Guardian Labs
Final Report

Ankita Avadhani
Steven Aramony
Maira Shahid
Gurjus Singh
Alfred Yi

Acknowledgement

Special thanks to Dr.Muhammad Ali Shahid DO, for helping the team manually label missing heartbeat audio data.

Table of Contents

About Us

Guardian Labs is a leading healthcare technology company, focused on applying innovative data and analytics solutions—including data integration and machine learning—to help improve clinical decision-making.

- We are a team of scientists, engineers, and consultants who are passionate about saving lives and transforming the healthcare field through Artificial Intelligence and personalized data.
- We are silent protectors that use state-of-the-art technology to save lives by helping clients—both providers and patients—make more informed choices and identify health risks earlier.
- We are a leader in the field of Data Integration and Machine Learning, developing algorithms, software, and APIs that seamlessly integrate with wearable technology.
- We are an effective partner for healthcare providers and insurance companies, offering unmatched software and data analytics solutions to lower healthcare costs for all stakeholders.

See **Appendix 1** for more details on team members, including roles and responsibilities.

“ Better Data. Better Lives”

Team Member Introductions

Guardian Labs consists of five members that bring diverse and global perspective, knowledge, and skill sets in the areas of business, finance, technology, and international affairs.

Steven Aramony

Product Manager

- Experienced Consultant and Technical Product Owner in the Financial Services and Fintech space
- M.S. in Data Science Candidate, Northwestern University
- B.S. in Finance, Real Estate, and Marketing, University of Virginia

Maira Shahid

Machine Learning Developer

- Experienced data analyst with 6 plus years of data analytics work in Finance, HR and Audit
- M.S. in Data Science Candidate, Northwestern University
- M.BA, Wayne State University
- B.S in Finance and minor in IT, University of the Punjab, Lahore, Pakistan

Alfred Yi

Technical Project Manager

- Experienced Political Officer with a demonstrated history of working in government, advising policymakers on developing strategies to analyze big data across mass/social media platforms.
- M.S in Data Science Candidate, Northwestern University
- B.A. in Political Science & Chinese, Middlebury College

Gurjus Singh

Machine Learning Developer

- 2 Years of experience in EHR/Medical Applications
- M.S in Data Science Candidate, Northwestern University
- B.A. Cognitive Science, UC Berkeley

Ankita Avadhani

Data Visualization Developer

- Experienced business insights developer working in marketing analytics in the software industry, specializing in developing ML models to drive marketing channels
- M.S in Data Science candidate, Northwestern University
- B.S Information Science, University of Michigan

Roles and Responsibilities

Technical Project Management

Alfred Yi will serve as the primary Technical Project Manager at Guardian Labs. Alfred brings a high level of technical expertise as well as organizational, leadership, and communication skills. As the technical project manager, he will be responsible for defining business requirements, deliverables, as well as roles and responsibilities for all team members. He is also in charge of coordinating regular technical meetings with machine learning engineers to ensure their research and findings are aligned with project goals and objectives.

Machine Learning

Maira Shahid and Gurjus Singh will be the primary Machine Learning Engineers at Guardian Labs. Maira and Gurjus bring a wealth of experience and technical knowledge in Python programming and data analytics. Their primary role will include assessing and cleaning datasets, and building/designing complex machine learning and deep learning models.

Data Visualizations

Ankita Avadhani will serve as our Data Visualization Expert and App Developer. Ankita will be responsible for developing data visualizations tools to generate insights and communicate stories in the most dynamic way to stakeholders. Ankita will also serve as the team's technical writer, which includes translating complex information and jargon into simple, polished, and engaging content for our stakeholders.

Product Ownership

Steven Aramony will serve as Product Owner at Guardian Labs. He is in charge of driving business value and ensuring the team's product meets the minimum specifications needed to accomplish the overall goal. He will also serve as the secondary *technical project manager*, which includes developing status report templates and ensuring the team adheres to the project timeline

Role Prioritization

P = Primary
S = Secondary

Roles	Alfred Yi	Steven Aramony	Maira Shahid	Gurjus Singh	Ankita Avadhani
Technical Project Manager	P	S			
Product Owner		P		S	S
Machine Learning Engineer	S	S	P	P	S
Data Visualization Expert & Technical Writer			S		P

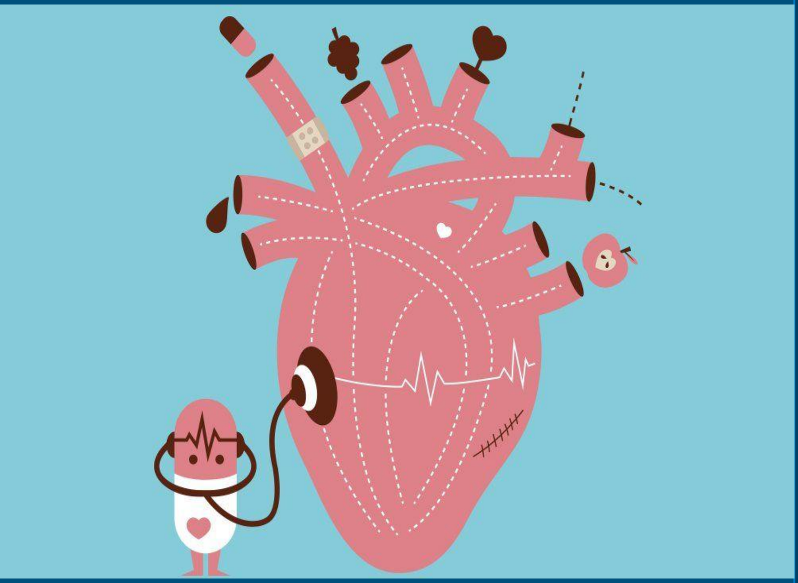
UnitedHealth Group

The future of digital health care holds tremendous potential. UnitedHealth Group, as America's largest health insurance company, is helping advance a consumer-centric, integrated, simple and safe digital health system (ValuePenguin and Price 2021). The company is prioritizing its efforts to improve health outcomes and lower costs; apply scientific best practices in health care to clinical and claims data; and build a digital health care workforce deeply committed to consumer and provider needs.

A 2019 study by Optum, a subsidiary of UnitedHealth Group, found that employers are increasingly embracing digital technology to engage workers in health and well-being programs (UnitedHealth Group 2019). Since 2016, the proportion of employers using health-related mobile apps rose by 46%, with now close to three-quarters of respondents reporting that the apps helped increase employee participation. Also, the number of employers reporting that their employee wellness programs include the use of fitness or activity devices increased by nearly 40% over the same time period, with 71% of employers reporting successful engagement by their employees.

With UnitedHealth Group's $13 billion investment in January 2021 to bolster its software and data analytics capabilities, the company is aggressively seeking opportunities to capitalize on mobile health-related data to lower costs, simplify user experience, and integrate data (Japsen 2021). UnitedHealth Group has partnered with Guardian Labs to tackle this *mobile health data initiative*.

Problem 1: Heart disease is costing the company too much money

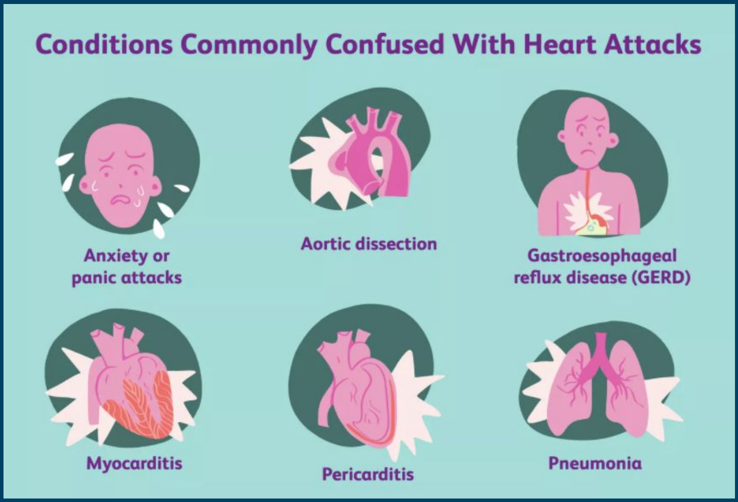


Heart disease is the leading cause of death for men, women, and people of most racial and ethnic groups in the United States (Centers for Disease Control and Prevention, 2020). In fact, one person dies every 36 seconds in the United States from cardiovascular disease. What does this mean for insurance companies? According to Hong Kong-based insurance advising company Pacific Prime, heart disease is the world's leading cause of medical claims, with cancer coming a distant second in some regions (Knapp 2014). Furthermore, cardiovascular patients typically cost more per member than cancer patients. Accordingly, UnitedHealth Group has decided to start with this specific segment of health risks for its *mobile health data initiative*.

Problem 2: Traditional heart disease detection methods lack accuracy and sophistication

Detecting and classifying abnormal heartbeats using a stethoscope has been the most widely used method due to its low cost, simplicity, and non-invasive nature. However, doctors still struggle to detect (early on) abnormal heart sounds that could be indicative of heart disease or abnormalities. It takes years of experience to properly distinguish heartbeat sounds. When doing this manually, there is always a possibility of human error (e.g., fatigue, inexperience, poor technique, too much background noise). Doctors must train their ears to hear the heart while teaching their brains to determine whether the sound is normal.

Although various automatic systems for arrhythmia classification (e.g., ECG signals) exist, UnitedHealth Group requires a more accurate, sophisticated, and faster method to detect heart diseases while also providing a great educational potential for all medical professionals.


Conditions Commonly Confused With Heart Attacks
Anxiety or panic attacks
Aortic dissection
Gastroesophageal reflux disease (GERD)
Myocarditis
Pericarditis
Pneumonia

Problem 3: Data between wearable tech and insurance company apps are not integrated

A fundamental shift has occurred in the fashion tech and medical fields:

- More Americans Are Wearing Fitness Bands: Roughly one-in-five U.S. adults (21%) regularly wear a smartwatch or wearable fitness tracker, according to a Pew Research Center survey conducted in 2019 (Vogels 2020).
- Insurance Companies Are Relying on Online Portals (Apps): An increasing number of insurance companies are using digital platforms to interact with customers, from communicating lab results, connecting patients with doctors, and handling medication refills (UnitedHealth Group 2019).

Despite such technological advances, the two entities—wearable tech and digital/mobile apps—are not talking! Like a nearsighted man stranded on an island with all the books in the world but no glasses, these two popular pieces of tech do not communicate well.

Imagine...

It's late at night, your loved one in his mid 30s has just returned home from a business trip. He checks his Apple Watch for the latest notifications on who just texted him—no rush, he will respond back once he is back in the house. He goes to lift his suitcase from his car to the ground when tragedy strikes. A massive heart attack sets in with zero warning! While he did not die, he will be bound to a wheelchair for the rest of his life.

The tragedy of this story—the root cause was an undetected heart condition (Murmur or Extrasystole), which if found early, could have been treated or managed, ultimately preventing that heart attack.

Guardian Labs to the rescue!

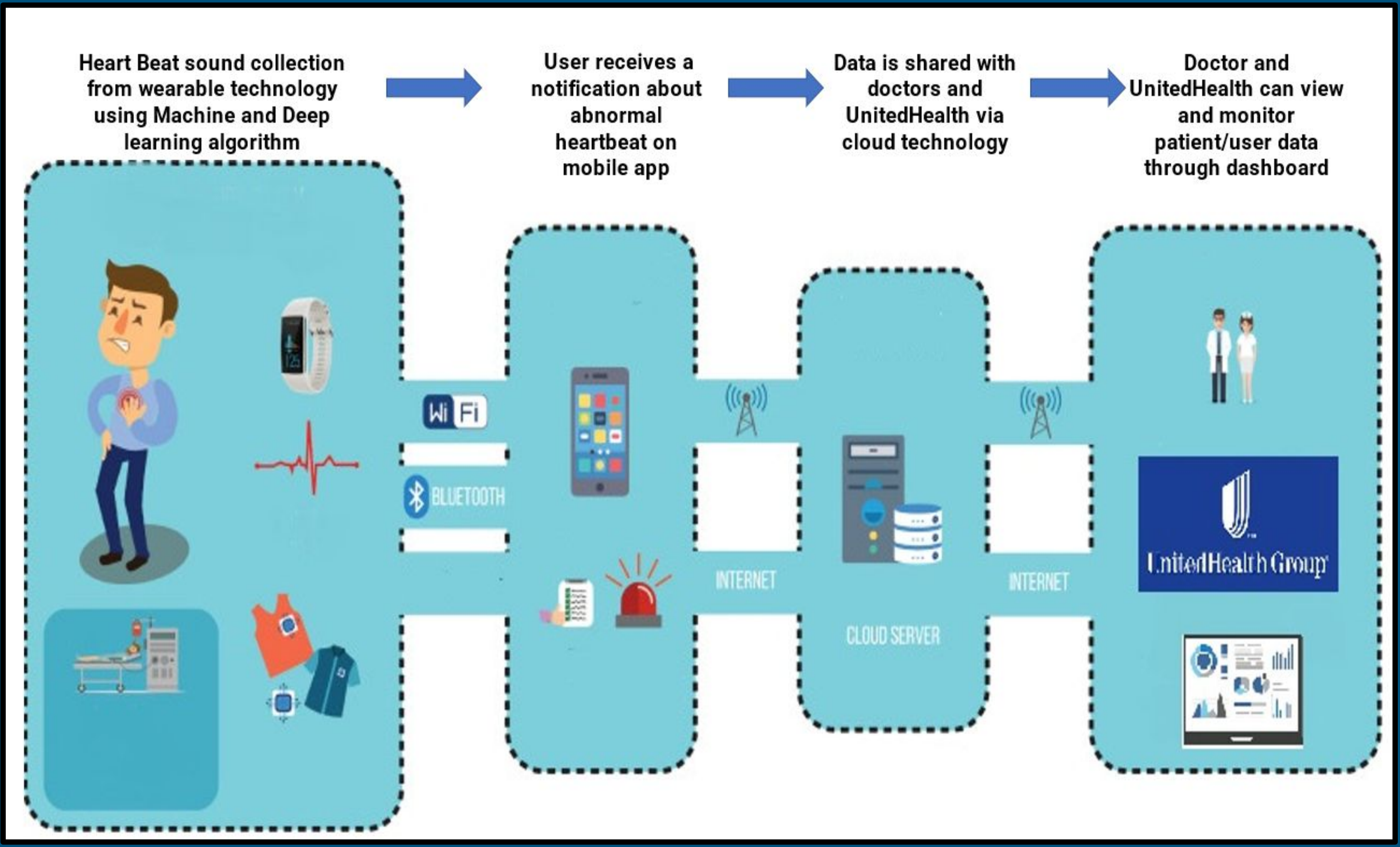
Project Goals, Objectives & Solutions

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.

Whether it's the Apple Watch, Fitbit, or Galaxy Watch, as long as the user opts in, we could integrate the heart beat data collected via the device with other health-related data in the various United Healthcare (UHC) apps (available on both Android and iOS devices).

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


Heart Beat sound collection from wearable technology using Machine and Deep learning algorithm
User receives a notification about abnormal heartbeat on mobile app
Data is shared with doctors and UnitedHealth via cloud technology
Doctor and UnitedHealth can view and monitor patient/user data through dashboard
Wi Fi
BLUETOOTH
INTERNET
CLOUD SERVER
INTERNET
UnitedHealth Group

Project Goals, Objectives & Solutions

	Goal	Deliverable / Success Criteria
01	**Learn to distinguish different heartbeat sounds (e.g., normal vs. abnormal heartbeats)**	<u>Deliverable:</u> • Exploratory data analysis (EDA) of heartbeat sounds • Creating of sound waves charts <u>Success Criteria</u> • Clear understanding of different heartbeats
02	**Create and extract features from audio data**	<u>Deliverable:</u> • Pre-processing of audio data using normalization & sampling techniques <u>Success Criteria</u> • Data ready to use for modeling purposes
03	**Creation of state-of-the-art classifier for the early detection of heart disease**	<u>Deliverable:</u> • Build machine and deep learning models to classify heartbeat sounds • Comparison of modeling approaches and algorithm performance <u>Success Criteria</u> • Data ready to use for modeling purposes heartbeats
04	**Implementation of classification model algorithm**	<u>Deliverable:</u> • Exploratory data analysis (EDA) of heartbeat sounds • Creating of sound waves charts <u>Success Criteria</u> • 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**	<u>Deliverable:</u> • Model deployment and user acceptance testing (UAT) by all team members <u>Success Criteria</u> • Easy usability of dashboard and mobile app by doctors and patients
06	**SUPPLEMENT GOAL: Develop strategy to expand product to wider client base**	<u>Deliverable:</u> • Target at least two new market segments: -new wearable technology (data input) -new client base <u>Success Criteria</u> • 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.

Programming Tools

Guardian Labs decided to use Python as the primary programming language to build the heartbeat classification models. A Python library called Librosa was used to process and load the audio data (.wav files). Python libraries Tensorflow, Keras, Scikit-learn, Numpy, Matplotlib were used to conduct EDA and build/train both traditional machine learning and deep learning models.

We decided to use Python for this project because implementation of Python is easier and more flexible compared to other languages like C, C++ and Java. Python integration with the web is better than R and because the project demands more than just statistics, so the integration of models for dashboarding will be easier on the web. Python also has a massive community which will help our machines learning engineers to benchmark the code and learn new techniques using the appropriate resources available online.

Training models, especially deep learning models, take hours and hours on a local machine to run. Therefore, Guardian Labs has decided to use Google Colaboratory to execute Python code. Google Colaboratory is a free cloud based Jupyter Notebook environment to train and test deep learning models on CPU, GPU and TPU. Since Google Colab gives free 12 hours of continuous runtime our engineers will use GPU, which will give them enough run time and computational power to train the models.

Visualization Tools

For visualization and interactive dashboarding, Guardian Labs will use powerBI to create two sets of dashboards. The first dashboard will be built for patients to view their heartbeat sound classification and their heartbeat history. The second dashboard will be built for doctors to monitor and assess their patients' heartbeat data.

Guardian Lab will use Figma to develop mobile app for user to monitor their heart sounds.

Description of Data

Guardian Labs will use a publicly available heartbeat sound dataset from Kaggle (https://www.kaggle.com/kinguistics/heartbeat-sounds?select=set_a)

The heartbeat data has been gathered from two 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)

The heartbeat dataset is primarily audio-based. All the heartbeat sounds are stored as WAV files that record either normal or abnormal heartbeats. The dataset consists of two subsets (Set A and Set B), which include the audio files and metadata files. Guardian Labs will use both datasets for modeling purposes.

Data is gathered in real-world situations and frequently contains background noise of every conceivable type.According to the data source (Bentley et al., n.d.), some audio has been clipped to reduce excessive noise and provide salient fragment of the sound. The dataset may have been collected from children or adults in calm or excited states; hence, heart rates in the data vary from 40 to 140 beats or higher per minute and are of different lengths.

Understanding the Dataset

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.

1. **Normal Category**
   In the normal category, there are normal, healthy heartbeat sounds. There are background noises in the final seconds of recording,as the device is removed from the body (Bentley et al., n.d.).The temporal description of normal heartbeat sound is below:
   ...lub..........dub................lub..........dub...............lub..........dub...............lub........dub

2. **Murmur Category**
   Heart murmurs sound as though there is a "whooshing, roaring, rumbling, or turbulent fluid" noise in one of two temporal locations: (1) between "lub" and "dub", or (2) between "dub" and "lub". They can be a symptom of many heart disorders, some serious. The asterisk* below shows the locations a murmur may be. (Bentley et al., n.d.)
   ...lub..****...dub............... lub..****..dub .... lub..****..dub .... lub..****..dub ...
   or
   ...lub..........dub...******....lub.......... dub...******....lub .... dub...******....lub ....dub...

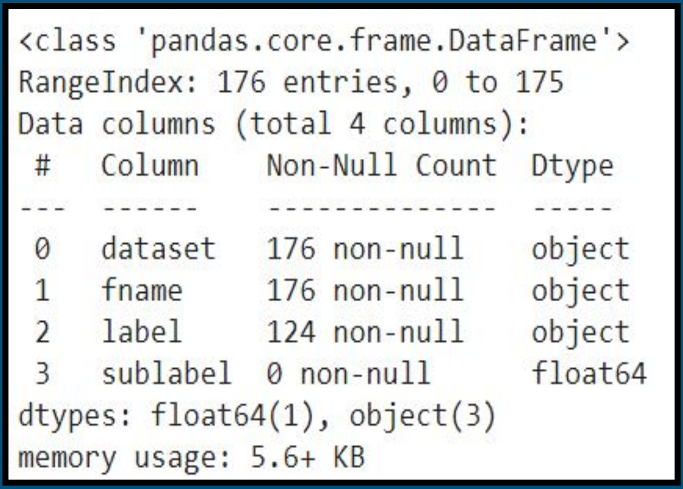
3. **ExtraSystole Category**
   Extrasystole sounds may appear occasionally and can be identified because there is a heart sound that is out of rhythm involving extra or skipped heartbeats, e.g. a "lub-lub dub" or a "lub dub-dub". An extrasystole may not be a sign of disease. Below, is the temporal description of the extra heart sounds:
   ....lub..........dub...................lub.....................dub...............lub.lub..............dub...or
   ...lub..........dub..........................lub.....................dub.dub...............lub..............dub

4. **Artifacts Category**
   In the Artifact category, there were a wide range of different sounds, including echoes, speech, music and noise. There is usually no discernable heart sound (Bentley et al., n.d.)

Data Structure

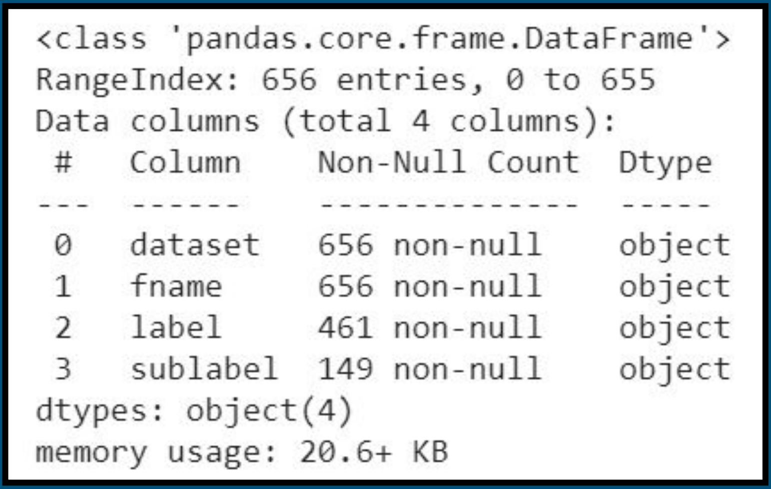
There are two primary datasets: A and B. The statistic for both datasets is shown in figures 2 and 3. Dataset A contains 176 audio files and Dataset B contains 656 audio files. The column “fname” in both sets contains the name of the audio file that serves as a unique key in the folder containing the audio files. The “label” column defines the heartbeat classification i.e, whether a heart beat is normal, murmur, extrasystole or artifact. The sublabel defines sub directory i.e. noisy normal and noisy murmur.

Figure 2. *Dataset A*



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 176 entries, 0 to 175
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   dataset   176 non-null    object
 1   fname     176 non-null    object
 2   label     124 non-null    object
 3   sublabel  0 non-null      float64
dtypes: float64(1), object(3)
memory usage: 5.6+ KB
```

Figure 3. *Dataset B*



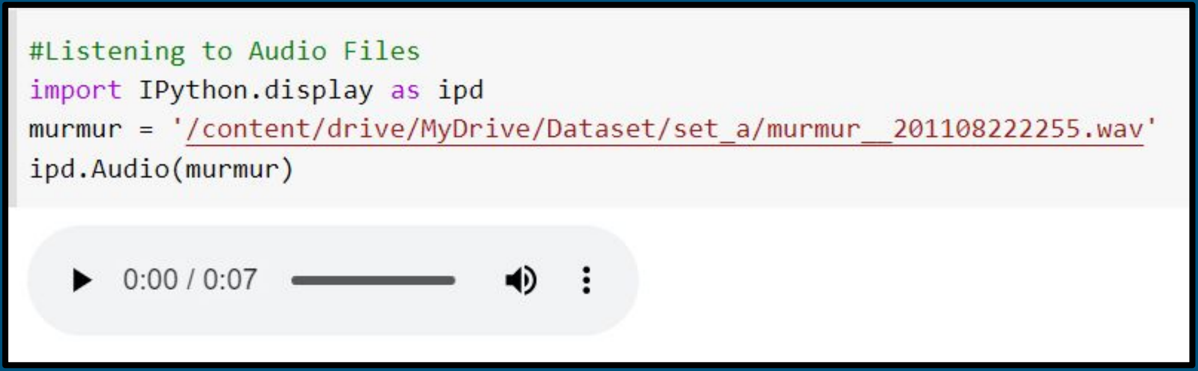
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 656 entries, 0 to 655
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   dataset   656 non-null    object
 1   fname     656 non-null    object
 2   label     461 non-null    object
 3   sublabel  149 non-null    object
dtypes: object(4)
memory usage: 20.6+ KB
```

Exploratory Data Analysis (EDA)

Extensive EDA was conducting to understand the data and the structure of the audio files. We used various graphs to identify key distinctions between the classes and their respective sound features.

Auditory Inspection

In order to inspect the data, we listened to audio files of each heartbeat class using the iPython library. The `IPython.display` package allowed us to play audio directly on Jupyter notebook.

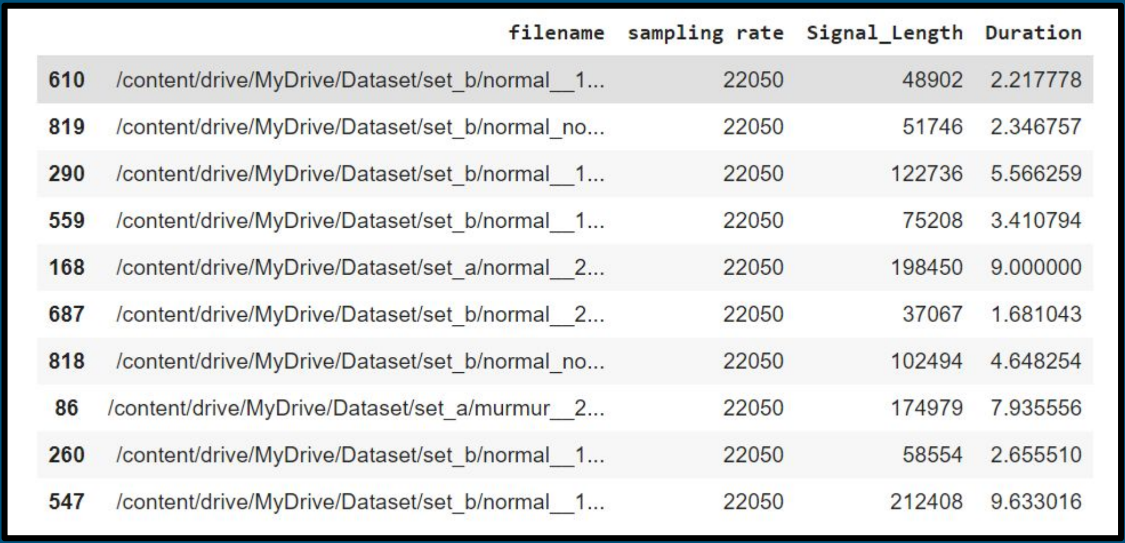


```
#Listening to Audio Files
import IPython.display as ipd
murmur = '/content/drive/MyDrive/Dataset/set_a/murmur__201108222255.wav'
ipd.Audio(murmur)
```

Sampling Rate

Sampling Rate indicates how many times (in one second) the data is collected (i.e., Hertz, abbreviated as Hz). Figure 4 shows that the heartbeat dataset was collected at 22050 Hz, which means each sample is taken 22050 times per second. Since all the files have uniform sampling rate, we did not have to apply any sampling conversion techniques.

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

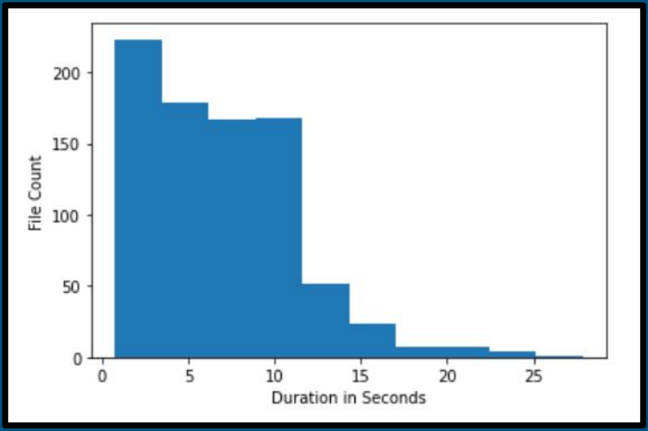


	filename	sampling rate	Signal_Length	Duration
610	/content/drive/MyDrive/Dataset/set_b/normal__1...	22050	48902	2.217778
819	/content/drive/MyDrive/Dataset/set_b/normal_no...	22050	51746	2.346757
290	/content/drive/MyDrive/Dataset/set_b/normal__1...	22050	122736	5.566259
559	/content/drive/MyDrive/Dataset/set_b/normal__1...	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

Audio Length

The histogram (figure 5) indicates that the audio lengths in the dataset vary between 1 and 30 seconds. Most information about the heartbeat sound is contained in the low-frequency components; the higher frequencies typically only contain noise.

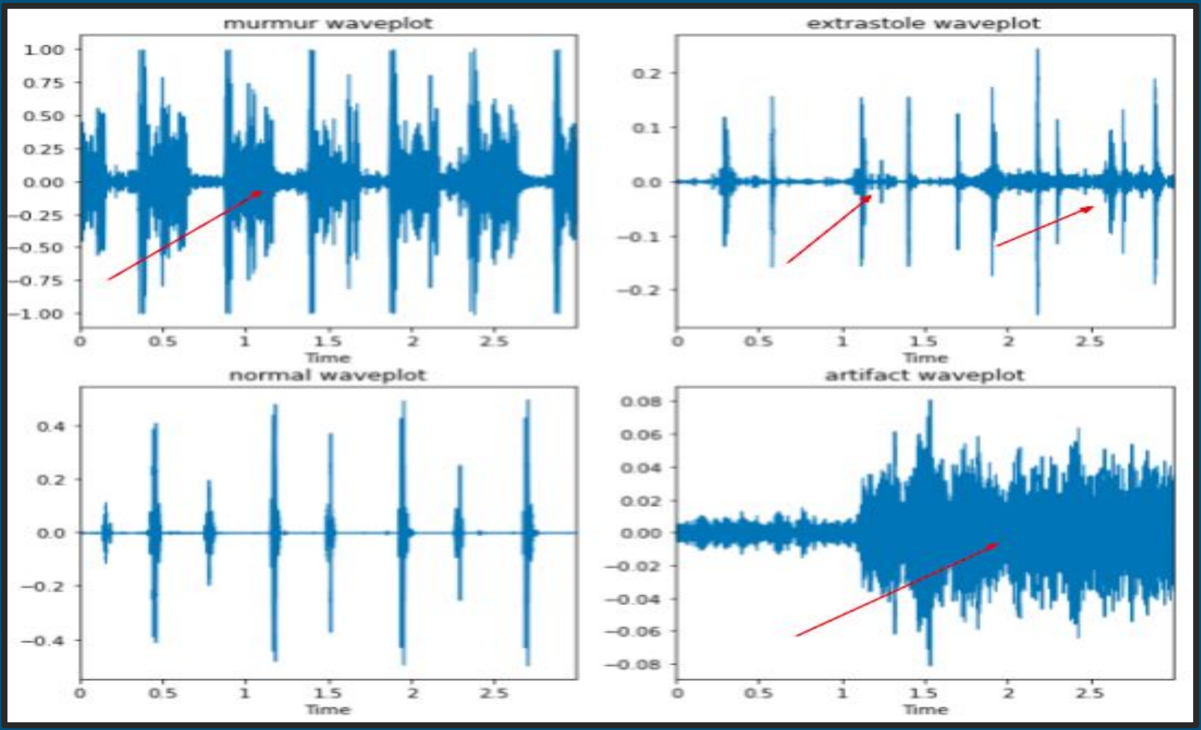
Figure 5. *Duration of audio files*



Wave Plots

The `librosa.display.waveplot` function was used to display waveform visualization of the amplitude vs the time representation of the signal (see figure 6). Waveform plots indicate how loud an audio is at a given time. The plots effectively visualize the difference between the four different heartbeat categories. The murmur category usually contains an echo or whooshing sound in between lub dub cycles. For the extrasystole category, there is an extra heartbeat sound between lub dub cycles. The artifact plots were easily distinguished as well because of the high frequency of noise in this category.

Figure 6. *Waveplot of heartbeat classes: red arrows highlight the distinct frequency patterns between each heartbeat class*



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.

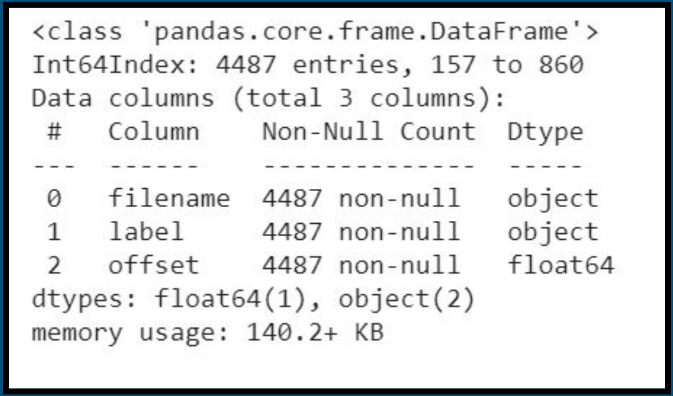
Audio File Loading

We applied various preprocessing steps on the combined dataset to convert audio signals into number arrays. We used the `librosa.load()` function in Python's Librosa library to load audio files.

Combining Datasets A & B

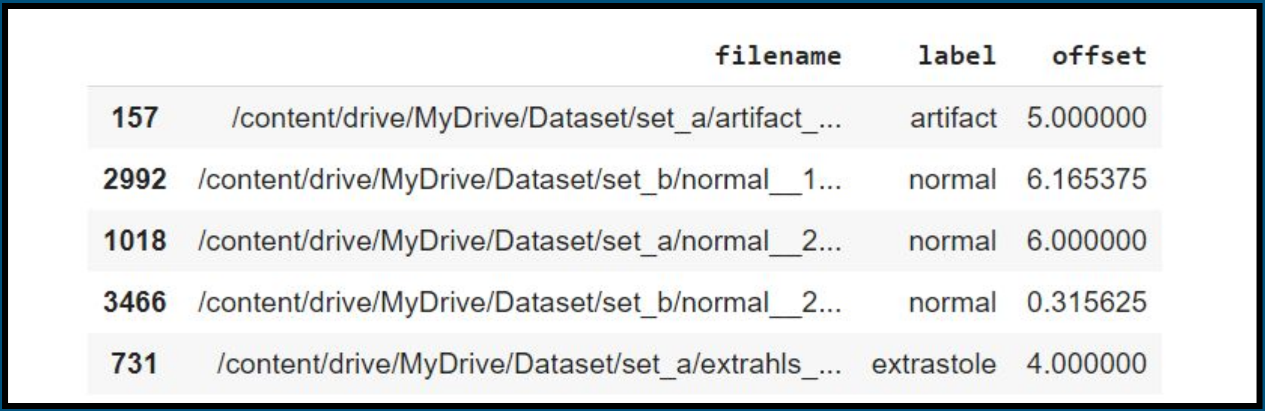
The two datasets were combined into one to facilitate the model training process. Figures 7 and 8 show a summary of this combined dataset, which has a total of 4487 audio files after data processing i.e. resizing etc

Figure 7. *Combined Dataset*



```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4487 entries, 157 to 860
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   filename  4487 non-null   object
 1   label     4487 non-null   object
 2   offset    4487 non-null   float64
dtypes: float64(1), object(2)
memory usage: 140.2+ KB
```

Figure 8. *Combined Dataset*



	filename	label	offset
157	/content/drive/MyDrive/Dataset/set_a/artifact_...	artifact	5.000000
2992	/content/drive/MyDrive/Dataset/set_b/normal__1...	normal	6.165375
1018	/content/drive/MyDrive/Dataset/set_a/normal__2...	normal	6.000000
3466	/content/drive/MyDrive/Dataset/set_b/normal__2...	normal	0.315625
731	/content/drive/MyDrive/Dataset/set_a/extrahls_...	extrastole	4.000000

Data Cleaning & Missing Labels

We decided to drop the “Sublabel" column from the dataset because it did not contain valuable information. i.e substantial amount of background noise and distortion. We combined ‘Extrahls’ with the extrasystole category because of few records of Extrahls and also both involve extra heartbeat.

Figure 9 shows the total number of records in each labeled category. There were a total of 247 audio files that had missing labels. Guardian Labs received assistance from Dr. Muhammad Ali Shahid from Henry Ford Medical Hospital to manually label the 247 records by listening to each audio file. We then used the *Librosa* library to load the audio files to listen to the heartbeats and plot the time amplitude graph of each unlabeled file. This helped us verify the label that Dr Muhammad Ali Shahid identified. Figure 10 shows the total number of records after labeling the missing data.

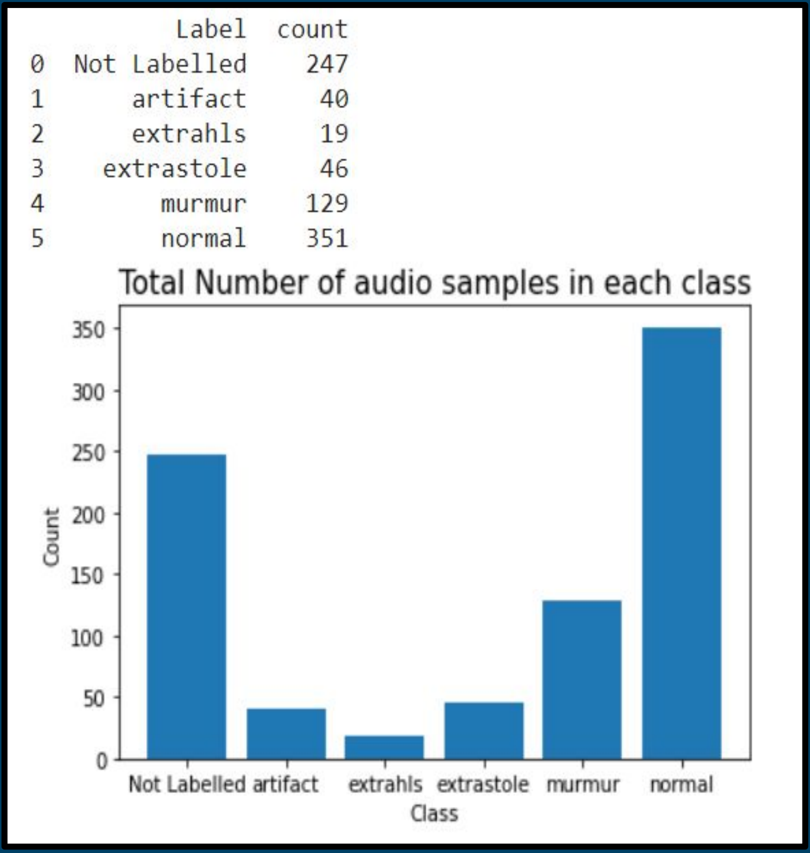
Reducing File Load Time

Python’s Librosa library contains optional parameters to reduce the time it takes to load audio files. We decided to use the ‘kaiser_fast’ parameter, which performed faster than the ‘scipy’ parameter.

Normalization

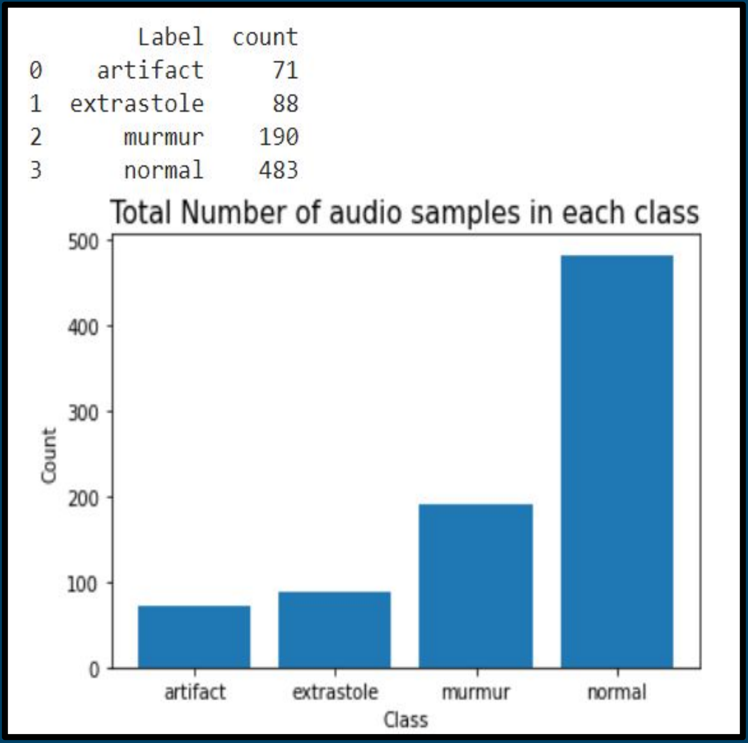
Since the heart sound files were recorded manually using the digital stethoscope, the amplitudes for each file were different. Thus, the audio files were normalized using the `librosa.util.normalize()` function so the values will range from -1 to a maximum amplitude of 1.

Figure 9. *Before labelling missing data*



```
         Label  count
0  Not Labelled    247
1      artifact     40
2      extrahls     19
3    extrastole     46
4        murmur    129
5        normal    351
```

Figure 10. *After labelling missing data*

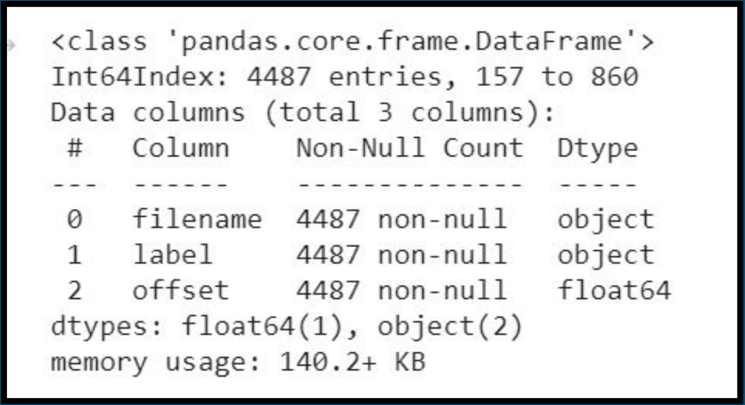


```
        Label  count
0    artifact     71
1  extrastole     88
2      murmur    190
3      normal    483
```

Resizing

All the audio files that have duration of fewer than 2 seconds were cut down because they do not contain enough data points to classify the heart beats. Those files do not capture a full heartbeat cycle, making it difficult to train whether or not those samples contain heart sound irregularities or not. Since the audio files were of different lengths, all of them were converted to fixed length prior to training. The files were cut down into a fixed length segment of 2 seconds. This approach enabled us to expand the dataset while retaining key information. Figure 11 shows the number of datapoints after resizing.

Figure 11. *Total number of records after resizing*

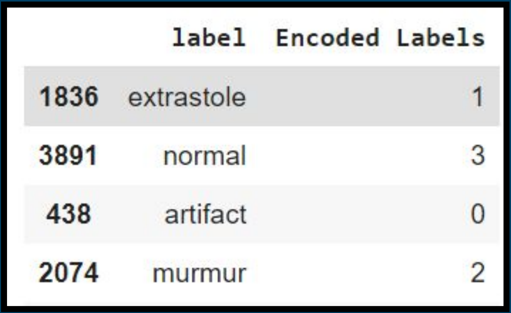


```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4487 entries, 157 to 860
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   filename  4487 non-null   object
 1   label     4487 non-null   object
 2   offset    4487 non-null   float64
dtypes: float64(1), object(2)
memory usage: 140.2+ KB
```

Converting Labels, Splitting Dataset

All the labels were encoded using `Sklearn.preprocessing.LabelEncode` (see figure 12) to convert from categorical text data into a model understandable numerical form.

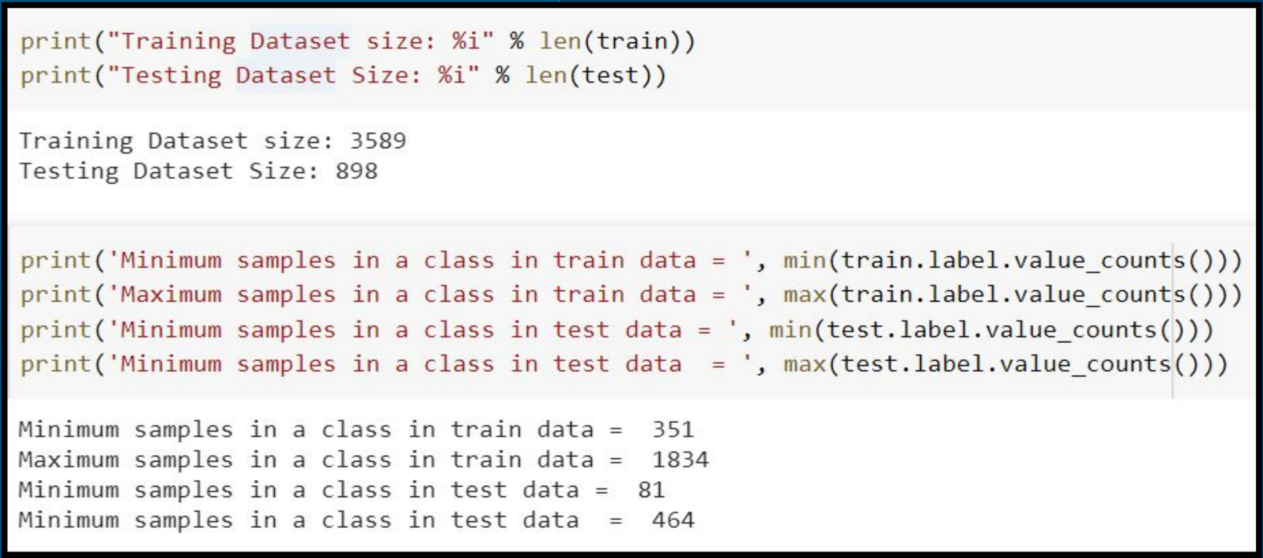
Figure 12. *Encoded Labels*



	label	Encoded Labels
1836	extrastole	1
3891	normal	3
438	artifact	0
2074	murmur	2

After performing the data processing steps and feature extraction steps (see the **Feature Extraction** section below), `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. Training dataset consists of **3,589** records and the testing dataset consists of **898** records. Figure 13 shows the maximum and minimum samples in a class in the train and test sets.

Figure 13. *Training and Testing Dataset Split*



```
print("Training Dataset size: %i" % len(train))
print("Testing Dataset Size: %i" % len(test))
```

```
Training Dataset size: 3589
Testing Dataset Size: 898
```

```
print('Minimum samples in a class in train data = ', min(train.label.value_counts()))
print('Maximum samples in a class in train data = ', max(train.label.value_counts()))
print('Minimum samples in a class in test data = ', min(test.label.value_counts()))
print('Minimum samples in a class in test data  = ', max(test.label.value_counts()))
```

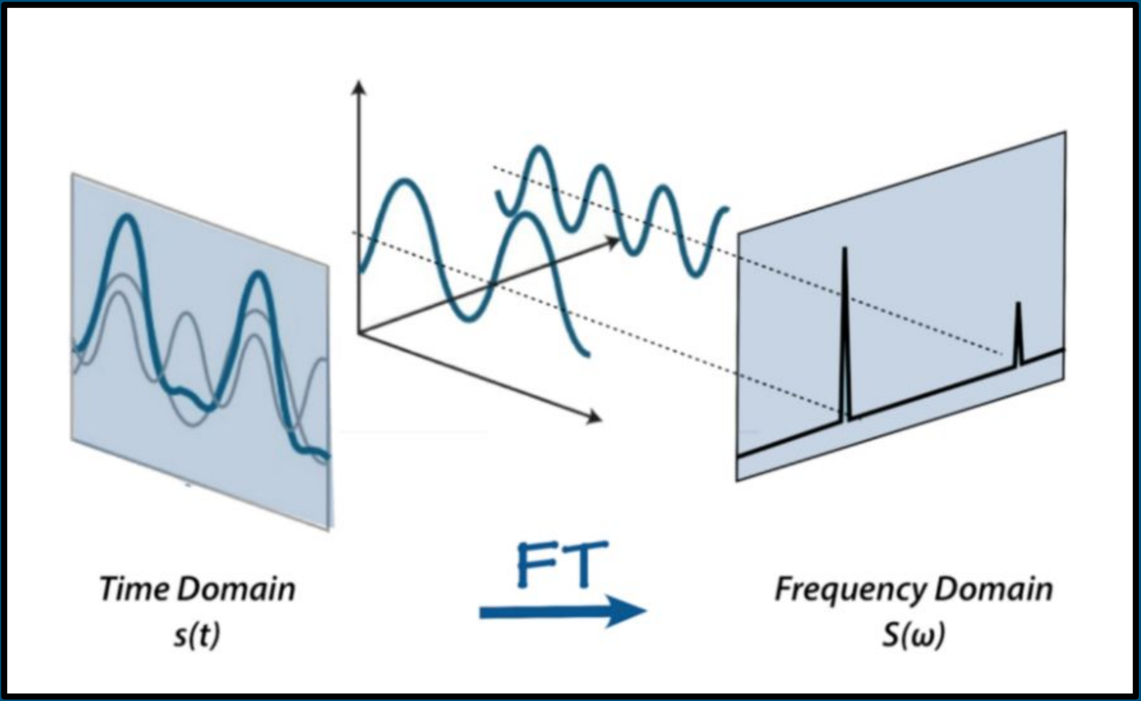
```
Minimum samples in a class in train data =  351
Maximum samples in a class in train data =  1834
Minimum samples in a class in test data =  81
Minimum samples in a class in test data  =  464
```

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. An audio signal is composed of several single-frequency sound waves. When taking samples of the signal over time, we only capture the resulting amplitudes. 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 14). The result is called a spectrum [3]

Since our raw audio files were in the .wav format, we transformed the one-dimensional time series signals into two-dimensional heat maps (amplitude vs time) that show the time frequency distribution of the signal. Since, the data was collected manually and there were many different amplitudes of audio signals in each file, converting the data to the frequency domain was more accurate. Accordingly, we used the **Mel-Frequency Cepstral Coefficient (MFCC)** and **Mel Spectrogram** as our feature extraction methods.

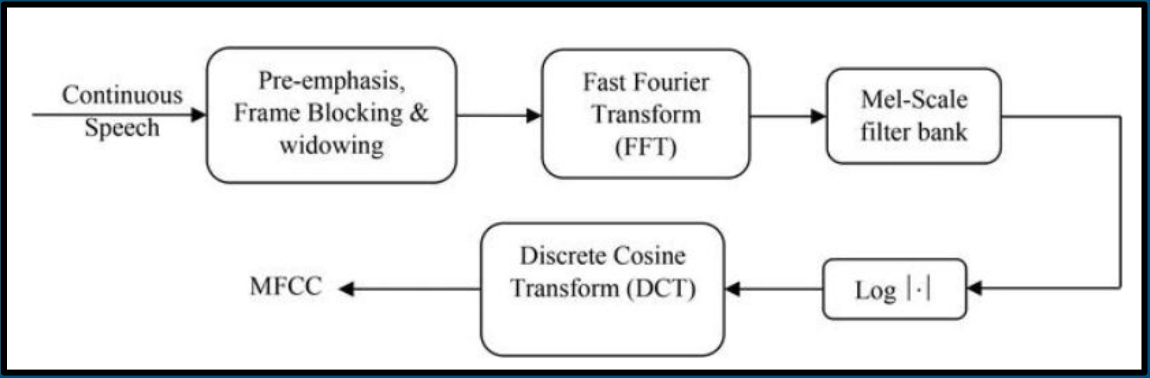
Figure 14. *Visualization of the Fourier transform*

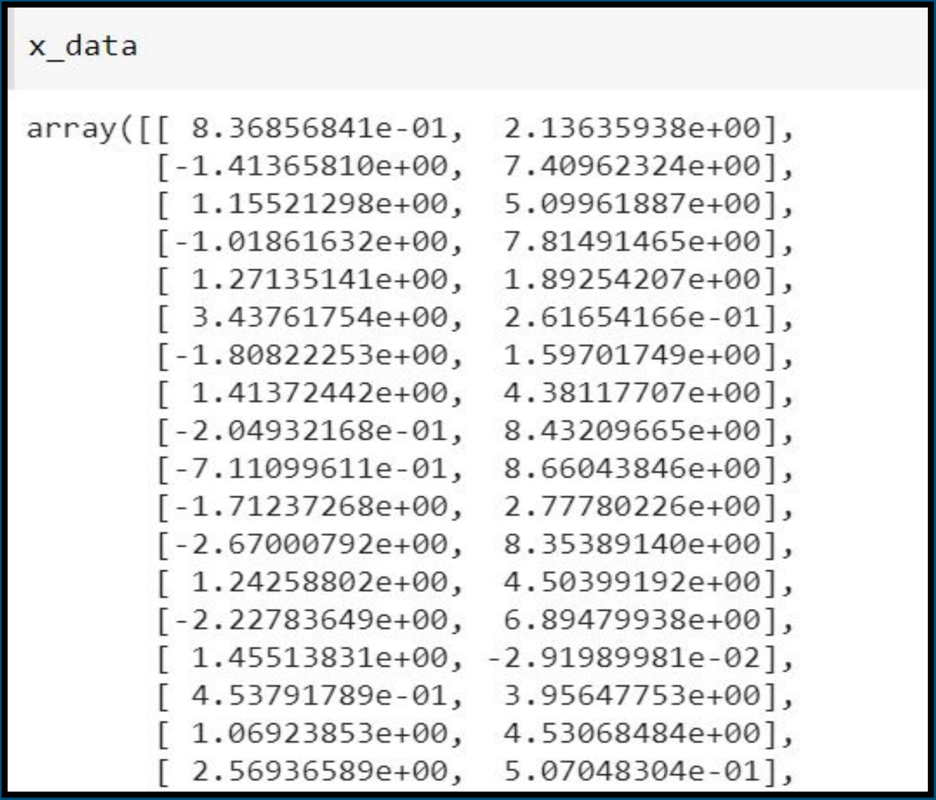


Mel-Frequency Cepstral Coefficient (MFCC)

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). MEL scale is based on the way humans distinguish between frequencies which makes it very convenient to process sounds or in simple words a "representation" of the vocal tract.
After extracting the MFCC's feature, the audio data was converted into an array of discrete numbers for machine learning models. Figure 15 shows the snapshot of data.

Figure 15. MFCC Extraction Process and Data after feature extraction


Continuous Speech
Pre-emphasis, Frame Blocking & widowing
Fast Fourier Transform (FFT)
Mel-Scale filter bank
Log |·|
Discrete Cosine Transform (DCT)
MFCC



```
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],
```

Mel Spectrogram

Deep learning models rarely take raw audio directly as input. The common practice is to convert the audio into a 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. The brighter the color the higher the energy of the signal.

A mel spectrogram is a spectrogram where the frequencies are converted to the mel scale. The following steps are involved in the conversion:

- Mapped the audio signal from the time domain to the frequency domain using the **fast Fourier transform**, and performed on overlapping segments of the audio signal.
- Convert the y-axis (frequency) to a log scale and the color dimension (amplitude) to decibels to form the spectrogram.
- Mapped the y-axis (frequency) onto the mel scale to form the mel spectrogram.

The Figure 16 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).

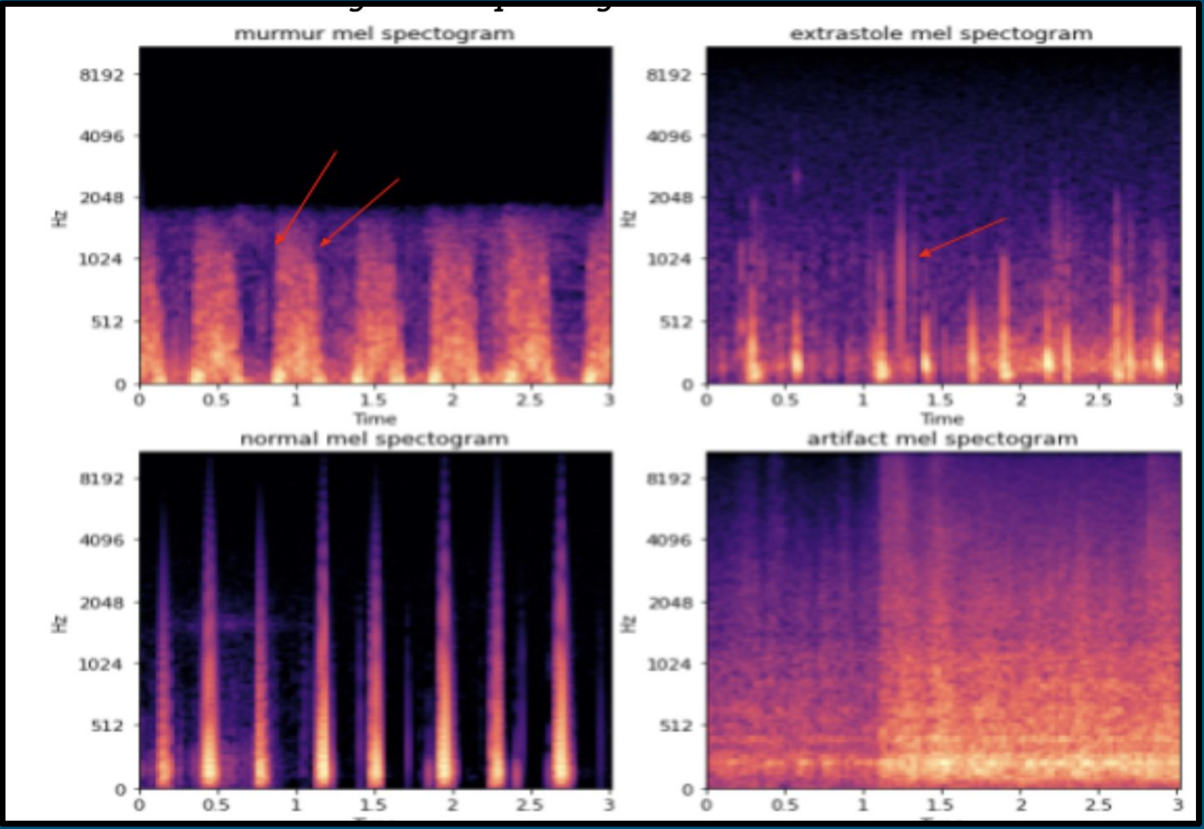
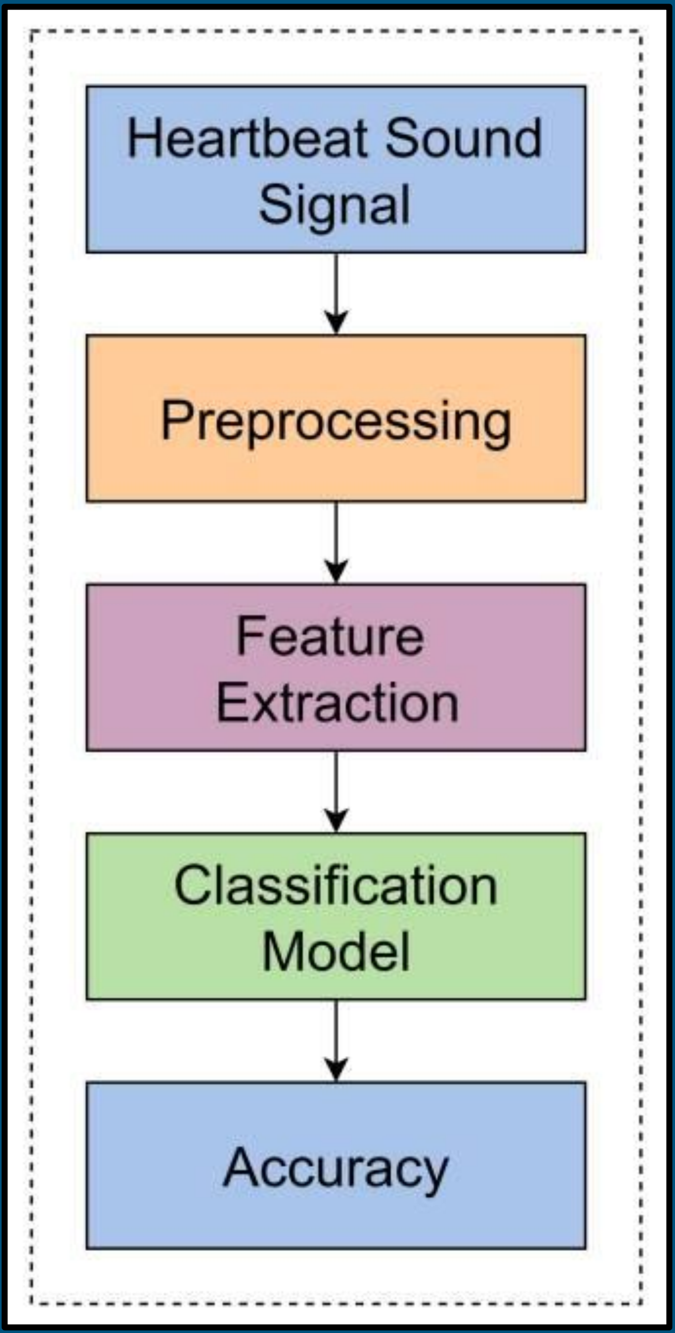


Figure 16. *Spectrograms of Heartbeats*

Modeling Process

The model building consists of three steps: pre-processing, feature extraction and the training the classification models. Figure 17 shows the process Guardian Labs took to create heartbeat classification models

Figure 17. *Guardian Labs Modeling Process*



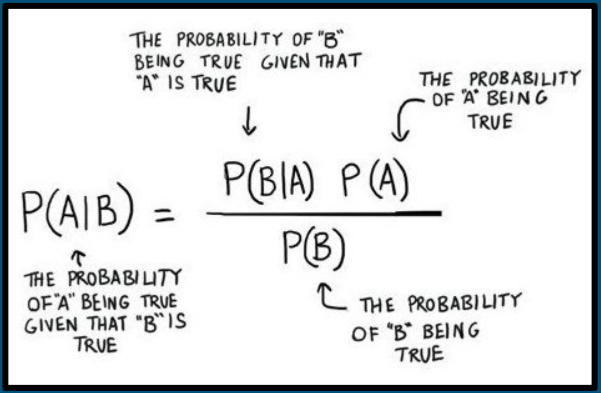
Overview of Machine Learning Models

Different classification algorithms were used in this study. These machine learning approaches include the Naive Bayesian Classifier, Support Vector Machines (SVM), K Nearest Neighbor (KNN), Random Forest and Gradient Boosted Trees. We decided to use the Machine Learning approaches as they each had their own strengths and weaknesses in classification problems. Each of them are explained below:

Naive Bayesian Classifier

Naive Bayesian is a collection of algorithms that is related to Bayes' Theorem formula shown in figure 18. Each feature is independent of the other features in the dataset and each feature contributes to a class. It uses probabilities to figure out the class of the data.

Figure 18. *Bayes' Theorem*


THE PROBABILITY OF "B" BEING TRUE GIVEN THAT "A" IS TRUE
THE PROBABILITY OF "A" BEING TRUE
P(A|B) = P(B|A) P(A) / P(B)
THE PROBABILITY OF "A" BEING TRUE GIVEN THAT "B" IS TRUE
THE PROBABILITY OF "B" BEING TRUE

Support Vector Machines (SVM)

SVMs are another algorithm that consists of a hyperplane that tries to find the best way to classify/divide the data. The goal of the hyperplane is to choose a big distance/margin that correctly classifies the data points.

K-Nearest Neighbor (KNN)

KNN is an algorithm that relies on a training set to classify new data. It groups data points together and tries to determine by distance which class a new data point belongs to. It does this by the value of K which signifies nearest neighbor. If the majority class wins in new data points space given by K then the new data point belongs to that class.

Random Forest

Random Forests is an algorithm that consists of a collection of decision trees. The decision trees take a random subset of the data and try to make rules on those data points. Once it has made rules on the training set, then all the decision trees vote on the class of a new data point. The new data point will belong to the majority voted class. we used n= 500 as estimators in our experiment.

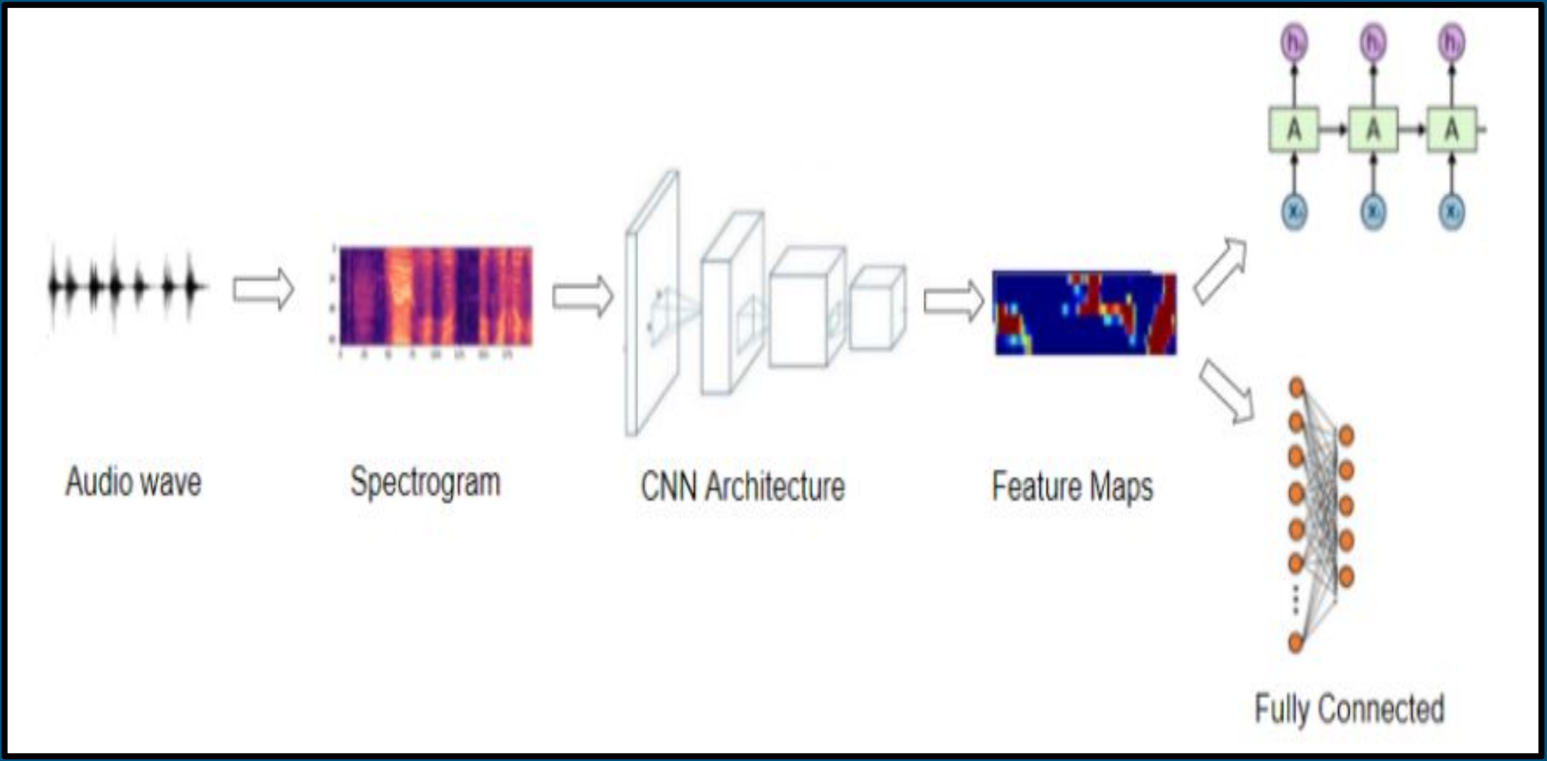
Gradient Boosted Trees

Gradient Boosted Trees uses Trees that are shallow in depth meaning that they have less leaves to train on. This model trains by first training one decision tree and then improving the loss by adding another decision tree to a group

Overview of Deep Learning Models

Various deep learning models were built by adjusting the hyperparameters (e.g., number of neurons, number of hidden layers, dropout regularization, and batch normalization). All the deep learning models used melspectogram for training the data and classification of the heartbeat sounds. The modeling process started with raw audio in the form of .wav files. Then they were converted from audio data into mel spectrogram as described in the feature extraction section above. After converting into image data, we used standard neural network architectures (as specified in the following section) to process the data and extract features. Then we generated the output predictions from this encoded representation by passing through a classifier consisting of fully connected layers (see figure 19).

Figure 19. Deep Learning Model Architecture



Deep Neural Network (DNN)

Deep Neural Network (DNN) - Type of Neural Network which is a Feed Forward Network meaning they only go in one direction and do not go backwards when computing data. The DNN model used to classify the heartbeat consists of 2 layers - input layer with 12 neurons, and output layer. **Relu** was used as the activation function for dense layers. The activation for output layer used was “**Softmax**.”

Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNN) - Expects images of inputs. Consists of a Convolutional Layer which has filters which activate to a specific part of the image. There are also pooling layers whose job is to reduce the number of parameters and computation in the network. The CNN model used to classify heartbeats consists of a sequential model with simple architecture. It consists of 3 Convolutional Layers with 16/32/64 filters, 4 Max Pooling Layers, 1 Global Average Pooling Layer, 2 Dropout Layers of 50 percent and one output layer. **Relu** was used as the activation function for Convolutional Layers. The activation for output layer used was “**Softmax**

Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) - Different from DNN in that it has internal memory. All the inputs are related to each other in this neural network unlike DNN. Problems with this neural network is that there is a Gradient vanishing and exploding problems making it hard for models to learn.

Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) - Resolves the vanishing gradient problem that RNN has. Has three gates mainly input, forget, and output gates. Uses Time Series Data, and also does backpropagation. The LSTM/RNN model consists of 2 LSTM layers with 64 and 32 neurons respectively and one output layer. **Relu** was used as the activation function and Softmax for the output layer.

Precision

Precision is one of the measures Guardian Labs used to calculate the performance of the machine learning models. Precision estimates the predictive value of a label, the higher the precision, the more likely the prediction is correct. This means that precision can be used to assess the predictive power of the machine learning model [sok]. The equation in figure 20 shows the formula for precision score. A prediction is considered true positive (TP) if the predicted class matches the target class. In the other case, false positive (FP) is if a predicted class is not the same as the target class. In a confusion matrix with rows being the target class and columns being the predicted class.

Figure 20. *Precision Score Equation*

$$\text{Precision} = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Positive}}$$

F-Score and Recall

F-score is the other measure used. F-score is a composite measure that challenges algorithms with higher recall and measures a classification model's usefulness, which is calculated by the equation shown in figure 21. In order to calculate F-score, recall is needed in addition to precision. Recall approximates the probability of a positive label being true and is shown in figure 22. This measure assesses the effectiveness of the models used.

Figure 21. *F-score Equation*

$$Fscore = \frac{(\beta^2 + 1) * precision * recall}{\beta * precision + recall}$$

Figure 22. *Recall Score Equation*

$$\text{Recall} = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Negative}}$$

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.

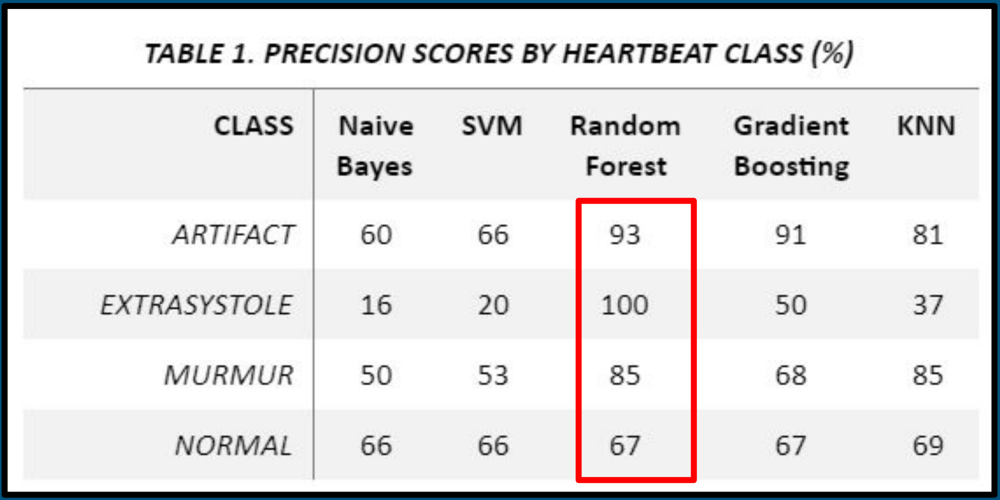


TABLE 1. PRECISION SCORES BY HEARTBEAT CLASS (%)

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

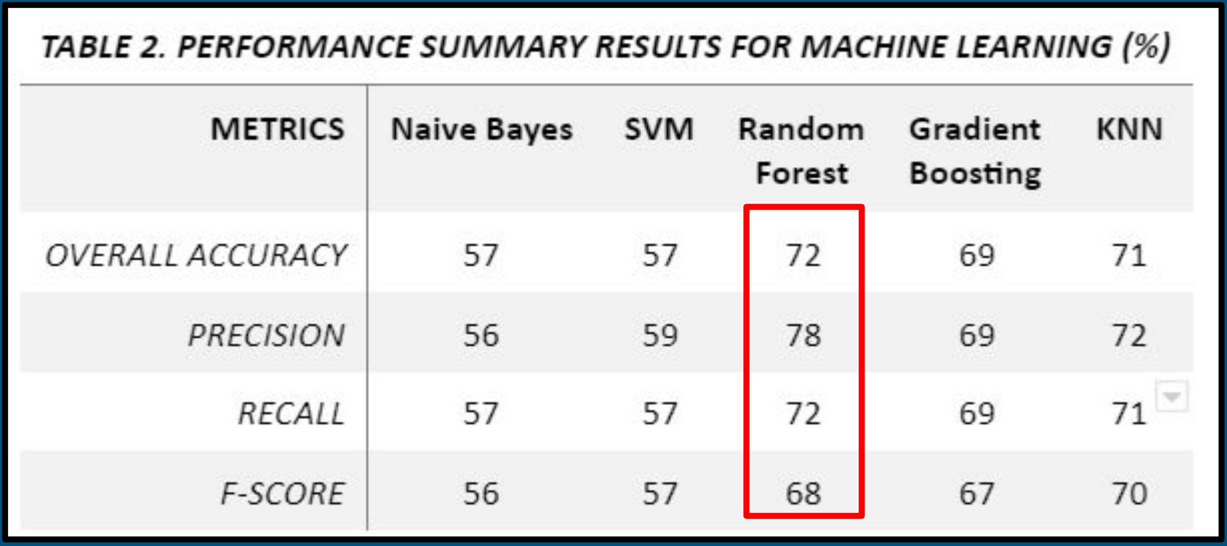


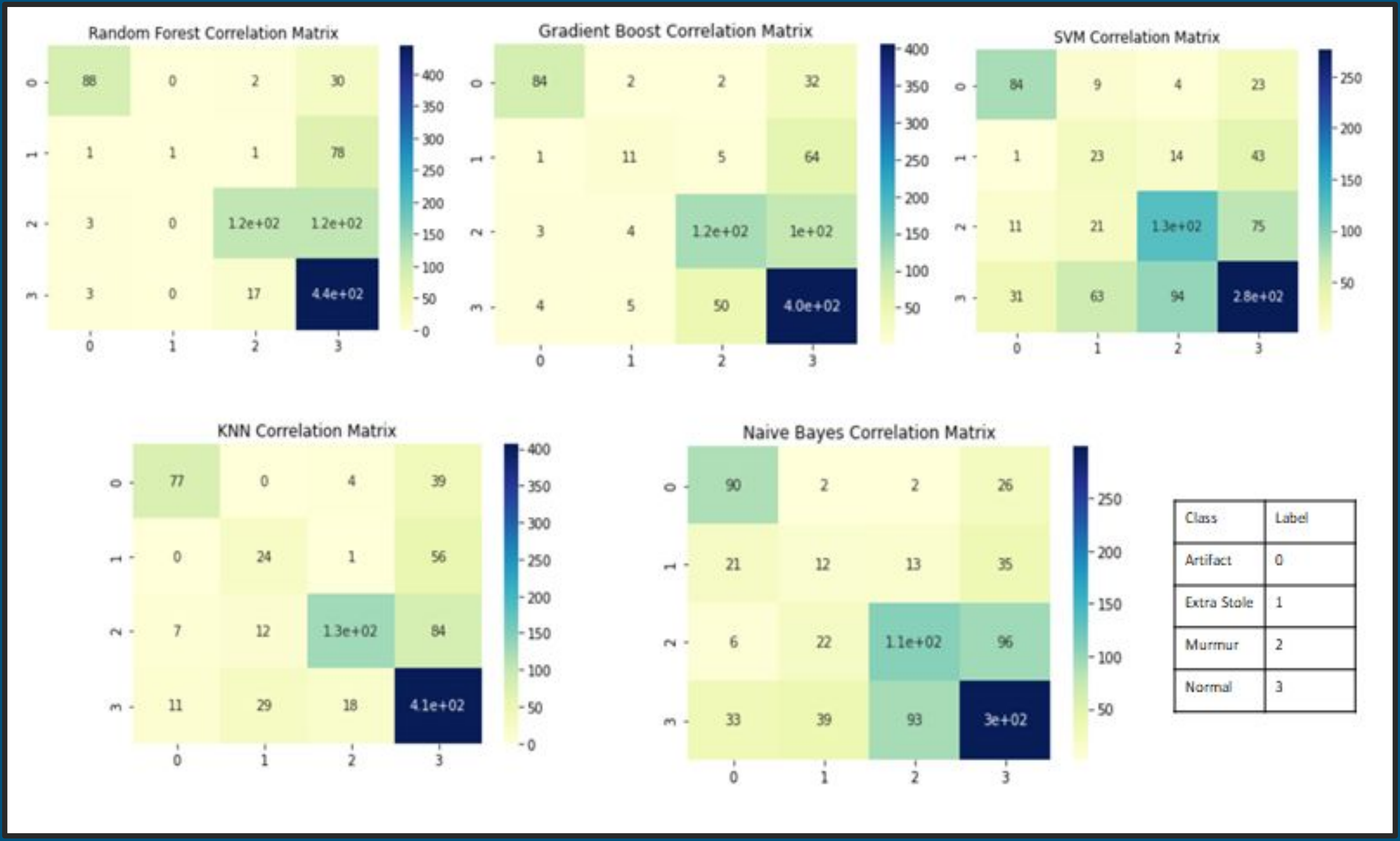
TABLE 2. PERFORMANCE SUMMARY RESULTS FOR MACHINE LEARNING (%)

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

Machine Learning Performance

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



Class	Label
Artifact	0
Extra Stole	1
Murmur	2
Normal	3

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.

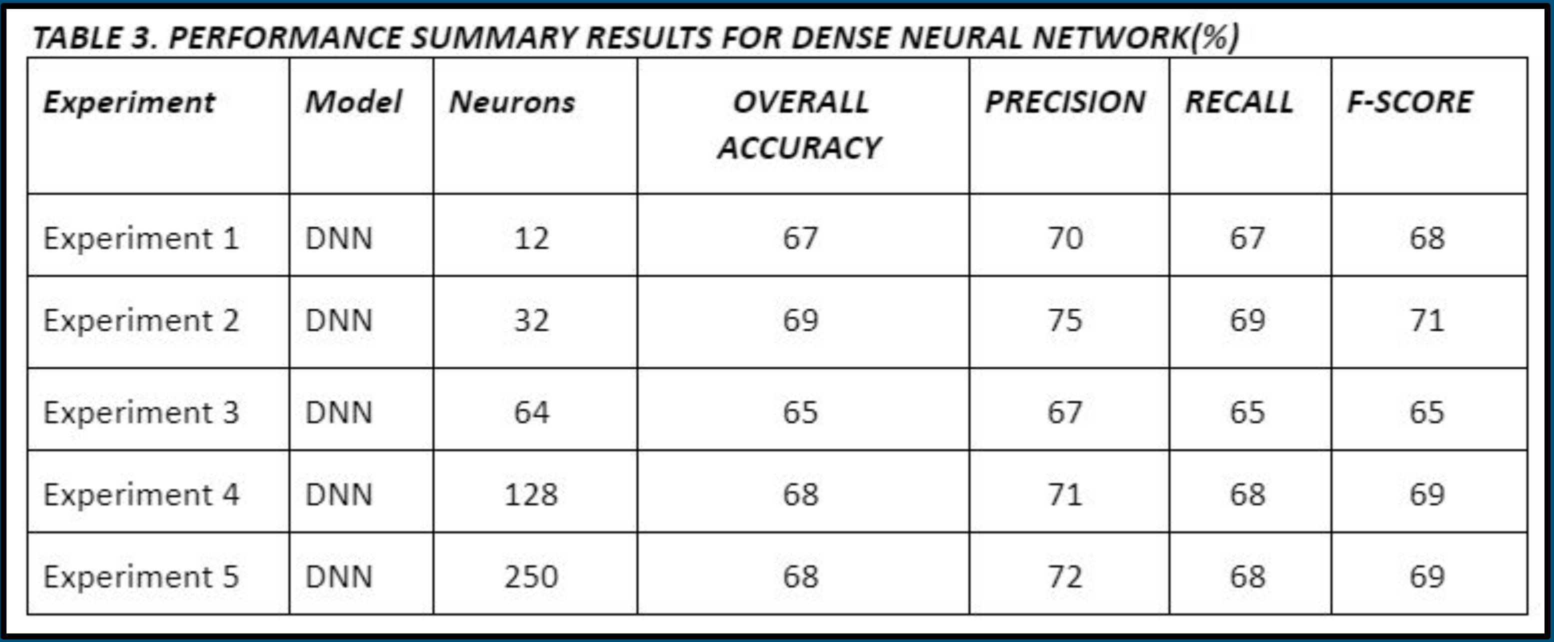


TABLE 3. PERFORMANCE SUMMARY RESULTS FOR DENSE NEURAL NETWORK(%)

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

Deep Learning Performance

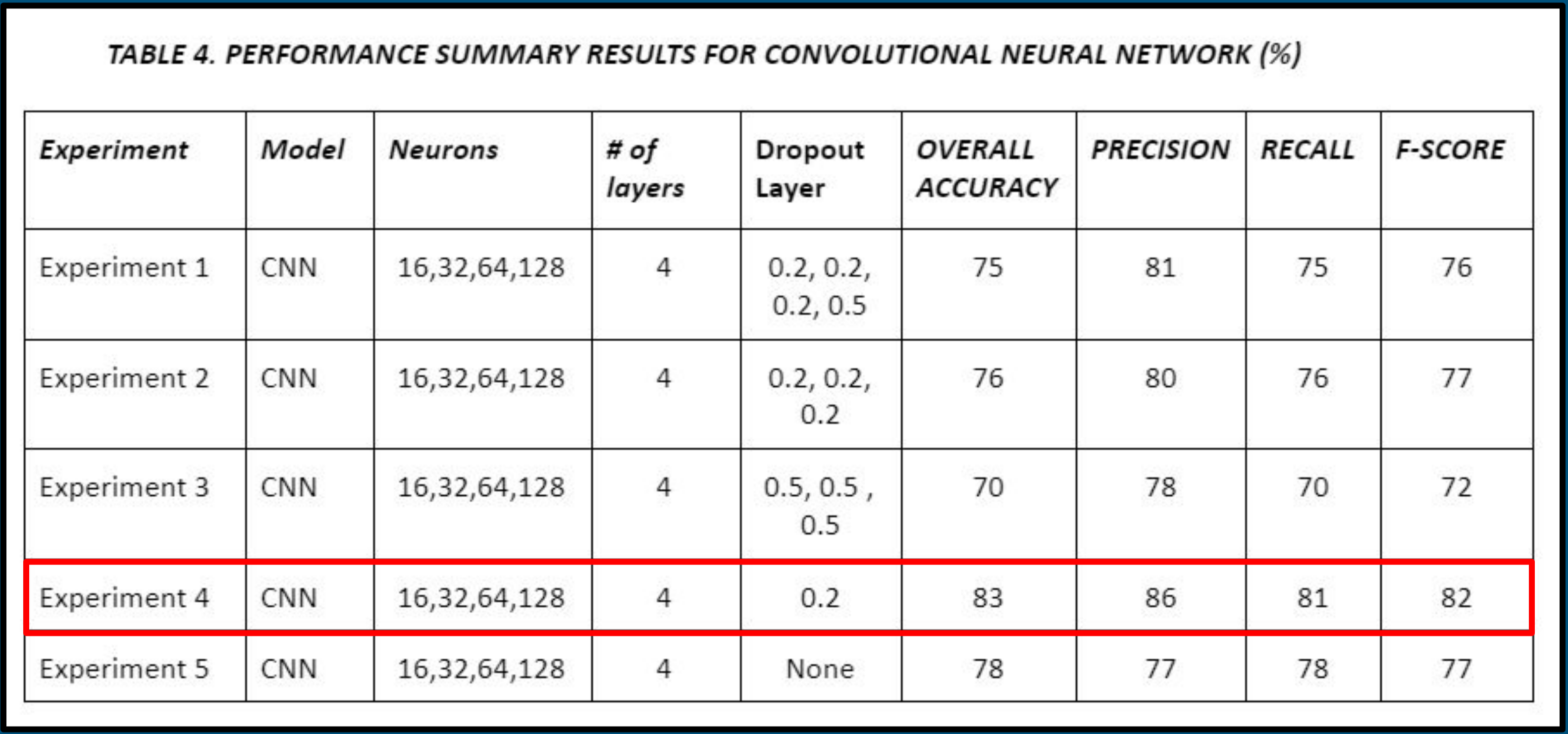


TABLE 4. PERFORMANCE SUMMARY RESULTS FOR CONVOLUTIONAL NEURAL NETWORK (%)

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

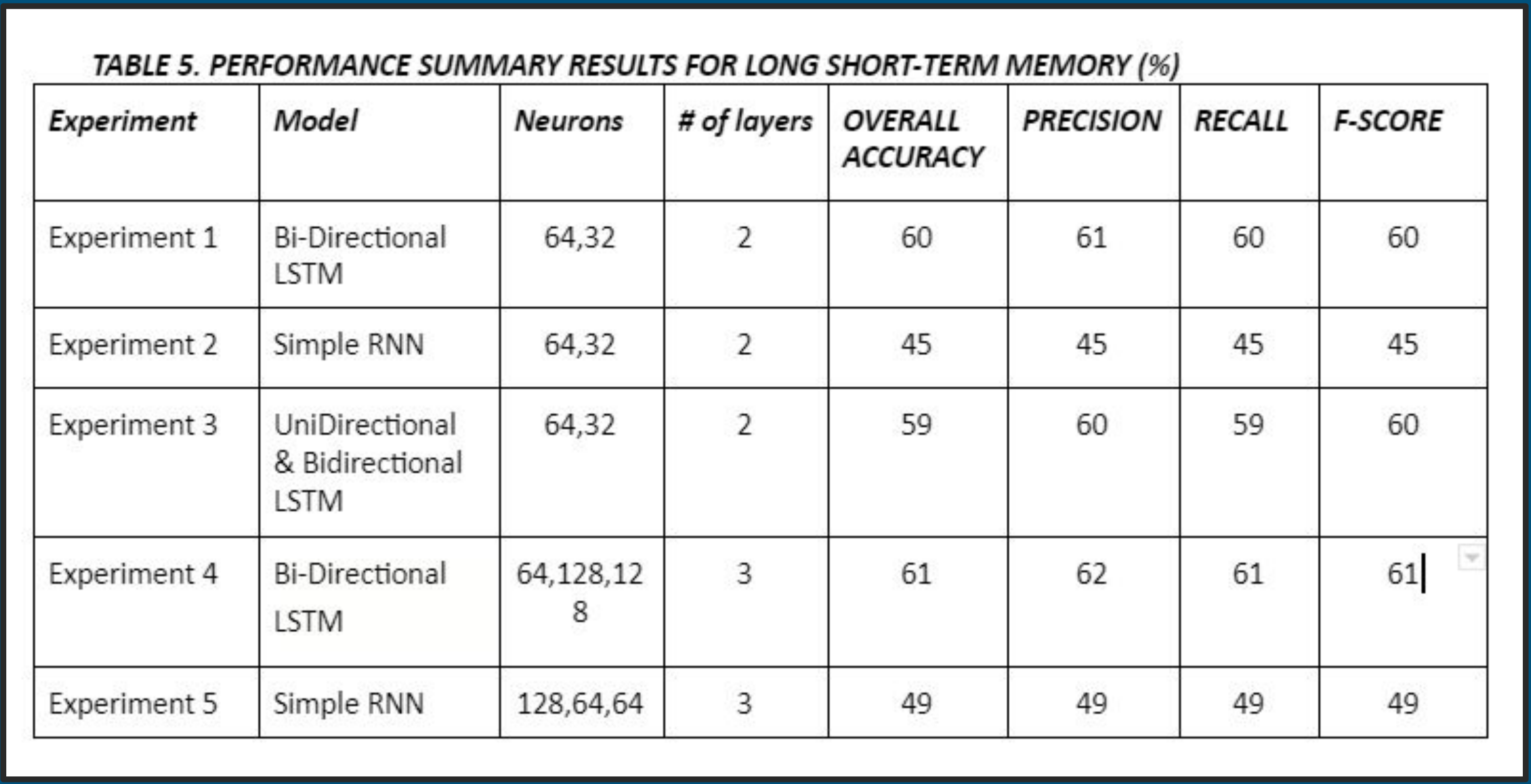


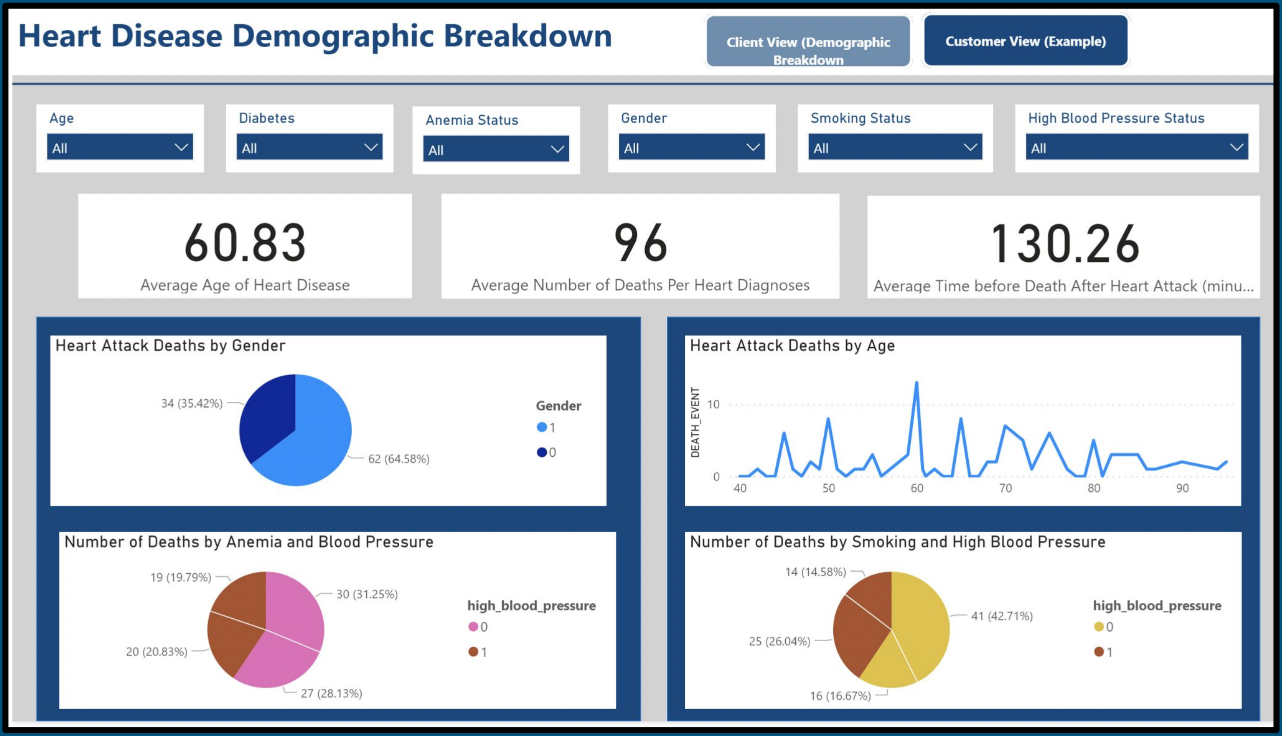
TABLE 5. PERFORMANCE SUMMARY RESULTS FOR LONG SHORT-TERM MEMORY (%)

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

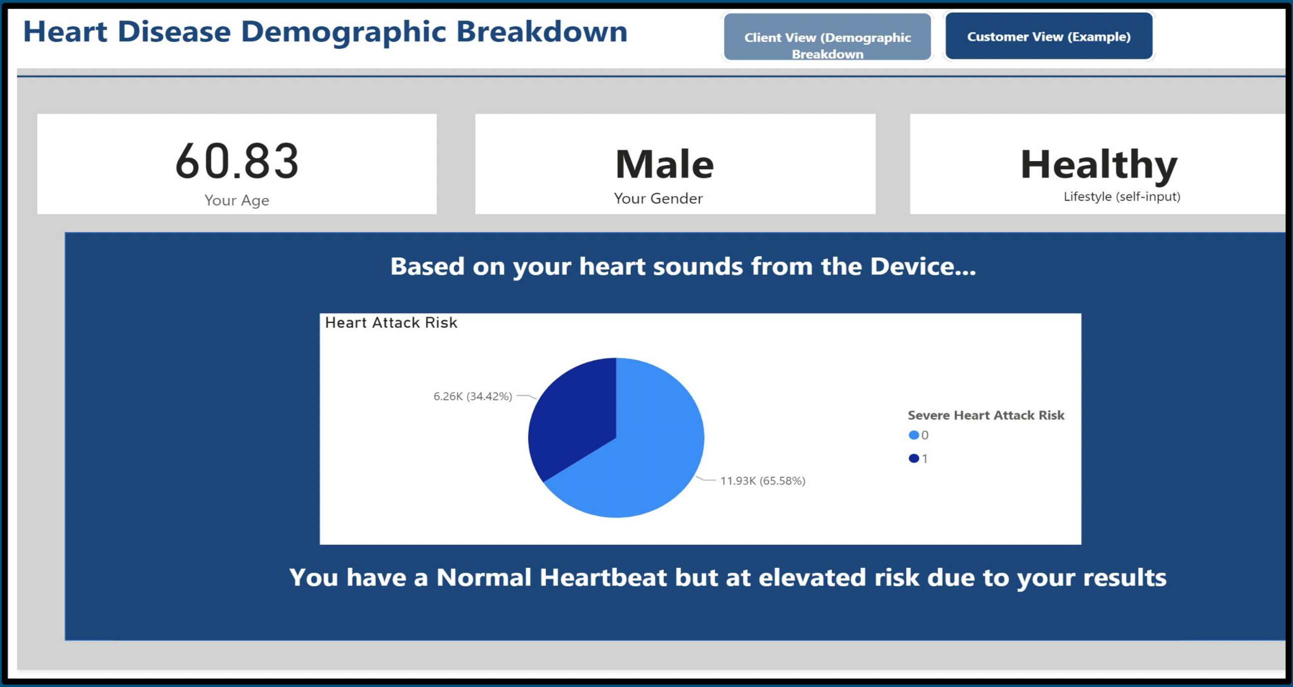
Dashboard Prototype

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.


Heart Disease Demographic Breakdown
Client View (Demographic Breakdown
Customer View (Example)
Age All
Diabetes All
Anemia Status All
Gender All
Smoking Status All
High Blood Pressure Status All
60.83
Average Age of Heart Disease
96
Average Number of Deaths Per Heart Diagnoses
130.26
Average Time before Death After Heart Attack (minu...
Heart Attack Deaths by Gender
34 (35.42%)
62 (64.58%)
Gender
● 1
● 0
Heart Attack Deaths by Age
DEATH_EVENT
10
0
40
50
60
70
80
90
Number of Deaths by Anemia and Blood Pressure
19 (19.79%)
30 (31.25%)
20 (20.83%)
27 (28.13%)
high_blood_pressure
● 0
● 1
Number of Deaths by Smoking and High Blood Pressure
14 (14.58%)
41 (42.71%)
25 (26.04%)
16 (16.67%)
high_blood_pressure
● 0
● 1

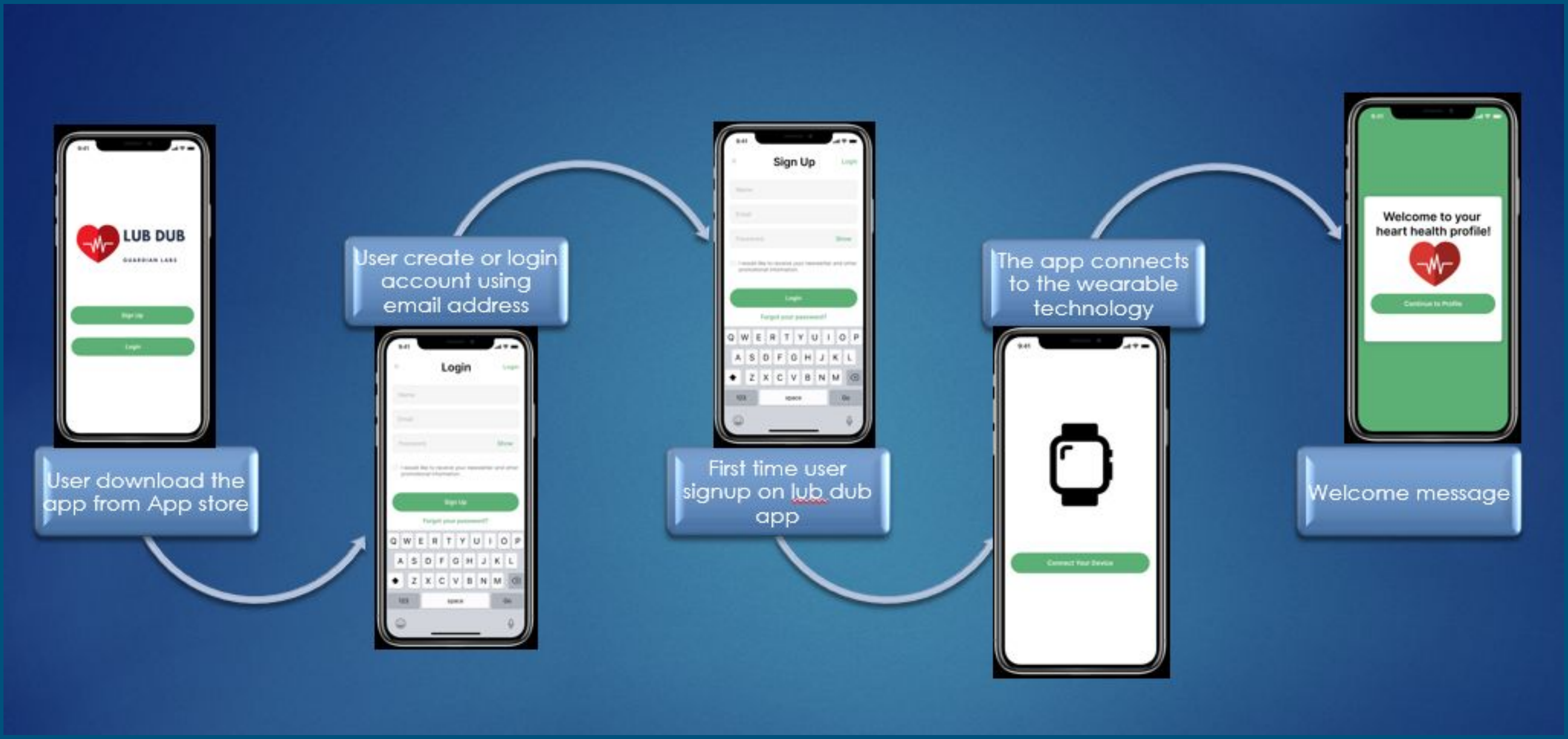
User View (e.g., app users or patients): Contains basic information about the user's heart health status based on data collected via wearable technology.

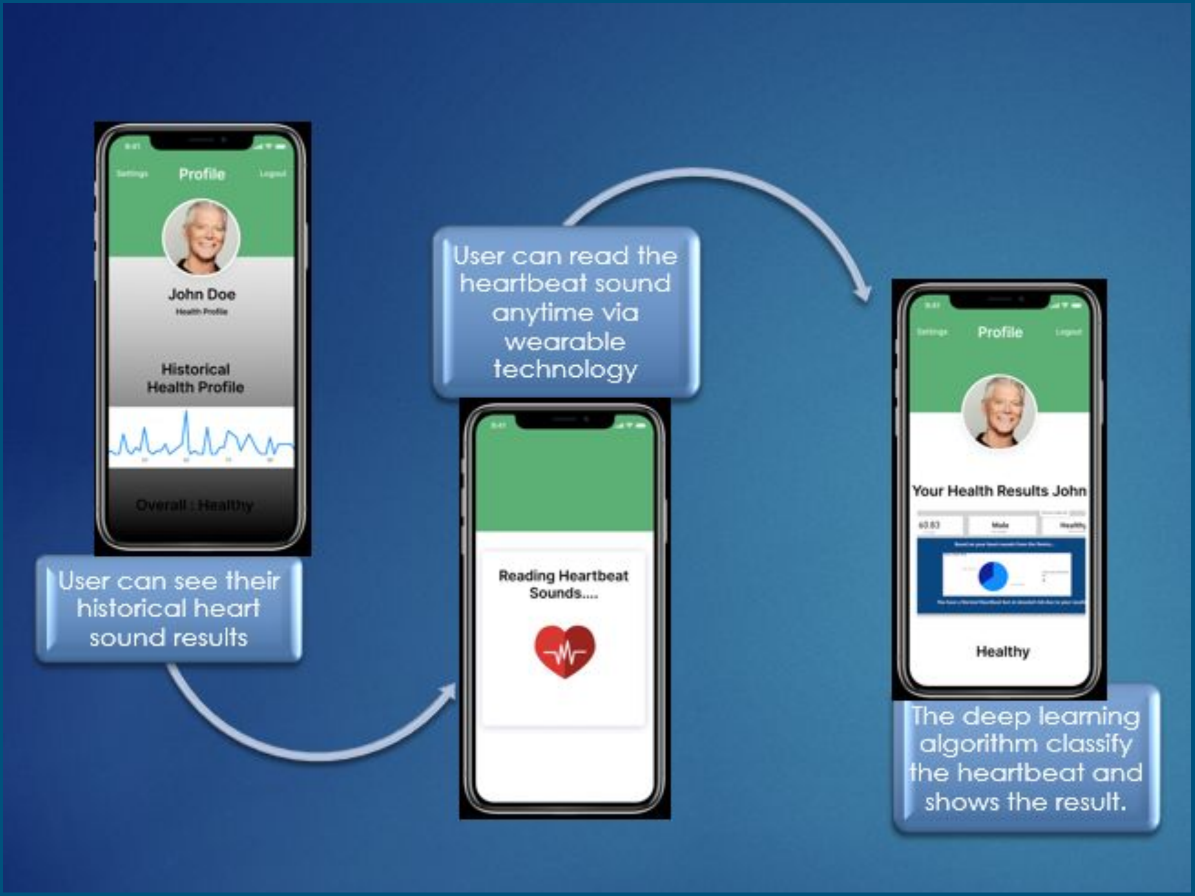

Heart Disease Demographic Breakdown
Client View (Demographic Breakdown
Customer View (Example)
60.83
Your Age
Male
Your Gender
Healthy
Lifestyle (self-input)
Based on your heart sounds from the Device...
Heart Attack Risk
6.26K (34.42%)
11.93K (65.58%)
Severe Heart Attack Risk
● 0
● 1
You have a Normal Heartbeat but at elevated risk due to your results

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

Mobile App Prototype

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.


LUB DUB
User download the app from App store
User create or login account using email address
Login
First time user signup on lub dub app
Sign Up
The app connects to the wearable technology
Connect Your Device
Welcome to your heart health profile!
Continue to Profile
Welcome message


Profile
John Doe
Historical Health Profile
Overall : Healthy
User can see their historical heart sound results
User can read the heartbeat sound anytime via wearable technology
Reading Heartbeat Sounds....
Your Health Results John
Healthy
The deep learning algorithm classify the heartbeat and shows the result.

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

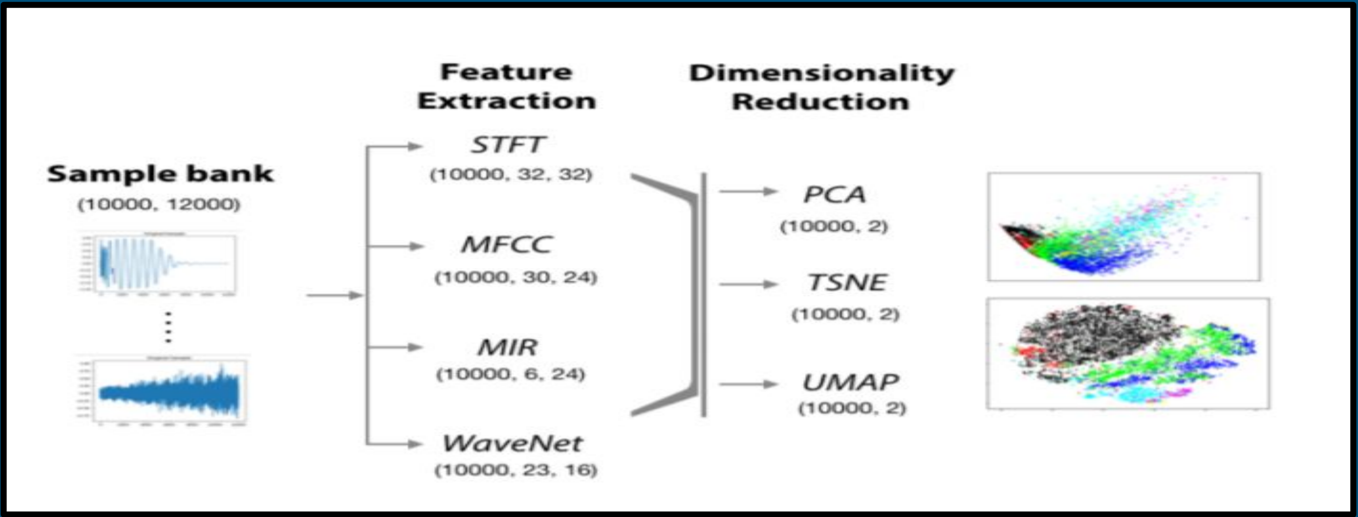
Guardian Lab is planning to expand its client base to more insurance and health care companies by actively connected with them and present the findings of this work.

More Data Collection

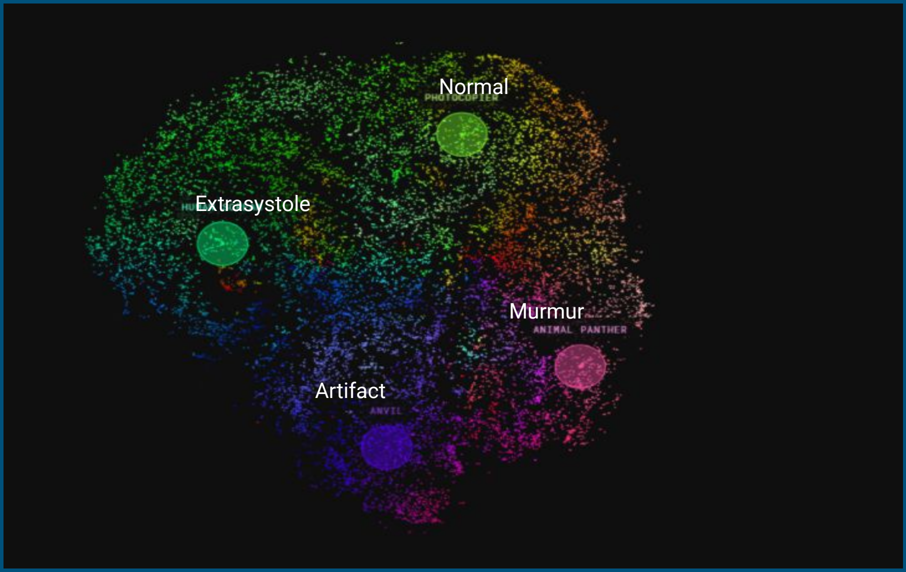
One of the very important future task for this project is to collect more data. By actively engaging with more insurance companies, Guardian Lab will not only increase its client base but will also be able to collect more data.The machine and deep models were built on small number of test and train data 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

More research work is needed to feature extraction techniques in order to improve predictive power of these models. For audio processing, it is important to reduce audio files down to a couple of values, so that they might be plotted and explored in a two-dimensional graph. Since t-SNE specializes in dimensionality reduction for visualization purposes as it transforms highly-dimensional data into a two- or three-dimensional space, therefore it is very useful to compare and group similar sounds.
As of now, Guardian labs have not tried the dimensionality reduction algorithm because of limited amount of training and testing dataset. But in future, Guardian Lab will collect more data points and will definitely use t-SNE on audio heartbeat sound to visualize and isolate different heartbeats. Below is a mockup example of t-SNE visualization for heartbeat sound data to help CEO and board of directors to understand the concept. Guardian Lab will use Librosa library to analyze each audio file and extract MFCC features.



After that t-SNE will be used to reduce the dimensionality of feature matrix into 2d coordinates to visualize as shown in figure below. 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.

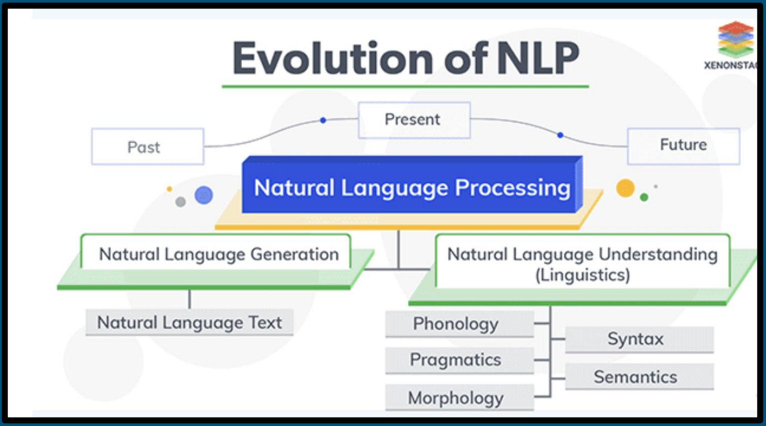


Transition to Cloud Computing Service

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) applications on the robust AWS ML infrastructure. Guardian Labs has been using Google Colaboratory as the primary platform to build and train machine/deep learning models (see Technology Overview section). While the service is free and has enabled the team to conduct initial research on heartbeat classification models, the limited continuous runtime and lack of compute processing power would affect the team's ability to expand models and scale products, especially as the amount of user data increases. AWS would not only accelerate the team's ML work, but also enhance our data collection/processing and project management efforts.

- **Streaming Data Collection & Processing** | We plan to leverage AWS's data streaming service, *Kinesis*, to continuously (in real-time) ingest user event logs (generated by the various wearable technology devices) into our AWS instance. This approach would enable us to securely store in Amazon's *Simple Storage Service (S3)* while simultaneously cleaning, processing, and normalizing the data in batch jobs using Amazon's *Elastic MapReduce (EMR)* service.
- **Efficient & Collaborative Modeling Process** | We would transition to Amazon *SageMaker* to provide data engineers/scientists at Guardian Labs an environment where they can collaboratively build, train, and deploy ML models quickly and iteratively.
- **Real-Time Progress Monitoring** | Amazon CloudWatch would provide technical managers and product owners data and actionable insights to monitor applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health.

Addition of Natural Language Processing Techniques



- Sound data relies on NLP in order to produce the highest accuracy of classification The core technology of Guardian Lab's is a wearable device that relies on sound data to accurately classify four types of heart health conditions: Normal, extrasystole, murmur and artifacts. We currently have different neural network models to predict classification of these heart conditions, however, we want to go beyond neural networks to improve the accuracy of our product. Our current accuracy is at 83%, we want to prioritize the best product for our customer and I believe that we need to implement NLP modeling into our company as soon as we have the resources and capital to do so, within one or two months from now.

- NLP will help us model risk adjustment condition categories for our insurance stakeholders In addition to our stakeholder of the customer, our other stakeholders are the health insurance companies, such as United Health Group. Insurance companies like United Health Group have hierarchical condition models, which essentially serve as a risk adjustment model for their patient. This could save our client millions of dollars by having extremely accurate classification of heart disease. It could also help mitigate the cost of diseases related to heart disease like high cholesterol, diabetes and more. In terms of a timeline for implementing NLP into our product lines, we should prioritize the technology of NLP first into our client portal, then focus our efforts of implementing NLP analytics to our insurance stakeholders. Therefore, we believe we need to prioritize NLP over a period of 4-5 months to benefit both stakeholders.

- The Cost of Implementing NLP is low According to the "NLP practitioner" (resource for NLP enthusiasts), the cost of implementing NLP methods are extremely low ($0-$100). Most enterprise software and 3 packages such as GLoVe and Comet, are open sourced datasets that can be used with standard data science packages and ide's such as jupyter lab. Hiring an additional engineer to improve the quality of our work outweighs the cost of the engineer itself. I believe we should hire an additional engineer as soon as possible, within 1-2 months of this proposition

Bibliography

Bambrick, Noel. n.d. “Support Vector Machines: A Simple Explanation.” KD Nuggets. Accessed April 28, 2021. https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html.

Bradley Boeh. n.d. “Chapter 12 Gradient Boosting.” Hands-On Machine Learning with R. Accessed April 28, 2021. https://bradleyboehmke.github.io/HOML/gbm.html.

Doshi, Sanket. 2018. “Music Feature Extraction in Python.” Towards Data Science. https://towardsdatascience.com/extract-features-of-music-75a3f9bc265d.

Github. n.d. “Pooling Layer.” cs231. Accessed April 28, 2021. https://cs231n.github.io/convolutional-networks/#pool.

Kaggle. 2019. “K-Nearest Neighbours.” Geeks for Geeks. https://www.geeksforgeeks.org/k-nearest-neighbours/.

Mittal, Aditi. 2019. “Understanding RNN and LSTM.” Aditi Mittal. https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e.

Ray, Sunil. 2017. “6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R.” Analytics Vidhya. https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/.

Roberts, Leland. 2020. “Understanding the Mel Spectrogram.” Analytics Vidhya. https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53.

SPRH LABS. 2019. “Understanding Deep Learning: DNN, RNN, LSTM, CNN and R-CNN.” SPRH Labs. https://medium.com/@sprhlabs/understanding-deep-learning-dnn-rnn-lstm-cnn-and-r-cnn-6602ed94dbff.

UJWALPAWAR913. 2020. “Lets Open the Black Box of Random Forests.” Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/12/lets-open-the-black-box-of-random-forests/.

Wikipedia. n.d. “Mel-frequency cepstrum.” Accessed April 28, 2021. https://en.wikipedia.org/wiki/Mel-frequency_cepstrum.

Bibliography

Bentley, Peter, Glenn Nordehn, Miguel Coimbra, Shie Mannor, and Rita Getz. n.d. "Classifying Heart Sounds Challenge." PeterJBentley. Accessed April 14, 2021. http://www.peterjbentley.com/heartchallenge/index.html.

Centers for Disease Control and Prevention. 2020. "Heart Disease Facts." Heart Disease Statistics and Maps. https://www.cdc.gov/heartdisease/facts.htm.

"Heart Disease: Facts, Statistics, and You." 2020. healthline. https://www.healthline.com/health/heart-disease/statistics.

Japsen, Bruce. 2021. "UnitedHealth's Optum To Buy Data Analytics Firm Change Healthcare In $13 Billion Deal." *Forbes*, January 6, 2021. https://www.forbes.com/sites/brucejapsen/2021/01/06/unitedhealths-optum-to-buy-data-analytics-firm-change-healthcare/?sh=441371e7307f.

Knapp, Amy. 2014. "Cancer vs. Heart Disease: Which Costs Insurers the Most Money?" Pacific Prime News Page. https://www.pacificprime.com/resources/news/cancer-vs.-heart-disease-which-costs-insurers-the-most-money/.

Mordor Intelligence. 2021. "Wearable Technology Market - Growth, Trends, COVID-19 Impact, and Forecasts (2021 - 2026)." Mordor Intelligence Reports. https://www.mordorintelligence.com/industry-reports/wearable-technology-market.

National Institutes of Health. 2020. "Heart Disease and Stroke." NIH: Our Biggest Health Challenges. https://www.nih.gov/about-nih/what-we-do/nih-turning-discovery-into-health-/heart-disease-stroke.

Rapaport, Lisa. 2019. "Almost half of U.S. heart disease patients struggle with medical bills." *Reuters*, February 11, 2019. https://www.reuters.com/article/us-health-heart-finances/almost-half-of-u-s-heart-disease-patients-struggle-with-medical-bills-idUSKCN1Q02HF.

Bibliography

UnitedHealth Group. 2019. “More than Twice as Many Employers than 10 Years Ago are Planning to Increase Investments in Employee Health and Wellness, Optum Study Shows.” UnitedHealth Group Newsroom Press Release. https://www.unitedhealthgroup.com/newsroom/2019/2019-08-21-optum-employee-wellness.html.

Vailshery, Lionel. 2021. “Wearable users penetration in the United States 2016-2022.” Wearable users penetration in the United States 2016-2022. https://www.statista.com/statistics/793800/us-adult-wearable-penetration/#:~:text=This%20statistic%20shows%20the%20penetration,at%20least%20once%20a%20month.

ValuePenguin and Sterling Price. 2021. “Largest Health Insurance Companies of 2021.” valuepenguin.com. https://www.valuepenguin.com/largest-health-insurance-companies.

Vogels, Emily A. 2020. “About one-in-five Americans use a smart watch or fitness tracker.” Pew Research Center. https://www.pewresearch.org/fact-tank/2020/01/09/about-one-in-five-americans-use-a-smart-watch-or-fitness-tracker/.

Heartbeat-Classifier/Heart Anomaly Detection.pdf at master · MananAgarwal/Heartbeat-Classifier · GitHub

Heart sounds analysis and classification with LSTM

Heartbeat-Classifier/Heart Anomaly Detection.pdf at master · MananAgarwal/Heartbeat-Classifier · GitHub

Heart sounds analysis and classification with LSTM

https://paulvanderlaken.com/2017/08/11/t-sne-the-ultimate-drum-machine-and-more

Comprehensive Guide on t-SNE algorithm with implementation in R & Python

Bibliography

Comprehensive Guide on t-SNE algorithm with implementation in R & Python

https://medium.com/@hanoi7/klustr-a-tool-for-dimensionality-reduction-and-visualization-of-large-audio[1]datasets-c3e958c0856c

https://paulvanderlaken.com/2017/08/11/t-sne-the-ultimate-drum-machine-and-more

https://github.com/LameesKadhim/Heart-Disease-Classifier/blob/main/Heart%20disease%20classifier.ipynb

https://github.com/MananAgarwal/Heartbeat-Classifier/blob/master/Heartbeat%20Classifier.ipynb

https://www.kaggle.com/bagussulistyo/heartbeat-prediction-with-mfccs-and-cnn

https://paulvanderlaken.com/2017/08/11/t-sne-the-ultimate-drum-machine-and-more