Midterm Report 1030

Shubham Makharia

October 2021

1 Introduction

This project explores the machine learning pipeline by working on Heart Failure Data, found on Kaggle. The target variable we classify is HeartDisease, a binary datatype where 1 indicates the person has Heart Disease, and 0 (normal) otherwise. Heart disease is the leading cause of death in the world, and machine learning models that can predict which persons are at a high risk for heart disease early can help spur action to increase the likelihood of positive health outcomes for those users, in addition to encouraging persons who might not be at risk currently to make pre-cautionary lifestyle changes. This dataset comprises 918 persons, recording 12 observations each. This datset is already described as follows (the annotation (CTS) means that attribute is continuous, while (CAT) indicates the attribute is categorical, and (ORD) indicates ordinal attribute):

- Age: age of the patient (years) (CTS)
- Sex: sex of the patient (M: Male, F: Female) (CAT)
- ChestPainType: chest pain type (TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic) (ORD)
- RestingBP: resting blood pressure (mm Hg) (CTS)
- Cholesterol: serum cholesterol (mm/dl) (CTS)
- FastingBS: fasting blood sugar (1: if FastingBS > 120 mg/dl, 0: otherwise) (ORD)
- RestingECG: resting electrocardiogram results (Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria) (CAT)
- MaxHR: maximum heart rate achieved (Numeric value between 60 and 202) (CTS)
- ExerciseAngina: exercise-induced angina (Y: Yes, N: No) (CAT)

- Oldpeak: oldpeak = ST (Numeric value measured in depression) (CTS)
- STSlope: the slope of the peak exercise ST segment (Up: upsloping, Flat: flat, Down: downsloping) (ORD)
- HeartDisease: output class (1: heart disease, 0: Normal)

Looking at the various public project submissions on Kaggle, we see all of them attempt to solve the same sort of classification problem. The two models I saw most frequently were RandomForestClassifiers and Logistic Regression (sometimes w/ PCA), and all of the public projects seemed to achieve accuracy scores of at least 80%.

2 Exploratory Data Analysis

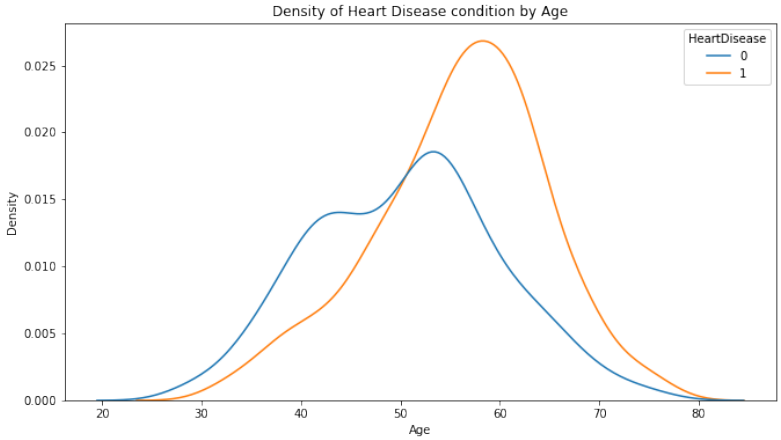

Figure 1: The distribution of Heart Disease patients centers around individuals in their upper 50s, while Normal patients are younger, and have more variance.

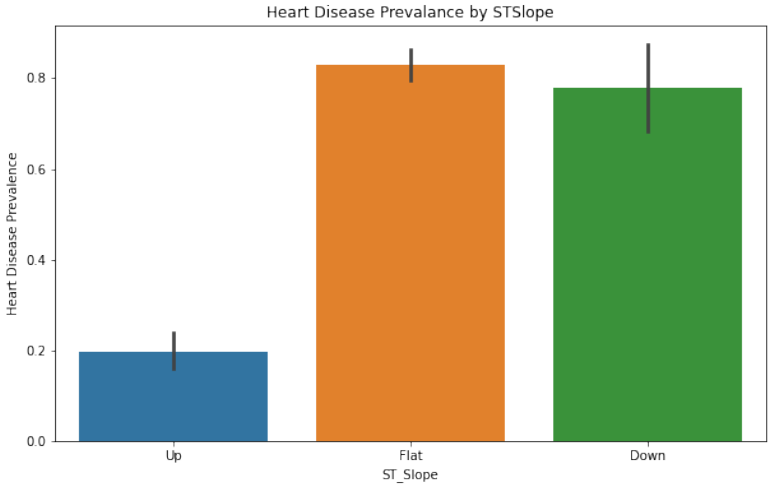

Figure 2: We see Heart Disease patients in general experience more depressed peak slopes

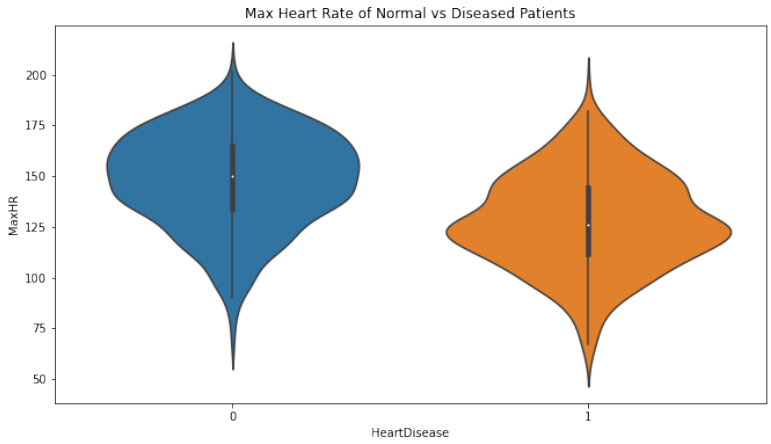

Figure 3: The distribution of heartrate amongst Normal and HeartDisease patients has similar variance, but the mean is much higher for Normal patients

3 Methods

3.1 Data Preprocessing

We split the dataset according to the following assumptions: There is no group structure, the data is IID, and the data is not time-series. We aim to split the dataset into 20% testing, and 80% other. The other set will be split into 25% for validation, and 75% for training. According to the annotations in the description, we applied a StandardScaler to the continuous variables, since we wanted to preserve the various values without skewing the training by the actual values. We used a OneHotEncoder for categorical variables, and we used an OrdinalEncoder for ChestPainType, so that the training could leverage the fact that the severity of a patient's chest pain. The original data consisted of 11 features, but the encoding increased the number of variables to 16. Additionally, it was observed that two features contained missing values: RestingBP and Cholesterol. RestingBP only had 1 missing value, and Cholesterol had hundreds of missing values. These missing values were stored as 0, so preprocessing additionally required replacing these with NaN values.

3.2 Machine Learning Pipeline

The machine learning pipeline implemented on this classification was a reduced features model to account for the missing values in Cholesterol and RestingBP. After applying a training-validation-testing split as described in the previous subsection, the model was trained according to a reduced features folding of the data. After identifying the unique missingness patterns in the data, the data was trained only on samples that exhibited the same missingness pattern. For example, the data with missing only Cholesterol values in the training data was the only data used to generate predictions on the validation and testing sets, for the data in those sets that were also missing only Cholesterol values, while testing/validation data with no missing values were predicted by training a model with only training data that also had no missing values.

We apply this pipeline to the data using 4 different models. The models used are XGBoost, Random Forests, Support Vector Machines, and Logistic Regression. We describe the parameters of interest and metrics for evaluation for each model. Since we are using a reduced features model, we can be fairly confident in the model identifying ideal parameters without KFold, since the missingness pattern slicing in reduced features models creates a fold of its own.

In order to assess a model's performance, we use an accuracy score to compare the predictions of the model to the actual condition of each patient. Accuracy seems to be the best suited metric given the context of the problem: We would hope to deploy this model in a healthcare setting, and we only want to deploy models with the highest accuracy.

3.2.1 XGBoost

This model was subjected to the reduced features pipeline, although in general XGBoost performs similarly well to a more standard pipeline on data with missing values. The parameters of interest with XGBoost are the learning rate, the number of estimators, and the maximum depth. The values investigated for these parameters were:

- Learning Rate: .01, .05, .1
- Estimators:60,80,100,120,140,160,180,200
- Maximum Depth:2,3,4,5,6,7,8,9

XGBoost is non-deterministic, so we further verify our confidence in the results by running XGBoost for 10 random states, and we examine the distribution of accuracy scores. If the histogram has low (but non-zero) variance, that reduces the degree of uncertainty we associate to the model. We might not know the underlying mechanics of this model, but we are sure that it is developing a consistent framework for generating predictions.

3.2.2 Random Forests

The parameters of interest with Random Forests are the maximum number of features each tree can consider before suggesting a split, and the maximum number of splits allowed before the model is stopped (regardless of convergence). The values investigated were:

- Maximum Features: 1, 3, 5, 10
- Maximum Depth: 1, 3, 5, 10, 20

Random Forests are non-deterministic, so we measure uncertainty with an approach identical to the one described when using XGBoost.

3.2.3 Support Vector Machines

The parameter of interest with SVC is C, a regularization constant that is inversely proportional to how significant the l2 penalty is in regularization.

- C = linspace(0.1,1,10)

While SVM is deterministic, we still measure uncertainty to generate confidence in the model's results.

3.2.4 Logistic Regression

The parameters of interest with Logistic Regression are the regularization term C (similar to SVM), the type of penalty applied, and in the case of an elasticnet penalty, we introduce an additional parameter that indicates the ratio between the l1 and l2 regularization terms.

- C = logspace(-3,4,7) (.001 to 1000 in multiples of 10)
- penalty = l1, l2, and elastic net
- ratio (if elastic net is applied) = .01, 0.1, 0.25, 0.5,0.75, 0.9, 0.99

While Logistic Regression is deterministic, we still measure uncertainty to generate confidence in the model's results.

4 Results

After applying the pipeline to these 4 models, we then compared the accuracy of each model, accounting for splitting variances by including an error bar. The model performances can be summarized by the following figure.

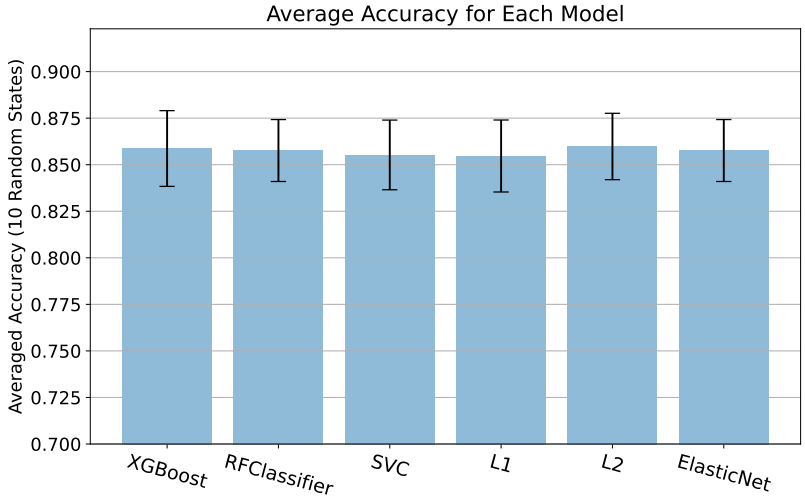

Figure 4: All of the models had fairly consistent performance, but L2-Logistic Regression edged out the rest.

We see that L2-Logistic Regression performed best, achieving global accuracy scores of 85.98%, exhibiting a standard deviation across random states of 1.66%. If we compare that to a baseline accuracy of 58%, then we quantify that L2-Logistic Regression outperforms the baseline by nearly 17 standard deviations.

Now, we seek to quantitatively describe feature importance from the results of the model. In order to do that, we employ three strategies. First, we examine the coefficients of the logistic regression model. Then, we apply feature permutation and apply L2 regression multiple times, one for each perturbed feature. Lastly, we use SHAP to generate model-agnostic description of feature importance. Additionally, we use SHAP to generate local feature importance. We can view the results of each of these measures in the following figures.

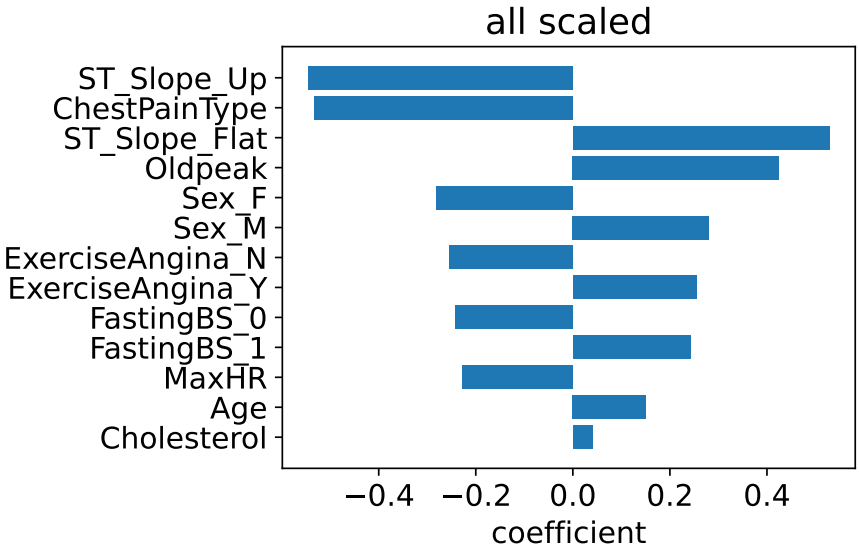

Figure 5: Ordered by magnitude, we see that the three leading coefficients in L2-Logistic Regression relate to the ECG of a patient or the severity of their chest pain.

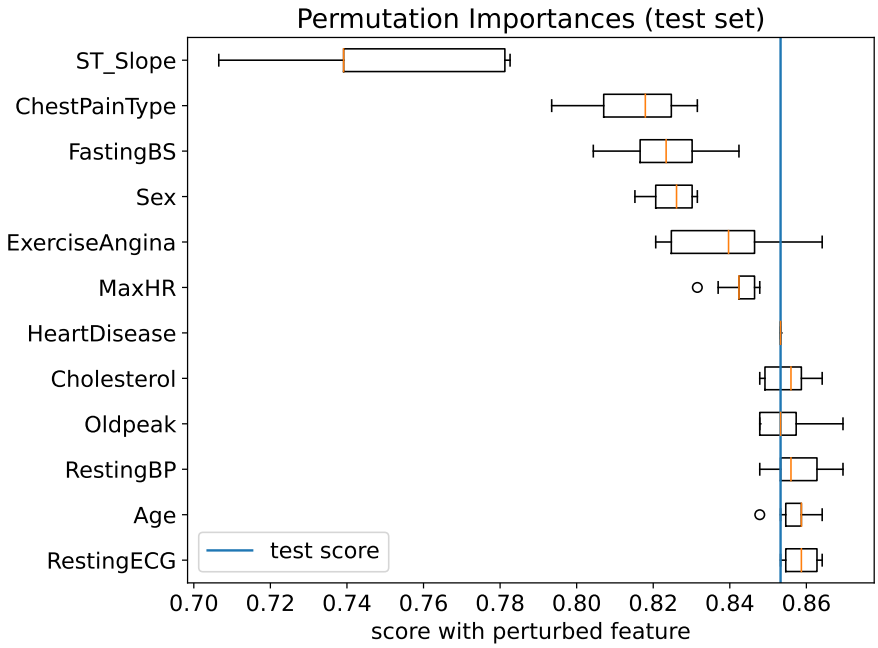

Figure 6: We note that the model performance tanks if the characteristic ECG data is removed from the dataset, but if other ECG data (Oldpeak, RestingECG) are removed, model performs similarly.

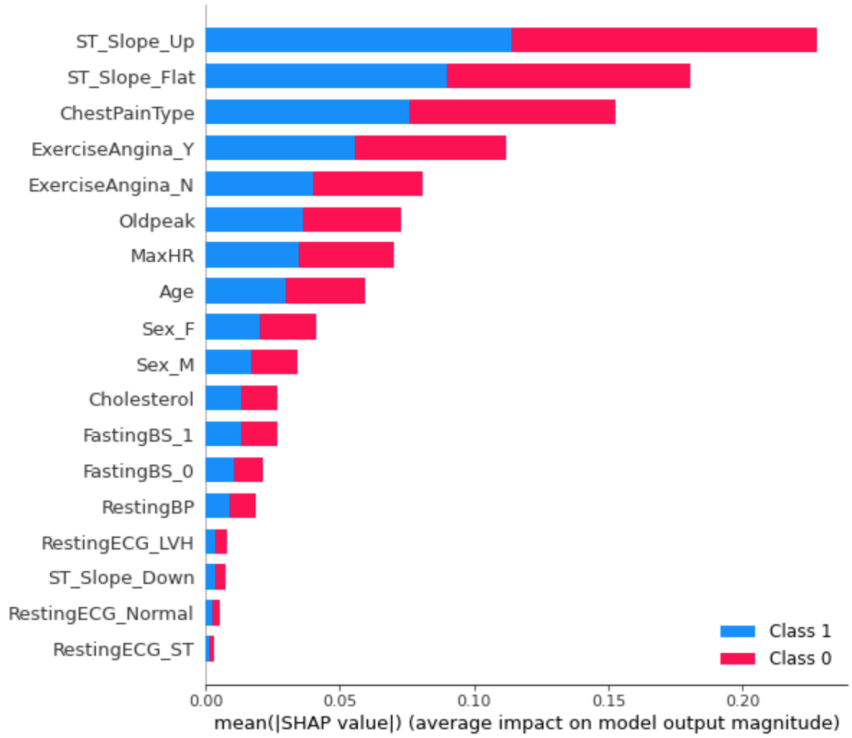

Figure 7: The SHAP Global Importance scores confirm the feature importances as illustrated in model coefficients and permutation importance.

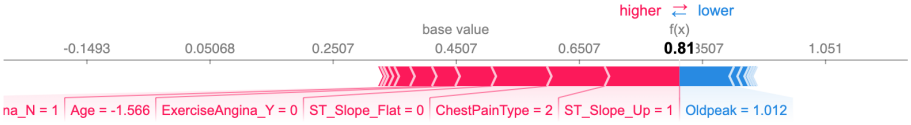

Figure 8: Local Feature Importance via SHAP

Examining the feature importance, we see that ST Slope is the most important feature, which was expected. Given that we are trying to deploy this model into a healthcare environment, we can use this to provide clinical decision support to providers. The model would suggest that understanding the ST Slope of a patient should be a priority for providers

The ST Slope is measured from taking an ECG of the patient, and when doctors are determining heart failure, they already give utmost priority to collecting an ECG and performing a similar analysis manually. While this model does not make a case for changing decisions, we can rest assured knowing we've strengthened the null hypothesis just a bit more. It was a bit surprising to see that only specific features related to ECG were important, like slope, while other measures, like RestingECG or OldPeak were very insignificant to strengthening the model. ECGs are highly accurate measurements, so I expected the entire measurement to matter to the model, but the model is able to distinguish that only certain segments of the model are critically important, just as doctors know to only pay attention to the ST segment of an ECG.

5 Outlook

The biggest hole in this entire pipeline and an opportunity for improvement is in the consideration of values with respect to Type I and Type II errors. For most patients with heart disease, the intervention prescribed is surgery in extreme cases, and lifestyle changes for the moderate cases. So, we expect that the cost of missing a patient with heart disease is far greater than the cost of misdiagnosing a healthy patient with heart disease, as a misdiagnosed patient is almost certain to be prescribed a lifestyle change initially, which carries no meaningful harm. Thus, we have a vested interest in minimizing Type II errors, while accepting some Type I errors. We assess the current best model's errors by displaying its confusion matrix.

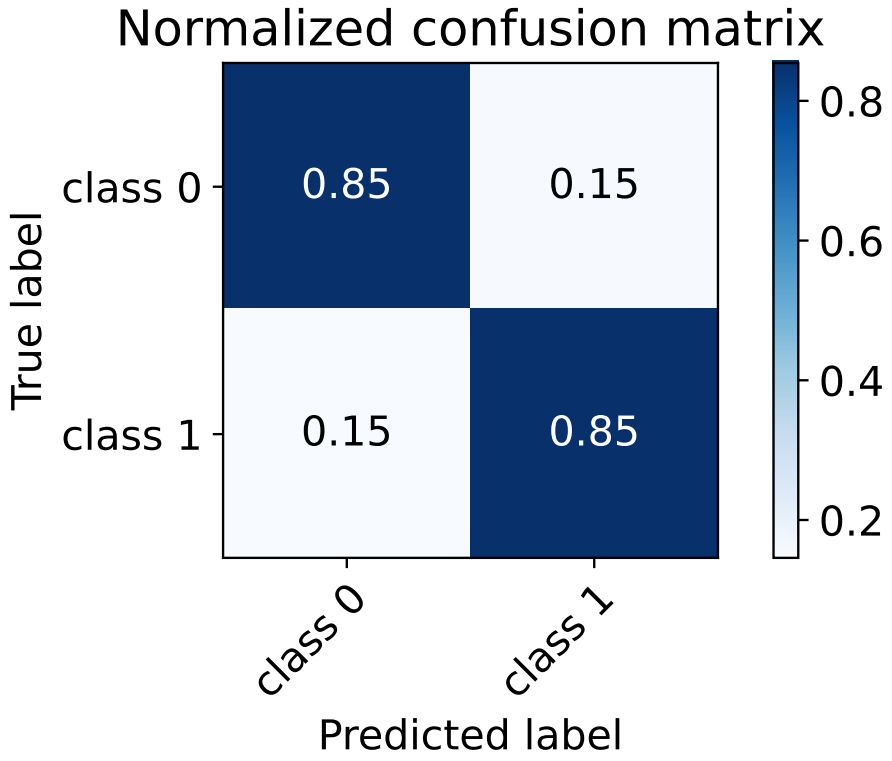

Figure 9: Confusion Matrix of L2 Regression

We see that currently the model is equally susceptible to Type I and Type II errors. We could seek to improve this model by plotting a Recieving Operator Characteristic curve, and determine an optimal threshold for predictions given our values in relation to this medical context. That would lead us to creating a more effective model that minimizes Type II errors, but its difficult to see if this would lead to better predictions in practice. If our dataset was larger, it would be easier to determine an optimal threshold. Apart from collecting more data by increasing that patients included in the study, we could theorize building a more robust prediction model by cultivating a dataset that incorporates other tests that providers use when assessing cardiac health. For example, this could be data measured through MRI or some other imaging technique.

Other techniques we could explore to minimize Type II errors is by introducing a highly specific loss function that introduces a more severe penalty when misclassification leads to a Type II error, and is lighter in the case of a Type I misclassification. This could be implemented as an adjusted update rule for these models, and could work fairly well, although we would not be able to rely on existing modules like sklearn to achieve this.

6 References

fedesoriano. (September 2021). Heart Failure Prediction Dataset. Retrieved 6 October 2021 from https://www.kaggle.com/fedesoriano/heart-failure-prediction.

7 Github

https://github.com/makhas/midterm-1030