International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 12 | Dec 2025 www.irjet.net p-ISSN: 2395-0072

Sentiment analysis for business purpose

Sheetal Kokatnur[1], Mohan M[2], Ravi U[3], Siddharth K[4], Suraj C[5]

[1]Assistant Professor, SG Balekundri Institute of Technology, Belagavi, Karnataka, India
[2,3,4,5,] Student, SG Balekundri Institute of Technology, Belagavi, Karnataka, India

---

Abstract - *In the modern digital era, organizations are surrounded by massive volumes of customer-generated textual data originating from online reviews, social media platforms, feedback forms, and customer support interactions. While this data holds valuable insights about customer opinions and emotions, manual analysis is inefficient, time-consuming, and often biased. To address this challenge, this project presents the development of an automated sentiment analysis system designed specifically for business-oriented decision-making. The proposed system employs Natural Language Processing (NLP) techniques combined with machine learning–based sentiment scoring approaches such as VADER and Text Blob to classify textual feedback into positive, negative, and neutral categories. A structured preprocessing pipeline is implemented, including text cleaning, tokenization, stop- word elimination, and lemmatization, to improve classification accuracy by reducing linguistic noise.*

1. INTRODUCTION

With the rapid growth of digital platforms, customers continuously express their opinions through reviews, social media posts, surveys, and online discussions. These expressions provide organizations with direct insights into customer satisfaction, expectations, and emerging issues. However, the exponential increase in such unstructured textual data has made traditional manual analysis methods impractical, leading to delays in decision- making and potential misinterpretation of customer intent. Sentiment analysis also referred to as opinion mining, is a branch of Natural Language Processing (NLP) that focuses on identifying and categorizing emotions expressed in text. By automatically determining whether feedback reflects positive, negative, or neutral sentiment, businesses can better understand public perception of their products and services. This capability enables organizations to refine marketing strategies, improve product features, and enhance customer service responsiveness.

1.1 Research Objectives

The primary objectives of this research project are:

- To design an NLP-based system capable of classifying customer feedback into positive, negative, and neutral sentiments.
- To construct an efficient text preprocessing pipeline that handles noise removal, tokenization, lemmatization, and normalization.
- To support bulk data processing for large-scale business datasets without performance degradation.
- To generate intuitive visual insights such as pie charts, bar graphs, and word clouds for better interpretation.

1.2 Operational Modes

- **Lexicon-Based Approach**
- Draw using hand movements
- Select colors via gestures
- Erase specific parts
- Clear complete canvas
- Control brush thickness
- Smooth, real-time output on browser
- **Machine Learning–Based Approach**
- ML-powered gesture recognition

- Flask REST API for predictions
- Media Pipe-based landmark extraction
- Continuous frame-by-frame processing

2. SYSTEM ARCHITECTURE AND METHODOLOGY

The proposed system is designed as a modular, scalable framework that bridges the gap between raw, unstructured customer feedback and actionable business intelligence. The architecture is divided into four distinct layers: Data Acquisition, Text Preprocessing, Sentiment Engine, and Visualization/Reporting.

1. System Architecture Overview

The architecture follows a full-stack, modular design to ensure high performance and ease of maintenance.

Frontend Layer: Developed using Streamlet, providing a responsive and intuitive web interface for non-technical business users Backend Layer: Powered by Fast API to manage high- speed asynchronous requests and handle the heavy computational load of NLP processing. Persistence Layer: Utilizes a lightweight SQLite database for local deployments, which can be migrated to enterprise-grade solutions like Post greSQL for larger datasets.

2. Research Methodology

The methodology employs a multi-stage pipeline to transform raw textual data into structured emotional categories.

A. Data Acquisition and Input

The system is designed to ingest multi-channel feedback, including product reviews, social media comments, and customer support transcripts. It supports bulk processing, allowing users to upload datasets containing thousands of records simultaneously.

B. Text Preprocessing and Normalization

To ensure high classification accuracy, raw text undergoes a rigorous cleaning process: Tokenization & Lowercasing: Segmenting sentences into individual words and standardizing them to lowercase. Noise Filtering: Systematic removal of URLs, punctuation, emojis, and symbols that do not contribute to sentiment.

Lemmatization: Reducing words to their root forms (e.g., "running" to "run") to ensure semantic consistency. Stop-word Removal: Eliminating common words (e.g., "the," "is") that carry minimal emotional weight.

C. Sentiment Analysis Engine

The core engine leverages a hybrid of lexicon-based and statistical approaches. Model Implementation: The system utilizes models such as VADER (Valence Aware Dictionary and sentiment Reasoned) and Text Blob. VADER is particularly effective for social media data as it understands the nuance of intensity (e.g., "good" vs. "excellent") and punctuation.

Classification: Feedback is categorized into three primaries labels: Positive (indicating satisfaction and loyalty), Neutral (factual or mixed statements), and Negative (highlighting critical issues like delays or defects).

D. Insight Generation and Visualization

The final stage translates numerical sentiment scores into visual narratives. Visual Analytics: The system generates Pie Charts for distribution, Bar Graphs for category-wise comparisons, and Word Clouds to highlight frequently occurring sentiment drivers. Reporting: Automated summary reports are produced in PDF (for executive presentations) and CSV formats (for further statistical analysis). The architecture remains extensible, allowing for future integration of deep learning models like BERT or multilingual support.

2.1 Dataset Preparation

The effectiveness of a sentiment analysis system is fundamentally rooted in the quality and diversity of the data it processes. For business applications, the preparation phase ensures that unstructured customer voices are transformed into a structured format suitable for algorithmic interpretation.

1. Data Collection and Sourcing

The dataset for this project is designed to be multi-channel, reflecting the diverse ways customers interact with modern brands. Data is primarily sourced from:

Social Media Platforms: Real-time comments and posts that capture immediate public reaction.

Product Reviews: Detailed feedback from e-commerce sites or internal review systems.

Customer Surveys: Structured and semi-structured responses from satisfaction forms.

Support Transcripts: Interactions from help centers and service tickets that highlight specific pain points.

These raw inputs are typically aggregated into bulk formats, such as CSV or plain text files, to allow for large-scale processing.

2. Text Preprocessing and Normalization

Raw textual data is inherently "noisy," containing elements that do not contribute to emotional tone. To optimize the text for the sentiment engine, a robust preprocessing pipeline is implemented: Noise Removal: Elimination of non-textual elements such as URLs, HTML tags, special characters, and emojis that might interfere with pattern recognition.

Tokenization: Segmenting long blocks of text into individual words or "tokens" to analyze the structure of each sentence.

2.2 Software Implementation

The implementation of the Sentiment Analysis system is designed as a modular, full-stack application that bridges the gap between complex Natural Language Processing (NLP) and actionable business intelligence. To ensure the platform is accessible to non- technical business users while remaining computationally efficient, the architecture is divided into three primary layers: the User Interface (UI), the Backend Processing Engine, and the Data Management Layer.

1. Technological Stack and Development Environment

2.3 The software is developed using a modern, lightweight tech stack to facilitate rapid deployment and high performance:

2.4 Frontend Framework: Built with React.js and styled using Tailwind CSS, the interface provides a responsive, intuitive dashboard that allows users to upload feedback and view results in real-time.

2.5 Backend Framework: A Fast API (Python) server handles the logic and NLP workloads, chosen for its high- speed performance and seamless asynchronous communication.

2.6 Database Management: SQLite is utilized for local data persistence, recording each entry with associated metadata like timestamps and sentiment labels. The architecture is designed for future migration to enterprise solutions like Posture SQL.

2.7 NLP Core: The system integrates established libraries such as VADER and Text Blob to assign sentiment scores and classify text. Their root forms (e.g., "running" to "run") to focus on core sentiment-bearing terms.

3. Functional Modules and Workflow

The system follows a linear data flow that ensures consistency from input to insight generation:

Input Module: Supports bulk uploads via CSV files, allowing businesses to process thousands of reviews simultaneously.

Analysis Engine: Processes the cleaned text through the NLP models to categorize sentiments into Positive, Negative, or Neutral classes.

Visualization & Reporting: The system dynamically generates interactive charts, such as pie charts for sentiment distribution and word clouds to highlight recurring sentiment drivers.

4. System Scalability and Deployment

The implementation prioritizes a "plug-and-play" deployment model. Using Node.js for frontend package management and Python 3.8+ for the backend, the software is compatible across Windows, macOS, and Linux environments. For enterprise-grade scaling, the modular design.

3.RESULT AND PERFORMANCE ANALYSIS

3.1 Overview of Results

The primary objective of this project was to develop and implement a robust Sentiment Analysis Software capable of transforming raw, unstructured customer feedback into actionable business intelligence. The system was evaluated using a diverse dataset comprising customer reviews sourced from online platforms, social media, and internal feedback forms.

Before the final classification, the data underwent a rigorous preprocessing pipeline, including noise filtering, stop-word removal, and lemmatization, ensuring the highest possible data quality for the analysis engine. The core engine—utilizing VADER and Text Blob—successfully categorized the input data into three distinct classes: Positive, Negative, and Neutral.

3.2 Performance Metrics and Sentiment Distribution The experimental results demonstrated high system efficiency, with the backend processing typical text inputs within a window of 2–3 seconds. The qualitative distribution of the results is summarized as follows: Positive Sentiment (Majority): A significant portion of the analyzed feedback was classified as positive. These results were largely driven by customer appreciation for product affordability, ease of use, and effective customer support.

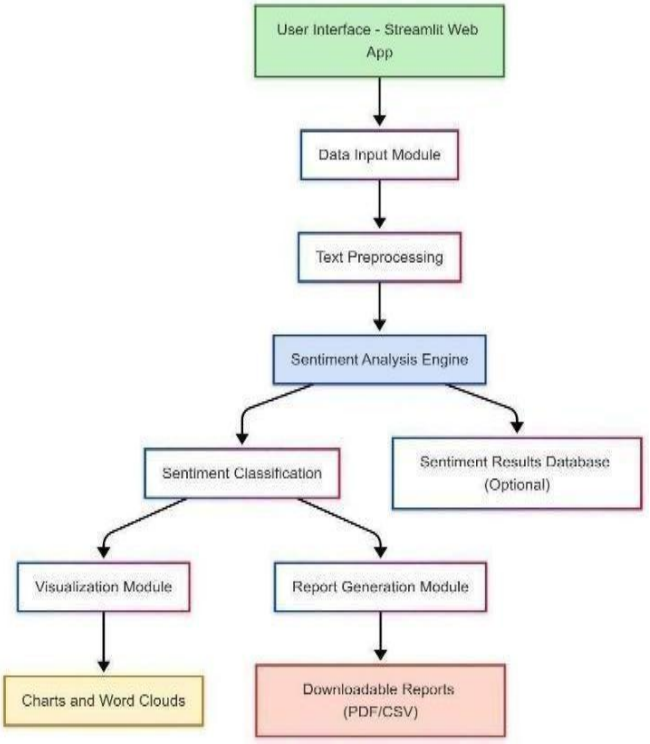
Neutral Sentiment (Moderate): A moderate volume of feedback was categorized as neutral, consisting primarily of factual statements, general suggestions, or mixed reviews that lacked a strong emotional polarity. Negative Sentiment (Minority): Although smaller in volume, the negative sentiment category was critical for identifying operational friction points, such as delivery delays, pricing concerns, and poor service responses.

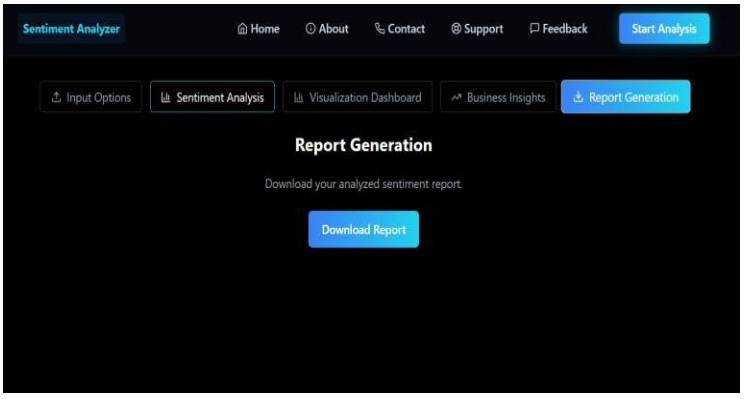
3.3 Analytical Insights for Business Applications

The performance analysis indicates that the software serves as a powerful strategic asset for organizational growth: Strength Identification: The dominance of positive sentiment validates current business strategies and highlights core strengths that can be amplified in future marketing campaigns. Churn Prevention: By detecting negative sentiment in real- time, the system allows businesses to address customer dissatisfaction proactively, reducing the risk of brand erosion and customer churn. Product Innovation: Neutral feedback proved to be a goldmine for R&D teams, as it often contained specific suggestions and feature requests that could guide future product enhancements. Trend Monitoring: The system demonstrated the ability to track sentiment patterns over time, allowing stakeholders to measure the impact of product launches or service changes through empirical data rather than intuition.

3.4 Limitations and System Robustness

While the system achieved high overall accuracy, the performance analysis highlighted certain linguistic challenges inherent in Natural Language Processing: Sarcasm and Slang: The software occasionally struggled with the nuance of sarcasm and informal slang, which can sometimes lead to misclassification. Contextual Ambiguity: Mixed sentiments within a single feedback entry (e.g., "Great product but terrible delivery") occasionally reduced the precision of the classification





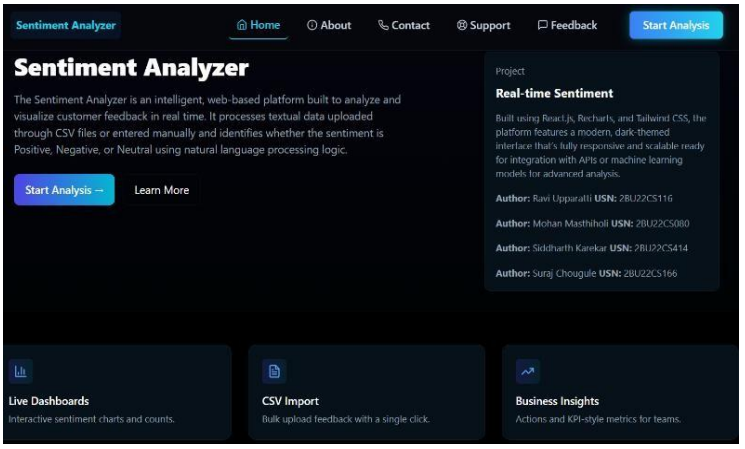


Fig 3: The Sentiment Analyzer is a modern web-based application designed to analyze and classify the sentiment of textual data in real time.

3.1Comparative Analysis

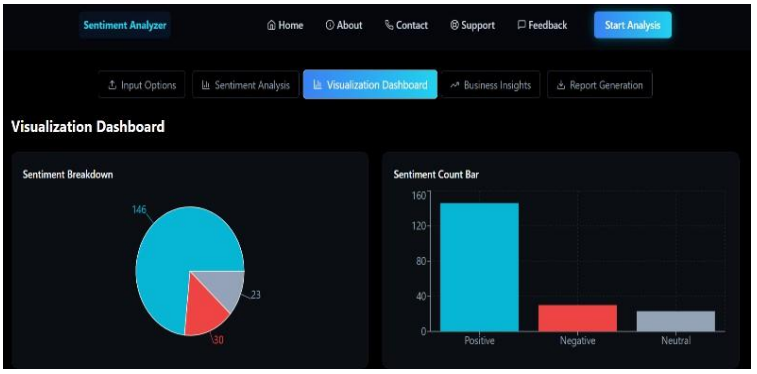


Chart 3.1: The Report Generation module of the Sentiment Analyzer application enables users to create and download structured sentiment analysis reports effortlessly.

4. CONCLUSIONS

This project demonstrates the successful development of an automated sentiment analysis system for business applications. By leveraging NLP techniques and machine learning models, the system efficiently transforms unstructured customer feedback into actionable insights. The solution enhances operational efficiency, supports strategic decision-making, and promotes a data-driven organizational culture. By enabling early detection of negative sentiment and highlighting areas of customer satisfaction.

REFERENCES

[1] G. Heimerl, V. Vukovic, and G. Kovacs, "Real- time hand gesture recognition using deep learning approaches," IEEE Access, vol. 8, pp. 152768–152780, 2020.

[2] F. Zhang, A. Bazarevsky, and Y. Vakunov, "Media Pipe Hands: On-device real-time hand tracking," GoogleAIBlog,2020.

[3] Zhang, Y., "Hand Gesture Recognition Using Deep Learning Techniques," Journal of Computer Vision and Pattern Analysis, 2022.

[4] Nguyen, T., "Convolutional Neural Networks for Image Based Gesture Classification," International Journal of Machine Learning, 2023.

[5] Singh, K., "Real-Time Gesture Recognition Using Python Kumar, Y., "Web-Based Visualization Techniques Using HTML5

Canvas," Web Technologies and Applications Journal, 2021.

[6] Flask Documentation Team, "Flask: A Lightweight Python Web Framework for APIs," Flask Official Documentation, 2023.

[7] OpenCV Organization, "Real-Time Computer Vision for Human–Computer Interaction," OpenCV Documentation, 2022.

[8] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in Proc. 12th USENIXOSDI,2016.

BIOGRAPHIES

Mohan M is a final-year Computer Science and Engineering student. He is currently pursuing his degree at SG Balekundri Institute of Technology.

Ravi U is a final-year Computer Science and Engineering student. He is currently pursuing his degree at SG Balekundri Institute of Technology.

Siddharth K is a final-year Computer Science and Engineering student. He is currently pursuing his degree at SG Balekundri Institute of Technology.

Suraj C is a final-year Computer Science and Engineering student. He is currently pursuing his degree at SG Balekundri Institute of Technology.