TEXT EMOTION CLASSFICATION AND PREDICTION

#### PROJECT REPORT

*Submitted by*

# DESDEMONA R (7376221CS131) DEVASRI S (7376221CS132) DHAARANI M (7376221CS133) DHARSHINI N (7376221EC146)

*In partial fulfilment for the award of the degree*

# BACHELOR OF ENGINEERING

**in**

COMPUTER SCIENCE AND ENGINEERING



#### BANNARIAMMANINSTITUTE OFTECHNOLOGY

**(An Autonomous Institution Affiliated to Anna University, Chennai) SATHYAMANGALAM - 638401**

# ANNAUNIVERSITY: CHENNAI 600025

#### NOVEMBER 2024

**BONAFIDE CERTIFICATE**

Certified that this project report **“Text Emotion Classification and Prediction”** is the Bonafide work of **"DESDEMONA R (221CS131), DEVASRI S (221CS132) DHAARANI M (221CS133) , DHARSHINI N (221EC146)"** who carried out the

project work under my supervision.

**Dr Sasikala D Dr. Swathypriyadharsini P**

**HEAD OF THE DEPARTMENT ASSISTANT PROFESSOR**

Department of Computer Science and Department of Computer Science and Engineering Engineering

Bannari Amman Institute of Technology Bannari Amman Institute of Technology

**Submitted for Project Viva Voice examination held on………………**

**Internal Examiner 1 Internal Examiner 2**

**DECLARATION**

We affirm that the project work titled **“TEXT EMOTION CLASSIFICATION AND PREDICTION”** being submitted in partial fulfillment for the award of the degree of Bachelor of Technology in Computer science is the record of original work done by us under the guidance of Dr. Swathypriyadharsini, Assistant Professor, Department of Computer Science and Engineering. It has not formed a part of any other project work(s) submitted for the award of any degree or diploma, either in thisor any other University.

**DESDEMONA R DEVASRI S**

**(221CS131) (221CS132)**

**DHAARANI M DHARSHINI N**

**(221CS133) (221EC146)**

I certify that the declaration made above by the candidates is true.

### Dr. SWATHYPRIYADHARSINI

###### ACKNOWLED GEMENT

We would like to enunciate heartfelt thanks to our esteemed Chairman **Dr. S.V. Balasubramaniam**, Trustee **Dr. M. P. Vijayakumar**, and the respected Principal **Dr.C. Palanisamy** for providing excellent facilities and support during the course of study in this institute.

We are grateful to **Dr Sasikala D**, **Head of the Department, Department of Computer Science and Engineering** for her valuable suggestions to carry out the project work successfully.

We wish to express our sincere thanks to Faculty guide **Dr. Swathypriyadharsini P, Assistant Professor**, **Department of Computer Science and Engineering,** for her constructive ideas, inspirations, encouragement, excellent guidance, and much needed technical support extended to complete our project work.

We would like to thank our friends, faculty and non-teaching staff who have directly and indirectly contributed to the success of this project.

**DESDEMONA R (221CS131) DEVASRI S (221CS132) DHAARANI M (221CS133) DHARSHINI N (221EC146)**

i

#### ABSTRACT

In order to handle the details of language and emotional expression, this study offers a thorough method for categorizing emotions in text using machine learning and deep learning approaches. The dataset consisting of text labelled with various emotions, undergoes extensive Preprocessing, Feature Extraction, Model Development, Evaluation and Deployment. The cleaned data is transformed using Count Vectorizer and TF-IDF Vectorizer to convert text into numerical features. Multiple classifiers are trained, including Logistic Regression (LR), Support Vector Classifier (SVC), Bagging Classifier, Extra Trees Classifier, and Light. Each model learns distinct patterns from the data, contributing to accurate emotion prediction. To enhance the overall performance, a Voting Classifier is used to combine the outputs of these models through soft voting, where the predicted probabilities from each classifier are averaged. The ensemble approach significantly boosts classification accuracy, ensuring robust and consistent results across different emotional categories. Evaluation metrics such as accuracy, classification reports, and confusion matrices demonstrate the effectiveness of the model. This solution is well-suited for applications in sentiment analysis, customer feedback interpretation, and real-time emotion detection in communication platforms.

Overall, this pipeline showcases a comprehensive approach to text emotion classification, blending traditional linear models with advanced ensemble techniques to handle the complex, high-dimensional nature of textual data. This methodology is valuable for real- world applications where nuanced emotion detection is essential for user experience enhancement and decision-making processes. The results demonstrate that the model achieves an accuracy of 0.64 (64%) with a precision of 0.66 and a recall of

0.65. The resulting F1 score was 0.64, based on a total support of instances.

***Keywords:*** Emotion Prediction, Ensemble Learning, Text Preprocessing, Sentiment Analysis, Natural Language Processing

**LIST OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **ChapterNo.** | **Title** | **Page.No** |
|  | **ACKNOWLEDGEMENT** | **4** |
|  | **ABSTRACT** | **5** |
|  | **TABLE OF CONTENTS** | **6** |
|  | **LIST OF TABLES** | **8** |
|  | **LIST OF FIGURES** | **9** |
| 1 | Introduction | 10 |
|  | 1.1 Advantages of Text Emotion Classification and Prediction | 10 |
|  | 1.2 Applications of Text Emotion Classification and Prediction | 11 |
| 2 | Literature Survey | 13 |
| 3 | Objectives And Methodology | 16 |
|  | 3.1 Objectives of the proposed work | 16 |
|  | 3.2 Methodology | 16 |

* + 1. [Data Collection and Preprocessing 17](#_TOC_250015)
    2. [Feature Extraction 19](#_TOC_250014)
    3. [Model Implementation 21](#_TOC_250013)
    4. [Real-world applications of text emotion classification 24](#_TOC_250012)
    5. Evaluation metrics 25

1. Proposed work module 28
   1. [Proposed Work 28](#_TOC_250011)
   2. [Methodology of the proposed work 28](#_TOC_250010)
   3. [Requirements and analyzing problems 29](#_TOC_250009)
      1. [Gathering and preparing data 29](#_TOC_250008)
      2. [Extraction of features 30](#_TOC_250007)
2. Result And Discussion 33
   1. [Results 33](#_TOC_250006)
      1. [Performance metrics 33](#_TOC_250005)
      2. [Classification Report 34](#_TOC_250004)
      3. [Ensemble model performance 35](#_TOC_250003)
      4. Implication of findings 36
   2. [Discussion 36](#_TOC_250002)
3. Conclusion & Future works 38
   1. [Conclusion 38](#_TOC_250001)
   2. [Suggestions for future works 39](#_TOC_250000)
4. References 41
5. Appendices 44
6. Acknowledgement 56

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No** | **Table Name** | **Page.No** |
| 5.1 | Comparison of Approaches | 34 |
| 5.3 | Classification Report | 37 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure Name** | **Page. No** |
| 3.1 | Workflow for sentiment Analysis | 17 |
| 3.2 | Text Preprocessing pipeline | 19 |
| 3.3 | The Process of Text vectorization | 20 |
| 3.4 | Building a robust text classification model | 24 |
| 5.2 | Model Accuracy Comparison | 34 |

# CHAPTER 1 INTRODUCTION

In recent years, there has been significant research into text-based emotion classification and prediction due to its increasing relevance in various fields, including artificial intelligence, natural language processing, and sentiment analysis. Emotion classification involves categorizing emotions expressed in text, such as joy, anger, sadness, and fear, while emotion prediction refers to anticipating future emotional states in text based on learned patterns. This technology is crucial for enhancing human- computer interaction, improving mental health monitoring, and enabling more empathetic responses in automated systems.

**Text Emotion Classification and Prediction** uses a combination of techniques, such as machine learning, deep learning, and natural language processing (NLP), to recognize and interpret emotions. It focuses on analyzing the text to determine the underlying emotional tone, providing valuable insights for a variety of applications. By understanding human emotions from text, systems can become more responsive, intuitive, and engaging.

### Advantages of Text Emotion Classification and Prediction

##### Improved Human-Computer Interaction:

Recognizing and responding to human emotions enhances communication between users and systems, making interactions more engaging and effective.

##### Enhanced Customer Support:

Automated systems, such as chatbots and virtual assistants, can offer empathetic and contextually appropriate responses based on the emotional state of the user.

##### Real-Time Emotional Insights:

It provides businesses with real-time emotional feedback from customers, allowing them to adapt their strategies or responses accordingly.

##### Mental Health Monitoring:

Emotion detection from text can help identify emotional distress or mental health concerns, offering timely intervention.

##### Personalized Experiences:

By understanding a user's emotional state, personalized content, recommendations, or services can be offered to improve user satisfaction.

##### Cost-Efficient Sentiment Analysis:

Automated emotion classification reduces the need for manual analysis, allowing companies to quickly and cost-effectively gauge public sentiment in large datasets

### Applications of Text Emotion Classification and Prediction

##### Customer Service:

Emotion detection can help tailor customer service interactions, enabling support systems to respond empathetically to user concerns, leading to higher satisfaction.

##### Social Media Monitoring:

Brands and businesses use text emotion analysis to track public sentiment across social media platforms, gaining insights into how their products or services are perceived emotionally.

##### Mental Health Applications:

Text emotion classification plays a vital role in identifying signs of emotional distress in patients, allowing for early interventions in mental health treatment and monitoring.

##### Personalized Marketing:

Marketers can tailor campaigns based on the emotional state of the target audience, increasing the relevance and effectiveness of advertisements.

##### Education:

Emotion detection in educational platforms can help identify when students are frustrated or disengaged, allowing for timely interventions to improve learning experiences.

##### Sentiment Analysis in Reviews:

By analyzing the emotional content of user reviews, companies can better

Understand customer satisfaction and areas for improvement in their products or services.

* + 1. Entertainment Industry:

Emotion classification can be used to recommend content based on a user’s emotional preferences, such as suggesting movies or music that match their current mood.

* + 1. Healthcare:

Emotion analysis can assist in understanding patients' emotional well-being, offering a more comprehensive approach to healthcare. Text-based systems can provide emotional insights from medical records, improving patient care.

#### INTRODUCTION:

Emotion classification and prediction from text is a rapidly developing field of study that seeks to understand and detect emotions conveyed in textual data. With the rapid expansion of social media, customer reviews, and other types of online communication, emotion detection has emerged as a vital tool in sentiment analysis, human-computer interaction, and a variety of applications such as mental health monitoring and customer service. This literature study examines current advances in text-based emotion categorization, including approaches, datasets, and problems in the field. The survey also tries to identify research gaps and suggest potential future directions.

#### EXISTING WORKS:

* + - * Traditional Machine Learning Approaches.
      * Deep Learning Approaches
      * Pertained Language Models
      * Multi-modal Emotion Detection

**Literature Survey**

1. Devlin et al. (2019) introduced BERT (Bidirectional Encoder Representations from Transformers), a groundbreaking model for language understanding, which significantly advanced the state of the art in various Natural Language Processing (NLP) tasks. BERT's key innovation lies in its bidirectional training of transformers, allowing it to consider the context from both the left and right sides of a word in a sentence. This pre-training strategy enabled BERT to capture deeper semantic and syntactic relationships, leading to improved performance in tasks like sentiment analysis, question answering, and emotion classification. Their experiments showed that BERT outperformed many existing models in terms of accuracy across multiple datasets. Future research was suggested to focus on model optimization techniques like distillation to reduce these challenges while retaining performance benefits [18].
2. The application of federated learning for speech emotion recognition (SER) was investigated by Li et al. (2020), with an emphasis on enhancing robustness and privacy preservation. Due to the sensitive personal information contained in speech data, traditional SER systems frequently encounter data privacy concerns. The authors used federated learning, which permits decentralized model training without necessitating the uploading of raw data to a central server, to address this. Their method kept high recognition accuracy while greatly improving privacy. According to the study, federated learning may strengthen emotion identification models' resistance to changes in speech patterns and background noise. They did,

however, draw attention to the additional complications that federated learning brings, including the need to ensure model consistency among dispersed devices and reduce communication overhead. The authors recommended more investigation into federated learning algorithm optimization to lower these overheads and preserve performance and robust privacy safeguards [14].

1. Liu et al. (2021) proposed an attention-based LSTM model for speech emotion recognition (SER), with the goal of improving the model's capacity to detect emotional cues in speech. The Long Short-Term Memory (LSTM) network was used to record temporal relationships in speech data, and an attention mechanism was added to prioritize emotion-relevant parts of the input. Their model showed considerable gains in emotion detection accuracy, particularly for complex emotions, by focusing on important speech regions that communicate emotional intensity. The study's findings surpassed traditional LSTM models by improving interpretability and providing greater performance in noisy conditions. However, the authors observed that, while the attention mechanism enhanced accuracy, the model's complexity rose, resulting in higher processing costs. They proposed additional optimization to balance accuracy and real-time processing requirements for actual applications [15].
2. Majumder et al. (2021) investigated sentiment and emotion classification using transformer-based language models, focusing on the advancements brought by transformers like BERT and GPT. The study demonstrated that these models significantly outperformed traditional machine learning approaches in identifying both sentiment and nuanced emotional states from text, due to their deep contextual understanding and ability to capture complex linguistic patterns. By fine-tuning pre- trained transformer models on emotion- labeled datasets, the authors achieved high accuracy across various classification tasks. However, they highlighted several challenges, including the high computational costs and the models’ occasional struggles with domain-specific language or rare emotions. The paper also called for future work on improving model efficiency and exploring better ways to handle mixed or overlapping emotional expressions in text [21].
3. Jha et al. (2022) explored multilingual emotion detection in text using transformer-based models, addressing the challenge of recognizing emotions across languages with limited resources. Their study leveraged pre-trained transformer models like mBERT (Multilingual BERT) and XLM-R to capture emotions from text in multiple languages, especially those with fewer linguistic resources. The models were fine-tuned on emotion-labeled datasets from various languages, achieving strong performance in emotion classification tasks. The authors demonstrated that transformer models could generalize well across different languages by learning language-agnostic features, significantly

improving over traditional methods that required separate models for each language. However, they pointed out challenges in handling low-resource languages with very small datasets and suggested further research into enhancing model performance for these languages. Additionally, they highlighted the need to refine cross-lingual transfer techniques to improve emotion detection in diverse and culturally distinct linguistic environments [22].

1. Ahmed et al. (2023) presented a text-based emotion detection framework that utilizes pre-trained embedding models in conjunction with a hybrid CNN-LSTM network. Their study aimed to leverage the strengths of both Convolutional Neural Networks (CNNs) for capturing local patterns in text and Long Short-Term Memory (LSTM) networks for modeling sequential dependencies. By incorporating pre- trained embeddings such as Word2Vec and GloVe, the authors improved the model's ability to understand semantic relationships and context within the text. The hybrid architecture demonstrated significant enhancements in emotion classification accuracy compared to traditional approaches and even pure CNN or LSTM models. However, the authors noted challenges related to the model’s complexity and the computational resources required for training, particularly with large datasets. They suggested future work to explore model optimization techniques and to investigate the model's performance in real-time applications, as well as its effectiveness across different domains and languages [23].
2. Chaturvedi et al. (2024) investigated zero-shot emotion recognition in text using cross-lingual pre-trained transformers, addressing the challenge of recognizing emotions in texts from languages without sufficient labelled training data. Their study leveraged models like mBERT and XLM-R, which are designed to understand and process multiple languages through cross-lingual representations. By applying these models in a zero-shot setting, the authors demonstrated that the approach could effectively classify emotions in unseen languages using knowledge learned from high-resource languages. The results indicated promising accuracy rates, significantly outperforming traditional methods in scenarios where labelled data was scarce. However, the authors noted that the model's performance varied across different languages and cultural contexts, suggesting a need for further refinement in handling language-specific nuances and idiomatic expressions. They proposed future research to enhance the model's robustness and adapt it for diverse linguistic environments, as well as exploring methods to improve interpretability and explainability in zero-shot emotion detection scenarios [24]

## CHAPTER 3

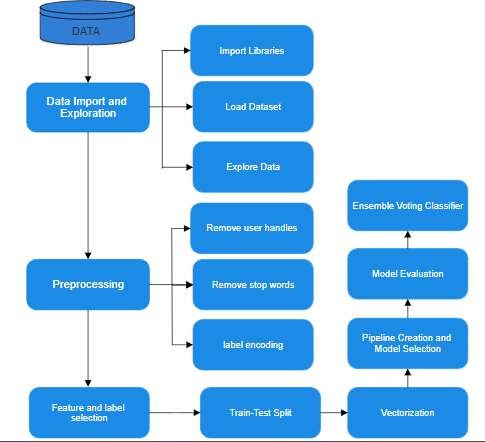
**OBJECTIVES AND METHODOLOGY**

#### OBJECTIVES OF THE PROPOSED WORK

1. Create an Ensemble Model: Using a collection of different machine learning methods, create a strong model for textual emotion classification.
2. Boost Predictive Precision: To increase overall accuracy and consistency in emotion prediction, combine classifiers with soft voting and a voting classifier.
3. Extensive Evaluation: Verify model performance across emotional categories using measures such as confusion matrices, classification reports, and accuracy.
4. Applications in the Real World: Create a scalable solution for real-time emotion recognition, sentiment analysis, and customer feedback interpretation applications.

#### METHODOLOGY

Model training entails training several classifiers (Logistic Regression, SVC, Bagging Classifier, Extra Trees Classifier, and LightGBM) to identify various patterns in the data. Data collection and preprocessing, which involves cleaning and tokenizing labelled text data and feature extraction, which uses CountVectorizer and TF-IDF to transform text into numerical features. The predictions from each classifier are combined using soft voting in a Voting Classifier to increase accuracy. Evaluation metrics like as confusion matrices, classification reports, and accuracy are used to verify the model's efficacy. A reliable system for text-based emotion prediction is ensured by this ensemble technique, which is scalable and perfect for applications in sentiment analysis and emotion detection in communication platforms.



**Figure 3.1** Workflow for Sentiment Analysis

#### DATA COLLECTION AND PREPROCESSING

Dataset Selection: The datasets used in this study comprises a total of 34,792 labeled text entries with a shape of (34792, 4), which indicates four features per entry chosen from publicly available resources that are widely accepted in text emotion detection research. The datasets include annotated textual data with emotional labels, ensuring diverse representation of emotions such as joy, sadness, anger, fear, and surprise.

Text Cleaning and Normalization: A critical preprocessing step in getting raw text data ready for machine learning models is text cleaning and normalization, especially for tasks like emotion categorization. The objective is to convert unstructured material into

a more standardized, clean format, which enhances the models' performance and accuracy.

Process: Text is cleaned to remove irrelevant elements. Remove punctuation ("I'm happy!" → "I’m happy").

Normalize to lowercase ("Happy" → "happy").

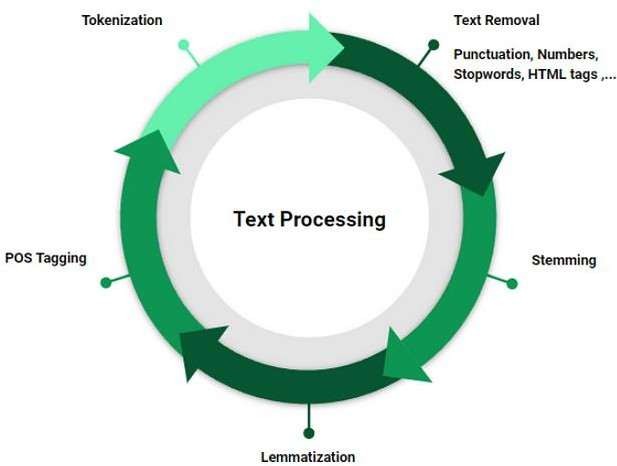
Stop word Removal: Common words that appear frequently but add nothing to the real meaning or emotional content, such as "is," "the," and "and," are eliminated from the dataset by stop word removal.

Example: "I am very happy today" → "happy today"

Stemming and Lemmatization: Lemmatization and stemming are methods for breaking down words into their most basic or root forms. By reducing terms like "running," "runs," and "ran" to just "run," for example, the data becomes less redundant and the model is able to concentrate on the word's essential meaning rather than its variants. This improves the model's capacity to generalize across various variants of the same term while simultaneously lowering the dimensionality of the data.

Example: "running" → "run"; "better" → "good"

Noise Removal: In order to guarantee that only pertinent data is used for model training, noise removal is also essential. This entails removing extraneous characters, punctuation, and special symbols—all of which sometimes hinder the ability to identify moods or emotions. Punctuation, hashtags, and special characters, for instance, might clog data and make it more difficult for the model to identify the words that are most important.



**Figure 3.2** Text Preprocessing Pipeline

#### FEATURE EXTRACTION

To convert textual data into numerical representations suitable for machine learning algorithms, the following feature extraction methods were used,

TF-IDF (Term Frequency-Inverse Document Frequency):

TF-IDF=TF(t)×IDF(t)

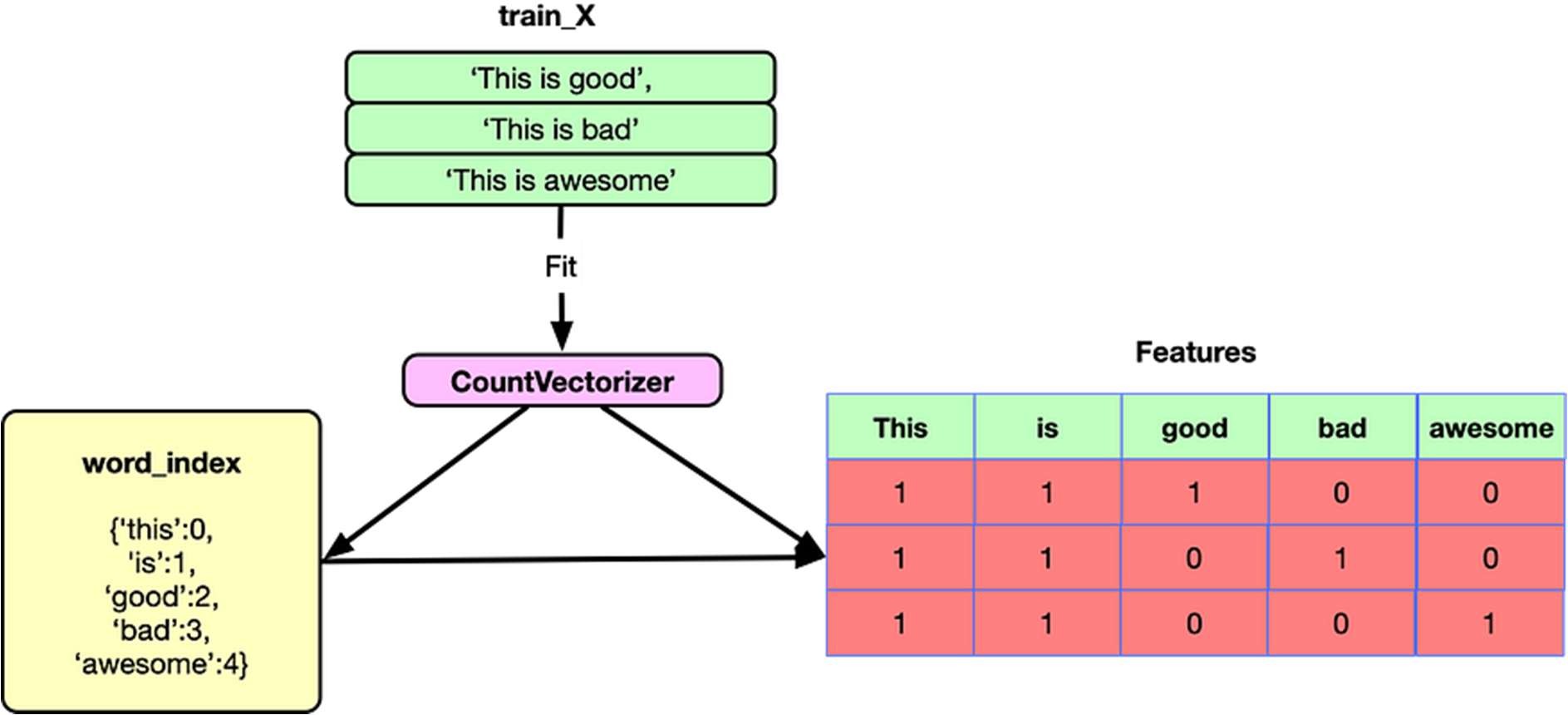
* + - * TF: Frequency of term t in document.
      * IDF: Logarithmic measure of term rarity across all documents.

A statistical method called TF-IDF (Term Frequency-Inverse Document Frequency) is used to assess a word's significance in a document in relation to a corpus of documents. Term Frequency (TF) and Inverse Document Frequency (IDF) are the two main components of the approach. Term frequency quantifies the number of times a term occurs in a given document. A word's phrase frequency increases with the number of times it occurs in a document. However, frequent words like "the" or "is" are not very

Significant because they have large term frequencies throughout multiple papers. Inverse document frequency is useful in this situation.

**CountVectorizer:**

CountVectorizer is a more straightforward method that creates a matrix of token counts from a collection of text. It separates the text into discrete tokens, often words, and determines the frequency with which each token occurs inside a document. CountVectorizer only generates a matrix in which each row denotes a document and each column denotes a word in the corpus, in contrast to TF-IDF, which accounts for word importance. The matrix's values show how frequently each term appears in each document. CountVectorizer is simpler and more computationally efficient than TF-IDF, but it is unable to distinguish between common and rare words. It just records a word's raw frequency without taking into account its significance, which occasionally causes noise in the dataset of popular terms.



**Figure 3.3** The Process of Text Vectorization

#### MODEL IMPLEMENTATION

Logistic Regression (LR):

A linear model called logistic regression determines the probability that a given text belongs to a specific emotion group. It uses a sigmoid function to predict binary outcomes for each label of emotion. It uses techniques like one-vs-rest and softmax to efficiently handle multiclass emotion prediction. Logistic regression functions well and is simple to comprehend when the text's emotional patterns can be linearly divided. Its simplicity and interpretability are among its main advantages since it offers clearly understood coefficients that aid in illuminating the connection between features and predictions. Furthermore, logistic regression is a computationally efficient method that works well with big datasets. Another virtue is its capacity to provide probabilistic outputs, which provide information on prediction certainty and are useful for tasks where determining a forecast's degree of confidence is crucial. However, when there are very complicated relationships among the data or when the emotional patterns are not linearly separable, its performance may suffer.

Support Vector Classifier (SVC) :

Support Vector Classifier (SVC) seeks to determine the optimal hyperplane that optimises the distance between emotion classifications. It performs well with both linearly and non-linearly separable data by utilising kernel functions such as linear kernels or the Radial Basis Function (RBF). SVC can work well with high-dimensional text features and is especially good at recognising subtle changes across emotions. For complicated emotion prediction tasks with text data, particularly when there is a wide range of emotional expressions, its resistance to overfitting makes it a preferred option. Even when the data is not linearly separable in the original space, SVC makes it simpler to locate a separating hyperplane by translating the input data onto a higher-dimensional space. SVC is an effective tool for text classification jobs containing delicate emotional nuances because of its capacity to handle complex, high-dimensional datasets.

Bagging Classifier:

Multiple instances of the same base classifier, each trained on a different random sample of the text input, are combined in bagging classifiers. The accuracy and stability of the model are improved by this procedure. Bagging lowers variance by averaging predictions from many models, which is essential for handling potentially noisy or different emotion data. Bagging is especially effective in situations where the underlying model is high variance, such decision trees, where overfitting may result from the model's sensitivity to even slight changes in the training data. Bagging offers more robust and generalized findings by training several classifiers on distinct subsets of data and aggregating their predictions, which makes it useful for handling complex and varied emotional expressions.

Extra Trees (Extremely Randomized Trees) :

By adding more randomness to the training process, Extra Trees also known as Extremely Randomized Trees, improve decision tree-based models. These classifiers produce a set of decision trees with randomly selected splits at each node. A wider variety of decision trees results from this increased randomness, strengthening the ensemble and lowering the likelihood of overfitting. Since emotions can have complicated and perhaps contradicting expressions, Extra Trees capacity to handle complex patterns in textual data is essential for emotion classification. Extra Trees are well-suited for capturing these intricate links in the data by producing a range of decision trees, providing enhanced performance in high-dimensionality and noisy data circumstances.

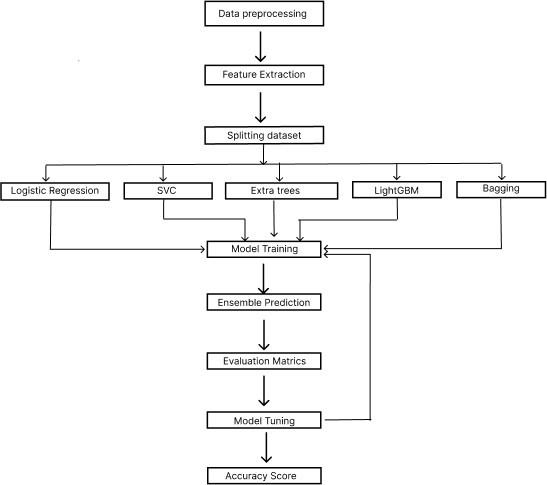
LightGBM:

A gradient boosting system called LightGBM is designed to be quick and effective, especially when handling big datasets. It is renowned for its capacity to build decision trees leaf by leaf, which enables it to effectively identify complex patterns in the data. Text-based applications frequently include sparse data, where many features could be useless for a particular instance. LightGBM performs well in handling this type of data. Because of its efficiency and speed, it is a viable option for real-time emotion classification because it can handle massive amounts of data rapidly while still achieving high predicted accuracy. Furthermore, LightGBM works well with datasets that are

Unbalanced, which is frequently the case in tasks involving emotion prediction where some emotions could be under-represented. It is a well-liked option for real-time applications and when handling complicated, high-dimensional data because of its efficiency and resilience.

Ensemble Methods:

Several models are used in ensemble approaches, such as the Voting Classifier, to enhance overall model performance. Ensemble approaches can outperform individual models in terms of accuracy and generalization by utilizing the advantages of classifiers like Logistic Regression, SVM, LightGBM, Extra Trees, and Bagging Classifiers. This method reduces the drawbacks of any one model by combining the predictions of multiple classifiers to arrive at a final judgement. When the individual models have complimentary qualities, such as distinct approaches to managing bias, variation, and the complexity of emotion data, ensemble methods are especially helpful. This method increases the emotion prediction task's robustness, particularly in real-world situations where emotional expression can be complicated, subtle, and diverse.



**Figure 3.4** Building a Robust Text Classification Model

#### REAL-WORLD APPLICATIONS OF TEXT EMOTION CLASSIFICATION:

Social Media Monitoring and Analysis:

* + - * Monitoring public opinion about a brand or product is known as brand reputation management.
      * Crisis management is the process of spotting possible problems or unfavorable opinions about a business.
      * Market research is the study of consumer preferences and attitudes.

Customer Service :

Sentiment Analysis of Customer Feedback:

* + - * Examining reviews and comments from customers to determine what needs to be improved.
      * Chatbot development is the process of building chatbots that are able to recognise and react to the emotions of users.

Mental Health :

Early Mental Health Issue Detection:

* + - * Keeping an eye out for suicide thoughts, anxiety, or depression in social media posts.
      * Finding people in distress and putting them in touch with the right resources is known as mental health crisis intervention.

Human-Computer Interaction :

* + - * Creating AI systems that are able to recognise and react to human emotions is known as emotionally intelligent AI.
      * The development of more engaging and sympathetic virtual assistants.

Healthcare :

* + - * Patient Sentiment Analysis: Examining evaluations and comments from patients to enhance medical care.
      * Remote Patient Monitoring: Using text messaging to keep an eye on a patient's feelings.

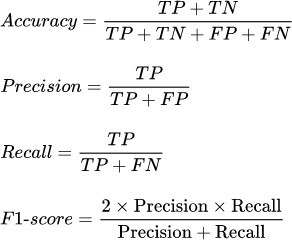
#### EVALUATION METRICS

The performance of each model was assessed using various evaluation metrics to determine the effectiveness of the text emotion classification system:

Accuracy: The overall percentage of correctly predicted emotions.

Precision, Recall, and F1-Score: These metrics were used to evaluate the models' ability to correctly classify emotions, especially for minority classes.

Confusion Matrix: A matrix used to visualize the model's performance in predicting each emotion class and identifying misclassifications. Quantitative analysis was conducted to compare the accuracy and performance of traditional machine learning models and ensemble methods, providing insights into the strengths and limitations of each approach.



**Figure 3.5** Mathematical formulas for evaluation metrics

Comparative Approach:

A comparative analysis of the various models was conducted to assess their effectiveness in text emotion classification. The comparison focused on: Model Efficiency: Evaluation of model training time, processing speed, and scalability when dealing with large datasets. Model Accuracy: Comparison of classification accuracy across different models and datasets, with a focus on identifying the best-performing approach. Model Robustness: Assessing the robustness of models in handling imbalanced datasets and ambiguous emotional expressions in text.

Scalability and Flexibility Evaluation:

The scalability and flexibility of the implemented models were evaluated by testing their ability to handle additional datasets and new emotion categories. This involved expanding the models to integrate additional features and testing their adaptability in different application domains such as customer service, mental health analysis, and social media monitoring. The study examined whether these models could be easily scaled and adapted for real-world use cases involving dynamic and complex text data.

# CHAPTER 4 PROPOSED WORK MODULE

This chapter discusses the proposed Text Emotion Classification &prediction system in a lengthened and time-consuming process, as it elaborates on the design, methodology, and processes involved in this system. The dataset consisting of text labelled with various emotions, undergoes extensive Preprocessing, Feature Extraction, Model Development, Evaluation and Deployment. The cleaned data is transformed using Count Vectorizer and TF-IDF Vectorizer to convert text into numerical features. Multiple classifiers are trained, including Logistic Regression (LR), Support Vector Classifier (SVC), Bagging Classifier, Extra Trees Classifier, and Light.

### Proposed Work

Designing and implementing a system for recognizing and classifying emotions in textual data is the aim of this project. Advanced machine learning and natural language processing (NLP) techniques will be used in this system to attain great accuracy and robustness. Textual emotional comprehension is essential for applications such as interactive AI systems, mental health monitoring, and consumer feedback analysis. Detecting nuanced emotions, processing unclear material, and managing multilingual data are all areas where current approaches frequently struggle. Addressing these constraints is the goal of this study.

### Methodology of the proposed work

To create a scalable and precise model for textual emotion detection by combining multiple models such as (Logistic Regression, SVC, Bagging Classifier, and Extra Trees Classifier. To manage language difficulties such as slang, sarcasm, and different phrase forms.

### Requirements and analyzing problems

Describe the system's scope and the target emotions (such as joy, sadness, anger, fear, and surprise.) that need to be categorized. .Collect pertinent data and examine issues including sarcasm, unclear wording, and multilingual content.

### Gathering and Preparing Data

##### Selection of the Dataset

The datasets utilized in this work include 34,792 18 labeled text entries with a shape of (34792, 4), meaning that each entry has four features selected from publically accessible sources that are commonly used in text emotion detection research. A wide range of emotions, including joy, sadness, rage, fear, and surprise, are represented in the datasets thanks to annotated textual data with emotional labels.

##### Text Preprocessing

Text cleaning and normalization are essential preparatory steps in preparing raw text data for machine learning models, particularly for tasks like emotion categorization. It includes removal of stop words, user handles. The goal is to transform unstructured data into a cleaner, more standardized format that improves the accuracy and performance of the models. Common words that appear frequently but add nothing to the real meaning or emotional content, such as "is," "the," and "and," are eliminated from the dataset by stop word removal and also removed the user handles.

### Extraction of Features

##### TF-IDF (Term Frequency-Inverse Document Frequency)

The significance of a word in a document is evaluated in connection to a corpus of documents using a statistical technique known as TF-IDF (Term Frequency-Inverse Document Frequency). The two primary elements of the strategy are Inverse Document Frequency (IDF) and Term Frequency (TF). The quantity of times a phrase appears in a particular document is measured by term frequency. The more times a word appears in a document, the higher its phrase frequency. However, because to their high term frequencies across several studies, common words like "the" and "is" are not highly important. In this case, inverse document frequency is helpful.

##### CountVectorizer:

A simpler technique called CountVectorizer uses a set of text to produce a matrix of token counts. It divides the text into distinct tokens, usually words, and calculates how frequently each token appears within a document. Unlike TF-IDF, which takes word importance into account, CountVectorizer merely creates a matrix where each row represents a document and each column represents a word in the corpus. The values in the matrix indicate the frequency with which each term occurs in each document. Compared to TF-IDF, CountVectorizer is easier to use and more computationally efficient, but it cannot tell the difference between common and uncommon words. It only captures a word's raw frequency without considering its meaning, which might occasionally introduce noise into the collection of well-known terms.

##### Model Training and Selection

For baseline results, compare traditional machine learning techniques like Support Vector Machines (SVM), Naïve Bayes, and Logistic Regression. Bagging Classifier to achieve better predictive performance by leveraging the strengths of each classifier

Analysis of the Systems

Split the dataset into training sets, validation sets, and tests. Evaluate the model via metrics such as:

Accuracy: 64% accuracy (The overall percentage of correctly predicted emotion). F1-Score, accuracy and Recall: Especially for data sets that are unbalanced.

Confusion Matrix: To examine cases that were incorrectly classified.

##### Deployment and Implementation

Include the learned model in an intuitive user interface. For real-time emotion detection, we used Stream lit.

Test the system in a variety of situations, including with inputs, slang, and different text lengths. The input data would be stored in the application module.

To enhance the model's functionality and performance, collect user input.

##### Case studies and their implications

Utilize the system in real-world circumstances, such analyzing client feedback. Strengthening emotional intelligence in ai interactions. Utilizing text-based analysis for tracking mental wellness. By analyzing the emotional content of user reviews, companies can better understand customer satisfaction and areas for improvement in their products or services.

##### Future Progress and Scalability

Merge text, audio, and video analysis to study multimodal emotion detection. Make the system as latency-efficient as possible for real-time applications. Increase the dataset's diversity by adding new emotional classifications and languages.

A simpler technique called CountVectorizer uses a set of text to produce a matrix of

token counts. It divides the text into distinct tokens, usually words, and calculates how frequently each token appears within a document. Unlike TF-IDF, which takes word importance into account, CountVectorizer merely creates a matrix where each row represents a document and each column represents a word in the corpus. The values in the matrix indicate the frequency with which each term occurs in each document. Compared to TF-IDF, CountVectorizer is easier to use and more computationally efficient, but it cannot tell the difference between common and uncommon words. It only captures a word's raw frequency without considering its meaning, which might occasionally introduce noise into the collection of well-known terms. Multiple classifiers Logistic Regression, Support Vector Classifier (SVC), Bagging, Extra Trees, and LightGBM were explored to address the varying complexities of emotional expression in language, The ensemble Voting Classifier, combining these models.

Logistic Regression demonstrated the highest individual classifier accuracy at 63.13%, followed closely by Extra Trees at 63.00%. SVC and Bagging classifiers achieved accuracies of 61.60% and 60.53%, respectively, while 29 LightGBM showed the lowest accuracy at 58.51%. Notably, the ensemble Voting Classifier, which aggregates the strengths of all five models, outperformed each individual classifier with an accuracy of 64.79%. . This result highlights how well ensemble models handle a variety of patterns in text data that is loaded with emotion, as the Voting Classifier skillfully strikes a compromise between linear techniques like Logistic regression using non-linear techniques such as LightGBM and Extra Trees. This study shows how several algorithms may be used to enhance text emotion identification, which makes it appropriate for a range of real-world sentiment analysis and user behavior comprehension applications. The method supports the larger objective of precise and scalable emotion prediction in textual data by improving classification accuracy and offering a strong, flexible model that can generalize across several emotion categories.

## CHAPTER 5 RESULTS AND DISCUSSION

#### RESULTS:

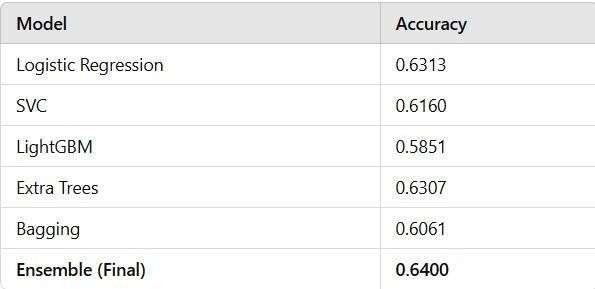
##### Performance metrics:

The accuracy statistic, which shows the percentage of correctly identified instances out of all instances, was used to gauge how effective the aforementioned classifiers were. Significant variations in the classifiers' performances were found by our investigation. With an accuracy of 0.6313, Logistic Regression (LR) was the model that performed the best. This method's efficacy can be attributed to its ability to handle classification problems involving extremely basic relationships in the data, both binary and multi-class.

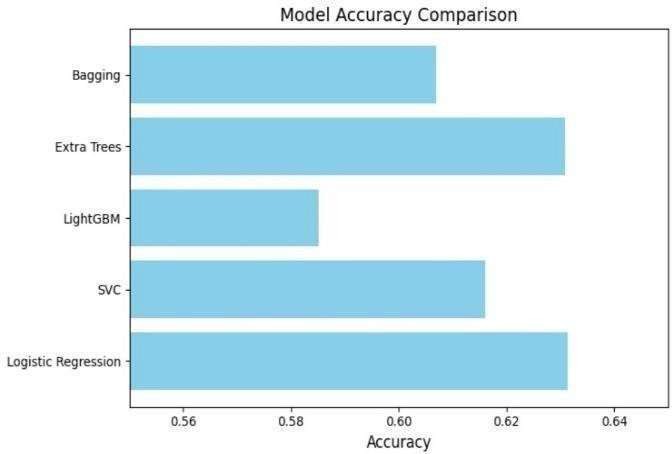
Second place went to Extra Trees, with an impressive accuracy of 0.6300. In order to detect data variances that a single decision tree could miss, this ensemble approach builds many decision trees and combines their predictions. It builds several randomized decision trees and combines their results to produce reliable predictions, which makes it a formidable candidate for high-dimensional data issues with substantial variability.

However, even though LightGBM is a powerful gradient boosting framework, it had the lowest accuracy (0.5851). Despite its effectiveness and scalability design, its ability to generalize from the training data might have been constrained in this specific application. Large datasets with intricate connections are often where LightGBM shines, but in this instance, the data properties might have reduced its typical benefit.

Additionally, the Support Vector Classifier (SVC) and Bagging classifier had respective accuracies of 0.6160 and 0.6053. The ensemble techniques outperformed the SVC in this Instance, despite the fact that it works well for various dataset types and is particularly helpful in high-dimensional spaces. Although SVC's dependence on using hyperplanes to separate data in higher dimensions produced good results, it was unable to outperform ensemble techniques like Extra Trees and Bagging.



**Figure 5.1 Comparison of Approaches**

****

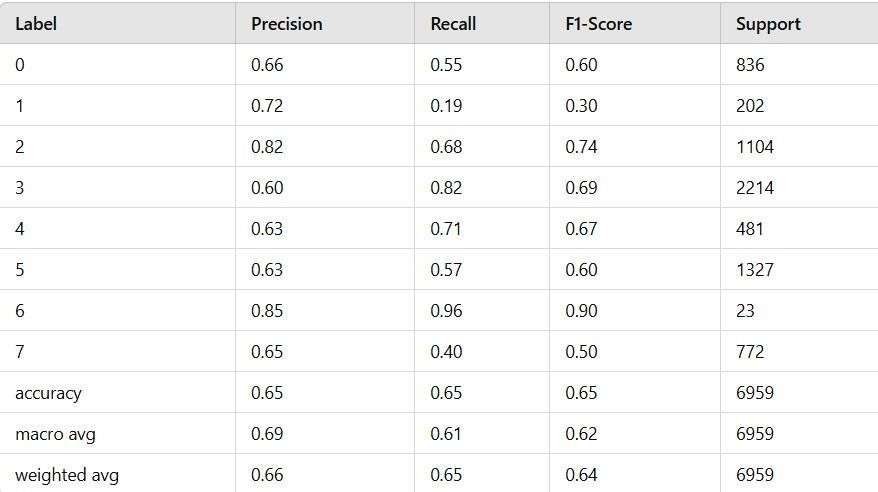
**Figure 5.2 Model Accuracy Comparison**

##### Classification report:

With an overall accuracy of 64.75%, this classification report assesses a voting classifier utilizing the top five models. Class 6 has the best model performance (precision: 0.85, recall: 0.96), but other classes perform differently, especially class 1 (0.19) and class 7 (0.40), which have poorer recall. This difference in recall and precision between classes shows that some classes were easier to distinguish from one another, suggesting that class- specific characteristics or problems with data imbalance require more research.

The model's high precision in class 6 indicates that the characteristics linked to this class were distinct and well-represented.

On the other hand, class 1 and class 7 have lower recall, which suggests that many instances of these classes were incorrectly identified. This could be because they lack enough training examples or have feature spaces that overlap with other classes.



**Figure 5.3 Classification Report**

##### Ensemble model Performance:

The Voting Classifier's performance, which achieved the highest accuracy of 0.6479, is one of the most compelling findings of our investigation. The Voting Classifier combines the predictions of the top five models, taking use of their combined strengths to produce a more accurate forecast. More reliable forecasts result from this aggregate, which lessens the possible biases of the different models. By merging the outputs, either by majority voting or by averaging probabilities, the Voting Classifier lowers the likelihood of overfitting to the training data and enhances overall performance.

Its performance in this application demonstrates the value of integrating more sophisticated ensemble techniques like Extra Trees and Bagging, which capture complicated data linkages, with linear models like Logistic Regression, which are excellent at spotting straightforward li near patterns. With an accuracy of 0.6313, Logistic Regression (LR) was

the model that performed the best. This method's efficacy can be attributed to its ability to handle classification problems involving extremely basic relationships in the data, both binary and multi-class.

This finding emphasizes that, with proper tuning, even the most basic models may perform competitively with more intricate ones.

##### Implications of Findings:

The study's findings highlight how different classifiers' efficacy varies when it comes to textual emotion classification. It is clear that ensemble approaches greatly improve predictive accuracy, even though individual classifiers can still function fairly well. This is especially important for applications like sentiment analysis, customer feedback evaluation, and social media monitoring where precision is critical.

Furthermore, a more thorough comprehension of the subtleties of emotion in text is made possible by ensemble models' capacity to capture both linear and non-linear correlations within the data. The knowledge gathered from this analysis will be crucial for choosing the best models to implement, as businesses depend more and more on automated systems to identify emotions.

#### DISCUSSION:

The results of our investigation demonstrate that ensemble models, specifically the Voting Classifier, can integrate the benefits of many classifiers to significantly increase the accuracy of emotion classification from text data. By integrating linear models like Logistic Regression with non-linear ensemble approaches like Extra Trees and Bagging, the ensemble approach provides a sophisticated interpretation of the data, capturing both straightforward and complex relationships.

The fact that the accuracy of the non-linear Extra Trees model (0.6300) and Logistic Regression (0.6313) was similar is noteworthy and suggests that, for some data distributions, linear approaches might be able to compete with more sophisticated ones. The usefulness of logistic regression is demonstrated by its strong performance in situations where the correlations between features are rather obvious. However, the lower

accuracy of 0.5851 for LightGBM suggests that some gradient boosting models may struggle with this task, possibly. due to their limitations with smaller, more diverse datasets. LightGBM generally performs well in high-dimensional datasets, but this analysis shows that it is limited when dealing with smaller or less structured data.

The Voting Classifier's top accuracy score of 0.6479 demonstrates how ensemble approaches combine various points of view to improve prediction dependability, which is particularly useful for applications like sentiment identification or customer feedback analysis. This finding suggests that an ensemble technique offers greater robustness in classification tasks with high unpredictability, in addition to collecting a greater range of patterns in the data.

These results demonstrate that, even while individual classifiers are helpful, combining models provides a definite performance benefit, particularly for tasks as complicated as textual emotion recognition. To further improve accuracy and flexibility across a variety of datasets, future research may examine the optimization of such ensembles using various model combinations or tuning strategies.

## CHAPTER 6 CONCLUSIONS & FUTURE WORKS

#### CONCLUSION

This research proposes an effective framework for text emotion classification and prediction by leveraging a combination of traditional and advanced machine learning models. Focusing on text data, this study aims to classify emotions accurately by capturing both simple and complex patterns within the text. Multiple classifiers Logistic Regression, Support Vector Classifier (SVC), Bagging, Extra Trees, and LightGBM were explored to address the varying complexities of emotional expression in language. The ensemble Voting Classifier, combining these models, proved particularly effective, highlighting the value of hybrid approaches in emotion classification.

The dataset underwent rigorous preprocessing to ensure high-quality input for model training. Techniques such as removing user handles, stop words, and punctuation, along with lemmatization, were applied to normalize the text data. This preprocessing step was critical in eliminating noise and standardizing words, allowing the model to focus on the linguistic patterns most relevant to emotional context. Feature extraction was performed using both CountVectorizer and TfidfVectorizer, enabling the model to capture key vocabulary terms and phrase patterns associated with specific emotions.

Each classifier’s performance was measured based on accuracy, precision, recall, and F1 score. Logistic Regression demonstrated the highest individual classifier accuracy at 63.13%, followed closely by Extra Trees at 63.00%. SVC and Bagging classifiers achieved accuracies of 61.60% and 60.53%, respectively, while LightGBM showed the lowest accuracy at 58.51%. Notably, the ensemble Voting Classifier, which aggregates the strengths of all five models, outperformed each individual classifier with an accuracy of 64.79%. This finding underscores the effectiveness of ensemble models in handling diverse patterns in emotion-laden text data, as the Voting Classifier adeptly

Balances linear methods like Logistic Regression with non-linear approaches like Extra Trees and LightGBM.

This study demonstrates the potential of combining different algorithms to improve emotion detection in text, making it suitable for various real-world applications in sentiment analysis and user behavior understanding. The approach not only enhances classification accuracy but also provides a robust, adaptable model that can generalize across multiple emotion categories, supporting the broader goal of accurate and scalable emotion prediction in textual data.

#### SUGGESTIONS FOR FUTURE WORKS

The promising results of this study open avenues for several advancements in text emotion classification. One primary area for improvement is the expansion of datasets to enhance the model's generalizability across diverse demographics and contexts. Future research should aim to incorporate larger, more varied datasets that reflect diverse linguistic, cultural, and contextual variations in emotional expression. Collaboration with social media platforms and data providers worldwide will be critical to ensure that the model performs robustly across different population groups, making it adaptable to various real-world applications.

Interpretability remains a significant challenge in emotion classification due to the "black-box" nature of many machine learning models, particularly ensemble and deep learning techniques. Future studies could focus on integrating interpretability tools like SHAP (Shapley Additive explanations) or LIME (Local Interpretable Model- agnostic Explanations). These methods allow visualization of the features contributing to model predictions, enabling researchers and end-users to understand the model’s reasoning process better. This transparency can help foster trust among stakeholders and improve the model's integration into user-facing applications.

With the increasing demand for real-time emotion analysis in mobile applications, optimizing the model for deployment on edge devices is essential. Techniques such as model pruning, quantization, and knowledge distillation could be

Explored to reduce model complexity and memory usage without compromising accuracy. By creating a lightweight model compatible with mobile and IoT devices, emotion classification tools could be used in diverse settings, such as customer support and mental health applications, ensuring accessible emotional insights in real time.

Another potential improvement involves the integration of multi-modal data sources to enrich emotion classification. Combining text data with other data type such as audio cues, facial expressions, and physiological signals could enhance the model's ability to capture nuanced emotional states. This multi-modal approach would allow the model to analyze emotions in a more holistic context, potentially improving accuracy in applications like sentiment analysis, mental health monitoring, and customer interaction analysis.

For smooth integration into existing digital ecosystems, developing application programming interfaces (APIs) for interoperability is essential. These APIs would enable seamless embedding of the emotion classification model into various software applications and platforms, such as Chatbot, social media monitoring tools, and mental health apps, streamlining workflows and making emotion analysis more accessible.

Improving text preprocessing techniques is another area for future exploration. Noise in text data, such as spelling errors, abbreviations, and sarcasm, can affect model accuracy. Future research could focus on advanced preprocessing algorithms, including neural-based spelling correction and context-aware word disambiguation, to improve the quality of input data. These improvements would enhance the model’s robustness and reliability in diverse and noisy text environments.

Lastly, implementing a continuous learning model is crucial for maintaining the model’s relevance as language evolves. Techniques like online learning and federated learning can enable the model to learn from new data sources and adapt to emerging language trends without compromising user privacy. This adaptability will ensure that the model remains accurate and up-to-date, keeping pace with evolving language dynamics and usage patterns, and thus providing a resilient tool for emotion classification.

## REFERENCES

1. Seguí, F.L.; Aguilar, R.A.E.; De Maeztu, G.; García-Altés, A.; Garcia-Cuyàs, F.; Walsh, S.; Castro, M.S.; Vidal-Alaball, J. Teleconsultations between Patients and Healthcare Professionals in Primary Care in Catalonia: The Evaluation of Text Classification Algorithms Using Supervised Machine Learning. *Int. J. Environ. Res. Public Health* **2020**, *17*, 1093.
2. Salur, M.U.; Aydin, I. A Novel Hybrid Deep Learning Model for Sentiment Classification. *IEEE Access* **2020**, *8*, 58080–58093.
3. Srivastava, S.K.; Singh, S.K.; Suri, J.S. A healthcare text classification system and its performance evaluation: A source of better intelligence by characterizing

healthcare text. In *Cognitive Informatics, Computer Modelling, and Cognitive Science*; Elsevier BV: Amsterdam, The Netherlands, 2020; pp. 319–369.

1. M. Hasan, E. Rundensteiner and E. Agu, "DeepEmotex: Classifying Emotion in Text Messages using Deep Transfer Learning," 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 2021, pp. 5143-5152, doi: 10.1109/BigData52589.2021.9671803.
2. Ibrahim, M.A.; Ghani Khan, M.U.; Mehmood, F.; Asim, M.N.; Mahmood,

W. GHS-NET a generic hybridized shallow neural network for multi-label biomedical text classification. *J. Biomed. Inform.* **2021**, *116*, 103699.

1. Chen, J., & Zhang, Y. (2021). Exploring the Role of Deep Learning in Emotion Detection from Text. Journal of Natural Language Engineering, 27(3), 251- 267.
2. V. Sundaram, S. Ahmed, S. A. Muqtadeer and R. Ravinder Reddy, "Emotion Analysis in Text using TF-IDF," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2021, pp. 292- 297, doi: 10.1109/Confluence51648.2021.9377159.
3. M. Xuanyuan, L. Xiao and M. Duan, "Sentiment Classification Algorithm Based on Multi-Modal Social Media Text Information," in IEEE Access, vol. 9, pp. 33410-33418, 2021, doi: 10.1109/ACCESS.2021.306145.
4. Minaee, S.; Kalchbrenner, N.; Cambria, E.; Nikzad, N.; Chenaghlu, M.; Gao,

J. Deep Learning–Based Text Classification: A Comprehensive Review. *Acm Comput. Surv.* **2021**, *54*, 1–40.

1. Ameer, G. Sidorov, H. Gómez-Adorno and R. M. A. Nawab, "Multi-Label Emotion Classification on Code-Mixed Text: Data and Methods," in IEEE Access, vol. 10, pp. 8779-8789, 2022, doi: 10.1109/ACCESS.2022.3143819.
2. R. Kumar, C. M. Sharma, V. M. Chariar, S. Hooda and R. Beri, "Emotion Analysis of News and Social Media Text for Stock Price Prediction using SVMLSTM-GRU Composite Model," 2022 International.
3. Liu C, Zhan Y, Li C, Du B, Wu J, Hu W, Liu T, Tao D (2022) Graph pooling for graph neural networks: Progress, challenges, and opportunities. arXiv preprint.
4. Yang, X., & Li, Y. (2022). Emotion Recognition in Text Using Transfer Learning Approaches. Journal of Information Processing Systems, 18(5), 10221035.
5. Juni, K., & Khyat, S. (2023). Advances in Emotion Detection Using Machine Learning Techniques: A Review. Journal of Artificial Intelligence Research, 68, 19- 41.
6. G. Singh, D. Brahma, P. Rai and A. Modi, "Text-Based Fine-Grained Emotion Prediction," in IEEE Transactions on Affective Computing, vol. 15, no. 2, pp. 405- 416, April-June 2024, doi: 10.1109/TAFFC.2023.3298405.

#### APPENDICES:

###### BILL OF MATERIALS:

Since only coding is utilized, there is no cost involved. Since my idea doesn't have any components, only the service cost which would be the bare minimum is considered. For this reason, I declare it free.

###### CODING:

app.py:

import streamlit as st import altair as alt

import plotly.express as px import pandas as pd import numpy as np

from datetime import datetime import joblib

from track\_utils import create\_page\_visited\_table, add\_page\_visited\_details, view\_all\_page\_visited\_details, add\_prediction\_details, view\_all\_prediction\_details, create\_emotionclf\_table, IST # Import IST from track\_utils

# Load Model

pipe\_lr = joblib.load(open("./models/emotion\_classifier\_pipe\_lr.pkl", "rb")) # Function

def predict\_emotions(docx): results = pipe\_lr.predict([docx]) return results[0]

def get\_prediction\_proba(docx):

results = pipe\_lr.predict\_proba([docx]) return results

emotions\_emoji\_dict = {"anger": "●·

•´`", "disgust": "<.¿",›● "fear": "· `ˆ

´ .ˆ•

", "happy":

"×●^•", "joy": "●v^-

"surprise": "•●˙."} # Main Application def main():

", "neutral": "●•−˙", "sad": "●`-\_-’", "sadness": "’-`\_-●", "shame": "●·−´•`",

st.title("Emotion Classifier App") menu = ["Home", "Monitor", "About"]

choice = st.sidebar.selectbox("Menu", menu) create\_page\_visited\_table() create\_emotionclf\_table()

if choice == "Home":

add\_page\_visited\_details("Home", datetime.now(IST)) st.subheader("Emotion Detection in Text")

with st.form(key='emotion\_clf\_form'): raw\_text = st.text\_area("Type Here")

submit\_text = st.form\_submit\_button(label='Submit') if submit\_text:

col1, col2 = st.columns(2)

prediction = predict\_emotions(raw\_text) probability = get\_prediction\_proba(raw\_text)

add\_prediction\_details(raw\_text,prediction,np.max(probability), datetime.now(IST))

with col1: st.success("Original Text") st.write(raw\_text) st.success("Prediction")

emoji\_icon = emotions\_emoji\_dict[prediction] st.write("{}:{}".format(prediction, emoji\_icon)) st.write("Confidence:{}".format(np.max(probability)))

with col2:

st.success("Prediction Probability")

proba\_df = pd.DataFrame(probability, columns=pipe\_lr.classes\_) proba\_df\_clean = proba\_df.T.reset\_index() proba\_df\_clean.columns = ["emotions", "probability"]

fig = alt.Chart(proba\_df\_clean).mark\_bar().encode(x='emotions', y='probability', color='emotions')

st.altair\_chart(fig, use\_container\_width=True) elif choice == "Monitor":

add\_page\_visited\_details("Monitor", datetime.now(IST)) st.subheader("Monitor App")

with st.expander("Page Metrics"):

page\_visited\_details = pd.DataFrame(view\_all\_page\_visited\_details(), columns=['Page Name', 'Time of Visit'])

st.dataframe(page\_visited\_details)

pg\_count=page\_visited\_details['Page Name'].value\_counts().rename\_axis('Page Name').reset\_index(name='Counts')

c = alt.Chart(pg\_count).mark\_bar().encode(x='Page Name', y='Counts', color='Page Name')

st.altair\_chart(c, use\_container\_width=True)

p = px.pie(pg\_count, values='Counts', names='Page Name') st.plotly\_chart(p, use\_container\_width=True)

with st.expander('Emotion Classifier Metrics'):

df\_emotions = pd.DataFrame(view\_all\_prediction\_details(), columns=['Rawtext', 'Prediction', 'Probability', 'Time\_of\_Visit'])

st.dataframe(df\_emotions)

prediction\_count =

df\_emotions['Prediction'].value\_counts().rename\_axis('Prediction').reset\_index(name= 'Counts')

pc = alt.Chart(prediction\_count).mark\_bar().encode(x='Prediction', y='Counts', color='Prediction')

st.altair\_chart(pc, use\_container\_width=True)

else:

add\_page\_visited\_details("About", datetime.now(IST))

st.write("Welcome to the Emotion Detection in Text App! This application utilizes the power of natural language processing and machine learning to analyze and identify emotions in textual data.")

st.subheader("Our Mission")

st.write("At Emotion Detection in Text, our mission is to provide a user-friendly and efficient tool that helps individuals and organizations understand the emotional content hidden within text. We believe that emotions play a crucial role in communication, and by uncovering these emotions, we can gain valuable insights into the underlying sentiments and attitudes expressed in written text.")

st.subheader("How It Works")

st.write("When you input text into the app, our system processes it and applies advanced natural language processing algorithms to extract meaningful features from the text. These features are then fed into the trained model, which predicts the emotions associated with the input text. The app displays the detected emotions, along with a confidence score, providing you with valuable insights into the emotional content of your text.")

st.subheader("Key Features:")

st.markdown("##### 1. Real-time Emotion Detection")

st.write("Our app offers real-time emotion detection, allowing you to instantly analyze the emotions expressed in any given text. Whether you're analyzing customer feedback, social media posts, or any other form of text, our app provides you with immediate insights into the emotions underlying the text.")

st.markdown("##### 2. Confidence Score")

st.write("Alongside the detected emotions, our app provides a confidence score, indicating the model's certainty in its predictions. This score helps you gauge the reliability of the emotion detection results and make more informed decisions based on the analysis.")

st.markdown("##### 3. User-friendly Interface")

st.write("We've designed our app with simplicity and usability in mind. The intuitive user interface allows you to effortlessly input text, view the results, and

interpret the emotions detected. Whether you're a seasoned data scientist or someone with limited technical expertise, our app is accessible to all.")

st.subheader("Applications") st.markdown("""

The Emotion Detection in Text App has a wide range of applications across various industries and domains. Some common use cases include:

* Social media sentiment analysis
* Customer feedback analysis
* Market research and consumer insights
* Brand monitoring and reputation management
* Content analysis and recommendation systems """)

if \_name\_ == '\_main\_': main()

track\_utils.py:

import sqlite3 import pytz

from datetime import datetime # Load Database Packages

conn = sqlite3.connect('./data/data.db', check\_same\_thread=False) c = conn.cursor()

IST = pytz.timezone('Asia/Kolkata') # Indian Standard Time # Function to create page visited table

def create\_page\_visited\_table():

c.execute('CREATE TABLE IF NOT EXISTS pageTrackTable(pagename TEXT, timeOfvisit TIMESTAMP)')

# Function to add page visited details

def add\_page\_visited\_details(pagename, timeOfvisit=None): if timeOfvisit is None:

timeOfvisit = datetime.now(IST).strftime("%Y-%m-%d %H:%M:%S") else:

timeOfvisit = timeOfvisit.astimezone(IST).strftime("%Y-%m-%d %H:%M:%S")

c.execute('INSERT INTO pageTrackTable(pagename, timeOfvisit) VALUES (?, ?)', (pagename, timeOfvisit))

conn.commit()

# Function to view all page visited details def view\_all\_page\_visited\_details():

c.execute('SELECT \* FROM pageTrackTable') data = c.fetchall()

return data

# Function to create emotion classifier table def create\_emotionclf\_table():

c.execute('CREATE TABLE IF NOT EXISTS emotionclfTable(rawtext TEXT, prediction TEXT, probability NUMBER, timeOfvisit TIMESTAMP)')

# Function to add prediction details

def add\_prediction\_details(rawtext, prediction, probability, timeOfvisit=None):

if timeOfvisit is None:

timeOfvisit = datetime.now(IST).strftime("%Y-%m-%d %H:%M:%S") else:

timeOfvisit = timeOfvisit.astimezone(IST).strftime("%Y-%m-%d %H:%M:%S")

c.execute('INSERT INTO emotionclfTable(rawtext, prediction, probability, timeOfvisit) VALUES (?, ?, ?, ?)', (rawtext, prediction, probability, timeOfvisit))

conn.commit()

# Function to view all prediction details def view\_all\_prediction\_details():

c.execute('SELECT \* FROM emotionclfTable') data = c.fetchall()

return data

Backend code:

import pandas as pd import numpy as np import seaborn as sns

import neattext.functions as nfx

from sklearn.preprocessing import LabelEncoder

from sklearn.feature\_extraction.text import TfidfVectorizer from sklearn.model\_selection import train\_test\_split

from sklearn.pipeline import Pipeline

from sklearn.ensemble import ExtraTreesClassifier from sklearn.linear\_model import LogisticRegression

from sklearn.ensemble import BaggingClassifier from sklearn.svm import SVC

import lightgbm as lgb

from sklearn.ensemble import VotingClassifier

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix import joblib

df = pd.read\_csv("D:\deva\OneDrive\Downloads\Emotion-Detection-in-Text- main\data\emotion\_dataset.csv")

df['Emotion'].value\_counts() sns.countplot(x='Emotion',data=df)

df['Clean\_Text'] = df['Text'].apply(nfx.remove\_userhandles) df['Clean\_Text'] = df['Clean\_Text'].apply(nfx.remove\_stopwords) label\_encoder = LabelEncoder()

y = df['Emotion']

y\_encoded = label\_encoder.fit\_transform(y) Xfeatures = df['Clean\_Text']

ylabels = df['Emotion']

x\_train, x\_test, y\_train, y\_test = train\_test\_split(Xfeatures, y\_encoded, test\_size=0.2, random\_state=42)

tfidf\_vectorizer = TfidfVectorizer(max\_features=5000)

pipe\_lr = Pipeline(steps=[('cv', CountVectorizer()), ('lr', LogisticRegression())])

pipe\_svc = Pipeline(steps=[('cv', CountVectorizer()), ('svc', SVC(kernel='linear', probability=True))])

pipe\_bagging = Pipeline([('tfidf', tfidf\_vectorizer), ('bagging', BaggingClassifier(n\_estimators=50))])

pipe\_lgbm = Pipeline([('tfidf', tfidf\_vectorizer), ('lgbm', lgb.LGBMClassifier())]) pipe\_et = Pipeline([('tfidf', tfidf\_vectorizer), ('et', ExtraTreesClassifier())]) pipe\_lr.fit(x\_train, y\_train)

pipe\_lr.fit(x\_train, y\_train) pipe\_svc.fit(x\_train, y\_train) pipe\_lgbm.fit(x\_train, y\_train) pipe\_et.fit(x\_train, y\_train) pipe\_bagging.fit(x\_train, y\_train) y\_pred\_et = pipe\_et.predict(x\_test) y\_pred\_lgbm = pipe\_lgbm.predict(x\_test)

y\_pred\_bagging = pipe\_bagging.predict(x\_test) svc\_accuracy = pipe\_svc.score(x\_test, y\_test) lr\_accuracy = pipe\_lr.score(x\_test, y\_test) acc\_lgbm = accuracy\_score(y\_test, y\_pred\_lgbm) acc\_et = accuracy\_score(y\_test, y\_pred\_et)

acc\_bagging = accuracy\_score(y\_test, y\_pred\_bagging) print(f"Logistic Regression Accuracy: {lr\_accuracy:.4f}") print(f"SVC Accuracy: {svc\_accuracy:.4f}")

print(f"LightGBM Accuracy: {acc\_lgbm:.4f}") print(f"Extra Trees Accuracy: {acc\_et:.4f}")

print(f"Bagging Accuracy: {acc\_bagging:.4f}") estimators\_5 = [

('extra\_trees', pipe\_et), ('log\_reg', pipe\_lr), ('bagging', pipe\_bagging), ('svc', pipe\_svc),

('lgbm', pipe\_lgbm)

]

voting\_clf = VotingClassifier(estimators=estimators\_5, voting='soft') voting\_clf.fit(x\_train, y\_train)

y\_pred = voting\_clf.predict(x\_test) accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Voting Classifier Accuracy with Top 5 Models: {accuracy:.4f}") print("\nClassification Report:\n", classification\_report(y\_test, y\_pred)) joblib.dump(voting\_clf, 'voting\_classifier\_model.pkl') print("\nConfusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))

from sklearn.metrics import confusion\_matrix import seaborn as sns

import matplotlib.pyplot as plt

cm = confusion\_matrix(y\_test, y\_pred) sns.heatmap(cm, annot=True, fmt="d", cmap="Blues") plt.xlabel("Predicted Label")

plt.ylabel("True Label")

plt.title("Confusion Matrix for Voting Classifier") plt.show()

* + 1. **PUBLICATION CERTIFICATE:**

#### WORK CONTRIBUTION:

###### MEMBER 1: DHARSHINI N

DATA COLLECTION AND PREPROCESSING

The program started by compiling a wide range of labelled textual data that represented different emotional categories in order to create an efficient emotion classification model. Basic emotions including joy, sadness, anger, disgust, fear, happiness, neutral, and shame were all included in these categories. In order to enable the model to learn from subtle linguistic patterns connected to various emotional states, the data gathering procedure made sure that the text samples represented a broad spectrum of emotional expressions. To guarantee that the model could generalize effectively across a range of applications, including sentiment analysis and feedback evaluation, this diversity was crucial.

The data then went through a crucial preparation step to improve its quality and machine learning applicability. To complete necessary cleaning activities, such as eliminating user handles (such as "@username") frequently included in social network data, we used the Neattext package. To further minimize noise in the dataset, stopwords common words like "the," "and," and "is" that do not add to the semantic meaning were removed. The model was able to concentrate on important characteristics pertinent to emotion categorization because this preprocessing phase made sure that only the most instructive words were left.

Lastly, we converted the category emotion labels into numerical values using label encoding. A distinct integer identity was given to each emotion category (such as

"happy," "sad," and "angry") so that the machine learning algorithms could digest these labels quickly. In order to facilitate effective emotion prediction, this encoding step is essential to supervised learning since it transforms qualitative data into a format that the classifiers can comprehend and use.

###### MEMBER 2 : DHAARANI M

FEATURE EXTRACTION

CountVectorizer and TF-IDF (Term Frequency-Inverse Document Frequency) were two feature extraction techniques used to transform textual input into numerical representations appropriate for machine learning algorithms. By ensuring that unstructured language was converted into numerical data, these techniques made it possible for the classifiers to efficiently learn from the input.

TF-IDF (Term Frequency-Inverse Document Frequency):

When it came to allocating variable weights to terms according to their frequency throughout the sample, TF-IDF was especially helpful. It increased the importance of terms that appeared less frequently but were essential for differentiating particular emotions while decreasing the impact of those that were used frequently. This method improved the model's capacity to identify minute changes in sentiment, including the difference between closely related emotions like happy and pleasure.

CountVectorizer:

By transforming the text into a matrix of token counts that represented the raw frequency of words in the dataset, CountVectorizer enhanced TF-IDF. This approach offered a straightforward yet effective means of quantifying textual data, which served as the foundation for training machine learning models. CountVectorizer provided a more comprehensive view of the dataset's linguistic patterns by capturing the general distribution of word occurrences, whereas TF-IDF highlighted the uniqueness of specific terms. When combined, these feature extraction methods gave the machine learning models a strong basis and improved their capacity to correctly categorize emotions by utilizing both word frequency and contextual significance.

###### MEMBER 3 : DEVASRI S

MODEL DEVELOPMENT AND TRAINING

We integrated a number of machine learning classifiers, such as Logistic Regression, Support Vector Classifier (SVC), Bagging, Extra Trees, and LightGBM, to create and train the emotion classification model in order to attain the best accuracy. These classifiers each contributed special capabilities to the challenge. For linearly separable data, the linear model known as logistic regression worked well since it was able to capture straightforward correlations between attributes and target labels. On the other hand, SVC is renowned for its effectiveness in high-dimensional spaces, where it divides classes using hyperplanes. By bootstrapping numerous versions of the dataset and training separate base learners to minimize variance and avoid overfitting, bagging made use of the potential of ensemble learning. While LightGBM, a gradient boosting approach, improved training efficiency and scalability by concentrating on data samples that were more difficult to accurately categorize, Extra Trees, an ensemble of decision trees, identified intricate, non-linear relationships within the data.

The outputs of these classifiers were combined using a Voting Classifier in order to further improve model accuracy through the use of ensemble approaches. Using either majority voting or average the estimated odds, this method combined predictions from several models to generate a final result. The ensemble approach increased overall predictive accuracy and robustness by utilizing the advantages of both non-linear models like Extra Trees and Bagging and linear models like Logistic Regression. Because the combined output softened the biases of individual models, this approach decreased the chance of overfitting. As a result, the ensemble technique made it possible for the model to identify a wider variety of patterns in the data, which increased accuracy and dependability particularly in challenging classification tasks requiring overlapping and subtle emotional expressions.

###### MEMBER 4 : DESDEMONA R

EVALUATION AND DEPLOYMENT

Using a wide range of evaluation measures, the effectiveness of each machine learning model utilized in the text emotion classification system was evaluated. These measures offered a thorough grasp of the models' classification accuracy for emotions in different categories.

The **accuracy score** provided a broad idea of the model's overall performance by calculating the percentage of successfully predicted cases among all forecasts. However, other metrics like precision, recall, and F1-score were also taken into consideration because accuracy alone might not adequately convey the subtleties of classification. **Precision**, which was computed at 0.66, showed the model's capacity to prevent false positives by showing the percentage of real positive predictions among all positive predictions. The model's ability to identify all pertinent emotional situations, including those that were more difficult to categorize, was demonstrated by its **recall**, which was measured at 0.65 and represented the percentage of actual positives accurately detected. The model's ability to manage both false positives and false negatives well was highlighted by the reported **F1-score** of 0.64, which strikes a solid balance between precision and recall. In order to provide insights into certain areas where the model could be improved, a confusion matrix was also used to depict the classification results by displaying the number of predictions that fell into the proper and erroneous categories.

**Streamlit**, an interactive tool for creating data-driven online applications, was used to deploy the model on a web-based platform in order to make it accessible and user-friendly. The model's outputs, such as accuracy measurements, confusion matrix visualizations, and dynamic input for real-time emotion classification, could all be seamlessly integrated thanks to Streamlit. This deployment made it possible for end users to engage with the system with ease and offered an easy-to-use interface for effective and fast textual emotion analysis.