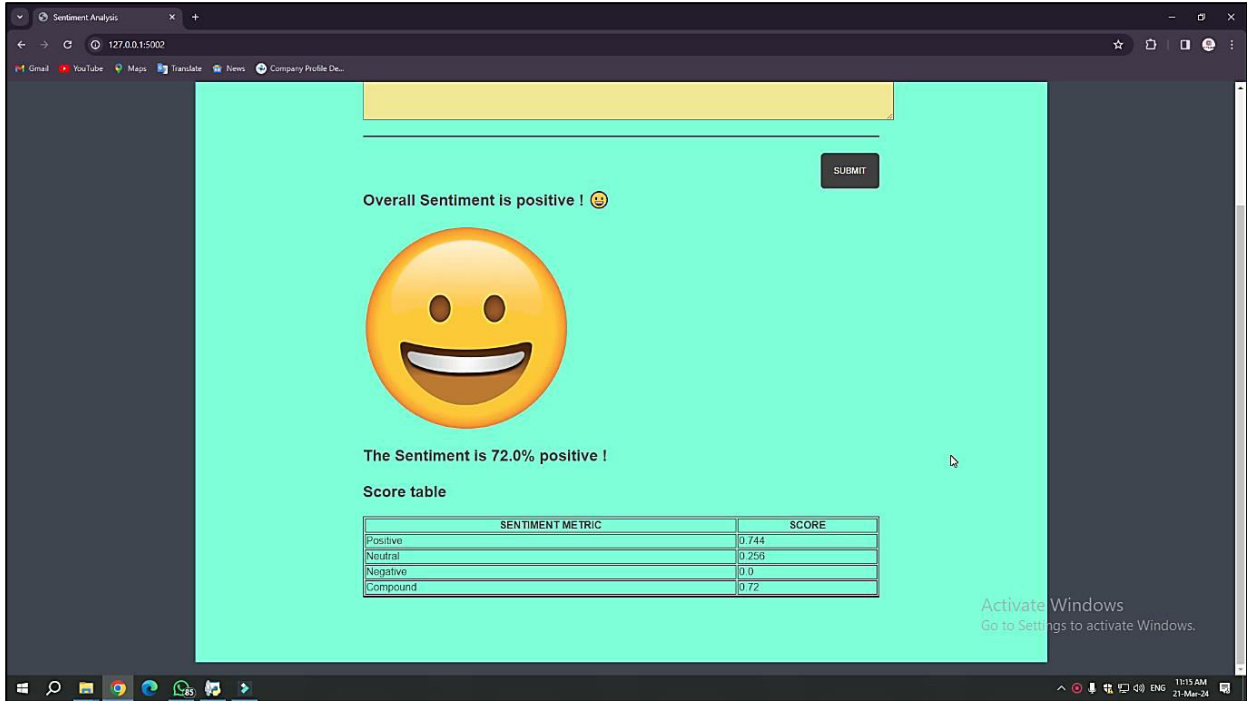


Figure 4: Model reacts for Positive text

As we provide neutral text “Bad product, not satisfied” to sentiment analysis model it will view result as follows:

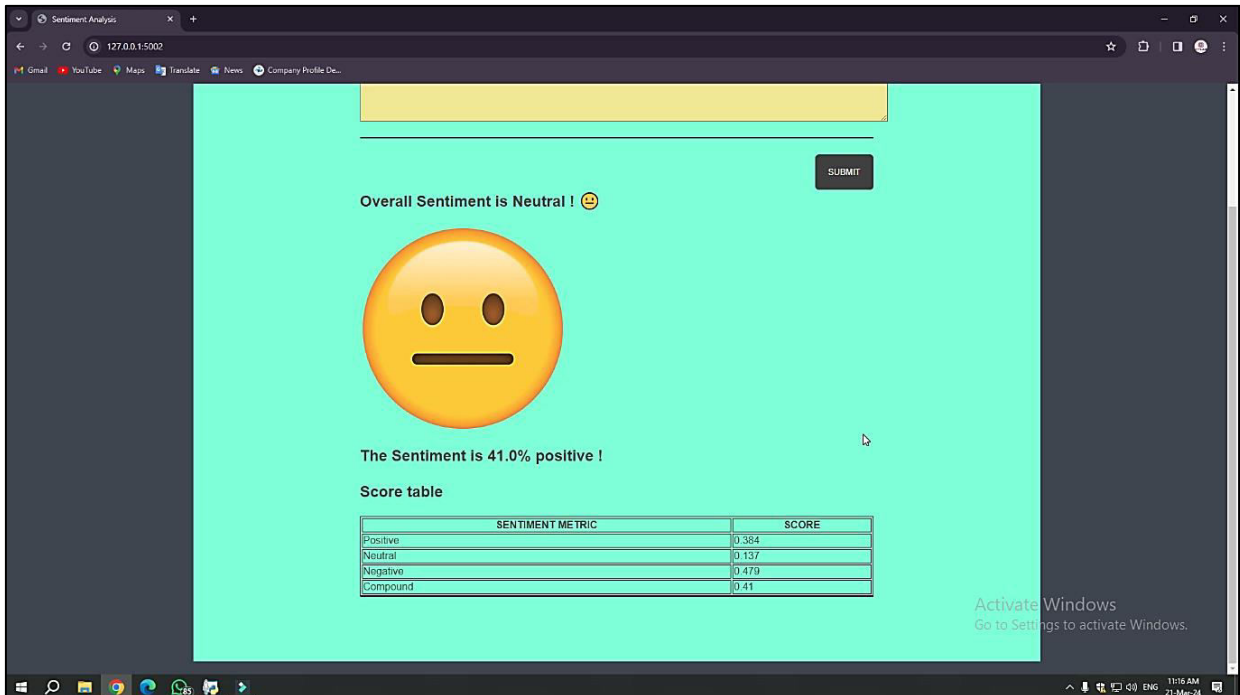


Figure 5: Model reacts for Neutral text

2. As we provide negative text “Worst product” to sentiment analysis model it will view result as follows:

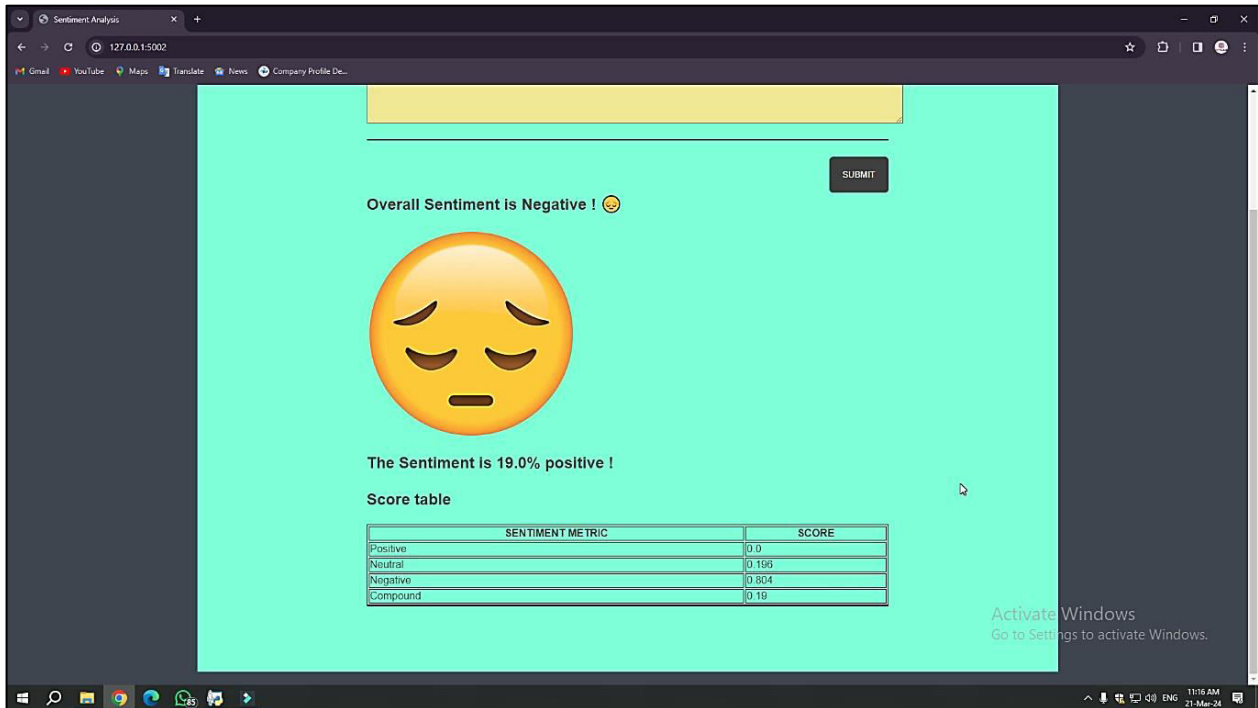


Figure 6: Model reacts for Negative text

VI. CONCLUSION AND FUTURE WORK

The sentiment analysis project presented here offers a powerful tool for businesses to analyze and understand customer sentiment from textual reviews. It processed user-provided text, calculates sentiment scores, and presented the results back to the user through a web interface. By leveraging machine learning and natural language processing techniques, the system provides actionable insights that can drive decision-making and improve customer satisfaction. While the project demonstrates the feasibility and effectiveness of sentiment analysis, there is ample opportunity for future enhancements and integrations to further enhance its utility and applicability in real-world scenarios.

Future Scope:

- Multilingual support: Expanding the system to support sentiment analysis in multiple languages can enhance its applicability in diverse markets.
- Fine-grained sentiment analysis: Implementing more sophisticated sentiment analysis techniques, such as aspect-based sentiment analysis, can provide deeper insights into specific aspects of products or services.
- Integration with CRM systems: Integrating sentiment analysis with customer relationship management (CRM) systems can enable businesses to automate response workflows and track sentiment-related metrics over time.

REFERENCES

- [1] Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.
- [2] Raschka, S., & Mirjalili, V. (2019). Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2 (3rd ed.). Packt Publishing.
- [3] Hutto, C. J., & Gilbert, E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media.



INNO  **SPACE**
SJIF Scientific Journal Impact Factor
Impact Factor: 8.379



ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

 **9940 572 462**  **6381 907 438**  **ijircce@gmail.com**



www.ijircce.com

Scan to save the contact details