**MEDI-VOICE**

**1. Tech Stack**

* **Programming Language:**
* Python 3.12
* **Audio Processing & Feature Extraction:**
* **Librosa** — for audio loading and feature extraction (MFCC, pitch, rms, mel frequency)
* **SciPy** — for signal processing (peak detection)
* **NumPy** — numerical operations
* **Selenium** — for auto testing
* **Machine Learning / Reinforcement Learning:**
* **Stable Baselines3 (SB3)** — for Deep Deterministic Policy Gradient (DDPG) implementation
* **Gymnasium** — custom environment creation for RL training
* **PyTorch** — backend framework used by SB3 for neural network models
* **Model Training:**
* Custom heart rate simulation environment built in Gymnasium
* DDPG RL algorithm for continuous action space prediction of heart rate
* Random forest classifier for age and gender classifier models

**2. Literature Survey**

|  |  |  |  |
| --- | --- | --- | --- |
| **Paper Name** | **Implemented** | **Drawbacks** | **How to solve** |
| **Heart Rate Detection and Classification from Speech Spectral Features Using Machine Learning (2020)** | Regression based algorithms on mel MEL frequency spectrum constants | Only used MEL spectrum constant feature from voice and has only been trained on English dataset | Include pitch and mfcc and other spectral constants from speech as features |
| **Heart rate monitoring using human speech spectral features (2015)** | 20 classifiers used on speech feature MEL spectrum constant | Low classification accuracy  Only one feature used  Results are in a range of heart rates not exact | Use more complex models  And use more features |
| **Analysis and prediction of heart rate using speech features from natural speech (2017)** | Used random forest regression models on SRI BioFrustration Corpus to classify emotional state and heart rate through live continuous speech | Heart rate predictive accuracy is not good | Integrating more features and using more complex models |
| **Heart Rate Extraction from Vowel Speech Signals (2012)** | Estimates heart rate from vowel speech signals mapped on a short-term Fourier transform (STFT) | Only focuses on vowel speech. | Integrating this with machine learning and expanding to more features |
| **Speech signal analysis for the estimation of heart rates under different emotional states (2016)** | Used a empirical linear predictor model to estimate heart rate. Trained on 4000 audio samples with ECG data as labels | Used Feature distances as metric to classify heart rate  And small dataset size | Improve dataset and feature extraction methods |
| **Extraction of Heart Rate Parameters Using Speech Analysis** | Entropy energy mean frequency standard deviation of the speech signal is being used to estimate the heart rate | Only showed correlation between speech and heart rate parameters | Integrating machine learning |
| **How speech processing can help with beat-to-beat heart rate estimation in ballistocardiograms** | Uses speech signlas to estimate heart rate from BCG’s | Limited generalizability | Expanded dataset |
| **Determining heart rate using speech signal** | Uses Fast fourier transforms to map the frequency differences in voice then a regression model to show correlation | Doesn’t estimate the heart rate only shows relation between speech and heart rate | Implementing complex machine learningtechniques on the concept |
| **Heart rate from read speech influenced by physical exercise** | Used pre-trained SBreathNet deep learning model to extract breathing patterns on which Independent component analysis was applied | Limited sample size  Only 7 participants data | Use other speech features along with breathing patterns to get a more accurate estimation |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**3. Abstract**

This project explores the estimation of heart rate from voice recordings by leveraging acoustic features extracted from speech signals, including MFCCs, pitch, energy, MEL spectrum constants, depth and speaking rate. Due to the scarcity of labelled heart rate data paired with voice, a heart rate simulation environment was developed to generate synthetic heart rate signals against the speech features, gender and the age group using Open AI’s gymnasium environment, for the training of a reinforcement learning model. The gender and age classifier were developed using simple random forests on labelled datasets and gave an output with 80% and 60 % accuracy respectively. Using the Deep Deterministic Policy Gradient (DDPG) algorithm within a custom Gymnasium environment, the system learns to predict heart rate values from the extracted speech features. This integration of speech signal processing with reinforcement learning provides a promising non-invasive method for estimating physiological parameters from audio data.

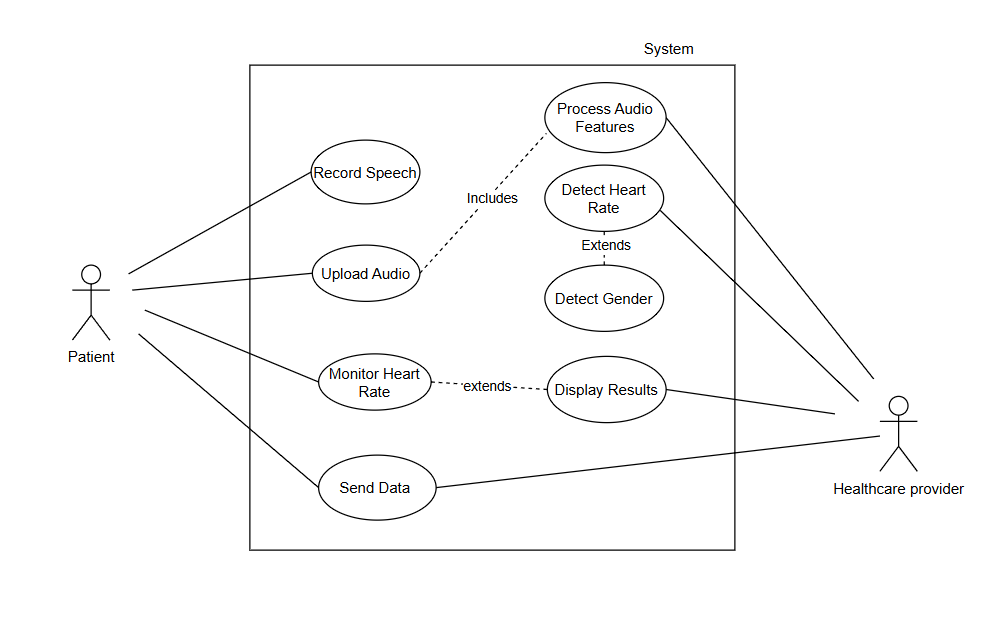
**4. Introduction**

Heart rate estimation through speech analysis uses the physiological links between vocal production and cardiovascular activity. Speech signals encode multiple features such as pitch, energy, MEL spectral coefficients, MFCCs, pitch depth, which can indirectly reflect heart rate. However, acquiring large, labelled datasets pairing speech with accurate heart rate measurements proved to be challenging.

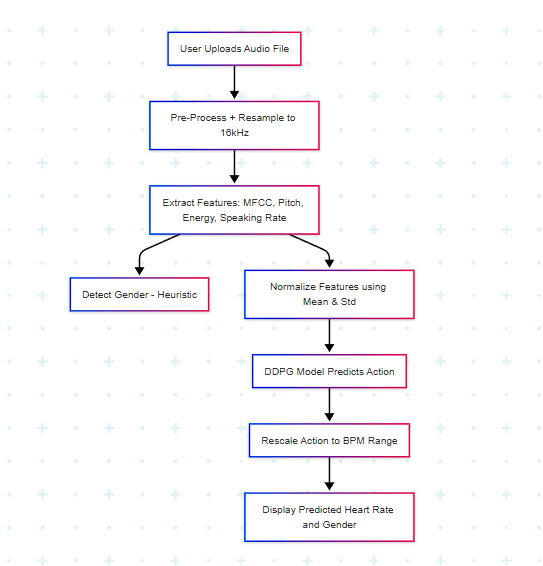
To address data scarcity, synthetic heart rate values are generated based on a heuristic function with different logics based on age and gender. An age and gender classifier are developed using simple random forest on a labelled dataset. All this allowing the creation of a labelled speech vs heart rate dataset allows reinforcement learning to learn the features. A reinforcement learning approach using Deep Deterministic Policy Gradient (DDPG) is used to predict continuous heart rate values from high-dimensional speech features. This method leverages simulated data and advances beyond traditional regression techniques, aiming for improved prediction and adaptability.

**5. System Design**

**5.1) Use Case Diagram**

****

**5.2) Workflow Diagram**

****

**6. Methodology**

This project combines speech processing, simulation, classification, and reinforcement learning to estimate heart rate from voice. All audio samples were first standardized by converting them to mono and resampling to 16 kHz. Acoustic features like MFCCs, pitch, energy, depth and MEL spectrogram constants were extracted using Librosa, and normalized using Z-score standardization to ensure uniform input for the model.

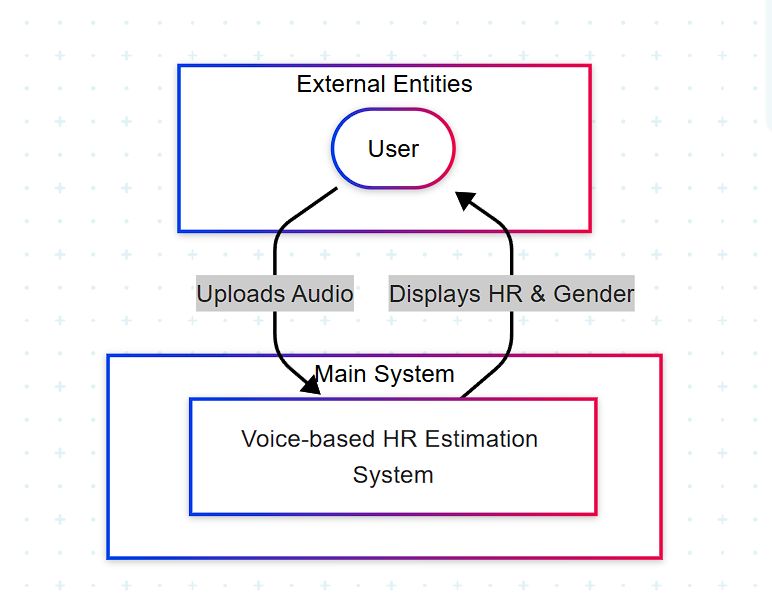
Since real-world datasets pairing speech and heart rate are scarce, heart rate values were synthetically generated based on the speaker’s predicted age and gender. Random forest classifiers trained on labelled datasets were used to assign each sample to an age group (young, mature, old) and gender(male or female). These categories were then used to assign simulated heart rate values within realistic physiological ranges, with added noise for variability using a heuristic function.

A custom environment was developed using the Gymnasium framework. It receives the extracted features as input and uses the predicted heart rate as the action. The environment calculates reward based on the absolute difference between the predicted and simulated heart rate. Additional reward shaping was introduced using adherence logic based on age. If the predicted heart rate falls within an appropriate range for the given age group, a bonus is awarded otherwise a penalty is applied.

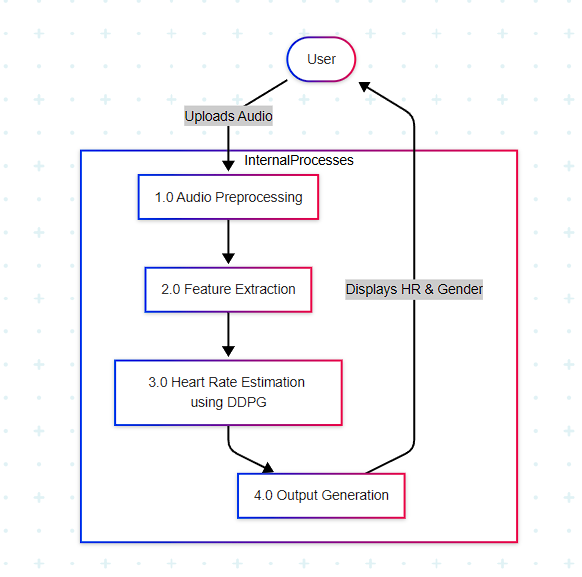
The DDPG (Deep Deterministic Policy Gradient) algorithm was used to train the agent, as it supports continuous action spaces. The model was trained across multiple episodes and over 50,000 timesteps with performance evaluated using average reward and prediction error along with standard evaluation metrics like MAE, MSE, RMSE and R^2 . This setup allowed the agent to learn from high-dimensional speech features and produce heart rate estimates that are not only accurate but physiologically appropriate for the speaker.

**7. Detailed System Design**

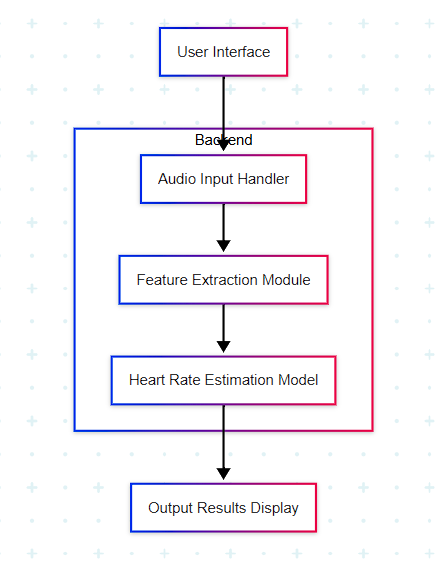
**7.1) DFD level 0 diagram**

****

**7.2) DFD level 1 diagram**

****

**7.3) Birds eye view diagram**

****

**8. Implementation**

The implementation was carried out using Pythonwith libraries like Librosa, NumPy, pandas, and scikit-learn. Audio features such as MFCCs, pitch, and energy were extracted using Librosa and normalized for model training. Random forest classifiers were trained to predict age group and gender, which were used to assign simulated heart rate values based on physiological ranges.

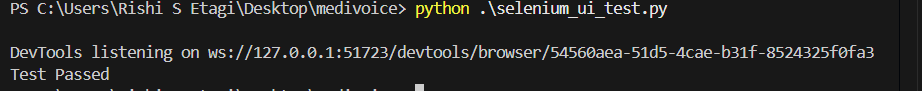
A custom Gymnasium environment was created to expose the extracted features as observations and accept heart rate predictions as actions. The environment calculated rewards based on prediction error, and included additional adherence logic that rewarded predictions within age-appropriate heart rate ranges.

The DDPG algorithm from Stable Baselines3 was used to train the agent. The training script handled environment setup and model configuration. Key training parameters like learning rate, buffer size, and number of episodes were tuned to improve performance. Debugging and testing were done using print and log statements to ensure the environment and reward mechanisms worked as intended.

All of these functionalities were used in a streamlit web based application as frontend.

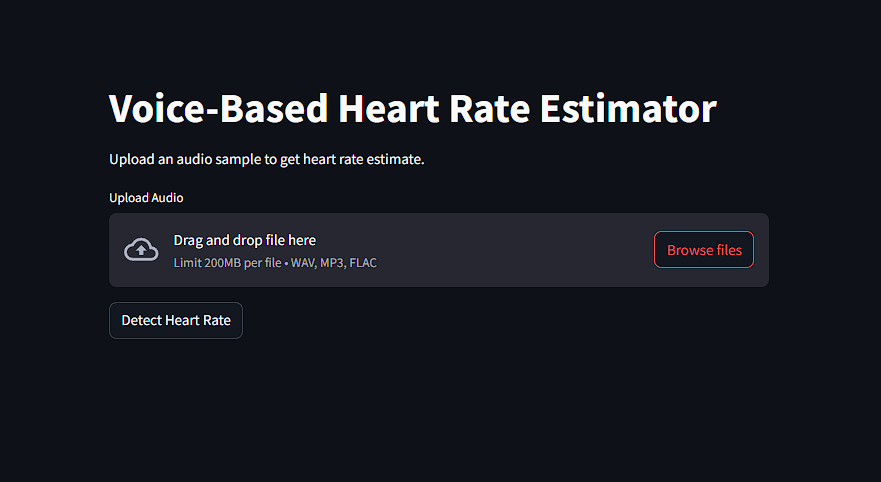
**9. Testing**

Functionality testing, UI/UX testing, smoke testing, Regression testing performed manually and automatically using selenium on the streamlit app.

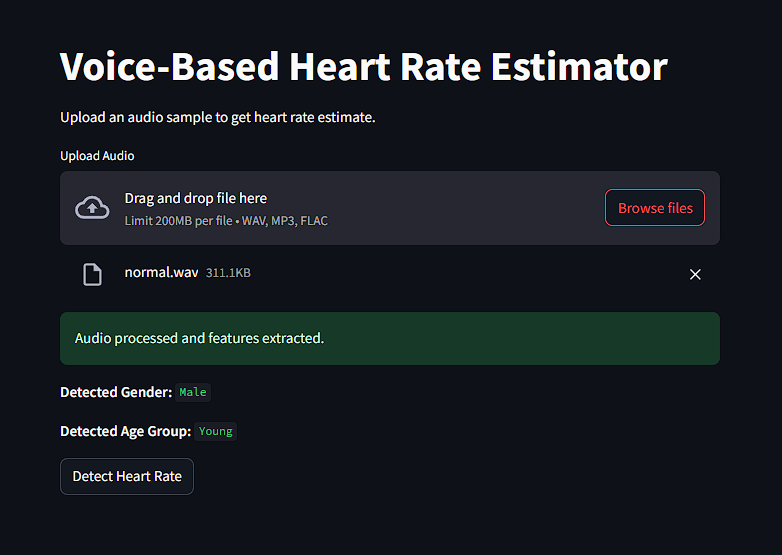
****

**10. Results and conclusion**

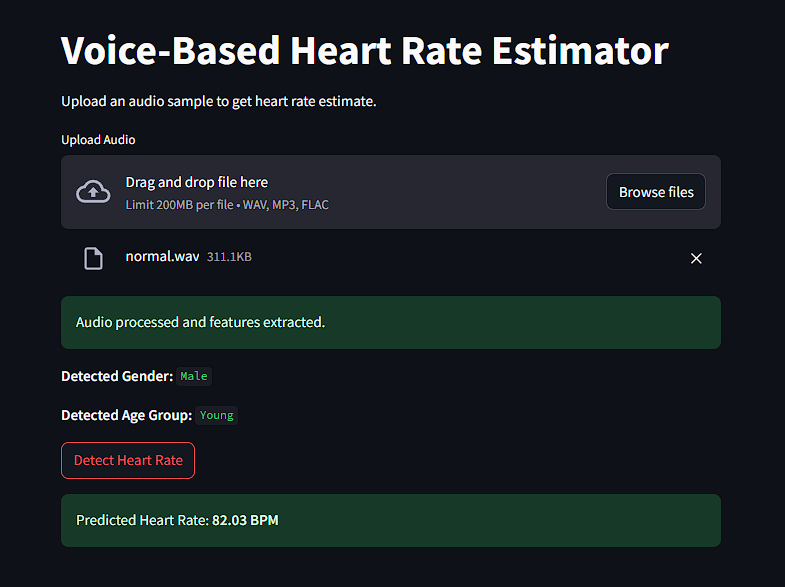
**Streamlit application**

****

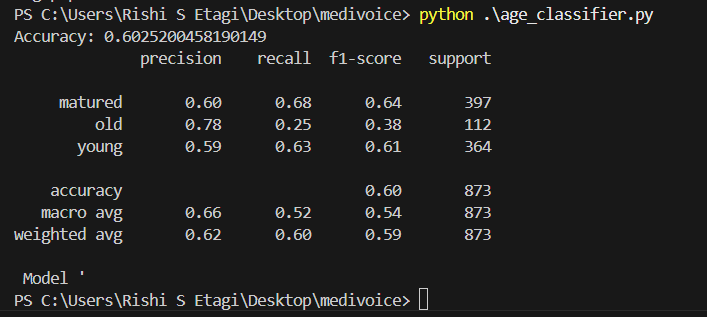
**After pre-processing feature extraction and age gender classification**

****

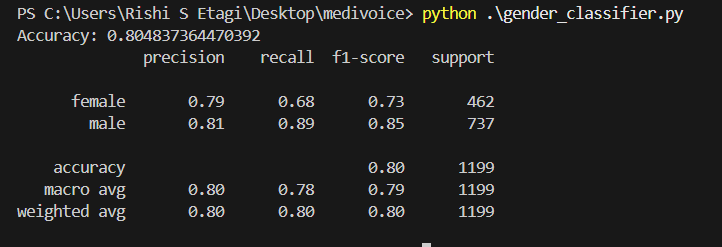
**After detecting heart rate**

****

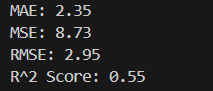
**Age classifier**

****

**Gender classifier**

****

**Heart rate detecting model**

****

We were able to achieve an accuracy of 80% on the gender classifier and an accuracy of 60% on the age group classifier.

The heart rate detecting model trained on DPPG policy under reinforcement learning environment was able to achieve an R^2 value of 0.55 and a RMSE value of 2.95

All the functionalities were used in a streamlit application as the final product