TEXT-BASED EMOTION CLASSIFICATION USING MACHINE LEARNING WITH NLP

A PROJECT REPORT

Submitted by

DHUNEESHA.E [REGISTER NO: 211419104066]

DIVYA.S [REGISTER NO:211419104072]

NIVEDHA.U [REGISTER NO:211419104185]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN
COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2023

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"Text-based Emotion Classification using Machine Learning with NLP"** is the bonafide work of "**Dhuneesha.E(211419104066) ,Divya.S(211419104072),Nivedha.U(211419104185)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.L.JABASHEELA,M.E.,Ph.D., **HEAD OF THE DEPARTMENT**	**Dr. KAVITHA SUBRAMANI M.E,Ph.D.,** **SUPERVISOR** **PROFESSOR**
DEPARTMENT OF CSE, PANIMALAR ENGINEERING COLLEGE, NASARATHPETTAI, POONAMALLEE, CHENNAI-600 123.	DEPARTMENT OF CSE, PANIMALAR ENGINEERING COLLEGE, NASARATHPETTAI, POONAMALLEE, CHENNAI-600 123.

Certified that the above candidate(s) was examined in the End Semester Project Viva-Voce Examination held on..........................

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

DECLARATION

We…**Dhuneesha.E(211419104066),Divya.S(211419104072), Nivedha.U(211419104185)**…. hereby declare that this project report titled "**Text-based Emotion Classification using Machine Learning with NLP**" , under the guidance of **Dr. Kavitha Subramani M.E,Ph.D.,** is the orginial work done by us and we have not plagiarized or submitted to any other degree in any university by us.

DHUNEESHA.E

DIVYA.S

NIVEDHA.U

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our beloved Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.MANI, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. L.JABASHEELA , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank our guide **Dr.KAVITHA SUBRAMANI M.E,Ph.D.,** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

DHUNEESHA.E
DIVYA.S
NIVEDHA.U

ABSTRACT

The collection and evaluation of emotions are the focus of the sentiment analysis subfield known as emotion detection. With the ease of obtaining data and the enormous advantages its deliverables provide, many research are being conducted in the area of text mining and analysis. The proposed approach in text has lately been more well-liked due to its numerous possible applications in marketing, development research, behavioural science, social interaction, automation, etc. In the proposed approach, the text emotion recognition used both speech as well as text to detect emotions. Hence, these methods fall short of creating a useful and flexible system for emotion recognition. A fresh approach was suggested and put into practise for detecting emotions in short entries. As opposed to conventional methods, which are mostly focused on statistical techniques, this approach attempts to infer and extract the causes of emotions by importing information and theories from other disciplines, such as sociology. The approach of emotion cause extraction is employed as a critical step to enhance the quality of chosen characteristics, and it is based on the idea that a prompting cause event is an essential component of emotion. NLP is used to build the supervised machine learning algorithms, and accuracy metrics are used for comparison. The best model, or the one with the highest accuracy, is then chosen after a comparison of the three, and it is implemented into a webpage. The algorithms used are Linear SVM, Random Forest, Decision Tree Classifier. The three models are then compared and the best one ie. the one with highest accuracy is deployed into a webpage. The highest accuracy obtained from the above algorithms is 90%. This work provides a webpage for emotion recognition that takes voice and text input with accurate results.

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

CHAPTER-1

INTRODUCTION

1.1. PROBLEM DEFINITION

Billions of users in this world are attracted by social media, which has been expanded quickly throughout the world for the past few years. Through these social networks there is a huge growth in the data generated by the users which has drawn the interest of the researchers to gain any useful information from this data which led to the need for developing NLP (Natural Language Processing) tools and methods. The main subjective data which is nothing but the sentiment analysis is dealt using this NLP. It deals with two different tasks 1.Opinion mining and 2.Emotion recognition. Emotion Recognition has more interest in the research community because it has many benefits in different fields for example, suicide prevention, children and student motivation and performance, reducing cyber-bullying, etc.., NLP researchers has applied different methodologies for the textual emotion recognition like Machine Learning(ML), rule based methods, lexical approaches. The most used method is the ML technique because it yields better results than many of the other methods. The performance and accuracy of a model depends on the size and quality of training dataset. Therefore, the training data must be carefully chosen to get more accuracy and good performance. This work demonstrates supervised machine learning technique along with NLP which is used to work with natural human language. Data visualization is done to get better insights about the data. Three algorithms are used for textual emotion recognition.The three models are compared based on accuracy and deployed.

CHAPTER 2
LITERATURE SURVEY

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

One of the numerous AI-based technologies that has significantly enhanced how people and society can address urgent issues is natural language processing (NLP).They include traditional algorithms-based models like the support vector machines (SVM), hidden markov model (hmm), and decision tree, which were developed in response to the need to improve efficiency and performance by streamlining the emotion annotation process.[1]In light of this, they developed EmoLabel, a tool that identifies the most common emotions through two stages: automatic pre-annotation and human annotation.

Emotions are used to categorize the thoughts expressed on social media.Among the various emotions are joy, sadness, fear, disgust, rage, surprise, and trust.In order to engage with people more closely, upcoming artificial intelligence will need to be able to recognise and express emotional states. Although the extraction of emotions from written representations of human conversation has shown encouraging results, the accuracy of acoustic feature-based emotion recognition from audio is still lacking.The suggested methodology is improved by a[2] novel feature extraction method based on Bag of Audio Words (BoAW) and cutting-edge recurrent neural networks.Several scholars have suggested methods for finding anomalies using different text mining approaches. Each remark or tweet is updated in casual human handwriting, and unstructured texts are standardised into a standard format to apply ML algorithms utilising NLPT Natural Language Processing methods. Views[3] expressed on social media are categorised by emotions. These categories include joyful, sad, fear, disgust, wrath, surprise, and trust.

This study examines how microblogs may be used to identify anomalies in social media. Text mining and emotion recognition methods from several authors deepen the study.In recent decades, a lot of research has been done on campus or institution security. Assailants have been deterred from accessing a facility by using licence plate recognition, voice verification, and facial identification separately. A hybrid recognition system may greatly boost security, according to various experts, however hybrid systems aren't frequently covered in the literature. In order to solve this problem,[4] a hybrid driver and vehicle identification module that can identify both the driver and the vehicle was introduced in this study. Face and speech recognition software is used to identify drivers.

FaceNet was used to identify faces, and multi-task cumulates convolutional networks were used to crop the faces for facial recognition.[5]Dynamic interactions occurring throughout the data procedure, and they suggest the use of a Dynamic Interactive Multiview Memory Network (DIMMN) model to include relevant information for emotion recognition. Particularly, numerous views are used inside DIMMN to combine information. TFIDF, a metric for word importance in a text, is used in certain research to propose a novel way of emotion detection[6].Using sentiment analysis, it can be said that if the material you're given has a positive, unfavorable, or neutral attitude. Nevertheless,emotion analysis goes farther than that and operates by segmenting the sentiment analysis categories[8]..The research put out makes use of machine learning neural networks to determine the gender and disposition of a speaker.Two modules, notably the blocks for gender identification and emotion detection, must be integrated to complete the assignment.

The emotion detection block is built by a convolutional neural network (CNN), trained and tested on a large dataset, while the gender identity block is formed by a simple feed-forward neural network[10].Individuals utilize websites such as Twitter to textually convey their views and opinions about relevant problems. For greater relationships between individuals and machines, it is crucial to comprehend emotions at a subtler level than just feeling something.[11] Consequently,in this the BERT language model towards emotion detection in Tweets written in Indonesian. Rather than pre-training, which necessitates a large amount of data and resources, employ fine-tuning. The efficiency and efficacy of the suggested model were evaluated using two pre-trained models.

2.2 COMPARISON TABLE BASED ON LITERATURE SURVEY

YEAR & AUTHOR	**TITLE**	**METHODOLOGY**	**MERITS**	**DEMERITS**	**FUTURESCOPE**
2022, Jennifer Santoso,et al.	Speech Emotion Recognition Based on Self-Attention Weight Correction for Acoustic and Text Features	BLSTM-and-self-attention-based SER method using self-attention weight correction (SAWC) with confidence measures	investigated the behavior of our proposed SAWC in each of the feature extractors.	accuracy is low	Increase in accuracy

YEAR & AUTHOR	TITLE	METHODOLOGY	MERITS	DEMERITS	FUTURESCOPE
2022,Sudarshan Pant,et al.	Korean Drama Scene Transcript Dataset for Emotion Recognition in Conversations	Bidirectional Encoder Representations from Transformers (BERT)	The dataset was annotated with three labels euphoria, dysphoria, and neutral to represent the emotional state of the characters, F1-score of 0.6288	Low accuracy	Improving accuracy
2022,Jing Zhang,et al.	Graph-Based Object Semantic Refinement for Visual Emotion Recognition	Graph Convolutional Networks (GCN)	four widely used benchmark datasets show that our proposed method can achieve competitive performance and outperform most of the state-of-the-art methods on visual emotion recognition.	Complex technique	Extract more effective semantic features through explicit or implicit modelling for visual emotion analysis.
2022,Namrata Chaudhari,et al.	Artificial Intelligence System for Emotion Recognition and Text Analytics	Deep neural network (DNN) for classifying emotions based on features extracted from facial expressions	accuracy of about 86.75%, The integrated system extracts video and audio simultaneously with a frame rate of 4-5 fps.	The emotion classified using speech is only three types(positive ,negative,neutral)	Types of emotions can be increased.
2022,Samuel Kakuba, et al	Deep Learning-Based Speech Emotion Recognition Using Multi-Level Fusion of Concurrent Features	Multi-level fusion first at the LFLB level, (DCC), bidirectional long short-term memory (Bi LSTM), transformer encoders (TE), multi-head and self-attention mechanisms	achieves 75.50% and 75.82% of weighted and unweighted accuracy, 75.32% and 75.57% of recall and F1 score respectively.	F1 score is not upto the mark	Explore concurrent feature learning of spatial, temporal,semantic tendencies in all modalities for emotion recognition

YEAR & AUTHOR	TITLE	METHODOLOGY	MERITS	DEMERITS	FUTURESCOPE
2022,Lea Canales, et al	EmoLabel: Semi-Automatic Methodology for Emotion Annotation of Social Media Text	BLSTM-and-self-attention-based SER method using self-attention weight correction (SAWC) with confidence measures	Accuracy is increased by 20%	Semi automatic technique is used	Accuracy can be increased.
2021,Samuel W. K. Chan	Multilabel Emotion Tagging for Domain-Specific Texts	Multilabel Emotion Parsing	This approach provides better modelling of compositional emotions by considering the emotion-bearing words, shifters, intensifiers, and overall sentence structure.	accuracy not mentioned	Model can be built for speech and gestures also.
2022,N.Susithra,et al	Speech based Emotion Recognition and Gender Identification using FNN and CNN Models	The work proposed in this paper makes use of machine learning neural networks to recognize the gender and emotion of a speaker	Gender detection accuracy is high(91.46)	accuracy for emotion detection is low(86%), Only four emotions are detected	Accuracy can be increased for emotion detection.
2022,Meenu S Nair,et al.	Transfer learning for Speech Based Emotion Recognition	This work uses the transfer learning method based Time Delay Neural Network (TDNN), which uses the knowledge learned from one domain and applies it to another domain with fewer data.	Transfer learning can be used to identify emotions in languages	Only one language dataset is used	Multiple language dataset can be used and accuracy can be increased

YEAR & AUTHOR	TITLE	METHODOLOGY	MERITS	DEMERITS	FUTURESCOPE
2022,Abhishek Kumar, et al.	Speech Emotion Recognition by using Feature Selection and Extraction	Emotional recognition based on previous technology that uses different dividers to monitor emotions in the review.	They have reviewed previous technologies	Not implemented any new technologies	New methods can be implemented

.

Table 2.1 Comparison table based on literature survey

CHAPTER 3
SYSTEM ANALYSIS

CHAPTER-3

SYSTEM ANALYSIS

3.1. EXISTING SYSTEM

The exponential growth of the amount of subjective information on the Web 2.0. has caused an increasing interest from researchers willing to develop methods to extract emotion data from these new sources. One of the most important challenges in textual emotion detection is the gathering of data with emotion labels because of the subjectivity of assigning these labels. Based on this rationale, the main objective of the research is to contribute to the resolution of this important challenge. The rationale behind the research is the need to simplify the emotion annotation task so that to improve its reliability and efficiency.The base paper shows the EmoLabel: a semi-automatic methodology consisting in two phases: (1) an automatic process to pre-annotate the un-labelled sentences with a reduced number of emotion categories; and (2) a manual refinement process where human annotators determines which is the dominant emotion between the predefined set of possibilities.

DISADVANTAGES

- Accuracy is not mentioned.
- Data visualization is not done.
- Deployment is not done.
- Voice input is not used

3.2. PROPOSED SYSTEM

The proposed model is to build a machine learning model that is capable of classifying what type of emotion in the text. The emotion text are considered to be widespread and controlling them is very difficult as the world is developing toward digital everyone now has access to internet and they can post whatever they want. So there is a greater chance for the people to get misguided. The machine learning is generally build to tackle these type of complicated task. It takes more amount of time to analyse these type of data manually. The machine learning can be used to classify the text whether what type of emotion it is by using the previous data and make them to understand the pattern and improve the accuracy of the model by adjusting parameters and use that model as the classification model. Different algorithms are compared and the best model can be used for classification purpose. The new feature added here is that it can take voice input and convert it into text , then detect the type of emotion in the text.

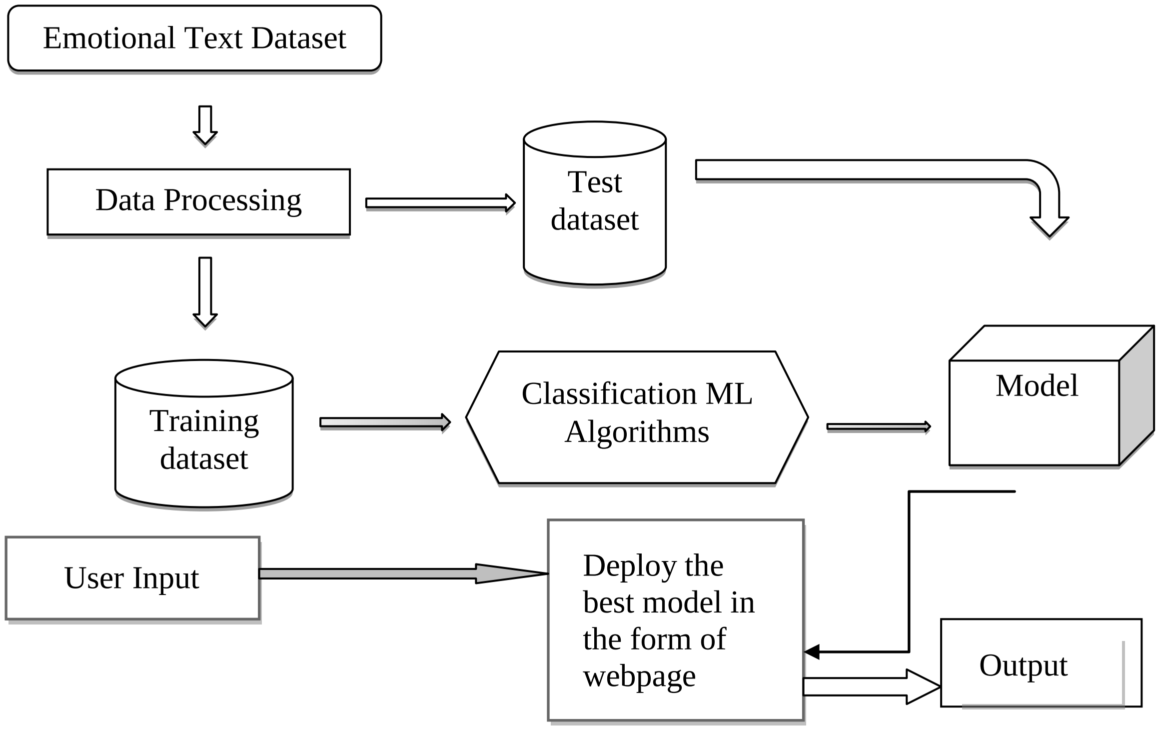

Figure 3.1 Working of proposed system

Advantages:

- Accuracy is between 85-90%.
- More algorithms will be implemented.
- Performance metrics are calculated.
- Data visualization is done.
- Voice input is used.

3.3. FEASIBILITY STUDY

DATA WRANGLING

The data wrangling is checking for cleanliness, and then trimming and cleaning given dataset for analysis.

DATA COLLECTION

The data set collected for predicting given data is split into Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The data model which was created using Random Forest, Linear Support Vector Machine and Decision Tree Classifier are applied on the Training set and based on the test result accuracy, Test set prediction is done. In this project, the train and test dataset ratio is 8:2.

PREPROCESSING

The dataset collected might contain missing values that may lead to inconsistency. To gain better results data need to be preprocessed so as to improve the efficiency of the algorithm. The outliers have to be removed and also variable conversion need to be done.

3.4. PROJECT REQUIREMENTS

FUNCTIONAL REQUIREMENTS

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists requirements of a particular software system. The following details to follow the special libraries like sk-learn, pandas, numpy, matplotlib and seaborn.

NON-FUNCTIONAL REQUIREMENTS

Process of functional steps,

1. Problem define
2. Preparing data
3. Evaluating algorithms
4. Improving results
5. Prediction the result

ENVIRONMENTAL REQUIREMENTS

1. Software Requirements:

Operating System : Windows 10 or later

Tool : Anaconda with Jupyter Notebook

2. Hardware requirements:

Processor : Intel i3

Hard disk : minimum 10 GB

RAM : minimum 4 GB

CHAPTER 4

SYSTEM DESIGN

CHAPTER-4

SYSTEM DESIGN

4.1. UML DIAGRAMS

4.1.1. USE CASE DIAGRAM

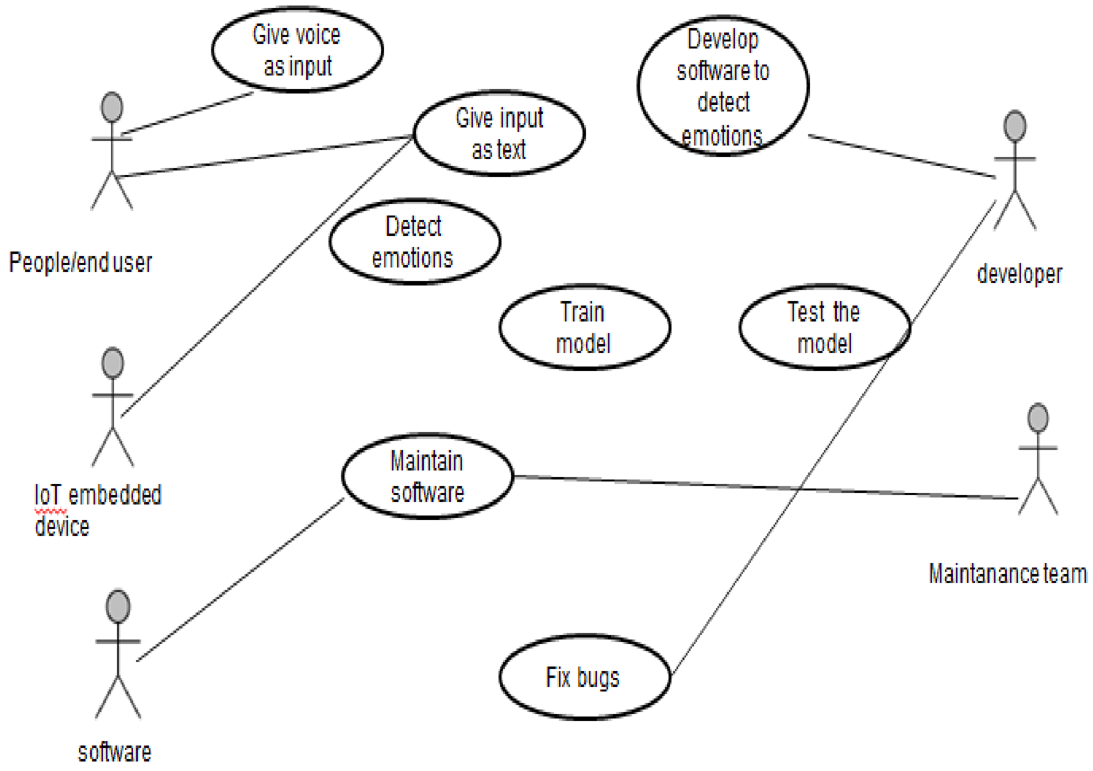

Figure 4.1 Use case diagram

The above use case diagram contains users like people who gives text, software that detects the emotions,Iot embedded devices,develper who fixes the bug and the maintenance team for further maintenance of the software developed. The developer also developes the software and trains and test the model using the algorithms.

4.1.2. CLASS DIAGRAM

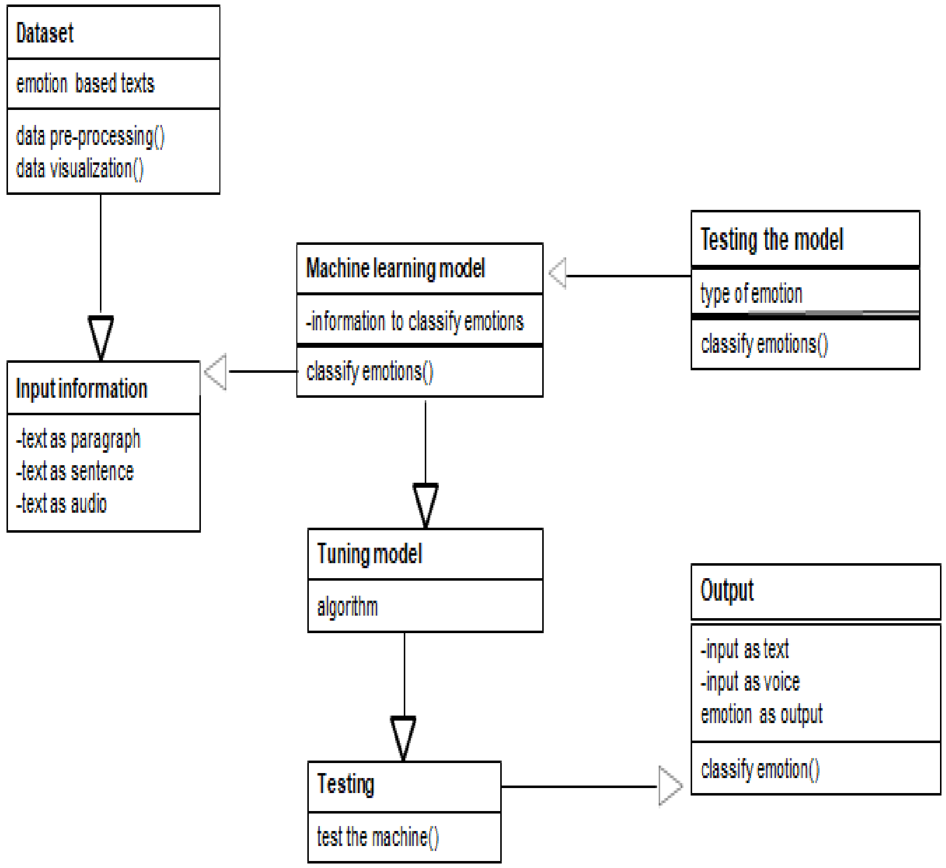

Figure 4.2 Class diagram

The above mentioned class diagram represents the detailed diagrammatic representation of the classes used. The dataset class is used for data preprocessing and data visualisation, the input information class contains textasparagraph and textassentence, the Machine learning model class is used to classify the emotions, testing the model class is also used for classify the emotions, tuning model contains the algorithm required for training and testing the model, testing class contains a testthemachine function and finally the output class is used to classify and represent the emotion as output.

4.1.3. ACTIVITY DIAGRAM

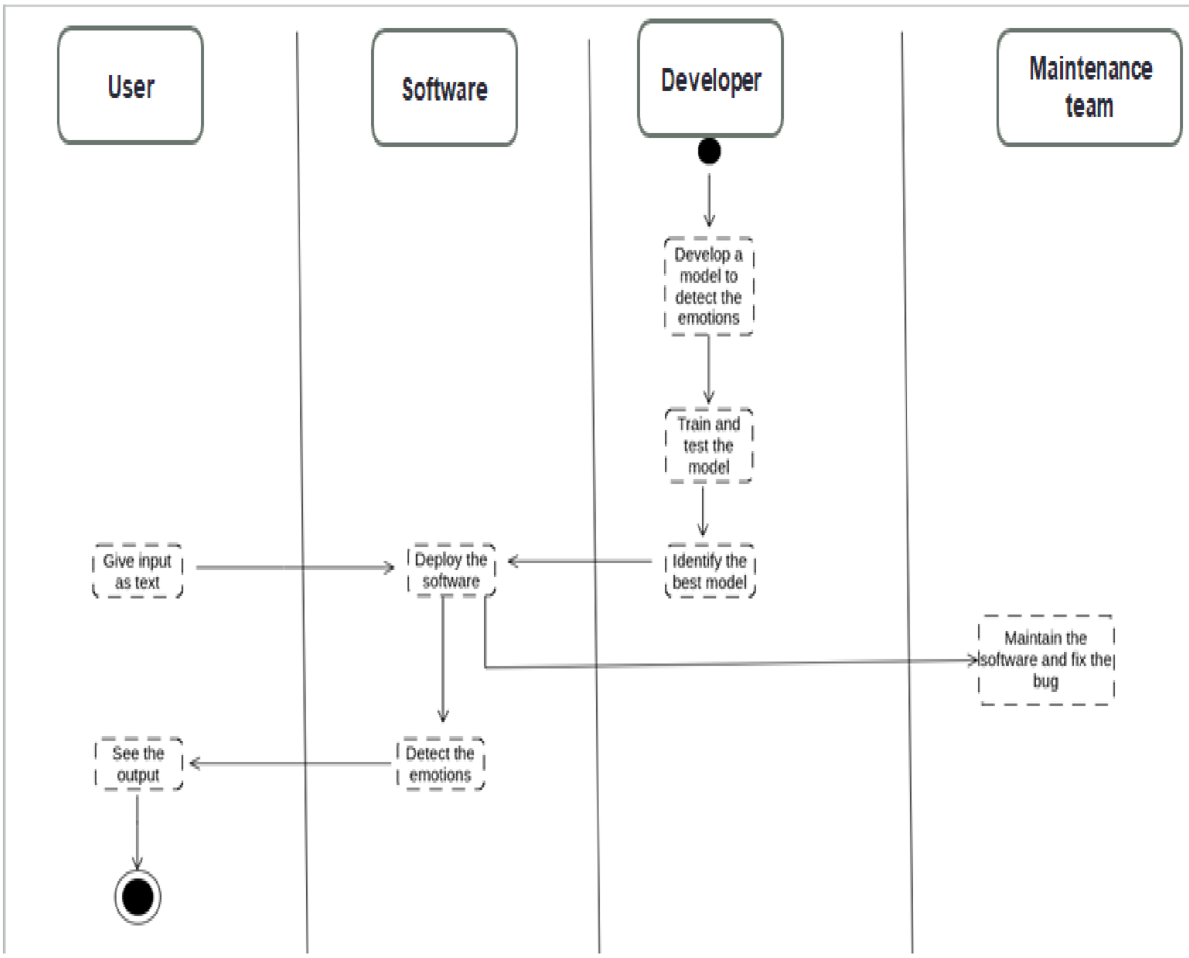

Figure 4.3 Activity diagram

The activity diagram represents the entire activities that are covered from the user giving the input to the software detecting the emotion and representing it as a output to the user. Initially the user gives the input as text to the software which is then used to detect the emotions by the software developed by the developer and represented as the output to the user . If there is any bug in the software it is then cleared by the maintenance team.

4.1.4. SEQUENCE DIAGRAM

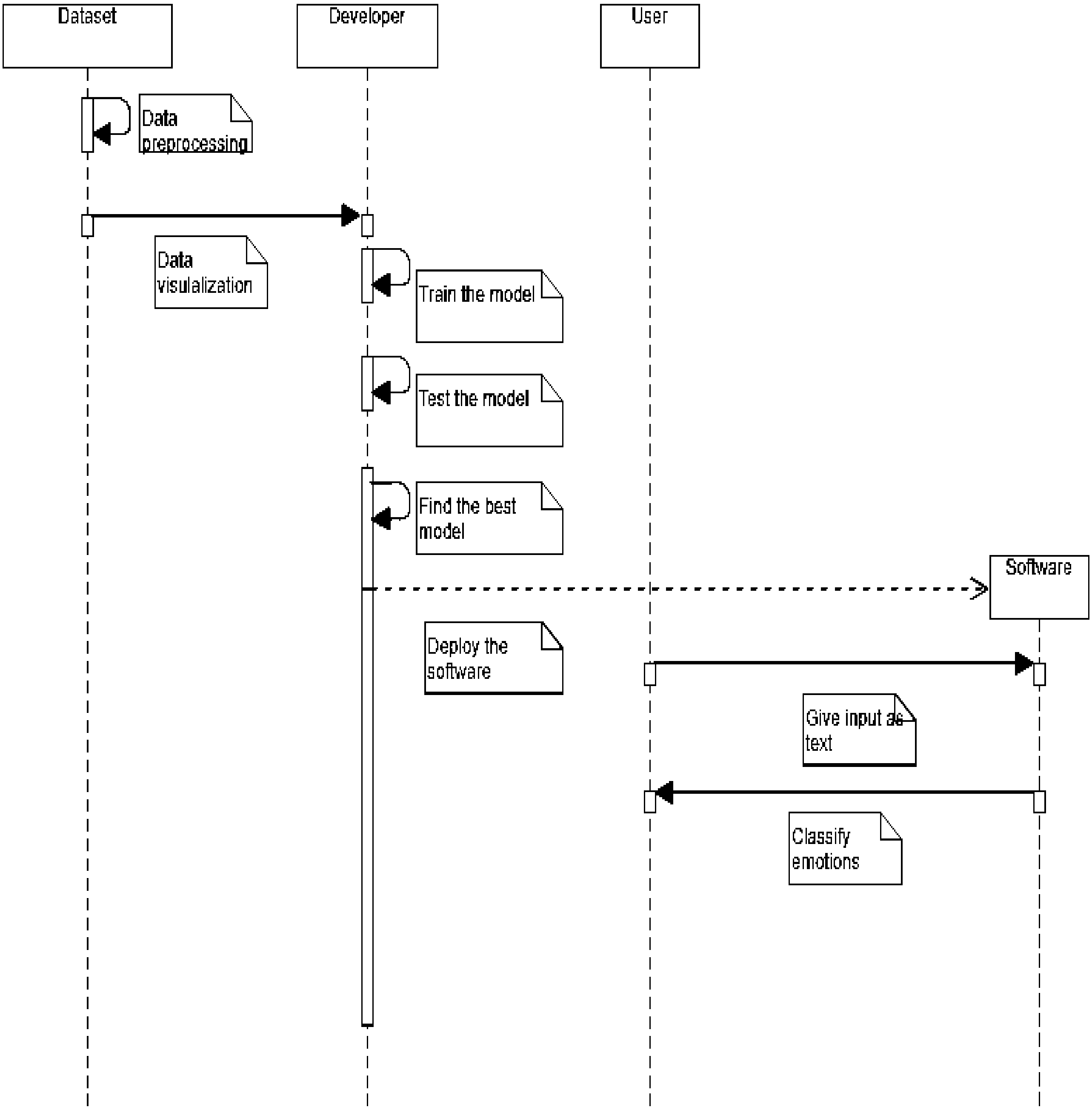

Figure 4.4 Sequence diagram

The above diagram depicts all the sequence of activities in detecting the emotion from the input given by the user. Initially the data preprocessing on the dataset is done to get the required data and data visualisation to represent the data in a understanding firmat. Then the developer uses this preprocessed data to train and test the model and finally finds the best model and deploys the software. It is then used by the users to give the input and detect the emotions.

4.2. DATA FLOW DIAGRAM

4.2.1. LEVEL 0

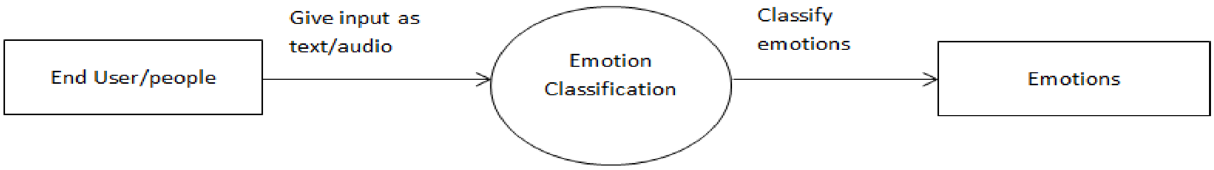

Figure 4.5 Level 0 Dataflow diagram

The Level 0 DFD diagram shows that the input will be given as text/audio by the end user and the emotions are classified .

4.2.2. LEVEL 1

The below diagram shows the flow of data in this project starting from collection of data to deployment containing data preprocessing,cleaning and the building model for the project.

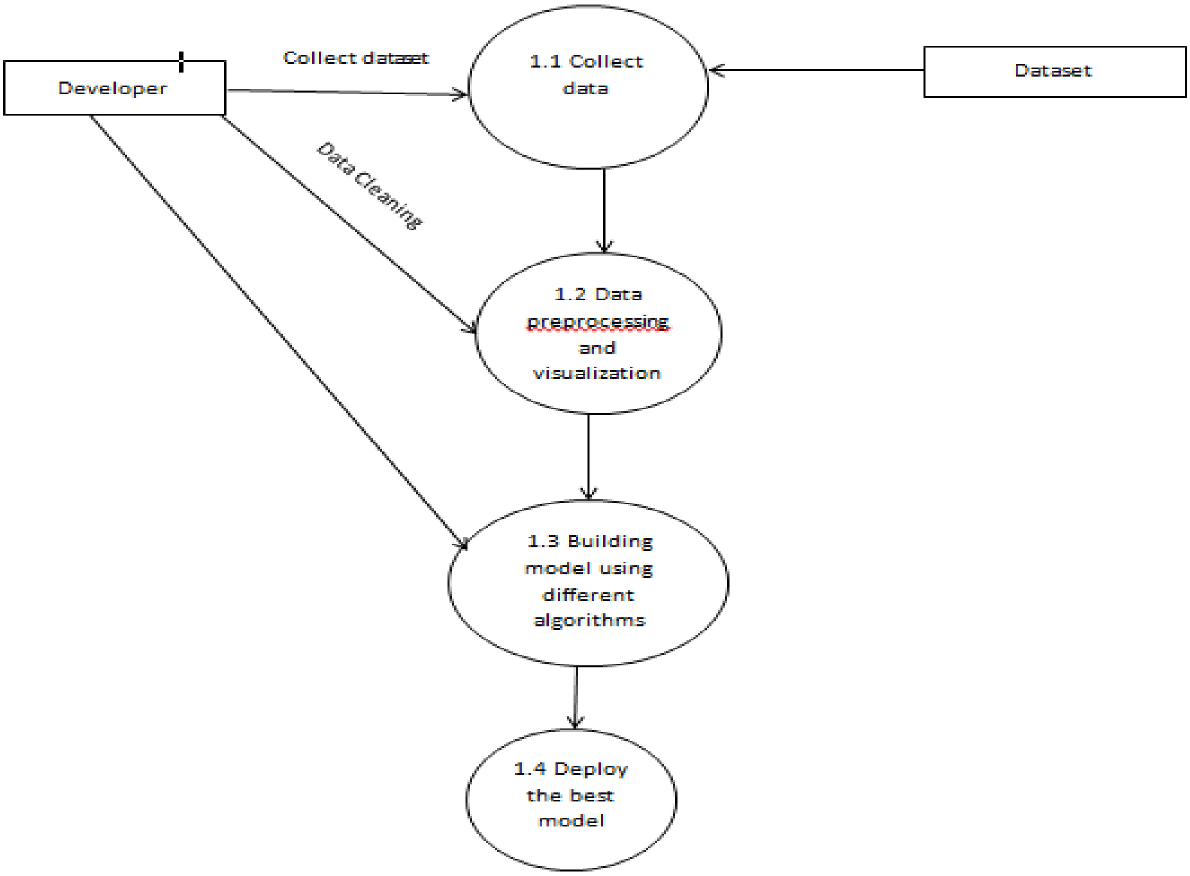

Figure 4.6 Level 1 Dataflow diagram

CHAPTER 5

SYSTEM ARCHITECTURE

CHAPTER -5

SYSTEM ARCHITECTURE

5.1. SYSTEM ARCHITECTURE

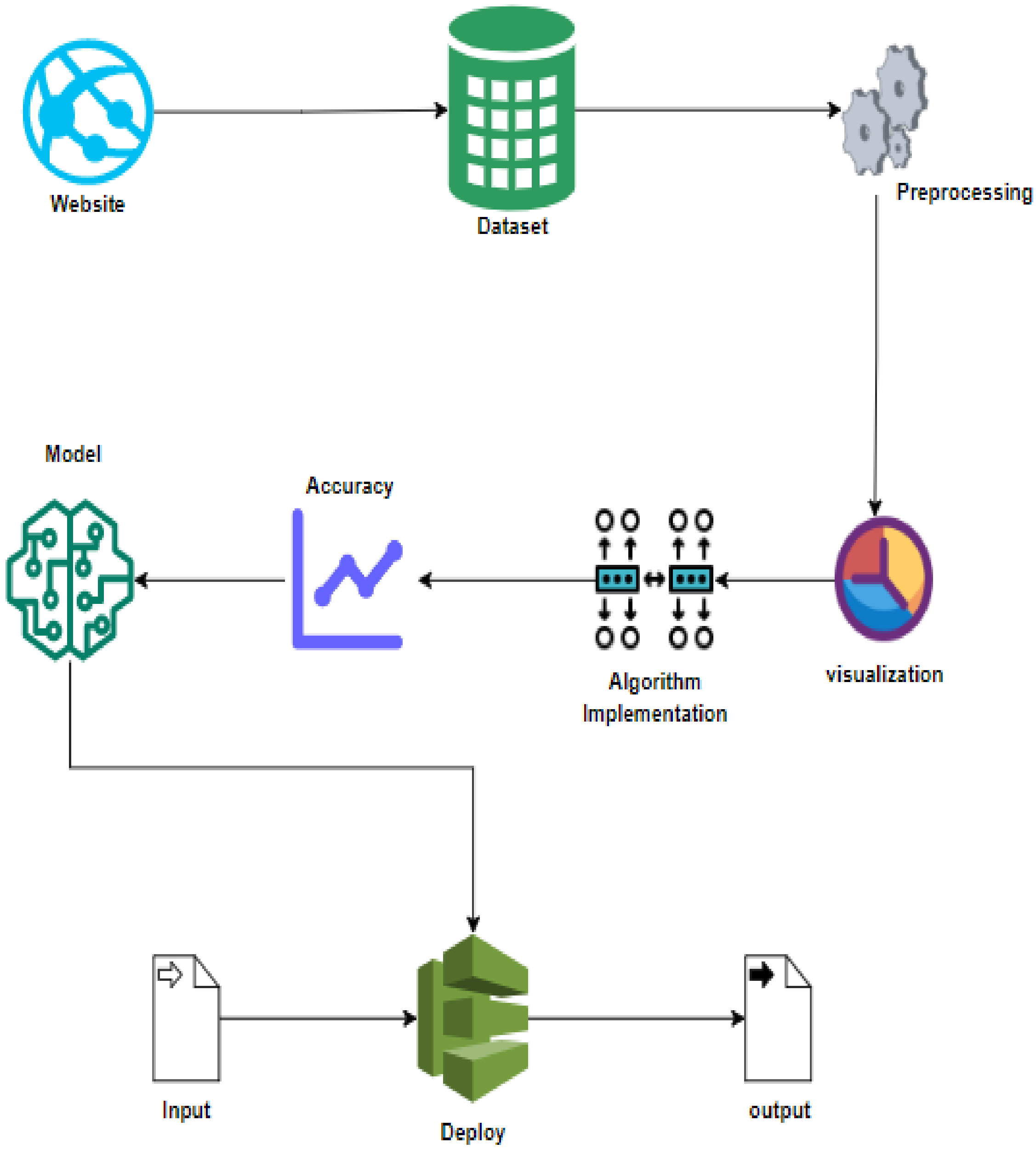

Figure 5.1 System Architecture diagram

The architecture diagram shows the processes involved for building the project. It involves collecting dataset from website, the processing it to remove the noisy data, visualizing it and then implementing algorithms and finding the best model based on accuracy and then deploying it in the form of webpage.

5.2. WORK FLOW DIAGRAM

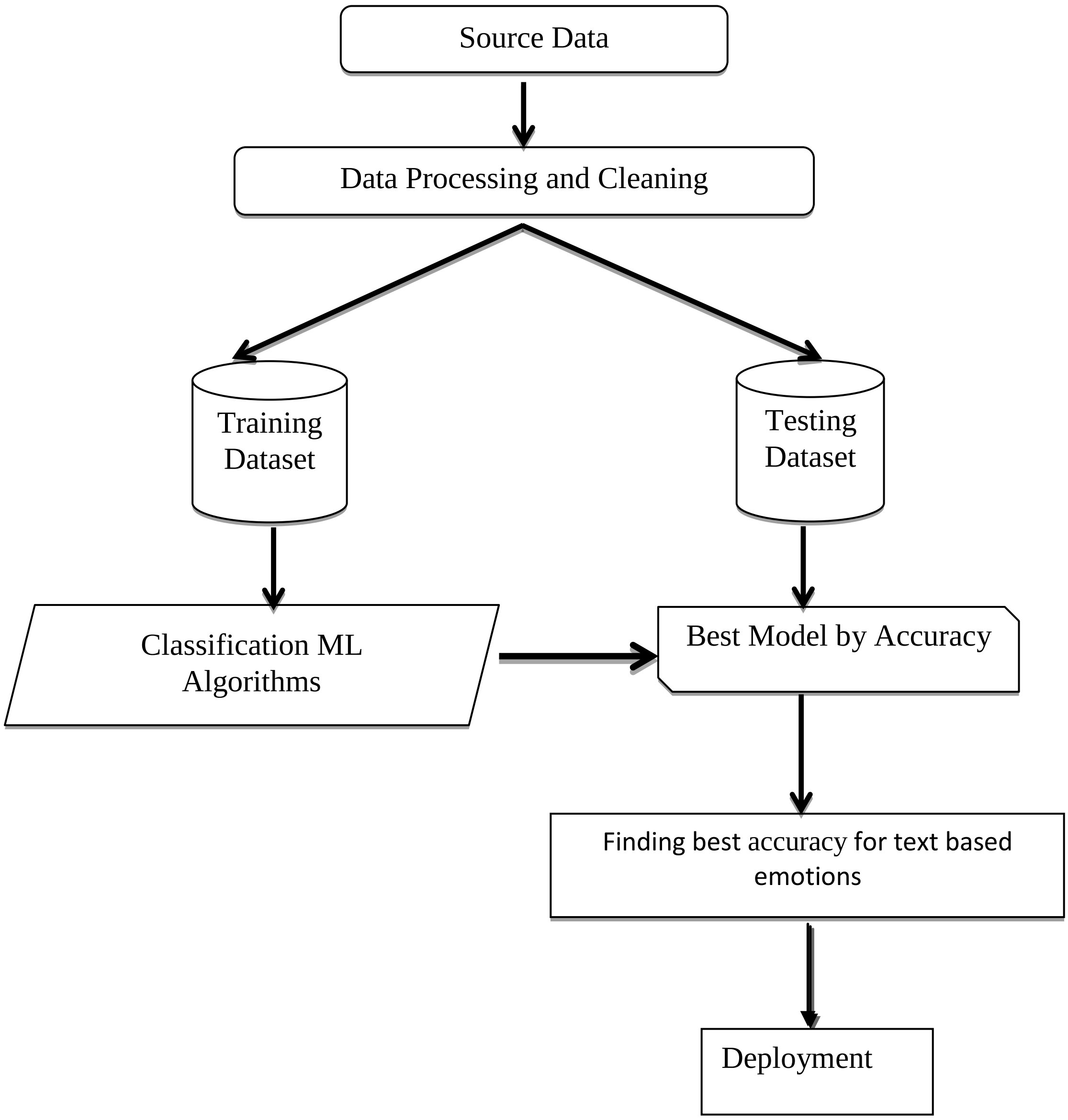

Figure 5.2 Work flow diagram for proposed system

The workflow diagram shows the flow of proposed work. That is it starts from dataset collection, then processing the data and cleaning. After splitting it into test and train data, the algorithm is implemented to train the model on emotion recognition and then deploying it as webpage by finding the best model.

CHAPTER 6

SYSTEM IMPLEMENTATION

CHAPTER-6

SYSTEM IMPLEMENTATION

6.1. MODULE DESCRIPTION

LIST OF MODULES

- Data Pre-processing
- Data Analysis of Visualization
- Implementing Random Forest Classifier Algorithm
- Implementing Decision Tree Algorithm
- Implementing Linear Support Vector Machine Algorithm
- Deployment Using Flask

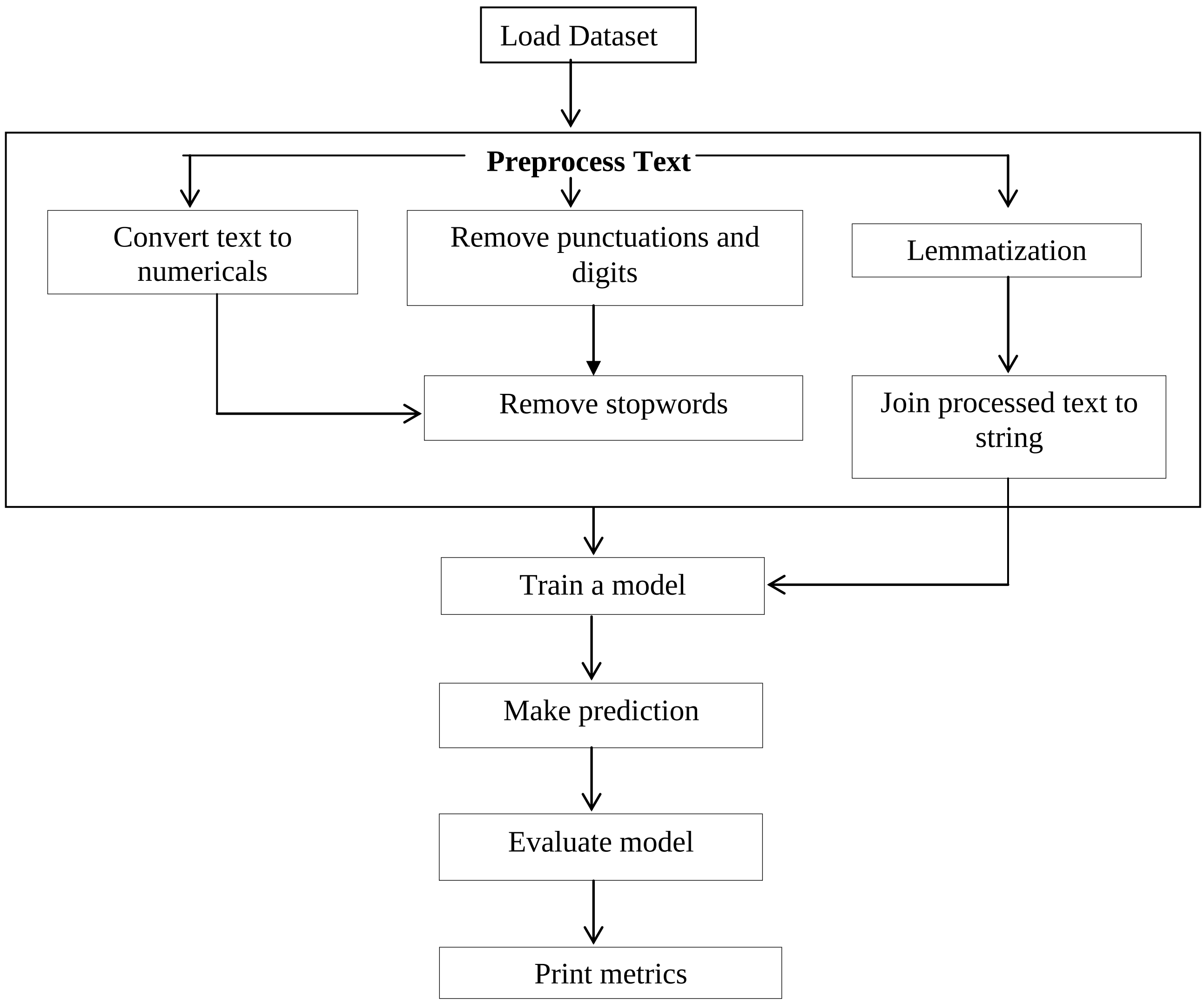

Figure 6.1 Flow of modules

6.1.1. DATA PRE-PROCESSING

DATASET COLLECTION

Anger	Fear	Happy	Love	Sadness	Surprise	Total
2993	2652	7029	1641	6265	879	21459

Table 6.1 Emotions in dataset

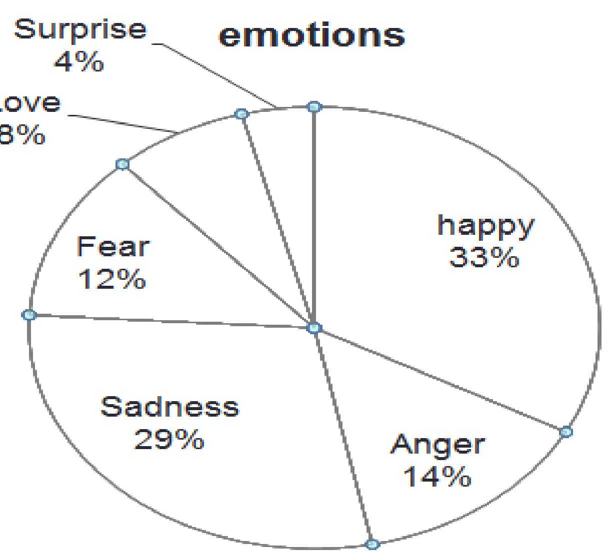

Figure 6.2 Emotions in dataset

Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand the data and its properties; this will help in choosing which algorithm to use to build your model. For this work, we have chosen "emotion in text" dataset from Kaggle. This contains six emotions namely anger, fear, joy, sad, surprise, love. The shape of dataset is (21549,2).

DATA CLEANING

A number of different data cleaning tasks using Python's Pandas library and specifically, it focus on probably the biggest data cleaning task, missing values and it able to more quickly clean data. The time for cleaning data should me less, and more time should be spent for exploring and modeling.

Some of these sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It's important to understand these different types of missing data from a statistics point of view. Data preprocessing takes dataset as input and removes noisy data from it. The type of missing data will influence how to deal with filling in the missing values and to detect missing values, and do some basicimputation anddetailed statistical approach for dealing with missing data. Before, joint into code, it's important to understand the sources of missing data.Here are some typical reasons why data is missing:

- User forgot to fill in a field.
- Data was lost while transferring manually from a legacy database.
- There was a programming error.
- Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.

Variable identification with Uni-variate, Bi-variate and Multi-variate analysis:

- import libraries for access and functional purpose and read the given dataset
- General Properties of Analyzing the given dataset
- Display the given dataset in the form of data frame
- show columns
- shape of the data frame
- To describe the data frame
- Checking data type and information about dataset
- Checking for duplicate data
- Checking Missing values of data frame
- Checking unique values of data frame
- Checking count values of data frame
- Rename and drop the given data frame
- To specify the type of values
- To create extra columns

It is the process of cleaning the imbalanced data, unwanted data, duplicate data, missing values. It involves

- Tokenising the data
- Removing stopwords

- Eliminating punctuation
- Encoding the data

6.1.2. DATA VISUALIZATION

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

In this project, we use “matplotlib” and “seaborn” library for data visualization. Word clouds are a type of data visualization that can be used to visualize the most commonly used words in a text. Word Clouds were generated for all the emotions in pairwise manner. Overall, data visualization can be a powerful tool for analyzing and understanding patterns in text-based emotion recognition, and can help researchers and practitioners make more informed decisions about how to interpret and use emotional data.

6.1.3. ML MODEL DEVELOPMENT

After preprocessing and visualization of data, algorithm implementation takes place. It starts with importing necessary libraries such as pandas, regular expressions, stopwords and WordNetLemmatizer from the Natural Language Toolkit (nltk) for text preprocessing.The preprocessing method is the same for all the three algorithms. Then it loads the dataset using Pandas and preprocesses the text data using the function preprocess_text(), which converts the text to lowercase, removes punctuation and digits, removes stopwords and lemmatizes the words. After preprocessing, the text data is converted to numerical vectors using TfidfVectorizer, which is a method for converting text data into a numerical form that can be used by machine learning algorithms. Then, the data is split into training and testing sets using train_test_split from scikit-learn, and data is trained on the by using the corresponding libraries and functions. For example, Decision Tree Classifier uses “from sklearn.tree import DecisionTreeClassifier”. Next, the trained model is used to predict the emotions on the testing data and evaluate the performance using metrics like accuracy, precision, recall, and f1-score.

The below 3 different algorithms are compared:

- Decision Tree Classifier model
- Random Forest Classifier model
- Linear Support Vector Machine classifier model

6.1.3.1. DECISION TREE CLASSIFIER

A tree has many analogies in real life, and turns out that it has influenced a wide area of machine learning, covering both classification and regression. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions. Though a commonly used tool in data mining for deriving a strategy to reach a particular goal, its also widely used in machine learning, which will be the main focus of this article.

The Decision Tree Classifier algorithm is trained used the “DecisionTreeClassifier” module form sklearn.tree library. The hyperparameters used in this algorithm are max_depth, criterion. The max_depth represents the depth of the decision tree and the criterion shows the measure of quality of split. The max_depth value is set to 1000. The criterion used for building decision tree is entropy.

The formula for entropy in decision tree is:

$$Entropy = -\sum_{i=1}^{n}p_i\log_2(p_i)$$

where n is the number of classes, and p_i is the proportion of the number of instances that belong to class i in the dataset. The entropy ranges from 0 to 1, where 0 represents a completely homogeneous dataset (all instances belong to the same class), and 1 represents a completely heterogeneous dataset (an equal number of instances belong to each class).

6.1.3.2. RANDOM FOREST CLASSIFIER

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random forest is a type of supervised machine learning algorithm based on ensemble learning. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision *trees*, resulting in a *forest of trees,* hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

The following are the basic steps involved in performing the random forest algorithm:

- Pick N random records from the dataset.
- Build a decision tree based on these N records.
- Choose the number of trees you want in your algorithm and repeat steps 1 and 2.

Random Forest is trained used the "RandomForestClassifier" module form sklearn.ensemble library. The dataset is split into train and test ie. "X_train, X_test, y_train, y_test". The size of test dataset is 20% of the total dataset. The value of random state used to train the model using Random Forest is 42. One of the hyperparameters used in random forest is "n_estimators". The default value is 10and it is set to 100 in this model. The n_estimators represents the number of decision trees.

The math equation for the prediction of a Random Forest model can be represented as follows:

$$\hat{y} - \frac{1}{B}\sum_{i\ \ 1}^{B} T_i(x)$$

where:

y-hat is the predicted class label for the input vector x

B is the number of decision trees in the forest

$T_i(x)$ is the predicted class label of the i^{th} decision tree in the forest for the input vector x

The output of each decision tree is combined by taking the majority vote (for classification) of the individual predictions of the trees.

6.1.3.3. LINEAR SUPPORT VECTOR MACHINE

A Linear Support Vector Machine (SVM) is a type of machine learning algorithm used for classification and regression analysis. It is a type of supervised learning algorithm that works by finding the hyperplane that best separates the data into different classes.

Linear SVMs can be extended for multiclass classification tasks in several ways. One common approach is the "one-vs-all" (OVA) method, where multiple binary classifiers are trained, each one distinguishing between one class and the rest. Another approach is the "one-vs-one" (OVO) method, where a binary classifier is trained for each pair of classes.

In both OVA and OVO methods, the binary classifiers can be trained using linear SVM or other classifiers, depending on the nature of the problem and the characteristics of the data. Overall, linear SVMs can be extended for multiclass classification tasks in various ways, and their performance can be improved by optimizing the choice of method and parameters

The hyperplane is determined by the equation:

$$\mathbf{w^T * x + b = 0}$$

where w is the weight vector, x is the feature vector, and b is the bias term.

The Linear Support Vector Machine algorithm is trained used the “LinearSVC” module form sklearn.svm library. The size of test dataset is 20% of the total dataset. The value of random state used to train the model using Random Forest is 42.

6.1.4. DEPLOYMENT

In this module the trained machine learning model is converted into pickle data format file (.pkl file) which is then deployed in our flask framework for providing better user interface and predicting the output of how much the given data is emotions based on texts. The webpage takes voice as well as text input to recognize emotions. The voice input method is achieved using HTML and Javascript code.

CHAPTER 7
PERFORMANCE EVALUATION

CHAPTER-7
PERFORMANCE EVALUATION
7.1. RESULTS AND DISCUSSIONS

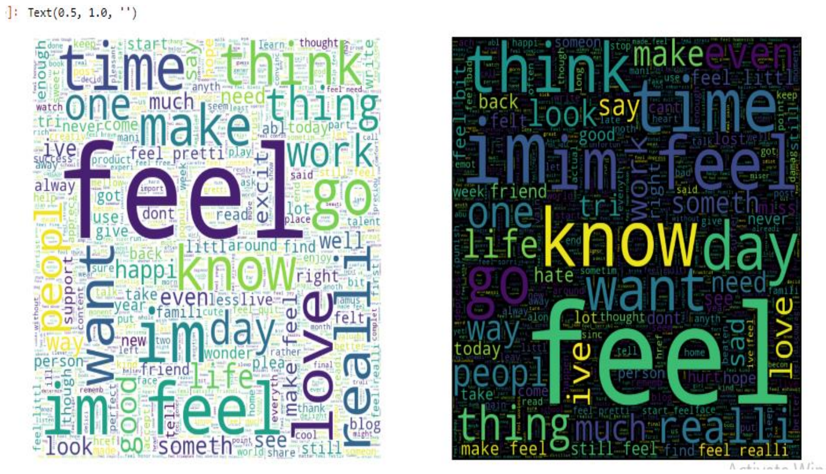

Figure 7.1 Screenshot ofWord clouds of happy and sadness

The above screenshot shows the output of data visualization done in module-2. It is the word cloud of happy and sadness ie. it shows some of the words that fall under happy and sadness. It was implemented was all other emotions.

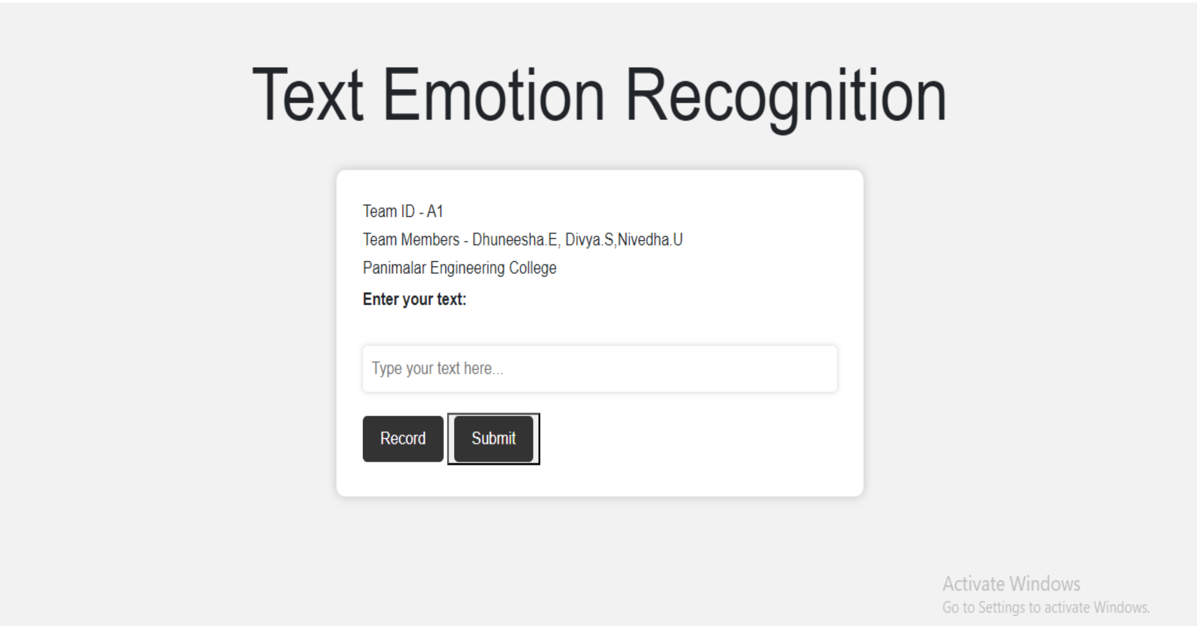

Figure 7.2 Screenshot of webpage for text emotion recognition

This screenshot shows the main page where the text/sentence/audio is given as input and the emotion is detected.

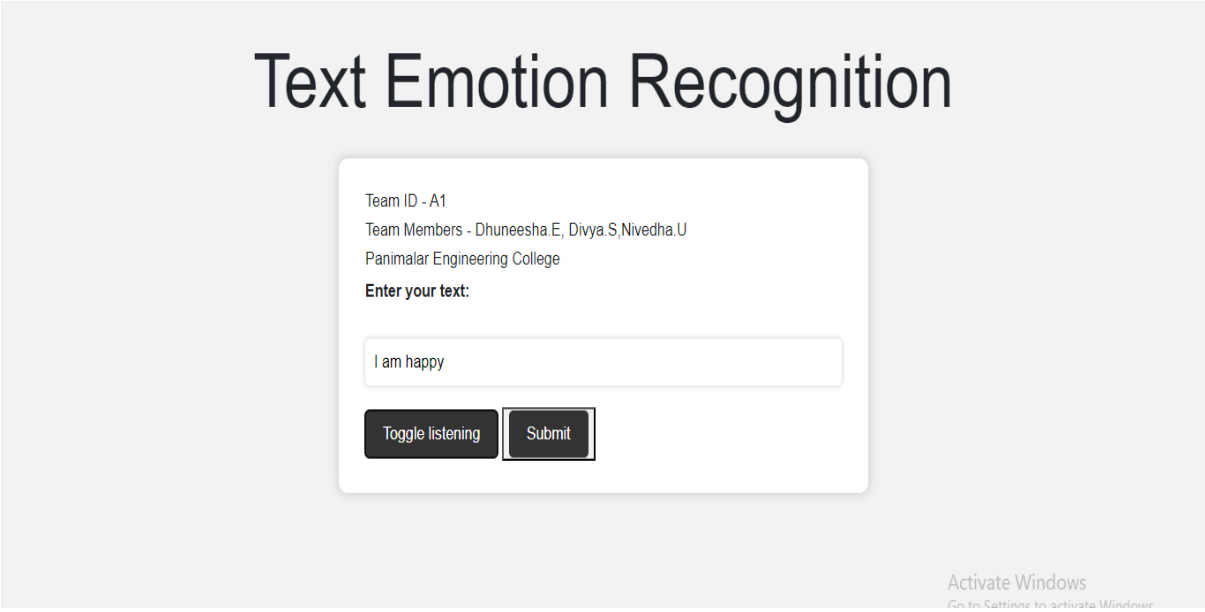

Figure 7.3 Screenshot of Entering the text for recognition

The above screenshot shows the way of entering text into the textbox either by typing or as an audio. The audio can be given by clicking Record / Toggle Listening button

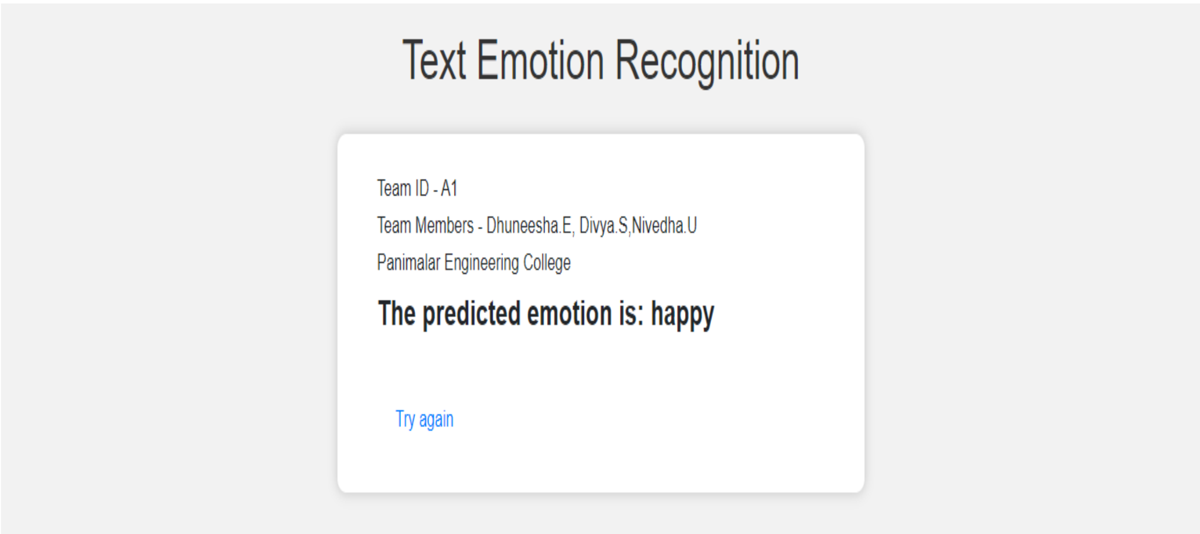

Figure 7.4 Screenshot of the predicted emotion is displayed

This figure shows how the type of emotion will be displayed in the webpage. By clicking on “Try Again”, it goes to the main webpage when input can be given.

7.2. COMPARATIVE ANALYSIS

```
In [6]: # Calculate evaluation metrics
        accuracy = accuracy_score(y_test, y_pred)
        precision = precision_score(y_test, y_pred, average='weighted')
        recall = recall_score(y_test, y_pred, average='weighted')
        f1 = f1_score(y_test, y_pred, average='weighted')

        print('Accuracy using Decision Tree classifier:', accuracy)
        print('Precision using Decision Tree classifier:', precision)
        print('Recall using Decision Tree classifier:', recall)
        print('F1-score using Decision Tree classifier:', f1)

        Accuracy using Decision Tree classifier: 0.8581081081081081
        Precision using Decision Tree classifier: 0.8588374686792589
        Recall using Decision Tree classifier: 0.8581081081081081
        F1-score using Decision Tree classifier: 0.8584058815749784
```

Figure 7.5 Screenshot of accuracy of Decision Tree algorithm

The above figure shows the accuracy,precision,recall,f1-score obtained by training the model using Decision Tree Classifier algorithm. The accuracy obtained is 86%.

```
In [6]: # Calculate evaluation metrics
        accuracy = accuracy_score(y_test, y_pred)
        precision = precision_score(y_test, y_pred, average='weighted')
        recall = recall_score(y_test, y_pred, average='weighted')
        f1 = f1_score(y_test, y_pred, average='weighted')

        print('Accuracy using Random Forest:', accuracy)
        print('Precision using Random Forest:', precision)
        print('Recall using Random Forest:', recall)
        print('F1-score using Random Forest:', f1)

        Accuracy using Random Forest: 0.8846691519105312
        Precision using Random Forest: 0.8843725461360431
        Recall using Random Forest: 0.8846691519105312
        F1-score using Random Forest: 0.883901052249225
```

Figure 7.6 Screenshot of accuracy of Random Forest algorithm

The above figure shows the accuracy,precision,recall,f1-score obtained by training the model using Random Forest Classifier algorithm. The accuracy obtained is 88%.

```
In [6]: # Calculate evaluation metrics
        accuracy = accuracy_score(y_test, y_pred)
        precision = precision_score(y_test, y_pred, average='weighted')
        recall = recall_score(y_test, y_pred, average='weighted')
        f1 = f1_score(y_test, y_pred, average='weighted')

        print('Accuracy using Linear SVM:', accuracy)
        print('Precision using Linear SVM:', precision)
        print('Recall using Linear SVM:', recall)
        print('F1-score using Linear SVM:', f1)

        Accuracy using Linear SVM: 0.9000465983224604
        Precision using Linear SVM: 0.8992366302005806
        Recall using Linear SVM: 0.9000465983224604
        F1-score using Linear SVM: 0.899027161380638
```

Figure 7.7 Screenshot of accuracy of Linear SVM algorithm

The above figure shows the accuracy,precision,recall,f1-score obtained by training the model using Linear Support Vector Machine algorithm. The accuracy obtained is 90%

In previous research papers on text emotion recognition, the maximum accuracy obtained was 86.75%. The number of emotions used to train the model were also less. In the proposed work, the model is trained to identify 6 types of emotions. Also the highest accuracy obtained from the three models is 90%.

COMPARISON OF ALGORITHMS

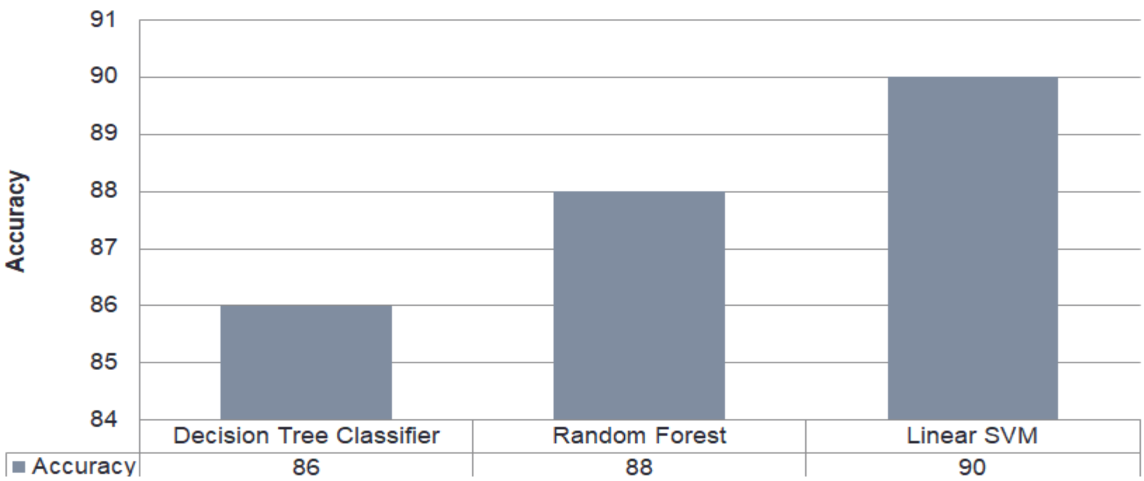

Figure 7.8 Comparison of three algorithms

The above graph shows the comparison between three algorithms of proposed work. From the above, it can be said that “Linear SVM” yields highest accuracy compared to other two

COMPARISON OF ACCURACIES IN PROPOSED AND RELATED WORK

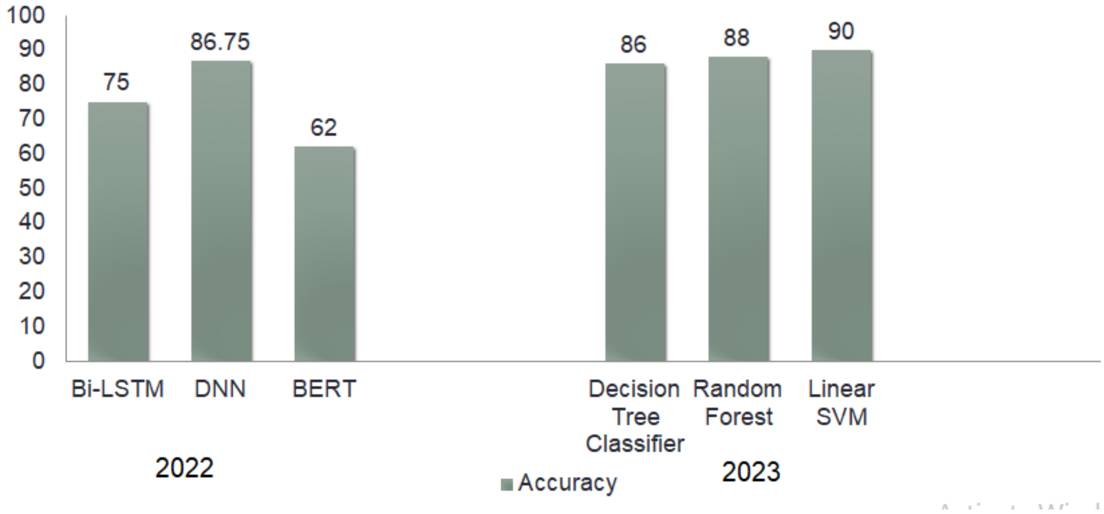

Figure 7.9 Comparison of accuracies in proposed and related work

Based on the above performance analysis, it is inferred that Linear Support Vector Machine yields highest accuracy. Other two algorithms has also yielded greater accuracy. In 2022 research papers, the algorithms like Bi-LSTM, DNN, BERT yielded 75%, 86.75%,62% accuracy respectively. From this, it is concluded that the proposed work has yielded higher accuracies compared to it's related work and all the three algorithms used in proposed work is greater than 85%.

CHAPTER 8
CONCLUSION

CHAPTER-8

CONCLUSION

8.1. CONCLUSION AND FUTURE ENHANCEMENTS

Various approaches have been proposed to tackle this problem, including rule-based systems, machine learning methods, and deep learning models. The performance of these models largely depends on the quality and size of the training data, feature engineering, and the choice of the classification algorithm. Nevertheless, recent studies have shown promising results, and text-based emotion recognition has practical applications in various domains, including social media monitoring, mental health diagnosis, and customer service. Furthermore, the development of cross-lingual and cross-cultural emotion recognition models can enable these systems to be used in multilingual and multicultural environments. Finally, the ethical implications of using emotion recognition technology must be carefully considered, and privacy concerns and biases in training data must be addressed to ensure that these systems are developed and used responsibly.

APPENDICES

A.SAMPLE DATASET

TEXT	EMOTION
i can go from feeling so hopeless to so damned hopeful just from being around someone who cares and is awake	Sadness
im grabbing a minute to post i feel greedy wrong	Anger
i am ever feeling nostalgic about the fireplace i will know that it is still on the property	Love
i am feeling grouchy	Anger
ive been feeling a little burdened lately wasnt sure why that was	Sadness
ive been taking or milligrams or times recommended amount and ive fallen asleep a lot faster but i also feel like so funny	Surprise
i feel as confused about life as a teenager or as jaded as a year old man	Fear
i have been with petronas for years i feel that petronas has performed well and made a huge profit	Happy
i feel romantic too	Love
i feel like i have to make the suffering i m seeing mean something	Sadness
i do feel that running is a divine experience and that i can expect to have some type of spiritual encounter	Happy
i think it s the easiest time of year to feel dissatisfied	Anger
i feel low energy i m just thirsty	Sadness

TEXT	EMOTION
i have immense sympathy with the general point but as a possible proto writer trying to find time to write in the corners of life and with no sign of an agent let alone a publishing contract this feels a little precious	happy
i do not feel reassured anxiety is on each side	happy
i didnt really feel that embarrassed	sadness
i feel pretty pathetic most of the time	sadness
i started feeling sentimental about dolls i had as a child and so began a collection of vintage barbie dolls from the sixties	sadness
i now feel compromised and skeptical of the value of every unit of work i put in	fear
i feel irritated and rejected without anyone doing anything or saying anything	anger
i am feeling completely overwhelmed i have two strategies that help me to feel grounded pour my heart out in my journal in the form of a letter to god and then end with a list of five things i am most grateful for	fear
i have the feeling she was amused and delighted	happy
i was able to help chai lifeline with your support and encouragement is a great feeling and i am so glad you were able to help me	happy

TEXT	EMOTION
i already feel like i fucked up though because i dont usually eat at all in the morning	Anger
i still love my so and wish the best for him i can no longer tolerate the effect that bm has on our lives and the fact that is has turned my so into a bitter angry person who is not always particularly kind to the people around him when he is feeling stressed	sadness
i feel so inhibited in someone elses kitchen like im painting on someone elses picture	sadness
i become overwhelmed and feel defeated	sadness
i feel kinda appalled that she feels like she needs to explain in wide and lenghth her body measures etc pp	anger
i feel more superior dead chicken or grieving child	happy
i get giddy over feeling elegant in a perfectly fitted pencil skirt	happy
i remember feeling acutely distressed for a few days	fear
i have seen heard and read over the past couple of days i am left feeling impressed by more than a few companies	surprise
i climbed the hill feeling frustrated that id pretty much paced entirely wrong for this course and that a factor that has never ever hampered me had made such a dent in the day	anger

TEXT	EMOTION
i can t imagine a real life scenario where i would be emotionally connected enough with someone to feel totally accepted and safe where it it morally acceptable for me to have close and prolonged physical contact and where sex won t be expected subsequently	happy
i am not sure what would make me feel content if anything	happy
i have been feeling the need to be creative	happy
i do however want you to know that if something someone is causing you to feel less then your splendid self step away from them	happy
i feel a bit rude writing to an elderly gentleman to ask for gifts because i feel a bit greedy but what is christmas about if not mild greed	anger
i need you i need someone i need to be protected and feel safe i am small now i find myself in a season of no words	happy
i plan to share my everyday life stories traveling adventures inspirations and handmade creations with you and hope you will also feel inspired	happy
i already have my christmas trees up i got two and am feeling festive which i m sure is spurring me to get started on this book	happy

TEXT	EMOTION
ive worn it once on its own with a little concealer and for the days im feeling brave but dont want to be pale then its perfect	happy
i feel very strongly passionate about when some jerk off decides to poke and make fun of us	happy
i was feeling so discouraged we are already robbing peter to pay paul to get our cow this year but we cant afford to not get the cow this way	sadness
i was feeling listless from the need of new things something different	sadness
i lost my special mind but don t worry i m still sane i just wanted you to feel what i felt while reading this book i don t know how many times it was said that sam was special but i can guarantee you it was many more times than what i used in that paragraph did i tell you she was special	happy
i can t let go of that sad feeling that i want to be accepted here in this first home of mine	love
on a boat trip to denmark	happy
i stopped feeling cold and began feeling hot	anger
i need to feel the dough to make sure its just perfect	happy
i found myself feeling a little discouraged that morning	sadness
i feel selfish and spoiled	anger
i was stymied a little bit as i wrote feeling unsure that i might go somewhere with the story unintended	fear

TEXT	EMOTION
i bag qaf look who s cryin now jacynthe lookin good feelin gorgeous rupaul the skins scissor sisters valentine the sun fed up kayle who s your daddy gerling awake the unkind u	happy
i feel you know basically like a fake in the realm of science fiction	sadness
i hate living under my dads roof because it gives him an excuse to be an asshole to me because hes providing for me to live here i think he feels that he needs to make me feel as unwelcome as possible so ill leave	sadness
i keep feeling pleasantly surprised at his supportiveness and also his ease in new situations	surprise
i have this feeling that if i have anymore vigorous sexual activity in the coming yes i misspelt that as cumming days parts of me will begin to fall off	happy
i feel my mom s graceful warm loving smile as i rob the time to nurture myself and heal	happy
i feel in they talk the brother in law is extremely popular the one that had no me to think is so stiff	happy
i ate i could feel a gentle tingle throughout almost as if i was feeling the healing taking place at a cellular level	love
i feel like we are pressured into being young beautiful thin and depending on the trend having the girls rejuvenated or butt implants	fear

TEXT	EMOTION
i began having them several times a week feeling tortured by the hallucinations moving people and figures sounds and vibrations	Fear
i am now nearly finished the week detox and i feel amazing	surprise
i feel selfish as i read back to my former posts how i have never asked for prayers for others how i never considered that there may be others out there that deserve their prayers answered before my own	anger
i know the pain parents feel when an enraged child becomes violent	anger
i have been on a roller coaster of emotions over these supposed feelings that something unpleasant was coming	sadness
i suppose my own truth needs to be shared i havent been feeling very faithful lately ive dwelled more in doubt and uncertainty than i have in faith	Love
i was feeling brave when i bought it and clearly when i was doing my makeup	happy
i am feeling miserable but c i am also the proudest mum on earth	sadness
i figure my family loves us no matter what but around anyone else i feel embarrassed when michelle goes ballistic	sadness
i don t necessarily think f bombs and sex are necessary in all stories but i feel reassured when i see them in print journals	happy

TEXT	EMOTION
i can feel my ovaries aching talking to me as i like to put it	sadness
i didn t feel like doing much chris and i mostly just took too many pictures of unimportant stuff	sadness
im tired of the book and ready to have it out of here and finding out that i was given unsuitable images and then feeling blamed for the result did not sit well	sadness
i did successfully manage to stretch a mxm canvas i feel that this is an achievement in itself for me and was a worthwhile usage of my money and time i will use the canvas for future briefs	happy
i think feelings are one of nay the most important things we have	happy
i feel completely honored to be an influence to this young talented fully alive beautiful girl woman	happy
i feel angered and firey	anger
i feel like a miserable piece of garbage	sadness
i feel like i need to make a list leanne would be appalled at the thought so that i dont miss anything	anger
i drove dannika to school i was feeling a little bit rushed and this is what greeted me as i turned the corner	anger
i remember feeling so hellip furious with the shooter	anger
i feel very happy and excited since i learned so many things	happy

TEXT	EMOTION
i feel that at shows and around show horses people are trusting and relaxed because most show horses are safe and quiet and are handled frequently	happy
i only have a couple of things left to make and at the start of december i am done and feeling smug	happy
i think about how u could make me feel and realize that everything will be ok	happy
i feel so worthless during those times i was struggling finding work	sadness
i will be able to lay on my bed in the dark and not feel terrified at least for a while	fear
i was ready to meet mom in the airport and feel her ever supportive arms around me	love
im feeling bitter today my mood has been strange the entire day so i guess it's that	anger
my mums brother passed away after having been involved in a car accident	sadness
i am letting go of the animosity that is towards anyone that i feel has wronged me	anger
i talk to dogs as i feel they cannot understand words but they can read emotions and know how to be supportive i decided i should go home	love
i feel like throwing away the shitty piece of shit paper	sadness
im starting to feel wryly amused at the banal comedy of errors my life is turning into	happy
i find every body beautiful and only want people to feel vital in their bodies	happy
i hear are owners who feel victimized by their associations the associations attorneys or the property manager	sadness
i say goodbye to the fam theyre all sad a crying and i feel like a heartless bitch because hey im pretty excited to be flying for the first time and you know also to spend a year in another country	anger

Table A1 Sample Dataset

B1. SAMPLE CODING

6.1. MODULE – 1

DATA PREPROCESSING:

```
import warnings
warnings.filterwarnings('ignore')

import pandas as pd

#Read data
data = pd.read_csv('data.csv')
data
data.shape

#Drop null values
df = data.dropna()
df.shape
df.isnull().sum()
df.info()
df.columns

#Check duplicates
df.duplicated()
df.duplicated().sum()
df.text.unique()
df.emotion.unique()
df.emotion.value_counts()
df.head()

#Encoding
from sklearn.preprocessing import LabelEncoder
var_mod = ['text']
le = LabelEncoder()
for i in var_mod:
    df[i] = le.fit_transform(df[i]).astype(int)
df.head()
```

6.2. MODULE -2

DATA VISUALIZATION

```
#import library packages
import pandas as p
import matplotlib.pyplot as plt
import seaborn as s
import numpy as n
import warnings
warnings.filterwarnings("ignore")

#Load given dataset
data = p.read_csv('e.csv')
df=data.dropna()
df

df.columns
df.groupby('label').describe()


#plotting graph for distribution
import matplotlib.pyplot as plt
import seaborn as sns
sns.countplot(x ="label", data = df)
df.loc[:, 'label'].value_counts()
plt.title('EMOTION')
df['label'].unique()

#!pip install nltk
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
import re
import string
# remove whitespaces
df['text']=df['text'].str.strip()
# lowercase the text
df['text'] = df['text'].str.lower()
#remove punctuation
punc = string.punctuation
table = str.maketrans('','',punc)
df['text']=df['text'].apply(lambda x: x.translate(table))
# tokenizing each message
df['word_tokens']=df.apply(lambda x: x['text'].split(' '),axis=1)
# removing stopwords
df['text'] = df.apply(lambda x: [word for word in x['word_tokens'] if word notin
stopwords.words('english')],axis=1)
```

```
# stemming
ps = PorterStemmer()
df['stemmed']= df.apply(lambda x: [ps.stem(word) for word in x['text']],axis=1)
# remove single letter words
df['final_text'] = df.apply(lambda x: ' '.join([word for word in x['stemmed'] if
len(word)>1]),axis=1)

# Now we'll create a vocabulary for the training set with word count
from collections import defaultdict
vocab=defaultdict(int)
for text in df['final_text'].values:
for elem in text.split(' '):
        vocab[elem]+=1

print(vocab)

# divide the set in training and test
from sklearn.model_selection import train_test_split
X,X_test,y,y_test = train_test_split(df.loc[:,'text':],df['label'],test_size=0.2)

#!pip install wordcloud
from wordcloud import WordCloud

ham=' '.join(X.loc[y==’happy'final_text'].values)
ham_text = WordCloud(background_color='white',max_words=2000,width = 800, height =
800).generate(ham)

spam=' '.join(X.loc[y=='sadness','final_text'].values)
spam_text = WordCloud(background_color='black',max_words=2000,width = 800, height =
800).generate(spam)

plt.figure(figsize=[30,50])

plt.subplot(1,3,1)
plt.imshow(ham_text,interpolation='bilinear')
plt.title('')
plt.axis('off')

plt.subplot(1,3,2)
plt.imshow(spam_text, interpolation='bilinear')
plt.axis('off')
plt.title('')
```

6.3. MODULE 3 :

6.3.1. DECISION TREE CLASSIFIER ALGORITHM

```
#Import necessary libraries
import pandas as pd
import re
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer


# Load the dataset
df = pd.read_csv('data.csv')

# Preprocess the text
def preprocess_text(text):
    # Convert to lowercase
    text = text.lower()
    # Remove punctuations and digits
    text = re.sub(r'[^\w\s]', '', text)
    text = re.sub(r'\d+', '', text)
    # Remove stop words
    stop_words = stopwords.words('english')
    text = ' '.join([word for word in text.split() if word not in stop_words])
    # Lemmatize words
    lemmatizer = WordNetLemmatizer()
    text = ' '.join([lemmatizer.lemmatize(word) for word in text.split()])
    return text

df['text'] = df['text'].apply(preprocess_text)

from sklearn.feature_extraction.text import TfidfVectorizer

# Convert text to numerical vectors
tfidf = TfidfVectorizer()
X = tfidf.fit_transform(df['text'])
y = df['emotion']

from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a linear support vector classifier
model = DecisionTreeClassifier(max_depth=1000, criterion='entropy')
model.fit(X_train, y_train)
```

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Make predictions on the testing set
y_pred = model.predict(X_test)

# Calculate evaluation metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

print('Accuracy using Decision Tree classifier:', accuracy)
print('Precision using Decision Tree classifier:', precision)
print('Recall using Decision Tree classifier:', recall)
print('F1-score using Decision Tree classifier:', f1)

from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test,y_pred))

from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

6.3.2. RANDOM FOREST ALGORITHM

```
#Import necessary libraries
import pandas as pd
import re
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
# Load the dataset
df = pd.read_csv('data.csv')

# Preprocess the text
def preprocess_text(text):
   # Convert to lowercase
   text = text.lower()
   # Remove punctuations and digits
   text = re.sub(r'[^\w\s]', '', text)
   text = re.sub(r'\d+', '', text)
   # Remove stop words
   stop_words = stopwords.words('english')
   text = ' '.join([word for word in text.split() if word not in stop_words])
   # Lemmatize words
   lemmatizer = WordNetLemmatizer()
   text = ' '.join([lemmatizer.lemmatize(word) for word in text.split()])
   return text

df['text'] = df['text'].apply(preprocess_text)

from sklearn.feature_extraction.text import TfidfVectorizer

# Convert text to numerical vectors
tfidf = TfidfVectorizer()
X = tfidf.fit_transform(df['text'])
y = df['emotion']

from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a linear support vector classifier
model = RandomForestClassifier(n_estimators=100)
model.fit(X_train, y_train)

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Make predictions on the testing set
y_pred = model.predict(X_test)
```

```
# Calculate evaluation metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

print('Accuracy using Random Forest:', accuracy)
print('Precision using Random Forest:', precision)
print('Recall using Random Forest:', recall)
print('F1-score using Random Forest:', f1)

from sklearn.metrics import classification_report
print("Classification Report of Random Forest\n",classification_report(y_test,y_pred))

from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test,y_pred))
```

6.3.3. LINEAR SVM ALGORITHM

```
#import necessary libraries
import pandas as pd
import re
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer

# Load the dataset
df = pd.read_csv('data.csv')

# Preprocess the text
def preprocess_text(text):
    # Convert to lowercase
    text = text.lower()
    # Remove punctuations and digits
    text = re.sub(r'[^\w\s]', '', text)
    text = re.sub(r'\d+', '', text)
    # Remove stop words
    stop_words = stopwords.words('english')
    text = ' '.join([word for word in text.split() if word not in stop_words])
    # Lemmatize words
    lemmatizer = WordNetLemmatizer()
    text = ' '.join([lemmatizer.lemmatize(word) for word in text.split()])
    return text

df['text'] = df['text'].apply(preprocess_text)

from sklearn.feature_extraction.text import TfidfVectorizer

# Convert text to numerical vectors
tfidf = TfidfVectorizer()
X = tfidf.fit_transform(df['text'])
y = df['emotion']

rom sklearn.svm import LinearSVC
from sklearn.model_selection import train_test_split

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a linear support vector classifier
model = LinearSVC()
model.fit(X_train, y_train)

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Make predictions on the testing set
y_pred = model.predict(X_test)
```

```
# Calculate evaluation metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')
print('Accuracy using Linear SVM:', accuracy)
print('Precision using Linear SVM:', precision)
print('Recall using Linear SVM:', recall)
print('F1-score using Linear SVM:', f1)

#Save the model as a pickle file
import pickle
with open('model_lsvc.pkl', 'wb') as file:
    pickle.dump(model, file)
with open('vectorizer_lsvc.pkl', 'wb') as file:
    pickle.dump(tfidf, file)

from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test,y_pred))

from sklearn.metrics import classification_report
print("Classification Report of Linear SVM\n",classification_report(y_test,y_pred))
```

6.4. MODULE 4 :

DEPLOY

6.4.1. HTML CODE:

home.html

```
<!DOCTYPE html>
<html>
<head>
	<title>Text Emotion Recognition</title>
<!-- Load Bootstrap CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
	<style>
		body {
			background-color: #f2f2f2;
			font-family: Arial, sans-serif;
		}
		h1 {
			text-align: center;
			margin-top: 50px;
			margin-bottom: 30px;
			color: black
		font-family: "Times New Roman";font-size: 70px;
		}
		form {
			background-color: #fff;
			border-radius: 10px;
			padding: 30px;
			box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.2);
			margin: 0 auto;
			max-width: 600px;
		}
		label {
			font-weight: bold;
		}
		input[type="text"] {
			width: 100%;
			padding: 10px;
			margin-bottom: 20px;
			border: none;
			border-radius: 5px;
			box-shadow: 0px 0px 5px rgba(0, 0, 0, 0.2);
		}
		input[type="submit"] {
			background-color: #333;
			color: #fff;
```

```
                        border: none;
                        padding: 10px 20px;
                        border-radius: 5px;
                        cursor: pointer;
                        transition: all 0.3s ease-in-out;
                }
                input[type="submit"]:hover {
                        background-color: #555;
                        transform: translateY(-2px);
                }
            #toggle{
                background-color: #333;
                        color: #fff;
                        border: none;
                        padding: 10px 20px;
                        border-radius: 5px;
                        cursor: pointer;
                        transition: all 0.3s ease-in-out;
            }
        </style>
</head>
<body>
<div class="container">
        <b><h1>Text Emotion Recognition</h1></b>
        <form action="/classify" method="post">
          <h6>Team ID - A1</h6>
          <h6>Team Members - Dhuneesha.E, Divya.S,Nivedha.U</h6>
          <h6>Panimalar Engineering College</h6>
                <label for="text">Enter your text:</label>
<br><br>
                <input type="text" id="text" name="text" placeholder="Type your text
here...">
<button type="button" id="toggle">Record</button>
           <button><input type="submit" value="Submit"></button>

        </form>
</div>
<!-- Load Bootstrap JS and jQuery -->
        <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
        <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
        <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
<script type="text/javascript" src="../static/script.js"></script>
</body>
</html>
```

result.html

```
<!DOCTYPE html>
<html>
<head>
	<title class="header">Text Emotion Recognition</title>
	<!-- Load Bootstrap CSS -->
	<link                                                                                rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
	<!-- Load custom CSS -->
	<style>
		body {
			background-color: #f2f2f2;
			font-family: Arial, sans-serif;
		}
		h1 {
			text-align: center;
			margin-top: 50px;
			margin-bottom: 30px;
			color: #333;
		}
	   p{
		font-size: 25px;
	   }
		h2 {
			margin-top: 30px;
			margin-bottom: 20px;
			color: black
		font-family: "Times New Roman";font-size: 100px;
		}
		.result {
			background-color: #fff;
			border-radius: 10px;
			padding: 30px;
			box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.2);
			margin: 0 auto;
			max-width: 600px;
		}
		.emotion {
			font-size: 24px;
			font-weight: bold;
			color: #333;
		}
		.confidence {
			margin-top: 20px;
			font-size: 16px;
			color: #555;
		}
		.btn {
			margin-top: 20px;
```

```
                padding: 10px 20px;
                border-radius: 5px;
                cursor: pointer;
                transition: all 0.3s ease-in-out;
            }
            .btn-primary {
                background-color: #333;
                border: none;
                color: #fff;
            }
            .btn-primary:hover {
                background-color: #555;
                transform: translateY(-2px);
            }
            .btn-secondary {
                background-color: #f2f2f2;
                border: none;
                color: #333;
            }
            .btn-secondary:hover {
                background-color: #ddd;
                transform: translateY(-2px);
            }
        </style>
</head>
<body>
<br>

<h1>Text Emotion Recognition</h1>
<div class="result">
<div class="container">

<h6>Team ID - A1</h6>
       <h6>Team Members - Dhuneesha.E, Divya.S,Nivedha.U</h6>
       <h6>Panimalar Engineering College</h6>
         <b><p>The predicted emotion is: {{ prediction }}</p></b>
         <button class='btn'><a href="/">Try again</a></button>
         </div>
</div>
</body>
</html>
```

6.4.2. JAVASCRIPT CODE

script.js

```
function init() {
  window.SpeechRecognition = window.SpeechRecognition ||
window.webkitSpeechRecognition;
  if (('SpeechRecognition' in window || 'webkitSpeechRecognition' in window)) {
    let speech = {
      enabled: true,
      listening: false,
      recognition: new window.SpeechRecognition(),
      text: ''
    }
    speech.recognition.continuous = true;
    speech.recognition.interimResults = true;
    speech.recognition.lang = 'en-US';
    speech.recognition.addEventListener('result', (event) => {
      const audio = event.results[event.results.length - 1];
      speech.text = audio[0].transcript;
      const tag = document.activeElement.nodeName;
      if (tag === 'INPUT' || tag === 'TEXTAREA') {
        if (audio.isFinal) {
          document.activeElement.value += speech.text;
        }
      }
      result.innerText = speech.text;
    });

    toggle.addEventListener('click', () => {
      speech.listening = !speech.listening;
      if (speech.listening) {
        toggle.classList.add('listening');
        toggle.innerText = 'Listening ...';
        speech.recognition.start();
      }
      else {
        toggle.classList.remove('listening');
        toggle.innerText = 'Toggle listening';
        speech.recognition.stop();
      }
    })
  }
}
init();
```

6.4.3. FLASK CODE:

```
from flask import Flask, request, render_template
import pandas as pd
import string
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import CountVectorizer
import pickle

# Load the pre-trained model and vectorizer
with open('model_lsvc.pkl', 'rb') as file:
    model = pickle.load(file)

with open('vectorizer_lsvc.pkl', 'rb') as file:
    vectorizer = pickle.load(file)

app = Flask(__name__)

# Preprocess the input text
def preprocess_text(text):
    # Convert to lowercase
    text = text.lower()
    # Remove punctuation
    text = text.translate(str.maketrans('', '', string.punctuation))
    # Remove stop words
    stop_words = set(stopwords.words('english'))
    word_tokens = word_tokenize(text)
    text = ' '.join([w for w in word_tokens if not w in stop_words])
    return text
# Render the homepage
@app.route('/')
def home():
    return render_template('home.html')

# Classify the input text
@app.route('/classify', methods=['POST'])
def classify():
    text = request.form['text']
    # Preprocess the input text
    text = preprocess_text(text)
    # Vectorize the input text
    features = vectorizer.transform([text])
    # Predict the emotion of the input text
    prediction = model.predict(features)[0]
    return render_template('result.html', prediction=prediction)

if __name__ == '__main__':
    app.run(debug=True,use_reloader=False)
```

B2. SAMPLE SCREENS

```
Out[14]: array(['sadness', 'anger', 'love', 'surprise', 'fear', 'happy'],
               dtype=object)

In [15]: df.emotion.value_counts()

Out[15]: happy       7029
         sadness     6265
         anger       2993
         fear        2652
         love        1641
         surprise     879
         Name: emotion, dtype: int64

In [16]: df.columns

Out[16]: Index(['text', 'emotion'], dtype='object')
```

Before LabelEncoder

```
In [17]: df.head()
```


Figure B2.1 Data pre-processing coding

```
In [7]: #plotting graph for distribution
        import matplotlib.pyplot as plt
        import seaborn as sns
        sns.countplot(x = "emotion", data = df)
        df.loc[:, 'emotion'].value_counts()
        plt.title('EMOTION')

Out[7]: Text(0.5, 1.0, 'EMOTION')
```

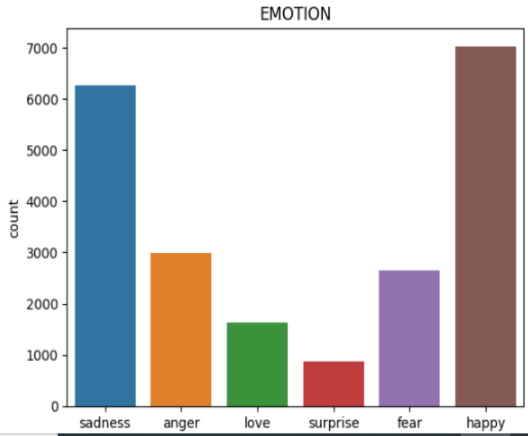

Figure B2.2 Data visualization coding

```
In [4]: from sklearn.tree import DecisionTreeClassifier
        from sklearn.model_selection import train_test_split

        # Split data into training and testing sets
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

        # Train a linear support vector classifier
        model = DecisionTreeClassifier(max_depth=1000, criterion='entropy')
        model.fit(X_train, y_train)

Out[4]: ▾ DecisionTreeClassifier
        DecisionTreeClassifier(criterion='entropy', max_depth=1000)

In [5]: from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

        # Make predictions on the testing set
        y_pred = model.predict(X_test)
```


Figure B2.3 Decision tree coding

```
In [4]: from sklearn.ensemble import RandomForestClassifier
        from sklearn.model_selection import train_test_split

        # Split data into training and testing sets
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

        # Train a linear support vector classifier
        model = RandomForestClassifier(n_estimators=100)
        model.fit(X_train, y_train)

Out[4]: ▾ RandomForestClassifier
        RandomForestClassifier()

In [5]: from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

        # Make predictions on the testing set
        y_pred = model.predict(X_test)
```


Figure B2.4 Random Forest coding

```
In [4]: from sklearn.svm import LinearSVC
        from sklearn.model_selection import train_test_split

        # Split data into training and testing sets
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

        # Train a linear support vector classifier
        model = LinearSVC()
        model.fit(X_train, y_train)

Out[4]: ▾ LinearSVC
        LinearSVC()

In [5]: from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

        # Make predictions on the testing set
        y_pred = model.predict(X_test)
```


Figure B2.5 Linear SVM coding

REFERENCES

[1]Lea Canales; Walter Daelemans; Ester Boldrini; Patricio Martínez-Barco, "EmoLabel: Semi-Automatic Methodology for Emotion Annotation of Social Media Text" in IEEE Transactions on Affective Computing (Volume: 13, Issue: 2, 01 April-June 2022)

[2] Jennifer Santoso; Takeshi Yamada; Kenkichi Ishizuka; Taiichi Hashimoto; Shoji Makino, "Speech Emotion Recognition Based on Self-Attention Weight Correction for Acoustic and Text Features" in IEEE Access (Volume: 10) **,**November 2022

[3] Sudarshan Pant; Eunchae Lim; Hyung-Jeong Yang; Guee-Sang Lee; Soo-Hyung Kim, "Korean Drama Scene Transcript Dataset for Emotion Recognition in Conversations" in IEEE Access (Volume: 10), November 2022

[4] Jing Zhang, Xinyu Liu, Zhe Wang, Hai Yang, "Graph-Based Object Semantic Refinement for Visual Emotion Recognition", in IEEE Transactions on Circuits and Systems for Video Technology (Volume: 32, Issue: 5, May 2022), July 2021

[5] Namrata Chaudhari; Reshu Agarwal; Meghna Narwade; Xinrui Yu; Jafar Saniie, "Artificial Intelligence System for Emotion Recognition and Text Analytics" in IEEE Xplore, July 2022

[6] Samuel Kakuba; Alwin Poulose; Dong Seog Han, "Deep Learning-Based Speech Emotion Recognition Using Multi-Level Fusion of Concurrent Features" in : IEEE Access (Volume: 10), November 2022

[7] Samuel W. K. Chan, "Multilabel Emotion Tagging for Domain-Specific Texts", in IEEE Transactions on Computational Social Systems (Volume: 9, Issue: 4, August 2022), November 2021

[8] N.Susithra, K.Rajalaksmi, P.Ashwath, B.Ajay, D. Rohit, S. Stewaugh, "Speech based Emotion Recognition and Gender Identification using FNN and CNN Models" in IEEE Access, July 2022

[9] Meenu S Nair; Deepa P Gopinath, "Transfer learning for Speech Based Emotion Recognition" in 2022 IEEE Access, May 2022

[10] Abhishek Kumar; Amit Kumar Goel, "Speech Emotion Recognition by using Feature Selection and Extraction", in IEEE Access, June 2022