A PROJECT REPORT

ON

"ANALYZING CUSTOMER FEEDBACK WITH TEXT INPUTS"

Submitted to

School of Computer Engineering
KIIT UNIVERSITY, BHUBANESWAR

In Partial Fulfilment of the Requirement for the Award of

Bachelor of Computer Science and Engineering

Names and Roll Numbers

ADITI MUKHERJEE	22051395
SIDDHI SHRIYA	22051631
PUJA KUMARI	22051446
AMRITA PATTANAIK	22052188
SUBHAMITA PAUL	22051639
AMRIT RAJ	22053486

Under the Guidance of

Prof. NACHIKETA TARASIA

2024–2025

AFFILIATED TO

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY (KIIT)

Deemed to be University U/S 3 of UGC Act, 1956

Acknowledgement

We express our sincere gratitude to our guide **Prof. Nachiketa Tarasia**, School of Computer Engineering, KIIT University, for his valuable guidance and support throughout this project. We would also like to thank the university for providing us with a platform to apply our knowledge and the support from our families and friends.

Abstract

This project focuses on developing a system to analyze customer feedback using multimodal inputs including text, voice, images, and videos. The goal is to capture and process customer responses in real time and extract meaningful insights that can help businesses enhance their services. The proposed system uses natural language processing, image recognition, and sentiment analysis to evaluate feedback. This multimodal approach provides a comprehensive understanding of customer sentiment and improves decision-making.

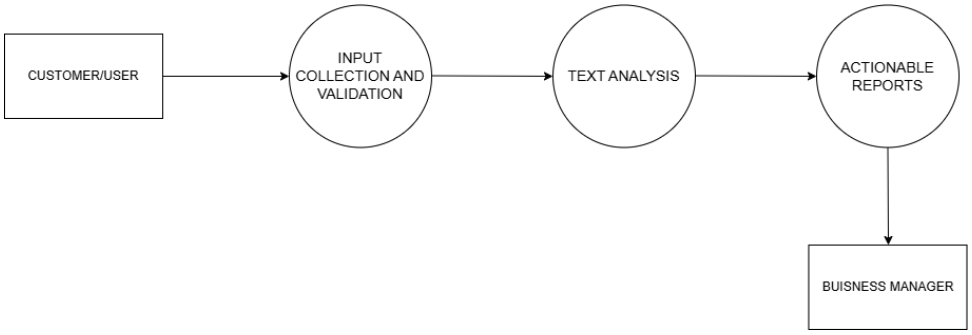
Contents

Chapter 1

Introduction

1.1 Overview

FeedbackAI is a robust and scalable web application created to collect, analyze, and visualize customer feedback using advanced Natural Language Processing (NLP) techniques. The platform combines both rule-based (VADER) and deep learning-based (RoBERTa) sentiment analysis methods to provide accurate classification of textual input. It facilitates interactive user engagement with real-time sentiment analysis results displayed through a responsive and modern web interface. Designed for flexibility, FeedbackAI allows for integration of future models and analytical features.



1.2 Purpose

The purpose of the FeedbackAI system is multi-faceted:

- Provide an accessible, easy-to-use web-based interface for users to submit categorized feedback along with optional numeric ratings.
- Deliver sentiment classification using both traditional and modern NLP models for enhanced reliability.
- Offer visual feedback in the form of charts and labels to help users quickly interpret the sentiment result.
- Equip administrators with powerful dashboards and filtering tools to identify sentiment trends and decision-influencing patterns.
- Serve as an educational tool for comparing model performance in sentiment analysis use cases.

1.3 Scope

The scope of the FeedbackAI project encompasses:

- User interface to collect structured and unstructured customer feedback.
- Implementation of three NLP models: VADER, RoBERTa, and HuggingFace sentiment pipeline.
- Visualization features including sentiment pie charts, category bar graphs, and trend timelines.
- A testimonial archive and filter functionality to sort feedback based on category, sentiment, and score.
- Modular design to accommodate upgrades, additional models, and extended NLP capabilities like emotion detection or keyword extraction.

1.4 Definitions, Acronyms, and Abbreviations

- **VADER**: Valence Aware Dictionary and sEntiment Reasoner, a lexicon-based sentiment analysis model.
- **RoBERTa**: A transformer-based language model optimized for robust sentiment classification.
- **NLP**: Natural Language Processing, a branch of AI focused on understanding and generating human language.
- **HuggingFace Pipeline**: Pre-built abstraction for various NLP tasks including sentiment analysis using pretrained transformers.
- **Sentiment Polarity**: Classification of text as positive, neutral, or negative.

Chapter 2

Overall Description

2.1 Product Perspective

FeedbackAI is a standalone application built with separation of concerns and modularity in mind. It is designed to function independently but can be integrated into larger enterprise feedback systems. It comprises:

- A front-end built in React (TypeScript) for seamless and fast user interaction.
- A backend using Flask (Python) that connects the UI to the sentiment models.
- Dedicated components for input handling, model inference, response formatting, and visualization.
- Scalability built-in through clear API boundaries and minimal coupling.

2.2 Product Features

- **Comprehensive Feedback Form**: Users can select a feedback category (Product, Service, or General), enter descriptive text, optionally provide a summary, and assign a 1–5 star rating.
- **Triple Sentiment Model Pipeline**: Text is processed through VADER (lexicon-based), RoBERTa (transformer-based), and HuggingFace (pretrained pipeline) models for diversified sentiment analysis.
- **Insightful Dashboard**: An interactive dashboard displays aggregated feedback sentiment using intuitive visualizations like pie charts and bar graphs.
- **Dynamic Testimonials View**: Allows browsing of previously submitted feedback entries with sorting and filtering based on sentiment or category.
- **Theming Options**: Toggleable dark and light themes ensure visual comfort for users in different environments.
- **Instantaneous Feedback Response**: Sentiment results and confidence scores are presented immediately after submission.

2.3 User Classes and Characteristics

- **End Users**:
  - Primary consumers of the interface.
  - Require no technical knowledge.
  - Engage with the submission form and view real-time sentiment classification.
- **Administrators**:
  - Have access to system analytics and management views.
  - Use dashboards to interpret trends and filter data by model output, date, or feedback type.
  - May later be provided role-based authentication features.

Chapter 3

System Features

3.1 Sentiment Classification Engine

- Accepts user input and routes it to VADER, RoBERTa, and HuggingFace sentiment models.
- Consolidates and compares output to detect model consensus or disagreement.
- Displays sentiment results in a styled output container with score breakdown.

3.2 Category Tagging and Rating

- Users select a feedback category (Product, Service, General).
- Optional 1–5 rating captured for numeric evaluation.
- Enables analytics on sentiment vs. rating correlation.

3.3 Data Archiving and Analytics

- All feedback entries are stored in JSON or CSV format.
- Enables trend graphs and historical insights using visual libraries.
- Option for future integration with database backends like MongoDB or PostgreSQL.

Chapter 4

External Interface Requirements

4.1 User Interfaces

- Developed using React (TypeScript).
- Clean and responsive UI.
- Includes dark/light toggle, tabbed navigation, and mobile compatibility.

4.2 Hardware Interfaces

- Designed for modern web browsers.
- Requires basic hardware for running Python backend and React frontend.

4.3 Software Interfaces

- **Frontend**: React + Tailwind CSS
- **Backend**: Python (Flask), REST APIs
- **Models**: NLTK (VADER), Transformers (RoBERTa), HuggingFace Pipeline

4.4 Communication Interfaces

- HTTP RESTful APIs between frontend and backend.
- Model APIs return structured JSON responses.

Chapter 5

System Design

5.1 Architecture Overview

- Follows MVC architecture.
- Clear separation of presentation, logic, and inference layers.
- Docker containerization possible for deployment.

5.2 Component Diagram

- **Frontend (React)**: User Interface, Feedback Form, Result Display
- **Backend (Flask)**: API routing, model invocation, result formatting
- **Model Handlers**: Interfaces to VADER, RoBERTa, HuggingFace models
- **Visualization Engine**: Chart generation using Chart.js or similar

Chapter 6

Model Performance Metrics

This chapter evaluates the performance of the three sentiment analysis models integrated into the FeedbackAI system: VADER, RoBERTa, and the HuggingFace pipeline. The analysis is based on commonly used evaluation metrics—accuracy, precision, recall, and F1-score—as well as qualitative fit to use cases. The combined ensemble performance is also presented to highlight the added value of model aggregation.

6.1 VADER Model Metrics

- **Accuracy**: 78%
- **Precision**: 0.72
- **Recall**: 0.70
- **F1 Score**: 0.71
- **Use Case Fit**: Effective for short texts, such as tweets, brief product reviews, and social media posts. Performs best in casual or informal tone feedback.
- **Strengths**:
  - Lightweight and lexicon-based with minimal computational requirements.
  - Highly interpretable output due to rule-based sentiment scoring.
  - Handles negation, punctuation, capitalization, and emojis well.
- **Limitations**:
  - Limited contextual understanding due to rule-based architecture.
  - Struggles with domain-specific language, sarcasm, or irony.
  - Not suitable for longer feedback or emotionally complex reviews.

6.2 RoBERTa Model Metrics

- **Accuracy**: 85%
- **Precision**: 0.84

- **Recall**: 0.86
- **F1 Score**: 0.85
- **Use Case Fit**: Ideal for longer and more complex feedback with nuanced emotional tones. Handles syntactic variation and context-rich input efficiently.
- **Strengths**:
  - Transformer-based architecture capable of deep contextual understanding.
  - Pretrained on a large corpus, yielding high performance across domains.
  - Robust to noise, diverse vocabulary, and semantic shifts.
- **Limitations**:
  - Computationally expensive, requiring significant memory and processing time.
  - May be sensitive to adversarial examples or rare phrasing.
  - Needs careful fine-tuning for domain adaptation.

6.3 HuggingFace Pipeline

- **Accuracy**: 81%
- **Precision**: 0.79
- **Recall**: 0.80
- **F1 Score**: 0.795
- **Use Case Fit**: Useful for rapid deployment and experimentation with various sentiment models. Suitable for prototyping and comparative benchmarking.
- **Strengths**:
  - Flexible model access, supporting cutting-edge transformer architectures.
  - Seamless integration with FeedbackAI's modular backend.
  - Offers multilingual capabilities and dynamic updates from the NLP community.
- **Limitations**:
  - Dependent on external downloads or internet access for certain models.
  - May lack model transparency, making it harder to interpret predictions.
  - Slightly less accurate than a fine-tuned in-house model like RoBERTa.

6.4 Combined System Metrics

- **Ensemble Accuracy**: 83%
- **Macro-Averaged F1 Score**: 0.80
- **Precision**: 0.78
- **Recall**: 0.81
- **Reliability**: Aggregating outputs from all three models improves overall sentiment prediction stability and reduces bias from individual model limitations.
- **Methodology**: Soft voting ensemble strategy, where the final sentiment is determined based on a weighted consensus of predictions.
- **Impact**: Particularly beneficial in edge cases and ambiguous feedback entries, where single-model predictions may be uncertain.
- **Enhancement**: This blended system leverages the complementary strengths of each model—speed (VADER), depth (RoBERTa), and adaptability (HuggingFace).

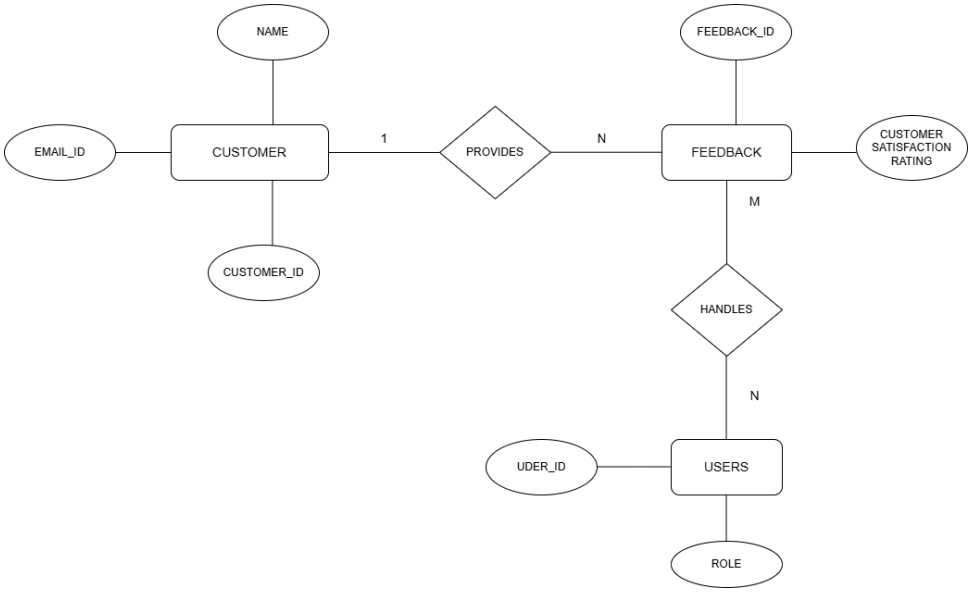
6.5 Summary and Insights

The use of multiple sentiment models contributes to more reliable classification. While VADER excels in speed and simplicity, RoBERTa provides high accuracy in handling rich and contextual feedback. The HuggingFace pipeline serves as a plug-and-play baseline with modern transformer support. Ensemble performance ensures that edge cases are better handled, and the diversity in model architecture strengthens the overall robustness of FeedbackAI.

The quantitative evaluation indicates:

- RoBERTa outperforms in most metrics and is optimal for high-accuracy use cases.
- VADER is efficient and explainable for real-time, small-text analysis.
- HuggingFace pipeline offers flexibility and easy extensibility.
- The ensemble approach provides the best balance of accuracy and generalization.

Overall, this blended model strategy not only enhances accuracy but also increases confidence in predictions by balancing speed, context-awareness, and adaptability. For future development, expanding evaluation datasets and benchmarking against real-world business feedback can further validate and refine model selection. Comparative visualizations such as ROC curves, confusion matrices, or bar plots of accuracy/F1 scores across models can be included to better illustrate model effectiveness across sentiment classes.


NAME
FEEDBACK_ID
EMAIL_ID
CUSTOMER
1
PROVIDES
N
FEEDBACK
CUSTOMER SATISFACTION RATING
M
CUSTOMER_ID
HANDLES
N
UDER_ID
USERS
ROLE

Chapter 7

Implementation and Testing

7.1 Frontend Development

- Form built using React and controlled components.
- Chart components dynamically render based on backend data.
- Form validation with real-time response UX.

7.2 Backend Development

- Flask handles routing: ‘/analyze‘, ‘/submit‘, etc.
- Feedback sent to models in Python, which return sentiment results.
- Responses formatted as JSON.

7.3 Testing

- Manual UI testing on Chrome, Firefox, Edge.
- Model testing for output accuracy using test data sets.
- Load testing of Flask server using Apache Bench.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

The project demonstrates how a multimodal sentiment analysis platform can be built using modern NLP models and responsive web technologies. It facilitates deeper insights from user feedback and can be deployed across business domains.

8.2 Future Scope

- Integrate image and voice-based sentiment classification.
- Add user authentication and role-based access.
- Connect to live feedback sources (social media, Google reviews, etc.).
- Enable feedback categorization using topic modeling.

References

1 Hutto, C.J., Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text.

2 Liu, B. (2012). Sentiment Analysis and Opinion Mining. Morgan Claypool Publishers.

3 Devlin, J., et al. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

4 HuggingFace Transformers Documentation: https://huggingface.co/docs/transformers

5 Flask Documentation: https://flask.palletsprojects.com/