Major Project Report on

EMOTION RECOGNITION FROM TEXT STORIES USING EMOTION-EMBEDDING MODEL

Submitted in partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted By

PEETHANI NAGASREE	**198R1A0401**
AMBALA NARESH	**198R1A0402**
TERATI ANAND	**198R1A0403**
ANKITA	**198R1A0404**

Under the Guidance of

Dr.S. POONGODI

Professor

ECE DEPARTMENT

CMR
ENGINEERING COLLEGE
EXPLORE TO INVENT

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

(Approved by AICTE, Affiliated to JNTUH, Accredited by NBA, NAAC)

Kandlakoya (v), Medchal, Telangana.

2022-23

CMR ENGINEERING COLLEGE

(Approved by AICTE, Affiliated to JNTUH, Accredited by NBA, NAAC)

Kandlakoya (v), Medchal, Telangana.

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the Major-Project work entitled "**EMOTION RECOGNITION FROM TEXT STORIES USING EMOTION EMBEDDING MODEL**" is being submitted by **PEETHANI NAGASREE** bearing Roll No: **198R1A0401**, **AMBALA NARESH** bearing Roll No:**198R1A0402**, **TERATI ANAND** bearing Roll No:**198R1A0403**, **ANKITA** bearing Roll No:**198R1A0404** in BTECH IV-II semester, Electronics and Communication Engineering is a record bonafide work carried out by them during the academic year 2022-2023.The results embodied in this report have not been submitted to any other University for the award of any degree.

INTERNAL GUIDE	HEAD OF THE DEPARTMENT
Dr.S. POONGODI	**Dr. SUMAN MISHRA**

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We sincerely thank the management of our college **CMR ENGINEERING COLLEGE** for providing during our project work.

We derive great pleasure in expressing our sincere gratitude to our principal **Dr. A. S. REDDY** for his timely suggestions, which helped us to complete the project work successfully.

It is a very auspicious moment we would like to express our gratitude to **Dr. SUMAN MISHRA**, Head of the Department, ECE for his consistent encouragement during the progress of this project.

We take it as a privilege to thank our project coordinator **Dr. T. SATYANARAYANA,**Associate Professor, Department of ECE for his ideas that led to completing the project work and we also thank his continuous guidance, support, and unfailing patience, throughout the course of this work.

We sincerely thank our internal project guide **Dr.S. POONGODI,** Professor, Department of ECE for her guidance and encouragement in carrying out this project.

.

DECLARATION

We hereby declare that the project work entitled **“EMOTION RECOGNITION FROM TEXT STORIES USING EMOTION EMBEDDING MODEL”** is the work done by us in campus at **CMR ENGINEERING COLLEGE,** Kandlakoya during the academic year 2022-23 and is submitted as Major Project in partial fulfilment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY** in **ELECTRONICS AND COMMUNICATION ENGINEERING** from **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD.**

PEETHANI NAGASREE	**198R1A0401**
AMBALA NARESH	**198R1A0402**
TERATI ANAND	**198R1A0403**
ANKITA	**198R1A0404**

CONTENTS

CHAPTERS **Page.No**

LIST OF FIGURES **i**

LIST OF TABLES **ii**

LIST OF ABBREVATION **iii**

CHAPTER 1 INTRODUCTION **1**

1.1 OVERVIEW OF PROJECT 1

1.2 OBJECTIVE OF EMOTION RECOGNITION 2

CHAPTER 2 LITERATURE SURVEY **5**

2.1 LITERATURE REVIEW 5

CHAPTER 3 EXISTING SYSTEM **9**

3.1 INTRODUCTION EXISTING SYSTEM 9

3.2 REQUIREMENTS SPECIFICATION 9

3.3 ADVANTAGES 10

3.4 DISADVANTAGES 10

CHAPTER 4 INTRODUCTION TO PYTHON **15**

4.1 INTRODUCTION TO PYTHON 15

4.2 PYTHON IN EMOTION RECOGNITION 15

4.3 MACHINE LEARNING 15

CHAPTER 5 PROPOSED SYSTEM **16**

5.1 OVERVIEW OF PROPOSED SYSTEM 16

5.2 SUPPORT VECTOR MACHINE 17

5.3 LOGISTIC REGRESSION 19

5.4 DECISION TREE 21

5.5 XGBOOST REGRESSION 25

5.6 NAIVE BYTES 26

5.7 BAYES THEOREM 28

5.8 RANDOM FOREST ALGORITHM 30

CHAPTER 6 SIMULATION RESULTS **33**

CHAPTER 7 CONCLUSION **37**

CHAPTER 8 REFERENCES **38**

ANNEXURE **41**

LIST OF FIGURES

Fig No.	NAME OF THE FIGURE	PAGE-NO
5.1	Block diagram of the proposed system	16
5.2	Support Vectors	18
5.3	Graph of Logistic Regression	20
5.4	Decision Tree	21
5.5	Working of RF Algorithm	30
6.1	Input data for Happy	33
6.2	Output data for Happy	33
6.3	Input data for unhappy	33
6.4	Output data for unhappy	34
6.5	View in Trained and Tested in Barchart	34
6.6	Predict Emotion from Dataset Details	35
6.7	Emotion Prediction Ratio of Dataset	35
6.8	Line Chart of Emotion Prediction	36

LIST OF TABLES

Table No.	NAME OF THE TABLE	PAGE-NO
5.1	Dataset Specification	17
5.7	Dataset	26
5.8	Weather Dataset	29

LIST OF ABBREVIATIONS

ACRONYMS		ABBREVIATIONS
SVM	-	SUPPORT VECTOR MACHINE
NLP	-	NATURAL LANGUAGE PROCESSING
LR	-	LOGISTIC REGRESSION
SGD	-	STOCHASTIC GRADIENT DESCENT
TF-IDF	-	TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY
VC	-	VOTING CLASSIFIER
RMSE	-	ROOT MEAN SQUARED ERROR
MAE	-	MEAN SQUARED ERROR
ML	-	MACHINE LEARNING
GPL	-	GENERAL PUBLIC LICENS

ABSTRACT

Emotion can be expressed in many ways, such as facial expressions, gestures, speech, and written text. Emotion Recognition in text documents is essentially a content-based classification problem involving concepts from the domains of Natural Language Processing (NLP) as well as Machine Learning. Sentiment analysis is a natural language processing technique used to analyze the sentiment of a piece of text, whether it is positive, negative, or neutral. In thisproject, We propose a new method for emotion recognition from text stories using an emotion-embedding model. We introduce an emotion-embedding model that learns to encode emotions asdense vector representations. By training this model on a large corpus of text stories, it learns to map different emotions to distinct regions of the embedding space. To recognize emotions in unseen text stories, employ a transfer learning strategy. The fine-tune the emotion-embedding model on a small, labeled dataset specific to the target domain. This fine-tuning process adapts the model to the emotional characteristics of the domain, enabling it to accurately recognize emotions in the target text stories. To evaluate the effectiveness of our approach, project is on a diverse set of text story datasets. Our results demonstrate that the proposed emotion-embedding model achieves competitive performance compared to existing supervised methods, even when trained on limited labeled data. Our approach presents a one method for emotion recognition from text stories using an emotion-embedding model. The proposed method has the potential to enhance various NLP applications, including sentiment analysis, chatbots, and recommendation systems, by enabling a more accurate understanding of human emotions from the text. Twitter is being used to collect views about products, trends, and politics. Sentiment analysis is a techniqueused to analyze the attitude, emotions, and opinions of different people towards anything, and it can be carried out on tweets to analyze public opinion on news, policies, social movements, and personalities. By employing Machine Learning models, opinion mining can be performed without reading tweets manually. Their results could assist governments and businesses in rolling out policies, products, and events. Machine Learning models are implemented for emotion recognition by classifying tweets as happy or unhappy. The work has been implemented using Python (Open-Source Computer Vision Library (OpenCV) and NumPy. The Text stories (testing dataset) are being compared to the training dataset and thus emotion is predicted. The objective of this project is to develop a system which can analyze and predict emotion. The project proves that this procedure is workable and produces valid results.

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

Automatic emotion recognition, pattern recognition, and computer vision have become significantly important in Artificial Intelligence lately with applications in a wide range of areas. Recently, social media platforms such as Twitter have generated enormous amounts of structured, unstructured, and semi-structured data. One of the most recent examples is COVID-19 infodemic which shows misinformation in social media can be far more important and devastating than a disaster such as a pandemic. There is a need to analyze to accurately assign sentiment classes on a large scale. To perform such tasks, accurate NLP techniques and machinelearning (ML) models for text classification are required. Efficient methods are important to automatically label text data due to its noisy nature. In the past many studies have been performed on Twitter sentiment classification. Twitter is very fast and an efficient micro-blogging examination that facilitates the end users to transmit small posts are said to be tweets. Twitter is a highly demanding app in the world and is a successful platform in social media. Free accountscan be created by using Twitter that can provide an enormous audience potential. With the purpose of business and marketing, Twitter can be proved as the best platform, through which one can get in touch with very rich and famous personalities like stars and celebrities, so their purchasing can be very charming for them as well as for advertisers. Using Twitter, every celebrity is linked with fans and grants communication to followers. Such a platform is one of the superlative approaches for lovers as well. But it has a short note range; only 140 letters for each post and it can type a post or link on the website since it has no cost and open as the advertisements as well. There is no problem with clusters of personal ads which are like other social networking sites. It is quick because as a tweet is posted on Twitter, the public who is after respective business will get it without delay. Companies and advertisers can compose utilizationof this source to check the diverse operational point of views which are very considerable. With help of this, they will obtain an immediate response from their followers. Remarkably, a lot of businesses with the intention of purchasing Twitter followers increase their deals.

manufactured goods or examine the products presented and to get share in profit. It is extremely effortless to utilize as people can follow to get the news and updates, as organizations can tweet or re-tweet, they can mark favorite or selected people to send the tweets, also know how to propel the posts plus to be able to endow their money and instance through it. People love to express their feelings about a particular product on social networks like Twitter. Product owners are ready to spend more money on social media platforms to better advertise their products and generate more revenue. When a person shares experience about a product, it helps the owner to change their market strategy, and selling schemes, and improve the quality. Customer reviews serve as feedback to the owners or manufacturers too.

1.2 OBJECTIVE OF EMOTION RECOGNITION

The objective of the project is toanalyze emotions from story texts based on the emotional words representing each story sentence. Extracting emotional words from a text story dataset and detecting emotions using the proposedNatural language Processing (NLP). To accurately identify and understand the emotional contentof the text to improve human-machine interactions and to gain insights into human emotions andbehavior. Machine Learning models are implemented for emotion recognition by classifying tweets as happy or unhappy. With an in-depth comparative performance analysis, it was observed that the proposed voting classifier (LR-SGD) with TF-IDF producesthe most optimal result with79% accuracy and 81% F1 score.

1.3 MACHINE LEARNING ALGORITHMS:

1. **SUPPORT VECTOR MACHINE(SVM):** It is a linear model for classification and regression models. The Algorithm creates a line or Hyperplane which separates the data into classes. The objective of the support vector machine algorithm is to find a hyperplane in N- dimensional space (N- the number of features) that distinctly classifies the data points Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.

2. **LOGISTIC REGRESSION:** It is an algorithm that provides a linear relationship between independent variable and dependent variable to predict the outcome of future events. Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1". In the logistic model, the log odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence thelabeling; the function that converts log-odds to probability is the logistic function.

3. **BAYES' THEOREM:** Bayes' Theorem finds the probability of an event occurring and probability of another event that has already occurred. Bayes theorem is also known with some other names such as Bayes rule or Bayes Law. Bayes theorem helps to determine the probability of an event with random knowledge. It is used to calculate the probability of occurring one event while another one has already occurred. It is the best method to relate the condition probability and marginal probability.

4. **NAIVE BAYES:** Naive Bayes classifiers are a collection of classification algorithms based on **Bayes' Theorem**. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e., every pair of features being classified is independent of each other.

5. XGBOOST FOR REGRESSION

The results of the regression problems are continuous or real values. Some commonly used regression algorithms are Linear Regression and Decision Trees. There are several metrics involved in regression like root-mean-squared error (RMSE) and mean-squared-error (MAE).

Boost is a powerful approach for building supervised regression models. The validity of this statement can be inferred by knowing about its (Boost) objective function and base learners. The objective function contains a loss function and a regularization term. It tells about the difference between actual values and predicted values.

5. DECISION TREE: A Decision Tree is a Graph that uses a Branching Method to Illustrateevery possible Output for a specific input. There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problemis the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

- Decision Trees usually mimic human thinking ability while deciding, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree - like structure.

6. RANDOM FOREST: The Random Forest algorithm is a machine learning technique that belongs to the ensemble learning category. It is designed to solve both classification and regression problems by constructing a multitude of decision trees and combining their predictions to make more accurate and robust predictions. The algorithm works by creating a set of decisiontrees, where each tree is trained on a random subset of the training data and a random subset of the input features. During the training process, each tree independently learns to predict the targetvariable based on different subsets of the data and features, resulting in a diverse set of weak learners.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE REVIEW

We have researched the several literatures from various project related papers of emotion recognition from text stories and then have come up with some most important literature reviews.

- Sentiment analysis inspires corporations to define clients' preferences about products, services, and brands. Further, it plays an important role in interpreting information about industries and corporations to preserve them in makingentityreview. Sarlan et al. established a sentiment analysis through extracting number of tweets with the help of prototyping and the results organized customers views via tweets into positive and negative. Their research divided into two phrases. The first part is based on literature study which involves the Sentiment analysis techniques and methods that nowadays are used. In the second part, the application necessities and operations are described preceding to its development.

- Alsaeedi and Zubair Khan analyzed various kinds of sentiment analysis that is applied to Twitter dataset and its conclusions. The distinct approaches and conclusions of algorithm performance were compared. Methods were used which were supervised ML based, lexicon-based, ensemble methods. Authors used four methods that were Twitter sentiment Analysis using Supervised ML Approaches: Twitter sentiment Analysis using Ensemble Approaches. Twitter sentiment Analysis uses lexicon-based Approaches. Lexicon based approaches have been explored by many researchers for emotion classification.

- Bandhakavi et al. performed emotion-based feature extraction using domain specific lexicon generation. They captured associationof wordsand emotions usinga unigram mixture model. They used tweets that are weakly labelled to classify emotions. Their proposed architecture outperformed other state-of-the-heart approaches such as Latent Dirichlet Allocation and Point wise Mutual Information. Event related tweets are identified by researchers on geo related tweets. They used specific tweets of local festivities in one year.

- Alsinet et al. analyzed tweets from political domains. They claimed accepted tweets are stronger as compared to the rejected tweets. Rumor detection in tweets is performed by usingan encoder to analyze human behavior in comments.

- Xia et al. created the proportional training of the efficiency about collaborative method on behalf of Sentiment's arrangement. They set two types of features in the context of sentimentanalysis. Firstly, the feature set totally depends on the part of speech and word relation was depending on the feature set. Secondly, the following familiar text classification algorithms were maximum entropy, support vector machines and naive Bayes. Thirdly, the following ensemble strategies, that was the fixed combination, meta-classifier combination and weighted combination. They used 5 document-level datasets broadly utilized along with the arena of Sentiment's arrangement.
- Rustam et al. presented a Tweets Classification for US Airline Companies Sentiments. The researcher applied pre-processing on the dataset. The influence about feature extraction methods, together with TF, TF-IDF, along with word2vec, proceeding the classification accuracy has been examined. In addition, execution about the long short-term memory (LSTM) was studied in certain dataset. Paper of researcher proposes a Voting Classifier (VC) who helps to process similar administrations.
- Santos and Bayser examined a sentiment analysis of short texts. In the experiment, researchers suggested a profound convolution neural network that achieves from character tosentence level material to accomplish sentiment analysis of little texts.
- Mohamed evaluated a sentiment analysis of mining halal food consumers. This examinationfills this gap through the investigation of an irregular example of 100,000 tweets managing halal food. To lead the examination, a specialist predefined dictionary of seed descriptors was computerized presents on impart about halal food.
- Parveen and Pandey studied sentiment analysis on a Twitter dataset that uses the NB algorithm. Analysts use Hadoop Framework for preparing film informational collection which is reachable on Twitter site as reviews, input, and opinions. Sentiment analysis on Twitter data is explored in three classes that are positive, negative, and neutral.
- Alomari et al. analyzed SVM utilizing TF-IDF. The study presented the Arabic Jordanian Twitter corpus where Tweets are explained seeing that any positive or negative. It researcheddistinctive directed machine learning opinion examination classifiers when applied to Arabic client's online life of general subjects that are found in either Modern Standard Arabic (MSA) or Jordanian tongue. Analyses were conducted to assess the utilization of various weight plans, stemming and N-grams terms strategies and situations.

- Gamal et al. built a Twitter benchmark dataset for Arabic Sentiment Analysis. A benchmark Arabic dataset was suggested in an experiment for estimation investigation demonstrating social event strategy about the latest tweets in various Arabic vernaculars. The experiment dataset incorporates more than 151,000 unique assessments which are marked into two classes, negative and positive.

- Kumar and Garg explored the sentiment analysis of multimodal Twitter data. The experiment utilized a multi-method feeling examination approach to decide slant After pre-processing, the content module utilizes an AI-based troupe strategy gradient boosting to characterize tweets into extremity classifications, to be specific, positive, negative, or neutral High execution exactness of 91.32% is watched on behalf of arbitrary multi-method tweet dataset utilize to assess the planned model.

- Sailunaz investigated the feeling through the dataset that was analyzed bya sentiment analysis from Twitter texts. The objective of this work was to recognize and investigate the assessment and feeling communicated by individuals from content in their Twitter posts and to use them for creating suggestions. The dataset is utilized to recognize slant and feeling from tweets and their answers and estimated the impact scores of clients dependent on different Tweet based.

CHAPTER 3

EXISTING SYSTEM

3.1 INTRODUCTION TO EXISTING SYSTEM:

Emotion recognition from text has been a widely studied topic in the field of natural language processing. Several existing methods have been proposed to recognize emotions from text, but they often rely on labeled emotion datasets and supervised learning approaches. Natural Language Processing is a branch of AI that could understand text and predict emotion whereas Sentiment Analysis is a core part of NLP which identifies the sentiment behind the text. Deep learning systems: These systems use deep neural networks to learn the relationships between words and emotions in text. Dependency on labeled emotion datasets: Most existing systems for emotion recognition require a significant amount of labeled data, where each text sample is annotated with the corresponding emotion label. Creating large-scale labeled emotion datasets can be time-consuming, expensive, and may not cover the diversity of emotions in different domains. Limited domain specificity: Many existing systems are trained and evaluated on specific emotion datasets, which limits their ability to generalize to different domains. Emotionscan vary in different contexts, and relying on domain-specific labeled data may not capture the full spectrum of emotions in other domains. Lack of unsupervised learning, Existing systems primarily rely on supervised learning techniques, where the model is trained on labeled data. Unsupervised learning methods that can capture the underlying emotional content of text without explicit emotion labels are often overlooked. Unsupervised learning can be more scalable and adaptable to different domains. Insufficient transfer learning approaches: Transfer learning, which leverages knowledge from pre-trained models on large-scale datasets, is not extensively explored in existing systems for emotion recognition from text stories. Fine-tuning models on domain-specific data can lead to improved performance and better adaptation to specific emotional characteristics of different domains. Performance limitations: Due to the challenges, existing systems may face limitations in accurately recognizing and capturing the nuances of emotions in text stories. The reliance on labeled data, lack of domain adaptation, and limited generalization capabilities can result in suboptimal performance in real-world applications.

3.2 REQUIREMENT ANALYSIS:

The project involved analyzing the design of a few applications to make the application more user-friendly. To do so, it was important to keep the navigation from one screen to the other well-ordered and at the same time reduce the amount of typing the user needs to do.

Software Requirements

For developing the application, the following are the Software Requirements:

1. Python
2. Django
3. HTML, CSS, JavaScript
4. MySQL (WAMP Server)

Operating Systems supported

- Windows 10 64-bit OS

Technologies and Languages used to Develop.

- Python

Debugger and Emulator

- Any Browser (Particularly Chrome)

Hardware Requirements

For developing the application, the following are the Hardware Requirements:

- Processor: Pentium-IV
- RAM: 4 GB
- Space on Hard Disk: minimum 1 TB
- Hard Disk: 20GB

3.3 ADVANTAGES:

1) The proposed system presents a voting classifier (LR-SGD) and aims to estimate the performance of famous ML classifiers on Twitter datasets.

2) Data Visualization helps to understand the hidden patterns lying inside the dataset. It helps to qualitatively get more details about the dataset by visualizing the characteristics of the attributes.

3) Quality Control Emotion detection helps companies analyze customer experience so that you can know what elements of their product and service need attention and improvement.

3.4 DISADVANTAGES:

1) The existing model which is an ensemble of LR and SGD is not applied to both the dataset and the results.

2) A Voting Classifier (VC) is not cooperative learning which engages multiple individual classifiers.

3) Lack of Context: Incapability of Recognizing Sentences without Keywords.

CHAPTER 4

INTRODUCTION TO SOFTWARE

4.1 INTRODUCTION TO PYTHON

Python is a **high-level, interpreted**, **interactive,** and **object-oriented scripting language**. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted:** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is like PERL and PHP.
- **Python is Interactive:** You can sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented:** Python supports an Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language:** Python is a great language for beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

HISTORYOFPYTHON

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, Unix shell and other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

PythonFeatures

Python's features include:

- **Easy-to-learn:** Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read:** Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain:** Python's source code is fairly easy-to-maintain.
- **A broad standard library:** Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh
- **Interactive Mode:** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable:** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable:** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases:** Python provides interfaces to all major commercial databases.
- **GUI Programming:** Python supports GUI applications that can be created and ported to many system calls, libraries, and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable:** Python provides a better structure and support for large programs than shell scripting.

Python has a big list of good features:

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to bytecode for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.

4.2 PYTHON IN EMOTION RECOGNITION

Text to Emotion is the python package which will help you to extract the emotions from content. Processes any textual message and recognize the emotions embedded in it.

1. TEXT PRE-PROCESSING

At first, we have the major goal to perform data cleaning and make the content suitable for emotion.

- Remove the unwanted textual part from the message.
- Perform the natural language processing techniques.
- Bring out the well pre-processed text from the text pre-processing.

2. EMOTION INVESTIGATION

Detect emotion from every word that we got from pre-processed text and take a count of it .

- Find the appropriate words that express emotions or feelings.
- Check the emotion category of each word.
- Store the count of emotions relevant to the words found.

3. EMOTION ANALYSIS

After emotion investigation, there is the time of getting the significant output for the textual message.

- The output will be in the form of dictionary.
- There will be keys as emotion categories and values as emotion score.
- Higher the score of a particular emotion category, we can conclude that the message belongs to that category.

4.3 Machine Learning

Machine Learning (ML) is that field of computer science with the help of which computer systems can provide sense to data in much the same way as human beings do. In simple words, ML is a type of artificial intelligence that extracts patterns out ofraw data by using an algorithm or method. The focus of ML is to allow computer systems to learn from experiencewithout being explicitly programmed or human intervention.

- **Need for Machine Learning**

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate, and solve complex problems. On the other side, AI is still in its initial stage and hasn’t surpassed human intelligence in many aspects.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programming logic, in problems that cannot be programmed inherently. The fact is that we cant does without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

Lack of human expertise:

The very first scenario in which we want a machine to learn and take data-driven decisions can be the domain where there is a lack of human expertise. The examples can be navigations in unknown territories or spatial planets.

Difficulty in translating expertise into computational tasks:

There can be various domains in which humans have their expertise, however, they are unable to translate this expertise into computational tasks. In such circumstances we want machine learning. Examples can be the domains of speech recognition, cognitive tasks etc.

➢ Applications of Machines Learning:

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

1. Emotion analysis

2. Sentiment analysis

3. Error detection and prevention

4. Weather forecasting and prediction

5. Stock market analysis and forecasting

6. Speech synthesis

7. Speech recognition

8. Customer segmentation

CHAPTER 5

PROPOSED SYSTEM

5.1 INTRODUCTION TO PROPOSED SYSTEM:

To analyze emotions from story texts based on the emotional words representing each story sentence. Extracting emotional words from a text story dataset and detecting emotions using the proposed Natural language Processing (NLP). To accurately identify and understand the emotional content of text to improve human-machine interactions and to gain insights into human emotions and behavior. Machine Learning models are implemented for emotion recognition by classifying tweets as happy or unhappy which involves fundamental steps from Data collection to Evaluation by using training and testing set.

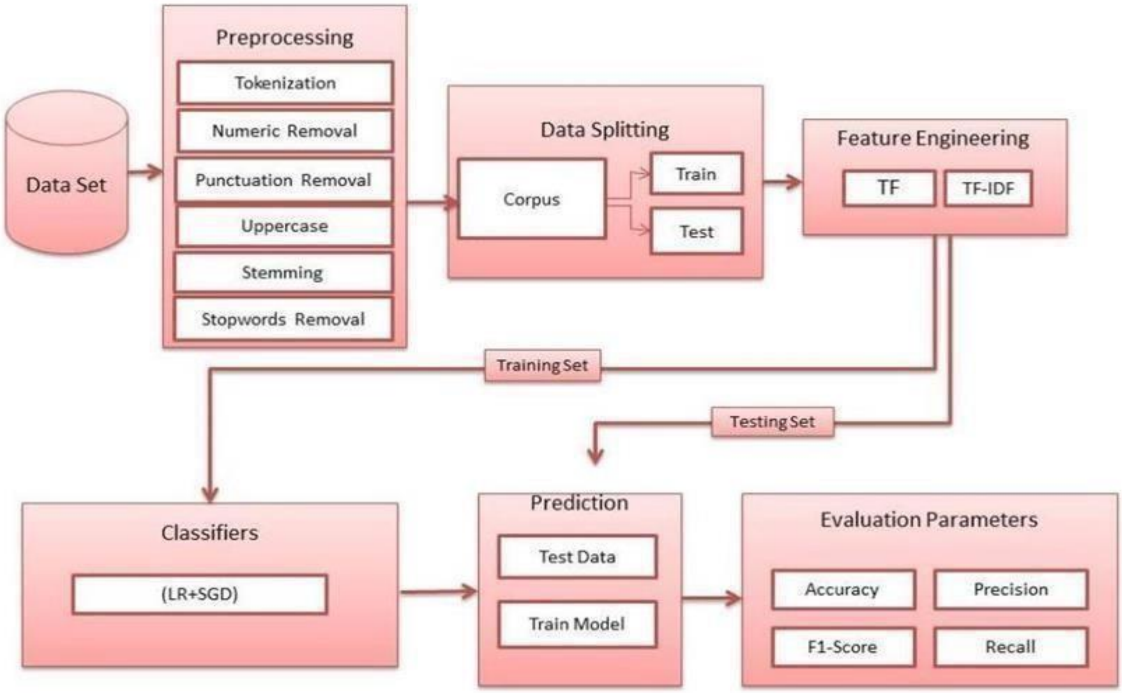


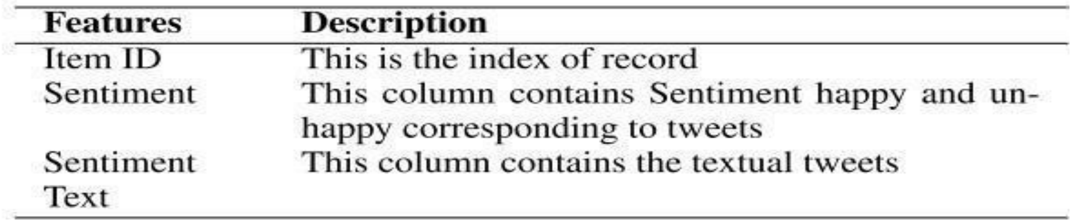
Fig 5.1 Block diagram of the proposed system

- **DATASET:** Dataset contains a lot of contrary tweets. The dataset is called “Sentiment Analysis on Twitter data" and contains 99989 records. Every record is labeled as happy and unhappy according to its sentimental polarity using symbols 1 and 0. Tweets which are in English are remembered for the finished dataset. The dataset contains different features.

Table 1 contains features and description of each feature.

Table 5.1 Dataset Specifications

TABLE 1. Dataset specifications.



Features	Description
Item ID	This is the index of record
Sentiment	This column contains Sentiment happy and unhappy corresponding to tweets
Sentiment Text	This column contains the textual tweets

- **DATA PRE-PROCESSING:** Datasets contain unnecessary data in raw form that can be unstructured or semi-structured. Such unnecessary data increases training time of the model and might degrade its performance. Pre-processing plays a vital role in improving the efficiency of ML models and saving computational resources. Text pre-processing boosts the prediction accuracy of the model. The following steps are performed in pre-processing; tokenization, case-conversion, stop words removal and removal of numbers.

- **DATA SPLITTING:** When performing emotion recognition from a text stories corpus, it is essential to split the data into appropriate subsets for training, validation, and testing.

- **FEATURE ENGINEERING:** After the data pre-processing step, the next essential step is the choice of features on a refined dataset. Supervised machine learning classifiers require textual data in vector form to get trained on it. The textual features are converted into vector form using TF and TF-IDF techniques in this work. Features extraction techniques not only convert textual features into vector form but also help to find significant features necessary to make predictions. For the most part all features do not contribute to the prediction of the target class. That is the reason feature extraction is the important part in the recognition of happy and unhappy related tweets.

  T F(t) = No. of times term t shows in a document/Total no. of terms inside document

The term frequency be frequently divided with the document length (the total number of terms in document)

- **CLASSIFIERS:** A Voting classifier with multiple parameters is used, that has used two individual classifiers that are LR and SGD and passes another parameter which is “voting” as “soft”. SGD is used to solve problems like redundancies in dataset and for big data. It performs classification by penalty and loss function. It is similar to gradient descent and looks at one sample for each step. On the other hand, LR calculates posterior probability p(Ct|v) by applying sigmoid function on input for binary classification.

- **PREDICTION:** Final prediction is the MaxProb (Avg− PosandAvg −Neg). An average probability is calculated for each class from the probability predicted by two classifiers. The decision function is then deciding the final class of the review which is based on the maximum average probability for a class. Use these embeddings as features and the labeled emotions as targets to train a classifier such as logistic regression, support vector machines (SVM), or a neural network. Experiment with different classifier models to find the best performance.

- **EVALUATION:** Evaluate the trained emotion recognition model on the testing dataset. Measure performance metrics such as accuracy, precision, recall, and F1 -score to assess how well the model recognizes emotions from text stories.

5.2 SUPPORT VECTOR MACHINE

- The objective of the support vector machine algorithm is to find a hyperplane in N-dimensional space (N — the number of features) that distinctly classifies the data points.
- Support vectors are data points that are closer to the hyperplane and influence the positionand orientation of the hyperplane. Using these support vectors, we maximize the marginof the classifier. Deleting the support vectors will change the position of the hyperplane.These are the points that help us build our SVM.
- Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane

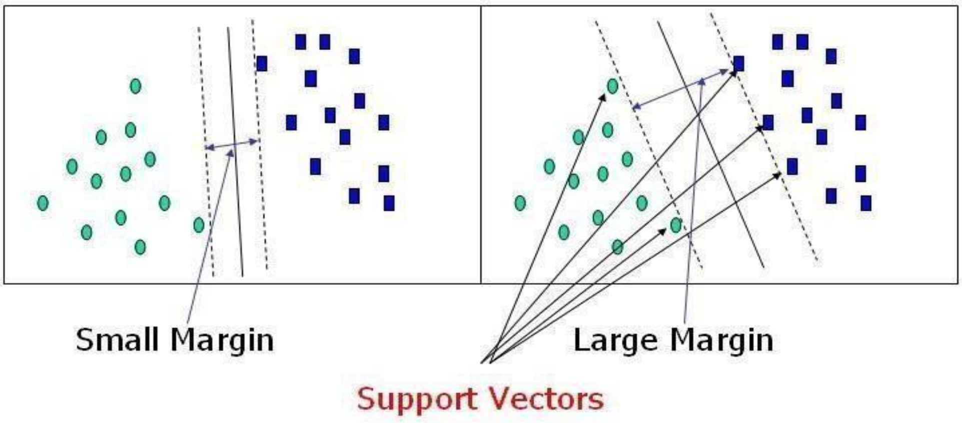


Fig 5.2 Support Vector Machine

Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM. In logistic regression, we take the output of the linear function and squash the value within the range of [0,1] using the sigmoid function. If the squashed value is greater than a threshold value(0.5) we assign it a label 1, else we assign it a label 0. In SVM, we take the output of the linearfunction and if that output is greater than 1, we identify it with one class and if the output is -1, we identify it with another class. Since the threshold values are changed to 1 and -1 in SVM, weobtain this reinforcement range of values ([-1,1]) which acts as margin.

Cost Function and Gradient Updates

In the SVM algorithm, we are looking to maximize the margin between the data points and the hyperplane. The loss function that helps maximize the margin is hinge loss.

$$(x, y, f^{(}x^{)}) = \{ \begin{array}{ll} 0 & if\ y*(x) \geq 1 \\ -y*(x), & else \end{array} \; 1$$ --------------------- (1)

$$(x, y, f(x)) = (1 - y*(x))$$ --------------------- (2)

Hinge loss function (function on left can be represented as a function on the right).

The cost is 0 if the predicted value and the actual value are of the same sign. If they are not, we then calculate the loss value. We also add a regularization parameter, the cost function. The objective of the regularization parameter is to balance the margin maximization and loss. After adding the regularization parameter, the cost functions look as below.

$$min_w \lambda \ ||w||^2 + \sum_{i=1}^{n} (1 - (x_i, w)) \qquad \text{-------------------}(3)$$

Gradient

When there is a misclassification, i.e., our model makes a mistake on the prediction of the class of our data point, we include the loss alongwith the regularization parameter to perform gradient update.

$$\omega = \omega + \alpha \,.\, (y_i \,.\, x_i - 2\lambda\omega) \qquad \text{----------------}(4)$$

5.3 LOGISTIC REGRESSION

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each bea binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labeling; the function that converts log-odds to probability is the logistic function, hence the name; the defining characteristic of the logistic model is that increasingone of the independent variablesmultiplicativelyscales the odds of the given outcome at a constant rate, with each independent variable having its own parameter; for a binary dependent variable this generalizes the odds ratio.

In a binary logistic regression model, the dependent variable has two levels (categorical). Outputs with more than two values are modeled by multinomial logistic regression and, if the multiple categories are ordered, by ordinal logistic regression though it can be used to make a classifier, for instance by choosing a cutoffvalue and classifying inputs with probabilitygreater thanthe cutoff as one class, belowthecutoff as theother; this is a common way to make a binary classifier. The logistic regressionas a general statistical model was originally developed and popularized primarily by JosephBerkson, beginning in Berkson (1944), where he coined "logit".

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be of a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.
- Logistic Regression is much like Linear Regression except for how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of somethingsuch as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it can provide probabilities and classify new data using continuous and discrete datasets.

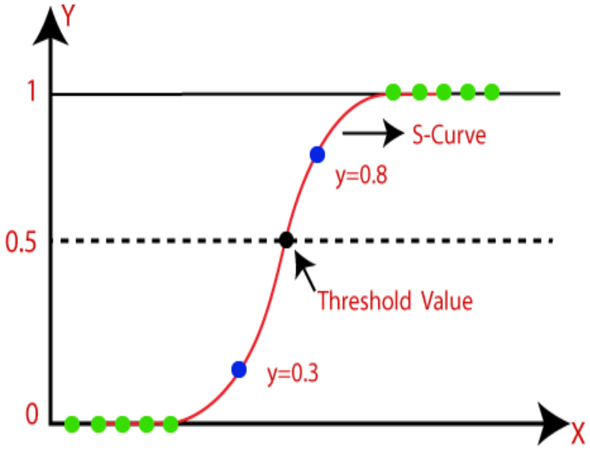


Fig 5.3 Graph of logistic regression

5.4 DECISION TREE

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

- Decision Trees usually mimic human thinking ability while deciding, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree -like structure.

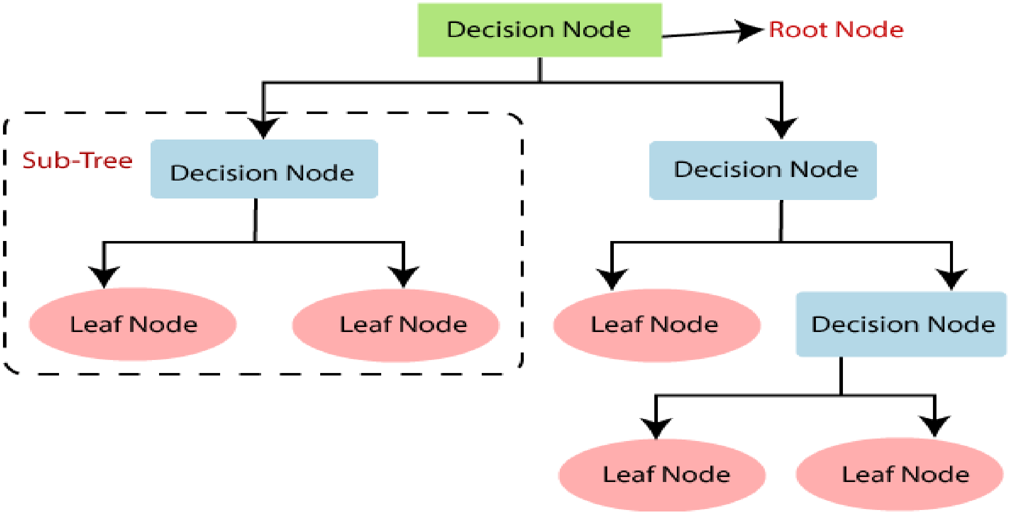


Fig 5.4 Structure of Decision Tree

Why use Decision Trees?

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.

Decision Tree Terminologies

□ **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

- **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

- **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

- **Branch/Sub Tree:** A tree formed by splitting the tree.

- **Pruning:** Pruning is the process of removing unwanted branches from the tree.

- **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

How does the Decision Tree algorithm Work?

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and based on the comparison, follows the branch, and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
- **Step-3:** Divide the S into subsets that contain possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.
- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node.

While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as **Attribute selection measure or ASM.** By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:

- **Information Gain**
- **Gini Index**

1. Information Gain:

- Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.
- It calculates how much information a feature provides us with about a class.
- According to the value of information gained, we split the node and build the decision tree.
- A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below.
- formula:

1. Information Gain= Entropy(S)- [(Weighted Avg) *Entropy (each feature)

Entropy: Entropy is a metric to measure the impurity in each attribute. It specifies randomness in data. Entropy can be calculated as:

Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)

Were,

- S= Total number of samples
- P(yes)= probability of yes
- P(no)= probability of no

2. **Gini Index**:

- The Gini index is a measure of impurity or purity used while creating a decision tree in the CART (Classification and Regression Tree) algorithm.
- An attribute with the low Gini index should be preferred as compared to the high Gini index.
- It only creates binarysplits, and the CART algorithm uses the Gini index to create binary splits.
- Gini index can be calculated using the below formula:

Gini Index= 1- $\sum_j P_j^2$

Advantages of the Decision Tree

- It is simple to understand as it follows the same process which a human follows while making any decision in real-life.
- It can be very useful for solving decision-related problems.
- It helps to think about all the possible outcomes for a problem.

Disadvantages of the Decision Tree

- The decision tree contains lots of layers, which makes it complex.
- It may have an overfitting issue, which can be resolved using the **Random Forest algorithm.**

Python Implementation of Decision Tree

Now we will implement the Decision Now we will implement the Decision tree using Python. For this, we will use the dataset "**user_data.csv**," which we have used in previous classification models. By using the same dataset, we can compare the Decision tree classifier with other classification models suchas KNN SVM, Logistic Regression, etc.

Steps will also remain the same, which are given below:

- Data Pre-processing step

- Fitting a Decision-Tree algorithm to the Training set
- Predicting the test result
- Test accuracy of the result (Creation of Confusion matrix)
- Visualizing the test set result.

5.5 XGBOOST FOR REGRESSION

The results of the regression problems are continuous or real values. Some commonly used regression algorithms are Linear Regression and Decision Trees. There are several metrics involved in regression like root-mean-squared error (RMSE) and mean-squared-error (MAE). These are some key members for XGBoost models, each plays their important roles.

- **RMSE:** It is the square root of mean squared error (MSE).
- **MAE:** It is an absolute sum of actual and predicted differences, but it lacks mathematically, that's why it is rarely used, as compared to other metrics.

XGBoost is a powerful approach for building supervised regression models. The validity of this statement can be inferred by knowing about its (XGBoost) objective function and base learners. The objective function contains a loss function and a regularization term. It tells about the difference between actual values and predicted values. The most common loss functions in XGBoost for regression problems is reg: linear, and that for binary classification is reg: logistics. Ensemble learning involves training and combining individual models to get a single prediction.

5.6 NAVIE BAYES

Naive Bayes classifiers are a collection of classification algorithms based on **Bayes' Theorem**. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e., every pair of features being classified is independent of each other.

To start with, let us consider a dataset.

Consider a fictional dataset that describes the weather conditions for playing a game of golf. Given the weather conditions, each tuple classifies the conditions as fit("Yes") or unfit("No") for playing golf.

Here is a tabular representation of our dataset.

Table 5.7 Dataset

	Outlook	Temperature	Humidity	Windy	Play Golf
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes

	Outlook	Temperature	Humidity	Windy	Play Golf
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	Yes
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	High	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

The dataset is divided into two parts, namely, **feature matrix** and the **response vector**.

- Feature matrix contains all the vectors(rows) of dataset in which each vector consists of the value of **dependent features**. In the above dataset, features are ‘Outlook’, ‘Temperature’, ‘Humidity’ and ‘Windy’.
- Response vector contains the value of **class variable** (prediction or output) for each row of feature matrix. In the above dataset, the class variable name is ‘Play golf’.

Assumption:

The fundamental Naive Bayes assumption is that each feature makes an:

- independent
- equal

contribution to the outcome.

With relation to our dataset, this concept can be understood as:

- We assume that no pair of features are dependent. For example, the temperature being 'Hot' has nothing to do with the humidity or the outlook being 'Rainy' has no effect on the winds. Hence, the features are assumed to be **independent**.
- Secondly, each feature is given the same weight (or importance). For example, knowing only temperature and humidity alone can't predict the outcome accurately. None of the attributes is irrelevant and assumed to be contributing **equally** to the outcome.

5.7 BAYES' THEOREM

Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \qquad \text{------------------ (1)}$$

where A and B are events and P(B)? 0.

- Basically, we are trying to find probability of event A, given event B is true. Event B is also termed as **evidence**.
- P(A) is the **priori** of A (the prior probability, i.e., Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance (here, it is event B).
- P(A|B) is a posteriori probability of B, i.e., probability of event after evidence is seen.

Now, with regards to our dataset, we can apply Bayes' theorem in following way:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)} \qquad \text{------------------------- (2)}$$

where, y is class variable and X is a dependent feature vector (of size n) where:

$$X = (x_1, x_2, x_3, \ldots \ldots, x_n) \qquad \text{-------------------- (3)}$$

Just to clear, an example of a feature vector and corresponding class variable can be: (refer 1st row of dataset)

X = (Rainy, Hot, High, False)

y = No

So basically, P(y|X) here means, the probability of "Not playing golf" given that the weather conditions are "Rainy outlook", "Temperature is hot", "high humidity" and "no wind".

Naive assumption

Now, it's time to put a naive assumption to the Bayes' theorem, which is, **independence** among the features. So now, we split **evidence** into independent parts.

Now, if any two events A and B are independent, then,P

(A, B) = P(A)P(B)

Hence, we reach to the result:

$$(y|x_1, \dots, x_n) = \frac{(x_1|y)(x_2|y)\dots P(x_n|y)P(y)}{(x_1)(x_2)\dots.(x_n)} \qquad \text{----------------} (4)$$

Now, we need to create a classifier model. For this, we find the probability of given set of inputs for all possible values of the class variable *y* and pick up the output with maximum probability. This can be expressed mathematically as:

$$y = argma(y)\pi \ {}_{i=1}{}^{n}p(x_i|y) \qquad \text{-----------------} (5)$$

So, finally, we are left with the task of calculating P(y) and P (x_i | y).

Please note that P(y) is also called **class probability** and P (x_i | y) is called **conditional probability**.

The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of P (x_i | y).

Let us try to apply the above formula manually on our weather dataset. For this, we need to do some precomputations on our dataset.

We need to find P (x_i | y_j) for each x_i in X and y_j in y. All these calculations have been demonstrated in the tables below:

Table 5.8 Weather dataset

Outlook

	Yes	No	P(yes)	P(no)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rainy	3	2	3/9	2/5
Total	9	5	100%	100%

Temperature

	Yes	No	P(yes)	P(no)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%

Humidity

	Yes	No	P(yes)	P(no)
Hot	3	4	3/9	4/5
Mild	6	1	6/9	1/5
Total	9	5	100%	100%

Play		P(Yes)/P(No)
Yes	9	9/14
No	5	5/14
Total	14	100%

5.8 RANDOM FOREST ALGORITHM

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, **"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:

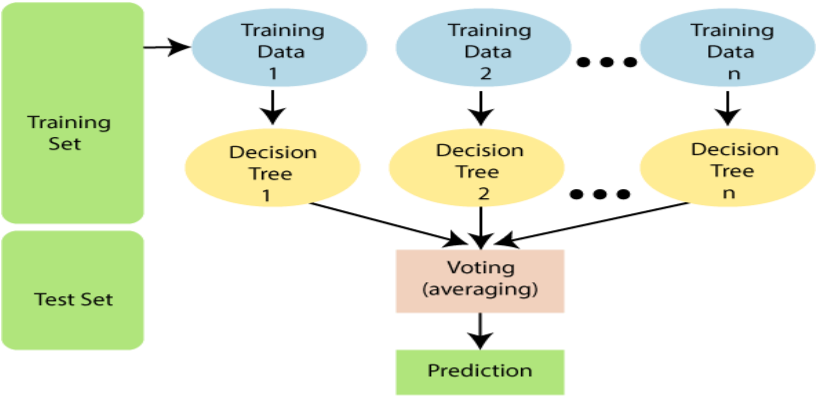


Fig 5.2 Working of Random Forest algorithm.

Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random Forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

Why use Random Forest?

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

How does Random Forest algorithm work?

Random Forest works in two-phases: first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Applications of Random Forest

There are mainly four sectors where Random Forest is mostly used:

1. **Banking:** Banking sector mostly uses this algorithm for the identification of loan risk.
2. **Medicine:** With the help of this algorithm, disease trends and risks of the disease can be identified.
3. **Land Use:** We can identify the areas of similar land use by this algorithm.
4. **Marketing:** Marketing trends can be identified using this algorithm.

Advantages of Random Forest

- Random Forest can perform both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

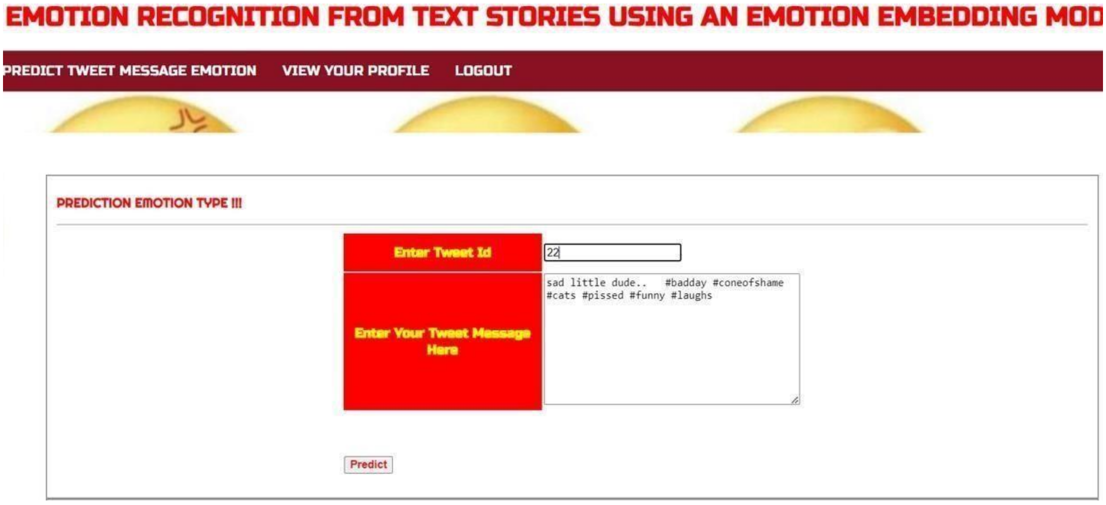
Disadvantages of Random Forest

- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.
- They may be overfitting for some noisy classification or regression tasks, as they rely on random sampling and splitting of the data.
- They can be computationally intensive and slow for large datasets, as they use multiple decision trees to make predictions and vote on the outcome.

CHAPTER 6

SIMULATION RESULTS

The result is obtained by using Python software which must be installed in our systems. Then, copy all the set of data collection in the form of files. Type the entire code in Python package and then run the code at command prompt, wait for the results and then it predicts the emotion whether it is happy or unhappy with help of algorithms where we use Logistic Regression and Stochastic Gradient Descent. From the below results, we can analyze the emotions through the given dataset.

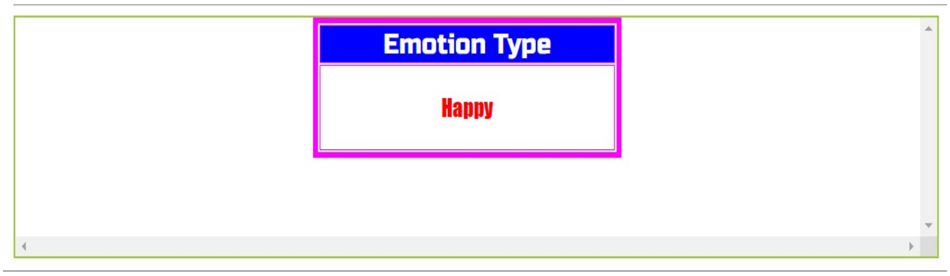

EMOTION RECOGNITION FROM TEXT STORIES USING AN EMOTION EMBEDDING MOD
PREDICT TWEET MESSAGE EMOTION
VIEW YOUR PROFILE
LOGOUT
PREDICTION EMOTION TYPE !!!
Predict

Enter Tweet Id: 22

Enter Your Tweet Message Here: sad little dude..   #badday #coneofshame #cats #pissed #funny #laughs

Fig 6.1 Input data for happy

From the Figure 6.1, we can predict the emotion from the given tweet. The data is taken from the dataset of tweets, we pick the required tweet and enter the page with the tweet id number.

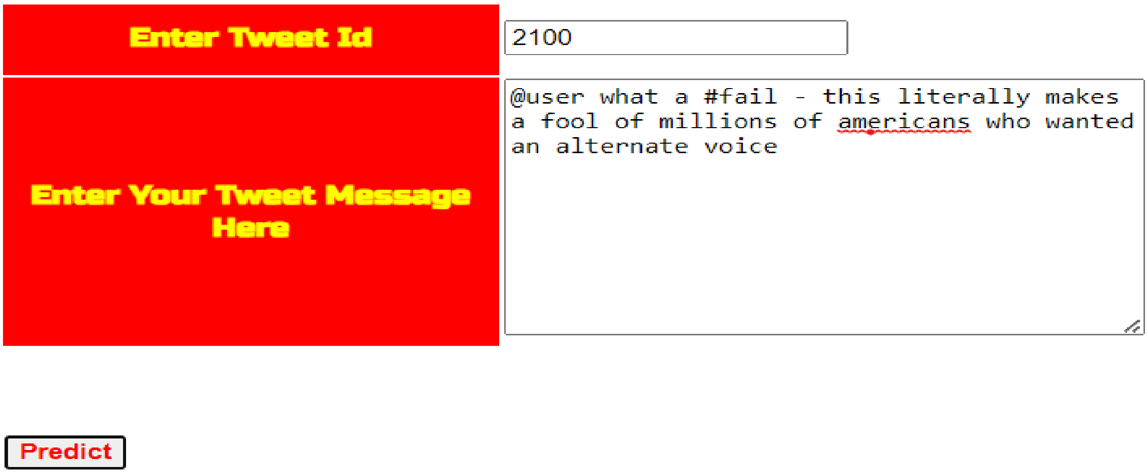


Emotion Type

Happy

Fig 6.2 Output data for happy

From the Figure 6.2, we can get the output of the tweet is shown as **Happy**. Thus, wecan predict the emotion type from the given data.

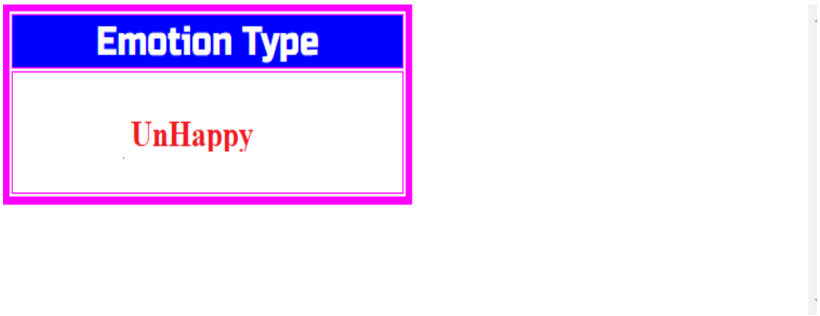


Enter Tweet Id: 2100

Enter Your Tweet Message Here: @user what a #fail - this literally makes a fool of millions of americans who wanted an alternate voice

Fig 6.3 Input data for unhappy

From the Figure 6.3, we can predict the emotion from the given tweet. The data is taken from the dataset of tweets, we pick the required tweet and enter the page with the tweet id number.



Emotion Type

UnHappy

Fig 6.4 Output data for unhappy

From the Figure 6.4, we can get the output of the tweet is shown **as unHappy**. Thus, we can predict the emotion type from the given data.

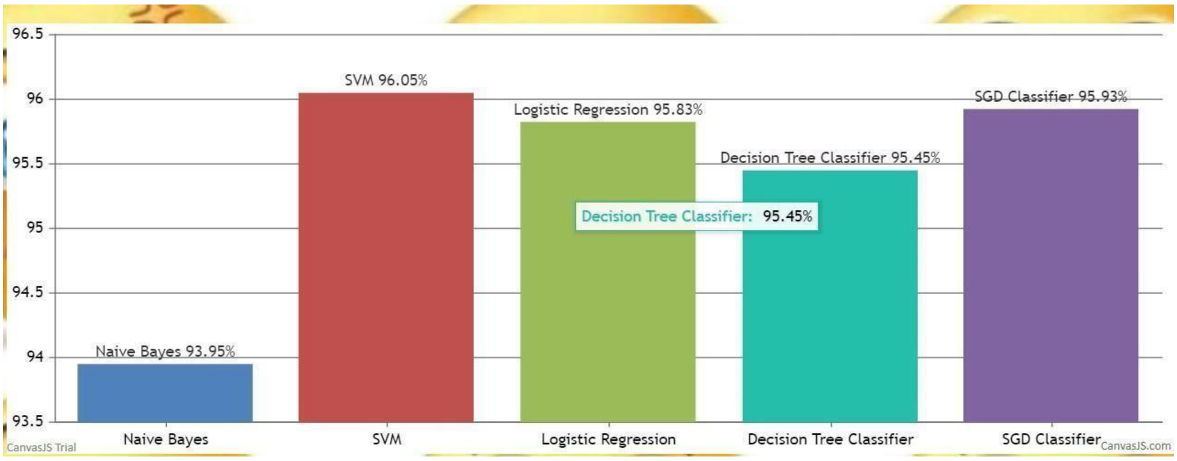
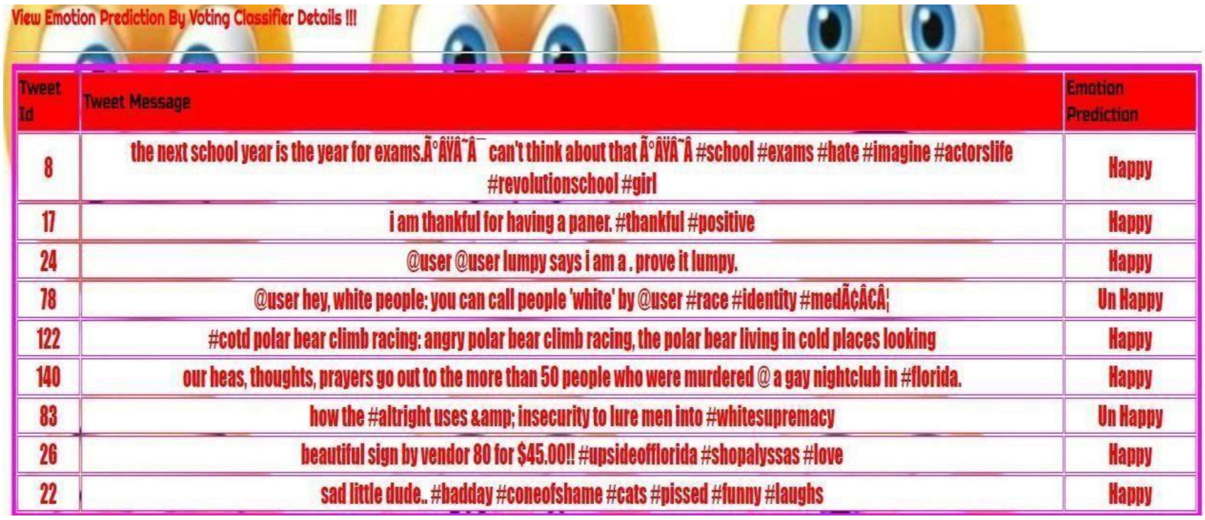


Fig 6.5 View in Trained and Tested in Barchart

From Figure 6.5, we can observe the view in trained and tested data by comparison between the machine learning algorithms where maximum percentage is **Support Vector Machine with 96.05% andminimum percentage is NaiveBayes with 93.95%**.

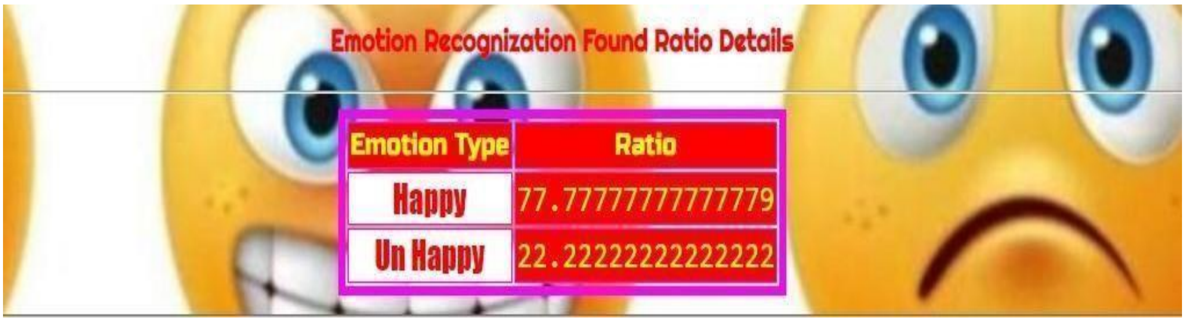


View Emotion Prediction By Voting Classifier Details !!!

Tweet Id	Tweet Message	Emotion Prediction
8	the next school year is the year for exams.Ã°ÂŸÂ˜Â¯ can't think about that Ã°ÂŸÂ˜Â #school #exams #hate #imagine #actorslife #revolutionschool #girl	Happy
17	i am thankful for having a paner. #thankful #positive	Happy
24	@user @user lumpy says i am a . prove it lumpy.	Happy
78	@user hey, white people: you can call people 'white' by @user #race #identity #medÃ¢ÂÂ¦	Un Happy
122	#cotd polar bear climb racing: angry polar bear climb racing, the polar bear living in cold places looking	Happy
140	our heas, thoughts, prayers go out to the more than 50 people who were murdered @ a gay nightclub in #florida.	Happy
83	how the #altright uses &amp; insecurity to lure men into #whitesupremacy	Un Happy
26	beautiful sign by vendor 80 for $45.00!! #upsideofflorida #shopalyssas #love	Happy
22	sad little dude.. #badday #coneofshame #cats #pissed #funny #laughs	Happy

Fig 6.6 Predict Emotion from Dataset Details

From the above figure, we can observe their set of tweets taken from the dataset details which shows many emotions of the tweets.



Emotion Recognization Found Ratio Details

Emotion Type	Ratio
Happy	77.77777777777779
Un Happy	22.22222222222222

Fig 6.7 Emotion Prediction Ratio of Dataset

The above figure shows the ratio of emotion recognition measured from the given dataset collection of tweets and then, we can also download the trained datasets.

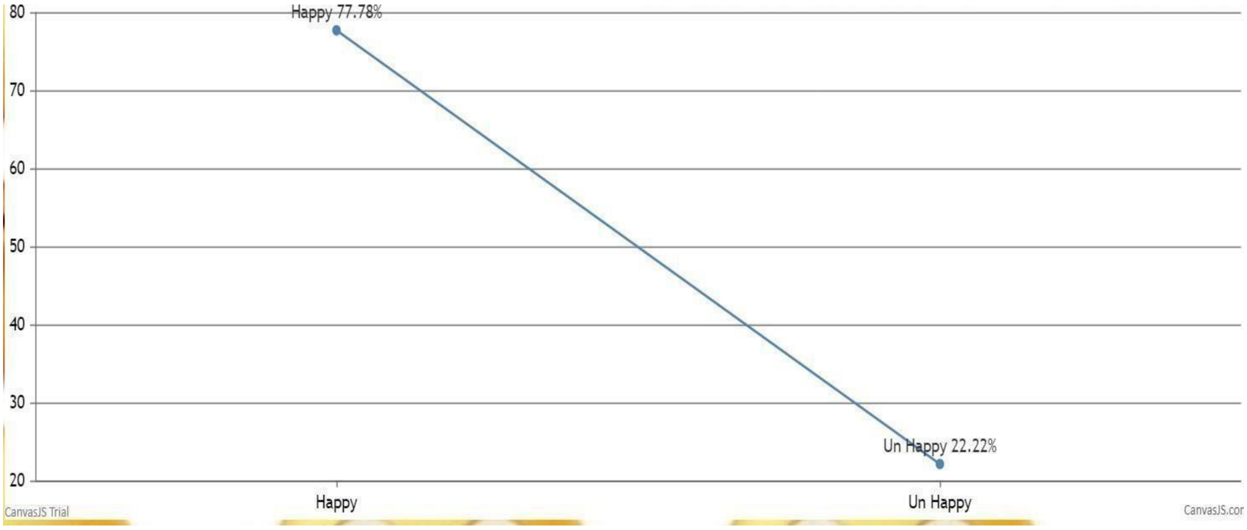


Fig 6.8 Line chart of Emotion Prediction

The graph above shows the comparison between Happy and unhappy where the percentage of happy is 77.8% and unhappyis 22.22%. From the dataset of tweets, **we can assure that most of the people are happy when compared to unhappy from the given dataset of emotions**. Thus, in this way, we can observe the emotion recognition through the given dataset. We can get percentage and results by using below formulas,

- **Accuracy:** Accuracy measures prediction correctness

Accuracy =Number of correctly classified predictions/Total predictions

- **Precision:** Precision measures the exactness of a classifier and determines the percentage of positive labeled tuples that are positive. It can be measured as:

Precision =T P/T P + F P

were,

TP: TP represents the positive predictions of a correctly predicted class.

FP: FP represents the negative predictions of an incorrectly predicted class.

TN: TN represents the negative predictions of a correctly predicted class.

FN: FN represents the positive predictions of an incorrectly predicted class.

- **Recall:** recall measures completeness and it presents the percentage of correctly labelled true positive tuples. Recall can be measured as:

Recall = T P/T P + FN

- **F1 Score**: It performs statistical analysis and computes scores between 1 and 0 by considering both precision and recall of the model. F1-score can be computed as:

F1score = 2*precision. Recall/precision + recall.

CHAPTER 7

CONCLUSION

Emotion recognition from text stories is a fascinating and challenging field that involves analyzing and understanding the emotional content conveyed through written text. Using computational methods and natural language processing techniques, researchers and practitioners aim to automatically detect and interpret emotions expressed in textual data. Advances in emotion recognition from text stories have significant implications in various domains. In social media analysis, emotion recognition can help monitor public sentiment and identify patterns of emotions expressed by users. In customer feedback analysis, it can assist in understanding customer sentiments towards products and services. The development of emotion recognition models typically involves the use of machine learning and deep learning algorithms. These models are trained on labeled datasets that associate specific emotions with corresponding textual features. Techniques such as sentiment analysis, natural language understanding, and semantic analysis are often employed to extract emotional cues from text, including sentiment words, linguistic patterns, and contextual information. Additionally, sarcasm, irony, and subtle emotional cues can be challenging to detect and interpret accurately. To overcome these challenges, ongoing research focuses on improving the robustness and generalizability of emotion recognition models. In conclusion, emotion recognition from text stories is a rapidly evolving field with significant potential for various applications. We can expect further progress in accuracy and the project proposed a novel combination of LR and SGD as a voting classifier for emotion recognition by classifying tweets as happy or unhappy. Projects are conducted to test machine learning models that are Logistic Regression, Naive Bayes, and Voting Classifier (LR-SGD). The results showed that all models performed well on the tweet dataset, then proposed model achieves the highest results using Term Frequency-Inverse document Frequency with **79%** Accuracy, **84%** Recall and **81%** F1-score. The future work will compare more feature engineering techniques and explore more combinations of ensemble models to improve the performance.

CHAPTER 8

REFERENCES

[1] N. F. F. da Silva, E. R. Hruschka, and E. R. Hruschka, ‘Tweet sentiment analysis with classifier ensembles,’’ Decis. Support Syst., vol. 66, pp. 170–179, Oct. 2014.

[2] C. Kariya and P. Khodke, ‘‘Twitter sentiment analysis,’ in Proc. Int. Conf. Emerg. Technol. (INCET), Jun. 2020, pp. 212–216.

[3] A. Alsaeedi and M. Zubair, ‘‘A study on sentiment analysis techniques of Twitter data,’’ Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 2, pp. 361–374, 2019.

[4] A. Bandhakavi, N. Wiratunga, D. Padmanabhan, and S. Massie, “Lexicon based feature extraction for emotion text classification,’ Pattern Recognit. Lett., vol. 93, pp. 133–142, Jul. 2017.

[5] J. Capdevila, J. Cerquides, J. Nin, and J. Torres, ‘‘Tweet-SCAN: An event discovery technique for geo-located tweets,’’ Pattern Recognit. Lett., vol. 93, pp. 58–68, Jul. 2017.

[6] T. Alsinet, J. Argelich, R. Béjar, C. Fernández, C. Mateu, and J. Planes, ‘‘An argumentative approach for discovering relevant opinions in Twitter with probabilistic valued relationships,’’Pattern Recognit. Lett., vol. 105, pp. 191–199, Apr. 2018.

[7] W. Chen, Y. Zhang, C. K. Yeo, C. T. Lau, and B. S. Lee, ‘‘Unsupervised rumor detection based on users’ behaviors using neural networks,’’ Pattern Recognit. Lett., vol. 105, pp. 226 –233, Apr. 2018.

[8] H. Hakh, I. Aljarah, and B. Al-Shboul, ‘‘Online social media-based sentiment analysis for us airline companies,’’ in New Trends in Information Technology. Amman, Jordan: Univ. of Jordan, Apr. 2017.

[9] R. Xia, C. Zong, and S. Li, ‘‘Ensemble of feature sets and classification algorithms for sentiment classification,’’ Inf. Sci., vol. 181, no. 6, pp. 1138–1152, Mar. 2011.

[10] M. Umer, S. Sadiq, M. Ahmad, S. Ullah, G. S. Choi, and A. Mehmood, ‘‘A novel stacked CNN for malarial parasite detection in thin blood smear images,’’ IEEE Access, vol. 8, pp. 93782–93792, 2020.

[11] S. Sadiq, A. Mehmood, S. Ullah, M. Ahmad, G. S. Choi, and B.-W. On, ‘‘Aggression detection through deep neural model on Twitter,’’ Future Gener. Comput. Syst., vol. 114, pp. 120 –129, Jan. 2021.

[12] F. Rustam, I. Ashraf, A. Mehmood, S. Ullah, and G. Choi, ‘‘Tweets classification on the base of sentiments for US airline companies,’’ Entropy, vol. 21, no. 11, p. 1078, Nov. 2019.

[13] C. D. Santos and M. G. D. Bayser, ‘‘Deep convolutional neural networks for sentiment analysis of short texts,’’ in Proc. 25th Int. Conf. Comput. Linguistics, Aug. 2014, pp. 69–78.

[14] M. Mohamed, ‘‘Mining and mapping halal food consumers: A geo-located Twitter opinion polarity analysis,’’ J. Food Products Marketing, vol. 24, pp. 1–22, Dec. 2017.

[15] H. Parveen and S. Pandey, ‘‘Sentiment analysis on Twitter data-set using naive Bayes algorithm,’’ in Proc. Int. Conf. Appl. Theor. Comput. Commun. Technol., Jan. 2016, pp. 416–41

[16] Emotion Recognition from Text Stories Using an Emotion Embedding Model | IEEE Conference Publication | IEEE Xplore.

[17] A Survey of Textual Emotion Recognition and Its Challenges | IEEE Journals & Magazine | IEEE Xplore.

[18] Emotion Recognition from Text Based on Automatically Generated Rules | IEEE Conference Publication | IEEE Xplore.

[19] Emotion Recognition from Multiple Modalities: Fundamentals and methodologies | IEEE Journals & Magazine | IEEE Xplore.

[20] A Survey of Textual Emotion Detection | IEEE Conference Publication | IEEE Xplore.’

WEBSITES:

- https://www.semanticscholar.org/paper/A-Study-on-Sentiment-Analysis-Techniques-of-Twitter-Alsaeedi-Zubair/6f34ad869da01abdf8d183a1786b54ff4217aefd
- https://onlinelibrary.wiley.com/doi/full/10.1002/eng2.12189
- https://www.hindawi.com/journals/cin/2022/2645381/
- https://devblogs.microsoft.com/cse/2015/11/29/emotion-detection-and-recognition-from-text-using-deep-learning/
- https://www.semanticscholar.org/paper/Emotion-Recognition-from-Text-Stories-Using-an-Park-Bae/

BOOKS:

1) **NAME:** A study on sentiment analysis techniques of Twitter data

   AUTHOR: A. Alsaeedi and M. Zubair

   PUBLISHED YEAR:2019

2) **NAME:** Unsupervised rumor detection based on users’ behaviors using neural networks’ Pattern Recognition.

   AUTHOR: W. Chen, Y. Zhang, C. K. Yeo, C. T. Lau

   PUBLISHED YEAR:2018

3) **NAME:** An argumentative approach for discovering relevant opinions in Twitter with probabilistic valued relationships

   AUTHOR: T. Alsinet, J. Argelich, R. Béjar, C. Fernández

   PUBLISHED YEAR:2018

4) **NAME:** Online social media-based sentiment analysis for us airline companies

   AUTHOR: H. Hakh, I. Aljarah, and B. Al-Shboul

   PUBLISHED YEAR:2017

ANNEXURE

```
from django.db.models import Count

from django.db.models import Q

from django.shortcuts import render, redirect, get_object_or_404

import datetime

import openpyxl

import  re

import string

from sklearn.feature_extraction.text import CountVectorizer

import pandas as pd

from  sklearn.feature_extraction.text  import  CountVectorizer

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

from sklearn.metrics import accuracy_score

from sklearn.metrics  import  f1_score

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import VotingClassifier

#Create your views here.

From Remote_User.models import ClientRegister_Model,Tweet_Message_details,detection_

Ratio,detection_accuracy,Emotion_prediction

def  login(request):

    if request.method == "POST" and 'submit1' in request.POST:

        username = request.POST.get('username')

        password = request.POST.get('password')

        try:

            enter  =  ClientRegister_Model.objects.get(username=username,password=password)
            request.session["userid"] = enter.idreturn redirect('ViewYourProfile') except:
            pass
   return render(request,'RUser/login.html')
```

```
def Add_DataSet_Details(request):

 if "GET" == request.method:

     return render(request, 'RUser/Add_DataSet_Details.html', {})

 else:

     excel_file = request.FILES["excel_file"]

     # you may put validations here to check extension or file size

     wb = openpyxl.load_workbook(excel_file)

     # getting all sheets

     sheets = wb.sheetnames

     print(sheets)

     # getting a particular sheet

     worksheet = wb["Sheet1"]

     print(worksheet)

     # getting active sheet

     active_sheet = wb.active

     print(active_sheet)

     # reading a cell

     print(worksheet["A1"].value)

     excel_data = list()

     # iterating over the rows and

     # getting value from each cell in row

     for row in worksheet.iter_rows():
            row_data = list()
            for cell in row:
                row_data.append(str(cell.value))
                print(cell.value)
            excel_data.append(row_data)
            Tweet_Message_details.objects.all().delete()
    for r in range(1, active_sheet.max_row+1):
        Tweet_Message_details.objects.create(
        Tweet_Id= active_sheet.cell(r, 1).value,
        Tweet_Label= active_sheet.cell(r, 2).value,
        Tweet_Message= active_sheet.cell(r, 3).value,
    )
    return render (request, 'RUser/Add_DataSet_Details.html', {"excel_data": excel_data})
def Register1(request):
    if request.method == "POST":
        username  = request.POST.get('username')
        email = request.POST.get('email')
        password = request.POST.get('password')
        phoneno = request.POST.get('phoneno')
        country = request.POST.get('country')
        state = request.POST.get('state')
        city = request.POST.get('city')
        ClientRegister_Model.objects.create(username=username, email=email, password=password,
phoneno=phoneno.,country=country, state=state, city=city)
        return render (request, 'RUser/Register1.html')
    else:
        return render(request,'RUser/Register1.html')
```

```
def ViewYourProfile(request):
    userid  = request.session['userid']
    obj = ClientRegister_Model.objects.get(id= userid)
    return render(request,'RUser/ViewYourProfile.html',{'object':obj})
def Search_DataSets(request):
    if request.method == "POST":
        kword = request.POST.get('keyword')
        if request.method == "POST":
            kword = request.POST.get('keyword')
            User_ID= request.POST.get('uid')
            print(kword)
            data = pd.read_csv("Datasets.csv")
          def clean_text(text):
             '''Make text lowercase, remove text in square brackets,remove links,remove punctuation
                and remove words containing numbers.'
                text = text.lower()
                text = re.sub('\[.*?\]', '', text)
                text = re.sub('https?://\S+|www\.\S+', '', text)
                text = re.sub('<.*?>+', '', text)
                text = re.sub('[%s]' % re.escape(string.punctuation), '', tex t)
                text = re.sub('\n', '', text)
                text = re.sub('\w*\d\w*', '', text)
                text = re.sub('@', '', text)
                text = re.sub('!', '', text)
                text = re.sub('#', '', text)
                return text
            data['text'] = data['tweet']. apply (lambda  x: clean_text(x))
            def remove_emoji(text):
                emoji_pattern = re.compile("["
                                            u"\U0001F600-\U0001F64F" # emoticons u"\U0001F300-
                                            \U0001F5FF" # symbols & pictographs u"\U0001F680-
                                            \U0001F6FF" # transport & map symbolsu"\U0001F1E0-
                                            \U0001F1FF" # flags (iOS) u"\U00002702-\U000027B0"
                                            u"\U000024C2-\U0001F251"
                                            "]+", flags=re.UNICODE)
                return emoji_pattern.sub(r'', text)
            data['text'] = data['tweet'].apply(lambda x: remove_emoji(x))
            data['text'].apply(lambda x: len(str(x).split())).max()
            # Creating a  mapping for Review Analysis
           x = data['tweet']
            y = data['label']
            cv = CountVectorizer()
            x = cv.fit_transform(x)
        models = []
            from sklearn.model_selection import train_test_split
            X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.20)
            X_train.shape, X_test.shape, y_train.shape
    print("Naive Bayes")
            from sklearn.naive_bayes import MultinomialNB
            NB = MultinomialNB()
            NB.fit(X_train, y_train)
            predict_nb = NB.predict(X_test)
            naivebayes = accuracy_score(y_test, predict_nb) * 100
```

```
print(naivebayes)

print(confusion_matrix(y_test, predict_nb))

print(classification_report(y_test, predict_nb))

models.append(('naive_bayes', NB))

# SVM Model

print("SVM")

from sklearn import svm

lin_clf = svm.LinearSVC()

lin_clf.fit(X_train, y_train)

predict_svm = lin_clf.predict(X_test)

svm_acc = accuracy_score(y_test, predict_svm) * 100

print(svm_acc)

print("CLASSIFICATION REPORT")

print(classification_report(y_test, predict_svm))

print("CONFUSION MATRIX")

print(confusion_matrix(y_test, predict_svm))

models.append(('svm', lin_clf))

print("Logistic Regression")

from sklearn.linear_model import LogisticRegression

reg = LogisticRegression(random_state=0, solver='lbfgs').fit(X_train, y_train)

y_pred = reg.predict(X_test)

print("ACCURACY")

print(accuracy_score(y_test, y_pred) * 100)

print("CLASSIFICATION REPORT")

print(classification_report(y_test, y_pred))

print("CONFUSION MATRIX")

print(confusion_matrix(y_test, y_pred))
```

```
    models.append(('logistic', reg))
print("Decision Tree Classifier")
    dtc = DecisionTreeClassifier()
    dtc.fit(X_train, y_train)
    dtcpredict = dtc.predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, dtcpredict) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, dtcpredict))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, dtcpredict))
    models.append(('DecisionTreeClassifier', dtc))
    print("SGD Classifier")
    from sklearn.linear_model import SGDClassifier
    sgd_clf = SGDClassifier(loss='hinge', penalty='l2', random_state=0)
    sgd_clf.fit(X_train, y_train)
    sgdpredict = sgd_clf.predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, sgdpredict) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, sgdpredict))
    print ("CONFUSION MATRIX")
    print (confusion_matrix(y_test, sgdpredict))
    models.append(('SGDClassifier', sgd_clf))
    classifier = VotingClassifier(models)
    classifier.fit(X_train, y_train)
    y_pred = classifier.predict(X_test)
```