A Comparative Assessment of Machine Learning Techniques in Kannada Multi- Emotion Sentiment Analysis

Dakshayani Ijeri and Pushpa B. Patil
Department of Computer Science and Engineering
BLDEA's V. P. Dr. P. G. Halakatti College of Engineering and Technology
(Affiliated to Visvesvaraya Technological University, Belagavi-590018)
Vijayapura-586103, Karanataka, INDIA

Abstract

In order to advance a firm, it is crucial to understand user opinions on social media. India has a diversity with Kannada being one of the widely spoken languages. Sentiment analysis, in Kannada offers a tool to assess opinion gather customer feedback and identify social media trends among the Kannada speaking community. This kind of analysis assists businesses, in comprehending the sentiments expressed in Kannada language customer reviews, social media posts and online conversations. It empowers them to make choices based on data and customize their offerings to better suit the needs of their customers. This work proposes a model to perform sentiment analysis in Kannada language with four emotions namely anger, fear, joy, and sadness using machine learning algorithms like Linear Support Vector Classifier, Logistic Regression, Stochastic Gradient Descent, K-Nearest Neighbors, Multinomial Naive Bayes, and Random Forest Classifier. The model achieved the accuracy of 87.25% with Linear Support Vector Classifier.

Introduction

The use of technology has been increasing rapidly during the previous few years. Because of the quicker communication methods made possible by social media platforms like Twitter, Facebook, and WhatsApp, among others, digital technology has revolutionized how people live. Over 3.2 billion people use the internet regularly at this time. All industries, including e- commerce, movie ticketing, education, and others, have gone online as technology has developed. To advance a firm to new heights, it is crucial to understand user opinions on social media. Sentiment Analysis can be used to examine the opinions of these users.

India is one of the most linguistically diverse country in the world with 22 national languages. Only five percentage of Indian population can communicate effectively in English, while rest of the people are comfortable with their regional languages. Due to lack of resource availability, Sentiment analysis in Kannada language has not been explored extensively. The field of natural language processing encompasses various techniques, methods, and tactics that provide insights into how language influences our thought processes and impacts the results we obtain. One area of interest within this domain is sentiment analysis - a method that involves using natural language processing (Roy ,2023), text analysis, computational

linguistics, and biometrics to identify, extract, measure, and study emotions and subjective information.

Sentiment analysis (Chundi et.al, 2023)**,** a task in natural language processing and information extraction, aims to identify the emotions expressed by writers in reviews, inquiries, and requests, regardless of any emotions. Its objective is to determine the overall attitude of a speaker or writer towards a subject or the polarity of a document. This analysis considers various factors such as judgement, affective state, and purposeful emotional communication. Sentiment analysis has become increasingly vital due to the widespread

use of the internet and the extensive exchange of public opinion. In the present study, we focus on developing a system that classifies emotions (such as Anger, Fear, Joy, and Sadness) for textual sentiment analysis specifically in the Kannada language. The model accepts input in the form of Kannada text of any length and is tested using six algorithms: Linear Support Vector Classifier, Logistic Regression (Hasan et.al,2023), Stochastic Gradient Descent, K-Nearest Neighbors (Hasan et.al,2023), Multinomial Naive Bayes (Hasan et.al,2023), and Random Forest Classifier (Hegde et.al, 2022).

Several classification methods have been employed in this work, including Linear Support Vector Classifier with an accuracy of 87.25%, Stochastic Gradient Descent with 85.25%, Logistic Regression with 84.25%, Random Forest Classifier with 85.75%, Multinomial Naive Bayes with 85.50% and K-Nearest Neighbors with an 74.50%. The Linear Support Vector Classifier, which has an overall accuracy of 87.25%, outperforms all other algorithms, while K-Nearest Neighbors, which has an accuracy of 74.50%, has got the least performance among all.

1 Related Work

(Pushpa Patil et. al, 2024) proposed the comparison of various machine learning algorithms for sentiment analysis on Kannada language. The system considered positive and negative emotions and achieved highest accuracy of 75% with Multinomial Naïve Bayes Classifier.

(Nag et. al, 2023) a model for sentiment analysis was put forth. in ten Indian languages including Kannada one among them. This approach is carried out in three phases in which the first phase identifies the Unicode of the first letter of a sentence to identify the language and converts it into the English language. The second phase involves the training of a dataset and the third phase is used to categorize the text into different domains and predict the sentiment to positive or negative.

(Sanghvi et.al ,2023) proposed an approach for sentiment analysis in Kannada-English code-mixed language using Cross-lingual Language Model. This model considers the input of length 256 characters and predicts the sentiment into positive, negative or mixed. The positive sentence predictions are more accurate than other two classes. The model achieved the accuracy of 73%.

(Chundi et.al, 2023) proposed a model for emotion detection in Kannada-English code-mixed language using lexicon-based approach. This approach uses the list of words based on different emotions namely anger, joy and trust to train the model. The model is trained with 1882 comments from YouTube. The model achieved the accuracy of 87%.

(Roy ,2023) proposed an approach for sentiment analysis in Kannada code mixed language using BERT, RoBERTa and DistilBERT. This approach predicts the class for positive, negative and neutral emotions. The input text is transliterated into English language. The model achieved the best results for non-Kannada category. It achieved the F1-score of 0.66 for Kannada code-mixed language.

(Hasan et.al,2023) proposed a model for sentiment analysis in Bangla language on Russia Ukraine comments from social media using mBERT, XLM-RoBERT and BanglaBERT. The model analyzed 10,860 comments and classified them as neutral, favoring Ukraine (positive), or favoring Russia (negative). The model had an accuracy rate of 86%. The input text is limited to token lengths of 128 and 200, respectively, for which padding and truncation are used.The training dataset consists of imbalance samples for different categories.

(Saumya et.al, 2022) proposed a model for sentiment analysis in Kannada language and Homophobia detection in Tamil, English,

Malayalam languages. This approach used positive, negative and mixed emotions for sentiment classification. The sentiment analysis is experimented using stacking ensemble based on logistic Regression, K nearest neighbor classifier, Decision tree classifier, support vector machine and naïve Bayes classifier. The gradient ensemble and model ensemble are based on logistic regression, random forest classifier and support vector machine. The model achieved accuracy of 51.5%.

(Hegde et.al, 2022) proposed an approach for sentiment analysis and homophobic content detection in Kannada, Malayalam and Tamil languages using Deep Learning based Short Term Memory model. This approach considers positive, negative and neutral emotions for sentiment analysis. The model achieved F1 score of 0.16, 0.61 and 0.44 for Tamil, Malayalam and Kannada languages respectively.

(Sumana ,2022) developed a model for sentiment analysis in Hindi and Kannada languages for Twitter data. The model is experimented with Naïve Bayes Classifier, KNeighbors Classifier, Decision Tree classifier and Random Forest classifier. Two emotions namely positive and negative are considered for sentiment analysis. This approach achieved the accuracy of 99.7% in Hindi language and 99.5% in Kannada language using Random Forest algorithm.

(Shetty et.al ,2022) proposed an approach for sentiment analysis in English, Kannada and Hindi languages. This approach predicts the text into positive negative classes and is experimented with 2000 sentences of Kannada dataset. The model is trained using Convolutional Neural Network (CNN) and achieved the accuracy of 99%. The length of the text is restricted to 70 words per sentence, in case of lesser length padding is used whereas for larger length the truncation is applied.

(Chakravarthi et.al ,2022) proposed a model for offensive language identification in Tamil, Malayalam and Kannada languages with code-mixed text using the fusion of MPNet and Convolutional Neural Network (CNN) algorithms. The dataset of Kannada language consists 8 words per sentence. The dataset consists of 4695 sentences for training and 592 for testing. The model classifies the result as offensive sentence or not offensive sentence. This approach achieved accuracy of 76%.

(Fadil et.al ,2022) developed a model for sentiment analysis in Tamil, Malayalam and Kannada languages using Deep Neural Network. The sentiment analysis categorizes the text into positive, negative, neutral and mixed. The dataset consists of 6212 sentences. The model achieved the accuracy of 57%.

1.1 Drawbacks of Existing Work:

- In most of the work, only positive and negative classes are considered for sentiment analysis.
- The existing system performs sentiment analysis by limiting the number of words per sentence for which the model ignores the remaining words after the maximum limit. This may lead to lose the complete meaning of an input text and prediction of a sentence may be incorrect.
- In most of the systems the Kannada code-mixed language is considered, due to which there is no clarity of actual Kannada Language used for sentiment analysis.
- In some system the input text is either translated to English language or English transliterated form is considered for sentiment analysis, due to which analysis performance is not measured directly on Kannada language.

2 Methodology

The steps involved in Kannada language sentiment analysis are shown in Figure 1 The sentences are considered from manually prepared dataset. A custom input sentence is considered as input text for testing which is initially tokenized. Data cleaning is taking out unnecessary information from reviews that doesn't provide any value, like punctuation, commas, and other elements. Stop words in sentences include the terms that have no semantic value. The technique of stemming involves tracing a word back to its origin. The practice of classifying groupings of texts into distinct categories is known as classification or text tagging.

Kannada is a Dravidian language which is highly rich in morphological and lexical feature. Its agglutinative allows the language to add variations of suffixes which results in complexity of language such as

ಓದಿಸಿಬಿಟ್ಟೆ (ōdisibiṭṭe - I made [someone] read).

ಓದು (ōdu - read) + ಇಸಿ (isi - causative) + ಬಿಟ್ಟೆ (biṭṭe - completed action).

2.1 Dataset Creation

In the domain of machine learning, a dataset refers to a compiled collection of data that serves as the training material for the model. By utilizing the dataset as a reference, the machine learning algorithm acquires the ability to make predictions. In order for the algorithm to comprehend what the desired output is, the data is typically first labelled or annotated. Data set consists of 3000 sentences with average of 18 words in each sentence. Out of entire dataset 30% are joy sentences, 20% are anger, 20% are sad and 30% are fear sentences, these sentences are collected from one of the popular Kannada news websites “kannada.webdunia.com”. 80% of dataset is used for training and remaining for testing.

2.2 Tokenization

Tokenization is the process of breaking up raw text into manageable chunks. The original text is divided into tokens, which may be single words or full sentences. The understanding of the context or the creation of natural language processing (NLP) models both heavily rely on these tokens. Tokenization aids in comprehension by examining the word order within the text. Tokenization can be done with a variety of programs and frameworks. NLTK, Gensim, and Keras are a few well-liked libraries for this task.

2.3 Data Cleaning

Data cleaning is a critical phase in NLP. Without data cleansing, the dataset is similar to a list of words that the computer cannot comprehend. In this process, duplicate, incorrect, and peripheral data elements are found, and the undesired material is modified, replaced, or deleted. In natural language processing (NLP), data cleaning entails removing numerous punctuation symbols, such as the comma ' , ', colon ' : ', exclamation mark ' ! ', hyphen ' - ', question mark ' ? ', apostrophe ' " ', brackets' {} , [] , () ', semicolon ' ; ', ellipsis (***), and (...).

2.4 Stop Words Removal

Stop words are any words or phrases that add no meaning to a statement in any language. The real meaning of the statement will not change if these stop words are removed. The data size will drop as a result of eliminating these stop words, and the model's training time will also shorten while performance and accuracy increase. The NLTK library is one of the oldest and most widely used Python libraries for natural language processing. In the corpus module, NLTK aids in locating the list of stop words and promotes their removal. The text must be broken up into words in order to remove

stop words; if the word is found in the list of stop words provided by NLTK, it is removed. It gives you the option to add or remove stop words from the list of stop words already present in NLTK.

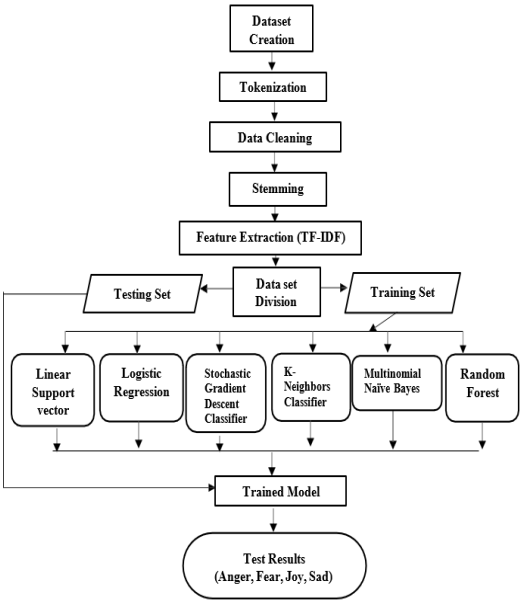

Figure 1. Methodology

2.5 Stemming

Natural language processing technique stemming breaks down words to their fundamental or root form in order to group together words that are variations of the same root word. The language is normalized by a stemming algorithm, as opposed to this, which simply reduces the variety of word forms to their standardized form. The words' base form is extracted using this technique by removing affixes. It is like cutting back the branches of a tree until they are at the trunk. For instance, the word "eat" is the root of the verbs "to eat," "to eat," and "to be eaten.". In order to index the words, search engines use stemming. The result is that a search engine can only store the word's stems rather than all of its variations. Stemming accomplishes this by reducing the size of the index and enhancing retrieval accuracy.

2.6 Feature Extraction

Term Frequency-Inverse Document Frequency is enforced to extract features for emotion detection which identifies the importance of a word in a document with respect to the collection of documents. TF-IDF is the text vectorization mechanism which assigns word's weight to a particular numeric value.

Term Frequency is the total number of appearances of a word as compared to the total number of words in a document as shown in equation 1.

$$TF = \frac{\text{Number of times the term appears in the document}}{\text{total number of terms in the document}} \quad (1)$$

Example: In the sentence ನಾನು ಪುಸ್ತಕ ಓದಿದೆ.

TF (ನಾನು)=1/3=0.33

TF (ಪುಸ್ತಕ)=1/3=0.33

TF (ಓದಿದೆ)=1/3=0.33

IDF is the frequency of usage of a word in all documents. The more frequency, lower is the score. It is calculated using the equation 2.

$$\text{IDF}=\log\left(\frac{\text{number of the documents in the corpus}}{\text{number of documents in the corpus that contain term}}\right) \quad (2)$$

Assume a corpus with multiple documents:
Total documents in the corpus: N=10.
Number of documents containing each term:

ನಾನು: Appears in 8 documents.

ಪುಸ್ತಕ: Appears in 5 documents.

ಓದಿದೆ: Appears in 6 documents.

ಸುಂದರವಾದ: Appears in 1 document.

IDF (ನಾನು)=log (10/8)=log1.25=0.0969

IDF (ಪುಸ್ತಕ)=log (10/5)=log2=0.3010

IDF (ಓದಿದೆ)=log (10/6)=log1.6667=0.2218

TF-IDF is calculated using following equation 3.

$$TF_{IDF} = TF * IDF \quad (3)$$

TF-IDF (ನಾನು)=TF (ನಾನು)×IDF

(ನಾನು)=0.33×0.0969=0.0320

TF-IDF (ಪುಸ್ತಕ)=TF (ಪುಸ್ತಕ)×IDF

(ಪುಸ್ತಕ)=0.33×0.3010=0.0993

TF- IDF (ಓದಿದೆ)=TF (ಓದಿದೆ)×IDF

(ಓದಿದೆ)=0.33×0.2218=0.0732

The final TF-IDF vector for the document is:
Vector= [0.0320,0.0993,0.0732]

The IDF score can be calculated using equation 4 with base 10 logarithm and denominator is added with 1 to avoid division by zero error.

$$IDF = \log\left(\frac{\text{number of the documents in the corpus}}{\text{number of documents in the corpus that contain term}+1}\right) \quad (4)$$

After the TF-IDF features are extracted, a sparse matrix is produced. Using this matrix, classification is carried out. The machine was trained using Kannada-language classification. This method trains the classifiers in the same language as the text. It is crucial that resources are available in the same language in order to analyze the sentiment. All training and exam materials are therefore in Kannada. We used a variety of classifiers, including Linear SVC, Logistic Regression, SGD Classifier, K-Neighbors Classifier, Multinomial NB, and Random Forest Classifier, to train and test the data.

2.7 Dataset Division

When the given data is split into two or more subsets so that a model can be trained, tested, and evaluated, data splitting enters the picture in data science or machine learning. Data splitting is a crucial component of practice or real-world projects, and it becomes essential when models are based on the data as it ensures the creation of machine learning models. It is divided into two splits; one will be used for training (80%) and the other for testing (20%).

2.8 Machine Learning Algorithms

As was already mentioned, preprocessing takes place before classification. The classification stage, an essential part of sentiment analysis, divides the dataset into four groups: joy, anger, fear, and sadness. To train the model this approach has employed following machine learning algorithms.

Linear Support Vector Classifier

The fundamental goal of linear SVC is to categorize the given data and provide the best fit hyperplane for the data.

Logistic Regression

It is a classification method that estimates the likelihood of the target variables through the use of supervised learning.

Stochastic Gradient Descent Classifier

The Stochastic Gradient Descent (SGD) Classifier is an efficient, scalable linear model for multiclass sentiment analysis that updates model weights incrementally using a subset of training data.

K-Neighbors Classifier

The K-Nearest Neighbors (KNN) algorithm for multiclass sentiment analysis classifies text by assigning the sentiment label most common among the k closest labeled examples in the feature space.

Multinomial Naïve Bayes

It predicts a set of texts to a particular class and is based on the Bayes theorem. Every tag will have its probability for the provided sample calculated, and the highest probability tag will be returned as the output.

Random Forest

The Random Forest algorithm is an efficient machine learning technique that brings together various decision trees to generate accurate predictions. The algorithm makes the final prediction by combining all of the individual trees' predictions during prediction.

3 Results and Discussions

3.1 Dataset Collection

The data is collected from the one of the Kannada news websites known as www.kannadawedunia.com for the model's training. It consists more than 2000 Kannada phrases from the internet for four classes—joy, fear, sad, and anger. These data are kept in XLSX (Microsoft Excel) format.

3.2 Classification Results

Figure 2 to Figure 5 illustrates a custom input example from popular news website of Kannada language, where users can manually enter sentences to be classified as one of the four emotions—joy, anger, fear, or sad. Additionally, it provides a phrase example that has been cleaned up and stemmed. The phrase will be categorized as joy, angry, fear, or sad by the categorization method.

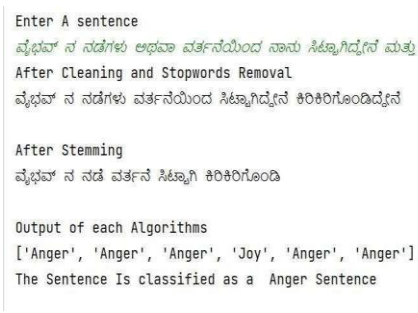

Figure 2: Final output of an anger sentence

The English version of Kannada input sentence from Figure 2 is "*Vaibhav's behavior had made me angry and irritated*" for which the model generated the result as ['Anger', 'Anger', 'Anger', 'Joy', 'Anger', 'Anger'] by ['Linear Support Vector Classifier', 'Logistic Regression', 'Stochastic Gradient Descent Classifier', 'Random Forest Classifier', 'Multinomial Naïve Bayes Classifier', 'K-Neighbors Classifier'] respectively.

The English version of input sentence from Figure 3 is "Fear of rejection can make it difficult to maintain confidence" for which the model generated the result as fear by all algorithms.

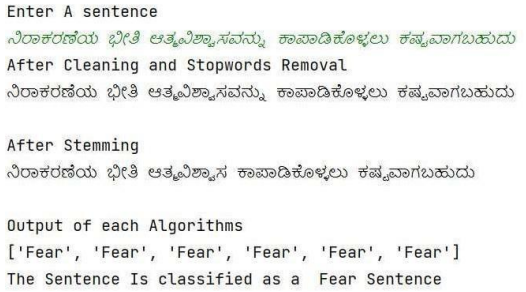

Figure 3: Final output of a Fear sentence

The English version of input sentence from Figure 4 is "*I appreciate the neighbors who always come forward to help and care others*." for which the model generated the result as joy by all algorithms.

The English version of input sentence from Figure 5 is "I feel a strong urge to shed tears when my co-actor's performance falls short on the stage." For which the model generated the result as ['Sad', 'Sad', 'Sad', 'Joy', 'Sad', 'Sad'] by ['Linear Support Vector Classifier', 'Logistic Regression', 'Stochastic Gradient Descent Classifier', 'Random Forest Classifier', 'Multinomial Naïve Bayes Classifier', 'K-Neighbors Classifier'] respectively.

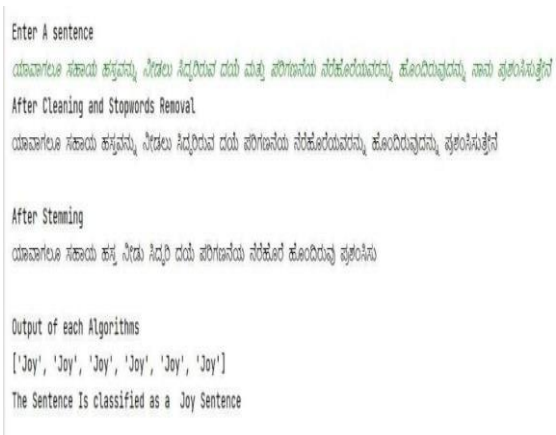

Figure 4: Final output of a Joy sentence

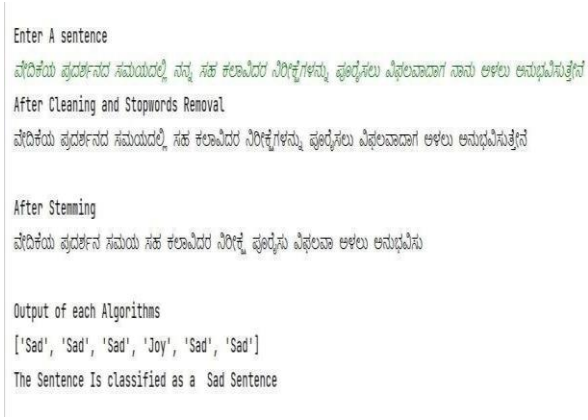

Figure 5: Final output of a Sad sentence

Table 1 shows the accuracy rates attained by various algorithms. It compares the accuracy of each algorithm. 2000+ Kannada phrases were used to train the model, with testing and training datasets representing two distinct categories. Of the entire dataset, the testing data accounted for 0.2. Based on the comparison, the Linear Support Vector Classifier performed better than all other classifiers, with an accuracy score of 87.25%. Logistic Regression had an accuracy of 84.25%, Stochastic Gradient Descent Classifier was 85.25% accurate, Stochastic Gradient Descent Classifier was 85.25% accurate, Random Forest Classifier was 84.75% accurate. K Neighbors Classifier performed poorly, as evidenced by its accuracy of 74.5%, which was the lowest.

Table 1: Accuracy of Algorithms

SL.NO	Classifier	Accuracy
1	**Linear Support Vector Classifier**	**87.25**
2	Stochastic Gradient Descent Classifier	85.25
3	Random Forest	84.75
4	Logistic Regression	84.25
5	Multinomial Naïve Bayes	82.5
6	K-Neighbors Classifier	74.5

Linear Support Vector Classifier had performed better for following reasons:

- Linear SVC uses a regularization parameter (C) that balances model complexity and classification accuracy on the training data. This helps avoid overfitting, especially on smaller datasets or datasets with noisy features.
- Linear SVC maximizes the margin between classes, meaning it finds the decision boundary that is farthest from the closest data points of each class (support vectors). This characteristic improves generalization to unseen data.
- TF-IDF vectors often have linearly separable patterns in sentiment analysis datasets

Table 2 shows the comparison between the proposed model and the existing model. Out of the existing 11 models, 5 were developed only in Kannada. Four models in Kannada code-mixed with other language (Sanghvi et.al ,2023), (Chundi et.al, 2023), (Roy ,2023) and (Fadil et.al ,2022). One in Bengali (Hasan et.al,2023), (Nag et. al, 2023) in English translated. In (Sanghvi et.al ,2023) the input data is restricted to 256 characters and the remaining text exceeding this limit is truncated due to which the meaning of a text may be disturbed. (Roy ,2023) experimented on transliterated English form due to which better performance is achieved for non-Kannada classification. (Hasan et.al,2023) and (Shetty et.al ,2022) also puts restriction on number of tokens for each sentence and truncation is applied because of which the actual sentiment analysis may be incorrect. (Saumya et.al, 2022) accuracy of this system is 51.5% for Kannada language. (Hegde et.al, 2022)The F1-score for Kannada language is 0.44. (Fadil et.al ,2022)The accuracy is 57% in Kannada language. Most of the work have two classification results as positive and negative. Our proposed model is experimented with 2000+ Kannada phrases from news websites with 6 different algorithms such as Logistic Regression, Stochastic Gradient Descent Classifier, Linear Support Vector Classifier, K Neighbors Classifiers, Multinominal Naïve Bayes and Random Forest

Classifier and attained the accuracy of 87.25% with Linear Support Classifier for the four emotions namely anger, joy, fear and sad.

Table 2: Comparison with Existing work

	Comparision
[2]	The language used in this work are multiple Indian languages translated in English with 4000 sentences. **Emotions:** Positive and Negative **Methods Adopted:** Unicode Identification.
[3]	Kannda-English code-mixed language with 4K sentences. **Emotions:** Positive, Negative and neutral. **Methods Adopted:** Transformer model with input sentence is restricted to 256 characters. **Accuracy:** 73%
[4]	Kannada- English code-mixed language with 7K sentences **Emotions:** Anger, Joy and trust. **Methods Adopted:** Naïve Bayes Algorithm **Accuracy:** 87%
[5]	Kannada code mixed language with 6K sentences. **Emotions:** Positive, Negative and neutral. **Methods Adopted:** CNN, BERT, ROBERT and DistillBERT. **Methods Adopted:** F1 score of 0.66 with Kannada code mixed language. Performed better for non-Kannada sentences
[6]	Bengali with 10K Youtube comments **Emotions:** Positive, negative and neutral. **Methods Adopted:** Maxnet, SVM, Decision Tree, KNN, SGDC and Random Forest. IT imposes restriction on number of tokens per comment. **Accuracy:** 86%
[7]	Kannada with 691 phrases. **Emotions:** Positive, negative and mixed. **Methods Adopted:** Logistic Regression, KNN, Decision Tree, SVM and Naïve Bayes classifier. **Accuracy:** 51.5%
[8]	Tamil Malayalam and Kannnda with 7K sentences. **Emotions:** Positive and Negative **Methods Adopted:** Text Vectorization and classifier. **Accuracy:** 0.441
[11]	Kannada 8K Sentences. **Emotions:** Offensive and non-offensive **Methods Adopted:** Fusion of MPnet and DeepNet, CNN and SVM F1 score 0f 0.76
[12]	Code mixed Tamil, Malayalam and Kannada language. With 6K sentences in Kannada. **Emotions:** Positive, negative and neutral. **Methods Adopted**: DNN. **Accuracy:** 57% in Kannada language.
Proposed Model	2000+ Kannada Phrases **Emotions:** Anger, Fear, Joy and Sad. **Methods Adopted**: Logistic Regression, Stochastic Gradient Descent Classifier, Linear Support Vector Classifier, K Neighbors Classifiers, Multinominal Naïve Bayes, and Random Forest Classifier. **Accuracy:** Linear Support Vector Classifier Has performed better with accuracy of 87.25%.

4 Conclusion

The use of social media for communication is widespread. Therefore, social media generates a lot of data each day. Sentiment analysis is therefore crucial in identifying company insights and achieving significant financial returns. For the English language, there are many sophisticated models for sentiment analysis. The work for Kannada language is very less. The proposed model had made an effort to provide a model that is effective for categorizing sentences in Kannada using several classification techniques. As part of this approach, 2000+ Kannada phrases were collected from Kannada news websites and manually labeled them as joy, angry, fear, or sad. Then these sentences are used to train our model. The preparation of data before classification is crucial. It improves model performance and

reduces the dataset to a higher level. Preprocessing techniques include tokenization, data cleaning, stop word removal, and stemming. Feature extraction comes after data has been preprocessed. For feature extraction, TF-IDF approach is used. The model is experimented with variety of classification techniques, including Linear SVC, Logistic Regression, SGD, K-Nearest Neighbors, Multinomial Naive Bayes, and Random Forest Classifier. The best performing algorithm overall, the Linear Support Vector Classifier, with an accuracy of 87.25%. Because the effectiveness of the model depends on the data, more data collection is still required.

References

Patil, P. B., Ijeri, D., Kulkarni, S. A., Burkaposh, S. S., Bhuyyar, R., & Gugawad, V. (2024). Comparative study of machine learning algorithms for Kannada twitter sentimental analysis. Multimedia Tools and Applications, 83(15), 45693-45713.

Nag, Shubhadip, et al. "A Knowledge based Approach for User Profiling from the Multilingual Texts in the Social Media Platforms." 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE). IEEE, 2023.

Sanghvi, Diya, et al. "Fine-Tuning of Multilingual Models for Sentiment Classification in Code-Mixed Indian Language Texts." International Conference on Distributed Computing and Intelligent Technology. Cham: Springer Nature Switzerland, 2023. Appendices

Chundi, Ramesh, Vishwanath R. Hulipalled, and Jay Bharthish Simha. "NBLex: emotion prediction in Kannada-English code-switch text using naïve bayes lexicon approach." International Journal of Electrical & Computer Engineering (2088-8708) 13.2 (2023).

Roy, Pradeep Kumar. "A Deep Ensemble Network for Sentiment Analysis in Bi-Lingual Low-Resource Languages." ACM Transactions on Asian and Low-Resource Language Information Processing (2023)

Hasan, Mahmud, et al. "Natural Language Processing and Sentiment Analysis on Bangla Social Media Comments on Russia–Ukraine War Using Transformers." Vietnam Journal of Computer Science (2023): 1-28.

Saumya, Sunil, Vanshita Jha, and Shankar Biradar. "Sentiment and Homophobia Detection on YouTube using Ensemble Machine Learning Techniques." Working Notes of FIRE 2022-Forum for Information Retrieval Evaluation (Hybrid). CEUR. 2022.

Hegde, Asha, and Hosahalli Lakshmaiah Shashirekha. "Leveraging Dynamic Meta Embedding for Sentiment Analysis and Detection of Homophobic/Transphobic Content in Code-mixed Dravidian Languages." (2022). Forum for Information Retrieval Evaluation, December 9-13, 2022, India.

Kanchan, Pradeep. "Hindi and Kannada Twitter Sentiment Analysis Using Machine Learning Algorithm." 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N). IEEE, 2022.

Shetty, Saritha, et al. "Sentiment Analysis of Twitter Posts in English, Kannada and Hindi languages." Recent Advances in Artificial Intelligence and Data Engineering: Select Proceedings of AIDE 2020. Springer Singapore, 2022.

Chakravarthi, Bharathi Raja, et al. "Offensive language identification in dravidian languages using MPNet and CNN." International Journal of Information Management Data Insights 3.1 (2023): 100151.

SFadila, N. Muhammad, and S. K. Lavanya. "Sentiment Analysis of YouTube comments in Dravidian Code-Mixed Language using Deep Neural Network." Forum for Information Retrieval Evaluation, Kolkata, India, December 9-13, 2022.