VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI

B.L.D.E ASSOCIATION'S

VACHANA PITAMAHA DR. P.G. HALAKATTI COLLEGE OF ENGINEERING & TECHNOLOGY
VIJAYAPURA - 586103

DEPARTMENT

OF

COMPUTER SCIENCE AND ENGINEERING

PROJECT REPORT
ON

"SENTIMENT ANALYSIS IN INDIAN REGIONAL LANGUAGE (KANNADA)"

UNDER THE GUIDANCE OF

Prof. D. M. IJERI

SUBMITTED BY

VINEET GIRGAONKAR	**2BL19CS110**
VAIBHAV THOBBI	**2BL19CS106**
NIKHIL GUGAWAD	**2BL19CS057**
ROHIT BIRADAR	**2BL19CS075**

2022-2023

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI

B.L.D.E ASSOCIATION'S

VACHANA PITAMAHA DR. P.G. HALAKATTI COLLEGE OF ENGINEERING & TECHNOLOGY
VIJAYAPURA - 586103

CERTIFICATE

This is to certify that the project entitled "Sentiment Analysis in Indian Regional Language (Kannada)" is a Bonafide work carried out by Vineet Girgaonkar (2BL19CS110),Vaibhav Thobbi (2BL19CS106), Nikhil Gugawad (2BL19CS057), Rohit Biradar (2BL19CS075). Submitted in 8th Semester of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi during year 2022- 2023. It is certified that all correction/suggestion indicated for Internal Assessment have been incorporated in report deposited in the departmental library. Project report has been approved as it satisfies the academic requirements in respect of project work.

GUIDE	HOD	PRINCIPAL
Prof. D.M. IJERI	Dr. PUSHPA .B. PATIL	Dr. V.G. SANGAM

EXAMINERS **SIGNATURE WITH DATE**

1.

2.

ACKNOWLEDGEMENT

It is overwhelming for me to bring out my project report titled **"SENTIMENT ANALYSIS IN INDIAN REGIONAL LANGUAGE (KANNADA)"** in the 8th Semester of the B.E. course in Computer Science and Engineering. The satisfaction that accompanies the successful completion of any work would be incomplete without naming the people who made it possible.

I would like to express my deep gratitude to our principal **Dr. V.G. SANGAM** for providing facilities for research in the college campus.

I would like to thank our head of the department **Dr. PUSHPA. B. PATIL** for providing all the facilities and fostering a congenial academic environment in the department.

I thank our guide **Prof. D. M. IJERI** for constant encouragement and support rendered to me throughout the course of work.

We also like to thank Project Coordinator **Prof. S.R. PATIL** and **Prof. D. M. IJERI** who provided their valuable guidance and suggestions in carrying out the project.

We would like to thank all the faculty members for their valuable suggestions and guidance.

Lastly, I would like to thank my parents and my friends who played an important role in completing this project.

VINEET GIRGAONKAR [2BL19CS110]

VAIBHAV THOBBI [2BL19CS106]

NIKHIL GUGAWAD [2BL19CS057]

ROHIT BIRADAR [2BL19CS075]

DECLARATION

We here by declare that the project entitled **'Sentiment Analysis in Indian Regional Language (Kannada)'** submitted us in partial fulfilment for the award of degree of BE in Computer Science & Engineering is a record of my original work carried out under the guidance of **Prof. D.M. Ijeri** Assistant Professor in Computer Science & Engineering department, BLDEA's V. P. Dr. P. G. Halakatti College of Engineering & Technology, Vijayapura

Name of the students Signature

Mr. VINEET GIRGAONKAR

Mr. VAIBHAV THOBBI

Mr. NIKHIL GUGAWAD

Mr. ROHIT BIRADAR

ABSTRACT

Natural Language Processing has a variety of uses, including sentiment analysis. Reading, comprehending, and extracting meaningful information from unstructured material are the objectives of natural language processing. Almost 3.2 billion people are active internet users right now. All business has gone online as a result of advances in technology, including e-commerce, movie ticket buying, education, and other areas.

In order to advance a firm, it is crucial to understand user opinions on social media. Sentiment Analysis can be used to examine these users' opinions. Sentiment analysis in the Kannada language has received a fair amount of earlier research, with an accuracy rate of 72%. The methodologies utilised for sentiment analysis of texts in Indian regional languages utilising natural language processing are discussed in this work. focused on classifying the four emotions of anger, fear, joy, and sadness using machine learning algorithms like Linear SVC, Logistic Regression, SGD, K-Nearest Neighbors, Multinomial Naive Bayes, and Random Forest Classifier. The Linear Support Vector Classifier, with an accuracy of 87.25%, is the algorithm that performs the best overall.

INDEX

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

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.

Models, approaches, and strategies used in natural language processing explain how the language we use affects how we think and the outcomes we get. The term "sentiment analysis" describes the systematic identification, extraction, quantification, and study of affective states and subjective information using natural language processing, text analysis, computational linguistics, and biometrics.

Sentiment analysis is a task in natural language processing and information extraction that looks for the writer's emotions as they are expressed in reviews, inquiries, and requests, whether they are positive or negative. Sentiment analysis broadly seeks to ascertain a speaker's or writer's attitude towards a subject or the general polarity of a document. His or her judgement, judgement, affective state, or purposeful emotional communication can all be categorized as attitude. Sentiment analysis has become increasingly necessary in recent years due to the exponential rise in internet usage and public opinion exchange. The proposed system focuses on classifying the emotions such as Anger, fear, joy, and sadness for textual based sentiment analysis. It may be sentence-based, classifying each and every sentence in the text that expresses sentiment. Sentiment analysis can be phrase-based, where the phrases in a sentence are categorized according to the polarity based on some patterns of their recurrence.

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.

CHAPTER 2

LITERATURE SURVEY

A.Pasumpon Pandian [1] proposed performance evaluation and comparison using deep learning techniques in sentiment analysis (2021). With the use of several ensemble models and by emphasizing the various features needed for sentiment analysis, the author of this study combined two techniques. The performance analysis of surface and deep ensembles, the advantages of deep learning, and the characterization of current approaches are all included in this work. The data is broken down into six data sets, three of which contain 43 tweets totaling 728 words. These studies demonstrate that the combination of information from many sources, such as impact word vectors, generic features, and surface characteristics, will lead to an improvement in sentiment analysis tasks.

Joshi and Pathak [2] proposed Sentiment Analysis on code-mixed Dravidian Languages, A Non-Linguistic Approach (2021). To perform sentiment analysis on this kind of language, many people utilize a combination of two or more languages, known as code mixed language. In this study, the authors extracted features from the data set using TF-IDF and the Keras Tokenizer API. Machine learning (MNB, DTC) and neural networks (ANN, CNN) approaches are used on these extracted characteristics, and the results are submitted to the HASOC 2021 shared problem to enhance transfer learning (BERT). They used 40230, 69675, and 15800 unique words from the Dravidian languages of Malayalam, Telugu, and Kannada, respectively, . Precision recall and F1 score of 0.78 were obtained for Malayalam language MNB utilizing combined word n-gram. With an F1 score of 0.65 for Tamil, ANN did better. In terms of Kannada language, CNN and ANN scored 0.63 and 0.64 on the F1 scale, respectively. The drawback of this strategy is that, in contrast to character-gram feature and combined word feature, machine learning approach fails horribly to code mixed sentences.

Rakshita et. al.[3] Proposed Sentiment analysis of Indian regional languages on social media (2021). This model primarily uses five regional languages, including Telugu, Tamil, Malayalam, Hindi, and Kannada. First, tweets are scraped from Twitter using Twitter's API. Next, text blob is used. Finally, customer reviews are given various sentiment scores and classified as positive, negative, or neutral by using test classification model. Sentiments are determined using the input data's semantic relations and the frequency of each pre-defined word. This results in precise output. Emojis and images might be taken into consideration for

this dataset for any additional information, but the downside of this study is that it only uses text data from Twitter.

Basiri and Nemati [4] proposed a novel fusion based deep learning model for sentiment analysis of COVID-19 tweets (2021). For the purpose of sentiment analysis of Tweets related to the Corona virus from eight different countries, the authors of this study implement a novel method based on the combination of four Deep Learning models (NBSVM, DistillBERT, BiGRU, and FastText) and one traditional supervised machine learning model (CNN). They used data from Twitter and Google trends as two sources for this investigation. The model is trained using the Stanford Sentiment140 Data set. Positive tweets in this study indicate that the individual is upbeat about the illness, but negative tweets indicate that they are concerned or unhappy about potential negative effects on their lives. The drawback of this study is that it ignores how worldwide Covid-19 news and statistics affect the general mood of other nations, and that the keywords chosen for information seeking and tweet filtering are independent of the nation for which the dataset is being extracted. The benefit of this work is that the fusion models that are suggested may accurately analyze the coronavirus-related dataset and can be utilized to identify feelings with promising performance.

Bera and Ghose [5] proposed Sentiment Analysis of Multilingual Tweets Based on Natural Language Processing (NLP) (2021). The authors of this study first conducted a thorough analysis of natural language processing (NLP) using simple neural networks (SNN), convolutional neural networks (CNN), and long short-term memory (LSTM) neural networks. They then created an amalgamated model by layering a CNN on top of the LSTM. A total of 4,000 samples of reviews in the languages of Bengali, Hindi, and English are used to create the outputs for the models mentioned above. The Sentiment Analysis dataset's accuracy of classification for the proportion of positive and negative statements is 84.1%, with an average time complexity. The drawback is that improved models may be anticipated in the future that provide better accuracy with faster execution to effectively manage human perception.

Ranjitha and Bhanu [6] proposed an improved sentiment analysis for Dravidian language Kannada using decision tree algorithm with efficient data dictionary (2021). The authors of this work suggested a method that computationally recognizes and classifies opinions conveyed in a piece of Kannada-language writing in order to ascertain if the writer has a favorable, negative, or neutral attitude toward a certain topic or product. The decision tree technique for Kannada sentiment analysis is used to achieve this. Additionally, the training's

data set is culled from a variety of websites, including Prajwani, One India News, and Wedunia. The results were obtained with 85% accuracy, 0.78 precision, and 0.79 recall. This study's drawback is that some terms in the Kannada language cause confusing messages to be produced by machine translation, which leads to incorrect results.

Madan and Ghose [7] proposed Sentiment Analysis for Twitter Data in the Hindi Language (2021). In this work, sentiment analysis for Hindi-language Twitter data was conducted. To connect to the Twitter app and extract the tweets, they used the Tweepy API. The resource utilized to create a sentiment classifier is an improved Hindi SentiWordNet. To categorize the various sentiments of tweets, they have used various classification approaches, including LR, MNB, SVM, Decision Tree, and Nearest Neighbor. They have completed sentiment classification and, at long last, sentiment analysis using both the LBA and HBA approaches. Only Hindi tweets are taken into account, which is a flaw in the system. Using LR and MI, this system has a 93.2% accuracy rate.

Kulkarni et. al [8] Proposed a Marathi tweet-based sentiment analysis data set (2021). In this paper, the authors show the baseline classification results utilizing CNN, LSTM, UMFiT, and BERT base deep learning models as well as the first significant publicly available Marathi Sentiment Analysis Dataset L3CubeMahaSent. There are 16,000 unique tweets in the data collection, which are divided into three categories. Using CNN yielded an accuracy of 83.24%, LSTM yielded an accuracy of 82.89%, UMFiT produced an accuracy of 80.80%, and BERT produced an accuracy of 84.13%.

Rajani Shree and Shambhavi [9] proposed POS tagger model for Kannada text with CRF++ and deep learning approaches (2020). In this article, authors demonstrate two strategies for training parts of speech tagging on Kannada words. Supervised machine learning is the first strategy. CRF ++0.50. The second strategy combines a deep learning technique with word embedding. The data set that was utilized in this implementation. Downloaded from TDIL are 1200 sentences with tags in Kannada. 76% accuracy when using CRF++ 0.50. is acquired. Additionally, 71% accuracy is achieved using the deep learning technique. The study's drawbacks include the necessity for a large amount of words to train the model (for example,different verb tenses where a single verb tag for all action words would increase accuracy) and the challenge in identifying proper and common nouns for the model.

Sharma and Ghose [10] proposed sentimental analysis of twitter data with respect to General Elections in India (2020). The authors of this study used the Twitter API to gather tweets. The

tweet is analyzed using the R Language, which is also utilized for pre-processing. Two applicants were given consideration for this study. AYLIEN, a text analysis extension of Rapid Miner, is used in this study to do named entity extraction. They also employed the NRC dictionary-based approach and the lexicon-based approach for sentiment analysis. The noise in the tweets and the lack of sentiment labels made the algorithm more difficult. The outcome of ten was in complete agreement with the actual election outcomes discovered in May 2019.

Shalini et. al [11] Proposed Sentiment analysis for code mixed Indian social media text with distributed representation (2018). By crawling Facebook comments, the authors of this work created a corpus of code-mixed Kannada and English. discussed how distributed representation techniques performed for the sentiment analysis job and reported comparisons between several deep learning (CNN, bi-LSTM) and machine learning (CNN) techniques (Fast Text, Doc2Vec). With a 71.50% accuracy rate, CNN gave the most accurate findings for the English-Kannada code-mixed language. In the future, we can enhance the representation by capturing the subtleties seen in social media content that has been code-mixed.

Nasr and Shaaban [12] proposed a building sentiment analysis model using Graphlab (2017). In this study, the authors employed graft lap to create sentiment models and a variety of text feature selection approaches, such as SVM, logistic regression, and boosted trees, to forecast positive and negative sentiment. To simulate client opinions, data was collected from hotel reviews obtained from the TripAdvisor website. Regression using logit. SVM using the bigrams feature had the highest accuracy, at 93.50%, with an accuracy of about 92.75%.

Naidu and Bharti [13] proposed sentiment analysis using telugu SentiWordNet (2017). In this study, SentiWordNet was used by the authors to propose a two-phase sentiment analysis for Telugu News sentences. The subjectivity of the sentence is categorized in the first phase. If a sentence is tagged as Objective, it is neutralized and disregarded in the following steps. And the SNS file stores subjective sentences. The second stage of the. The input is a sinus file. Additionally, SentiWordNet's positive and negative keyword files are matched to each word in the phrase to evaluate its attitude. Eenadu, Sakshi, Andhra Jyothi, and Andhra Bhumi are four Telugu newspapers from which data for this study has been gathered. This includes 1400 sentences in Telugu. For classification of subjectivity and mood, this system has accuracy rates of 74% and 81%, respectively. The drawbacks include the difficulty in locating annotated datasets for NLP tasks on regional languages, such as POS tagging and text summarization.

Impana and Kallimani [14] proposed cross-lingual sentiment analysis for Indian regional languages (2017). The authors of this study used the bilingually constrained recursive autoencoder model to analyze the sentiment of two languages (BRAE). On the Word Net data set, they have worked. The model runs learning of phrase embedding first to improve the analysis's precision. Furthermore, cross-training is used to fine-tune the embedding. To create the final classifier, the supervised training performance was carried out utilizing the smaller Kannada Labelled Corpora and the resource-rich English Labelled Corpora.

Rohini et. al [15] Proposed Domain-based sentiment analysis in regional language. Kannada, using a machine learning algorithm (2016). In this study, the authors use machine learning techniques to do a domain-based sentiment analysis of movies in a particular regional language. Approximately 100 movie reviews from Kannada websites that have undergone POS tagging serve as the training data set for the decision tree algorithm ID 3 used for this investigation. The system's disadvantages include the inability to handle ambiguous words and the lack of tools available for a given topic. For the Kannada data set, this system has a precision of 0.78 and recall of 0.79, and for the English data set, it has a precision of 0.86 and recall of 0.67.

Phani et. al [16] Proposed Sentiment Analysis of tweets in three Indian languages. The SAIL dataset was utilized by the authors of this study to acquire outcomes in three languages. utilizing six various classifiers. Multinomial Naive Bayes, Logistic Regression, Decision Tree, Random Forest SVM SVC, and SVM linear SVC are all available in the Scikit Learn package. The absence of test data and stop words, at least for Tamil, are limitations. Bengali had 23 misclassified cases out of 53 speakers, whereas Hindi had just three out of 56. The accuracy for the Tamil language is 51.25%. The accuracy rate for Bengali is 60.83%. In contrast, Hindi has a 96.43% accuracy rate.

Hegde and Padma [17] proposed sentiment analysis for kannada language using mobile product reviews (2015). In this study, the authors attempted to extract aspects using a lexicon-based strategy. Due to its computational complexity and robustness, the Naive Bayes classification model is also used to analyse the polarity of the statement. The data set is a modified corpus taken from gadget local's weekly review column. U.B. Pavanaja are taken into account.Sentiment shifts are not handled by the system, and multi-class is not addressed. This method has a recall of 75%, accuracy of 65%, and precision of 62.5%.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Problem Statement

The proposed system aims to develop a sentiment classification model capable of accurately categorizing kannada sentences into multiple sentiments such as Sadness, Fear, Anger, and Happiness.

3.2 Drawbacks of Existing System

- The existing system focuses solely on discerning positive and negative sentiments, without taking into account other sentiment categories.
- The majority of the existing system relied on translating the text into English before performing the classification task.
- The sentences processed by the existing system contain a restricted number of words.
- The existing systems work with a limited dataset due to resource constraints in the Kannada language, which resulted in a reduced amount of training data available for classification.

3.3 Proposed System

- The proposed system offers classification of Kannada sentences into various sentiment categories, including Sad, Fear, Anger, and Joy, allowing for a more nuanced analysis of emotions.
- The proposed system utilizes a substantial labelled dataset comprising 2000 sentences, enabling a more comprehensive training and evaluation process.
- The proposed system handles lengthy sentences, typically averaging around 10 words, ensuring a sufficient linguistic context for accurate classification.

3.4 Applications

- **Social Media Monitoring**

  Twitter sentiment analysis enables us to monitor what is being said about the product or service on social media and can assist in identifying irate clients or unfavourable mentions before they become a huge catastrophe.

- **Customer Service**

  The objections that were expressed on Twitter can be resolved with the aid of Twitter sentiment analysis. Sixty percent of clients who complain on social media say they want a response within an hour.

- **Market Research**

  We can stay one step ahead of the competition by using Twitter sentiment analysis. We can concentrate on these areas when promoting the firm by locating the problems that our rivals have.

- **Political Campaigns**

  Monitoring the tweets on particular dates, such as the day of the presidential debate, and observing unfavourable or good replies as well as the primary phrases expressed that day.

- **Product analysis**

  As soon as a new product is released, find out what the public is saying about it. You may also examine feedback from years ago that you may have never seen. You can train sentiment analysis models to identify only the information you require by doing keyword searches for certain product features (such as functionality, interface, and user experience).

CHAPTER 4

SYSTEM REQUIREMENTS AND SPECIFICATION

4.1 Hardware Requirements

- Processor: Minimum 1 GHz, Recommended 2GHz or more.
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi).
- Hard Drive: Minimum 32 GB, Recommended 64 GB or more.
- Memory (RAM): Minimum 1 GB, Recommended 4 GB or above.

4.2 Software Requirements

- Programming Language: Python 3.9.1
- Integrated Development Environment (IDE): IDLE/PyCharm 2022.3.2
- Library: NLTK (Natural Language Tool Kit).
- Operating System: Windows 10/ Ubuntu.
- Web Browser: Google Chrome/Firefox.

CHAPTER 5

SYSTEM DESIGN AND IMPLEMENTATION

The steps involved in kannada language sentiment analysis are shown in Fig. 5.1 above. The sentences will be taken from manually prepared dataset. After taking a custom input sentence, a sentence is tokenized, or broken up into smaller words. 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 the, which, and other words 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.

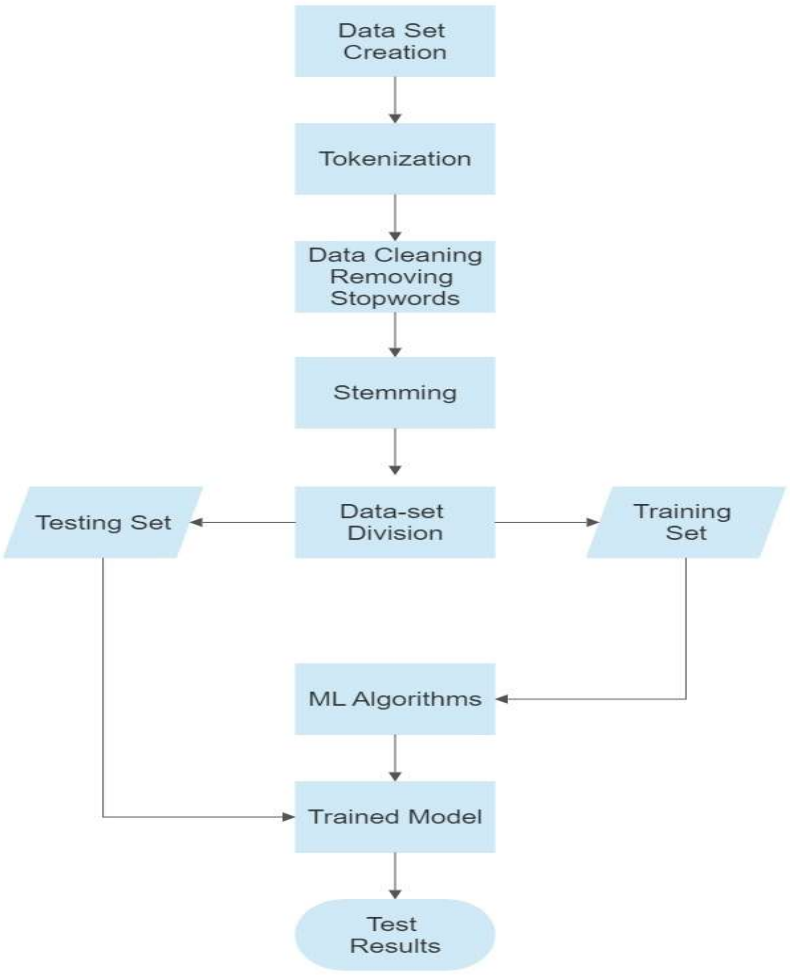

Fig 5.1 Steps involved in Sentiment Analysis of Indian Regional Language (Kannada)

5.1 Dataset Creation

A machine learning dataset is a collection of data that is used to train the model. The machine learning algorithm learns how to make predictions using a dataset as an example. Text, image, audio, video, and numeric data are among the common sorts of data. In order for the algorithm to comprehend what the desired output is, the data is typically first labelled or annotated.Data set consists of 604 joy, 520 anger, 520 sad and 356 fear sentences as shown in fig. 5.1

5.2 Tokenization

As shown in the fig. 5.1 Tokenization involves cutting the raw text into manageable pieces. Tokenization divides the original text into tokens, which are words and sentences. These tokens aid in context comprehension or model development for NLP. By examining the word order in the text, tokenization aids in comprehending the text's meaning. For example, the text “It is raining” can be tokenized into ‘It’, ‘is’, ‘raining’. Tokenization can be done using a variety of tools and frameworks. Some of the libraries that can be utilized to do the work include NLTK, Gensim, and Keras.

5.3 Data Cleaning

In NLP, data cleaning is a crucial step. The dataset is like a collection of words that the computer cannot understand without cleansing the data as shown in fig.5.1. 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 (...).

5.4 Removing Stopwords

From the fig.5.1, Stopwords 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 stopwords are removed. The data size will drop as a result of eliminating these stopwords, 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.

5.5 Stemming

In Natural Language Processing (NLP), the process of stripping a word down to its root or suffix- and prefix-affixed word stem is known as stemming as shown in fig. 5.1. In contrast, a stemming algorithm normalizes language by condensing various word forms to their standard form. Through the removal of affixes, the base form of the words is extracted using this method. It is analogous to trimming a tree's branches back to the trunk. For instance, the stem of the words "eating," "eats," and "eaten" is "eat". Stemming is used by search engines to index the words. Because of this, a search engine can only record the stems of a word rather than all of its variations. Stemming does this by reducing the size of the index and improving retrieval precision.

5.6 Dataset Division

In data science or machine learning, data splitting comes into the picture when the given data is divided into two or more subsets so that a model can get trained, tested and evaluated as shown in fig. 5.1. In practice or in real-life projects, data splitting is an important aspect, and it becomes a must when models are based on the data as it ensures the making of machine learning models. Usually, we create two or three parts of the main dataset. If two splits are there, it means one will be utilised for training and another one will be used for testing.

5.7 Training Set

The training data is the biggest (in -size) subset of the original dataset, which is used to train or fit the machine learning model. Firstly, the training data is fed to the ML algorithms, which lets them learn how to make predictions for the given task.

5.8 Testing Set

Once we train the model with the training dataset, it's time to test the model with the test dataset. This dataset evaluates the performance of the model and ensures that the model can generalize well with the new or unseen dataset. The test dataset is another subset of original data, which is independent of the training dataset. However, it has some similar types of features and class probability distribution and uses it as a benchmark for model evaluation once the model training is completed. Test data is a well-organized dataset that contains data for each type of scenario for a given problem that the model would be facing when used in the real world. Usually, the test dataset is approximately 20-25% of the total original data for an ML project.

5.9 TF-IDF

For feature extraction, we employed the Term Frequency Inverse Document Frequency (TF-IDF) approach. Term Frequency - Inverse Document Frequency (TF-IDF) is a widely used statistical method in natural language processing and information retrieval. It measures how important a term is within a document relative to a collection of documents (i.e., relative to a corpus). Words within a text document are transformed into importance numbers by a text vectorization process. There are many different text vectorizations scoring schemes, with TF-IDF being one of the most common.

As its name implies, TF-IDF vectorizes/scores a word by multiplying the word's Term Frequency (TF) with the Inverse Document Frequency (IDF)

Term Frequency: TF of a term or word is the number of times the term appears in a document compared to the total number of words in the using eqn. 5.1

$$TF = \frac{\text{number of times the term appears in the document}}{\text{total number of terms in the document}} \quad \text{... eqn. 5.1}$$

Inverse Document Frequency: IDF of a term reflects the proportion of documents in the corpus that contain the term using eqn. 5.2. Words unique to a small percentage of documents (e.g., technical jargon terms) receive higher importance values than words common across all documents (e.g., a, the, and).

$$IDF = log(\frac{\text{number of the documents in the corpus}}{\text{number of documents in the corpus contain the term}}) \quad \text{... eqn. 5.2}$$

The TF-IDF of a term is calculated by multiplying TF and IDF scores using eqn. 5.3.

$$TF\text{-}IDF = TF * IDF \quad \text{... eqn. 5.3}$$

TF-IDF is useful in many natural language processing applications. For example, Search Engines use TF-IDF to rank the relevance of a document for a query. TF-IDF is also employed in text classification, text summarization, and topic modelling.

Note that there are some different approaches to calculating the IDF score. The base 10 logarithm is often used in the calculation using eqn.5.4. However, some libraries use a natural logarithm. In addition, one can be added to the denominator as follows in order to avoid division by zero.

$$IDF = log(\frac{\text{number of the documents in the corpus}}{\text{number of documents in the corpus contain the term} + 1}) \quad \text{... eqn. 5.4}$$

A sparse matrix is created following the extraction of TF-IDF features. Classification is done using this matrix. To train the machine, we employed in-language classification. The classifiers are trained using the same language as the text in this method. To analyze the sentiment, it significantly depends on resources being available in the same language. As a result, all training and testing materials are in Kannada. To train and test the data, we employed a number of classifiers, including Linear SVC, Logistic Regression, SGD Classifier, K-Neighbors Classifier, Multinomial NB and Random Forest Classifier.

5.10 Machine Learning Algorithms

Preprocessing will take place before classification, as was stated in prior sections. The classification process, which divides tweets into four categories—joy, anger, fear, and sad—is a crucial part of sentiment analysis. We employ categorization algorithms in order to categorize sentences. SGD Classifier, Linear SVC,K-Neighbors Classifier, Multinomial NB, and Random Forest Classifier are some of the classification methods we employed in our model. Using these algorithms, sentences are to be classified, and effectiveness is to be evaluated.

5.10.1 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. Some features can be fed to the classifier after obtaining the hyperplane to see what class it would predict. First, draw a straight line that is unbiased to all classes and has the distance between the +support vector and line equal to the -support vector. Second, the narrow margin values indicate that the support vectors are overly sensitive. Support vector variance won't be robust if the dataset or classifier change.

5.10.2 Logistic Regression

It is a classification technique that uses supervised learning to predict the likelihood of the target variables. Since the goal or dependent variable has a dichotomous nature, there are two alternative classes. The binary nature of the target variable has data coded as 1 for success and 0 for failure. The p(A=1) function of B is predicted by the logistic regression model. It is the simplest algorithm in machine learning that may be applied to different classification issues.

5.10.3 Stochastic Gradient Descent Classifier

A useful approach for adjusting is stochastic gradient descent (SGD), which is used to find the parameters of a function that reduces a cost function. It is utilized for convex loss function-based differential learning of linear classifiers, such as SVM and logistic regression. Because it updates the coefficient for every training instance rather than at the conclusion of the instance, it can be successfully used to bigger data scales. SGD essentially establishes a simple SGD learning method that supports various loss functions and classification penalties. Scikit Learn, which has an SGD classifier module, can be used to implement SGD classification.

5.10.4 K-Neighbors Classifier

The machine learning algorithm K-Nearest Neighbor is based on the supervised learning method. It takes into account how similar existing cases and new data are, and it places the new instance in a category that is comparable to existing categories. This algorithm is usually only used to solve classification problems, while it is also used for regression. Since it doesn't make any assumptions about the underlying data, it is regarded as a non-parametric algorithm. This algorithm also goes by the label "lazy learner algorithm" because, rather than learning from the

training set and performing some measurements on the dataset, it saves the dataset throughout the classification phase.

5.10.5 Multinomial Naïve Bayes

The Multinomial Naive Bayes algorithm is based on the Selective learning technique that is primarily employed in NLP. It predicts a set of texts as a newspaper story or email 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 using eqn.5.4.

$$P(X|Y) = \frac{P(Y|X)\,P(X)}{P(Y)} \quad \text{... eqn. 5.4}$$

5.10.6 Random Forest Classifier

Supervisory learning algorithms like Random Forests are used. Both classification and regression are accomplished with it. The algorithm is simple and very adaptable. The trees that make up a forest. A forest will grow larger as more trees are present. On arbitrary picked data samples, it creates a few decision trees, receives predictions from each tree, and then uses voting to determine which is the best option. For example, picture classification and recommendation engines are only two examples of how Random Forest is used. Additionally, Random Forest is utilized to categorize dependable loan applicants, uncover fraudulent behavior, and anticipate diseases. A dataset's key attributes are chosen via the Boruta Algorithm, which is at its core.

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Performance Metrics for Classification

6.1.1 Precision

Precision, used in document retrievals, may be defined as the number of correct documents returned by our ML model. We can easily calculate it by confusion matrix with the help of following eqn. 6.1

$$Precision = \frac{TP}{TP + FP} \quad \text{... eqn. 6.1}$$

6.1.2 Recall

Recall may be defined as the number of positives returned by our ML model. We can easily calculate it by confusion matrix with the help of following formula eqn. 6.2

$$Recall = \frac{TP}{TP + FN} \quad \text{... eqn. 6.2}$$

6.1.3 F1 Score

This score will give us the harmonic mean of precision and recall. Mathematically, F1 score is the weighted average of the precision and recall. The best value of F1 would be 1 and worst would be 0. We can calculate F1 score with the help of following formula eqn. 6.3

$$\boldsymbol{F1} = \mathbf{2} * (\boldsymbol{precision} * \boldsymbol{recall}) / (\boldsymbol{precision} + \boldsymbol{recall}) \quad \text{…eqn.6.3}$$

F1 score is having equal relative contribution of precision and recall.

We can use classification_report function of sklearn.metrics to get the classification report of our classification model.

6.1.4 Support

Support is the actual number of instances of the class in the given dataset. Unbalanced support in the training data may be a sign of structural flaws in the classifier's reported scores and may point to the need for stratified sampling or rebalancing. Support doesn't alter depending on the model; instead, it diagnoses the evaluation procedure.

6.1.5 Accuracy

The accuracy metric is one of the simplest Classification metrics to implement, and it can be determined as the number of correct predictions to the total number of predictions.
It can be formulated using eqn.6.4:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ number\ of\ predictions} \quad \text{... eqn. 6.4}$$

To implement an accuracy metric, we can compare ground truth and predicted values in a loop, or we can also use the scikit-learn module for this.

6.1.6 Macro Average

Macro averaging is perhaps the most straightforward among the numerous averaging methods. The macro-averaged F1 score (or macro F1 score) is computed using the arithmetic mean (aka unweighted mean) of all the per-class F1 scores.
This method treats all classes equally regardless of their support values.

6.1.7 Weighted Average

The weighted-averaged F1 score is calculated by taking the mean of all per-class F1 scores while considering each class's support.
The 'weight' essentially refers to the proportion of each class's support relative to the sum of all support values. With weighted averaging, the output average would have accounted for the contribution of each class as weighted by the number of examples of that given class.

6.2 Data Collection

The information we gathered for the sentiment analysis of the kannada language is shown in the fig. 6.1 below. For the model's training, we have gathered more than 2000 Kannada phrases from the internet. Four categories—joy, fear, sad, and anger—were assigned to these expressions. These data are kept in XLSX (Microsoft Excel) format.

	A	B
1	Sentences	sentiment
2	ಕೆಲವು ವ್ಯಕ್ತಿಗಳ ಅಡೆತಡೆಗಳನ್ನು ಸೃಷ್ಟಿಸುವ ಪಕ್ಷಪಾತದ ವರ್ತನೆಗಳನ್ನು ನಾನು ನೋಡಿದಾಗ ನಾನು ಕೋಪಗೊಳ್ಳುತ್ತೇನೆ	Anger
3	ಅವರು ನ್ಯಾಯಾಧೀಶರ ಮೇಲೆ ತಮ್ಮ ಕೋಪವನ್ನು ವ್ಯಕ್ತಪಡಿಸಿದರು	Anger
4	ರಾಜಕಾರಣಿಗಳು ದ್ವೇಷದ ಮಾತುಗಳನ್ನು ಬಳಸುವುದನ್ನು ನೋಡುವುದು ನೋವಿನ ಅನುಭವವಾಗಿರುತ್ತದೆ	Sad
5	ಭ್ರಷ್ಟಾಚಾರವು ಅಸಮಾನತೆ ಮತ್ತು ಅನ್ಯಾಯಕ್ಕೆ ಕಾರಣವಾದಾಗ, ದುರ್ಬಲ ಜನಸಂಖ್ಯೆಗೆ ಹಾನಿಯಾದಾಗ ಅದು ನನಗೆ ಸಿ	Anger
6	ನಿರುದ್ಯೋಗ ತರುವ ಕಳಂಕ ಮತ್ತು ತಾರತಮ್ಯವನ್ನು ಎದುರಿಸುತ್ತಿರುವುದಕ್ಕೆ ನಾನು ವಿಷಾದಿಸುತ್ತೇನೆ	Sad
7	ಗುಣಮಟ್ಟದ ಶಿಕ್ಷಣವನ್ನು ಪಡೆಯಲು ಸಾಧ್ಯವಾಗುತ್ತಿಲ್ಲ ಎಂಬ ದುಃಖವು ದೀರ್ಘಕಾಲೀನ ಪರಿಣಾಮಗಳನ್ನು ಉಂಟುಮಾ	Sad
8	ಜಾಗತಿಕ ತಾಪಮಾನ ಏರಿಕೆಯಿಂದಾಗಿ ಪರಿಸರ ವ್ಯವಸ್ಥೆಗಳು ಮತ್ತು ಆವಾಸಸ್ಥಾನಗಳ ನಾಶವನ್ನು ವೀಕ್ಷಿಸುವ ಖಿನ್ನತೆಯು	Sad
9	ರೋಗಗಳು ಚಿಕಿತ್ಸೆ ಮತ್ತು ಔಷಧಿಗಳಿಗೆ ಹೆಚ್ಚು ನಿರೋಧಕವಾಗುತ್ತಿವೆ ಎಂಬ ಭಾವನೆಯ ದುಃಖವು ರೋಗಲಕ್ಷಣಗಳನ್ನು ಗ	Sad
10	ಕಾಯಿಲೆಯಿಂದ ಪ್ರಯಾಣಿಸಲು ಸಾಧ್ಯವಾಗುತ್ತಿಲ್ಲ ಎಂಬ ಭಯ ನನ್ನನ್ನು ಗಾಬರಿಗೊಳಿಸುತ್ತಿದೆ	Fear
11	ಶಾಲಾ ಜೀವನವು ನನಗೆ ಅಪಾರ ಸಂತೋಷವನ್ನು ನೀಡಿದೆ	Joy
12	ನ್ಯಾಯೋಚಿತತೆಯ ತತ್ವಗಳಿಗೆ ವಿರುದ್ಧವಾದ ಪಕ್ಷಪಾತವನ್ನು ನಾನು ಗಮನಿಸಿದಾಗ ನಾನು ಆಕ್ರೋಶ ಮತ್ತು ಕೋಪವನ್ನು	Anger
13	ಒಂದು ಸಮುದಾಯವು ಒಗ್ಗೂಡಿ ಸವಾಲುಗಳನ್ನು ಜಯಿಸಿ ಸುಖಾಂತ್ಯವನ್ನು ಕಂಡಾಗ ಸಿಗುವ ಆನಂದ ಉತ್ಕೃಷ್ಟವಾಗಿದೆ	Joy
14	ಸಮಯದ ಕ್ಷಣಿಕ ಸ್ವಭಾವವು ನಮಗೆ ಚಿಂತೆವನ್ನುಂಟುಮಾಡುತ್ತದೆ, ಅದು ನಮ್ಮ ಜೀವನದ ಮೇಲೆ ನಾವು ನಿಯಂತ್ರಣವನ	Sad
15	ಸಾವು ಇತರರಿಗೆ ಉಂಟುಮಾಡುವ ನೋವು ಮತ್ತು ನೋವನ್ನು ನಾನು ನೋಡಿದಾಗ ನನಗೆ ಹತಾಶೆ ಉಂಟಾಗುತ್ತದೆ	Sad
16	ಶಾಲೆಯಲ್ಲಿ ಕಾರ್ಯನಿರ್ವಹಿಸುವ ಒತ್ತಡವು ವಿದ್ಯಾರ್ಥಿಗಳಿಗೆ ಬಹಳಷ್ಟು ದುಃಖ ಮತ್ತು ಚಿಂತೆಯನ್ನು ಉಂಟುಮಾಡಬಹ	Sad
17	ಸರ್ಕಾರಗಳು ಮತ್ತು ಸಂಸ್ಥೆಗಳು ಹಸಿವಿನ ಸಮಸ್ಯೆಯನ್ನು ಪರಿಹರಿಸಲು ಆದ್ಯತೆ ನೀಡಲು ವಿಫಲವಾಗುವುದನ್ನು ನೋಡಿದ	Anger
18	ಭಾರತದ ಸಾಂಸ್ಕೃತಿಕ ಪರಂಪರೆ, ಅದರ ಪ್ರಾಚೀನ ದೇವಾಲಯಗಳು, ಸ್ಮಾರಕಗಳು ಮತ್ತು ಹೆಗ್ಗುರುತುಗಳು, ದೇಶದ ಶ್ರೀಮಂ	Joy
19	ನನ್ನ ವೃತ್ತಿಜೀವನದ ಭವಿಷ್ಯದ ಮೇಲೆ ಪರಿಣಾಮ ಬೀರುವ ಹಿನ್ನಡೆಗಳು ಅಥವಾ ವೈಫಲ್ಯಗಳನ್ನು ನಾನು ಎದುರಿಸಿದಾಗ ನ	Sad
20	ಚಿನ್ನದಿಂದ ಮಾಡಿದ ಉಡುಗೊರೆಯನ್ನು ಸ್ವೀಕರಿಸುವ ಆನಂದವು ಒಂದು ಅಮೂಲ್ಯ ಕ್ಷಣವಾಗಿದೆ	Joy
21	ಪರಿಸರ ವ್ಯವಸ್ಥೆಗಳು ಮತ್ತು ನಮ್ಮ ಗ್ರಹದ ಸೂಕ್ಷ್ಮ ಸಮತೋಲನದ ಮೇಲೆ ಮಾಲಿನ್ಯದ ಪರಿಣಾಮವನ್ನು ನೋಡುವ ನೋ	Sad
22	ನಾಯಿಗಳು ಅವುಗಳ ಮಾಲೀಕರಿಂದ ನಿಂದನೆ ಮತ್ತು ನಿರ್ಲಕ್ಷಿಸುವುದನ್ನು ನೋಡಿದಾಗ ನನಗೆ ದುಃಖವಾಗುತ್ತದೆ	Sad
23	ಧನ್ಯವಾದಗಳು! ನಾನು ಅವನನ್ನು ಪ್ರತಿ ವಾರ ವೀಕ್ಷಿಸಲು ಇಷ್ಟಪಡುತ್ತೇನೆ	Joy
24	ಪೋಷಕರು ಮತ್ತು ಮಗುವಿನ ನಡುವಿನ ಬೇಷರತ್ತಾದ ಪ್ರೀತಿ ಮತ್ತು ಬಾಂಧವ್ಯವು ನನಗೆ ಆನಂದದಾಯಕ ನೆರವೇರಿಕೆಯ	Joy
25	ದೀಪಾವಳಿ, ಹೋಳಿ ಮತ್ತು ನವರಾತ್ರಿಯ ಭಾರತದ ಹಬ್ಬಗಳು, ಅವರ ಸಂತೋಷದಾಯಕ ಆಚರಣೆಗಳು ಮತ್ತು ಸಂಪ್ರದಾ	Joy
26	ಪುರಾತನ ದೇವಾಲಯಗಳ ಸಮೀಪದಲ್ಲಿ ಸಾಮಾನ್ಯವಾಗಿ ಕಂಡುಬರುವ ನೈಸರ್ಗಿಕ ಪರಿಸರದ ಸೌಂದರ್ಯವು ನನಗೆ ಉತ್ತಾ	Joy
27	ಹಿಂದೆ ಗುಡುಗು ಸಿಡಿಲಿನ ಸಮಯದಲ್ಲಿ ಆಘಾತಕಾರಿ ಘಟನೆಗಳನ್ನು ಅನುಭವಿಸಿದ ವ್ಯಕ್ತಿಗಳಲ್ಲಿ ಗುಡುಗಿನ ಭಯವು ಹೆಚ್ಚಾ	Fear
28	ಶಿಕ್ಷಣವನ್ನು ಎಲ್ಲಾ ವ್ಯಕ್ತಿಗಳಿಗೆ ಮೂಲಭೂತ ಮಾನವ ಹಕ್ಕು ಎಂದು ಪರಿಗಣಿಸುವ ಬದಲು ಒಂದು ಸವಲತ್ತು ಎಂದು ಪ	Anger
29	ನನ್ನ ನಂಬಿಕೆಗಳನ್ನು ಇತರರು ತಪ್ಪಾಗಿ ಅರ್ಥೈಸಿಕೊಂಡಾಗ ನಾನು ಅಸಮಾಧಾನಗೊಳ್ಳುತ್ತೇನೆ	Sad
30	ಸಾರ್ವಜನಿಕ ಹೋರಾಟದಿಂದ ಉಂಟಾದ ಅವ್ಯವಸ್ಥೆ ಮತ್ತು ಗೊಂದಲದಿಂದ ನಾನು ಆಕ್ರೋಶಗೊಂಡಿದ್ದೇನೆ	Anger
31	ನ್ಯಾಯಯುತ ವೇತನಗಳ ಪ್ರವೇಶದಲ್ಲಿ ಅಸಮಾನತೆಗಳನ್ನು ಉಂಟುಮಾಡಿದಾಗ ಅದು ನನ್ನನ್ನು ಸಿಟ್ಟುಗೊಳಿಸುತ್ತದೆ	Anger
32	ಮುಜುಗರಕ್ಕೊಳಗಾಗುವ ಭಯವು ಆತ್ಮವಿಶ್ವಾಸವನ್ನು ಕಳೆದುಕೊಳ್ಳಲು ಕಾರಣವಾಗಬಹುದು	Fear
33	ಚಂಡಮಾರುತದ ಭಯವು ನಿಯಂತ್ರಣವನ್ನು ಕಳೆದುಕೊಳ್ಳುವ ಭಯಕ್ಕೆ ಸಂಬಂಧಿಸಿರಬಹುದು, ಏಕೆಂದರೆ ನೈಸರ್ಗಿಕ ವಿಪ	Fear
34	ನನ್ನ ಸಂಗಾತಿಯೊಂದಿಗೆ ರೊಮ್ಯಾಂಟಿಕ್ ವಾರ್ಷಿಕೋತ್ಸವವನ್ನು ಆಚರಿಸಿದ ಆನಂದವು ವರ್ಣನಾತೀತವಾಗಿದೆ	Joy
35	ನನಗೆ ಯಾವುದೇ ಸ್ನೇಹಿತರು ಅಥವಾ ಕುಟುಂಬ ಉಳಿದಿಲ್ಲ, ಅವರೆಲ್ಲರೂ ಹಸಿವಿನಿಂದ ಸಾವನ್ನಪ್ಪಿದ್ದಾರೆ	Sad
36	ಕೆಲಸದ ಒತ್ತಡವು ಅಸಮಂಜಸ ಮಟ್ಟಕ್ಕೆ ಹೆಚ್ಚಾದಾಗ ನಾನು ಆವೇಶಗೊಳ್ಳುತ್ತೇನೆ, ಇದು ನನ್ನ ಮಾನಸಿಕ ಮತ್ತು ದೈಹಿಕ ಆ	Anger
37	ಯುದ್ಧದ ಸಮಯದಲ್ಲಿ ನಿಮ್ಮ ಕುಟುಂಬವನ್ನು ಪೂರೈಸಲು ಸಾಧ್ಯವಾಗದ ಭಯವು ಕರುಳು ಹಿಂಡುತ್ತದೆ	Fear
38	ರಸ್ತೆಯ ಪ್ರವೇಶದಲ್ಲಿ ಸಂಚಾರಿಗಳು ಹೆದರಿಸುತ್ತಾರೆ	Fear
39	ಉತ್ಸಾಹಭರಿತ ಪ್ರೇಕ್ಷಕರನ್ನು ಹೊಂದಿರುವ ಚಿತ್ರಮಂದಿರದಲ್ಲಿ ಚಲನಚಿತ್ರವನ್ನು ಅನುಭವಿಸುವ ಉತ್ಸಾಹ ಮತ್ತು ಖುಷಿ	Joy

Fig. 6.1 Data collection

6.3 Before Tokenization

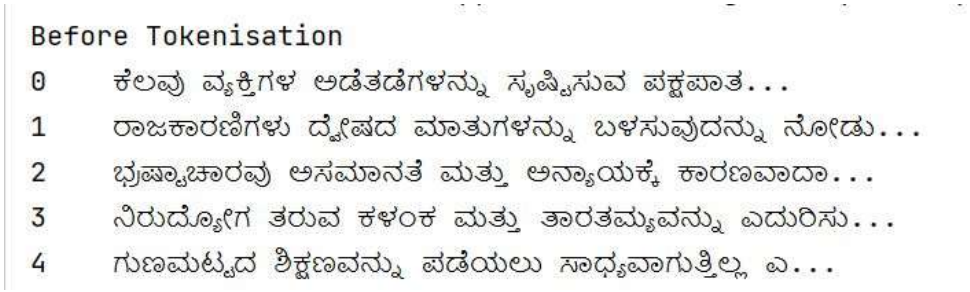

Fig. 6.2 Before Tokenization

6.4 After Tokenization

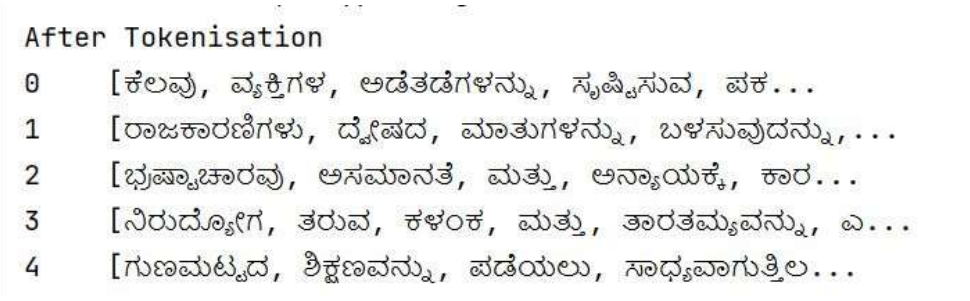

Fig. 6.3 After Tokenization

Figure 6.2 provides a sample of phrases without tokenization. Following the tokenization process, phrases are shown in Fig. 6.3. In the process of tokenization, sentences are divided up into various tokens or words. These tokens are used as input in the cleansing of the data process. For example, the sentence "ಕೆಲವು ವ್ಯಕ್ತಿಗಳ ಅಡೆತಡೆಗಳನ್ನು ಸೃಷ್ಟಿಸುವ ಪಕ್ಷಪಾತದ ವರ್ತನೆಗಳನ್ನು ನಾನು ನೋಡಿದಾಗ ನಾನು ಕೋಪಗೊಳ್ಳುತ್ತೇನೆ". Tokens from this phrase are broken down as follows: "ಕೆಲವು", "ವ್ಯಕ್ತಿಗಳ", "ಅಡೆತಡೆಗಳನ್ನು",. etc

6.5 Before Cleaning and Stopword Removal

```
Before Cleaning and Stopword Removal
0    ಕೆಲವು ವ್ಯಕ್ತಿಗಳ ಅಡೆತಡೆಗಳನ್ನು ಸೃಷ್ಟಿಸುವ ಪಕ್ಷಪಾತ...
1    ರಾಜಕಾರಣಿಗಳು ದ್ವೇಷದ ಮಾತುಗಳನ್ನು ಬಳಸುವುದನ್ನು ನೋಡು...
2    ಭ್ರಷ್ಟಾಚಾರವು ಅಸಮಾನತೆ ಮತ್ತು ಅನ್ಯಾಯಕ್ಕೆ ಕಾರಣವಾದಾ...
3    ನಿರುದ್ಯೋಗ ತರುವ ಕಳಂಕ ಮತ್ತು ತಾರತಮ್ಯವನ್ನು ಎದುರಿಸು...
4    ಗುಣಮಟ್ಟದ ಶಿಕ್ಷಣವನ್ನು ಪಡೆಯಲು ಸಾಧ್ಯವಾಗುತ್ತಿಲ್ಲ ಎ...
```

Fig. 6.4 Before Cleaning and stopword removal

6.6 After Cleaning and Stopword Removal

```
After Cleaning and Stopword Removal
0    ವ್ಯಕ್ತಿಗಳ ಅಡೆತಡೆಗಳನ್ನು ಸೃಷ್ಟಿಸುವ ಪಕ್ಷಪಾತದ ವರ್ತ...
1    ರಾಜಕಾರಣಿಗಳು ದ್ವೇಷದ ಮಾತುಗಳನ್ನು ಬಳಸುವುದನ್ನು ನೋಡು...
2    ಭ್ರಷ್ಟಾಚಾರವು ಅಸಮಾನತೆ ಅನ್ಯಾಯಕ್ಕೆ ಕಾರಣವಾದಾಗ ದುರ್...
3    ನಿರುದ್ಯೋಗ ತರುವ ಕಳಂಕ ತಾರತಮ್ಯವನ್ನು ಎದುರಿಸುತ್ತಿರು...
4    ಗುಣಮಟ್ಟದ ಶಿಕ್ಷಣವನ್ನು ಪಡೆಯಲು ಸಾಧ್ಯವಾಗುತ್ತಿಲ್ಲ ದ...
```

Fig. 6.5 After cleaning and stopword removal

A sample of phrases before cleaning is shown in Fig. 6.4. A sample using the data cleaning technique is shown in Fig. 6.5. In the data cleaning process, undesirable data is removed, such as punctuation that isn't useful for sentiment analysis. Punctuation markers like the comma, full stop, and dollar sign are deleted in the example above.

6.7 Before Stemming

```
Before Stemming
0    ವ್ಯಕ್ತಿಗಳ ಅಡೆತಡೆಗಳನ್ನು ಸೃಷ್ಟಿಸುವ ಪಕ್ಷಪಾತದ ವರ್ತ...
1    ರಾಜಕಾರಣಿಗಳು ದ್ವೇಷದ ಮಾತುಗಳನ್ನು ಬಳಸುವುದನ್ನು ನೋಡು...
2    ಭ್ರಷ್ಟಾಚಾರವು ಅಸಮಾನತೆ ಅನ್ಯಾಯಕ್ಕೆ ಕಾರಣವಾದಾಗ ದುರ್...
3    ನಿರುದ್ಯೋಗ ತರುವ ಕಳಂಕ ತಾರತಮ್ಯವನ್ನು ಎದುರಿಸುತ್ತಿರು...
4    ಗುಣಮಟ್ಟದ ಶಿಕ್ಷಣವನ್ನು ಪಡೆಯಲು ಸಾಧ್ಯವಾಗುತ್ತಿಲ್ಲ ದ...
```

Fig. 6.6 Before Stemming

6.8 After Stemming

```
After Stemming
0    ವ್ಯಕ್ತಿ ಅಡೆತಡೆ ಸೃಷ್ಟಿಸುವ ಪಕ್ಷಪಾತದ ವರ್ತನೆ ನೋಡಿ ...
1    ರಾಜಕಾರಣಿ ದ್ವೇಷದ ಮಾತು ಬಳಸುವು ನೋಡು ನೋವಿನ ಅನುಭವವಾ...
2    ಭ್ರಷ್ಟಾಚಾರವು ಅಸಮಾನತೆ ಅನ್ಯಾಯ ಕಾರಣವಾ ದುರ್ಬಲ ಜನಸಂ...
3         ನಿರುದ್ಯೋಗ ತರುವ ಕಳಂಕ ತಾರತಮ್ಯ ಎದುರಿಸು ವಿಷಾದಿಸು
4    ಗುಣಮಟ್ಟದ ಶಿಕ್ಷಣ ಪಡೆಯಲು ಸಾಧ್ಯವಾಗು ದುಃಖವು ದೀರ್ಘಕ...
```

Fig. 6.7 After Stemming

above Fig. 6.6 and Fig. 6.7. In essence, stemming is taking a word's suffix off to get to the core of the word. First sentence in the example above the word “ವ್ಯಕ್ತಿಗಳ” is reduced to “ವ್ಯಕ್ತಿ”.

6.9 Final Output

Fig. 6.8,6.9,6.10,6.11 illustrates a custom input example 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.

```
Enter A sentence
ವೈಭವ್ ನ ನಡೆಗಳು ಅಥವಾ ವರ್ತನೆಯಿಂದ ನಾನು ಸಿಟ್ಟಾಗಿದ್ದೇನೆ ಮತ್ತು ಕಿರಿಕಿರಿಗೊಂಡಿದ್ದೇನೆ
After Cleaning and Stopwords Removal
ವೈಭವ್ ನ ನಡೆಗಳು ವರ್ತನೆಯಿಂದ ಸಿಟ್ಟಾಗಿದ್ದೇನೆ ಕಿರಿಕಿರಿಗೊಂಡಿದ್ದೇನೆ

After Stemming
ವೈಭವ್ ನ ನಡೆ ವರ್ತನೆ ಸಿಟ್ಟಾಗಿ ಕಿರಿಕಿರಿಗೊಂಡಿ

Output of each Algorithms
['Anger', 'Anger', 'Anger', 'Joy', 'Anger', 'Anger']
The Sentence Is classified as a  Anger Sentence
```

Fig. 6.8 Final output of an anger sentence

```
Enter A sentence
ನಿರಾಕರಣೆಯ ಭೀತಿ ಆತ್ಮವಿಶ್ವಾಸವನ್ನು ಕಾಪಾಡಿಕೊಳ್ಳಲು ಕಷ್ಟವಾಗಬಹುದು
After Cleaning and Stopwords Removal
ನಿರಾಕರಣೆಯ ಭೀತಿ ಆತ್ಮವಿಶ್ವಾಸವನ್ನು ಕಾಪಾಡಿಕೊಳ್ಳಲು ಕಷ್ಟವಾಗಬಹುದು

After Stemming
ನಿರಾಕರಣೆಯ ಭೀತಿ ಆತ್ಮವಿಶ್ವಾಸ ಕಾಪಾಡಿಕೊಳ್ಳಲು ಕಷ್ಟವಾಗಬಹುದು

Output of each Algorithms
['Fear', 'Fear', 'Fear', 'Fear', 'Fear', 'Fear']
The Sentence Is classified as a  Fear Sentence
```

Fig. 6.9 Final output of a fear sentence

```
Enter A sentence
ಯಾವಾಗಲೂ ಸಹಾಯ ಹಸ್ತವನ್ನು ನೀಡಲು ಸಿದ್ಧರಿರುವ ದಯೆ ಮತ್ತು ಪರಿಗಣನೆಯ ನೆರೆಹೊರೆಯವರನ್ನು ಹೊಂದಿರುವುದನ್ನು ನಾನು ಪ್ರಶಂಸಿಸುತ್ತೇನೆ
After Cleaning and Stopwords Removal
ಯಾವಾಗಲೂ ಸಹಾಯ ಹಸ್ತವನ್ನು ನೀಡಲು ಸಿದ್ಧರಿರುವ ದಯೆ ಪರಿಗಣನೆಯ ನೆರೆಹೊರೆಯವರನ್ನು ಹೊಂದಿರುವುದನ್ನು ಪ್ರಶಂಸಿಸುತ್ತೇನೆ

After Stemming
ಯಾವಾಗಲೂ ಸಹಾಯ ಹಸ್ತ ನೀಡು ಸಿದ್ಧರಿ ದಯೆ ಪರಿಗಣನೆಯ ನೆರೆಹೊರೆ ಹೊಂದಿರುವು ಪ್ರಶಂಸಿಸು

Output of each Algorithms
['Joy', 'Joy', 'Joy', 'Joy', 'Joy', 'Joy']
The Sentence Is classified as a  Joy Sentence
```

Fig. 6.10 Final output of a joy sentence

```
Enter A sentence
ವೇದಿಕೆಯ ಪ್ರದರ್ಶನದ ಸಮಯದಲ್ಲಿ ನನ್ನ ಸಹ ಕಲಾವಿದರ ನಿರೀಕ್ಷೆಗಳನ್ನು ಪೂರೈಸಲು ವಿಫಲವಾದಾಗ ನಾನು ಅಳಲು ಅನುಭವಿಸುತ್ತೇನೆ
After Cleaning and Stopwords Removal
ವೇದಿಕೆಯ ಪ್ರದರ್ಶನದ ಸಮಯದಲ್ಲಿ ಸಹ ಕಲಾವಿದರ ನಿರೀಕ್ಷೆಗಳನ್ನು ಪೂರೈಸಲು ವಿಫಲವಾದಾಗ ಅಳಲು ಅನುಭವಿಸುತ್ತೇನೆ

After Stemming
ವೇದಿಕೆಯ ಪ್ರದರ್ಶನ ಸಮಯ ಸಹ ಕಲಾವಿದರ ನಿರೀಕ್ಷೆ ಪೂರೈಸು ವಿಫಲವಾ ಅಳಲು ಅನುಭವಿಸು

Output of each Algorithms
['Sad', 'Sad', 'Sad', 'Joy', 'Sad', 'Sad']
The Sentence Is classified as a  Sad Sentence
```

Fig. 6.11 Final output of a sad sentence

6.10 Algorithm Results Comparison

The accuracy comparison of all the methods utilized in our model is shown in figure 6.12 above. With 2000 Kannada sentences, our model was trained. Our data has been divided into training and testing. 0.2 of the total data set was testing data.

We concluded from the comparison that Linear SVC outperformed all other classifiers. The accuracy rating was 87.25%. SGD classifier accuracy was 85.25%, while that of Logistic Regression was 84.25%. The accuracy of the Random Forest classifier was 85.75%. The accuracy for multinomial naive bayes was 85.50%. K Neighbors Classifier did poorly since it had the lowest accuracy 74.50%.

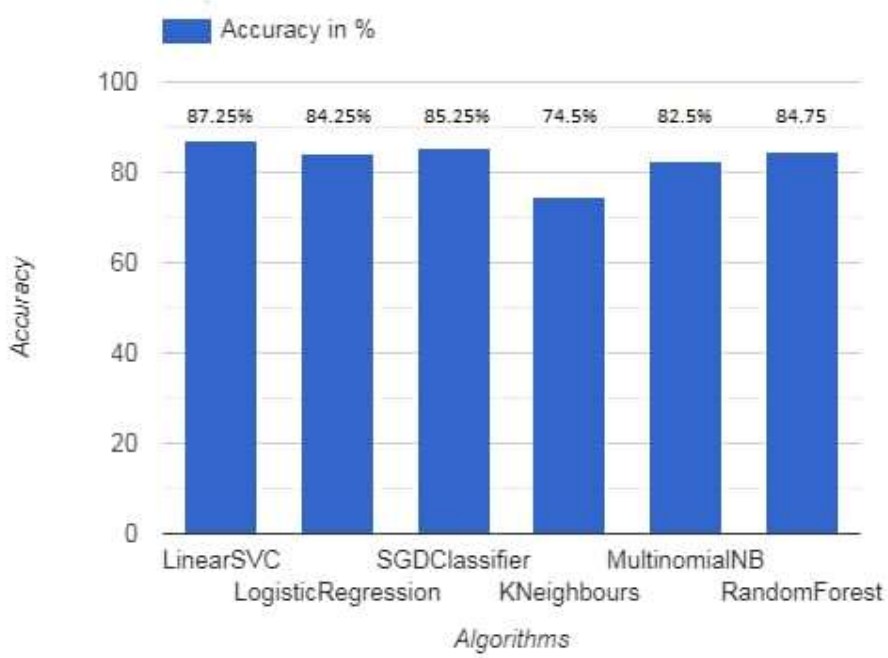

Fig. 6.12 Results comparison

Table. 6.1 Classification Report of Classifiers

Classifier	Accuracy
Linear Support Vector Classifier	Linear SVC precision recall f1-score support Anger 0.89 0.81 0.85 112 Fear 0.95 0.87 0.91 86 Joy 0.86 0.95 0.90 110 Sad 0.77 0.82 0.79 92 accuracy 0.86 400 macro avg 0.87 0.86 0.86 400 weighted avg 0.87 0.86 0.86 400
Stochastic Gradient Descent Classifier	SGDClassifier precision recall f1-score support Anger 0.83 0.79 0.81 112 Fear 0.94 0.86 0.90 86 Joy 0.87 0.92 0.89 110 Sad 0.77 0.82 0.79 92 accuracy 0.85 400 macro avg 0.85 0.85 0.85 400 weighted avg 0.85 0.85 0.85 400
Logistic Regression Classifier	LogisticRegression precision recall f1-score support Anger 0.85 0.79 0.82 112 Fear 0.94 0.76 0.84 86 Joy 0.80 0.94 0.87 110 Sad 0.75 0.82 0.78 92 accuracy 0.83 400 macro avg 0.84 0.82 0.83 400 weighted avg 0.84 0.83 0.83 400

Random Forest Classifier

RandomForestClassifier

	precision	recall	f1-score	support
Anger	0.89	0.78	0.83	112
Fear	0.95	0.85	0.90	86
Joy	0.81	0.92	0.86	110
Sad	0.76	0.83	0.79	92
accuracy			0.84	400
macro avg	0.85	0.84	0.84	400
weighted avg	0.85	0.84	0.84	400

Multinomial Naïve Bayes Classifier

MultinomialNB

	precision	recall	f1-score	support
Anger	0.87	0.78	0.82	112
Fear	0.95	0.71	0.81	86
Joy	0.80	0.95	0.87	110
Sad	0.72	0.82	0.77	92
accuracy			0.82	400
macro avg	0.83	0.81	0.82	400
weighted avg	0.83	0.82	0.82	400

K-Neighbors Classifier

KNeighborsClassifier

	precision	recall	f1-score	support
Anger	0.86	0.64	0.73	112
Fear	0.96	0.53	0.69	86
Joy	0.70	0.87	0.78	110
Sad	0.56	0.80	0.66	92
accuracy			0.72	400
macro avg	0.77	0.71	0.72	400
weighted avg	0.77	0.72	0.72	400

All of the classifiers' classification reports are displayed in Table 6.1. We may assess the accuracy of classification algorithm predictions using the Classification report. It forecasts how many predictions will be right and wrong. Precision, recall, and f1-score are the three primary classification metrics displayed in the classification report.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

In this study, a classification utilizing machine learning is proposed to Sentiment Analysis of Indian Regional Language using Kannada NLP. Internet usage has significantly increased in recent years. From shopping to schooling, everything has shifted online. 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 Kannada language has relatively little sentiment analysis, though. We have made an effort to provide a model that is effective for categorizing sentences in kannada using several classification techniques.

As part of our project, we gathered 2000 Kannada sentences and manually classified them as joy, angry, fear, or sad sentences. We then used these sentences 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, we employed the TF-IDF approach. We've employed a 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.

7.2 Future Scope

The future scope of the project involves several advancements to enhance the sentiment analysis system. Firstly, considering sentences with emoticons would improve the system's ability to interpret emotions expressed through visual cues. Expanding the range of recognized emotions to include surprise, disgust, nervousness, and confusion would enable a more nuanced

understanding of sentiment. Handling code-mixed language sentences would allow the system to accurately analyze sentiment in multilingual contexts. Lastly, incorporating the capability to identify and classify sentences with multiple emotions would provide richer insights into complex emotional states. These future developments would enhance the accuracy and versatility of the sentiment analysis system, catering to a wider range of linguistic expressions and emotional nuances.

REFERENCES

[1] Pandian, A. Pasumpon. "Performance evaluation and comparison using deep learning techniques in sentiment analysis." Journal of Soft Computing Paradigm (JSCP) 3.02 (2021): 123-134.

[2] Joshi, Prasad A., and Varsha M. Pathak. "Sentiment Analysis on Code-mixed Dravidian Languages, A Non-linguistic Approach." (2021).

[3] Rakshitha, Kakuthota, et al. "Sentimental analysis of Indian regional languages on social media." *Global Transitions Proceedings* 2.2 (2021): 414-420.

[4] Basiri, Mohammad Ehsan, et al. "A novel fusion-based deep learning model for sentiment analysis of COVID-19 tweets." Knowledge-Based Systems 228 (2021): 107242.

[5] Bera, Abhijit, Mrinal Kanti Ghose, and Dibyendu Kumar Pal. "Sentiment analysis of multilingual tweets based on Natural Language Processing (NLP)." International Journal of System Dynamics Applications (IJSDA) 10.4 (2021): 1-12.

[6] Ranjitha, P., and K. N. Bhanu. "Improved sentiment analysis for dravidian language-kannada using dicision tree algorithm with efficient data dictionary." IOP Conference Series: Materials Science and Engineering. Vol. 1123. No. 1. IOP Publishing, 2021. (2021).

[7] Madan, Anjum, and Udayan Ghose. "Sentiment Analysis for Twitter Data in the Hindi Language." 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2021.

[8] Kulkarni, Atharva, et al. "L3cubemahasent: A marathi tweet-based sentiment analysis dataset." arXiv preprint arXiv:2103.11408 (2021).

[9] Rajani Shree, M., and B. R. Shambhavi. "POS tagger model for Kannada text with CRF++ and deep learning approaches." Journal of Discrete Mathematical Sciences and Cryptography 23.2 (2020): 485-493.

[10] Sharma, Ankita, and Udayan Ghose. "Sentimental analysis of twitter data with respect to general elections in India." Procedia Computer Science 173 (2020): 325-334.

[11] Shalini, K., et al. "Sentiment analysis for code-mixed Indian social media text with distributed representation." 2018 International conference on advances in computing, communications and informatics (ICACCI). IEEE, 2018.

[12] Nasr, Mona Mohamed, Essam Mohamed Shaaban, and Ahmed Mostafa Hafez. "Building sentiment analysis model using graphlab." International Journal of Scientific and Engineering Research 8 (2017): 1155-1160.

[13] Naidu, Reddy, et al. "Sentiment analysis using telugu sentiwordnet." 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). IEEE, 2017.

[14] Impana, P., and Jagadish S. Kallimani. "Cross-lingual sentiment analysis for Indian regional languages." 2017 International conference on electrical, electronics, communication, computer, and optimization techniques (ICEECCOT). IEEE, 2017.

[15] Rohini, V., Merin Thomas, and C. A. Latha. "Domain based sentiment analysis in regional Language-Kannada using machine learning algorithm." 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT). IEEE, 2016.

[16] Phani, Shanta, Shibamouli Lahiri, and Arindam Biswas. "Sentiment analysis of tweets in three Indian languages." Proceedings of the 6th workshop on south and southeast asian natural language processing (WSSANLP2016). 2016.

[17] Hegde, Yashaswini, and S. K. Padma. "Sentiment analysis for Kannada using mobile product reviews: a case study." 2015 IEEE International Advance Computing Conference (IACC). IEEE, 2015.

Vineet Girgaonkar

ORIGINALITY REPORT

30%	19%	16%	17%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.javatpoint.com Internet Source	3%
2	"Sentimental Analysis and Deep Learning", Springer Science and Business Media LLC, 2022 Publication	3%
3	www.learndatasci.com Internet Source	2%
4	medium.com Internet Source	1%
5	prutor.ai Internet Source	1%
6	Submitted to Asian Institute of Management Student Paper	1%
7	Submitted to Liverpool John Moores University Student Paper	1%
8	Submitted to SSN COLLEGE OF ENGINEERING, Kalavakkam Student Paper	1%