Predicting Heart Attacks and Other Health Risks

Ryan Canfield

Dr. Fotios Kokkotos

DSE6111 Predictive Modeling

**Executive Summary:**

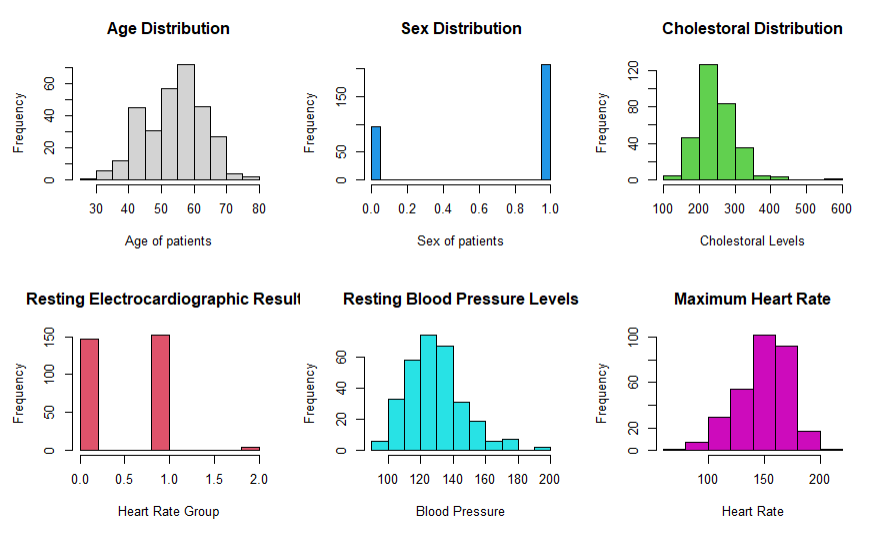
In the United States, someone has a heart attack every 40 seconds. Every year, about 805,000 people in the United States have a heart attack. It is a common result of cardiovascular disease that can easily be prevented. The main goal of this research project was to examine what key factors can lead to a heart attack. We will also be looking into what cholesterol level puts a person into the redzone of having a heart attack. By knowing these factors, we can help prevent heart attacks from happening because we will be able catch symptoms early. If the population learns what the most important factors are, it will lead to them having a better lifestyle because they can focus on keeping those leading risk factors in healthy zones. By trying and comparing different predictive models, the best one can be used on a patient's data to tell if they are in the danger zone to have a heart attack. The information from this report can be helpful to hospital workers and health clinics since it can be used to determine if that person will have a heart attack. This project has three main objectives. Objective one looks at cholesterol levels and tries to determine what number is safe and what number is unsafe. It uses the “cholesterol” variable, as a quantitative response variable. Objective two looks at whether someone is more or less likely to have a heart attack and uses an output variable, which is a qualitative response variable. It puts patients into groups and based on their other data, we are able to determine a solution. Objective three uses principal component regression (PCR) to answer the question “what is the most common factor when indicating if someone will have a heart attack”. The exact research aims are below.

1. Objective 1: To identify and quantify the factors that contribute to variations in cholesterol levels.
2. Object 2: To see what factors contribute the most to having a higher chance or a lower chance of a heart attack.
3. Objective 3: Using principal component regression, what is the most important factor when looking for signs of a heart attack.

Some key findings from this project are that while cholesterol can be used to predict heart attacks, it is difficult to determine if there is a cholesterol level that is more prone to attacks. This is due to the nature of the data and how it was recorded. From the second objective you can tell if someone is more or less likely to have a heart attack based on their gender, resting heart rate, and number of major blood vessels. Finally, after using PCR, the most common sign of a heart attack is chest pain. For future work, the data on cholesterol was limited, so more research and data collection is required in order to give a sufficient answer. If patients are observed again, they should measure the cholesterol level of patients who have had a heart attack. This would help determine what are safe and unsafe levels of cholesterol in regards to a heart attack. To help reduce the factors found from the research, I would recommend some form of light to moderate cardio every day, especially if you are male. This is because based on my findings, males are more likely to have heart attacks than females, and if the other factors are based on resting heart rate and the amount of major blood vessels, I would do activities that would lower a resting heart rate, which would in turn lower the chances of having a heart attack. Finally, since chest pain is the number one factor in determining a heart attack, if you are experiencing chest pain, I would recommend calling a doctor or ambulance immediately because there is a strong likelihood you are having a heart attack. In the event of a heart attack you should seek medical attention as soon as possible.

**Data & Approach:**

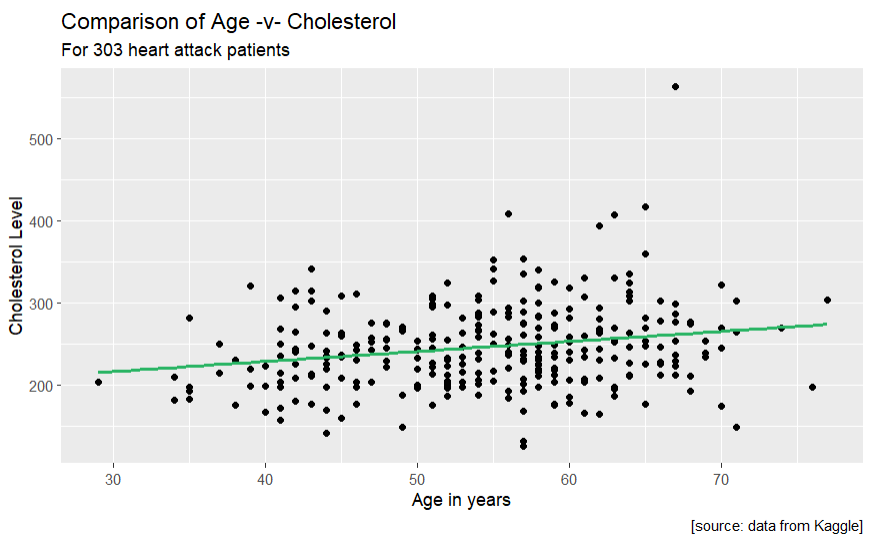
The data used in this analysis contains information about heart attacks, and was found on [www.kaggle.com](http://www.kaggle.com). Kaggle is a platform where data scientists and machine learning engineers can find data sets to analyze and use in case studies or research projects like this one. After performing analysis on a data set, if they choose, they can publish or share their results back on that website in hopes to share their findings and their techniques. They can also publish a data set that they collected themselves like the one from this project. The heart attack data set has 303 rows and fourteen columns, where each row represents a patient and each column represents an attribute or recorded measure. Some of the attributes measured are age, sex, resting heart rate, blood pressure, chest pain, cholesterol, maximum heart rate, and whether someone is more or less likely to have a heart attack. After loading the data into R, which is a statistical computing software, I was able to explore the data. From this exploration, I found that the ages of patients range from 29 years old to the oldest being 77 years old. The average age from this data set was 54 years old. This data set is made up of two hundred males and one hundred females, making it a 2 to 1 ratio of men and women. The data was split with either having a resting ecg that was normal or having ST-T wave abnormality. The average resting blood pressure is about 131.6mm Hg, and if you experienced chest pain it was most likely a typical angina. Further exploration was done in which a correlation matrix of all the variables was created and some histograms were created to further cement the distribution of data. Many of these variables were already mentioned above.



**Figure 1 (above):** Histogram of some of the variables.

Not much data re-engineering was done to this data set. It had no far reaching outliers which could have skewed data, and the data did not have any missing values. For my approach in this project, I decided to split the model into two groups. For one group, I used models that could take a continuous variable as its target variable and for the other I used models that needed a categorical variable. I used R^2 values and MSE values to measure and determine how well the continuous or regression models performed. For the models that used categorical variables, I was able to predict the accuracy, sensitivity, and precision, by looking at values in a confusion matrix. A confusion matrix lets you see all correct and incorrect labels a model made and this it's presented in a nice visual. The analytic goal for this is to find the strongest/ most accurate model for predicting and then interpret that model in order to tell a story with the data. Throughout the project, seventeen models were used. For the quantitative models, I used multiple regression in which I picked the best subset and used a stepwise method to find the best variables, ridge regression, lasso regression, partial least squares, regression trees, bagging methods, random forests, and boosting methods. For the qualitative models, I used K-Nearest Neighbors (KNN), logistic regression, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), classification trees, bagging methods for classification, and random forests for classification. Finally for the last aim, I used principal component regression in order to reduce some of the dimensions of the data set while retaining most of the information that each variable holds.

**Detailed Findings:**

For the first objective which looked at different cholesterol levels, I used nine different models. The first was a multiple linear regression model that used the best subset selection method to choose its predictors. The best subset selection method is when you run a linear regression model on all the variables first, and pick the most significant ones based on their p-values. After summarizing that first model, you pick the best variables from that model and use them in the actual model. From the first run of the model, the most significant variables were age, sex, and resting electrocardiographic results (restecg). This table shows that all p-values are below the 0.05 threshold which is commonly used to test for significance.

| Variable Name | P-Value |
| --- | --- |
| age | 0.003854 |
| sex | 0.000127 |
| restecg | 0.018983 |

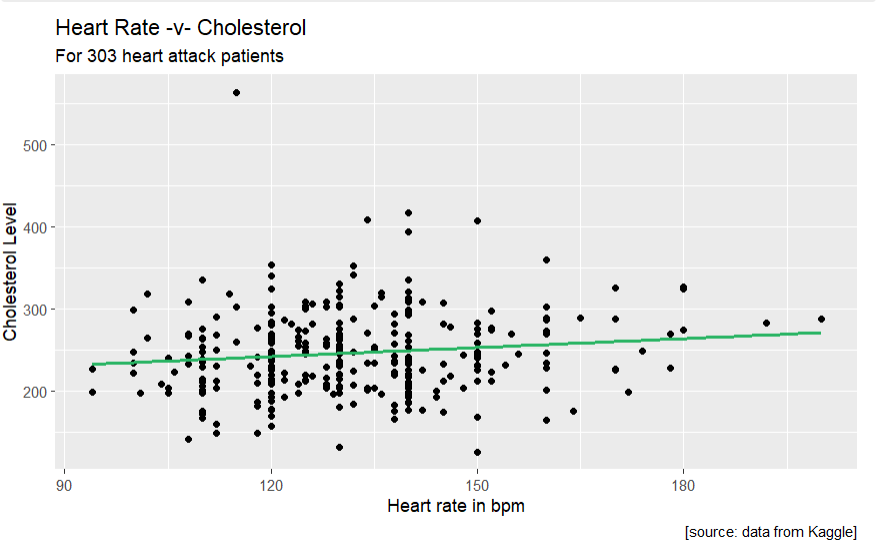
**Figure 2 (above):** Table of P-Values of the BSS linear regression model.

**Figure 3 (right):** Visualization of the linear regression model.

When graphing this model, I used age because it appeared first in the summary table. Even though all the values are significant, the data has a weak correlation which you can see from the image to the right. As a reminder, we are trying to get all of the data points on the line or as close to the line as possible. This model had an R^2 value of 0.087. This means that it can explain 8% of the variability in the data. In all cases, the higher the R^2 value, the better the model is, so this shows that the model struggles when predicting cholesterol level. The next linear regression model I used was forward stepwise regression. Forward stepwise regression works by having the model try every possible combination until it finds the best combination of variables for the model.

| Variable Name | P-Value |
| --- | --- |
| age | 0.002871 |
| sex | 0.000139 |
| restecg | 0.010005 |
| thall | 0.019664 |

**Figure 4 (above):** Table of P-Values of the forward stepwise regression model.

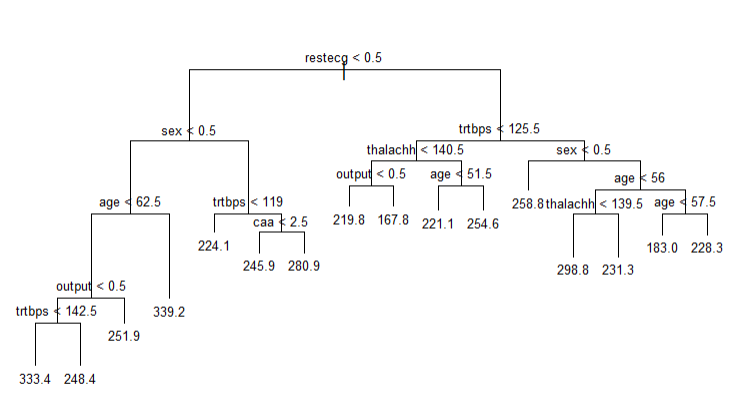
**Figure 5 (left):** Visualization of the forward stepwise linear regression model.

After running the model, I get similar variables but this time thall (maximum heart rate) was added and the p-values were obtained. Even with the model finding another variable to add, it only adds one percent to the R^2 value making it go from a 0.08 to a 0.09. A similar graph shows that while the output is better, it is still not good enough to make predictions. From both of these models, we can conclude that the correlations are not strong enough to make a decision about cholesterol. We can however use other stronger regression models. Ridge, lasso, and partial least squares (PLS) regression are other models we can use to show the relationship between the explanatory variables and cholesterol. Ridge regression is used to analyze data when the data set has high multicollinearity. Multicollinearity is a statistical concept where several independent variables in a model are correlated. Lasso regression is a method that performs both selection and regularization in order to enhance the accuracy of a model. Finally PLS can be used similarly to ridge regression to get rid of variables that show similar information. Below is a table that shows each model’s MSE or mean square error. MSE is a way to look at how well a model does in predicting the outcome for a certain variable. When looking at the table, we are trying to get the numbers as close to zero as possible so the lower the number the better. I also included the other two models MSE for comparison.

| Model | BBS linear Regression | Forward Stepwise | Ridge Regression | Lasso Regression | Partial Least Squares |
| --- | --- | --- | --- | --- | --- |
| MSE | 2474.45 | 2450.777 | 2077.706 | 2107.66 | 2134.263 |

**Figure 6 (above):** Table comparing Mean square errors.

As you can see from the table, while the three new models did better, their mean square error is still very high which means not one of them is strong enough to make valid predictions. Out of the five models, ridge regression is the best. This could be because the data has a lot of overlap so ridge regression helps to weed out what is not as important. Instead of looking at these variables in a linear fashion, we can use other models that can still handle continuous numbers but look at each variable in a different way. The models that do this are regression trees, bagging random forests, and boosting. Regression trees is a model that splits up data based on continuous data instead of labels. At each split, two more branches are created until it ends at a node. These nodes hold the predicted answer for that variable. Below is an image that demonstrates this.



**Figure 7 (left):** Regression tree.

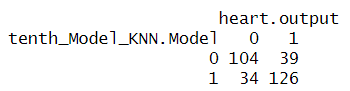
On this tree, the model splits it on different variables until it reaches an answer. To follow this tree you would start at the top. If your restecg value was below a 0.5 you would travel on the left side until the next branch. Here you would see if your sex value is higher or lower than 0.5. If it is below that number then you will continue to follow the left branch. You keep following the tree until you get at the bottom. For example, if you follow the left side all the way down, then your predicted cholesterol level would be 333.4. Bagging, boosting and random forests are all techniques you can do which involve combining the predictions of multiple individual models to the best overall model. The idea of bagging or bootstrap aggregation aims to reduce overfitting to improving the accuracy of a model by training multiple instances of the same model. Boosting focuses on improving accuracy of multiple weak models by combining them into a stronger model. Random forests are an extension of bagging that takes a majority vote from the individual trees to make predictions. Below is how well these models performed when trying to predict cholesterol levels. I also added in the regression tree MSE and the bridge recession MSE since it was the best out of the other five models.

| Model | Ridge Regression | Regression Tree | Bagging | Random Forests | Boosting |
| --- | --- | --- | --- | --- | --- |
| MSE | 2077.706 | 2263.066 | 2087.262 | 2111.701 | 2,227.433 |

**Figure 8 (above):** 2nd Table of MSE when finding cholesterol levels.

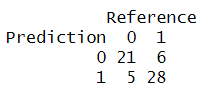
As you can see from the table above, ridge regression is still the best model when predicting cholesterol. Some reasons why ensemble techniques like bagging, boosting, and random forests might not be as powerful as regression is because they are complex and need a large set of data to perform well. Ensemble techniques like the ones used above are good at finding nonlinear relationships. However, they might not be as efficient as linear regression because they add a layer of complexity that might not be necessary. These techniques usually require a lot more data to perform well and capture the variability that is being introduced during the combining process. Since this is a smaller data set, these models can tend to over fit the data where the model will send all the data to one branch and only have a few values in each other node instead of it being spread out evenly. After looking at all the models that can take a continuous target variable, we can say that we were not able to identify and quantify any factors that contribute to cholesterol levels because all the models we tried were not strong enough to make accurate predictions. Ridge regression worked the best but that had a MSE of over 2,000 which is way too high to make appropriate decisions. We also have to take into account that we are dealing with real people with heart diseases so models have to be even stronger, have really low MSE’s, and accuracies in the high nineties. This is because you do not want to misclassify a patient and harm them even more.

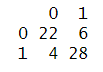
For the second objective, seeing what factors contribute the most for a patient to have a more or less likely chance of a heart attack, we used seven models, first was KNN. KNN uses proximity to make classifications or predictions about the grouping of an individual data point. For all of these classification models I was able to produce a confusion matrix used to show accuracy for which points are classified correctly and incorrectly.



**Figure 9 (left):** Knn Confusion Matrix

From this matrix you can see that the model classified 104 people correctly who were less likely to have a heart attack and correctly classified 126 people who were more likely to have a heart attack. However it misclassified 34 people saying they were less likely to have a heart attack when in actuality they were more likely to have a heart attack. 39 people the opposite way. KNN gets an accuracy of 76% which you get from 230 / 303. Next is logistic regression. Logistic regression is used to predict binary outcomes whether a target is 0 or 1 in this case 0 is less likely to have a heart attack and 1 is more likely. I made a forward stepwise logistic regression model which has the same idea as the other stepwise model. Here it tries all the combinations of variables until it finds the strongest one. After running the model, it found these variables to be most useful: cp (Chest Pain), oldpeak, caa(Major Vessels), sex, thalachh (Max Heart Rate), thall, exng (Exercise induced pain), trtbps (blood pressure) and chol. Its confusion matrix looks like this.

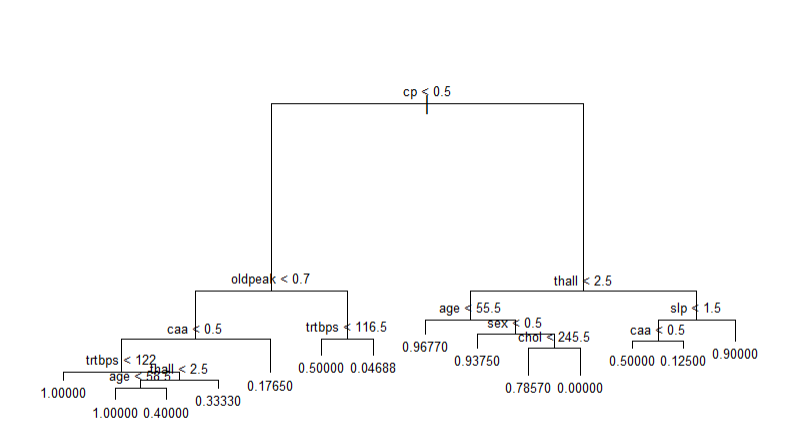
 **Figure 10 (left):** Logistic regression confusion matrix.

The reason why this confusion matrix has much lower numbers is because I split the data into a training and testing set. The idea behind this is to build your models on most of the data, and then you test how well the model does on new similar data that your model has never seen before. Here the accuracy of the logistic regression model is 81.67%. The next two models are very similar. LDA and QDA are classification models that reduce dimensions in order to try and find the best accuracy. LDA reduces dimensions or removes columns by finding a linear combination of variables that separates two or more classes. It aims to maximize the means while minimizing the variance. QDA reduces dimensions by a quadratic decision boundary. This allows for more flexibility in choosing classes. Both are techniques for classification and dimensionality reduction. Their confusion matrices look like this. 

**Figure 11 (left):** LDA confusion matrix.

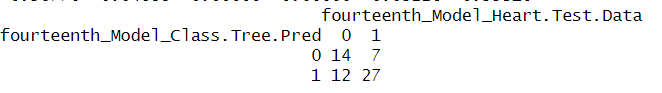
**Figure 12 (right):** QDA confusion matrix.

When reducing dimensions both models take a little bit of each variable to make up its new dimension. For LDA it is mostly made up of sex and exng (exercise induced angina). LDA’s accuracy is 81.67% classifying 49 out of the 60 labels correctly. QDA’s dimension is reduced down to chest pain and cholesterol, and its accuracy is 83.33%. The last three models are the ensemble methods, but this time they are applied to classification instead of regression. A classification tree is similar to a regression tree, but instead of a number it chooses a class label to make its decision. The classification tree I made to determine what factors are more likely to have a heart attack looks like this.



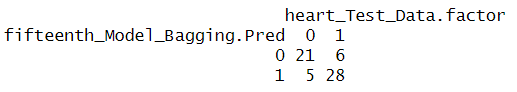
**Figure 13 (above):** Classification tree

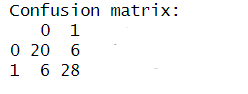
This tree above mostly uses chest pain, oldpeak, and thall to make its decisions. The nodes in this case are not actually the value, but are the probabilities that someone will be more likely to have a heart attack. So if you follow the branch and get to .1765, you have a 17.65% chance to be more likely to have a heart attack. The confusion matrix for this model is as follows.



**Figure 14 (above):** Classification tree confusion matrix.

This confusion matrix did not do as well as the others only scoring a 68%. This is due to the overfitting we talked about earlier which can be seen in the false negative value seen in the bottom left of the confusion matrix. Here are the accuracies for the bagging and random forests models.



**Figure 15 (left):** Bagging confusion matrix.

**Figure 16 (left):** Random forest confusion matrix.

The ensemble methods did a much better job on the classification data, then the regression data. Bagging had a total of 49 correct labels and 11 misclassified labels giving it an overall accuracy of 81.67%. Random forests had 48 correct and 12 incorrect giving it an accuracy of 80%. Below is the final summary table of all the classification models. This table includes accuracy which is the total correct divided by the total number of patients in the table. It also includes sensitivity and precision which are just other measures people use to tell a model's performance. Sensitivity is calculated by dividing the true positive number by the true positive plus the false negatives (bottom right value). Precision is calculated by the number of true positives divided by the true positives plus false positives (top right value).

**Figure 17 (below):** Total summary of performance measures on all the classification models.

| Model Used | KNN | Logistic Regression | LDA | QDA | Classification tree | Bagging | Random Forests |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Accuracy | 75.91% | 81.67% | 81.67% | 83.33% | 68.33% | 81.67% | 80% |
| Sensitivity | 75.36% | 80.77% | 76.92% | 84.62% | 53.85% | 80.77% | 76.92% |
| Precision | 72.73% | 77.78% | 80% | 78.57% | 66.67% | 77.78% | 76.92% |

From this table the top three models are QDA, LDA, and logistic regression. Most likely reason why QDA and LDA did so well is probably due to the nature of the data. This data appears to have a lot of correlations, meaning that certain columns hold the same or similar data. Which is why these models can be built with different variables but still have relatively the same accuracy. To answer the objective and see what factors contribute the most to having a more likely chance of a heart attack or a less likely chance of a heart attack, I would use the QDA model. QDA was built using chest pain, caa or major vessels, and cholesterol level. To decrease the chance of the likelihood of you having a heart attack I would focus on these three variables and try to mitigate my chances by having no chest pain, all of my major vessels working, and having low cholesterol.

For the last aim it was geared towards solving it using principal component regression. PCR is a predictive model that combines principal component analysis (PCA) and multiple linear regression. PCA is another dimension reduction technique that removes variables based on variance. The point of PCA is to keep the most information in as few columns as possible. To do this it takes variance from every variable but some variables have influence on the dimension then others. Some advantages of PCR can be found with highly correlated predictors. My third objective is to PCR to see what is the most important factor when looking for signs of a heart attack. After performing PCR the model determined that four dimensions should be used to create the model. These four dimensions contained 50% of the variance. Its MSE was 0.1351897. After analyzing the results further it appears that chest pain is the variable that gives dimensions two and three most of its variance. From this I can conclude that chest pain is the most common symptom of having a heart attack.

**Validity & Reliability Assessment:**

Throughout the various research and testing done on this project, the models that have been created are not accurate enough to be reliable, therefore should be invalid when used to predict symptoms of heart attacks. This data is geared strongly towards predicting heart attacks which can clearly be seen throughout the first aim. Both R^2 values I was able to pull out of the linear regression models were below .1 which means that there is very low correlation between those variables and trying to predict cholesterol levels. The MSE on all the regression models were very high too which further proves this point. Part of this could be the data itself. It is a relatively small data set with only three hundred rows. This gives us a sample that is very hard to work with because it is hard to find definite emerging patterns within such little data. The column values while at first seemed very promising had a lot of issues. First, there were a lot of categorical variables where numerical would have been better. Second, when fitting the data it seemed like the data was fitted many times. In both tree models, the model focused on the noise in the data rather than finding the underlying patterns. If a bunch of random samples were created on the built models I think they would perform poorly and give biased answers due to the small data set as well. Another problem was due to the multicollinearity. This was seen in most of all the models. Since the data was all highly correlated with each other it made most models perform poorly. The only models that did well were the ones that were meant to deal with this problem. The ridge regression LDA, QDA, and PCR were all models made to deal with this and reduce its dimensions or eliminate some of the columns that were used as explanatory variables. In conclusion, after performing a thorough validity and reliability assessment, it is seen that these models did not meet the standard required to predict outcomes. Especially in data with real patients, it is essential that these models are accurate, strong, and applicable to other future implementations.

**Appendix:**

Predictive Modeling Assignment #8 – Ryan Canfield – 2023-12-15 – R Markdown

Module 08: Assignment 01 - Final Project

# Opening the data set and getting some different previews of the data.

heart <- read.csv("../data sets/heart.csv")  
 head(heart)

# About this data set:

# Age : Age of the patient

# Sex : Sex of the patient

# exang : exercise induced angina (1 = yes; 0 = no)

# ca : number of major vessels (0-3)

# cp : Chest Pain type chest pain type Value 1: typical angina Value 2: atypical angina Value 3:non-anginal pain Value 4: asymptomatic

# trtbps : resting blood pressure (in mm Hg)

# chol : cholesterol in mg/dl fetched via BMI sensor

# fbs : (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)

# rest\_ecg : resting electrocardiographic results Value 0: normal Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV) Value 2: showing probable or definite left ventricular hypertrophy by Estes’ criteria thalach : maximum heart rate achieved

# target : 0 = less chance of heart attack 1= more chance of heart attack

## Exploratory data analysis

summary(heart)

cat("\n")

glimpse(heart)

*# Correlation matrix* pairs(heart)

*# Looking at different histograms to get a better idea of the data* par(mfrow = c(2,3))  
 hist(heart$age, xlab = "Age of patients", main = "Age Distribution")  
 hist(heart$sex, col = 4, breaks = 20, xlab = "Sex of patients", main = "Sex Distribution")  
 hist(heart$chol, col = 3, xlab = "Cholestoral Levels", main = "Cholestoral Distribution")  
 hist(heart$restecg, col = 2, breaks = 10, xlab = "Heart Rate Group", main = "Resting Electrocardiographic Results")  
 hist(heart$trtbps, col = 5, xlab = "Blood Pressure", main = "Resting Blood Pressure Levels")  
 hist(heart$thalach , col = 6, breaks = 8, xlab = "Heart Rate", main = "Maximum Heart Rate")

## Preprocessing

*# count total missing values* print("Count of total missing values - ")

sum(is.na(heart))

Objective 1 (Quantitative Response Variable): To identify and quantify the factors that contribute to variations in cholesterol levels.

First Model: Multiple Regression – Best Subset Selection

*# Preliminary linear regression model to see what variables are statsitcally significiant.* first\_Model <- lm(chol ~ ., data = heart)  
 summary.lm(first\_Model)

*# Taking the significant variables to produce the find BSSmultiple linear regression model.* first\_Model\_Final <- lm(chol ~ age + sex + restecg, data = heart)  
 summary.lm(first\_Model\_Final)

#Plotting the model.

model1 <- ggplot(data = heart, aes(x = trtbps, y = chol)) +

geom\_point () +

labs(title = "Heart Rate -v- Cholesterol ",

subtitle = "For 303 heart attack patients",

x = "Heart rate in bpm",

y = "Cholesterol Level",

caption = "[source: data from Kaggle]") +

geom\_smooth(method = "lm", se = 0, colour = "#28B463")

model1

Second Model: Forward Stepwise Regression.

*# To run stepwise regression:* *# First, defining the null model. We need this for forward stepwise regression:* second\_Model\_Intercept\_Only <- lm(chol ~ 1, data = heart)  
  
 *# Next, have a model with all explanatory variables included.* second\_Model\_All\_Variables <- lm(chol ~ ., data = heart)  
 *# The "." is a symbol that lets us include all the variables.* *# Perform forward stepwise regression here, using the step function.* *# We start from the intercept\_only model and try every combonation.  
 # We tell "step" the direction of stepwise regression we want. .* second\_Model <- step(second\_Model\_Intercept\_Only, direction = 'forward', scope = formula(second\_Model\_All\_Variables), trace = 0)  
  
 *# If we want to see the output of the forward stepwise regression, we can use this command:* second\_Model$anova

*# This gets us our coefficients for the model.* second\_Model$coefficients

*# Adding the variables above to find the best linear model.* second\_Model\_Final <- lm(chol ~ age + sex + restecg + thall, data = heart)  
 summary.lm(second\_Model\_Final)

#Plotting the model.

model2 <- ggplot(data = heart, aes(x = thall , y = chol)) +

geom\_point () +

labs(title = "Comparison of Age -v- Cholesterol ",

subtitle = "For 303 heart attack patients",

x = "Age in years",

y = "Cholesterol Level",

caption = "[source: data from Kaggle]") +

geom\_smooth(method = "lm", se = 0, colour = "gold")

model2

For future models.

Splitting the data into training and testing set.

*# Split the data into training and test set.* set.seed(310)  
  
 *# This has all the data* training\_Samples <- heart$output %>%  
 createDataPartition(p = 0.8, list = FALSE)  
  
 heart\_Train\_Data<- heart[training\_Samples, ]  
 heart\_Test\_Data <- heart[-training\_Samples, ]  
  
 *# If need here is the split separated by the data and then the target variable.* heart\_TrainX <- heart\_Train\_Data[c(1:13)]  
 heart\_Trainy <- heart\_Train\_Data[c(14)]  
 heart\_TestX <- heart\_Test\_Data[c(1:13)]  
 heart\_Testy <- heart\_Test\_Data[c(14)]

Third Model: Ridge Regression

set.seed(310)  
  
 *# Setting up variables for Ridge Regression.* heart\_TrainSet\_Matrix <- model.matrix(chol ~ ., data = heart\_Train\_Data)  
 heart\_TestSet\_Matrix <- model.matrix(chol ~ ., data = heart\_Test\_Data)  
 grid <- 10 ^ seq(10, -2, length = 100)  
  
 *# Fitting the model following the book.* third\_Model\_RidgeR.fit <- glmnet(heart\_TrainSet\_Matrix, heart\_Train\_Data$chol, alpha = 0, lambda = grid, thresh = 1e-12)  
 third\_Model\_RidgeR.cv <- cv.glmnet(heart\_TrainSet\_Matrix, heart\_Train\_Data$chol, alpha = 0, lambda = grid, thresh = 1e-12)  
 third\_Model\_RidgeR.lambda.cv <- third\_Model\_RidgeR.cv$lambda.min  
  
 third\_Model\_RidgeR.pred <- predict(third\_Model\_RidgeR.fit, s = third\_Model\_RidgeR.lambda.cv, newx = heart\_TestSet\_Matrix)  
 print("The MSE on the Ridge Regression's testing set is")

mean((third\_Model\_RidgeR.pred - heart\_Test\_Data$chol)^2)

Fourth Model: Lasso Regression

set.seed(310)  
  
 *# Fitting a lasso model on the training set* fourth\_Model\_Lasso.fit <- glmnet(heart\_TrainSet\_Matrix, heart\_Train\_Data$chol, alpha = 1, lambda = grid, thresh = 1e-12)  
  
 *# Choosing lambda by crossvalidation* fourth\_Model\_Lasso.cv <- cv.glmnet(heart\_TrainSet\_Matrix, heart\_Train\_Data$chol, alpha = 1, lambda = grid, thresh = 1e-12)  
 fourth\_Model\_Lasso.lambda.cv <- fourth\_Model\_Lasso.cv$lambda.min  
  
 *# Reporting the testing error* fourth\_Model\_Lasso.pred <- predict(fourth\_Model\_Lasso.fit, s = fourth\_Model\_Lasso.lambda.cv, newx = heart\_TestSet\_Matrix)  
 print("The MSE on the Lasso Model's testing set is")

mean((fourth\_Model\_Lasso.pred - heart\_Test\_Data$chol)^2)

*# Finding the Lasso's coefficients* fourth\_Model\_Lasso.coef <- predict(fourth\_Model\_Lasso.fit, s = fourth\_Model\_Lasso.lambda.cv, type = "coefficients")  
 print("The Lasso coefficients are:")

## [1] "The Lasso coefficients are:"

round(fourth\_Model\_Lasso.coef, 3)

Fifth Model: Partial Least Squares

*# Fitting the PLS model on the training set and looking at which M to choose.* set.seed(310)  
 fifth\_Model\_PLS.fit <- plsr(chol ~ ., data = heart\_Train\_Data , scale = TRUE, validation = "CV")  
 summary(fifth\_Model\_PLS.fit)

*# Finding out where dimensional stops being reduced used for ncomp below.* validationplot(fifth\_Model\_PLS.fit, val.type = "MSEP")

*# Finding the PLR testing error.* fifth\_Model\_PLS.pred <- predict(fifth\_Model\_PLS.fit, heart\_Test\_Data, ncomp = 2)  
 print("The MSE on the PLS Model's testing set is")

mean((fifth\_Model\_PLS.pred - heart\_Test\_Data$chol)^2)

Sixth Model: Regression Trees

*# Fitting the regression trees.* sixth\_Model\_Reg.Tree.Model <- tree(chol ~ ., heart\_Train\_Data)  
  
 *# Plotting the regression trees.* plot(sixth\_Model\_Reg.Tree.Model)  
 text(sixth\_Model\_Reg.Tree.Model, pretty = 0, cex = 0.65)

*# Summarizing the results.* summary(sixth\_Model\_Reg.Tree.Model)

*# Getting the test MSE* sixth\_Model\_Test.Pred <- predict(sixth\_Model\_Reg.Tree.Model, heart\_Test\_Data)  
 mean((sixth\_Model\_Test.Pred - heart\_Test\_Data$chol)^2)

*# Getting the train MSE.* sixth\_Model\_Train.Pred <- mean(heart\_Train\_Data$chol)  
 mean((sixth\_Model\_Train.Pred - heart\_Test\_Data$chol)^2)

*# Using cross-validation in order to determine the optimal level of tree complexity and seeing if it helped improve the MSE.* sixth\_Model\_.CV.Reg.Tree.Model <- cv.tree(sixth\_Model\_Reg.Tree.Model)  
 plot(sixth\_Model\_.CV.Reg.Tree.Model$size , sixth\_Model\_.CV.Reg.Tree.Model$dev, type = "b")

sixth\_Model\_Pruned.Tree.Model <- prune.tree(sixth\_Model\_Reg.Tree.Model, best = 3)  
  
 sixth\_Model\_Test.Pred <- predict(sixth\_Model\_Pruned.Tree.Model, heart\_Test\_Data)  
 mean((sixth\_Model\_Test.Pred - heart\_Test\_Data$chol)^2)

Seventh Model: Bagging

set.seed(310)  
  
 *#Creating the bagging model.* Seventh\_Model\_Bagged.Trees.Model <- randomForest(y = heart\_Train\_Data$chol, x = heart\_Train\_Data[ , -5],  
mtry = ncol(heart\_Train\_Data) - 5, importance = T)  
 *# Looking at the preformance of the model.* Seventh\_Model\_Test.Pred <- predict(Seventh\_Model\_Bagged.Trees.Model, heart\_Test\_Data)  
 mean((Seventh\_Model\_Test.Pred - heart\_Test\_Data$chol)^2)

## [1] 2087.262

*# Getting the node purity here.* importance(Seventh\_Model\_Bagged.Trees.Model) %>%  
 as.data.frame() %>%  
 rownames\_to\_column() %>%  
 arrange(desc(IncNodePurity))

Eighth Model Random Forests

set.seed(310)  
 eighth\_Model\_RF.High <- randomForest(chol ~ ., data = heart\_Train\_Data, mtry = 3, importance = TRUE)  
  
 eigth\_Model\_Yhat.RF <- predict(eighth\_Model\_RF.High, heart\_Test\_Data)  
 mean((eigth\_Model\_Yhat.RF - heart\_Test\_Data$chol)^2)

importance(eighth\_Model\_RF.High)

varImpPlot(eighth\_Model\_RF.High)

Ninth Model: Boosting

set.seed(310)  
  
 ninth\_Model\_Lambda.Seq <- 10^seq(-5, 0, 0.1)  
  
 ninth\_Model\_Heart.Train.MSE <- c()  
 ninth\_Model\_Heart.Test.MSE <- c()  
  
 for (i in 1:length(ninth\_Model\_Lambda.Seq)) {  
  
 ninth\_Model\_Boost.Heart <- gbm(chol ~ . - chol, data = heart\_Train\_Data, distribution = "gaussian", n.trees = 1000, interaction.depth = 4,  
 shrinkage = ninth\_Model\_Lambda.Seq[i])  
  
 ninth\_Model\_Heart.Train.MSE[i] <- mean((predict(ninth\_Model\_Boost.Heart, heart\_Train\_Data, n.trees = 1000) - heart\_Train\_Data$chol)^2)  
 ninth\_Model\_Heart.Test.MSE[i] <- mean((predict(ninth\_Model\_Boost.Heart, heart\_Test\_Data, n.trees = 1000) - heart\_Test\_Data$chol)^2)  
  
 }  
  
 summary(ninth\_Model\_Boost.Heart)

Object 2 (Qualitative Response Variable): To see what factors contribute the most to having a more likely of a chance for a heart attack or a less likely of a chance for a heart attack.

Tenth Model: KNN

set.seed(310)  
 tenth\_Train.X = data.frame(heart$cp)  
 tenth\_Test.X = data.frame(heart$cp)  
 heart.output = heart$output  
  
 *# Creating the modeling.* tenth\_Model\_KNN.Model = knn(tenth\_Train.X, tenth\_Test.X, heart.output, k = 1)  
 table(tenth\_Model\_KNN.Model, heart.output)

11th Model: Logistic Regression

*# Forward Stepwise Logestic Regression  
  
 # Fit an intercept-only model* heart\_Null\_Model <- glm(output ~ 1, data = heart\_Train\_Data, family = binomial)  
  
 *# fit a model with everything* heart\_All\_Model <- glm(output ~ ., data = heart\_Train\_Data, family = binomial)  
  
 *# Forward stepwise selection using AIC with both null and full models* heart\_Final\_Model <- stepAIC(heart\_Null\_Model, scope = list(lower = heart\_Null\_Model, upper = heart\_All\_Model), direction = "forward", trace = 0)  
  
 *# Display the final model summary* summary(heart\_Final\_Model)

*# Obtain predicted probabilities on the testing set* predicted\_probs <- predict(heart\_Final\_Model, newdata = heart\_Test\_Data, type = "response")  
  
 *# Assuming you have the true outcomes for the testing set (test\_data$output)* observed\_responses <- as.factor(heart\_Test\_Data$output)  
  
 *# Convert predicted probabilities to binary predictions (e.g., using a threshold of 0.5)* predicted\_classes <- as.factor(ifelse(predicted\_probs >= 0.5, 1, 0))  
  
 *# Create and displaying the confusion matrix* conf\_matrix <- confusionMatrix(predicted\_classes, observed\_responses)  
 Conf\_matrix

12th Model: LDA

*# Creating the model based on variables from logistic regression* twelfth\_Model\_LDA.Model = lda(output ~ cp + oldpeak + caa + sex + thalachh + thall + exng + trtbps + chol, data = heart\_Train\_Data)  
 twelfth\_Model\_LDA.Model

*# Making a confusion matrix to check accuracy.* twelfth\_Model\_LDA.Model.Pred = predict(twelfth\_Model\_LDA.Model, heart\_Test\_Data)  
 table(twelfth\_Model\_LDA.Model.Pred$class, heart\_Test\_Data$output)

13th Model: QDA

*# Creating the model.* thirteenth\_Model\_QDA.Model = qda(output ~ cp + oldpeak + caa + sex + thalachh + thall + exng + trtbps + chol, data = heart\_Train\_Data)  
 thirteenth\_Model\_QDA.Model

*# Checking its accuracy.* thirteenth\_Model\_QDA.Model.Pred = predict(thirteenth\_Model\_QDA.Model, heart\_Test\_Data)  
 table(thirteenth\_Model\_QDA.Model.Pred$class, heart\_Test\_Data$output)

14th Model: Classification Trees

*# Creating the model* fourteenth\_Model\_Class.Tree <- tree(output ~ .-output, data = heart\_Train\_Data)  
 summary(fourteenth\_Model\_Class.Tree)

*# Plotting the tree.* plot(fourteenth\_Model\_Class.Tree)  
 text(fourteenth\_Model\_Class.Tree , pretty = 0, cex = 0.65)

*# Checking the accuracy with a confusion matrix.* fourteenth\_Model\_Class.Tree.Pred <- predict(fourteenth\_Model\_Class.Tree, heart\_Test\_Data)  
 fourteenth\_Model\_Class.Tree.Pred <- as.factor(ifelse(fourteenth\_Model\_Class.Tree.Pred >= 0.5, 1, 0))  
 fourteenth\_Model\_Heart.Test.Data <- as.factor(heart\_Test\_Data$output)  
 table(fourteenth\_Model\_Class.Tree.Pred, fourteenth\_Model\_Heart.Test.Data)

15th Model: Bagging

set.seed(310)  
  
 heart\_Train\_Data.factor <- as.factor(heart\_Train\_Data$output)  
 heart\_Test\_Data.factor <- as.factor(heart\_Test\_Data$output)  
  
 *#Creating the bagging model.* fifteenth\_Model\_Bagging <- randomForest(y = heart\_Train\_Data.factor, x = heart\_Train\_Data[ , -14],ntree = 100, importance = T)  
  
 *# Looking at the performance of the model.* fifteenth\_Model\_Bagging.Pred <- predict(fifteenth\_Model\_Bagging, heart\_Test\_Data)  
 table(fifteenth\_Model\_Bagging.Pred, heart\_Test\_Data.factor)

16th Model: Random Forests

set.seed(310)  
  
  
  
 sixteenth\_Model.Random.Forests.Train<- randomForest(heart\_Train\_Data.factor ~ cp + oldpeak + caa + sex + thalachh + thall + exng + trtbps + chol,  
 data = heart\_Train\_Data, ntree = 200, mtry = 3)  
 sixteenth\_Model.Random.Forests.Train

sixteenth\_Model.Random.Forests.Test<- randomForest(heart\_Test\_Data.factor ~ cp + oldpeak + caa + sex + thalachh + thall + exng + trtbps + chol,  
data = heart\_Test\_Data, ntree = 200, mtry = 3)  
 sixteenth\_Model.Random.Forests.Test

Objective 3 (Principal Components Regression)

Final Model: Principal Components Regression

*# Fitting the PCR model on the training set and looking at which M to choose.* set.seed(310)  
 final\_Model\_PCR.Fit <- pcr(output ~ ., data = heart\_Train\_Data , scale = TRUE, validation = "CV")  
 summary(final\_Model\_PCR.Fit)

*# Finding out where dimensional stops being reduced used for ncomp below.* validationplot(final\_Model\_PCR.Fit, val.type = "MSEP")

*# Finding the PCR testing error.* final\_Model\_PCR.Pred <- predict(final\_Model\_PCR.Fit, heart\_Test\_Data, ncomp = 4)  
 print("The MSE on the PCR Model's testing set is")

mean((final\_Model\_PCR.Pred - heart\_Test\_Data$output)^2)

heart\_Train\_Data2 <- heart\_Train\_Data[c(-14)]

pca\_result <- prcomp(heart\_Train\_Data2, center = TRUE, scale. = TRUE)

pca\_result