SOMAIYA VIDYAVIHAR UNIVERSITY

HINGLISH SENTIMENT ANALYSIS

Submitted In Partial Fulfillment of Requirements
For the Degree Of

Honors in Data Science and Analytics (Offered by Department of Computer Engineering)

By

Sarthak Pokale
Roll No: 16010122146

Khushi Poojary
Roll No: 16010122147

Hyder Presswala
Roll No: 16010122151

Vedant Rathi
Roll No: 16010122154

Guide

Veena Badgujar

DECLARATION

We declare that this written report submission represents the work done based on our and / or others' ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in my submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

Signature of the Student **Roll No. 16010122146**	**Signature of the Student** **Roll No. 16010122147**
Signature of the Student **Roll No. 16010122151**	**Signature of the Student** **Roll No. 16010122154**

Date: 09 January 2026

Place: Mumbai-77

Abstract

Understanding patient sentiment from online drug reviews is an important component of modern healthcare analytics, as such reviews often contain valuable insights regarding drug effectiveness, side effects, and overall patient satisfaction. However, the large volume of unstructured textual data makes manual analysis impractical, while existing automated sentiment analysis approaches in the medical domain often lack systematic comparative evaluation. In particular, practitioners face uncertainty when choosing between computationally efficient classical machine learning models and more resource-intensive deep learning architectures.This project presents a comprehensive comparative analysis of Classical Machine Learning and Deep Learning approaches for drug review sentiment classification. Using the UCI Drug Review Dataset, a binary sentiment classification task was formulated by categorizing reviews into positive and negative classes based on user ratings, while neutral reviews were excluded to strengthen decision boundaries. A robust preprocessing pipeline was designed, with distinct strategies tailored to each paradigm. Classical Machine Learning employed TF–IDF vectorization combined with Logistic Regression as a strong baseline, while the Deep Learning approach utilized a Long Short-Term Memory (LSTM) network to capture sequential and contextual information in medical text.A controlled experimental framework was adopted to ensure fair comparison across models, including identical data splits, consistent evaluation metrics, and targeted preprocessing interventions such as stopword retention for context preservation in deep learning. Model performance was evaluated using accuracy, precision, recall, and F1-score, with particular emphasis on negative sentiment detection due to its importance in identifying adverse drug reactions. Experimental results demonstrate that the LSTM model achieves superior performance, with an absolute accuracy improvement of 2.11% and a significant 8.51% increase in negative recall compared to the classical baseline, albeit at higher computational cost.The project further includes qualitative error analysis and a computational cost–benefit evaluation to guide practical deployment decisions. An interactive Streamlit-based dashboard was developed to enable real-time prediction and comparison of both models. Overall, this work provides a structured analytical blueprint for selecting sentiment analysis models in healthcare applications, balancing predictive accuracy, interpretability, and computational efficiency.

Key words**:** *Drug Review Sentiment Analysis, Classical Machine Learning, Deep Learning, LSTM, Logistic Regression, Healthcare NLP, Comparative Analysis.*

CONTENTS

Chapter 4: Implementation and Experimentation

4.1 Implementation Environment

4.2 Dataset Preparation and Loading

4.3 Preprocessing Implementation

4.4 Feature Engineering and Tokenization Implementation

4.5 Model Training and Configuration

4.6 Interpretation and Comparative Insights

4.7 Summary

Chapter 5: Results and Discussion

5.1 Overview of Experiments

5.2 Model Performance Comparison

5.3 Qualitative Error Analysis

5.4 Practical Usability and Cost–Performance Discussion

5.5 Summary

Chapter 6: Conclusion and Future Work

6.1 Conclusion
6.2 Key Contributions of the Project Implementation of a Hinglish NLP Pipeline
6.3 Future Work
6.4 Summary

Bibliography

Acknowledgment

Chapter 1: Introduction

1.1 Background

The rapid growth of social media and digital communication platforms has resulted in an enormous volume of user-generated textual content. In multilingual societies like India, this content is rarely confined to a single language. Instead, users frequently employ **code-mixed languages**, with **Hinglish**—a combination of Hindi and English written primarily in Roman script—being the most dominant form. Hinglish is widely used on platforms such as Twitter, Instagram, YouTube comments, and product review websites.

Hinglish sentiment analysis plays a crucial role in understanding public opinion, customer feedback, political discourse, and social trends in the Indian context. Studies indicate that a significant percentage of Indian social media posts contain mixed Hindi-English text, making Hinglish a critical linguistic medium for opinion mining and business intelligence.

However, traditional Natural Language Processing (NLP) systems are primarily designed for **monolingual text**, usually English. Hinglish text violates these assumptions by mixing vocabulary, grammar, and linguistic structures from two languages within the same sentence. Variations in spelling, informal grammar, transliteration inconsistencies, and region-specific slang further complicate analysis. As a result, standard sentiment classifiers often fail to correctly interpret sentiment polarity in code-mixed text.

Sentiment analysis, a key task in NLP, focuses on identifying emotional polarity—positive, negative, or neutral—from text. While classical machine learning techniques such as Naïve Bayes, Support Vector Machines (SVM), and Logistic Regression have been widely used due to their simplicity and efficiency, they struggle to capture contextual dependencies in Hinglish text. In contrast, deep learning models like Long Short-Term Memory (LSTM) networks and transformer-based models offer improved contextual understanding but require higher computational resources and careful preprocessing.

This project is positioned within this context, aiming to systematically analyze and compare **Classical Machine Learning and Deep Learning approaches** for Hinglish sentiment analysis.

1.2 Motivation

India has over half a billion bilingual or multilingual speakers, making code-mixed communication the norm rather than the exception. Ignoring Hinglish content leads to biased or incomplete sentiment analysis results, which can negatively impact applications such as social media monitoring, market research, and public opinion analysis.

Despite the importance of Hinglish sentiment analysis, existing tools and datasets are limited. Many sentiment analysis systems are trained exclusively on English data and perform poorly on Hinglish text. Moreover, while several studies propose individual models for Hinglish sentiment classification, **systematic comparative evaluations** between classical and deep learning approaches remain scarce.

Another motivating factor is the **sensitivity of Hinglish sentiment analysis to preprocessing choices**. Decisions related to stopword removal, transliteration, normalization, and negation handling significantly influence model performance. Words such as *"nai"*, *"nahi"*, and *"not"* often reverse sentiment polarity, and improper preprocessing can lead to incorrect classification.

Additionally, real-world deployment requires balancing **accuracy, interpretability, and computational efficiency**. Deep learning models may achieve higher accuracy but are resource-intensive, while classical models are lightweight but may fail to capture linguistic nuances. This project is motivated by the need to quantify these trade-offs and provide practical guidance for selecting appropriate sentiment analysis models for Hinglish text.

1.3 Scope of the Project

The scope of this project focuses on the design, implementation, and evaluation of a bilingual sentiment analysis system for Hinglish text. The primary objective is to analyze sentiment in Hindi–English code-mixed text using classical machine learning models and a voting-based ensemble approach.

The analysis is limited to English–Hindi code-mixed text written in Roman script, sourced from publicly available Twitter sentiment datasets. The project addresses a supervised sentiment classification problem, categorizing text into positive and negative sentiment classes.

The scope includes:

- Dataset selection, cleaning, and preprocessing of noisy Twitter data
- Text normalization including spelling correction, lemmatization, contraction expansion, and negation handling
- Feature extraction focused on sentiment-bearing linguistic features such as adjectives, adverbs, and abstract nouns
- Translation-based handling of bilingual words to resolve Hindi–English language ambiguity
- Model training using multiple classical machine learning classifiers
- Development of a voting-based hybrid ensemble model

- Evaluation using standard performance metrics such as accuracy, precision, recall, F1-score, and confusion matrix analysis

Advanced deep learning architectures, transformer-based models, and multi-class emotion classification are outside the current scope. The project is intended as an academic research prototype rather than a full-scale production deployment.

1.4 Brief Description of the Project Undertaken

This project undertakes a comprehensive study of sentiment analysis techniques applied to Hinglish text using classical machine learning approaches. A labeled Twitter dataset is prepared by selecting relevant textual fields and cleaning noisy elements such as URLs, emojis, HTML tags, and special characters. Linguistically important tokens, especially negations, are carefully handled during preprocessing.

A bilingual sentiment analysis pipeline is developed consisting of the following components:

1. **Classical Machine Learning Pipeline**
   Text is converted into numerical representations using feature extraction techniques, followed by training classifiers such as Logistic Regression, Naïve Bayes, Support Vector Machines, Stochastic Gradient Descent, and Maximum Entropy models. This pipeline emphasizes efficiency, interpretability, and robustness.

2. **Hybrid Ensemble Pipeline**
   A voting-based ensemble model combines predictions from multiple classifiers to improve overall sentiment classification accuracy and reduce individual model bias.

A controlled experimental setup ensures fair evaluation using identical datasets, splits, and performance metrics. The project also analyzes the impact of preprocessing strategies such as negation handling, normalization, and bilingual translation on sentiment prediction performance.

The final system enables both qualitative and quantitative evaluation of Hinglish sentiment classification and highlights the effectiveness of ensemble learning for code-mixed text.

1.5 Organisation of the Report

This report is organized into six chapters:

- **Chapter 1** introduces the project, outlining the background, motivation, scope, and objectives of Hinglish sentiment analysis.
- **Chapter 2** presents a detailed literature survey of existing research in code-mixed and Hinglish sentiment analysis.
- **Chapter 3** describes the system architecture and overall methodology.
- **Chapter 4** details the implementation and experimental setup.
- **Chapter 5** discusses results, error analysis, and comparative performance.

- **Chapter 6** concludes the report and outlines future research directions.

Chapter 2: Literature Survey

1.1 Introduction to Literature Survey

Sentiment analysis of code-mixed text has emerged as an important research area due to the prevalence of multilingual communication on social media. Hinglish sentiment analysis presents unique challenges related to language mixing, informal spelling, and contextual ambiguity. This chapter reviews existing research on Hinglish and code-mixed sentiment analysis, focusing on methodologies, limitations, and identified research gaps.

1.2 Sentiment Analysis of Code-Mixed Text

Joshi et al. (2016) introduced one of the earliest studies on sentiment analysis of Hindi-English code-mixed text, highlighting the inadequacy of monolingual sentiment models. Their work demonstrated that sub-word level representations significantly improve performance on code-mixed data. However, the approach required extensive linguistic preprocessing.

1.3 Deep Learning Approaches for Hinglish Sentiment Analysis

Singh and Lefever (2020) explored cross-lingual word embeddings for Hinglish sentiment classification. Their BiLSTM-based model effectively captured contextual dependencies but suffered from increased computational complexity and limited generalization to unseen slang.

Bhange and Kasliwal (2020) fine-tuned transformer-based models such as mBERT for Hinglish sentiment detection. While these models achieved improved accuracy, they struggled with noisy transliterations and required large labeled datasets.

1.4 Classical Machine Learning Approaches

Several studies have evaluated classical machine learning algorithms such as Naïve Bayes, Logistic Regression, and SVM for Hinglish sentiment classification using TF–IDF features. These models demonstrated strong baseline performance and low computational cost but failed to handle negation and sarcasm effectively.

1.5 Linguistic Challenges in Hinglish Sentiment Analysis

Research highlights challenges such as:

- Negation handling (“nai pasand”)
- Informal grammar and spelling variations
- Transliteration inconsistencies
- Lack of Hinglish-specific NLP resources

Most existing studies focus on coarse sentiment polarity and do not address deeper linguistic phenomena.

1.6 Impact of Preprocessing on Hinglish Sentiment Models

Studies emphasize that preprocessing decisions significantly affect sentiment classification performance. Proper handling of negation, spelling correction, and bilingual vocabulary improves accuracy, especially for classical models.

1.7 Summary of Literature Findings

The literature indicates that:

- Classical models are efficient and interpretable
- Performance depends heavily on preprocessing quality
- Ensemble learning improves robustness
- Controlled comparative evaluations are limited

1.8 Research Gap Identified

The following gaps are identified:

- Limited use of ensemble models for Hinglish sentiment analysis
- Insufficient analysis of preprocessing impact
- Lack of detailed error and confusion matrix analysis

1.9 Objectives of the Thesis Work

- Develop a classical machine learning-based Hinglish sentiment analysis system
- Analyze preprocessing impact on sentiment classification
- Evaluate individual classifiers and an ensemble model
- Provide practical insights for real-world deployment

Chapter 3: Project Design

3.1 System Architecture of the Hinglish Sentiment Analysis System

The Hinglish Sentiment Analysis system is designed as a modular Natural Language Processing framework focused on classical machine learning–based sentiment classification of Hindi–English code-mixed text.

The overall system architecture consists of the following components:

- **Data Input Layer**
  This layer accepts Hinglish text data collected from publicly available Twitter datasets. The input consists of code-mixed Hindi–English tweets written in Roman script, annotated with sentiment labels.

- **Preprocessing and Feature Engineering Layer**
  This layer performs extensive text preprocessing, including noise removal, spelling correction, lemmatization, negation handling, and bilingual word translation. Linguistically important sentiment-bearing words are retained for feature extraction.

- **Model Processing Layer**
  The processed text is passed through multiple classical machine learning classifiers such as Logistic Regression, Naïve Bayes, Support Vector Machines, Stochastic Gradient Descent, and Maximum Entropy.

- **Ensemble Decision Layer**
  A voting-based hybrid ensemble combines predictions from all individual classifiers to produce the final sentiment output.

This architecture ensures robustness, interpretability, and reproducibility while maintaining low computational complexity.

3.2 End-to-End Workflow for Hinglish Sentiment Classification

The end-to-end workflow is divided into four major stages:

1) Data Collection and Selection
A labeled Twitter dataset is selected, and a balanced subset of tweets is used for sentiment classification.

2) Data Preprocessing
Raw tweets undergo cleaning steps such as removal of URLs, emojis, HTML tags, punctuation, and user mentions. Normalization, spelling correction, lemmatization, and negation handling are applied.

3) Feature Extraction and Model Training
Sentiment-rich features such as adjectives, adverbs, and abstract nouns are extracted. Multiple classical machine learning models are trained independently on the processed dataset.

4) Ensemble Prediction and Evaluation

Predictions from individual classifiers are combined using majority voting, and performance is evaluated using standard metrics.

3.3 System Flow Design

The system flow begins with loading the Hinglish Twitter dataset, followed by data cleaning and preprocessing. Each tweet is labeled as positive or negative.

The cleaned data is then passed through:

- A feature extraction module that selects sentiment-bearing words
- Multiple classical classifiers that independently predict sentiment
- A voting-based ensemble model that generates the final prediction

Evaluation results are analyzed using confusion matrices and classification reports.

3.4 Dataset Description and Preprocessing Pipeline

The project uses a publicly available Twitter sentiment dataset containing code-mixed Hindi–English text written in Roman script.

Preprocessing Pipeline

Preprocessing includes:

- Removal of HTML tags, URLs, emojis, punctuation, and special characters
- Conversion of text to lowercase
- Removal of unnecessary stopwords while retaining negation words
- Spelling correction and lemmatization
- Replacement of negated words with antonyms
- Translation of Hindi words written in Roman script into English

The final dataset consists of cleaned tweets ready for feature extraction.

3.5 Feature Engineering and Tokenization Strategy

Only linguistically significant words contributing to sentiment are retained.

- Feature Selection
- Adjectives
- Adverbs
- Abstract nouns

These features are extracted using a predefined linguistic dictionary, avoiding reliance on outdated POS taggers.

Vectorization
The extracted features are converted into numerical form using suitable vectorization techniques for classical machine learning models.

3.6 Model Development and Selection

The following classifiers are implemented:

- Logistic Regression
- Naïve Bayes
- Support Vector Machine
- Stochastic Gradient Descent
- Maximum Entropy

Each model is trained independently using the same training dataset to ensure fair comparison.

3.7 Comparative Evaluation Framework

A controlled experimental framework is adopted where:

- The same dataset is used across all models
- Identical train–test splits are applied
- Common evaluation metrics are used

Performance is measured using accuracy, precision, recall, F1-score, and confusion matrix analysis.

3.8 Interpretability and Error Analysis Design

Error analysis focuses on identifying:

- Misclassification due to negation
- Errors caused by bilingual ambiguity
- Impact of spelling and transliteration variations

Feature weight analysis is used to interpret classical model predictions.

3.9 Summary

This chapter described the system architecture, data flow, preprocessing pipeline, feature engineering strategy, model selection, and evaluation framework used for Hinglish sentiment analysis

Chapter 4: Implementation and Experimentation

1.1 Implementation Environment

The system is implemented using Python and standard NLP and machine learning libraries such as NLTK, Scikit-learn, and supporting translation tools.

1.2 Dataset Preparation and Loading

The dataset is loaded from CSV files. Missing and duplicate records are removed. A balanced subset of tweets is selected for training and testing.

1.3 Preprocessing Implementation

Text preprocessing includes:

- HTML parsing and URL removal
- Emoji conversion and removal
- Case normalization
- Stopword filtering with negation retention
- Spelling correction
- Lemmatization and negation substitution

1.4 Feature Engineering and Tokenization Implementation

Extracted sentiment-bearing words are transformed into feature vectors suitable for classical classifiers.

1.5 Model Training and Configuration

Each classifier is trained using the same training set. Hyperparameters are tuned to optimize classification performance while preventing overfitting

1.6 Interpretation and Comparative Insights

The voting-based hybrid ensemble consistently outperforms individual classifiers by combining their strengths and reducing bias.

1.7 Summary

This chapter presented the implementation details and experimental evaluation of the Hinglish sentiment analysis system.

Chapter 5: Results and Discussion

5.1 Overview of Experiments

All experiments are conducted on a held-out test set. Evaluation focuses on both overall and class-wise performance.

5.2 Model Performance Comparison

Results demonstrate that the ensemble model achieves the highest accuracy and F1-score compared to individual classifiers.

5.3 Qualitative Error Analysis

A qualitative analysis of misclassified samples revealed important insights into the strengths and weaknesses of both approaches.

The Classical Machine Learning model frequently misclassified sentences involving negation or contrast. Phrases such as *"movie achhi nahi hai"* or *"service theek hai but response slow hai"* were often incorrectly labeled due to the removal of stopwords and the lack of sequential understanding. Since TF–IDF representations ignore word order, the presence of positive keywords alone was sometimes sufficient to skew predictions.

The LSTM model handled such cases more effectively by analyzing the entire word sequence. However, it struggled with:

- Sarcastic expressions common in Hinglish
- Extremely short texts with limited context
- Rare slang terms and regional expressions

These limitations indicate that while deep learning models improve contextual understanding, they still lack robust handling of sarcasm and evolving informal language

5.4 Practical Usability and Cost–Performance Discussion

The experimental results highlight a trade-off between computational efficiency and predictive sensitivity.

The Classical Machine Learning approach offers:

- Fast training and inference
- Low hardware requirements
- High interpretability

This makes it suitable for large-scale, cost-sensitive applications where latency is critical.

In contrast, the Deep Learning approach requires higher computational resources and longer training time but delivers superior performance in capturing contextual and negation-based sentiment. For applications such as social media monitoring, opinion mining, or policy analysis—where misinterpreting negative sentiment can have significant consequences—the Deep Learning model is the preferred choice.

Thus, the selection of the model should be guided by application requirements rather than accuracy alone.

5.5 Summary

This chapter presented a comprehensive evaluation of the Hinglish sentiment analysis system through both quantitative metrics and qualitative analysis. The results demonstrated that while Classical Machine Learning provides a strong and efficient baseline, Deep Learning models offer superior contextual understanding and improved negative sentiment detection. The analysis reinforces the importance of sequence modeling for code-mixed language processing and validates the architectural decisions made in earlier chapters.

Chapter 6: Conclusion and Future Work

6.1 Conclusion

This project successfully developed and evaluated a sentiment analysis system for Hinglish (Hindi–English code-mixed) text using classical machine learning techniques. The work addressed key challenges of Hinglish sentiment analysis such as code-switching, informal spelling, transliteration inconsistencies, and negation handling, which are commonly observed in social media data.

By applying robust preprocessing and carefully selected linguistic features, the system was able to accurately classify Hinglish tweets into positive and negative sentiment categories. The use of multiple classical classifiers allowed for a comparative analysis of their strengths and limitations, while ensuring interpretability and computational efficiency. Overall, the results demonstrate that classical machine learning models remain effective for Hinglish sentiment analysis when supported by appropriate preprocessing strategies.

6.2 Key Contributions of the Project Implementation of a Hinglish NLP Pipeline

The major contributions of this project include:

- Development of a Hinglish-specific preprocessing pipeline to handle noisy and code-mixed text
- Effective handling of bilingual vocabulary through translation-based normalization
- Implementation and evaluation of multiple classical machine learning classifiers
- Construction of a voting-based hybrid ensemble model to improve sentiment classification accuracy
- Detailed performance and error analysis using standard evaluation metrics

6.2.1 Comparative Analysis

A controlled experimental setup was used to compare the performance of individual classifiers and the ensemble model. While individual models showed competitive results, their predictions varied due to differences in learning behavior. The voting-based ensemble model consistently achieved better performance by combining the strengths of individual classifiers and reducing classification errors.

Error analysis revealed that most misclassifications occurred in tweets containing sarcasm, ambiguous bilingual expressions, or extremely short text. These observations highlight the inherent complexity of Hinglish sentiment analysis and the limitations of keyword-based approaches.

6.2.2 Practical Deployment Using Streamlit

The findings of this project have practical relevance for applications such as social media monitoring, opinion mining, and customer feedback analysis in multilingual environments. Classical machine learning models offer advantages in terms of low computational cost, faster training, and better interpretability, making them suitable for real-world deployment where resources may be limited.

6.3 Future Work

Several directions can be explored to extend this work:

- Expanding the dataset to include Hinglish text from multiple domains such as product reviews.
- Improving handling of sarcasm, idiomatic expressions, and emerging slang
- Exploring lightweight transformer-based models for comparison under limited computational resources.
- Incorporating multi-class sentiment or emotion classification for finer-grained analysis

6.4 Summary

This chapter summarized the outcomes of the project and discussed its contributions and limitations. The study confirms that classical machine learning models, when combined with strong preprocessing and ensemble learning, provide an effective and practical solution for sentiment analysis of Hinglish text.

Acknowledgment

The successful completion of this project would not have been possible without the guidance, encouragement, and support of many individuals and institutions. We feel truly privileged to express our heartfelt gratitude to all of them.

We sincerely thank **K. J. Somaiya School of Engineering** for providing us with the opportunity, resources, and an encouraging academic environment to undertake and complete this project.

We would like to extend our deepest appreciation to our project guide, **Veena Badgujar**, for her invaluable guidance, constant support, and insightful feedback throughout the course of this work. Her patience, motivation, and expertise played a pivotal role in shaping our understanding of the subject and in achieving the project objectives.

We are also grateful to the **Department of Computer Engineering** and all the faculty members for their assistance and constructive suggestions at various stages of the project.

Finally, we thank our families and peers for their encouragement and understanding, which inspired us to persevere and bring this project to successful completion.