DATA ANALYTICS REPORT

By Meryl Gabrielle Tubio

100763231

Table of Contents

Introduction

Princess Margret Hospital (in partnership with the UHN Echocardiography Lab) aims to save lives through early detection of heart murmur. With the dataset provided by the Hospital, Lab Researchers aims to detect early signs of heart murmur.

The purpose of this report is to identify insight about the heart data and produce an early detection model that would allow healthcare professionals for early-diagnosis and improve patient prognosis.

Key Questions

1. What features are important in the model?
2. What are the criteria for selecting these features?

Identifying key features within the data is critical in developing an accurate model. Key features that are integral to the overall accuracy of the model should be acknowledged and documented to allow modifications and possibility of an in-depth analysis on key features.

3. How to measure model performance?

Through the precision and recall metric[1]. For the base model a Receiver Operator Characteristic (ROC) curve would be included for model evaluation[2]. Models included in the report have a threshold of 87% score in model performance.

---

[1] the precision metric allows one to understand how the model is performing based on the patients correctly diagnosed among all the diagnosed patients diagnosed positive. While recall, also known as sensitivity, focuses on the probability that a patient would have heart murmur out of patients who were misdiagnosed to not have heart murmur. Precision and recall are based on an understanding and measure of relevance.

[2] The ROC learning curve highlights the trade-off between recall and specificity (1-False Positive Rate)

Dataset Summary

Prior to understanding the different data transformations, one must first understand the different variables. To rehash, the focus of this study is to identify the unique heart measurements of patients with no heart ailment and patients diagnosed with heart valve defects, the data to be used is a Binomial classification structure. The naming convention for this study is as follows:

Class 0 – No Heart Ailment

Class 1 – Heart Ailment (Congenital Defect or Heart Valve Defect)

The raw murmur dataset consisted of 5,010 lines patient heart measurements. One row in the dataset, indicates 1 patient. This has since been shrunk and pre-processed to a total of 5512 lines. The following are the major data transformations that were executed on the murmur dataset:

1. Identified and removed 450 missing values
2. Identified and removed 742 Outliers care of Turkey Method[1]
3. Identified class imbalance and performed class balance care of Synthetic Minority Oversampling Technique (SMOTE)[2]. From an original class ratio of 1:2 (Class 0 - 1438: Class 1 - 2820) to an improved class ratio of (Class 0 - 2256: Class 1 - 2256)

Furthermore, the following observations were evident in the dataset when the dataset was visualized:

1. Extreme fluctuations in correlations from features X1 to X20, feature from high to lows correlations. While all other features, namely X21 to X40, remain relatively low in terms of correlation.
2. The dataset is normally distributed. Although there seems to be a sharp spike in 0.40 and -0.5 features in the dataset.

For an in-depth discussion of the dataset pre-processing and EDA[3], please refer to appendix A.

---

[1]Turkey Method – a statistical method in identifying outliers and removing them outliers to normalize the data.

[2] Synthetic Minority Oversampling Technique (SMOTE) – a statistical technique for increasing the number of cases in the dataset to balance the dataset.

[3]Exploratory Data Analysis (EDA) – an approach to analyze the main characteristics of the dataset. This could be done through visuals and other python libraries.

Model Analysis

Researchers have utilized (3) three main models: these are (A) Logistic Regression, (B) Decision Tree, and (C) Random Forest.

A. Logistic Regression

Logistic Regression is a linear model that is designed for binary variables. For this report, the logistic regression is the base model used to compare the other models, primarily due to its performance in the learning curve[1] and metrics. See figure 1 below:

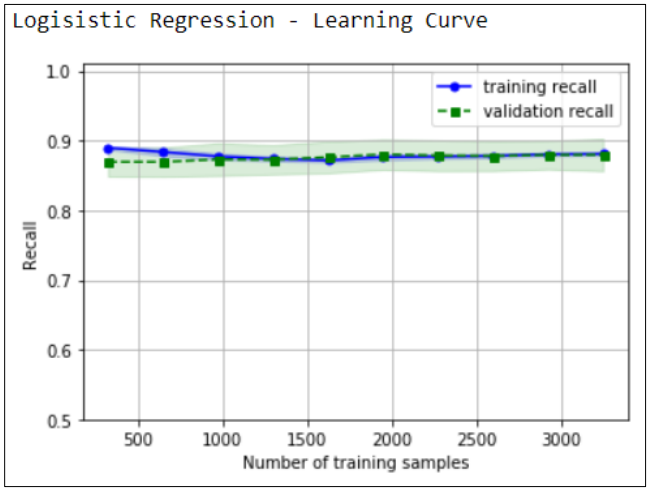


Figure 1

The Logistic Regression learning curve indicates a good bias-variance trade-off. Making it an ideal base model.

In addition, Logistic regression assumes the following characteristics of the data:

1. Binary – Heart murmur is binary (Class 0 and 1)
2. A linear relationship between the independent variables and the link function[2]
3. The dependent variable ("Class") is mutually exclusive and exhaustive

[1]Learning Curve - a learning curve shows the validation and training score of an estimator for varying numbers of training samples. It is a tool to find out how much a model benefits from adding more training data and whether the estimator suffers more from a variance error or a bias error.

[2]Link Function - In generalized linear models, there is a link function, which is the link between the mean of Y on the left and the fixed component on the right.

Meanwhile the results of the optimized model (Figure 2) indicates that the model has higher precision score for Outcome 1 (heart murmur), while there is a higher recall score for Outcome 0 (patients with no heart ailments).

```
Optimized Model

Model Name: LogisticRegression(C=1.0, class_weight='balanced', dual=False,
                   fit_intercept=True, intercept_scaling=1, l1_ratio=None,
                   max_iter=100, multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=100, solver='liblinear', tol=0.0001, verbose=0,
                   warm_start=False)

Best Parameters: {'clf__C': 0.01, 'clf__penalty': 'l2'}

 [[436  16]
 [ 91 360]]

              precision    recall  f1-score   support

   Outcome 0       0.83      0.96      0.89       452
   Outcome 1       0.96      0.80      0.87       451

    accuracy                           0.88       903
   macro avg       0.89      0.88      0.88       903
weighted avg       0.89      0.88      0.88       903


NestedCV Accuracy(weighted) :0.88 +/-0.01
NestedCV Precision(weighted) :0.89 +/-0.01
NestedCV Recall(weighted) :0.88 +/-0.01
```

Figure 2

To focus more on important features, Analysts have decided to utilize key features identified by the SelectFromModel[1] function, Figure 3 below shows the results of this function. *Moving forward, the results are solely based on the attributes indicated below for the 5512 patients.*

```
Key Features: Index(['x3', 'x4', 'x5', 'x6', 'x7', 'x9', 'x10', 'x11', 'x12', 'x13', 'x15',
       'x16', 'x17', 'x18'],
      dtype='object')
```

Figure 3

[1]SelectFrom Model – a simple method that removes less important features based on a threshold given as a parameter. No iteration is involved.

B. Decision Tree

A decision tree is a type of supervised learning algorithm that can be used for both regression and classification problems. It works for both categorical and continuous input and output variables. Unlike the logistic regression this algorithm does not consider any assumptions about the data and requires minimal data pre-processing. Although one drawback of this algorithm is that it is prone to overfit. The result of the decision tree is in figure 4, we can identify that there is high variance in the model.

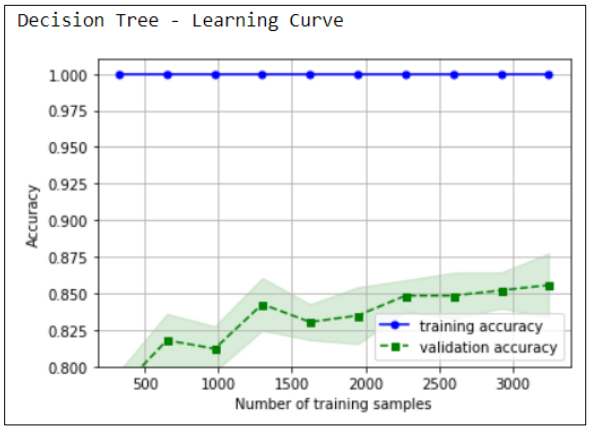


Figure 4

```
[[401  54]
 [ 75 373]]

              precision    recall  f1-score   support

         0.0       0.84      0.88      0.86       455
         1.0       0.87      0.83      0.85       448

    accuracy                           0.86       903
   macro avg       0.86      0.86      0.86       903
weighted avg       0.86      0.86      0.86       903


NestedCV Accuracy(weighted) :0.86 +/-0.03
NestedCV Precision(weighted) :0.86 +/-0.03
NestedCV Recall(weighted) :0.86 +/-0.03
```

Figure 5: Optimized Decision Tree

Figure 5 indicates the results of the decision tree, precision and recall are at the lower side compared to the logistic regression and a comparatively lower accuracy[1] score.

[1]Accuracy - true positive and true negatives added together, and then divided by the total. An in-depth discussion about accuracy and classification matrix is in appendix B.

C. Random Forest

The random forest algorithm consists of clusters of decisions trees and derives an overall output that is more accurate. Many decision trees are trained, but each tree only receives a bootstrapped[1] sample of observations. The dataset is resampled with replacement repeatedly.

Like decision tree, the algorithm has overfitted the model (see figure 6). However, the results (see figure 7) are substantially higher and is the highest accuracy among the three models.

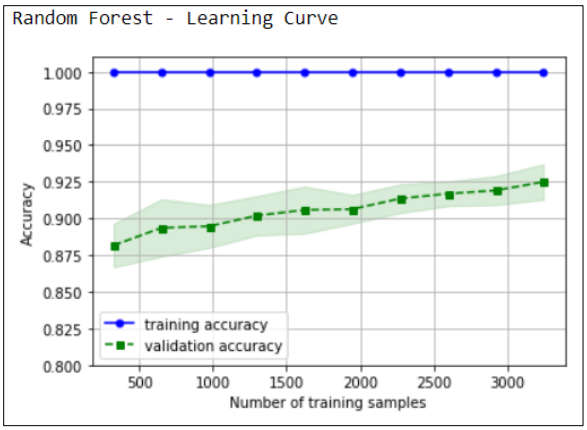


Figure 6

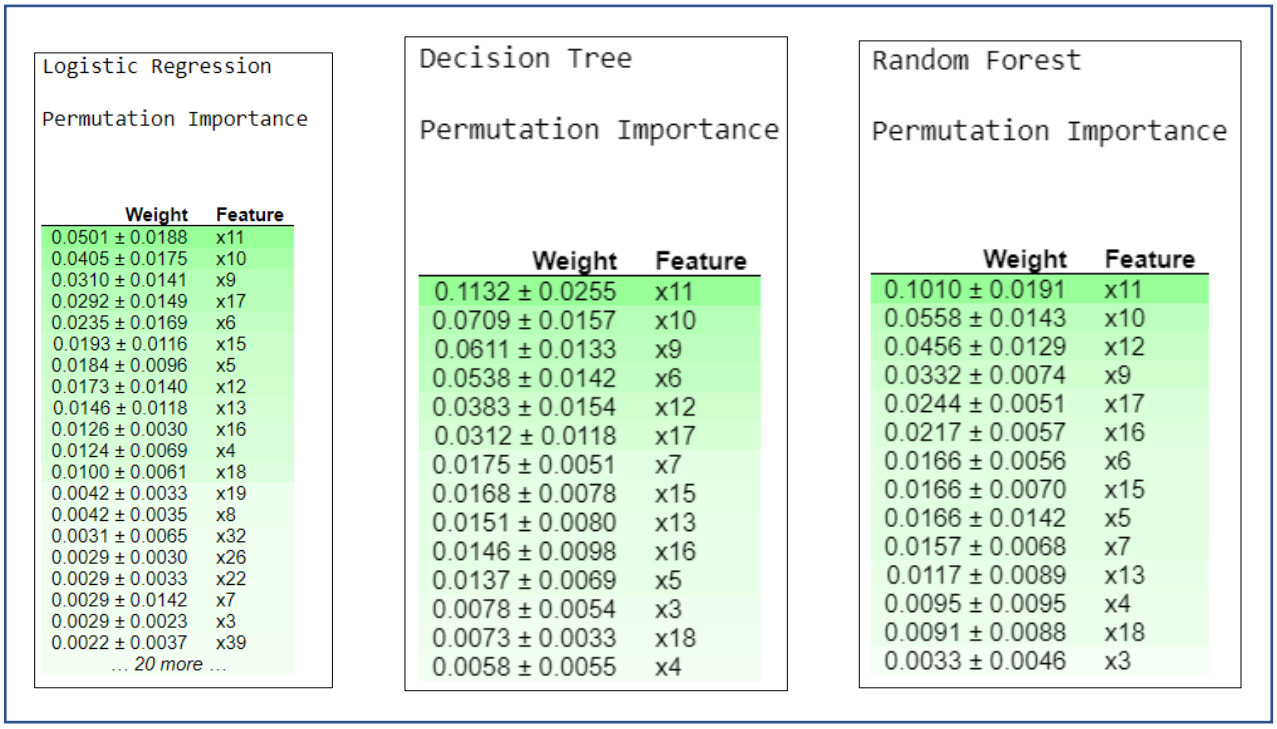
```
[[421  34]
 [ 50 398]]

              precision    recall  f1-score   support

         0.0       0.89      0.93      0.91       455
         1.0       0.92      0.89      0.90       448

    accuracy                           0.91       903
   macro avg       0.91      0.91      0.91       903
weighted avg       0.91      0.91      0.91       903


NestedCV Accuracy(weighted) :0.92 +/-0.02
NestedCV Precision(weighted) :0.92 +/-0.02
NestedCV Recall(weighted) :0.92 +/-0.02
```

Figure 7: Optimized Random Forest

[1]Bootstrap - method is a technique for making estimations by taking an average of the estimates from smaller data samples.

D. Model Comparison

From the 3 algorithms discussed, the random forest algorithm has the highest nested CV precision and accuracy score at 0.92. Regarding feature importance, feature x11, x10, x9, x17, and x6 are consistently ranked the highest in terms of weights in the model.



Logistic Regression

Permutation Importance

Weight	Feature
0.0501 ± 0.0188	x11
0.0405 ± 0.0175	x10
0.0310 ± 0.0141	x9
0.0292 ± 0.0149	x17
0.0235 ± 0.0169	x6
0.0193 ± 0.0116	x15
0.0184 ± 0.0096	x5
0.0173 ± 0.0140	x12
0.0146 ± 0.0118	x13
0.0126 ± 0.0030	x16
0.0124 ± 0.0069	x4
0.0100 ± 0.0061	x18
0.0042 ± 0.0033	x19
0.0042 ± 0.0035	x8
0.0031 ± 0.0065	x32
0.0029 ± 0.0030	x26
0.0029 ± 0.0033	x22
0.0029 ± 0.0142	x7
0.0029 ± 0.0023	x3
0.0022 ± 0.0037	x39
… 20 more …	

Decision Tree

Permutation Importance

Weight	Feature
0.1132 ± 0.0255	x11
0.0709 ± 0.0157	x10
0.0611 ± 0.0133	x9
0.0538 ± 0.0142	x6
0.0383 ± 0.0154	x12
0.0312 ± 0.0118	x17
0.0175 ± 0.0051	x7
0.0168 ± 0.0078	x15
0.0151 ± 0.0080	x13
0.0146 ± 0.0098	x16
0.0137 ± 0.0069	x5
0.0078 ± 0.0054	x3
0.0073 ± 0.0033	x18
0.0058 ± 0.0055	x4

Random Forest

Permutation Importance

Weight	Feature
0.1010 ± 0.0191	x11
0.0558 ± 0.0143	x10
0.0456 ± 0.0129	x12
0.0332 ± 0.0074	x9
0.0244 ± 0.0051	x17
0.0217 ± 0.0057	x16
0.0166 ± 0.0056	x6
0.0166 ± 0.0070	x15
0.0166 ± 0.0142	x5
0.0157 ± 0.0068	x7
0.0117 ± 0.0089	x13
0.0095 ± 0.0095	x4
0.0091 ± 0.0088	x18
0.0033 ± 0.0046	x3

Figure 8: Comparative Feature Selection Results

Results

Random forest has the best results, but these results need to be proceeded with caution because of the current overfitted results from the learning curve. The Decision tree had the lowest results and prone to overfit. These first two models generalize the model, by not accounting training data. While Logistic regression had better precision and recall scores that wasn't prone to overfitting or underfitting.

Lastly, Features x11 and x10 have scaled the highest in feature importance for heart measurements.

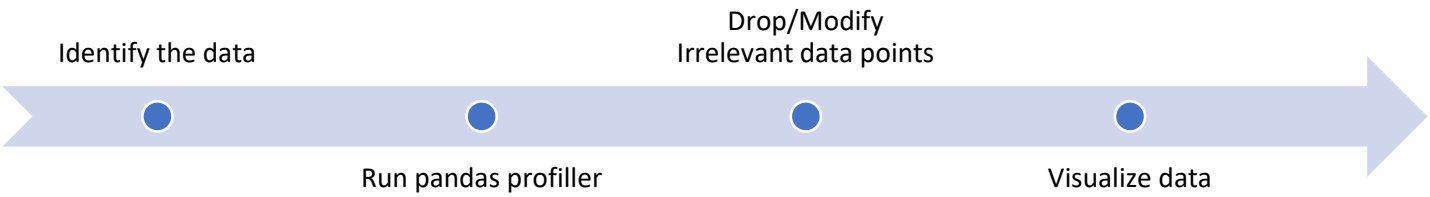
Conclusion

In conclusion, an ensemble model like random forest performed the best amongst the three models. Although prone to overfitting, Analyst could investigate more in revising the hyperparameter tuning method of the algorithms and include more relevant training data. Features x11 and x10 have consistently been ranked as the most important feature for heart measurements. In other words, slight fluctuations in these heart measurements would be critical to a patient's heart diagnosis.

Appendices

Appendix A - Full Dataset Summary

a. Action Plan



The following are the detailed descriptions of the Action Plan steps:

Step 1: Identify the data

This can be done by executing simple statistics and data information in python. Identify the dependent and independent variables.

RESULTS:

	x1	x2	x3	x4	x5
count	5001.000000	5000.000000	5000.000000	5000.00000	5000.000000
mean	-0.018634	0.328100	0.653080	1.00654	1.357060
std	1.007407	1.044018	1.183895	1.42673	1.673702
min	-3.500000	-3.200000	-3.100000	-3.30000	-3.500000
25%	-0.700000	-0.400000	-0.200000	0.00000	0.100000
50%	0.000000	0.300000	0.600000	0.90000	1.200000
75%	0.700000	1.000000	1.400000	2.00000	2.500000
max	3.400000	4.300000	4.800000	5.70000	6.700000

Figure 1

From figure 1, the dataset is imbalanced and has inconsistent fluctuations in standard deviations. Identifying the dependent and independent variables. – X1 to X40 are independent variables, although the column "classes" is the only dependent variable in the dataset.

Step 2: Run Pandas profiler

Afterwards run the dataset through pandas profiler (pandas profiler is a separate library that needs to be downloaded for this code to run). Pandas profiler allows you to identify information outside of simple statistics and data type information. This includes: Warnings, Attributes, visualization of items (i.e. column headers, heatmap of correlation) etc. Pandas profiler also helps identify missing values.

RESULTS: pandas profiling report, primarily for its quick insight into the dataset. The following are the results that need immediate action:

a) Missing Cells – 450 (0.2%)
b) Duplicate Rows – 9 (0.2%) Duplicate rows are not dropped in this instance because each row signifies a patient in the dataset. Having similar measurements with another patient should remain in the data

Step 3: Drop/Modify Irrelevant data

Cleaning the data would be broken down to 3 sub-processes:
Step 3a: First, check for missing values and drop values (if applicable)
Step 3b: Check for any outliers and remove (if applicable)
Step 3c: Check if dataset is balanced and perform SMOTE (if applicable)

RESULTS:

Step 3a: Check for missing values – There were 450 missing values or NaNs identified in Python. Although 450 missing values accounts 0.2% of the dataset. The missing values were dropped. Primarily for rows that were completely NaNs. There also one patient that registered only 1 feature out of the 40, particularly the X1 measurement. Therefore, the last patient was dropped from the dataset due to lack of information.

Step 3b: Check for outliers – There were 742 outliers identified through the turkey method. After removing the dataset, there were 4258 patient measurements left.

Step 3c: Check if dataset is balanced – The data is imbalanced, therefore there is a need to balance the dataset first. The Researchers did SMOTE to balance the dataset Making the total dataset at 5292 items.

Step 4: Visualize Data

Visualize the data through boxplot and a correlation heatmap. This allows outliers to be highlighted.

RESULTS:

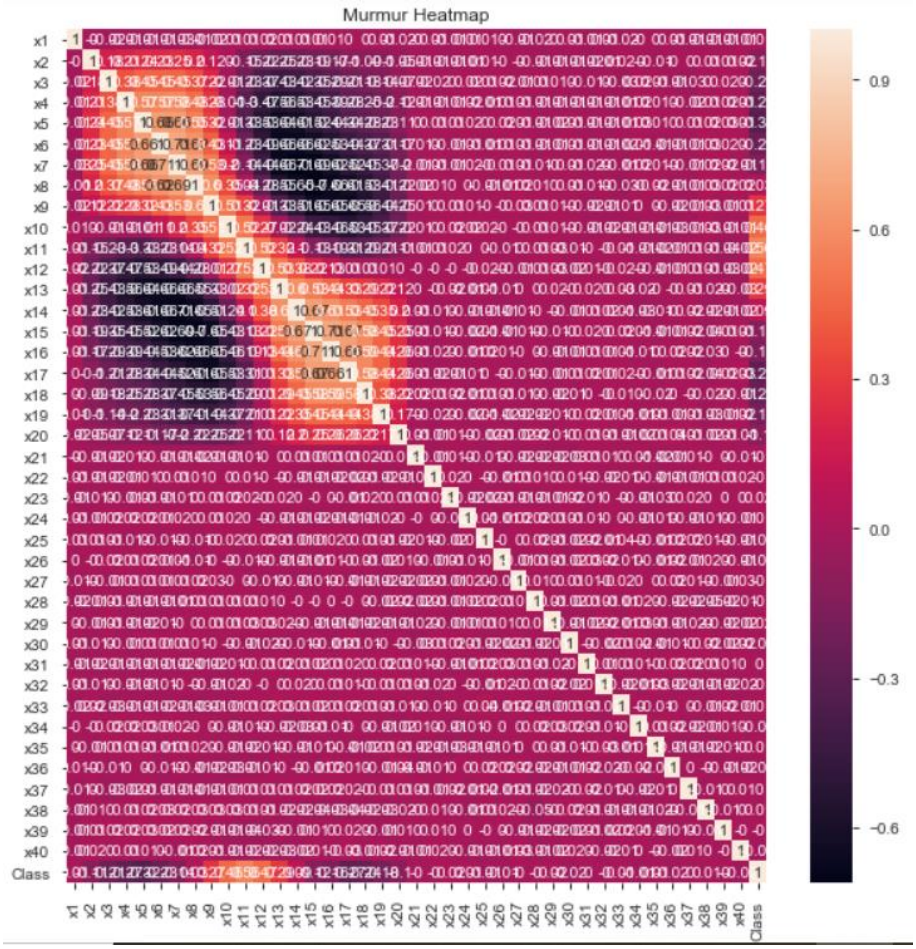


Figure 2

From figure 2, we can identify that there is an extreme range from high to low correlation in the X1 to X20 features. While all other features remain relatively low in terms of correlation.

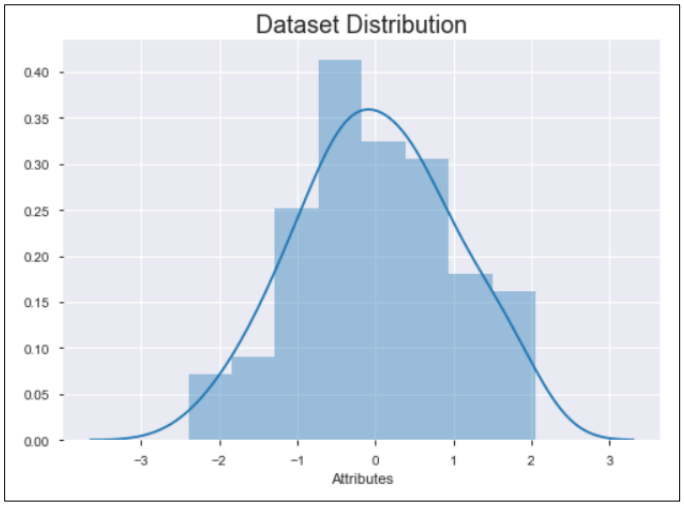


Figure 3

From figure 3, we can identify that the dataset is normally distributed. Although there seems to be a sharp spike in 0.40 and -0.5 features in the dataset.

b. Assumptions

There are three assumptions about the dataset:

1. That the data has been vetoed and cleaned
2. That the Murmur data is based on valid individual patient data
3. That the data is normally distributed, unless proven otherwise through the EDA (Exploratory Data Analysis).

c. Constraints

Basically, a hard constraint has a strong negative impact to the model, a strong negative impact would mean a need to remodel the data if new attributes or information were added. While the soft constraint would have a weaker effect to the model, meaning if there is new information there is no need to remodel the dataset. There are (3) three constraints about the data and within these constraints we can identify (2) two hard constraints and (1) one soft constraint, these are:

1. The dataset cannot be increased. The data was forwarded by the client, if the client forwards additional data this would force the Researchers to perform EDA and data remodeling again. This is categorized as a hard constraint.
2. The dataset cannot have additional features. Again, this data is from the client forcing the researchers to work with the current data. Any additional features not from the client would damage the dataset. In retrospect, if the client also forwarded additional features this would force the Researchers to do EDA and data remodeling again This is also a hard constraint.
3. Limited Knowledge of the dataset. Although this constraint causes a slight inconvenience, if there was more knowledge about the data, depending on the extent of the information, this would still not require the Researchers to recreate EDA and data remodeling. This is a soft constraint.

Appendix B – Assessment of Predictive Models

Precision – What percent of predictions were correct?

Precision is the ability of a classifier not to label an instance positive that is negative. For each class it is defined as the ratio of true positives to the sum of true and false positives.

TP – True Positives
FP – False Positives

Precision – Accuracy of positive predictions
Precision = TP/(TP + FP)

Recall – What percent of the positive cases did you catch?

Recall is the ability of a classifier to find all positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives.

FN – False Negatives

Recall: Fraction of positives that were correctly identified
Recall = TP/(TP+FN)

		Predicted Class: Positive	**Predicted Class: Negative**	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	**Negative Predictive Value** $\frac{TN}{(TN + FN)}$	**Accuracy** $\frac{TP + TN}{(TP + TN + FP + FN)}$