Predicting Presence of Heart Disease

Intro to Problem and Data

Problem

I know a lot of people aging and developing heart diseases, so I was curious about the determining factors. So, for my final project, I want to develop a model that is capable of accurately predicting the presence of heart disease with a given amount of medical data. I'm not entirely sure who would want to use this, as medical professionals have their own rationale and guidelines to use but I wanted to try anyway. It can possibly be integrated into a backend patient system to provide a second opinion to a doctor's initial evaluation, either confirming their prediction, or if opposite, causing another checkup. However, general use by the public might just result in fear mongering and inaccurate results.

Dataset Description

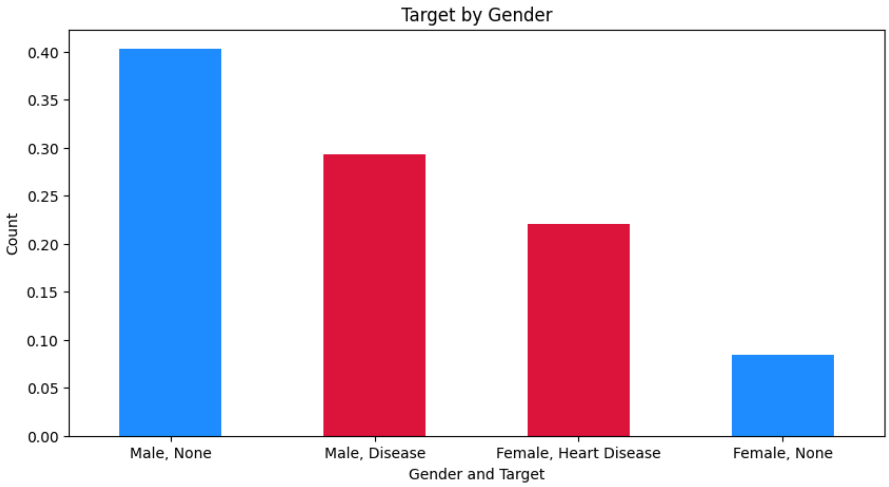
This dataset is sourced from Kaggle as a csv file. It is originally sourced from medical research, but is filtered to only contain a subset of data that is significant, so the actual data doesn't require cleaning. However, some of the columns were mislabeled, so I did some renaming. I will also replace the binary values with text (in my EDA only) to make my analysis easier to read. I don't think there will really be any issues constructing the models, as the data is objective and solely quantitative.

The features of this dataset are 13 encompassing measurements of patients, including age, sex, cholesterol levels, resting blood pressure, fasting blood sugar, max heart rate, etc. There are ~1000 rows and 13 columns for me to create a predictive classification model.

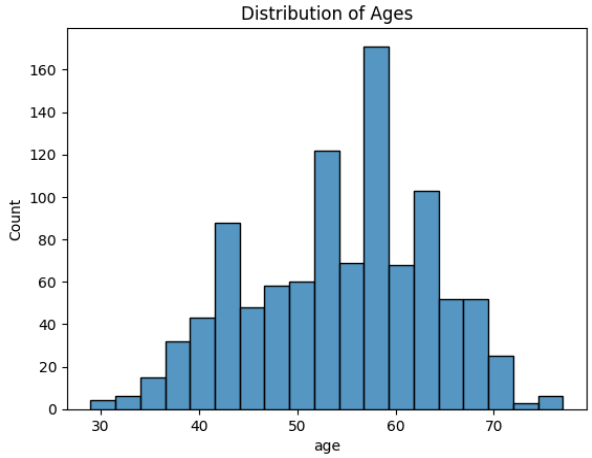
Preliminary Data Examination

Target	Total
1	51.32%
0	48.38%

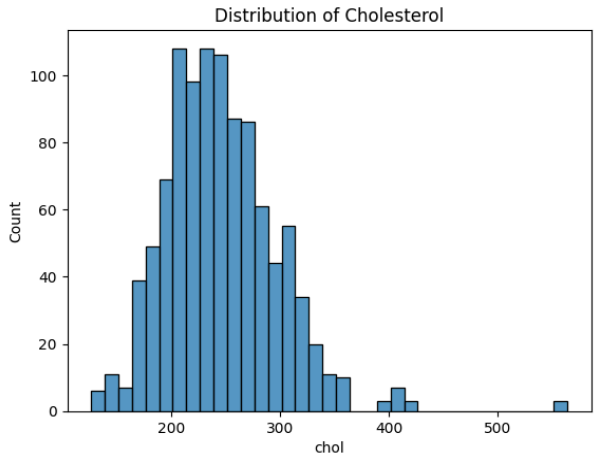
Sex	Total
Male	69.56%
Female	30.44%



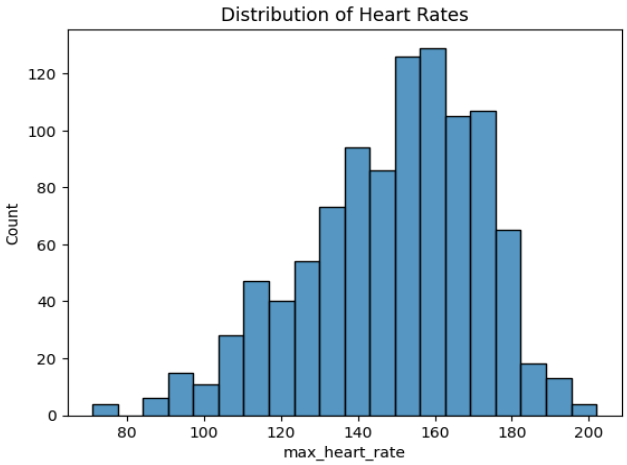
Here, I just wanted to see how the target is distributed. We can see that the data is relatively even across those with and without heart disease. However, there are a lot more men in the data compared to women, which may be a cause for concern. In addition, women tend to have heart disease at a higher rate than men, which may be a flaw in the data, or it could be an actual trend.



The ages range from 29 to 77 years, with a mean of 54.4 years, reflecting a population primarily at risk for cardiovascular diseases. We can see the ages are relatively normally distributed between the minimum and maximum.

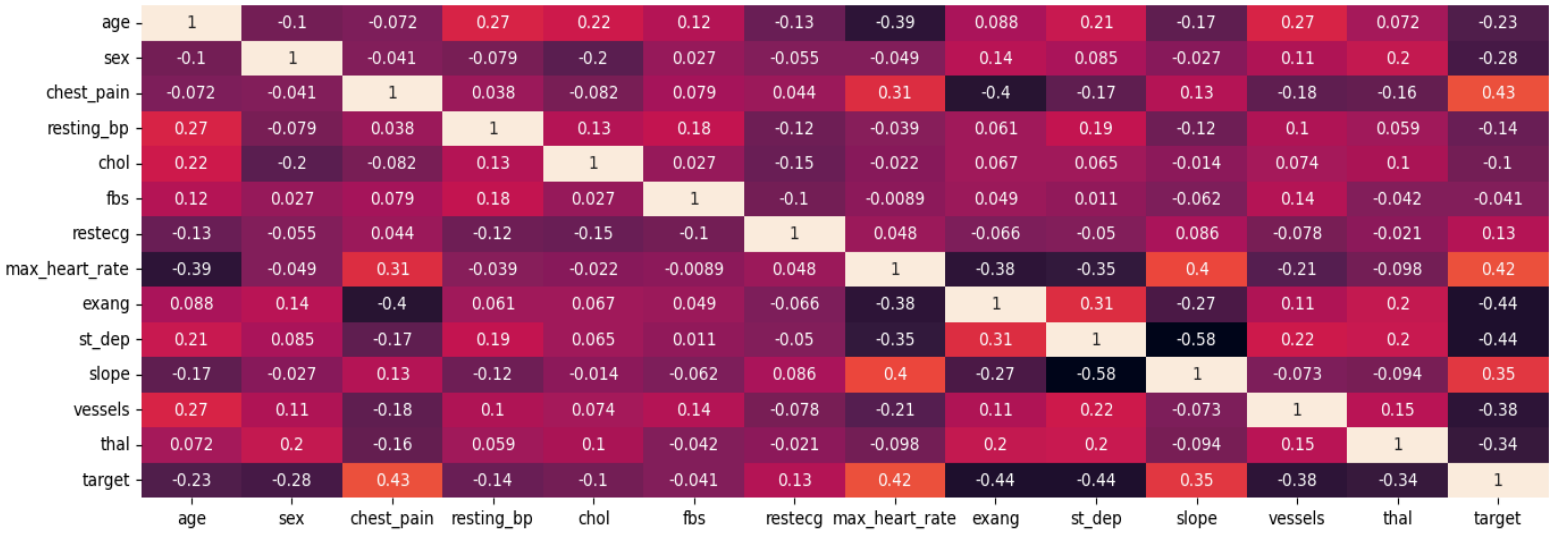


The cholesterol levels in this dataset seems to be normally distributed from the median, with a few outliers in the 400-500+ mg/dl range.

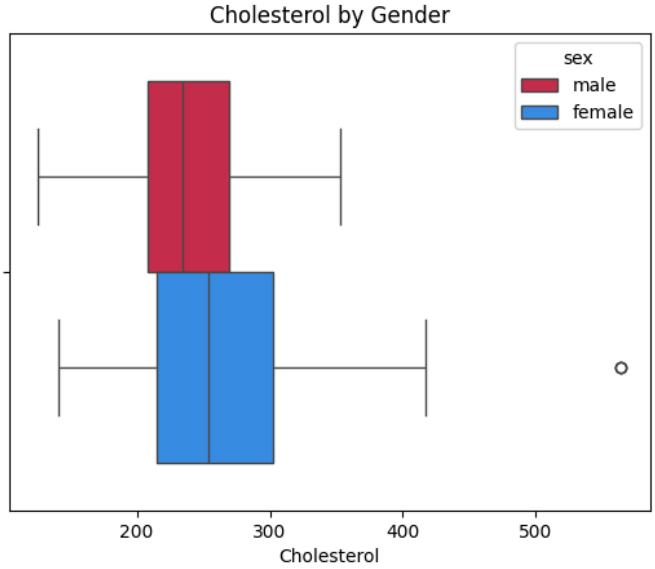


Here we can see the distribution of heart rates, which looks relatively skewed to the left. Not too sure why, but it seems interesting.

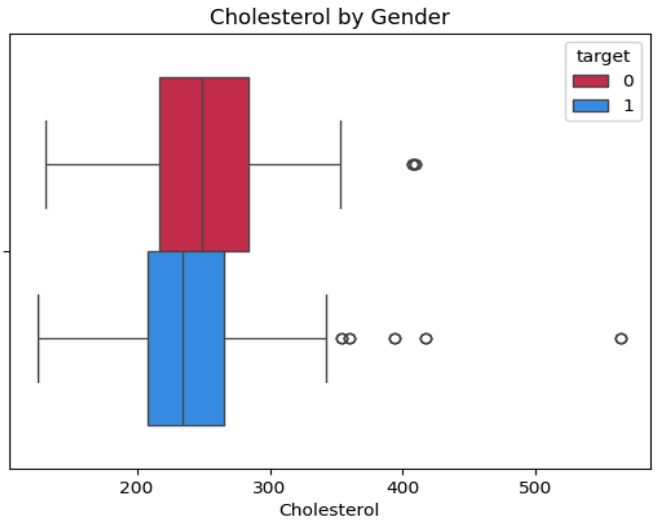
Exploratory Data Analysis



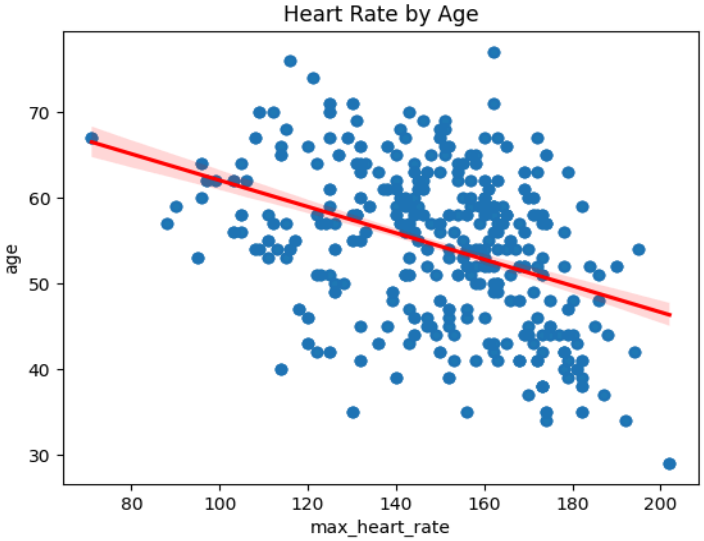
To figure out how the data might be related, I wanted to use a heatmap. The results aren't exactly what I expected, but some of the standout features (correlation with target) are Chest Pain, Exercise Induced Angina, Maximum Heart Rate, and ST Depression. I would also say that Max HR is the most correlated with other features, which may make it an important parameter in my models. Most of the magnitudes of correlations are below or at .40, which should mean the data isn't too collinear. A standout feature would be slope and st_dep, as these two features are related through the ST segment. Now, I want to explore the distributions of my other features.



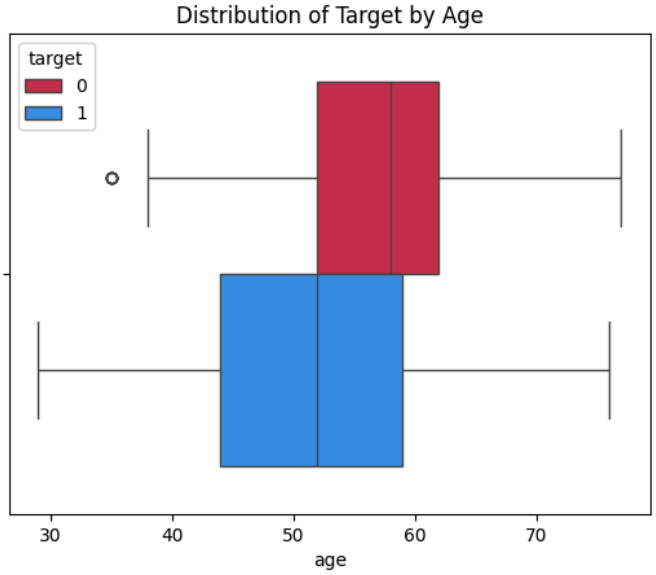
One interesting note I found was the average cholesterol levels of the females were higher than the men's. This could have to do with the fact that within the dataset, more women have heart disease than not, and increased cholesterol increases risk of heart disease. However, the next graph slightly disproves that.



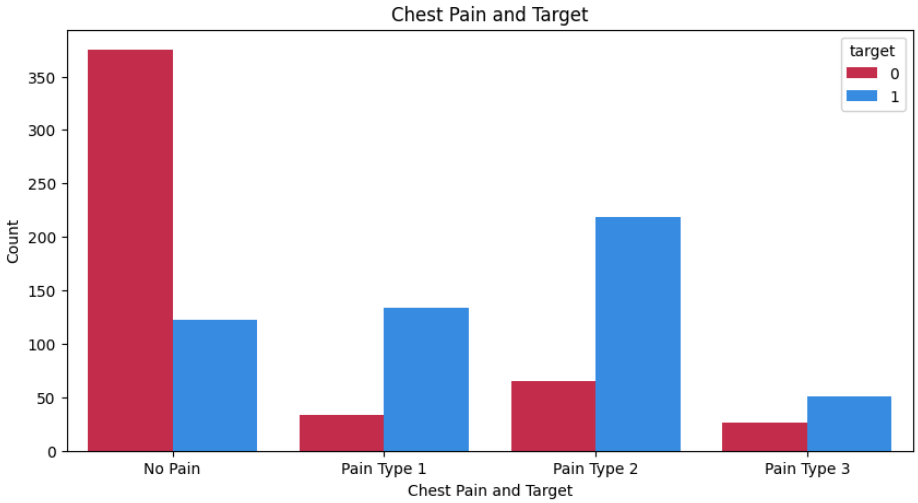
Something I didn't fully expect, but saw in the heatmap, was that increased cholesterol didn't strongly correlate with heart disease. In fact, it was negatively correlated, which surprised me. In fact, the median and quartile ranges were lower for those who had heart disease.



Two features with a medium correlation were max heart rate and age. While this may indicate a weakening body, it doesn't really point to the presence of heart disease. Given the skewness of the heart rates and normalness of the age, I didn't expect for the relationship to be like this.



Another interesting graph I found was the separation of targets by age. I expected those with heart disease to be older, but the median age for those with heart disease was lower.



Here, we can see that for all chest pains other than none, those with heart disease are more likely to have chest pain. This is an indicator it should be an important feature.

Models and Methods Used

I chose to use Logistic Regression, Random Forest, and XGBoost. While I do have the baseline accuracy, 51%, I still wanted to see which models were more effective at predicting heart disease. In order to ensure the best training data, I had to create a column transformer, where I encoded binary values with OHE and used Standard Scaler for all the continuous variables, since they were in different units. This would ensure fair weightings for all the parameters in my models. Then, I split my data into training and testing splits to ensure my models could predict accurately on new data.

Baseline

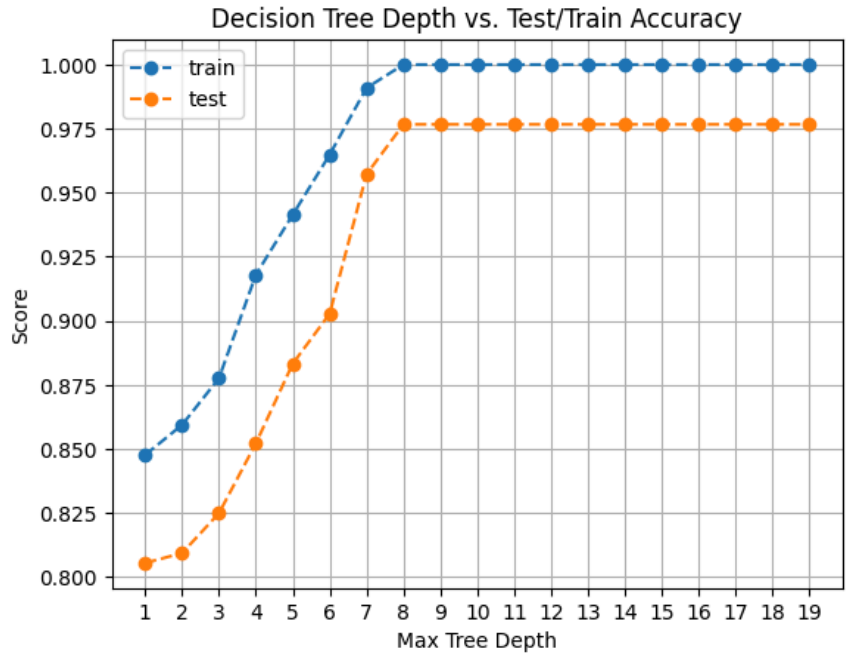
My baseline is the probability of guessing someone has heart disease based on the percentages in the target column. 51.32% of the sample indicated the presence of heart disease, so this will be my baseline accuracy.

Logistic Regression

I chose to use this model because I wanted to have a simple stepping stone to my better models. The Logistic regression model was pretty simple. All I had to do was put the model and transformer into a Pipeline, fit it, and then I was able to predict using my test data. This model returned a score of 81.95%. This model outperformed the baseline by a significant amount, which means it was able to account for a general pattern within the data.

Random Forest

Random Forest, as an ensemble method, captures the non-linear relationships and provides feature importance rankings. I took a few extra steps by using Grid Search with Random Forest, just to tune the model for the best parameters. My GridSearch parameters consisted of n_estimators and max_depth, since there were no features left in the remainder. My model's best parameters were a model max depth of 2 and the # of estimators as 10. Here, since the max was quite low, my test score was 80.48% and my train score was slightly above. Overall for this model, my train and test data still performed better than the baseline, but it underperformed compared to the logistic regression model. Below, I made a graph of Random Forest scores (no GridSearch) while increasing the max depth. As we can see, there is a point of marginal return and possible overfitting, which might be why GridSearch does not use those depths.



XGBoost

XGBoost is the new model that we haven't seen in class. It uses an ensemble of decision trees sequentially to minimize the gradient while complimenting less complex models. It's somewhat similar to the random forest model, but it integrates the neural network aspect of gradient minimization. The model minimizes BCE Loss because it is a classification model. When instantiating it, I used a 'hist' tree method and stopped early rounds at 2. My test score was

98.83%, which is a lot higher than the baseline and previous models, which is interesting. I thought this meant overfitting, but because of the early stopping, I believe it shouldn't be overfit.

Results and Interpretation

Logistic Regression

I had a lot of features, but a few important ones were sex, chest pain, and thal. Sex_male, coefficient -1.44, and chest pain_0, coefficient -1.27, were a little more obvious, since the data distribution gives women a higher probability of having heart disease compared to men and no chest pain indicates a healthy person. For thal, one thing I noticed was both ends had negative values, while the middles were positive. Thal_0: -0.43, Thal_1: 0.71, Thal_2: 0.57, and Thal_3: -0.85. I'm not entirely sure what this means, but I believe thal_0 and thal_3 are predictive of no heart disease. Lastly, another significant feature was the number of vessels, coefficient of -.87. While it had some negative correlation with the target in the heatmap, I wasn't sure what the medical meaning of the feature was.

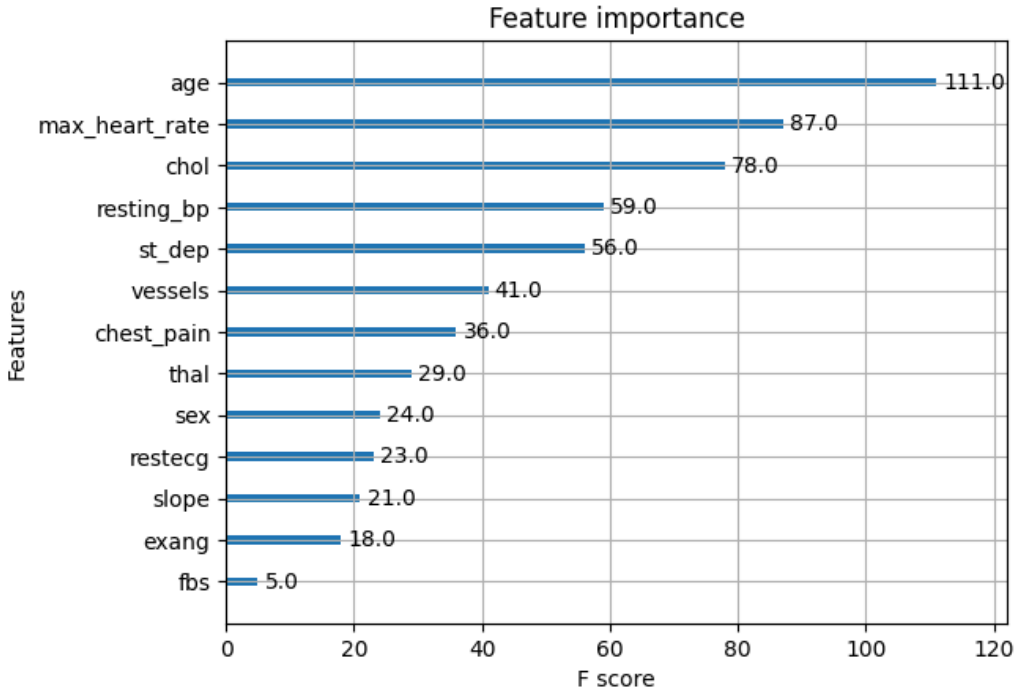
Random Forest

These were my most weighted features in the random forest model. Contrary to the Logistic regression, some of these values are positive and their magnitudes are a lot lower. This is probably because they don't appear to be encoded, so for example, an increase in chest pain leads to an increase in probability of heart disease. Similarly, this is also seen in thal, but something to consider is that the negative coefficient of thal_4 is not really considered. In this model, cholesterol was somewhat important. Max heart rate also showed up as one of the stronger indicators in the random forest model. It was also in the logistic regression model, meaning it should be an important feature to consider. However, vessels did not show up, which is interesting.

thal	0.135
chest_pain	0.035
chol	0.012
max_heart_rate	0.010

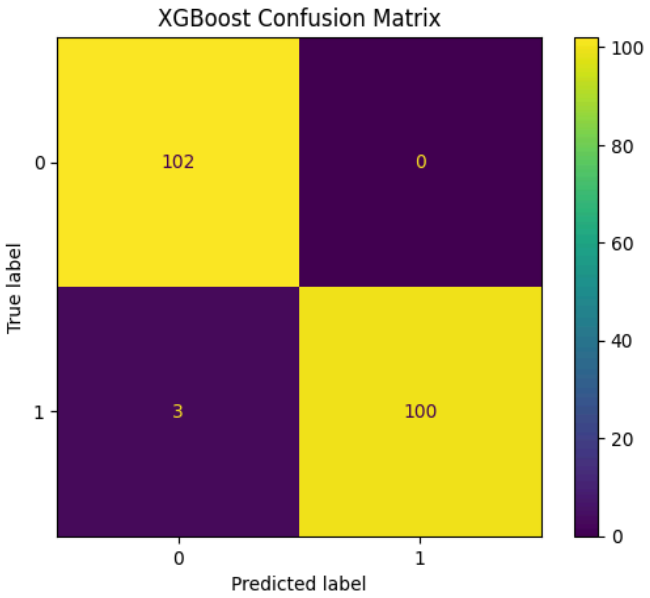
XGBoost

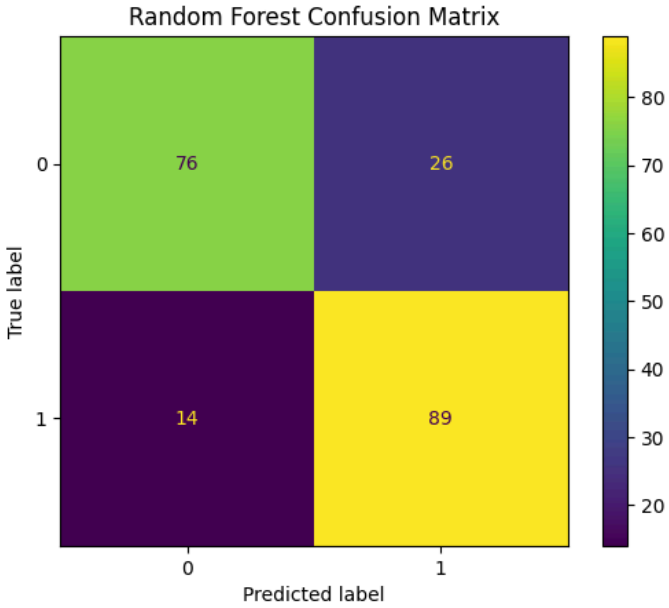
While XGBoost Classifier doesn't necessarily have coefficients, it can use feature importance to describe the model. We can see how age is extremely influential in this model and to a lesser extent, heart rate, cholesterol, blood pressure, and ST depression. Cholesterol seems to have popped up in all three models as a relatively strong or strong indicator. However, what is interesting is the differences between feature importance and coefficient values between XGBoost and the other two models. Logistic regression and Random Forest prioritize thal and chest pain, while XGBoost doesn't. In addition, another confusing part is that age is not important in the other two, but extremely important with XGBoost.

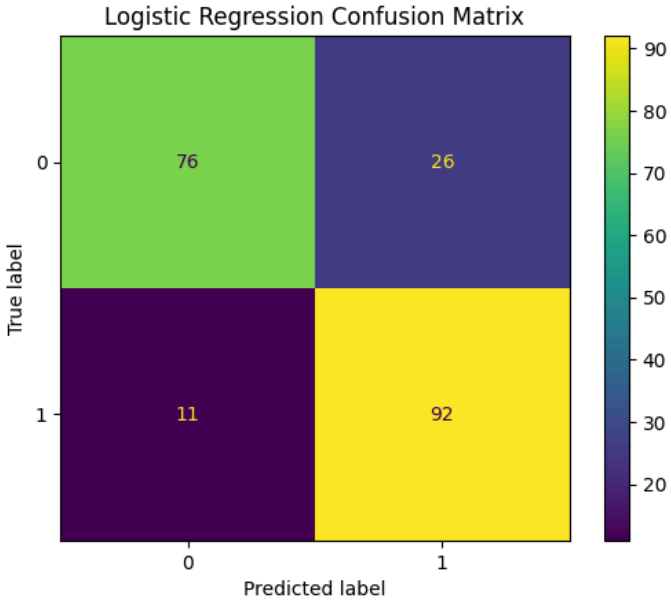


Confusion Matrices

The confusion matrices for healthcare should point to a very specific case. While we'd like to have a score of 100%, we would rather falsely diagnose someone as having heart disease than to falsely diagnose someone as free of heart disease. This means we want to minimize our false negatives. These occur in the bottom left corner, with a True label of 1 and a Predicted label of 0.







Given these confusion matrices, we can see the differences between the predictions of each model. For our logistic regression, we have a FN of 11. For random forest, 18, and for XGBoost, just 3. Clearly, XG has the best accuracy, given its 3 total mistakes. However, I want to compare the recalls of logistic regression and Random Forest. Using true positives and false negatives, I will be able to calculate the recalls.

LGR: 92 / (92 + 11) = 0.893

RF: 89 / (89 + 14) = 0.864

We can see that LGR slightly performs better, in terms of score, and also in terms of recall, which we find important to maximize. So, for our model, LGR is the better choice of the two in both ways.

Conclusion and Next Steps

Key Insights

Overall, Gradient Boosting emerged as the top-performing model with the highest accuracy and recall in both training and testing data. Logistic regression placed second and random forest in a close third. Features that were highlighted by all models were cholesterol, thalassemia, and chest pain. Features that were confused between the two

Future Steps

To enhance the predictive capabilities of the models and gain deeper insights into classifying heart disease, I would like to incorporate these additional features into my models.

1. Enhanced Data Collection
    a. Incorporate additional features such as genetics (family history of heart disease), and lifestyle factors to improve model robustness.
2. External Validation
    a. Test the models on external datasets and possibly new features to evaluate generalizability.
3. Medical Understanding
    a. I'd like to look deeper into the medical reasoning for the feature importances, but that is very science based and something I'm not very good at.

By integrating these next steps into the model, I'd be able to refine the model further and look deeper into predictions of heart disease and also minimize the false negatives. This approach could potentially lead to more accurate predictions and insights for doctors. I'd also like to dig deeper into neural networks, as I tried to implement an ANN, but couldn't do so successfully. My losses would not change and I was stuck on how to improve the model.