Predicting Heart Disease Using Patient Health Data

Andrew Luo - *Brown University* - GitHub

1. Introduction

Heart disease is one of the leading causes of death globally, accounting for approximately one third of all deaths [6]. In particular, coronary artery disease is a condition in which the buildup of plaque in the coronary arteries reduces their diameters through a process called atherosclerosis [4]. As a preventable disease that may be mitigated from early detection and treatment, being able to predict coronary artery disease in patients could help them get a formal diagnosis, seek treatment, and achieve better health outcomes [5]. So, this is an impactful classification problem.

1.1 Heart Disease Dataset

The Heart Disease dataset from the UC Irvine ML Repository is a collection of datasets from four locations including Cleveland, Budapest, Zurich, and Long Beach. Each of the datasets contains 76 features, but only the 14 most frequently used features are included in the "processed" datasets. The target variable is the diagnosis of heart disease where a value of 0 indicates less than 50% narrowing while a value of 1 indicates more than 50% narrowing as a result of coronary artery disease. For the purposes of this project, I combined the "processed" Cleveland, Budapest, and Long Beach datasets into a single general heart disease dataset [7].

1.2 Existing Research

There is a wide variety of existing research featuring the Heart Disease datasets. However, different works often choose to use different datasets, different features, or different machine learning models. One study, using only the Cleveland dataset, found that the SVM model had the best accuracy of 84.12% [2]. Another study, using a set of 25 features from the Heart Disease dataset but data from 3 hospitals instead, found that the SVM model had the best F1 score of 69.6% [1]. Lastly, one other study, using the Cleveland dataset for training, calculated a range of accuracies but found that their algorithms tended to over predict the probability of disease [3].

2. Exploratory Data Analysis

The heart disease dataset I generated contains 797 data points (which each represents a patient), 9 categorical features, 5 numerical features, and 1 target variable. The additional categorical feature that was not in the original dataset is the location feature which was added to indicate the location a data point was collected from. Additionally, this dataset does include missing values.

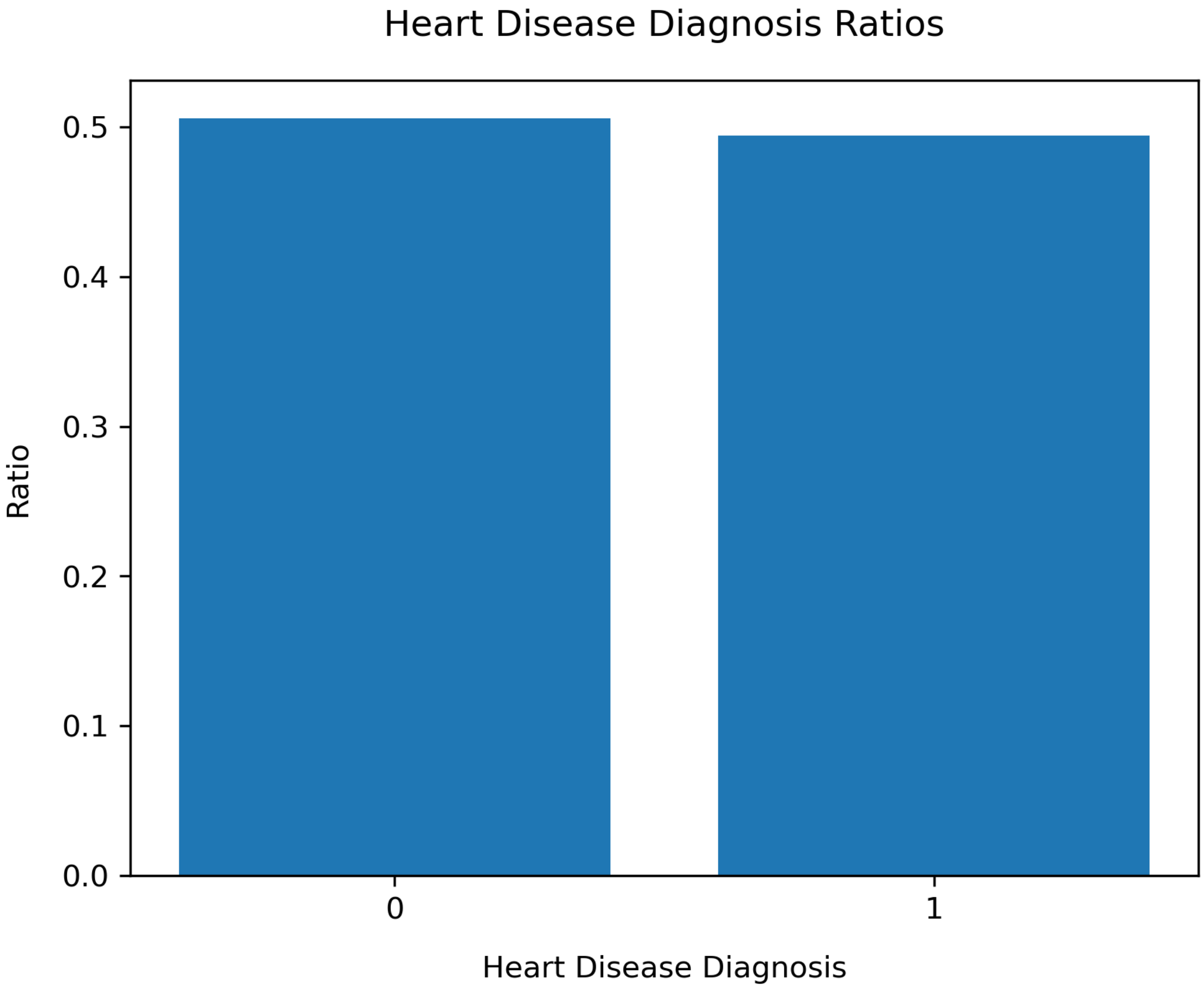
Figure 1. A table showing the features with missing values, their data types, as well as the fraction of missing values. There are 6 categorical features and 4 continuous features.

Feature	Data Type	Fraction Of Missing Values
trestbps	float64	0.072773
chol	float64	0.099122
fbs	object	0.018821
restecg	object	0.001255
thalach	float64	0.067754
exang	object	0.067754
oldpeak	float64	0.070263
slope	object	0.366374
ca	object	0.618570
thal	object	0.544542

2.1 Target Variable

First, I evaluated the target variable in the dataset and its balance by plotting its ratios as shown in Figure 2. Since the target classes have about a 50.56% - 49.44% split, the dataset is balanced.

Figure 2. A bar chart visualizing the ratios of the target classes. Since 50.56% are negative diagnoses and 49.44% are positive diagnoses, it is clear that the dataset is well balanced.

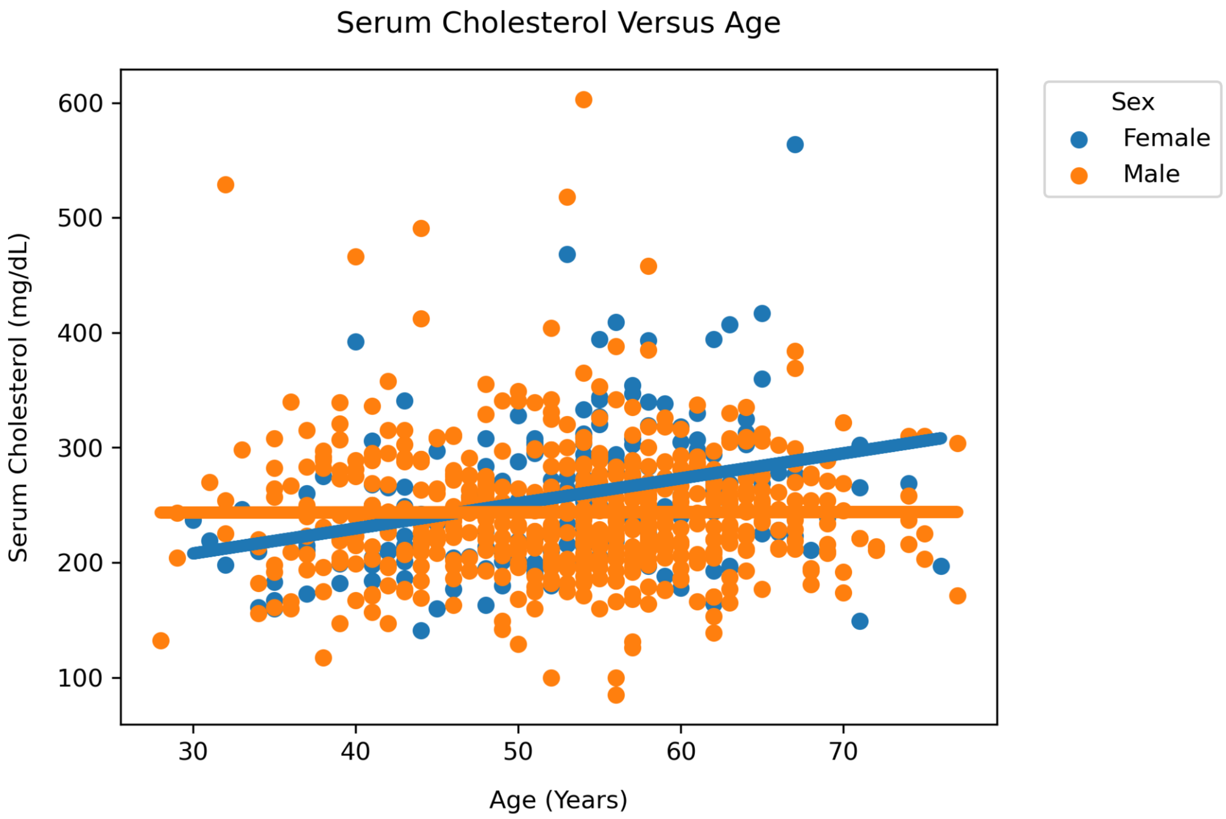


2.2 Feature Analysis

Additionally, I investigated the features in the dataset to gain insights that may not be obvious or intuitive without any proper exploration. Some of the most interesting findings are as follows.

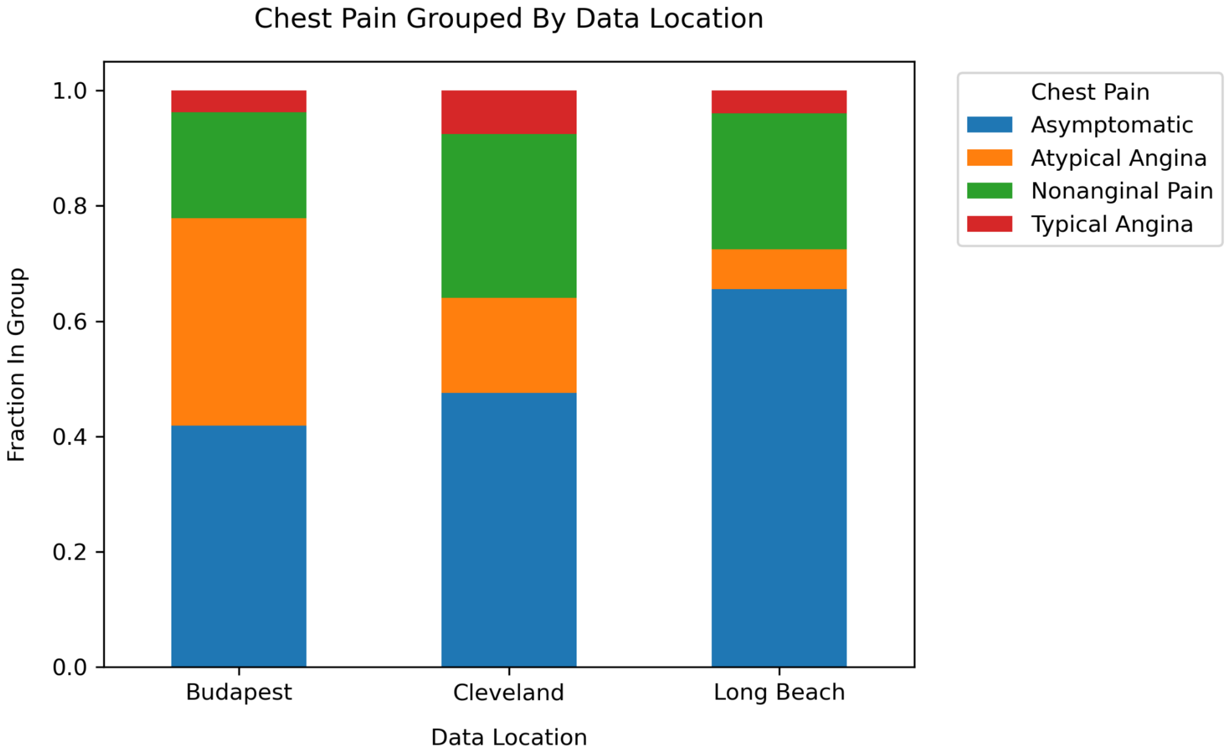
First, consider the box plot shown in Figure 3. Intuitively, one would expect cholesterol to increase with age. However, this is not the case for the overall dataset as the correlation between serum cholesterol and age is positive for females but negative for males. Consequently, the correlation between serum cholesterol and age for the overall dataset is incredibly low at 0.083.

Figure 3. A categorical scatter plot visualizing serum cholesterol versus age grouped by sex which shows that females and males have opposite serum cholesterol growths as they age.



Next, consider the stacked bar chart shown in Figure 4. It illustrates that patients in Budapest experience notably more atypical angina while patients in Long Beach tend to be asymptomatic.

<u>Figure 4.</u> A stacked bar chart visualizing chest pain grouped by data location which shows that patients in different locations will likely experience different types of chest pain.



3. Methodology

I developed two machine learning pipelines to train and evaluate the performance of 5 models including Logistic Regression, Random Forest, K Nearest Neighbors, Support Vector Machine, and XGBoost. The reason I developed two pipelines was mostly out of curiosity. Originally, I chose accuracy as the evaluation metric since it is easy to interpret and the dataset is well balanced. However, since the cost of false positives and false negatives are high when predicting a patient's heart disease diagnosis, I decided that F1 score might be the more appropriate metric.

3.1 Dataset Splitting

Since the dataset has group structure by location as well as balanced target classes, I used GroupShuffleSplit with 1 split, a test size of 0.2, and a random state of (42 * *i*) where *i* is the current trial to derive a 20% test set and 80% other set. Then, I used the 80% other set and GroupKFold with 2 splits for cross validation by passing them as arguments to GridSearchCV.

3.2 Preprocessing

Next, I implemented a preprocessing pipeline featuring a categorical transformer and numerical transformer. The categorical transformer used OneHotEncoder to preprocess the categorical features and StandardScaler afterwards for the LinearRegression model and SVM model (with a linear kernel) to improve the interpretability of their coefficients. The numerical transformer used StandardScaler to preprocess the continuous features and IterativeImputer beforehand to impute the missing values using a LinearRegression estimator for all of the models except XGBoost.

3.3 Cross Validation

The five models I trained using my two machine learning pipelines were optimized using GridSearchCV for hyperparameter tuning and cross validation. Additionally, by training and testing the models over 20 trials with a unique random state per trial, I was able to measure uncertainties from splitting and non-deterministic models. The parameter grids are in Figure 5.

Figure 5. A table showing the parameter grid and optimal parameters for each of the 5 models that were derived from the F1 score pipeline which uses GridSearchCV.

ML Model	Parameter Grid	Optimal
Logistic Regression	penalty = [l1, l2]	l2
	C = 1 / np.logspace(-2, 2, 21)	0.01
Random Forest	max_depth = [1, 3, 10, 30, 100]	30
	max_features = [0.25, 0.5, 0.75, 1.0]	0.25
K Nearest Neighbors	n_neighbors = np.arange(1, 100, 10)	31
	weights = [uniform, distance]	uniform
Support Vector Machine	C = np.logspace(-1, 1, 10)	0.46
	gamma = list(np.logspace(-1, 1, 10)) + ['scale']	0.10
XGBoost	learning_rate = [0.03, 0.05, 0.08, 0.1, 0.2]	0.03
	max_depth = [1, 3, 6, 10, 30, 100]	1

4. Results

As a benchmark, the baseline of the dataset was approximately 0.51. This number was calculated by dividing the majority target class by the total number of data points and verified using the DummyClassifier. Using the F1 score pipeline to measure model performance across 20 trials with different train, validation, and test set splits to account for uncertainty, the mean test score, standard deviation, and standard deviations above the baseline were calculated and collected in Figure 6. These results are graphically represented in Figure 7. Similarly, the results of the 5 models using the accuracy score pipeline are graphically represented in Figure 8 for reference.

Analyzing the results in Figure 6, Figure 7, and Figure 8, the mean F1 score of each of the 5 models is well above the baseline of 0.51. Additionally, the RF model had the best overall performance for both the F1 score pipeline and accuracy pipeline. The overall test performance confusion matrix is visualized in Figure 9, illustrating an overall false negative rate of 0.23.

Figure 6. A table showing the mean test scores, the standard deviations, and the number of standard deviations above the baseline for the 5 models derived using the F1 score pipeline.

ML Model	Mean Test Score	Standard Deviation	(Mean - Base) / STD
Logistic Regression	0.73	0.046	4.80
Random Forest	0.77	0.015	17.0
K Nearest Neighbors	0.74	0.039	6.00
Support Vector Machine	0.76	0.037	7.00
XGBoost	0.73	0.030	7.80

Figure 7. **A plot showing the test results for the 5 models derived using the F1 score pipeline. RF was the best performing model followed by SVM which was comparable.**

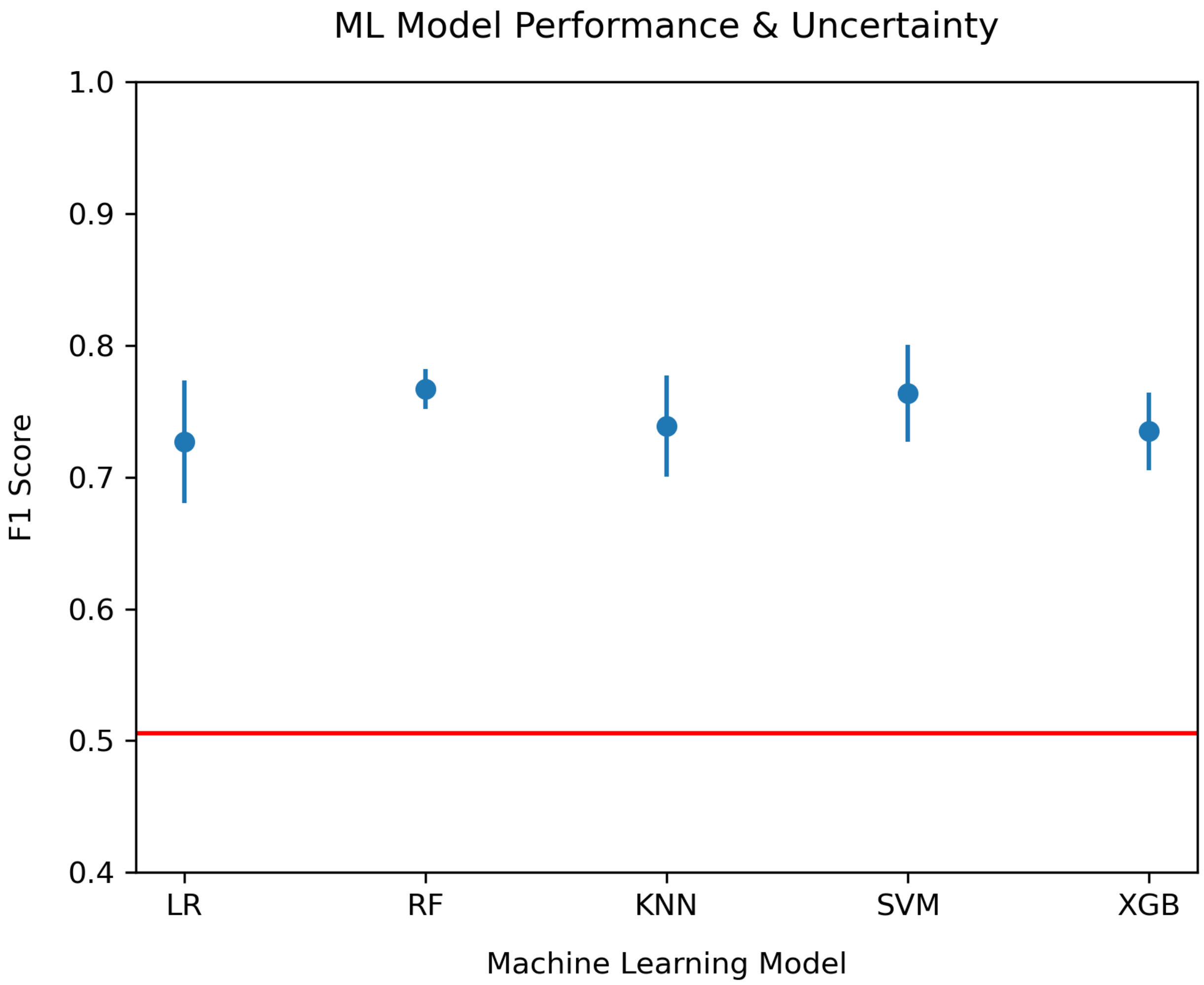


Figure 8. **A plot showing the test results for the 5 models derived using the accuracy score pipeline. SVM was the best performing model, followed by RF which was comparable.**

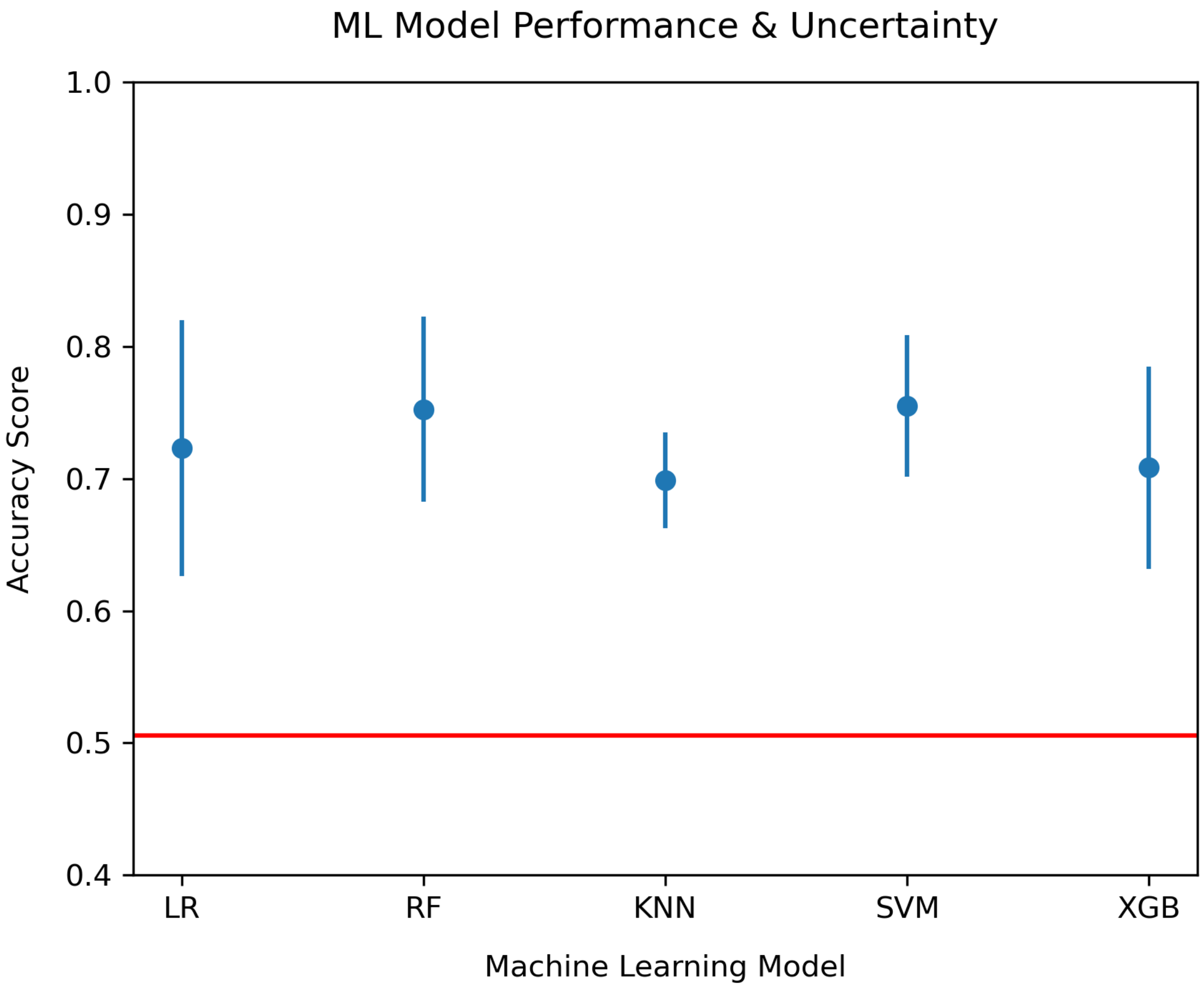
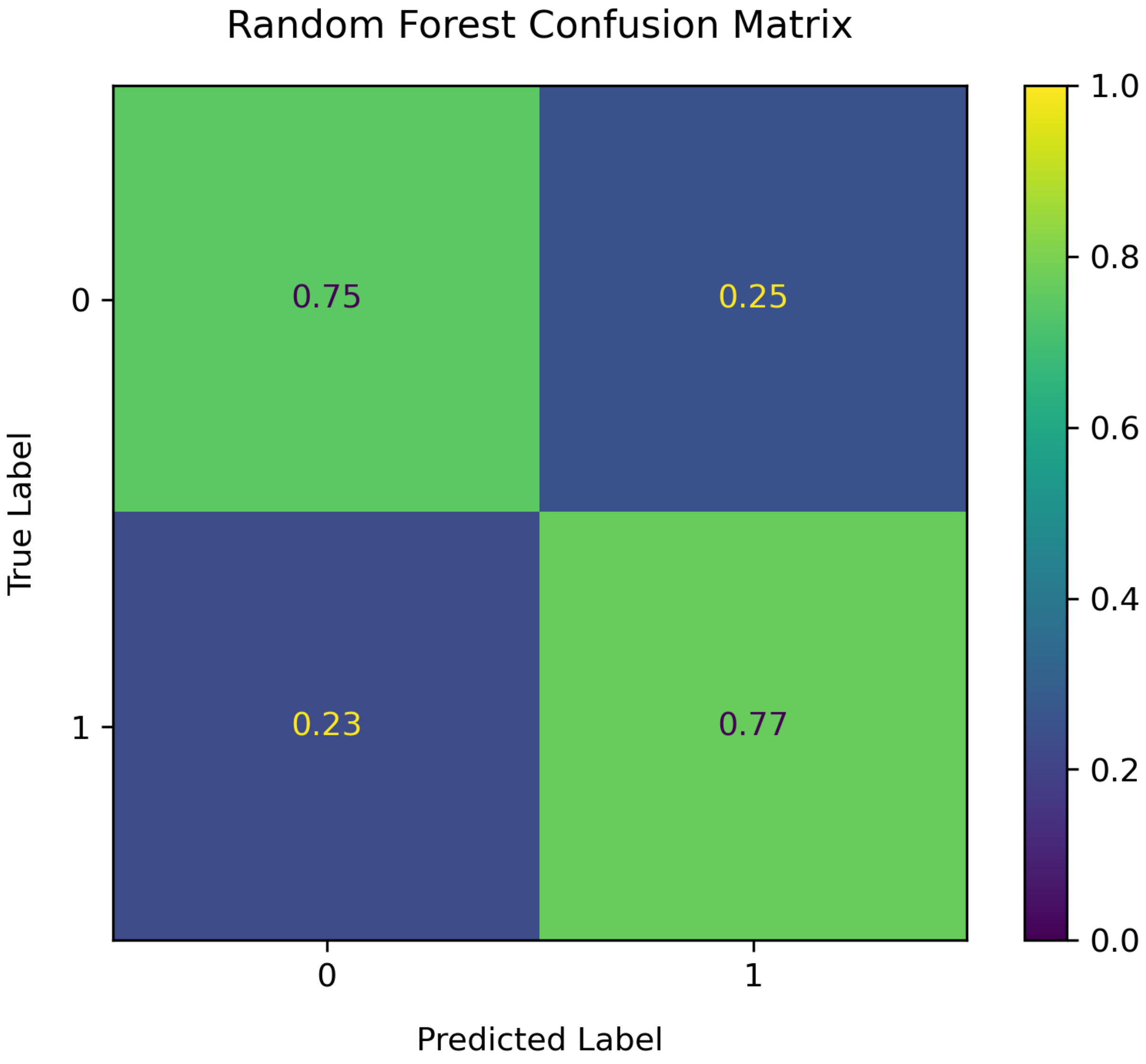


Figure 9. **A confusion matrix for the overall test performance of the RF model that was trained and evaluated using the F1 score pipeline. Note that the false negative rate is 0.23.**



4.1 Global Feature Importance

From Figure 10, Figure 11, and Figure 12, the global feature importance analyses reveal that *num__oldpeak*, *num__thalach*, *num__age*, and *cat__cp_Asymptomatic* are in the top 10 most important features for the RF model by permutation feature importance, SHAP value, and node impurity. However, the top 10 most important features still vary significantly between analyses.

Figure 10. A permutation feature importance plot visualizing the top ten most important features for the RF model that was trained and evaluated using the F1 score pipeline.

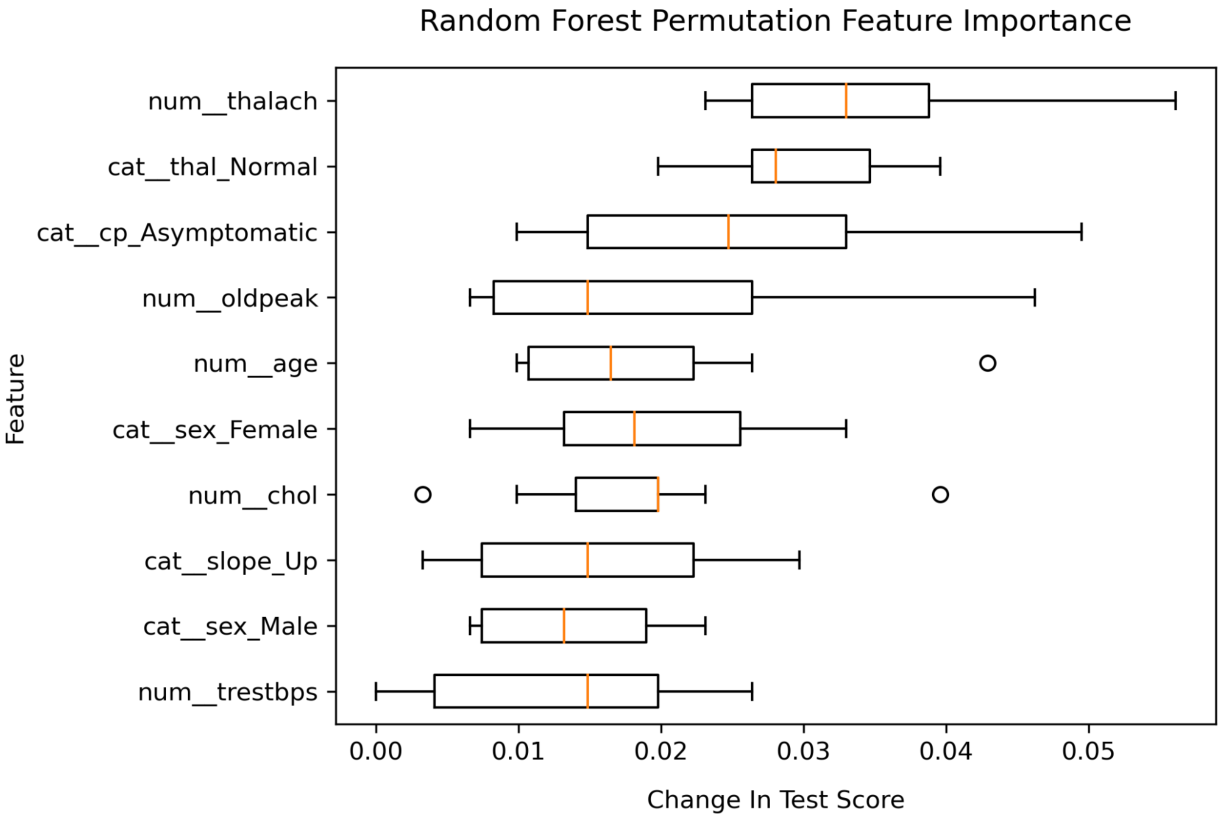


Figure 11. A SHAP beeswarm plot visualizing the top ten most important features for the RF model that was trained and evaluated using the F1 score pipeline by SHAP value.

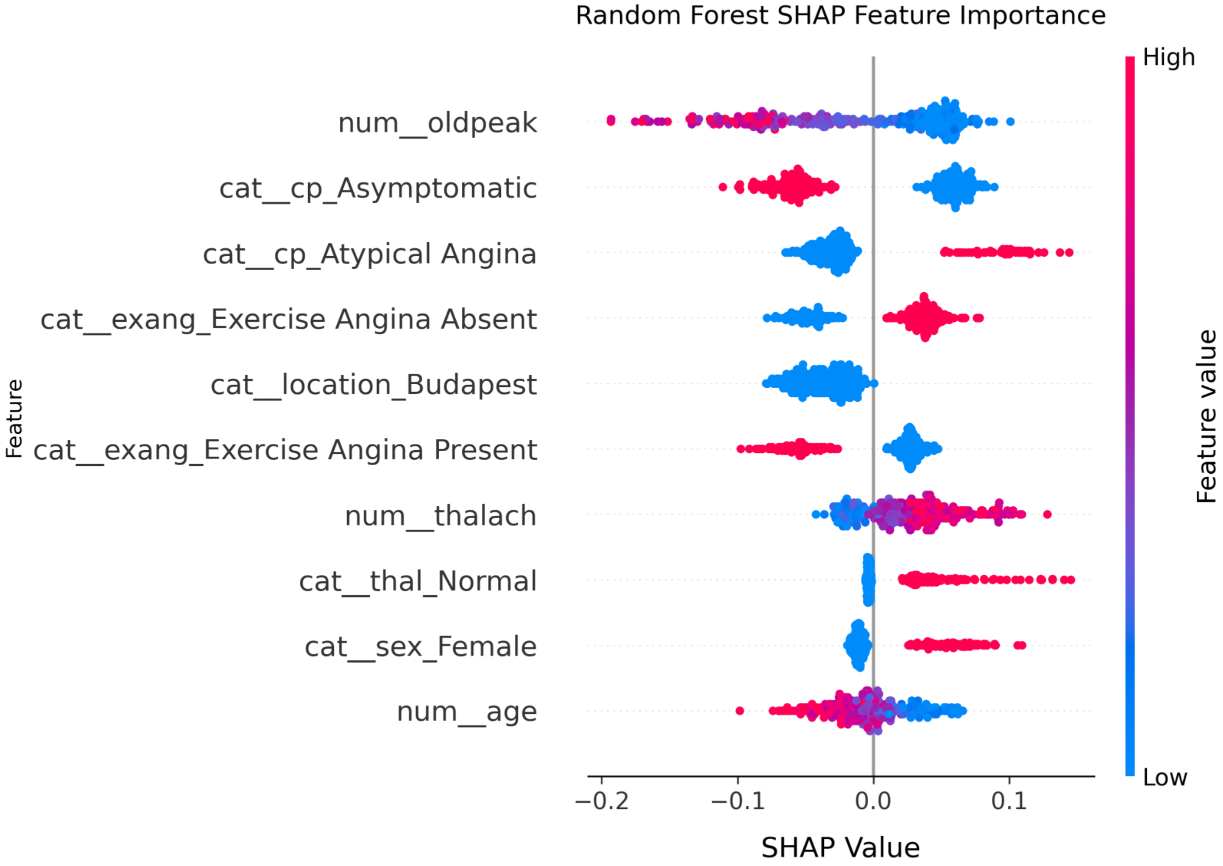
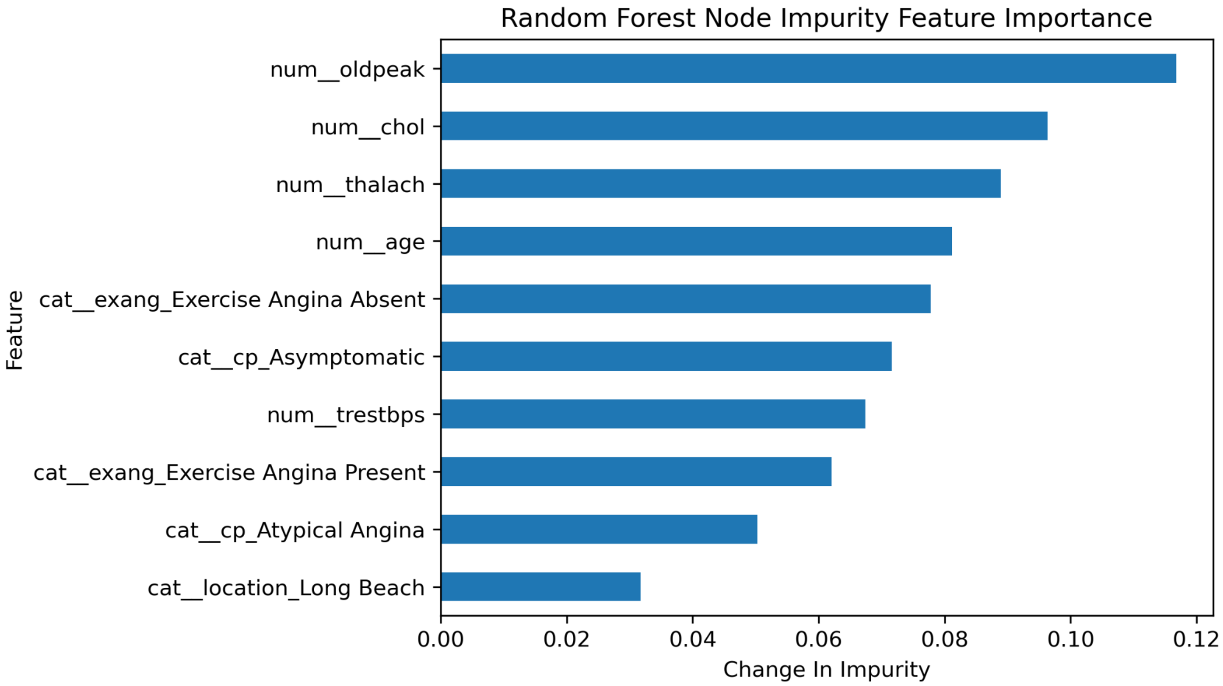


Figure 12. **A horizontal bar plot visualizing the top ten most important features for the RF model that was trained and evaluated using the F1 score pipeline by node impurity.**



4.2 Local Feature Importance

Figure 13, Figure 14, and Figure 15 are local feature importance analyses, helping us better understand how features contribute to a particular prediction. For instance, the force plot for the point at index 60 indicates that the patient is classified as class 0 with the greatest positive contribution coming from *cat__sex_Female* and the greatest negative contribution coming from *cat__exang_Exercise Angina Absent*. A similar analysis can be conducted for all other points.

Figure 13. **A SHAP force plot showing the most important features for the point at *i = 0*.**

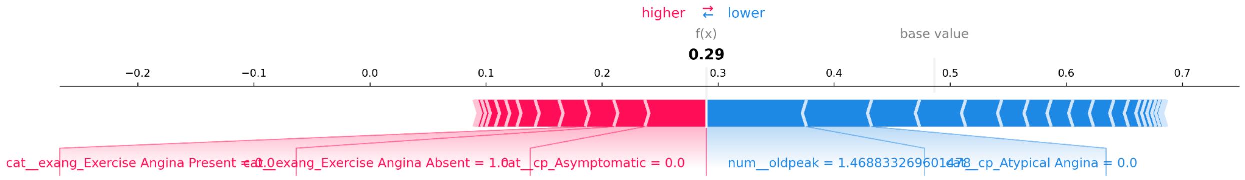


Figure 14. **A SHAP force plot showing the most important features for the point at *i = 60*.**

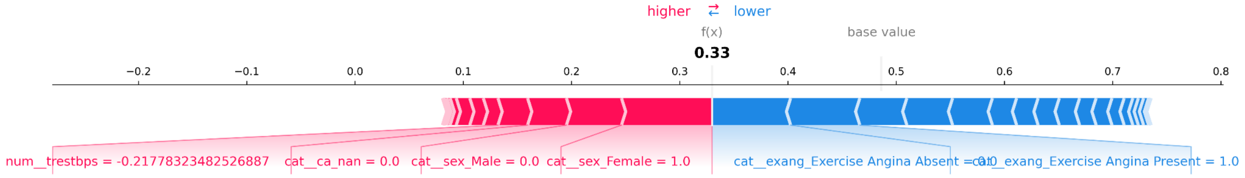
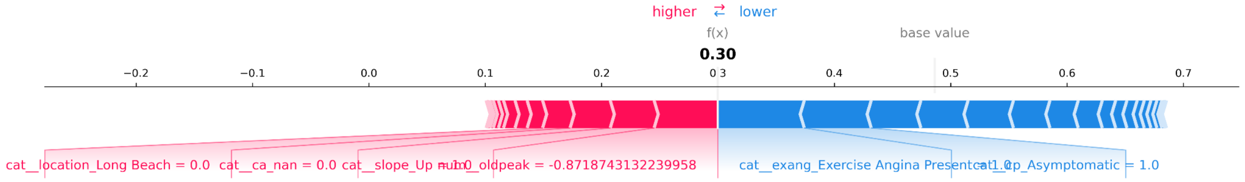


Figure 15. **A SHAP force plot showing the most important features for the point at *i = 120*.**



5. Outlook

There are multiple ways that this project could be improved. First, additional features and data could be included in the dataset. Since the unprocessed datasets contain 76 features, there is room to experiment with other features or even conduct feature engineering. Another notable way this project could be improved is by training additional models and continuing the parameter tuning of the existing models. One particular weakness of my modeling approach is that I use two general machine learning pipelines (one for accuracy and one for F1 score). So, the current implementation of XGBoost does not have early stopping. Lastly, I could recalculate feature importance by analyzing the correlation matrix of the models and dropping correlated features.

References

[1] Ayatollahi, H., Gholamhosseini, L., & Salehi, M. (2019). Predicting coronary artery disease: a comparison between two data mining algorithms. *BMC Public Health, 19*.

[2] Chaki, D., Das, A., & Zaber, M.I. (2015). A comparison of three discrete methods for classification of heart disease data. *Bangladesh Journal of Scientific and Industrial Research, 50*, 293-296.

[3] Detrano, R.C., Jánosi, A., Steinbrunn, W., Pfisterer, M.E., Schmid, J., Sandhu, S., Guppy, K., Lee, S., & Froelicher, V. (1989). International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American journal of cardiology, 64 5*, 304-10.

[4] Center for Disease Control and Prevention: Coronary Artery Disease

[5] Coronary Artery Disease: Prevention, Treatment and Research

[6] Our World In Data: Causes of Death

[7] UC Irvine Machine Learning Repository: Heart Disease

[8] Source Code: https://github.com/aluo918/DATA1030-Final