International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Exploring Sentiment Analysis in Kannada Language: A Comprehensive Study on COVID-19 Data using Machine Learning and Ensemble Algorithms

Shankar R.*[1], Suma Swamy[2], Shashiroop Hegde[3]

Submitted: 03/11/2023 **Revised**: 22/12/2023 **Accepted**: 04/01/2024

Abstract: COVID-19 changed several lives in the past few years. The world came to a halt and the situation forced everyone to stay home and this brought a huge shift in people's sentiments for several reasons. Various countries were severely affected due to the lockdown including various regions of India. India is a multi-lingual country which consists of various regional languages. In this paper, Sentiment Analysis is performed on COVID-19 data which is in Kannada using three Machine Learning algorithms along with three Ensemble algorithms. Employing a diverse set of models, the study aims to enhance prediction accuracy, feature selection, and overall understanding of the pandemic's dynamics. By leveraging advanced techniques, the research contributes valuable insights for optimizing decision-making processes in healthcare and public policy. The findings demonstrate the potential of integrating machine learning into epidemiological studies for a more nuanced and effective response to global health crises.

Keywords: *Sentiment Analysis; Machine Learning; Ensemble; Scikit Learn; COVID-19.*

1. Introduction

Sentiment is a feeling or an opinion, especially one based on emotions. A human's decision can sometimes be entirely based on raw emotions. Hence, studying about emotions and sentiments embedded in a statement can be very precious. Sentiment Analysis is one of the techniques in which human sentiment can be studied. Sentiment Analysis is a technique under Natural Language Processing (NLP) where we study and quantify information based on a particular subject into either positive, negative, or neutral. There are 4 types of sentiment analysis: Fine-Grained, Emotion Detection, Aspect Based, Intent Analysis. Each of these types does the same action but in different ways depending on the purpose of use. Apart from their types, they can be applied on different levels such as Word level, Sentence level and Document level. Sentiment Analysis is applied in various fields such as regulating comments on social media website, product reviews, Business intelligence and Brand reputation management and so on.

The sentiment of a sentence is typically based on a word or a phrase revolving around the main subject in that sentence. These words or phrases are important as they decide the tone of the sentence and also reflect the individual's opinions on that subject. Capturing these words and phrases to understand the underlying tone and opinion behind it is what Sentiment Analysis is all about.

There has been numerous research on sentiment analysis in different domains and majority of them are in English. India is a diverse, multilinguistic nation with several languages such as Kannada, Hindi, Tamil etc. and these languages span over 28 states with each language being local to their respective region. Each language has its vocabulary and way of expressing emotions. Hence, sentiment analysis in these different languages becomes necessary. The scarcity of sentiment analysis in these languages becomes a challenge as well as the driving force to perform deeper research to gain more knowledge.

In this paper, sentiment analysis is applied on the regional language Kannada. Another highlighting feature is the use of a COVID-19 related dataset. COVID-19 has vastly affected the world in the last few years to a point where every aspect of the human life was completely changed. A pandemic which negatively affected us humans is expected to contain a lot of negative data, so performing sentiment analysis on such data is another challenge that needs to be overcome. With the use of several machine learning and Ensembling algorithms, we try to understand the emotions of people who communicated in Kannada during the pandemic.

2. Related Work

Sentiment Analysis using semantic and machine learning approaches is performed which is predominately used on English data set, is applied on Kannada web documents.

[1] *1Research Scholar, Sir M. Visvesvaraya Institute of Technology, Visvesvaraya Technological University, Belagavi, India*
ORCID ID : 0000-0001-6389-2198
[2] *2Professor, Sir M. Visvesvaraya Institute of Technology, Visvesvaraya Technological University, Belagavi, India*
ORCID ID : 0000-0002-6207-5898
[3] *3Associate-II Software Engineer, Capgemini Technology Services, India*
** Corresponding Author Email: shankar@bmsit.in*

Finally, the average accuracy of both is compared and the results show machine learning approach to be better [1]. In a comparison between direct Kannada dataset and machine translation in English, where opinion extraction and decision tree classifier were used respectively, the results obtained for analysing data in regional language gave better and accurate results compared to machine translated English data [2]. Similarly, lexicon-based approach was compared against Naïve Bayes classifier and was found to give better accuracy, but better results were obtained using Naïve Bayes for incomplete or uncertain data. Considering these outcomes, when a hybrid approach of analysis which contained both Lexicon-based and Naïve Bayes approach was implemented, a more accurate outcome was obtained [3]. When the Kannada Code Mixed Dataset [KanCMD], a multi-task learning dataset for sentiment analysis and offensive language identification containing comments from YouTube users, was subjected to a variety of machine learning algorithms, including SVM, Random Forest, Logistic Regression, Multinomial Nave Bayes, and K-Nearest Neighbour, it performed well in recognising emotions but struggled to identify offensive language [4].

A Dual-Channel BERT-based model that uses best of two datasets: Code-mixed and Translated English texts produce various results based on the combination used for the respective datasets [5]. The ability to do sentiment analysis was greatly developed for several languages, including Hindi, Bengali, Tamil, and Telugu. Hindi language scored an accuracy of 81.97% using Lexicon Based Approach and an overall accuracy of 87.1% using Naïve Bayes Classifier. Whereas other under-resourced languages pose a concern and allow opportunities to develop quality resources for these languages [6]. Convolutional neural networks (CNN) and bidirectional long short-term memory (BiLSTM), two deep learning models, are used in a hybrid model called SAEKCS for sentiment analysis of Kannada-English code switch text. This approach produces an accuracy of 77.6% [7].

The Bayesian Recognizer was shown to have a greater error rate than the Hidden Markov Model-based recognizer. From the Emotion Error Rate (EER) of Hidden Markov Model, it was clear that it was able to recognize emotions better and was consistent compared to the Bayesian Recognizer [8]. Using Turney's Pattern Algorithm on Kannada dataset provides an accuracy of 51%. When used with negators, we observe an increase in accuracy to 53%. On creating new patterns and applying them to document-based approach, there is an increase in accuracy by 33% and 31% [9]. Two machine learning algorithms, namely, Random Forest and SVM are used. Random Forest seems to be outperformed by SVM in the case of accuracy and other parameters of measurement seem to be almost similar. Random Forest produces an accuracy of 86.73% and SVM produces an accuracy of 89.08% [10]. The KES database, which has two male and two female native actors delivering the four main emotions of happiness, grief, rage, fear, and neutral, was subjected to spectral analysis. Fast Fourier Transform (FFT) and linear prediction (LPC) methods were employed in the investigation [11].

An artificial neural network (ANN)-based emotion identification system and contrast it with a hidden markov modelling (HMM)-based system is developed. Consequently, it may be concluded that each emotion of Kannada speaker, that results from Kannada vowels or consonants can be successfully modelled using ANN, with 10 hidden layers per model [12]. Using characteristics like Character N-Grams, Word N-Grams, Repetitive characters, and others on SVM and LSTM on our corpus, experimentation was done with machine learning prediction models, and the accuracy was 30 and 32 percent, respectively [13]. To assess the impact of the coronavirus on daily life, the mood of tweets concerning work from home (WFH) and online learning was tracked over time. On the dataset of tweets containing coronaviruses that was labelled using VADER, LSTM and ANN both achieved accuracy of 76 and 84.5 percent, respectively. Additionally, it was noted that overall attitudes have continually favoured online learning and work from home, as opposed to the opposite [14].

During the early months of the COVID-19 outbreak in Europe, Twitter messages (or tweets) were gathered. A total of 4.6 million tweets were examined, and of them, 79,000 included at least one COVID-19 term in them. It is observed that different countries, aggregated and represented through curves on a graph of sentiment over time, have a relatively similar development with evident drops after the announcement of lockdown and increases during holidays [15]. Using an ensemble model that merges deep learning based CNN and transformer-based BERT, sentiment prediction is performed on bi-lingual low resource languages such as Kannada and Malayalam. Ensembling of the transformed based models produced an accuracy of 87.5% and precision, recall, f-1 score of 0.66 each in Kannada and an accuracy of 86.3% and precision, recall, f-1 score of 0.52, 0.54 and 0.53 respectively in Malayalam [16]. A hybrid text and aspect grading model for sentiment summarization is studied. In general, the paper offers studies on opinion mining and sentiment analysis, suggesting practical methods for examining consumer attitudes toward products and services [17]. Building a Malayalam dataset for sentiment analysis on Malayalam texts is the main goal of this research, which also examines how well a pre-trained deep learning model performs sentiment analysis on latent Malayalam texts. BERT outperformed all other ML and DL models with an accuracy of 88.06%, followed by Bi-GRU with 83% [18].

A data collection of Twitter posts connected to the current

14th Gujarat Legislative Assembly Election, 2017, to predict the likelihood of winning party by employing popular opinion. Furthermore, it broadens the research by applying a latent Dirichlet allocation (LDA)-based topic model to determine the correlated topic for the event across several categories [19]. The feasibility of a novel fine-grained sentiment annotation system is investigated. The results of fine-grained sentiment analysis tests were reviewed, revealing that the proposed approach attained an F1-score of 0.71 for identifying the target of the sentiment expression and an F1-score of 0.68 for identifying the polarity of the sentiment expression [20]. This research examines sentiment analysis on Twitter corpora using several methodologies, including monolingual and multilingual models. The authors also discovered that contextual features perform poorly due to the short amount of the training corpus but become effective as the training data grows larger [21].

The use of machine translation systems for multilingual sentiment analysis is investigated in this research article. Overall, this work offers useful insights on the use of machine translation systems for multilingual sentiment analysis, emphasizing the significance of taking language-specific aspects into account in this sort of analysis [22]. The translation of English text sentiment analysis tools and methodologies to Spanish is examined. They discover that SVMs, at least the rather simple SVMs they have tried, perform poorly in their Spanish corpora. [23]. This research article focuses on the problems and achievements in applying concept-level techniques to analyze internet attitudes and feelings. Overall, this article provides a thorough examination of the obstacles and potential in concept-level sentiment analysis [24]. This paper describes research that used deep learning approaches to analyze sentiment in code-mixed Bambara-French social media material. The highest performing model outperformed the traditional machine learning techniques with an accuracy of 83.23% [25]. By concentrating on emotion tokens, this research provides a unique way to assessing sentiment in short and informal tweets. The suggested technique obtains an F1-score of 0.68 for English tweets and 0.62 for non-English tweets, which is much higher than previous approaches' F1-scores [26].

This research provides a language-independent Bayesian model for sentiment analysis of brief social-network status updates, with a focus on Twitter. The suggested model and extension have significant implications for corporations and organizations interested in monitoring sentiment on social media platforms such as Twitter [27]. The SemEval-2013 Task 2 competition promoted research that will contribute to a better understanding of how sentiment is transmitted in Tweets and SMS messages. The job was divided into two subtasks: expression-level and message-level. The top team received an F1 score of 0.692 on subtask A and 0.676 on subtask B. The top-scoring team received strong F1 scores in both subtasks, showing that sentiment analysis in Twitter and SMS communications is a difficult but doable topic [28]. This paper gives a complete investigation on sentiment recognition in Malayalam tweets using deep learning and traditional approaches. LSTM with ReLU activation function had the greatest accuracy of 87.5% among deep learning techniques, while CNN with ELU activation function achieved the highest F1-score of 0.87 [29]. The outcomes of the TASS 2014 workshop on the problem of aspect-based sentiment analysis are presented. Overall, the workshop indicated that the Spanish sentiment analysis research community is progressing toward the development of appropriate approaches for aspect-based sentiment analysis [30].

3. Methodology

The entire process flow can be presented as below.

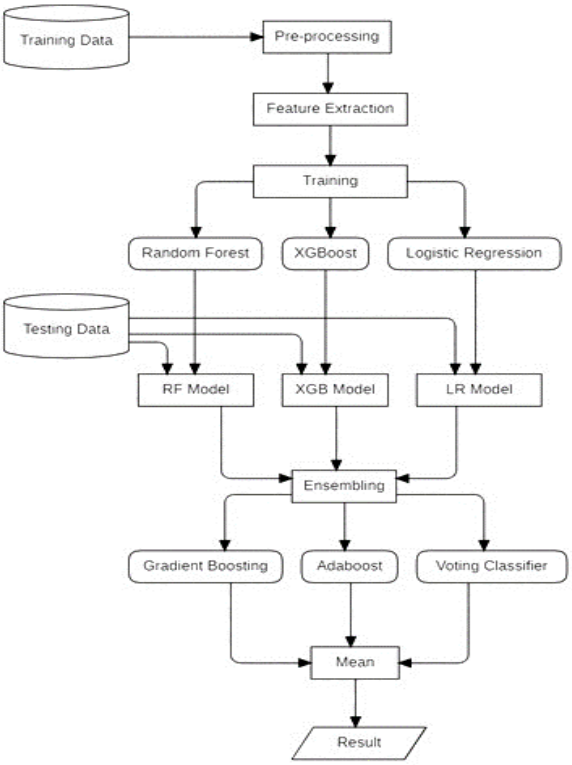

Fig 1. Process Flow

The above process can be divided into several steps as follows:

3.1. Data Collection

The dataset used was found/collected from Kaggle from Shravan C and contained 10,000 sentences. Each sentence in this dataset is specifically directed towards COVID-19 and its effect. Python is used to perform all our sentiment analysis and natural language processing.

3.2. Data Visualization

For a deeper understanding of the dataset being used, a visual study of the same was performed. To start with, the dataset was displayed to understand its contents. On a high level, we observed that the dataset included the 'Text' or the sentences that were in Kannada based on COVID-19 and its corresponding 'Label'. 'Label' here refers to the sentiment a particular sentence is insinuating.

The following snippet shows the same:

	Text	Label
0	ಇತ್ತೀಚೆಗೆ ಮಧ್ಯ ಚೀನಾ ಪ್ರವಾಸದಿಂದ ಮರಳಿದ ಯುನೈಟೆಡ್ ...	0.0
1	ಕರೋನವೈರಸ್ 2019 ಎನ್ ಕೋವಿ ಕಾದಂಬರಿಯು 5 ವರ್ಷದ ಚೀನೀ...	0.0
2	ಯು.ಎಸ್. ಸೆಂಟರ್ಸ್ ಫಾರ್ ಡಿಸೀಸ್ ಕಂಟ್ರೋಲ್ ಅಂಡ್ ಪ್ರ...	0.0
3	ಚಿತ್ರ ಕೃತಿಸ್ವಾಮ್ಯ ಗೆಟ್ಟಿ ಇಮೇಜಸ್ ಚಿತ್ರ ಶೀರ್ಷಿಕೆ...	0.0
4	ಬೋಸ್ಟನ್ ಸಿಬಿಎಸ್ ಸಿಎನ್ಎನ್ ಡಿಸೆಂಬರ್ನಲ್ಲಿ ವುಹಾನ್ ...	0.0

Fig 2. Dataset Understanding

To get a hold of the count of different labels, we plotted a bar graph as follows:

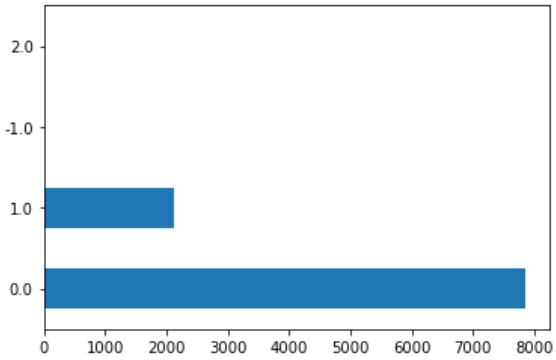

Fig 3. Count of Positive and Negative labels in dataset

As we can see in the above graph, there are 4 labels: 0.0, 1.0, -1.0, -2.0. On further analysis it was determined that the sentences labelled 0.0 were 'Negative' and the ones labelled 1.0 were 'Positive'. The labels -1.0 and -2.0 were negligible in count and did not classify as either Positive or Negative. Next, we look at the distribution of words based on their length. We use a distribution plot to visualize the density of words against the length of statements.

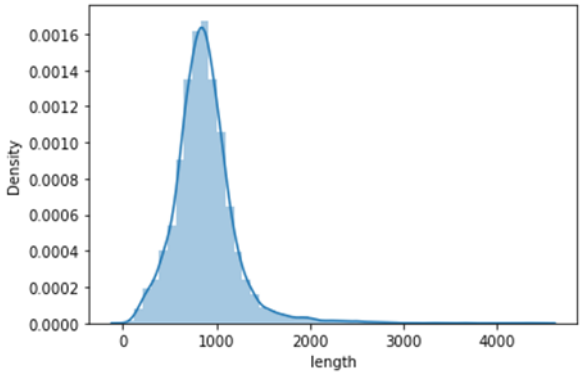

Fig 4. Distribution plot to visualize the density of words against the length of statements

We also use a boxplot to understand the variability of the values or 'Labels' against the length of statements. It also helps us identify any outliers.

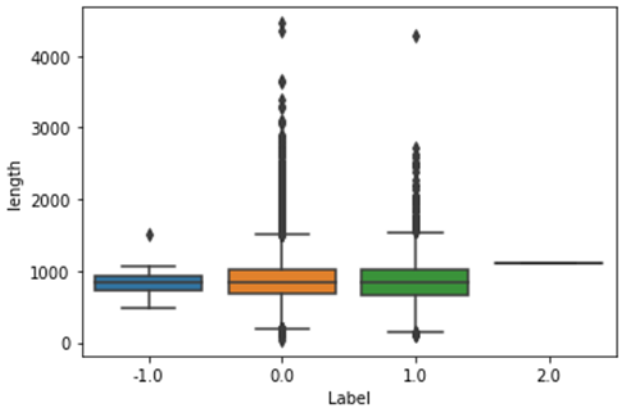

Fig 5. Boxplot to understand the variability of the values or 'Labels' against the length of statements

3.3. Data Pre-Processing

The data collected for the purpose of the experiment was raw. So necessary pre-processing steps were taken. The sentences in the dataset were manually labelled to identify their polarity to train a model in future. In this experiment, the polarity of data was divided into two classes, Positive and Negative. Any redundant or repetitive data was eliminated. After removing all the redundant and repetitive data, the dataset was imported into a Jupyter Notebook to compare it with the two classification classes to check if any of the classes were skewed. In this case, the quantity of negative data massively outnumbered the positive data. An attempt was made to balance out the data using sampling techniques. Techniques such as Synthetic Minority Oversampling Technique (SMOTE), Borderline SMOTE and Oversample using Adaptive Synthetic (ADASYN) were experimented on the data. Unfortunately, these techniques could not provide fruitful results. Hence, it was necessary to balance the data before proceeding and this was done by under sampling the majority class.

3.4. Feature Extraction

Before feature extraction is performed on the dataset, we split the data into training and testing data. Here, we use an 80:20 split for training and test data respectively. To perform the split, we use the train_test_split function from Sklearn library. Using the TF-IDF Vectorizer, feature extraction is performed where it converts the Kannada data into a usable vector. It produces a sparse matrix of tf-idf features from the provided data. With the Pickle library, we dump the tf-idf vectorizer for future use. To preserve the configuration of the vectorizer during both training and testing, we dump the tf-idf vectorizer for future purposes.

3.5. Training the Model using ML Algorithms

We use several different algorithms on the dataset to train different models. Some of the algorithms used by us

include: XGBoost [XGB], Logistic Regression [LR], Random Forest [RF].

3.5.1. XGBoost

Extreme gradient boosting, or XGBoost, is a gradient boosting technique that was created to be very effective, adaptable, and efficient. Using this method, decision trees are created successively, with the results of each classifier being eventually combined to create a robust and accurate model.

3.5.2. Logistic Regression

A categorical dependent variable is predicted using the supervised learning method logistic regression. It is based on a few independent factors. It forecasts the output into a certain category using a logistic function.

3.5.3. Random Forest

Random Forest mixes numerous decision trees, whereas the decision tree produces results based on the combination of a few decisions. Large datasets may be processed effectively while keeping accuracy.

3.6. Ensembling Approach

Ensembling approaches involve combining several independent models or algorithms by voting or averaging that produces a higher performance compared to all the individual models. Here, we use several Ensembling models on our dataset. They are mentioned as follows:

3.6.1. Adaboost

AdaBoost is an iterative approach to improve subpar classifiers by gaining knowledge from their errors. Although a single classifier might not be able to accurately predict an object's class, we can create a strong model by merging several weak classifiers, each of which learns from the mistakenly classified items of the others.

3.6.2. Gradient Boosting

The consistency and speed of gradient boosting make it a standout method, especially when working with large and complex datasets. Gradient-boosted trees, the resulting technique, outperforms random forest when a decision tree is the weak learner. The development of a gradient-boosted trees model follows the same stage-wise process as earlier boosting techniques, but it generalises those techniques by allowing optimization of any differentiable loss function.

3.6.3. Voting Classifier

A voting classifier is a machine learning model that receives training on a collection of different models and predicts an output (class) based on the likelihood that the outcome would fall into the selected class. It simply compiles the output from each classifier that is input into the voting classifier and forecasts the class that will receive the most votes. Instead of building several specialised models separately and evaluating their performance, we develop a single model that trains on numerous models and predicts output based on the total number of votes cast for each output class. Also, we make use of the GridSearchCV from the Scikit Learn Library for selecting the best parameters possible. GridSearchCV is a method for finding the optimal parameter values from a given set of parameters in a grid. In other words, it can be seen a type of cross-validator that selects only the best parameters out of a set of parameters. It is used to fine tune the parameters of a model to obtain the best possible result. We use it on the ensembling approaches to add an extra layer of enhancement on the already ensembled results.

3.7. Results and Discussions

Perform sentiment analysis on a dataset consisting of 10,000 reviews in total for two emotions: Positive and Negative. This dataset has reviews that are mainly focused on COVID-19. It was collected from Kaggle by Shravan C. The results are divided into two sections, results of machine learning algorithms and Ensembling methods.

3.7.1. Machine Learning Results

For the machine learning algorithms, namely XGBoost, Random Forest and Logistic Regression, respective precision, recall and F1-score have been listed in the tables. The average precision of all algorithms is 64.67% (or 0.6467). The finely tuned libraries of Sklearn produce the best possible results due to their high efficiency.

3.7.2. Ensembling Results

Next, we perform Ensembling on the dataset using the following algorithms: Adaboost, Gradient Boosting and Voting Classifier. We observe an average precision of 0.67, which is an improvement compared to the previous approaches. We make use of GridSearchCV to pick the only the best hyper-parameters for Ensembling to ensure better results.

A Random Forest Classifier is well-known for its efficiency on large datasets. Its ability to work well with non-linear data along with lower chances of overfitting makes it a suitable option for this dataset. When applied on the dataset, it produces a precision of 0.68 or 68% for the negative class and 0.65 or 65% for the positive class. Although Random Forest is an ensembling approach at its core, we use it as a supervised learning technique for our dataset as we perform Ensembling explicitly later on.

Table 1. Performance Matrix of Random Forest Algorithm

RANDOM FOREST				
Class	***Precision***	***Recall***	***F1-Score***	***Support***
Negative	0.68	0.67	0.67	304
Positive	0.65	0.65	0.65	329
Accuracy			0.66	633
M-avg	0.66	0.66	0.66	633
W-avg	0.66	0.66	0.66	633

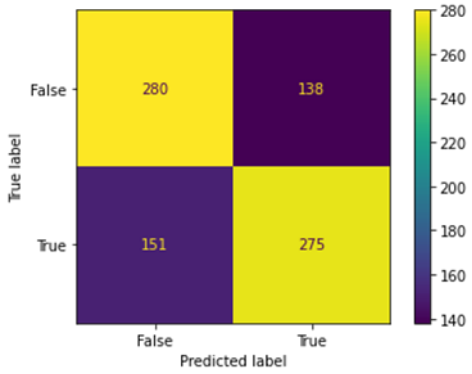

Fig 6. Confusion Matrix of Random Forest Algorithm

We consider only two sentiments or classes, Positive and Negative. Hence, Logistic Regression is a good match for this dataset. Based on earlier dataset observations, it can accurately forecast binary outcomes. In our case, Logistic Regression can produce a precision of 0.69 or 69% for Negative class and 0.65 or 65% for positive class.

Table 2. Performance Matrix of Logistic Regression Algorithm

LOGISTIC REGRESSION				
Class	***Precision***	***Recall***	***F1-Score***	***Support***
Negative	0.69	0.66	0.67	304
Positive	0.65	0.68	0.66	329
Accuracy			0.67	633
M-avg	0.67	0.67	0.67	633
W-avg	0.67	0.67	0.67	633

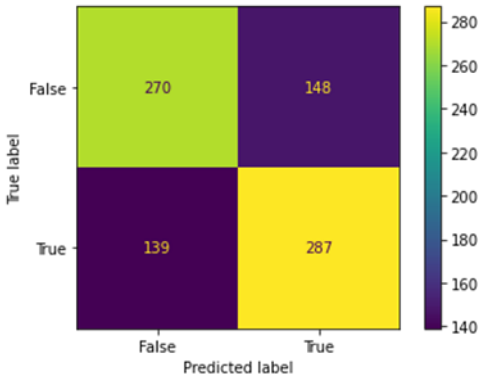

Fig 7. Confusion Matrix of Logistic Regression Algorithm

XGBoost is an enhanced and regularized version of Gradient Boosting. It is well-known for its parallel tree boosting capabilities and can handle any amount of data. In regression and classification problems, it likewise reigns supreme. Here, we observe a precision of 0.68 or 68% for Negative and 0.64 or 64% for Positive class.

Table 3. Performance Matrix of XGBoost Algorithm

XGBOOST				
Class	***Precision***	***Recall***	***F1-Score***	***Support***
Negative	0.68	0.65	0.67	304
Positive	0.64	0.66	0.65	329
Accuracy			0.66	633
M-avg	0.66	0.66	0.66	633
W-avg	0.66	0.66	0.66	633

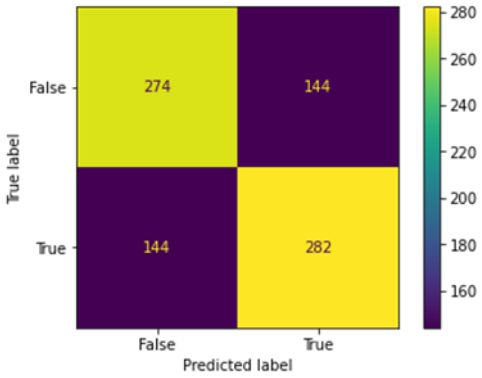

Fig 8. Confusion Matrix of XGBoost Algorithm

When the three algorithms are compared with each other, we find that the overall accuracy all three algorithms, XGBoost (XGB), Random Forest (RF) and Logistic Regression (LR) is similar.

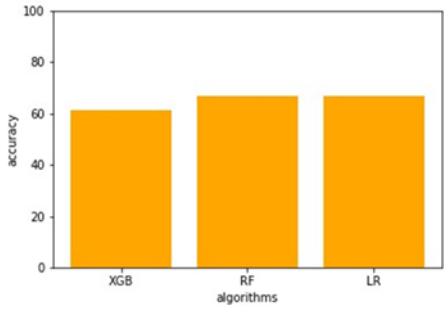

Fig 9. Comparison Study

As illustrated in the graph, all three algorithms produce an accuracy in the range 66% - 67%. Also, when we check the Area Under the Curve (AUC) of the individual models against a No-Skill classifier, we observe that there is a gradual, linear rise until a point followed by a gradual linear fall.

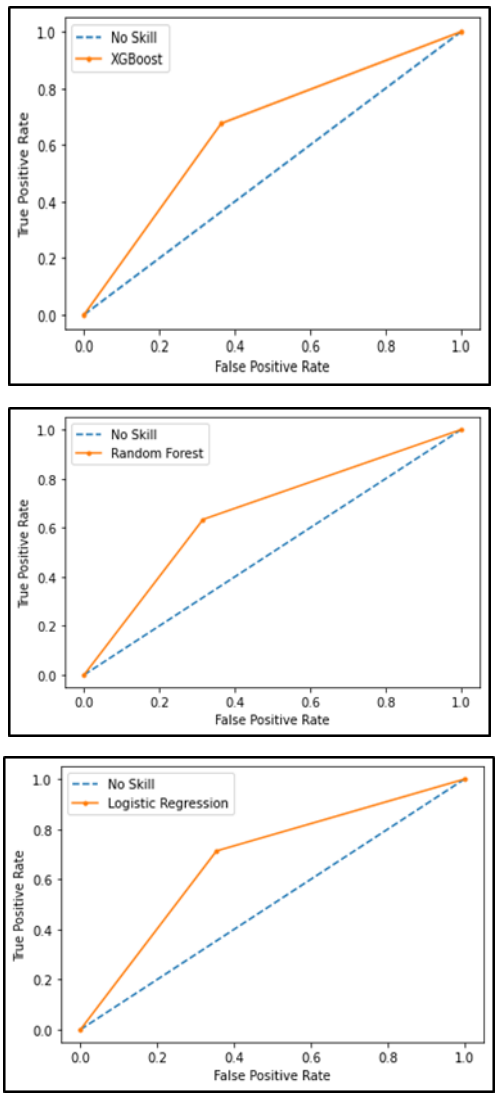

Fig 10. Area Under the Curve (AUC) of the individual models against a No-Skill classifier

This gives us an insight on the capability of the models to separate the classes. Here, we observe that the models can somewhat distinguish between the classes they are predicting, but not as effectively as a model with a steeper rise and fall in its AUC. Hence, we try to perform Ensembling on the models to get better results. The AUC graphs of each of the model is as shown above.

Further, we consider enhancing these results by applying Ensembling approaches. The following is observed and noted. According to Adaboost's roots, it was intended to improve the efficiency of binary classifiers. Adaboost is also known for having a lower likelihood of overfitting.

Table 4. Performance Matrix of Adaboost Algorithm

ADABOOST				
Class	***Precision***	***Recall***	***F1-Score***	***Support***
Negative	0.62	0.54	0.58	304
Positive	0.62	0.69	0.65	329
Accuracy			0.62	633
M-avg	0.62	0.61	0.61	633
W-avg	0.62	0.62	0.62	633

Gradient boosting has frequently been shown to be one of the most effective techniques for developing predictive models in both classification and regression. Gradient Boosting is commonly used in bias-prone problems, and in this dataset, it appears to be a good alternative because the background of the comments is overwhelmingly unfavourable. When it comes to flexibility, Gradient Boosting is more flexible as is a generic algorithm unlike Adaboost. Here, we can see a considerable increase in the precision and recall scores compared to the previously used algorithms. It produces a precision of 0.61 or 61% for Negative and 0.60 or 60% for the Positive class.

Table 5. Performance Matrix of Gradient Boosting Algorithm

GRADIENT BOOST				
Class	***Precision***	***Recall***	***F1-Score***	***Support***
Negative	0.61	0.48	0.54	304
Positive	0.60	0.71	0.65	329
Accuracy			0.60	633
M-avg	0.60	0.60	0.60	633
W-avg	0.60	0.60	0.60	633

A voting classifier is a machine learning model that gains experience by training on a collection of several models and forecasts an output (class) based on the class with the highest likelihood of being the output. It supports two types of voting namely, Hard Voting and Soft Voting. Hard Voting considers the majority of votes out of a set of classifiers. Soft Voting produces a prediction based on the average of all the considered classifiers. In our scenario, we

use soft voting for the classifiers.

Table 6. Performance Matrix of Voting Classifier Algorithm

VOTING CLASSIFIER - m				
Class	***Precision***	***Recall***	***F1-Score***	***Support***
Negative	0.61	0.48	0.54	304
Positive	0.60	0.71	0.65	329
Accuracy			0.66	633
M-avg	0.66	0.66	0.66	633
W-avg	0.66	0.66	0.66	633
VOTING CLASSIFIER – m2				
Class	***Precision***	***Recall***	***F1-Score***	***Support***
Negative	0.66	0.63	0.64	304
Positive	0.67	0.70	0.68	329
Accuracy			0.66	633
M-avg	0.66	0.66	0.66	633
W-avg	0.66	0.66	0.66	633

Finally, a comparison study of the above Ensembling techniques has been presented below. Overall, around 68% accuracy has been achieved which gets slightly better when the mean/average of all the ensembling techniques is considered.

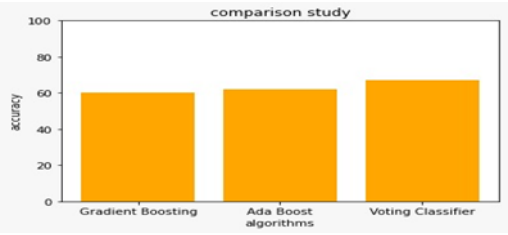

Fig 11. Final Comparison of Ensembling Techniques

4. Future Enhancements

A different way of approaching Sentiment Analysis is through Natural Language Processing (NLP) techniques. NLP enables computers to understand language as humans do. It involves techniques such as Vectorisation, Tokenisation, Lemmatisation, Normalisation, Stop Word Removal etc. which can be paired with powerful NLP libraries like iNLTK (Natural Language Tool Kit for Indic Languages). iNLTK is a Python library that is used to perform NLP operations in Indian languages and covers almost all of the most common Indian languages. It is easy to setup and has lets us perform some of the basic NLP tasks. One advantage of this approach is being able to use data sampling techniques such as SMOTE, Borderline SMOTE and ADASYN to balance out skewed dataset.

5. Conclusion

Sentiment Analysis in Kannada is a vast field with very little research on it. Hence, performing sentiment analysis on this topic was very challenging as well as rewarding. It was found that performing sentiment analysis on a dataset that was based on a negative topic adds an extra layer of challenge as the majority of the content is skewed towards being negative. Nevertheless, it was necessary to perform sentiment analysis as it was an important topic which provided deep knowledge about the people's emotions in the lockdown period showing a shift in their emotions and causes for the shift.

5.1. Acknowledgment

Authors would like to thank the Management of Sir M. Visvesvaraya Institute of Technology for their continued support in this endeavor.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Anil Kumar K.M, N. Rajasimha, "Analysis of users' Sentiments from Kannada Web Documents", Procedia Computer Science, pp 247-256, 2015.

[2] Shankar R, Suma Swamy, "Corpora Based Classification to Perform Sentiment Analysis in Kannada Language", The Design Engineering, pp 647-656, 2021.

[3] Shankar R, Suma Swamy, "Sentiment Analysis of Kannada Political Tweets using Support Vector Machines", International Journal of Recent Technology and Engineering, pp 5186-5191, 2020.

[4] Adeep Hande , Ruba Priyadharshini , Bharathi Raja Chakravarthi," KanCMD: Kannada CodeMixed Dataset for Sentiment Analysis and Offensive Language Detection", Proceedings of the Third Workshop on Computational Modeling of People's Opinions, PersonaLity, and Emotions in Social media, pp 54–63, 2020.

[5] Adeep Hande et al, "Hope Speech detection in under-resourced Kannada language", Computation and Language, pp 2108-2119, 2021.

[6] Mahesh B. Shelke , Sachin N. Deshmukh, "Recent Advances in Sentiment Analysis of Indian Languages", International Journal of Future Generation Communication and Networking, pp 1656-1675, 2020.

[7] Ramesh Chundi; Vishwanath R. Hulipalled; J.B Simha, "SAEKCS: Sentiment Analysis for English – Kannada Code SwitchText Using Deep Learning Techniques", International Conference on Smart Technologies in Computing, Electrical and Electronics, pp 327-331, 2020.

[8] Prashanth, Vidya Bhat, "A comparison of Bayesian and HMM based approaches in machine learning for emotion detection in native Kannada speaker", IEEE International WIE Conference on Electrical and Computer Engineering pp 1-6, 2019.

[9] Anil Kumar KM, Asmita Poojari, Mohana kumari M, "Pattern based approach for mining users opinion from Kannada web documents", Discovery Journal, pp 138-143, 2015.

[10] Venkatachalam Kandasamy et al, "Sentimental Analysis of COVID-19 Related Messages in Social Networks by Involving an N-Gram Stacked Autoencoder Integrated in an Ensemble Learning Scheme", Sensors (Basel), pp 15-21, 2021.

[11] Geethashree A and Dr. D.J Ravi, "Acoustic and Spectral Analysis of Kannada Emotional Speech", Third International Conference on Current Trends in Engineering Science and Technology, pp 42-49, 2017.

[12] Prashanth, Vidya Bhat, "Comparison of Hidden Markov Model and Artificial Neural Network Based Machine Learning Techniques Using DDMFCC Vectors for Emotion Recognition in Kannada", IEEE International WIE Conference on Electrical and Computer Engineering, pp 1-6, 2019.

[13] Abhinav Reddy Appidi et al, "Creation of Corpus and analysis in Code-Mixed Kannada-English Twitter data for Emotion Prediction", Proceedings of the 28th International Conference on Computational Linguistics, pp 6703—6709, 2020.

[14] Muvazima Mansoor et al, "Global Sentiment Analysis Of COVID-19 Tweets Over Time", Computation and Language, pp 1423-1430, 2020.

[15] Anna Kruspe et al, "Cross-language sentiment analysis of European Twitter messages duringthe COVID-19 pandemic", Social and Information Networks, pp 2008-2015, 2020.

[16] 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, pp 2375-4699, 2023.

[17] Ramanathan, Bing Liu and Alok Choudhary, "Sentiment Analysis of Conditional Sentences", Proceedings of Conference on Empirical Methods in Natural Language Processing, pp 213-221, 2009.

[18] Soumya et al, "Sentiment analysis of malayalam tweets using machine learning techniques", ICT Express, pp 300-305, 2009.

[19] Rajesh Bose, Raktim Kumar Dey, Sandip Roy and Debabrata Sarddar, "Analyzing Political Sentiment Using Twitter Data", Smart Innovation, Systems and Technologies, pp 427–436, 2018.

[20] Marjan Van de Kauter et al, "Fine-grained analysis of explicit and implicit sentiment in financial news articles", Expert Systems with Applications, pp 4999-5010, 2015.

[21] David Vilares, Miguel A. Alonso and Carlos G´omez-Rodr´ıguez, "Sentiment Analysis on Monolingual, Multilingual and Code-Switching Twitter Corpora", Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp 2-8, 2015.

[22] Alexandra Balahur, Marco Turchi, "Computer Speech & Language", ICT Express, pp 56-75, 2014.

[23] Julian Brooke et al, "Cross-Linguistic Sentiment Analysis: From English to Spanish", International Conference RANLP, pp 50-54, 2009.

[24] Erik et al, "Knowledge-Based Approaches to Concept-Level Sentiment Analysis", Intelligent Systems, IEEE, pp 12-14, 2013.

[25] Arouna et al, "Sentiment Analysis of Code-Mixed Bambara-French Social Media Text Using Deep Learning Techniques", Wuhan University Journal of Natural Sciences, pp 237–243, 2018.

[26] Anqi Cui et al, "Emotion Tokens: Bridging the Gap among Multilingual Twitter Sentiment Analysis", Asia Information Retrieval Symposium, pp 238-249, 2011.

[27] Alex et al, "Language-independent Bayesian sentiment mining of Twitter." The 5th SNA-KDD Workshop, pp 11-19, 2011.

[28] Preslav Nakov et al, "SemEval-2013 Task 2: Sentiment Analysis in Twitter", Second Joint Conference on Lexical and Computational Semantics, pp 312–320, 2013.

[29] Sachin Kumar et al, "Identifying Sentiment of Malayalam Tweets Using Deep Learning", Lecture Notes on Data Engineering and Communications Technologies, pp 391-408, 2018.

[30] Julio et al, "TASS 2014-The Challenge of Aspect-based Sentiment Analysis", Procesamiento de Lenguaje Natural, 61-68, 2015.