Heartbeat Classification

General Assembly

Data Science Immersive 13 Capstone

Ng Gim Pei

Summary Topics

1. Problem Statements
2. Project Framework
3. Exploratory Data Analysis
4. Preprocessing
5. Audio Feature Extraction Techniques
6. Classification Models
7. Model Evaluation, Results and Discussions
8. Summary and Recommendations

Problem Statements

Singapore Statistics:

Every day, 17 people die from cardiovascular disease (heart diseases and stroke) in Singapore. Cardiovascular disease accounted for 29.2% of all deaths in 2018. This means that almost 1 out of 3 deaths in Singapore, is due to heart diseases or stroke.

Source: Singapore Heart Foundation, Ministry of Health

Problem Statements

Stethoscope is an instrument that is widely used in medical field to diagnose heart disease.

Doctors can find heart diseases from listening to the heartbeat by using the stethoscope.

Though, digital stethoscope is available for consumer, but to distinguish whether or not it is abnormal heartbeat requires experienced or trained clinical persons with stethoscope hearings.

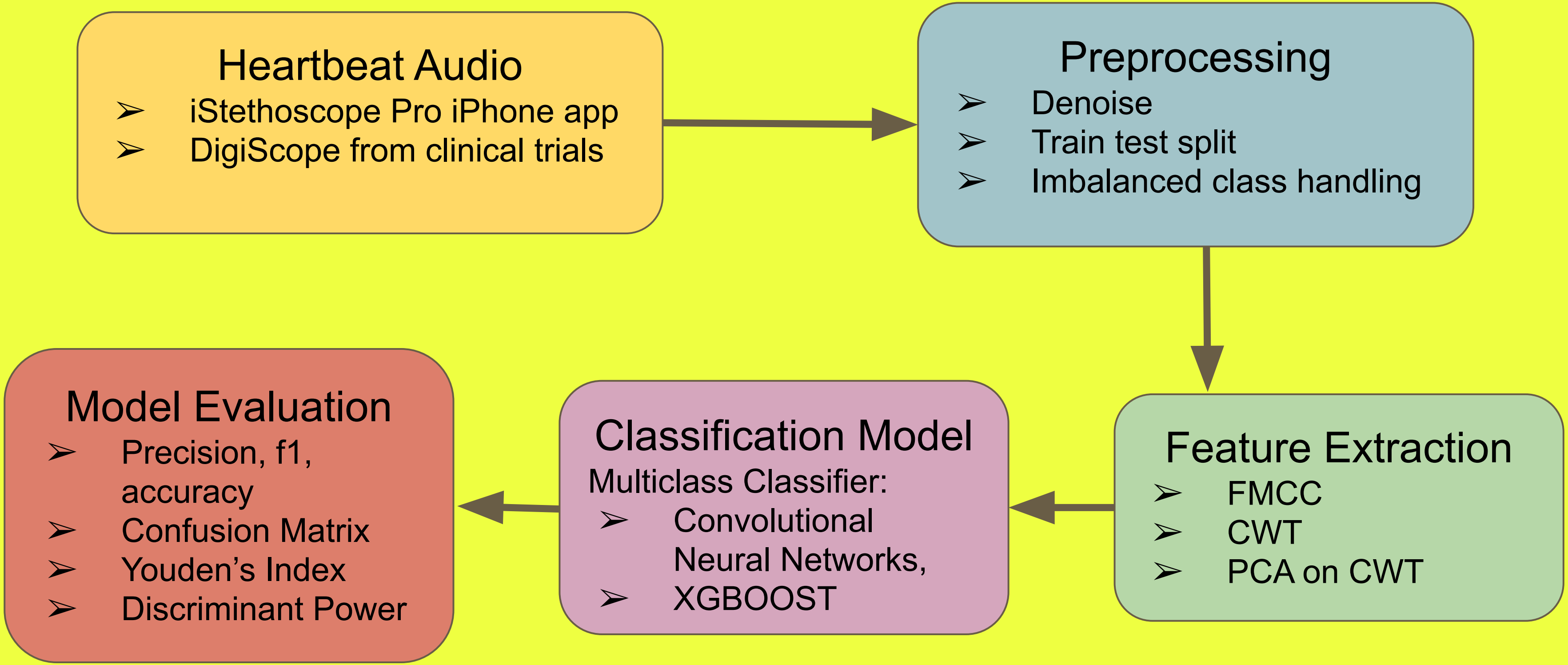
Problem Statements

Goal: Build a Machine Learning Model to classify heartbeat audio from stethoscope into normal versus various non-normal heartbeat category.

Target audience:

General consumer.
We only seek medical attention when we feel ill. The success of this project would be handy to complement with the digital stethoscope where consumer can monitor their heartbeat condition at their convenient and seek medical attention soonest possible, if abnormal heartbeat is detected.

Project Framework:



Exploratory Data Analysis (EDA)

- Data type
- Distribution of categories
- Audio length

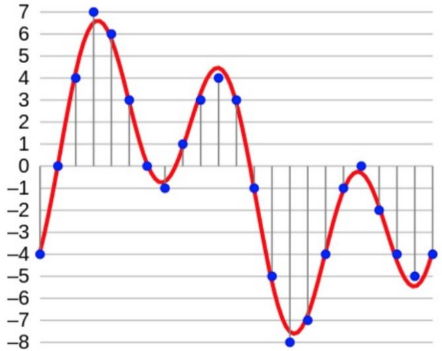
EDA

Dataset	Number of Categories	Categories	Sources	Recorded by
A	4	Normal, Murmur, Extra Heart Sound, Artifact	general public	iStethoscope Pro iPhone app
B	3	Normal, Murmur, Extrsystole	clinic trial in hospitals	digital stethoscope

Data Type

Audio wav:

- Set a: mono channel, bit-depth = 16, sampling rate = 44.1kHz
- Set b: mono channel, bit-depth = 16, sampling rate = 4kHz

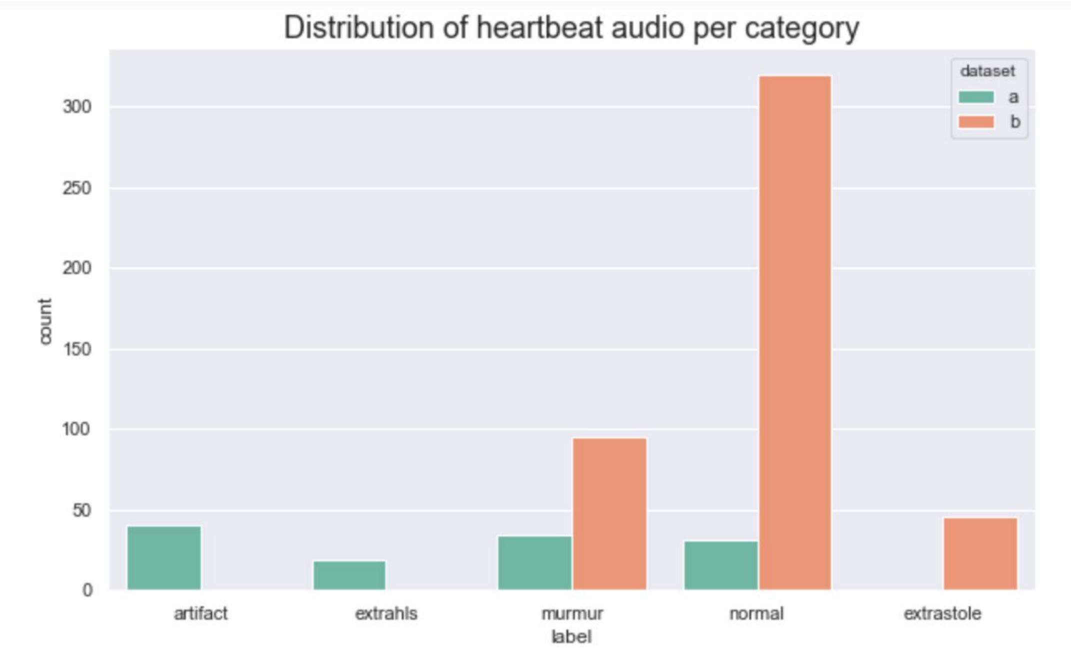


Sound wave

[-4, 0, 4, 7, 6, 3, 0, -1, 1, 3, 4, 3, -1, -5, -8, -7, -4, -1, 0, -2, -4, -5, -4]

Array

Distribution of categories



Imbalance Class

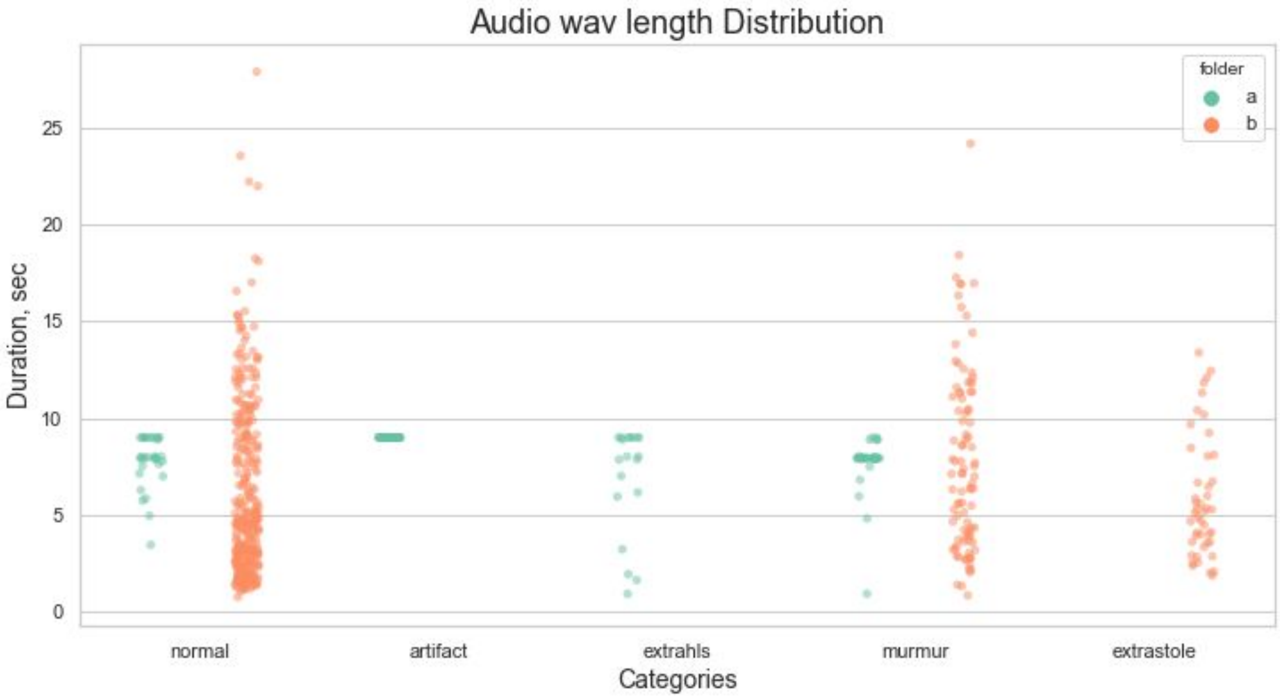
Majority:

- Normal

Minority:

- Extra heart sound, Extrasystole

Audio length

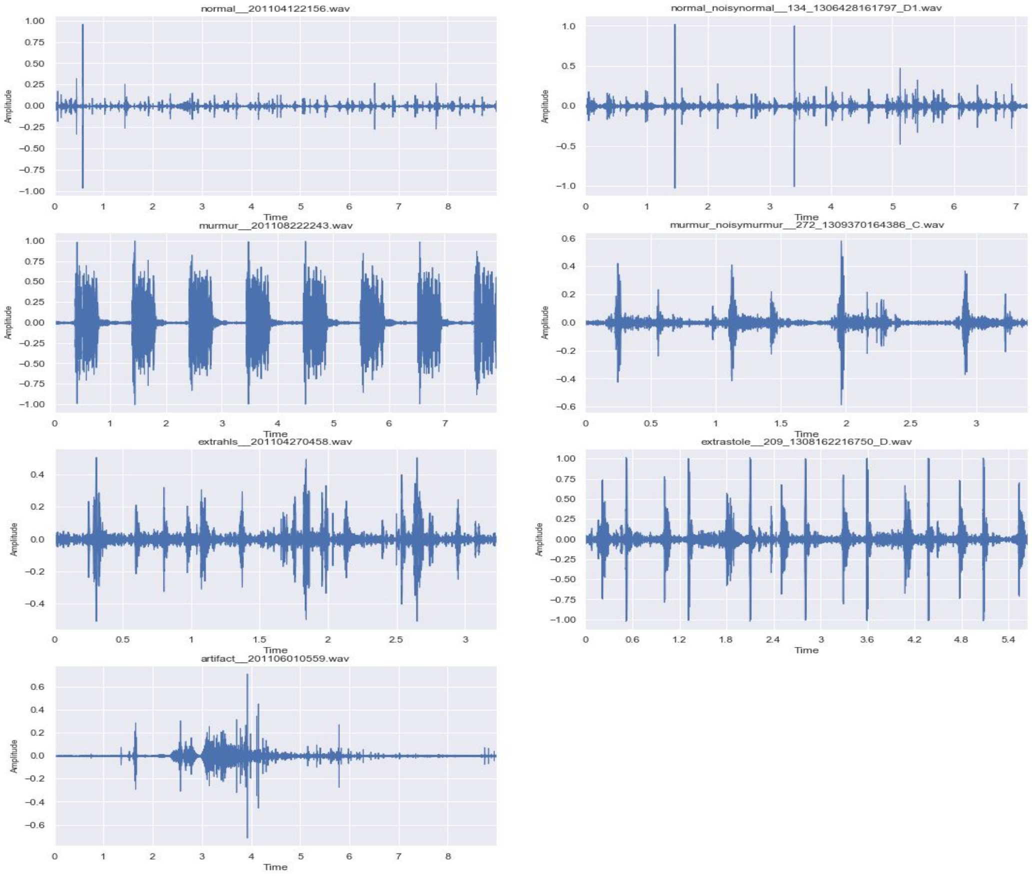


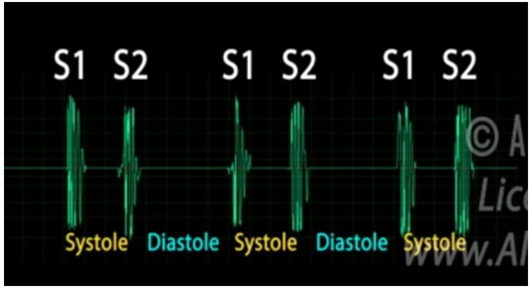
label	min	max	mean	count
artifact	9.000000	9.00000	9.000000	40
extrahls	0.936372	9.00000	6.872237	19
extrastole	1.874500	13.38075	5.858043	46
murmur	0.856750	24.16000	7.774043	129
normal	0.763250	27.86700	6.318183	351

Preprocessing

- Denoise
- Train Test Split
- Imbalanced Class

Categories at a glance





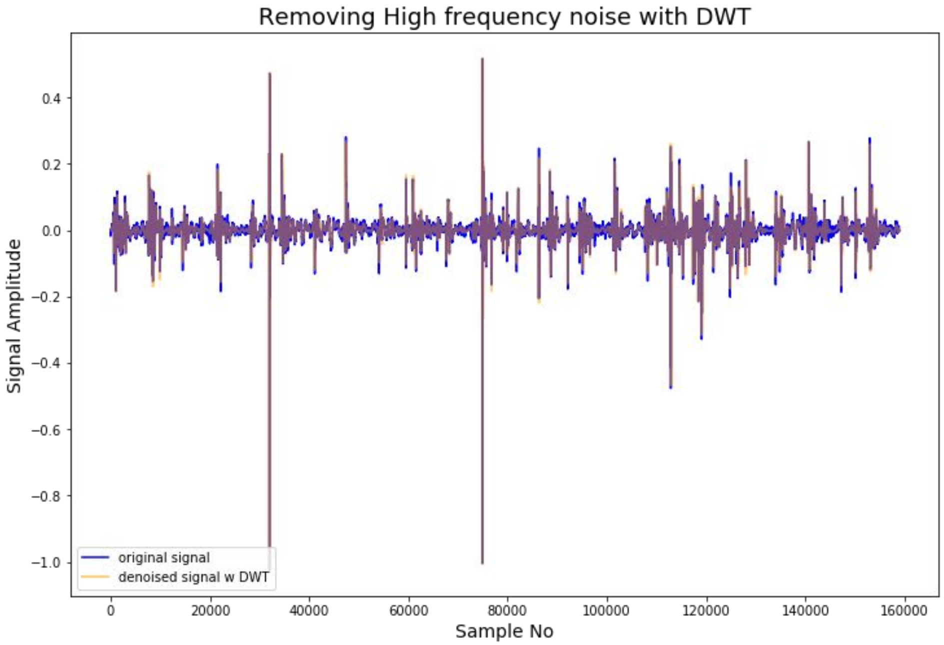
Denoise

Technique

Discrete Wavelet Transformation (DWT)

Settings:

- Wavelet family: ‘Daubechies’ (‘db’)
- Subcategories: ‘db6’
- Level = 10
- Threshold = 0.3*max(x)

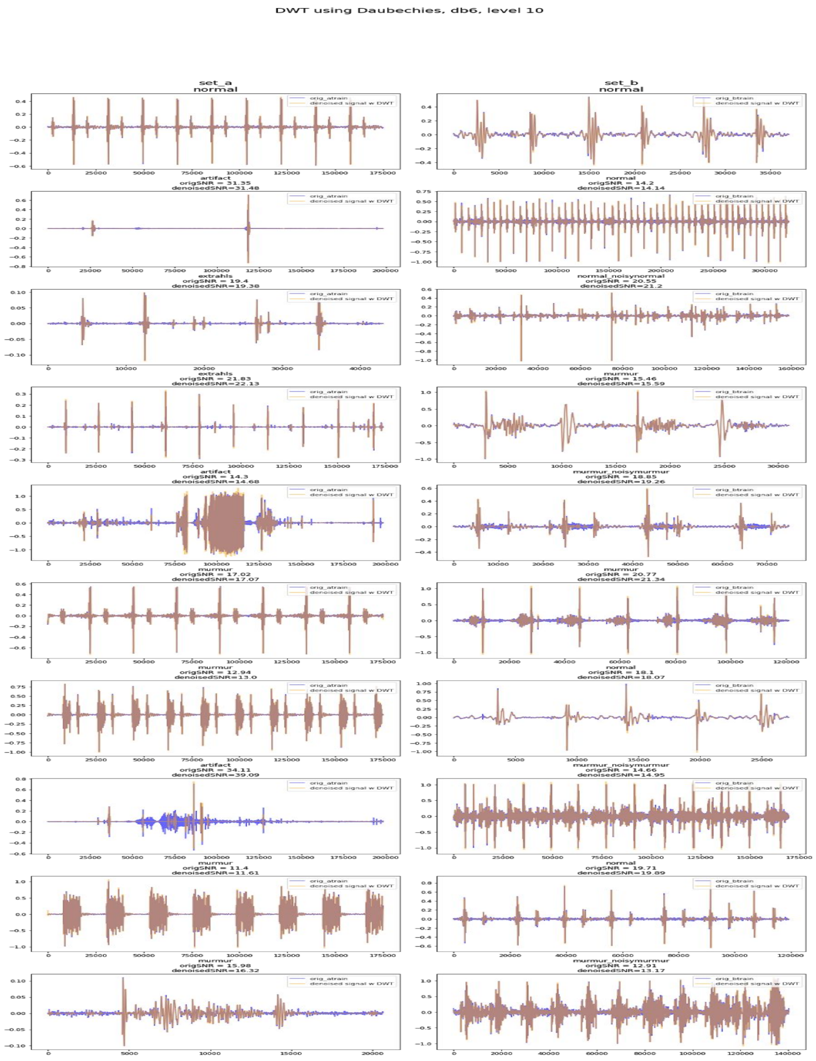


Denoise

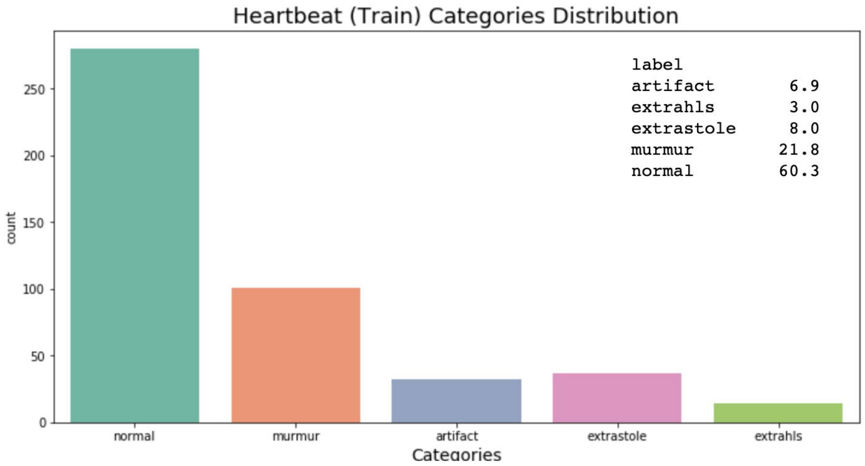
Metrics:

Wavelet
sub-categories
decided based on:

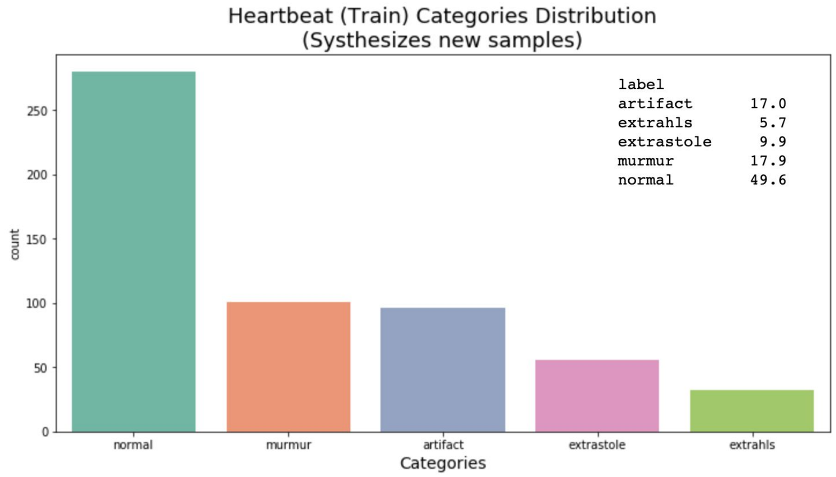
Signal-to-Noise
Ratio (SNR)

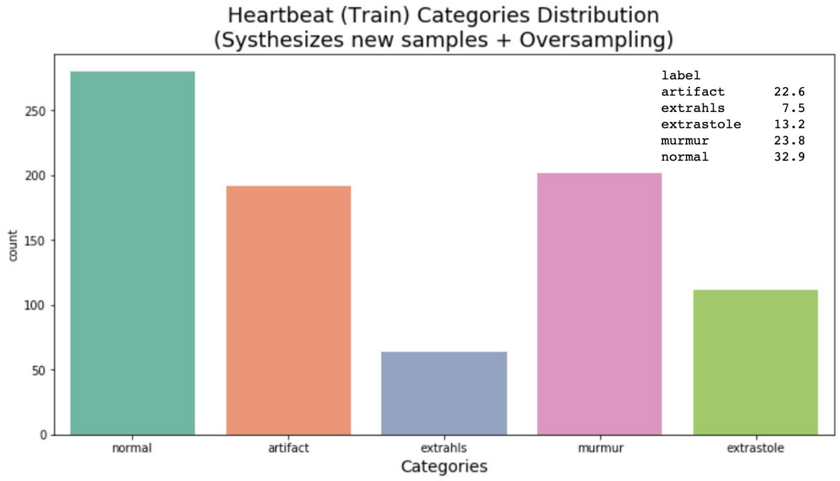


Imbalance Class



- Models work best when each class (heartbeat category) has similar sized in the training dataset.
- Imbalance class with majority category at 60%





Audio Feature Extraction

- MFCC

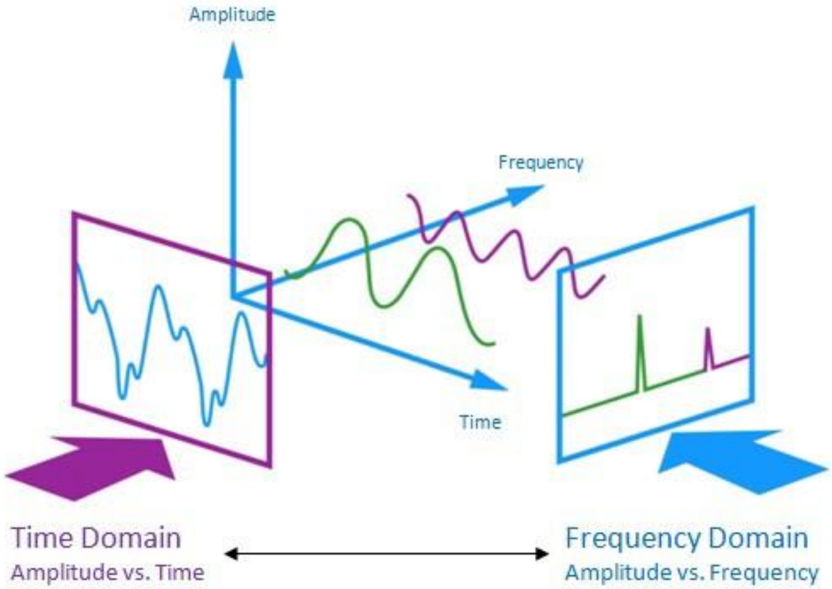
  (Mel Frequency Cepstral Coefficients)

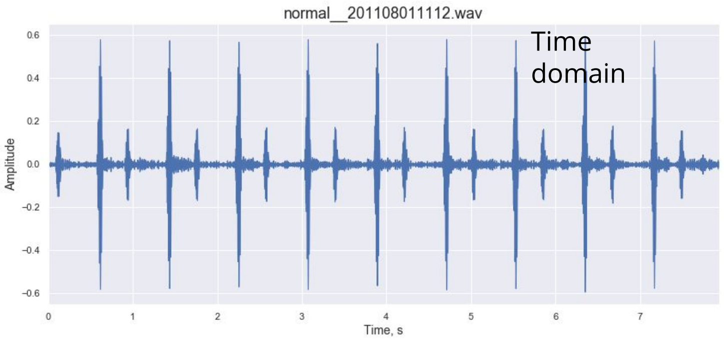
- CWT

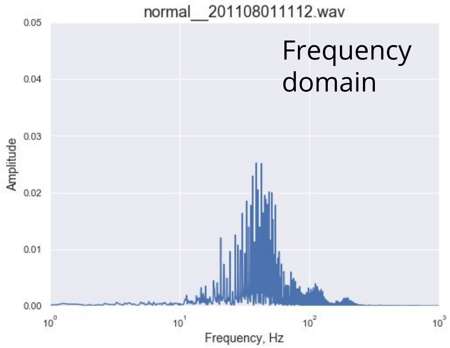
  (Continuous Wavelet Transform

Audio Feature

Audio signal can be represented in time domain and frequency domain

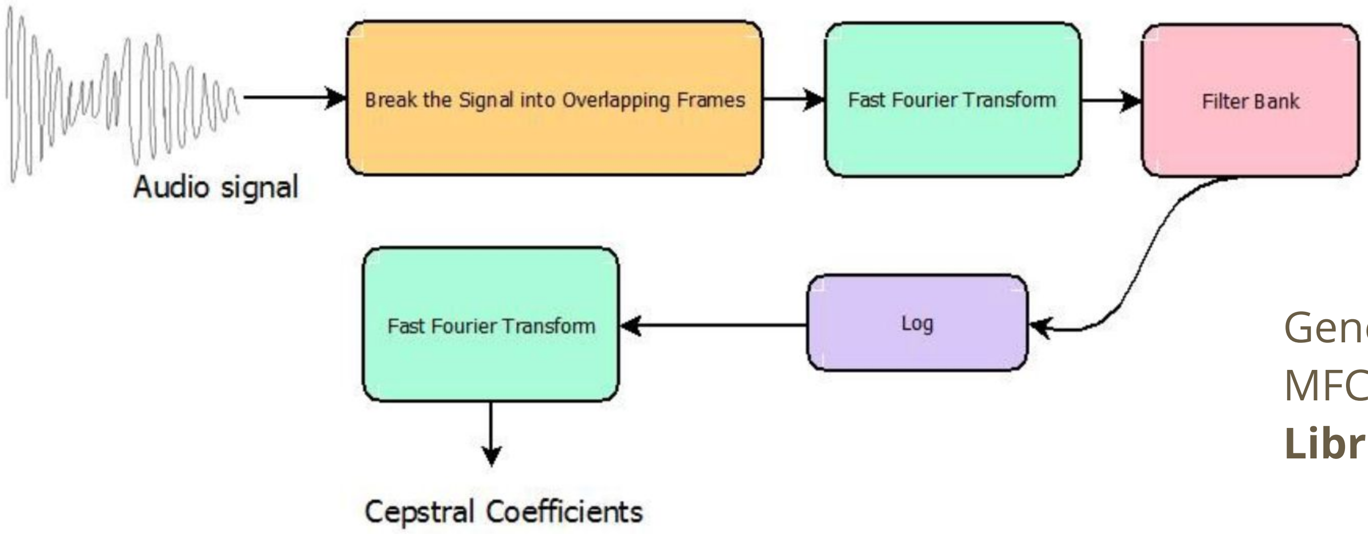






MFCC (Mel Frequency Cepstral Coefficients)

Popular techniques to extract features from audio signals is computing the MFCC from the raw audio signal.



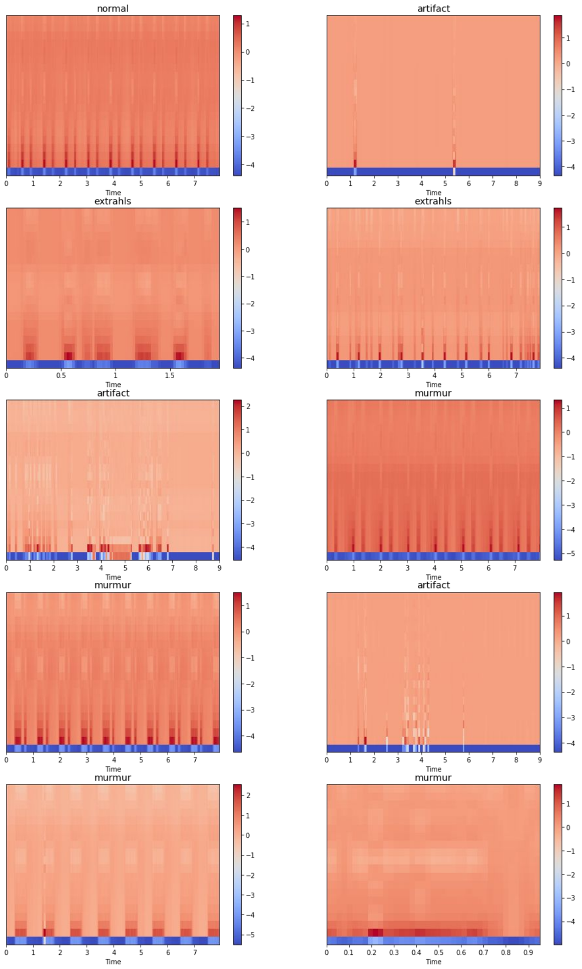
MFCC

Generating MFCC using **Librosa** library

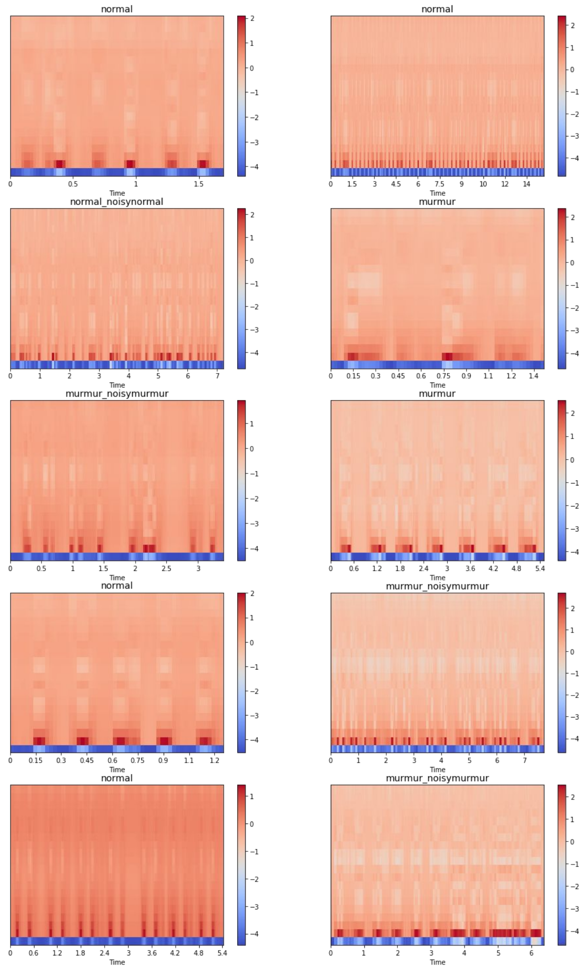
MFCC

There are patterns on each audio wav, which, can use it to train deep learning model to classify the heartbeat category based on the pattern!

Set a



Set b

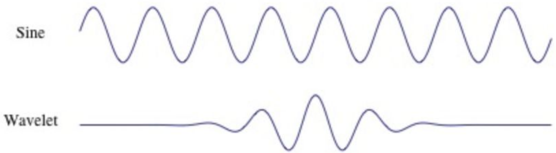


CWT (Continuous Wavelet Transform)

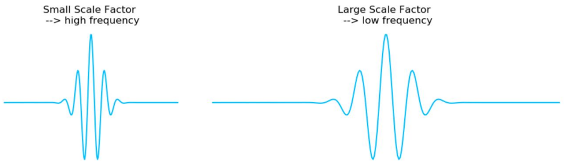
Wavelet transform an 1D signal to 2D. The output of CWT is time-pitch-power representation of the signal in the form of a **scalogram.**

$$cwt(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} x(t)\psi\left(\frac{t-\tau}{s}\right) dt$$

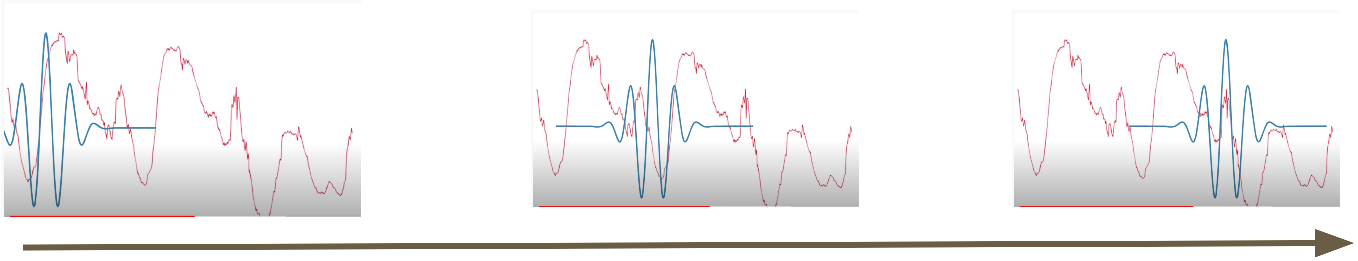
$$\tau = translation, s = scale, \Psi(t) = mother\ wavelet, \left(\frac{t-\tau}{s}\right) = sale\ factor$$



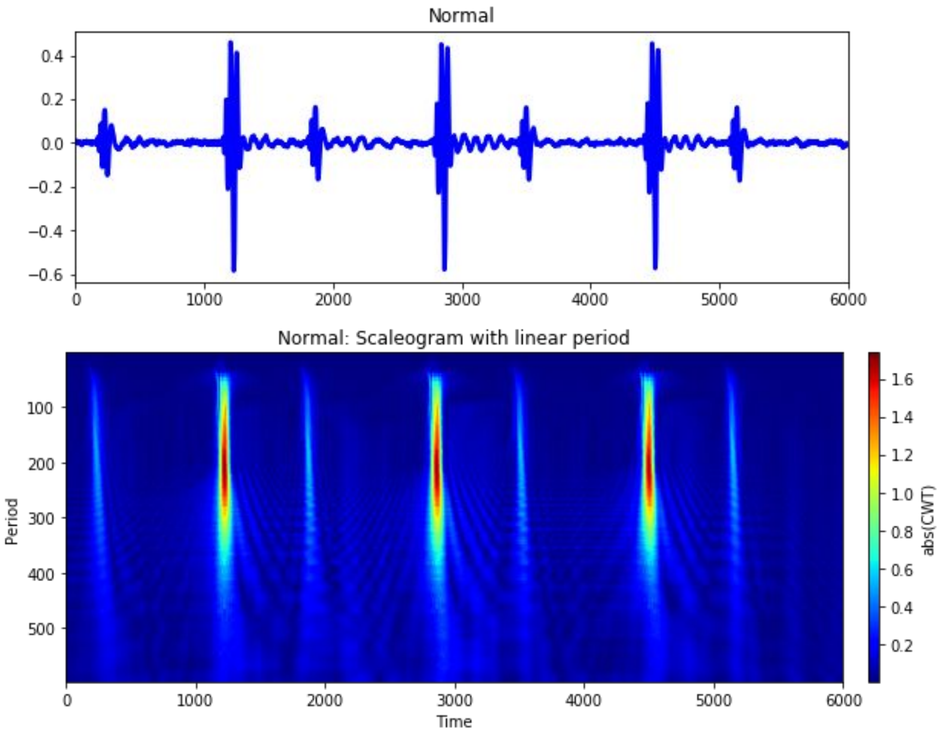
scaling

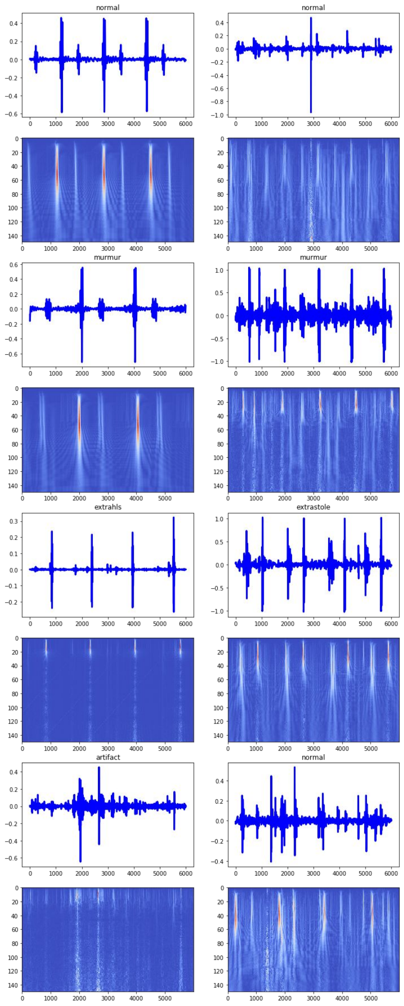


translation



Scalogram





Similarly, there are patterns on each audio wav, which, can use it to train deep learning model to classify the heartbeat category based on the pattern!

Classification Model

(Multiclass classifier)

2D Convolutional Neural Networks

XGBOOST

Building CNN Model

Raw audio file (44.1kHz, mono)

Resampling (22050 Hz) + Denoise (DWT)

Fix audio duration to 3 sec via zero padding or post truncating

MFCC extraction + Normalization

Conv2D (filters = 16, kernel_size =3, activation = ‘relu’, padding = ‘same’)

MaxPool2D (pool_size = ())

Conv2D(filters = 32, kernel_size =3, activation = ‘relu’, padding = ‘same’)

MaxPool2D(pool_size =())

Dropout(0.2)

Conv2D(filters = 32, kernel_size =5, activation = ‘relu’, padding = ‘same’)

MaxPool2D(pool_size =())

Dropout(0.2)

Flatten()

Dense (128, activation = ‘relu’, kernel_regularizer= regularizers.l2(0.0001)

BatchNormalization()

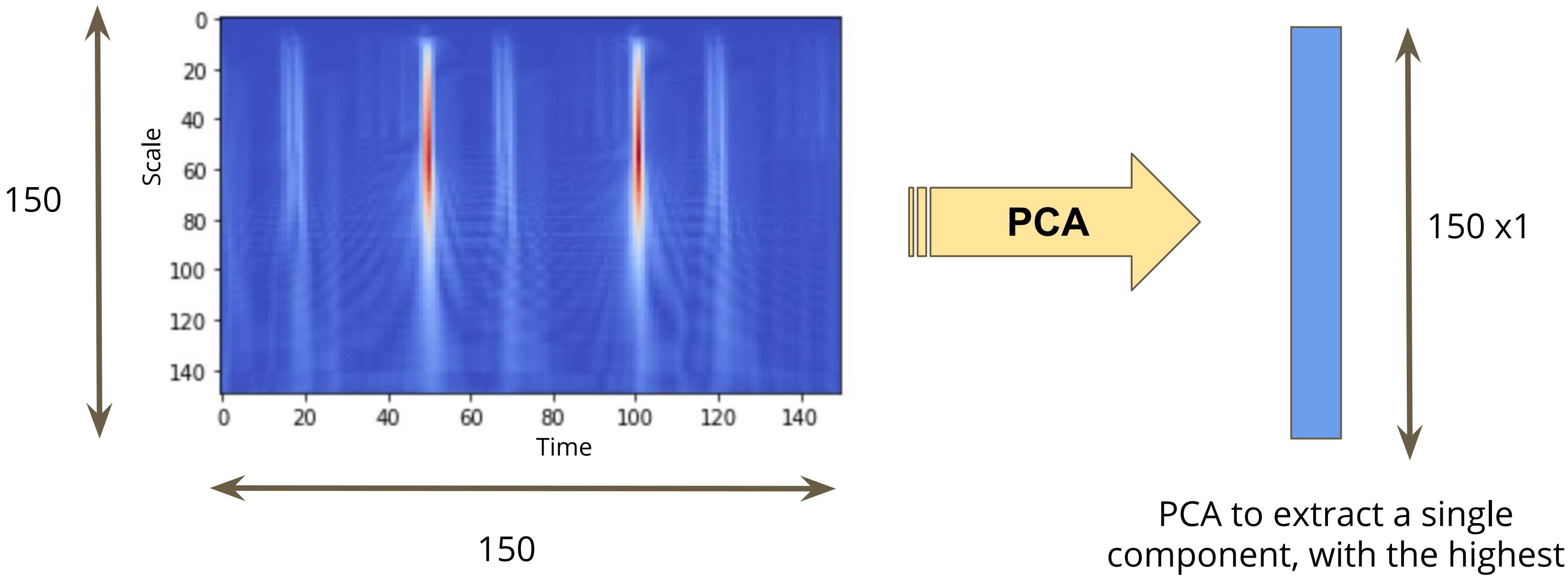
Dropout (0.5)

Dense (5)

Softmax Output

Same CNN architech for MFCC & CWT, but with varying hyperparameters

Building Model using PCA on CWT + XGBOOST



PCA to extract a single component, with the highest variation per scale

Model Evaluation Results Discussions

Metrics:

1. Precision
2. f1
3. Accuracy
4. Youden’s Index
5. Discriminant Power

Data:

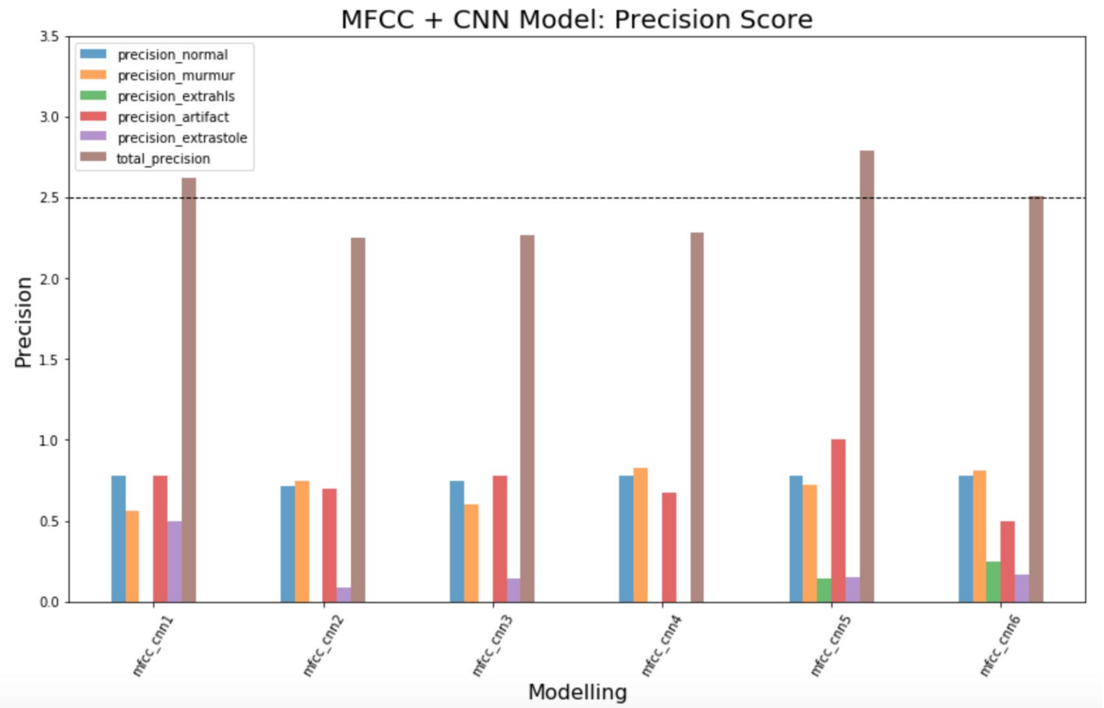
1. TEST set
2. Unlabel set_a, set_b

Results

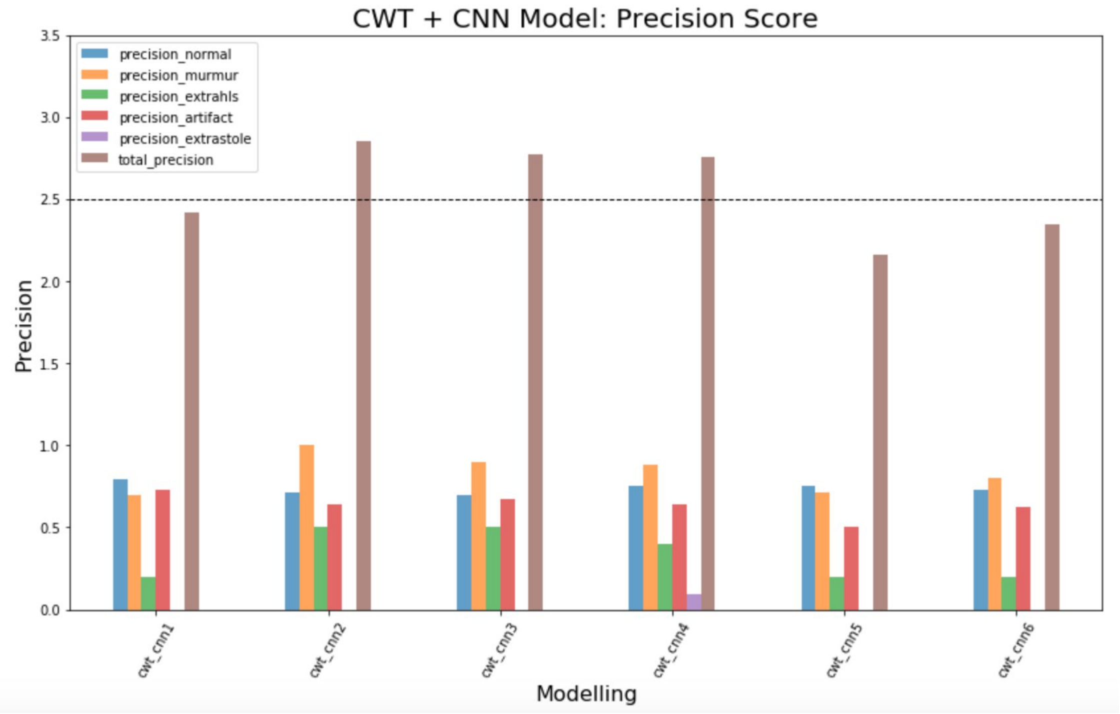
$$Precision = \frac{TP}{FP + TP}$$

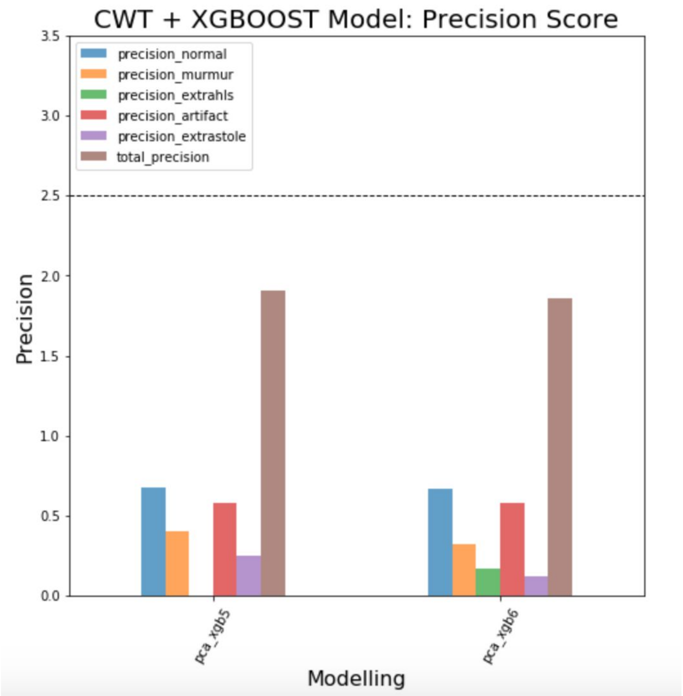
Example: Murmur category

Among all predicted murmur heartbeat, how many did I predict correctly?



Results : Precision

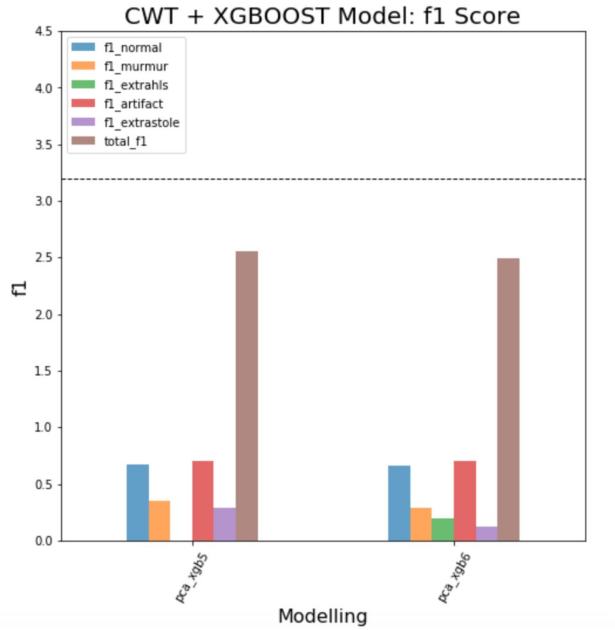


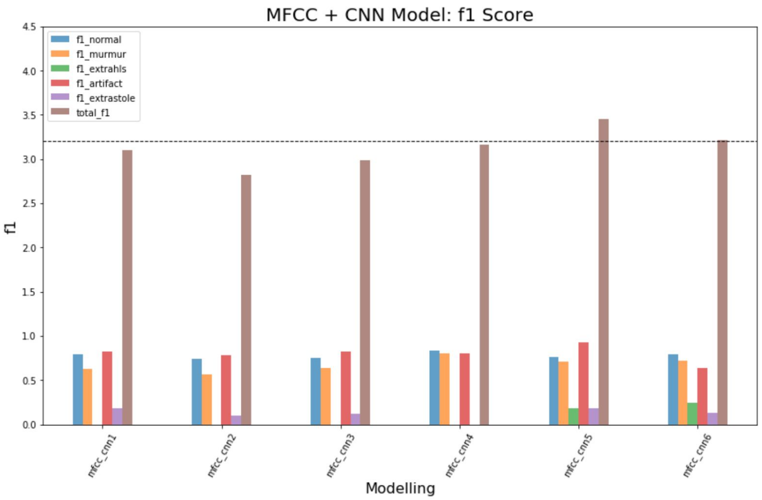


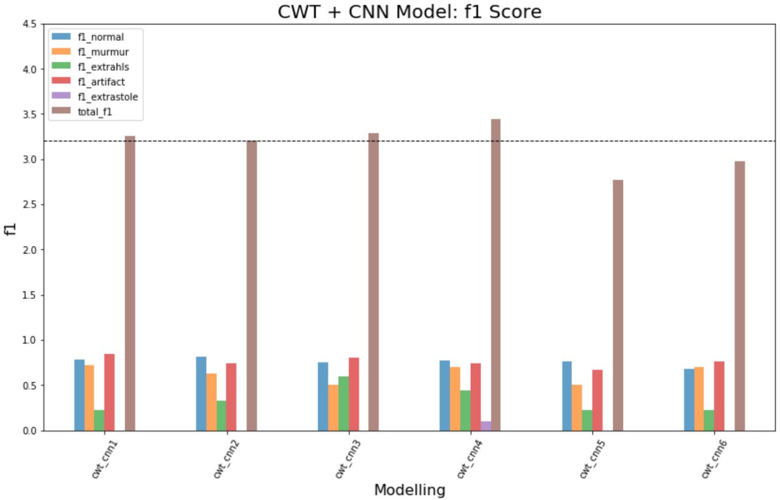
Results

$$f1 = 2\frac{Precision * Recall}{Precision + Recall}$$

f1 is weighted average for precision and recall



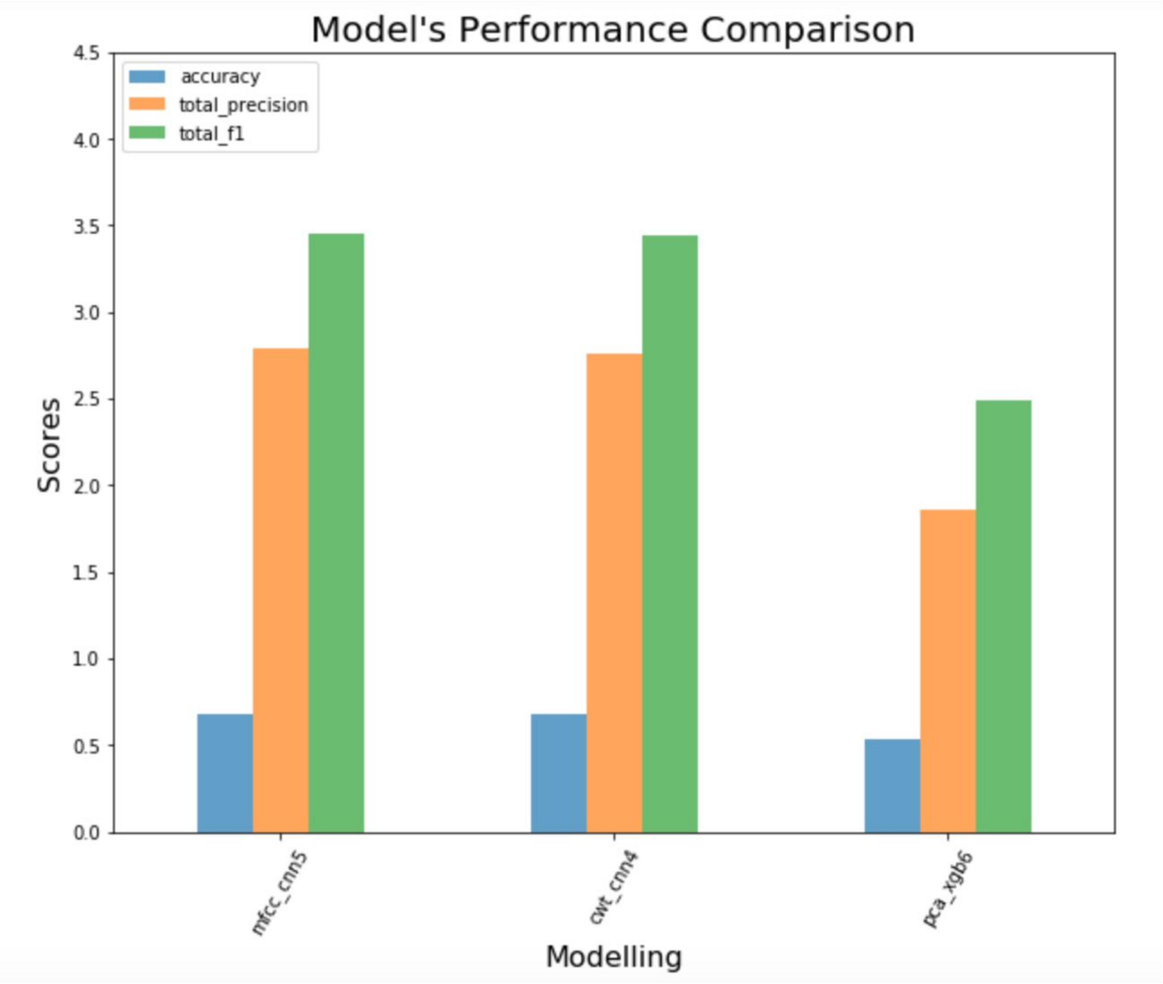




Results

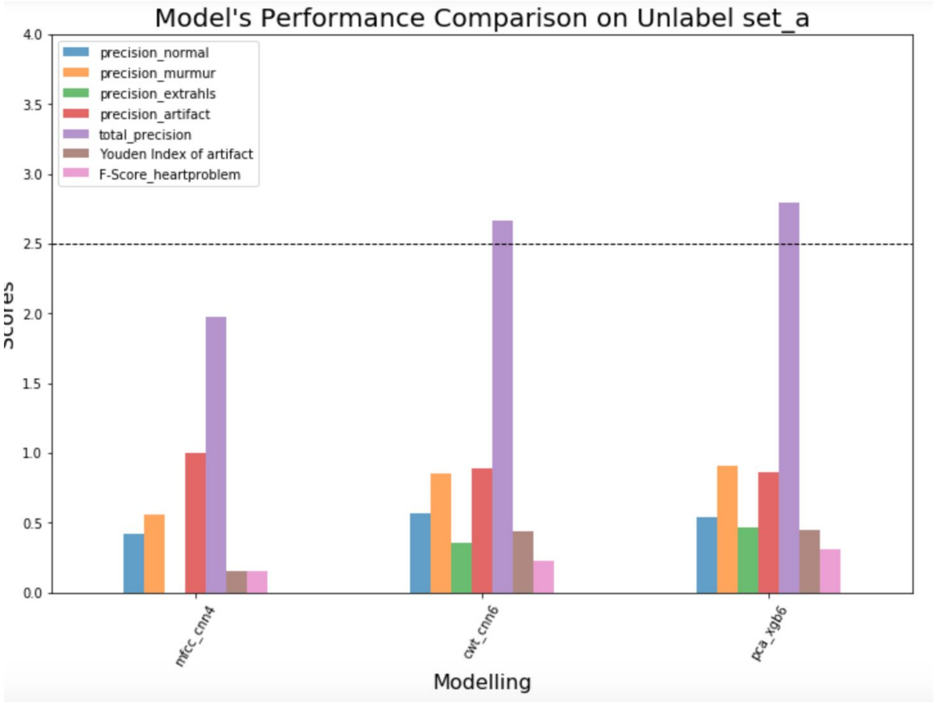
TEST set

from label data

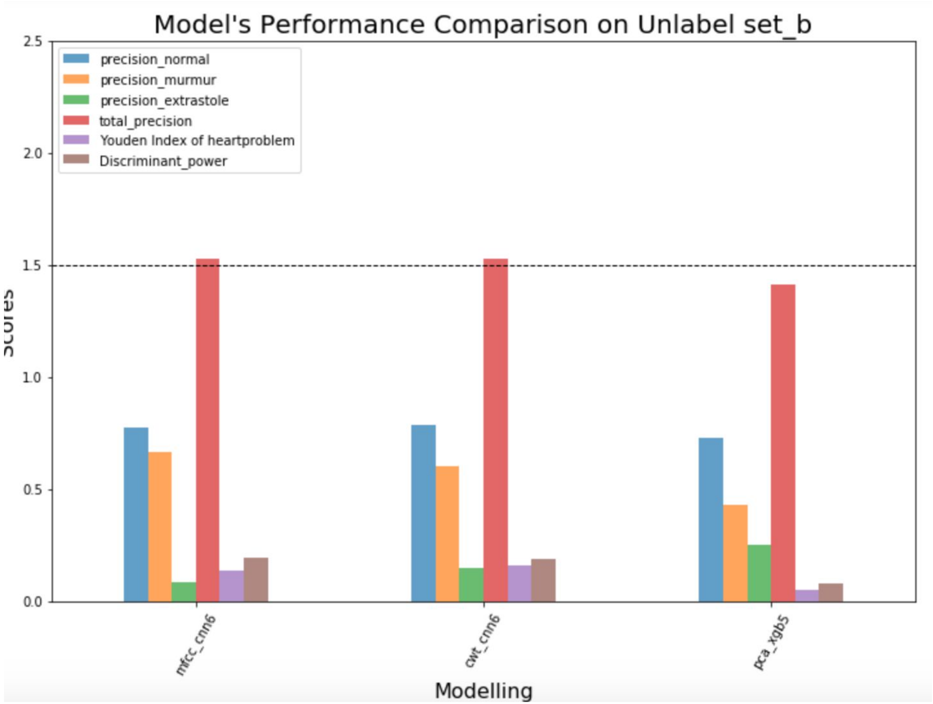


Results

set_a un-label data

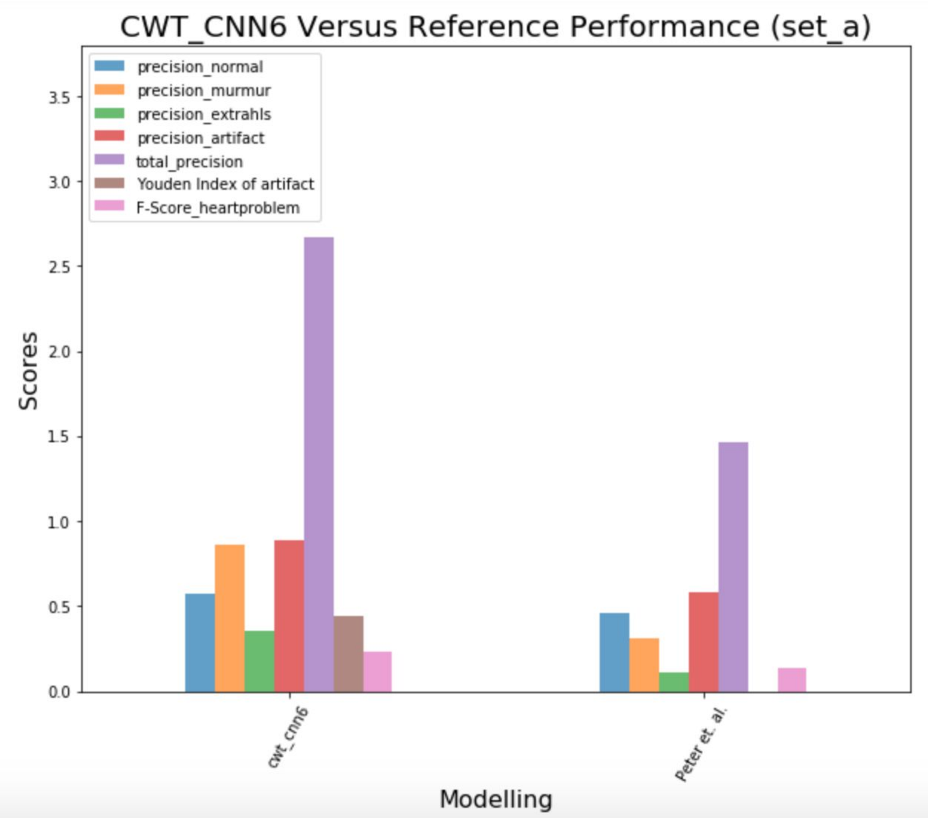


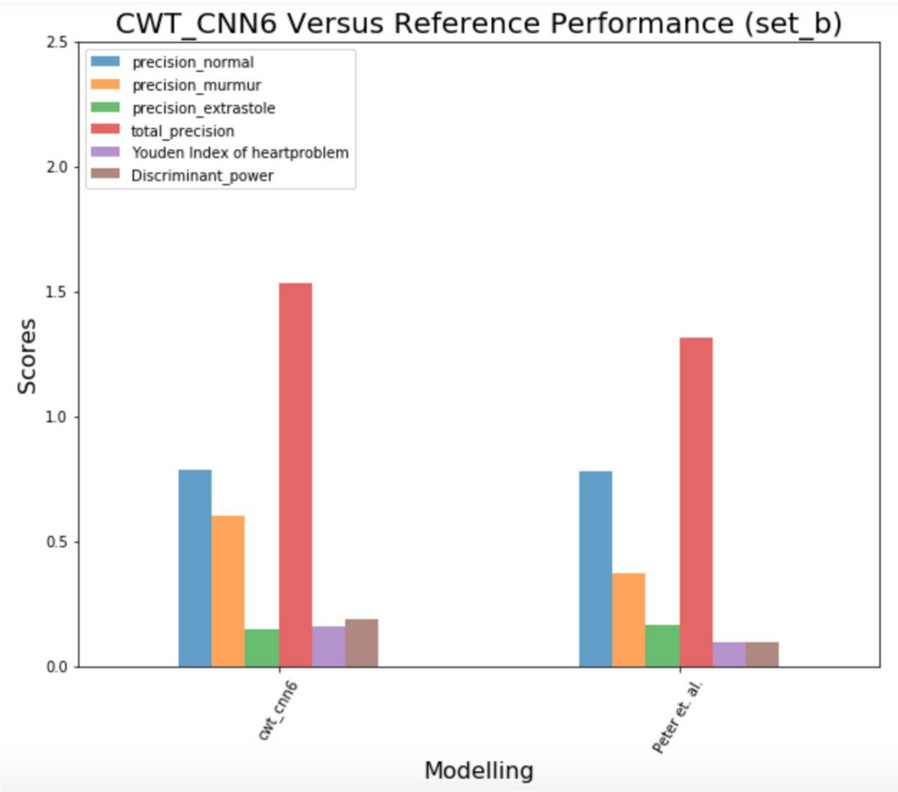
set_b un-label data



CWT_CNN6 appears in BOTH unlabel set_a and set_b!

Reference Performance





CWT_CNN6 perform better!

Summary and Recommendations

Proposed approaches

- **Imbalance class treatment**
- **Feature extraction using CWT**
- **Scalogram pattern detection using CNN**
- **Hyperparameter optimization using RandomizedSearchCV**

Classifier efficiency still need improvement!

- More data. Train dataset of 585 audio wav is simply too little
- Segmentation approach in preprocessing could be improve by first detect the position of heartbeat (S1, S2), segment it into smaller chunk by locating the beginning of the heartbeat and ensuring x amount of heartbeat cycles in it.
- Explore RNN (improve model efficiency in heartbeat classification) and XGBOOST (reduce computational time)

Thank You