# Heart Disease Classification

Group 5: Almin Raja, Dinh Mai, Julio Amaya, Subhan Mehmood, Nicholas Pham

## Introduction

According to the CDC, heart disease is the leading cause of death for men and women in the United States. It is estimated that a person dies every 33 seconds from cardiovascular disease, with a proportion of 1 in every 5 deaths being from heart disease in 2022.

As such, we chose this particular dataset focusing on heart disease in patients from 4 databases: Cleveland, Hungary, Switzerland, and the VA Long Beach. Our goal is to build and train classification models to be able to classify whether a person has heart disease or not so that they can adequately get the medical attention they need to reduce the statistics of deaths by heart disease. Additionally, another reason we chose this dataset is that it has a broad range of features consisting of categorical and quantitative aspects pertaining to the health measurements of a patient that would allow us to properly train our models, as well as having enough instances that would allow for enough observations to train while at the same time, be manageable. Also, the information about this dataset's databases, features, etc., is extensive, allowing us to get the complete picture.

## Description of Data

The dataset is called Heart Disease and comes from the UC Irvine Machine Learning Repository. It contains data from 4 databases: Cleveland, Hungary, Switzerland, and the VA Long Beach. There are a total of 303 instances and 13 features in addition to a response variable signifying whether a patient has heart disease or not, so a total of 14 variables. The dataset contains categorical, integer, and real feature types. The subject area of this dataset is health and medicine.

## Description of Variables

Total of 14 variables:

*age*: age in years - numerical
*sex*: sex (1 = male, 0 = female) - categorical
*cp*: type of chest pain (0 = typical angina, 1 = atypical angina, 2 = non-anginal pain, 3 = asymptomatic) - categorical
*trestbps*: resting blood pressure (in mm Hg on admission to the hospital) - numerical
*chol*: serum cholesterol in mg/dl - numerical
*fbs*: fasting blood sugar $> 120$ mg/dl (1 = true, 0 = false) - categorical
*restecg*: resting electrocardiographic results (0 = normal, 1 having ST-T wave abnormality [T wave inversions and/or ST elevation or depression of $> 0.05$ mV], 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria) - categorical

*thalach*: maximum heart rate achieved - numerical
*exang*: exercise induced angina (1 = yes, 0 = no) - categorical
*oldpeak*: ST depression induced by exercise relative to rest - numerical
*slope*: the slope of the peak exercise ST segment (0 = upsloping, 1 = flat, 2 = downsloping) - categorical
*ca*: number of major vessels (0-3) colored by fluoroscopy - categorical
*thal*: 0 = normal, 1 = fixed defect, 2 = reversable defect - categorical
*num*: diagnosis of heart disease (1 = disease, 0 = no disease) - categorical

## Main Questions

We will use *num* as our response, with 0 indicating a patient has no heart disease and a 1 indicating a patient does have heart disease. We will potentially use *age*, *sex*, *cp*, *trestbps*, *chol*, *fbs*, *restecg*, *thalach*, *exang*, *oldpeak*, *slope*, and *ca* as predictors. Additionally, we will prioritize prediction as our final goal because our focus is to classify and predict whether a patient has heart disease or not.

Questions:
Which features are the most important predictors of heart disease?
What is the best model to predict heart disease based on this dataset? (Primary Question)
Does age play a significant role in predicting heart disease?
Is heart disease more prominent in males or females?

## Models

We will be using Logistic Regression as our first model because our response is categorical, and it will allow us to classify whether or not a patient has heart disease since Logistic Regression produces the probability of an event occurring. Also, the ease of interpretability of Logistic Regression because it provides how each feature impacts the response variable and its simplicity in implementing this model further solidified our choice. However, Logistic Regression has difficulty with non-linear data and is not able to capture these trends, and this model is sensitive to outliers, which can cause disarray in the performance and statistics of the model.

We will be using Random Forests as our second model because it will enable us to accurately predict our classifier and provide further insights into complex, non-linear interactions between features that Logistic Regression would not be able to capture, which will come in handy for our analysis, as well as the fact that this model will be able to determine the rank of importance of the predictors for this dataset. However, Random Forest can

be computationally expensive, prone to overfitting if not tuned properly, and can be less interpretable than Logistic Regression.

Additionally, we will use cross-validation for model comparison and evaluation. It will provide a fair comparison between the Logistic Regression and the Random Forest Models, aiding our analysis.

## Logistic Regression Model (Subhan, Dinh, Nicholas)

Since our scenario is a classification problem, predicting whether a patient has heart disease based on attributes, our response variable, *num*, is categorical. As such, our first method will be Logistic Regression. Our goal would be to use significant features found through our analysis within our model to answer specific questions with Logistic Regression; the questions we will answer are specified under the Main Questions section. Additionally, a benefit of utilizing Logistic Regression in this situation is that it is simple to implement and easy to understand the results that are outputted. However, there are still downsides to using this model, such as overfitting our model, fitting the noise of the dataset rather than the true data, and if many variables are factored into training the model, then the results can be convoluted, and the Logistic Regression could be out-performed by another model, such as Random Forest.

$$p(x) = \frac{\exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_{13} X_{13})}{1 + \exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_{13} X_{13})}$$

p(X), also known as Y, will be the response variable, num, B0 will be the intercept, B1 through B13 will be the predictors or features, which are *age*, *sex*, *cp*, *trestbps*, *chol*, *fbs*, *restecg*, *thalach*, *exang*, *oldpeak*, *slope*, *ca*, and *thal.* Initially, we include all the predictors from the dataset into the model, and we will statistically analyze and reduce features till we are only left with the significant ones to this dataset and the ones important in predicting heart disease.

To train the Logistic Regression Model, we wanted first to find the significant variables for this dataset. Initially, we started by training Logistic Models with each variable included individually in relation to the predicted variable, *num*. For example, we trained a Logistic Model with *num* and *age*, *num* and *sex*, etc. We found that *age*, *sex*, *cp*, *trestbps*, *restecg*, *thalach*, *exang*, *oldpeak*, *slope*, and *ca* were significant when fitted individually based on the p-value. Next, we trained a Logistic Model with all the variables included, as seen below. From this fit, we found that *sex*, *cp*, *trestbps*, *thalach*, *oldpeak*, and *slope* were significant via the p-value. At this point, we noted that the variables, *sex*, *cp*, *trestbps*, *thalach*, *oldpeak*, and *slope*, were both significant when trained individually with the model and when trained all together; hence, we considered these variables significant.

```r
# Sex, cp, trestbps, thalach, oldpeak, and slope are significant
heart.fit = glm(num ~ ., family = "binomial", data = HeartDisease)
summary(heart.fit)
```

```
Call:
glm(formula = num ~ ., family = "binomial", data = HeartDisease)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.800e+00  4.413e+00  -1.314 0.188789
age          3.079e-03  3.199e-02   0.096 0.923344
sex1         2.255e+00  7.131e-01   3.162 0.001569 **
cp2          1.435e+00  9.672e-01   1.484 0.137777
cp3          6.985e-01  9.686e-01   0.721 0.470779
cp4          3.185e+00  9.386e-01   3.393 0.000691 ***
trestbps     3.502e-02  1.491e-02   2.348 0.018851 *
chol         6.133e-03  4.923e-03   1.246 0.212842
fbs1        -7.916e-01  7.309e-01  -1.083 0.278783
restecg1     1.710e+01  5.019e+03   0.003 0.997282
restecg2     8.272e-01  4.892e-01   1.691 0.090869 .
thalach     -2.895e-02  1.468e-02  -1.971 0.048667 *
exang1      -1.807e-02  5.928e-01  -0.030 0.975691
oldpeak0,1  -1.813e+00  1.255e+00  -1.444 0.148688
oldpeak0,2  -2.495e+00  1.374e+00  -1.816 0.069337 .
oldpeak0,3   1.187e-01  1.425e+00   0.083 0.933644
oldpeak0,4  -3.802e+00  1.470e+00  -2.586 0.009723 **
oldpeak0,5  -3.550e+00  1.607e+00  -2.209 0.027185 *
oldpeak0,6  -1.811e+00  1.191e+00  -1.521 0.128344
oldpeak0,7  -1.273e+01  1.075e+04  -0.001 0.999056
oldpeak0,8   3.945e-01  9.967e-01   0.396 0.692249
oldpeak0,9   8.711e-01  1.064e+01   0.082 0.934754
oldpeak1,0  -4.734e-01  1.180e+00  -0.401 0.688193
oldpeak1,1  -3.319e+01  7.958e+03  -0.004 0.996673
oldpeak1,2  -8.075e-01  1.080e+00  -0.748 0.454524
oldpeak1,3  -1.822e+01  1.075e+04  -0.002 0.998648
oldpeak1,4   5.745e-01  1.150e+00   0.499 0.617528
oldpeak1,5  -4.318e+00  3.113e+00  -1.387 0.165411
oldpeak1,6  -9.263e-01  1.245e+00  -0.744 0.456920
oldpeak1,8  -2.579e-01  1.370e+00  -0.188 0.850672
```

```
oldpeak1,9  -1.299e+00  1.685e+00  -0.771 0.440794
oldpeak2,0  -2.527e+00  1.588e+00  -1.591 0.111624
oldpeak2,1   1.381e+01  1.075e+04   0.001 0.998975
oldpeak2,2   1.682e+01  4.103e+03   0.004 0.996729
oldpeak2,3  -2.040e+01  6.793e+03  -0.003 0.997604
oldpeak2,4  -2.084e+00  5.211e+00  -0.400 0.689220
oldpeak2,5   1.634e+01  7.370e+03   0.002 0.998231
oldpeak2,6   6.581e-01  2.727e+00   0.241 0.809298
oldpeak2,8   1.663e+01  3.365e+03   0.005 0.996057
oldpeak2,9   1.425e+01  1.075e+04   0.001 0.998943
oldpeak3,0   4.487e-02  1.589e+00   0.028 0.977469
oldpeak3,1   1.672e+01  1.075e+04   0.002 0.998759
oldpeak3,2   1.640e+01  6.519e+03   0.003 0.997993
oldpeak3,4   1.277e+01  5.436e+03   0.002 0.998125
oldpeak3,5  -1.686e+01  1.075e+04  -0.002 0.998749
oldpeak3,6   1.824e+01  4.149e+03   0.004 0.996493
oldpeak3,8   2.091e+01  1.075e+04   0.002 0.998448
oldpeak4,0   1.351e+01  6.026e+03   0.002 0.998211
oldpeak4,2  -4.811e+00  2.840e+00  -1.694 0.090307 .
oldpeak4,4  -1.441e+00  1.187e+04   0.000 0.999903
oldpeak5,6   1.539e+01  1.075e+04   0.001 0.998858
oldpeak6,2   1.501e+01  1.075e+04   0.001 0.998886
slope2       2.188e+00  6.906e-01   3.169 0.001531 **
slope3       1.799e+00  1.353e+00   1.329 0.183804
ca0,0       -1.459e-01  2.225e+00  -0.066 0.947704
ca1,0        2.063e+00  2.237e+00   0.922 0.356299
ca2,0        3.324e+00  2.472e+00   1.345 0.178768
ca3,0        2.233e+00  2.402e+00   0.929 0.352719
thal3,0     -2.297e+00  2.165e+00  -1.061 0.288683
thal6,0     -1.830e+00  2.449e+00  -0.747 0.454911
thal7,0     -2.932e-01  2.177e+00  -0.135 0.892880
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 417.98  on 302  degrees of freedom
Residual deviance: 148.31  on 242  degrees of freedom
AIC: 270.31

Number of Fisher Scoring iterations: 18
```

Afterward, we utilized the step() function to acquire a model with appropriate features based on the lowest AIC, as seen below. From this, we noted that the model giving the lowest AIC included *sex*, *cp*, *trestbps*, *thalach*, *exang*, *slope*, *ca*, and *thal*. This enforced some of our recognized significant predictors from previous analyses. Therefore, we considered *sex*, *cp*, *trestbps*, *thalach*, and *slope* as confirmed significant predictors. Upon further investigation by fitting test models with combinations of exang, ca, and thal with the previously confirmed significant predictors, we found that including *exang*, *ca*, and *thal* with the already recognized important features gave us better model statistics than not including them specifically, we achieved lower residual deviance and lower AIC while the null deviance stayed the same throughout the test models. Additionally, from our earlier studies, we also initially included *oldpeak* and *slope* within our list of significant predictors; however, after running test models by including a combination of *oldpeak* and *slope* with the already existing variables, we found that these features significantly increased the AIC and heightened the residual deviance; hence, we knew these features were not appropriate for our model and were not included.

```
# Implement step function to get predictors that result in lowest AIC
step(heart.fit)
```

```
Start:  AIC=270.31
num ~ age + sex + cp + trestbps + chol + fbs + restecg + thalach +
    exang + oldpeak + slope + ca + thal

           Df Deviance    AIC
- oldpeak  39   187.42 231.42
- exang     1   148.31 268.31
- age       1   148.32 268.32
- fbs       1   149.54 269.54
- restecg   2   151.66 269.65
- chol      1   149.87 269.87
<none>          148.31 270.31
- thalach   1   152.63 272.63
- trestbps  1   154.42 274.42
- slope     2   159.92 277.92
- thal      3   163.37 279.37
- sex       1   159.90 279.90
- ca        4   172.73 286.73
- cp        3   173.21 289.21

Step:  AIC=231.42
```

```
num ~ age + sex + cp + trestbps + chol + fbs + restecg + thalach +
    exang + slope + ca + thal

           Df Deviance    AIC
- restecg   2   189.18 229.18
- fbs       1   188.35 230.35
- age       1   188.83 230.83
- chol      1   189.02 231.02
<none>          187.42 231.42
- exang     1   190.42 232.42
- thalach   1   191.00 233.00
- trestbps  1   194.17 236.17
- thal      3   201.06 239.06
- slope     2   200.43 240.43
- sex       1   199.72 241.72
- cp        3   208.62 246.62
- ca        4   230.73 266.73

Step:  AIC=229.18
num ~ age + sex + cp + trestbps + chol + fbs + thalach + exang +
    slope + ca + thal

           Df Deviance    AIC
- fbs       1   190.18 228.18
- age       1   190.47 228.47
<none>          189.18 229.18
- chol      1   191.45 229.45
- exang     1   192.16 230.16
- thalach   1   192.78 230.78
- trestbps  1   197.03 235.03
- thal      3   202.07 236.07
- slope     2   203.10 239.10
- sex       1   202.08 240.08
- cp        3   210.46 244.46
- ca        4   233.39 265.39

Step:  AIC=228.18
num ~ age + sex + cp + trestbps + chol + thalach + exang + slope +
    ca + thal

           Df Deviance    AIC
```

```
- age       1   191.41 227.41
<none>          190.18 228.18
- chol      1   192.29 228.29
- exang     1   192.83 228.83
- thalach   1   193.97 229.97
- trestbps  1   197.39 233.39
- thal      3   203.37 235.37
- slope     2   203.68 237.68
- sex       1   202.62 238.62
- cp        3   213.56 245.56
- ca        4   233.39 263.39

Step:  AIC=227.41
num ~ sex + cp + trestbps + chol + thalach + exang + slope +
    ca + thal

           Df Deviance    AIC
- chol      1   193.15 227.15
<none>          191.41 227.41
- exang     1   194.19 228.19
- thalach   1   194.20 228.20
- trestbps  1   197.52 231.52
- thal      3   204.59 234.59
- slope     2   204.68 236.68
- sex       1   204.48 238.48
- cp        3   215.47 245.47
- ca        4   234.70 262.70

Step:  AIC=227.15
num ~ sex + cp + trestbps + thalach + exang + slope + ca + thal

           Df Deviance    AIC
<none>          193.15 227.15
- thalach   1   195.60 227.60
- exang     1   196.07 228.07
- trestbps  1   199.99 231.99
- thal      3   206.89 234.89
- sex       1   204.69 236.69
- slope     2   207.11 237.11
- cp        3   217.57 245.57
- ca        4   237.46 263.46
```

```
Call:  glm(formula = num ~ sex + cp + trestbps + thalach + exang + slope +
    ca + thal, family = "binomial", data = HeartDisease)

Coefficients:
(Intercept)         sex1          cp2          cp3          cp4     trestbps
   -6.06623      1.67748      1.09696      0.22624      2.30184      0.02668
    thalach       exang1       slope2       slope3        ca0,0        ca1,0
   -0.01615      0.74352      1.61783      1.33283      1.55594      3.65051
      ca2,0        ca3,0      thal3,0      thal6,0      thal7,0
    4.74341      3.72928     -1.83412     -2.14409     -0.44397

Degrees of Freedom: 302 Total (i.e. Null);  286 Residual
Null Deviance:      418
Residual Deviance: 193.2    AIC: 227.2
```

Therefore, for our significant features and the answer to one of our questions, which features are the most important predictors of heart disease, we found *sex*, *cp*, *trestbps*, *thalach*, *exang*, *slope*, *ca*, and *thal* were important and thus included and trained with a Logistic Regression Model to give us the optimal model. Additionally, given the best model, the null hypothesis would be that all the coefficients in the model equate to zero, and the alternative hypothesis would be that at least one coefficient is non-zero. From our analysis, we would reject the null hypothesis since there are significant predictors based on their p-value. Below is the training of the best model with the significant features. However, there are some features that are not significant based on their p-values. The metrics that we achieve from training the Logistic Regression Model result in the lowest AIC and residual deviance.

```r
# Fit model based on predictors chosen by step function
# Considered our optimal model
heart.fit.optimal = glm(num ~ sex + cp + trestbps + thalach + exang
                        + slope + ca + thal, family = "binomial",
                        data = HeartDisease)
summary(heart.fit.optimal)
```

```
Call:
glm(formula = num ~ sex + cp + trestbps + thalach + exang + slope +
    ca + thal, family = "binomial", data = HeartDisease)

Coefficients:
```

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.06623    4.09035  -1.483 0.138058
sex1         1.67748    0.51802   3.238 0.001203 **
cp2          1.09696    0.76808   1.428 0.153238
cp3          0.22624    0.68274   0.331 0.740362
cp4          2.30184    0.68119   3.379 0.000727 ***
trestbps     0.02668    0.01054   2.530 0.011406 *
thalach     -0.01615    0.01058  -1.526 0.126980
exang1       0.74352    0.43387   1.714 0.086586 .
slope2       1.61783    0.45527   3.554 0.000380 ***
slope3       1.33283    0.76074   1.752 0.079772 .
ca0,0        1.55594    1.66564   0.934 0.350231
ca1,0        3.65051    1.72538   2.116 0.034364 *
ca2,0        4.74341    1.82728   2.596 0.009435 **
ca3,0        3.72928    1.85744   2.008 0.044670 *
thal3,0     -1.83412    2.97346  -0.617 0.537347
thal6,0     -2.14409    3.04988  -0.703 0.482051
thal7,0     -0.44397    2.97648  -0.149 0.881427
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 417.98  on 302  degrees of freedom
Residual deviance: 193.15  on 286  degrees of freedom
AIC: 227.15

Number of Fisher Scoring iterations: 6
```

Using the significant predictors, we would implement cross-validation via an 80/20 split, where 80% of the dataset would be used to train the model and the remaining 20% to train it. We outputted a confusion matrix to visualize the model's performance in predicting whether a patient has a heart disease, denoted as 0, or does not, denoted as 1. This 80/20 split was run with 10 iterations with a different seed each time, resulting in 10 different matrices. The implementation of this cross-validation is seen below.

```r
# Logistic Regression Cross Validation - 80/20 Split
store.errorRate = rep(0, 10)

for(i in 1:10){
```

```
  set.seed(i + 100)
  sample = sample.int(n = nrow(HeartDisease), size =
                        floor(.8 * nrow(HeartDisease)), replace = F)

  train = HeartDisease[sample,]
  test = HeartDisease[-sample,]

  heart.glm.cv = glm(num ~ sex + cp + trestbps + thalach + exang
                     + slope + ca + thal, family = "binomial",
                     data = train)

  heart.glm.pred = predict(heart.glm.cv, newdata = test, type = "response")
  heart.binary = ifelse(heart.glm.pred < 0.5, "0", "1")

  conf.mat = table(Predicted = heart.binary, Actual = test$num)
  store.errorRate[i] = (conf.mat[1, 2] + conf.mat[2, 1])/sum(conf.mat)
}

conf.mat # 0 does not have heart disease, 1 has heart disease
```

```
         Actual
Predicted  0  1
        0 29  6
        1  5 21
```

```
store.errorRate
```

```
 [1] 0.2131148 0.2786885 0.1639344 0.1311475 0.1803279 0.1639344 0.1475410
 [8] 0.1475410 0.1475410 0.1803279
```

```
mean(store.errorRate)
```

```
[1] 0.1754098
```

```
# Accuracy
1 - mean(store.errorRate)
```

```
[1] 0.8245902
```

The results show that the average error rate of the Logistic Regression model is 0.1754 or 17.54%, meaning that the model has an error of approximately 17.54%. Each of these error rates was calculated by taking the false positives in addition to the false negatives and dividing it by the total observations. This means that, on average, our model is 0.8246 or 82.46% accurate in predicting *num*, i.e., whether or not a patient has a heart disease, given that the significant predictors are known.

Moreover, one of the questions we specified concerning this dataset is whether *age* plays a significant role in predicting heart disease. Hence, with our best model determined, we took this opportunity to establish this. It is to be noted that *age* is not included within our set of important features, so it can be said that *age* does not play a significant role in predicting heart disease. However, we did further analysis to solidify this claim.

As previously discussed, whenever *age* is modeled individually concerning *num*, the variable is significant based on the p-value. However, when we plot a Linear Model with all the features included, as we have already seen, we notice that *age* is no longer a significant predictor. Additionally, the step() function that we formerly implemented granted us the predictors that resulted in the lowest AIC and led us to the significant variables; age was not included among these variables. Moreover, when including *age* along with the features of our optimal model, as seen below, we can note that it increases our AIC and is not significant based on its p-value, indicating that including *age* gives our model poorer performance than previously seen. Hence, we can attest that *age* does not play a significant role in predicting heart disease.

```
# Role of age
# Including age alongside the significant predictors
heart.age = glm(num ~ age + sex + cp + trestbps + thalach + exang
                        + slope + ca + thal, family = "binomial",
                        data = HeartDisease)
summary(heart.age)
```

```
Call:
glm(formula = num ~ age + sex + cp + trestbps + thalach + exang +
    slope + ca + thal, family = "binomial", data = HeartDisease)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.89162    4.23173  -1.156 0.247706
age         -0.02247    0.02430  -0.925 0.355091
sex1         1.64530    0.52259   3.148 0.001642 **
```

```
cp2          1.12853    0.77383   1.458 0.144741
cp3          0.24385    0.68174   0.358 0.720582
cp4          2.30014    0.68428   3.361 0.000776 ***
trestbps     0.02973    0.01114   2.668 0.007638 **
thalach     -0.01957    0.01124  -1.741 0.081754 .
exang1       0.73015    0.43494   1.679 0.093205 .
slope2       1.64242    0.45821   3.584 0.000338 ***
slope3       1.33158    0.75948   1.753 0.079555 .
ca0,0        1.63682    1.72981   0.946 0.344023
ca1,0        3.82702    1.79681   2.130 0.033180 *
ca2,0        5.01213    1.91007   2.624 0.008689 **
ca3,0        3.95500    1.92744   2.052 0.040175 *
thal3,0     -1.78106    2.87531  -0.619 0.535633
thal6,0     -2.11480    2.95547  -0.716 0.474266
thal7,0     -0.38975    2.87907  -0.135 0.892316
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 417.98  on 302  degrees of freedom
Residual deviance: 192.29  on 285  degrees of freedom
AIC: 228.29

Number of Fisher Scoring iterations: 6
```

Furthermore, another question we posed was: is heart disease more prominent in males or females? We trained a Multiple Logistic Regression Model via the optimal model with the significant predictors to analyze this. Then, we split the model based on gender, male and female, and extracted the mean from the predictors for each gender to get a reasonable metric for each sex.

```r
# Convert num to numeric
HeartDisease$num <- as.numeric(as.character(HeartDisease$num))
# Prominence of heart disease in males or females
# Female is 0, male is 1
sex.split = split(HeartDisease, HeartDisease$sex)

summary(sex.split[[1]]) # Split for females
```

```
      age        sex     cp        trestbps          chol       fbs     restecg
```

```
 Min.   :34.00   0:97   1: 4   Min.   : 94.0   Min.   :141.0   0:85   0:49
 1st Qu.:50.00   1: 0   2:18   1st Qu.:120.0   1st Qu.:215.0   1:12   1: 3
 Median :57.00          3:35   Median :132.0   Median :254.0          2:45
 Mean   :55.72          4:40   Mean   :133.3   Mean   :261.8
 3rd Qu.:63.00                 3rd Qu.:140.0   3rd Qu.:302.0
 Max.   :76.00                 Max.   :200.0   Max.   :564.0
    thalach      exang    oldpeak          slope    ca      thal
 Min.   : 96.0   0:75   Length:97          1:47      : 0      : 1
 1st Qu.:142.0   1:22   Class :character   2:45   0,0:65   3,0:80
 Median :157.0          Mode  :character   3: 5   1,0:15   6,0: 1
 Mean   :151.2                                    2,0:13   7,0:15
 3rd Qu.:165.0                                    3,0: 4
 Max.   :192.0
      num
 Min.   :0.0000
 1st Qu.:0.0000
 Median :0.0000
 Mean   :0.2577
 3rd Qu.:1.0000
 Max.   :1.0000
```

```
mean(sex.split[[1]]$num) # Mean split for females
```

```
[1] 0.257732
```

```
summary(sex.split[[2]]) # Split for males
```

```
      age         sex     cp         trestbps          chol       fbs
 Min.   :29.00   0:  0   1: 19   Min.   : 94.0   Min.   :126.0   0:173
 1st Qu.:47.00   1:206   2: 32   1st Qu.:120.0   1st Qu.:208.8   1: 33
 Median :54.50           3: 51   Median :130.0   Median :235.0
 Mean   :53.83           4:104   Mean   :130.9   Mean   :239.6
 3rd Qu.:59.75                   3rd Qu.:140.0   3rd Qu.:268.5
 Max.   :77.00                   Max.   :192.0   Max.   :353.0
 restecg    thalach      exang     oldpeak          slope    ca       thal
 0:102   Min.   : 71.0   0:129   Length:206         1:95      :  4      :  1
 1:  1   1st Qu.:132.0   1: 77   Class :character   2:95   0,0:111   3,0: 86
 2:103   Median :150.5           Mode  :character   3:16   1,0: 50   6,0: 17
         Mean   :148.8                                    2,0: 25   7,0:102
```

```
        3rd Qu.:167.5                                          3,0: 16
        Max.   :202.0
     num
 Min.   :0.0000
 1st Qu.:0.0000
 Median :1.0000
 Mean   :0.5534
 3rd Qu.:1.0000
 Max.   :1.0000
```

```r
mean(sex.split[[2]]$num) # Mean split for males
```

```
[1] 0.5533981
```

From the results, we can see that females, measured as 0, have a 0.2577 or 25.77% probability of having a heart disease, and the males, measured as 1, have a 0.5534 or 55.34% probability of having a heart disease. It is clear that heart disease is more prominent in males than females, given this higher statistic for males derived from the dataset, which also explains why *sex* is a significant feature within the model, as the gender of a patient can play an important role in determining whether a patient has heart disease.

### Random Forest Model (Julio and Almin)

For our second method, we employed the Random Forest model. This approach offers reduced variability compared to other tree-based techniques like bagging and decision trees, while effectively capturing non-linear and complex relationships within the dataset. However, a negative of Random Forest is that its interpretability can be difficult, as it involves the construction of numerous decision trees.

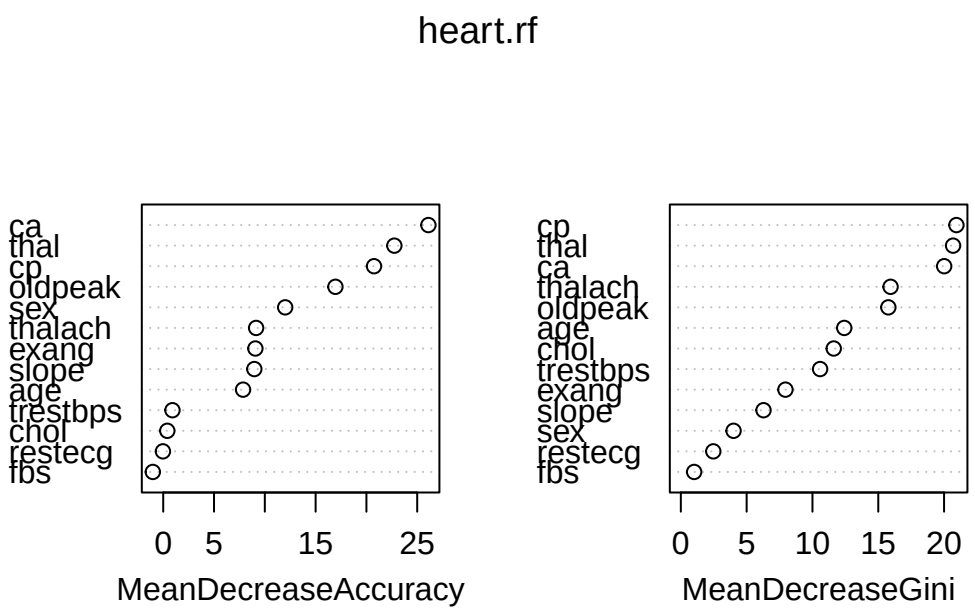
Initially, the Random Forest Model will follow the form num ~ age + sex + cp + trestbps + chol + fbs + restecg + thalach + exang + oldpeak + slope + ca + thal, which is the complete model with all the predictors included. Below is the training of this model and its respective variable importance plot.

```r
# Convert num to factor
HeartDisease$num <- as.factor(HeartDisease$num)
set.seed(123)

heart.rf = randomForest(num ~ ., data = HeartDisease,
```

```
                         ntree = 500, mtry = sqrt(13), importance = T)

heart.rf
```

```
Call:
 randomForest(formula = num ~ ., data = HeartDisease, ntree = 500,      mtry = sqrt(13), im
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 17.82%
Confusion matrix:
    0   1 class.error
0 140  24   0.1463415
1  30 109   0.2158273
```

```
varImpPlot(heart.rf)
```



The Random Forest Model, which included all the features, was trained with 500 trees. From it, we achieved an error rate of 17.16%, which is only slightly better than our best-performing Logistic Regression Model.

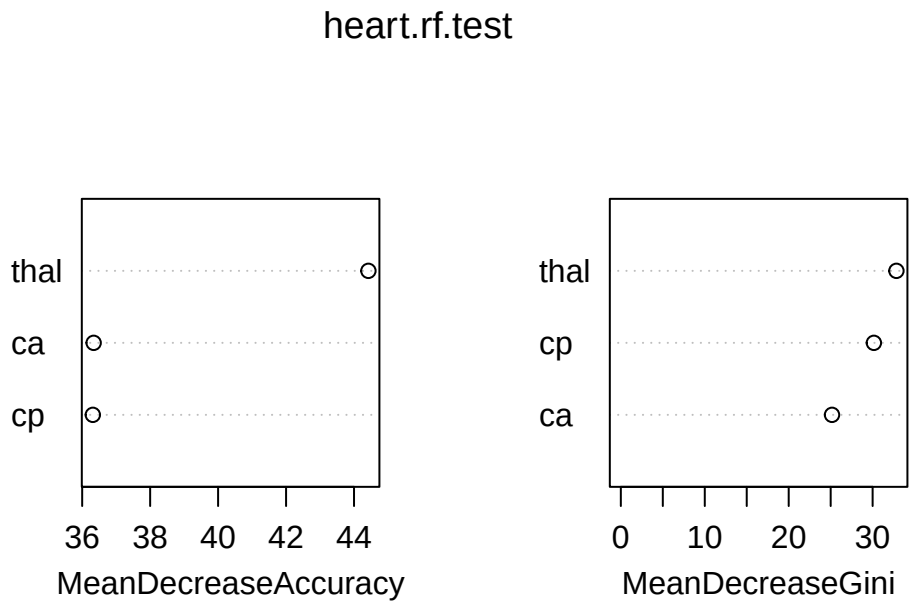
Additionally, from the variable importance plot, we can note that *ca*, *thal*, and *cp* are significant variables in this instance, which is measured by the total amount that the Gini Index decreases after each split over that predictor. The larger this value is, the more important this predictor is within our model(s). The there predictors we identified as important reinforce some of the predictors we had already deemed significant from our analysis that being *ca*, *thal*, and *cp*.

For testing purposes, we created a Random Forest with the three significant predictors we found by plotting the full Random Forest Model, which were *ca*, *thal*, and *cp*. Below is the variable importance plot and statistics of this model.

```
# Convert num to factor (if needed)
HeartDisease$num <- as.factor(HeartDisease$num)
set.seed(123)

heart.rf.test = randomForest(num ~ ca + thal + cp, data = HeartDisease,
                         ntree = 500, mtry = sqrt(3), importance = T)

heart.rf.test
```

```
Call:
 randomForest(formula = num ~ ca + thal + cp, data = HeartDisease,      ntree = 500, mtry =
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 16.83%
Confusion matrix:
    0   1 class.error
0 143  21   0.1280488
1  30 109   0.2158273
```

```
varImpPlot(heart.rf.test)
```



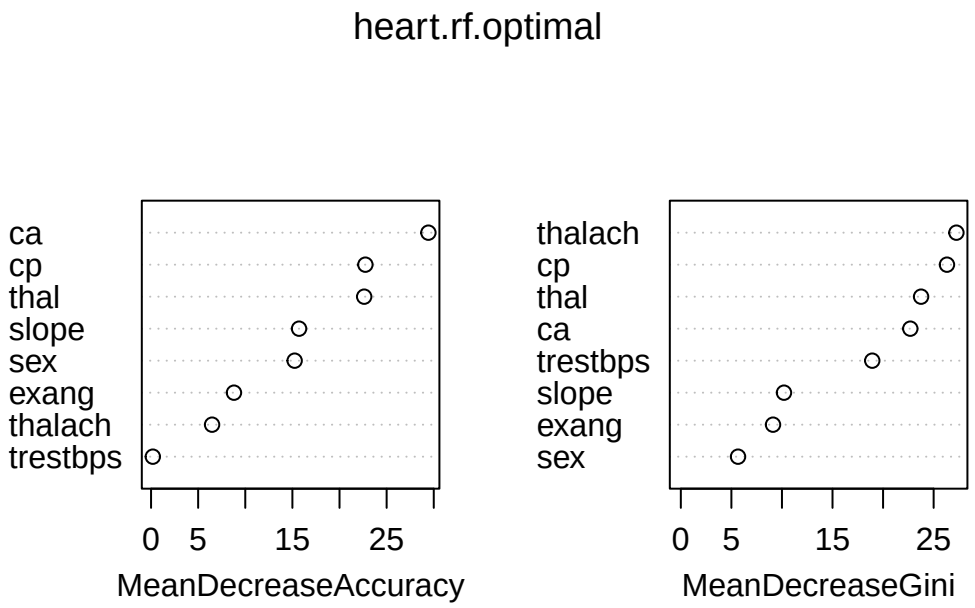

This particular model attained an error rate of 16.83% which is slightly better than the better-performing Logistic Regression Model, and slightly better than the full Random Forest Model we implemented previously. However, though the statistics of this model with these predictors give better performance, we decided to go with the significant predictors we obtained via Logistic Regression and train the Random Forest Model with these features.

Therefore, we trained a Random Forest Model with only the significant predictors that we established from our analysis with the Logistic Regression Model. Below is the training of the model.

```
# Convert num to factor (if needed)
HeartDisease$num <- as.factor(HeartDisease$num)
set.seed(123)
heart.rf.optimal = randomForest(num ~ sex + cp + trestbps + thalach + exang
                                + slope + ca + thal, data = HeartDisease,
                                ntree = 500,
                                mtry = sqrt(8), importance = T)

heart.rf.optimal
```

```
Call:
 randomForest(formula = num ~ sex + cp + trestbps + thalach +      exang + slope + ca + tha
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 20.13%
Confusion matrix:
    0   1 class.error
0 137  27   0.1646341
1  34 105   0.2446043
```

```
varImpPlot(heart.rf.optimal)
```




Here, the Random Forest Model trained with the significant features previously established and with 500 trees gave us an error of 20.13%, which is higher than the Logistic Model and higher than the Random Forest Model when all predictors were considered. However, this could potentially mean that a Logistic Model would be best in this scenario between the two. As such, we will continue our Random Forest analysis with the eight significant features that we noted. Also, from the variable importance plot, we can note that *cp* and *thal* are significant features here.

Next, we did an 80/20 split of the data to train the Random Forest Model on 80% of the data and train it on the remaining 20%. This was run for 10 iterations, and the error rate was calculated and stored for each iteration, from which we calculated the mean error rate to give us an overall error rate of the Random Forest Model via cross-validation. The model was trained with 500 trees.

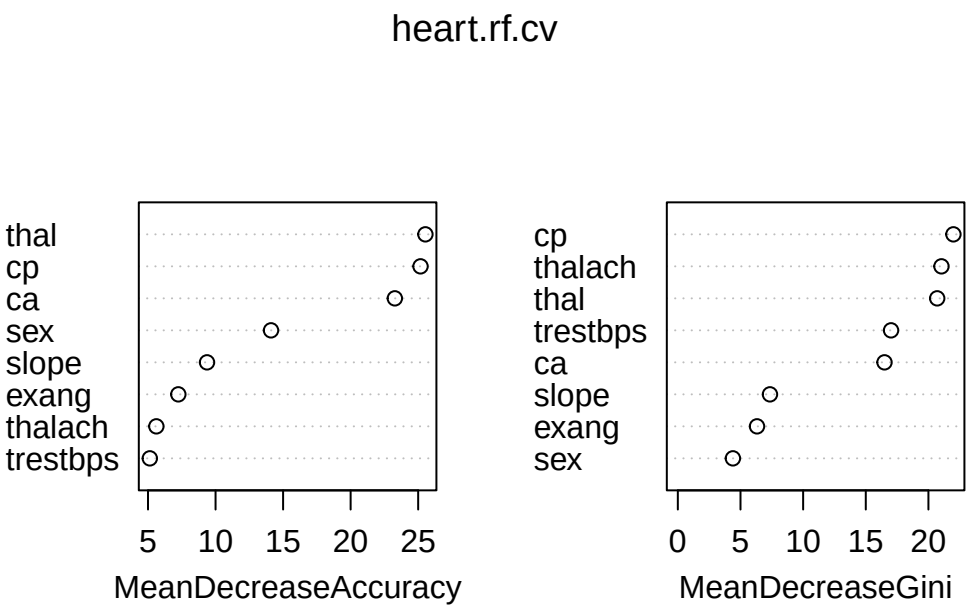
```
# Convert num to factor (if needed)
HeartDisease$num <- as.factor(HeartDisease$num)
# Random Forest Cross Validation - 80/20 Split
store.errorRate = rep(0, 10)

for(i in c(1: 10)){
  set.seed(i + 100)

  sample = sample.int(n = nrow(HeartDisease), size =
                        floor(.8 * nrow(HeartDisease)), replace = F)

  train = HeartDisease[sample,]
  test = HeartDisease[-sample,]

  heart.rf.cv = randomForest(num ~ sex + cp + trestbps + thalach + exang
                             + slope + ca + thal, data = train,
                             ntree = 500, mtry = sqrt(8), importance = T)

  heart.rf.pred = predict(heart.rf.cv, newdata = test)
  conf.mat = table(Predicted = heart.rf.pred, Actual = test$num)

  store.errorRate[i] = (conf.mat[1, 2] + conf.mat[2, 1])/sum(conf.mat)
}

store.errorRate
```

```
 [1] 0.1639344 0.2131148 0.2131148 0.1311475 0.1967213 0.2295082 0.2131148
 [8] 0.1803279 0.2295082 0.1967213
```

```
mean(store.errorRate)
```

```
[1] 0.1967213
```

```
varImpPlot(heart.rf.cv)
```




The average test error rate of the Random Forest Model results is 0.1967213 or 19.67%. This is higher than the mean test error rate of the Logistic Regression Model. Hence, we can say that the Logistic Regression Model is the better model to predict heart disease based on this dataset, which answers our main question. The Logistic Regression Model performed better, and we can assert that it is the better model to predict heart disease on this dataset based on the test error rates and model metrics/statistics. This could possibly be attributed to the low noise of the dataset, the size of the dataset, balanced class distribution, etc. Also, from the variable importance plot, we notice that *cp* and *thal* are significant predictors.

## Conclusion

In this project, we posed many questions concerning the prediction and inference of heart disease in patients and its inner workings. The Logistic Regression Model was utilized in feature importance to find the significant predictors of heart disease for this dataset. It proved useful and effective as it is one of the strengths of Logistic Regression Models and can be difficult when using Random Forest. This process answered one of our questions

of which features were important predictors of heart disease, which were *sex*, *cp*, *trestbps*, *thalach*, *exang*, *slope*, *ca*, and *thal*.

Through our testing, implementation, and analysis, we found the Logistic Regression to be a better-performing model when it comes to predicting heart disease, which was surprising, as we had assumed the Random Forest Model would have performed better due to its functionalities, but that proved to be untrue. Our Logistic Regression Model yielded a lower test error than the Random Forest Model. This answered the main question that we initiated.

Additionally, we also asked ourselves whether age played a significant role in predicting heart disease, which we found to be untrue, as it was not an important attribute concerning heart disease in patients.

Moreover, we raised the question of whether heart disease was more prominent in males or females. Our examination revealed that males were more susceptible to heart diseases than females, which was an eye-opening discovery.

Overall, through our exploration and research, we answered important questions in relation to heart disease in patients and trained two models in classifying patients with such an ailment.

Bibliography

Andras Janosi, William Steinbrunn. Heart Disease. UCI Machine Learning Repository, 1989. DOI.org (Datacite), https://doi.org/10.24432/C52P4X.

CDC. "Heart Disease Facts." Heart Disease, 7 Feb. 2025, https://www.cdc.gov/heart-disease/data-research/facts-stats/index.html.