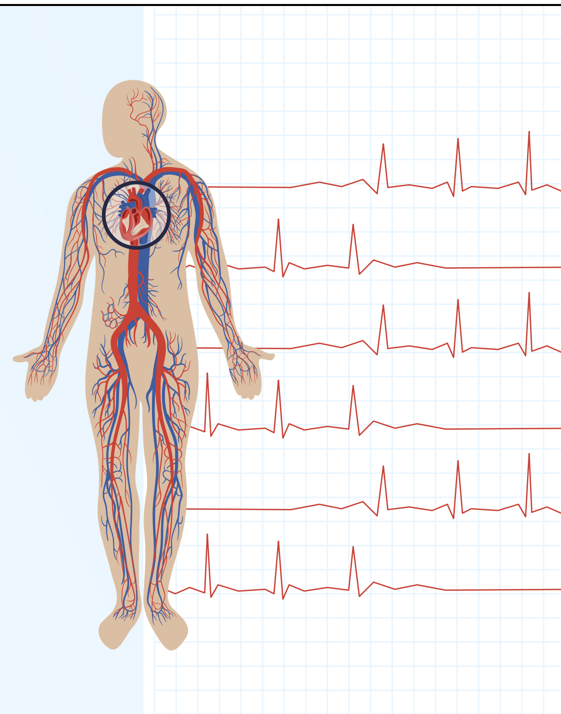
ECG Diagnostic Tool: A Machine Learning Approach



Daniel Kim, Trisha Sanghal, Zachary Fenton

I want to Thank you for joining us today. Along with Trisha and Daniel we will be presenting our exploration into an EKG Diagnostic tool utilizing a Machine learning approach.

Table of contents

First we will discuss the motivation to our project and the data that we used. Next we will look at the preprocessing and modeling of that data. After that we will discuss the iterations we went through with our model for optimization and finally our conclusions.

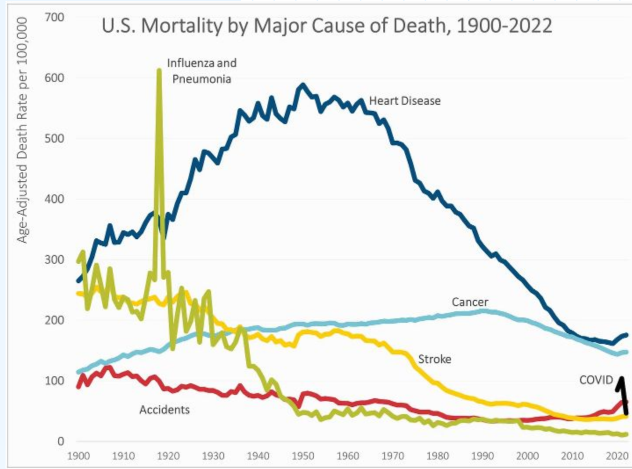
Motivation

Heart disease is the leading cause of death in America.

Electrocardiography (ECG; Pronounced 'Eee-Kay-Gee') is a painless, non-invasive diagnostic tool.

AI support systems for classifying ECGs could provide significant assistance; however, there are 2 major obstacles:

1. The lack of available datasets
2. A well trained model

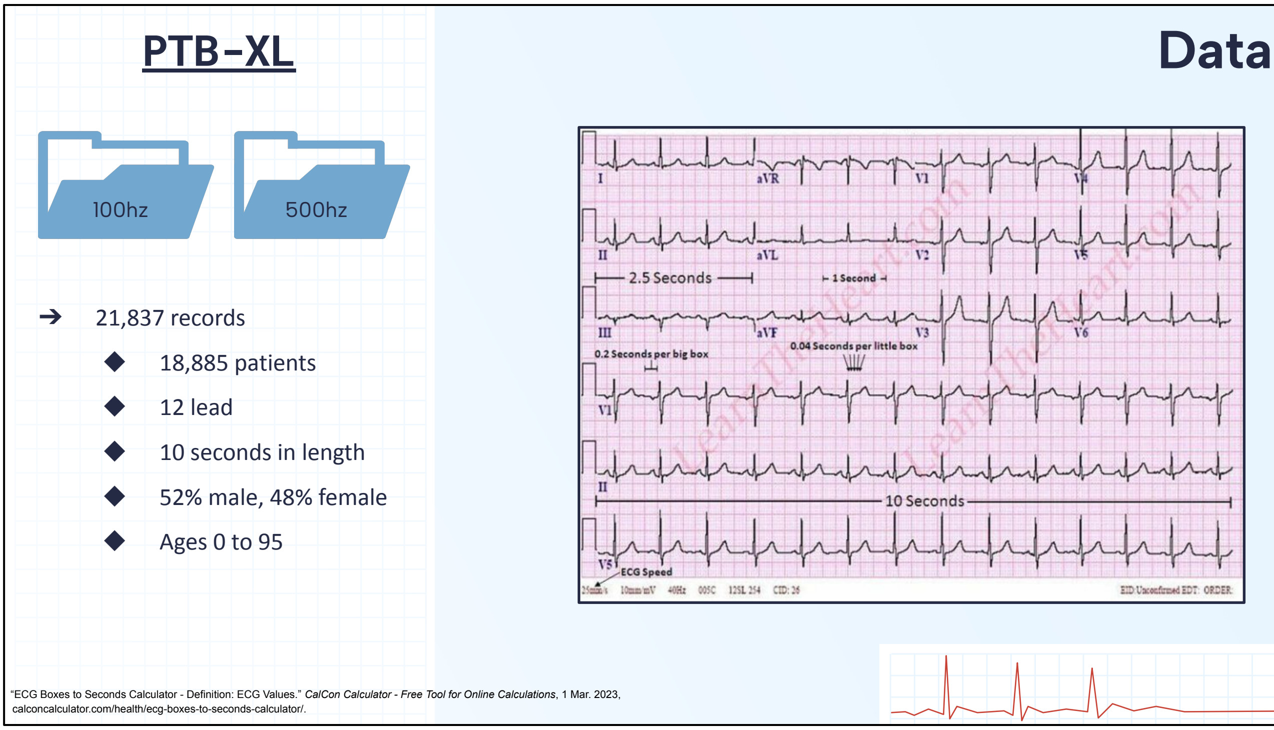


Campbell, Mary Pat. "The Shape of U.S. Mortality 2: 1900-2022." *The Shape of U.S. Mortality 2: 1900-2022*, STUMP - Meep on public finance, pensions, mortality and more, 15 Nov. 2023, marypatcampbell.substack.com/p/the-shape-of-us-mortality-2-1900.

According the National Center for Health Statistics, the leading cause of death in the United States is heart disease and has been for the past century. A simple, painless and non-invasive tool for diagnosing various forms of heart disease is Electrocardiography, or EKG. In spite of the fact that the EKG was first invented in the very early 1900's, the rate of diagnosing heart disease successfully the first time utilizing an EKG is less than 60% percent and in some cases, as low as 33%.

Employing an ML approach to provide support for classifying EKGs could provide significant improvements in first time success. Of course, historically there have been two major obstacles accomplishing this; 1- the lack of available datasets, and 2 - a well trained model.

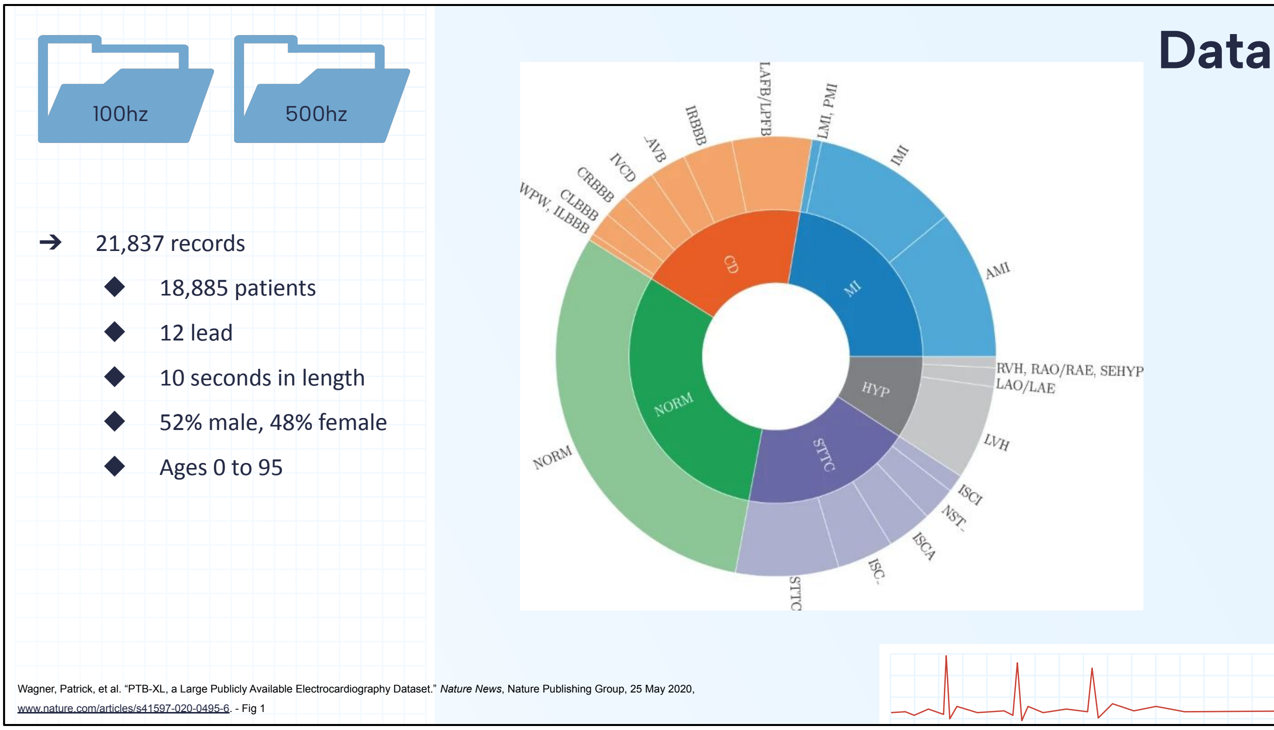
I would not be talking about his if both of those were still true today. So onto the data:



The dataset we used comes from Fee-zee-kah-lish Tek-NEE-shuh BOON-dess-ahn-shtahlt, or PTB-XL. It is the largest, publicly available 12-lead waveform EKG dataset.

The dataset is broken into a 100hz and a 500hz dataset. The 100hz is for convenience of use due to fewer datapoints from smaller frequency.

The data consists of 21,837 records collected from 18,885 patients. Each record is 10 seconds in length. The image here shows the output of a 10 second, 12 lead EKG.



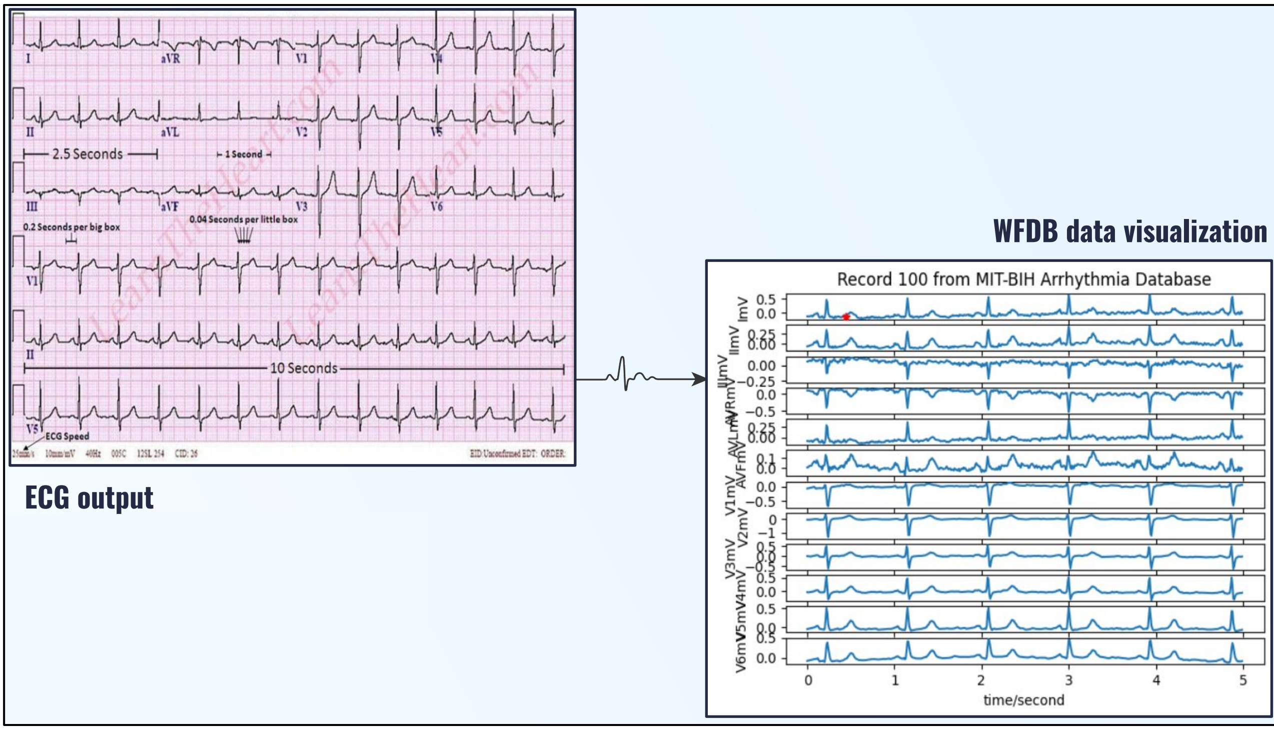
The data is broken into 5 Superclasses and 24 subclasses. For our model, we decided to utilize the superclasses. The superclasses are:

- NORM - which signifies normal rhythm
- CD - Conduction Disturbance
- MI - Myocardial Infarction
- HYP - Hypertrophy
- STTC - ST/T-Change (Supraventricular Tachycardia).

The data was collected from 1989 to 1996 utilizing devices from Schiller AG. The data was approved for release by The Institutional Ethics Committee after all determination of consent was approved and all PII was removed.

Each record has been annotated by at least 2 physicians. PTB-XL is considered a good dataset for ML based applications because it is not perfect.

The raw data is stored in a 16-bit binary format and transformed to a 1000 by 12 matrix utilizing the WFDB format (Waveform Database) publicly available package for waveform type data such as this.



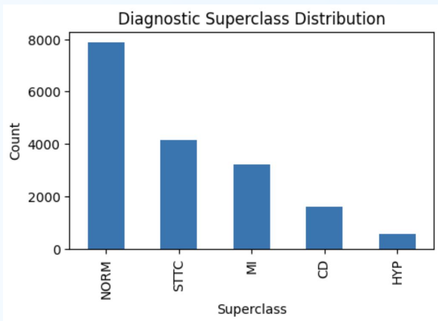
The package also provides a way to visualize the data, which can be seen here.

This shows a comparison of and EKG standard output compared to the visualization function of the WFDB package.

After the data is read from the raw form using the package, we began exploring preprocessing the data which I will now hand off to Trisha to speak on this.

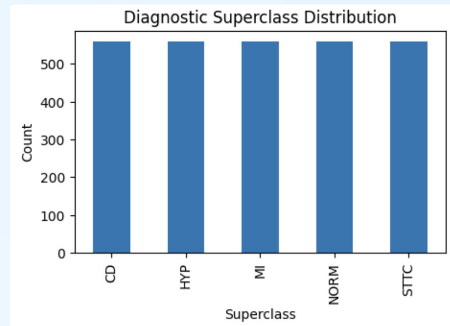
Preprocessing: Correct the Data Imbalance

Before



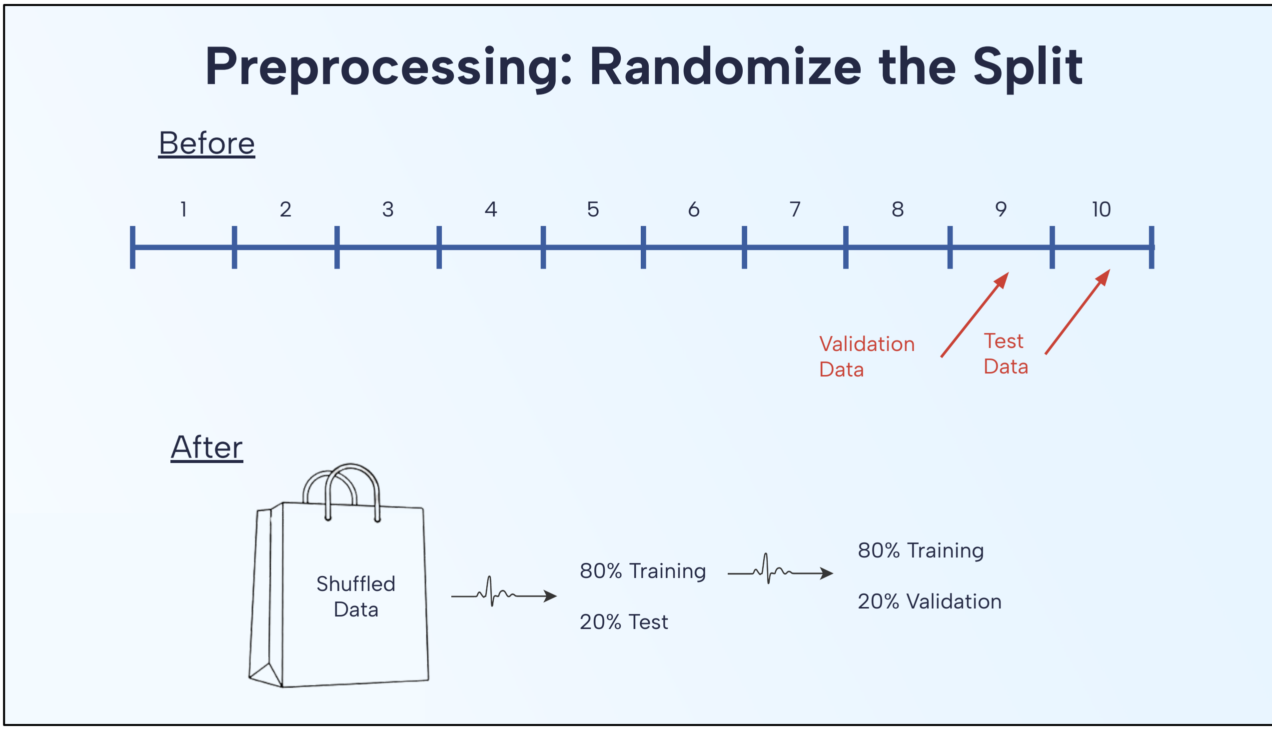
Total # of Training Samples:
17,439

After

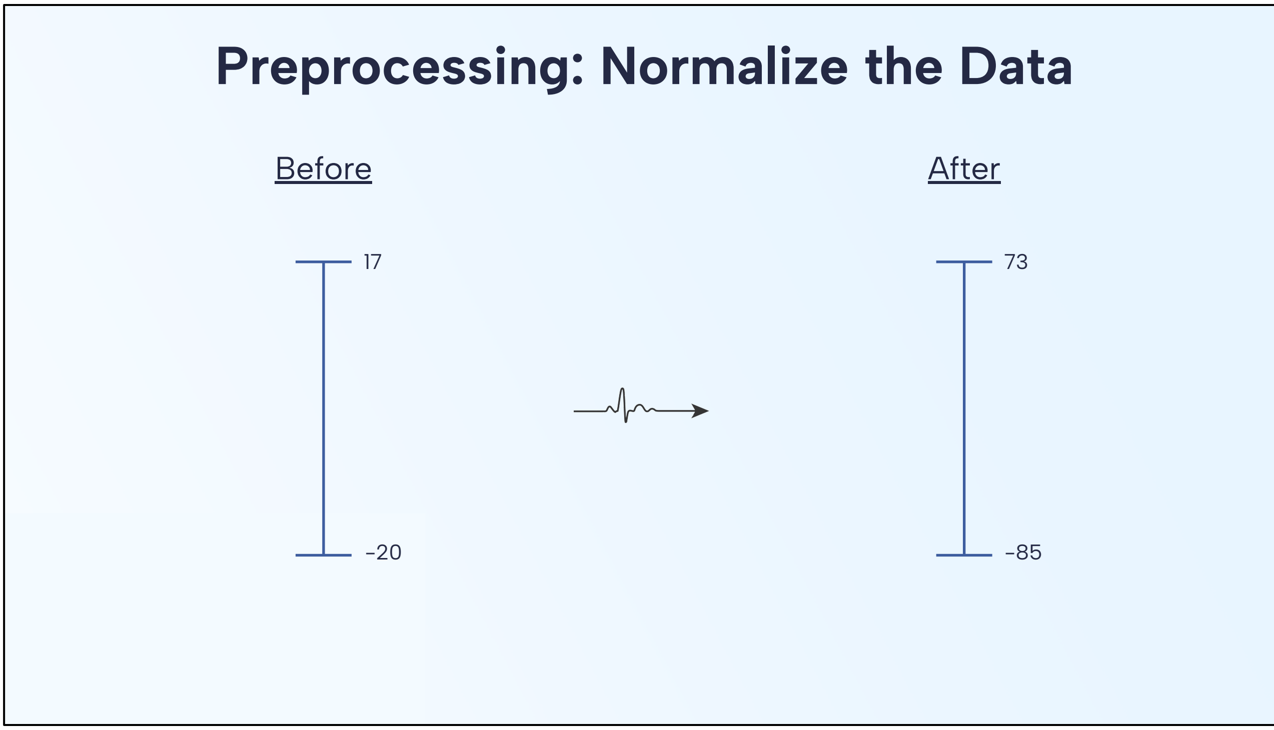


Total # of Training Samples:
2,800

One issue we hoped to address during our data preprocessing was the data imbalance across each class of EKG. Since we weren't working with image data, instead of applying some of the image data augmentation practices we learned, we instead tried to even the spread across EKG class by randomly sampling the same number of samples from each class. So, to be more specific we sampled *without replacement* from each class *n times* where n was the size of the smallest class. These plots show the distribution of samples across each class of EKG both before and after our data imbalance correction. This approach did drastically reduced the amount of data available for training, and so it did end up impacting our our validation accuracy somewhat, but we decided to include this step in our final model anyway to mitigate for potential bias.



Next in preprocessing - randomize the split. The folks who put together the dataset had already split the data into 10 sets, and said that the 9th and 10th sets had a "particularly high label quality" and should "be used as validation and test sets". But, we thought that including some of these higher quality labels in our training set might improve our model's performance. So, rather than using the data breakdown provided, we shuffled the data and then split it into training, validation, and test sets. However, because this approach did not impact our validation accuracy much, we opted not to include it in our final model.



The last thing we addressed when preprocessing our data was normalizing it, which did improve our validation accuracy and we ended up including this in our final model.

Modeling: Baseline Models

Predict at Random

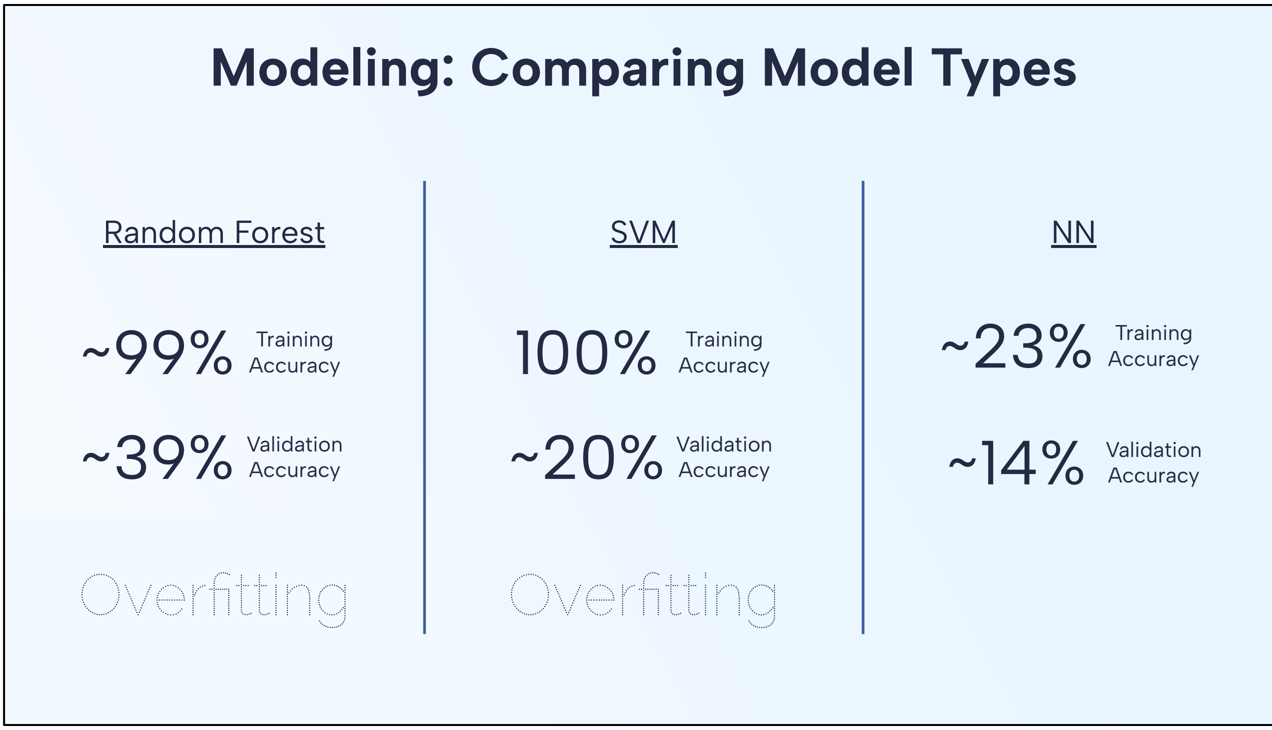
~20% Validation Accuracy

Always Predict Normal

~46% Validation Accuracy

Moving on to modeling. We started with two baseline models: (1) predicting a class at random, and (2) always predicting normal, which was our majority class. And hopefully these validation accuracies make intuitive sense - there were 5 classes, so predicting at random has a 1 in 5 chance of predicting correctly, resulting in the 20% validation accuracy. And, about 46% of our training data was in the “Normal” class, which means that 46% of the predictions will be correct.

Note: Baselines were established without running the data imbalance correction.



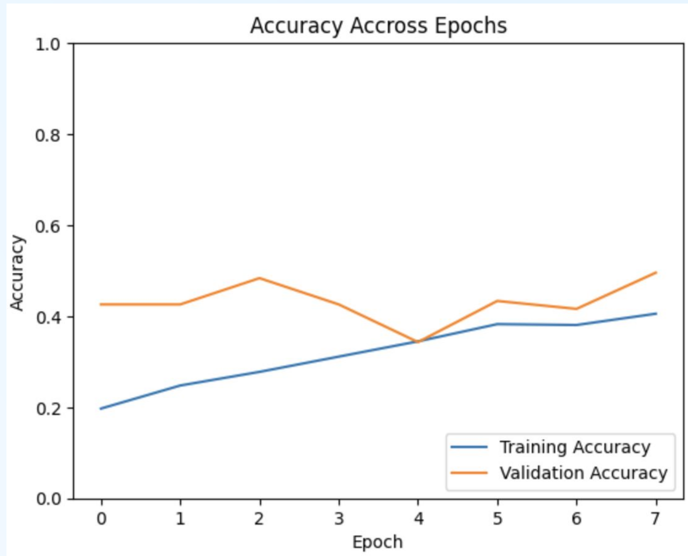
In addition to our baseline models, we also wanted to get a sense of how the neural network performed with respect to other commonly used multi-class classification model types. There are a lot of numbers on this slide, but we'll focus on a few interesting things. First, without any fine tuning of the models, none of these classifiers beat the validation accuracy of our previously established baseline of always predicting normal. In fact, our initial attempt at a neural network had the lowest validation accuracy of them all. However, the random forest and SVM both showed significant signs of overfitting to the training data (shown by the large difference in training and validation accuracies).

Note: These models were built after running the data imbalance correction.

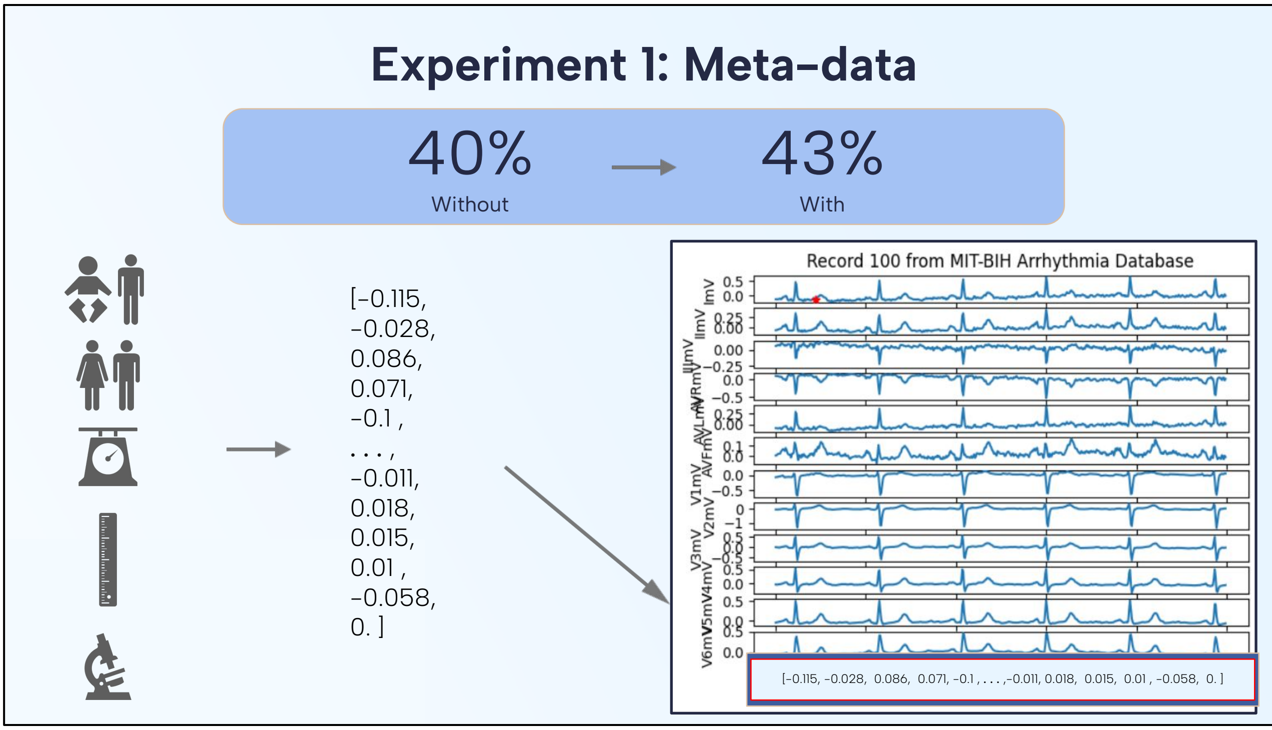
Modeling: What We Landed On

Layer (type)	Output Shape	Param #
conv1d_12 (Conv1D)	(None, 998, 64)	2368
max_pooling1d_12 (MaxPooling1D)	(None, 499, 64)	0
conv1d_13 (Conv1D)	(None, 497, 128)	24704
max_pooling1d_13 (MaxPooling1D)	(None, 248, 128)	0
conv1d_14 (Conv1D)	(None, 246, 256)	98560
max_pooling1d_14 (MaxPooling1D)	(None, 123, 256)	0
flatten_4 (Flatten)	(None, 31488)	0
dense_18 (Dense)	(None, 256)	8061184
dropout_10 (Dropout)	(None, 256)	0
dense_19 (Dense)	(None, 128)	32896
dropout_11 (Dropout)	(None, 128)	0
dense_20 (Dense)	(None, 64)	8256
dropout_12 (Dropout)	(None, 64)	0
dense_21 (Dense)	(None, 32)	2080
dropout_13 (Dropout)	(None, 32)	0
dense_22 (Dense)	(None, 16)	528
dropout_14 (Dropout)	(None, 16)	0
dense_23 (Dense)	(None, 12)	204

Total params: 8230780 (31.40 MB)
Trainable params: 8230780 (31.40 MB)
Non-trainable params: 0 (0.00 Byte)



We ended up moving forward with the neural network. By making some adjustments to the number and types of layers in our model (without doing too much experimentation), we were able to get a validation accuracy of about 40%. Next, Daniel will cover some of the experiments we ran to improve our neural network further.


Experiment 1: Meta-data
40%
Without
43%
With
[-0.115,
-0.028,
0.086,
0.071,
-0.1 ,
. . . ,
-0.011,
0.018,
0.015,
0.01 ,
-0.058,
0.]
Record 100 from MIT-BIH Arrhythmia Database
[-0.115, -0.028, 0.086, 0.071, -0.1 , . . . ,-0.011, 0.018, 0.015, 0.01 , -0.058, 0.]

Experiment 2: Manual Hyperparameter Optimization

43% → 43%

Before → Best Observed

# Conv layers	Conv filters	Dense units	Pool Size
1	[64]	[256, 128, 64, 32, 16]	2
2	[64, 64]	[256, 128, 64, 32, 16]	2
3	[64, 64, 64]	[256, 128, 64, 32, 16]	2
4	[64, 64, 64, 64]	[256, 128, 64, 32, 16]	2
5	[64, 64, 64, 64, 64]	[256, 128, 64, 32, 16]	2
3	[64, 128, 256]	[256, 128, 64, 32, 16]	2
3	[128, 128, 128]	[256, 128, 64, 32, 16]	2
3	[256, 256, 256]	[256, 128, 64, 32, 16]	2
3	[256, 128, 64]	[256, 128, 64, 32, 16]	2
3	[128, 128, 128]	[256, 128, 64, 32, 16]	1
3	[256, 256, 256]	[256, 128, 64, 32, 16]	2
3	[256, 128, 64]	[256, 128, 64, 32, 16]	3
3	[256, 256, 256]	[64]	1
3	[256, 256, 256]	[64, 64]	1
3	[256, 256, 256]	[64, 64, 64]	1
3	[256, 256, 256]	[64, 64, 64, 64]	1
3	[256, 256, 256]	[64, 64, 64, 64, 64]	1

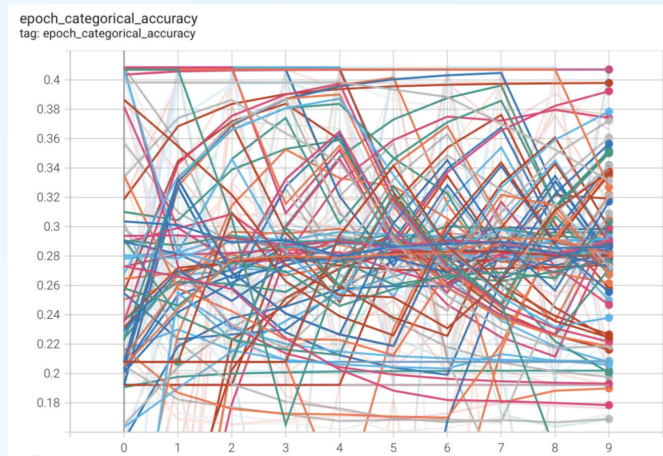
Experiment 3: Keras Tuner Optimization

43% Before → **47%** Best Model

50% Test Accuracy

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 1000, 13)]	0
conv1d (Conv1D)	(None, 998, 224)	8960
max_pooling1d (MaxPooling1D)	(None, 499, 224)	0
conv1d_1 (Conv1D)	(None, 497, 224)	150752
max_pooling1d_1 (MaxPooling1D)	(None, 248, 224)	0
conv1d_2 (Conv1D)	(None, 246, 256)	172288
max_pooling1d_2 (MaxPooling1D)	(None, 123, 256)	0
conv1d_3 (Conv1D)	(None, 121, 256)	196864
max_pooling1d_3 (MaxPooling1D)	(None, 60, 256)	0
conv1d_4 (Conv1D)	(None, 58, 256)	196864
max_pooling1d_4 (MaxPooling1D)	(None, 29, 256)	0
flatten (Flatten)	(None, 7424)	0
dropout (Dropout)	(None, 7424)	0
dropout_1 (Dropout)	(None, 7424)	0
dense (Dense)	(None, 32)	237600
dense_1 (Dense)	(None, 32)	1056
dense_2 (Dense)	(None, 32)	1056
dense_3 (Dense)	(None, 32)	1056
dense_4 (Dense)	(None, 32)	1056
dense_5 (Dense)	(None, 32)	1056
dense_6 (Dense)	(None, 12)	396

Total params: 969,004
Trainable params: 969,004
Non-trainable params: 0



Conclusions

Model

Metric	Value	Conclusion
Test Accuracy	50%	✗
Validation Accuracy	47%	✗
F1	0.59	~
Precision	0.5	~
Recall	0.7	~

Implications

- Without training: medical students have a 42% accuracy while residents have a 55.8% accuracy.[5]
- This implies that this initial model may perform better than your average medical student if you want to get your EKG looked at (not recommended)
- Some potential negative societal impacts include inequity due to biased datasets for certain classes (sex, race, etc.)

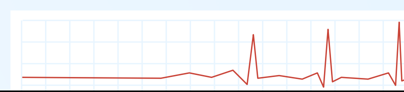
Next Steps

- Continue to optimize data pre-processing
- Re-encode using all relevant metadata
- Scope different model types (RNN, MLP, Hybrid)
- Continue to optimize model hyperparameters (try Adamax loss function)

Contributions

	Daniel	**Trisha**	**Zach**
Code	Hyperparameter optimization	Data preprocessing, baseline modeling	Data loading, Model development
Slides/Presentation	• Experiments • Conclusion	• Preprocessing • Modeling	• Motivation • Data
Research	• Tensorflow hyperparameter optimization automation • ECG studies	• Supervised vs Unsupervised models	• Initial data set • ECG research

– All members contributed to NeurIPS verification with equal effort.



References

1. *Wagner, Patrick, et al. "PTB-XL, a large publicly available electrocardiography dataset." Scientific Data, vol. 7, no. 1, 25 May 2020, https://doi.org/10.1038/s41597-020-0495-6.*
2. *Wagner, P., Strodthoff, N., Bousseljot, R., Samek, W., & Schaeffter, T. (2022). PTB-XL, a large publicly available electrocardiography dataset (version 1.0.3). PhysioNet. https://doi.org/10.13026/kfzx-aw45.*
3. Goldberger, A., et al. "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. e215–e220." (2000).
4. Wagner, Patrick, et al. "PTB-XL, a Large Publicly Available Electrocardiography Dataset." *Nature News*, Nature Publishing Group, 25 May 2020, www.nature.com/articles/s41597-020-0495-6.
5. Cook DA, Oh SY, Pusic MV. Accuracy of Physicians' Electrocardiogram Interpretations: A Systematic Review and Meta-analysis. *JAMA Intern Med*. 2020;180(11):1461-1471. doi:10.1001/jamainternmed.2020.3989

Data License:
https://physionet.org/content/ptb-xl/view-license/1.0.3/

GitHub:
https://github.com/zfenton/UCBMIDS_W207_finalProject

NeurIPS Checklist

❏ 1. For all authors...
- ✓ (a) Do the main claims made in the abstract and introduction accurately reflect the papers contributions and scope? **- N/A**
- ✓ (b) Have you read the ethics review guidelines and ensured that your paper conforms to them? **- YES**
- ✓ (c) Did you discuss any potential negative societal impacts of your work? **- YES**
- ✓ (d) Did you describe the limitations of your work? **- YES**

❏ 2. If you are including Theoretical Results…
- ✓ (a) Did you state the full set of assumptions of all theoretical results? **- N/A**
- ✓ (b) Did you include complete proofs of all theoretical results? **- N/A**

❏ 3. If you ran experiments…
- ✓ (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? **- YES**
- ✓ (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? **- YES**
- ✓ (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? - **YES** - in github
- ✓ (d) Did you include the amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? **- YES**

❏ 4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets…
- ✓ (a) If your work uses existing assets, did you cite the creators? **- YES**
- ✓ (b) Did you mention the license of the assets? **- YES**
- ✓ (c) Did you include any new assets either in the supplemental material or as a URL? **- YES**
- ✓ (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? **- YES**
- ✓ (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? **- YES**

✓ 5. If you used crowdsourcing or conducted research with human subjects…
- ✓ (a) Did you include the full text of instructions given to participants and screenshots, if applicable? **- N/A**
- ✓ (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? **- N/A**
- ✓ (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? **- N/A**