

EMOTION DETECTION TO UPGRADE CUSTOMER EXPERIENCE

A Project Report

Submitted by

**SOWMIYANARAYAN S [cb.en.u4cse18053]
ABISHECK KATHIRVEL [cb.en.u4cse18404]
APOORVAA S RAGHAVAN [cb.en.u4cse18409]
ARVIND T [cb.en.u4cse18410]**

Under the guidance of

Dr. Sabarish B.A

(Asst. Prof, Department of Computer Science & Engineering)

*in partial fulfillment for the award of the degree
of*

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

AMRITA VISHWA VIDYAPEETHAM



Amrita Nagar PO, Coimbatore - 641 112, Tamilnadu

May 2022

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF ENGINEERING, COIMBATORE-641 112



BONAFIDE CERTIFICATE

Certified that this project report titled “**EMOTION DETECTION TO UPGRADE CUSTOMER EXPERIENCE**” is the bonafide work of “**SOWMIYANARAYAN S [cb.en.u4cse18053], ABISHECK KATHIRVEL [cb.en.u4cse18404], APOORVAA S RAGHAVAN [cb.en.u4cse18409], ARVIND T [cb.en.u4cse18410],** ”, 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 any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. Sabarish B.A

GUIDE

Asst. Prof

Dept. of Computer Science & Engineering

SIGNATURE

Dr. (Col.) Kumar P. N.

HEAD OF THE DEPARTMENT

Dept. of Computer Science & Engineering

Signature of the Internal Examiner

Signature of the External Examiner

DECLARATION

I the undersigned solemnly declare that the project report **Emotion detection to upgrade Customer Experience** is based on my own work carried out during the course of our study under the supervision of Dr. Sabarish B.A, Assistant Professor, Computer Science & Engineering, and has not formed the basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgement have been made wherever the findings of others have been cited.



Sowmiyanarayan S[CB.EN.U4CSE18053]



Abisheck Kathirvel[CB.EN.U4CSE18404]



Apoorvaa S Raghavan[CB.EN.U4CSE18409]



Arvind T[CB.EN.U4CSE18410]

ABSTRACT

Growing technological development business platforms have shifted to web and cloud-based environments. Every customer needs privacy with respect to the purchases and services they seek. All business centers provide customer service support to clarify their queries in services or products. The present feedback system uses QA which ends in a biased feedback that leads to inaccurate customer experience. Interaction between the customer and executive gets affected by various reasons including inappropriate executive, extended wait time, etc. This project provides a framework that identifies the mood/emotion of the customer based on the initial chat query in a chat application and then uses sentiment analysis and routes the request to the appropriate technical expert to solve the issue which in turn improves the customer experience. After the call is established, using speech recognition the system identifies the emotion of the customer for the clarification/service provided by the person and grades them automatically. This framework for improving customer experience through sentiment analysis and emotion detection involves two phases: emotion-based call routing and auto-grading of service professionals.

The project proposes a Convolutional Neural Network(CNN) based architecture for Speech Emotion Recognition and classifies speech into angry, neutral, disappointed or happy. The outcome of this proposed system is to upgrade the customer experience by analyzing the calls at the customer care center. This system can also be used in various fields other than feedback systems like in the diagnosis of physiological disorders and counseling as emotion is an important topic in psychology and neuroscience.

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to our beloved Satguru **Sri Mata Amritanandamayi Devi** for providing the bright academic climate at this university, which has made this entire task appreciable. This acknowledgement is intended to be a thanksgiving measure to all those people involved directly or indirectly with my project. I would like to thank our Pro Chancellor **Bramachari Abhayamrita Chaitanya**, Vice Chancellor **Dr. Venkat Rangan. P** and **Dr. Sasangan Ramanathan**, Dean Engineering of Amrita Vishwa Vidyapeetham for providing us the necessary infrastructure required for the completion of the project. I express my thanks to **Dr. (Col) P.N. Kumar**, Chairperson of Department of Computer Science Engineering, **Dr. G. Jeyakumar** and **Dr. C. Shunmuga Velayutham**, Vice Chairpersons of the Department of Computer Science and Engineering for their valuable help and support during our study. I express my gratitude to my guide, **Dr. Sabarish B.A**, Assistant Professor, support and supervision. I feel extremely grateful to Review panel members **Dr. Vidhya Balasubramanian**, Professor, **Ms. Aswathi T**, Assistant Professor, **Dr. Dhanya N. M**, Assistant Professor(SG) for their feedback and encouragement which helped us to complete the project. I would also like to thank the entire staff of the Department of Computer Science and Engineering. I would like to extend my sincere thanks to my family and friends for helping and motivating me during the course of the project. Finally, I would like to thank all those who have helped, guided and encouraged me directly or indirectly during the project work. Last but not the least, I thank God for His blessings which made my project a success.

Sowmiyanarayan S[CB.EN.U4CSE18053]

Abisheck Kathirvel[CB.EN.U4CSE18404]

Apoorvaa S Raghavan[CB.EN.U4CSE18409]

Arvind T[CB.EN.U4CSE18410]

TABLE OF CONTENTS

ABSTRACT	iv
ACKNOWLEDGEMENTS	v
List of Tables	viii
List of Figures	ix
ABBREVIATIONS	x
List of Symbols	xi
1 Introduction	1
1.1 Problem Definition	1
1.1.1 Justification for the proposed problem	1
2 LITERATURE SURVEY	2
2.1 Emotion Recognition using speech recognition using Python	2
2.1.1 Commonalities/differences	2
2.2 Multi-Modal Emotion Recognition from Speech and Facial Expression Based on Deep Learning	3
2.2.1 Commonalities/differences	3
2.3 Ordinal Learning for Emotion Recognition in Customer Service Calls	3
2.3.1 Commonalities/differences	4
2.4 Speech Sentiment and Customer Satisfaction Estimation in Social Bot Conversations	4
2.4.1 Commonalities/differences	5
2.5 A Literature Review on Application of Sentiment Analysis Using Ma- chine Learning Techniques	5
2.5.1 Commonalities/differences	5
2.6 Data Set	6
2.7 Software/Tools Requirements	6
3 Proposed System	7
3.1 System Analysis	7
3.1.1 System requirement analysis	8
3.1.2 Module details of the system	10
3.2 System Design	14
3.2.1 Flow diagram of the system	14
3.2.2 Architecture diagram	15
4 Implementation and Testing	16
4.1 Implementation of module 1 (Speech Emotion Recognition)	16

4.1.1	Feature extraction	16
4.1.2	Training the model	17
4.2	Implementation of module 2 (Sentiment analysis)	17
4.2.1	Data gathering and feature extraction	17
4.2.2	Training the model	18
4.2.3	Testing the model	18
4.3	Implementation of module 3 (Customer care executive video-based emotion recognition)	19
4.3.1	Data gathering and feature extraction	19
4.3.2	Training the model	19
4.3.3	Testing the model	19
4.4	Implementation of module 4 (Integrating the two modules with chat application and calling system)	20
4.4.1	Algorithm Design	20
4.4.2	User Interface	21
5	Results and Discussion	25
5.1	Results from the SER model	25
5.1.1	Testing the model	25
5.2	Results from the sentiment analysis model	25
5.2.1	VADER tool	26
5.2.2	Observations from LSTM model	26
5.3	Results from the Customer care executive video-based emotion recognition	30
6	Conclusion	32
7	Future Enhancement	33

LIST OF TABLES

2.1	Dataset details	6
3.1	Modularization	10

LIST OF FIGURES

3.1	Flow diagram for speech emotion recognition	11
3.2	Flow diagram for sentiment analysis	12
3.3	Flow diagram for Customer care executive video-based emotion recog- nition	13
3.4	Flow diagram	14
3.5	Architecture diagram	15
4.1	Feature extraction code	16
4.2	MLP model	17
4.3	Tokenizing the words	18
4.4	Live testing of model	18
4.5	Tensed and Happy	19
4.6	score calculation	20
4.7	Tensed	21
4.8	Happy	21
4.9	customer UI	22
4.10	Positive Sentiment	22
4.11	Neutral Sentiment	22
4.12	Negative Sentiment	22
4.13	customer care UI	23
4.14	Video analysis score	23
4.15	Customer ID- a temporary ID assigned to a specific customer just to keep a track of them while they are in the system.	24
4.16	Overall Customer Care executive performance table	24
4.17	Performance of Customer care executive in each call with a specific customer	24
5.1	Observed accuracy in SER model	25
5.2	Testing SER model	26
5.3	Testing VADER tool	27
5.4	Accuracy changes after each epoch	27
5.5	Loss in LSTM	28
5.6	TPR vs FPR	29
5.7	Classification report	29
5.8	Testing the trained model in real time	29
5.9	Loss Accuracy Graphs	30
5.10	Epoch	31
5.11	Model Summary	31

ABBREVIATIONS

List of Symbols

Chapter 1

INTRODUCTION

1.1 Problem Definition

To provide an improved customer service experience by emotion-based call routing using Sentiment Analysis in a chat and automatic grading for the customer service executives using Speech Emotion Recognition.

1.1.1 Justification for the proposed problem

- The current customer feedback system is not accurate and the customer service executive has no idea of the mood of the customer.
- Today's KPIs use single questions at the end of customer interaction which mostly is biased by the outcome. Hence, we cannot identify the overall customer experience (the ups and downs) in the call.
- In the proposed system the customer interacts with the chat application where the system carries out sentiment analysis to identify the mood of the user.
- The call is then routed to the best executive that can handle such situations based on prior grades the model assigned to the customer service executive.
- In every call received at the call centre the system uses speech emotion recognition for automatic grading of the executive.

The following chapters in the report involve the entire summary of the project. Starting from the literature survey, observations from the survey and collection of datasets, the report progresses with the explanation of the proposed system with the help of diagrams as well as modules to be completed during the course of the project.

Chapter 2

LITERATURE SURVEY

2.1 Emotion Recognition using speech recognition using Python

Rohan et al. (2020) aimed at presenting a comprehensive review on emotion recognition through speech using various python libraries and comparison of various classifiers. The critical part in this study is to identify the properties and characteristics of the speed signal. The following were the different classifiers used to detect emotions after feature extraction: Support Vector Machine (SVM), Random Forest classifier (RF), Multiple Layer Perception (MLP), Keras, Gaussian Naive Byes classifier (GNB), K-Nearest Neighbor (KNN). The paper proposes that speech signal is divided into subparts by framing where each frame is typically 20ms. The features which are extracted from the frames are then classified into modules based on the algorithms mentioned above. The variation of emotions is clearly identified using speech signal spectrum. After comprehensive study, the paper concludes that the highest accuracy of 82 was obtained upon training with the random forest classifier model.

2.1.1 Commonalities/differences

This paper has details about the models which we plan to use in our speech emotion recognition module. The novelty in our approach is the use of multi-modal methods where the accuracy we obtain would be based on the combined results as a result of analyzing audio-visual features.

2.2 Multi-Modal Emotion Recognition from Speech and Facial Expression Based on Deep Learning

Cai et al. (2020) aimed at developing a multi-modal emotion recognition model using facial expression benefits from the complementary information of audio-visual features. In this paper the pre-processing is followed by use of deep neural networks to extract features. The paper uses an emotion database recorded by university of south California. It contains about 12 hours of audiovisual data, namely video, audio and voice text, facial expressions, which is 10 actors (5 females and 5 male actors) in the lines or impromptu scenes, leading to emotional expression. This paper uses confusion matrix as the evaluation index of the algorithm model. The paper uses CNN and LSTM to learn global and context high-level speech emotion features, and design multiple small-scale kernel convolution blocks to extract facial expression features.

2.2.1 Commonalities/differences

The multi-modal emotion recognition used here is using the speech and face expression features that our project proposes to use. The novelty in our idea is in making use of the data obtained from the model to carry out emotion-based call routing. We also plan to have separate modules for audio and visual emotion recognition unlike the method followed here.

2.3 Ordinal Learning for Emotion Recognition in Customer Service Calls

Han et al. (2020) aimed at using a Consistent Rank Logits (CORAL) based model for ordinal speech emotion recognition. The VG-Gish is reformed in a such a way that the consecutive outputs are designed to deal with binary speech emotion recognition subtasks and the final ordinal result that is generated is based on the series of subtasks. This paper uses a call center dataset divided into three partitions with an 8:1:1 split (i.e., 3,655 utterances for the training set, 458 for the development set, 424 for the test set). The database is created from recorded customer support calls in Chinese

(8 kHz, mono). It consists of 129 conversations in total. Each utterance is rated on a three-point scale: 1-non-negative, 2-Somewhat Negative, and 3-Obviously Negative. The paper successfully shows that the VG-Gish CORAL model improves performance as compared to the generic VG-Gish model. The main reason for this improvement is that the coral strategy imposes a greater penalty when larger classification errors are detected.

2.3.1 Commonalities/differences

This paper talks about using speech emotion recognition using models that we intend to use in our project. The novelty in our project is that we use sentiment analysis and facial expression-based emotion recognition and we are going to provide rating to the customer care executive on a scale of 5.

2.4 Speech Sentiment and Customer Satisfaction Estimation in Social Bot Conversations

Kim et al. (2020) aimed at showing the advantage of Bidirectional Long Short-Term Memory Networks (BLSTM) over static models through correlation analysis Customer Satisfaction (CSAT) and mean data. In addition, Kim Y, Levy J, Liu Y (2020) evaluated regression models to predict CSAT based on embeddings provided by automatic sentiment analysis system. Alexa Prize (AP) Social Data includes 6308 AP conversations and 93,671 utterances, corresponding to an average conversation's length of 14.85 utterances. The paper addresses: Activation (excited vs calm), Valence (positive vs negative) and satisfaction as the different sentiment dimensions. Kim Y, Levy J, Liu Y (2020) trained the acoustic and lexical using both acoustic and lexical cues at the utterance level. This paper uses static (SVM and RBF kernels) and temporal regression models to predict estimated CSAT score. Kim Y, Levy J, Liu Y (2020) proposed a method to automate generation of sentiment embeddings to construct model that predicts CSAT with very high accuracy.

2.4.1 Commonalities/differences

This paper talks about sentiment analysis and distinguishes different models and their accuracy which we intend to use in our project to decide on which approach to follow to implement our Chatbot based Sentiment Analysis module. The novelty in our idea is to create our own chatbot system in which we will be performing sentiment analysis with more sentiments other than activation and valence as mentioned in the paper.

2.5 A Literature Review on Application of Sentiment Analysis Using Machine Learning Techniques

Anvar Shathik and Krishna Prasad (2020) aimed at summarizing different machine learning techniques to identify emotion/sentiment. The paper also presents the different applications of sentiment analysis as a part of their literature survey. A number of sentiment analysis models and its application in various domains gives a clearer picture of the wide range of applications of sentimental analysis possible. The paper presents a tabulated summary of research works wherein the findings and research gaps in each paper are tabulated. This paper is actually a review and an extensive literature survey of existing applications of sentiment analysis and the different methods and machine learning models used for sentiment analysis.

2.5.1 Commonalities/differences

This paper is an effort to summarize numerous researches which have already been done in the field of sentiment analysis using machine learning. This will help us decide the approach we opt in our sentiment analysis model. The novelty in our idea is to create our own chatbot system in which we will be performing sentiment analysis.

2.6 Data Set

The summary of the chosen dataset is listed in Table 2.1

Dataset Name	IEMOCAP
Recorded by	University of Southern California
Dataset URL	https://sail.usc.edu/iemocap/
Area	Emotion
Data type	Audio-Visual data
Number of Instances	10 (5 male actors + 5 female actors)
Number of classes in output	4 (Excited, angry, sad, neutral)
Justification	The combination of audio and visual data in a single dataset makes it ideal to train models for emotion detection from speech and visual data in a call.
Keywords	Emotional, Multimodal, Acted, Dyadic

Table 2.1: Dataset details

2.7 Software/Tools Requirements

- Flask (Python library) - Front End
- Librosa (Python library)- used for processing and extracting features from the audio file.
- Soundfile- Sound file reading/writing
- Pyaudio- cross-platform audio input/output stream library
- Sklearn- Predictive data analysis

Chapter 3

PROPOSED SYSTEM

3.1 System Analysis

Multi-Layer Perceptron(MLP)

Multi-layer Perceptron (MLP) is a supervised learning algorithm that can learn a non-linear function approximator for either classification or regression with one or more non-linear layers, called hidden layers. For the training, we store the numerical values of emotions, features correspondingly in different arrays. These arrays are given as an input to the MLP Classifier that has been initialized. The Classifier identifies different categories in the datasets and classifies them into different emotions. The model will now be able to understand the ranges of values of the speech parameters that fall into specific emotions.

Longest Short-Term Memory(LSTM)

Primarily used for classification. LSTM contains a memory unit that stores the short-term findings of the model. This is particularly helpful in processing multiple sentences at the same time. If a GPU is available and all the arguments to the layer meet the requirement of the CuDNN kernel, the layer will use a cuDNN implementation which is faster.

RASA Open Source 2.0

Rasa Open Source 2.0 is an open-source machine learning framework that is used to automate the conversations based on text and voice inputs. It can understand the messages given as input, conduct conversation based on input, and can seamlessly connect across multiple messaging channels such as slack, messenger, and with APIs.

3.1.1 System requirement analysis

TensorFlow

TensorFlow we have used TensorFlow as the neural engine to develop the different deep learning models in our project. It is an open-source platform for machine learning and its libraries. TensorFlow provides us with high-level APIs like Keras which develops and train machine learning models very easily. It provides flexibility by allowing fast execution for immediate iterations and debugging.

Keras

Keras is an open-source software library that is used in our project and provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow libraries. It contains numerous implementations of commonly used neural network building blocks such as layers, trainers, activation functions, optimizers, etc. It contains a large array of tools that simplifies working with image and text data.

SciKit-Learn

Scikit-learn is a machine learning library for Python. It contains various classification, regression, and clustering algorithms including SVM, random forests, gradient boosting, k-means, and DB-SCAN, etc. It is designed to interoperate with various numerical and scientific libraries in python such as NumPy and SciPy.

Librosa

Librosa is a Python package for music and audio analysis. Librosa is basically used when we work with audio data like in music generation(using LSTM's), Automatic Speech Recognition. It provides the building blocks necessary to create the music information retrieval systems. It is used to extract mfcc, Mel, chroma features from the sound file.

HTML, CSS, JavaScript

We have used HTML, CSS, and Javascript to develop the front End of the project. HTML can be defined as the markup language to design the documents for display in a

web browser. It is usually assisted with tools such as Cascading Style Sheets (CSS) for designing and JavaScript for scripting. CSS is a simple tool to add styles such as fonts, colors, spacing, background to HTML web documents. JavaScript (JS) is a compiled programming language with first-class functions and is used to program the behavioral nature of HTML web pages.

Tokenizer

This class allows vectorizing a text corpus, by turning each text into either a sequence of integers each integer being the index of a token in a dictionary or into a vector where the coefficient for each token could be binary, based on word count.

Flask

We have used Flask to connect the back end with the front end of our project and host our application on a local server. Flask is a web framework written in Python and it does not need any particular tools or libraries. Flask doesn't enforce the project developer to follow any particular framework or adopt any dependencies.

Hardware requirements

Hardware requirements – Since most of the project time was remote, google colab was used (with GPU enabled) our models were trained with comparatively smaller datasets so they can run on any computer with any configuration. We took the advantage of the internal microphones of the personal computers while deploying the application. To give microphone access to google colab a javascript file is included to request and gain audio permission from chrome. Therefore, when run on an environment without the internal mic, it is necessary to add an mic to ensure smooth running.

3.1.2 Module details of the system

The project is divided into 4 main modules

S. No	Module	Module description
1	Speech Emotion Recognition	Using speech recognition, the system identifies the emotion of the customer for the clarification/service provided by the person and grades them automatically
2	Sentiment Analysis	Identifies the mood/emotion of the customer based on the initial chat query in a chat application
3	Salesperson video-based emotion recognition	Using video emotion recognition
4	Integrating the two modules with chat application and calling system	Combining SER and Sentiment analysis into a real-world system

Table 3.1: Modularization

The next few pages has the detailed explanation of each module

Module 1: Speech Emotion Recognition Module

In this module the goal is to develop a model which can recognize emotions of the speakers in the call at the call center. This module is prioritized as it is of high importance in the project. The results that we get from this module is used to grade executive and this will henceforth enable successful call routing.

To start with we plan to explore the existing speech emotion recognition modules and do a comparison based the results and accuracy level. The next step in this module is choosing the ideal model for our project. Once the model is chosen, we will move on to optimizing the parameters and increasing the accuracy level. The model will finally be tried out in real time wherein we will be the actors. This will be the final phase of this module.

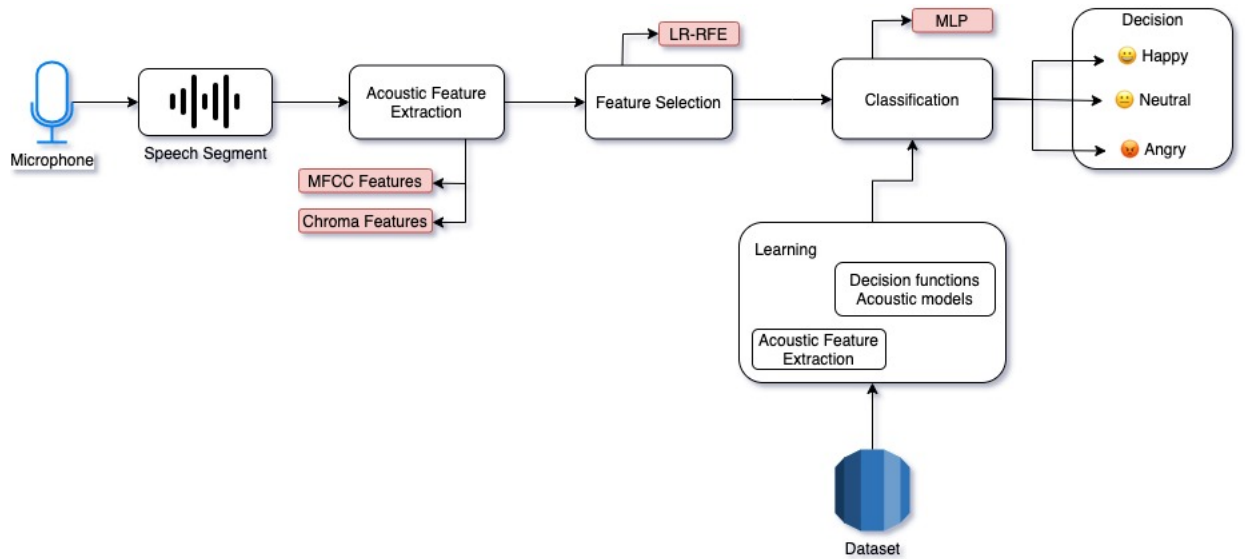


Figure 3.1: Flow diagram for speech emotion recognition

Details of Sub-modules

- **Feature Extraction** – The speech signal contains large number of parameters that reflect emotional characteristics. MFCC and chroma(pitch) are extracted.
 - **MFCC - Mel-frequency cepstrum coefficient** is the most representation of spectral property of voice signals. 15 high order features will be extracted from the 60-dimensional MFCC feature Vector
- **Feature Selection** – Done to reduce running time of learning algorithm using Recursive feature elimination(RFE) to select the best or reject the worst performing feature. This will improve classification Accuracy. LR-RFE or SVM-RFE can be used.
- **Classification** – Multi Layer Perceptron(MLP) Classifier will be used.

Module 2: Sentiment Analysis

This module is entirely based on the data we obtain from our chat application. Initially we plan to use a popular dataset for the purpose of analyzing sentiments in chatbot applications. The chatbot application is initially planned to be a simple website where the user interacts with a bot before he gets connected to the call center executive. A model must be built which can accurately predict the emotions of the customer based on the data from the chatbot. The process of call routing will be done based on the results from this module.

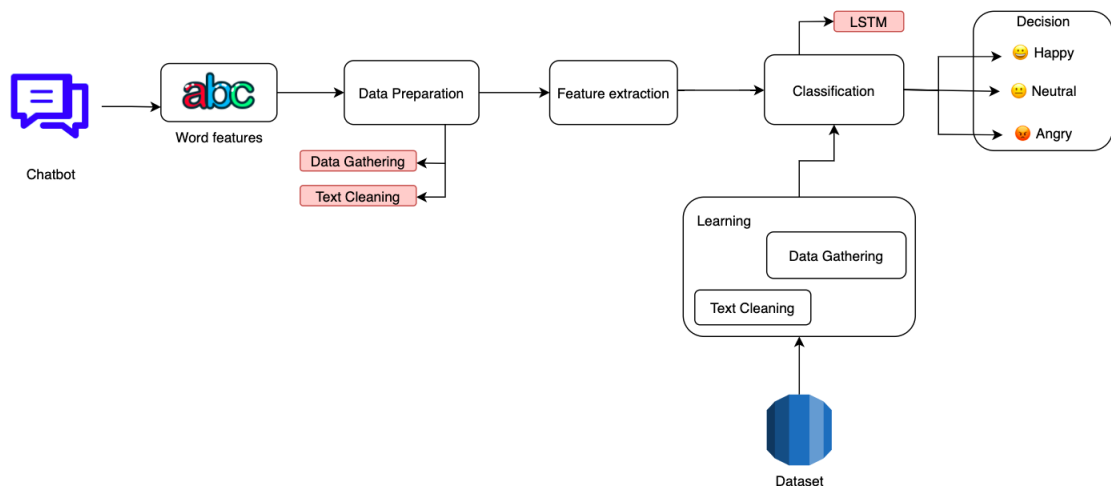


Figure 3.2: Flow diagram for sentiment analysis

Details of Sub-modules

The Sentiment Analysis module consists of three main sub modules namely Data Preparation, Build the Text Classifiers and Train the model.

- **Data Preparation – 1) Data Gathering 2) Text Cleaning**
 - **Data Gathering** – Data is required to analyze so gathering data from user reviews, social media using scraping tools, API's etc.
 - **Text Cleaning** – Removing stop words (a, and, or etc.), punctuations and whichever maybe irrelevant to the analysis.
- **Build the Text Classifier** – For sentiment analysis project, using LSTM layers with dropout mechanism to avoid over-fitting.
- **Train the model** – Train the Sentiment Analysis Model on the whole dataset for 5 epochs and a validation split of 20%

Module 3: Customer care executive video-based emotion recognition

This module brings in video-based emotion recognition wherein the emotion of the customer executive is identified by their real time facial expressions recorded by their webcam. This feature is in addition to speech emotion recognition in order to make our system more accurate.

The successful completion of this module hence ensure better results and reliability in the overall system that is built. The main tasks in this module is to explore the existing CNN models and techniques to dynamically recognize emotions and then map the model to identify emotions from a real-time video call.

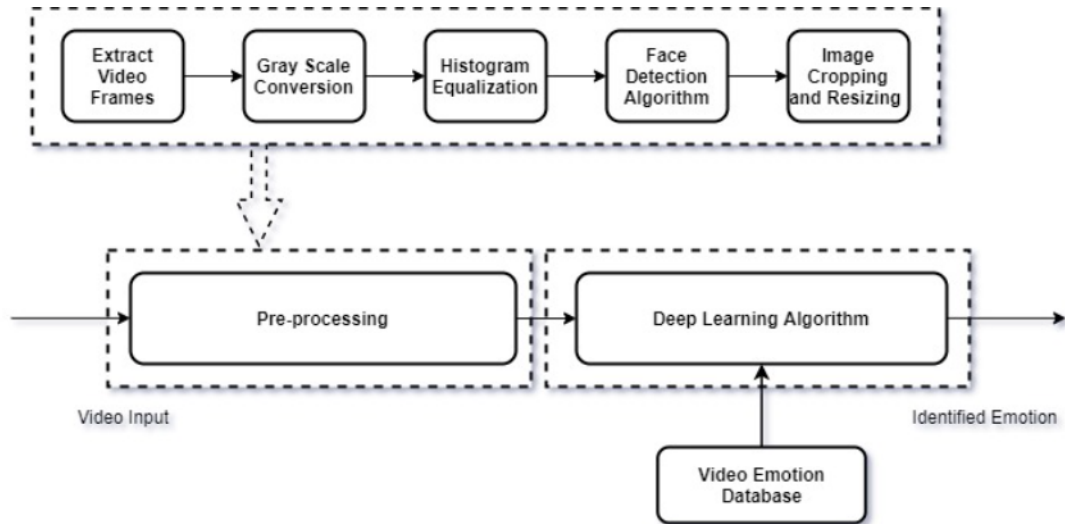


Figure 3.3: Flow diagram for Customer care executive video-based emotion recognition

Details of Sub-modules

The Customer care executive video-based emotion recognition module consists of three main sub modules namely Pre-Processing, Face Detection, Image cropping and resizing and Classification Model.

- **Preprocessing** – Frames are extracted from the input video which is converted to gray-scale on which Histogram equalization is done to loosen up the intensity scope of the picture which reduces and computational time.
- **Face Detection** – Emotions are featured mainly from face. CNN can be used to improve the exactness of face acknowledgement calculations which would incorporate features like width, surface etc.
- **Image cropping and resizing** – The face detected is cropped to obtain a broader and clearer facial image which reduces processing times. Optimization of selected features will be done to improve accuracy.
- **Classification Model** – CNN based model to classify emotions.

Module 4: Integrating the two modules with chat application and calling system

This is the final modal wherein the task is to combine the and integrate the work done in the previous three modules. The completion of this module indicates the successful completion the solution that is proposed.

Details of Sub-modules

- **Chat and Calling System** – RASA is an open source machine learning framework that is used to automate conversations which are trained in order to route to

the correct customer care personnel according to the mood found by the sentiment analysis module.

- **Ranking System** – A Ranking system which would be updated after every call would be made which would be looked upon to do dynamic call routing according to the mood of the user.
- For the Customer Module-1 will be used and for the executive Module-3 will be used and based on the found-out emotions the Ranking system will be updated.

3.2 System Design

3.2.1 Flow diagram of the system

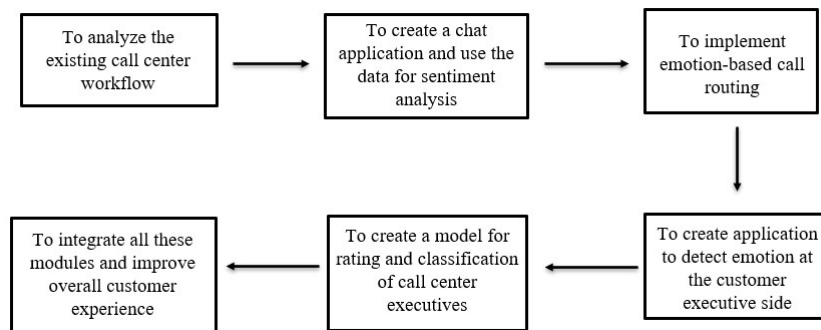


Figure 3.4: Flow diagram

3.2.2 Architecture diagram

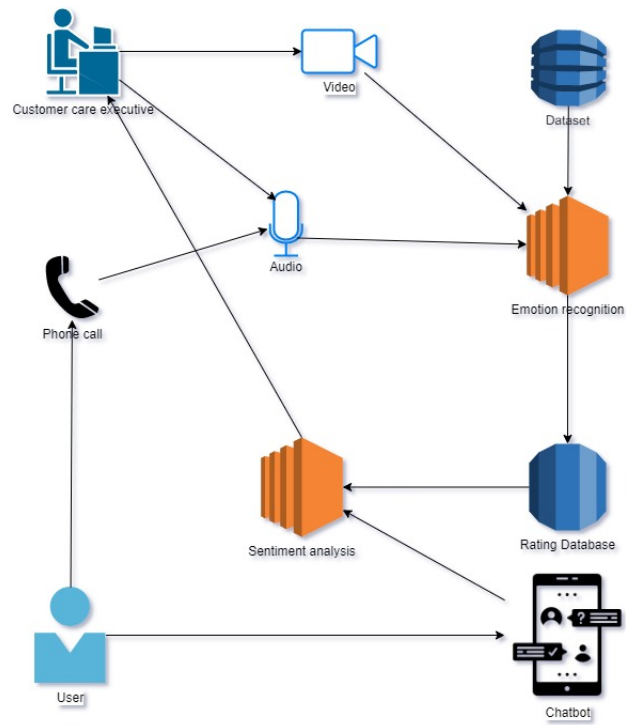


Figure 3.5: Architecture diagram

Chapter 4

IMPLEMENTATION AND TESTING

The implementation of the first two modules have been completed as planned in the timeline. The implementation details and the respective results are mentioned in the following sections.

4.1 Implementation of module 1 (Speech Emotion Recognition)

The goal to develop a model which can recognize emotions of the speakers in the call at the call center is implemented in this module.

4.1.1 Feature extraction

For the feature extraction the Librosa library is the main library that is used. The inbuilt feature of the library is used to split and obtain the chroma(pitch) , MFCC and mel spectrogram data. Mel spectrogram is a spectrogram that is converted to a Mel scale. The Mel scale basically mimics how the human ear works. A snippet of the core feature extraction code is below.

```
def extract_feature(file_name, mfcc, chroma, mel):
    with soundfile.SoundFile(file_name) as sound_file:
        X = sound_file.read(dtype="float32")
        sample_rate=sound_file.samplerate
        if chroma:
            stft=np.abs(librosa.stft(X))
            result=np.array([])
        if mfcc:
            mfccs=np.mean(librosa.feature.mfcc(y=X, sr=sample_rate, n_mfcc=40).T, axis=0)
            result=np.hstack((result, mfccs))
        if chroma:
            chroma=np.mean(librosa.feature.chroma_stft(S=stft, sr=sample_rate).T,axis=0)
            result=np.hstack((result, chroma))
        if mel:
            mel=np.mean(librosa.feature.melspectrogram(X, sr=sample_rate).T,axis=0)
            result=np.hstack((result, mel))
    return result
```

Figure 4.1: Feature extraction code

```

print(f'Features extracted: {x_train.shape[1]}')
Features extracted: 180

model=MLPClassifier(alpha=0.01, batch_size=256, epsilon=1e-08, hidden_layer_sizes=(300,), learning_rate='adaptive', max_iter=500)

model.fit(x_train,y_train)

/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimizer:
ConvergenceWarning,
MLPClassifier(alpha=0.01, batch_size=256, hidden_layer_sizes=(300,),
learning_rate='adaptive', max_iter=500)

y_pred=model.predict(x_test)

```

Figure 4.2: MLP model

4.1.2 Training the model

The model is trained using the MLP classifier. MLPClassifier relies on an underlying Neural Network to perform the task of classification. This model optimizes the log-loss function using stochastic gradient descent. We fine tuned the model by changing and comparatively choosing the ideal hyper-parameters like hidden layer sizes, learning rate and maximum iterations. The model was trained to identify 3 specific emotions: 'calm', 'happy', 'angry'. The number of features extracted from the audio file is 180. These features are then used by the MLP model to train and build the classifier. After hyper-parameter tuning and optimizing the model we were able to obtain an accuracy of 84.72 percentage.

4.2 Implementation of module 2 (Sentiment analysis)

The goal to develop a model which can recognize sentiment of the user by the means the messages received in the chatbot application that we create. Multiple ways of sentiment analysis were attempted and compared.

4.2.1 Data gathering and feature extraction

For the sentiment analysis module the dataset chosen is a vast one from Amazon wherein the reviews of customers are analyzed to find sentiment. It is a labelled dataset where the negative and positive reviews are labelled accordingly. The pre-processing steps involved removing null values and using clean text to choose the words and tokens of importance to us. A tokenizer is used to create dictionary where a unique integer is assigned for each word in the dataset.

```

num_words = 50000

tokenizer = Tokenizer(num_words=num_words,oov_token="unk")
tokenizer.fit_on_texts(X_train)
print(str(tokenizer.texts_to_sequences(['xyz how are you'])))

[[48311, 64, 14, 9]]

```

Figure 4.3: Tokenizing the words

```

y=[]
x = input("Enter String for Sentiment Analysis: ")
y.append(x)
xy = np.array(tokenizer.texts_to_sequences(y))
xy = pad_sequences(xy, padding='post', maxlen=100)
predictions = new_model.predict(xy)
if predictions[0][0] <0.5:
    print("Negative")
elif predictions[0][0] >0.78:
    print("Positive")
else:
    print("Neutral")

Enter String for Sentiment Analysis: Sad Worst Product
Negative

```

Figure 4.4: Live testing of model

4.2.2 Training the model

The model used for sentiment analysis is an LSTM based model to classify sentiments. LSTM contains a memory unit that stores the short-term findings of the model. This enables the previous words in the text also to be given equal importance when we try to predict the sentiment. The dropout mechanism is also used is a regularization technique for reducing overfitting in artificial neural networks. The hyper-parameter tuning is done changing the activation fuction and using appropriatæ kernel regularizer parameters. The model is then run for 10 epochs, until we get a stable accuracy pattern.

4.2.3 Testing the model

Live demo was performed and the results were observed.The model worked really well and predicted the sentiment with great accuracy. Evaluation metrics is discussed in the following chapter.

4.3 Implementation of module 3 (Customer care executive video-based emotion recognition)

The goal is to develop a model which can recognize the emotions of a person by analyzing the video.

4.3.1 Data gathering and feature extraction

For the video analysis the customer care executive's video feed during a call will be recorded and analyzed.

4.3.2 Training the model

A deep learning model is trained in image classification to identify emotions. It constantly identifies emotions in frames of a particular video.

4.3.3 Testing the model

A demo was performed using multiple recorded videos. The model worked well and captured emotions continuously in the frames of the video analysed.

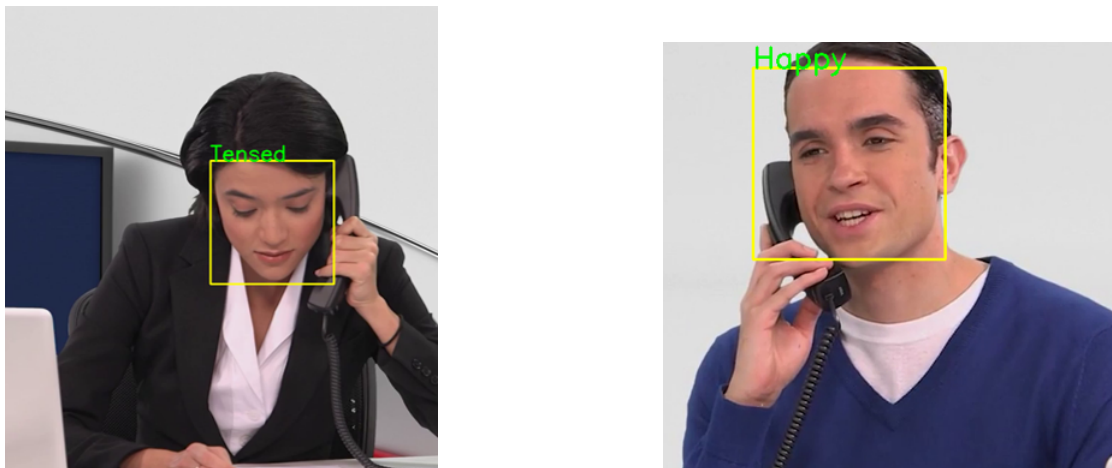


Figure 4.5: Tensed and Happy

4.4 Implementation of module 4 (Integrating the two modules with chat application and calling system)

This module uses the combination of the previous modules and gets a score which is then stored and ranked in a dynamic database that gets updated after every call placed in server.

4.4.1 Algorithm Design

The speech emotion recognition module is used to assign scores based on basic emotions(happy, tensed, and angry) that were analyzed and observed from the audio. The customer care executive video-based emotion recognition assigns a default score to the video initially and then deducts full or half points for angry and tense and adds points for happy emotions that were observed. These scores are then combined using a 40-60 weight-age since visual information is always superior to detect emotions from. But we can not neglect the audio emotions since the customers only source of emotion identification is through call. Hence 40-60 seemed to be a approximate weight-age.

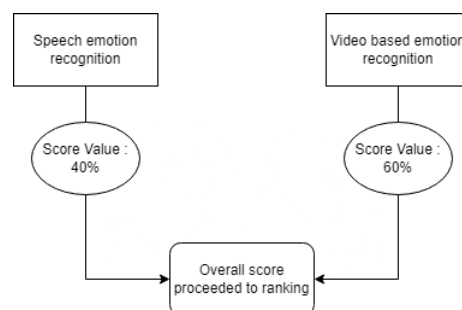
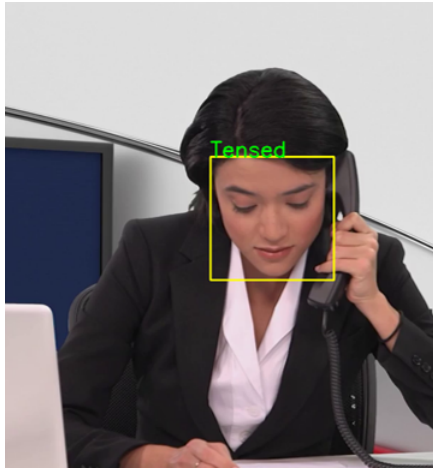


Figure 4.6: score calculation

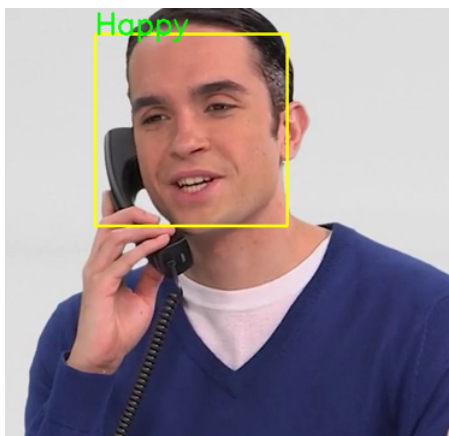


```
print("Tensed = ", (tensed/total)*100)
videoscore=5-(angry/total*10)+(happy
score=(audioscore*40+videoscore*60)
print("Score:",score)
cap.release()
cv2.destroyAllWindows()

C:\Users\ABISHE~1\AppData\Local\Temp
... 0.00112915 0.00115967 0.0008239
n error
mel=np.mean(librosa.feature.melsp

angry
Angry = 6.111111111111111
Happy = 2.4074074074074074
Tensed = 44.074074074074076
Score: 1.8555555555555556
```

Figure 4.7: Tensed



```
score=(audioscore*40+videoscore*60)/
print("Score:",score)
cap.release()
cv2.destroyAllWindows()

C:\Users\ABISHE~1\AppData\Local\Temp
... 0.00067139 0.00082397 0.00054932
n error
mel=np.mean(librosa.feature.melspe

happy
Angry = 0.0
Happy = 6.36604774535809
Tensed = 5.570291777188329
Score: 7.2148541114058355
```

Figure 4.8: Happy

4.4.2 User Interface

A flask app was designed to implement the integration of modules 1, 2, and 3. These include the speech emotion recognition, sentiment analysis, and video based emotion recognition. The modules 1 and 3 are used on the end of the customer care executive whereas the module 2 is solely to analyse any query the customer has on their side.

Customer end - Sentiment Analysis

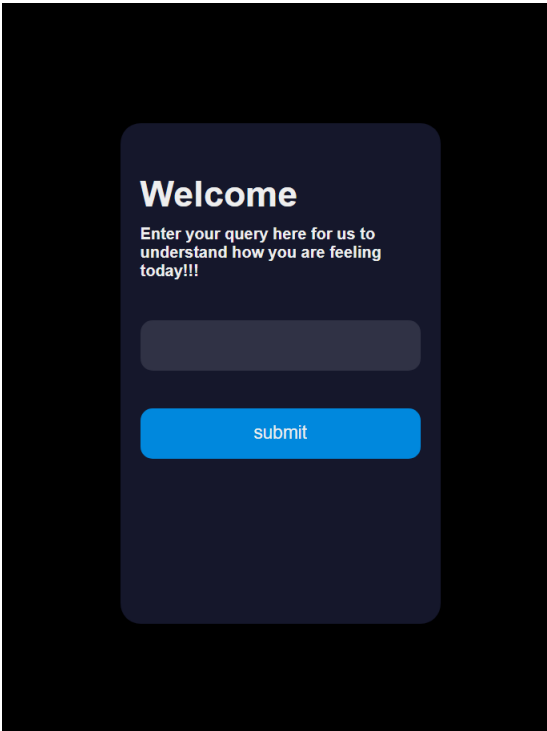


Figure 4.9: customer UI

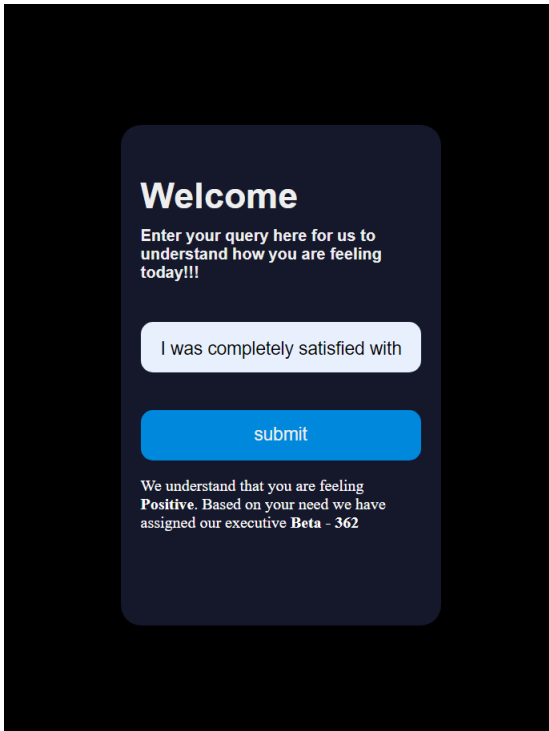


Figure 4.10: Positive Sentiment

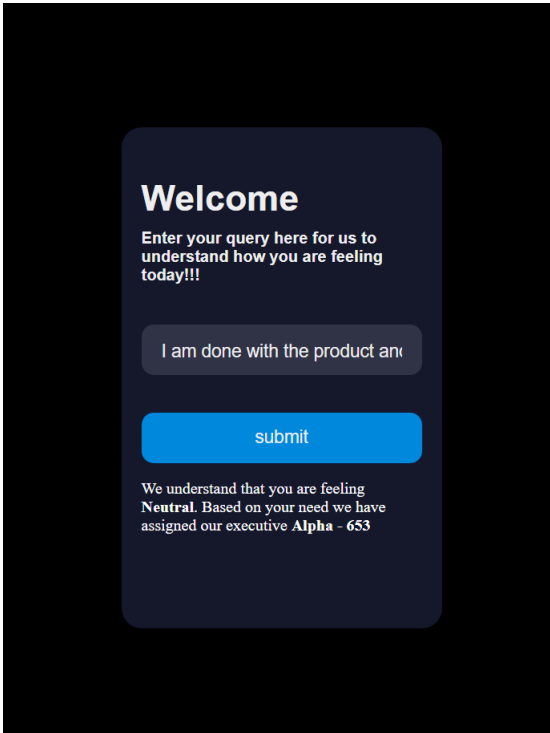


Figure 4.11: Neutral Sentiment

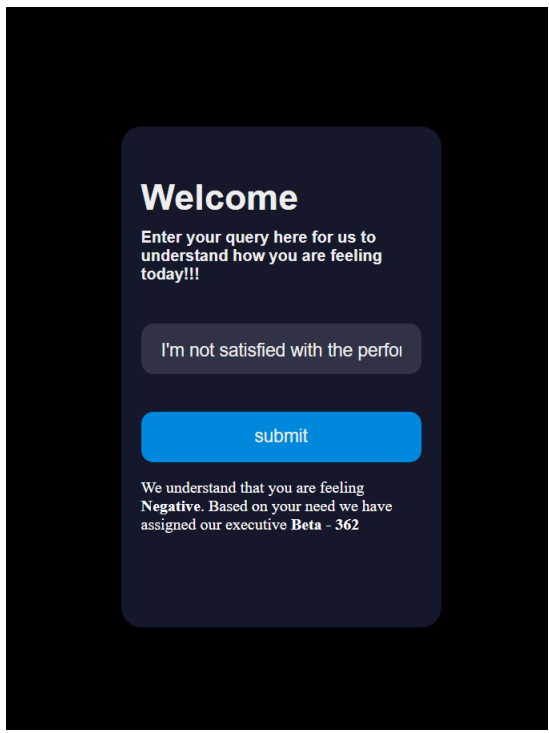
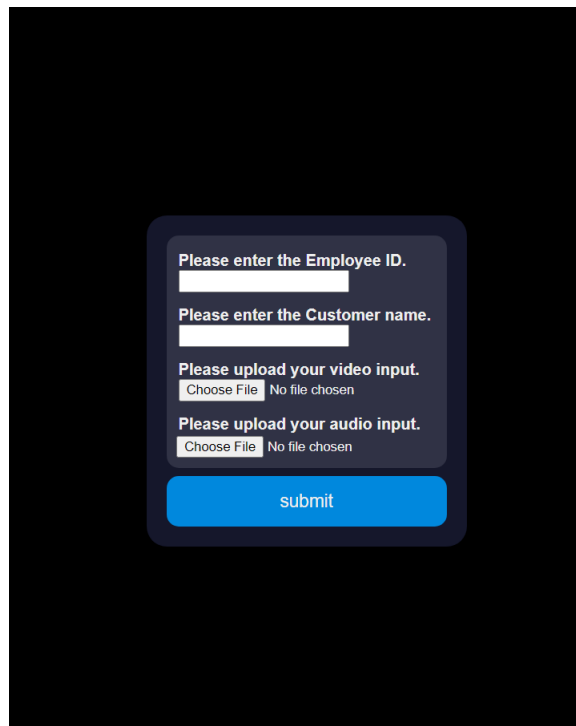


Figure 4.12: Negative Sentiment

Customer care executive end - Video Analysis



A dark-themed user interface for video analysis. It features a central light blue rounded rectangle containing four input fields and a submit button. The fields are labeled: 'Please enter the Employee ID.', 'Please enter the Customer name.', 'Please upload your video input.', and 'Please upload your audio input.'. Each upload field has a 'Choose File' button and the text 'No file chosen'. A blue 'submit' button is at the bottom.

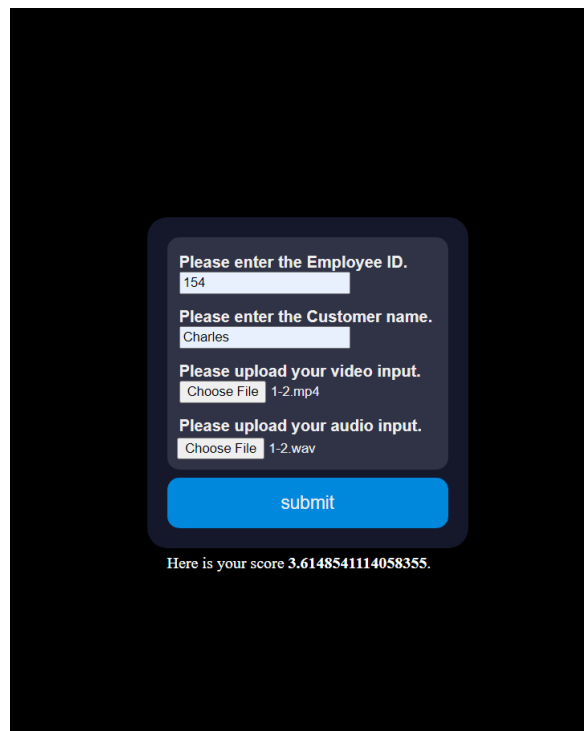
Please enter the Employee ID.

Please enter the Customer name.

Please upload your video input.
 No file chosen

Please upload your audio input.
 No file chosen

Figure 4.13: customer care UI



The same dark-themed user interface as Figure 4.13, but with data entered into the fields. The Employee ID field contains '154' and the Customer name field contains 'Charles'. The video upload field shows '1-2.mp4' and the audio upload field shows '1-2.wav'. Below the submit button, a message displays the analysis score.

Please enter the Employee ID.

Please enter the Customer name.

Please upload your video input.
 1-2.mp4

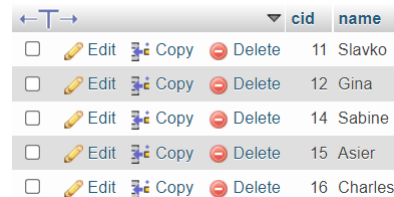
Please upload your audio input.
 1-2.wav

Here is your score 3.6148541114058355.

Figure 4.14: Video analysis score

Database

The front end is connected to a database that has tables that record the Customer's detail, the customer care executive's performance overall and their performance in a specific call with a customer they were assigned to.



					cid	name
<input type="checkbox"/>		Edit		Copy		Delete
					11	Slavko
<input type="checkbox"/>		Edit		Copy		Delete
					12	Gina
<input type="checkbox"/>		Edit		Copy		Delete
					14	Sabine
<input type="checkbox"/>		Edit		Copy		Delete
					15	Asier
<input type="checkbox"/>		Edit		Copy		Delete
					16	Charles

Figure 4.15: Customer ID- a temporary ID assigned to a specific customer just to keep a track of them while they are in the system.






















						eid	name	score	1
<input type="checkbox"/>		Edit		Copy		Delete	495	Morgan	9.654
<input type="checkbox"/>		Edit		Copy		Delete	362	Beta	7.674
<input type="checkbox"/>		Edit		Copy		Delete	265	Victor	7.362
<input type="checkbox"/>		Edit		Copy		Delete	653	Alpha	6.536
<input type="checkbox"/>		Edit		Copy		Delete	154	John	6.156
<input type="checkbox"/>		Edit		Copy		Delete	936	Charlie	3.201
<input type="checkbox"/>		Edit		Copy		Delete	656	Barns	1.956

Figure 4.16: Overall Customer Care executive performance table

						call_id	eid	cid	score	
<input type="checkbox"/>		Edit		Copy		Delete	1	653	9	8.16589
<input type="checkbox"/>		Edit		Copy		Delete	2	362	21	3.10236
<input type="checkbox"/>		Edit		Copy		Delete	3	154	10	7.21485
<input type="checkbox"/>		Edit		Copy		Delete	4	936	11	4.96584
<input type="checkbox"/>		Edit		Copy		Delete	5	936	12	7.69584
<input type="checkbox"/>		Edit		Copy		Delete	6	656	14	7.03659
<input type="checkbox"/>		Edit		Copy		Delete	7	495	15	8.98021

Figure 4.17: Performance of Customer care executive in each call with a specific customer

Chapter 5

RESULTS AND DISCUSSION

5.1 Results from the SER model

The dataset we chose initially had audio-visual data but it was not available publicly for usage and hence we took another dataset which consists audio data alone for the SER model. The model upon being trained with an MLP classifier was tested for accuracy and an accuracy of 84.72 percentage was obtained.

5.1.1 Testing the model

Upon observing an 84.72 percentage the next task undertaken was to do a real world testing of the model. To do this, the voice was recored in live usng a javascript code block in colab. The recored voice is then converted to an acceptable format using pydub library and this .wav that we create is then tested with the pre trained model using MLP classifer. The results were good and the model was able to predict the right emotion.

5.2 Results from the sentiment analysis model

The sentiment analysis was intially tried out through an existing library called VADER and after observing the results, we had moved on to building an LSTM model for better results.

```
y_pred=model.predict(x_test)

accuracy=accuracy_score(y_true=y_test, y_pred=y_pred)

print("Accuracy: {:.2f}%".format(accuracy*100))

Accuracy: 84.72%
```

Figure 5.1: Observed accuracy in SER model

```
[ ] record()

    'audio.flac'

[ ] pip install pydub

[ ] from pathlib import PurePath
    from pydub import AudioSegment

    file_path = PurePath("audio.flac")

    flac_tmp_audio_data = AudioSegment.from_file(file_path)

    flac_tmp_audio_data.export(file_path.name.replace(file_path.suffix, "1") + ".wav", format="wav")

    <_io.BufferedRandom name='audio1.wav'>

[ ] feature=extract_feature('/content/audio1.wav', mfcc=True, chroma=True, mel=True)

[ ] x_new=feature
    x_new

[ ] y_new=model.predict([x_new])
    y_new[0]

    'happy'
```

Figure 5.2: Testing SER model

5.2.1 VADER tool

The VADER (Valence Aware Dictionary and sEntiment Reasoner) was tested out initially to observe its performance. This tool is mainly used for sentiment analysis in social media and it works based on the lexical analysis performed. Lexical approach in this case looks at the score of each word in the sentence. Upon observing the results we felt that the way the sentence is presented doesn't impact the result much and it is only the specific words in the sentence that highly influence the scores. The flask app built to test out the VADER tool is in the figure below.

5.2.2 Observations from LSTM model

This model gave better results as it considered the sequence of words in the sentence rather than sticking onto single scores for specific words. In real life cases the words are not the only deciding factors in sentence and hence LSTM gave more acceptable results. After using a tokenizer an LSTM model was built and the results after 10 epochs were visualized (Figure 5.4 and 5.5).

Team 43

SENTIMENT ANALYSIS MODULE

Type to check sentiment

I am unhappy and irritated.

Check result

Result:

Negative 0.744 %

Neutral 0.256 %

Positive 0.0 %

Overall result is:

Sentence was finally rated as: Negative

Figure 5.3: Testing VADER tool

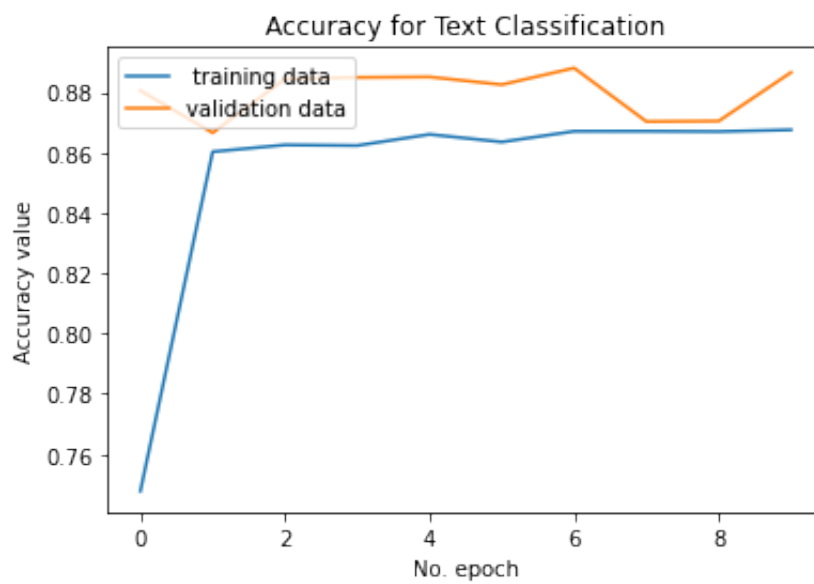


Figure 5.4: Accuracy changes after each epoch

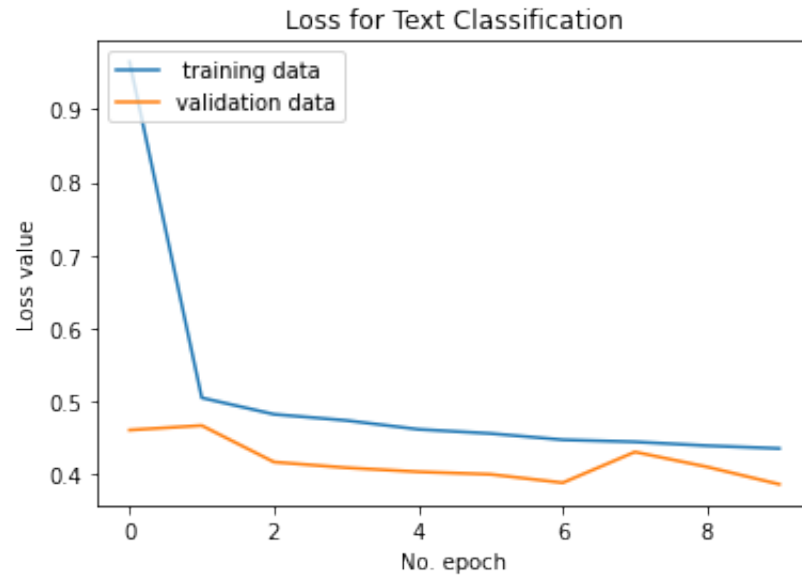


Figure 5.5: Loss in LSTM

A plot of the true positive rate vs the false positive rate is in figure 5.6

Thus the results observed were as expected and classification report gave us confidence to stick on with the chosen model. The report is in figure 5.7

Finally the live testing was carried out and the model return results as expected which marked the end of this module and the results and observations were recorded. A snippet from the live test carried out is in figure 5.8 .

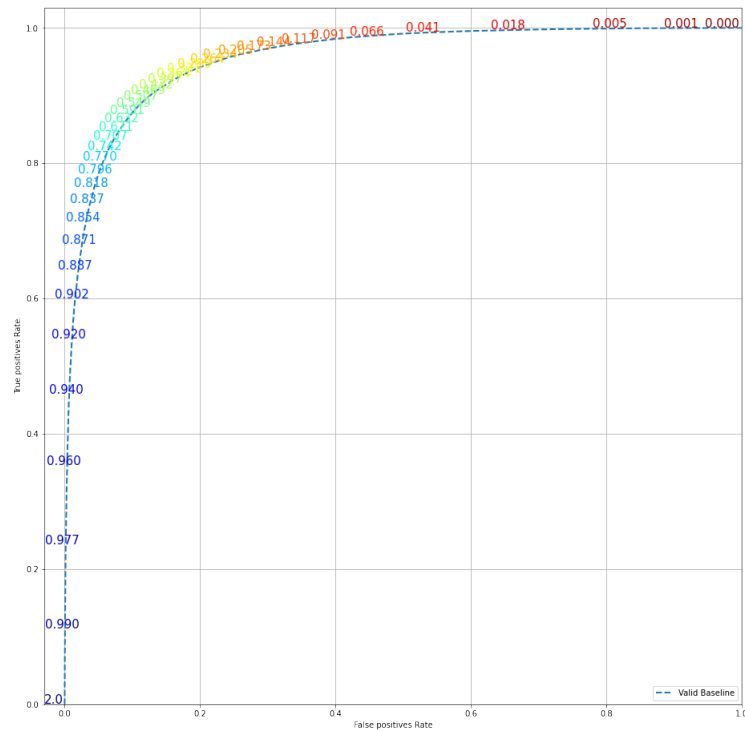


Figure 5.6: TPR vs FPR

	precision	recall	f1-score	support
0	0.83	0.95	0.89	200000
1	0.94	0.80	0.87	200000
accuracy			0.88	400000
macro avg	0.88	0.88	0.88	400000
weighted avg	0.88	0.88	0.88	400000

Figure 5.7: Classification report

```

y=[]
x = input("Enter String for Sentiment Analysis: ")
y.append(x)
xy = np.array(tokenizer.texts_to_sequences(y))
xy = pad_sequences(xy, padding='post', maxlen=100)
predictions = new_model.predict(xy)
if predictions[0][0] <0.5:
    print("Negative")
elif predictions[0][0] >0.78:
    print("Positive")
else:
    print("Neutral")

```

Enter String for Sentiment Analysis: Worst Customer care
Negative

Figure 5.8: Testing the trained model in real time

5.3 Results from the Customer care executive video-based emotion recognition

The video analysis was done using a deep learning model. The numerous epoch that were run, the model summary and the loss accuracy metrics are all done during the training phase and their output are as follows.

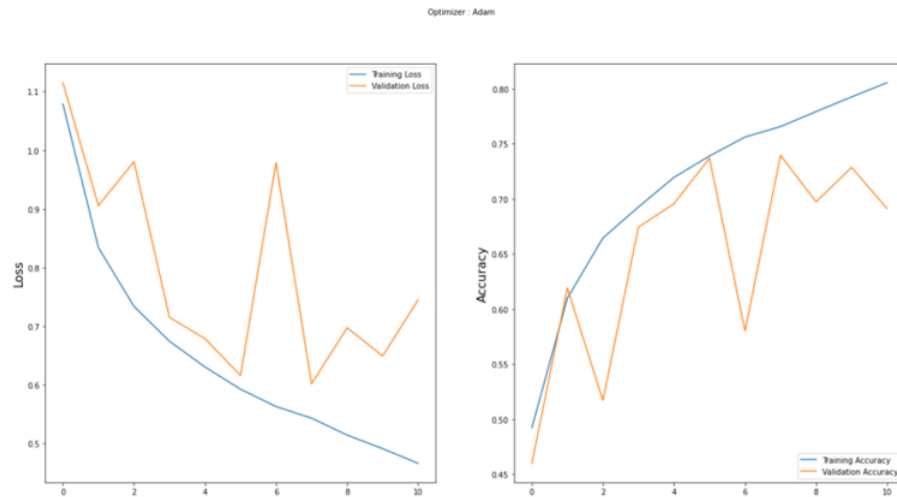


Figure 5.9: Loss Accuracy Graphs

```

Epoch 1/48
133/133 [=====] - ETA: 0s - loss: 1.0788 - accuracy: 0.4919
Epoch 1: val_accuracy improved from -inf to 0.45972, saving model to -\test1.h5
133/133 [=====] - 108s 1s/step - loss: 1.0788 - accuracy: 0.4919 - val_loss: 1.1151 - val_accuracy: 0.4597 - lr: 0.0010
Epoch 2/48
133/133 [=====] - ETA: 0s - loss: 0.8338 - accuracy: 0.6097
Epoch 2: val_accuracy improved from 0.45972 to 0.61890, saving model to -\test1.h5
133/133 [=====] - 19s 140ms/step - loss: 0.8338 - accuracy: 0.6097 - val_loss: 0.9053 - val_accuracy: 0.6189 - lr: 0.0010
Epoch 3/48
133/133 [=====] - ETA: 0s - loss: 0.7336 - accuracy: 0.6644
Epoch 3: val_accuracy did not improve from 0.61890
133/133 [=====] - 20s 148ms/step - loss: 0.7336 - accuracy: 0.6644 - val_loss: 0.9804 - val_accuracy: 0.5168 - lr: 0.0010
Epoch 4/48
133/133 [=====] - ETA: 0s - loss: 0.6741 - accuracy: 0.6926
Epoch 4: val_accuracy improved from 0.61890 to 0.67432, saving model to -\test1.h5
133/133 [=====] - 22s 160ms/step - loss: 0.6741 - accuracy: 0.6926 - val_loss: 0.7149 - val_accuracy: 0.6743 - lr: 0.0010
Epoch 5/48
133/133 [=====] - ETA: 0s - loss: 0.6382 - accuracy: 0.7195
Epoch 5: val_accuracy improved from 0.67432 to 0.69556, saving model to -\test1.h5
133/133 [=====] - 23s 174ms/step - loss: 0.6382 - accuracy: 0.7195 - val_loss: 0.6784 - val_accuracy: 0.6956 - lr: 0.0010
Epoch 6/48
133/133 [=====] - ETA: 0s - loss: 0.5922 - accuracy: 0.7389
Epoch 6: val_accuracy improved from 0.69556 to 0.73706, saving model to -\test1.h5
133/133 [=====] - 25s 188ms/step - loss: 0.5922 - accuracy: 0.7389 - val_loss: 0.6158 - val_accuracy: 0.7371 - lr: 0.0010
Epoch 7/48
133/133 [=====] - ETA: 0s - loss: 0.5627 - accuracy: 0.7561
Epoch 7: val_accuracy did not improve from 0.73706
133/133 [=====] - 23s 174ms/step - loss: 0.5627 - accuracy: 0.7561 - val_loss: 0.9786 - val_accuracy: 0.5708 - lr: 0.0010
Epoch 8/48
133/133 [=====] - ETA: 0s - loss: 0.5429 - accuracy: 0.7656
Epoch 8: val_accuracy improved from 0.73706 to 0.73950, saving model to -\test1.h5
133/133 [=====] - 20s 152ms/step - loss: 0.5429 - accuracy: 0.7656 - val_loss: 0.6915 - val_accuracy: 0.7395 - lr: 0.0010
Epoch 9/48
133/133 [=====] - ETA: 0s - loss: 0.5141 - accuracy: 0.7793
Epoch 9: val_accuracy did not improve from 0.73950
133/133 [=====] - 20s 150ms/step - loss: 0.5141 - accuracy: 0.7793 - val_loss: 0.6971 - val_accuracy: 0.6973 - lr: 0.0010
Epoch 10/48
133/133 [=====] - ETA: 0s - loss: 0.4909 - accuracy: 0.7926
Epoch 10: val_accuracy did not improve from 0.73950
133/133 [=====] - 24s 181ms/step - loss: 0.4909 - accuracy: 0.7926 - val_loss: 0.6488 - val_accuracy: 0.7285 - lr: 0.0010
Epoch 11/48
133/133 [=====] - ETA: 0s - loss: 0.4658 - accuracy: 0.8050
Epoch 11: restoring model weights from the end of the best epoch: 8.
Epoch 11: val_accuracy did not improve from 0.73950

Epoch 11: ReduceLROnDPlatform reducing learning rate to 0.000200000000049549026.
133/133 [=====] - 24s 170ms/step - loss: 0.4658 - accuracy: 0.8050 - val_loss: 0.7448 - val_accuracy: 0.6912 - lr: 0.00010
Epoch 11: early stopping

```

Figure 5.10: Epoch

Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 32)	640
batch_normalization_12 (Batch Normalization)	(None, 48, 48, 32)	256
activation_12 (Activation)	(None, 48, 48, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 24, 24, 32)	0
dropout_12 (Dropout)	(None, 24, 24, 32)	0
conv2d_2 (Conv2D)	(None, 24, 24, 128)	20480
batch_normalization_13 (Batch Normalization)	(None, 24, 24, 128)	512
activation_13 (Activation)	(None, 24, 24, 128)	0
max_pooling2d_2 (MaxPooling2D)	(None, 12, 12, 128)	0
dropout_13 (Dropout)	(None, 12, 12, 128)	0
conv2d_3 (Conv2D)	(None, 12, 12, 512)	100352
batch_normalization_14 (Batch Normalization)	(None, 12, 12, 512)	2048
activation_14 (Activation)	(None, 12, 12, 512)	0
max_pooling2d_3 (MaxPooling2D)	(None, 6, 6, 512)	0
dropout_14 (Dropout)	(None, 6, 6, 512)	0
conv2d_4 (Conv2D)	(None, 6, 6, 512)	277376
batch_normalization_15 (Batch Normalization)	(None, 6, 6, 512)	2048
activation_15 (Activation)	(None, 6, 6, 512)	0
max_pooling2d_4 (MaxPooling2D)	(None, 3, 3, 512)	0
dropout_15 (Dropout)	(None, 3, 3, 512)	0
flatten_1 (Flatten)	(None, 4096)	0
dense_1 (Dense)	(None, 256)	117984
batch_normalization_16 (Batch Normalization)	(None, 256)	1024
activation_16 (Activation)	(None, 256)	0
dropout_16 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 512)	117984
batch_normalization_17 (Batch Normalization)	(None, 512)	2048
activation_17 (Activation)	(None, 512)	0
dropout_17 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 1)	513

Total params: 4,476,472
Trainable params: 4,472,707
Non-trainable params: 3,765

Figure 5.11: Model Summary

Chapter 6

CONCLUSION

To sum up the overall workflow of our project, the system starts with the user entering a query they are with in a form and submitting it. As the session starts the submitted query is pre-processed and sent to Long Short-Term Memory(LSTM) classifier. In the pre-processing stage, the text is cleaned removing punctuation and numbers. The pre-processed text is sent as input to the LSTM classifier to identify the emotion. The identified emotion is used as input to the call routing module which will identify the appropriate technical expert to solve the issue which in turn improves the customer experience. After the call is established, the speech signals are pre-processed and the required features are extracted and then sent to MLP classifier which does speech recognition and thus the system identifies the emotion of the customer for the clarification/service provided by the person and grades them automatically. In feature extraction, Mfcc, Mel, and Chroma features are extracted from the sound file and stored in hstack. From the data set, the required emotions are selected and the MLP model is trained using the features extracted. Hyper-parameters were tweaked to get higher accuracy. This classifier returns the emotion of the customer during the call at certain intervals which will be used in the Integration module to automatically grade the customer service personnel.

Chapter 7

FUTURE ENHANCEMENT

The future developments of this project should revolve around increasing the accuracies of the models in the speech emotion recognition, sentiment analysis, and video analysis. The User interface could be made a bit more user friendly and platform independent. The routing of the calls could be automated to decrease the latency.

REFERENCES

1. Anvar Shathik, J. and Krishna Prasad, K. (2020). “A literature review on application of sentiment analysis using machine learning techniques.” *International Journal of Applied Engineering and Management Letters(IJAEML)*.
2. Cai, L., Dong, J., and Wei, M. (2020). “Multi-modal emotion recognition from speech and facial expression based on deep learning.” *Chinese Automation Congress (CAC)*.
3. Han, W., Jiang, T., Li, Y., Schuller, B., and Ruan, H. (2020). “Ordinal learning for emotion recognition in customer service calls.” *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
4. Kim, Y., Levy, J., and Liu, Y. (2020). “Speech sentiment and customer satisfaction estimation in socialbot conversations.” *arXiv:2008.1237*.
5. Rohan, M., Swaroop, K., Mounika, B., Renuka, K., and Nivas, S. (2020). “Emotion recognition through speech signal using python.” *International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*.