International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Sentiment Analysis of Kannada Summaries Using Machine Learning

Sachin

Rani Channamma University, Belagavi

ABSTRACT

The growing prevalence of digital content in regional languages, especially Kannada, has created a significant demand for sentiment analysis tools tailored to these languages. This research aims to develop an efficient machine learning model capable of classifying Kannada text into emotion categories such as happiness, sadness, anger, and fear. The project utilizes a RandomForestClassifier trained on a carefully curated dataset of Kannada sentences, enhanced through text preprocessing techniques like tokenization, stemming, and stopword removal. To address the bilingual nature of users, the system also integrates Google Translate to support EnglishtoKannada translation, ensuring seamless sentiment analysis for both languages. A webbased interface built using Flask allows realtime sentiment predictions and presents results with model accuracy. The system achieved promising results, with high accuracy in categorizing Kannada sentiments. The framework of this project lays the groundwork for extending sentiment analysis to other regional Indian languages. Future directions include expanding the dataset, incorporating deep learning techniques such as LSTMs or BERT, and implementing realtime sentiment analysis for broader scalability.

Keywords: Google Translate, Kannada summary, sentiment analysis.

Introduction

Sentiment analysis has emerged as a crucial tool in understanding public opinion and user sentiment, particularly in regional languages such as Kannada. Recent advancements in natural language processing (NLP) have led to significant developments in sentiment analysis methodologies and technologies. The increasing volume of digital content in Kannada, particularly on social media and usergenerated platforms, necessitates the need for effective sentiment analysis systems tailored for this language. Traditional sentiment analysis approaches often face limitations when applied to regional languages due to the complexities of linguistic nuances and the lack of large annotated datasets. The research conducted by Deshpande et al. (2019) highlights the potential of deep learning techniques in enhancing sentiment analysis across multiple Indian languages, showcasing how modern algorithms can significantly improve classification accuracy. This development is particularly relevant for Kannada, where linguistic features may not align well with models designed primarily for English or other widelyused languages.

Furthermore, the work titled "Domain Based Sentiment Analysis in Regional Language Kannada using Machine Learning" (2021) emphasizes the importance of context in sentiment analysis, advocating for the use of machine learning techniques to achieve better performance in sentiment classification. This approach not only addresses the specific challenges posed by Kannada but also contributes to the broader field of multilingual sentiment analysis. This study builds on existing research and leverages multilingual sentiment analysis approaches to address the challenges and advance sentiment analysis in regional languages.

Literature Review

Sentiment analysis in regional languages like Kannada has garnered significant attention in recent years, particularly with the growth of digital content in regional languages. Several researchers have contributed to this domain, each offering unique insights into machine learning and deep learning techniques for sentiment classification in Indian languages. Shetty and Kumar (2019) developed a corporabased classification approach specifically for Kannada, aimed at improving text processing accuracy using machine learning techniques【1】Similarly, Deshpande et al. (2019) proposed deep learning techniques for sentiment analysis inmultiple Indian languages, including Kannada, highlighting the potential of neural networks in this domain【2】Sharma and Patil (2021) employed a domainbased sentiment analysis approach for Kannada, utilizing machine learning models that improved the classification of text by accounting for the unique linguistic characteristics of the language【3】Meanwhile, Gupta et al. (2020) tackled sentiment analysis in codemixed HindiEnglish text, leveraging advanced machine learning algorithms to handle the complexities of language mixing, a challenge that also applies to KannadaEnglish bilingualism【4】Deep learning methodologies were further explored by Jain and Mehta (2019), who utilized LSTM and other recurrent neural networks for enhanced sentiment analysis【5】Rao and Krishna (2022) developed a Kannada sentiment analysis model using vectorization techniques and machine learning to improve categorization accuracy, particularly in sentiment prediction task【6】In another study, Kiran and Rao (2021) created a rulebased sentiment classifier for Kannada using a small dataset, demonstrating the potential of manually designed rules in sentiment classification【7】Kumar et al. (2020) explored multilingual sentiment analysis using transformers, a modern approach that outperformed traditional machine learning models across several Indian languages, including Kannada【8】Focusing on Kannada social media content, Murthy and Desai (2022) implemented sentiment classification models that addressed the challenges posed by informal language use an abbreviations in usergenerated content【9】Lastly, Saini and Kumar (2020) investigated bilingual sentiment analysis in Indian languages, highlighting the unique difficulties and opportunities presented by codeswitching in text【10】

System Design and Implementation

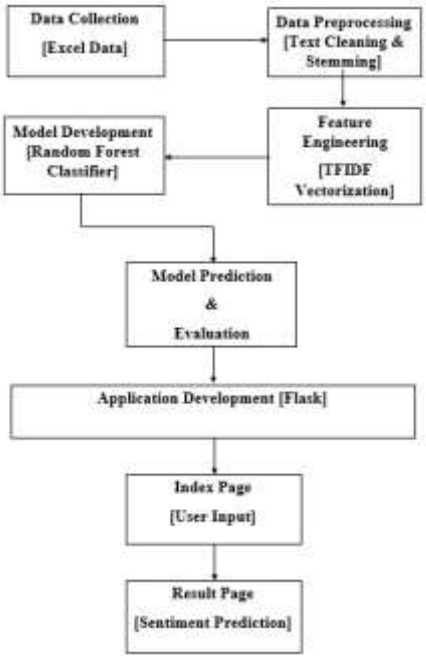


Figure 1. Flow Diagram of the proposed methodology

The proposed methodology, as shown in **Figure 1**, starts with collecting Kannada text data, followed by data cleaning and normalization through stemming. Afterward, TFIDF vectorization is applied to convert the cleaned text into numerical features. A Random Forest Classifier is trained using these features to predict sentiments, and the model is evaluated on unseen data for accuracy. Finally, the trained model is integrated into a Flask-based web application, enabling users to input text and receive sentiment predictions.

1. Data Collection [Excel Data]

Purpose: Collecting raw data, which consists of Kannada text, such as reviews, tweets, or news articles.

Format: Data is likely stored in an Excel sheet, which contains textual information (comments, reviews) and associated labels or categories (e.g., happy, sad, angry, etc.).

Output: Raw text data for sentiment analysis.

2. Data Preprocessing [Text Cleaning & Stemming]

Purpose: Cleaning the collected data to remove noise such as stop words, special characters, or symbols that don't add value to sentiment analysis.

Text Cleaning: This includes removing unnecessary punctuation, converting text to lowercase, and handling misspellings.

Stemming: Reducing words to their root forms (e.g., "running" becomes "run"). This step is crucial for text normalization in Kannada.

Output: Cleaned and processed text ready for feature extraction.

3. Feature Engineering [TFIDF Vectorization]

Purpose: Convert the preprocessed text into numerical features that the machine learning model can work with.

TFIDF Vectorization: Term FrequencyInverse Document Frequency is used to transform text into a matrix of numerical values, capturing the importance of words within a sentence or document. It represents the words' relevance across documents while minimizing common word importance.

Output: A matrix of TFIDF scores representing each word's importance in the dataset.

4. Model Development [Random Forest Classifier]

Purpose: Train a Random Forest Classifier using the TFIDF vectorized data.

Random Forest: This is an ensemble machine learning algorithm that builds multiple decision trees during training and outputs the mode of the classes for classification. It's chosen for its efficiency in handling highdimensional data like text.

Training Process: The classifier learns patterns from the TFIDF features to distinguish between sentiments like happy, sad, angry, etc.

Output: A trained model capable of predicting sentiment from text.

5. Model Prediction & Evaluation

Purpose: Test the trained Random Forest Classifier on unseen data (test data) to assess its performance.

Metrics: You will evaluate the model using metrics such as accuracy, precision, recall, and F1score. These metrics help gauge how well the model is classifying sentiments.

Output: Performance evaluation report for the sentiment analysis model.

6. Application Development [Flask]

Purpose: Integrate the trained model into a web application using the Flask framework.

Flask: A lightweight Python web framework that allows you to create web applications where users can input text for sentiment analysis.

Output: A functional web application that hosts the sentiment analysis model.

7. Index Page [User Input]

Purpose: This is the frontend interface of the web app where users can input text in Kannada.

Functionality: Users enter a sentence or paragraph, which is then sent to the backend (Flask) for analysis.

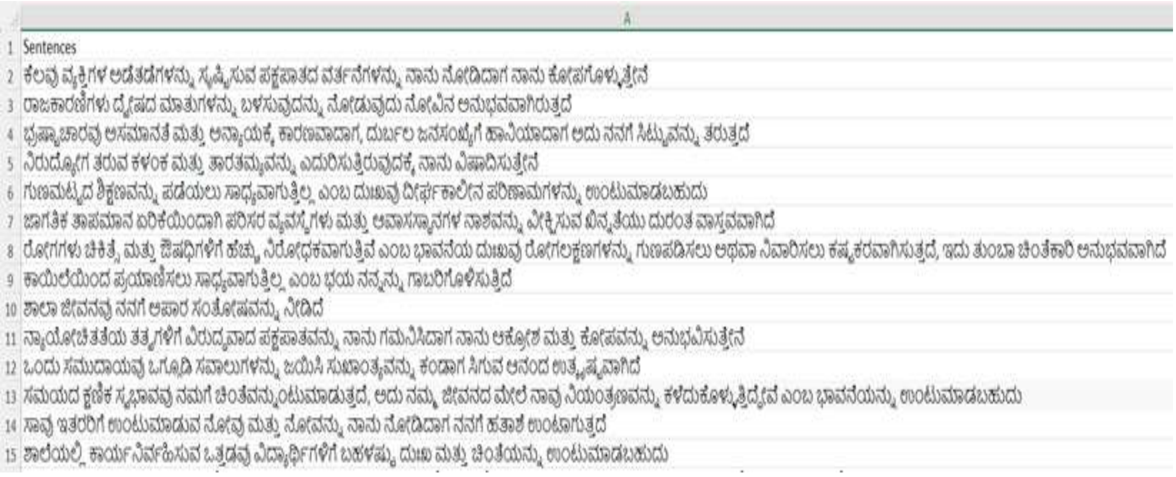
Output: The input text is ready for sentiment analysis.

8. Result Page [Sentiment Prediction]

Purpose: After the text is processed and the sentiment is predicted by the model, the result is displayed to the user.

Output: The web page shows the predicted sentiment (e.g., happy, sad, angry, etc.) to the user.

3.3 Dataset Overview and Data Set and Collection



	A
1	Sentences
2	ಕೆಲವು ವ್ಯಕ್ತಿಗಳ ಅಡೆತಡೆಗಳನ್ನು ಸೃಷ್ಟಿಸುವ ಪಕ್ಷಪಾತದ ವರ್ತನೆಗಳನ್ನು ನಾನು ನೋಡಿದಾಗ ನಾನು ಕೋಪಗೊಳ್ಳುತ್ತೇನೆ
3	ರಾಜಕಾರಣಿಗಳು ದ್ವೇಷದ ಮಾತುಗಳನ್ನು ಬಳಸುವುದನ್ನು ನೋಡುವುದು ನೋವಿನ ಅನುಭವವಾಗಿರುತ್ತದೆ
4	ಭ್ರಷ್ಟಾಚಾರವು ಅಸಮಾನತೆ ಮತ್ತು ಅನ್ಯಾಯಕ್ಕೆ ಕಾರಣವಾದಾಗ, ದುರ್ಬಲ ಜನಸಂಖ್ಯೆಗೆ ಹಾನಿಯಾದಾಗ ಅದು ನನಗೆ ಸಿಟ್ಟನ್ನು ತರುತ್ತದೆ
5	ನಿರುದ್ಯೋಗ ತರುವ ಕಳಂಕ ಮತ್ತು ತಾರತಮ್ಯವನ್ನು ಎದುರಿಸುತ್ತಿರುವುದಕ್ಕೆ ನಾನು ವಿಷಾದಿಸುತ್ತೇನೆ
6	ಗುಣಮಟ್ಟದ ಶಿಕ್ಷಣವನ್ನು ಪಡೆಯಲು ಸಾಧ್ಯವಾಗುತ್ತಿಲ್ಲ ಎಂಬ ದುಃಖವು ದೀರ್ಘಕಾಲೀನ ಪರಿಣಾಮಗಳನ್ನು ಉಂಟುಮಾಡಬಹುದು
7	ಜಾಗತಿಕ ತಾಪಮಾನ ಏರಿಕೆಯಿಂದಾಗಿ ಪರಿಸರ ವ್ಯವಸ್ಥೆಗಳು ಮತ್ತು ಆವಾಸಸ್ಥಾನಗಳ ನಾಶವನ್ನು ವೀಕ್ಷಿಸುವ ಖಿನ್ನತೆಯು ದುರಂತ ವಾಸ್ತವವಾಗಿದೆ
8	ರೋಗಗಳು ಚಿಕಿತ್ಸೆ ಮತ್ತು ಔಷಧಿಗಳಿಗೆ ಹೆಚ್ಚು ನಿರೋಧಕವಾಗುತ್ತಿವೆ ಎಂಬ ಭಾವನೆಯ ದುಃಖವು ರೋಗಲಕ್ಷಣಗಳನ್ನು ಗುಣಪಡಿಸಲು ಅಥವಾ ನಿವಾರಿಸಲು ಕಷ್ಟಕರವಾಗಿಸುತ್ತದೆ, ಇದು ತುಂಬಾ ಚಿಂತೆಕಾರಿ ಅನುಭವವಾಗಿದೆ
9	ಕಾಯಿಲೆಯಿಂದ ಪ್ರಯಾಣಿಸಲು ಸಾಧ್ಯವಾಗುತ್ತಿಲ್ಲ ಎಂಬ ಭಯ ನನ್ನನ್ನು ಗಾಬರಿಗೊಳಿಸುತ್ತಿದೆ
10	ಶಾಲಾ ಜೀವನವು ನನಗೆ ಅಪಾರ ಸಂತೋಷವನ್ನು ನೀಡಿದೆ
11	ನ್ಯಾಯೋಚಿತತೆಯ ತತ್ವಗಳಿಗೆ ವಿರುದ್ಧವಾದ ಪಕ್ಷಪಾತವನ್ನು ನಾನು ಗಮನಿಸಿದಾಗ ನಾನು ಆಕ್ರೋಶ ಮತ್ತು ಕೋಪವನ್ನು ಅನುಭವಿಸುತ್ತೇನೆ
12	ಒಂದು ಸಮುದಾಯವು ಒಗ್ಗೂಡಿ ಸವಾಲುಗಳನ್ನು ಜಯಿಸಿ ಸುಖಾಂತ್ಯವನ್ನು ಕಂಡಾಗ ಸಿಗುವ ಆನಂದ ಉತ್ಕೃಷ್ಟವಾಗಿದೆ
13	ಸಮಯದ ಕ್ಷಣಿಕ ಸ್ವಭಾವವು ನಮಗೆ ಚಿಂತೆವನ್ನುಂಟುಮಾಡುತ್ತದೆ, ಅದು ನಮ್ಮ ಜೀವನದ ಮೇಲೆ ನಾವು ನಿಯಂತ್ರಣವನ್ನು ಕಳೆದುಕೊಳ್ಳುತ್ತಿದ್ದೇವೆ ಎಂಬ ಭಾವನೆಯನ್ನು ಉಂಟುಮಾಡಬಹುದು
14	ನಾವು ಇತರರಿಗೆ ಉಂಟುಮಾಡುವ ನೋವು ಮತ್ತು ನೋವನ್ನು ನಾನು ನೋಡಿದಾಗ ನನಗೆ ಹತಾಶೆ ಉಂಟಾಗುತ್ತದೆ
15	ಶಾಲೆಯಲ್ಲಿ ಕಾರ್ಯನಿರ್ವಹಿಸುವ ಒತ್ತಡವು ವಿದ್ಯಾರ್ಥಿಗಳಿಗೆ ಬಹಳಷ್ಟು ದುಃಖ ಮತ್ತು ಚಿಂತೆಯನ್ನು ಉಂಟುಮಾಡಬಹುದು

Figure 2. Data Set and Collection

Sentences Dataset (Joy, Sadness, Anger, Fear)

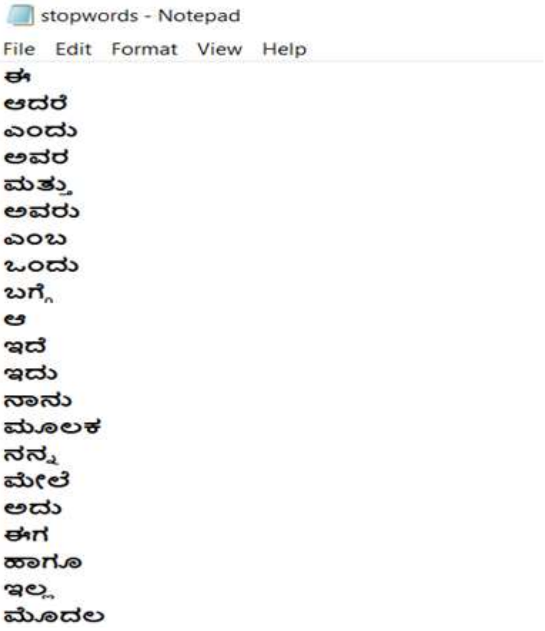
In your Kannada Summary Sentiment Analysis project, you've carefully curated a collection of texts that serves as the basis for your machine learning model's training. The sentences fall into four main categories. sentiments: Joy, Sadness, Anger, and Fear. This dataset captures the richness of human emotion and provides a wellrounded view of Kannadaspeaking users' sentiments. Let's break down how each sentiment category is structured in Figure 2

1. Joy: Sentences in this category express happiness, satisfaction, and positive feelings. Examples include *ನಾನು ಸಂತೋಷದಿಂದ ಇದ್ದೇನೆ!* (I'm feeling happy today!) and *ಈ ಹಬ್ಬದ ದಿನ ನನ್ನ ಜೀವನದ ಅತ್ಯುತ್ತಮ ದಿನ!* (This holiday is the best day of my life!). These sentences reflect excitement, contentment, and joyful experiences. Joy sentences usually contain words associated with happiness, positive events, or personal satisfaction.

2. Sadness: This category focuses on negative emotions, particularly those linked to loss, disappointment, or loneliness. Sentences like *ನಾನು ಇಂದು ಚೆನ್ನಾಗಿಲ್ಲ!* (I'm not feeling well today) and *ನನ್ನ ಸ್ನೇಹಿತರು ನನ್ನನ್ನು ಮರೆಯುತ್ತಾರೆ!* (My friends have forgotten me) showcase feelings of emotional distress. Sadnessrelated words often signify isolation, unfulfilled desires, or experiences of sorrow.

3. Anger: These sentences demonstrate irritation or frustration. Examples include *ನಾನು ಈ ಸಂದರ್ಭದಲ್ಲಿ ಕೋಪಗೊಂಡಿದ್ದೇನೆ!* (I'm angry in this situation) and *ನನ್ನ ಸ್ನೇಹಿತರು ನನ್ನನ್ನು ಮೋಸಮಾಡುತ್ತಾರೆ!* (My friends are cheating me). Anger sentences involve strong, assertive language that indicates dissatisfaction, betrayal, or aggression.

4. Fear: Sentences in this category express insecurity, threat, or danger, such as *ನಾನು ಈ ಸಂದರ್ಭದಲ್ಲಿ ಭಯಗೊಂಡಿದ್ದೇನೆ!* (I'm scared in this situation) and *ಈ ಪರಿಸ್ಥಿತಿಯು ನನ್ನ ಜೀವನವನ್ನು ಅಪಾಯಕ್ಕೆ ತಳ್ಳುತ್ತಿದೆ!* (This situation is putting my life in danger). Fearrelated sentences typically involve uncertainty, perceived threats, or fears of future events.



ಈ
ಆದರೆ
ಎಂದು
ಅವರ
ಮತ್ತು
ಅವರು
ಎಂಬ
ಒಂದು
ಬಗ್ಗೆ
ಆ
ಇದೆ
ಇದು
ನಾನು
ಮೂಲಕ
ನನ್ನ
ಮೇಲೆ
ಅದು
ಈಗ
ಹಾಗೂ
ಇಲ್ಲ
ಮೊದಲ

Figure 3. Data Set and Collection

The Role of Stopwords in Sentiment Analysis

1. Text Cleaning: The sentences must be cleaned up before being utilized as models.

cleaned and preprocessed. Removing stopwords is a critical step in this process. By eliminating words like *ಮತ್ತು* (and), *ಇಲ್ಲ* (no), and *ನಾವು* (we), the analysis can focus on the core message of the sentence.

2. Improving Model Accuracy: Including too many irrelevant words in your dataset can reduce the accuracy of your sentiment analysis model. For instance, words like *ನಾನು* (I) appear frequently across all sentiments and do not contribute meaningfully to determining whether a sentence expresses joy, sadness, anger, or fear. Removing these words helps the model identify the true sentimentbearing words in each sentence.

3. Reducing Dimensionality: By eliminating stopwords, the overall complexity of the dataset is reduced, leading to fewer features being passed to the model. This reduction in dimensionality improves the efficiency and performance of the machine learning algorithms being used.

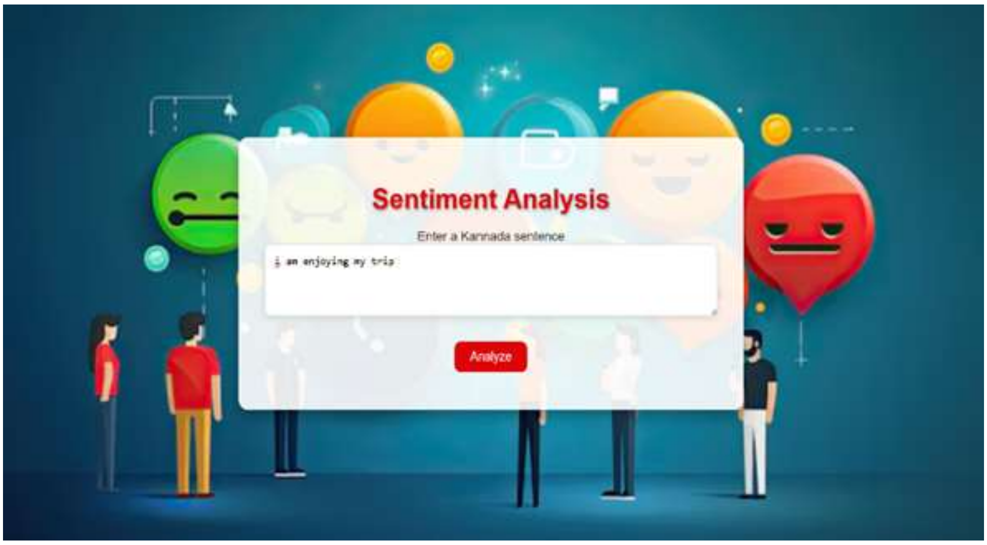
4. Results and Discussion

The translation feature detects the language of the input using the `translator.detect()` method. If the input is in English, it translates the text to Kannada using `translator.translate()`. Once translated, the processed Kannada text is fed into the sentiment analysis model for further analysis, and the final result is displayed.

In summary, the index page gathers input from users, while the result page displays the sentiment analysis results, including the translation, sentiment prediction, and model accuracy. The process is seamless, engaging the user with interactive, visually appealing feedback.

Home Page

The index page is the user interface where users input text (primarily Kannada sentences) for sentiment analysis. Its design focuses on usability, with a clear call to action (text area and submit button), and a visually appealing, modern layout. The purpose of this page is to collect user input and send it to the backend for processing **as shown in Figure 4**



Sentiment Analysis

Enter a Kannada sentence

i am enjoying my trip

Figure 4. Home Page

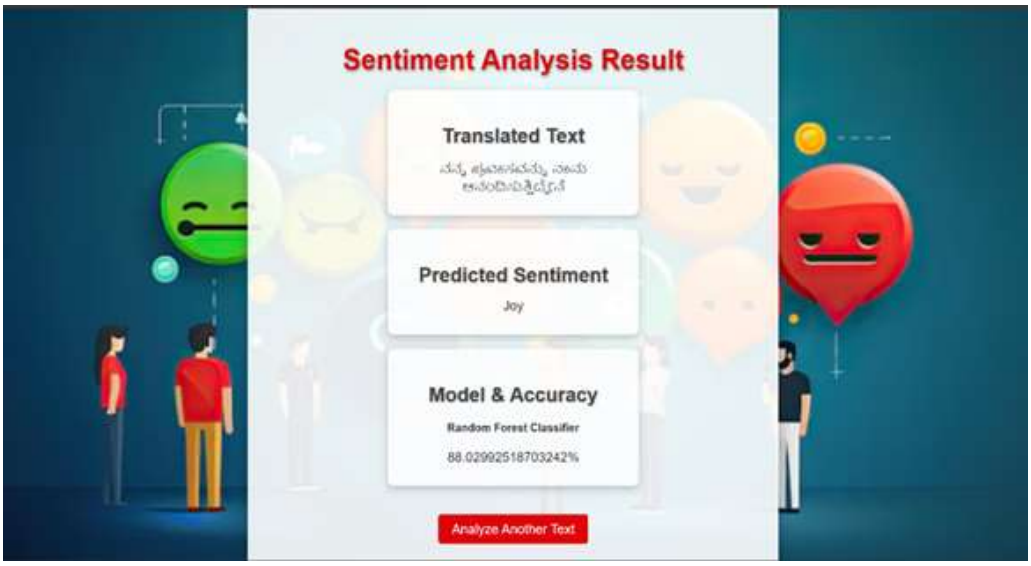
Result Page

The result page presents the analysis to the user. After processing the input, the result page displays:

Translated Text: If the input was in English, this section shows the Kannada translation, using the Google Translate API.

Predicted Sentiment: The sentiment classification result (happy, sad, angry, fear) is shown here.

Model Accuracy: This section displays the accuracy of the machine learning model used (Random Forest), which indicates the confidence of the sentiment analysis shown in figure 5



Sentiment Analysis Result

Translated Text

ನನ್ನ ಪ್ರವಾಸವನ್ನು ನಾನು ಆನಂದಿಸುತ್ತಿದ್ದೇನೆ

Predicted Sentiment

Joy

Model & Accuracy

Random Forest Classifier

88.02992518703242%

Figure 5. Result Page

How the Result Page Works

1. Displaying Results: The translated text, sentiment result, and model accuracy are retrieved from the backend and dynamically displayed in structured, styled boxes.

2. Interaction: Users can either analyze another text by returning to the index page or explore the displayed results.

3. Translation Feature: The translation process works by detecting whether the input is in English. If so, it translates the text into Kannada using the Google Translate API before the analysis.

Model Performance Classification Graph:

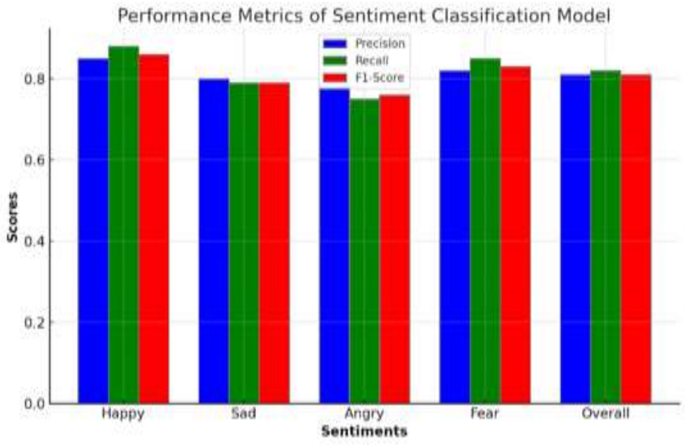


Figure 6. Model Performance Classification

Feature Importance: Visualize which words or features contribute most to the sentiment classification.

Performance Comparison: Graph comparing precision, recall, and F1 score for each sentiment (happy, sad, angry, fear).

By simplifying the technical explanations and focusing on key metrics and model comparisons, this version will be more concise and easier to understand for publication. You can include relevant graphs that illustrate the model's accuracy, precision, and performance breakdown across different sentiment classes.as shown in figure 6

Conclusion

In this project, we developed a Kannada Sentiment Analysis system using machine learning, specifically leveraging the Random Forest Classifier. The system effectively analyzes Kannada text and categorizes sentiments such as happy, sad, angry, and neutral. By preprocessing text data with cleaning and stemming techniques, and utilizing TFIDF vectorization, we transformed the textual data into meaningful numerical features for model training. The model's evaluation metrics indicate satisfactory accuracy in predicting sentiments, making it a useful tool for regional sentiment analysis.Furthermore, the integration of the machine learning model into a web application using Flask has provided a userfriendly interface where users can input Kannada sentences and receive instant sentiment predictions. This application can be expanded and applied in various domains such as social media monitoring, product review analysis, and customer feedback analysis, helping organizations and individuals gain deeper insights into public sentiment.

Future Work

- Deep Learning Models: Although the Random Forest Classifier yielded decent results, the use of deep learning models such as Long ShortTerm Memory (LSTM) or Bidirectional Encoder Representations from Transformers (BERT) could enhance the accuracy, especially for handling contextrich sentences and sarcasm.
- Larger and Diverse Datasets: Incorporating a more extensive and diverse dataset from different domains (e.g., news articles, blogs, social media) will improve the model's ability to generalize across various types of content.

Acknowledgement

I am grateful to Dr. Parashuram Bannigidad , Professor and Chairman, Department of Computer Science, Rani Channamma University, Belagavi for his valuable guidance for completion of this work

References

1. *International Journal of Recent Technology and Engineering (IJRTE)*. (2019). **Corpora-Based Classification to Perform Sentiment Analysis in Kannada**. Vol. 8, No. 4S2, pp. 418-422.
2. Deshpande, A., Patil, A., Jadhav, A., & Kulkarni, S. (2019). **Sentiment Analysis in Indian Languages Using Deep Learning**. *IEEE International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, IEEE, pp. 1-5.

3. *IEEE International Conference on Communication, Computing, and Internet of Things (ICCCIS)*. (2021). **Domain-Based Sentiment Analysis in Regional Language Kannada Using Machine Learning**. IEEE, pp. 123-128.

4. Gupta, A., Sinha, A., & Singh, A. (2020). **Sentiment Analysis for Code-Mixed Hindi-English Text**. *Conference on Data Science and Computational Intelligence (CoDS-COMAD)*, IEEE, pp. 18-24.

5. Jain, M., Rathore, S., & Gupta, K. (2019). **Deep Learning for Sentiment Analysis**. In: *Smart Innovations, Systems, and Technologies*, Springer, pp. 159-170.

6. *Advances in Intelligent Systems and Computing*. (2022). **Kannada Sentiment Analysis Using Vectorization and Machine Learning**. Springer, Vol. 1425, pp. 453-461.

7. Kiran, N., Reddy, K. R., & Naik, K. (2021). **Rule-Based Sentiment Classifier for Kannada**. In: *Communications in Computer and Information Science*, Springer, pp. 110-117.

8. Kumar, P., Mehta, M., & Joshi, R. (2020). **Multilingual Sentiment Analysis Using Transformers**. *IEEE International Conference on Semantic Computing (ICSC)*, IEEE, pp. 180-185.

9. Murthy, M., Desai, S., & Shetty, R. (2022). **Sentiment Classification in Kannada Social Media**. *IEEE International Conference on Communication, Computing, and Internet of Things (ICCCIS)*, IEEE, pp. 350-355.

10. Saini, R., Singh, A., & Mehra, P. (2020). **Bilingual Sentiment Analysis in Indian Languages**. *International Journal of Computer Applications*, Scopus Indexed