

A PROJECT REPORT ON
Sentiment analysis

ARTIFICIAL INTELLIGENCE & DATA SCIENCE

21AI3612 – Data Science Lab

YEAR: 2024-2025

Submitted By:

ARUL GNANAKUMAR.R - 95072317016

MARI DINESH N - 95072317057

AMARNATH - 95072317301

HARIPRASATH - 95072317302

INDEX

S.no	Contents	Pg.no	Signature
1	Abstract	1	
2	Problem Statement	1	
3	Existing System	2	
4	Proposed System	3	
5	Block Diagram	4	
6	Program	5	
7	Output	7	
8	Conclusion	8	

Abstract:

This project focuses on the development of a sentiment analysis tool using two popular natural language processing (NLP) techniques: VADER (Valence Aware Dictionary and sEntiment Reasoner) and TextBlob. Sentiment analysis plays a pivotal role in understanding user emotions, opinions, and attitudes expressed through text. The tool takes user-provided text, analyzes it for sentiment (positive, negative, neutral), and outputs sentiment scores using both VADER and TextBlob. The purpose of this system is to simplify the sentiment analysis process for various applications, such as social media analysis, product reviews, and customer feedback analysis.

Problem Statement:

In the digital age, a massive volume of unstructured data is generated through social media, product reviews, blogs, and customer feedback. Understanding the sentiment behind these data is crucial for businesses to make informed decisions. However, manually analyzing such large datasets is impractical and time-consuming. Therefore, an automated sentiment analysis tool that is efficient, reliable, and accurate is required to help in categorizing and understanding user opinions.

Existing Systems:

Existing systems for sentiment analysis include both rule-based approaches and machine learning models. Some of the commonly used sentiment analysis tools are:

- **Rule-Based Approaches:** Tools like VADER and TextBlob, which use predefined dictionaries and heuristics to determine sentiment polarity. These tools are easy to use, fast, and often effective for social media text and informal language.
- **Machine Learning Models:** Advanced systems use machine learning and deep learning algorithms, such as Support Vector Machines (SVM), Naive Bayes, and neural networks. These models require large amounts of labeled training data and are often more complex to implement and fine-tune.

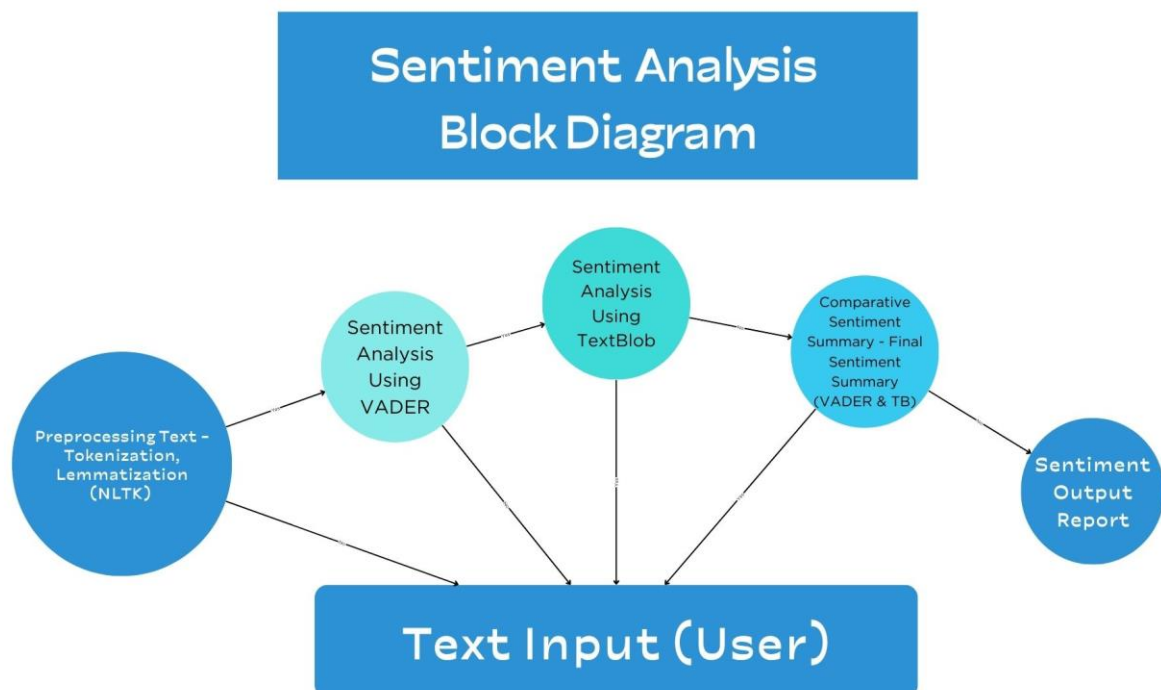
Proposed System:

The proposed system aims to combine the strengths of two rule-based sentiment analysis tools: **VADER** and **TextBlob**. These tools will be integrated into a Python program that allows users to input text and receive sentiment scores along with categorized sentiments (positive, negative, or neutral). The system will handle real-time input, analyze the text, and output detailed sentiment scores.

- **VADER**: A lexicon and rule-based sentiment analysis tool that works well with social media text, emojis, and informal language. It provides detailed sentiment scores for positive, neutral, and negative sentiment.
- **TextBlob**: A text-processing library that can determine the polarity of the text. It uses natural language processing and is particularly effective for formal texts.

The proposed system will compare both VADER and TextBlob analysis results for each text input and provide users with comprehensive sentiment feedback. Additionally, the system will generate a summary report of the sentiment distribution across multiple text inputs.

BLOCK DIAGRAM:



PROGRAM:

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
from textblob import TextBlob
import matplotlib.pyplot as plt
import nltk
nltk.download('punkt') # Download necessary data for TextBlob

vader_analyzer = SentimentIntensityAnalyzer()

def vader_sentiment(text):
    return vader_analyzer.polarity_scores(text)

def textblob_sentiment(text):
    blob = TextBlob(text)
    polarity = blob.sentiment.polarity
    if polarity > 0:
        return {'polarity': polarity, 'label': 'Positive'}
    elif polarity < 0:
        return {'polarity': polarity, 'label': 'Negative'}
    else:
        return {'polarity': polarity, 'label': 'Neutral'}

def advanced_sentiment_analysis(text):
    vader_result = vader_sentiment(text)
    textblob_result = textblob_sentiment(text)

    print(f"Text: {text}")
    print("VADER Sentiment Scores:", vader_result)
    print(f"TextBlob Sentiment Polarity: {textblob_result['polarity']}, Label: {textblob_result['label']}")

    return vader_result, textblob_result

def analyze_and_visualize(sentences):
    vader_scores = []
    textblob_scores = []
    categories = {"Positive": 0, "Negative": 0, "Neutral": 0}

    for sentence in sentences:
        vader_result, textblob_result = advanced_sentiment_analysis(sentence)
        vader_scores.append(vader_result['compound'])
```

```

        textblob_scores.append(textblob_result['polarity'])
        categories[textblob_result['label']] += 1

    plot_sentiment(vader_scores, textblob_scores, categories)

def plot_sentiment(vader_scores, textblob_scores, categories):
    fig, axs = plt.subplots(2, 1, figsize=(10, 8))

    axs[0].plot(vader_scores, label="VADER Compound Score",
                marker='o', linestyle='-', color='b')
    axs[0].plot(textblob_scores, label="TextBlob Polarity Score",
                marker='x', linestyle='--', color='g')
    axs[0].set_title("Sentiment Scores Comparison")
    axs[0].set_xlabel("Sentence Index")
    axs[0].set_ylabel("Sentiment Score")
    axs[0].legend()
    axs[0].grid(True)

    labels = list(categories.keys())
    values = list(categories.values())
    axs[1].bar(labels, values, color=['green', 'red', 'gray'])
    axs[1].set_title("Sentiment Distribution (TextBlob)")
    axs[1].set_ylabel("Count")

    plt.tight_layout()
    plt.show()

sentences = []

print("Enter sentences one by one. Type 'done' when finished.")

while True:
    sentence = input("Enter a sentence: ")
    if sentence.lower() == 'done':
        break
    sentences.append(sentence)

if sentences:
    analyze_and_visualize(sentences)
else:
    print("No sentences provided for analysis.")

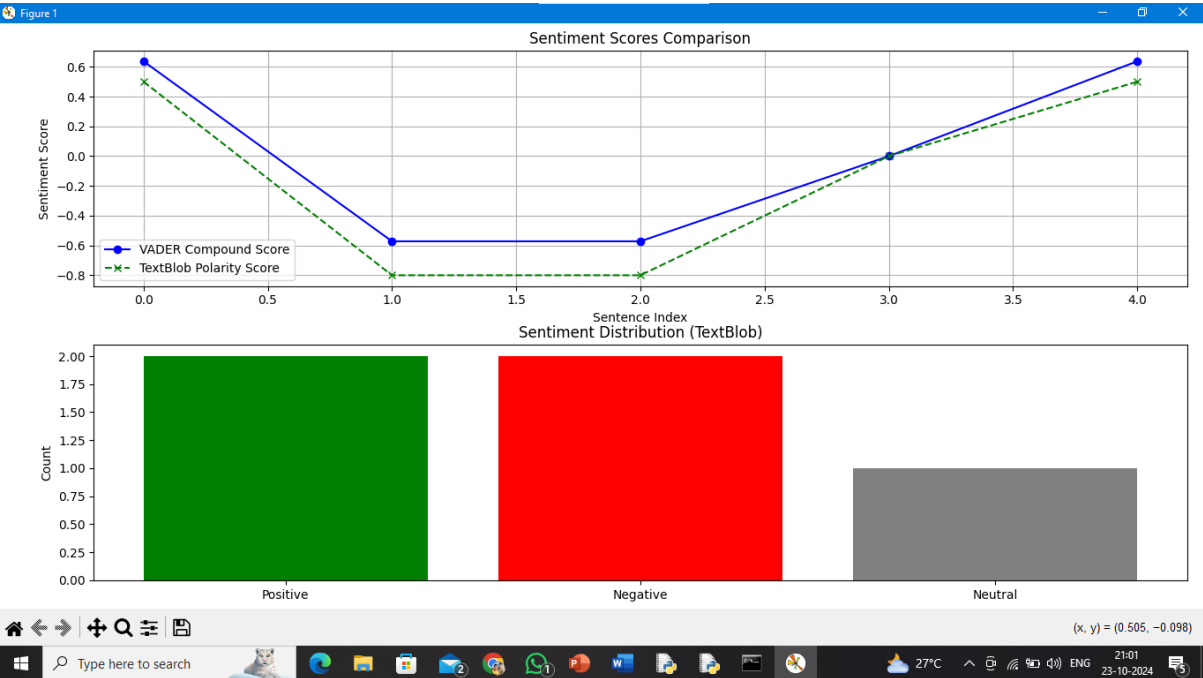
```

OUTPUT:

```
Python 3.13.0 (tags/v3.13.0:60403a5, Oct 7 2024, 09:38:07) [MSC v.1941 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
===== RESTART: C:/Users/Admin/Desktop/DSE 2.py =====
[nltk_data] Downloading package punkt to
[nltk_data] C:\Users\Admin\AppData\Roaming\nltk_data...
[nltk_data] Package punkt is already up-to-date!
Enter sentences one by one. Type 'done' when finished.
Enter a sentence: i love you
Enter a sentence: i hate you
Enter a sentence: i hate you
Enter a sentence: who are you?
Enter a sentence: i love you
Enter a sentence: done
Text: i love you
VADER Sentiment Scores: {'neg': 0.0, 'neu': 0.323, 'pos': 0.677, 'compound': 0.6369}
TextBlob Sentiment Polarity: 0.5, Label: Positive
Text: i hate you
VADER Sentiment Scores: {'neg': 0.649, 'neu': 0.351, 'pos': 0.0, 'compound': -0.5719}
TextBlob Sentiment Polarity: -0.8, Label: Negative
Text: i hate you
VADER Sentiment Scores: {'neg': 0.649, 'neu': 0.351, 'pos': 0.0, 'compound': -0.5719}
TextBlob Sentiment Polarity: -0.8, Label: Negative
Text: who are you?
VADER Sentiment Scores: {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
TextBlob Sentiment Polarity: 0.0, Label: Neutral
Text: i love you
VADER Sentiment Scores: {'neg': 0.0, 'neu': 0.323, 'pos': 0.677, 'compound': 0.6369}
TextBlob Sentiment Polarity: 0.5, Label: Positive
```

!



Conclusion:

This report proposes the development of a hybrid sentiment analysis tool that leverages the strengths of VADER and TextBlob for real-time sentiment classification. The tool can be highly useful for small- to medium-scale text analysis tasks, especially in scenarios where informal text is predominant (e.g., social media analysis). It also offers simplicity, efficiency, and accuracy without the overhead of complex machine learning models.

The integration of these tools ensures that users have access to both granular and overall sentiment results, providing better insight into text data. Future work could focus on extending the system by integrating machine learning techniques to enhance accuracy, especially for more complex sentiment analysis tasks.