# Module 6 — Final Project: Final Report

Utkarsh Kothari, Jatin Satija, Pruthvi Samir Patel

Master of Professional Studies in Analytics, Northeastern University

ALY6015: Intermediate Analytics

Dr. Harpreet Sharma

Feb 11, 2024

## Introduction

Proceeding with the "Cardiovascular Diseases Risk Prediction" (Cardiovascular Diseases Risk Prediction Dataset, 2017) project. This report will discuss the model that we plan to construct in phases, with the goal of identifying the most promising model that can more precisely address each of our questions.

## Methods

We will apply two techniques to create the model that would respond to our questions: logistic regression and linear regression. The link between one or more independent factors and a continuous dependent variable (like BMI) is modeled using linear regression. On the other hand, when the dependent variable is binary (such as the existence or absence of cardiovascular disease), logistic regression is utilized to forecast the likelihood that the outcome would develop based on one or more independent variables.

## Analysis

### Question 1: How do lifestyle decisions, including exercise habits, General health and eating habits, influence Body Mass Index (BMI) and, in turn, impact cardiovascular health?

***Hypothesis Summary***

**Figure 1**

*Hypothesis Test Results*

```
variance.
> if (p_value_for_interaction < 0.05) {
+   print("Reject the null hypothesis for interaction. Hence, 'Exercise' and 'General Health' interact significantly to expla
in BMI variance.")
+ } else {
+   print("Fail to reject the null hypothesis for interaction. Hence, There is not enough interaction between 'Exercise' and
'General Health' to explain the variation in BMI.")
+ }
[1] "Reject the null hypothesis for interaction. Hence, 'Exercise' and 'General Health' interact significantly to explain BMI
variance."
```

```
> if (summary(linear_model)$coefficients["Fruit_Consumption", "Pr(>|t|)"] < significance_level) {
+   cat("Reject the null hypothesis. There is a substantial link between mean BMI and fruit consumption.\n")
+ } else {
+   cat("Accept the null hypothesis. There is no significant relationship between mean BMI and levels of fruit consumptio
n.\n")
+ }
Reject the null hypothesis. There is a substantial link between mean BMI and fruit consumption.
```

```
> if (summary(linear_model_two)$coefficients["Green_Vegetables_Consumption", "Pr(>|t|)"] < significance_level) {
+   cat("Reject the null hypothesis. There is a substantial relationship between BMI and green vegetable consumption.\n")
+ } else {
+   cat("Accept the null hypothesis.There is no significant relationship between BMI and green vegetable consumption.\n")
+ }
Reject the null hypothesis. There is a substantial relationship between BMI and green vegetable consumption.
```

Note. Based on the hypothesis test, we obtained the following test results: exercise and general health interact significantly to explain BMI variance; second, there is an important correlation between mean BMI and fruit consumption; and third, there is a significant link between BMI and vegetable consumption.

To begin with model development, it is helpful to understand BMI. Body mass index (BMI) is a good measure of your risk for developing certain diseases that are linked to increased body fat levels. It also acts as a stand-in for body fat (*Assessing Your Weight and Health Risk*, n.d.).

After completing a few hypothesis tests and reviewing a few insights, we acquired several ideas on how specific features are affecting the BMI of individuals and moving forward with the research. We developed a a linear regression model that is capable of considerably predicting an individual's BMI based on various promising input features.

Prior to beginning the model's development, a few categorical features need to be carefully considered. Specifically, features with yes and no categories should have their values converted to 1 and 0, respectively, and features with more than two categories are handled using one-hot encoding.

**Model 1**

The first model is built around a few noteworthy features that may lead to the prediction of BMI. As this is the first model, the data it contains is unprocessed. Based on the overall idea, we later deal with outliers, transformation techniques, and selection techniques to obtain the most promising model.

**Figure 2**

*First Model Summary*

```
Call:
lm(formula = BMI ~ Fruit_Consumption + Green_Vegetables_Consumption +
    Alcohol_Consumption + FriedPotato_Consumption + Exercise_conv +
    Smoking_History_conv + General_HealthExcellent + General_HealthFair +
    General_HealthGood + General_HealthPoor + General_HealthVery_Good,
    data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-20.036  -4.070  -0.798   3.095  70.987

Coefficients: (1 not defined because of singularities)
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                 29.6777069  0.0363365  816.75   <2e-16 ***
Fruit_Consumption           -0.0095828  0.0004751  -20.17   <2e-16 ***
Green_Vegetables_Consumption -0.0086061  0.0007892  -10.90   <2e-16 ***
Alcohol_Consumption         -0.0592277  0.0013977  -42.37   <2e-16 ***
FriedPotato_Consumption      0.0283528  0.0013141   21.57   <2e-16 ***
Exercise_conv               -1.3746002  0.0284851  -48.26   <2e-16 ***
Smoking_History_conv        -0.2077693  0.0235033   -8.84   <2e-16 ***
General_HealthExcellent     -1.6185577  0.0325261  -49.76   <2e-16 ***
General_HealthFair           2.7255012  0.0391148   69.68   <2e-16 ***
General_HealthGood           1.7262956  0.0279291   61.81   <2e-16 ***
General_HealthPoor           2.2251723  0.0633356   35.13   <2e-16 ***
General_HealthVery_Good             NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.247 on 308843 degrees of freedom
Multiple R-squared:  0.08276,   Adjusted R-squared:  0.08273
F-statistic:  2787 on 10 and 308843 DF,  p-value: < 2.2e-16
```

Note. The summary of the linear regression model indicates that a number of lifestyle factors, including the consumption of fruit, vegetables, alcohol, exercise routines, and others, have a substantial impact on body mass index (BMI). Notably, consuming more fruits and vegetables is linked to a lower BMI, whereas consuming fried potatoes and alcohol is linked to a higher BMI. Furthermore, BMI readings are generally lower in those who report exercising, having smoked in the past, and having excellent or fair general health. On the other hand, people with worse overall health typically have higher BMIs. It's crucial to remember that singularities prevent the coefficient estimate for the "General_HealthVery_Good" category from being defined. Overall, the adjusted R-squared value shows that the model explains around 8.3% of the variability in BMI, which is not very satisfactory and needs improvement.

**Model 1 Extended Version**

**Figure 3**

*First Model Extended Version  Summary*

```
Call:
lm(formula = BMI ~ Fruit_Consumption + Green_Vegetables_Consumption +
    Height_.cm. + Weight_.kg. + Diabetes_conv + Alcohol_Consumption +
    FriedPotato_Consumption + Exercise_conv + Smoking_History_conv +
    General_HealthExcellent + General_HealthFair + General_HealthGood +
    General_HealthPoor + General_HealthVery_Good, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-13.628  -0.319   0.046   0.334  42.500

Coefficients: (1 not defined because of singularities)
                              Estimate Std. Error   t value Pr(>|t|)
(Intercept)                  5.813e+01  2.779e-02  2091.770  < 2e-16 ***
Fruit_Consumption            1.352e-04  6.887e-05     1.963 0.049639 *
Green_Vegetables_Consumption -8.519e-04  1.144e-04    -7.450 9.37e-14 ***
Height_.cm.                  -3.407e-01  1.801e-04 -1891.721  < 2e-16 ***
Weight_.kg.                   3.427e-01  9.109e-05  3762.672  < 2e-16 ***
Diabetes_conv                 5.386e-02  5.110e-03    10.540  < 2e-16 ***
Alcohol_Consumption           1.886e-04  2.047e-04     0.921 0.356857
FriedPotato_Consumption       6.118e-04  1.915e-04     3.194 0.001402 **
Exercise_conv                -4.203e-02  4.159e-03   -10.105  < 2e-16 ***
Smoking_History_conv         -4.085e-03  3.407e-03    -1.199 0.230583
General_HealthExcellent       2.321e-02  4.731e-03     4.905 9.34e-07 ***
General_HealthFair            4.301e-02  5.783e-03     7.437 1.03e-13 ***
General_HealthGood            1.257e-02  4.085e-03     3.078 0.002081 **
General_HealthPoor            3.235e-02  9.268e-03     3.491 0.000481 ***
General_HealthVery_Good              NA         NA        NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9042 on 308840 degrees of freedom
Multiple R-squared:  0.9808,    Adjusted R-squared:  0.9808
F-statistic: 1.213e+06 on 13 and 308840 DF,  p-value: < 2.2e-16
```

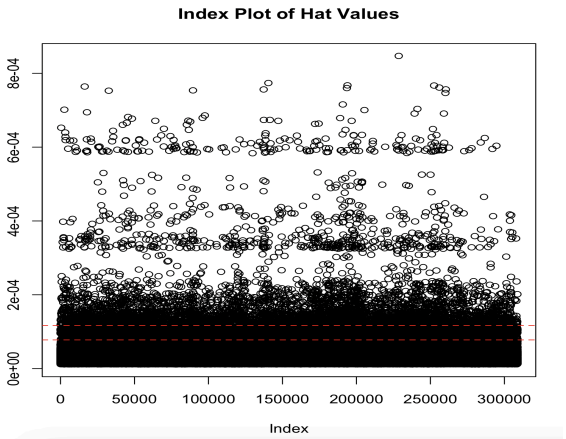
**Note.** As the adjusted R squared score above is not very impressive, a few additional parameters like height, weight, and the existence of diabetes in each individual are specified to create the mode excellent fit. Surprisingly, the adjusted R-squared score shows a huge improvement from 8.9% straight to 98%.

Handling Outliers

**Figure 4**

*Handling Outliers found in First Model Extended Version*

```
> #---------- Checking for Model Outliers ----------
> outlierTest(model = firstModel_extended)
       rstudent unadjusted p-value Bonferroni p
30024  46.60740         0.0000e+00   0.0000e+00
53677  46.59307         0.0000e+00   0.0000e+00
195511 43.86087         0.0000e+00   0.0000e+00
208800 41.42159         0.0000e+00   0.0000e+00
253619 45.04399         0.0000e+00   0.0000e+00
255796 47.18127         0.0000e+00   0.0000e+00
292794 41.04286         0.0000e+00   0.0000e+00
194794 35.60024        5.0903e-277  1.5722e-271
236327 35.26051        8.2638e-272  2.5523e-266
288706 34.97446        1.8450e-267  5.6983e-262
```




Note. A few outliers that must be fixed in order to optimise the model effectively were found using the outlierTest() function and an index plot of Hat values. In order to deal with outliers, we decided to eliminate them. A few clear outliers can be observed in the images above, also the OutliersTest() method tells significantly which of the possible outliers are present.

**Model 2 (After addressing Outliers)**

Following the removal of the outliers, Model 2 was created using a dataset free of outliers.

**Figure 5**

*Second Model Summary*

```
Call:
lm(formula = BMI ~ Fruit_Consumption + Green_Vegetables_Consumption +
    Height_.cm. + Weight_.kg. + Diabetes_conv + Alcohol_Consumption +
    FriedPotato_Consumption + Exercise_conv + Smoking_History_conv +
    General_HealthExcellent + General_HealthFair + General_HealthGood +
    General_HealthPoor + General_HealthVery_Good, data = outlier_free_dataset)

Residuals:
     Min       1Q   Median       3Q      Max
-2.22850 -0.27197  0.04172  0.29522  2.30555

Coefficients: (1 not defined because of singularities)
                             Estimate Std. Error   t value Pr(>|t|)
(Intercept)                 5.608e+01  1.904e-02  2945.877  < 2e-16 ***
Fruit_Consumption           1.995e-04  4.445e-05     4.488 7.19e-06 ***
Green_Vegetables_Consumption -5.763e-04  7.486e-05    -7.698 1.39e-14 ***
Height_.cm.                -3.288e-01  1.252e-04 -2626.108  < 2e-16 ***
Weight_.kg.                 3.430e-01  6.813e-05  5035.164  < 2e-16 ***
Diabetes_conv               3.248e-02  3.346e-03     9.706  < 2e-16 ***
Alcohol_Consumption        -2.179e-04  1.310e-04    -1.663   0.0963 .
FriedPotato_Consumption     5.416e-04  1.316e-04     4.116 3.86e-05 ***
Exercise_conv              -4.771e-02  2.700e-03   -17.670  < 2e-16 ***
Smoking_History_conv       -2.730e-03  2.185e-03    -1.249   0.2115
General_HealthExcellent     1.379e-02  2.997e-03     4.602 4.19e-06 ***
General_HealthFair          4.802e-02  3.773e-03    12.729  < 2e-16 ***
General_HealthGood          1.720e-02  2.601e-03     6.615 3.72e-11 ***
General_HealthPoor          5.227e-02  6.403e-03     8.162 3.30e-16 ***
General_HealthVery_Good            NA         NA        NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5625 on 291446 degrees of freedom
Multiple R-squared:  0.9897,    Adjusted R-squared:  0.9897
F-statistic: 2.145e+06 on 13 and 291446 DF,  p-value: < 2.2e-16
```

Note. Here, we can see that the modified R-square score has increased somewhat following the removal of the outliers. It was previously 98%, but is now nearly 99% (98.97%), indicating a minor improvement in the model's goodness of fit. Even so, as shown by the NA values for the feature General_HealthVery_Good, multicollinearity is still a problem.

**Model 3 (Using Transformation approach)**

Additionally to ensure that the model is free from problems with heteroscedasticity and non-normality of residuals. Using the sqrt() function, we applied a transformation approach to the BMI values.

**Figure 6**

*Third Model Summary*

```
Call:
lm(formula = BMI_sqrt ~ Fruit_Consumption + Green_Vegetables_Consumption +
    Alcohol_Consumption + FriedPotato_Consumption + Height_.cm. +
    Weight_.kg. + Diabetes_conv + Exercise_conv + Smoking_History_conv +
    General_HealthExcellent + General_HealthFair + General_HealthGood +
    General_HealthPoor + General_HealthVery_Good, data = outlier_free_dataset)

Residuals:
     Min       1Q   Median       3Q      Max
-1.35050 -0.02478  0.01716  0.03765  0.18808

Coefficients: (1 not defined because of singularities)
                              Estimate Std. Error   t value Pr(>|t|)
(Intercept)                  7.808e+00  2.077e-03  3759.057  < 2e-16 ***
Fruit_Consumption            6.418e-06  4.850e-06     1.323 0.185720
Green_Vegetables_Consumption -2.062e-05  8.167e-06    -2.525 0.011569 *
Alcohol_Consumption           4.168e-05  1.429e-05     2.916 0.003546 **
FriedPotato_Consumption       5.157e-05  1.436e-05     3.592 0.000328 ***
Height_.cm.                  -2.997e-02  1.366e-05 -2194.066  < 2e-16 ***
Weight_.kg.                   3.150e-02  7.433e-06  4238.260  < 2e-16 ***
Diabetes_conv                 6.350e-03  3.651e-04    17.393  < 2e-16 ***
Exercise_conv                -2.848e-03  2.946e-04    -9.666  < 2e-16 ***
Smoking_History_conv          7.736e-04  2.384e-04     3.245 0.001175 **
General_HealthExcellent      -2.763e-03  3.270e-04    -8.448  < 2e-16 ***
General_HealthFair           -1.663e-03  4.116e-04    -4.041 5.32e-05 ***
General_HealthGood           -1.664e-05  2.837e-04    -0.059 0.953242
General_HealthPoor           -2.074e-03  6.986e-04    -2.969 0.002991 **
General_HealthVery_Good              NA         NA        NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.06137 on 291446 degrees of freedom
Multiple R-squared:  0.9855,    Adjusted R-squared:  0.9855
F-statistic: 1.519e+06 on 13 and 291446 DF,  p-value: < 2.2e-16
```

Note. There are few changes in estimates found when the BMI value is transformed using the sqrt() function. For instance, in Model 2, it was shown that persons with excellent general health had high BMIs, but, in this model, people with excellent general health appear to have low BMIs, and the same is for those with fair, good,

or poor health status. Additionally, the Adjusted R square score decreased somewhat, going from roughly 99% (98.97%) to 98.55%.

**Model 4 (Using Stepwise Selection Approach)**

Now that we have identified the model with a significantly higher adjusted R square score, it is time to simplify the model and address the multicollinearity issue. To do this, we need to identify a small number of features that will address the mentioned problem and not in any way compromise the model's accuracy. To find these features, we will use the feature selection technique, more specifically the stepwise selection method.

**Figure 7**

*List of Suggested Features by Setpwise Model and Model Summary Along with Multicollinearity Test*

```
> selected_features_stepwise <- names(coef(step_wise_selection))
>
> cat("Best Features based on stepwise selection:", selected_features_stepwise, "\n")
Best Features based on stepwise selection: (Intercept) Green_Vegetables_Consumption Alcohol_Consumption FriedPotato_Consumption Height
_.cm. Weight_.kg. Diabetes_conv Exercise_conv Smoking_History_conv General_HealthExcellent General_HealthFair General_HealthPoor
```

```
Call:
lm(formula = BMI_sqrt ~ Green_Vegetables_Consumption + Alcohol_Consumption +
    FriedPotato_Consumption + Height_.cm. + Weight_.kg. + Diabetes_conv +
    Exercise_conv + Smoking_History_conv + General_HealthExcellent +
    General_HealthFair + General_HealthPoor, data = outlier_free_dataset)

Residuals:
     Min       1Q   Median       3Q      Max
-1.35040 -0.02478  0.01718  0.03764  0.18796

Coefficients:
                             Estimate Std. Error   t value Pr(>|t|)
(Intercept)                 7.808e+00  2.056e-03  3797.330  < 2e-16 ***
Green_Vegetables_Consumption -1.789e-05  7.902e-06    -2.264 0.023593 *
Alcohol_Consumption          4.113e-05  1.427e-05     2.882 0.003955 **
FriedPotato_Consumption      5.048e-05  1.433e-05     3.522 0.000428 ***
Height_.cm.                 -2.997e-02  1.363e-05 -2199.655  < 2e-16 ***
Weight_.kg.                  3.150e-02  7.391e-06  4262.482  < 2e-16 ***
Diabetes_conv                6.354e-03  3.635e-04    17.477  < 2e-16 ***
Exercise_conv               -2.810e-03  2.923e-04    -9.611  < 2e-16 ***
Smoking_History_conv         7.515e-04  2.374e-04     3.166 0.001546 **
General_HealthExcellent     -2.744e-03  3.055e-04    -8.981  < 2e-16 ***
General_HealthFair          -1.661e-03  3.838e-04    -4.328 1.51e-05 ***
General_HealthPoor          -2.064e-03  6.811e-04    -3.031 0.002438 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.06137 on 291448 degrees of freedom
Multiple R-squared:  0.9855,    Adjusted R-squared:  0.9855
F-statistic: 1.796e+06 on 11 and 291448 DF,  p-value: < 2.2e-16
```

```
> vif(FourthModel)
Green_Vegetables_Consumption          Alcohol_Consumption      FriedPotato_Consumption                 Height_.cm.
                    1.030304                     1.060121                     1.018205                     1.436922
                Weight_.kg.                Diabetes_conv               Exercise_conv         Smoking_History_conv
                    1.450442                     1.097578                     1.113208                     1.049013
    General_HealthExcellent           General_HealthFair           General_HealthPoor
                    1.091397                     1.102426                     1.064374
>
```

Note. By employing the stepwise selection technique, we were able to identify the features that not only minimize the complexity of the model but also address the multicollinearity issue, which is confirmed by using the vif() function. As can be seen above, each feature's score is approximately 1, indicating that there is no multicollinearity issue. Furthermore, the adjusted r-square score remained consistent at 98.55%, meaning that the model's accuracy was not compromised.

**Model Comparison**

Up till now, five models have been examined in order to determine which is the most promising. The AIC score has been used to select the top model from the group. AIC establishes the ideal balance between the goodness of fit and complexity of the model. Here, the model with the lowest AIC would be preferred.

**Figure 8**

*Model Perfromance Comparision Using AIC*

```
> AIC(firstModel)
[1] 2008175
> AIC(firstModel_extended)
[1] 814267.7
> AIC(secondModel)
[1] 491775.5
> AIC(ThirdModel)
[1] -799660.4
> AIC(FourthModel)
[1] -799662.7
```

Note. The fourth model, created through the stepwise selection approach, had the lowest AIC score. As we can see from the results above, it also provides a significantly comparable R-squared score like the other models, requires fewer features to accomplish the same score by reducing complexity, also resolves multicollinearity issues.

**Model Evaluation**

Since we ultimately settled on the fourth model, it is necessary to evaluate it in order to gain a general understanding of the model's performance.

**Figure 9**

*Model Evaluation (of Fourth Model)*

```
> cat("Mean Absolute Error (MAE):", mae, "\n")
Mean Absolute Error (MAE): 0.04517708
> cat("Root Mean Squared Error (RMSE):", rmse, "\n")
Root Mean Squared Error (RMSE): 0.06137176
> cat("R-squared:", rsquared, "\n")
R-squared: 0.9854588
```

Note. The evaluation metrics show that the model performs well. The model's average prediction error (MAE) is 0.045, meaning that it differs from the actual values by about 0.045 units. With a Root Mean Squared Error (RMSE) of 0.061, the model's predictions are expected to have an average error of roughly 0.061 units. Furthermore, a strong fit to the data is indicated by the high R-squared value of 0.985, which shows that the model explains around 98.5% of the variance in the target variable.

**Model development utilising the LASSO and Ridge method**

**Figure 10**

*LASSO Model Summary and Its Performance Metrics*

```
> coef(fifth_model_lasso)
12 x 1 sparse Matrix of class "dgCMatrix"
                                        s0
(Intercept)                   7.7699009532
Green_Vegetables_Consumption  .
Alcohol_Consumption           .
FriedPotato_Consumption       .
Height_.cm.                  -0.0296939787
Weight_.kg.                   0.0313664078
Diabetes_conv                 0.0033665937
Exercise_conv                -0.0005791687
Smoking_History_conv          .
General_HealthExcellent      -0.0006226354
General_HealthFair            .
General_HealthPoor            .
```

```
> print("Lasso Model:")
[1] "Lasso Model:"
> print(paste("  MAE:", lasso_mae))
[1] "  MAE: 0.0451699253691863"
> print(paste("  RMSE:", lasso_rmse))
[1] "  RMSE: 0.0614714344943438"
> print(paste("  R-squared:", lasso_r_squared))
[1] "  R-squared: 0.985411563018523"
```

Note. We created the fifth model using the LASSO method, utilizing the predictors from the fourth model, which was the best of all. LASSO reduces the number of features in the model, which further reduces its complexity. Additionally, the pair of features the lasso model suggests closely matches the performance score provided by the fourth model, so it makes sense to choose this fifth model, which is built using the lasso method, as it offers a less complex model with comparable performance.

**Figure 11**

*RIDGE Model Summary and Its Performance Metrics*

```
> coef(sixth_model_ridge)
12 x 1 sparse Matrix of class "dgCMatrix"
                                       s0
(Intercept)                  7.123867e+00
Green_Vegetables_Consumption -9.740141e-05
Alcohol_Consumption          -7.594844e-04
FriedPotato_Consumption       9.663966e-05
Height_.cm.                  -2.383884e-02
Weight_.kg.                   2.732450e-02
Diabetes_conv                 3.040280e-02
Exercise_conv                -1.954464e-02
Smoking_History_conv         -1.493229e-03
General_HealthExcellent      -2.731086e-02
General_HealthFair            1.038082e-02
General_HealthPoor           -2.709502e-03
```

```
[1] "Ridge Model:"
> print(paste("  MAE:", ridge_model_mae))
[1] "  MAE: 0.0680789364571338"
> print(paste("  RMSE:", ridge_model_rmse))
[1] "  RMSE: 0.0912425522158195"
> print(paste("  R-squared:", ridge_model_r_squared))
[1] "  R-squared: 0.967859182607646"
```

Note. The model produced with the ridge method can be seen in the above image, it performs less well than the model produced by Lasso. Notably, even when compared to Ridge, where features are not removed but rather have their coefficients decreased in order to optimize the model, Lasso offers a model with fewer features but better performance.

Therefore, in order to predict the individual's BMI, we choose to utilise the LASSO model, which provides the same results as model 4 but is less complex because it uses less features to provide the same results.

**Question 2: Can we build a predictive model using logistic regression to assess the risk of cardiovascular diseases based on various health indicators and lifestyle factors? Specifically, we want to understand the influence of exercise, cancer history, mental health (depression), diabetes, arthritis, and other factors on the likelihood of developing heart disease.**

Data Preprocessing

In the exploratory data analysis (EDA) of our dataset, we observed that the target variable 'Heart_Disease' is imbalanced. One class of "Yes" for heart disease is more prevalent than "No" for no heart disease. Imbalanced datasets lead to biased models where they predict accurately for the majority class ("No" in our dataset) compared to the minority class ("Yes" in our dataset).

Balancing the Dataset

```
data_yes <- subset(data, Heart_Disease == "Yes")
data_no <- subset(data, Heart_Disease == "No")

# Sample 10,000 rows from each subset
data_yes_sampled <- data_yes[sample(nrow(data_yes), 10000), ]
data_no_sampled <- data_no[sample(nrow(data_no), 10000), ]

# Combine the sampled subsets
balanced_data <- rbind(data_yes_sampled, data_no_sampled)

# Shuffle the rows to randomize the order
balanced_data <- balanced_data[sample(nrow(balanced_data)), ]

data <- balanced_data
```

Based on the target variable Heart_Disease, the code subsets the data into two datasets based on the output of the variable Heart_Disease.

'data_yes' contains rows with an output of Heart_Disease as 'Yes'.

'data_no' contains rows with an output of Heart_Disease as 'No'.

10,000 sample rows from each subset are combined into a single dataset which will be used for modeling tasks.

Creating Logistic regression model

**Model 1**

It is necessary to convert the categorical variables before starting the model's construction. In particular, one-hot encoding is used for features with more than two categories, and features with yes and no categories should have their values changed to binary(1 and 0), respectively.

```
logistic_model_heart <- glm(Heart_Disease ~ ., data = train_data, family = binomial)
summary(logistic_model_heart)

coef(logistic_model_heart)
exp(coef(logistic_model_heart))
```

We constructed a logistic regression model using 'Heart_Disease' as the response variable and all other variables as predictor variables.

Then, we extracted the odd ratios of the coefficients. The ratios represent the change in odds of the response variable with one unit change in the predictor variables.

**Figure 12**

*Odd ratios of the logistic regression model*

```
                        NA                        NA                        NA
> exp(coef(logistic_model_heart))
                (Intercept)                Height_.cm.                Weight_.kg.                        BMI
                 0.00116541                 1.03957840                 0.98120328                 1.05928435
        Alcohol_Consumption           Fruit_Consumption Green_Vegetables_Consumption     FriedPotato_Consumption
                 0.99247190                 0.99766307                 1.00151439                 1.00081415
             Exercise_conv       Smoking_History_conv             Arthritis_conv           Skin_Cancer_conv
                 1.00802230                 1.59936989                 1.30142506                 1.09770007
           Other_Cancer_conv                No_Diabetes               Pre_Diabetes               Yes_Diabetes
                 1.10294021                 1.37703612                 1.92272213                 2.59621376
         Pregnancy_Diabetes     General_HealthExcellent          General_HealthFair          General_HealthGood
                         NA                 0.55638174                 3.72174966                 1.80894899
         General_HealthPoor     General_HealthVery_Good          `Age_Category18-24`        `Age_Category25-29`
                 6.21650207                         NA                 0.03466110                 0.05369670
        `Age_Category30-34`        `Age_Category35-39`        `Age_Category40-44`        `Age_Category45-49`
                 0.07162388                 0.06238887                 0.08766752                 0.12495181
        `Age_Category50-54`        `Age_Category55-59`        `Age_Category60-64`        `Age_Category65-69`
                 0.16007404                 0.27317912                 0.30393819                 0.40890250
        `Age_Category70-74`        `Age_Category75-79`          `Age_Category80+`        `More than 5 years`
                 0.53011587                 0.73876729                         NA                 0.56649819
                      Never             `Past 2 years`             `PAst 5 years`             `Past 1 year`
                 1.26448532                 0.73699619                 0.55200585                         NA
```

The odd ratios greater than 1 represent the positive relationship between the predictor variable and the likelihood of having heart disease, whereas values less than1 represent the negative relationship for the same.

```
#Confusion matrix for train data
train_predictions <- predict(logistic_model_heart, train_data, type = "response")
train_predictions

train_predictions_binary <- as.factor(ifelse(train_predictions > 0.5, "Yes", "No"))
train_predictions_binary

head(train_predictions_binary)

confusionMatrix(train_predictions_binary, train_data$Heart_Disease, positive = "Yes")
```

We will check the performance metrics of the logistic regression model by creating a confusion matrix on the training data.

It contains the following elements:

True positive (TP): Predicted ‘Yes’ for heart disease correctly.

False positive (FP): Predicted ‘No’ for heart disease correctly.

True negative (TN): Incorrect prediction for having a heart disease.

False negative (FN): Incorrect prediction for not having a heart disease.

**Figure 13**

*Confusion matrix for train data*

```
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  5056 1545
       Yes 1944 5455

               Accuracy : 0.7508
                 95% CI : (0.7435, 0.7579)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5016

 Mcnemar's Test P-Value : 1.606e-11

            Sensitivity : 0.7793
            Specificity : 0.7223
         Pos Pred Value : 0.7373
         Neg Pred Value : 0.7659
             Prevalence : 0.5000
         Detection Rate : 0.3896
   Detection Prevalence : 0.5285
      Balanced Accuracy : 0.7508

       'Positive' Class : Yes
```

Accuracy: 0.7508, the model's overall performance is 75.08%, which is the proportion of correct predictions made out of all predictions.

Precision (Positive Pred value): 0.7373, the proportion of correctly predicted positives out of all instances predicted as positive. 73.73% of the cases predicted positive (having heart disease) were actually positive.

Recall (Sensitivity): 0.7793, the proportion of predicted positives out of all instances actually positive. 77.93% of the actual positive cases were correctly identified by the model.

Specificity: 0.7223, the proportion of correctly predicted negatives out of all negatives.

To gain insights into the performance metrics of the logistics regression model, we apply it to unseen data 'test_data'.

```
#Confusion matrix for test data
test_predictions <- predict(logistic_model_heart, test_data, type = "response")
test_predictions

test_predictions_binary <- as.factor(ifelse(test_predictions > 0.5, "Yes", "No"))
test_predictions_binary

head(test_predictions_binary)

confusionMatrix(test_predictions_binary, test_data$Heart_Disease, positive = "Yes")
```

**Figure 14**

*Confusion matrix for test data*

```
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  2183  655
       Yes  817 2345

               Accuracy : 0.7547
                 95% CI : (0.7436, 0.7655)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5093

 Mcnemar's Test P-Value : 2.712e-05

            Sensitivity : 0.7817
            Specificity : 0.7277
         Pos Pred Value : 0.7416
         Neg Pred Value : 0.7692
             Prevalence : 0.5000
         Detection Rate : 0.3908
   Detection Prevalence : 0.5270
      Balanced Accuracy : 0.7547

       'Positive' Class : Yes
```

The performance metrics of accuracy, recall, specificity etc. are similar between training and test data, which signifies that the model generalizes well to unseen data as well.

**Model 2 (Using stepwise approach)**

To further refine and simplify the model, we will use stepwise feature selection method, which performs the feature selection based on the AIC value.

```
#Model 2 (stepwise feature selection approach)
step_wise_selection <- step(logistic_model_heart, direction = "both", trace=TRUE)
selected_features_stepwise <- names(coef(step_wise_selection))

cat("Best Features based on stepwise selection:", selected_features_stepwise, "\n")

logistic_model_refined <- glm(Heart_Disease ~ Height_.cm. + Weight_.kg. + BMI + Alcohol_Consumption + Fruit_Consumption +
                              Smoking_History_conv + Arthritis_conv + Skin_Cancer_conv + Other_Cancer_conv +
                              Pre_Diabetes + Yes_Diabetes + General_HealthExcellent + General_HealthFair +
                              General_HealthGood + General_HealthPoor + `Age_Category18-24` + `Age_Category25-29` +
                              `Age_Category30-34` + `Age_Category35-39` + `Age_Category40-44` + `Age_Category45-49` +
                              `Age_Category50-54` + `Age_Category55-59` + `Age_Category60-64` + `Age_Category65-69` +
                              `Age_Category70-74` + `Age_Category75-79` + `More than 5 years` + `Past 2 years` + `PAst 5 years`, data = train_data, family = binomial)

summary(logistic_model_refined)
```

```
Step:  AIC=14254.84
Heart_Disease ~ Height_.cm. + Weight_.kg. + BMI + Alcohol_Consumption +
    Fruit_Consumption + Smoking_History_conv + Arthritis_conv +
    Skin_Cancer_conv + Other_Cancer_conv + Pre_Diabetes + Yes_Diabetes +
    General_HealthExcellent + General_HealthFair + General_HealthGood +
    General_HealthPoor + `Age_Category18-24` + `Age_Category25-29` +
    `Age_Category30-34` + `Age_Category35-39` + `Age_Category40-44` +
    `Age_Category45-49` + `Age_Category50-54` + `Age_Category55-59` +
    `Age_Category60-64` + `Age_Category65-69` + `Age_Category70-74` +
    `Age_Category75-79` + `More than 5 years` + `Past 2 years` +
    `PAst 5 years`
```

The following function performs both forward and backward variable selection technique to keep the most relevant variables required to predict whether a person has a heart disease or not. This technique also handles the multicollinearity issue in the model.

**Figure 15**

*Confusion matrix for train data after stepwise feature selection*

```
> #Confusion matrix for train data
> train_predictions <- predict(logistic_model_refined, train_data, type = "response")
> train_predictions_binary <- as.factor(ifelse(train_predictions > 0.5, "Yes", "No"))
> confusionMatrix(train_predictions_binary, train$Heart_Disease, positive = "Yes")
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  5065 1546
       Yes 1935 5454

               Accuracy : 0.7514
                 95% CI : (0.7441, 0.7585)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5027

 Mcnemar's Test P-Value : 4.824e-11

            Sensitivity : 0.7791
            Specificity : 0.7236
         Pos Pred Value : 0.7381
         Neg Pred Value : 0.7661
             Prevalence : 0.5000
         Detection Rate : 0.3896
   Detection Prevalence : 0.5278
      Balanced Accuracy : 0.7514

       'Positive' Class : Yes
```

Here, we observe that the performance metrics between models before and after stepwise feature selection are quite similar.

Therefore, we have successfully reduced the complexity of the model (number of predictor variables are

reduced to half compared to the first model) without compromising on the performance metrics such as accuracy, sensitivity, specificity etc.

**Figure 16**

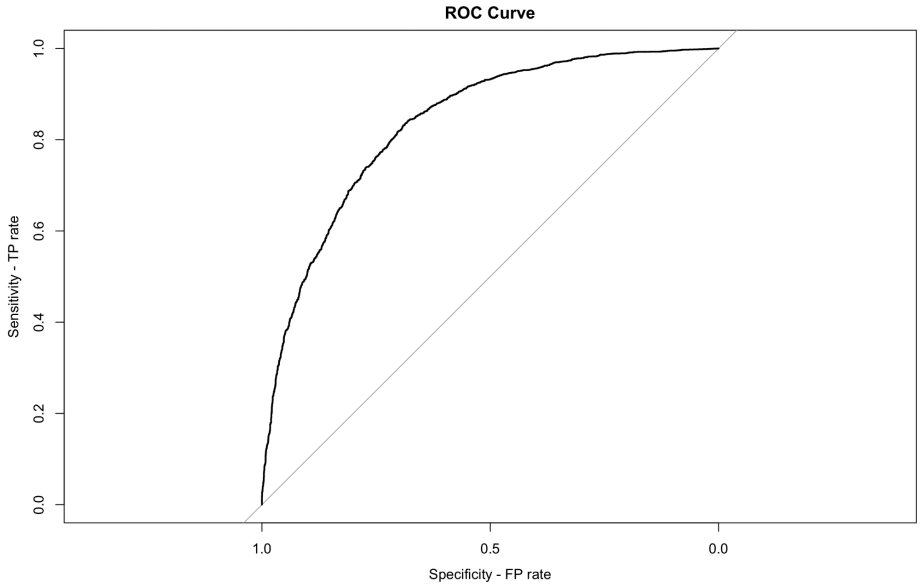
*Confusion matrix for train data after stepwise feature selection*

```
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  2184  654
       Yes  816 2346

               Accuracy : 0.755
                 95% CI : (0.7439, 0.7658)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.51

 Mcnemar's Test P-Value : 2.679e-05

            Sensitivity : 0.7820
            Specificity : 0.7280
         Pos Pred Value : 0.7419
         Neg Pred Value : 0.7696
             Prevalence : 0.5000
         Detection Rate : 0.3910
   Detection Prevalence : 0.5270
      Balanced Accuracy : 0.7550

       'Positive' Class : Yes
```

Similarly, we applied the refined model on unseen ‘test_data’ to check the robustness of the model. Based on the provided test data, the logistic regression model performs well in predicting heart disease after being refined by stepwise feature selection.

**Figure 17**

*ROC curve*




Using the actual and predicted probabilities of the response variable 'Heart_Disease', we plotted a ROC curve that represents the 'true positive rate' against the 'false positive rate' at different threshold values.

The diagonal line represents random guessing and the less curve hugging the diagonal line, the better the model's performance.

Further, we calculated the Area under the curve (AUC) which represents the model's discrimination ability.

The values close to 1 represent the model's better ability to classify whereas values close to 0.5 represent that the model performs no better than random guessing.

**Figure 18**

*Area Under the curve*

```
> #AUC
> auc <- auc(ROC1)
> auc
Area under the curve: 0.836
```

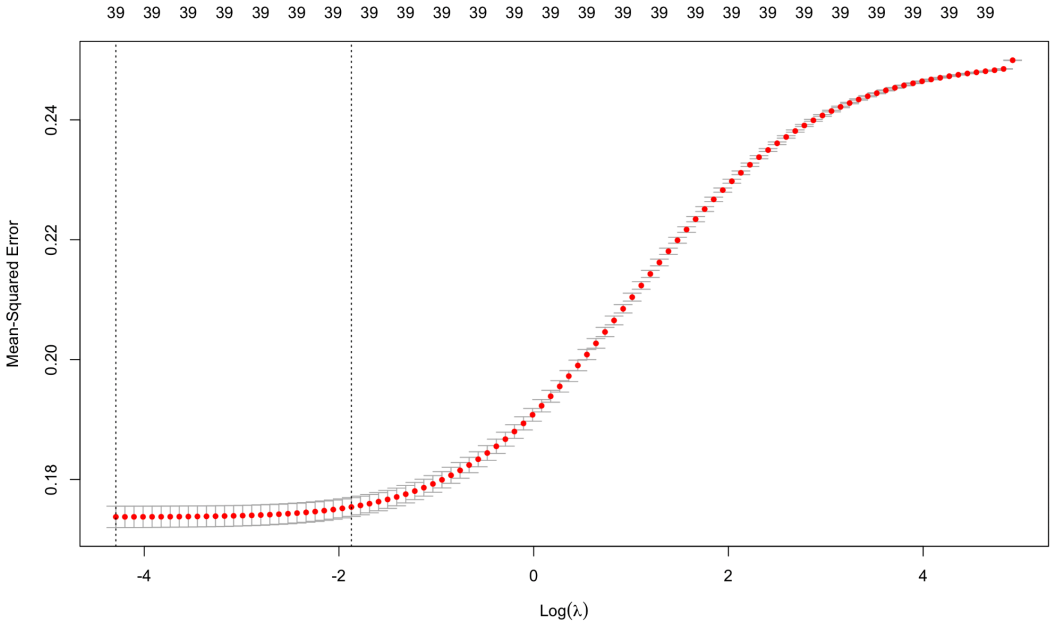
For our model, an AUC of 0.836 represents that the model performs well in distinguishing between individuals having heart disease and those not.

**Model 3 (Ridge regression)**

```
set.seed(123)
cv.ridge <- cv.glmnet(train_x, train_y, alpha = 0, nfolds = 10)
plot(cv.ridge)
```

**Figure 19**

*Plot of the Regression regularization*




The cv.glmnet() method uses cross-validation to determine the optimal λ for a prediction model. The graph shows the mean square error on the y-axis and the natural log of lambda on the x-axis. The amount of variables considered at each point is shown across the top. The lambda with the lowest mean square error is the dotted line closest to the y-axis. The second dotted line represents the lambda, which is within one standard error of the minimum.

```
> log(cv.ridge$lambda.min)
[1] -4.291191
> log(cv.ridge$lambda.1se)
[1] -1.872314
```

The log of lambda value selected by cross-validation that minimizes the mean cross-validated error is -4.291191.

The log of lambda value selected by cross-validation that is within one standard error of the minimum is -1.872314.

**Figure 20**

*Coefficients after Ridge regularization at lambda value that minimizes the mean cross-validated error*

```
> ridge.model.min <- glmnet(train_x, train_y, alpha = 0, lambda = cv.ridge$lambda.min)
> coef(ridge.model.min)
40 x 1 sparse Matrix of class "dgCMatrix"
                                        s0
(Intercept)                  -6.074107e-02
Height_.cm.                   2.910720e-03
Weight_.kg.                   5.985170e-05
BMI                           8.305373e-04
Alcohol_Consumption          -1.548190e-03
Fruit_Consumption             8.492926e-06
Green_Vegetables_Consumption -8.002099e-05
FriedPotato_Consumption       6.691886e-04
Exercise_conv                -1.540525e-02
Smoking_History_conv          7.670158e-02
Arthritis_conv                5.275493e-02
Skin_Cancer_conv              1.965440e-02
Other_Cancer_conv             1.257486e-02
No_Diabetes                  -6.474897e-02
Pre_Diabetes                  1.281008e-02
Yes_Diabetes                  6.338827e-02
Pregnancy_Diabetes           -1.109019e-01
General_HealthExcellent      -1.601638e-01
General_HealthFair            1.207687e-01
General_HealthGood            6.508329e-03
General_HealthPoor            2.204795e-01
General_HealthVery_Good      -9.612920e-02
`Age_Category18-24`          -2.745652e-01
`Age_Category25-29`          -2.633340e-01
`Age_Category30-34`          -2.543862e-01
`Age_Category35-39`          -2.626801e-01
`Age_Category40-44`          -2.313245e-01
`Age_Category45-49`          -1.710193e-01
`Age_Category50-54`          -1.189968e-01
`Age_Category55-59`          -4.330887e-02
`Age_Category60-64`          -2.703187e-02
`Age_Category65-69`           6.697250e-02
`Age_Category70-74`           1.085318e-01
`Age_Category75-79`           1.488104e-01
`Age_Category80+`             2.184860e-01
`More than 5 years`          -3.254278e-02
Never                        -8.243403e-02
`Past 2 years`               -2.702768e-02
`PAst 5 years`               -4.836257e-02
`Past 1 year`                 3.347156e-02
```

The lambda value that minimizes the cross validation error was used to fit the Ridge regression model.

The predicted impact of each predictor variable on the response variable (Heart_Disease) is shown by the model coefficients.

When other predictors are held constant, the coefficients show how the response variable changes in response to a one-unit change in the predictor.

**Figure 21**

*Coefficients after Ridge regularization at lambda within one standard error of the minimum*

```
> ridge.model.1se <- glmnet(train_x, train_y, alpha = 0, lambda = cv.ridge$lambda.1se)
> coef(ridge.model.1se)
40 x 1 sparse Matrix of class "dgCMatrix"
                                        s0
(Intercept)                   1.118045e-01
Height_.cm.                   1.836871e-03
Weight_.kg.                   3.696836e-04
BMI                          -2.756251e-05
Alcohol_Consumption          -1.233225e-03
Fruit_Consumption            -1.342875e-05
Green_Vegetables_Consumption -1.759398e-04
FriedPotato_Consumption       3.371648e-04
Exercise_conv                -2.797477e-02
Smoking_History_conv          6.982044e-02
Arthritis_conv                6.199040e-02
Skin_Cancer_conv              3.586665e-02
Other_Cancer_conv             2.849747e-02
No_Diabetes                  -6.405290e-02
Pre_Diabetes                  1.959202e-02
Yes_Diabetes                  7.031742e-02
Pregnancy_Diabetes           -1.018350e-01
General_HealthExcellent      -1.355876e-01
General_HealthFair            1.002643e-01
General_HealthGood            7.818638e-03
General_HealthPoor            1.749831e-01
General_HealthVery_Good      -8.059058e-02
`Age_Category18-24`          -2.090371e-01
`Age_Category25-29`          -1.997314e-01
`Age_Category30-34`          -1.931249e-01
`Age_Category35-39`          -2.001651e-01
`Age_Category40-44`          -1.764506e-01
`Age_Category45-49`          -1.276018e-01
`Age_Category50-54`          -8.711049e-02
`Age_Category55-59`          -2.643974e-02
`Age_Category60-64`          -1.475437e-02
`Age_Category65-69`           5.829665e-02
`Age_Category70-74`           9.067370e-02
`Age_Category75-79`           1.219728e-01
`Age_Category80+`             1.755361e-01
`More than 5 years`          -3.988698e-02
Never                        -7.379129e-02
`Past 2 years`               -2.961816e-02
`PAst 5 years`               -5.457711e-02
`Past 1 year`                 4.304876e-02
```

The lambda value chosen by using the one standard error rule (lambda.1se) was used to fit this Ridge regression model.

Compared to the first model (with lambda value that minimizes the cross-validation error), this model has smaller coefficients indicating less of an influence on the response variable (Heart_Disease).

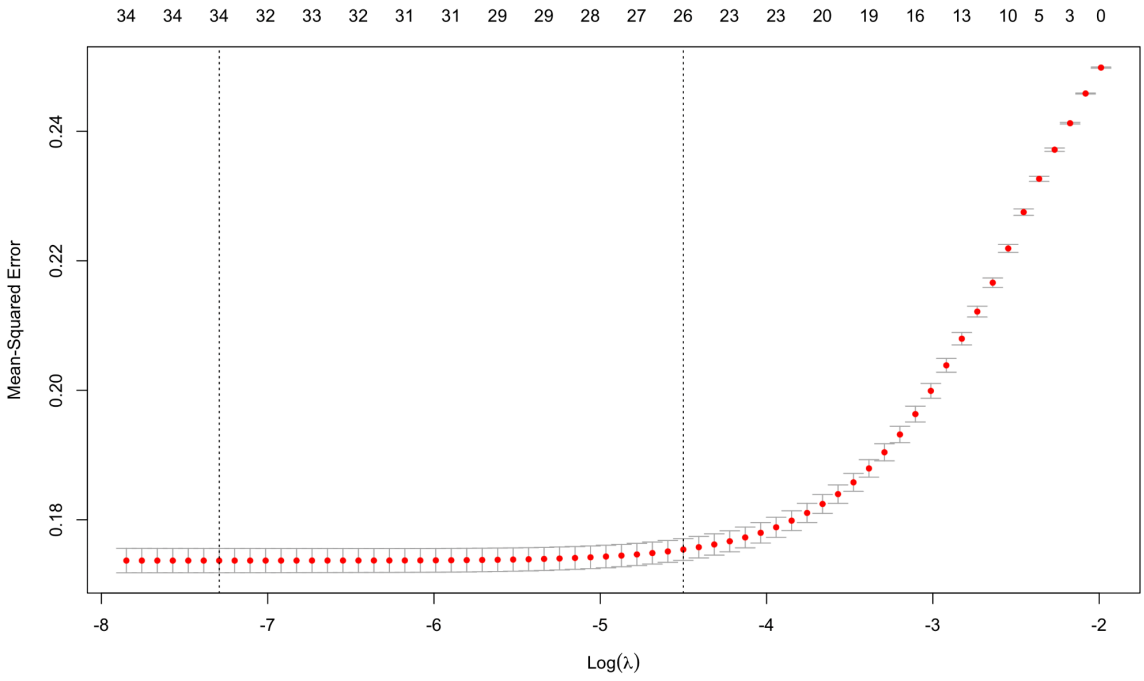
In contrast to the model where lambda minimizes cross-validation error, this model with lambda within one standard error, tends to reduce the coefficients towards zero.

**Model 3 (LASSO regression)**

```
set.seed(256)
cv.lasso <- cv.glmnet(train_x, train_y, nfolds = 10)
plot(cv.lasso)
```

**Figure 22**

*Plot for LASSO regularization*




The function cv.glmnet () selects the best predictive model λ by cross-validation, similar to the Ridge regression model described above.

```
> log(cv.lasso$lambda.min)
[1] -7.29153
> log(cv.lasso$lambda.1se)
[1] -4.500517
```

Lambda.min has a logarithm of about -7.29153.

This is the optimal lambda value for reducing prediction error as it correlates with the cross-validated error.

Lambda.1se has a logarithm of about -4.500517.

During cross-validation, this lambda value is determined using the single standard error rule.

Selecting a less complex model that reduces cross-validated error and falls within one standard error of the lambda value becomes easy.

**Figure 23**

*Coefficients after LASSO regularization at lambda value that minimizes the mean cross-validated error*

```
> lasso.model.min <- glmnet(train_x, train_y, alpha = 1, lambda = cv.lasso$lambda.min)
> coef(lasso.model.min)
40 x 1 sparse Matrix of class "dgCMatrix"
                                        s0
(Intercept)                  -1.156964e-01
Height_.cm.                   3.006996e-03
Weight_.kg.                   .
BMI                           9.566300e-04
Alcohol_Consumption          -1.513262e-03
Fruit_Consumption             .
Green_Vegetables_Consumption -2.740461e-05
FriedPotato_Consumption       6.247219e-04
Exercise_conv                -1.299168e-02
Smoking_History_conv          7.674650e-02
Arthritis_conv                5.144560e-02
Skin_Cancer_conv              1.652533e-02
Other_Cancer_conv             9.505591e-03
No_Diabetes                  -7.475367e-02
Pre_Diabetes                  .
Yes_Diabetes                  5.221913e-02
Pregnancy_Diabetes           -1.141217e-01
General_HealthExcellent      -1.693025e-01
General_HealthFair            1.164398e-01
General_HealthGood            .
General_HealthPoor            2.190461e-01
General_HealthVery_Good      -1.039429e-01
`Age_Category18-24`          -2.506645e-01
`Age_Category25-29`          -2.387481e-01
`Age_Category30-34`          -2.301302e-01
`Age_Category35-39`          -2.387054e-01
`Age_Category40-44`          -2.063203e-01
`Age_Category45-49`          -1.445153e-01
`Age_Category50-54`          -9.122364e-02
`Age_Category55-59`          -1.366084e-02
`Age_Category60-64`           .
`Age_Category65-69`           9.602669e-02
`Age_Category70-74`           1.389851e-01
`Age_Category75-79`           1.802211e-01
`Age_Category80+`             2.520250e-01
`More than 5 years`          -6.480251e-04
Never                        -4.495106e-02
`Past 2 years`                .
`PAst 5 years`               -1.736854e-02
`Past 1 year`                 6.012227e-02
```

**Figure 24**

*Coefficients after LASSO regularization at lambda value within one standard error*

```
> lasso.model.1se <- glmnet(train_x, train_y, alpha = 1, lambda = cv.lasso$lambda.1se)
> coef(lasso.model.1se)
40 x 1 sparse Matrix of class "dgCMatrix"
                                      s0
(Intercept)                  0.0964939975
Height_.cm.                  0.0016750184
Weight_.kg.                  0.0000592972
BMI                          .
Alcohol_Consumption         -0.0001873537
Fruit_Consumption            .
Green_Vegetables_Consumption .
FriedPotato_Consumption      .
Exercise_conv               -0.0075703250
Smoking_History_conv         0.0687075589
Arthritis_conv               0.0600996816
Skin_Cancer_conv             0.0117856567
Other_Cancer_conv            0.0019730370
No_Diabetes                 -0.0543609873
Pre_Diabetes                 .
Yes_Diabetes                 0.0798163718
Pregnancy_Diabetes           .
General_HealthExcellent     -0.1605693756
General_HealthFair           0.1044859833
General_HealthGood           .
General_HealthPoor           0.1964885281
General_HealthVery_Good     -0.0940888682
`Age_Category18-24`         -0.1911201351
`Age_Category25-29`         -0.1697629334
`Age_Category30-34`         -0.1720643508
`Age_Category35-39`         -0.1839131938
`Age_Category40-44`         -0.1526365016
`Age_Category45-49`         -0.0913083432
`Age_Category50-54`         -0.0440327616
`Age_Category55-59`          .
`Age_Category60-64`          .
`Age_Category65-69`          0.0778715135
`Age_Category70-74`          0.1218856169
`Age_Category75-79`          0.1594210997
`Age_Category80+`            0.2301140290
`More than 5 years`          .
Never                        .
`Past 2 years`               .
`PAst 5 years`               .
`Past 1 year`                0.0637935464
```

Lasso regression selects variables by decreasing coefficients to zero, reducing the effect of less relevant variables.Coefficients in the lasso model.1se lowering to zero indicates that these factors are less relevant in predicting heart disease for the specified lambda value.

By efficiently removing less important variables, Lasso regression with lambda chosen using one standard error rule contributes to the creation of a more robust model that is easier to understand and comprehend.

## Figure 25

*Confusion matrix for train data after LASSO regularization*

```
> confusionMatrix(train_predictions_binary, train_data$Heart_Disease, positive = "Yes")
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  5033 1559
       Yes 1967 5441

               Accuracy : 0.7481
                 95% CI : (0.7409, 0.7553)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4963

 Mcnemar's Test P-Value : 7.174e-12

            Sensitivity : 0.7773
            Specificity : 0.7190
         Pos Pred Value : 0.7345
         Neg Pred Value : 0.7635
             Prevalence : 0.5000
         Detection Rate : 0.3886
   Detection Prevalence : 0.5291
      Balanced Accuracy : 0.7481

       'Positive' Class : Yes
```

## Figure 26

*Confusion matrix for test data after LASSO regularization*

```
> confusionMatrix(test_predictions_binary, test_data$Heart_Disease, positive = "Yes")
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  2225  657
       Yes  775 2343

               Accuracy : 0.7613
                 95% CI : (0.7503, 0.7721)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5227

 Mcnemar's Test P-Value : 0.001989

            Sensitivity : 0.7810
            Specificity : 0.7417
         Pos Pred Value : 0.7514
         Neg Pred Value : 0.7720
             Prevalence : 0.5000
         Detection Rate : 0.3905
   Detection Prevalence : 0.5197
      Balanced Accuracy : 0.7613

       'Positive' Class : Yes
```

**Figure 27**

*ROC curve after LASSO regularization*

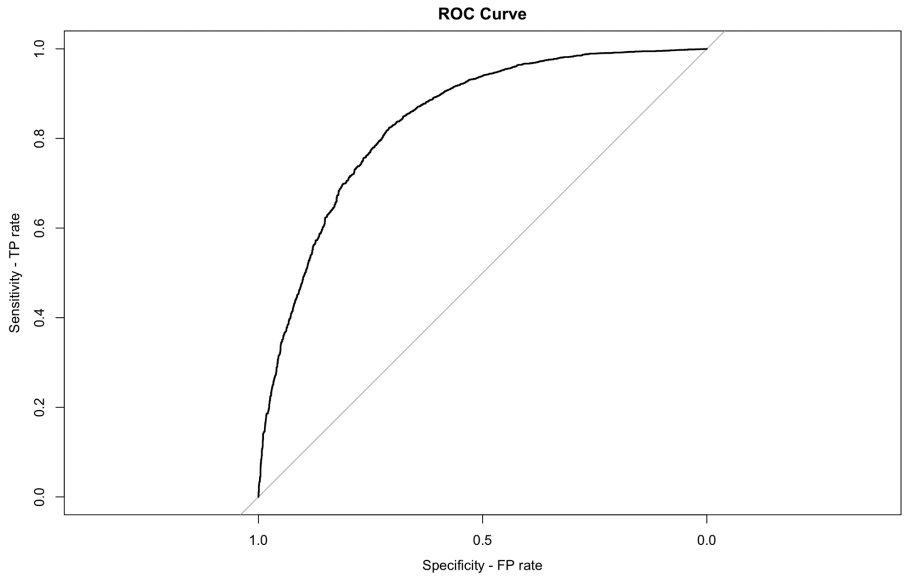



**Figure 28**

*Area Under the curve after LASSO regularization*

```
> auc
Area under the curve: 0.8387
```

**Comparison of Models**

Compared to the stepwise feature selection model, the performance metrics of the LASSO regression model are quite similar but it shrinks the coefficients of the predictor variables further reducing the complexity of the model by only keeping the predictor variables which contribute significantly to predicting heart disease. The predictor variables after performing LASSO regularization have been reduced to 24 compared to 30 predictor variables in stepwise feature selection. Also, the performance metrics of the model have not been compromised.

**Question 3: How does exercise influence the occurrence of depression in an individual?**

For this, we made the model and performed logistic regression. We used the glm function in R, which stands for "generalized linear model". This function can handle logistic regression when you set the family argument to "binomial"

**Figure 29:**

```
> # Summary of the model to check the coefficients
> summary(model)

Call:
glm(formula = Depression ~ Exercise, family = binomial, data = data)

Coefficients:
            Estimate Std. Error z value             Pr(>|z|)
(Intercept) -1.17266    0.01366  -85.84 <0.0000000000000002 ***
ExerciseYes -0.37986    0.01604  -23.68 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 123861  on 127732  degrees of freedom
Residual deviance: 123318  on 127731  degrees of freedom
AIC: 123322

Number of Fisher Scoring iterations: 4
```

Note. We can observe from the summary that exercise is a significant predictor of depression.

Interpretation of the results:

**Estimate:** These are the coefficients of the logistic regression model. The intercept is -1.17266, and the coefficient for Exercise is -0.37986. In the context of logistic regression, these coefficients represent the log odds. For example, for each unit increase in Exercise (going from "No" to "Yes"), the log odds of Depression decrease by 0.37986.

**Std. Error:** These values represent the standard errors of the coefficients. They are used in hypothesis testing for the coefficients (to determine whether the predictors are statistically significant) and in the construction of confidence intervals.

**z value:** These are the z-statistics for the coefficients. They are calculated as (Estimate/Std. Error).

**Pr(>|z|):** These are the p-values associated with the z-statistics. They represent the probability that you would see these results (or more extreme) if the null hypothesis were true. In this case, the null hypothesis is that the coefficient is zero (i.e., the predictor has no effect). The very small p-values here suggest that both the intercept and Exercise are statistically significant predictors in the model.

**Null deviance and Residual deviance:** These are goodness-of-fit statistics for the model. The null deviance represents the deviance of a model with no predictors, while the residual deviance represents the deviance of the model with predictors. The smaller the residual deviance is compared to the null deviance, the better your model is doing.

**AIC:** This is the Akaike Information Criterion for the model. It's a measure of the relative quality of statistical models for a given dataset. Lower AIC values indicate better-fitting models.

**Number of Fisher Scoring iterations:** This is the number of iterations it took for the logistic regression model to converge.

In conclusion, the logistic regression model suggests that exercise is a significant predictor of depression, with individuals who exercise having lower odds of depression.

Firstly to get a rough estimate of the accuracy of the model we used a confusion matrix and used Yes & No to simplify our understanding of the result.

**Figure 30:**

```
> # To check the accuracy of the model using a confusion matrix
> table(Predicted = predicted_depression, Actual = data$Depression)
         Actual
Predicted     No    Yes
       No 103585  24148
```

Note. We can see that our model has a good accuracy of prediction from the confusion matrix.

This tells us that 103585 times it predicted the correct conclusion of not having depression and 24148 times had the incorrect prediction.

For further confirmation, we calculated other metrics to check the performance of the model. We first split the data into training (70%) and test set (30%). Then we fit the model on the training data and made predictions on the test data. Later calculated precision, recall, F1 score and the AUC-ROC.

**Figure 31:**

```
> print(paste("Precision: ", round(precision, 2)))
[1] "Precision:  0.81"
> print(paste("Recall: ", round(recall, 2)))
[1] "Recall:  1"
> print(paste("F1 Score: ", round(f1_score, 2)))
[1] "F1 Score:  0.9"

> print(paste("AUC-ROC: ", round(auc, 2)))
[1] "AUC-ROC:  0.53"
```

Note. These show the metric values for the performance of the model.

- Precision of 0.81 means that when the model predicts an individual has depression, it is correct about 81% of the time.
- Recall of 1 means that the model correctly identifies 100% of the individuals with depression.
- The F1 Score is the harmonic mean of Precision and Recall and is a better measure than Accuracy in cases where the class distribution is imbalanced. An F1 Score of 0.9 is quite high and suggests that the model has a good balance between precision and recall.
- An AUC-ROC score of 0.53 means that your model has a slight ability to distinguish between positive and negative classes. It's slightly better than random guessing.

Now to check additional factors that affected depression we chose more variables like Smoking History, Alcohol Consumption, Fruit Consumption and Green vegetable consumption including Exercise.

We then used feature selection methods like forward, backward and stepwise to make the optimum model. Results of backward elimination, forward and stepwise give us the same summary since all the factors were statistically significant:

**Figure 32:**

```
> summary(model)

Call:
glm(formula = Depression ~ Smoking_History + Exercise + Alcohol_Consumption +
    Fruit_Consumption + Green_Vegetables_Consumption, family = binomial,
    data = data)

Coefficients:
                             Estimate Std. Error z value             Pr(>|z|)
(Intercept)                -1.1945881  0.0214152 -55.782 < 0.0000000000000002 ***
Smoking_HistoryYes          0.4786581  0.0144998  33.011 < 0.0000000000000002 ***
ExerciseYes                -0.2789275  0.0165324 -16.872 < 0.0000000000000002 ***
Alcohol_Consumption        -0.0125589  0.0022728  -5.526          0.0000000328 ***
Fruit_Consumption          -0.0064147  0.0007352  -8.725 < 0.0000000000000002 ***
Green_Vegetables_Consumption -0.0128319  0.0011540 -11.119 < 0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 123861  on 127732  degrees of freedom
Residual deviance: 121924  on 127727  degrees of freedom
AIC: 121936

Number of Fisher Scoring iterations: 4
```

Note. This model summary tells us that all variables in this model are significant predictors.

Then we fit the logistic regression model and check the performance from the metrics like Accuracy, Precision, Recall, F1 score and the AUC-ROC.

**Figure 33:**

```
> # Print the metrics
> print(paste("Accuracy: ", round(accuracy, 2)))
[1] "Accuracy:  0.81"
> print(paste("Precision: ", round(precision, 2)))
[1] "Precision:  0.81"
> print(paste("Recall: ", round(recall, 2)))
[1] "Recall:  1"
> print(paste("F1 Score: ", round(f1_score, 2)))
[1] "F1 Score:  0.9"


 > print(paste("AUC-ROC: ", round(auc, 2)))
 [1] "AUC-ROC:  0.59"
```

Note. These show the values of the metrics to tell us about the performance of the model.

Interpretation of the metrics:

- Accuracy of 0.81 means that the model correctly predicts whether an individual has depression 81% of the time.
- Precision of 0.81 means that when the model predicts an individual has depression, it is correct about 81% of the time.
- Recall of 1 means that the model correctly identifies 100% of the individuals with depression.
- The F1 Score is the harmonic mean of Precision and Recall and is a better measure than Accuracy in cases where the class distribution is imbalanced. An F1 Score of 0.9 is quite high and suggests that the model has a good balance between precision and recall.
- The AUC-ROC is 0.59. The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) is a measure of how well the model can distinguish between individuals with depression and those without. An AUC-ROC of 0.59 suggests that the model has a moderate ability to distinguish between the two classes.

The hypotheses for the models would be:

1st Model:

**Null Hypothesis (H0):** There is no association between exercise habits and depression status

**Alternate Hypothesis (H1):** There is an association between exercise habits and depression status.

2nd Model:

**Null Hypothesis (H0):** There is no association between the factors (Smoking_History, Exercise, Alcohol_Consumption, Fruit_Consumption, and Green_Vegetables_Consumption) and depression status.

**Alternate Hypothesis (H1):** At least one of the factors (Smoking_History, Exercise, Alcohol_Consumption, Fruit_Consumption, and Green_Vegetables_Consumption) is associated with depression status.

In both cases we reject the null hypothesis.

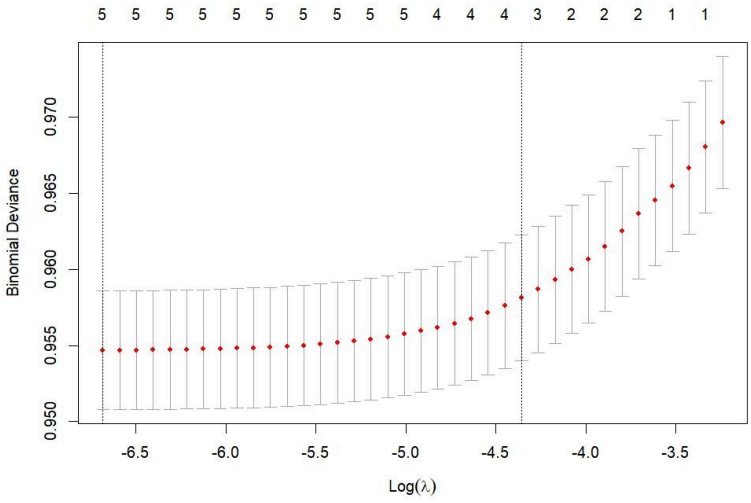
We also performed Lasso Regression for our model:

Figure 35:

```
> confusionMatrix(as.factor(lasso.preds.train.binary), as.factor(train_y))
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 103585  24148
         1      0      0

               Accuracy : 0.8109
                 95% CI : (0.8088, 0.8131)
    No Information Rate : 0.8109
    P-Value [Acc > NIR] : 0.5017

                  Kappa : 0

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 1.0000
            Specificity : 0.0000
         Pos Pred Value : 0.8109
         Neg Pred Value :    NaN
             Prevalence : 0.8109
         Detection Rate : 0.8109
   Detection Prevalence : 1.0000
      Balanced Accuracy : 0.5000

       'Positive' Class : 0
```

Note. This gives us the similar accuracy, recall as our previous model.

Figure 36:




Note. Plot for Lasso regularization

## Conclusion

In conclusion, answering every question gives us a great deal of information, and we try to significantly contribute to the creation of models that can affect various aspects of individual health.

We develop five different models in total to determine the best model to predict an individual's BMI. Each model goes through several stages of development, including handling outliers, data transformation, and feature selection. Finally, we identify the model (the fourth model mentioned in question 2 section above) that is more appropriate, less complex, and more accurate, with an adjusted R-squared score of 98.55%, MAE of 0.045, and RMSE of 0.061. After further model optimisation using the LASSO and RIDGE methods, we discovered that the LASSO method produced the best model with fewer features and the same performance.

The logistic regression model shows promising performance in predicting heart disease based on many health indicators and lifestyle variables. The model's Area Under the Curve (AUC) result of 0.836 indicates that it has strong discriminative capability through thorough development, analysis and feature selection procedures. This shows that the model can distinguish between those who have cardiac disease and those who don't.

We developed the model to see the impact of exercise on depression using logistic regression. We found it to be a significant factor that affected depression. We measured the metrics to check the performance of the model like Precision, Recall, and F1 score. This gave us conclusions like our model was good at predictions and had a good balance between precision and recall. We later added more variables like Smoking History, Alcohol Consumption, Fruit Consumption and Green vegetable consumption to check the impact and made the newer model accordingly.

We found them all to be making an impact on depression and all selection methods ended up giving the same output. We then checked the metrics of our model and it improved the AUC-ROC. While these metrics can provide a good summary of our model’s performance, they don’t tell the whole story. It’s also important to consider the context and the cost of false positives and false negatives. For example, in a medical context, a false negative (missing a true case of depression) might be more serious than a false positive (incorrectly predicting depression).

## References

*Assessing Your Weight and Health Risk*. (n.d.). NHLBI. Retrieved February 10, 2024, from https://www.nhlbi.nih.gov/health/educational/lose_wt/risk.htm

*Cardiovascular Diseases Risk Prediction Dataset*. (n.d.). Kaggle. Retrieved February 11, 2024, from https://www.kaggle.com/datasets/alphiree/cardiovascular-diseases-risk-prediction-dataset

Cho, D. (2022, December 16). *Logistic Regression: Feature Selection Methods*. RPubs. Retrieved February 11, 2024, from https://rpubs.com/ohcsnad/feature_selection_methods

Jain, S. (2023, August 1). *Encoding Categorical Data in R*. GeeksforGeeks. Retrieved February 11, 2024, from https://www.geeksforgeeks.org/encoding-categorical-data-in-r/