EMOTION BASED SONG SUGGESTION SYSTEM FOR TAMIL LANGUAGE

A PROJECT REPORT

Submitted by

AJAY R	**(185002006)**
V ANOOSH	**(185002013)**

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

Department of Information Technology

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

Rajiv Gandhi Salai (OMR), Kalavakkam – 603 110

JUNE 2022

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this project titled "**Emotion Based Song Suggestion System For Tamil Language"** is the bonafide work of "**Ajay R** (185002006), and **V Anoosh Solayappan** (185002013)", who carried out the project work under my supervision.

Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. C. ARAVINDAN	**Dr. N. RADHA**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Professor	Assistant Professor
Department of IT	Department of IT
SSN College of Engineering	SSN College of Engineering
Kalavakkam – 603110	Kalavakkam – 603110

Submitted for project viva-voce examination held on …………………………........

EXTERNAL EXAMINER **INTERNAL EXAMINER**

ACKNOWLEDGEMENT

We are honoured to acknowledge the below mentioned people who have been instrumental in the completion of this project.

We would like to thank and show our deep sense of gratitude to our guide **Dr. N. RADHA**, Assistant Professor, Department of Information Technology, for her valuable advice and suggestions as well as her continued guidance, patience and support that helped us to shape and refine our work.

Our sincere thanks to **Dr. C. ARAVINDAN**, Professor and Head of the Department of Information Technology for his words of advice and encouragement.

We express our deep respect to the founder **PADMA BHUSHAN Dr. SHIV NADAR**, Chairman, SSN Institutions. We also express our appreciation to our **Dr. V. E. ANNAMALAI**, Principal, for all the help he has rendered during this course of study.

We sincerely thank our project panel members in the Department of Information Technology for providing us with helpful pointers throughout the course of the project. Finally, we would like to thank our parents and friends for their patience, cooperation and moral support throughout our life.

AJAY R **V ANOOSH**

ABSTRACT

Playlists' uses include allowing a particular desired musical atmosphere to be created and maintained without constant user interaction. The project aims to generate an automated playlist to overcome the above-mentioned difficulties. By doing so, user satisfaction and the end product obtained is in its fullest. For the above problem statement, the dataset is obtained from the Spotify Developer Application, where features like Acousticness, Danceability, Energy, Instrumentalness, Key, Liveness, Loudness, Mode, Speechiness, Tempo, Time Signature, Valence are extracted from the tracks and the playlist is suggested based on the user requirements.

The Machine Learning model will be trained on the dataset obtained from the Spotify Developer Application. Since this is a classification problem, this work intends to use different classification algorithms (Decision tree, eXtreme Gradient Boosting) and suggest a playlist using Ensemble Learning having attained an accuracy of 89.6%. A real-time picture of the user is taken. Emotion of the user is then detected using MobileNet model trained on FER2013 dataset which obtained a training accuracy of 75% and validation accuracy of 62%. After retrieving the tracks based on the user requirements, the tracks are displayed in an appealing manner by providing a smooth interface which will be integrated using Flask, which is a python-based web-framework.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	Abstract List of Tables List of Figures List of Acronyms and Abbreviations	iv vii vii ix
1.	**INTRODUCTION** **1.1** Motivation **1.2** Problem Statement **1.3** Objective **1.4** Challenges **1.5** Expected output and outcomes **1.6** Report Organization	 12 13 14 15 15 16
2.	**LITERATURE SURVEY** **2.1** Pre-existing Works **2.2** Inference	 17 21
3.	**SYSTEM DESIGN** **3.1** System Architecture	 22
4.	**DATA ANALYSIS** **4.1** Dataset Used **4.2** Algorithms Used **4.2.1** Decision Tree **4.2.2** Adaptive Boosting **4.2.3** eXtreme Gradient Boosting **4.2.4** Gaussian Naïve Bayes **4.2.5** Ensemble Learning **4.2.6** MobileNet **4.3** Implementation **4.3.1** Song Emotion Prediction **4.3.2** Emotion Recognition Model **4.3.3** Web Interface	 26 30 30 31 31 32 32 33 35 39 40

5.	**SOFTWARE TOOLS** **5.1** Software Tools and Technologies: **5.1.1** OpenCV **5.1.2** Python **5.1.3.** PIP **5.2** Hardware Devices Used	 45 45 46 50 51
6.	**RESULTS AND DISCUSSION** **6.1** Metrics Used **6.1.1** Accuracy **6.1.2** Area Under Curve **6.2** Results **6.2.1** Machine Learning Algorithm 1 (Decision Tree) **6.2.2** Machine Learning Algorithm 2 (Adaptive Boosting) **6.2.3** Machine Learning Algorithm 3 (Gaussian Naive Bayes) **6.2.4** Machine Learning Algorithm 4 (eXtreme Gradient Boosting) **6.2.5** Ensemble Learning (Weighted Averaging) **6.2.6** Deep Learning Algorithm1 (MobileNet)	 52 53 54 55 56 57 58 59 60 61
7	**CONCLUSION AND FUTURE SCOPE** 7.1 Conclusion 7.2 Future Scope	 63 64
8	**REFERENCES**	65

LIST OF TABLES

6.1	Classifier Results	55
6.2	MobileNet Results	61

LIST OF FIGURES

1.1	Growth of Music Streaming Apps	13
3.1	Phase I	22
3.2	Phase II	23
4.1	FER2013 Dataset	27
4.2	The architecture of MobileNet	34
4.3	Working of Classifier for a Happy song	35
4.4	Working of Classifier for a Sad song	36
4.5	Working of Classifier for an Angry song	37
4.6	Working of Classifier for a Calm song	38

4.7	Emotion Recognition Model	39
4.8	Home Page	40
4.9	Emotion Recognition Module (i)	40
4.10	Suggested Playlist for Happy mood	41
4.11	Emotion Recognition Module (ii)	41
4.12	Suggested Playlist for Happy mood	42
4.13	Emotion Recognition Module (iii)	42
4.14	Suggested Playlist for Angry mood	43
4.15	Emotion Recognition Module (iv)	43
4.16	Suggested Playlist for Sad mood	44
6.1	Confusion Matrix	53
6.2	Area Under Curve	55

6.3	Decision Tree Results	56
6.4	Adaptive Boosting Results	57
6.5	Gaussian Naive Bayes Results	58
6.6	eXtreme Gradient Boosting Results	59
6.7	Ensemble Learning Results	60
6.8	MobileNet Accuracy	61
6.9	MobileNet Loss	61
6.10	MobileNet AUC	62

LIST OF ACRONYMS AND ABBREVIATIONS

EEG	ElectroEncephaloGram
PPG	PhotoPlethysmoGraphy
GSR	Galvanic Skin Response
API	Application Programming Interface

SVM	Support Vector Machine
RBF	Radial Basis Function
AV	Arousal-Valence
FER	Facial Emotion Recognition
SIFT	Scale-Invariant Feature Transform
RCNN	Region-based Convolutional Neural Network
JSON	JavaScript Object Notation
REST	Representational State Transfer
CSV	Comma Separated Values
HCI	Human Computer Interaction
CNN	Convolutional Neural Network
DSC	Depthwise Separable Convolution
CV	Computer Vision
WSGI	Web Server Gateway Interface
AUC	Area Under Curve

CHAPTER 1

INTRODUCTION

With the advent of digital music and music-streaming platforms, the amount of music available for selection is now greater than ever. Sorting through all this music is impossible for anyone. Music recommendation systems reduce human effort by automatically recommending music based on genre, artist, instrument, and user reviews. Music recommendation systems aim to provide real-time recommendations of both new and old songs to the users. E-commerce giants like Amazon, Spotify provide personalized recommendations to users based on their taste and history. Facial emotion is a fundamental means to communicate with people. The ability to understand the facial emotion of others is a key to successful communication. Emotion becomes an intermediate between human beings to assist interactions. Facial Emotion Recognition is the one that recognizes the state of emotion of the individual personality by utilizing the facial image that is captured by various means like high-resolution cameras, surveillance cameras, and several other means that capture the face images. Facial expression, being a fundamental mode of communicating human emotions, finds its applications in human–computer interaction (HCI), health care, surveillance, driver safety, deceit detection, etc.

1.1 Motivation

The Music Streaming industry has seen a big boom in recent times. The number of subscribers has increased from 76.8 million to 487 million over a span of 6 years. As a country, India has over 10 mainstream music streaming applications. With the increase in the number of users, the demand for advanced features have gone up. Manual creation of a playlist is a tiresome process which is why streaming services like Spotify provide their own playlists. One of the most common problems faced in such apps are that the recommendation systems often do not understand the needs of the user and rather provide suggestions based on their history and the liked songs which are not always right. People all over the world somehow find time to listen to music either with the help of radio stations while driving or gym speakers while working out. Music is intertwined in our day-to-day activities and it will continue to do so. Music is more than something that is used to while away time, it has many benefits physically and psychologically.

Here are some of the benefits:

- Boosting Cognition
- Improving Performance under pressure
- Reducing Depression Symptoms
- Increasing Memory Power
- Lowering Stress Levels
- Better functioning of Blood Vessels
- Reducing Pain
- Increasing Endurance

1.2 Problem Statement

The number of music listeners in this developing world is exponentially increasing day by day. Music is not about just listening, but also to vibe along with it and get immersed in a different world. The number of streaming platforms are also increasing day by day. The first ever streaming platform Napster formerly known as Listen.com (1999–2001) Rhapsody (2001–2016) was found in the year of 2001 after which more than 20 streaming platforms have been found right from Spotify, Amazon Music, Jio Saavn, Gaana, Wynk Music, etc. As part of this work, an Emotion Based Song Suggestion System is created. The approach that is being explored is a system detecting the emotion of the user and suggest a playlist based on the current emotion of the user.

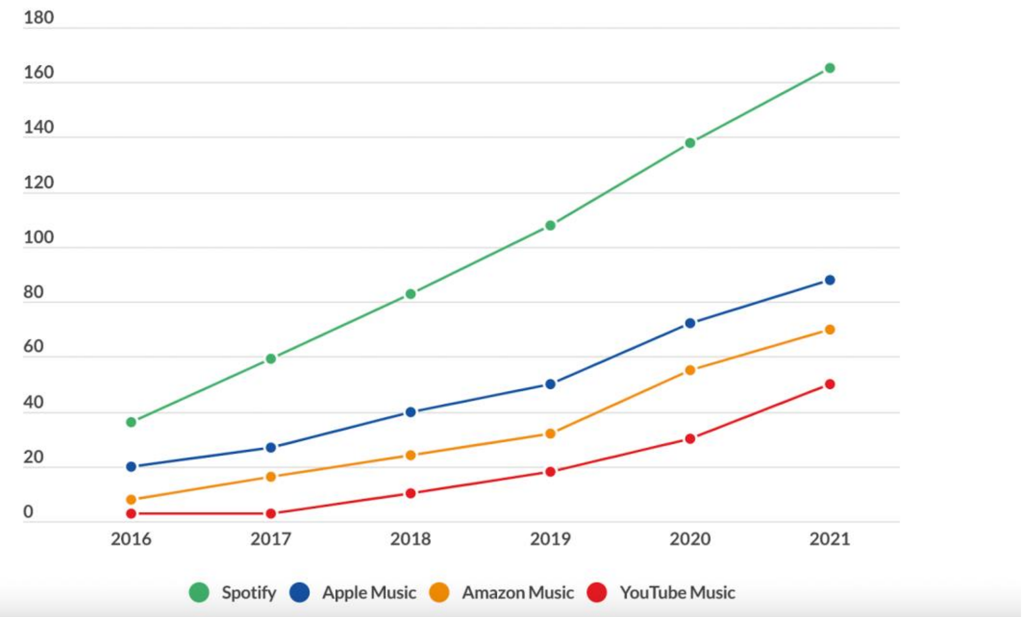

Fig 1.1 Growth of Music Streaming Apps

The above is a pictorial representation of the Music streaming subscribers by app, which clearly shows the booming of the music streaming industry economically.

1.3 Objective

Some of the pre-existing works on Emotion Based Song Recommendation are:

- Devices such as EEG, PPG and GSR are used to monitor the user’s physiological signals based on which their mood is detected. Such methods require the constant need of wearing sensors and they are not very effective.
- Using social media handles of users to analyse their posts and comments based on which the mood is predicted. The drawback of this method is that it does not work based on the real-time emotion of the user.

To counter the above-mentioned problems, this work uses an Emotion Based Song Suggestion System for Tamil Language. Our objectives are as follows:

- Create a dataset exclusively for Tamil songs containing the audio features of songs such as Acousticness, Danceability, Energy, Instrumentalness, Key, Liveness, Loudness, Mode, Speechiness, Tempo, Time Signature, Valence.
- Use multiple Machine Learning classifiers (Decision Tree, eXtreme Gradient Boosting, Adaptive Boosting and Gaussian Naives Bayes) and use Ensemble Learning to create a model that predicts the mood of the song accurately to add it to the existing songs dataset.

- Detect the emotion of the user (Calm, Happy, Sad, Angry) with the help of Face Recognition algorithms. Once the emotion is obtained that is mapped to a particular mood.
- Create a Web Interface using Flask and present the playlist to the user in accordance with his current emotion.

1.4 Challenges

There are challenges with recognizing the emotion of the user using the face recognition system as there are not many proper datasets available. Another limitation relates to the classification of the new songs (emotion-wise) which is got from the spotify playlist as in many languages the sad songs get mixed up with calm songs and happy songs too.

1.5 Expected output and outcome

The developed model will be an emotion based song suggestion system, which takes the user's emotion using the face recognition model and maps it with the dataset created using the Spotify API call. A machine learning model is also developed to predict the mood of the new songs and adds it to the existing songs dataset. A comparative study discussing the performance of all the classification methods utilised will be provided. This project will showcase a real time generation of playlist in accordance with the current emotion of the user. A web application devised and developed using Flask was used to incorporate the above work and the final music player is presented to the user using this web interface.

1.6 Organization Report

The detailed Organization of the report is as follows:

- Chapter 2: Literature Review
- Chapter 3: System Design
- Chapter 4: Data Analysis
- Chapter 5: Software Modules
- Chapter 6: Results and Discussions
- Chapter 7: Conclusion and Future Scope

CHAPTER 2

LITERATURE REVIEW

2.1 Pre-existing Works

Paul *et al.* (2020) proposes a recommendation model where initially during the sign-up process the system will ask the user for information like age, gender, location, and then music preference like which language the music user likes, what genre they prefer, and then the artists the user likes. Using this data, the system will initially recommend songs in the following way to solve cold-start problems. To address this, a Metadata filtering, Context-based model, and Content-based model was used in this work. User-centric models are far more effective in predicting accurately than other models. Scope for hybrid recommendation system in the future. The conclusion of this work is that User-centric models are far more effective in predicting accurately than other models. Scope for hybrid recommendation system in the future.

Karl-Arnold Bodarwé *et al.* (2015) proposes a system that provides recommendations based on the user's selection of emotion. Therefore the user interface provides a set of buttons associated with the emotion categories (angry, sad, calm, happy), which allow the user to select the desired music emotion. The system displays the resulting recommendations as a playlist. To address this, Tellegen-Watson-Clark-Model was used in this work. The dataset consisted of a collection of 424 songs of various genres and artists. To achieve a preferably even distribution over the four emotions (Angry, sad, relaxing, happy). This work promised to give an accuracy of 74%.

Kunjal Gajjar *et al* (2015) proposes a system which Maps the artists to a particular mood and the mood of the user is obtained using scraping tweets. In addition to this, it focuses on the mood classification also known as emotion classification based on the lyrics. To automatically identify a user's mood, the proposed model considers only widely used social platforms such as Facebook, Twitter and WhatsApp.

Champika H. P. D. Wishwanath *et al* (2020) proposes a system where Lyrics and audio are analyzed and matched using Random Forest Classifier. To address this, a Naive Bayes classifier was used. The dataset consisted of 1 Million songs, which has a wide variety of songs, spread across different artists and genres. This work promised to give an accuracy of 61.32%.

Jongseol Lee *et al* (2018) proposes a system which utilizes EEG electrodes to analyse emotion of the user and takes history of previously listened songs to suggest based on artist, composer etc. To address this, an SVM model with RBF kernel was used. A music corpus, KETI AFA2000, which contains approximately 2,400 Korean Songs, was used as the dataset for this work. This work promised to give an accuracy of 81.07%.

Luís Cardoso *et al* (2011) proposes a system which uses Arousal and Valence to identify the mood and provide the playlist. To address this, an SVM Classifier and Regressor was used. The dataset consists of AV annotations gathered using a subjective study involving 253 volunteers, containing 194 music clips with 25-sec duration. This work promised to give an accuracy of 56.30%.

Renato Panda *et al* (2012) proposes a system where 253 features are extracted using PsySound, MIRToolbox and Marsyas and to reduce the number of features RReliefF algorithm was used. To address this, the model made use of multiple regressors. A Dataset of 903 songs similar to MIREX dataset was used. This work promised to give an accuracy of 68.90%.

Deger Ayata *et al* (2013) proposes a system which makes use of Wearable Computing. Wearable Computing is used where the person of interest is made to wear a device which provides human-computer connection in this case GSR (Galvanic Skin Response) and PPG (Photoplethysmography) are used. GSR is attached to the subject's hand or leg and measures skin resistance while PPG is attached to the thumb of the subject and detects change in blood volume. These devices measure the arousal and map it to an emotion based on the valence-arousal model. To address this, a Random Forest Classifier was used. The dataset was collected using the DEAP Emotion Database. This work promised to give an accuracy of 72.06%.

Immanuel *et al* (2019) proposes a system which makes use of Face Recognition as the preliminary step. Face Recognition is used to detect the emotion of the user. The user is required to record a video and frames are extracted from which the model deduces the emotion after which tagged songs which are stored in a database are recommended for the corresponding emotion. This is done for 4 emotions which are Happy, Angry, Sad and Surprise. This makes use of a Hidden Markov Model for face recognition and linear SVM Classifier.

Jaepyeong Cha *et al* (2014) proposes a Stress Relieving Music Recommending System with the help of a portable wireless photoplethysmography module with a finger-type sensor which measures the stress level of the person. The subject is made to select 3 songs they like, 3 songs they hate and then they are made to listen to the 6 songs, during which their stress levels are monitored and then songs similar to least stressful ones are suggested. This work promised to give an accuracy of 56.1%.

Adit Jamdar *et al* (2015) proposes an approach where the emotion of the song is analysed based on the lyrics and audio features. WordNet, a lexical database has been used to calculate the valence and arousal values for the lyrics and the song features are extracted using The Echo Nest. Based on the Russell's Model of mood the song is classified into a particular group and it is validated. The model used here is K-Nearest Neighbors. This makes use of the LAST.FM Dataset. This work promised to give an accuracy of 83.40%.

Zhang *et al* (2022) proposes a music recommendation system to classify songs into 4 genres namely rock, popularity, classical and light music. The dataset for this work is NetEase Cloud Music and Global Music Network where each song had the duration of 240 seconds and was split into 93 fragments and fed into the convolution neural network. An average accuracy of 50.35% was obtained.

Wen *et al* (2021) uses deep learning and IoT to build intelligent music recommendation system where image recognition is done with SIFT algorithm and music classification is done with SVM algorithm.

Faster RCNN algorithm is used for feature extraction. A maximum accuracy of 87.2% was achieved.

Nan *et ali* (2022) uses A-MobileNet a modified version of MobileNetV1 on the FERPlus dataset which is another modification of FER2013 dataset. Emotion recognition is done using this model. The images are of size (48,48) in RGB format. An accuracy of 88.11% was achieved on FERPlus and 84.49% was achieved on RAFDB dataset.

2.2 Inference

Previous works have either made use of biological sensors to obtain real-time emotion or apply Natural Language Processing to texts and tweets and arrive at the emotion of the user. Datasets used before were open source datasets like last.fm which had stored the whole song in this work the features of the songs are stored rather than the whole song. This work focuses on generating a playlist based on the real time emotion of the user, where the emotion is recognised using a face recognition system.

CHAPTER 3

SYSTEM DESIGN

3.1 System Architecture

The System has been built as two separate modules and then integrated as one. Initially, i) The Song Emotion Prediction Module and ii) The User Emotion Recognition Module. Figure 3.1 describes the system architecture of the proposed application.

In module one, A tamil songs feature dataset is created with the help of spotipy package and GET requests. This is used to train two classifiers namely Decision Tree and XgBoost Classifiers. Ensemble Learning has been applied in the form of Weighted Averaging and the emotion of the song is predicted.

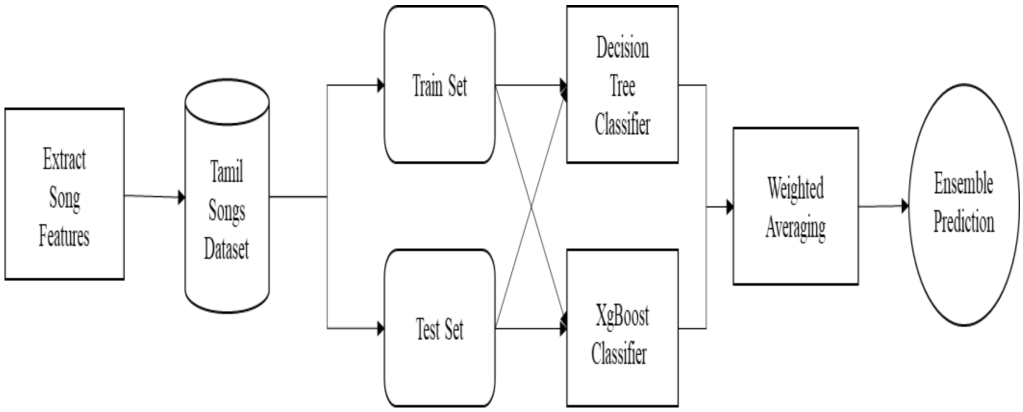

Fig 3.1 Phase I

In module two, the emotion of the user is obtained using the MobileNet model after which a mapping is done to emotion of the user and songs in the dataset with the same emotion. These songs are presented to the user in the form of a playlist.

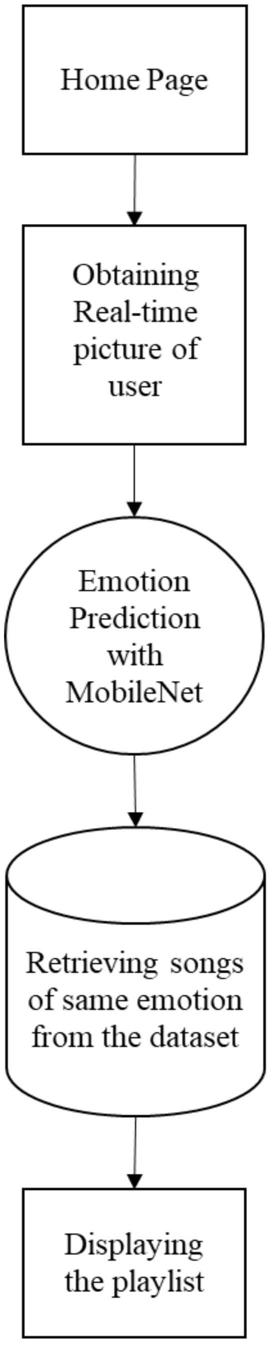

Fig 3.2 Phase II

As part of this work, the intent is to create an emotion based song suggestion system for the Tamil language. The dataset for this problem statement was obtained using the spotify developer's page. Based on simple REST principles, the Spotify Web API endpoints return JSON metadata about music artists, albums, and tracks, directly from the Spotify Data Catalogue. In this way, the audio features of the songs are extracted. As part of the dataset, audio features like Acousticness, Danceability, Energy, Instrumentalness, Key, Liveness, Loudness, Mode, Speechiness, Tempo, Time Signature, Valence were extracted for more than 600 songs along with their mood in a CSV file. In addition to the extraction of features of the songs, the cover image of all the songs was extracted so as to present it as a music player.

To add new songs to the existing dataset, a machine learning model was developed which took the above dataset as input with input features as Acousticness, Danceability, Energy, Instrumentalness, Key, Liveness, Loudness, Mode, Speechiness, Tempo, Time Signature, Valence and the Class. Class is the mood of the song which was manually entered after extracting the dataset.

The labels for the Class are as follows (Mood of the Song):

- 0 - Calm
- 1 - Happy
- 2 - Sad
- 3 - Angry

To solve the purpose of predicting the emotion of the new songs, machine learning classifiers namely Decision tree and eXtreme Gradient Boosting were developed using which an ensemble of the above classifiers was done using the weighted average technique to enhance the mood prediction of the new songs. The new songs along with their class (mood of the song) is appended to the existing CSV file using the CSV Writer Python module.

To get the emotion of the user, a face recognition model was developed using the MobileNet Architecture which was trained using the FER2013 Dataset. The MobileNet model is based on a streamlined architecture that uses depth wise separable convolutions to build lightweight deep neural networks that can have low latency for mobile and embedded devices. Four emotions of the user were broadly classified namely Calm, Sad, Happy and Angry.

After getting the emotion of the user, this is mapped with the songs dataset which was created using the Spotify REST call. The above work is incorporated using a web application which is designed and developed using flask. The final playlist is presented as a music player with next, play and previous buttons.

CHAPTER 4

DATA ANALYSIS

In this Chapter, the dataset used and created is seen, and the different algorithms used are explained.

4.1 Dataset Used

FER2013 dataset was used for the purpose of emotion recognition from an image. FER stands for Facial Emotion Recognition, this is a very important aspect for Human-Computer Interaction (HCI). The dataset was introduced at the International Conference on Machine Learning (ICML) in 2013 and became a benchmark in comparing model performance in emotion recognition. Human performance on this dataset is estimated to be 65.5%. The dataset consists of 28,709 images categorised into 7 emotions. The images in the dataset are of size (48x48) and are in grayscale mode. For training purposes 21,000 images belonging to 4 emotions (anger, sad, calm and happy) are used. For testing purposes 5,200 images belonging to the same 4 emotions are used. 4953 images belonging to the emotion anger, 8989 images belonging to the emotion happy, 6060 images belonging to the emotion sad and 6198 images belonging to the calm are present in the dataset used.

(i) (ii)

(iii) (iv)

Fig 4.1 FER2013 Dataset

(i) Angry person (ii) Happy person

(iii) Calm person (iv) Sad person

Happy

Happy expression is symbolized by joy with a smile, raised cheeks, tightened eyelids, drawn down eyebrows and mouth can be open or closed.

Sad

Sad expression is generally showed by the corner lip down movement, dropped jaw and upper eyelids are dropped with the inner corner of eyebrow brought together.

Anger

Angry expression is symbolized by the raised upper eyelids with tightened up area around eyes, eyebrows are lowered and joined together, jaw tightly clenched, thinning of lips, forward lower jaw. In case of teeth closed position, mouth tends to get a rectangular shape.

Calm

Calm expression is relaxed facial muscles. Eyelids are tangent to the iris. The mouth is closed and lips are in contact.

Song Features

- Acousticness**:** A confidence measure from 0.0 to 1.0 of whether the track is acoustic.
- Danceability**:** Describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity.
- Duration**:** The duration of the track in milliseconds.
- Energy**:** Represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale.

- Instrumentalness**:** Predicts whether a track contains no vocals. “Ooh” and “aah” sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly “vocal”.
- Key**:** The estimated overall key of the track. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on.
- Liveness**:** Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live.
- Loudness**:** The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks.
- Mode**:** Indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
- Speechiness**:** This detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value.
- Tempo**:** The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece, and derives directly from the average beat duration.
- Time Signature**:** An estimated overall time signature of a track. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).
- Valence**:** Describes the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

A dataset of 635 Tamil songs has been created which consists of the above-mentioned features. This was done with the help of Spotipy API (GET requests). These songs have been categorised into classes based on emotion (calm, happy, sad and anger).

4.2 Algorithms Used

Four Machine Learning Models (Decision Tree, Gaussian Naïve Bayes, Adaptive Boosting and eXtreme Gradient Boosting) and an Ensemble Learning Model are used to predict the emotion of songs. One Deep Learning Model (MobileNet) is used for Facial Emotion Recognition purposes.

4.2.1 Decision Tree

Decision trees (Hastie 2009) is a well-established algorithm used to learn from data. Decision trees are of two types Classification tree and Regression tree, in this work a Classification tree (Gupta 2012) is used. These trees are generally resistant to methods such as scaling and normalization as internal feature selection happens within the tree. Tree based data structures handle data different datatypes without the need of complications like encoding for datasets with numerical and categorical data. Random Forest algorithm makes use of multiple Decision Trees and averages the final prediction of each tree to give the result. Gini impurity is the probability of misclassifying a new random data. This is the criterion used to decide the split of the Decision Tree.

4.2.2 Adaptive Boosting

Adaptive Boosting (Freund 1997) is one of the most commonly used Boosting Algorithm which makes use of multiple Decision Trees. The trees do not contribute uniformly rather varyingly. The intent of the algorithm is to reduce the net loss which is given by Equation (4.1).

$$L_A - \min_i L_i \tag{4.1}$$

$$L_A = \sum_{t=1}^{T} p^t . l^t$$

L_A stands for total cumulative loss

$$L_i = \sum_{t=1}^{T} l_i^t$$

p^t and l^t indicate the probability and loss vector.

This Ensemble Learning method uses weak learners called Decision Stumps (Decision Trees with one root and two leaves), combines their learning to form a strong learner.

4.2.3 eXtreme Gradient Boosting

eXtreme Gradient Boosting (Chen 2015) is an improvement on gbm (Gradient Boosting Machine) as parallel computing is possible. This algorithm can be used for regression and classification. In this work

XGBoost is used for classification purposes. The base learners are a set of classification and regression trees.

4.2.4 Gaussian Naïve Bayes

Gaussian Naive Bayes (Rish 2001) classifiers consists of many probabilistic classifiers which are based on Bayes' Theorem given by Equation (4.2). The main assumption in this classifier high independence between features. The Bayes Classifier is given by Equation (4.3)

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \tag{4.2}$$

$$h^*(x) = \arg\max_{i} P\ (\mathrm{X} = \mathrm{x}|\mathrm{C} = \mathrm{i})\ P(C = i) \tag{4.3}$$

C indicates the classes, i indicates the i^{th} class, x denotes the value. Gaussian Naïve Bayes is used in many real-world situations, famously document classification and spam filtering.

4.2.5 Ensemble Learning

Ensemble Learning (Dietterich 2002) utilises multiple classifiers and combines their learning to yield a better result. Various ensemble strategies exist such as Voting, Bagging, Boosting, etc. The ensemble strategy used in this work is Weighted Average. The contribution of each classifier used is not uniform rather it varies with the weight associated with it. Ensemble Learning is a technique to make use of multiple classifiers to enhance stability and accuracy of the model. Weighted Average is given by Equation (4.4).

$$W = \frac{\sum_{i=1}^{n} w_i X_i}{\sum_{i=1}^{n} w_i} \tag{4.4}$$

W is the weighted average, n is the number of terms to be averaged, w is the weight to be applied to values and X stands for value to be averaged.

4.2.6 MobileNet

Since AlexNet used a convolutional neural network (CNN) for image classification and won the first place in the ImageNet competition in 2012. Researchers have designed more and more deep neural network models, such as classical VGGNet16/19, GoogleNet, ResNet50, and so on. Compared with traditional classification algorithms, these have been excellent. However, as people continue to deepen the network, huge storage pressure and computational burden caused by the model calculations have begun to limit the application field of the deep learning models. Traditional CNN has large memory requirements and a large computational amount, which makes it impossible to run on mobile devices and embedded devices. To this end, Google has proposed a lightweight deep neural network called MobileNetV1. It is a CNN with a smaller model size, less trainable parameters and calculation amount, and is suitable for mobile devices. It takes full advantage of its computing resources and improves the accuracy of the model to the greatest extent. The core idea of MobileNetV1 network is to replace the standard convolution operation with depthwise separable convolution (DSC) to reduce model parameters.

Specifically, DSC is to use the 3 x 3 convolution kernel with only one layer thickness, sliding layer by layer on the input tensor, and generate an output channel after each convolution. When the convolution was completed, use 1 x 1 pointwise convolution.

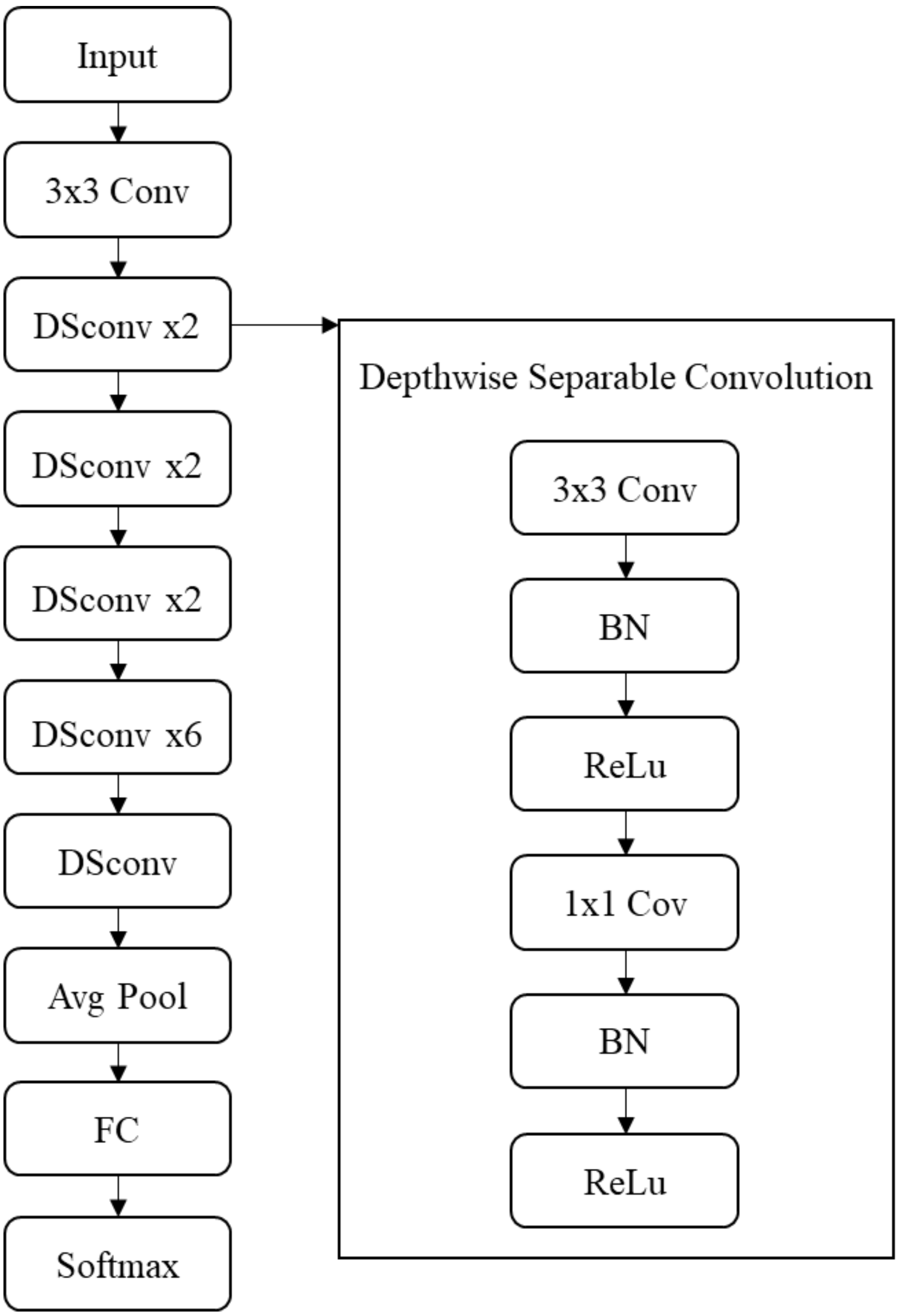

Fig 4.2 The architecture of MobileNet

4.3 Implementation

4.3.1 Song Emotion Prediction

Once the id of the song is fed to the Spotipy API, the features of that particular song is extracted. These extracted features are provided as input to the Ensemble Learning model which predicts the category (angry, happy, sad, calm) to which the song belongs to.

```
[55] q=get_songs_features('0Yup6SQDmHmYKG8Faigkqk')

q

([0.583, 0, 0.46, 4, -11.509, 0.035, 0.751, 0.000147, 0.146, 0.534, 129.904],
 ['Danceability',
  'Mode',
  'Energy',
  'Key',
  'Loudness',
  'Speechiness',
  'Acousticness',
  'Instrumentalness',
  'Liveness',
  'Valence',
  'Tempo'],
 'Thendral Vandhu Theendumbothu (From "Avatharam")')
```

```
[87]
     if final1[0]==0:
       emo="calm"
     elif final1[0]==1:
       emo="happy"
     elif final1[0]==2:
       emo="sad"
     else:
       emo="angry"

emo

'happy'

[89] print("The song",s_name,"is a",emo,"song")

The song Thendral Vandhu Theendumbothu (From "Avatharam") is a happy song
```

Fig 4.3 Working of Classifier for Happy song

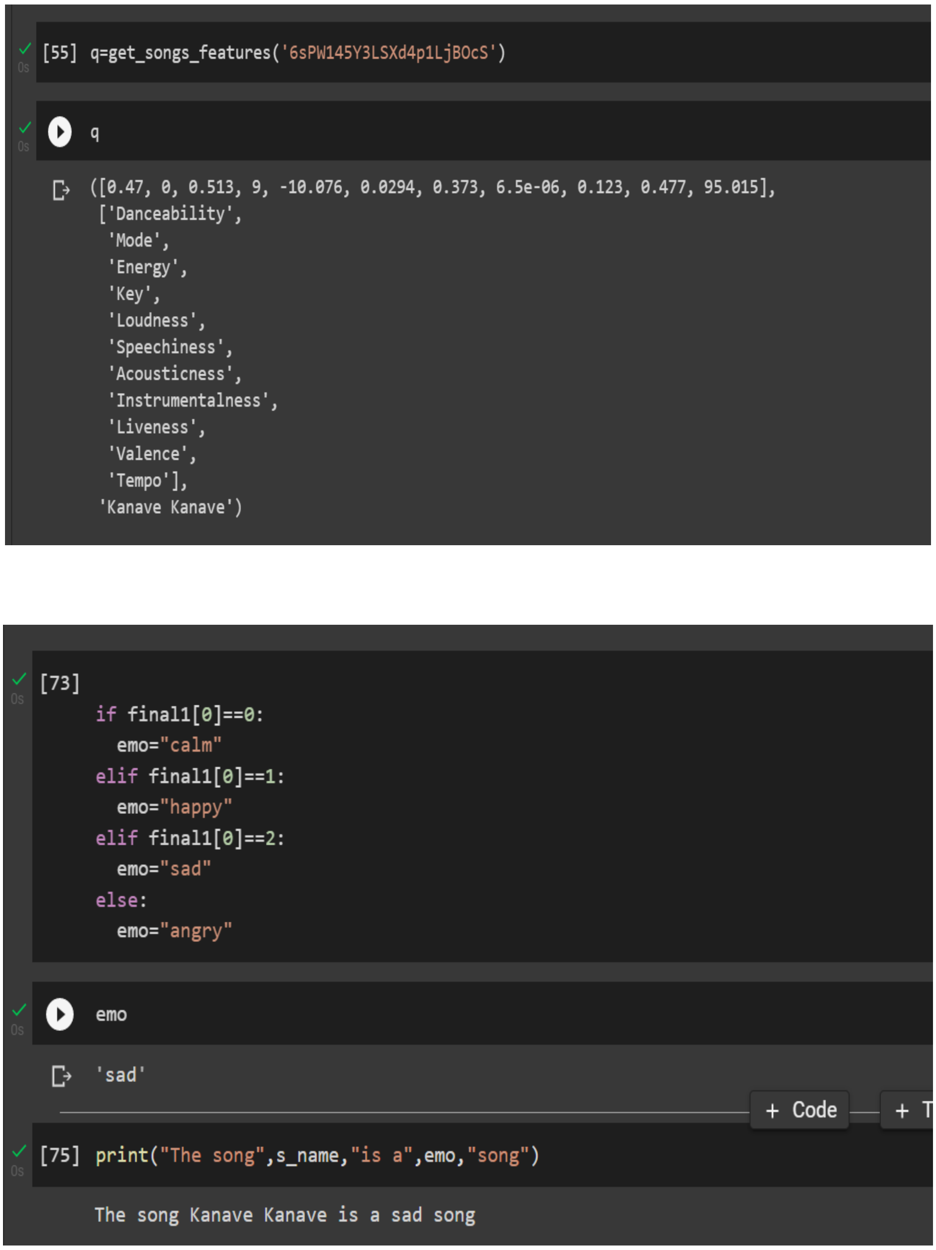

Fig 4.4 Working of Classifier for Sad song

```
[95] q=get_songs_features('11qgUXxN2YEGNOfDDUOT2K')

[96] q

([0.61, 1, 0.864, 2, -3.465, 0.203, 0.0833, 0, 0.0351, 0.413, 159.944],
 ['Danceability',
  'Mode',
  'Energy',
  'Key',
  'Loudness',
  'Speechiness',
  'Acousticness',
  'Instrumentalness',
  'Liveness',
  'Valence',
  'Tempo'],
 'Neeye Oli')

q[0]

[0.568, 0, 0.581, 7, -13.044, 0.0553, 0.106, 0, 0.125, 0.468, 149.984]
```

```
if final1[0]==0:
  emo="calm"
elif final1[0]==1:
  emo="happy"
elif final1[0]==2:
  emo="sad"
else:
  emo="angry"

[137] emo

'angry'

[138] print("The song",s_name,"is a",emo,"song")

The song Vikram - Title Track is a angry song
```

Fig 4.5 Working of Classifier for Angry song

```
[76] q=get_songs_features('1Fg7iPvC0uugUZAjKfN3Wz')

q

([0.568, 0, 0.581, 7, -13.044, 0.0553, 0.106, 0, 0.125, 0.468, 149.984],
 ['Danceability',
  'Mode',
  'Energy',
  'Key',
  'Loudness',
  'Speechiness',
  'Acousticness',
  'Instrumentalness',
  'Liveness',
  'Valence',
  'Tempo'],
 'Enna Vilai')

[78] q[0]

[0.568, 0, 0.581, 7, -13.044, 0.0553, 0.106, 0, 0.125, 0.468, 149.984]
```

```
[92]
if final1[0]==0:
  emo="calm"
elif final1[0]==1:
  emo="happy"
elif final1[0]==2:
  emo="sad"
else:
  emo="angry"

emo

'calm'

[94] print("The song",s_name,"is a",emo,"song")

The song Enna Vilai is a calm song
```

Fig 4.6 Working of Classifier for Calm song

4.3.2 Emotion Recognition Model

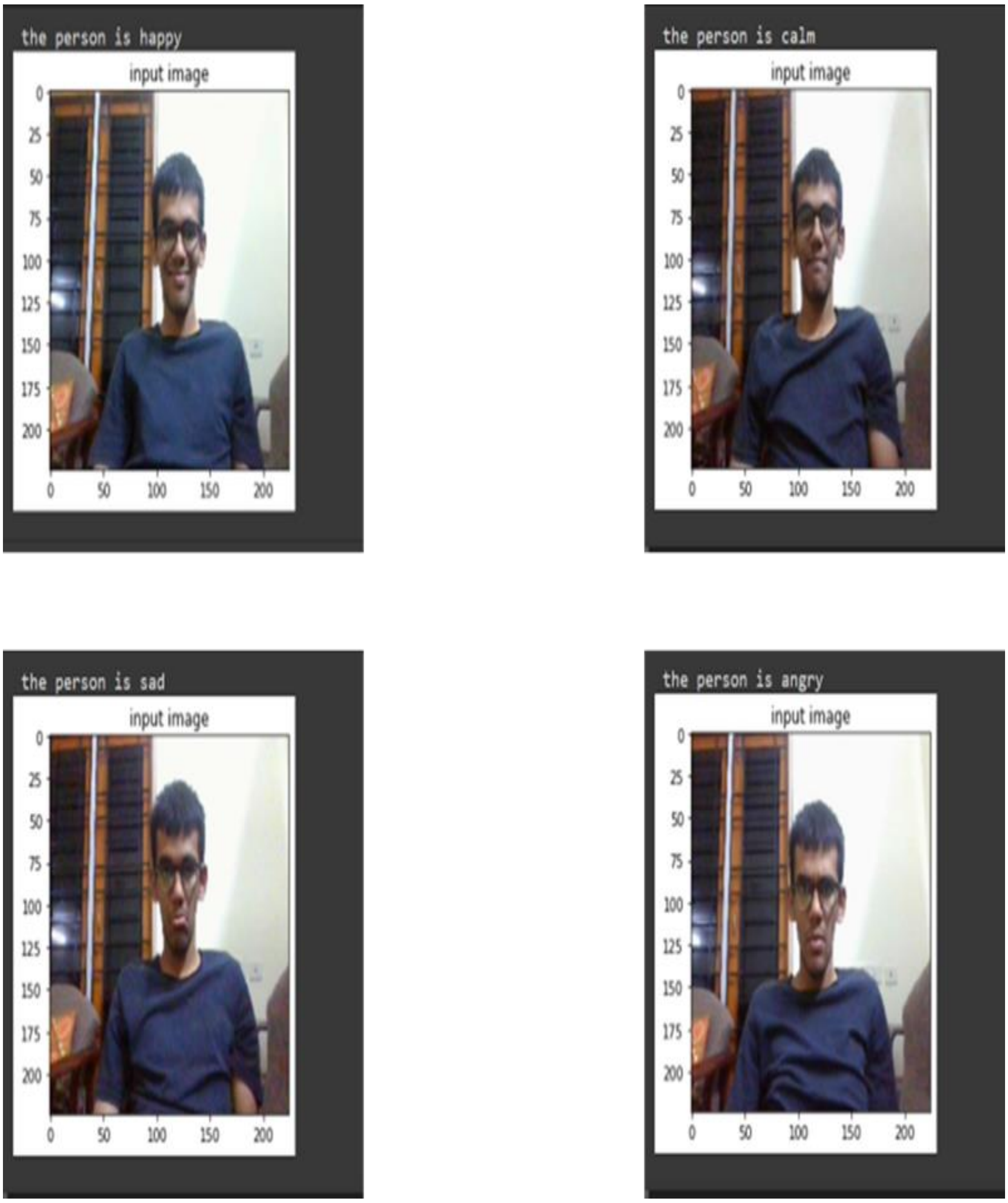

Fig 4.7 Emotion Recognition Model

4.3.3 Web Interface

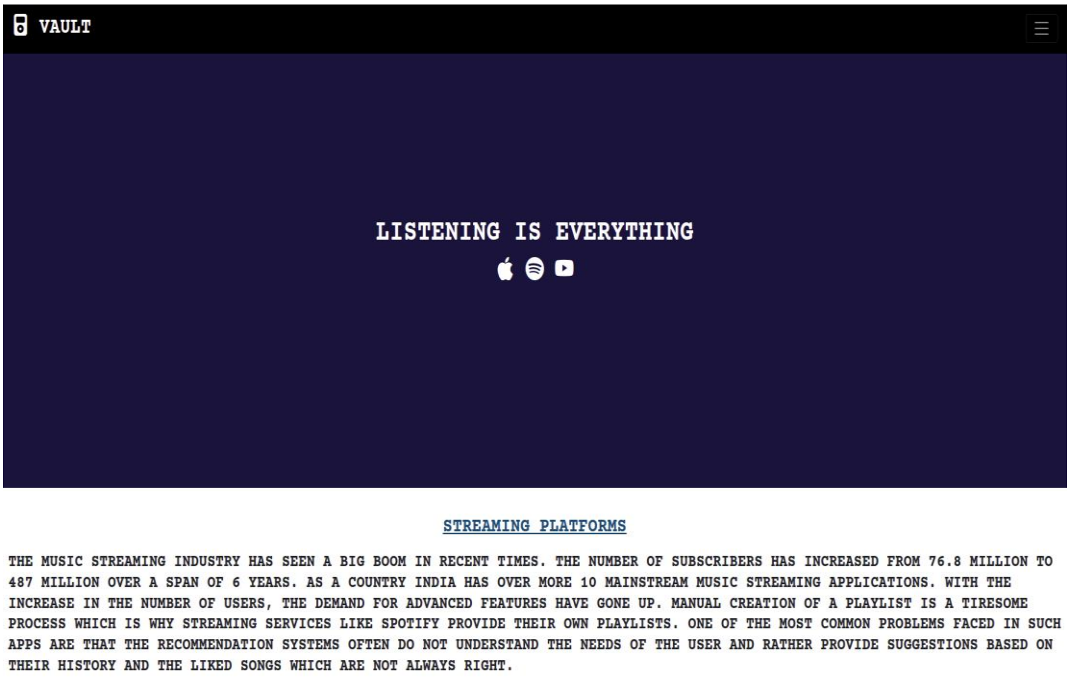

Fig 4.8 Home Page

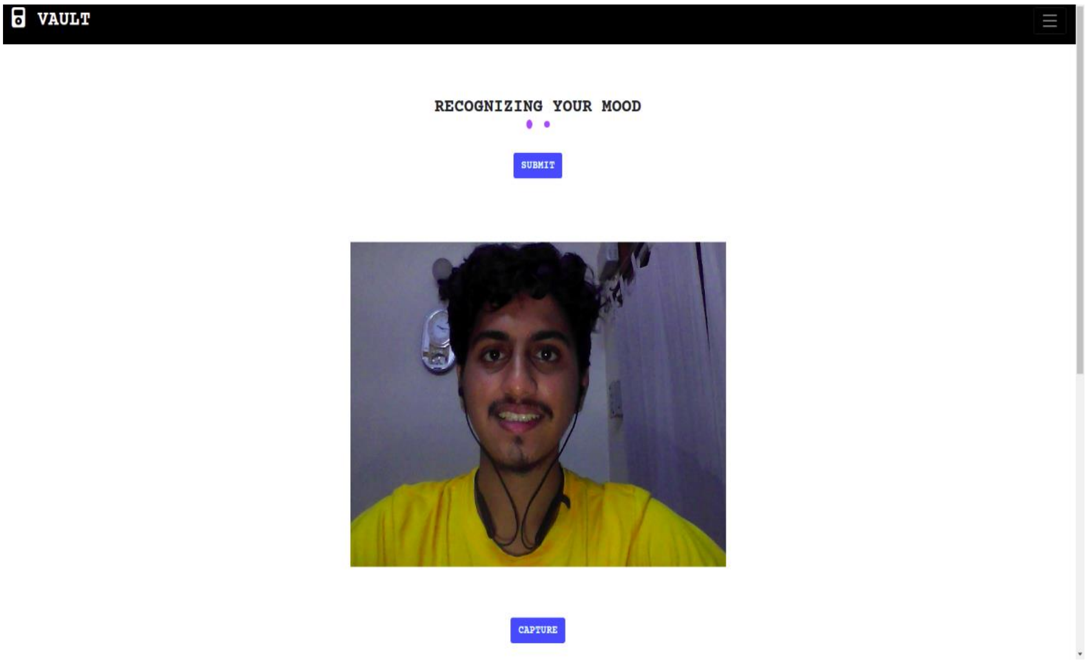

Fig 4.9 Emotion Recognition Module (i)

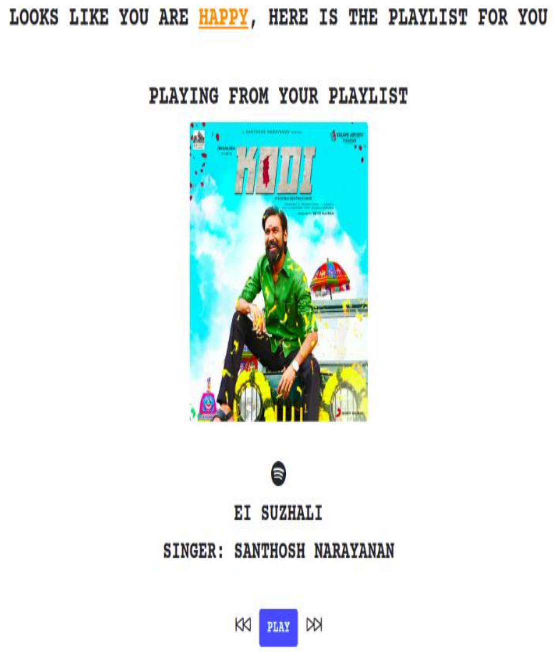

Fig 4.10 Suggested Playlist for Happy mood

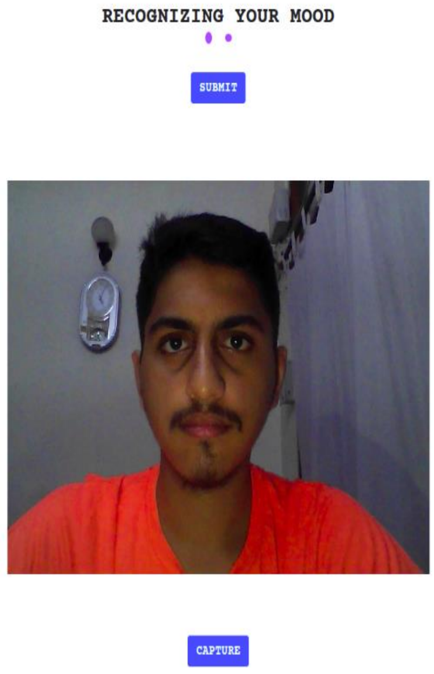

Fig 4.11 Emotion Recognition Module (ii)

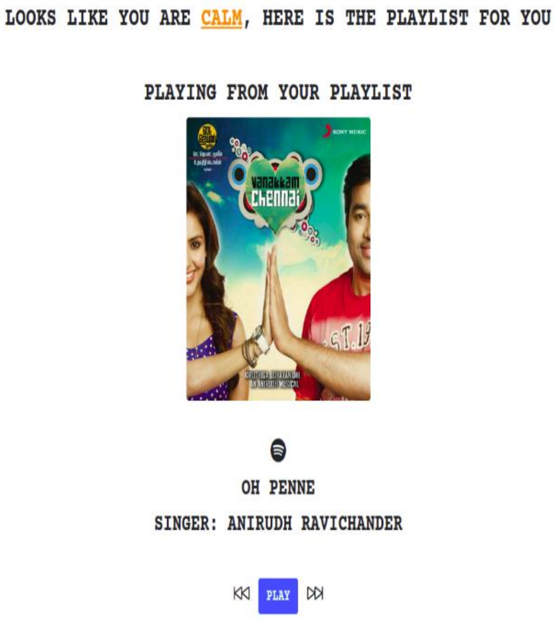

Fig 4.12 Suggested Playlist for Calm mood

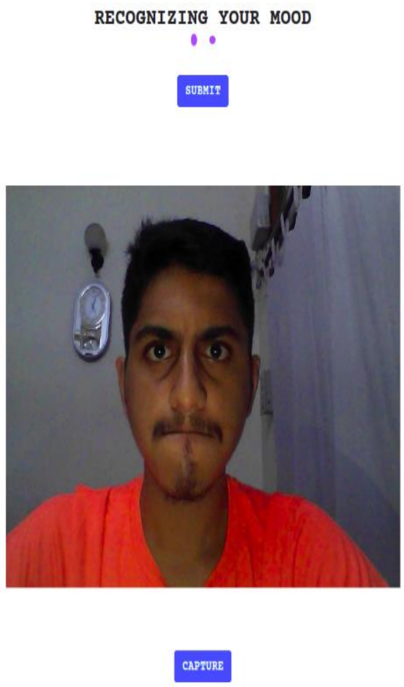

Fig 4.13 Emotion Recognition Module (iii)

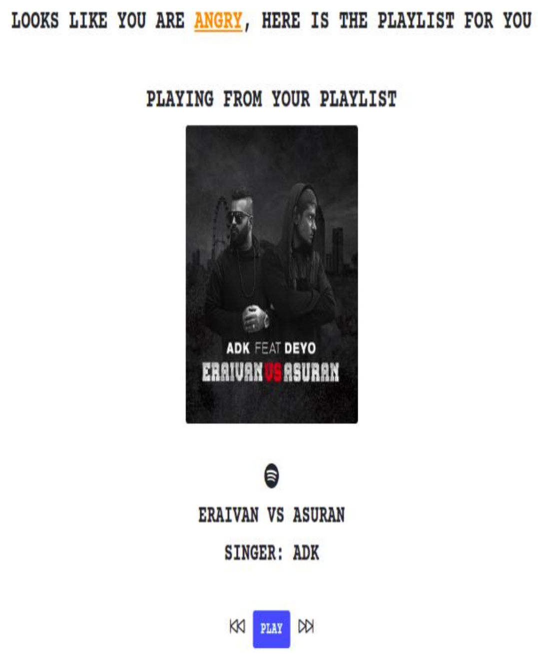

Fig 4.14 Suggested Playlist for Angry mood

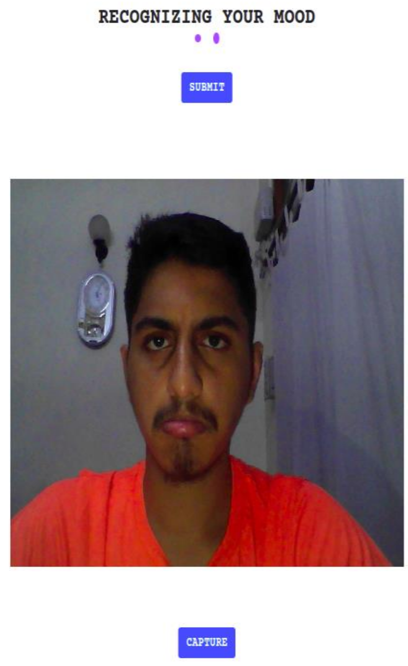

Fig 4.15 Emotion Recognition Module (iv)

Fig 4.16 Suggested Playlist for Sad mood

CHAPTER 5

5.1 Software Tools and Technologies:

5.1.1 OpenCV

OpenCV (Open Computer Vision Library) is a library of predefined functions and packages that aid in solving complex or common real time Computer Vision related problems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features, and use vector space and perform mathematical operations on these features.

The various packages that are used are:

i) Numpy:

NumPy is a python based external library that is imported to deal with data structures like arrays, sophisticated broadcasting functions, linear algebra, fourier transform, etc. NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows seamless and speedy integration with a wide variety of databases. OpenCV python uses Numpy to perform numerical operations. Any array declared using OpenCV is converted into or stored as a Numpy array. It also enables platform compatibility.

ii) CV2:

It mainly focuses on image processing, video capture and analysis. cv is a subclass inside cv2. Cv2 is used to import SciPy libraries for numerical processing.

iii) Time - This is used to compute time of execution and other such time complexities. In addition to this, it allows functionality like getting the current time, pausing the Program from executing, etc.

5.1.2 Python:

Python is an Object oriented language which when used with OpenCV, represents every image as a non-dimensional array of numerical values. The submodules of python such as SciPy are specifically concentrated for Image Processing. It is a high-level, general-purpose and a very popular programming language. Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting-edge technology in the Software Industry.

i) Math:

Python Math module is defined as the most famous mathematical functions, which includes trigonometric functions, representation functions, logarithmic functions, etc. Furthermore, it also defines two mathematical constants, i.e., Pie and Euler numbers, etc. This also performs statistical tasks.

ii) Spotipy:

Spotipy is a lightweight Python library for the Spotify Web API. With Spotipy you get full access to all of the music data provided by the Spotify platform. Spotipy supports all of the features of the Spotify Web API including access to all endpoints, and support for user authorization.

iii) Matplotlib:

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It is a visualisation library supported by python. It can be used to create dynamic, static and interactive visualisations using python. Matplotlib.pyplot is a collection of functions that makes matplotlib work like MATLAB. This library has been used extensively in plotting the graphs in the statistics page of the UI.

iv) Seaborn:

Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. It is a library mostly used for statistical plotting in Python. It is built on top of

Matplotlib and provides beautiful default styles and colour palettes to make statistical plots more attractive.

v) Flask:

Flask is a web framework that provides libraries to build lightweight web applications in python. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. Like most widely used Python libraries, the Flask package is installable from the Python Package Index (PPI).

vi) CSV Writer:

CSV writer class is used to insert data to the CSV file. This class returns a writer object which is responsible for converting the user's data into a delimited string. This is a very essential package to add new data to the existing data.

vii) WSGI:

The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI is the specification of a common interface between web servers and web applications. It is used to forward requests from a web server (such as Apache or NGINX) to a backend Python web application or framework. From there, responses are then passed back to the web server to reply to the requestor.

viii) scikit-learn

scikit - learn is one of the eminent machine learning packages available in python. It is largely written in python and is built upon NumPy, SciPy and Matplotlib. Scikit Learn focuses more on data modelling. It contains supervised models like Regression and Unsupervised models like Clustering to which the data set can be subjected to. It also contains a few predefined ensemble models which can be used to add up predictions of the individual model.

ix) Pandas

Pandas is a BSD - licensed library used to provide ready to use data structures and data analysis tools for Python. Pandas library is used to make data more interactive and flexible. The two major data structures supported by pandas are Series- a 1D data structure and a DataFrame - a 2D data structure. These data structures are used the most in various data analysis procedures.

xi) Tensorflow

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. TensorFlow was developed by the Google Brain team for internal Google use in research and production. TensorFlow can be used in a wide variety of programming languages, most notably Python, as well as Javascript, C++, and Java. This flexibility lends itself to a range of applications in many different sectors.

xii) Keras

Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple backend neural network computation.

xiii) Save and Load Models

To save weights manually, use tf.keras.Model.save_weights. By default, tf.keras—and the Model.save_weights method in particular—uses the TensorFlow Checkpoint format with a .ckpt extension. To save in the HDF5 format with a .h5 extension, refer to the Save and load models guide. The SavedModel format is another way to serialise models. Models saved in this format can be restored using tf.keras.models.load_model and are compatible with TensorFlow Serving.

5.1.3 PIP:

“ Preferred Installer Program ” or “ PIP install Packages' ' is used to install or uninstall Python packages required to operate OpenCV. ‘pip install spotipy’ is used to install the spotify package in the machine. PIP is a package management system used to install and manage software packages written in Python.

5.2 Hardware Devices Used:

A Webcam or USB camera is used to capture the real time picture of the user. The input image is captured using this camera and stored as an image frame. To implement the proposed idea, a HP wide vision HD-RGB camera with 0.92 MP and resolution of 1280×720 that is inbuilt within the laptop is used.

CHAPTER 6

RESULTS AND DISCUSSION

In this Chapter, the results obtained from the 5 models (Decision Tree, Gaussian Naive Bayes, Adaptive Boosting, eXtreme Gradient Boosting and MobileNet) are presented. The inferences obtained from the results are presented.

6.1 Metrics Used

Confusion Matrix

As the name suggests, confusion matrix gives us a matrix as output and describes the complete performance of the model.

●**True Positives** : The cases in which the model predicted YES and the actual output was also YES.

●**True Negatives** : The cases in which the model predicted NO and the actual output was NO.

●**False Positives** : The cases in which the model predicted YES and the actual output was NO.

●**False Negatives** : The cases in which the model predicted NO and the actual output was YES.

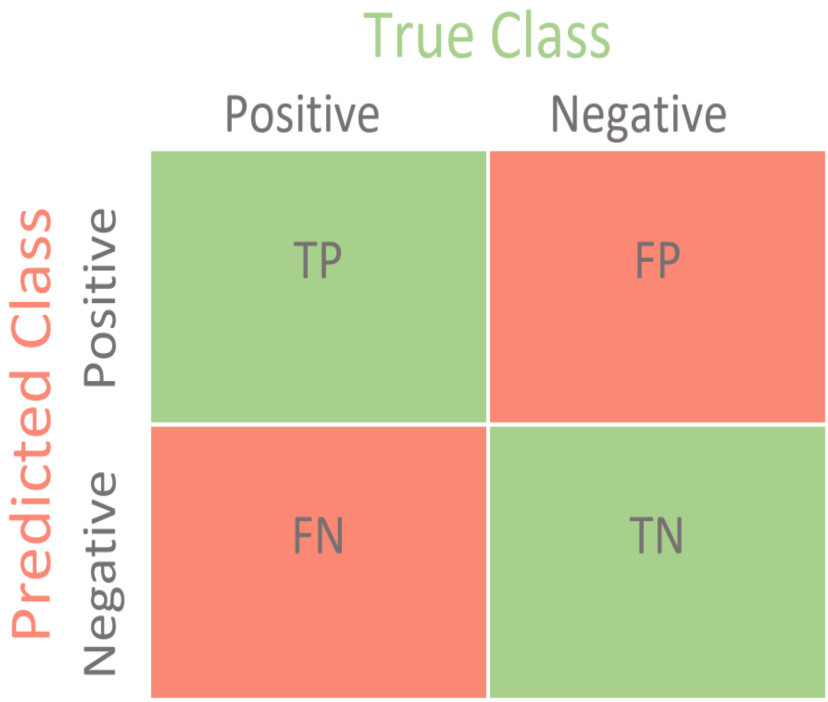

Fig 6.1 Confusion Matrix

6.1.1 Accuracy

Accuracy for the matrix can be calculated by taking average of the values lying across the main diagonal

$$\textbf{Accuracy} = \frac{TP + TN}{TS}$$

Where TP stands for number of true positive instances, TN stands for number of true negatives instances and TS stands for the total number of samples in the test set.

6.1.2 Area Under Curve

Area Under Curve (AUC) is one of the most widely used metrics for evaluation. It is used for binary classification problems. The AUC of a classifier is equal to the probability that the classifier will rank randomly chosen positive example higher than a randomly chosen negative example.

True Positive Rate (Sensitivity): True Positive rate is defined as ***TP / (FN + TP)***

TruePositiveRate = $TP/(FN + TP)$

Where TP stands for number of True Positive instances and FN stands for number of False Negative instances.

True Negative Rate (Specificity) : True Negative Rate is defined as **TN / (FP+TN).**

TrueNegativeRate = $TN/(FP + TN)$

where FP stands for number of False Positive instances, TN stands for True Negative instances in the test set.

False Positive Rate : False Positive Rate is defined as **FP / (FP+TN)**.

where FP stands for number of False Positive instances, TN stands for True Negative instances in the test set.

False Positive Rate and True Positive Rate both have values in the range [0, 1].

AUC has a range of [0, 1]. The greater the value, the better is the performance of our model.

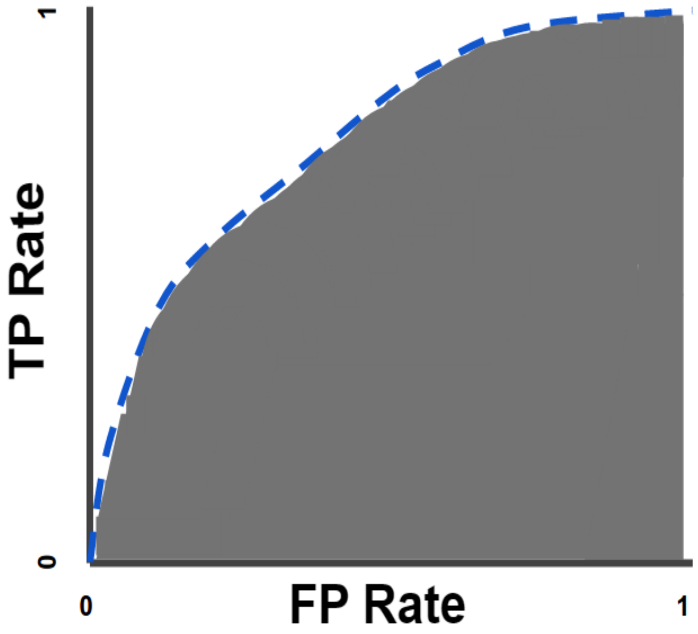

Fig 6.2 Area Under Curve

6.2 RESULTS

Classifier	Accuracy
Decision Tree	59.7%
Adaptive Boosting	52.3%
eXtreme Gradient Boosting	59.2%
Gaussian Naive Bayes	56.5%
Ensemble Model	89.6%

Table 6.1 Classifier Results

6.2.1 Machine Learning Algorithm 1 (Decision Tree)

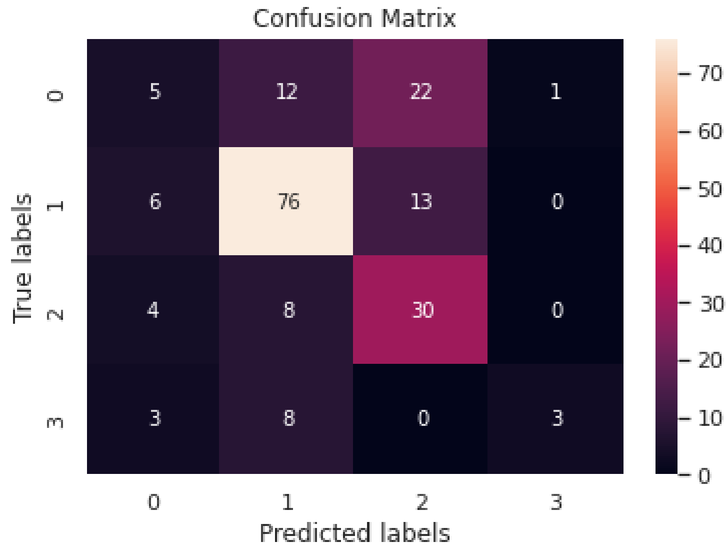

Fig 6.3 Decision Tree Results

The songs have been tagged with values indicating their emotion (calm=0, happy=1, sad=2 and angry=3). The above confusion matrix is the result of the predictions made by the Decision Tree Classifier. The values in the diagonal regions indicate correctly predicted emotion of the respective song. Many songs belonging to class 0 (calm) have been predicted as belonging to class 2 (sad), indicating that the classifier does not have a high accuracy and does not perform very well. The max_depth is set to 3 implying that there are 4 levels in the tree including the root node. The overall accuracy of Decision Tree Classifier is 59.7%.

6.2.2 Machine Learning Algorithm 2 (Adaptive Boosting)

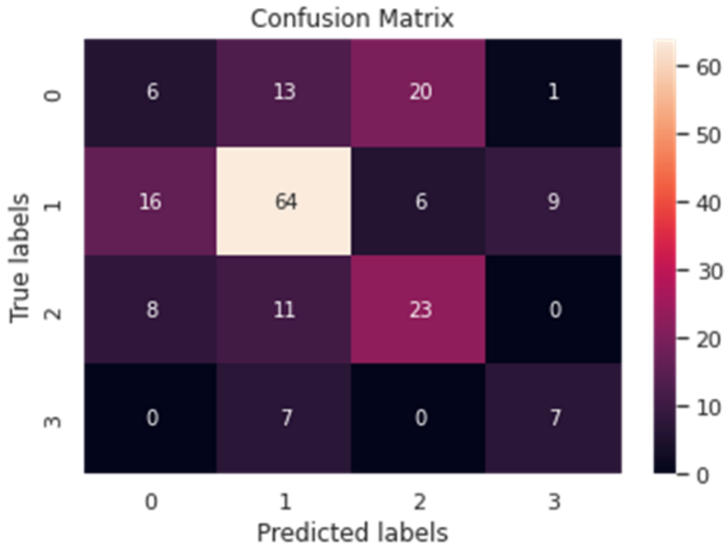

Fig 6.4 Adaptive Boosting Results

The above confusion matrix is the result of the predictions made by the Adaptive Boosting Classifier. The values in the diagonal regions indicate correctly predicted emotion of the respective song. Many songs belonging to class 0 (calm) have been predicted as belonging to class 2 (sad) and songs belonging to class 1 (happy) have been predicted as belonging to class 0 (calm) indicating that the classifier does not have a high accuracy and does not perform very well. The n_estimators are set to 16 and the learning rate to 0.97. The overall accuracy of Adaptive Boosting Classifier is 52.3%.

6.2.3 Machine Learning Algorithm 3 (Gaussian Naive Bayes)

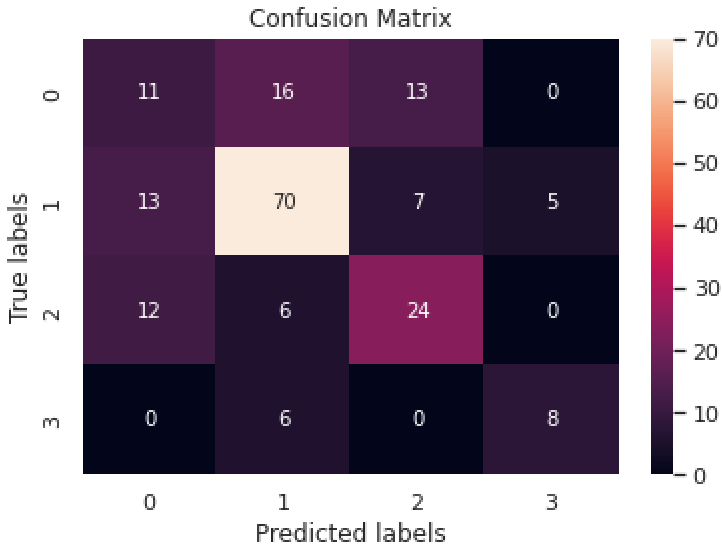

Fig 6.5 Gaussian Naive Bayes Results

The above confusion matrix is the result of the predictions made by the Gaussian Naïve Bayes Classifier. The values in the diagonal regions indicate correctly predicted emotion of the respective song. Many songs belonging to class 0 (calm) have been predicted as belonging to class 1 (happy) indicating that the classifier does not have a high accuracy and does not perform very well. The overall accuracy of Gaussian Naïve Bayes Classifier is 56.5%.

6.2.4 Machine Learning Algorithm 4 (eXtreme Gradient Boosting)

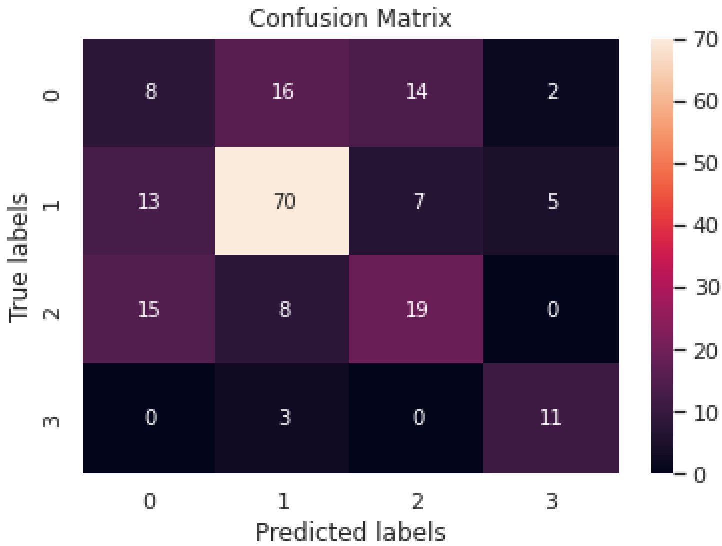

Fig 6.6 eXtreme Gradient Boosting Results

The above confusion matrix is the result of the predictions made by the eXtreme Gradient Boosting Classifier. The values in the diagonal regions indicate correctly predicted emotion of the respective song. Many songs belonging to class 0 (calm) have been predicted as belonging to class 1 (happy) and songs belonging to class 2 (sad) have been predicted as belonging to class 0 (calm), indicating that the classifier does not have a high accuracy and does not perform very well. The overall accuracy of eXtreme Gradient Boosting Classifier is 59.2%.

6.2.5 Ensemble Learning (Weighted Averaging)

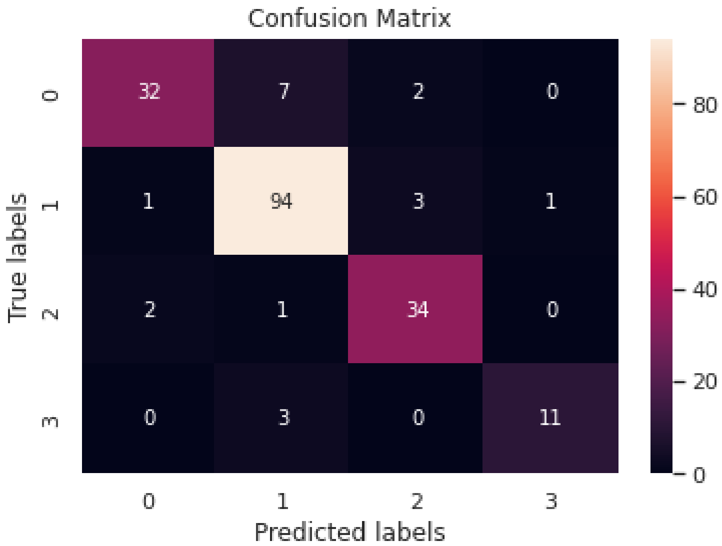

Fig 6.7 Ensemble Learning Results

The above confusion matrix is the result of the predictions made by the Ensemble Learning Classifier. The Ensemble approach yielded an accuracy of 89.6%. The top two classifiers (based on accuracy) were used for weighted averaging i.e, Decision Tree and eXtreme Gradient Boosting. The weight applied to the prediction of eXtreme Gradient Boosting classifier is 1.667 and the weight applied to Decision Tree classifier is 0.333.

6.2.6 Deep Learning Algorithm 1 (MobileNet)

	Accuracy	Precision	Recall
Training	75%	65.9%	66%
Validation	62%	65.8%	65.8%

Table 6.2 MobileNet Results

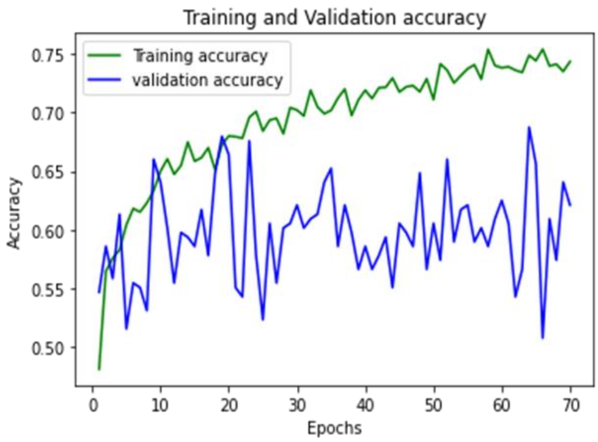

Fig 6.8 MobileNet Accuracy

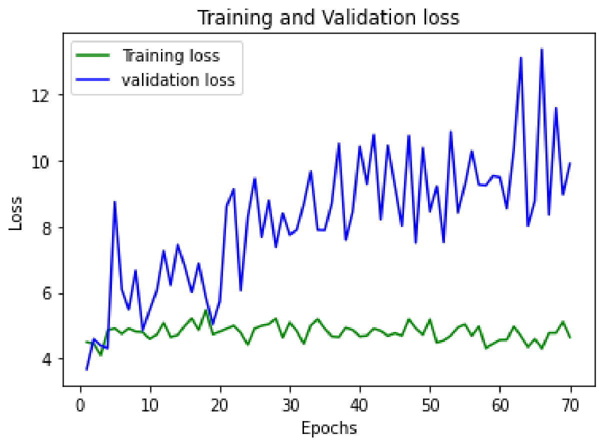

Fig 6.9 MobileNet Loss

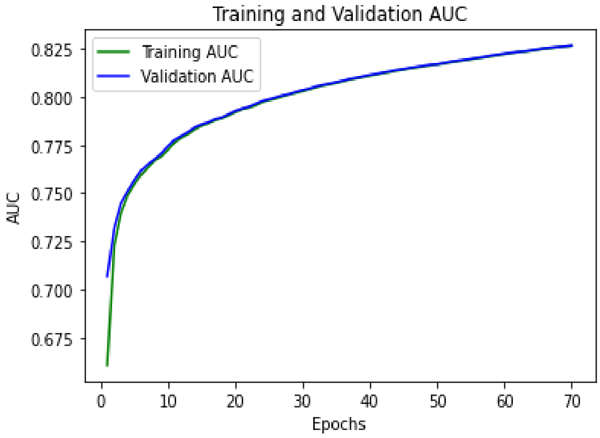

Fig 6.10 MobileNet AUC

The MobileNet model was used for Emotion Recognition purposes. The dataset used to train the model is FER2013 which is a collection of images which are tagged with a particular emotion. The emotions in which the model was trained are (calm, happy, sad and angry). The input images are of size (224x224) and in RGB format. This architecture was instantiated with ImageNet weights and uses Adam optimizer along with Categorical Cross Entropy as the loss function. MobileNet was trained for 70 epochs with batch size of 100. Maximum training accuracy of 75% is obtained along with a validation accuracy of 62%. The model took 2610 seconds to train.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

As part of this work, an emotion based song suggestion system was developed. To recognize the emotion of the user, a face recognition system was trained using the FER2013 dataset where the model makes use of the MobileNet Architecture which is a lightweight convolution neural network designed to be deployed in mobile applications. They are based on a streamlined architecture that uses depth wise separable convolutions to build lightweight deep neural networks that can have low latency for mobile and embedded devices.

MobileNet is a class of CNN that was open-sourced by Google, and therefore, this gives us an excellent starting point for training our classifiers that are insanely small and insanely fast. Mobile Nets split the convolution into a 3x3 depth-wise convolution and a 1x1 pointwise convolution. After getting the emotion of the user, this emotion was mapped with the songs dataset, which was prepared using the Spotify API call. In addition to this, a machine learning model was created using the ensemble learning of 2 classifiers namely Decision Tree and XG Boost (eXtreme Gradient Boosting) to predict the mood of the new song and was added to the existing songs dataset. An accuracy of 89.6% was obtained as a result of ensemble learning. By making use of Ensemble Learning, the accuracy had seen a major improvement by 29.9% from the classifier with maximum accuracy (59.7% obtained by Adaptive Boosting). A web application devised and developed using Flask was used to incorporate the above work and this presents the final playlist as a music player to the user.

7.2 Future Scope

Further work aims at improving the accuracy of the face recognition system which makes use of the MobileNet architecture trained using the FER2013 dataset to recognize the current emotion of the user. Currently, a training accuracy of 75% and a validation accuracy of 62% was obtained, the objective is to improve the accuracy in the future. In addition to this, the objective is to create a Database to store the credentials of the user. This in turn will save the playlist generated by the model in the database and the user can view it at any point of time.

CHAPTER 8

REFERENCES

- Bodarwé, Karl-Arnold, Jenny Noack, and Philipp Jean-Jacques. "Emotion-based music recommendation using supervised learning." In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*, pp. 341-344. 2015.
- Cardoso, Luís, Renato Panda, Rui Pedro Paiva. "MOODetector: A prototype software tool for mood-based playlist generation." *3º Simpósio de Informática-InForum 2011* (2011).
- Chen, Tianqi, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, and Kailong Chen. "Xgboost: extreme gradient boosting." *R package version 0.4-2* 1, no. 4 (2015): 1-4.
- Dietterich, Thomas G. "Ensemble learning." *The handbook of brain theory and neural networks* 2, no. 1 (2002): 110-125.
- Freund, Yoav, and Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting." *Journal of computer and system sciences* 55, no. 1 (1997): 119-139.
- Gajjar, Kunjal, and Siddhi Shah. "Mood based playlist generation for Hindi popular music: a proposed model." *International Journal of Computer Applications* 975 (2015): 8887.
- Gupta, D. L., A. K. Malviya, and Satyendra Singh. "Performance analysis of classification tree learning algorithms." *International Journal of Computer Applications* 55, no. 6 (2012).
- Hastie, Trevor, Robert Tibshirani, Jerome H. Friedman, and Jerome H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. New York: springer, 2009.

- Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).
- Jamdar, Adit, Jessica Abraham, Karishma Khanna, and Rahul Dubey. "Emotion analysis of songs based on lyrical and audio features." *arXiv preprint arXiv:1506.05012* (2015).
- James, H. Immanuel, J. James Anto Arnold, J. Maria Masilla Ruban, M. Tamilarasan, and R. Saranya. "Emotion based music recommendation system." *Emotion* 6, no. 03 (2019).
- LEE, JONGSEOL, KYOUNGRO YOON, DALWON JANG, SEI-JIN JANG, SAIM SHIN, and JI-HWAN KIM. "MUSIC RECOMMENDATION SYSTEM BASED ON GENRE DISTANCE AND USER PREFERENCE CLASSIFICATION." *Journal of Theoretical & Applied Information Technology* 96, no. 5 (2018).
- Nan, Yahui, Jianguo Ju, Qingyi Hua, Haoming Zhang, and Bo Wang. "A-MobileNet: An approach of facial expression recognition." *Alexandria Engineering Journal* 61, no. 6 (2022): 4435-4444.
- Panda, Renato, and Rui Pedro Paiva. "Music emotion classification: analysis of a classifier ensemble approach." In *5th International Workshop on Music and Machine Learning–MML 2012–in conjunction with the 19th International Conference on Machine Learning–ICML 2012*. 2012.
- Paul, Dip, and Subhradeep Kundu. "A survey of music recommendation systems with a proposed music recommendation system." In *Emerging technology in modeling and graphics*, pp. 279-285. Springer, Singapore, 2020.

- Rish, Irina. "An empirical study of the naive Bayes classifier." In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22, pp. 41-46. 2001.
- Shin, Il-hyung, Jaepyeong Cha, Gyeong Woo Cheon, Choonghee Lee, Seung Yup Lee, Hyung-Jin Yoon, and Hee Chan Kim. "Automatic stress-relieving music recommendation system based on photoplethysmography-derived heart rate variability analysis." In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6402-6405. IEEE, 2014.
- Wen, Xinglin. "Using deep learning approach and IoT architecture to build the intelligent music recommendation system." *Soft Computing* 25, no. 4 (2021): 3087-3096.
- Wishwanath, Champika HPD, Supuni N. Weerasinghe, Kanishka H. Illandara, A. S. T. M. R. D. S. Kadigamuwa, and Supunmali Ahangama. "A personalized and context aware music recommendation system." In *International Conference on Human-Computer Interaction*, pp. 616-627. Springer, Cham, 2020.
- Zhang, Yezi. "Music Recommendation System and Recommendation Model Based on Convolutional Neural Network." *Mobile Information Systems* 2022 (2022).