ATRIAL FIBRILLATION DETECTION

I. Introduction

a. *Problem Statement*

How to help patients and physicians detect Atrial Fibrillation (AF) in an early stage and provide treatment promptly as needed to reduce 10% of AF patients who are at high risk of having a stroke, a heart attack, and even a heart failure by the end of 2021?

b. *Context*

Atrial Fibrillation (AF) increases a patient's risk of life-threatening complications, such as stroke, heart attack, and heart failure. AF is also independently associated with a significantly greater risk of mortality. For instance, AF patients have a 46% greater risk of mortality than patients without AF and the rate of mortality is 40% among new patients diagnosed with AF. Around 15-30% of patients are asymptomatic, which is of concern as AF is a major risk factor for stroke. As AF progresses, patients are more likely to experience greater impairments in their quality of life, such as increased pain and discomfort. Early detection and appropriate management reduce stroke risk by two-thirds. As a result, early detection of AF is important to ensure prompt and adequate management which not only aims to control symptoms but to avoid later complications.

c. *Criteria for success*

By the end of 2021, we need to reduce 10% of AF patients who do not know that they are having AF, which puts their lives at risk as they may have a stroke or a heart attack anytime.

d. *Scope of solution space*

AF patients in the United States.

e. *Constraints*

The dataset is sort of large, which will consume a lot of power resources and time to train the model.

f. *Stakeholders*

AF physicians, patients, and everyone in general.

g. *Data source(s) and References*

https://www.kaggle.com/arjunascagnetto/ptbxl-atrial-fibrillation-detection
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3402179/

II. Datasets

a. coorteeqsrafva.csv: This is a subset of the PTB-XL, a large publicly available electrocardiography dataset, found on Kaggle. This dataset includes 3 ECG rhythms in the ***ritmi*** column: Normal (SR), Atrial Fibrillation (AF), all other arrhythmia (VA). Please see below for the codebook.

Section	Variable	Data Type	Description
Identifiers	ecg_id	integer	unique ECG identifier
	patient_id	integer	unique patient identifier
	filename_lr	string	path to waveform data (100 Hz)
	filename_hr	string	path to waveform data (500 Hz)
General Metadata	age	integer	age at recording in years (see Fig. 3 left)
	sex	categorical	sex (male 0, female 1)
	height	integer	height in centimeters (see Fig. 3 right)
	weight	integer	weight in kilograms (see Fig. 3 middle)
	nurse	categorical	involved nurse (pseudonymized)
	site	categorical	recording site (pseudonymized)
	device	categorical	recording device
	recording_date	datetime	ECG recording date and time
ECG Statements	report	string	ECG report from diagnosing cardiologist
	scp_codes	dictionary	SCP ECG statements (see Tables 6, 7 and 8)
	heart_axis	categorical	heart's electrical axis (see Table 10)
	infarction_stadium1	categorical	infarction stadium (see Table 11)
	infarction_stadium2	categorical	second infarction stadium (see Table 11)
	validated_by	categorical	validating cardiologist (pseudonymized)
	second_opinion	boolean	flag for second (deviating) opinion
	initial_autogenerated_report	boolean	initial autogenerated report by ECG device
	validated_by_human	boolean	validated by human
Signal Metadata	baseline_drift	string	baseline drift or jump present
	static_noise	string	electric hum/static noise present
	burst_noise	string	burst noise
	electrodes_problems	string	electrodes problems
	extra_beats	string	extra beats
	pacemaker	string	pacemaker
Cross-validation Folds	strat_fold	integer	suggested stratified folds

Figure 1 - PTB-XL codebook

b. ecgeq-500hzsrfava.npy: This numpy file contains the 12-leads ECG of the patients in 'coorteeqsrfava' file. These are the recording (6528) of the ECG that have one and only one of these conditions:

- Sinusal Rhythm (SR): The condition of a normal ECG.
- Atrial Fibrillation (AF): The condition of having the specific arrhythmia of Atrial Fibrillation.
- Various Arrhythmia (VA): The condition of having one of the possible other types of arrhythmias.

To simply put, this is a 3D array, which contains 6428 layers, 5000 rows, and 12 columns. 12 columns represent for 12 leads, which are lead I, II, III, aVF, aVR, aVL, V1, V2, V3, V4, V5, V6. Leads I, II, III, aVR, aVL, aVF are denoted the limb leads while the V1, V2, V3, V4, V5, and V6 are precordial leads.

III. Data Cleaning/Wrangling

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6428 entries, 0 to 6427
Data columns (total 30 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   diagnosi                      6428 non-null   object
 1   ecg_id                        6428 non-null   int64
 2   ritmi                         6428 non-null   object
 3   patient_id                    6428 non-null   float64
 4   age                           6394 non-null   float64
 5   sex                           6428 non-null   int64
 6   height                        1866 non-null   float64
 7   weight                        2428 non-null   float64
 8   nurse                         6097 non-null   float64
 9   site                          6423 non-null   float64
 10  device                        6428 non-null   object
 11  recording_date                6428 non-null   object
 12  report                        6428 non-null   object
 13  scp_codes                     6428 non-null   object
 14  heart_axis                    4124 non-null   object
 15  infarction_stadium1           1800 non-null   object
 16  infarction_stadium2           26 non-null     object
 17  validated_by                  3676 non-null   float64
 18  second_opinion                6428 non-null   bool
 19  initial_autogenerated_report  6428 non-null   bool
 20  validated_by_human            6428 non-null   bool
 21  baseline_drift                444 non-null    object
 22  static_noise                  1021 non-null   object
 23  burst_noise                   265 non-null    object
 24  electrodes_problems           10 non-null     object
 25  extra_beats                   851 non-null    object
 26  pacemaker                     294 non-null    object
 27  strat_fold                    6428 non-null   int64
 28  filename_lr                   6428 non-null   object
 29  filename_hr                   6428 non-null   object
dtypes: bool(3), float64(7), int64(3), object(17)
memory usage: 1.4+ MB
```

Figure 2 - Info of the dataset

First, we need to look at the dataset in general. We can notice that there are some columns that have null values, such as ***age, height, weight, nurse, site, heart_axis, infarction_stadium1, infarction_stadium2, validated_by, baseline_drfit, static_noise, burst_noise, electrodes_problems, extra_beats, pacemaker***. We might not want to drop all the null values in this phase since it will reduce our data points because we all know that the larger our data is, the better. Instead, we could replace the null values with the mean or the median for the quantitative columns in the feature engineering phase. We are most likely to drop the columns that do not provide any significant insights later.

Now, let us take a look at the descriptive statistics for each column to detect the outliers. We only included the age, height, and weight columns as these columns do provide us useful information when we are looking at the descriptive statistics.

	age	height	weight
count	6394.000000	1866.000000	2428.000000
mean	61.740069	166.796356	69.841845
std	17.739252	10.249504	16.795521
min	4.000000	95.000000	5.000000
25%	52.000000	160.000000	58.000000
50%	64.000000	167.000000	69.000000
75%	75.000000	174.000000	79.000000
max	95.000000	195.000000	210.000000

Figure 3 - Descriptive Statistics

Looking at min, max, and mean for each column, we can conclude that there is no outlier exist in these three columns. Thus, we do not need to remove any outliers.

Next, we need to determine which column we should drop before moving onto the Exploratory Data Analysis phase. One way to determine unnecessary columns is checking unique values for each column. For instance, if a column has too many different categories that make it hard to visualize, we are more likely to drop that column. We also drop a column that contains all the id of a specific object, such as the column that contains patients' id.

After successfully checking unique values for each column, we decided to drop ***ecg_id, patient_id, nurse, site, report, scp_codes, validated_by, second_opinion, initial_autogenerated_report, baseline_drift, static_noise, burst_noise, electrodes_problems, extra_beats, pacemaker, filename_lr, filename_hr*** as some of them do not provide significant information, while some of them will make it very hard to visualize.

Last step, we recoded values for ***ritmi*** and ***validated_by_human*** from string and boolean to numeric values. For the ***ritmi*** column, "SR" will be 0, "AF" will be 1, and "VA" will be 2. For the ***validated_by_human*** column, False will be 0 and True will be 1.

We also created grouped variables for ***age, height, weight, and recording_date*** to make it easier for visualizing.

For the ***age*** column, we created a grouped variable for age called ***age_group***. There will be 10 groups:

- 0 - 9 Years
- 10 - 19 Years
- 20 - 29 Years
- 30 - 39 Years
- 40 - 49 Years
- 50 - 59 Years
- 60 - 69 Years
- 70 - 79 Years
- 80+ Years
- Missing (For null values)

Similarly, we created a grouped variable for height called ***height_group***. There will be 7 groups:

- <1.50m: Less than 1.50m
- 1.50m +: 1.50m and above
- 1.60m +: 1.60m and above
- 1.70m +: 1.70m and above
- 1.80m +: 1.80m and above
- 1.90m +: 1.90m and above
- Missing (For null values)

Likewise, we created a grouped variable for weight called ***weight_group***. There will be 7 groups:

- <60kg: Less than 60kg
- 60kg +: 60kg and above
- 70kg +: 70kg and above
- 80kg +: 80kg and above
- 90kg +: 90kg and above
- 100kg +: 100kg and above
- Missing (For null values)

We also want to get a year for each record instead of the date; thus, we created a grouped variable called ***recording_year*** by using the pd.to_datetime() function on the ***recording_date*** column.

Finally, we had a final dataset that consists of 6428 observations and 14 columns and exported to a csv file. We used this dataset in the Exploratory Data Analysis phase.

	diagnosi	ritmi	age	sex	height	weight	recording_date	heart_axis	validated_by_human	strat_fold	age_group	height_group	weight_group	recording_year
2	STACH	2	54	0			9/1/1993 11:31	MID	0	6	50-59 Years	Missing	Missing	1993
3	AFLT	1	54	0			9/1/1993 11:31	MID	0	6	50-59 Years	Missing	Missing	1993
4	SR	0	55	0			6/9/1992 15:52	LAD	1	10	50-59 Years	Missing	Missing	1992
5	STACH	2	29	1	164	56	2/8/1997 18:33		1	1	20-29 Years	1.60m +	<60kg	1997
6	SBRAD	2	57	0			9/13/1994 10:21	MID	0	1	50-59 Years	Missing	Missing	1994
7	SR	0	59	0	156	75	2/8/1997 6:17		1	9	50-59 Years	1.50m +	70kg +	1997
8	PACE	2	60	0			8/27/1997 15:24		0	5	60-69 Years	Missing	Missing	1997
9	AFIB	1	82	1			12/22/1999 10:18	MID	1	3	80+ Years	Missing	Missing	1999
10	SR	0	52	0			10/11/1994 14:04	ALAD	0	7	50-59 Years	Missing	Missing	1994

Figure 4 - Modified dataframe was exported to a csv file.

IV. Exploratory Data Analysis

1. General

a) Diagnosis

There are 12 classes composed on the rhythm diagnosis from PTB-XL:

- SR: Sinus Rhythm
- AFIB: Atrial Fibrillation
- STACH: Sinus Tachycardia
- SARRH: Sinus Arrhythmia
- SBRAD: Sinus Bradycardia
- PACE: Normal Functioning Artificial Pacemaker
- SVARR: Supraventricular Arrhythmia
- BIGU: Bigeminal Pattern (unknown origin, SV or Ventricular)
- AFLT: Atrial Flutter
- SVTAC: Supraventricular Tachycardia
- PSVT: Paroxysmal Supraventricular Tachycardia
- TRIGU: Trigeminal Pattern (unknown origin, SV or Ventricular)

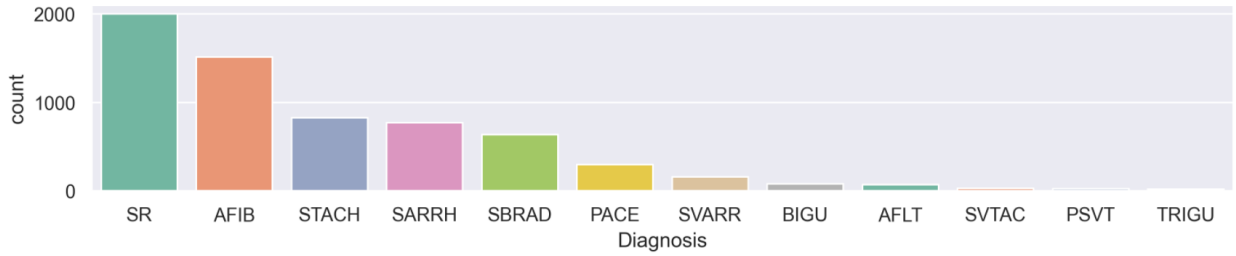


Figure 5 - diagnosi column

As expected, SR has the most records, then AFIB, next is STACH and then SARRH. We can say that approximately 2000 individuals in this dataset have Sinus Rhytm, 1514 individuals with Atrial Fibrillation, 826 individuals with Sinus Tachycardia and 772 individuals with Sinus Arrhythmia. Additionally, there are very small number of individuals who have Supraventricular Tachycardia (27), Paroxysmal Supraventricular Tachycardia (24), and Trigeminal Pattern (20).

b) Rhythm

As we recoded the values in the data wrangling phase, 0 is SR, 1 is AF, and 2 is VA.

- Sinusal Rhythm (SR). The condition of a normal ECG.
- Atrial Fibrillation (AF). The condition of having the specific arrhythmia of Atrial Fibrillation.
- Various Arrhythmia (VA). The condition of having one of the possible other types of arrhythmias.

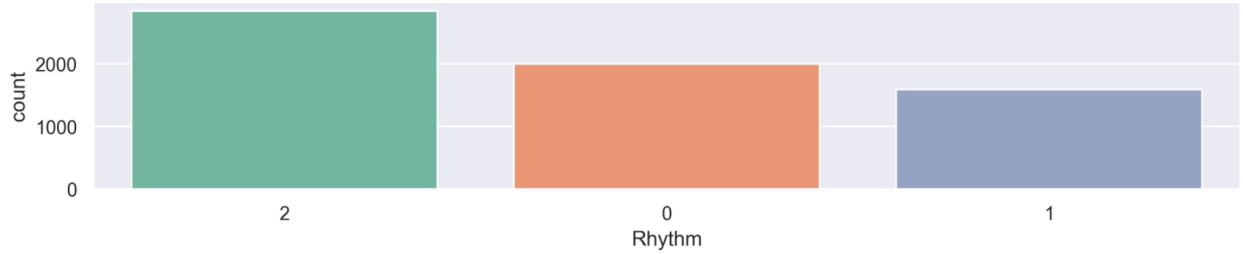


Figure 6 - ritmi column

2841 individuals who have the condition of having one of the possible other types of arrhythmias. 2000 individuals who have the condition of a normal ECG, and 1587 individuals who have the condition of having the specific arrhythmia of Atrial Fibrillation. This column is a simplified version of the ***diagnosis*** column. Nevertheless, we can notice that the number for individuals with AF reported in the ***ritmi*** column do not match with the number reported in the ***diagnosi*** column as there are more 73 individuals with AF in the ***ritmi*** column. However, since it is not a significant difference, we do not have to worry much about this.

c) Age

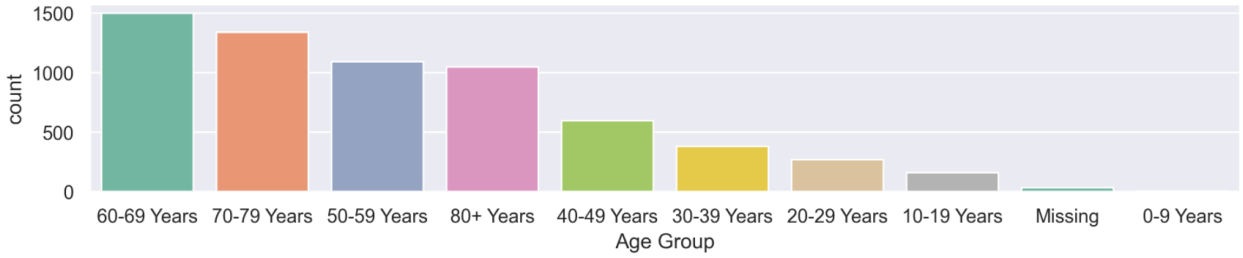


Figure 7 - age_group column

There are four outstanding groups, 60-69 Years (1500), 70-79 Years (1338), 50-59 Years (1093), and 80+ Years (1049). This totally makes sense as most individuals who are from 50 to 80 years old and above usually have heart diseases. For individuals who are young, they have a lower risk of getting heart diseases as 6 people from 0 to 9 years old, and 162 people from 20-29 years old. We also discover that the average age of patients is 62.0 years old. The minimum age of patients is 4.0 years old, while the maximum age of patients is 95.0 years old.

d) Sex (Male is 0, female is 1)

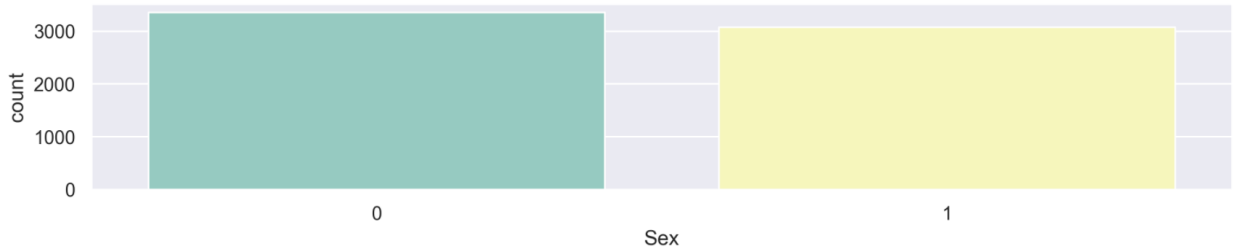


Figure 8 - sex column

There are more male patients (3353) than female patients (3075), not a huge difference though.

e) Height

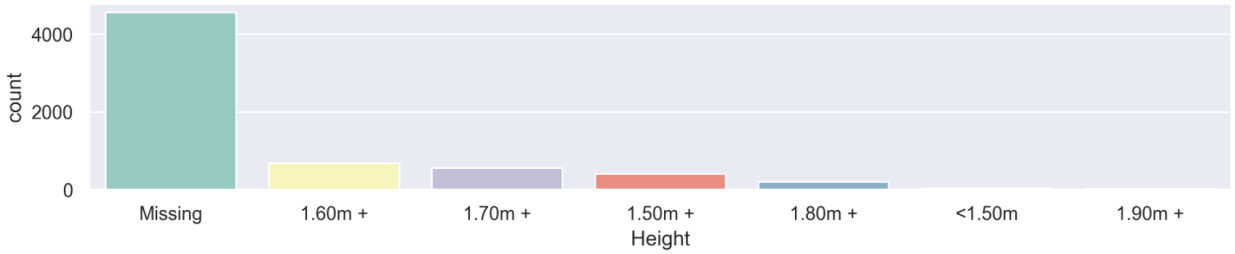


Figure 9 - height_group column

We see a huge number of missing values in ***height_group*** column. Despite that, most patients are 1.60/1.70 m and above, very few patients are 1.90 m and above. The average height of patients is 1.7 m. The minimum height of patients is 0.95 m, while the maximum height of patients is 1.95 m.

f) Weight

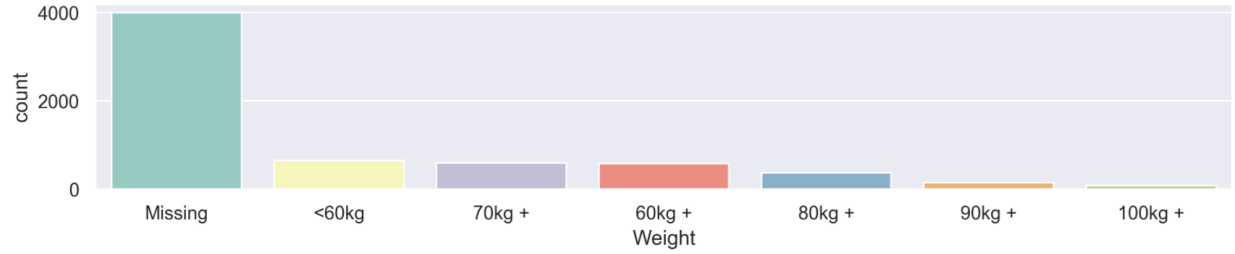


Figure 10 - weight_group column

Similar to ***height_group***, there are 4000 missing values in ***weight_group***. Most patients are less than 60 kg, and more than 60/70kg, very few patients are 100 kg and above. Based on this, we can assume that some patients with heart diseases do not have obesity, while some do. The average weight of patients is 70.0 kg. The minimum weight of patients is 5.0 kg, and the maximum weight of patients is 210.0 kg.

g) Recording Year

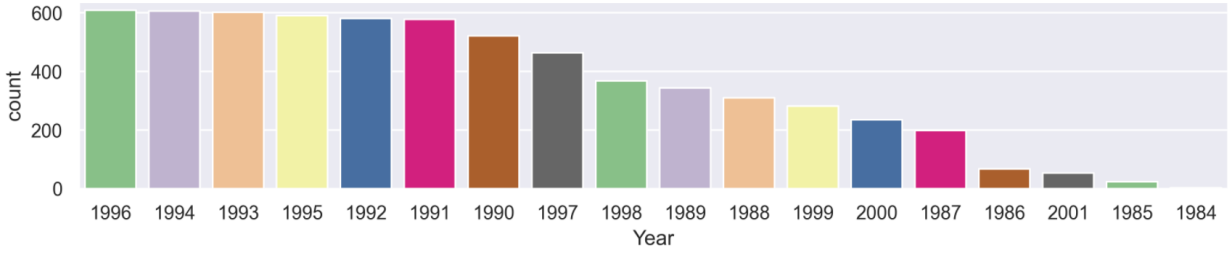


Figure 11 - recording_year column

This dataset comprises lots of old records as 2001 is the latest year. We can see that the records started to be collected from 1984 to 2001. Specifically, most records are in 1996 (608), 1994 (605), 1993 (601) and 1995 (590). Only a few records are in 1984 (3), 1985 (23), and 2001 (53).

h) Heart's Electrical Axis

These are the keywords of each code:

- MID - Normal axis
- LAD - Left axis deviation
- ALAD - Abnormal
- LAD, extreme left axis deviation
- RAD - Right axis deviation
- ARAD - Abnormal
- RAD, extreme right axis deviation
- AXR - Vertical axis
- AXL - Horizontal axis
- SAG - Saggital type (S1-S2-S3 Pattern)

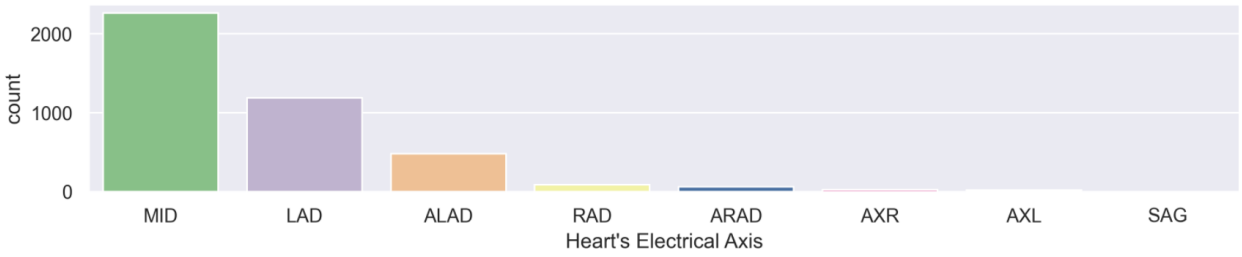


Figure 12 - heart_axis column

Based on figure 12, we can say that most patients have normal axis for heart's electrical axis as there are 2262 of them. 1187 patients with left axis deviation and 482 patients with Abnormal LAD, extreme left axis deviation. Only 1 patient who has Saggital type (S1-S2-S3 Pattern).

i) Validated by Human (False is 0, true is 1)

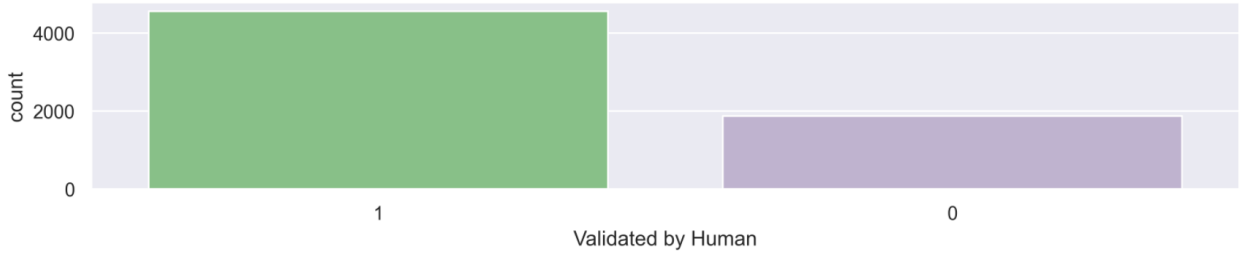


Figure 13 - validated_by_human column

There are 4559 records are validated by human, while 1869 records are not validated by human. We can assume that most records in this dataset are valid.

j) Suggested Stratified Folds

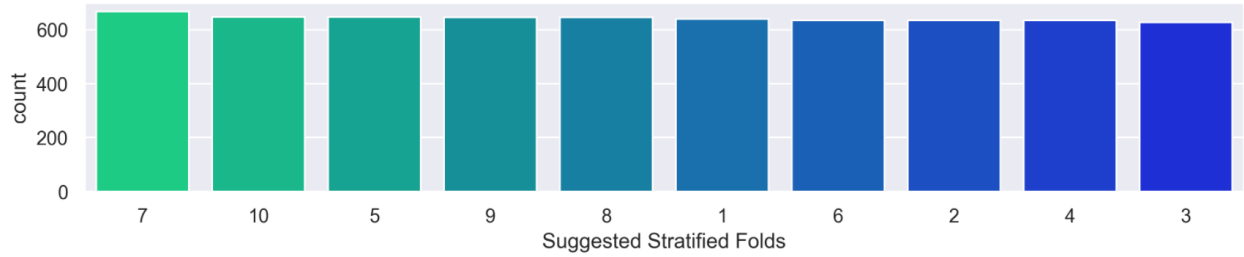


Figure 14 - strat_fold column

As shown above, there is a slightly difference between 10 groups; however, some groups have the exact the same values, such as group 10 (648) and group 5 (648), group 9 (646) and group 8 (646), group 6 (635), group 2 (635), and group 4 (635). Based on this result, we can understand which value we should use to set cross_validation when we are tuning the model. Specifically, 7, 10, and 5 are the best number for cv.

2. Atrial Fibrillation-related Questions

We generate some plots to help us answer Atrial Fibrillation-related questions.

a) Which gender usually has a higher risk of getting AF?

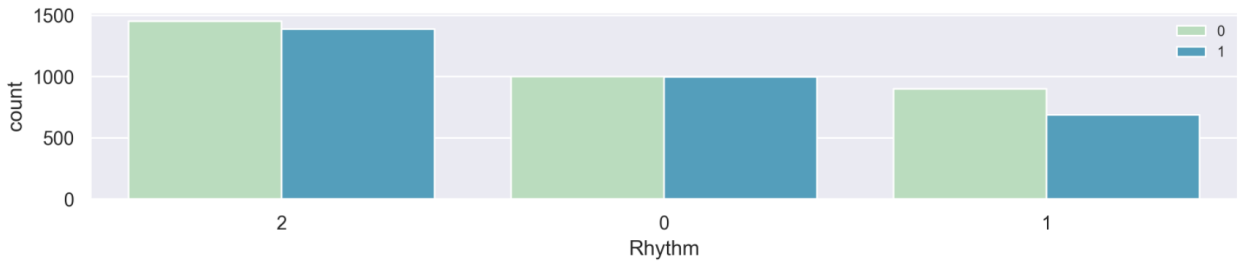


Figure 15 - ritmi by sex

➔ Female patients are at higher risk of getting AF than male patients.

b) Which age-group is associated with higher risk of having AF than others?

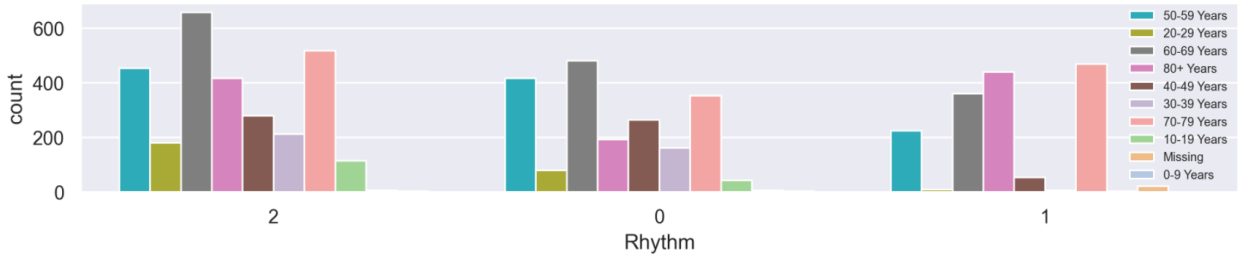


Figure 16 - ritmi by age_group

➔ Patients who are 70 to 89 years old have a higher risk of having AF than others.

c) What is the common weight of patients who have AF?

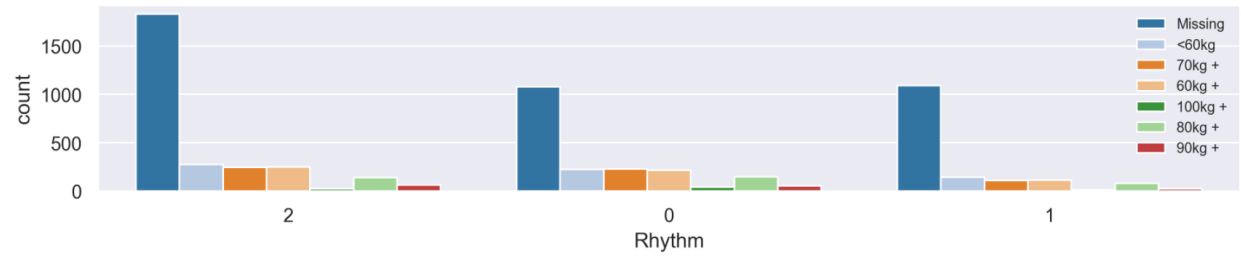


Figure 17 - ritmi by weight_group

➔ Patients who have AF are usually less than 60kg, or 60 to 79kg.

d) What is the common height of patients who have AF?

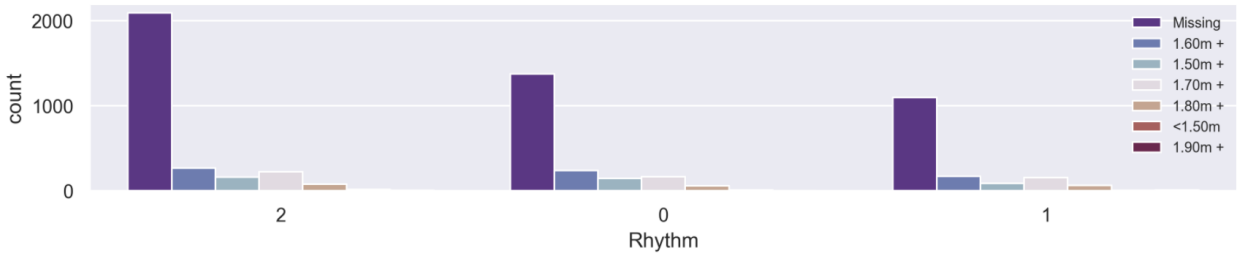


Figure 18 - ritmi by height_group

➔ Patients who have AF are usually from 1.50m to 1.79m.

e) What is the most common heart's electrical axis associated with AF patients?

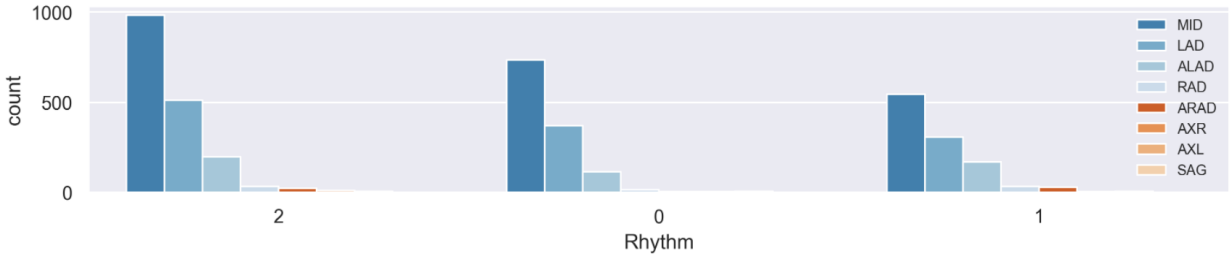


Figure 19 - ritmi by heart_axis

➔ Most AF patients have normal heart's electrical axis.

3. ECGs

We generate 5 random ECG for each rhythm using the numpy data file.

a. Normal ECG

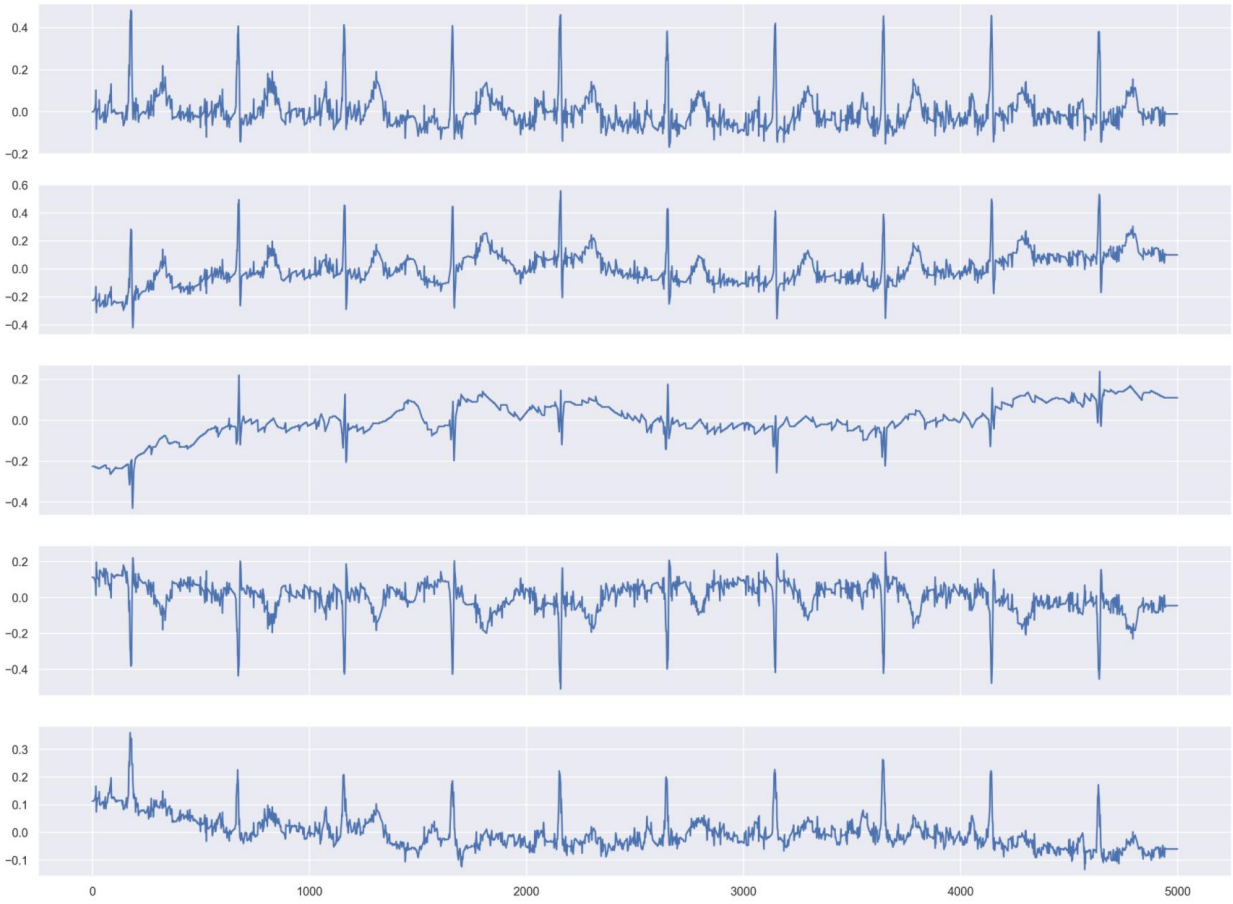


Figure 20 – ECGs for normal rhythm

Since I do not have knowledge about reading ECGs, I cannot interpret each plot exactly like the way it is. However, I could make some judgements based on the shape of the line for each category to differentiate them. For the normal ECG, they have a fixed space between each peak. For instance, as shown in figure 20, the last plot shows the heart rate fluctuates from 2000 to 2200 then peaks at 2300. Similarly, the heart rate fluctuates from 2400 to 2600 then peaks at 2700.

b. Atrial Fibrillation ECG

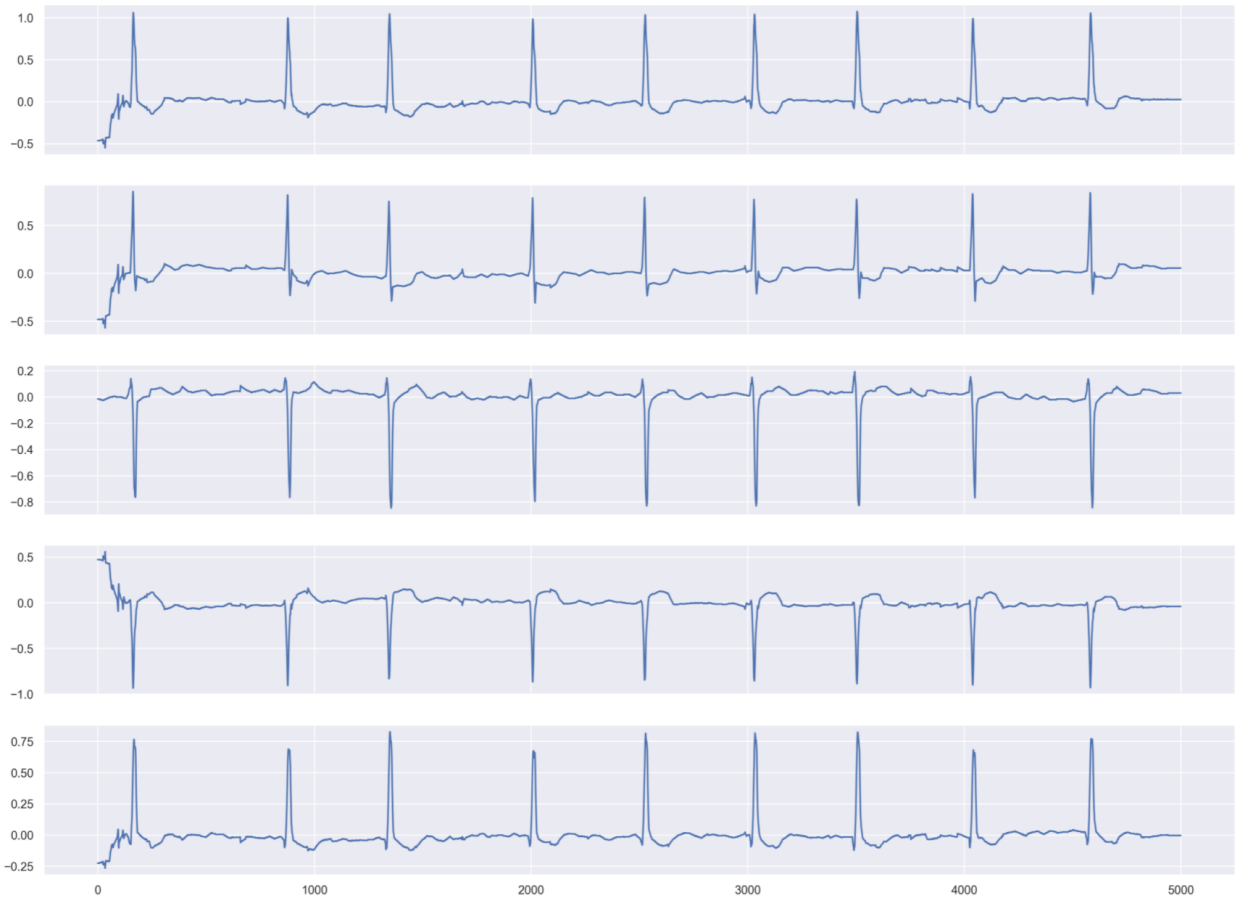
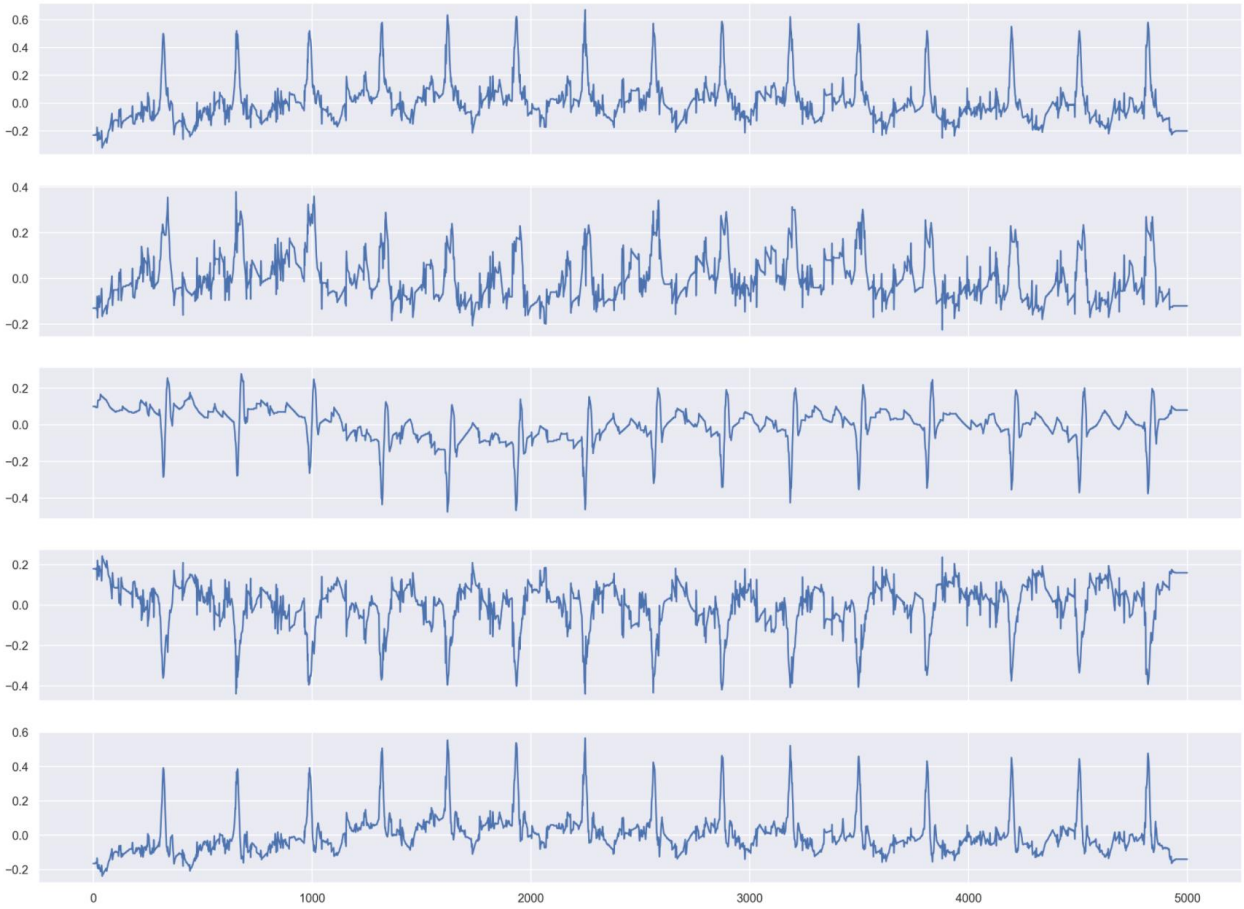


Figure 21 - ECGs for Atrial Fibrillation rhythm

For the Atrial Fibrillation ECG, they do not have a fixed space between each peak, some spaces are long, and some are short. Moreover, the heart rate is very irregular as shown in the last plot in figure 21. Most of the time, the heart rate is above 0.75, but sometimes the heart rate is below 0.75.

c. Other Arrhythmia ECG



For the other arrhymthmia ECG, similar to normal ECG, they do have a fixed space between each peak. However, the difference is the peak seems much higher in other arrhymthmia ECG compared to normal ECG.

V. Feature Engineering

Since we have two files, one is the CSV file, and the other is the numpy file. We are not sure which file would help us in the modeling phase. Since the csv file has so many missing values, we are also not certain whether filling the missing values with the mean/median values or dropping the missing values would yield the higher accuracy score. In addition, we want to make use of the numpy file as we hope it would help us in detecting the Atrial Fibrillation cases since this project is all about the pattern recognition. Without using 12 leads from the numpy file, detecting AF cases only using height, weight, age, etc. is not persuading the cardiology physicians. As a result, we created three datasets. Two datasets are created from the csv file, and one dataset is created by merging the features from the csv file and the numpy file.

a) *First Dataset: 11 variables (10 features and 1 label)*

In this dataset, we dropped all the missing values for the height and weight columns after reading in the csv file. We also filled the missing values for the age column with the mean values. For the other columns, we filled in the missing values with either 0 or "missing". We ended up having 1803 data points with 10 features and 1 label.

10 features		*1 label*
○ age	○ validated_by	ritmi
○ sex	○ second_opinion	
○ height	○ validated_by_human	
○ weight	○ pacemaker	
○ heart_axis	○ strat_fold	

	ritmi	age	sex	height	weight	heart_axis	validated_by	second_opinion	validated_by_human	pacemaker	strat_fold
0	2	29.0	1	164.0	56.0	0	0.0	0	1	0	1
1	0	59.0	0	156.0	75.0	0	0.0	0	1	0	9
2	2	84.0	1	152.0	51.0	0	0.0	0	1	0	7
3	0	79.0	0	172.0	66.0	0	0.0	0	1	0	5
4	1	67.0	0	178.0	73.0	4	0.0	0	1	0	5

Figure 22 - First dataset

b) Second dataset: 14 variables (13 features and 1 label)

In the second dataset, after reading in the csv file, we filled the missing values with the mean values for the age, height, weight columns, and filled missing values with 0 for the nurse, site, validated_by, heart_axis, and pacemaker columns. Therefore, we ended up having 6366 data points with 13 features and 1 label.

13 features		***1 label***
○ age	○ heart_axis	ritmi
○ sex	○ validated_by	
○ height	○ second_opinion	
○ weight	○ validated_by_human	
○ nurse	○ pacemaker	
○ site	○ strat_fold	
○ device		

	ritmi	age	sex	height	weight	nurse	site	device	heart_axis	validated_by	second_opinion	validated_by_human	pacemaker	strat_fold
0	2	54.0	0	166.796356	69.841845	0.0	0.0	0	3.0	0.0	0	0	0.0	6
1	1	54.0	0	166.796356	69.841845	0.0	0.0	0	3.0	0.0	0	0	0.0	6
2	0	55.0	0	166.796356	69.841845	1.0	2.0	1	1.0	1.0	0	1	0.0	10
3	2	29.0	1	164.000000	56.000000	7.0	1.0	10	0.0	0.0	0	1	0.0	1
4	2	57.0	0	166.796356	69.841845	0.0	0.0	0	3.0	0.0	0	0	0.0	1

Figure 23 - Second dataset

c) Third dataset: 25 variables (24 features and 1 label)

As mentioned above, we have a numpy file, which is a 3D array that contains 6428 layers, 5000 rows, and 12 columns. We decided to transform the 3D array to a 2D array and then converted it to a dataframe. However, we only kept 700 rows instead of 5000 rows to reduce the size of the dataset, which helps reduce the time and computing resources when training the model. The converted dataframe has 4361843 observations and 13 variables (12 leads and 1 index column).

Afterwards, we read in the csv file created from the second dataset above, which consists of 14 variables. We then merged the converted dataframe (13 variables) and the dataframe from the second dataset (14 variables). As a result, we have a total of 26 variables after dropping the index column. The final dataset has a total of 4319176 observations and 26 variables.

24 features		***1 label***
○ I	○ sex	ritmi
○ II	○ height	
○ III	○ weight	
○ aVF	○ nurse	
○ aVR	○ site	
○ aVL	○ device	
○ V1	○ heart_axis	
○ V2	○ validated_by	
○ V3	○ second_opinion	
○ V4	○ validated_by_human	
○ V5	○ pacemaker	
○ V6	○ strat_fold	
○ age		

	I	II	III	aVF	aVR	aVL	V1	V2	V3	V4	...	weight	nurse	site	device	heart_axis	validated_by
0	-0.005	0.135	0.140	-0.065	-0.073	0.137	-0.125	-0.090	-0.110	-0.210	...	69.841845	0.0	0.0	0	3.0	0.0
1	-0.005	0.135	0.140	-0.065	-0.073	0.137	-0.125	-0.090	-0.110	-0.211	...	69.841845	0.0	0.0	0	3.0	0.0
2	-0.005	0.131	0.136	-0.063	-0.070	0.133	-0.125	-0.082	-0.102	-0.190	...	69.841845	0.0	0.0	0	3.0	0.0
3	-0.005	0.130	0.135	-0.063	-0.070	0.132	-0.122	-0.077	-0.094	-0.172	...	69.841845	0.0	0.0	0	3.0	0.0
4	-0.005	0.128	0.133	-0.062	-0.069	0.130	-0.119	-0.071	-0.084	-0.157	...	69.841845	0.0	0.0	0	3.0	0.0

Figure 24 - Third dataset

VI. Modeling

Even though we have three datasets created in the feature engineering phase, we do not know which dataset will help us in yielding the high accuracy score yet. As a result, in the modeling phase, we read in each file and applied different algorithms to compare the accuracy score. Besides the accuracy metric, we also focused on the recall metric since we want to detect as many Atrial Fibrillation cases as possible.

a) *First dataset: 11 variables (10 features and 1 label)*

1. *Random Forest*

We used the RandomForest algorithm and tuned the model with GridSearchCV, we got 0.45 for the highest performance score (accuracy metric). Afterwards, we used the model to predict X_test. Based on the recall metric, we can conclude that the model has 47% of accurately detecting normal cases, 41% of accurately detecting Atrial Fibrillation,

	precision	recall	f1-score	support
0	0.45	0.44	0.45	162
1	0.45	0.43	0.44	117
2	0.44	0.47	0.45	172
accuracy			0.45	451
macro avg	0.45	0.45	0.45	451
weighted avg	0.45	0.45	0.45	451

Figure 25 - Random Forest on the first dataset

2. *LightGBM*

We also used the LightGBM algorithm and tuned the model with BayesianOptimization. We got 0.64 for the highest performance score using the AUC metric. We can say that the score has been improved a lot, but it is in a different metric. Since the metric is not the same, we will apply other algorithms later to see if there is another algorithm that helps us improve our accuracy score.

```
|   iter    |  target   | lambda_l1 | lambda_l2 | max_depth | min_ch... | min_da... | num_le... |
-------------------------------------------------------------------------------------------------
[LightGBM] [Warning] min_data_in_leaf is set=1488, min_child_samples=5433 will be ignored. Current value: min_data_
in_leaf=1488
|  1        |  0.5      |  0.001754 |  0.01623  |  13.16    |  5.433e+0 |  1.488e+0 |  1.711e+0 |
|  2        |  0.5      |  0.042    |  0.02319  |  49.17    |  1.11e+03 |  1.822e+0 |  1.603e+0 |
|  3        |  0.6337   |  0.04281  |  0.03493  |  39.03    |  226.1    |  218.8    |  3.964e+0 |
|  4        |  0.5      |  0.03825  |  0.01928  |  12.82    |  8.253e+0 |  1.422e+0 |  1.689e+0 |
|  5        |  0.5      |  0.002021 |  0.04044  |  57.3     |  7.747e+0 |  890.2    |  3.305e+0 |
|  6        |  0.642    |  0.03399  |  0.02087  |  42.14    |  361.3    |  153.9    |  158.1    |
|  7        |  0.6405   |  0.007697 |  0.004628 |  33.75    |  9.711e+0 |  164.3    |  103.5    |
|  8        |  0.6424   |  0.0252   |  0.004836 |  24.14    |  7.766e+0 |  102.4    |  140.5    |
|  9        |  0.6428   |  0.02575  |  0.03534  |  25.51    |  253.4    |  107.9    |  2.15e+03 |
|  10       |  0.6423   |  0.03853  |  0.03429  |  53.52    |  2.539e+0 |  115.3    |  105.0    |
|  11       |  0.6419   |  0.005657 |  0.04969  |  34.51    |  1.434e+0 |  101.1    |  40.52    |
|  12       |  0.6418   |  0.03422  |  0.01324  |  22.84    |  5.41e+03 |  104.9    |  49.81    |
=================================================================================================
```

Figure 26 - LightGBM on the first dataset

3. *K-Neighbors*

After trying different algorithms, such as LogisticRegression, LogisticRegressionCV, SVC, etc., we can see that KNeighborsClassifier returned 0.45 for the highest score using the same accuracy metric. Based on the recall metric, the model has 41% of accurately detecting normal cases, 47% of accurately detecting Atrial Fibrillation, and 49% of accurately detecting other arrhythmia cases. Since the accuracy score in this algorithm is the same as the random forest algorithm, which is 0.45, we can assume that KNeighborsClassifier is the most suitable algorithm in this dataset based on the recall metric as it has 47% of accurately detecting AF cases, while the RandomForest algorithm only has 43% of accurately detecting AF cases.

	precision	recall	f1-score	support
0	0.47	0.41	0.44	162
1	0.49	0.47	0.48	117
2	0.42	0.49	0.45	172
accuracy			0.45	451
macro avg	0.46	0.46	0.46	451
weighted avg	0.46	0.45	0.45	451

Figure 27 - K-Neighbors on the first dataset

b) *Second dataset: 14 variables (13 features and 1 label)*

1. *Random Forest*

Similar to the first dataset, we used the RandomForest algorithm and tuned the model with GridSearchCV, we got 0.50 for the accuracy score. Afterwards, we used the model to predict X_test. Based on the recall metric, we can conclude that the model has 38% of accurately detecting normal cases, 49% of accurately detecting Atrial Fibrillation, and 60% of accurately detecting other arrhythmia cases.

	precision	recall	f1-score	support
0.0	0.43	0.38	0.40	407
1.0	0.51	0.48	0.49	318
2.0	0.54	0.60	0.57	549
accuracy			0.50	1274
macro avg	0.49	0.49	0.49	1274
weighted avg	0.50	0.50	0.50	1274

Figure 28 - Random Forest on the second dataset

2. *LightGBM*

We also used the LightGBM algorithm, and tuned the model with BayesianOptimization, and we got 0.64 for the highest performance score using the AUC metric. Thus, we can conclude that it has the same score as the first dataset.

iter	target	lambda_l1	lambda_l2	max_depth	min_ch...	min_da...	num_le...
1	0.6391	0.04647	0.02277	18.49	8.727e+0	134.5	1.371e+0
2	0.6044	0.02907	0.006896	52.64	8.533e+0	1.595e+0	3.268e+0
3	0.6383	0.003875	0.03582	21.89	70.68	153.7	117.0
4	0.6081	0.01025	0.01776	45.82	4.809e+0	1.457e+0	1.265e+0
5	0.6408	0.04423	0.02645	41.34	1.185e+0	153.9	3.993e+0
6	0.6319	0.045	0.02148	35.5	3.555e+0	487.9	3.331e+0
7	0.5	0.02452	0.02734	19.78	585.2	1.967e+0	1.528e+0
8	0.5	0.03278	0.000656	39.37	9.971e+0	1.886e+0	37.67
9	0.6365	0.02465	0.004804	36.71	9.703e+0	254.8	3.98e+03
10	0.642	0.01797	0.02621	41.39	4.317e+0	108.9	40.98
11	0.6354	0.0395	0.02178	20.65	2.415e+0	264.6	3.861e+0
12	0.6401	0.003655	0.04305	35.25	7.257e+0	114.0	3.895e+0

Figure 29 - LightGBM on the second dataset

3. *K-Neighbors*

Finally, we applied the KNeighborsClassifier algorithm. Based on the recall metric, the model has 23% of accurately detecting normal cases, 40% of accurately detecting Atrial Fibrillation, and 70% of accurately detecting other arrhythmia cases. Since the accuracy score and the recall scores in this algorithm is lower than the RandomForest algorithm, we can conclude that the RandomForest algorithm works best for this dataset as it returns 0.50 for the accuracy score and it has 48% of accurately detecting Atrial Fibrillation cases.

	precision	recall	f1-score	support
0.0	0.38	0.23	0.29	407
1.0	0.54	0.40	0.46	318
2.0	0.49	0.70	0.58	549
accuracy			0.48	1274
macro avg	0.47	0.44	0.44	1274
weighted avg	0.47	0.48	0.45	1274

Figure 30 - K-Neighbors on the second dataset

c) *Third dataset: 25 variables (24 features and 1 label)*

1. *Random Forest*

After training and tuning the model for the second dataset, we learned that the Random Forest algorithm is the best for detecting Atrial Fibrillation cases. Since this dataset is generated by merging the numpy features with the second dataset above, we only need to apply the Random Forest algorithm to get the accuracy score. Surprisingly, we got 0.99 for the accuracy score. We can conclude that the third dataset is the best dataset among the three datasets. Based on the recall metric, the model has 99% of accurately detecting normal cases, 98% of accurately detecting Atrial Fibrillation cases, and 99% of accurately detecting other arrhythmia cases.

	precision	recall	f1-score	support
0.0	1.00	0.99	0.99	335792
1.0	0.98	0.98	0.98	267846
2.0	0.99	0.99	0.99	476156
accuracy			0.99	1079794
macro avg	0.99	0.99	0.99	1079794
weighted avg	0.99	0.99	0.99	1079794

Figure 31 - Random Forest on the third dataset

d) *Summary*

The third dataset, which consists of 25 features and 1 label, is the most suitable dataset to be used in training the model.

Datasets	***(1) 11 features and 1 label***	***(2) 13 features and 1 label***	***(3) 25 features and 1 label***
Random Forest	0.45 (Accuracy)	0.48 (Accuracy)	0.99 (Accuracy)
K-Neighbors	0.45 (Accuracy)	0.50 (Accuracy)	N/A
LightGBM	0.64 (AUC)	0.64 (AUC)	N/A

VII. Conclusion and Feature Work

After going through 4 phrases, data wrangling, exploratory data analysis, feature engineering, and modeling, we have gained insights into all the features in the dataset and learned that we can get a very high accuracy score if we use all the features in both the csv file and the numpy file, which is the third dataset. Moreover, the Random Forest is the best algorithm that helps the model get such a high accuracy score. Since the numpy file is very large (~3GB), we could only choose 700 rows in the numpy file and merged it with the csv file. For the future work, if we have more ram that can train the model with a very large file like that, we should try to train the model using all 5000 rows to see whether we can get 99.9% of the accuracy score since the current accuracy score is 98.8% but rounded up to 99%.