*A project report on*

# VOICE BASED EMOTION DETECTION ANALYSIS

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

## *Integrated M.tech (software Engineering)*

***by***

**Mannava Siri Chandana – 19mis7114**

**Mallela Jahnavi – 19mis7094**

**Galla Harika – 19mis7077**


**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**July , 2022**

## CERTIFICATE

This is to certify that the Summer Project/ summer Internship work titled **"Voice based emotion detection"** that is being submitted by **Mannava Siri Chandana (19mis7114)** is in partial fulfillment of the requirements for the award of **Master of Technology (Integrated 5 Year**) **software engineering**, is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

Signature

Ch Anil Carie

The thesis is satisfactory

**Approved by**

**PROGRAM CHAIR** **DEAN**

M. Tech. SE School Of Computer Science and Engineering

# ABSTRACT

Sentiment analysis has evolved over past few decades, most of the modern applications are using voice based feature for their applications.

Our proposed system is leveraging this voice sentiment analysis technology to take care of the aged people. Our proposed system will detect the different emotion from the audio and if the emotion is panic then the alert will be send to the guardian so that they can help them on time.

We perform sentiment analysis on audios to detect the emotions of the individual speakers. We analyzed different techniques to perform speaker sentiment analysis to find efficient algorithms to perform this task.

This proposed system will working as an alerting system which will give the alert to the guardians whenever it will find “panic” emotion in the audio**.**

The emotions that we are going to detect are Happy , calm , panic , disgust

# ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude *Dr.Anil Carie* , Scope, VIT-AP for his constant guidance, continual encouragement, understanding; more than all, he taught me patience in my endeavor .My association with him is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of area.

In jubilant mood I express ingeniously my whole-hearted thanks to all teaching staff and members working as limbs of our university for their not-self-centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

# CONTENTS

# INTRODUCTION

## 1.1 PROJECT PLAN

## 1.2 ABOUT THE PROJECT

During pandemic people who are alone in their home are feeling depressed and loneliness, and especially those persons who are aged. In this period guardian of the aged people don't know exactly about the emotion of that person, even there is not a type of system which can detect and give the alerts to the care taker of the aged people.

There are some system in the market that is available which can detect the body motions via camera so that guardian can see and get alert like they are fall or they are in some kind of problem. But this system is solving real world problem which can be seen from the eyes. But during pandemic people are suffered more from the mental issues compare to the physical issue.

In such scenario, we need a system which can detect the emotion of the aged people so that guardian can detect the emotion of the aged people and get alert whenever "panic" emotion is detected.

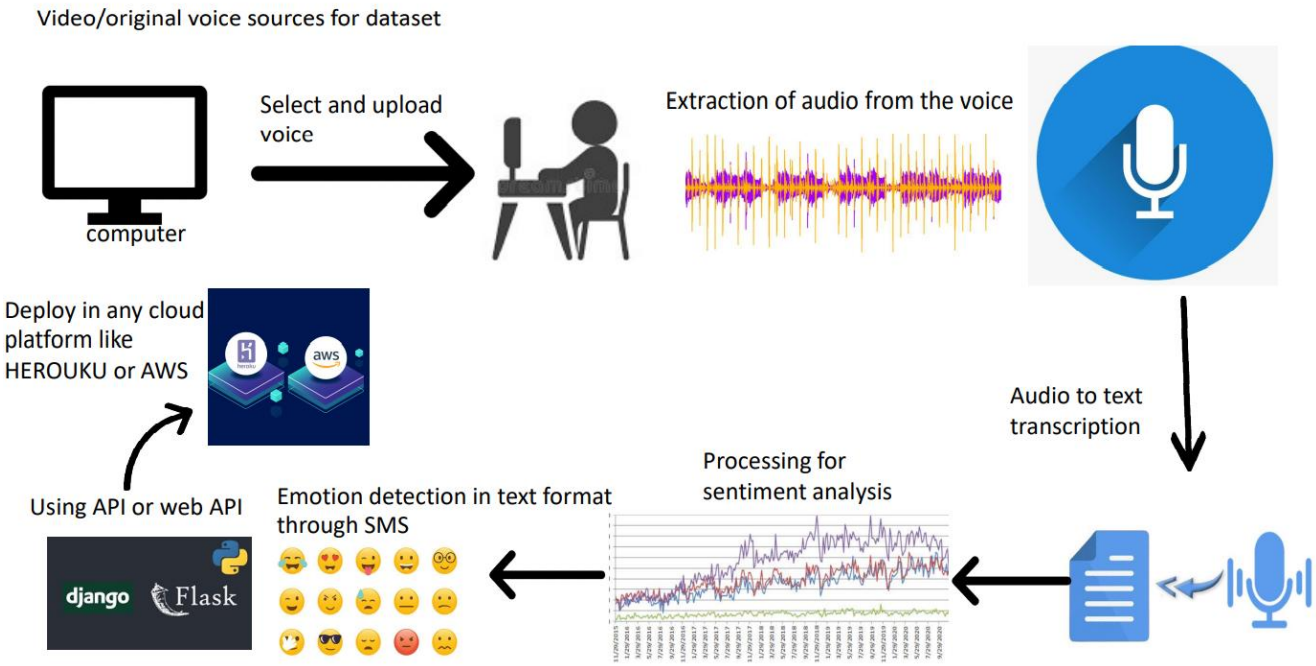
Our proposed system came up with the emotion detection feature which is implemented with the help of sentiment analysis technology to get the different emotions in the voice.

Our system will detect the emotions first and then filter that out and if the emotion is "panic" then the SMS will be sent to the guardian as an alert.

## 1.3 PURPOSE AND SCOPE

Our proposed system's aim is to provide a system which can help guardian in taking care of the aged people. Our proposed system will mainly focus on detecting the emotions of the people using voice input and give the alerts via SMS.
Our proposed system can be use in the different scenarios like parents can keep track the emotion of their children they will get alert whenever there will be detection of emotion like – panic, happy, sad, etc.
So this system has wide scope and it is generic type of voice sentiment analysis system with SMS alert feature.

## 1.4 OVER VIEW OF PROJECT



## 1.5 OBJECTIVE OF PROJECT

- To help aged peoples.
- To solve real world problem.
- To get better understanding of aged people thoughts.
- To help physically disabled people.

## 1.6 TECHNOLOGIES USED

- Django
- Python
- HTML
- CSS
- Java Script
- Machine learning

# 2 . BACKGROUND WORK

## 2.1 Literature Survey of our project

In paper “Audio and Text based multimodal sentiment analysis using features extracted from selective regions and deep neural networks” An improved multimodal approach to detect the sentiment of products based on their multi-modality natures (audio and text) is proposed. The basic goal is to classify the input data as either positive or negative sentiment.

Learning utterance-level representations for speech emotion and age/gender recognition. Accurately recognizing speaker emotion and age/gender from speech can provide better user experience for many spoken dialogue systems. In this study, we propose to use Deep Neural Networks (DNNs) to encode each utterance into a fixed-length vector by pooling the activations of the last hidden layer over time.

The paper “Towards Real-time Speech Emotion Recognition using Deep Neural Networks” proposes a real-time SER system based on end-to-end deep learning. Namely, a Deep Neural Network (DNN) that recognizes emotions

From paper “Machine Learning and Sentiment Analysis Approaches for the Analysis of Parliamentary Debates” the

author seeks to establish the most appropriate mechanism for conducting sentiment analysis with respect to political

debates; Firstly so as to predict their outcome and secondly to support a mechanism to provide for the visualisation of

such debates in the context of further analysis. To this end two alternative approaches are considered, a classification-

based approach and a lexicon-based approach. In the context of the second approach both generic and domain specific sentiment lexicons are considered. Two techniques to generating domain-specific sentiment lexicons are also proposed:

(i) direct generation
(ii) adaptation
(iii) The first was founded on the idea of generating a dedicated lexicon directly from labelled source data.

This paper “Techniques and Applications of Emotion Recognition in Speech” gives a brief overview of the current state of the research in this area with the aim to underline different techniques that are being used for detecting emotional states in vocal expressions. Furthermore, approaches for extracting speech features from speech datasets and machine learning methods with special emphasis on classifiers are analysed. In addition to the mentioned techniques, this paper also gives an outline of the areas where emotion recognition could be utilised such as healthcare, psychology, cognitive sciences and marketing.

The reported comparison indicates that the attitude of

speakers can be effectively predicted using sentiment mining. The author then goes on to propose a framework, the Debate Graph Extraction (DGE) framework, for extracting debate graphs from transcripts of political debates. The idea is to represent the structure of a debate as a graph with speakers as nodes and\exchanges" as links. Links between nodes were established according to the exchanges between the speeches. Nodes were labelled according to the \attitude"(sentiment) of the speakers, \positive" or \negative", using one of the three proposed sentiment mining approaches.

# 3. SOFTWARE DEVELOPMENT LIFE CYCLE

## 3.1 REQUIREMENT ANALYSIS PHASE

The Requirements Analysis Phase begins when the previous phase objectives have been achieved. Documentation related to user requirements from the Concept Development Phase and the Planning Phase shall be used as the basis for further user needs analysis and the development of detailed requirements. Multiple-release projects require only one iteration of the Requirements Analysis Phase, which should involve requirements definition for all planned releases.

The objective of this phase is to define in more detail the system inputs, processes, outputs and interfaces. At the end of this phase the system's processes will be defined at the functional level, meaning the functions to be performed will be known, but not necessarily how they will be performed. Unless specifically constrained by the Project Charter, Requirements Analysis should not consider the computer programs, files and data streams.

Requirements Analysis will identify and consider the risks related to how the technology will be integrated into the standard operating procedures. Requirements Analysis will collect the functional and system requirements of the business process, the user requirements and the operational requirements (e.g., when operational what is necessary to keep the system up and running).

## 3.2 System Requirement Specification

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function (in a market-driven project, these roles may be played by the marketing and development divisions). Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules. Used appropriately, software requirements specifications can help prevent software project failure.

The software requirements specification document lists sufficient and necessary requirements for the project development. To derive the requirements, the developer needs to have clear and thorough understanding of the products under development. This is achieved through detailed and continuous communications with the project team and customer throughout the software development process.

- **Purpose**

The purpose of this document is to give a detailed description of the requirements for the "Amazing Lunch Indicator" (ALI) software. It will illustrate the purpose and complete declaration for the development of system. It will

also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

## 3.3 Hardware and Software Requirement

1. **Hardware Specification**

    3.3.1.1 RAM 4 GB

    3.3.1.2 GPU

2. **Software Requirements:**

    3.3.1.3 Python

    3.3.1.4 PyCharm

    3.3.1.5 Browser to Test

## 3.4 Functional Requirement

In Software engineering and systems engineering, a functional requirement defines a function of a system or its

component. A function is described as a set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioural requirements describing all the cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements which impose constraints on the design or implementation.

As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability. Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system.

## 3.5 Non-Functional Requirement

In systems engineering and requirements engineering, a non-functional requirement is

a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviour. They are contrasted with functional requirements that define specific behaviour or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually Architecturally Significant Requirements.

Broadly, functional requirements define what a system is supposed to do and non-functional requirements define how a system is supposed to be. Functional requirements are usually in the form of, an individual action or part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, process and control functional model or IPO Model. In contrast, non-functional requirements are in the form of, an overall property of the system as a whole or of a particular aspect and not a specific function. The system's overall properties commonly mark the difference between whether the development project has succeeded or failed.

Non-functional requirements are often called "quality attributes" of a system. Other terms for non-functional requirements are "qualities", "quality goals", "quality of service requirements", "constraints" and "non-behavioural requirements". Informally these are sometimes called the "ileitis", from attributes like stability and portability. Qualities—that is non-functional requirements—can be divided into two main categories:

1. Execution qualities, such as safety, security and usability, which are observable during operation.
2. Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the system.

### 3.5.1 Feasibility Study

Feasibility study is made to see if the project on completion

will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness.

A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources.

Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

Following are its feature

## 3.6 TECHNICAL FEASIBILITY

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures.

Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

## 3.7 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project,

which will give best, return at the earliest.

One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development

## 3.8 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?

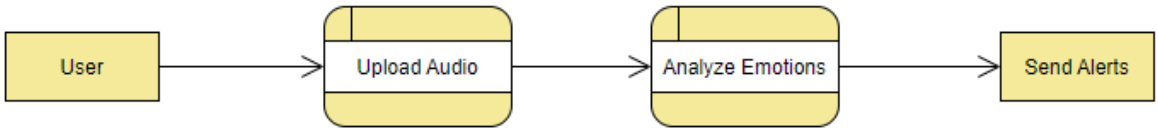
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is behaviourally feasible.

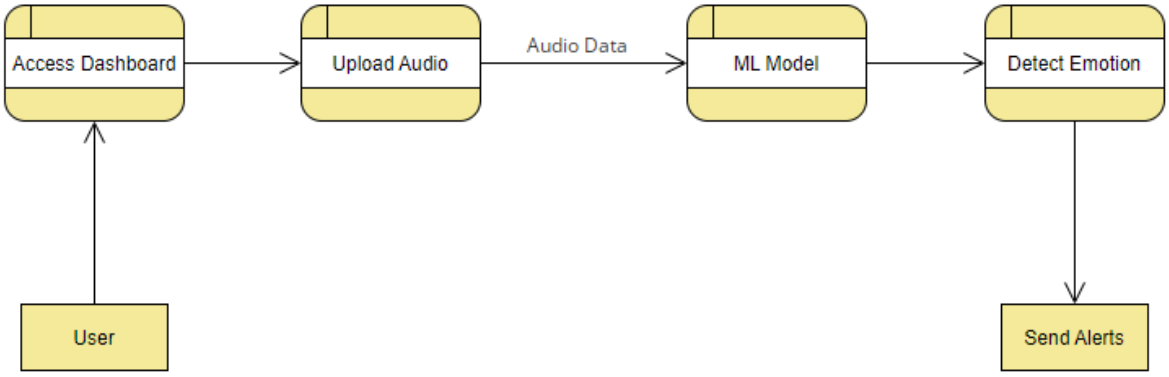
# 4. DIAGRAMS

## 4.1 DATA FLOW DIAGRAM

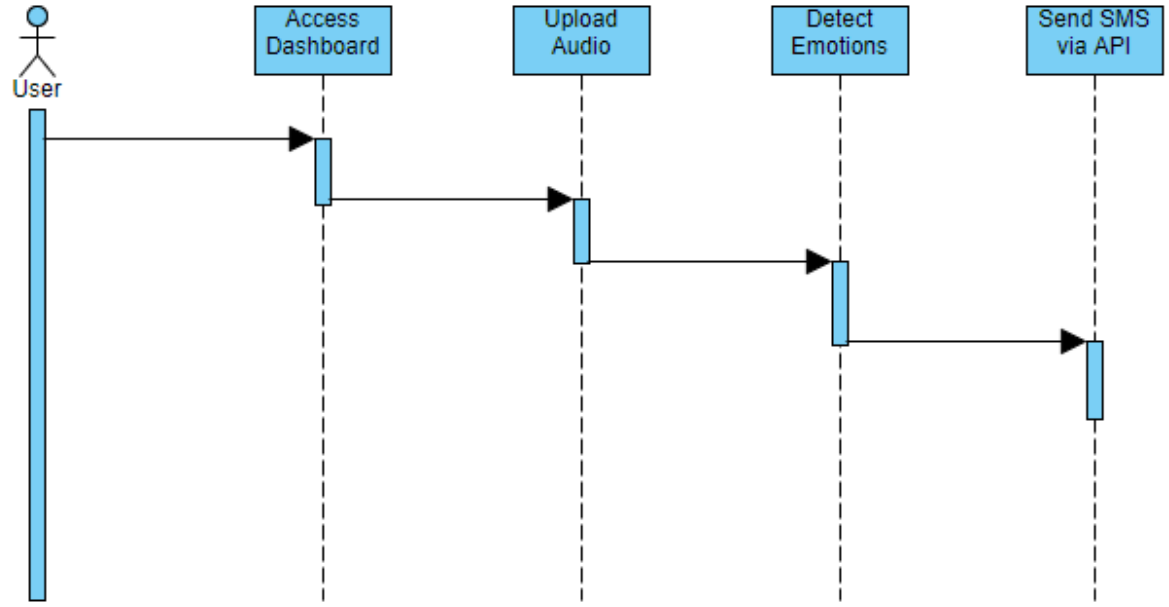
# Level 0 DFD



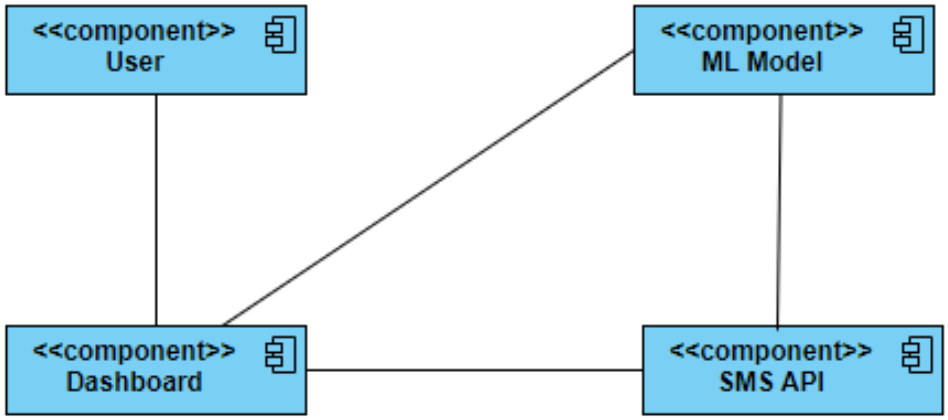

# Level 1 DFD



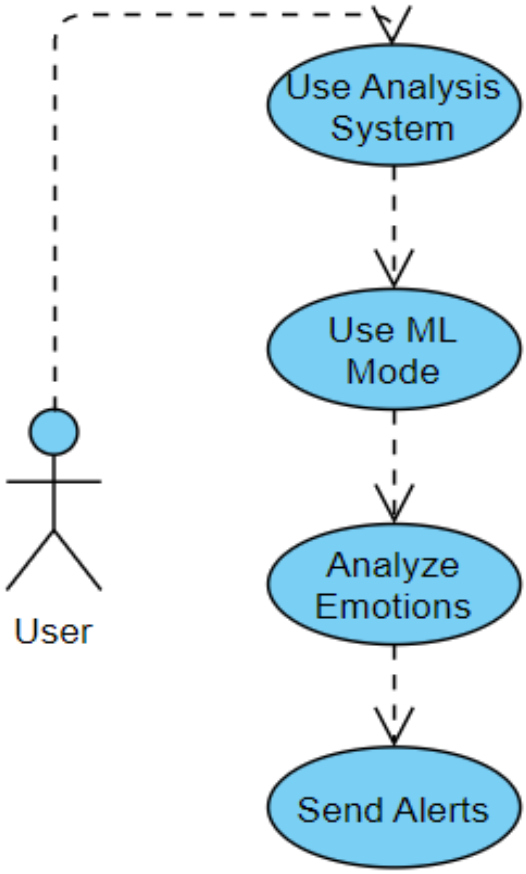
## 4.2 SEQUENCE DIAGRAM



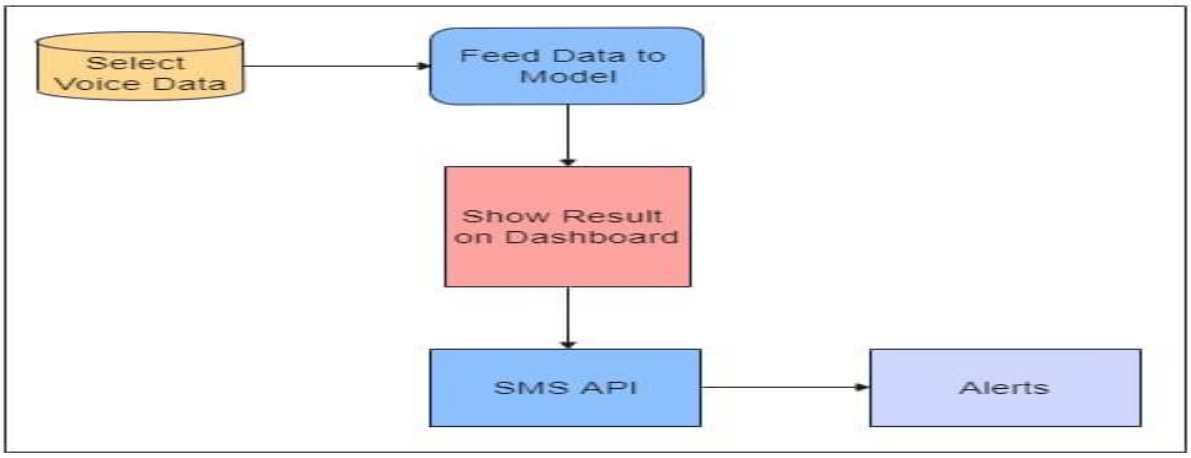

## 4.3 COMPONENT DIAGRAM

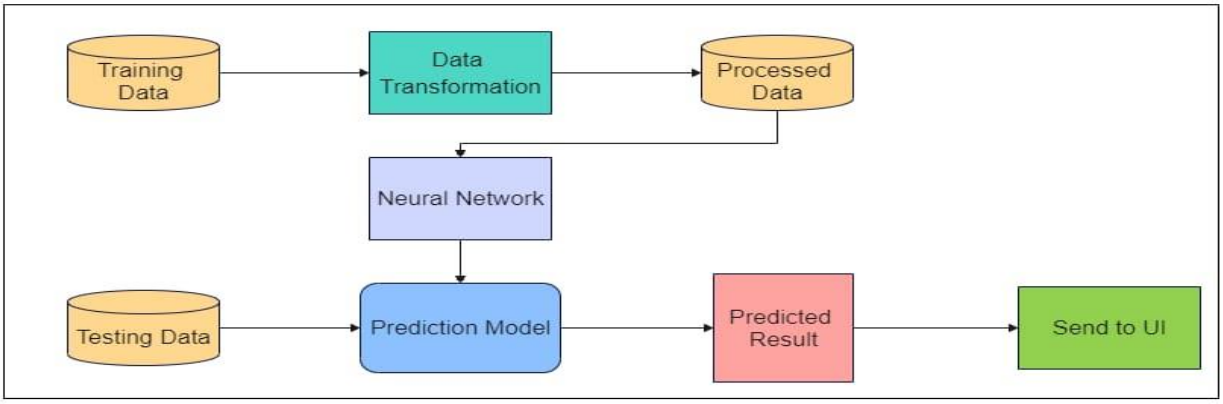


## 4.4 USECASE DIAGRAM



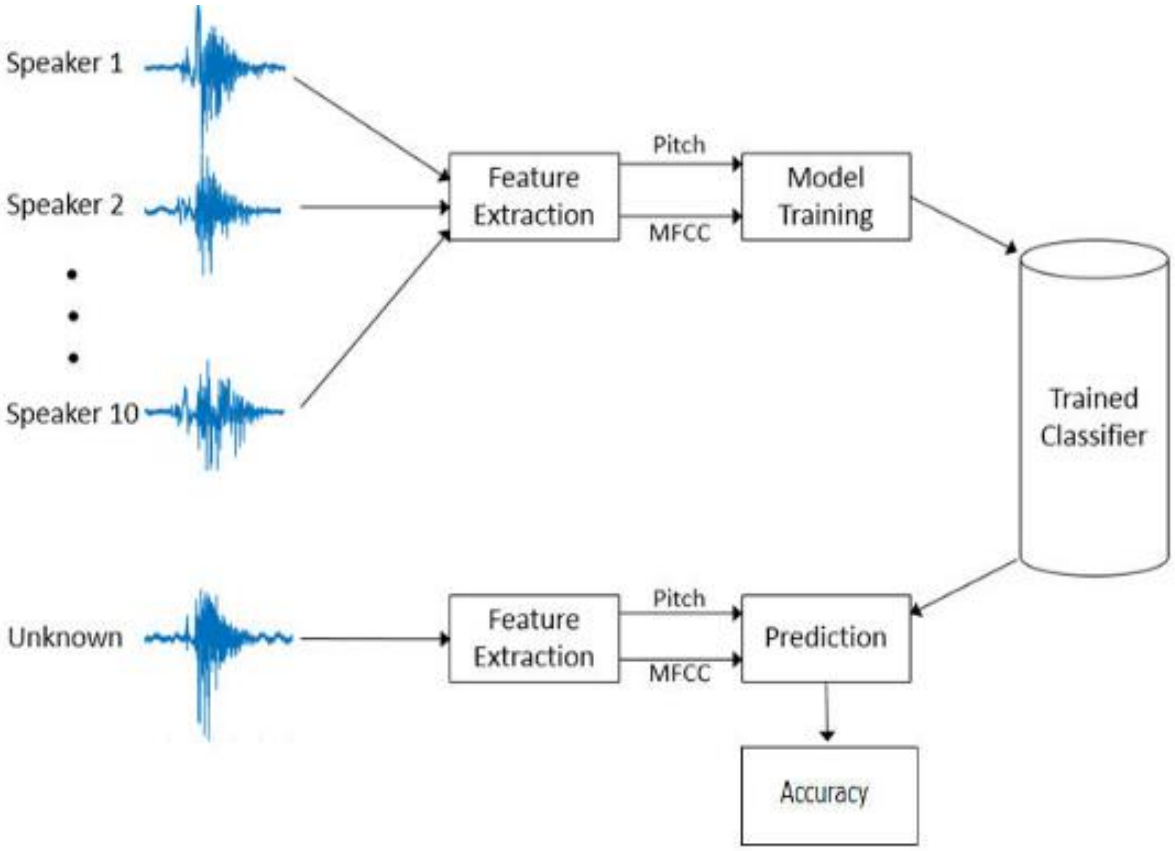
# 4.5 ARCHITECTURE DIAGRAM



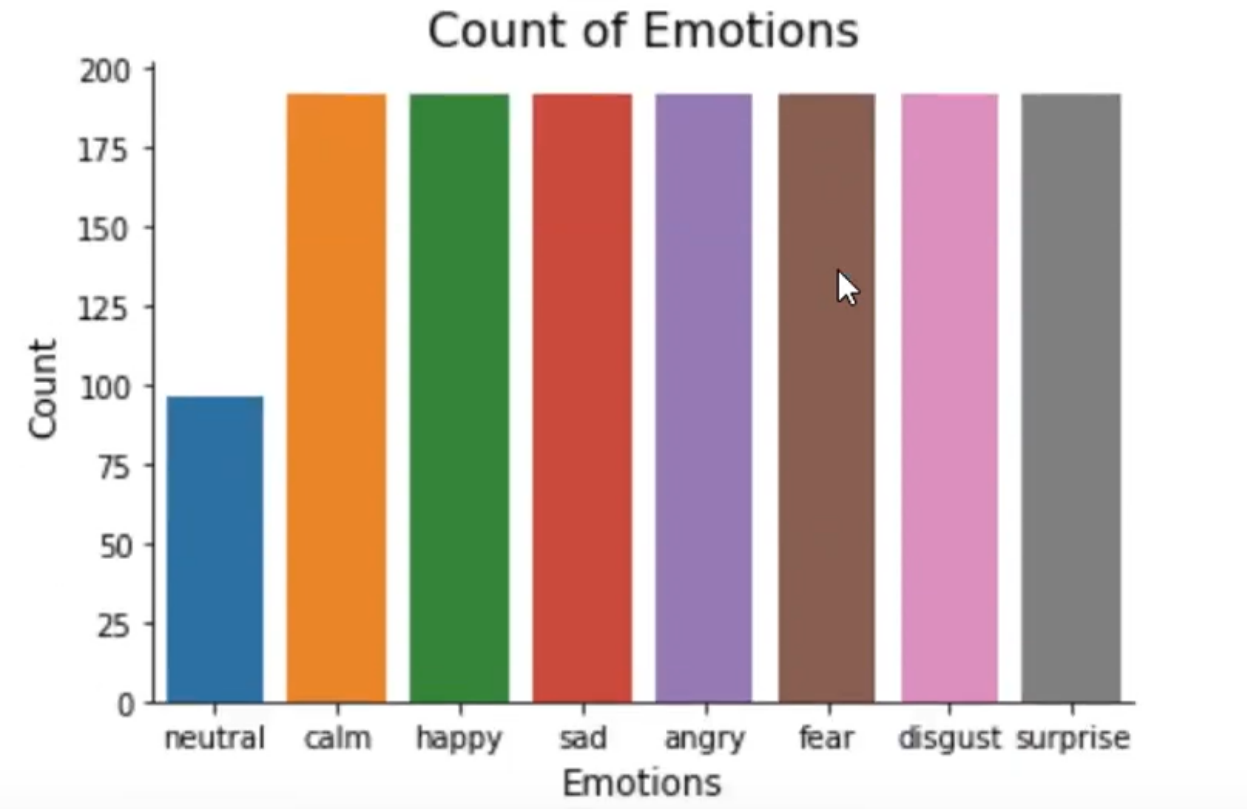



## 5. DATASET

## FEATURE EXTRACTION PROCESS

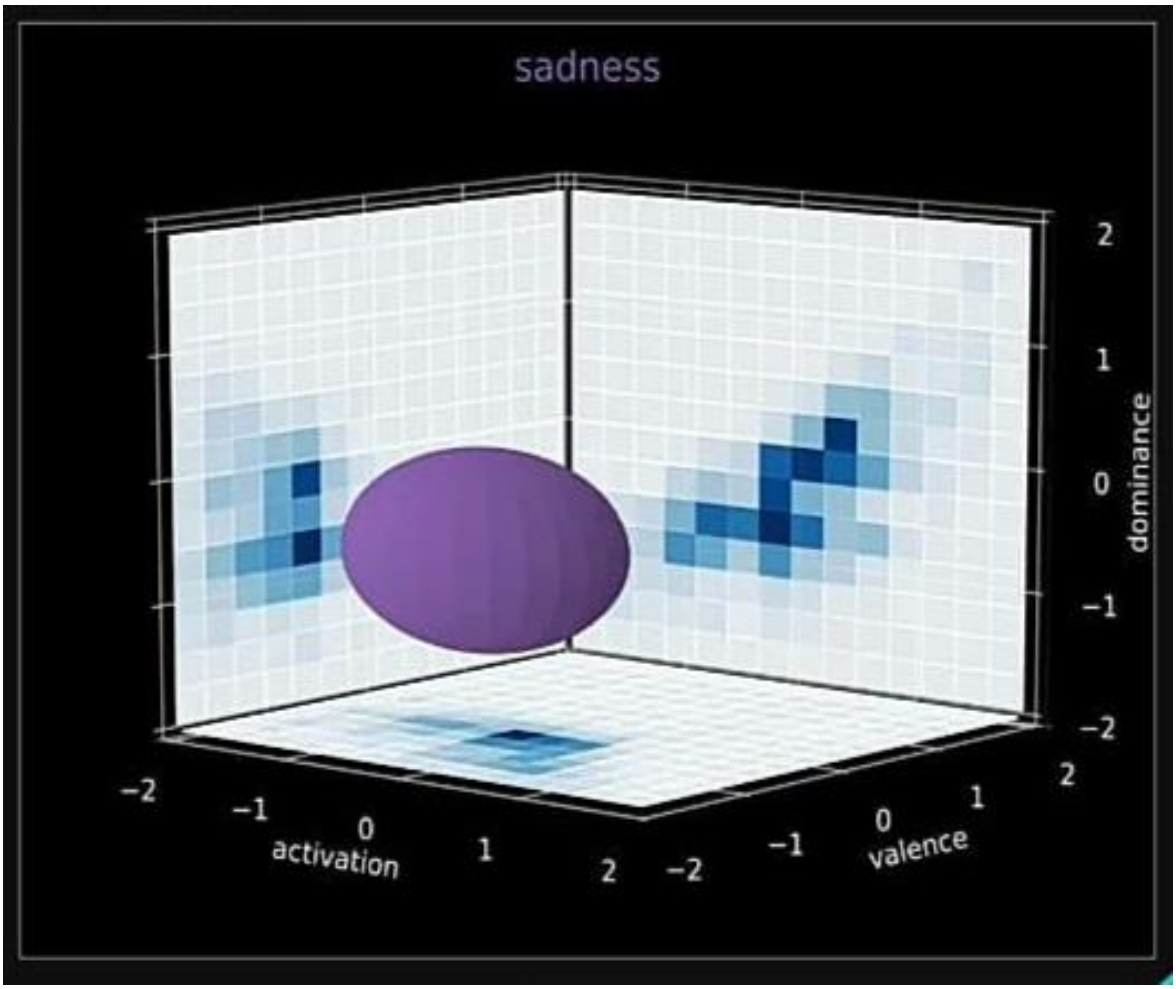


## 5.1 OBSERVATION FOR DATASET

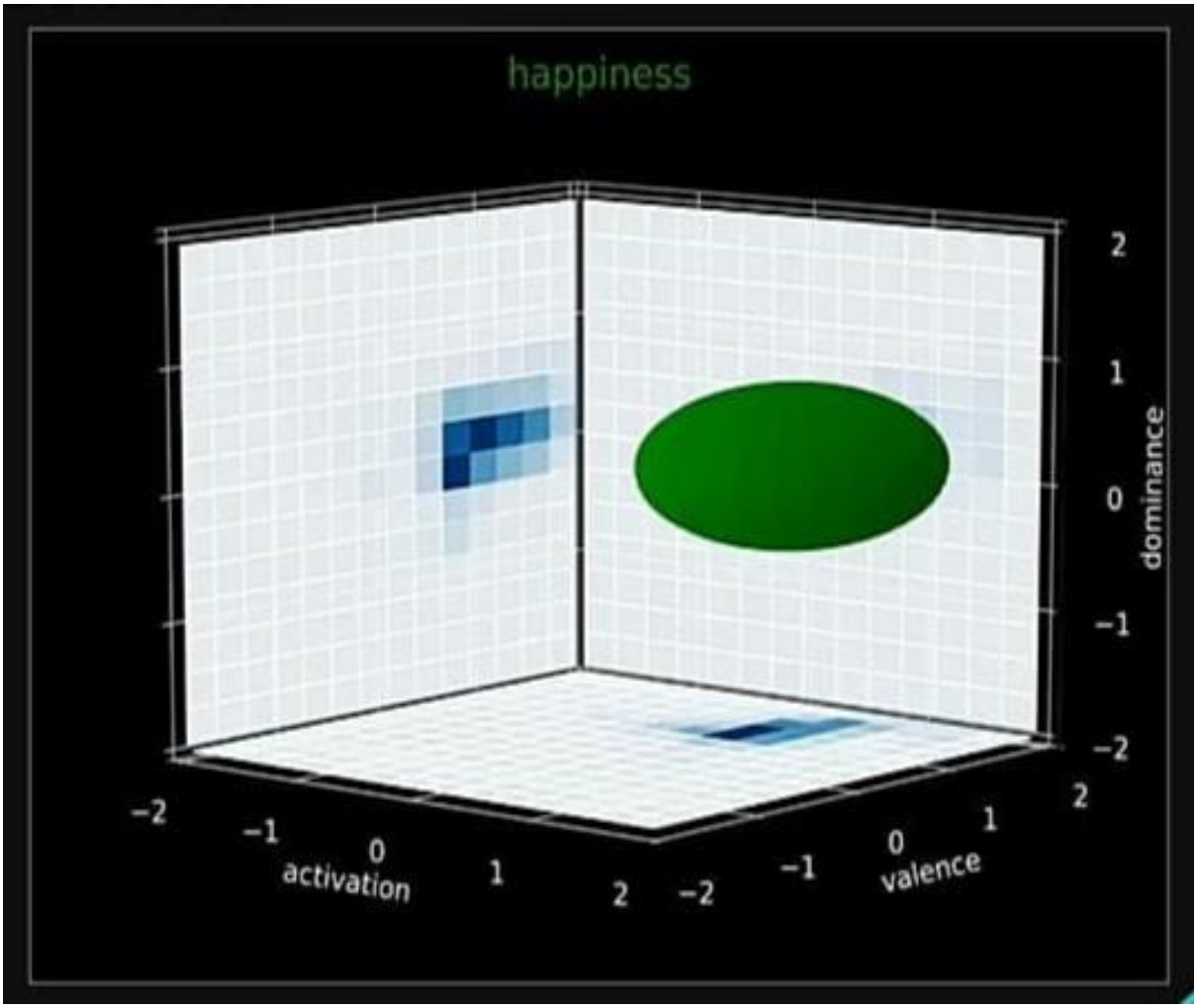


## 5.2 EMOTION IDENTIFICATION

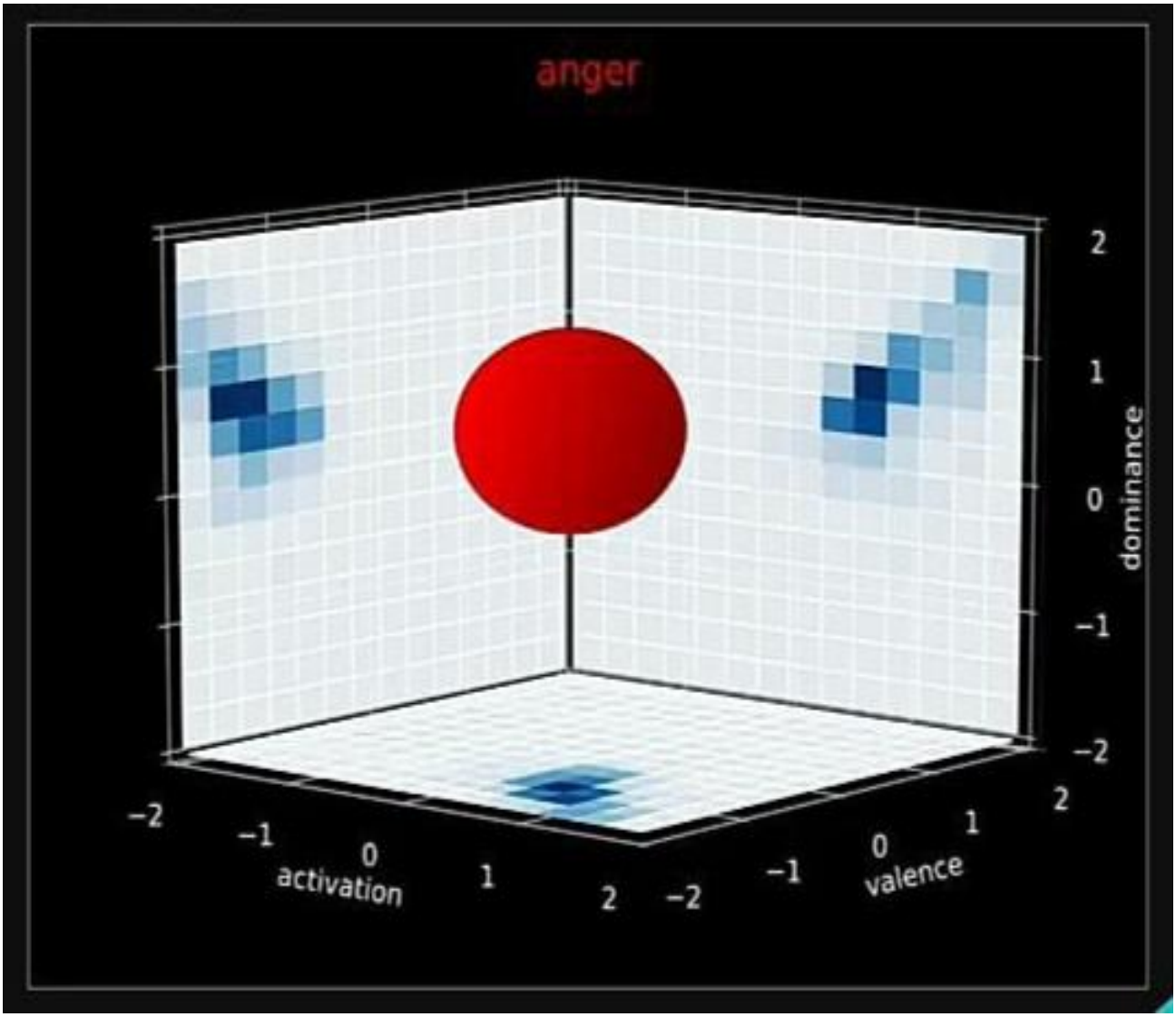
## FOR SADNESS



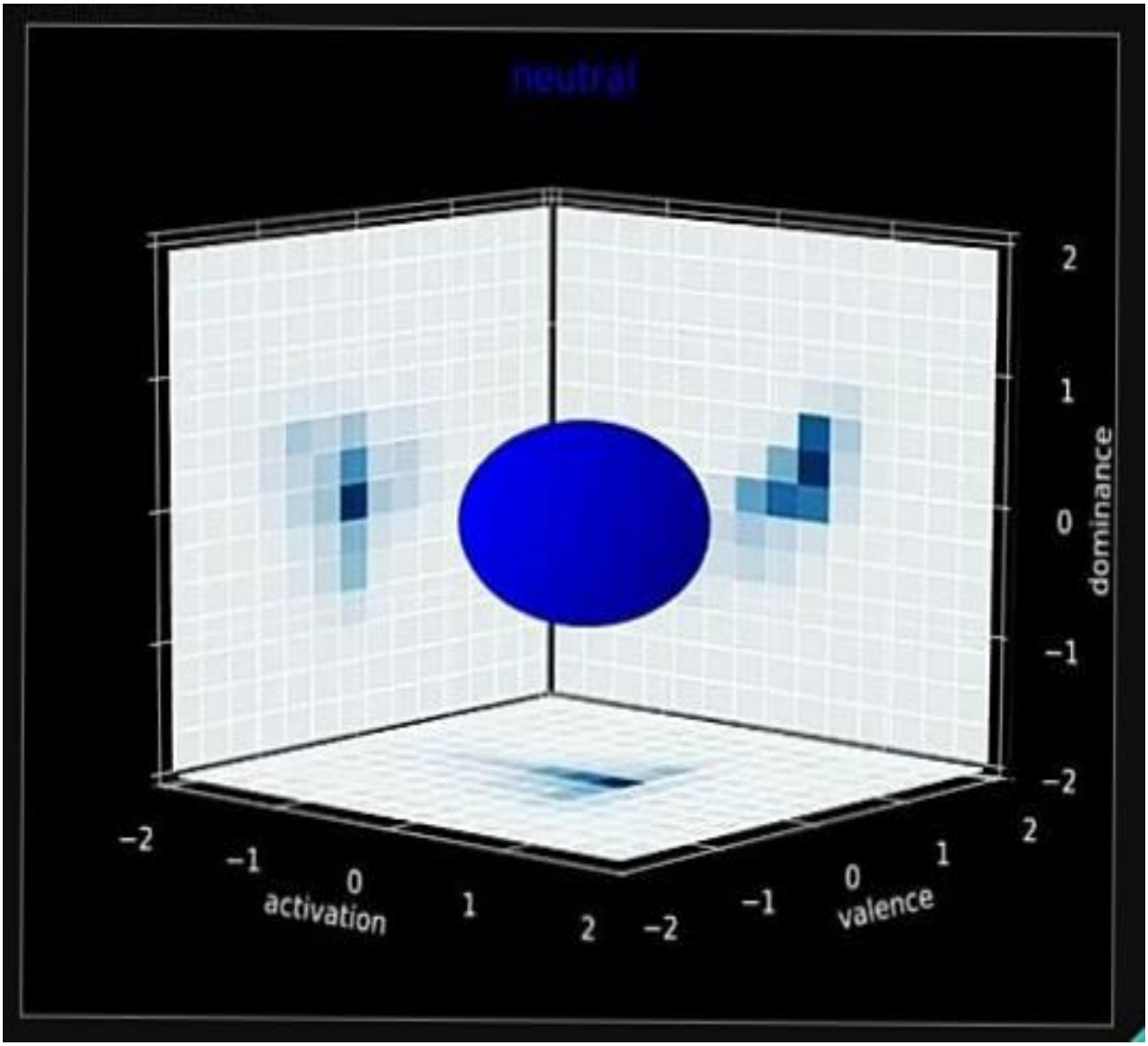
## FOR HAPPINESS



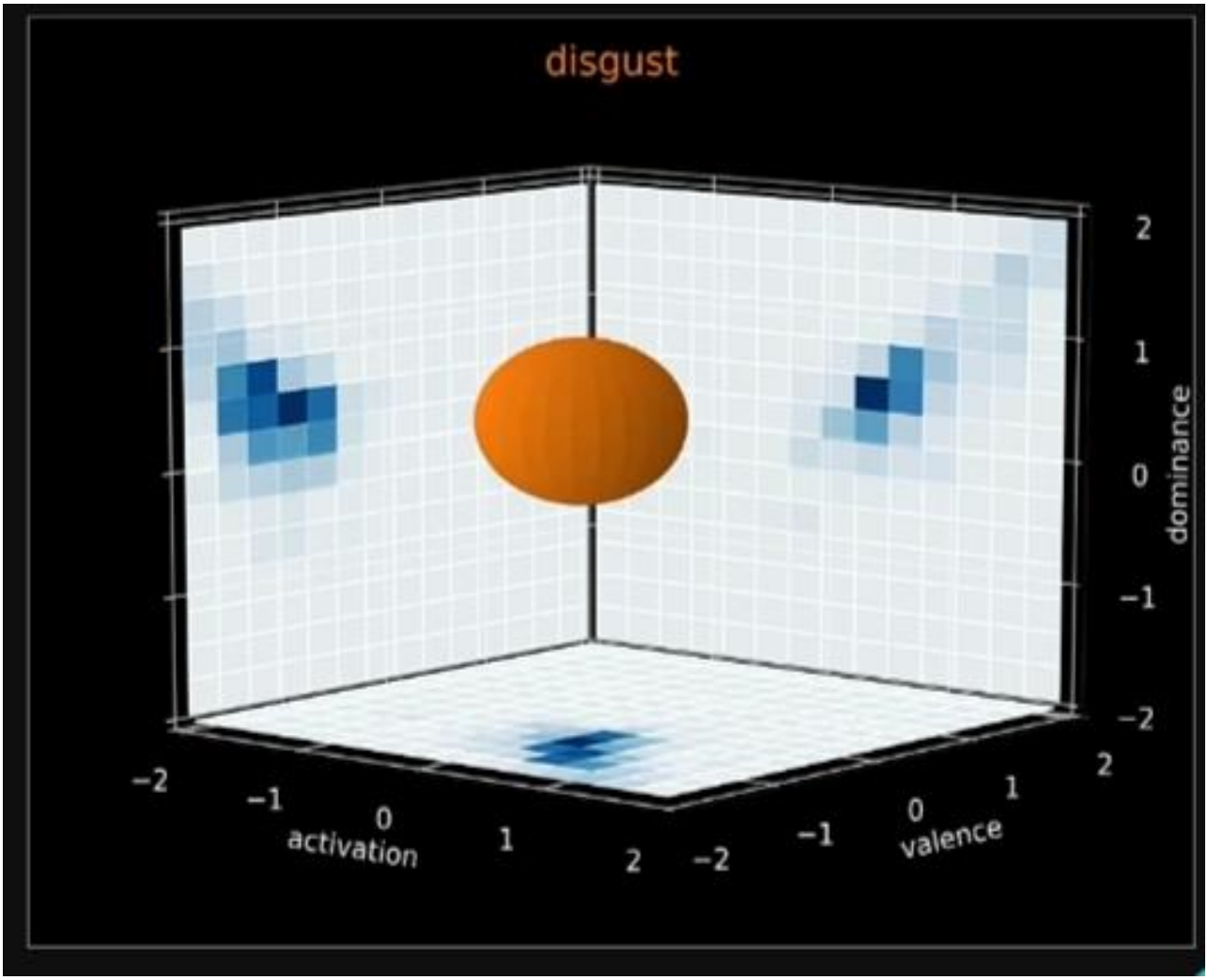
## FOR ANGER



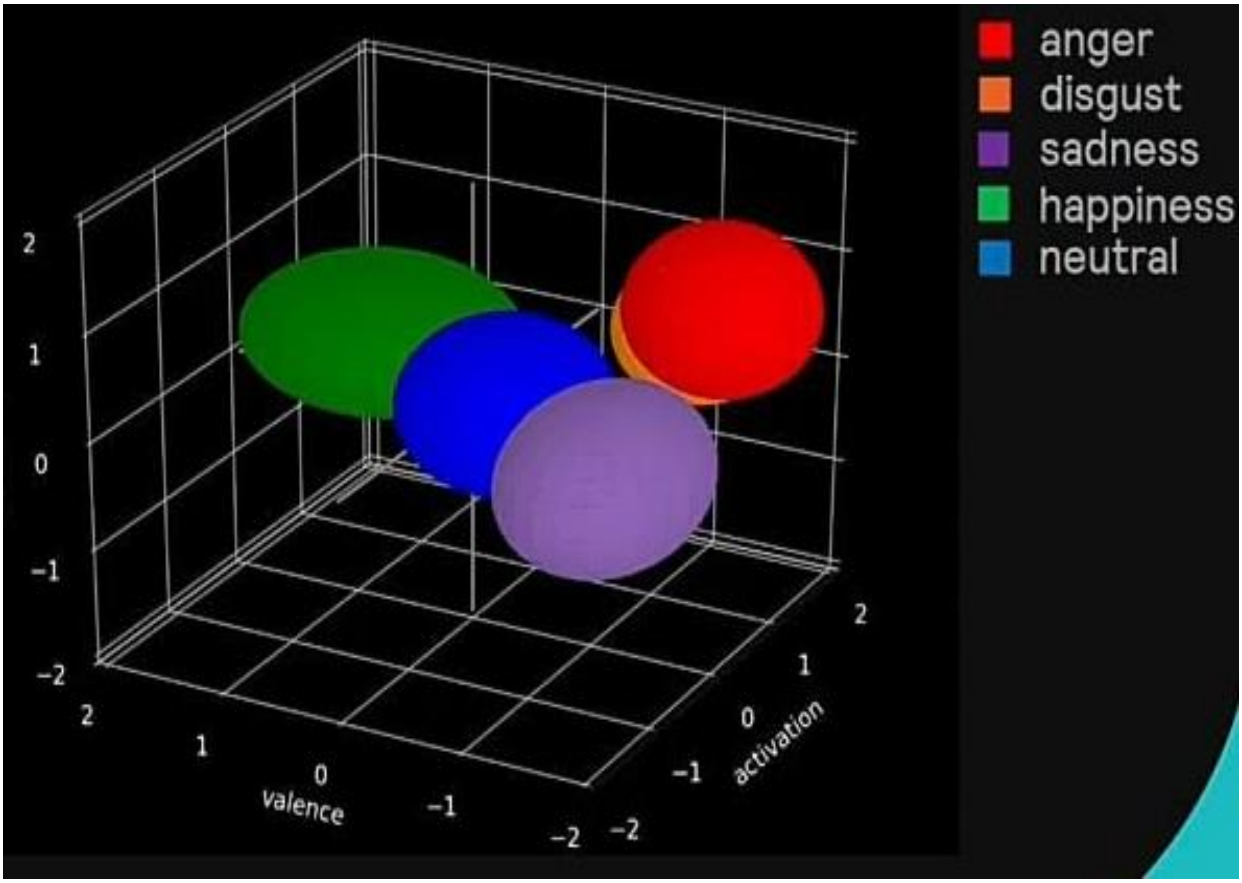
## FOR NEUTRAL



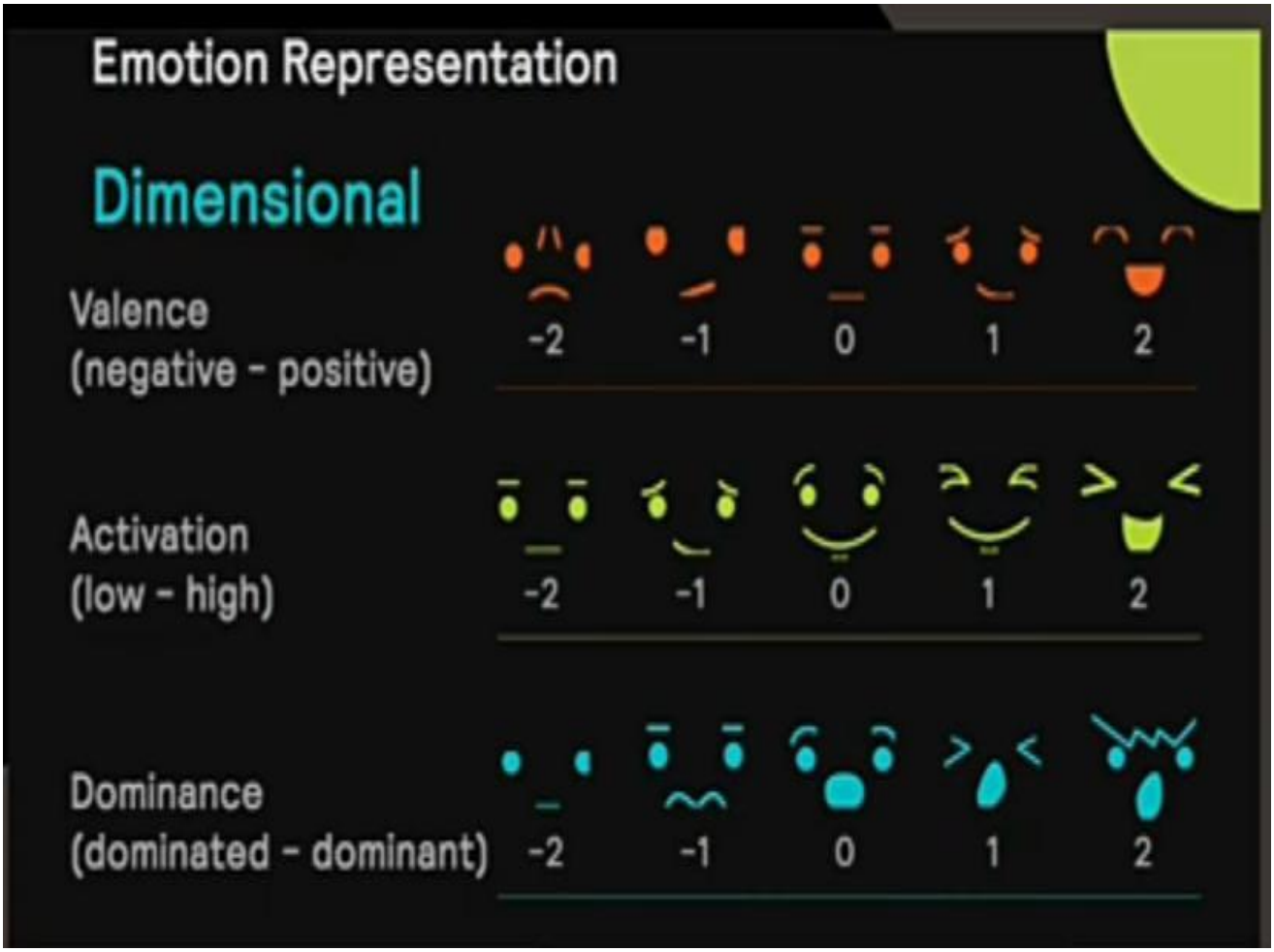
## FOR DISGUST



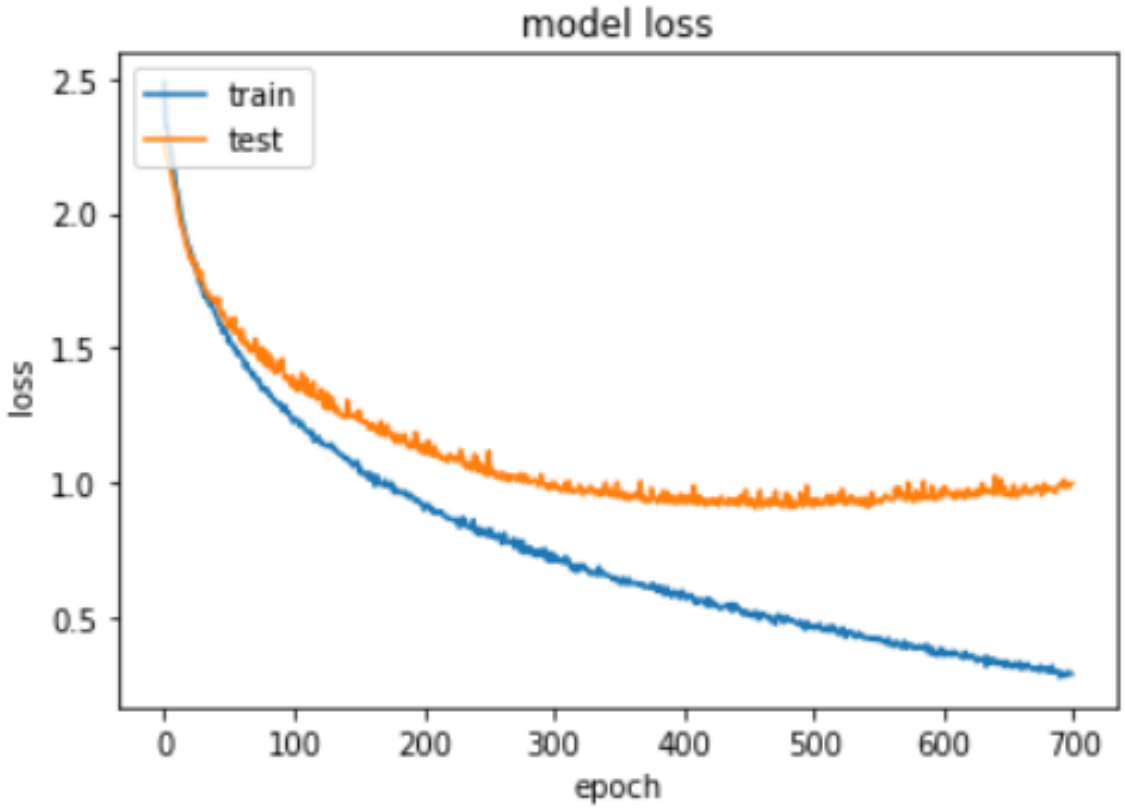
## 5.3 OVER ALL EMOTIONS IN ONE GRAPH



## 5.4 PITCH RANGES



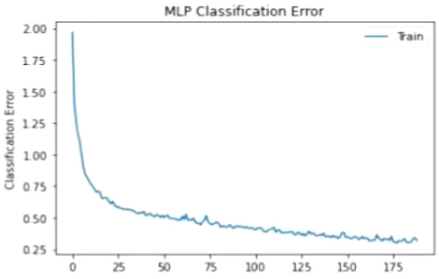
## 5.5 TRAINING AND TESTING GRAPH



# 5.6 ACCURACY FOR MLP CLASSFIER

**MLP Classifier Results**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| calm | 0.85 | 0.88 | 0.87 | 121 |
| happy | 0.67 | 0.76 | 0.71 | 34 |
| sad | 0.94 | 0.68 | 0.79 | 114 |
| angry | 0.73 | 0.88 | 0.80 | 120 |
| fearful | 0.83 | 0.77 | 0.80 | 119 |
| surprised | 0.82 | 0.85 | 0.84 | 123 |
| accuracy | | | 0.81 | 631 |
| macro avg | 0.81 | 0.81 | 0.80 | 631 |
| weighted avg | 0.82 | 0.81 | 0.81 | 631 |




# ACCURACY FOR XGB BOOSTER

**XGB Classifier Results**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| calm | 0.86 | 0.79 | 0.82 | 121 |
| happy | 0.54 | 0.62 | 0.58 | 34 |
| sad | 0.80 | 0.75 | 0.77 | 114 |
| angry | 0.74 | 0.74 | 0.74 | 120 |
| fearful | 0.79 | 0.77 | 0.78 | 119 |
| surprised | 0.76 | 0.85 | 0.80 | 123 |
| accuracy | | | 0.77 | 631 |
| macro avg | 0.75 | 0.75 | 0.75 | 631 |
| weighted avg | 0.78 | 0.77 | 0.77 | 631 |

# 6. CODING

# 6.1 PYTHON CODE

## Python Code for Main App ( Controller )

```
from django.contrib.auth import authenticate, login
from django.shortcuts import render, redirect
from pathlib import Path

import librosa
import soundfile
import os, glob, pickle
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score

import requests as r
import json


BASE_DIR = Path(__file__).resolve().parent.parent

def index(request):
    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

    if request.method == "POST":
        my_uploaded_file = request.FILES['my_uploaded_file'].read()
```

```
        name = "test"
        number = "1"
        file_name = "{}{}.wav".format(os.path.join(BASE_DIR,
'myapp/static/'),name+"_"+number)
        print(file_name)
        with open(file_name,'wb') as f:
            f.write(my_uploaded_file)

        modelname = 'trained_model.sav'

        path = os.path.join(BASE_DIR, 'myapp/static/')
```

```
        print(path)

        loaded_model = pickle.load(open(modelname, 'rb')) # loading the model
file from the storage

        feature=extract_feature(file_name, mfcc=True, chroma=True, mel=True)

        feature=feature.reshape(1,-1)

        prediction=loaded_model.predict(feature)

        print(prediction)

        service_plan_id = "ec9eae3c15704184b8aff9b7ba21c742"
        access_token    = "b0ca53e01801441da0d4a54def053b48"

        from_ = "447520651530"
        to_   = "918439321860"

        message = "Emotion Detected : "+str(prediction[0])

        headers = {
            "Authorization":f"Bearer {access_token}",
            "Content-Type":"application/json"
        }

        payload = {
            "from":from_,
            "to":[to_],
            "body":message
        }
        response = r.post(
            f'https://sms.api.sinch.com/xms/v1/{service_plan_id}/batches',
            headers=headers,
            data=json.dumps(payload)
        ).json()

        print(response)
```

```
        return render(request, "index.html",context = {"mymessage":"Emotion
Detected : "+str(prediction[0]),"Flag":"True"})

    return render(request, "index.html")
```

# Project Settings Code

```
import os
from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent
```

```
# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-
e#ov2dd3i228hlk82p!(1dl4)wur!1e!v(+@8_de&w!(tkl2mv'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'myapp',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

```
ROOT_URLCONF = 'mysite.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR,'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'mysite.wsgi.application'


# Database
# https://docs.djangoproject.com/en/3.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}


# Password validation
# https://docs.djangoproject.com/en/3.2/ref/settings/#auth-password-
validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator'
,
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]
```

```
# Internationalization
# https://docs.djangoproject.com/en/3.2/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.2/howto/static-files/

STATICFILES_DIRS = [
   os.path.join(BASE_DIR, 'myapp/static/')
]
STATIC_URL = '/static/'

# Default primary key field type
# https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

## Urls Code for Main App

```
from django.contrib import admin
from django.urls import include, path

urlpatterns = [
    path('', include('myapp.urls')),
    path('admin/', admin.site.urls),
]
```

## HTML Index Page Code

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.
css" rel="stylesheet" integrity="sha384-
KyZXEAg3QhqLMpG8r+8fhAXLRk2vvoC2f3B09zVXn8CA5QIVfZOJ3BCsw2P0p/We"
crossorigin="anonymous">
```

```
    <script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>

    <title>Home</title>
  {% load static %}
  </head>

  <nav class="navbar navbar-dark bg-dark justify-content-center">
    <div class="container">
      <a class="navbar-brand" href="/">
        Voice Sentiment Analysis
      </a>
    </div>
      </nav>

  {% load static %}
  <body background="{% static 'myapp/back2.jpg' %}" style="background-
size: cover;background-repeat: no-repeat;background-attachment: fixed;">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/js/bootstrap.bundle
.min.js" integrity="sha384-
U1DAWAznBHeqEIlVSCgzq+c9gqGAJn5c/t99JyeKa9xxaYpSvHU5awsuZVVFIhvj"
crossorigin="anonymous"></script>
    <br><br>
    <section class="vh-100 gradient-custom">
      <div class="container py-4 h-90">
        <div class="row d-flex justify-content-center align-items-center
h-100">
          <div class="col-12 col-md-8 col-lg-6 col-xl-5">
            <div class="card bg-dark text-white" style="border-radius:
1rem;">
              <div class="card-body p-3 text-center">

                <div class="mb-md-5 mt-md-4 pb-5">

                  <h2 class="fw-bold mb-1 text-uppercase">Upload File</h2>
                  <p class="text-white-50 mb-5">Please upload voice data
for sentiment analysis.</p>

                  <form name="voiceform" method="POST"
enctype="multipart/form-data">
                    {% csrf_token %}

                  <div class="form-outline form-white mb-4">
                    <input type="file" id="my_uploaded_file"
name="my_uploaded_file" accept="audio/*">
                  </div>

                  <br>

                  <button class="btn btn-outline-light btn-lg px-5"
name="login" type="submit">Analyse</button>
                   <br><br>
                  </form>
                </div>
              </div>
            </div>
```

```
          </div>
        </div>
      </div>
    </section>


  </body>

<script type="text/javascript">

  {% if Flag %}
      swal({
        text: "{{ mymessage }}",
      })
      .then((homemessage) => {
        if(homemessage)
        {
          {% if Navigate %}
            window.location.href = "/";
        {% endif %}
        }
      });
  {% else %}
      None
  {% endif %}


</script>

</html>
```

## 7. TESTING

## TESTING PHASE

Testing refers to test the software so it is also called software testing. **Software testing** is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test.[1] Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test-

- meets the requirements that guided its design and development,
- responds correctly to all kinds of inputs,
- performs its functions within an acceptable time
- is sufficiently usable
- can be installed and run in its intended environments, and
- achieves the general result its stakeholders desire.

## 7.1 Unit Testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

Parameterized unit tests (PUTs) are tests that take parameters. Unlike traditional unit tests, which are usually closed methods, PUTs take any set of parameters. PUTs have been supported by Testing, JUnit and various .NET test frameworks.

Suitable parameters for the unit tests may be supplied manually or in some cases are automatically generated by the test framework. Testing tools like Quick Check exist to generate test inputs for PUTs.

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

In our tool we tested each and every unit (or module) and it was successfully executed.

We perform sentiment analysis process by our tool and its accuracy is around 75% and which is up to the expectation.

## 7.2 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. Some different types of integration testing are big-bang, mixed (sandwich), risky- hardest, top-down, and bottom-up. Other Integration Patterns are: collaboration integration, backbone integration, layer integration, client-server integration, distributed services integration and high-frequency integration. In the big-bang approach, most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing. This method is very effective for saving time in the integration testing process. However, if the test cases and their results are not recorded properly, the entire integration process will be more complicated and may prevent the testing team from achieving the goal of integration testing.

In our tool we tested all components by merge together and it passed all our criteria.

## 7.3 System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system.

The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behaviour and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification.

After performing system testing with our tool we find that it follow all requirements and working as per requirement, it takes the input as audio to process data and give result as detected emotion and send alert to the mobile phone via SMS

API.

## 7.4 Functional Testing

Functional testing is a quality assurance (QA) process and a type of black-box testing that bases its test cases on the specifications of the software component under test.

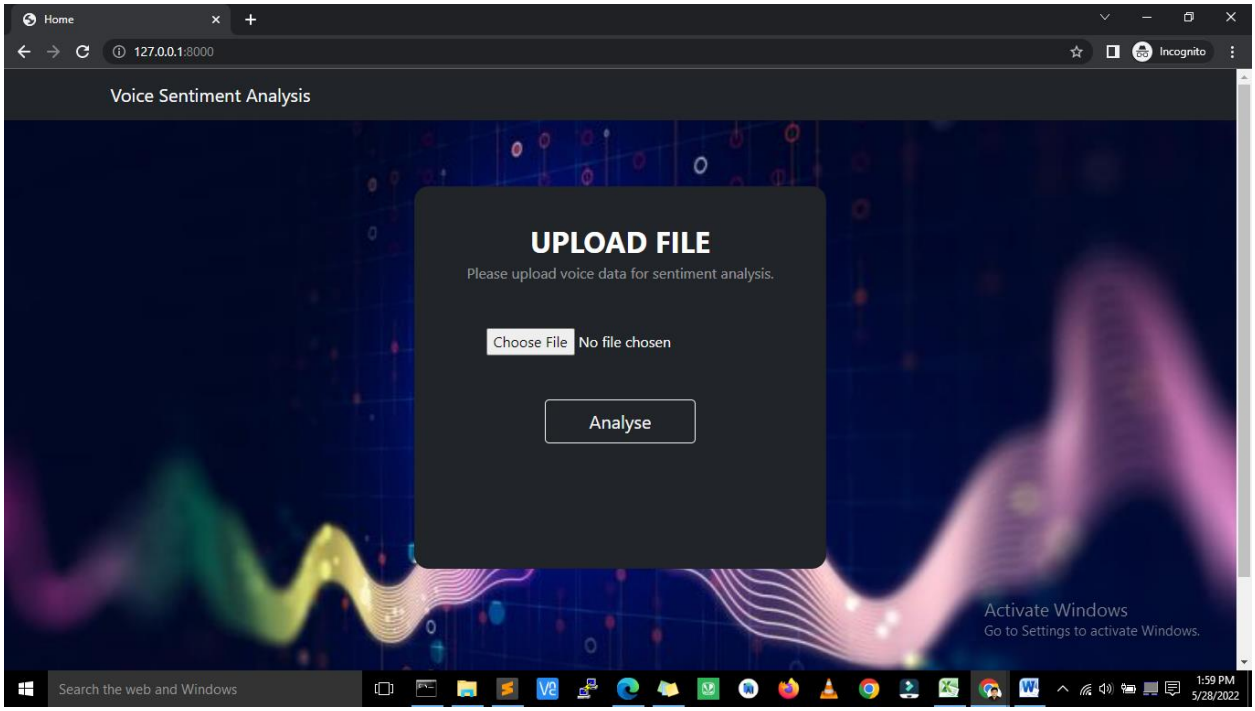
Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (unlike white-box testing). Functional testing is conducted to evaluate the compliance of a system or component with specified functional requirements. Functional testing usually describes what the system does.

All functionality voice sentiment analysis tool is working properly.

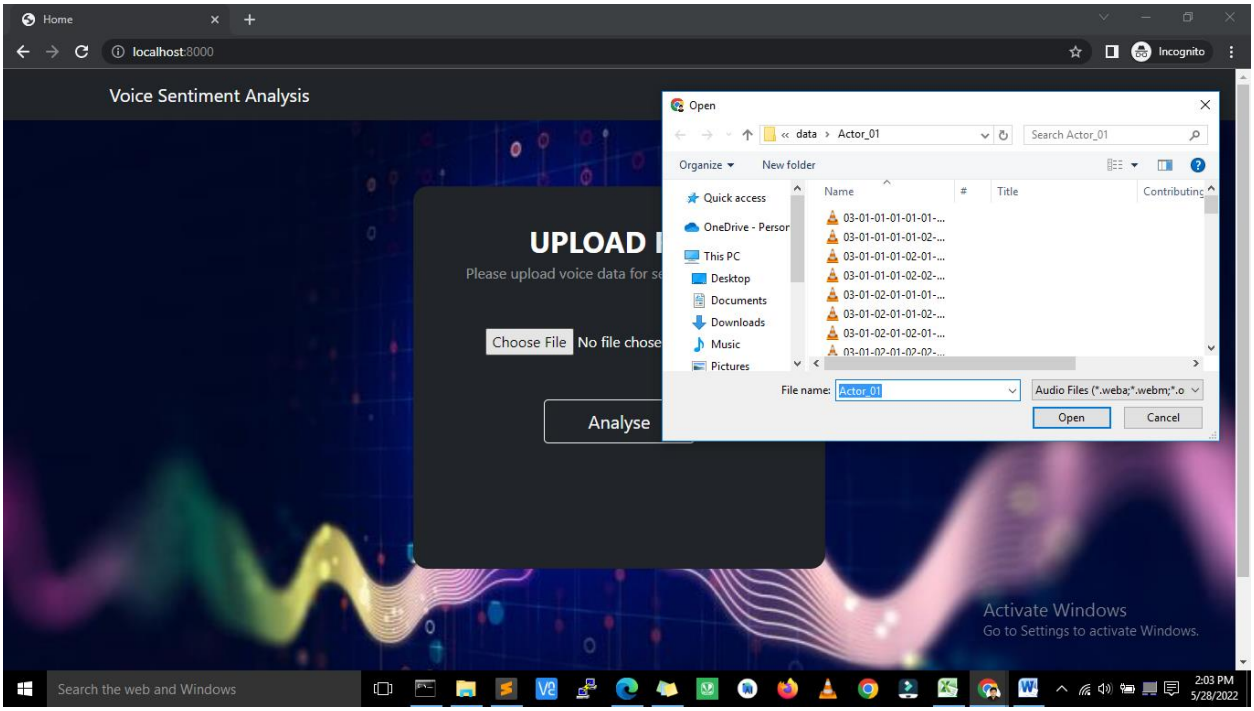
It took input (as audio) properly and gives the output as detected emotion from the audio.

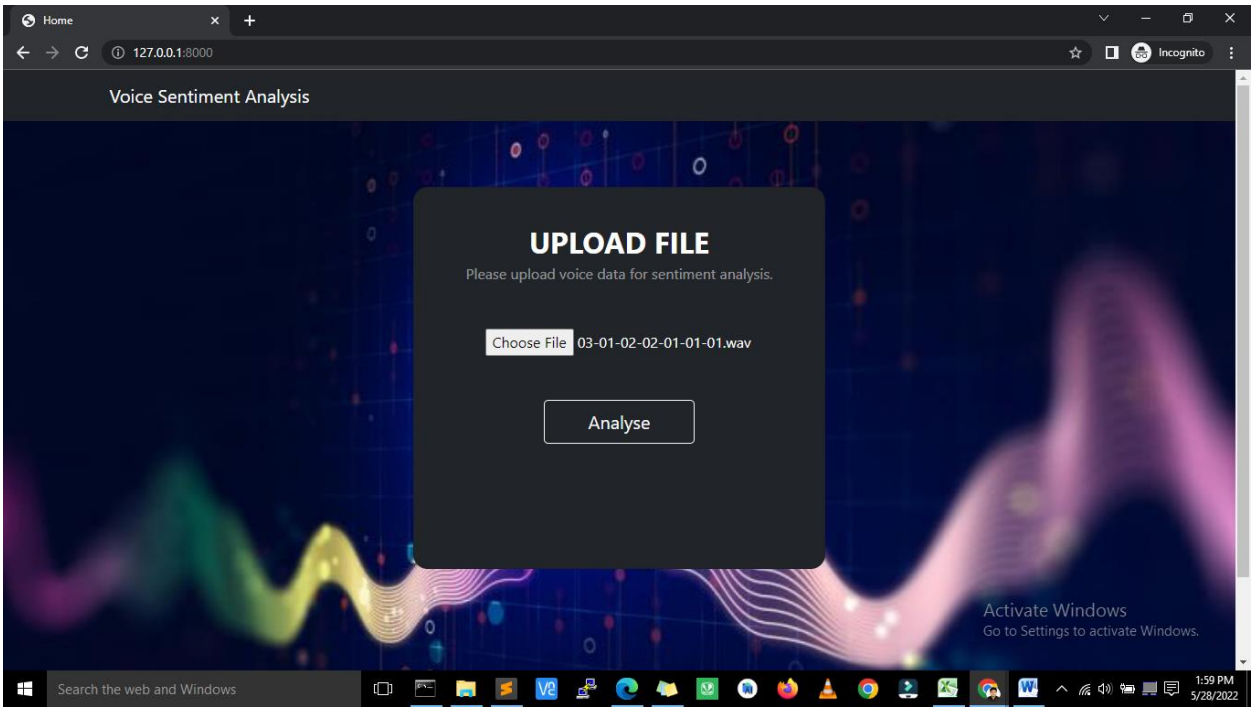
# 8. SNAPSHOTS

# 1. INDEX PAGE

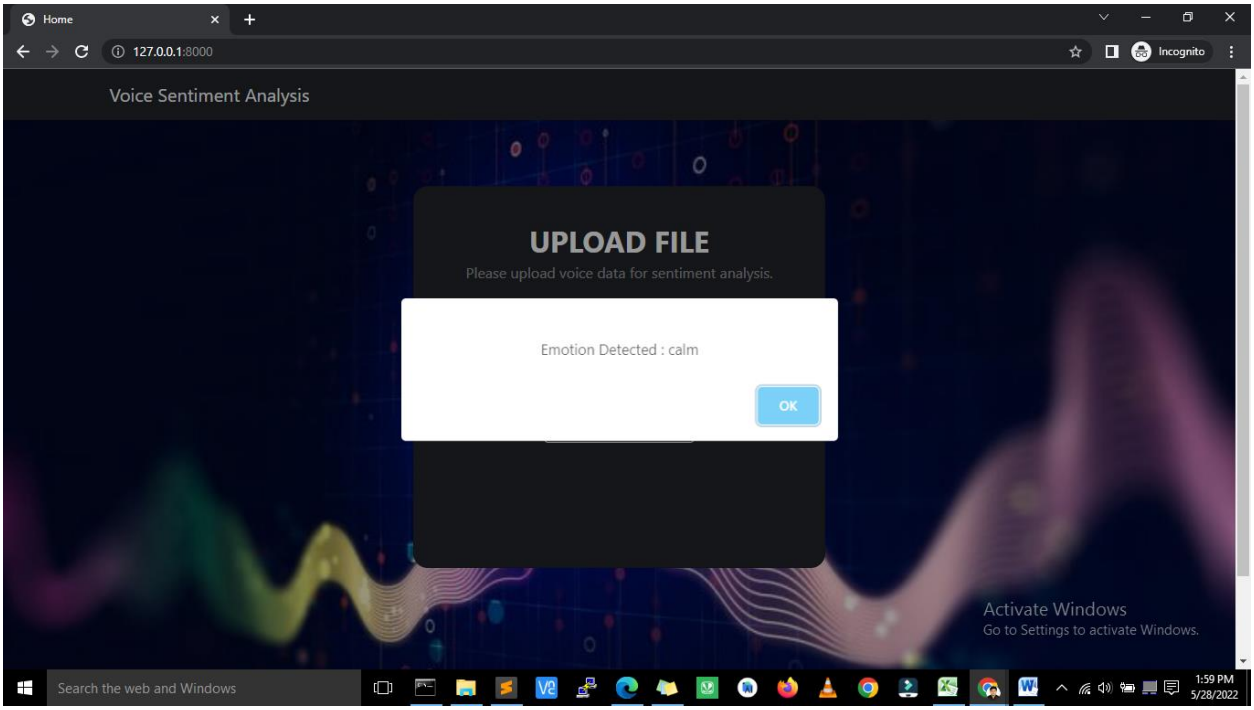



# 2.Audio Selection


Home
localhost:8000
Incognito
Voice Sentiment Analysis
UPLOAD
Please upload voice data for
Choose File
No file chose
Analyse
Open
data
Actor_01
Search Actor_01
Organize
New folder
Quick access
OneDrive - Persor
This PC
Desktop
Documents
Downloads
Music
Pictures
Name
#
Title
Contributing
03-01-01-01-01-01-...
03-01-01-01-01-02-...
03-01-01-01-02-01-...
03-01-01-01-02-02-...
03-01-02-01-01-01-...
03-01-02-01-01-02-...
03-01-02-01-02-01-...
File name:
Actor_01
Audio Files (*.weba;*.webm;*.o
Open
Cancel
Activate Windows
Go to Settings to activate Windows.
Search the web and Windows
2:03 PM
5/28/2022


Home
127.0.0.1:8000
Incognito
Voice Sentiment Analysis
UPLOAD FILE
Please upload voice data for sentiment analysis.
Choose File
03-01-02-02-01-01-01.wav
Analyse
Activate Windows
Go to Settings to activate Windows.
Search the web and Windows
1:59 PM
5/28/2022

# 3.Output




# 4.SMS Alerts

## 9. Future Enhancement

In Future we will add some more features to our Voice sentiment analysis tool like-

1. In future we can add WhatsApp alerts so that users can get alerts on WhatsApp also.

2. In future we can add the features which can continuously listen to the voice from the surrounding and detect emotions continuously.

3. In future we will give option where user can pass the audio link directly for those audios which are stored in online medium.

4. In future we will make this application live so that it can be accessible from anywhere.

## REFERENCES

## BOOKS:


- Ian Somerville ‘Software engineering’
- Rajeev mall ‘Software engineering’


## ONLINE REFERENCE:


1. www.google.co.in
2. https://stackoverflow.com/
3. https://docs.djangoproject.com/en/3.2/
4. https://www.w3schools.com/