

Project Title:

Sentiment Analysis Web Application using Flask and VADER

TABLE OF CONTENT:

S.NO.	TOPICS
1.	Introduction
2.	Objectives
3.	Tools and Technologies
4.	Methodology
5.	Implementation
6.	Future Scope
7.	Screenshots
8.	Conclusion
9.	References

1. Introduction:

Sentiment Analysis is one of the most widely used applications of Natural Language Processing (NLP). It deals with identifying the emotional tone behind textual data. Organizations use sentiment analysis to understand customer opinions, product reviews, and feedback.

This project implements a Flask-based Web Application that allows users to input any text and instantly get its sentiment classification (Positive, Neutral, or Negative). It uses VADER (Valence Aware Dictionary and sEntiment Reasoner), a pre-trained sentiment analysis model optimized for social media text.

2. Objectives:

- To build an interactive web-based sentiment analysis system.
- To classify text as Positive, Neutral, or Negative using VADER.
- To display detailed sentiment scores (positive, neutral, negative, compound).
- To visualize sentiment distribution using a bar chart.
- To provide a simple and user-friendly graphical interface.

3. Tools and Technologies:

Programming Language:	Python
Framework:	Flask(for backend web application)
NLP Library:	NLTK(VADER sentiment analyzer)
Visualization:	Matplotlib
Frontend:	HTML, CSS
Environment:	Virtual Environment in python

5. Methodology:

1. **Data Input:** User provides text via a form.
2. **Pre-processing:** Text is passed directly (VADER handles normalization internally).
3. **Sentiment Analysis:**
 - VADER computes four metrics: Positive, Neutral, Negative, Compound.
 - Compound score determines the sentiment class.
4. **Output Visualization:**
 - Results are shown as:
 - Sentiment Label with Emoji.
 - Sentiment Score Table.
 - Sentiment Distribution Chart (bar chart).

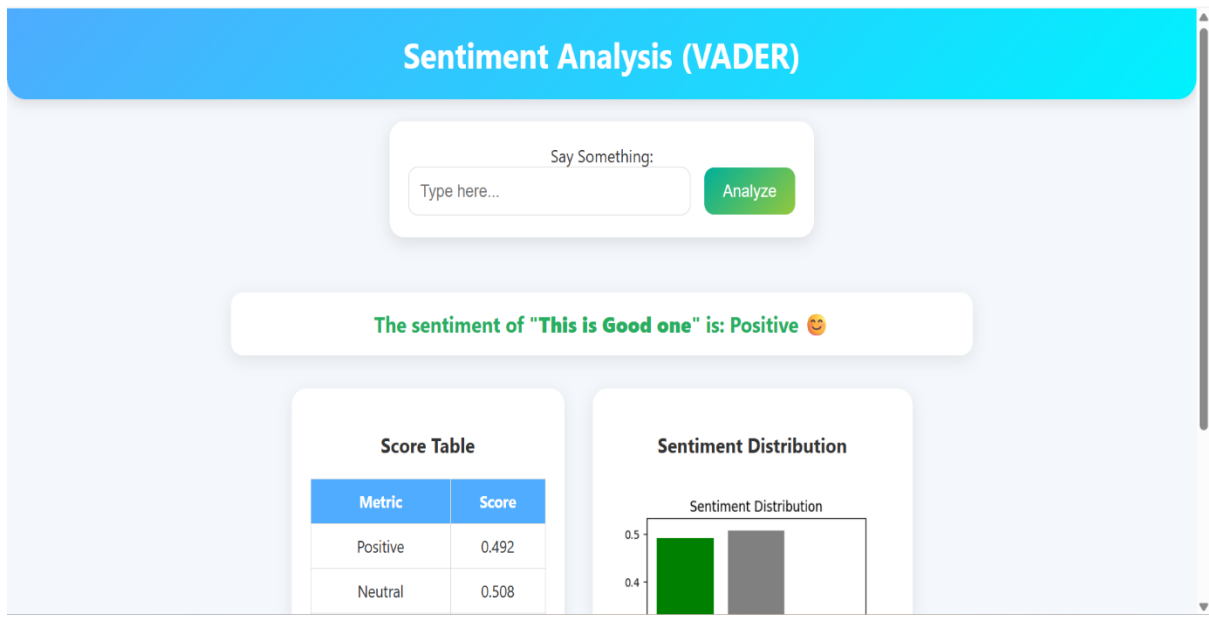
5. Implementation:

- Flask server handles routes (/ for input and results).
- NLTK VADER used for sentiment scoring.
- Matplotlib generates sentiment distribution charts, saved in the static/ folder.
- CSS improves UI (center alignment, clean tables, styled buttons, side-by-side chart + table).

6.Future Scope:

- Train a custom deep learning model (e.g., LSTM, BERT).
- Add database integration to store results.
- Deploy application on cloud platforms (Heroku, AWS, Azure).
- Create an interactive dashboard with advanced visualizations (Plotly/Dash)

7.Snapshots:



Sentiment Analysis (VADER)

Say Something:

This is worst experience ever

Analyze

The sentiment of "This is worst experience ever" is: Negative 😞

Score Table

Metric	Score
Positive	0.0
Neutral	0.494

Sentiment Distribution

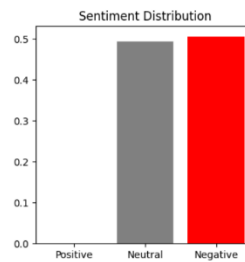


The sentiment of "This is worst experience ever" is: Negative 😞

Score Table

Metric	Score
Positive	0.0
Neutral	0.494
Negative	0.506
Compound	-0.6249

Sentiment Distribution

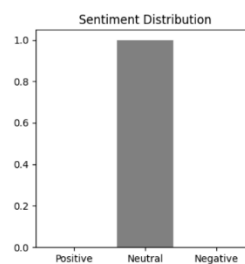


The sentiment of "I have seen this" is: Neutral 😐

Score Table

Metric	Score
Positive	0.0
Neutral	1.0
Negative	0.0
Compound	0.0

Sentiment Distribution



8. Conclusion:

This project successfully demonstrates intermediate-level sentiment analysis using Flask and VADER. It provides a real-time web interface for analyzing user text, displaying both quantitative scores and visual representation of sentiment. The system is efficient, lightweight, and easy to use, making it suitable for academic purposes and small-scale applications.

9. References:

1. NLTK Library Documentation – <https://www.nltk.org/>
2. Flask Framework – <https://flask.palletsprojects.com/>