Lab Report 4

Prepared by: Heidi Eren

Date: 11/23/2025

Executive summary:

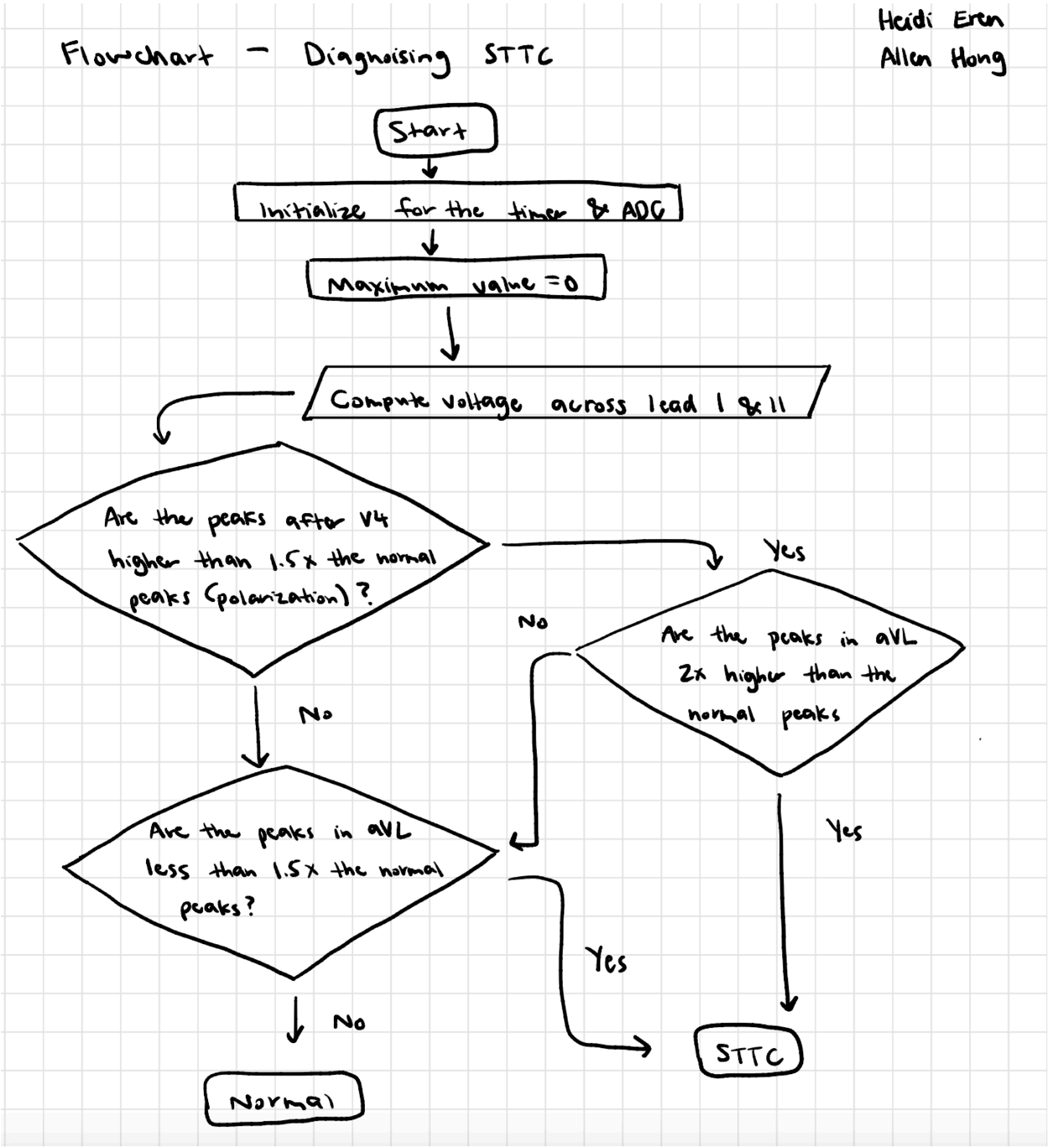
Describe what you are trying to accomplish/demonstrate. What is the purpose of this test? (**5 pts)**

The purpose of this experiment is to demonstrate an understanding of electrocardiogram signal processing through supervised learning of a machine learning model (CNN) with heart rate readings to diagnose varying conditions. By designing a flowchart, the structure of the algorithm to diagnose such a condition can be determined. In Python, a convolutional neural network can be supplied with labeled ECG data that is split between a training set and testing set to identify and classify features accordingly. Using a confusion matrix and outputting descriptive statistics, the accuracy of the model can be assessed across different combinations of test-train sets, number of CNN filters, number of epochs, and various hyperparameters to evaluate the best performing model with the least amount of time.

Procedure:

1. **On paper,** develop an algorithmic flowchart to diagnose the condition you've been assigned. **(15 pts)**

Condition: STTC



2. Cognitively walk through the algorithm of the team across from you. How accurate is it? **(5 pts)**

The algorithm utilizes quantitative metrics to compare the peaks of each wave, which was helpful in clearly making distinctions before proceeding to the next condition or decision. For example, the algorithm is specified to examine an elevated ST or depressed ST (inverted T wave) and whether it crosses a threshold (number of boxes in mV) to distinguish between a normal case and a diagnosis of the condition. However, some terms were vague (e.g. " Any abnormalities present without the MI pattern") so I think additional clarity through a quantitative or qualitative distinction would be helpful for all flowchart steps. In terms of accuracy, 4 of the ECG images were correct using this algorithm, while 2 of the ECG images were not correct in identifying the condition (STTC). Of the ones that were correct, the normal ECG graph was correctly identified using the algorithm.

3. In python, use descriptive statistics to identify (You can write, list, or show plots below):
    a. The number of cases of each diagnosis **(2 pts)**
    b. The average and standard deviation of age at each diagnosis **(3 pts)**
    c. How many men and women receive each diagnosis **(3 pts)**
    d. Select one additional piece of metadata you think may be relevant. Analyze by diagnosis **(3 pts)**.

See the descriptive statistics below for each diagnosis **(*****this is run for cases where the only diagnosis was itself, with overlapping cases excluded completely)**:

	NORM	MI	CD	STTC	HYP
Number of cases	9069	2532	1708	2400	535
Average of age	52.816959	65.880332	69.154567	68.812083	62.568224
Standard deviation of age	22.309173	24.817905	39.020620	34.599731	23.737376
Sex distribution	4119 (F) 4950 (M)	1602 (F) 930 (M)	1045 (F) 663 (M)	1019 (F) 1381 (M)	302 (F) 233 (M)
Weight distribution (average, std)	71.242135, 15.204256	74.515913, 19.169375	71.812500, 14.875458	70.967196, 16.818258	70.874552, 14.345757

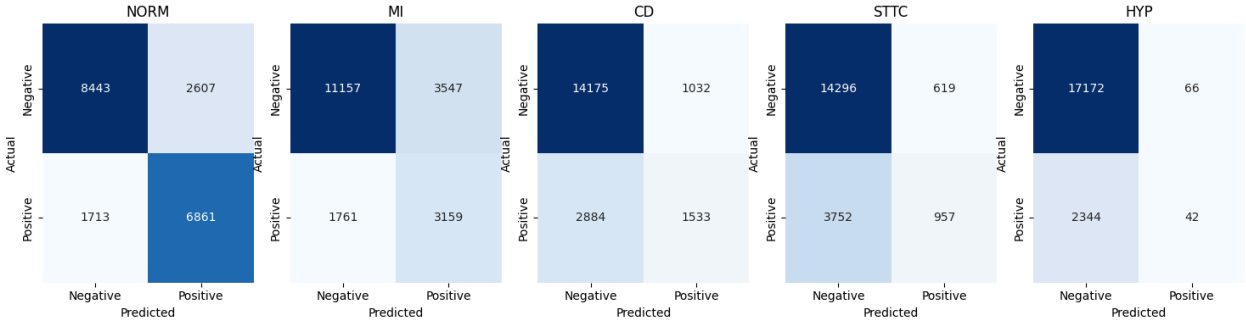
4. Use the ML algorithm to classify independent ECG recordings into one of five categories: “Normal”, “Myocardial infarction” “Conduction disturbance” “ST/T change”, and “Hypertrophy”.

5. Examine the algorithm under 6 different conditions:
    a. 10% train/90% test
    b. 50% train/50% test
    c. 90% train/10% test
    d. 50% train/50% test; 32 epochs
    e. 50% train/50% test; 64 filters in both CNNs
    f. Pick a combination that you think will achieve the greatest accuracy in the least amount of time. Justify the choice **(3 pts)**.

Based on the previous combinations, it was evident that having a smaller training set decreased the accuracy of the model across all diagnoses. Increasing the number of epochs slowed the processing time, but did not change that drastically in accuracy nor precision. Changing the number of filters in the first ReLU activation CNN to 64 filters did increase the accuracy across each of the models, as this hyperparameter can help in capturing more complex features for each diagnosis. Hence, I chose a **90% train - 10% test with 64 filters** in both CNNs with **one epoch** to achieve the greatest accuracy while being time conscious (~1-2 min). The accuracy was significantly higher across all diagnoses (>80%), as well as the specificity and sensitivity metrics. (see below)

6. For each of the above, report out the confusion matrix, the overall accuracy, the sensitivity, and the specificity for each diagnosis (**1 pt each for accuracy, sensitivity, specificity for each case**).

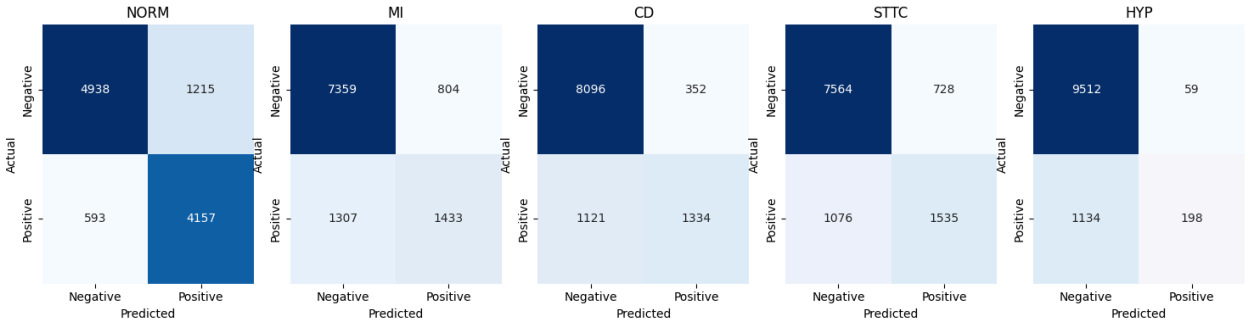
10% train/90% test



Overall Model Metrics
Precision: 0.6853
Recall: 0.3771
binary_accuracy: 0.7970
loss: 0.4463

Diagnosis	Sensitivity	Specificity	Accuracy
NORM	0.8002	0.7641	0.7799
MI	0.6421	0.7588	0.7295
CD	0.3471	0.9321	0.8004
STTC	0.2032	0.9585	0.7773
HYP	0.0176	0.9962	0.8772

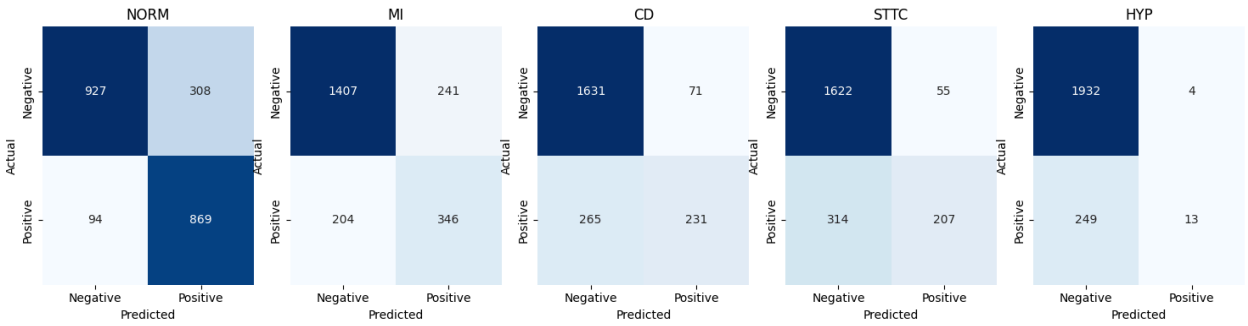
50% train/50% test



Overall Model Metrics
Precision: 0.7560
Recall: 0.5860
binary_accuracy: 0.8463
loss: 0.3598

Diagnosis	Sensitivity	Specificity	Accuracy
NORM	0.8752	0.8025	0.8342
MI	0.5230	0.9015	0.8064
CD	0.5434	0.9583	0.8649
STTC	0.5879	0.9122	0.8345
HYP	0.1486	0.9938	0.8906

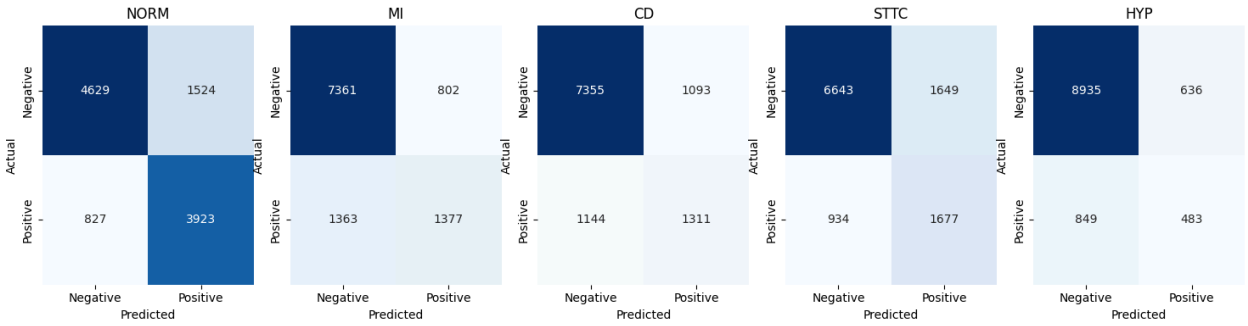
90% train/10% test



Overall Model Metrics
Precision: 0.7388
Recall: 0.5419
binary_accuracy: 0.8349
loss: 0.3894

Diagnosis	Sensitivity	Specificity	Accuracy
NORM	0.9024	0.7506	0.8171
MI	0.6291	0.8538	0.7975
CD	0.4657	0.9583	0.8471
STTC	0.3973	0.9672	0.8321
HYP	0.0496	0.9979	0.8849

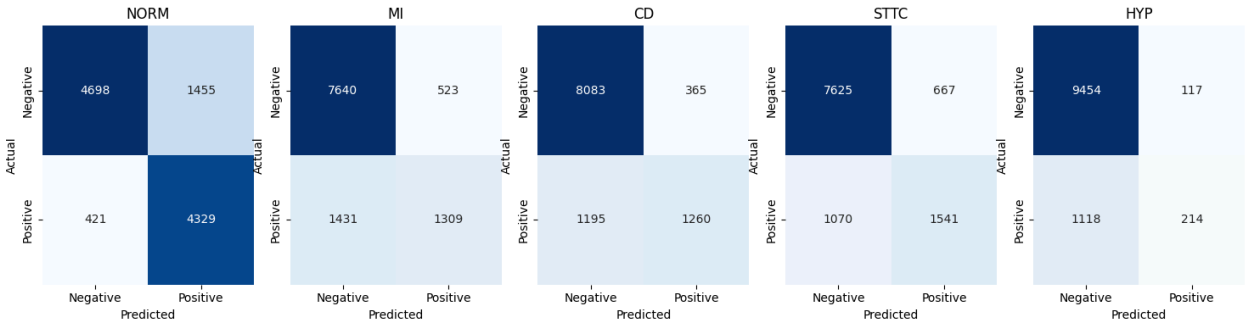
50% train/50% test; 32 epochs



Overall Model Metrics
Precision: 0.6211
Recall: 0.5926
binary_accuracy: 0.8041
loss: 4.0037

Diagnosis	Sensitivity	Specificity	Accuracy
NORM	0.8259	0.7523	0.7844
MI	0.5026	0.9018	0.8014
CD	0.5340	0.8706	0.7948
STTC	0.6423	0.8011	0.7631
HYP	0.3626	0.9335	0.8638

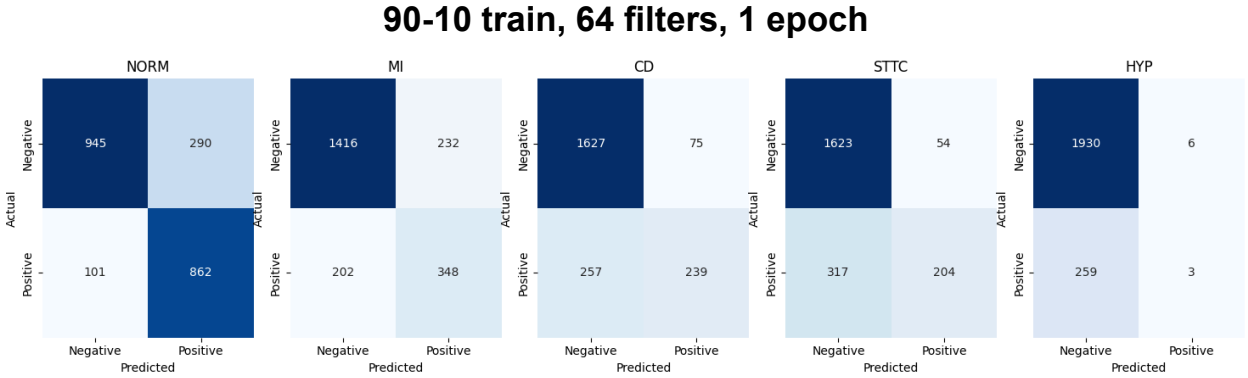
50% train/50% test; 64 filters in both CNNs



Overall Model Metrics
Precision: 0.7466
Recall: 0.5977
binary_accuracy: 0.8458
loss: 0.3652

Diagnosis	Sensitivity	Specificity	Accuracy
NORM	0.9114	0.7635	0.8279
MI	0.4777	0.9359	0.8208
CD	0.5132	0.9568	0.8569
STTC	0.5902	0.9196	0.8407
HYP	0.1607	0.9878	0.8867

Custom combination:



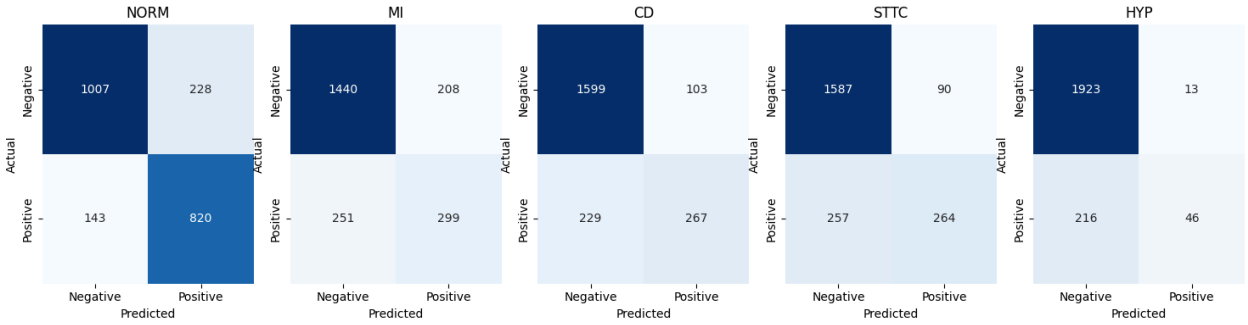
Overall Model Metrics
Precision: 0.7289
Recall: 0.5681
binary_accuracy: 0.8366
loss: 0.3832

Diagnosis	Sensitivity	Specificity	Accuracy
NORM	0.8951	0.7652	0.8221
MI	0.6327	0.8592	0.8025
CD	0.4819	0.9559	0.8490
STTC	0.3916	0.9678	0.8312
HYP	0.0115	0.9969	0.8794

7. Based on the descriptive statistics above, select the item of metadata that you think will improve accuracy the most and add it to your ML input. Justify this choice. **(3 pts for justification, 1 pt each for accuracy, sensitivity, specificity)**

I chose “sex” as the metadata to improve the accuracy of the ML model. This makes sense because across all the diagnoses, there was a distinct correlation between the sex of the patient being diagnosed. For patients diagnosed with myocardial infarction, conduction disturbance, and hypertrophy, there were a higher number of cases that were females than males. For the other diagnoses, there were a higher number of males than females. By including this metadata in the ML model, the model can use it as an additional feature to better understand distinctions between the diagnoses and interpret patterns more easily, improving the overall accuracy of the model.

90% train-10% test, 64 filters, 1 epoch, with “sex” metadata



Overall Model Metrics
Precision: 0.7433
Recall: 0.5476
binary_accuracy: 0.8370
loss: 0.3718

Diagnosis	Sensitivity	Specificity	Accuracy
NORM	0.8515	0.8154	0.8312
MI	0.5436	0.8738	0.7912
CD	0.5383	0.9395	0.8490
STTC	0.5067	0.9463	0.8421
HYP	0.1756	0.9933	0.8958

The above metrics demonstrate how although the “sex” metadata was not as significant in improving all aspects, the accuracy did improve from 87.9% to 89.5% when comparing to the 90-10 train with 1 epoch model. For this model, the sensitivity is not as low, and the specificity is not as high. Although this is not a huge improvement, it is enough of a difference that the model is improved by a small margin for the overall accuracy of the model. The binary accuracy also improved from 83.66% to 83.70%. Hence, by incorporating and training the model with the “sex” metadata, the model was able to better associate patterns between the diagnoses and make slightly more accurate predictions on the testing set.

8. **Bonus (1 pt each):** Test the sensitivity of the model to at least 3 additional items of metadata (alone or in combination). What metadata had the largest impact on model accuracy? Was this surprising or unsurprising?

Based on the table below, by evaluating a combination of sex, weight, and age metadata, the accuracy of the model diminished significantly. The sensitivity of the normal diagnoses was 1, which means that the metadata combination maximized the detection of true positive cases, or

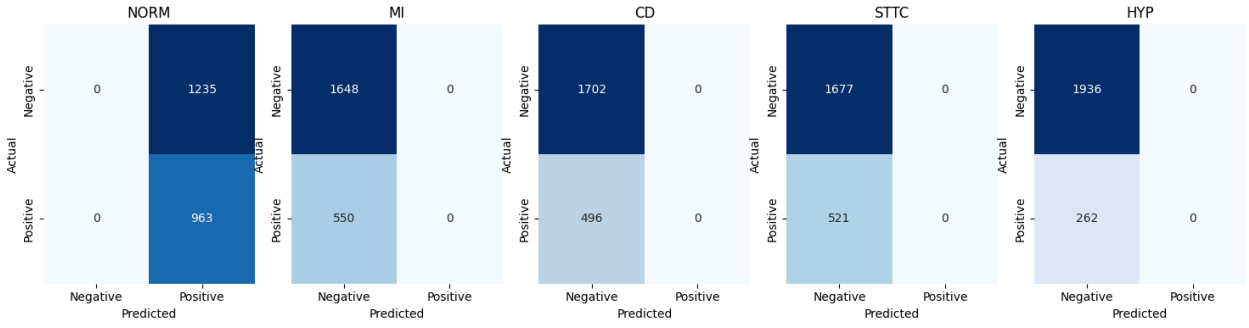
minimized false negatives. When combined, the model accuracy was marginally lower (by 1%), which is consistent with what is expected when sex, weight, and age may lean more towards arbitrary features that do not correlate with a specific diagnosis. In addition, the specificity and sensitivity resulting in binary values of 1 and 0 respectively convey that adding more metadata to the model does not improve nor greatly hinder the predictions made from the model.

Between the three items, the “sex” metadata had the largest impact on model accuracy as shown from the previous question. The accuracy was almost all ~80% and above for all diagnoses, which is surprising but consistent based on the descriptive statistics table that conveys how there may be a correlation between diagnosis and sex.

Sex, weight, age metadata combined

Diagnosis	Sensitivity	Specificity	Accuracy
NORM	1	0	0.4381
MI	0	1	0.7498
CD	0	1	0.7743
STTC	0	1	0.7630
HYP	0	1	0.8808

Overall Model Metrics
Precision: 0.0000
Recall: 0.0000
binary_accuracy: 0.7460
loss: 0.5455



Notes and recommendations:

1. Discuss the pros and cons of “traditional” signal processing techniques vs. machine learning. List three medical applications where traditional signal processing makes more sense vs. three where machine learning makes more sense. Justify. **(12 pts - 2 each)**

   Traditional signal processing is simple to implement, requires lower processing requirements, and does not require a prior dataset. However, it can be more complicated in computing higher order algorithms and the patterns must be known in advance. For medical applications, it can be best for low dimensional datasets, real time classification, and can be used when known patterns help underlie the data.

   One example is interpreting ECG signals. Traditional signal processing techniques can help in classifying such conditions and patterns with filtering and feature extraction, and since existing patterns are known and the dataset is smaller, there is less computational demand required for such analysis.

   Another example is photoplethysmography (pulse oximetry). By processing blood oxygen saturation, traditional signal processing can compute heart rate through optical detection. By pre-processing with noise reduction from light or motion interference, signal processing with known algorithms such as FFT can compute accurate SPO2 calculations. This is suitable for traditional signal processing because the lower dimensional dataset with known patterns can detect and calculate oxygen saturation more accurately.

   One last example is glucose monitors. Traditional signal processing can filter raw electrical or optical sensor data and use linear regression to predict blood glucose levels. Given its lower complexity in processing and known patterns for blood glucose levels, signal processing is ideal for this application to simply convert sensor signals to a reliable and accurate blood glucose reading.

   For machine learning, the ability to recognize patterns, process higher-order algorithms, and combine multiple models can lead to accurate and rapid processing. However, these models are less transparent in the output and require large, labelled datasets across multiple devices. There are many requirements and tradeoffs for what the training data can be as it can create bias in the model. Another issue with machine learning models is that they can easily underfit or overfit the data, capturing features in ways that are not optimal. For medical applications, machine learning models work best for high dimensional datasets with unknown or variable data patterns, and can be used for diagnose and higher order processing.

One example is Nabla, an AI notetaking for clinicians. By having AI help create medical notes for patients, it can ease the lives of clinicians when they may not have sufficient time to create very elaborate or detailed notes during sessions. Since there is a large pool of digital data pertaining to clinician notes that have accumulated over decades, it can be easily implemented and trained with a machine learning model to recognize patterns in writing based on context. In addition, because AI notetaking does not directly diagnose or recognize a condition for patients, there is no legal liability that is tied to the resource.

Another example is Brainomix, an AI imaging model for stroke and lung fibrosis. Given the high-dimensional, large, and labelled datasets available from countless published papers that distinguish such conditions in images, a machine learning model can easily recognize patterns and perform more advanced image processing analysis that traditional signal processing cannot do. Image recognition and segmentation can be automated, making it ideal for machine learning.

One last example is Whoop, a fitness and health wearable device that uses machine learning to analyze biometric data from users to create personalized health and fitness insights. Because the watch is continuously collecting data including heart rate and sleep cycles, the large high dimensional dataset that the model is trained on helps to rapidly identify patterns and provide suggestions for the user to optimize their health and wellbeing. This analysis may not be easily done with traditional signal processing methods.

2. Describe the setup of the ML algorithm in detail. **(4 pts)**

   The algorithm first loads the metadata (filename, sex, age, etc) into a dataframe Y. Then each ECG dataset is loaded as another dataframe X. This dataframe is then split into a training set and testing set based on the inputted strat_fold variable. Each diagnosis is also converted into a binary array and is set as a dataframe Z. As an optional step, the metadata can be included by adding an array and transposing onto the dataframe X. Then, a CNN model is implemented with the first layer as 32 filters with ReLU activation, the second layer as 64 filters with RELU, and the final layer with sigmoid activation. The filtering reduces the dimensionality of the data and the sigmoid activation is the classification. After fitting the model, the accuracy score can be outputted. The model is then predicted on the testing data set to assess the accuracy on an unknown data set, and this accuracy is outputted as a confusion matrix. Additional metrics including sensitivity, specificity, and accuracy is computed from this confusion matrix and saved as a dataframe.

3. How was the accuracy of the ML algorithm? Could it continue to be improved? Why or why not? **(4 pts)**

I think the accuracy of the ML algorithm was moderately well, as some combinations of hyperparameters achieved over 80% accuracy across all diagnoses (such as the 50-50% 64 filters). Given the large dataset with extensive metadata, the ML algorithm was easily able to be trained on, and since we were able to optimize it by testing different combinations of test-train splits, number of filters, number of epochs, and inclusion of metadata, there were many ways in which the accuracy was refined. I think the model could still be improved by evaluating alternative activation functions for each layer in CNN, such as Tanh or sigmoid, to capture the nonlinearity relationship between features. We could also include an additional step of normalization for greater stability and convergence in the loss function while reducing overfitting. We could also add another convolutional layer to make more meaningful patterns between features of the ECG data.

4. How did the inclusion of metadata affect the accuracy of the neural network? Was this surprising? Why or why not? **(5 pts)**

   The inclusion of specific metadata did in fact change the accuracy of the neural network. When the “sex” metadata was added to the model, the overall accuracy improved by ~2% across all diagnoses. Since this provided more context and structure given that there was a correlation between sex and specific diagnoses, the model was trained better to perform with greater accuracy and quality. Hence, the metadata provides additional context for the model to make better predictions at diagnosis.

   What was surprising was that after adding 3 items to the model, the precision of the model decreased to nearly 0. This is surprising because it shows that not all metadata is significant in correlating the specific diagnoses with the corresponding ECG signals. Hence, when combining less related metadata such as weight and age, the model may not be able to make as reliable of predictions with this highly variable information for each diagnosis. When looking at the descriptive statistics, the variability in these categories was very high, which could convey the lack of correlation between the category and the specific disease diagnosed. Hence, this metadata would not improve, but rather, decrease the accuracy and precision of the model’s predictions.

5. Examine your accuracy/sensitivity/specificity data. Which one is the most important for this application? For which diagnoses is the model most accurate? Is this surprising? **(6 pts)**

   The accuracy is the overall number of true positives and true negatives of the model in making predictions. The sensitivity is the number of times a positive result is truly positive. The specificity is the number of times a negative result is truly negative. For this application, sensitivity would be of the most importance. Having high sensitivity is critical because if a positive case was not correctly identified as a condition, then this would be

a serious concern since the patient would be misdiagnosed and unknowingly be putting their life on the line without proper information. Although accuracy is equally important in knowing that both the positive cases and negative cases are true, the overall measure of evaluating the reliability of all cases does not outweigh the consequences of obtaining true positive cases alone. Having a high specificity, where the number of true negative cases is high, is not as significant because misdiagnosing a condition when the patient is normal is not as dangerous or fatal as failing to diagnose any condition in the first place.

Of the diagnoses, the model was most accurate for hypertrophy, which is when the heart muscle becomes enlarged. This is not surprising because there is a clear distinction in ECG signals with this condition because of the large amplitude in the QRS complex and the T-wave inversion for each wave. This distinct feature in the ECG signals would make it easier for the model to distinguish the hypertrophy diagnosis from others since it can identify such robust patterns. The second most accurate was for conduction disturbance, where the ECG signals do not propagate through the heart in a normal pattern. A slowed conduction pattern or complete block pattern is an indication of this condition, with the most obvious feature being the change in timing. Because timing can be easily distinguished or stand out from other ECG signals, perhaps the model is able to learn from such distinct patterns to identify and predict the condition.

6. Discuss the role of ML/AI as decision support for physicians. Is the FDA justified in not yet allowing ML/AI to provide diagnosis without physician oversight? What would you consider minimum viable performance for AI/ML to diagnose or offer treatment options without physician supervision? **(6 pts)**

   ML/AI has a plethora of opportunities to support physicians and clinicians in healthcare such as through notetaking or imaging. There are many ways in which mundane tasks can be automated or validated for decision making. However, it does make sense that the FDA does not allow for such applications without physician oversight because there is always the possibility of automation bias in the model. This assumes that because there is no human bias, the model would always be considered correct. However, the training and testing data may be bias based on whether or not it is representative of the true population with the underlying medical condition (geographic aspect, socio-economic status, access to healthcare). If the model is fed with biased data, then the model itself will continue to reflect that bias. The labelling of the data may also be incorrect or inconsistent across equipment in hospitals (and knowledge across countries), which would affect the way in which the model is trained on such labels. The level of performance of the model would also be dependent on what would be considered “acceptable” which may vary significantly based on the dataset and the objectives of the model itself. Because the ability to diagnose or provide treatments without physician supervision would be putting people’s lives at stake, the accuracy of the model must be extremely high. This minimum performance of the model must match

the level of reliability and accuracy that real physicians have on their patients, which could be up to 85.6% (https://pmc.ncbi.nlm.nih.gov/articles/PMC6484633/) in providing correct diagnoses and offering treatments. When comparing to expert physicians, this accuracy threshold could be up to 95% as the minimum viable performance of AI models. By matching this accuracy, AI models will have the ability to perform the same as individual physicians, so the stakes of impacting patient lives will be the same.