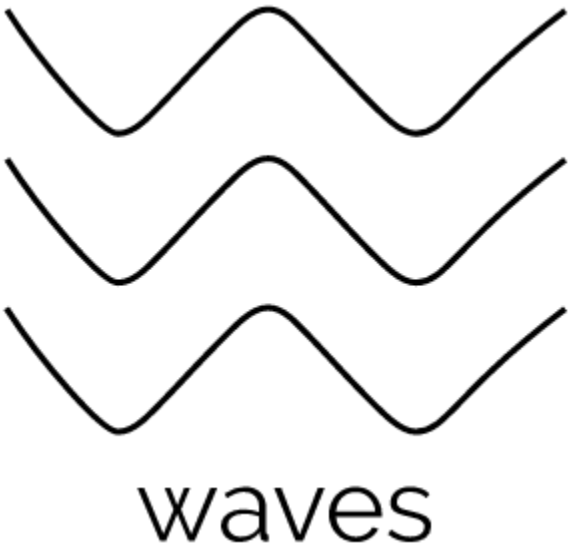
Project Report



A Web Application to Detect Emotion From Speech

Sajal Ganjewala - 1810110204
Sparsh Ailawadi - 1810110251
Savitha G - 1810110223

1. Introduction

1.1. Document Purpose

This document serves as a reference to the software development process as well as a detailed explanation of the entire application as developed under the name of Wave, a web-based portal developed with the intention of allowing users to identify emotions based on speech. This document will outline the software design and specification of our workflow task management system in addition to system architecture, system components, and software requirements. Moreover, this document contains specifics about the design of our software. It is a high-level explanation of how the project functions, and which components have been developed to help it do so. This document also serves to explain the technologies used to develop the website, as well as the user's interaction with various features.

1.2. Feasibility Report

Introduction

Emotions are fundamental for humans, impacting perception and everyday activities such as communication, learning, and decision-making. They are expressed through speech, facial expressions, gestures, and other non-verbal clues. Communication is essential for clearly expressing one's thoughts and ideas. Speech is the most favored and powerful means of communication among humans among all types of communication. Speech emotion detection is the study of vocal behavior as an indicator of emotion, with an emphasis on nonverbal elements of speech. Its core premise is that a person's voice contains a collection of objectively quantifiable characteristics that represent the affective state they are now conveying. The voice reflects underlying emotion through tone and pitch allowing us to analyze the emotion of the human speaking.

Proposed Solution

The idea is to develop an application that uses audio from either the one that has been uploaded or recorded live on the app to find and display the emotion being shown by the speaker.

2. Project Overview

2.1. Goals

The goal of the project is to develop an end-to-end web-based solution that enables users to identify and recognize emotions based on the speech being uploaded in an audio format. The application will serve as the foundation of a larger idea that leads to bringing forth discussions about Alexithymia, a subclinical disease that is characterized by a lack of the ability to recognize emotions.

2.2. Scope

The scope of this particular application is to establish the basic foundation of a cross intersection of information and data-based web application running a machine learning

model that effectively recognizes emotions based on the audio input. Further, the idea is also to allow for the dissemination of information about Alexithymia and related information.

2.3. Objectives

The following are the primary underlying objectives of the project being undertaken,

- Identify a suitable dataset to allow for testing and training of the model in order to identify emotions.
- Clean and preprocess the dataset in order to make it ready for the model. Further conduct a basic exploratory data analysis in order to extract information from the dataset.
- Develop and deploy a machine learning model that identifies the information from the audio being uploaded.
- Develop a suitable user interface allowing ease of use of the application.

2.4. Deliverables

The following are the key deliverables for the project being undertaken

- An end-to-end user-friendly web-based application that allows the user to identify and display the emotion being shown in the audio file provided by them.
- User experience-centric design for the web application that displays information as deemed relevant.

2.5. Success Criteria

The project will be considered a success if a user-friendly interface allowing the user to identify the emotion being conveyed via the audio is developed. The model must be running on a web server with further analysis benign done upon the accuracy and other relevant metrics.

3. Project Approach

3.1. Process Model

The project was broken down into 3 modules and the team members worked on their respective independent modules of the project in alignment with an agile process model. The team engaged in weekly coding sprints where we worked on our respective tasks of the day and had a meeting after the sprint to discuss goals, progress, updates, and backlogs. The regular scrum meetings and coding sprints and through this iterative development strategy, the requirements and solutions evolved through collaboration between members working on different aspects of the project. This process allowed us to develop the deliverables of Wave keeping in mind the backend and frontend, working in unison from Day 1.

3.2. Project Team

The Wave development team comprises:

- Sajal Ganjewala - Frontend Developer & Product Owner
- Sparsh Ailawadi - Backend Developer & Quality Assurance
- Savitha G - Backend Developer & Testing

3.3. Roles and responsibilities

In addition to our individual strengths in development, each of us took up roles that complemented our characters in the agile framework.

Sajal Ganjewala - Product Owner

Sajal was responsible for designing the overall architecture of the application and laying out tasks that had to be completed over the course of the project. He also took on the role of scrum master where his responsibilities included:

- Lead Sprint planning
- Organizing bi-weekly scrum meetings
- Assisting with product backlog
- Participating in upfront planning of the agile development cycles
- Conflict resolution and removing roadblocks

Sparsh Ailawadi - Frontend developer & Project Manager

Sparsh was majorly involved in the frontend development of the web application and integration between the frontend and backend. He also acted as the Project Manager at Wave where the duties included:

- Defining the project scope and gathering requirements
- Identifying activities, sequencing, and time estimates
- Identifying deliverables and milestones
- Serving as a single-point-of-contact between teams
- Communicates with users via multiple channels and gathers feedback

Savitha G - Backend Developer & Testing

Savitha took on the backend development of the project making sure that the model ran well over the webserver and the data transfer between the web services and the machine learning model was seamless. As a test engineer, her responsibilities also comprised:

- Identifying bugs and potential problems within the product

- Planning and performing different stages of testing
- Developing new tools and testing strategies

3.4. Evolution of the plan

After identifying the deliverables of Wave, we decided on an agile framework for development and divided our roles and responsibilities according to our strengths. We broadly broke down our coding phases into — Backend Development, Frontend Development, and Integration & Testing. We had biweekly coding sprints and scrum meetings for the development of Wave where we worked on our individual modules and discussed our progress. In these progress meetings, we often had criteria in one team member's module which reflected changes in another team member's functionality since they are all closely related. This method and communication helped us to stay on track and make changes accordingly, avoiding larger bugs and issues in later stages of development.

Post integration, we moved onto the last phase which is testing and quality assurance, and heavily focused our efforts on the same to provide the users with a smooth bug-free experience.

4. Communications

Team Wave majorly used messaging and emails in order to communicate with each other.
We held our bi-weekly scrum meetings on the Google Meet platform and would communicate our individual tasks for each other during the scheduled call.

5. Work Plan

5.1. Work breakdown structure

The development of Wave can be broken down into 3 phases that were treated as independent modules:

❖ **Model Design and Development**

This was a particularly extensive process that started with identifying a dataset for the purposes of developing the model. For the purposes of this project, we chose The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) which contains 7356 files. The database contains 24 professional actors (12 female, 12 male),

vocalizing two lexically-matched statements in a neutral North American accent. Speech includes calm, happy, sad, angry, fearful, surprise, and disgust expressions

After the dataset was finalized we performed some basic data analysis in order to gather more insights from the data. After this was done, key features were identified that will enable the development of the model. Once these key features were established, a suitable model - an MLP Classifier was used.

The dataset was divided into test and training data and the MLP Classifier was then used on the following set of features,

- MFCC(Mel Frequency Cepstral Coefficients): The special characteristic of MFCC is that it is taken on a Mel scale. It scales the frequency in order to match more closely what the human ear can hear
- Mel Spectrogram: Depicts amplitude mapped on a Mel scale
- Chroma: It is a 12-element feature vector indicating how much energy of each pitch class is present in the signal on a standard chromatic scale.

The Python implementation of the Librosa package was used in their extraction.

- **Backend Development**

  The backend development was relatively simple with most of the work being done in order to ensure

- **Frontend/UI development**
  - Figma Design & Theme Ideation
  - Login and Registration Page
  - Search Page (Home Page)
    - Running Search Display Cards
    - Search results Display Cards
  - Chat Page
    - Chat interface
  - Predictor Page
    - Predictor interface
- **Integration and testing**
  - Unit Testing on Individual Backend Modules
  - Linking together appropriate backend and frontend services
  - Integration Testing

5.2. Resources

We made a conscious effort to marshal our resources and use them diligently to meet Waves deliverables. The responsibility of allocation of resources was handled by the Project Manager who took into consideration the following:

- Time
  We managed our time effectively by planning early and regularly holding our 3 hour biweekly sprints, thus ensuring maximum productivity from all team members during those invested 6 hours every week.
- Human Resources
  We each applied our different skill sets as programmers, graphic designers, and database administrators to the project.
- Computer Resources
  We required a web server.
- Money
  All the tools and frameworks that were used are free of cost. They can be upgraded in the future through payments, if necessary.

5.3. Conflict resolution

Conflict resolution in Wave was well handled by the Project Manager. Since we are a small team of 3, the conflicts that we encountered mainly revolved around differences in views about frontend design ideas, deliverables, etc., and schedule changes. These were usually resolved through an open discussion or by taking a vote, accommodating all or majority of the team with a suitable compromise.

6. Technical Process Plans

6.1. Methods, Tools, and techniques

Method

The project followed an iterative software life cycle development model throughout in order to go through multiple successive iterations that allowed timely detection of bugs and the addition of new features over the course of development. The method further allowed the end product to be visualized as the development progressed given that the initial visualization of the end product was vague.

Tools

The development of the project leveraged multiple tools - both for project management as well as technology development. The following is the set of tools used,

Project Management,

- Google Sheets
- Google Meet
- Microsoft Teams
- Google Docs

Design,

- Figma
- Online Whiteboards

Version Control,

- Git/Github

Technology Stack,

- Flask
- JavaScript
- ReactJS
- HTML/CSS
- Tailwind
- Python

Techniques

The project followed a lean model with an agile approach. Reusable components were used throughout based on React to ensure code reusability and efficiency. Moreover, frameworks like Tailwind were also leveraged to ensure a high amount of design elements could be added while keeping the code to a minimum. The focus was on getting more done in the most efficient manner possible and hence the technologies were chosen keeping the same in mind. Work was distributed based on skillset and constant communication across members was maintained.

6.2. Software Documentation

To run the project locally, clone the Github repository. Now open your terminal in the downloaded directory and run the following commands:
Npm install or yarn install
Then, run
Npm run dev or yarn run dev
Now Waves will run locally on your machine

6.2.1. Software Requirements Specification

Purpose

The SRS defines the User Interface, Frontend, and Backend of the web app, Wave version 1.0, and is intended for the following group of people:-

- Developers for the purpose of maintenance and new releases of the software
- Waves Administrators
- Testers
- Documentation writers

Intended Audience

The target audience of Waves is users trying to find out the emotion being shown through a speech. Further, the idea is to set a foundation for further discussion on Alexithymia, the inability to identify or relate with emotions.

Product Perspective

Waves is a web application that uses audio input to run an analysis of it in order to identify the emotion being shown by the speaker in the audio input. Further, the idea is to set a foundation for a further discussion on Alexithymia, the inability to identify or relate with emotions.

User Characteristics

The user need not possess any technical background or undergo special training in order to use the app as the interface will be familiar and easy to use.

External Interface Requirements

a) User Interface Requirements

- Upon logging in to the app the user can choose to either upload an audio file to be analyzed or use the recording function by clicking on the record button. The recorded audio will then be analyzed and the identified emotion displayed.

b) Hardware Interface Requirements

The user must have a laptop, phone, or tablet with an active internet connection.

6.2.2. Software Design Description

Database and Backend Design

The Backend architecture of the Waves web application is such that the web server interacts with the file system (HTML, CSS, Images) and the machine learning module in order to use the audio input, get an output, and display the same in the UI function.

The backend coding was mostly done using Javascript and Python to train the machine learning models for the module. The backend development was coded separately from the frontend using basic HTML elements for the necessary inputs and outputs. The backend uses Flask in order to allow for easy data transfer to the front-end module.

UI Interface Design

The frontend designs were ideated using Figma and developed using ReactJS,

HTML, and Tailwind CSS. ReactJS enables us to build composable user interfaces, encouraging the creation of reusable UI components which present data that changes over time.
Tailwind CSS is a CSS framework that focuses on the display rather than the functionality of the item being styled, This made it easier for the team to test out new styles and change the layout easily.

6.2.3. Software Test Plan / Quality Assurance

Routine testing was carried out in the frontend as well as the backend during the course of the development in order to provide the user with a smooth bug-free experience with Waves.

After different modules that comprised the backend were developed, unit testing was carried out on them before the integration of the backend, making sure that there were no bugs in the individual modules that could cause trouble in further stages of development or usage. In addition to this, we also ensured that there were no problems in accessing the database and retrieving the required data.

Front-end testing included unit testing of the individual React UI components for responsiveness and border cases. Integration tests were then carried out on these various components put together.

The integration of the frontend and the backend posed many new challenges and bugs from page navigation issues to server issues. We ensured that all these bugs were fixed and made an exhaustive list of every component that we could test, checking them off as we did so.

7. Project Schedule and Milestones

Total estimation of man-hours: 200

Milestones	Tasks	Hrs	Date
1 - Analysis			
1.1	Identify Deliverables	6	9/07/21
1.2	Architecture Design	4	28/07/21
1.3	UI Design on Figma	8	30/07/21
1.4	Design work plan (distribution of tasks to development teams)	10	12/08/21

2 - Development			
2.1	Identify the Dataset	10	17/08/21
2.2	Dataset processing and analysis	30	15/09/21
2.3	Model Development	25	19/09/21
2.4	Backend development	28	23/09/21
2.5	Frontend Development	35	20/11/21
3 - Integration & Testing			
3.1	Unit Testing of modules	16	21/11/21
3.2	Integration of frontend and backend services	22	25/11/21
3.3	Integration Testing	6	30/11/21
4 - Deployment			
4.1	Deploy on the web	-	Future

8. Performance and Metrics

The module has an accuracy of 81% which was deemed to be fair by the developers after having done extensive trial and error combinations with various weights and the testing to training division of the dataset.

9. Open Issues

There are various open issues with regards to Waves that we have identified are:

- The accuracy of the model is heavily reliant on existing algorithms, more work can be done to tweak and increase the accuracy of the model.
- The UI can be improved to allow for real-time detection as and when the user speaks and changes tone.

10. Security Aspects

There are several security aspects that needed to be kept in mind while developing the project. The app does not share the data of the users with any third party or any other user. All the recordings are stored locally and destroyed once the server stops running.

11. Future Scope

In the future, we plan to expand Waves to comprise many additional features such as -

- The algorithm will be optimized to provide real-time speech to emotion detection.
- Incorporating more types of emotions by collecting data from recordings.

12. Abbreviations and Definitions

- SRS - Software Requirement Specification
  It is a document that captures the complete description of how the system is expected to perform.
- ML - Machine Learning
  Study of computer algorithms that improve automatically through experience and by the use of data.
- HTML - HyperText Markup Language
  Standard markup language for documents designed to be displayed in a web browser.
- CSS - Cascading Style Sheets
  Style sheet language is used for describing the presentation of a document written in a markup language such as HTML.

13. References

[1] R. A. Khalil, E. Jones, M. I. Babar, T. Jan, M. H. Zafar and T. Alhussain, "Speech Emotion Recognition Using Deep Learning Techniques: A Review," in IEEE Access, vol. 7, pp. 117327-117345, 2019, doi: 10.1109/ACCESS.2019.2936124.

[2]N. Krüger, “How to Write a Software Requirements Specification (SRS Document),” *Perforce Software*. [Online]. Available: https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#:~:text=A%20software%20requirements%20specification%20(SRS)%20is%20a%20document%20that%20describes,A%20purpose.

[3]“Fixing Top Issues in Software Development Project Management,” *Intellectsoft Blog*, 13-Aug-2020. [Online]. Available: https://www.intellectsoft.net/blog/software-development-project-management-issues-and-solutions/.

[4]Positive Technologies, “How to approach secure software development,” *Secure Software Development: Best Practices and Methodologies for Secure SDL (LifeCycle)*, 27-Nov-2020. [Online]. Available: https://www.ptsecurity.com/ww-en/analytics/knowledge-base/how-to-approach-secure-software-development/.

[5]“What is a Project Team and who all are Involved?,” *What is a project team and who all are involved? | Invensis Learning*. [Online]. Available: https://www.invensislearning.com/articles/pmp/what-is-a-project-team-and-who-all-are-involved.

[6]“What is a Software Process Model? Top 7 models explained,” *Educative*. [Online]. Available: https://www.educative.io/blog/software-process-model-types.