Guardian Labs
Executive Summary

Ankita Avadhani
Steven Aramony
Maira Shahid
Gurjus Singh
Alfred Yi

Key Problems

The **three key problems** we are trying to solve are the following:

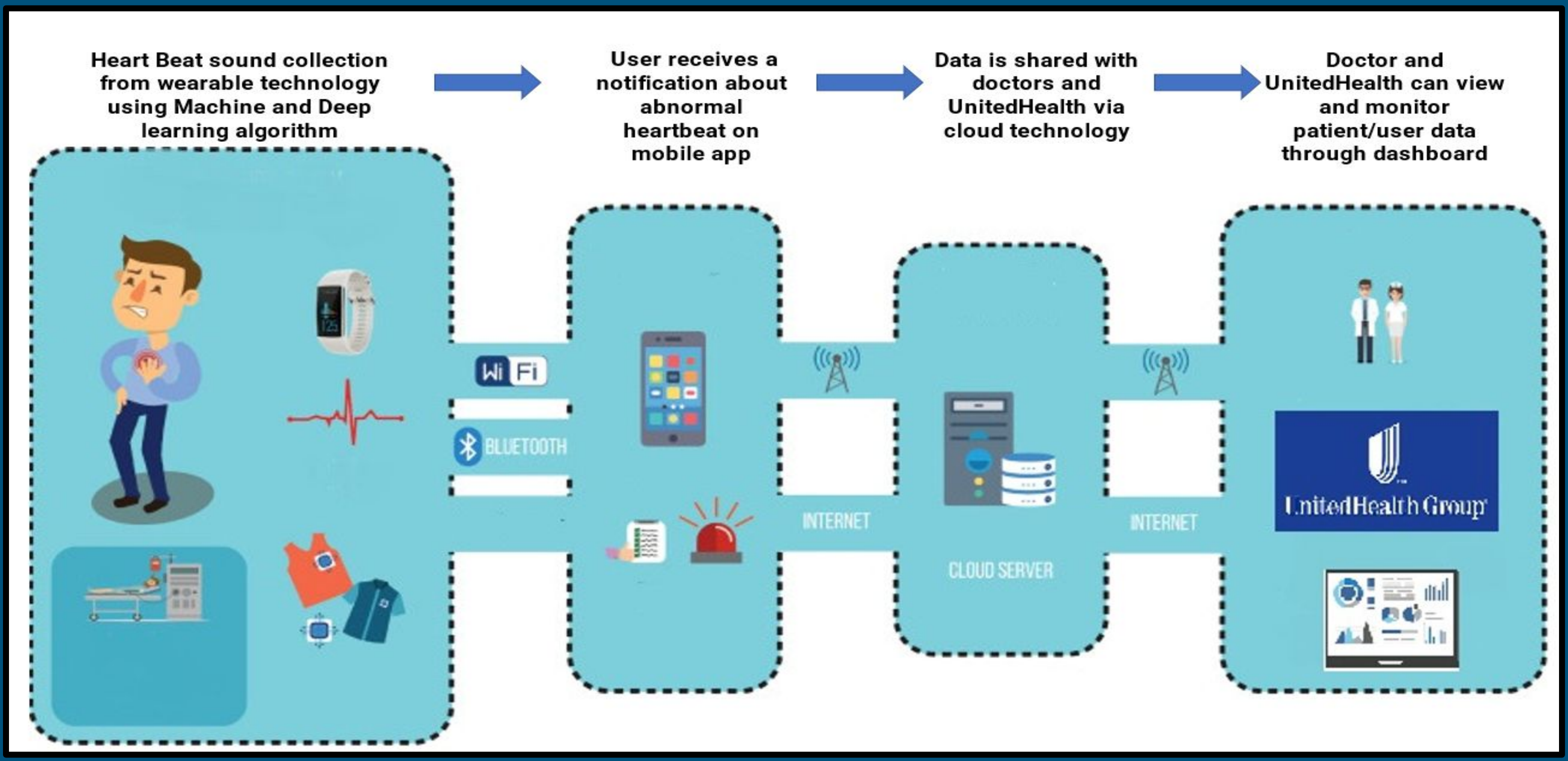
1. Heart disease is costing the companies too much money.
2. Traditional heart disease detection methods lack accuracy and sophistication.
3. Data between wearable tech and insurance company apps are not integrated.

See the full **Initial Findings Report** for more context and background on each of these problems.

Guardian Labs is creating a solution to pull wearable technology data and create a machine learning algorithm that can predict when abnormal heartbeats exist by classifying the heartbeat into one of four categories: Normal, Murmur, Extrasystole, or Artifacts.

By partnering with UnitedHealth Group, our data services and algorithms can be embedded in a wide range of devices, including the company's internal and external apps across its vast network of healthcare providers and patients, to seamlessly monitor heart beat data via wearable tech and alert users and primary care physicians (or cardiologists) of a potential heart condition before it's too late. See figure 1 for an overview of our mobile health data workflow.

Figure 1. *Overview of the Mobile Health Data Workflow*



Project Goals, Objectives & Solutions

01 — **Learn to distinguish different heartbeat sounds (e.g., normal vs. abnormal heartbeats)**

Deliverable:
- Exploratory data analysis (EDA) of heartbeat sounds
- Creating of sound waves charts

Success Criteria
- Clear understanding of different heartbeats

02 — **Create and extract features from audio data**

Deliverable:
- Pre-processing of audio data using normalization & sampling techniques

Success Criteria
- Data ready to use for modeling purposes

03 — **Creation of state-of-the-art classifier for the early detection of heart disease**

Deliverable:
- Build machine and deep learning models to classify heartbeat sounds
- Comparison of modeling approaches and algorithm performance

Success Criteria
- Data ready to use for modeling purposes heartbeats

04 — **Implementation of classification model algorithm**

Deliverable:
- Exploratory data analysis (EDA) of heartbeat sounds
- Creating of sound waves charts

Success Criteria
- Clear understanding of different heartbeats

05 — **Build interactive dashboard and mobile app that clients (e.g., medical providers and patients) can use to monitor/detect abnormal heartbeats**

Deliverable:
- Model deployment and user acceptance testing (UAT) by all team members

Success Criteria
- Easy usability of dashboard and mobile app by doctors and patients

06 — **SUPPLEMENT GOAL: Develop strategy to expand product to wider client base**

Deliverable:
- Target at least two new market segments:
  - -new wearable technology (data input)
  - -new client base

Success Criteria
- High-level strategic analysis confirming which wearables and consumers we could target and establish partnerships with

Benefits of Solution Implementation

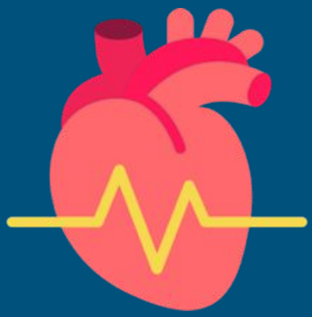
Reduce Cost for Stakeholders

Guardian Labs is seeking to reduce UnitedHealth Group's spending on cardiovascular disease by at least 5%. Similarly, our product would enable any individual—doctor or patient—to detect heart abnormalities faster and more accurately, resulting in more affordable preventative treatment.

Enhance Marketing Capabilities

Our product will equip UnitedHealth Group and its affiliates with the precise data and insights necessary to target specific users and patients with relevant information, including customized healthcare plans and treatment options

Contribute to Heart Health & Disease Research



Our product would enable UnitedHealth Group to attract more mobile app users/participants through multiple apps, which in turn would increase the amount of data Guardian Labs could use to improve our heartbeat detection algorithms..

Key Findings 5

Guardian Labs has made significant progress in analyzing the publicly available heartbeat dataset derived from the following sources:

- **iPhone Data:** general public via the iStethoscope Pro iPhone app (dataset A)
- **Medical Grade Trials:** clinical trial in hospitals using the digital stethoscope DigiScope (dataset B)

Most Effective Model

After extensive EDA, various layers of data preprocessing, feature extraction, and finally model training work, we determined that using a **Convolutional Neural Network** (CNN) model is the most effective in classifying heartbeat sounds. The CNN model yielded the best sound classification accuracies and precision among all the models with an accuracy of 83% and precision of 86%.

Balanced Data is Essential

In comparing the accuracy metrics, we observed that some models were underperforming in classification. Most of the models are overfitting normal heartbeats as murmur, extrasystole and artifacts. We believe that this is because of an imbalance of the dataset (i.e., the majority of the audio files are normal heartbeat sounds).

More Data Samples Required

The number of training and testing sets is small, which also leads to overfitting of data. Experimenting with data augmentation methods to account for the imbalanced records resulted in poorer performance; suggesting that collecting more data samples in the future can mitigate this issue (i.e., increasing the training dataset). However, when selecting methods to use in practice, more work needs to be done. Improvements to feature extraction techniques and collection of more data need to be made to improve the result of these models.

Exploratory Data Analysis

We conducted extensive EDA to understand the data and structure of the audio files. We used various graphs to identify key distinctions between the classes and their respective sound features. See the full **Initial Findings Report** for more details on each EDA step.

- **Auditory Inspection**: We used `IPython.display` to listen to each audio file.
- **Sampling Rate**: Sampling Rate indicates how many times (in one second) the data is collected (i.e., Hertz, abbreviated as Hz). Since all the files have uniform sampling rate ie.22050, we did not have to apply any sampling conversion techniques.(figure 4.)
- **Audio Length**: The audio lengths in the dataset vary between 1 and 30 seconds. (figure 3.)
- **Wave Plots**: We used the `librosa.display.waveplot` function in Python's Librosa library to display waveform visualization of the amplitude vs the time representation of the signal. Waveform plots indicate how loud an audio is at a given time (see figure 2).

Figure 3. *Duration of audio files*

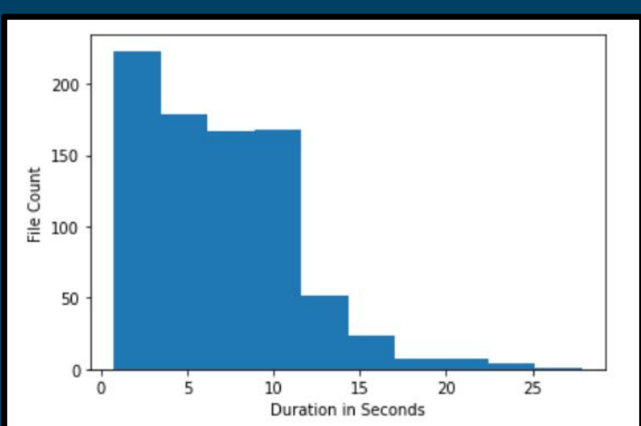


Figure 2. *Waveplot of heartbeat classes: red arrows highlight the distinct frequency patterns between each heartbeat class*

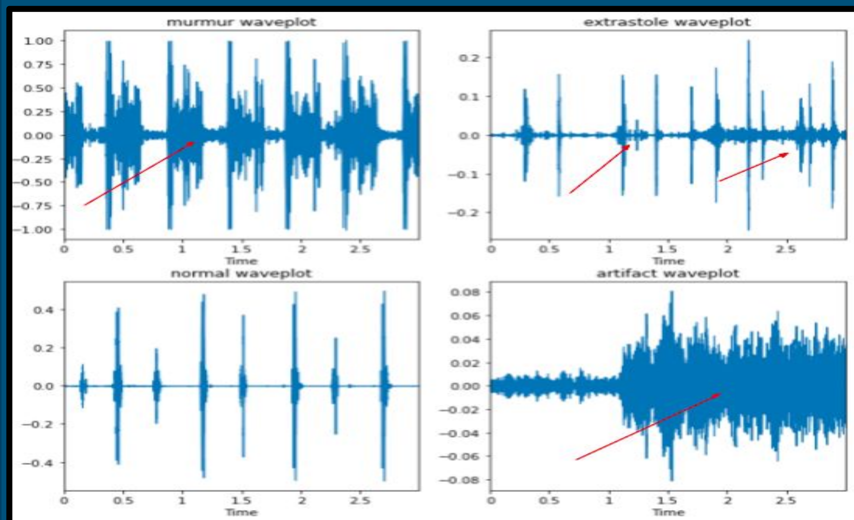


Figure 4. *Sampling rate, signal length and duration of audio files*

	filename	sampling rate	Signal_Length	Duration
610	/content/drive/MyDrive/Dataset/set_b/normal__1...	22050	48902	2.217778
819	/content/drive/MyDrive/Dataset/set_b/normal_no...	22050	51746	2.346757
290	/content/drive/MyDrive/Dataset/set_b/normal__1...	22050	122736	5.566259
559	/content/drive/MyDrive/Dataset/set_b/normal__1...	22050	75208	3.410794
168	/content/drive/MyDrive/Dataset/set_a/normal__2...	22050	198450	9.000000
687	/content/drive/MyDrive/Dataset/set_b/normal__2...	22050	37067	1.681043
818	/content/drive/MyDrive/Dataset/set_b/normal_no...	22050	102494	4.648254
86	/content/drive/MyDrive/Dataset/set_a/murmur__2...	22050	174979	7.935556
260	/content/drive/MyDrive/Dataset/set_b/normal__1...	22050	58554	2.655510
547	/content/drive/MyDrive/Dataset/set_b/normal__1...	22050	212408	9.633016

Data Overview

The dataset A and B consist of audio files which contain the heartbeat sound. The heartbeats were classified into four different categories: Normal, Murmur, extrasystole and Artifacts. Prior to diving into the analysis of the data, it is essential to establish a firm understanding of these four heartbeat categories. See the full **Initial Findings Report** for more details on the dataset.

Data Preprocessing

Prior to inputting the audio file paths into the models, we had to process and convert the data into a format the models could understand and accept. The following preprocessing steps were taken to achieve this goal. See the full **Initial Findings Report** for more details on each data preprocessing step.

- **Audio File Loading:** We used the `librosa.load()` function in to load audio files.
- **Combining Datasets:** Datasets A and B were combined into one to facilitate the model training process.
- **Data Cleaning and Missing Value:** We dropped irrelevant columns from the dataset and manually labeled files with missing values to improve model training efforts.
- **Reducing File Load Time:** We used specific parameters to load audio files faster.
- **Normalization:** We normalized the data since each file had a different amplitude.
- **Resizing**: For consistency, all files were resized to a 2-second sugement. Files with a duration of under 2 seconds were dropped. This approach enabled us to expand the dataset while retaining key information.
- **Converting Labels, Splitting Dataset**: All the labels were encoded using `Sklearn.preprocessing.LabelEncode` to convert from categorical text data into a model understandable numerical form. After performing the data processing steps and feature extraction steps, `sklearn.model_selection.train_test_split` was used to split the data into training and testing sets. **80%** of the data was used for training purposes and **20%** was used for testing.

Figure 5. *Before labelling missing data*

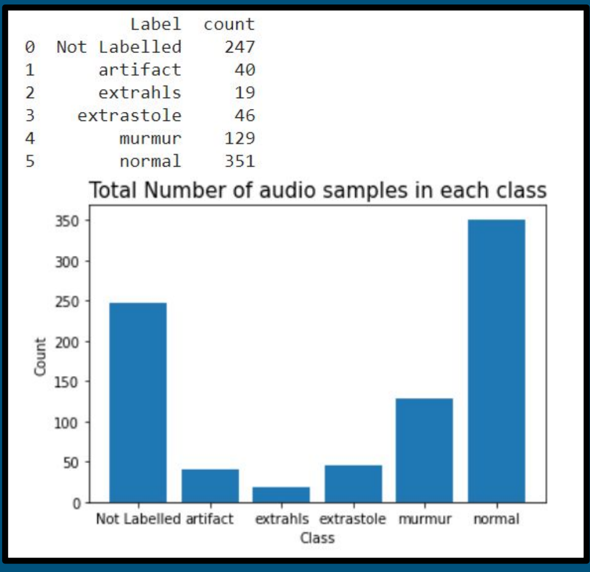
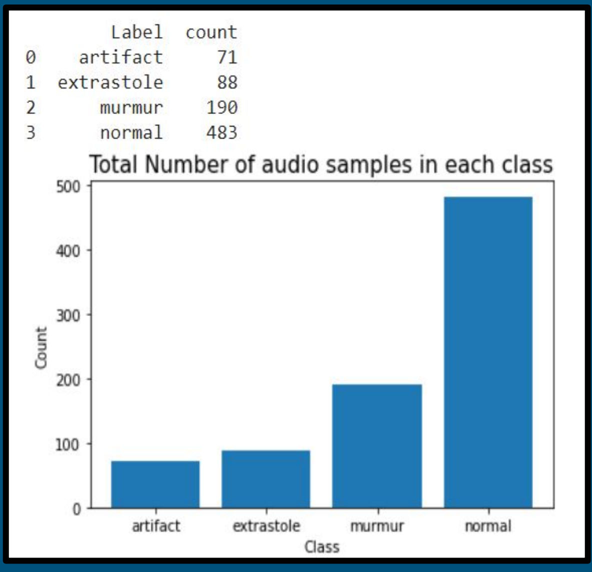


Figure 6. *After labelling missing data*

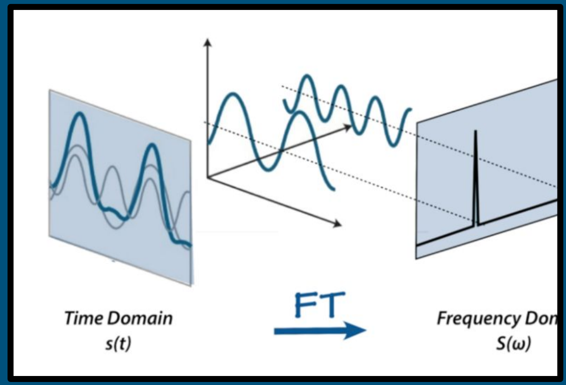


Fourier transform

In machine learning, feature extraction is the process of identifying and extracting the most impactful information that is used to train a classification model.

The **Fourier transform** is a mathematical formula that allows us to decompose a signal into its individual frequencies and the frequency's amplitude. In other words, it converts the signal from the time domain into the frequency domain (see figure 7).

Figure 7. Visualization of the Fourier transform



Mel-Frequency Cepstral Coefficient (MFCC)

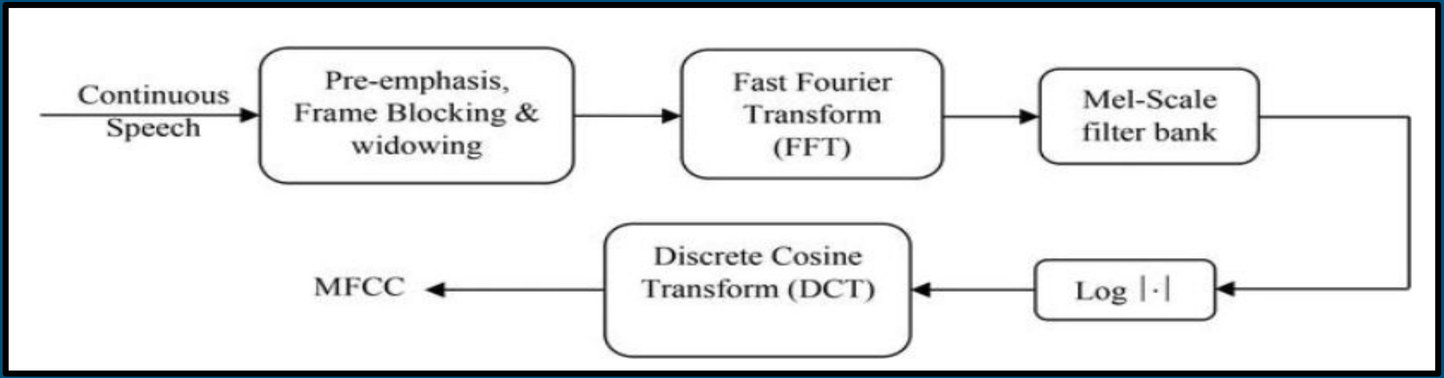
This feature is one of the most important methods to extract a feature of an audio signal and is used majorly whenever working on audio signals. The MFCCs of a signal are a small set of features (usually about 10–20) that extract the Cepstral Coefficients using Discrete Cosine Transform (DCT).

After extracting the MFCC's 39 feature, the audio data was converted into an array of discrete numbers for machine learning models. Figure 8 shows the snapshot of data.

```
x_data

array([[ 8.36856841e-01,  2.13635938e+00],
       [-1.41365810e+00,  7.40962324e+00],
       [ 1.15521298e+00,  5.09961887e+00],
       [-1.01861632e+00,  7.81491465e+00],
       [ 1.27135141e+00,  1.89254207e+00],
       [ 3.43761754e+00,  2.61654166e-01],
       [-1.80822253e+00,  1.59701749e+00],
       [ 1.41372442e+00,  4.38117707e+00],
       [-2.04932168e-01,  8.43209665e+00],
       [-7.11099611e-01,  8.66043846e+00],
       [-1.71237268e+00,  2.77780226e+00],
       [-2.67000792e+00,  8.35389140e+00],
       [ 1.24258802e+00,  4.50399192e+00],
       [-2.22783649e+00,  6.89479938e+00],
       [ 1.45513831e+00, -2.91989981e-02],
       [ 4.53791789e-01,  3.95647753e+00],
       [ 1.06923853e+00,  4.53068484e+00],
       [ 2.56936589e+00,  5.07048304e-01],
```

Figure 8 *MFCC Extraction Process and Data after feature extraction*



Mel Spectrogram

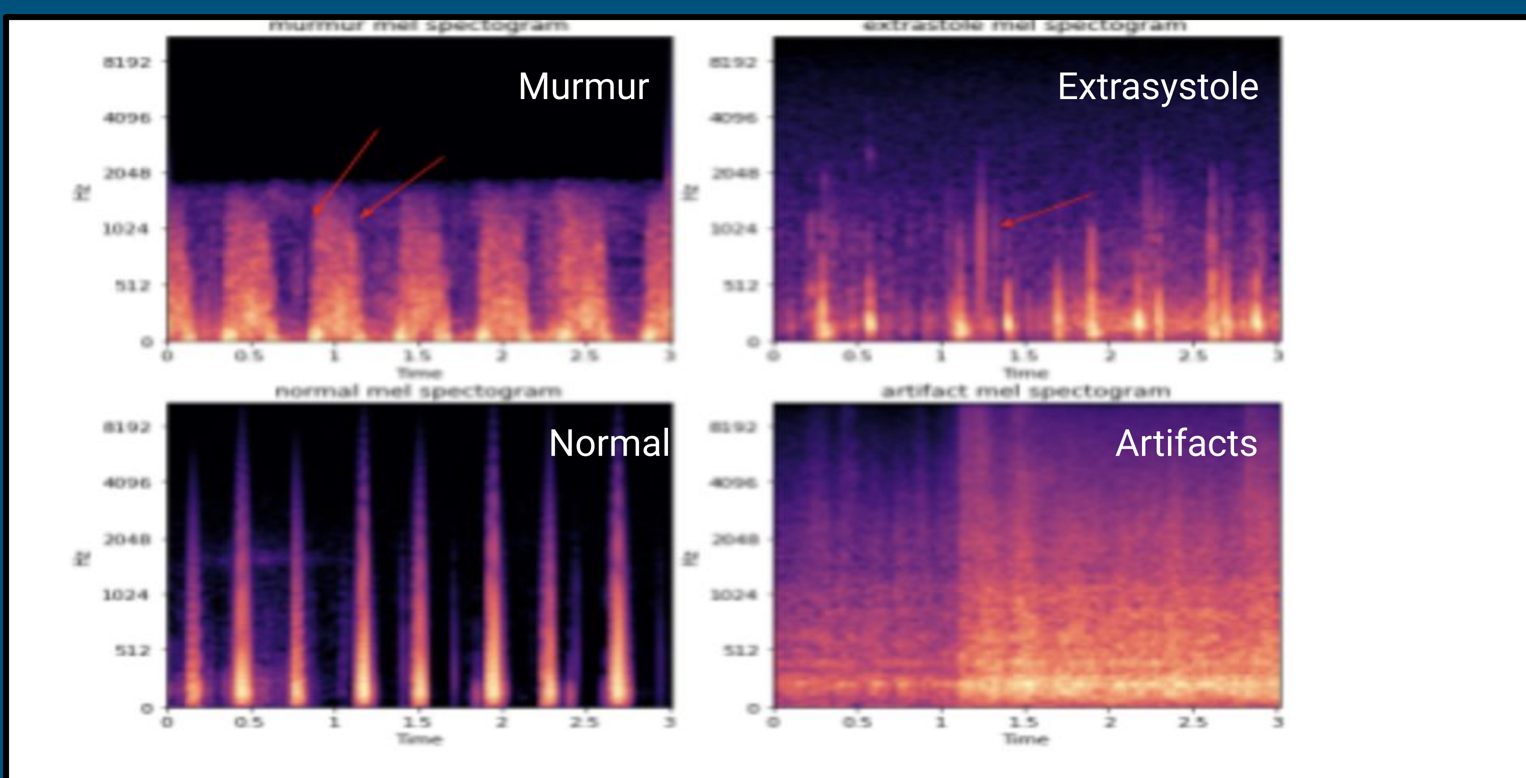
The spectrogram is a concise ‘snapshot’ of an audio wave and since it is an image, it is well suited to being input to CNN-based architectures developed for handling images.It plots Frequency (y-axis) vs Time (x-axis) and uses different colors to indicate the Amplitude of each frequency.

Figure 9 shows the spectrogram graphs of each category. The energy content of a signal expressed as a function of frequency, time, and amplitude are represented as a heat map. The darker orange color in the spectrogram indicates a higher sound frequency.

- The **artifact spectrogram** shows a lot of dark orange which explains that there is an abundance of high frequency energy like background noise, music or people talking.
- The **murmur spectrogram** shows intermittent noise sequences of amplitude with a wider frequency spectrum. These are whooshing sounds between heart beat cycles.
- The **extrasystole spectrogram** shows the peaks of high energy between cycles which indicates extra heart sounds.

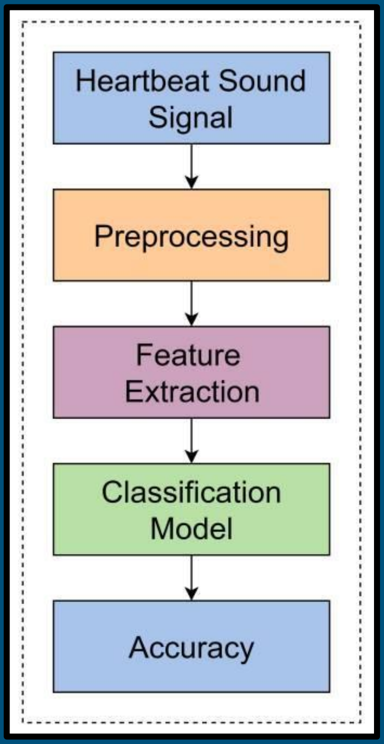
All the three spectrograms clearly deviate from a normal spectrogram, which has clear and consistent spikes for each heart cycle (i.e., lub dub).

Figure 9. *Spectrograms of Heartbeats*



The model building consists of three steps: pre-processing, feature extraction and the training the classification models. Figure 10 shows the process Guardian Labs took to create heartbeat classification models

Figure 10. *Guardian Labs Modeling Process*



Overview of Models

Different classification algorithms were used in this study. The following methods were used: Naive Bayesian Classifier, Support Vector Machines (SVM), K Nearest Neighbor (KNN), Random Forest and Gradient Boosted Trees. Various deep learning models were also built by adjusting hyperparameters (e.g., number of neurons, number of hidden layers, dropout regularization, and batch normalization). The following neural networks were used: Deep Neural Network (DNN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short Term Memory (LSTM).

See the full **Initial Findings Report** for more details on each of these models/algorithms.

Machine Learning Performance

Tables 1 and 2 show the different accuracy metrics for machine learning models. The performance of each machine learning model is recorded with respect to its given accuracy. We conducted different experiments and evaluated the averaged precision, F1 score, and accuracy of all models with MFCC's features.

Among the traditional machine learning models, random forest with n-estimators=500 performed the best with an overall precision of 78%. It performed strongly on artifact, murmur, and extrasystole, but underperformed on normal heartbeats.

TABLE 1. PRECISION SCORES BY HEARTBEAT CLASS (%)

CLASS	Naive Bayes	SVM	Random Forest	Gradient Boosting	KNN
ARTIFACT	60	66	93	91	81
EXTRASYSTOLE	16	20	100	50	37
MURMUR	50	53	85	68	85
NORMAL	66	66	67	67	69

TABLE 2. PERFORMANCE SUMMARY RESULTS FOR MACHINE LEARNING (%)

METRICS	Naive Bayes	SVM	Random Forest	Gradient Boosting	KNN
OVERALL ACCURACY	57	57	72	69	71
PRECISION	56	59	78	69	72
RECALL	57	57	72	69	71
F-SCORE	56	57	68	67	70

Figure 23 shows the correlation matrix of machine learning models. The results suggest that our models are overfitting the data. SVM is performing worst among all models, whereas random forest performs best among all five models.

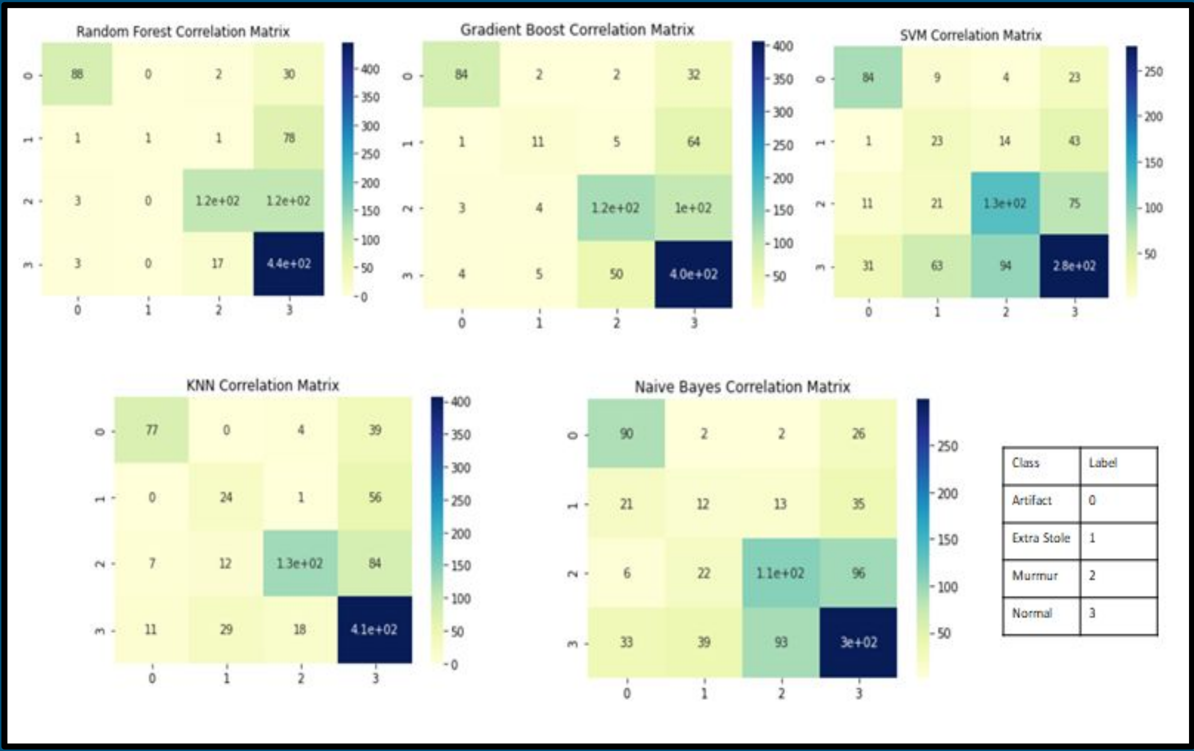


***Figure 23.** Correlation matrices of machine learning models*

Deep Learning Performance

Different deep learning models were conducted to evaluate which model yields better results. We experimented with changing the hyperparameters of the models like number of neurons, changing the number of layers, changing the dropout ratio in each layers etc. Table 3,4,5 shows the experiments and results of DNN, CNN and LSTM/RNN.

The CNN model 4 with four layers and only one dropout ratio yielded the best sound classification accuracy and precision among all the models with an accuracy of 83% and precision of 86%. The CNN classifier performed extremely well on normal heartbeats and decently for murmur and artifacts, but was unable when classifying extrasystole heartbeats. This is most likely due to the fact that the dataset contains far more normal, murmur and artifacts data than extrasystole samples; hence, affecting the model's ability to distinguish between classes.

Our both machine and deep learning models generally underperformed on the minority extrasystole class. In order to improve the performance, we attempted to balance the data using data augmentation methods, such as Synthetic Minority Oversampling Technique (SMOTE). Unfortunately, fitting and evaluating the models after applying the SMOTE method yielded lower scores.

TABLE 3. PERFORMANCE SUMMARY RESULTS FOR DENSE NEURAL NETWORK(%)

Experiment	*Model*	*Neurons*	*OVERALL ACCURACY*	*PRECISION*	*RECALL*	*F-SCORE*
Experiment 1	DNN	12	67	70	67	68
Experiment 2	DNN	32	69	75	69	71
Experiment 3	DNN	64	65	67	65	65
Experiment 4	DNN	128	68	71	68	69
Experiment 5	DNN	250	68	72	68	69

TABLE 5. PERFORMANCE SUMMARY RESULTS FOR LONG SHORT-TERM MEMORY (%)

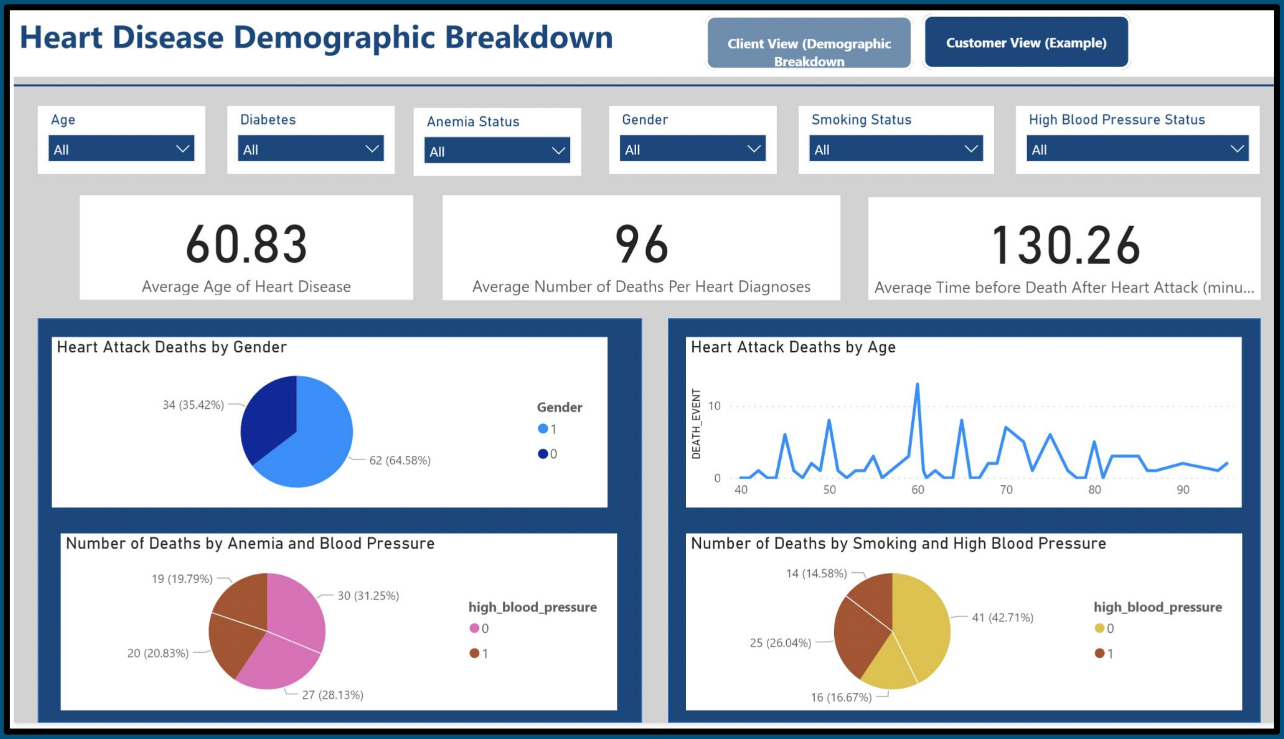
Experiment	*Model*	*Neurons*	*# of layers*	*OVERALL ACCURACY*	*PRECISION*	*RECALL*	*F-SCORE*
Experiment 1	Bi-Directional LSTM	64,32	2	60	61	60	60
Experiment 2	Simple RNN	64,32	2	45	45	45	45
Experiment 3	UniDirectional & Bidirectional LSTM	64,32	2	59	60	59	60
Experiment 4	Bi-Directional LSTM	64,128,128	3	61	62	61	61
Experiment 5	Simple RNN	128,64,64	3	49	49	49	49

TABLE 4. PERFORMANCE SUMMARY RESULTS FOR CONVOLUTIONAL NEURAL NETWORK (%)

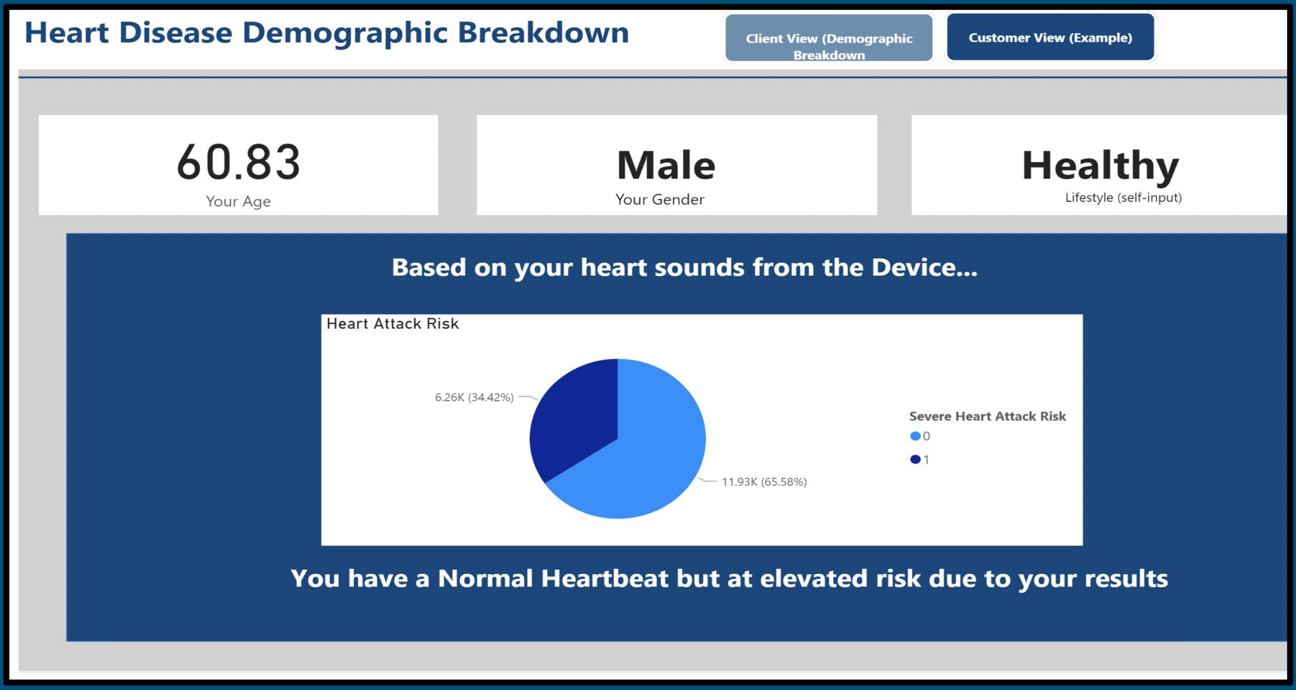
Experiment	*Model*	*Neurons*	*# of layers*	*Dropout Layer*	*OVERALL ACCURACY*	*PRECISION*	*RECALL*	*F-SCORE*
Experiment 1	CNN	16,32,64,128	4	0.2, 0.2, 0.2, 0.5	75	81	75	76
Experiment 2	CNN	16,32,64,128	4	0.2, 0.2, 0.2	76	80	76	77
Experiment 3	CNN	16,32,64,128	4	0.5, 0.5 , 0.5	70	78	70	72
Experiment 4	CNN	16,32,64,128	4	0.2	83	86	81	82
Experiment 5	CNN	16,32,64,128	4	None	78	77	78	77

Data is aggregated, processed, and ingested into our models to populate a summary of the user/patient heart health status. Two views will be available depending on the user:

Client View (e.g., insurance or healthcare providers): Contains detailed information about disease demographics based on users' data.

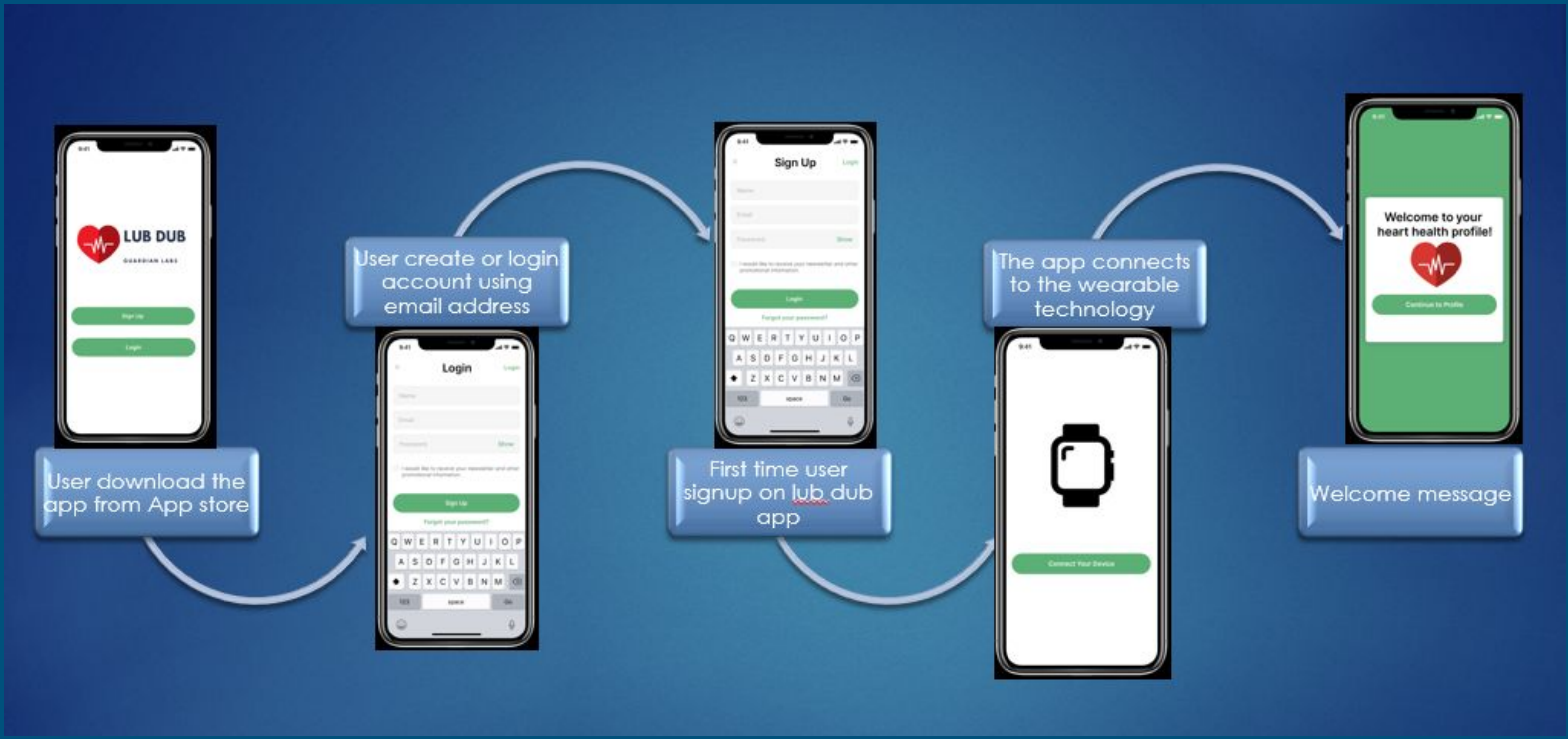


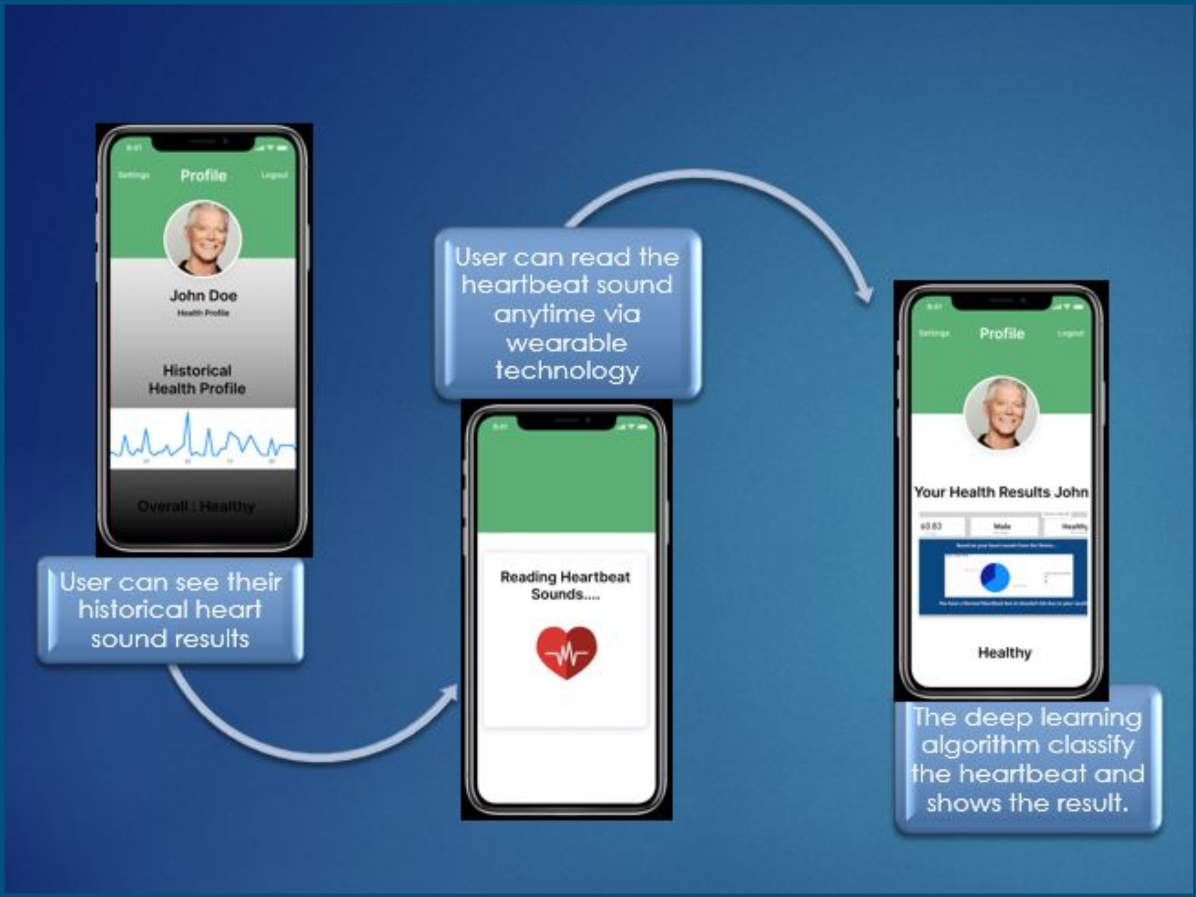
User View (e.g., app users or patients): Contains basic information about the user's heart health status based on data collected via wearable technology.



Link to live demo: https://www.youtube.com/watch?v=2Bq1Kf7Lupk

Our LUB DUB mobile app is designed to provide users a fun and seamless experience in integrating their heartbeat data with any wearable technology. Data will be streamed to the dashboard, which is viewable within in the app.





Link to live demo:
https://www.youtube.com/watch?v=lbQVLWLXoDk

Challenges

We have identified several challenges while working with the heartbeat sound dataset. The key challenges to keep in mind as we continue to build our model and products are the following:

- **Unlabeled Data:** Dealing with unlabeled data was a significant challenge because falsely classifying the audio files will directly impact our model performance in future. In order to overcome this challenge, we will continue our collaboration with substantive experts in the medical field to resolve this issue..
- **Varying Audio Lengths:** The different length of audio files needs to be standardized.
- **Varying Frequencies:** Heart sounds in the low frequency components, with noise in the higher frequencies.
- **Background Noise:** The data is gathered in real-world situations and frequently contains background noise of every conceivable type. The differences between heart sounds corresponding to different heart symptoms can also be extremely subtle and challenging to separate. Success in classifying this form of data requires extremely robust classifiers. (Bentley et al., n.d.)
- **Limited Datasets:** The dataset is limited in terms of the number of training and testing examples.

Conclusion

This report studied the leading cause of death for men, women, and people of most racial and ethnic groups in the United states i.e. heartbeat abnormality. Guardian Labs investigated the problem by using audio recording collected from digital stethoscope. From the four possible classes, the machine and deep learning models were trained to predict the class of testing heartbeat recording.

After conducting various experiments and techniques, our findings suggest using Convolutional Neural network with four layers, number of neurons as 12,32,64 and 128 and one dropout layer with a ratio of 0.2 as optimal model to identify the abnormal heartbeats due to its higher accuracy of 83 %, precision of 86% and f-score of 83%.Guardian Lab decided to use this model for classify heart beat sounds in our wearable technology.

Expand Product to Wider Client Base

Guardian Labs is planning to expand its client base to more insurance and health care companies by actively connected with them and present the findings of this work.

More Data Collection

One of the very important future task for this project is to collect more data. By actively engaging with more insurance companies, Guardian Lab will not only increase its client base but will also be able to collect more data.The machine and deep models were built on small number of test and train data sets. Increasing the number of sample size will yield better results and mitigate the effect of imbalance classes on model performance.

Better Feature Extraction with Dimensionality Reduction

Use t-SNE for dimensionality reduction with more data. t-SNE can help Guardian lab in following ways:

- Labelling the missing data with the help of t-SNE visualization.
- Help with manual search in collections of samples which often lead to cumbersome task of listening to the contents of the audio in sequential playback.
- If applied, t-SNE visualization could be a capable tool for the exploration and understanding of sound collections.

Cloud Computing

As a healthcare technology company focused on applying innovative data and analytics solutions to transform the healthcare field through AI, Guardian Labs is planning to use Amazon Web Services (AWS) to efficiently build, train, and deploy our machine learning (ML).A WS would not only accelerate the team's ML work, but also enhance our data collection/processing and project management efforts.

- Streaming Data Collection & Processing
- Efficient & Collaborative Modeling Process
- Real-Time Progress Monitoring

Addition of Natural Language Processing Techniques

We want to achieve the following by implementing NLP processes:

- Improve sound data accuracy with unstructured data for our customer clients through surveys.
- Improve risk adjustment models for our insurance clients.
- Lower the overall cost by using open sourced packages to enhance the qualities of our product.