Heart Disease classification

Barzan Giorgia, Olivato Matteo

29-05-2023

Introduction

This project focuses on a binary classification problem concerning heart diseases. According to the Centers for Disease Control and Prevention (CDC)'s 2020 annual survey, heart disease ranks among the leading causes of death across various racial groups in the US, including African Americans, American Indians, Alaska Natives, and white individuals. Approximately 47% of Americans possess at least one of three primary risk factors for heart disease: high blood pressure, high cholesterol levels, and smoking. Other significant indicators include diabetes, obesity (measured by a high BMI), lack of physical activity, and excessive alcohol consumption. Detecting and preventing these influential factors is of utmost importance in healthcare. The purpose of this project is to develop a prediction on the causes related to the presence or absence of heart diseases.

Libraries

The followings are the libraries that has been used in this project.

```
library(tidyverse)
library(corrplot)
library(ppcor)
library(ggplot2)
library(gridExtra)
library(caret)
library(pROC)
library(tidymodels)
library(glmnet)
library(car)
library(class)
library(MASS)
library(dplyr)
```

Uploading data

This dataset is part of the Behavioral Risk Factor Surveillance System (BRFSS) dataset, collected by the CDC, and contains health-related data from a telephone survey conducted in 2020 in the United States. It includes information on risk behaviors and chronic health conditions. This dataset consists in 319795 observations and 18 variables. With the following function we upload the dataset and look at the first observations.

```
# We now upload the file with the function read.csv and look at the first
# observations for each variable
dati=read.csv("heart_2020_cleaned.csv")
head(dati)
```

```
##   HeartDisease   BMI Smoking AlcoholDrinking Stroke PhysicalHealth MentalHealth
## 1           No 16.60     Yes              No     No              3           30
```

```
## 2           No 20.34      No              No    Yes              0           0
## 3           No 26.58     Yes              No     No             20          30
## 4           No 24.21      No              No     No              0           0
## 5           No 23.71      No              No     No             28           0
## 6          Yes 28.87     Yes              No     No              6           0
##   DiffWalking    Sex AgeCategory  Race Diabetic PhysicalActivity GenHealth
## 1          No Female       55-59 White      Yes              Yes Very good
## 2          No Female 80 or older White       No              Yes Very good
## 3          No   Male       65-69 White      Yes              Yes      Fair
## 4          No Female       75-79 White       No               No      Good
## 5         Yes Female       40-44 White       No              Yes Very good
## 6         Yes Female       75-79 Black       No               No      Fair
##   SleepTime Asthma KidneyDisease SkinCancer
## 1         5    Yes            No        Yes
## 2         7     No            No         No
## 3         8    Yes            No         No
## 4         6     No            No        Yes
## 5         8     No            No         No
## 6        12     No            No         No
```

Here is a brief description of each variable:

Variable	Description
HeartDisease	Has or not a heart disease
BMI	Body mass index
Smoking	Has smoked at least 100 cigarettes in their life
AlcoholDrinking	Heavy drinker
Stroke	Has had a stroke
PhysicalHealth	For how many days in the past 30 days the physical health was not good
MentalHealth	For how many days in the past 30 days the mental health was not good
DiffWalking	Has serious difficulty walking or climbing stairs
Sex	Male or female
AgeCategory	To which age category they belong
Race	Imputed race/ethnicity value
Diabetic	Has diabetes
PhysicalActivity	Exercise in the past 30 days
GenHealth	General health condition
SleepTime	Hours they sleep on average
Asthma	Has asthma
KidneyDisease	Has a kidney disease
SkinCancer	Has skin cancer

Data pre-processing

Before delving into the analysis of this dataset, it is essential to examine its structure. To preprocess the data, we start by checking for missing values. Specifically, we use the following command to calculate the total count of NA values for each variable.

```
# look for NAs
sapply(dati, function(x) sum(is.na(x)))
```

```
##     HeartDisease              BMI          Smoking  AlcoholDrinking
##                0                0                0                0
##           Stroke   PhysicalHealth     MentalHealth      DiffWalking
```

```
##                 0               0               0               0
##               Sex      AgeCategory            Race        Diabetic
##                 0               0               0               0
## PhysicalActivity       GenHealth       SleepTime          Asthma
##                 0               0               0               0
##    KidneyDisease       SkinCancer
##                 0               0
```

As the sum of missing values is zero for each variable, we can proceed with the analysis. We now examine the characteristics of each variable:

```
str(dati)
```

```
## 'data.frame':    319795 obs. of  18 variables:
##  $ HeartDisease    : chr  "No" "No" "No" "No" ...
##  $ BMI             : num  16.6 20.3 26.6 24.2 23.7 ...
##  $ Smoking         : chr  "Yes" "No" "Yes" "No" ...
##  $ AlcoholDrinking : chr  "No" "No" "No" "No" ...
##  $ Stroke          : chr  "No" "Yes" "No" "No" ...
##  $ PhysicalHealth  : num  3 0 20 0 28 6 15 5 0 0 ...
##  $ MentalHealth    : num  30 0 30 0 0 0 0 0 0 0 ...
##  $ DiffWalking     : chr  "No" "No" "No" "No" ...
##  $ Sex             : chr  "Female" "Female" "Male" "Female" ...
##  $ AgeCategory     : chr  "55-59" "80 or older" "65-69" "75-79" ...
##  $ Race            : chr  "White" "White" "White" "White" ...
##  $ Diabetic        : chr  "Yes" "No" "Yes" "No" ...
##  $ PhysicalActivity: chr  "Yes" "Yes" "Yes" "No" ...
##  $ GenHealth       : chr  "Very good" "Very good" "Fair" "Good" ...
##  $ SleepTime       : num  5 7 8 6 8 12 4 9 5 10 ...
##  $ Asthma          : chr  "Yes" "No" "Yes" "No" ...
##  $ KidneyDisease   : chr  "No" "No" "No" "No" ...
##  $ SkinCancer      : chr  "Yes" "No" "No" "Yes" ...
```

As we can see from these results, there are many variables of the type 'char', we therefore need to change their nature in order to use the variables for our analysis. In particular, we convert the char variables into numeric ones, assigning to each char type a number.

```
# we convert the response variable from "Yes" and "No" to 1 and 0 and do the
# same for the other binary variables
dati$HeartDisease <- ifelse(dati$HeartDisease == "Yes", 1, 0)
dati$Smoking <- ifelse(dati$Smoking == "Yes", 1, 0)
dati$AlcoholDrinking <- ifelse(dati$AlcoholDrinking == "Yes", 1, 0)
dati$Stroke <- ifelse(dati$Stroke == "Yes", 1, 0)
dati$DiffWalking <- ifelse(dati$DiffWalking == "Yes", 1, 0)
dati$Diabetic <- ifelse(dati$Diabetic == "Yes", 1, 0)
dati$PhysicalActivity <- ifelse(dati$PhysicalActivity == "Yes", 1, 0)
dati$Asthma <- ifelse(dati$Asthma == "Yes", 1, 0)
dati$KidneyDisease <- ifelse(dati$KidneyDisease == "Yes", 1, 0)
dati$SkinCancer <- ifelse(dati$SkinCancer == "Yes", 1, 0)
# here we assign "1" to "Male" and "0" to "Female"
dati$Sex <- ifelse(dati$Sex == "Male", 1, 0)

conv <- c("18-24" = 1, "25-29" = 2, "30-34" = 3, "35-39" = 4, "40-44" = 5,
          "45-49" = 6, "50-54" = 7, "55-59" = 8, "60-64" = 9,
          "65-69" = 10, "70-74" = 11, "75-79" = 12, "80 or older" = 13)
```

```r
# transfrom the char variable into a numeric var
dati$AgeCategory <- ifelse(dati$AgeCategory %in% names(conv), conv[dati$AgeCategory], NA)

conv2=c("White" = 1,  "Black" = 2,  "Asian" = 3, "American Indian/Alaskan Native" = 4,
        "Other" = 5, "Hispanic" = 6)
dati$Race <- ifelse(dati$Race %in% names(conv2), conv2[dati$Race], NA)

conv3 <- c("Very good" = 1, "Fair" = 2, "Good" = 3, "Poor" = 4, "Excellent" = 5)
dati$GenHealth <- ifelse(dati$GenHealth %in% names(conv3), conv3[dati$GenHealth], NA)
```

We can now attach the dataset using the following function:

```r
attach(dati)
```

An imbalanced dataset, where the number of observations in one class differs significantly from the other, can yield unreliable results during model training. Therefore, we examine the distribution of the response variable *HeartDisease*. It is evident that the dataset is highly imbalanced, with 91.44% of the observations indicating the absence of heart disease. Later in the analysis, we will consider both the unbalanced and balanced datasets to explore potential improvements through modified data.

```r
table(HeartDisease)/nrow(dati) *100
```

```
## HeartDisease
##         0         1
## 91.440454  8.559546
```

Training and Test split

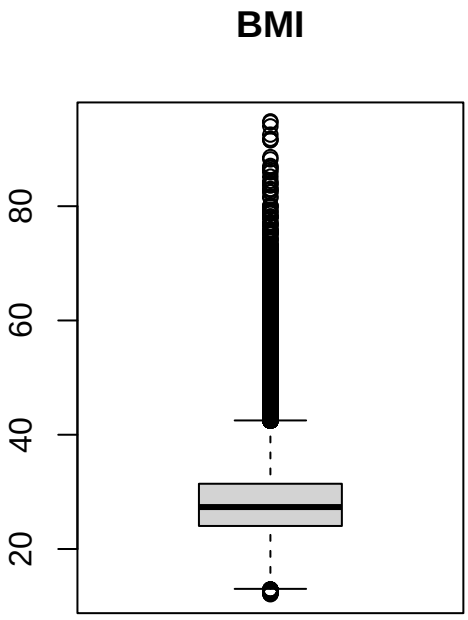
To train our models and evaluate their performance, we need to split the imbalanced dataset into training and test sets. We will allocate 75% of the data for training and reserve 25% for testing purposes.

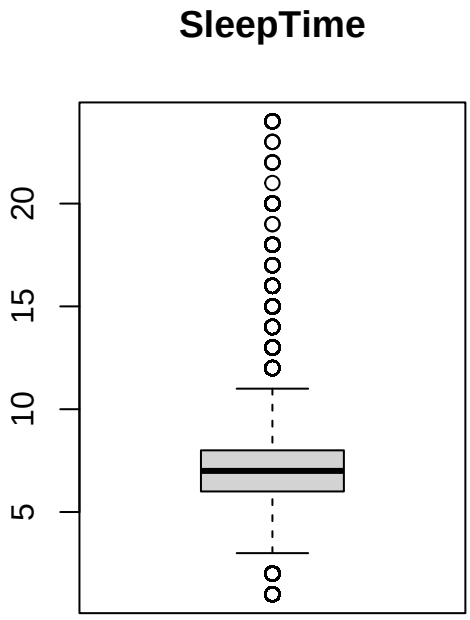
```r
# set a random seed to get always the same results
set.seed(123)

# divide the unbalanced dataset  in 75% train and 25% test
index <- sample(1:nrow(dati), size = round(0.75*nrow(dati)), replace = FALSE)
dati_train <- dati[index, ]
dati_test <- dati[-index, ]
```

Outliers

After dividing our data sets into train and test sets, we need to check for the presence of outliers. We decided to report in particular the boxplots of the independent variables *BMI* and *SleepTime*, which happen to have a lot of outliers. Despite identifying outliers in these two variables, we have made the decision not to remove them. In the case of the BMI variable, an exceptionally high value indicates severe obesity, while extremely low or high values in SleepTime can be characteristic of a chronically ill individual. It is important to retain these outliers because they hold valuable information for predicting whether a person has a heart disease. Outliers can provide insights into extreme cases or rare scenarios that are crucial for accurately predicting the presence of heart disease. By excluding them, we risk losing important data points that could impact the model's performance. Furthermore, they can help improve the robustness and generalizability of the predictive model, allowing it to handle a wider range of scenarios and variations in the data.

```r
par(mfrow=c(1,2))
boxplot(dati_train$BMI, main="BMI")
boxplot(dati_train$SleepTime, main="SleepTime")
```



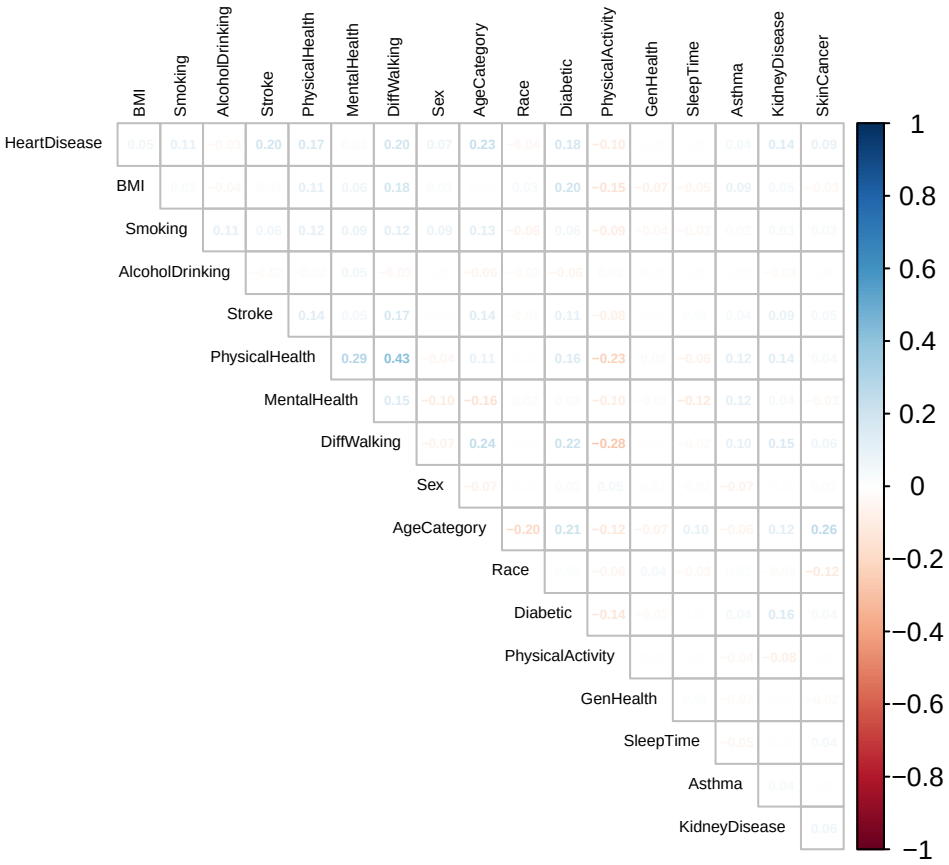


Exploratory Analysis

Correlations

We begin the exploratory analysis by looking at the correlations on the training dataset. To do so, we first compute the correlation matrix and then use the function *corrplot* from the library *corrplot* to visualize the correlations. In the graph below, the correlations between variables are represented by the numbers in each cell.

```
# correlation matrix on unbalanced dataset
library(corrplot)
corr_matrix <- cor(dati_train)
corrplot(corr_matrix, method = "number", diag = FALSE, number.cex = 0.4, tl.cex = 0.5,
         type = "upper", tl.col = "black")
```



By analyzing this graph, we can identify the variables that exhibit the highest correlation with our response variable. These variables are summarized in the following table:

Variable	Correlation_with_HeartDisease
Smoking	0.11
Stroke	0.20
PhysicalHealth	0.17
DiffWalking	0.20
AgeCategory	0.23
Diabetic	0.18
PhysicalActivity	-0.10
KidneyDisease	0.14

The most significant correlations can be summarized in the following table. The highest correlation is 0.43, while the remaining correlations range slightly above or below 0.20. These moderate correlations indicate a lower risk of collinearity, which will be further assessed during the model selection process.

Variable1	Variable2	Correlations
BMI	Diabetic	0.20
BMI	DiffWalking	0.18
BMI	PhysicalActivity	-0.15
Stroke	DiffWalking	0.17
PhysicalHealth	MentalHealth	0.29
PhysicalHealth	DiffWalking	0.43

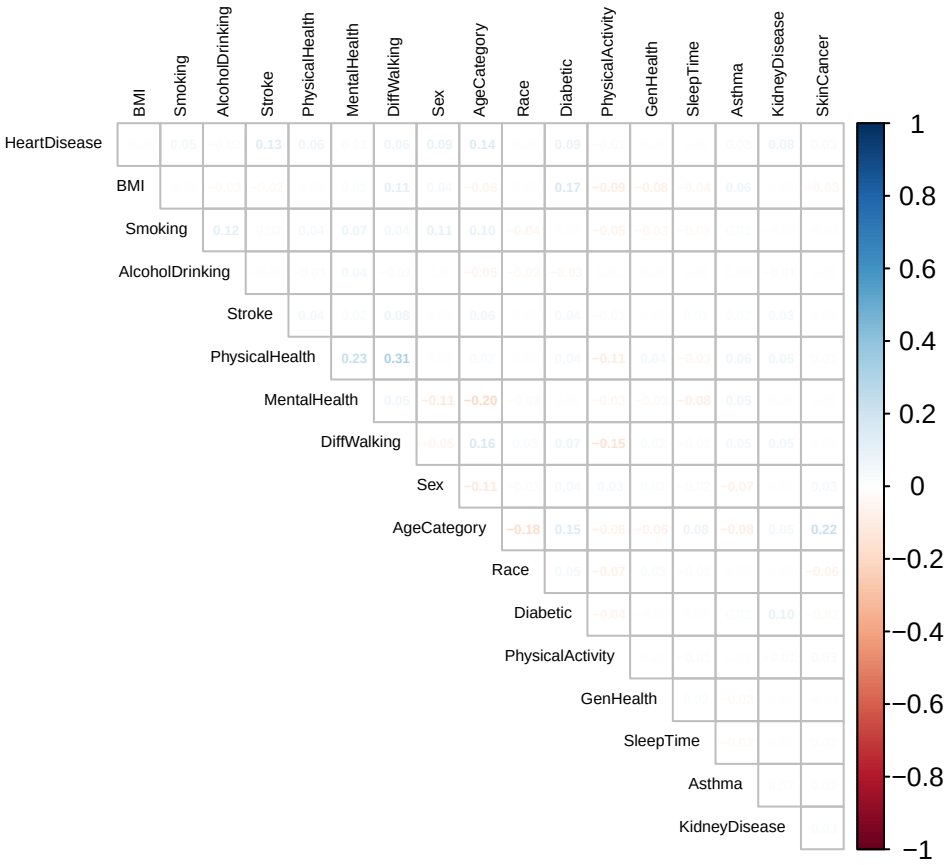
Variable1	Variable2	Correlations
PhysicalHealth	Diabetic	0.16
PhysicalHealth	PhysicalActivity	-0.23
MentalHealth	AgeCategory	-0.16
DiffWalking	AgeCategory	0.24
DiffWalking	Diabetic	0.22
DiffWalking	PhysicalActivity	-0.28
AgeCategory	Race	-0.20
AgeCategory	Diabetic	0.21
AgeCategory	SkinCancer	0.26
Diabetic	KidneyDisease	0.16

The preceding plot reveals the correlations among variables, taking into account their mutual influence. However, to mitigate this issue, we utilize the *correlation* library and its corresponding function to calculate the partial correlations between variables. Additionally, we employ the *pcor* function from the *ppcor* library to visualize these partial correlations in a similar plot. Due to the large size of the *correlation* output, we will not display it here.

```
# partial correlations
library(correlation)
#correlation(dati_train, partial=TRUE)
```

Based on the obtained partial correlations, we observe that the variables most strongly correlated with the response variable, when accounting for the influence of other variables, are *Stroke* and *AgeCategory*.

```
library(ppcor)
part_cor <- pcor(dati_train)$estimate
corrplot(part_cor, method = "number", diag = FALSE, number.cex = 0.4, tl.cex = 0.5,
         type = "upper", tl.col = "black")
```



The most significant partial correlations can be summarized in the following table:

Variable1	Variable2	Correlations
PhysicalHealth	MentalHealth	0.23
PhysicalHealth	DiffWalking	0.31
MentalHealth	AgeCategory	-0.20
DiffWalking	AgeCategory	0.16
DiffWalking	PhysicalActivity	-0.15
AgeCategory	Race	-0.18
AgeCategory	Diabetic	0.15
AgeCategory	SkinCancer	0.22

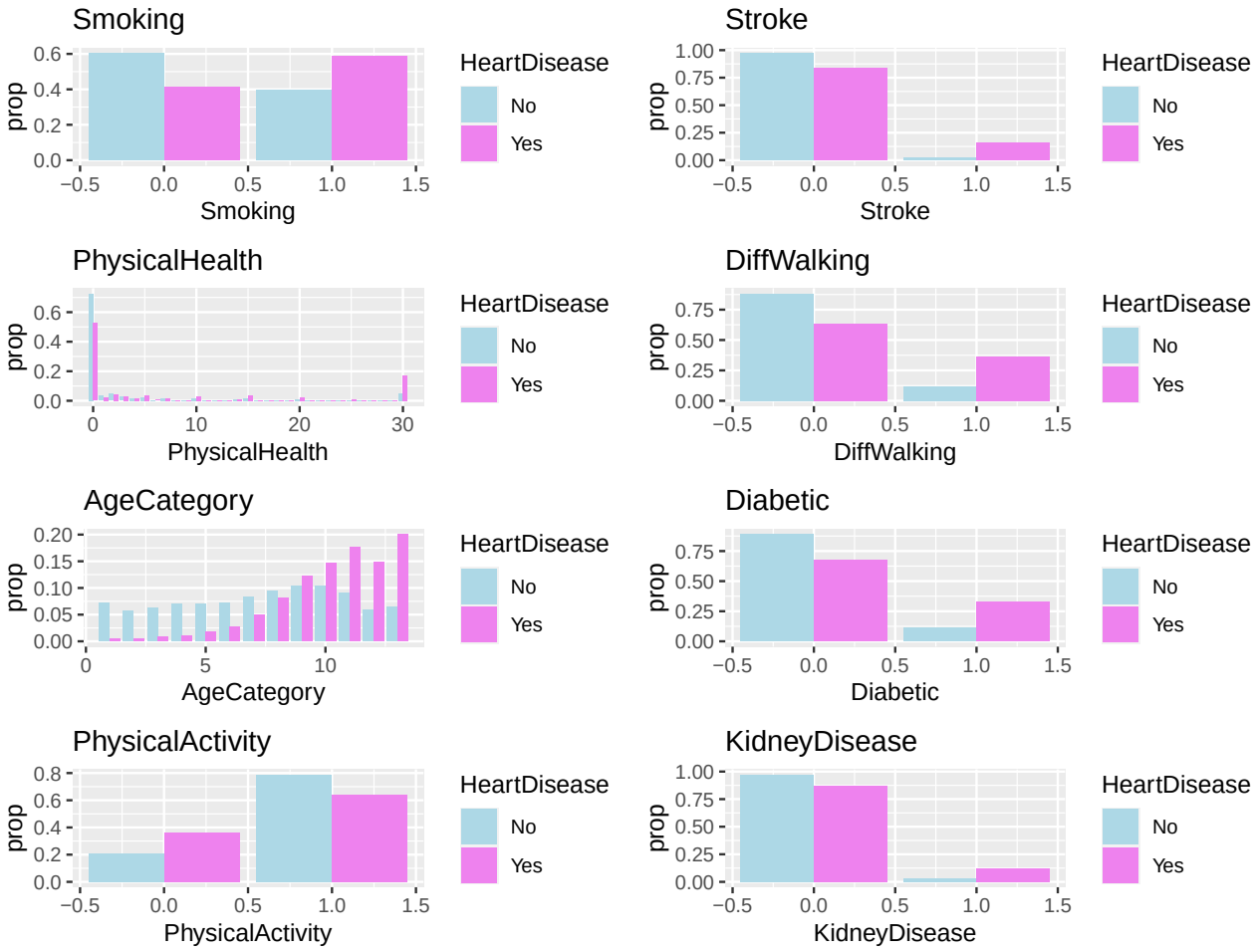
Plots

We will now delve deeper into the behavior of the most correlated independent variables concerning the response variable *HeartDisease*. Through the following plots, we can observe the distributions of each variable in relation to the values of the response variable. Notably, individuals who have smoked at least 100 cigarettes in their lifetime are more likely to have a heart disease compared to those who have not. Regarding stroke, although it affects a small portion of the population, almost all individuals with a stroke also have a heart disease. While the majority of people who experience poor physical health are at risk, even those who have had a month of good physical health can still develop the disease. Regarding difficulty in walking, a significant number of individuals with this issue have a heart disease, but there are still instances of the disease among those without this difficulty. As the age increases, the proportion of individuals with a heart disease also grows. Although most individuals with diabetes have a heart disease, a considerable portion of non-diabetic individuals can still be affected. Among those who do not engage in physical activity, there is a higher presence of heart disease, but even those who engage in physical activity can still fall ill. While the majority

of the population does not have kidney disease, among those who do, almost all of them also have a heart disease.

```
library(gridExtra)
library(ggplot2)

dati_train$HeartDisease <- as.factor(dati_train$HeartDisease)
# ggplot of the variables
a<-ggplot(dati_train, aes(x=Smoking, fill=HeartDisease))+
  geom_bar(aes(y=after_stat(prop)), position="dodge")+
  ggtitle("Smoking")+
  xlab("Smoking")+
  scale_fill_manual(values = c("lightblue", "violet"), labels = c("No", "Yes"))

b<-ggplot(dati_train, aes(x=Stroke, fill=HeartDisease))+
  geom_bar(aes(y=after_stat(prop)), position="dodge")+
  ggtitle("Stroke")+
  xlab("Stroke")+
  scale_fill_manual(values = c("lightblue", "violet"), labels = c("No", "Yes"))

c<-ggplot(dati_train, aes(x=PhysicalHealth, fill=HeartDisease))+
  geom_bar(aes(y=after_stat(prop)), position="dodge")+
  ggtitle("PhysicalHealth")+
  xlab("PhysicalHealth")+
  scale_fill_manual(values = c("lightblue", "violet"), labels = c("No", "Yes"))

d<-ggplot(dati_train, aes(x=DiffWalking, fill=HeartDisease))+
  geom_bar(aes(y=after_stat(prop)), position="dodge")+
  ggtitle("DiffWalking")+
  xlab("DiffWalking")+
  scale_fill_manual(values = c("lightblue", "violet"), labels = c("No", "Yes"))

e<-ggplot(dati_train, aes(x=AgeCategory, fill=HeartDisease))+
  geom_bar(aes(y=after_stat(prop)), position="dodge")+
  ggtitle("AgeCategory")+
  xlab("AgeCategory")+
  scale_fill_manual(values = c("lightblue", "violet"), labels = c("No", "Yes"))

f<-ggplot(dati_train, aes(x=Diabetic, fill=HeartDisease))+
  geom_bar(aes(y=after_stat(prop)), position="dodge")+
  ggtitle("Diabetic")+
  xlab("Diabetic")+
  scale_fill_manual(values = c("lightblue", "violet"), labels = c("No", "Yes"))

g<-ggplot(dati_train, aes(x=PhysicalActivity, fill=HeartDisease))+
  geom_bar(aes(y=after_stat(prop)), position="dodge")+
  ggtitle("PhysicalActivity")+
  xlab("PhysicalActivity")+
  scale_fill_manual(values = c("lightblue", "violet"), labels = c("No", "Yes"))

h<-ggplot(dati_train, aes(x=KidneyDisease, fill=HeartDisease))+
  geom_bar(aes(y=after_stat(prop)), position="dodge")+
  ggtitle("KidneyDisease")+
  xlab("KidneyDisease")+
```

```
  scale_fill_manual(values = c("lightblue", "violet"), labels = c("No", "Yes"))

grid.arrange(a,b,c,d,e,f,g,h, nrow = 4, ncol = 2)
```



Data balancing

As we saw before, the dataset is very unbalanced, meaning that one class has a significantly larger number of instances compared to the other. To address this issue, we explored different techniques such as downsampling and oversampling. Downsampling involves reducing the size of the majority class by randomly selecting a subset of instances to match the number of instances in the minority class. On the other hand, oversampling involves increasing the size of the minority class by replicating or synthesizing new instances. After experimenting with both downsampling and oversampling approaches, we found that downsampling resulted in similar performance compared to oversampling. Therefore, we decided to proceed with the most efficient one, downsampling. By using the *downSample* formula from the *caret* library, we downscaled the training dataset, achieving a perfect balance between the *HeartDisease* classes. This approach allowed us to mitigate the impact of class imbalance and obtain reliable results for further analysis.

```
# balance the dataset
library(caret)
set.seed(123)
dati_balanced_train <- downSample(x = dati_train[,-1], y = as.factor(dati_train$HeartDisease))

# we rename the response variable and put it as first variable
names(dati_balanced_train)[names(dati_balanced_train) == "Class"] <- "HeartDisease"
```

```
table(dati_balanced_train$HeartDisease)/nrow(dati_balanced_train) *100
```

```
##
##  0  1
## 50 50
```

```
dati_balanced_train=cbind(dati_balanced_train$HeartDisease, dati_balanced_train[,-18])
names(dati_balanced_train)[names(dati_balanced_train) == "dati_balanced_train$HeartDisease"]<-"HeartDis
dati_balanced_train$HeartDisease <- ifelse(dati_balanced_train$HeartDisease == "1", 0, 1)
```

Model selection

In this section we will propose some modelling techniques and then compare their results. We will do this for the unbalanced and balanced datasets, in order to compare the performances.

Logistic Model

As a first modelling technique we propose the logistic regression. Logistic regression, specifically the binomial generalized linear model (GLM), is a widely used technique for analyzing binary response variables. In our analysis, the binary response variable *HeartDisease*, indicates whether a person has heart disease or not. The logistic regression model estimates the probability of heart disease based on predictor variables, using the logistic function. We begin by modeling the complete model that includes all the predictor variables. We then employ the stepwise technique, which combines both backward and forward selection, to refine the model. Stepwise selection is a variable selection method used to identify a subset of predictor variables that are most relevant to the outcome. It iteratively adds or removes variables based on their statistical significance and contribution to the model's performance. Backward selection begins with the full model and successively deletes variables that have the least impact on the model's fit. It repeatedly removes the least significant variable until. Forward selection, on the other hand, incrementally adds variables based on their individual significance. At each step, it selects the variable that contributes the most to the model's fit. By combining backward and forward selection in the stepwise technique, we aim to identify the most influential predictor variables and the most significant interactions, while maintaining a parsimonious model. This iterative process helps us refine the model, ensuring that it includes only the most relevant variables for predicting the outcome variable, thus enhancing its interpretability and predictive accuracy. After initializing the complete model that includes every variable, we check for the presence of collinearity by using the *VIF* function, which calculates the *Variance Inflation Function* of the independent variables. Collinearity occurs when independent variables in a regression model are highly correlated, making it difficult to determine their individual effects on the dependent variable. This can constitute a problem because it makes it difficult to determine the individual effects of these variables on the dependent variable, hence addressing it improves the model's accuracy and reliability.

Unbalanced dataset

Initially, we construct the complete model using the unbalanced dataset. By reviewing the summary, it is evident that the independent variables *Race* and *GenHealth* are not statistically significant. Hence, they will later be eliminated during the stepwise selection process. The *VIF* output reveals that the values are approximately 1, indicating the absence of collinearity among the variables.

```
# Complete model definition and summary

glm1u <- glm(HeartDisease ~ ., data = dati_train, family = binomial(link = "logit"))
summary(glm1u)
```

```
##
## Call:
## glm(formula = HeartDisease ~ ., family = binomial(link = "logit"),
```

```
##     data = dati_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2034  -0.4176  -0.2607  -0.1469   3.4847
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -6.2237126  0.0723407 -86.033  < 2e-16 ***
## BMI               0.0147583  0.0012977  11.373  < 2e-16 ***
## Smoking           0.4471138  0.0163251  27.388  < 2e-16 ***
## AlcoholDrinking  -0.2536899  0.0383431  -6.616 3.68e-11 ***
## Stroke            1.1475536  0.0260015  44.134  < 2e-16 ***
## PhysicalHealth    0.0221129  0.0008696  25.428  < 2e-16 ***
## MentalHealth      0.0106102  0.0010070  10.536  < 2e-16 ***
## DiffWalking       0.3889262  0.0206048  18.876  < 2e-16 ***
## Sex               0.7344163  0.0166433  44.127  < 2e-16 ***
## AgeCategory       0.2848345  0.0034842  81.751  < 2e-16 ***
## Race             -0.0055545  0.0059659  -0.931    0.352
## Diabetic          0.6127787  0.0188417  32.522  < 2e-16 ***
## PhysicalActivity -0.0962087  0.0183430  -5.245 1.56e-07 ***
## GenHealth         0.0001887  0.0061481   0.031    0.976
## SleepTime        -0.0301541  0.0050234  -6.003 1.94e-09 ***
## Asthma            0.3680845  0.0219512  16.768  < 2e-16 ***
## KidneyDisease     0.6809847  0.0281467  24.194  < 2e-16 ***
## SkinCancer        0.1347549  0.0221859   6.074 1.25e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 140171  on 239845  degrees of freedom
## Residual deviance: 111538  on 239828  degrees of freedom
## AIC: 111574
##
## Number of Fisher Scoring iterations: 6
```

```
# VIF of the complete model
library(car)
vif(glm1u)
```

```
##              BMI          Smoking  AlcoholDrinking           Stroke
##         1.165804         1.036943         1.018442         1.030563
##   PhysicalHealth     MentalHealth      DiffWalking              Sex
##         1.407689         1.228949         1.420195         1.070793
##      AgeCategory             Race         Diabetic PhysicalActivity
##         1.251939         1.048665         1.119960         1.169820
##        GenHealth        SleepTime           Asthma    KidneyDisease
##         1.020871         1.041198         1.057197         1.043690
##       SkinCancer
##         1.069767
```

Stepwise selection

We can now proceed with the stepwise selection. To avoid excessive length, the complete stepwise selection process, as in the addition and removal of non-significant interactions, was not included. Variables were added or removed based on statistical criteria such as p-values or information criteria (*AIC*), which help evaluate their significance. The process begins with the first modification of the complete model, glm2u, where the variables *Race* and *GenHealth* are removed from glm1u. The following addition of interactions derive from the partial correlation analysis, where we identified the most statistically significant ones. Moving to glm3u, we see the addition of interaction of *BMI* with *Diabetic* and *AlcoholDrinking*. In glm4u, two more interaction terms are introduced: *Smoking* * *KidneyDisease* and *Smoking* * *AgeCategory*. The subsequent steps, glm5u to glm11u, continue this process of adding or removing variables and interactions. Interaction terms related to variables such as *Stroke*, *DiffWalking*, *PhysicalHealth*, *MentalHealth*, *Sex*, *AgeCategory*, *Diabetic*, *Asthma*, *Smoking*, *PhysicalActivity*, and *KidneyDisease* are included. In glm11u, *Smoking* * *Stroke*, *AgeCategory* * *PhysicalActivity* and *AgeCategory* * *KidneyDisease* are removed. This operation has been done because these interactions do not have a very high significance. We then selected glm10u as the best model based on the lower *AIC* value, as it can be shown below.

```
glm2u <- update(glm1u, . ~ .- Race - GenHealth)

glm3u <- update(glm2u, . ~ . + BMI * Diabetic + BMI * AlcoholDrinking)

glm4u <- update(glm3u, . ~ . + Smoking * KidneyDisease +  Smoking * AgeCategory)

glm5u <- update(glm4u, . ~ . + Stroke * DiffWalking + Stroke * AgeCategory +
                  Stroke * Smoking)

glm6u <- update(glm5u, . ~ . + PhysicalHealth * DiffWalking +
                  PhysicalHealth * MentalHealth +PhysicalHealth * AgeCategory
                + PhysicalHealth * Diabetic)

glm7u <- update(glm6u, . ~ . + MentalHealth * Sex)

glm8u <- update(glm7u, . ~ . + DiffWalking * AgeCategory + DiffWalking * Sex)

glm9u <- update(glm8u, . ~ . + Sex * AgeCategory)

glm10u <- update(glm9u, . ~ . + AgeCategory * Diabetic +
                   AgeCategory * Asthma  + AgeCategory * Smoking +
                   AgeCategory * PhysicalActivity + AgeCategory * KidneyDisease)

glm11u <- update(glm10u, . ~ . - Smoking:Stroke - AgeCategory:PhysicalActivity -
                   AgeCategory:KidneyDisease)
```

```
# compare the models with the AIC
AIC(glm1u, glm2u, glm3u, glm4u, glm5u, glm6u, glm7u, glm8u, glm9u, glm10u, glm11u)
```

```
##        df      AIC
## glm1u  18 111573.7
## glm2u  16 111570.6
## glm3u  18 111554.5
## glm4u  20 111492.4
## glm5u  23 111178.9
## glm6u  27 111023.7
## glm7u  28 110993.5
## glm8u  30 110910.5
```

```
## glm9u  31 110880.8
## glm10u 35 110768.9
## glm11u 32 110777.5
```

```
summary(glm10u)
```

```
##
## Call:
## glm(formula = HeartDisease ~ BMI + Smoking + AlcoholDrinking +
##     Stroke + PhysicalHealth + MentalHealth + DiffWalking + Sex +
##     AgeCategory + Diabetic + PhysicalActivity + SleepTime + Asthma +
##     KidneyDisease + SkinCancer + BMI:Diabetic + BMI:AlcoholDrinking +
##     Smoking:KidneyDisease + Smoking:AgeCategory + Stroke:DiffWalking +
##     Stroke:AgeCategory + Smoking:Stroke + PhysicalHealth:DiffWalking +
##     PhysicalHealth:MentalHealth + PhysicalHealth:AgeCategory +
##     PhysicalHealth:Diabetic + MentalHealth:Sex + DiffWalking:AgeCategory +
##     DiffWalking:Sex + Sex:AgeCategory + AgeCategory:Diabetic +
##     AgeCategory:Asthma + AgeCategory:PhysicalActivity + AgeCategory:KidneyDisease,
##     family = binomial(link = "logit"), data = dati_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9697  -0.4244  -0.2469  -0.1249   3.6496
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -6.761e+00  1.156e-01 -58.500  < 2e-16 ***
## BMI                           1.552e-02  1.612e-03   9.628  < 2e-16 ***
## Smoking                       8.294e-01  6.828e-02  12.147  < 2e-16 ***
## AlcoholDrinking              -6.853e-01  1.718e-01  -3.989 6.63e-05 ***
## Stroke                        2.896e+00  1.205e-01  24.035  < 2e-16 ***
## PhysicalHealth                4.330e-02  3.726e-03  11.622  < 2e-16 ***
## MentalHealth                  1.573e-02  1.672e-03   9.405  < 2e-16 ***
## DiffWalking                   1.076e+00  9.754e-02  11.030  < 2e-16 ***
## Sex                           4.753e-01  7.295e-02   6.516 7.20e-11 ***
## AgeCategory                   3.235e-01  9.089e-03  35.599  < 2e-16 ***
## Diabetic                      1.811e+00  1.318e-01  13.738  < 2e-16 ***
## PhysicalActivity             -2.490e-01  7.640e-02  -3.260  0.00111 **
## SleepTime                    -2.513e-02  4.961e-03  -5.065 4.08e-07 ***
## Asthma                        6.702e-01  8.106e-02   8.267  < 2e-16 ***
## KidneyDisease                 1.133e+00  1.361e-01   8.322  < 2e-16 ***
## SkinCancer                    1.352e-01  2.188e-02   6.180 6.39e-10 ***
## BMI:Diabetic                 -1.363e-02  2.672e-03  -5.100 3.39e-07 ***
## BMI:AlcoholDrinking           1.545e-02  5.855e-03   2.638  0.00834 **
## Smoking:KidneyDisease        -1.655e-01  5.440e-02  -3.043  0.00234 **
## Smoking:AgeCategory          -3.828e-02  6.566e-03  -5.829 5.56e-09 ***
## Stroke:DiffWalking           -2.444e-01  5.053e-02  -4.837 1.32e-06 ***
## Stroke:AgeCategory           -1.557e-01  1.067e-02 -14.594  < 2e-16 ***
## Smoking:Stroke               -7.615e-02  5.097e-02  -1.494  0.13518
## PhysicalHealth:DiffWalking   -1.410e-02  1.643e-03  -8.578  < 2e-16 ***
## PhysicalHealth:MentalHealth  -3.578e-04  7.382e-05  -4.847 1.26e-06 ***
## PhysicalHealth:AgeCategory   -1.073e-03  3.476e-04  -3.085  0.00203 **
## PhysicalHealth:Diabetic      -4.256e-03  1.561e-03  -2.726  0.00642 **
## MentalHealth:Sex             -5.466e-03  1.906e-03  -2.868  0.00413 **
## DiffWalking:AgeCategory      -4.022e-02  8.591e-03  -4.682 2.84e-06 ***
```

```
## DiffWalking:Sex              -2.245e-01  3.555e-02  -6.315 2.70e-10 ***
## Sex:AgeCategory               3.574e-02  6.873e-03   5.200 1.99e-07 ***
## AgeCategory:Diabetic         -7.497e-02  8.332e-03  -8.999  < 2e-16 ***
## AgeCategory:Asthma           -3.507e-02  8.168e-03  -4.293 1.76e-05 ***
## AgeCategory:PhysicalActivity  1.477e-02  7.317e-03   2.019  0.04347 *
## AgeCategory:KidneyDisease    -3.513e-02  1.234e-02  -2.847  0.00441 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 140171  on 239845  degrees of freedom
## Residual deviance: 110699  on 239811  degrees of freedom
## AIC: 110769
##
## Number of Fisher Scoring iterations: 7
```

After selecting the best model we proceed on the analysis with the model validation. In this section we implement a prediction on the test set, using first the best model and then the complete model, then check for the best threshold by looking at three options: 0.4, 0.5 and 0.6. The selection of the best threshold is based on the proportion of observations assigned to class 1 that is closer to the actual proportion of the class, which in this case is 1%, as well as the one that brings better values in the metrics. We then compute the confusion matrix and extrapolate the true positive rate, the true negative rate, precision and recall in order to compute the F1 score. The F1 score is a metric used to evaluate the performance of a model. It combines precision and recall into a single value, providing a measure of the model's accuracy in identifying positive samples correctly. Precision represents the proportion of correctly predicted positive samples out of all samples predicted as positive, while recall measures the proportion of correctly predicted positive samples out of all actual positive samples. The F1 score combines these two metrics to provide a balanced evaluation of the model's ability to correctly identify positive samples while minimizing false positives and false negatives. It is no surprise, as the dataset is very unbalanced on the negative class, that the value of TPR is very low and TNR is almost equal to 1, as well as an average value for precision and a low one for recall, which result in the F1 score very close to zero. We will see later how this situation changes with the balanced dataset.

```
### MODEL VALIDATION

pred_u <- predict(glm10u, dati_test[,-1], type = "response")

# threshold = 0.4
pred_u_4 <- ifelse(pred_u >= 0.4, 1, 0)
mean(pred_u_4)
```

```
## [1] 0.02725487
```

```
# threshold = 0.5
pred_u_5 <- ifelse(pred_u >= 0.5, 1, 0)
mean(pred_u_5)
```

```
## [1] 0.01155737
```

```
# threshold = 0.6
pred_u_6 <- ifelse(pred_u >= 0.6, 1, 0)
mean(pred_u_6)
```

```
## [1] 0.004252711
```

```
# the best threshold is 0.5
```

```r
# Confusion matrix
conf_mat <- table(dati_test$HeartDisease, pred_u_5)
conf_mat
```

```
##    pred_u_5
##         0     1
##   0 72701   403
##   1  6324   521
```

```r
# true positive rate
TPR <- conf_mat[2, 2]/sum(conf_mat[2, ])
TPR
```

```
## [1] 0.07611395
```

```r
# true negative rate
TNR <- conf_mat[1, 1]/sum(conf_mat[1, ])
TNR
```

```
## [1] 0.9944873
```

```r
# precision
PREC <- conf_mat[2, 2]/sum(conf_mat[, 2])
PREC
```

```
## [1] 0.5638528
```

```r
# recall
REC <- conf_mat[2, 2]/sum(conf_mat[2, ])
REC
```

```
## [1] 0.07611395
```

```r
# F1 score
F1_glmu <- 2 * (PREC * REC)/(PREC + REC)
F1_glmu
```

```
## [1] 0.1341228
```

We now compute the validation of the model without interactions. The results are very similar to the best model selected, but with a lower F1 score.

```r
# we look now at the complete model
pred_uc <- predict(glm1u, dati_test[,-1], type = "response")

# threshold = 0.4
pred_uc_4 <- ifelse(pred_uc >= 0.4, 1, 0)
mean(pred_uc_4)
```

```
## [1] 0.02905602
```

```r
# threshold = 0.5
pred_uc_5 <- ifelse(pred_uc >= 0.5, 1, 0)
mean(pred_uc_5)
```

```
## [1] 0.0155224
```

```r
# threshold = 0.6
pred_uc_6 <- ifelse(pred_uc >= 0.6, 1, 0)
mean(pred_uc_6)
```

```
## [1] 0.007629864
```

```
# the best threshold is 0.6
```

```
# Confusion matrix
conf_mat <- table(dati_test$HeartDisease, pred_uc_6)
conf_mat
```

```
##    pred_uc_6
##         0     1
##   0 72845   259
##   1  6494   351
```

```
# true positive rate
TPR <- conf_mat[2, 2]/sum(conf_mat[2, ])
TPR
```

```
## [1] 0.05127831
```

```
# true negative rate
TNR <- conf_mat[1, 1]/sum(conf_mat[1, ])
TNR
```

```
## [1] 0.9964571
```

```
# precision
PREC <- conf_mat[2, 2]/sum(conf_mat[, 2])
PREC
```

```
## [1] 0.5754098
```

```
# recall
REC <- conf_mat[2, 2]/sum(conf_mat[2, ])
REC
```

```
## [1] 0.05127831
```

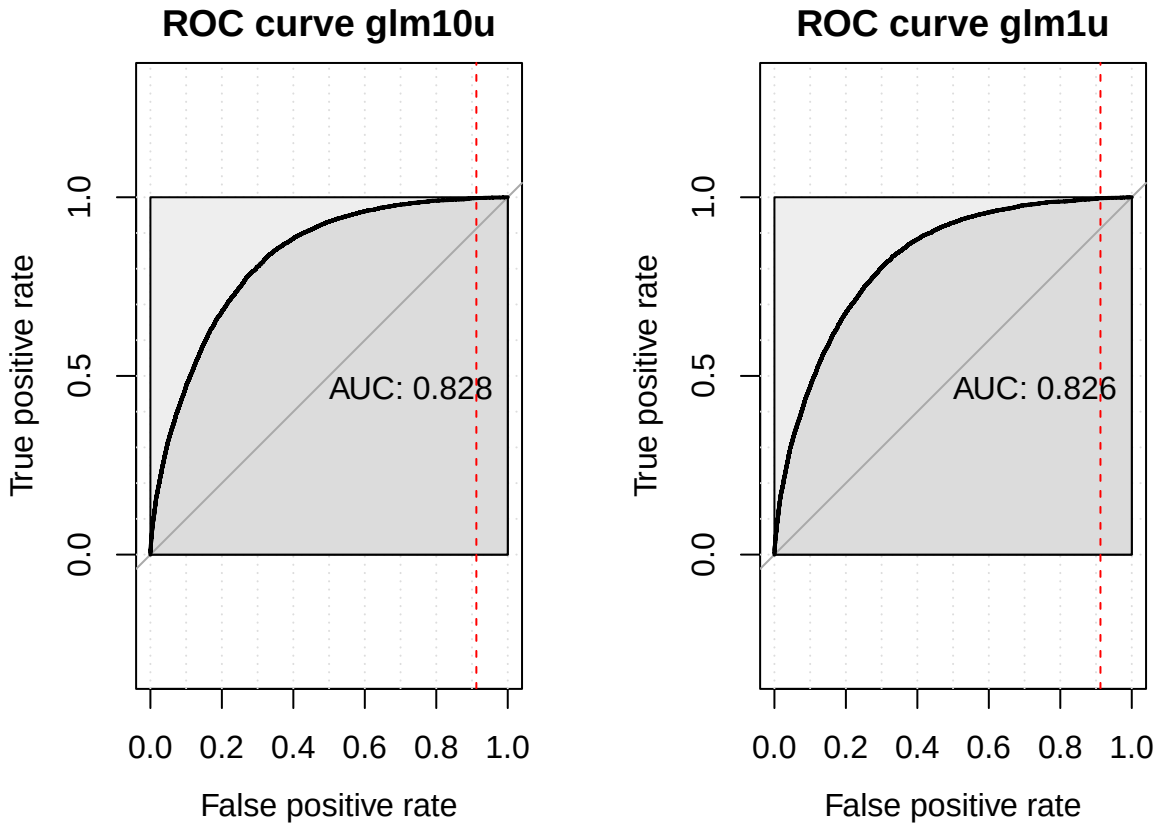
```
# F1 score
F1 <- 2 * (PREC * REC)/(PREC + REC)
F1
```

```
## [1] 0.09416499
```

We now explore the ROC curve of the prediction of the *glm10u* and *glm1u* models. The ROC curve and AUC summarize the performance of the model in predicting positive and negative instances. The ROC curve plots the true positive rate against the false positive rate, providing a visual representation of the model's discrimination ability. The AUC is a single metric that quantifies the overall performance of the model based on the ROC curve. It helps evaluate and compare the predictive accuracy of different models. Since the first model has a slightly lower F1 score than the *glm10u* model and lower AIC as well as lower AUC, we decide to stick with the *glm10u* model, as it indicates a better trade-off between the model's goodness-of-fit and its complexity.

```
# ROC curve
library(pROC)
par(mfrow=c(1,2))
roc_u <- roc(dati_test$HeartDisease, pred_u)
auc_u <- auc(roc_u)
plot(roc_u, main = "ROC curve glm10u", print.auc = TRUE, auc.polygon = TRUE,
     max.auc.polygon = TRUE, grid = TRUE, legacy.axes = TRUE,
     xlab = "False positive rate", ylab = "True positive rate")
coords <- coords(roc_u, "best", best.method = "closest.topleft")
```

```r
opt_thr <- coords$threshold
abline(v = opt_thr, col = "red", lty = 2)

roc_uc <- roc(dati_test$HeartDisease, pred_uc)
auc_uc <- auc(roc_uc)
plot(roc_uc, main = "ROC curve glm1u", print.auc = TRUE, auc.polygon = TRUE,
     max.auc.polygon = TRUE, grid = TRUE, legacy.axes = TRUE,
     xlab = "False positive rate", ylab = "True positive rate")
coords <- coords(roc_uc, "best", best.method = "closest.topleft")
opt_thr <- coords$threshold
abline(v = opt_thr, col = "red", lty = 2)
```



Balanced dataset

Next, we examine the glm models using the balanced dataset and establish the complete model. By reviewing the summary, it is evident that the independent variable *Race* is not statistically significant. Consequently, it will be eliminated during the stepwise selection process. The *VIF* output indicates that all the values are close to 1, suggesting the absence of collinearity among the variables.

```r
# Complete model definition and summary

glm1 <- glm(HeartDisease ~ ., data = dati_balanced_train, family = binomial(link = "logit"))
summary(glm1)
```

```
##
## Call:
## glm(formula = HeartDisease ~ ., family = binomial(link = "logit"),
##     data = dati_balanced_train)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8194  -0.8529   0.0600   0.8243   3.1728
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       4.143701   0.107020  38.719  < 2e-16 ***
## BMI              -0.017067   0.002014  -8.475  < 2e-16 ***
## Smoking          -0.469911   0.024432 -19.233  < 2e-16 ***
## AlcoholDrinking   0.216108   0.053480   4.041 5.32e-05 ***
## Stroke           -1.336971   0.052622 -25.407  < 2e-16 ***
## PhysicalHealth   -0.026324   0.001494 -17.618  < 2e-16 ***
## MentalHealth     -0.011232   0.001608  -6.985 2.84e-12 ***
## DiffWalking      -0.385275   0.033497 -11.502  < 2e-16 ***
## Sex              -0.742805   0.024893 -29.840  < 2e-16 ***
## AgeCategory      -0.297219   0.004797 -61.961  < 2e-16 ***
## Race             -0.005072   0.008672  -0.585 0.558639
## Diabetic         -0.679220   0.031453 -21.594  < 2e-16 ***
## PhysicalActivity  0.085148   0.028543   2.983 0.002853 **
## GenHealth         0.030880   0.009200   3.357 0.000789 ***
## SleepTime         0.021121   0.007848   2.691 0.007115 **
## Asthma           -0.451186   0.035104 -12.853  < 2e-16 ***
## KidneyDisease    -0.764638   0.053450 -14.306  < 2e-16 ***
## SkinCancer       -0.166115   0.036263  -4.581 4.63e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 56916  on 41055  degrees of freedom
## Residual deviance: 41899  on 41038  degrees of freedom
## AIC: 41935
##
## Number of Fisher Scoring iterations: 5
```

```
# VIF of the complete model
library(car)
vif(glm1)
```

```
##              BMI          Smoking  AlcoholDrinking           Stroke
##         1.128601         1.041268         1.023404         1.014725
##   PhysicalHealth     MentalHealth      DiffWalking              Sex
##         1.322279         1.225235         1.329145         1.077139
##      AgeCategory             Race         Diabetic PhysicalActivity
##         1.274593         1.060056         1.083889         1.137474
##        GenHealth        SleepTime           Asthma    KidneyDisease
##         1.017174         1.043478         1.062542         1.028040
##       SkinCancer
##         1.066323
```

Stepwise selection

The following step involves the stepwise selection. To avoid excessive length, the complete stepwise selection process, as in the addition and removal of non-significant interactions, was not included. Variables were

added or removed based on statistical criteria such as p-values or information criteria (*AIC*), which help evaluate their significance. In the initial model, the variable *Race* was included, but since it did not contribute significantly to the model's predictive power, it was removed. To enhance the model's performance, several variables and their interactions were added. These included *BMI* with *DiffWalking* and *GenHealth*. Further improvements were made by incorporating additional interactions such as *Smoking* with *AgeCategory* and *GenHealth*. Other interactions of the following variables were considered: *Stroke* , *PhysicalHealth*, *AgeCategory*, *DiffWalking*, *MentalHealth*,*Sex*, *GenHealth*,*SleepTime* and *Asthma*. In glm11, *AgeCategory* * *Asthma*, *PhysicalHealth* * *AgeCategory* and *MentalHealth* * *Sex* are removed. This operation has been done because these interactions do not have a very high significance. We then compared the AIC values of each model and selected the best one, which was the glm10 model.

```
glm2 <- update(glm1, . ~ . - Race)

glm3 <- update(glm2, . ~ . + BMI * DiffWalking + BMI * GenHealth)

glm4 <- update(glm3, . ~ . + Smoking * AgeCategory + Smoking * GenHealth)

glm5 <- update(glm4, . ~ . + Stroke * PhysicalHealth + Stroke * AgeCategory +
                 Stroke * GenHealth + Stroke * PhysicalHealth + Stroke * DiffWalking)

glm6 <- update(glm5, . ~ . + PhysicalHealth * DiffWalking +
                 PhysicalHealth * MentalHealth +
                 PhysicalHealth * AgeCategory + PhysicalHealth*AgeCategory)

glm7 <- update(glm6, . ~ . + MentalHealth * Sex + MentalHealth * GenHealth)

glm8 <- update(glm7, . ~ . + DiffWalking * AgeCategory + DiffWalking * Sex)

glm9 <- update(glm8, . ~ . + Sex * AgeCategory)

glm10 <- update(glm9, . ~ . + AgeCategory * Asthma  + AgeCategory * Smoking)

glm11 <- update(glm10, . ~ . - AgeCategory * Asthma  - PhysicalHealth * AgeCategory -
                  MentalHealth * Sex)
```

```
# compare the models' AIC
AIC(glm1, glm2, glm3, glm4, glm5, glm6, glm7, glm8, glm9, glm10, glm11)
```

```
##       df      AIC
## glm1  18 41934.62
## glm2  17 41932.96
## glm3  19 41919.15
## glm4  21 41898.78
## glm5  25 41744.75
## glm6  28 41666.41
## glm7  30 41613.70
## glm8  32 41555.88
## glm9  33 41521.20
## glm10 34 41517.95
## glm11 26 44341.00
```

```
summary(glm10)
```

```
##
## Call:
## glm(formula = HeartDisease ~ BMI + Smoking + AlcoholDrinking +
```

```
##     Stroke + PhysicalHealth + MentalHealth + DiffWalking + Sex +
##     AgeCategory + Diabetic + PhysicalActivity + GenHealth + SleepTime +
##     Asthma + KidneyDisease + SkinCancer + BMI:DiffWalking + BMI:GenHealth +
##     Smoking:AgeCategory + Smoking:GenHealth + Stroke:PhysicalHealth +
##     Stroke:AgeCategory + Stroke:GenHealth + Stroke:DiffWalking +
##     PhysicalHealth:DiffWalking + PhysicalHealth:MentalHealth +
##     PhysicalHealth:AgeCategory + MentalHealth:Sex + MentalHealth:GenHealth +
##     DiffWalking:AgeCategory + DiffWalking:Sex + Sex:AgeCategory +
##     AgeCategory:Asthma, family = binomial(link = "logit"), data = dati_balanced_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.95696  -0.82269   0.01688   0.80864   2.89635
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  4.0425794  0.1819365  22.220  < 2e-16 ***
## BMI                         -0.0089193  0.0048559  -1.837 0.066241 .
## Smoking                     -0.6035919  0.1007790  -5.989 2.11e-09 ***
## AlcoholDrinking              0.2035753  0.0539802   3.771 0.000162 ***
## Stroke                      -3.1266133  0.2412409 -12.961  < 2e-16 ***
## PhysicalHealth              -0.0511204  0.0054429  -9.392  < 2e-16 ***
## MentalHealth                 0.0011219  0.0040649   0.276 0.782557
## DiffWalking                 -2.1507396  0.2051358 -10.484  < 2e-16 ***
## Sex                         -0.3276522  0.0933357  -3.510 0.000447 ***
## AgeCategory                 -0.3127023  0.0086321 -36.225  < 2e-16 ***
## Diabetic                    -0.6611399  0.0314548 -21.019  < 2e-16 ***
## PhysicalActivity             0.0931575  0.0286095   3.256 0.001129 **
## GenHealth                    0.2031401  0.0475331   4.274 1.92e-05 ***
## SleepTime                    0.0172698  0.0078314   2.205 0.027440 *
## Asthma                      -0.6441605  0.1065185  -6.047 1.47e-09 ***
## KidneyDisease               -0.7492236  0.0532009 -14.083  < 2e-16 ***
## SkinCancer                  -0.1571858  0.0361697  -4.346 1.39e-05 ***
## BMI:DiffWalking              0.0200368  0.0041033   4.883 1.04e-06 ***
## BMI:GenHealth               -0.0048165  0.0016543  -2.911 0.003597 **
## Smoking:AgeCategory          0.0295339  0.0088641   3.332 0.000863 ***
## Smoking:GenHealth           -0.0433852  0.0188070  -2.307 0.021062 *
## Stroke:PhysicalHealth        0.0114470  0.0050131   2.283 0.022406 *
## Stroke:AgeCategory           0.1919962  0.0197369   9.728  < 2e-16 ***
## Stroke:GenHealth            -0.1160772  0.0477735  -2.430 0.015110 *
## Stroke:DiffWalking           0.3561807  0.1147658   3.104 0.001912 **
## PhysicalHealth:DiffWalking   0.0216823  0.0029428   7.368 1.73e-13 ***
## PhysicalHealth:MentalHealth  0.0006004  0.0001263   4.755 1.99e-06 ***
## PhysicalHealth:AgeCategory   0.0011896  0.0005468   2.175 0.029597 *
## MentalHealth:Sex             0.0074319  0.0030734   2.418 0.015601 *
## MentalHealth:GenHealth      -0.0072771  0.0012860  -5.659 1.52e-08 ***
## DiffWalking:AgeCategory      0.0831020  0.0129454   6.419 1.37e-10 ***
## DiffWalking:Sex              0.3100698  0.0598766   5.178 2.24e-07 ***
## Sex:AgeCategory             -0.0548747  0.0093015  -5.900 3.64e-09 ***
## AgeCategory:Asthma           0.0265878  0.0115263   2.307 0.021071 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 56916  on 41055  degrees of freedom
## Residual deviance: 41450  on 41022  degrees of freedom
## AIC: 41518
##
## Number of Fisher Scoring iterations: 5
```

In parallel to what we did for the unbalanced dataset, after selecting the best model we proceed with the model validation. The selection of the best threshold is based on the one that balances the trade-off between precision and recall and the one that maximizes the F1 score. The F1 score is now increased of almost 1%.

```
### MODEL VALIDATION

pred_b <- predict(glm10, dati_test[,-1], type = "response")

# threshold = 0.1
pred_b_1 <- ifelse(pred_b >= 0.1, 1, 0)
mean(pred_b_1)
```

```
## [1] 0.9798622
```

```
# threshold = 0.2
pred_b_2 <- ifelse(pred_b >= 0.2, 1, 0)
mean(pred_b_2)
```

```
## [1] 0.9249146
```

```
# threshold = 0.3
pred_b_3 <- ifelse(pred_b >= 0.3, 1, 0)
mean(pred_b_3)
```

```
## [1] 0.8509675
```

```
# threshold = 0.4
pred_b_4 <- ifelse(pred_b >= 0.4, 1, 0)
mean(pred_b_4)
```

```
## [1] 0.7721923
```

```
# threshold = 0.5
pred_b_5 <- ifelse(pred_b >= 0.5, 1, 0)
mean(pred_b_5)
```

```
## [1] 0.6863125
```

```
# the best threshold is 0.1
```

```
# Confusion matrix
conf_mat <- table(dati_test$HeartDisease, pred_b_1)
conf_mat
```

```
##    pred_b_1
##         0     1
##   0   817 72287
##   1   793  6052
```

```
# true positive rate
TPR <- conf_mat[2, 2]/sum(conf_mat[2, ])
TPR
```

```
## [1] 0.884149
```

```r
# true negative rate
TNR <- conf_mat[1, 1]/sum(conf_mat[1, ])
TNR
```

```
## [1] 0.01117586
```

```r
# precision
PREC <- conf_mat[2, 2]/sum(conf_mat[, 2])
PREC
```

```
## [1] 0.07725399
```

```r
# recall
REC <- conf_mat[2, 2]/sum(conf_mat[2, ])
REC
```

```
## [1] 0.884149
```

```r
# F1 score
F1_glmb <- 2 * (PREC * REC)/(PREC + REC)
F1_glmb
```

```
## [1] 0.1420924
```

We now compute the model validation of the *glm1* model. The results are very similar to the best model selected, and the F1 score is slightly lower.

```r
# we look now at the complete model
pred_bc <- predict(glm1, dati_test[,-1], type = "response")

# threshold = 0.1
pred_bc_1 <- ifelse(pred_bc >= 0.1, 1, 0)
mean(pred_b_1)
```

```
## [1] 0.9798622
```

```r
# threshold = 0.2
pred_bc_2 <- ifelse(pred_bc >= 0.2, 1, 0)
mean(pred_bc_2)
```

```
## [1] 0.9266908
```

```r
# threshold = 0.3
pred_bc_3 <- ifelse(pred_bc >= 0.3, 1, 0)
mean(pred_bc_3)
```

```
## [1] 0.8628501
```

```r
# threshold = 0.4
pred_bc_4 <- ifelse(pred_bc >= 0.4, 1, 0)
mean(pred_bc_4)
```

```
## [1] 0.7864013
```

```r
# threshold = 0.5
pred_bc_5 <- ifelse(pred_bc >= 0.5, 1, 0)
mean(pred_bc_5)
```

```
## [1] 0.6954934
```

```r
# the best threshold is 0.1
```

```r
# Confusion matrix
conf_mat <- table(dati_test$HeartDisease, pred_bc_1)
conf_mat
```

```
##    pred_bc_1
##         0     1
##   0  1108 71996
##   1   998  5847
```

```r
# true positive rate
TPR <- conf_mat[2, 2]/sum(conf_mat[2, ])
TPR
```

```
## [1] 0.8542001
```

```r
# true negative rate
TNR <- conf_mat[1, 1]/sum(conf_mat[1, ])
TNR
```

```
## [1] 0.01515649
```

```r
# precision
PREC <- conf_mat[2, 2]/sum(conf_mat[, 2])
PREC
```

```
## [1] 0.07511273
```

```r
# recall
REC <- conf_mat[2, 2]/sum(conf_mat[2, ])
REC
```

```
## [1] 0.8542001
```

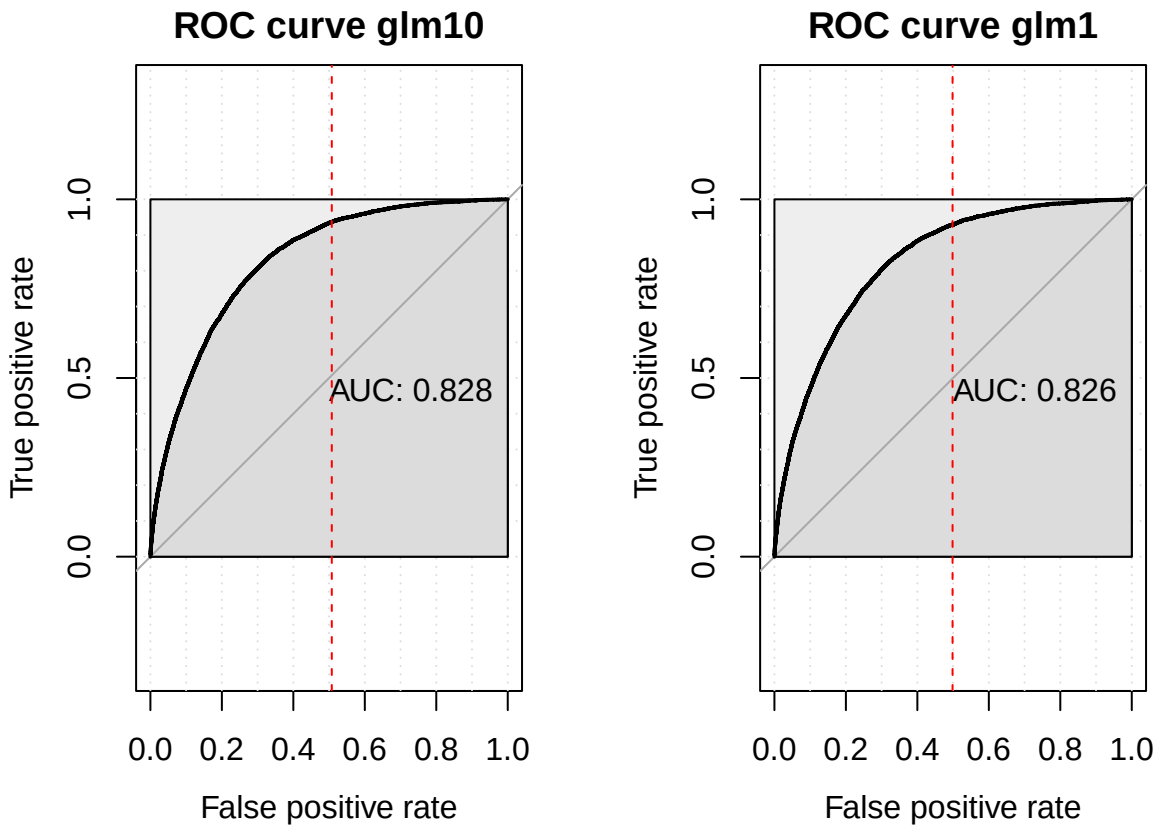
```r
# F1 score
F1 <- 2 * (PREC * REC)/(PREC + REC)
F1
```

```
## [1] 0.1380833
```

We now explore the ROC curve of the prediction of the *glm10* and *glm1* models. By looking at the F1 scores, the AIC and the AUC values, we are confident to choose *glm10* as the best model.

```r
# ROC curve
library(pROC)
par(mfrow=c(1,2))
roc_b <- roc(dati_test$HeartDisease, pred_b)
auc_b <- auc(roc_b)
plot(roc_b, main = "ROC curve glm10", print.auc = TRUE,
     auc.polygon = TRUE, max.auc.polygon = TRUE, grid = TRUE, legacy.axes = TRUE,
     xlab = "False positive rate", ylab = "True positive rate")
coords <- coords(roc_b, "best", best.method = "closest.topleft")
opt_thr <- coords$threshold
abline(v = opt_thr, col = "red", lty = 2)

roc_bc <- roc(dati_test$HeartDisease, pred_bc)
auc_bc <- auc(roc_bc)
plot(roc_bc, main = "ROC curve glm1", print.auc = TRUE,
     auc.polygon = TRUE, max.auc.polygon = TRUE, grid = TRUE, legacy.axes = TRUE,
     xlab = "False positive rate", ylab = "True positive rate")
coords <- coords(roc_bc, "best", best.method = "closest.topleft")
```

```r
opt_thr <- coords$threshold
abline(v = opt_thr, col = "red", lty = 2)
```



Discriminant Analysis

Discriminant Analysis is a technique used to analyze and classify data into distinct groups or categories based on a set of predictor variables. It aims at identifying the decision boundaries that separate different groups or classes in the dataset. It achieves this by maximizing the between-class variance and minimizing the within-class variance. Discriminant Analysis encompasses two main variants: Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA), that we will both apply to our unbalanced and balanced datasets.

Linear Discriminant Analysis

We begin the discriminant analysis with the LDA. It assumes that the predictor variables have a multivariate normal distribution within each class and that the covariance matrices are equal across all classes. It projects the data onto a lower-dimensional space while maximizing the ratio of between-class scatter to within-class scatter, leading to a linear decision boundary.

LDA on unbalanced dataset

For this LDA model, we will consider the variables and interactions from the GLM analysis that we have previously examined, as they have shown improved results, in particular thoes from the *glm10u* model. This integration of information aims to maximize the discriminatory power of the model and provide us with a robust framework for accurately classifying new observations in our binary response variable dataset.

```r
library(MASS)
lda_u <- lda(HeartDisease ~ BMI + Smoking + AlcoholDrinking +
               Stroke + PhysicalHealth + MentalHealth + DiffWalking + Sex +
               AgeCategory + Diabetic + PhysicalActivity + SleepTime + Asthma +
               KidneyDisease + SkinCancer + BMI:Diabetic + BMI:AlcoholDrinking +
               Smoking:KidneyDisease + Smoking:AgeCategory + Stroke:DiffWalking +
               Stroke:AgeCategory + Smoking:Stroke + PhysicalHealth:DiffWalking +
               PhysicalHealth:MentalHealth + PhysicalHealth:AgeCategory +
               PhysicalHealth:Diabetic + MentalHealth:Sex + DiffWalking:AgeCategory +
               DiffWalking:Sex + Sex:AgeCategory + AgeCategory:Diabetic +
               AgeCategory:Asthma + AgeCategory:PhysicalActivity +
               AgeCategory:KidneyDisease, data = dati_train, family="binomial")
lda_u
```

```
## Call:
## lda(HeartDisease ~ BMI + Smoking + AlcoholDrinking + Stroke +
##     PhysicalHealth + MentalHealth + DiffWalking + Sex + AgeCategory +
##     Diabetic + PhysicalActivity + SleepTime + Asthma + KidneyDisease +
##     SkinCancer + BMI:Diabetic + BMI:AlcoholDrinking + Smoking:KidneyDisease +
##     Smoking:AgeCategory + Stroke:DiffWalking + Stroke:AgeCategory +
##     Smoking:Stroke + PhysicalHealth:DiffWalking + PhysicalHealth:MentalHealth +
##     PhysicalHealth:AgeCategory + PhysicalHealth:Diabetic + MentalHealth:Sex +
##     DiffWalking:AgeCategory + DiffWalking:Sex + Sex:AgeCategory +
##     AgeCategory:Diabetic + AgeCategory:Asthma + AgeCategory:PhysicalActivity +
##     AgeCategory:KidneyDisease, data = dati_train, family = "binomial")
##
## Prior probabilities of groups:
##          0          1
## 0.91441175 0.08558825
##
## Group means:
##        BMI   Smoking AlcoholDrinking     Stroke PhysicalHealth MentalHealth
## 0 28.22466 0.3950246       0.0706326 0.02602614       2.956328     3.823001
## 1 29.40018 0.5868570       0.0418940 0.15944076       7.832570     4.633281
##   DiffWalking       Sex AgeCategory  Diabetic PhysicalActivity SleepTime
## 0   0.1177012 0.4646358    7.257361 0.1091611        0.7880384  7.094643
## 1   0.3644778 0.5922642   10.240403 0.3282346        0.6383476  7.139273
##      Asthma KidneyDisease SkinCancer BMI:Diabetic BMI:AlcoholDrinking
## 0 0.1298480    0.02833785 0.08487675     3.466216            1.931563
## 1 0.1821415    0.12504871 0.18155690    10.361335            1.210194
##   Smoking:KidneyDisease Smoking:AgeCategory Stroke:DiffWalking
## 0            0.01322281            3.063957         0.01009949
## 1            0.07185308            5.968580         0.08539556
##   Stroke:AgeCategory Smoking:Stroke PhysicalHealth:DiffWalking
## 0          0.2559753     0.01371524                   1.319098
## 1          1.6449240     0.10069174                   5.146775
##   PhysicalHealth:MentalHealth PhysicalHealth:AgeCategory
## 0                    27.56564                   23.77618
## 1                    73.51174                   78.45611
##   PhysicalHealth:Diabetic MentalHealth:Sex DiffWalking:AgeCategory
## 0               0.6215222         1.390196                1.110301
## 1               3.2929170         2.123344                3.813231
##   DiffWalking:Sex Sex:AgeCategory AgeCategory:Diabetic AgeCategory:Asthma
## 0      0.04264584        3.207603             1.008736          0.8653553
```

```
## 1      0.17970577        6.051101             3.387617          1.7392829
##   AgeCategory:PhysicalActivity AgeCategory:KidneyDisease
## 0                     5.562102                  0.267347
## 1                     6.496882                  1.323022
##
## Coefficients of linear discriminants:
##                                        LD1
## BMI                           1.964142e-03
## Smoking                      -1.814618e-01
## AlcoholDrinking              -3.807080e-01
## Stroke                        1.845165e+00
## PhysicalHealth               -1.631714e-03
## MentalHealth                  6.003213e-03
## DiffWalking                  -1.594323e-01
## Sex                          -2.811745e-01
## AgeCategory                   5.249283e-02
## Diabetic                     -2.552818e-01
## PhysicalActivity              4.762441e-02
## SleepTime                    -1.159331e-02
## Asthma                        7.965540e-02
## KidneyDisease                 4.298959e-02
## SkinCancer                    2.270411e-01
## BMI:Diabetic                  4.891828e-04
## BMI:AlcoholDrinking           8.242151e-03
## Smoking:KidneyDisease         1.424638e-01
## Smoking:AgeCategory           5.660916e-02
## Stroke:DiffWalking            2.116165e-01
## Stroke:AgeCategory           -2.569500e-02
## Smoking:Stroke                4.016817e-01
## PhysicalHealth:DiffWalking   -8.028798e-03
## PhysicalHealth:MentalHealth  -2.695009e-05
## PhysicalHealth:AgeCategory    3.091611e-03
## PhysicalHealth:Diabetic       1.667132e-02
## MentalHealth:Sex              2.095779e-03
## DiffWalking:AgeCategory       6.823685e-02
## DiffWalking:Sex               2.079033e-01
## Sex:AgeCategory               9.908738e-02
## AgeCategory:Diabetic          9.286061e-02
## AgeCategory:Asthma            2.463327e-02
## AgeCategory:PhysicalActivity -1.775098e-02
## AgeCategory:KidneyDisease     9.694537e-02
```

We now proceed on using the model and the test set for predicting the data. We first compute the error rate, then we look for the best threshold that gave the lower percentage of misclassified data. After selecting 0.6 as threshold, we compute the confusion matrix as well as the true positive rate, true negative rate and F1 score. With respect to the *glm10u* model there has been an improvement, but it's still very low. We can also notice how the AUC is lower than the one of the GLM model.

```
# predictions
lda_u.pred <- predict(lda_u, dati_test[,-1])
lda_u.post=lda_u.pred$posterior
lda_u.class <- lda_u.pred$class

# error rate
mean(lda_u.class != dati_test$HeartDisease) * 100
```

```
## [1] 9.456028
```

```
# threshold = 0.4
lda_u.pred4=as.factor(ifelse(lda_u.post[,2] >= 0.4, 1, 0))
mean(lda_u.pred4!=dati_test$HeartDisease)
```

```
## [1] 0.1000638
```

```
# threshold = 0.5
lda_u.pred5=as.factor(ifelse(lda_u.post[,2] >= 0.5, 1, 0))
mean(lda_u.pred5!=dati_test$HeartDisease)
```

```
## [1] 0.09456028
```

```
# threshold = 0.6
lda_u.pred6=as.factor(ifelse(lda_u.post[,2] >= 0.6, 1, 0))
mean(lda_u.pred6!=dati_test$HeartDisease)
```

```
## [1] 0.09072033
```

```
# the best threshold is 0.6
```

```
# Confusion matrix
conf_mat <- table(dati_test$HeartDisease, lda_u.pred6)
conf_mat
```

```
##    lda_u.pred6
##         0     1
##   0 71296  1808
##   1  5445  1400
```

```
# true positive rate
TPR <- conf_mat[2, 2]/sum(conf_mat[2, ])
TPR
```

```
## [1] 0.2045289
```

```
# true negative rate
TNR <- conf_mat[1, 1]/sum(conf_mat[1, ])
TNR
```

```
## [1] 0.9752681
```

```
# precision
PREC <- conf_mat[2, 2]/sum(conf_mat[, 2])
PREC
```

```
## [1] 0.436409
```

```
# recall
REC <- conf_mat[2, 2]/sum(conf_mat[2, ])
REC
```

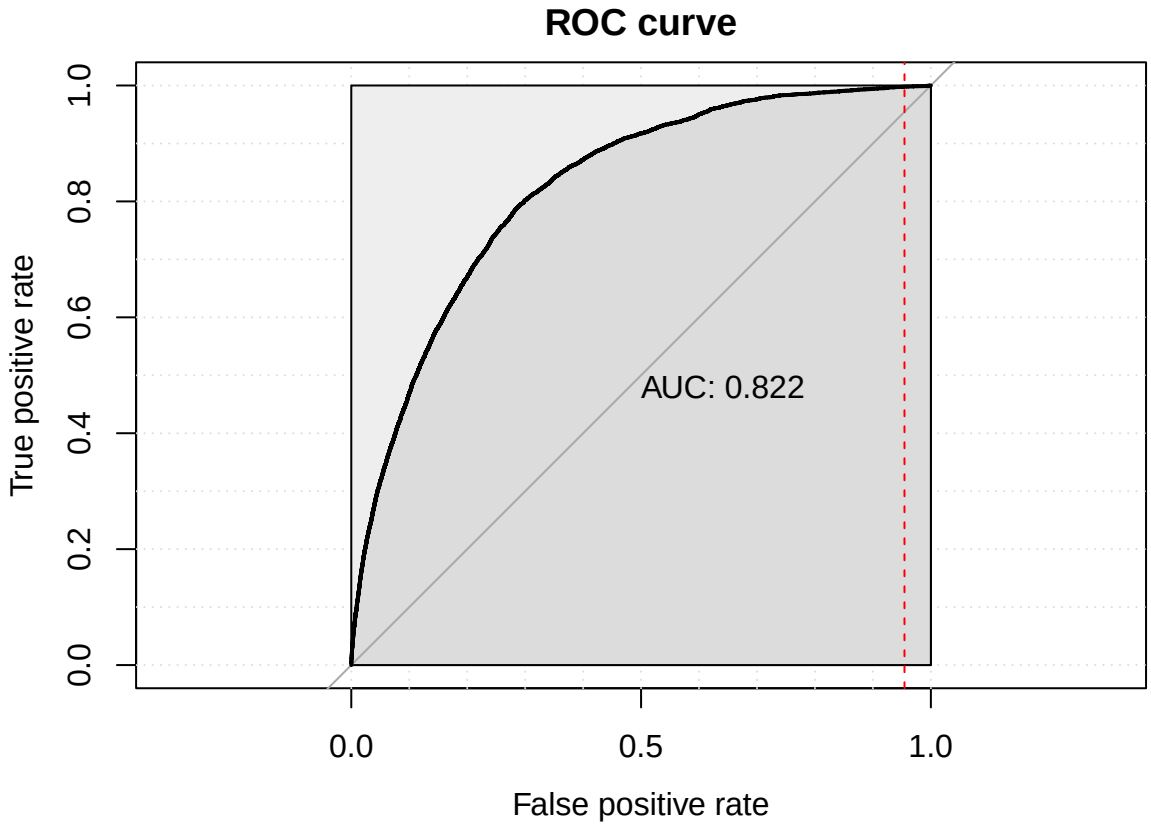
```
## [1] 0.2045289
```

```
# F1 score
F1_ldau <- 2 * (PREC * REC)/(PREC + REC)
F1_ldau
```

```
## [1] 0.2785238
```

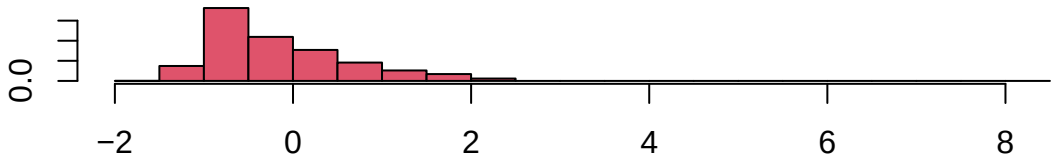
```
# ROC curve
library(pROC)
```

```
roc_u <- roc(dati_test$HeartDisease, lda_u.post[,2])
coords <- coords(roc_u, "best", best.method = "closest.topleft")
opt_thr <- coords$threshold
plot(roc_u, main = "ROC curve", print.auc = TRUE, auc.polygon = TRUE,
     max.auc.polygon = TRUE, grid = TRUE, legacy.axes = TRUE,
     xlab = "False positive rate", ylab = "True positive rate")
auc_u <- auc(roc_u)
abline(v = opt_thr, col = "red", lty = 2)
```

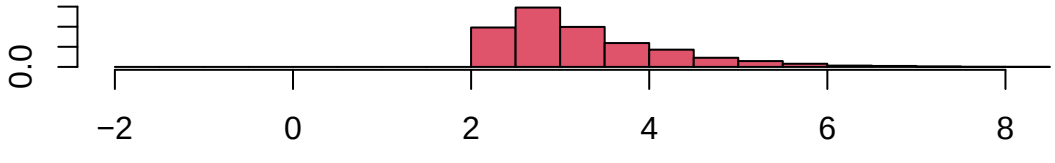


From the graph below, by taking the x values as linear combination of the variables that better describe the observations and as *class* the assigned class, we can notice how the LDA function can clearly discriminate the two classes, without overlaps.

```
ldahist(lda_u.pred$x[,1], g=lda_u.pred$class, col=2)
```



group 0



group 1

LDA on balanced dataset

We are now repeating the same steps for the balanced dataset. To define the LDA model we used the variables and interactions of the *glm10* model.

```
lda_b <- lda(formula = HeartDisease ~ BMI + Smoking + AlcoholDrinking +
    Stroke + PhysicalHealth + MentalHealth + DiffWalking + Sex +
    AgeCategory + Diabetic + PhysicalActivity + GenHealth + SleepTime +
    Asthma + KidneyDisease + SkinCancer + BMI:DiffWalking + BMI:GenHealth +
    Smoking:AgeCategory + Smoking:GenHealth + Stroke:PhysicalHealth +
    Stroke:AgeCategory + Stroke:GenHealth + Stroke:DiffWalking +
    PhysicalHealth:DiffWalking + PhysicalHealth:MentalHealth +
    PhysicalHealth:AgeCategory + MentalHealth:Sex + MentalHealth:GenHealth +
    DiffWalking:AgeCategory + DiffWalking:Sex + Sex:AgeCategory +
    AgeCategory:Asthma, data = dati_balanced_train, family="binomial")
lda_b
```

```
## Call:
## lda(HeartDisease ~ BMI + Smoking + AlcoholDrinking + Stroke +
##     PhysicalHealth + MentalHealth + DiffWalking + Sex + AgeCategory +
##     Diabetic + PhysicalActivity + GenHealth + SleepTime + Asthma +
##     KidneyDisease + SkinCancer + BMI:DiffWalking + BMI:GenHealth +
##     Smoking:AgeCategory + Smoking:GenHealth + Stroke:PhysicalHealth +
##     Stroke:AgeCategory + Stroke:GenHealth + Stroke:DiffWalking +
##     PhysicalHealth:DiffWalking + PhysicalHealth:MentalHealth +
##     PhysicalHealth:AgeCategory + MentalHealth:Sex + MentalHealth:GenHealth +
##     DiffWalking:AgeCategory + DiffWalking:Sex + Sex:AgeCategory +
##     AgeCategory:Asthma, data = dati_balanced_train, family = "binomial")
##
## Prior probabilities of groups:
##   0   1
## 0.5 0.5
##
```

```
## Group means:
##        BMI   Smoking AlcoholDrinking     Stroke PhysicalHealth MentalHealth
## 0 29.40018 0.5868570      0.04189400 0.15944076       7.832570     4.633281
## 1 28.21378 0.3948266      0.06931995 0.02572097       2.980076     3.885425
##   DiffWalking       Sex AgeCategory  Diabetic PhysicalActivity GenHealth
## 0   0.3644778 0.5922642   10.240403 0.3282346        0.6383476  2.605417
## 1   0.1201286 0.4621493    7.266124 0.1089244        0.7853663  2.645216
##   SleepTime    Asthma KidneyDisease SkinCancer BMI:DiffWalking BMI:GenHealth
## 0  7.139273 0.1821415    0.12504871  0.1815569       11.313819      76.80310
## 1  7.077260 0.1285074    0.02762081  0.0831060        3.742345      73.88757
##   Smoking:AgeCategory Smoking:GenHealth Stroke:PhysicalHealth
## 0            5.968580          1.549006             1.8087977
## 1            3.069417          1.010425             0.2022603
##   Stroke:AgeCategory Stroke:GenHealth Stroke:DiffWalking
## 0          1.6449240       0.44276111         0.08539556
## 1          0.2569661       0.06586126         0.01081450
##   PhysicalHealth:DiffWalking PhysicalHealth:MentalHealth
## 0                   5.146775                    73.51174
## 1                   1.383379                    28.75638
##   PhysicalHealth:AgeCategory MentalHealth:Sex MentalHealth:GenHealth
## 0                   78.45611         2.123344               13.42751
## 1                   23.96093         1.431557                9.88674
##   DiffWalking:AgeCategory DiffWalking:Sex Sex:AgeCategory AgeCategory:Asthma
## 0                3.813231      0.17970577        6.051101           1.739283
## 1                1.145265      0.04525526        3.202358           0.837539
##
## Coefficients of linear discriminants:
##                                       LD1
## BMI                         -0.0034610144
## Smoking                     -0.1059268005
## AlcoholDrinking              0.1722699771
## Stroke                      -2.7359748203
## PhysicalHealth              -0.0377805145
## MentalHealth                 0.0006324687
## DiffWalking                 -1.5776352210
## Sex                         -0.1057054740
## AgeCategory                 -0.1934063470
## Diabetic                    -0.5176309988
## PhysicalActivity             0.0862432351
## GenHealth                    0.1241523856
## SleepTime                    0.0072714841
## Asthma                      -0.3340385422
## KidneyDisease               -0.5003362376
## SkinCancer                  -0.1445924151
## BMI:DiffWalking              0.0114068814
## BMI:GenHealth               -0.0028615142
## Smoking:AgeCategory         -0.0166168707
## Smoking:GenHealth           -0.0224417916
## Stroke:PhysicalHealth        0.0155222423
## Stroke:AgeCategory           0.1597260353
## Stroke:GenHealth            -0.0331011233
## Stroke:DiffWalking           0.3912233480
## PhysicalHealth:DiffWalking   0.0173302806
## PhysicalHealth:MentalHealth  0.0004072301
```

```
## PhysicalHealth:AgeCategory   0.0006399980
## MentalHealth:Sex             0.0055918996
## MentalHealth:GenHealth      -0.0051406077
## DiffWalking:AgeCategory      0.0483266422
## DiffWalking:Sex              0.3863465446
## Sex:AgeCategory             -0.0642557237
## AgeCategory:Asthma           0.0057720551
```

We now proceed on using the model and the test set for predicting the data. After selecting 0.1 as threshold, we compute the confusion matrix as well as the true positive rate, true negative rate and F1 score. The results are quite similar to the ones of the *glm10* model, including the AUC value.

```
lda_b.pred <- predict(lda_b, dati_test[,-1])
lda_b.post=lda_b.pred$posterior
lda_b.class <- lda_b.pred$class

# error rate
mean(lda_b.class != dati_test$HeartDisease) * 100
```

```
## [1] 73.21668
```

```
# threshold = 0.1
lda_b.pred1=as.factor(ifelse(lda_b.post[,2] >= 0.1, 1, 0))
mean(lda_b.pred1!=dati_test$HeartDisease)
```

```
## [1] 0.9115686
```

```
# threshold = 0.6
lda_b.pred2=as.factor(ifelse(lda_b.post[,2] >= 0.2, 1, 0))
mean(lda_b.pred2!=dati_test$HeartDisease)
```

```
## [1] 0.883876
```

```
# threshold = 0.3
lda_b.pred3=as.factor(ifelse(lda_b.post[,2] >= 0.3, 1, 0))
mean(lda_b.pred3!=dati_test$HeartDisease)
```

```
## [1] 0.8406609
```

```
# threshold = 0.4
lda_b.pred4=as.factor(ifelse(lda_b.post[,2] >= 0.4, 1, 0))
mean(lda_b.pred4!=dati_test$HeartDisease)
```

```
## [1] 0.7908667
```

```
# threshold = 0.5
lda_b.pred5=as.factor(ifelse(lda_b.post[,2] >= 0.5, 1, 0))
mean(lda_b.pred5!=dati_test$HeartDisease)
```

```
## [1] 0.7321668
```

```
# the best threshold is 0.1
```

```
# Confusion matrix
conf_mat <- table(dati_test$HeartDisease, lda_b.pred1)
conf_mat
```

```
##    lda_b.pred1
##         0     1
##   0  1297 71807
##   1  1072  5773
```

```r
# true positive rate
TPR <- conf_mat[2, 2]/sum(conf_mat[2, ])
TPR
```

```
## [1] 0.8433893
```

```r
# true negative rate
TNR <- conf_mat[1, 1]/sum(conf_mat[1, ])
TNR
```

```
## [1] 0.01774185
```

```r
# precision
PREC <- conf_mat[2, 2]/sum(conf_mat[, 2])
PREC
```

```
## [1] 0.07441351
```

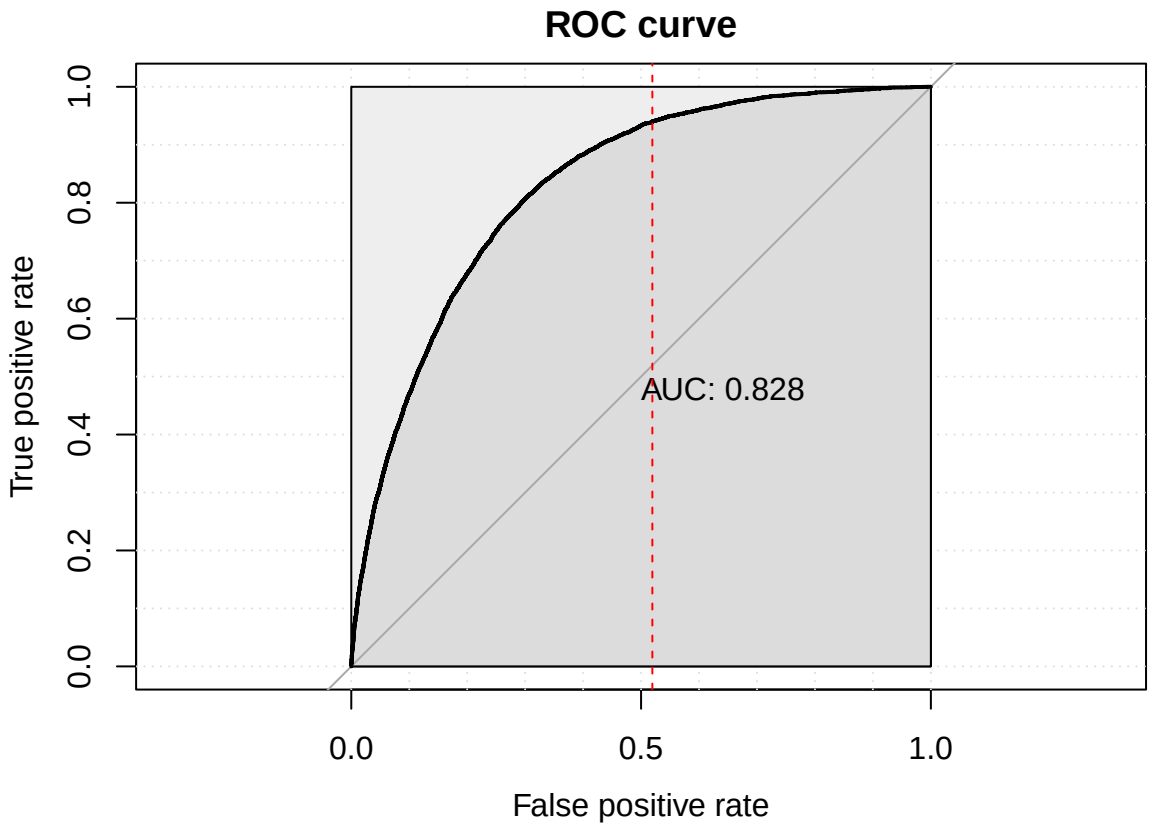
```r
# recall
REC <- conf_mat[2, 2]/sum(conf_mat[2, ])
REC
```

```
## [1] 0.8433893
```

```r
# F1 score
F1_ldab <- 2 * (PREC * REC)/(PREC + REC)
F1_ldab
```

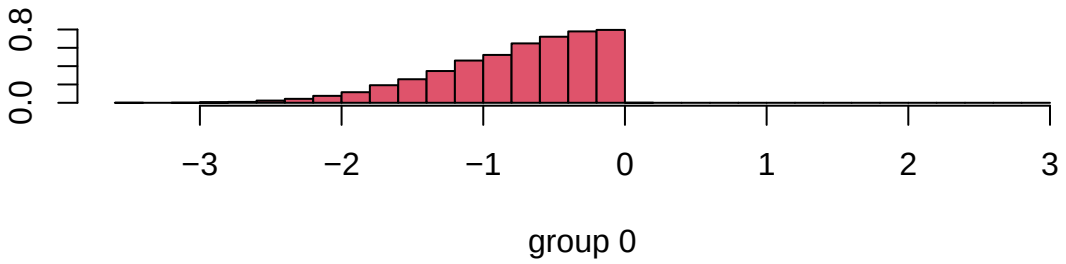
```
## [1] 0.1367604
```

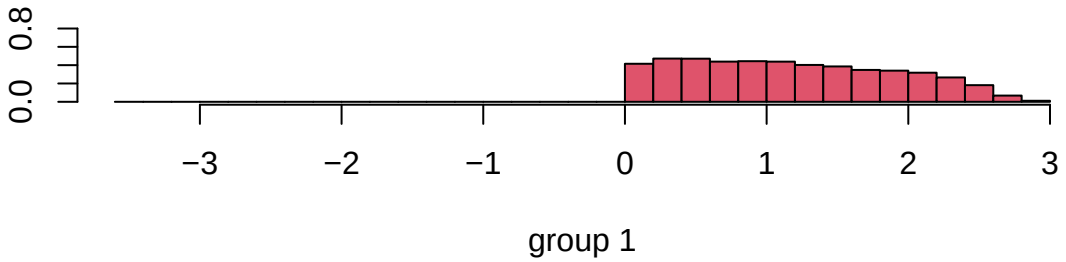
```r
# ROC curve
library(pROC)
roc_b <- roc(dati_test$HeartDisease, lda_b.post[,2])
auc_b <- auc(roc_b)
coords <- coords(roc_b, "best", best.method = "closest.topleft")
opt_thr <- coords$threshold
plot(roc_b, main = "ROC curve", print.auc = TRUE, auc.polygon = TRUE,
     max.auc.polygon = TRUE, grid = TRUE, legacy.axes = TRUE,
     xlab = "False positive rate", ylab = "True positive rate")
abline(v = opt_thr, col = "red", lty = 2)
```



From the graph below we can notice how the LDA function clearly discriminates the two classes, without overlaps.

```
ldahist(lda_b.pred$x[,1], g=lda_b.pred$class, col=2)
```





Quadratic Discriminant Analysis

The QDA relaxes the assumption of equal covariance matrices across classes and allows for different quadratic decision boundaries for each class. It can capture more complex relationships between the predictor variables and the response variable, at the cost of increased computational complexity.

QDA on unbalanced dataset

As we did for the LDA model, we are considering a model with the same variables and interactions of the best glm model.

```
qda_u <- lda(HeartDisease ~ BMI + Smoking + AlcoholDrinking +
               Stroke + PhysicalHealth + MentalHealth + DiffWalking + Sex +
               AgeCategory + Diabetic + PhysicalActivity + SleepTime + Asthma +
               KidneyDisease + SkinCancer + BMI:Diabetic + BMI:AlcoholDrinking +
               Smoking:KidneyDisease + Smoking:AgeCategory + Stroke:DiffWalking +
               Stroke:AgeCategory + Smoking:Stroke + PhysicalHealth:DiffWalking +
               PhysicalHealth:MentalHealth + PhysicalHealth:AgeCategory +
               PhysicalHealth:Diabetic + MentalHealth:Sex + DiffWalking:AgeCategory +
               DiffWalking:Sex + Sex:AgeCategory + AgeCategory:Diabetic +
               AgeCategory:Asthma + AgeCategory:PhysicalActivity +
               AgeCategory:KidneyDisease, data = dati_train, family="binomial")
qda_u
```

```
## Call:
## lda(HeartDisease ~ BMI + Smoking + AlcoholDrinking + Stroke +
##     PhysicalHealth + MentalHealth + DiffWalking + Sex + AgeCategory +
##     Diabetic + PhysicalActivity + SleepTime + Asthma + KidneyDisease +
##     SkinCancer + BMI:Diabetic + BMI:AlcoholDrinking + Smoking:KidneyDisease +
##     Smoking:AgeCategory + Stroke:DiffWalking + Stroke:AgeCategory +
##     Smoking:Stroke + PhysicalHealth:DiffWalking + PhysicalHealth:MentalHealth +
##     PhysicalHealth:AgeCategory + PhysicalHealth:Diabetic + MentalHealth:Sex +
##     DiffWalking:AgeCategory + DiffWalking:Sex + Sex:AgeCategory +
##     AgeCategory:Diabetic + AgeCategory:Asthma + AgeCategory:PhysicalActivity +
##     AgeCategory:KidneyDisease, data = dati_train, family = "binomial")
##
## Prior probabilities of groups:
##          0          1
## 0.91441175 0.08558825
##
## Group means:
##        BMI   Smoking AlcoholDrinking     Stroke PhysicalHealth MentalHealth
## 0 28.22466 0.3950246       0.0706326 0.02602614       2.956328     3.823001
## 1 29.40018 0.5868570       0.0418940 0.15944076       7.832570     4.633281
##   DiffWalking       Sex AgeCategory  Diabetic PhysicalActivity SleepTime
## 0   0.1177012 0.4646358    7.257361 0.1091611        0.7880384  7.094643
## 1   0.3644778 0.5922642   10.240403 0.3282346        0.6383476  7.139273
##      Asthma KidneyDisease SkinCancer BMI:Diabetic BMI:AlcoholDrinking
## 0 0.1298480    0.02833785 0.08487675     3.466216            1.931563
## 1 0.1821415    0.12504871 0.18155690    10.361335            1.210194
##   Smoking:KidneyDisease Smoking:AgeCategory Stroke:DiffWalking
## 0            0.01322281            3.063957         0.01009949
## 1            0.07185308            5.968580         0.08539556
##   Stroke:AgeCategory Smoking:Stroke PhysicalHealth:DiffWalking
## 0          0.2559753     0.01371524                   1.319098
```

```
## 1          1.6449240     0.10069174                   5.146775
##   PhysicalHealth:MentalHealth PhysicalHealth:AgeCategory
## 0                    27.56564                   23.77618
## 1                    73.51174                   78.45611
##   PhysicalHealth:Diabetic MentalHealth:Sex DiffWalking:AgeCategory
## 0               0.6215222         1.390196                1.110301
## 1               3.2929170         2.123344                3.813231
##   DiffWalking:Sex Sex:AgeCategory AgeCategory:Diabetic AgeCategory:Asthma
## 0      0.04264584        3.207603             1.008736          0.8653553
## 1      0.17970577        6.051101             3.387617          1.7392829
##   AgeCategory:PhysicalActivity AgeCategory:KidneyDisease
## 0                     5.562102                  0.267347
## 1                     6.496882                  1.323022
##
## Coefficients of linear discriminants:
##                                       LD1
## BMI                          1.964142e-03
## Smoking                     -1.814618e-01
## AlcoholDrinking             -3.807080e-01
## Stroke                       1.845165e+00
## PhysicalHealth              -1.631714e-03
## MentalHealth                 6.003213e-03
## DiffWalking                 -1.594323e-01
## Sex                         -2.811745e-01
## AgeCategory                  5.249283e-02
## Diabetic                    -2.552818e-01
## PhysicalActivity             4.762441e-02
## SleepTime                   -1.159331e-02
## Asthma                       7.965540e-02
## KidneyDisease                4.298959e-02
## SkinCancer                   2.270411e-01
## BMI:Diabetic                 4.891828e-04
## BMI:AlcoholDrinking          8.242151e-03
## Smoking:KidneyDisease        1.424638e-01
## Smoking:AgeCategory          5.660916e-02
## Stroke:DiffWalking           2.116165e-01
## Stroke:AgeCategory          -2.569500e-02
## Smoking:Stroke               4.016817e-01
## PhysicalHealth:DiffWalking  -8.028798e-03
## PhysicalHealth:MentalHealth -2.695009e-05
## PhysicalHealth:AgeCategory   3.091611e-03
## PhysicalHealth:Diabetic      1.667132e-02
## MentalHealth:Sex             2.095779e-03
## DiffWalking:AgeCategory      6.823685e-02
## DiffWalking:Sex              2.079033e-01
## Sex:AgeCategory              9.908738e-02
## AgeCategory:Diabetic         9.286061e-02
## AgeCategory:Asthma           2.463327e-02
## AgeCategory:PhysicalActivity -1.775098e-02
## AgeCategory:KidneyDisease    9.694537e-02
```

We now proceed on using the model and the test set for predicting the data. The results are almost identical to the ones we observed before on the LDA.

```r
qda_u.pred <- predict(qda_u, dati_test[,-1])
qda_u.post=qda_u.pred$posterior
qda_u.class <- qda_u.pred$class

# error rate
mean(qda_u.class != dati_test$HeartDisease) * 100
```

```
## [1] 9.456028
```

```r
# threshold = 0.4
qda_u.pred4=as.factor(ifelse(qda_u.post[,2] >= 0.4, 1, 0))
mean(qda_u.pred4!=dati_test$HeartDisease)
```

```
## [1] 0.1000638
```

```r
# threshold = 0.5
qda_u.pred5=as.factor(ifelse(qda_u.post[,2] >= 0.5, 1, 0))
mean(qda_u.pred5!=dati_test$HeartDisease)
```

```
## [1] 0.09456028
```

```r
# threshold = 0.6
qda_u.pred6=as.factor(ifelse(qda_u.post[,2] >= 0.6, 1, 0))
mean(qda_u.pred6!=dati_test$HeartDisease)
```

```
## [1] 0.09072033
```

```r
# the best threshold is 0.6
```

```r
# Confusion matrix
conf_mat <- table(dati_test$HeartDisease, qda_u.pred6)
conf_mat
```

```
##    qda_u.pred6
##         0     1
##   0 71296  1808
##   1  5445  1400
```

```r
# true positive rate
TPR <- conf_mat[2, 2]/sum(conf_mat[2, ])
TPR
```

```
## [1] 0.2045289
```

```r
# true negative rate
TNR <- conf_mat[1, 1]/sum(conf_mat[1, ])
TNR
```

```
## [1] 0.9752681
```

```r
# precision
PREC <- conf_mat[2, 2]/sum(conf_mat[, 2])
PREC
```

```
## [1] 0.436409
```

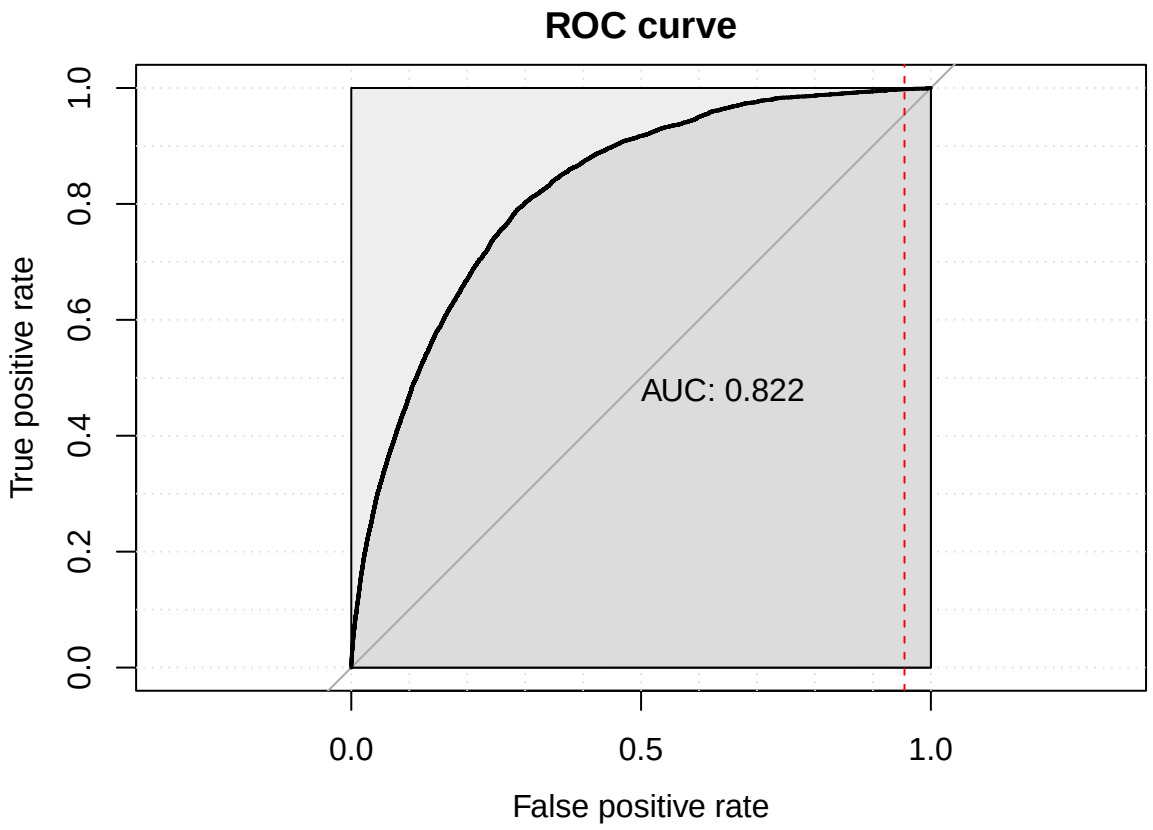
```r
# recall
REC <- conf_mat[2, 2]/sum(conf_mat[2, ])
REC
```

```
## [1] 0.2045289
```

```r
# F1 score
F1_qdau <- 2 * (PREC * REC)/(PREC + REC)
F1_qdau
```

```
## [1] 0.2785238
```

```r
# ROC curve
library(pROC)
roc_u_qda <- roc(dati_test$HeartDisease, qda_u.post[,2])
auc_u_qda <- auc(roc_u_qda)
coords <- coords(roc_u_qda, "best", best.method = "closest.topleft")
opt_thr <- coords$threshold
plot(roc_u_qda, main = "ROC curve", print.auc = TRUE,
     auc.polygon = TRUE, max.auc.polygon = TRUE, grid = TRUE, legacy.axes = TRUE,
     xlab = "False positive rate", ylab = "True positive rate")
abline(v = opt_thr, col = "red", lty = 2)
```



QDA on balanced dataset

We are now repeating the same operations but on the balanced dataset. We notice that the F1 score, but especially the AUC value, are lower than the other models.

```r
qda_b <- qda(formula = HeartDisease ~ BMI + Smoking + AlcoholDrinking +
    Stroke + PhysicalHealth + MentalHealth + DiffWalking + Sex +
    AgeCategory + Diabetic + PhysicalActivity + GenHealth + SleepTime +
    Asthma + KidneyDisease + SkinCancer + BMI:DiffWalking + BMI:GenHealth +
    Smoking:AgeCategory + Smoking:GenHealth + Stroke:PhysicalHealth +
    Stroke:AgeCategory + Stroke:GenHealth + Stroke:DiffWalking +
    PhysicalHealth:DiffWalking + PhysicalHealth:MentalHealth +
```

```
    PhysicalHealth:AgeCategory + MentalHealth:Sex + MentalHealth:GenHealth +
    DiffWalking:AgeCategory + DiffWalking:Sex + Sex:AgeCategory +
    AgeCategory:Asthma, data = dati_balanced_train, family="binomial")
qda_b
```

```
## Call:
## qda(HeartDisease ~ BMI + Smoking + AlcoholDrinking + Stroke +
##     PhysicalHealth + MentalHealth + DiffWalking + Sex + AgeCategory +
##     Diabetic + PhysicalActivity + GenHealth + SleepTime + Asthma +
##     KidneyDisease + SkinCancer + BMI:DiffWalking + BMI:GenHealth +
##     Smoking:AgeCategory + Smoking:GenHealth + Stroke:PhysicalHealth +
##     Stroke:AgeCategory + Stroke:GenHealth + Stroke:DiffWalking +
##     PhysicalHealth:DiffWalking + PhysicalHealth:MentalHealth +
##     PhysicalHealth:AgeCategory + MentalHealth:Sex + MentalHealth:GenHealth +
##     DiffWalking:AgeCategory + DiffWalking:Sex + Sex:AgeCategory +
##     AgeCategory:Asthma, data = dati_balanced_train, family = "binomial")
##
## Prior probabilities of groups:
##   0   1
## 0.5 0.5
##
## Group means:
##        BMI   Smoking AlcoholDrinking     Stroke PhysicalHealth MentalHealth
## 0 29.40018 0.5868570      0.04189400 0.15944076       7.832570     4.633281
## 1 28.21378 0.3948266      0.06931995 0.02572097       2.980076     3.885425
##   DiffWalking       Sex AgeCategory  Diabetic PhysicalActivity GenHealth
## 0   0.3644778 0.5922642   10.240403 0.3282346        0.6383476  2.605417
## 1   0.1201286 0.4621493    7.266124 0.1089244        0.7853663  2.645216
##   SleepTime    Asthma KidneyDisease SkinCancer BMI:DiffWalking BMI:GenHealth
## 0  7.139273 0.1821415    0.12504871  0.1815569       11.313819      76.80310
## 1  7.077260 0.1285074    0.02762081  0.0831060        3.742345      73.88757
##   Smoking:AgeCategory Smoking:GenHealth Stroke:PhysicalHealth
## 0            5.968580          1.549006             1.8087977
## 1            3.069417          1.010425             0.2022603
##   Stroke:AgeCategory Stroke:GenHealth Stroke:DiffWalking
## 0          1.6449240       0.44276111         0.08539556
## 1          0.2569661       0.06586126         0.01081450
##   PhysicalHealth:DiffWalking PhysicalHealth:MentalHealth
## 0                   5.146775                    73.51174
## 1                   1.383379                    28.75638
##   PhysicalHealth:AgeCategory MentalHealth:Sex MentalHealth:GenHealth
## 0                   78.45611         2.123344               13.42751
## 1                   23.96093         1.431557                9.88674
##   DiffWalking:AgeCategory DiffWalking:Sex Sex:AgeCategory AgeCategory:Asthma
## 0                3.813231      0.17970577        6.051101           1.739283
## 1                1.145265      0.04525526        3.202358           0.837539
```

```
qda_b.pred <- predict(qda_b, dati_test[,-1])
qda_b.post=qda_b.pred$posterior
qda_b.class <- qda_b.pred$class

# error rate
mean(qda_b.class != dati_test$HeartDisease) * 100
```

```
## [1] 82.05106
```

```
# threshold = 0.1
qda_b.pred1=as.factor(ifelse(qda_b.post[,2] >= 0.1, 1, 0))
mean(qda_b.pred1!=dati_test$HeartDisease)
```

```
## [1] 0.83532
```

```
# threshold = 0.2
qda_b.pred2=as.factor(ifelse(qda_b.post[,2] >= 0.2, 1, 0))
mean(qda_b.pred2!=dati_test$HeartDisease)
```

```
## [1] 0.829729
```

```
# threshold = 0.3
qda_b.pred3=as.factor(ifelse(qda_b.post[,2] >= 0.3, 1, 0))
mean(qda_b.pred3!=dati_test$HeartDisease)
```

```
## [1] 0.8260766
```

```
# threshold = 0.4
qda_b.pred4=as.factor(ifelse(qda_b.post[,2] >= 0.4, 1, 0))
mean(qda_b.pred4!=dati_test$HeartDisease)
```

```
## [1] 0.8231748
```

```
# threshold = 0.5
qda_b.pred5=as.factor(ifelse(qda_b.post[,2] >= 0.5, 1, 0))
mean(qda_b.pred5!=dati_test$HeartDisease)
```

```
## [1] 0.8205106
```

```
# the best threshold is 0.1
```

```
# Confusion matrix
conf_mat <- table(dati_test$HeartDisease, qda_b.pred1)
conf_mat
```

```
##    qda_b.pred1
##         0     1
##   0  9732 63372
##   1  3411  3434
```

```
# true positive rate
TPR <- conf_mat[2, 2]/sum(conf_mat[2, ])
TPR
```

```
## [1] 0.5016801
```

```
# true negative rate
TNR <- conf_mat[1, 1]/sum(conf_mat[1, ])
TNR
```

```
## [1] 0.1331254
```

```
# precision
PREC <- conf_mat[2, 2]/sum(conf_mat[, 2])
PREC
```

```
## [1] 0.05140257
```

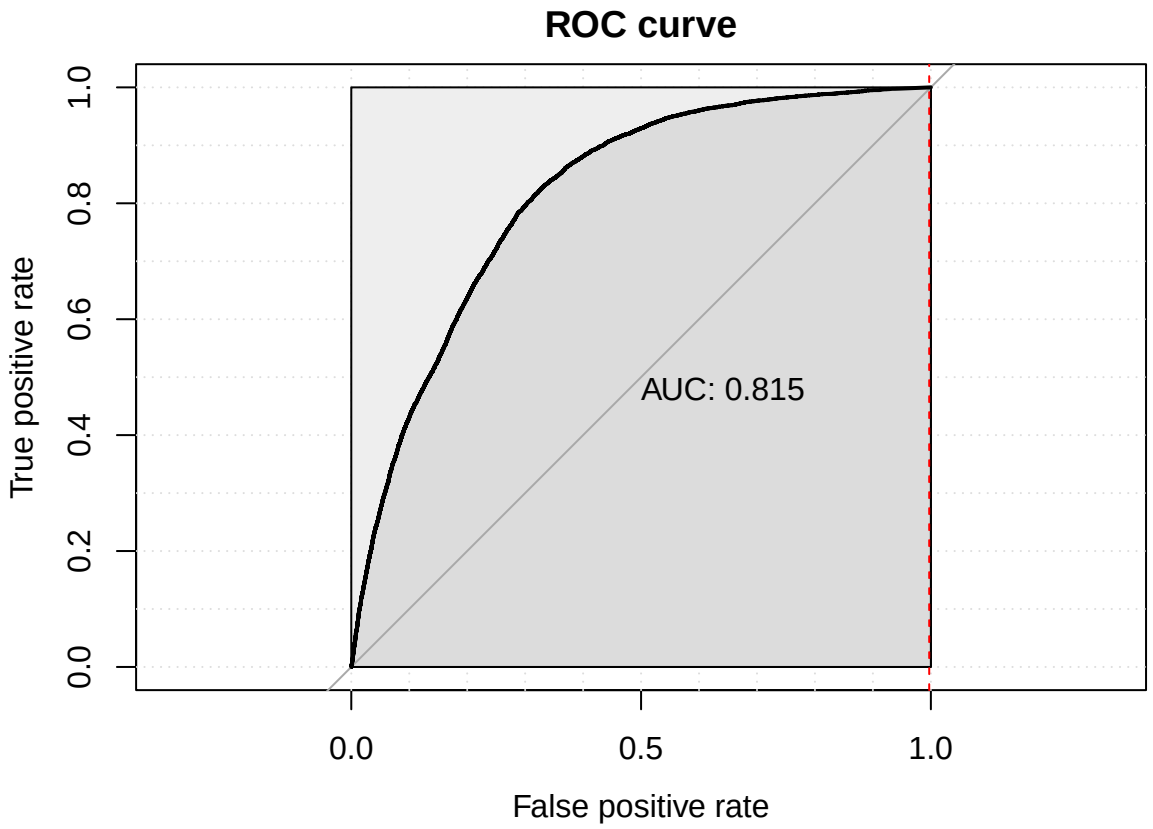
```
# recall
REC <- conf_mat[2, 2]/sum(conf_mat[2, ])
REC
```

```
## [1] 0.5016801
```

```
# F1 score
F1_qdab <- 2 * (PREC * REC)/(PREC + REC)
F1_qdab
```

```
## [1] 0.0932506
```

```
# ROC curve
library(pROC)
roc_b <- roc(dati_test$HeartDisease, qda_b.post[,2])
auc_b <- auc(roc_b)
coords <- coords(roc_b, "best", best.method = "closest.topleft")
opt_thr <- coords$threshold
plot(roc_b, main = "ROC curve", print.auc = TRUE, auc.polygon = TRUE,
     max.auc.polygon = TRUE, grid = TRUE, legacy.axes = TRUE,
     xlab = "False positive rate", ylab = "True positive rate")
abline(v = 1-opt_thr, col = "red", lty = 2)
```



Shrinkage methods

We will now model our dataset with shrinkage methods: ridge and lasso. Shrinkage methods aim to address two common challenges in regression analysis: multicollinearity and overfitting. Ridge regression and lasso regression both introduce a penalty term to the ordinary least squares (OLS) objective function, which helps

to control the complexity of the model and mitigate the effects of multicollinearity. Ridge regression adds a penalty term that is proportional to the sum of squared coefficients, effectively shrinking the coefficients towards zero but without setting them exactly to zero. This leads to more stable coefficient estimates, reducing the impact of multicollinearity and improving model performance. Lasso regression, on the other hand, adds a penalty term that is proportional to the sum of the absolute values of the coefficients. This has the effect of both shrinking coefficients towards zero and performing variable selection by setting some coefficients exactly to zero. Lasso regression can effectively identify and exclude irrelevant predictors from the model, providing a sparse solution that only includes the most important predictors. Although multicollinearity and overfitting are not significant concerns in our dataset, it is still worthwhile to explore whether ridge and lasso regression can yield improved performance compared to other models.

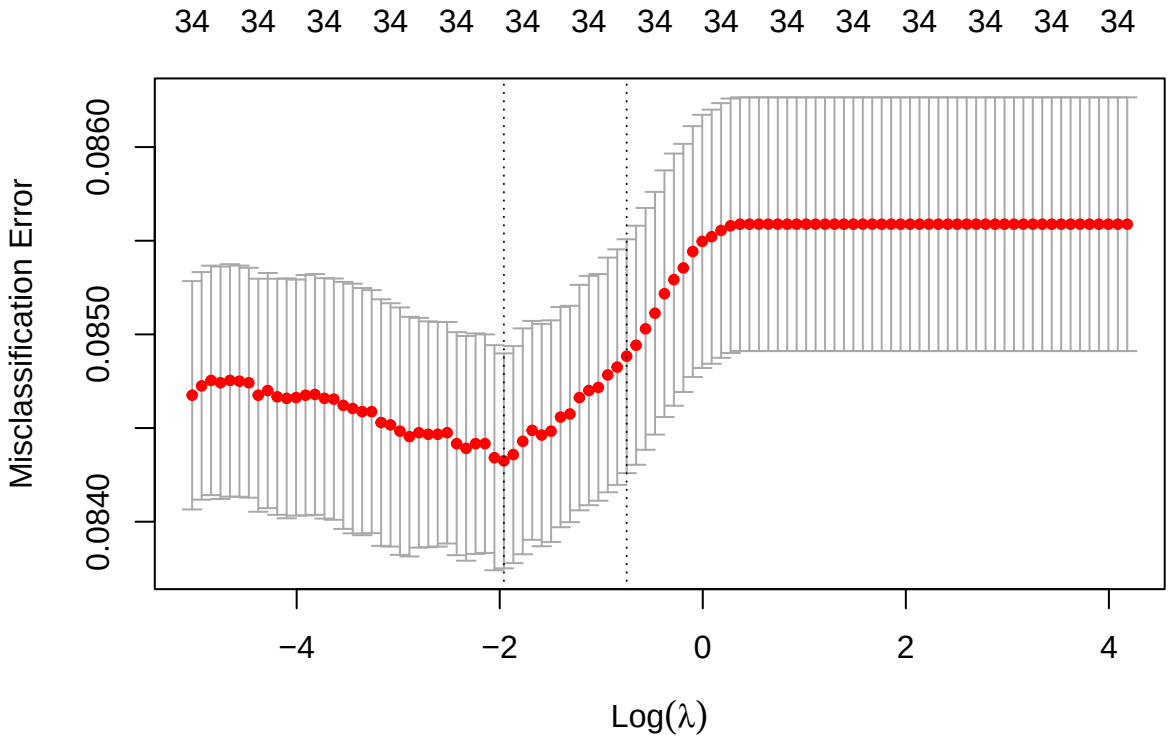
Ridge regression

Unbalanced dataset

We are now applying the ridge regression to the unbalanced dataset. As we did for the LDA and QDA, we will consider the significant variables and interactions of the best glm model. With the object *ridge_u.cv* we are defining the cross-validation with ridge logistic regression, and the following plot displays the cross-validated error as a function of the log(lambda) values, where lambda represents the tuning parameter controlling the amount of shrinkage. We then extract the optimal lambda value that minimizes the cross-validated error and use it to obtain predictions on the test dataset. We notice the same pattern we saw with the other models, so a very low F1 score. The AUC value is bigger than the previous models, but still lower than the glm.

```
library(glmnet)
library(tidyverse)

# interaction matrix
dati_inter <- transform(dati_train[,-c(11,14)],
                        BMI_Diabetic = BMI * Diabetic,
                        BMI_AlcoholDrinking = BMI * AlcoholDrinking,
                        Smoking_KidneyDisease = Smoking * KidneyDisease,
                        Smoking_AgeCategory = Smoking * AgeCategory,
                        Stroke_DiffWalking = Stroke * DiffWalking,
                        Stroke_AgeCategory = Stroke * AgeCategory,
                        Smoking_Stroke = Smoking * Stroke,
                        PhysicalHealth_DiffWalking = PhysicalHealth * DiffWalking,
                        PhysicalHealth_MentalHealth = PhysicalHealth * MentalHealth,
                        PhysicalHealth_AgeCategory = PhysicalHealth * AgeCategory,
                        PhysicalHealth_Diabetic = PhysicalHealth * Diabetic,
                        MentalHealth_Sex = MentalHealth * Sex,
                        DiffWalking_AgeCategory = DiffWalking * AgeCategory,
                        DiffWalking_Sex = DiffWalking * Sex,
                        Sex_AgeCategory = Sex * AgeCategory,
                        AgeCategory_Diabetic = AgeCategory * Diabetic,
                        AgeCategory_Asthma = AgeCategory * Asthma,
                        AgeCategory_PhysicalActivity = AgeCategory * PhysicalActivity,
                        AgeCategory_KidneyDisease = AgeCategory * KidneyDisease)

formula <- HeartDisease ~ BMI + Smoking + AlcoholDrinking + Stroke + PhysicalHealth +
  MentalHealth + DiffWalking + Sex + AgeCategory + Diabetic +
  PhysicalActivity + SleepTime + Asthma + KidneyDisease + SkinCancer +
  BMI_Diabetic + BMI_AlcoholDrinking + Smoking_KidneyDisease + Smoking_AgeCategory +
  Stroke_DiffWalking + Stroke_AgeCategory + Smoking_Stroke +
  PhysicalHealth_DiffWalking + PhysicalHealth_MentalHealth +
  PhysicalHealth_AgeCategory + PhysicalHealth_Diabetic + MentalHealth_Sex +
```

```
  DiffWalking_AgeCategory + DiffWalking_Sex + Sex_AgeCategory +
  AgeCategory_Diabetic + AgeCategory_Asthma +
  AgeCategory_PhysicalActivity + AgeCategory_KidneyDisease

design_matrix <- model.matrix(formula, data = dati_inter)

ridge_u.cv <- cv.glmnet(x = design_matrix, y = dati_inter$HeartDisease,
                        family = "binomial", alpha = 0, type.measure = "class")

dati_test_inter <-transform(dati_test[,-c(11,14)],
                            BMI_Diabetic = BMI * Diabetic,
                            BMI_AlcoholDrinking = BMI * AlcoholDrinking,
                            Smoking_KidneyDisease = Smoking * KidneyDisease,
                            Smoking_AgeCategory = Smoking * AgeCategory,
                            Stroke_DiffWalking = Stroke * DiffWalking,
                            Stroke_AgeCategory = Stroke * AgeCategory,
                            Smoking_Stroke = Smoking * Stroke,
                            PhysicalHealth_DiffWalking=PhysicalHealth*DiffWalking,
                            PhysicalHealth_MentalHealth=PhysicalHealth*MentalHealth,
                            PhysicalHealth_AgeCategory=PhysicalHealth*AgeCategory,
                            PhysicalHealth_Diabetic = PhysicalHealth * Diabetic,
                            MentalHealth_Sex = MentalHealth * Sex,
                            DiffWalking_AgeCategory = DiffWalking * AgeCategory,
                            DiffWalking_Sex = DiffWalking * Sex,
                            Sex_AgeCategory = Sex * AgeCategory,
                            AgeCategory_Diabetic = AgeCategory * Diabetic,
                            AgeCategory_Asthma = AgeCategory * Asthma,
                            AgeCategory_PhysicalActivity=AgeCategory*PhysicalActivity,
                            AgeCategory_KidneyDisease = AgeCategory * KidneyDisease)

design_matrix_test <- model.matrix(formula, data = dati_test_inter)

plot(ridge_u.cv)
```




```
# optimal lambda
lmin <- ridge_u.cv$lambda.min
lmin
```

```
## [1] 0.1409873
```

```
# predictions with optimal lambda
ridge_u.predmin <- predict(ridge_u.cv, newx= design_matrix_test %>%
                                    as.matrix(), type="response", s = lmin)
ridge_u.predmin_class <- predict(ridge_u.cv, newx= design_matrix_test %>%
                                    as.matrix(), type="class", s = lmin)
```

```
# Confusion matrix
conf_mat <- table(dati_test$HeartDisease, ridge_u.predmin_class)
conf_mat
```

```
##    ridge_u.predmin_class
##         0     1
##   0 72823   281
##   1  6475   370
```

```
# true positive rate
TPR <- conf_mat[2, 2]/sum(conf_mat[2, ])
TPR
```

```
## [1] 0.05405405
```

```
# true negative rate
TNR <- conf_mat[1, 1]/sum(conf_mat[1, ])
TNR
```

```
## [1] 0.9961562
```

```r
# precision
PREC <- conf_mat[2, 2]/sum(conf_mat[, 2])
PREC
```

```
## [1] 0.5683564
```

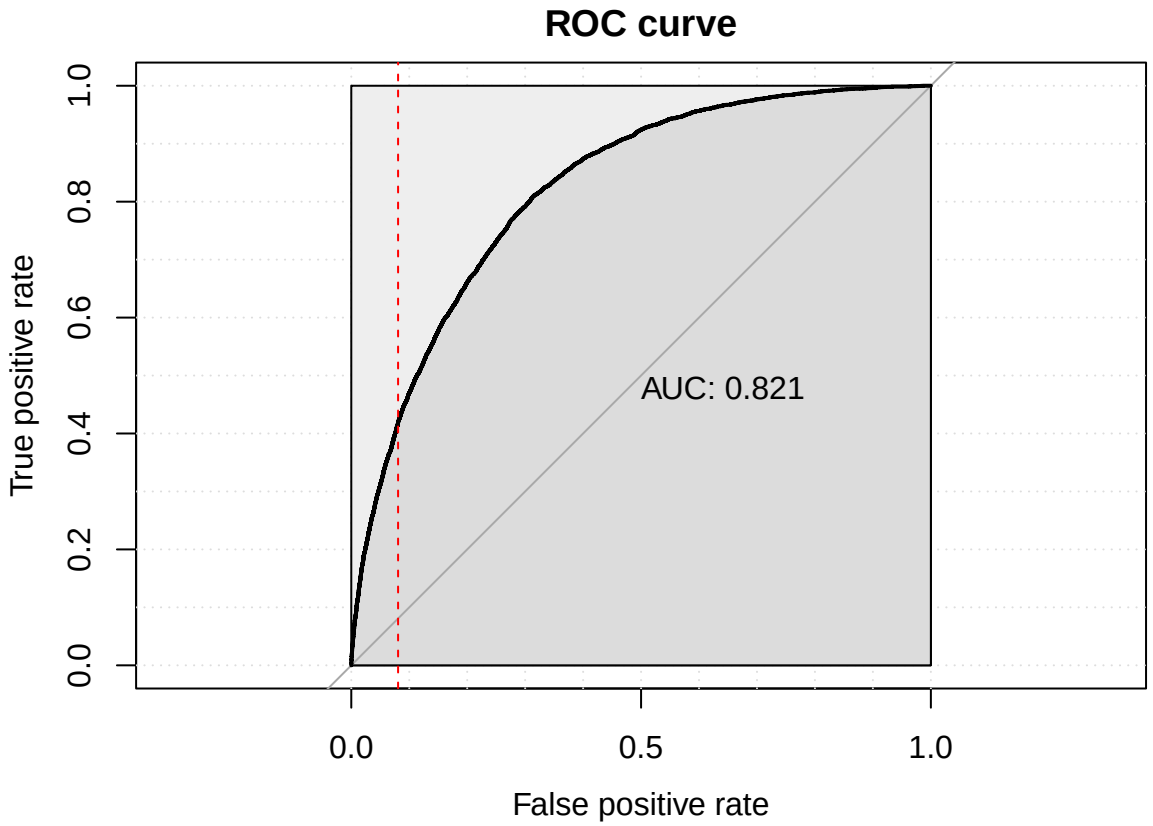
```r
# recall
REC <- conf_mat[2, 2]/sum(conf_mat[2, ])
REC
```

```
## [1] 0.05405405
```

```r
# F1 score
F1_ridgeu <- 2 * (PREC * REC)/(PREC + REC)
F1_ridgeu
```

```
## [1] 0.09871932
```

```r
# ROC curve
library(pROC)
roc_u <- roc(dati_test$HeartDisease, as.numeric(ridge_u.predmin))
auc_u <- auc(roc_u)
coords <- coords(roc_u, "best", best.method = "closest.topleft")
opt_thr <- coords$threshold
plot(roc_u, main = "ROC curve", print.auc = TRUE, auc.polygon = TRUE,
     max.auc.polygon = TRUE, grid = TRUE, legacy.axes = TRUE,
     xlab = "False positive rate", ylab = "True positive rate")
abline(v = 1-opt_thr, col = "red", lty = 2)
```

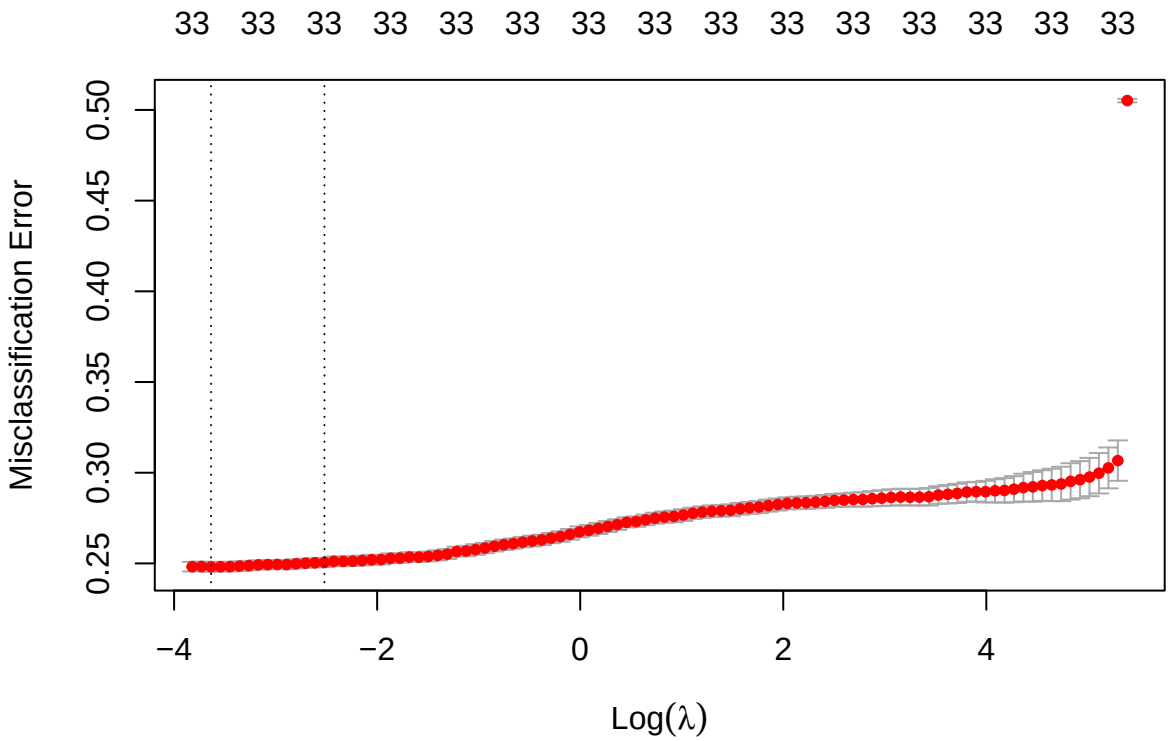


Balanced dataset

We are now applying the ridge regression to the balanced dataset, considering the variables and interactions of the best glm model. As we did for the unbalanced dataset, we are defining the cross-validation with ridge logistic regression, and the following plot displays the cross-validated error as a function of the log(lambda) values. We then extract the optimal lambda value that minimizes the cross-validated error and use it to obtain predictions on the test dataset. We can notice that the performance metrics lowered, as well as the AUC value.

```
library(glmnet)
library(tidyverse)

# interaction matrix
dati_inter <- transform(dati_balanced_train[,-11],
                        BMI_DiffWalking = BMI * DiffWalking,
                        BMI_GenHealth = BMI * GenHealth,
                        Smoking_AgeCategory = Smoking * AgeCategory,
                        Smoking_GenHealth = Smoking * GenHealth,
                        Stroke_PhysicalHealth = Stroke * PhysicalHealth,
                        Stroke_AgeCategory = Stroke * AgeCategory,
                        Stroke_GenHealth = Stroke * GenHealth,
                        Stroke_DiffWalking = Stroke * DiffWalking,
                        PhysicalHealth_DiffWalking = PhysicalHealth * DiffWalking,
                        PhysicalHealth_MentalHealth = PhysicalHealth * MentalHealth,
                        PhysicalHealth_AgeCategory = PhysicalHealth * AgeCategory,
                        MentalHealth_Sex = MentalHealth * Sex,
                        MentalHealth_GenHealth = MentalHealth * GenHealth,
                        DiffWalking_AgeCategory = DiffWalking * AgeCategory,
                        DiffWalking_Sex = DiffWalking * Sex,
                        Sex_AgeCategory = Sex * AgeCategory,
                        AgeCategory_Asthma = AgeCategory * Asthma)

formula <- HeartDisease ~ BMI + Smoking + AlcoholDrinking +
    Stroke + PhysicalHealth + MentalHealth + DiffWalking + Sex +
    AgeCategory + Diabetic + PhysicalActivity + GenHealth + SleepTime +
    Asthma + KidneyDisease + SkinCancer + BMI:DiffWalking + BMI:GenHealth +
    Smoking:AgeCategory + Smoking:GenHealth + Stroke:PhysicalHealth +
    Stroke:AgeCategory + Stroke:GenHealth + Stroke:DiffWalking +
    PhysicalHealth:DiffWalking + PhysicalHealth:MentalHealth +
    PhysicalHealth:AgeCategory + MentalHealth:Sex + MentalHealth:GenHealth +
    DiffWalking:AgeCategory + DiffWalking:Sex + Sex:AgeCategory +
    AgeCategory:Asthma

design_matrix <- model.matrix(formula, data = dati_inter)

dati_test_inter <- transform(dati_test[,-11],
                             BMI_DiffWalking = BMI * DiffWalking,
                             BMI_GenHealth = BMI * GenHealth,
                             Smoking_AgeCategory = Smoking * AgeCategory,
                             Smoking_GenHealth = Smoking * GenHealth,
                             Stroke_PhysicalHealth = Stroke * PhysicalHealth,
                             Stroke_AgeCategory = Stroke * AgeCategory,
                             Stroke_GenHealth = Stroke * GenHealth,
                             Stroke_DiffWalking = Stroke * DiffWalking,
                             PhysicalHealth_DiffWalking = PhysicalHealth*DiffWalking,
```

```
                            PhysicalHealth_MentalHealth = PhysicalHealth*MentalHealth,
                            PhysicalHealth_AgeCategory = PhysicalHealth*AgeCategory,
                            MentalHealth_Sex = MentalHealth * Sex,
                            MentalHealth_GenHealth = MentalHealth * GenHealth,
                            DiffWalking_AgeCategory = DiffWalking * AgeCategory,
                            DiffWalking_Sex = DiffWalking * Sex,
                            Sex_AgeCategory = Sex * AgeCategory,
                            AgeCategory_Asthma = AgeCategory * Asthma)

design_matrix_test <- model.matrix(formula, data = dati_test_inter)

ridge_b.cv <- cv.glmnet(x = design_matrix, y = dati_inter$HeartDisease, family = "binomial", alpha = 0,

plot(ridge_b.cv)
```




```
# optimal lambda
lmin <- ridge_b.cv$lambda.min
lmin
```

```
## [1] 0.02636199
```

```
# predictions with optimal lambda
ridge_b.predmin <- predict(ridge_b.cv, newx= design_matrix_test %>%
                                  as.matrix(), type="response", s = lmin)
ridge_b.predmin_class <- predict(ridge_b.cv, newx= design_matrix_test %>%
                                  as.matrix(), type="class", s = lmin)
```

```
# Confusion matrix
conf_mat <- table(dati_test$HeartDisease, ridge_b.predmin_class)
conf_mat
```

```
##    ridge_b.predmin_class
```

```
##         0     1
##   0 18485 54619
##   1  5144  1701
```

```
# true positive rate
TPR <- conf_mat[2, 2]/sum(conf_mat[2, ])
TPR
```

```
## [1] 0.2485026
```

```
# true negative rate
TNR <- conf_mat[1, 1]/sum(conf_mat[1, ])
TNR
```

```
## [1] 0.2528589
```

```
# precision
PREC <- conf_mat[2, 2]/sum(conf_mat[, 2])
PREC
```

```
## [1] 0.03020241
```

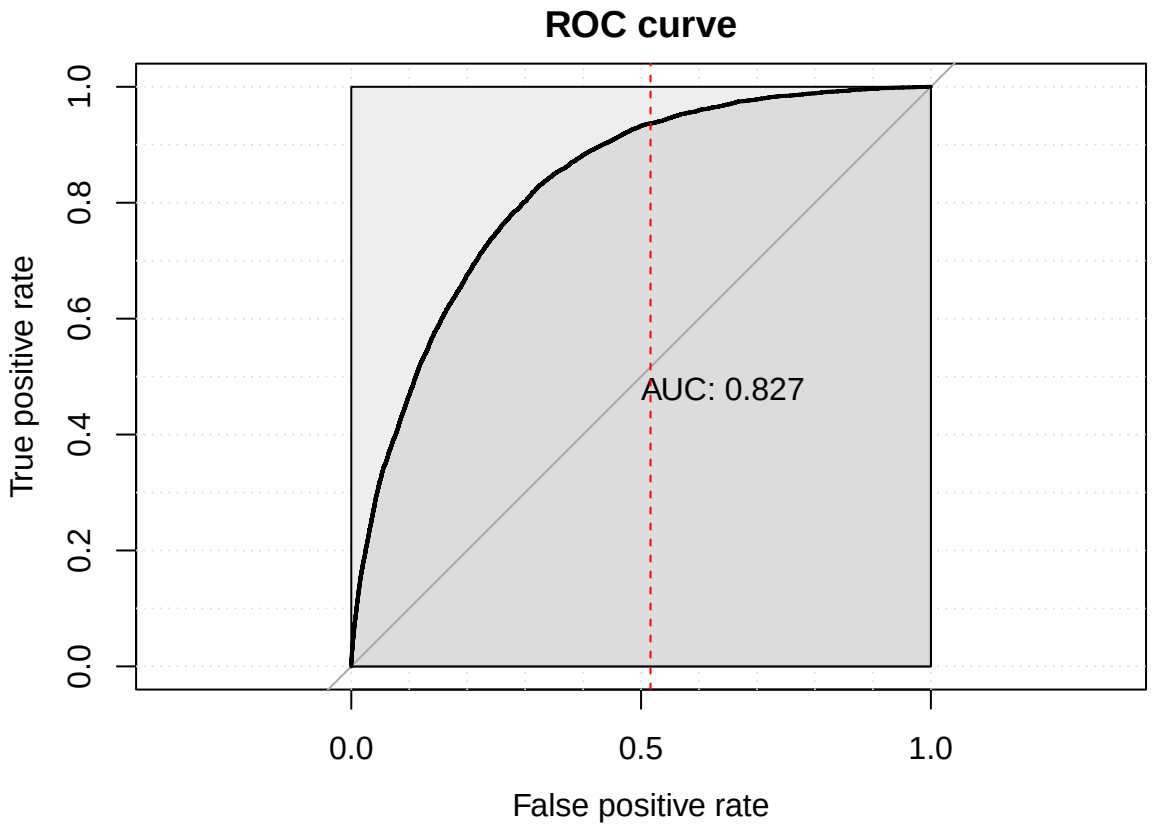
```
# recall
REC <- conf_mat[2, 2]/sum(conf_mat[2, ])
REC
```

```
## [1] 0.2485026
```

```
# F1 score
F1_ridgeb <- 2 * (PREC * REC)/(PREC + REC)
F1_ridgeb
```

```
## [1] 0.05385894
```

```
# ROC curve
library(pROC)
roc_b <- roc(dati_test$HeartDisease, as.numeric(ridge_b.predmin))
auc_b <- auc(roc_b)
coords <- coords(roc_b, "best", best.method = "closest.topleft")
opt_thr <- coords$threshold
plot(roc_b, main = "ROC curve", print.auc = TRUE,
     auc.polygon = TRUE, max.auc.polygon = TRUE, grid = TRUE, legacy.axes = TRUE,
     xlab = "False positive rate", ylab = "True positive rate")
abline(v = 1-opt_thr, col = "red", lty = 2)
```



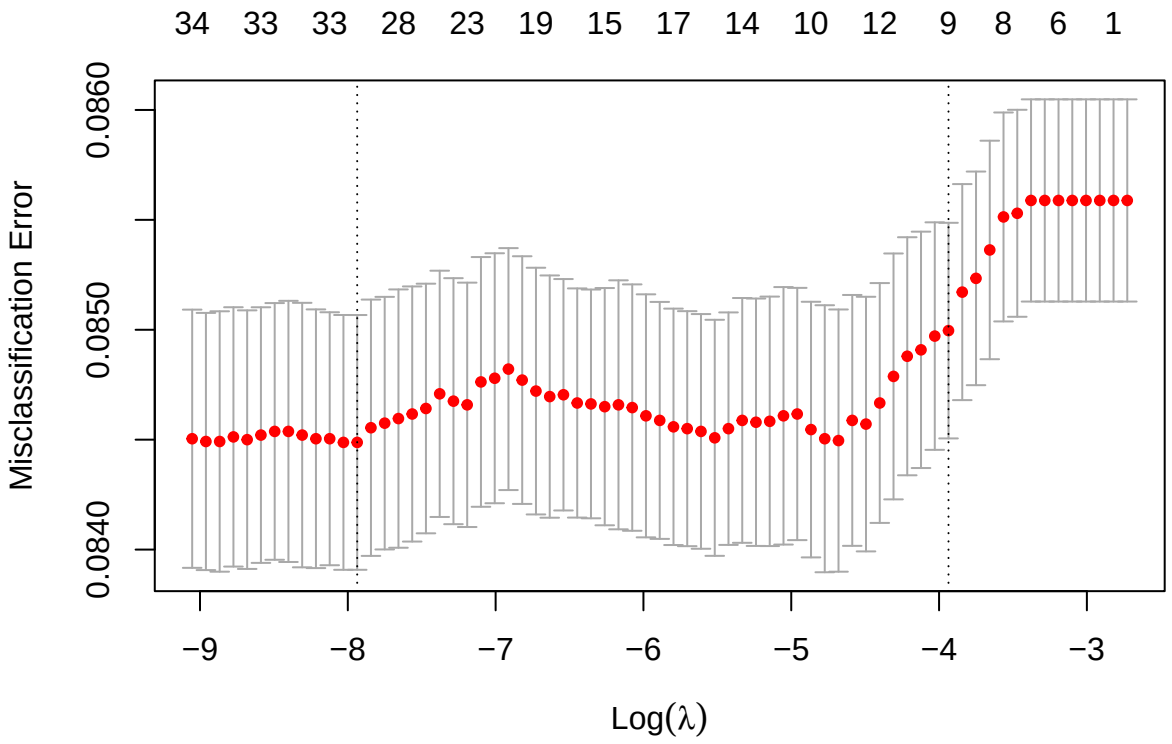
Lasso regression

Unbalanced dataset

Similarly as we did for the ridge regression, we are now applying the lasso regression. From the results we can notice how it performs better than the ridge regression in terms of F1 score.

```
library(glmnet)
library(tidyverse)
dati_inter <- transform(dati_train[,-c(11,14)],
                        BMI_Diabetic = BMI * Diabetic,
                        BMI_AlcoholDrinking = BMI * AlcoholDrinking,
                        Smoking_KidneyDisease = Smoking * KidneyDisease,
                        Smoking_AgeCategory = Smoking * AgeCategory,
                        Stroke_DiffWalking = Stroke * DiffWalking,
                        Stroke_AgeCategory = Stroke * AgeCategory,
                        Smoking_Stroke = Smoking * Stroke,
                        PhysicalHealth_DiffWalking = PhysicalHealth * DiffWalking,
                        PhysicalHealth_MentalHealth = PhysicalHealth * MentalHealth,
                        PhysicalHealth_AgeCategory = PhysicalHealth * AgeCategory,
                        PhysicalHealth_Diabetic = PhysicalHealth * Diabetic,
                        MentalHealth_Sex = MentalHealth * Sex,
                        DiffWalking_AgeCategory = DiffWalking * AgeCategory,
                        DiffWalking_Sex = DiffWalking * Sex,
                        Sex_AgeCategory = Sex * AgeCategory,
                        AgeCategory_Diabetic = AgeCategory * Diabetic,
                        AgeCategory_Asthma = AgeCategory * Asthma,
                        AgeCategory_PhysicalActivity = AgeCategory * PhysicalActivity,
```

```
                        AgeCategory_KidneyDisease = AgeCategory * KidneyDisease)

formula <- HeartDisease ~ BMI + Smoking + AlcoholDrinking + Stroke + PhysicalHealth +
         MentalHealth + DiffWalking + Sex + AgeCategory + Diabetic + PhysicalActivity +
         SleepTime + Asthma + KidneyDisease + SkinCancer + BMI_Diabetic +
         BMI_AlcoholDrinking + Smoking_KidneyDisease + Smoking_AgeCategory +
         Stroke_DiffWalking + Stroke_AgeCategory + Smoking_Stroke +
         PhysicalHealth_DiffWalking + PhysicalHealth_MentalHealth +
         PhysicalHealth_AgeCategory + PhysicalHealth_Diabetic + MentalHealth_Sex +
         DiffWalking_AgeCategory + DiffWalking_Sex + Sex_AgeCategory +
         AgeCategory_Diabetic + AgeCategory_Asthma +
         AgeCategory_PhysicalActivity + AgeCategory_KidneyDisease

design_matrix <- model.matrix(formula, data = dati_inter)

lasso_u.cv <- cv.glmnet(x = design_matrix, y = dati_inter$HeartDisease,
                        family = "binomial", alpha = 1, type.measure = "class")

dati_test_inter <- transform(dati_test[,-c(11,14)],
                             BMI_Diabetic = BMI * Diabetic,
                             BMI_AlcoholDrinking = BMI * AlcoholDrinking,
                             Smoking_KidneyDisease = Smoking * KidneyDisease,
                             Smoking_AgeCategory = Smoking * AgeCategory,
                             Stroke_DiffWalking = Stroke * DiffWalking,
                             Stroke_AgeCategory = Stroke * AgeCategory,
                             Smoking_Stroke = Smoking * Stroke,
                             PhysicalHealth_DiffWalking=PhysicalHealth*DiffWalking,
                             PhysicalHealth_MentalHealth=PhysicalHealth*MentalHealth,
                             PhysicalHealth_AgeCategory=PhysicalHealth*AgeCategory,
                             PhysicalHealth_Diabetic = PhysicalHealth * Diabetic,
                             MentalHealth_Sex = MentalHealth * Sex,
                             DiffWalking_AgeCategory = DiffWalking * AgeCategory,
                             DiffWalking_Sex = DiffWalking * Sex,
                             Sex_AgeCategory = Sex * AgeCategory,
                             AgeCategory_Diabetic = AgeCategory * Diabetic,
                             AgeCategory_Asthma = AgeCategory * Asthma,
                             AgeCategory_PhysicalActivity=AgeCategory*PhysicalActivity,
                             AgeCategory_KidneyDisease=AgeCategory*KidneyDisease)

design_matrix_test <- model.matrix(formula, data = dati_test_inter)
plot(lasso_u.cv)
```



It can be noticed that the lambda is very low (close to 0), which implies that the regularization effect is weak, and the lasso regression model will closely resemble ordinary linear regression. The coefficients will be estimated without much constraint, potentially leading to a larger number of non-zero coefficients.

```
# optimal lambda
lmin <- lasso_u.cv$lambda.min
lmin
```

```
## [1] 0.0003574541
```

```
# MSE with best lambda
lasso_u.predmin <- predict(lasso_u.cv, newx= design_matrix_test %>%
                                    as.matrix(), type="response", s = lmin)
lasso_u.predmin_class <- predict(lasso_u.cv, newx= design_matrix_test %>%
                                    as.matrix(), type="class", s = lmin)
```

```
# Confusion matrix
conf_mat <- table(dati_test$HeartDisease, lasso_u.predmin_class)
conf_mat
```

```
##    lasso_u.predmin_class
##         0     1
##   0 72633   471
##   1  6293   552
```

```
# true positive rate
TPR <- conf_mat[2, 2]/sum(conf_mat[2, ])
TPR
```

```
## [1] 0.0806428
```

```
# true negative rate
TNR <- conf_mat[1, 1]/sum(conf_mat[1, ])
TNR
```

```
## [1] 0.9935571
```

```
# precision
PREC <- conf_mat[2, 2]/sum(conf_mat[, 2])
PREC
```

```
## [1] 0.5395894
```

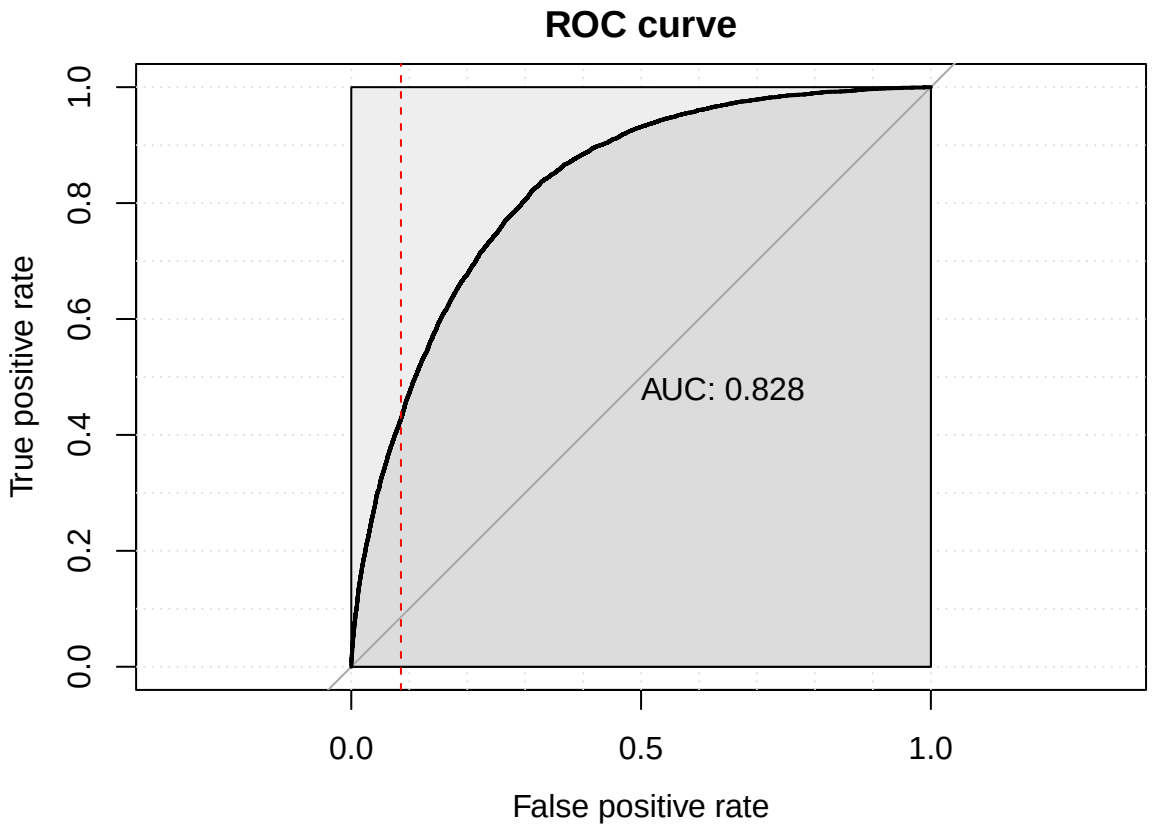
```
# recall
REC <- conf_mat[2, 2]/sum(conf_mat[2, ])
REC
```

```
## [1] 0.0806428
```

```
# F1 score
F1_lassou <- 2 * (PREC * REC)/(PREC + REC)
F1_lassou
```

```
## [1] 0.1403152
```

```
# ROC curve
library(pROC)
roc_u <- roc(dati_test$HeartDisease, as.numeric(lasso_u.predmin))
auc_u <- auc(roc_u)
coords <- coords(roc_u, "best", best.method = "closest.topleft")
opt_thr <- coords$threshold
plot(roc_u, main = "ROC curve", print.auc = TRUE,
     auc.polygon = TRUE, max.auc.polygon = TRUE, grid = TRUE, legacy.axes = TRUE,
     xlab = "False positive rate", ylab = "True positive rate")
abline(v = 1-opt_thr, col = "red", lty = 2)
```

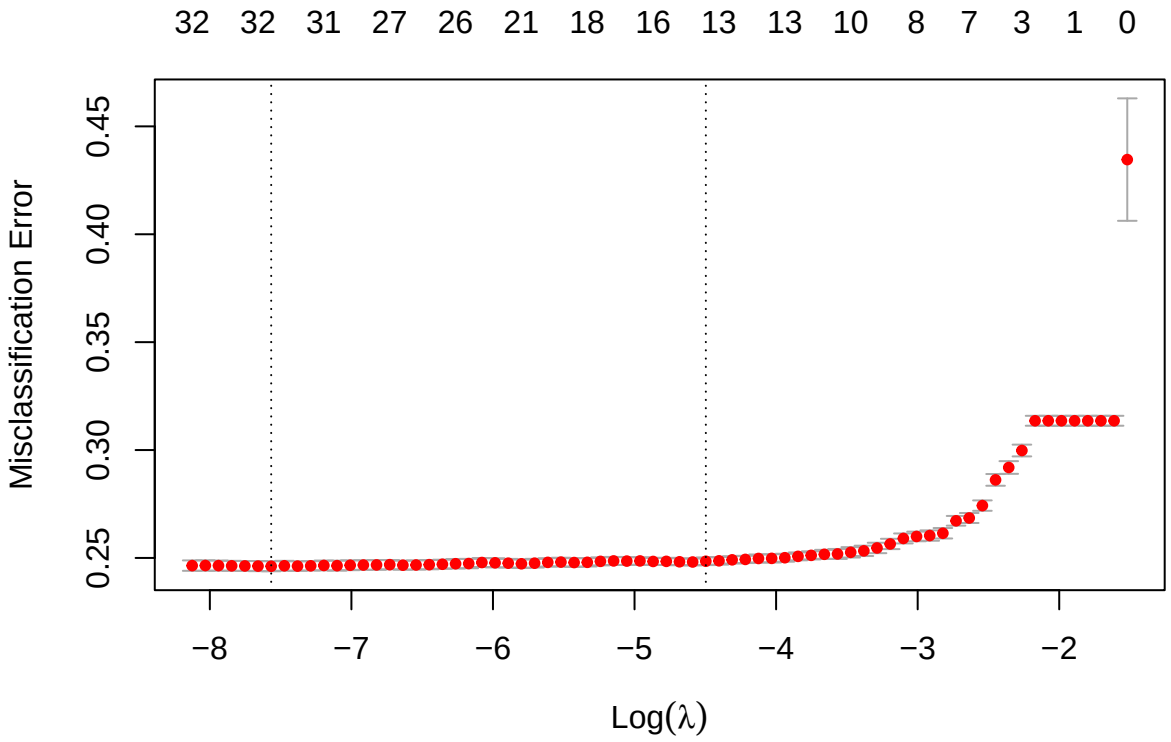


Balanced dataset

We are now applying the lasso regression to the balanced dataset, which performs similarly to the ridge regression but better in terms of AUC value.

```
library(glmnet)
library(tidyverse)

dati_inter <- transform(dati_balanced_train[,-11],
                        BMI_DiffWalking = BMI * DiffWalking,
                        BMI_GenHealth = BMI * GenHealth,
                        Smoking_AgeCategory = Smoking * AgeCategory,
                        Smoking_GenHealth = Smoking * GenHealth,
                        Stroke_PhysicalHealth = Stroke * PhysicalHealth,
                        Stroke_AgeCategory = Stroke * AgeCategory,
                        Stroke_GenHealth = Stroke * GenHealth,
                        Stroke_DiffWalking = Stroke * DiffWalking,
                        PhysicalHealth_DiffWalking = PhysicalHealth * DiffWalking,
                        PhysicalHealth_MentalHealth = PhysicalHealth * MentalHealth,
                        PhysicalHealth_AgeCategory = PhysicalHealth * AgeCategory,
                        MentalHealth_Sex = MentalHealth * Sex,
                        MentalHealth_GenHealth = MentalHealth * GenHealth,
                        DiffWalking_AgeCategory = DiffWalking * AgeCategory,
                        DiffWalking_Sex = DiffWalking * Sex,
                        Sex_AgeCategory = Sex * AgeCategory,
                        AgeCategory_Asthma = AgeCategory * Asthma)

formula <- HeartDisease ~ BMI + Smoking + AlcoholDrinking +
    Stroke + PhysicalHealth + MentalHealth + DiffWalking + Sex +
    AgeCategory + Diabetic + PhysicalActivity + GenHealth + SleepTime +
    Asthma + KidneyDisease + SkinCancer + BMI:DiffWalking + BMI:GenHealth +
    Smoking:AgeCategory + Smoking:GenHealth + Stroke:PhysicalHealth +
    Stroke:AgeCategory + Stroke:GenHealth + Stroke:DiffWalking +
    PhysicalHealth:DiffWalking + PhysicalHealth:MentalHealth +
    PhysicalHealth:AgeCategory + MentalHealth:Sex + MentalHealth:GenHealth +
    DiffWalking:AgeCategory + DiffWalking:Sex + Sex:AgeCategory +
    AgeCategory:Asthma

design_matrix <- model.matrix(formula, data = dati_inter)

dati_test_inter <- transform(dati_test[,-11],
                        BMI_DiffWalking = BMI * DiffWalking,
                        BMI_GenHealth = BMI * GenHealth,
                        Smoking_AgeCategory = Smoking * AgeCategory,
                        Smoking_GenHealth = Smoking * GenHealth,
                        Stroke_PhysicalHealth = Stroke * PhysicalHealth,
                        Stroke_AgeCategory = Stroke * AgeCategory,
                        Stroke_GenHealth = Stroke * GenHealth,
                        Stroke_DiffWalking = Stroke * DiffWalking,
                        PhysicalHealth_DiffWalking = PhysicalHealth * DiffWalking,
                        PhysicalHealth_MentalHealth = PhysicalHealth * MentalHealth,
                        PhysicalHealth_AgeCategory = PhysicalHealth * AgeCategory,
                        MentalHealth_Sex = MentalHealth * Sex,
                        MentalHealth_GenHealth = MentalHealth * GenHealth,
                        DiffWalking_AgeCategory = DiffWalking * AgeCategory,
```

```
                        DiffWalking_Sex = DiffWalking * Sex,
                        Sex_AgeCategory = Sex * AgeCategory,
                        AgeCategory_Asthma = AgeCategory * Asthma)

design_matrix_test <- model.matrix(formula, data = dati_test_inter)

lasso_b.cv <- cv.glmnet(x = design_matrix, y = dati_inter$HeartDisease,
                        family = "binomial", alpha = 1, type.measure = "class")
plot(lasso_b.cv)
```



As for the unbalanced dataset, the minimum lambda is close to zero.

```
# optimal lambda
lmin <- lasso_b.cv$lambda.min
lmin
```

```
## [1] 0.0005174966
```

```
# predictions with optimal lambda
lasso_b.predmin <- predict(lasso_b.cv, newx= design_matrix_test %>%
                                  as.matrix(), type="response", s = lmin)
lasso_b.predmin_class <- predict(lasso_b.cv, newx= design_matrix_test %>%
                                  as.matrix(), type="class", s = lmin)
```

```
# Confusion matrix
conf_mat <- table(dati_test$HeartDisease, lasso_b.predmin_class)
conf_mat
```

```
##    lasso_b.predmin_class
##         0     1
##   0 19418 53686
##   1  5275  1570
```

```r
# true positive rate
TPR <- conf_mat[2, 2]/sum(conf_mat[2, ])
TPR
```

```
## [1] 0.2293645
```

```r
# true negative rate
TNR <- conf_mat[1, 1]/sum(conf_mat[1, ])
TNR
```

```
## [1] 0.2656216
```

```r
# precision
PREC <- conf_mat[2, 2]/sum(conf_mat[, 2])
PREC
```

```
## [1] 0.0284132
```

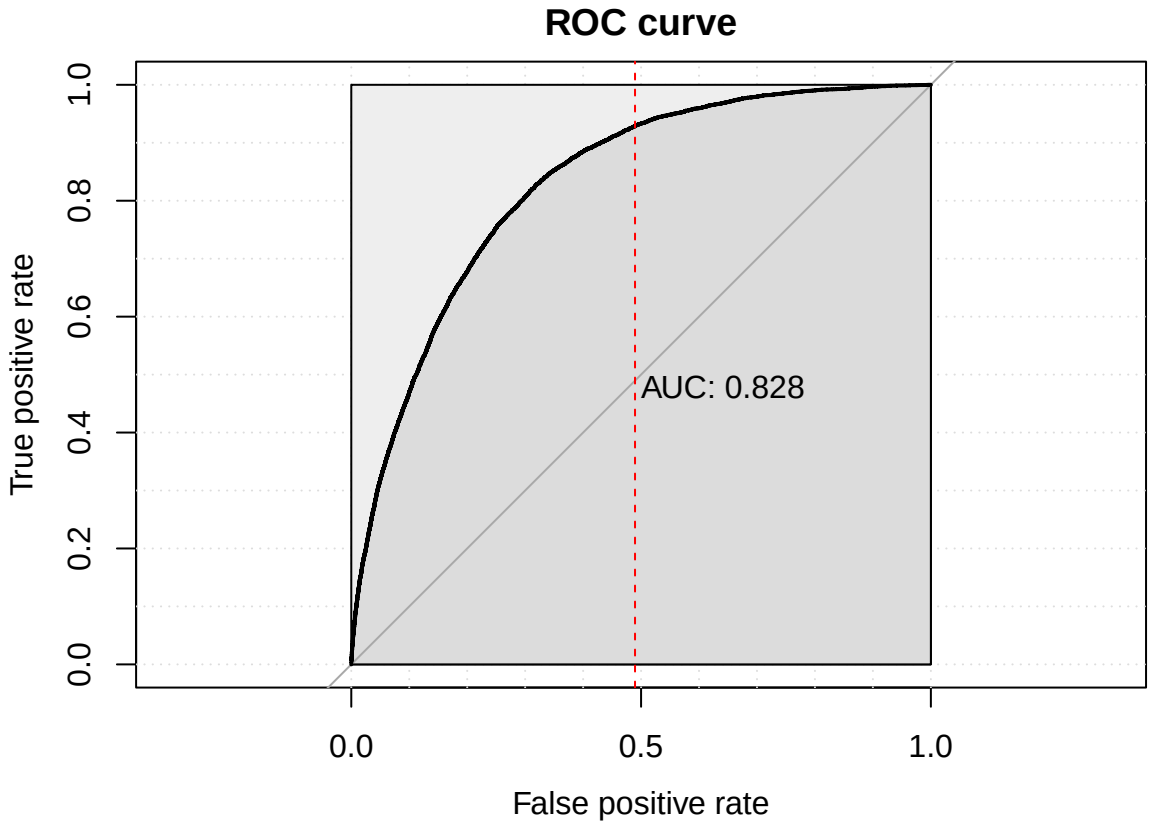
```r
# recall
REC <- conf_mat[2, 2]/sum(conf_mat[2, ])
REC
```

```
## [1] 0.2293645
```

```r
# F1 score
F1_lassob <- 2 * (PREC * REC)/(PREC + REC)
F1_lassob
```

```
## [1] 0.05056279
```

```r
# ROC curve
library(pROC)
roc_b <- roc(dati_test$HeartDisease, as.numeric(lasso_b.predmin))
auc_b <- auc(roc_b)
coords <- coords(roc_b, "best", best.method = "closest.topleft")
opt_thr <- coords$threshold
plot(roc_b, main = "ROC curve", print.auc = TRUE,
     auc.polygon = TRUE, max.auc.polygon = TRUE, grid = TRUE, legacy.axes = TRUE,
     xlab = "False positive rate", ylab = "True positive rate")
abline(v = 1-opt_thr, col = "red", lty = 2)
```



Non-parametric model

K-Nearest Neighbors (KNN) is a non-parametric classification algorithm that we are considering as the final model for our dataset. It is a simple yet effective algorithm that makes predictions based on the similarity of a new observation to its k nearest neighbors in the training data. In KNN, the value of k represents the number of neighbors to consider when making a prediction. When a new observation needs to be classified, the algorithm finds the k nearest neighbors in the training data based on a chosen distance metric, such as Euclidean distance. The majority class among these k neighbors is then assigned to the new observation.

KNN

Unbalanced dataset

Before training our final knn model, we do a cross-validation to find the optimal value of k.

```
k_grid <- seq(1, 10, by = 2)

# Cross-validation on different k values
knn_model <- train(dati_train, dati_train$HeartDisease , method = "knn",
                   trControl = trainControl(method = "cv", number = 10),
                   tuneGrid = data.frame(k = k_grid))

# Optimal value of k
best_k <- knn_model$bestTune$k
```

We find by this cross-validation that the optimal value of k is equal to 3.

```
best_k
```

```
## [1] 3
```

We can now define the knn model and evaluate its performance using the error rate and F1 score. By looking at these results we can say that this model does not bring improvements in terms of F1 score, as well as AUC value.

```
library(class)
knn_u <- knn(train = dati_train[,-1], test = dati_test[,-1],
             cl = dati_train$HeartDisease, k = best_k)

# error rate
mean(knn_u != dati_test$HeartDisease) * 100
```

```
## [1] 10.17274
```

```
# Confusion matrix
conf_mat <- table(dati_test$HeartDisease, knn_u)
conf_mat
```

```
##    knn_u
##         0     1
##   0 71049  2055
##   1  6078   767
```

```
# true positive rate
TPR <- conf_mat[2, 2]/sum(conf_mat[2, ])
TPR
```

```
## [1] 0.1120526
```

```
# true negative rate
TNR <- conf_mat[1, 1]/sum(conf_mat[1, ])
TNR
```

```
## [1] 0.9718894
```

```
# precision
PREC <- conf_mat[2, 2]/sum(conf_mat[, 2])
PREC
```

```
## [1] 0.2717931
```

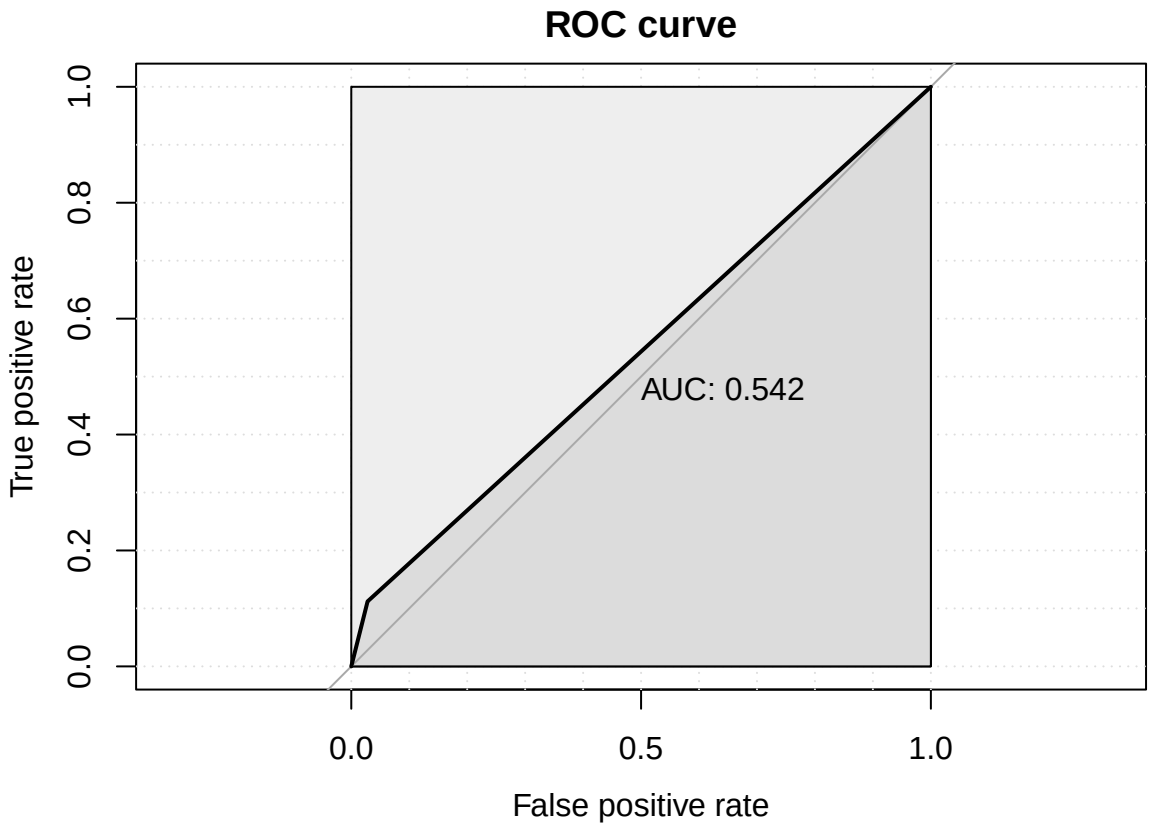
```
# recall
REC <- conf_mat[2, 2]/sum(conf_mat[2, ])
REC
```

```
## [1] 0.1120526
```

```
# F1 score
F1_knnu <- 2 * (PREC * REC)/(PREC + REC)
F1_knnu
```

```
## [1] 0.1586842
```

```
# ROC curve
library(pROC)
roc_u <- roc(dati_test$HeartDisease, as.numeric(knn_u))
auc_u <- auc(roc_u)
plot(roc_u, main = "ROC curve", print.auc = TRUE, auc.polygon = TRUE,
     max.auc.polygon = TRUE, grid = TRUE, legacy.axes = TRUE,
     xlab = "False positive rate", ylab = "True positive rate")
```



Balanced dataset

We define the knn model for the balanced dataset as we did for the unbalanced dataset so, before training the model, we do a cross-validation to find the optimal value of k.

```
k_grid <- seq(1, 10, by = 2)

# Cross-validation on different k values
knn_model <- train(dati_balanced_train, as.factor(dati_balanced_train$HeartDisease),
                   method = "knn",
                   trControl = trainControl(method = "cv", number = 10),
                   tuneGrid = data.frame(k = k_grid))

# Optimal value of k
best_k <- knn_model$bestTune$k
```

We find by this cross-validation that the optimal value of k is equal to 7.

```
best_k
```

```
## [1] 7
```

The model can now be trained. This model brings no improvements with respect to the previous models.

```
library(class)
knn_b <- knn(train = dati_balanced_train[,-1], test = dati_test[,-1],
             cl = dati_balanced_train$HeartDisease, k = best_k)

# error rate
mean(knn_b != dati_test$HeartDisease) * 100
```

```
## [1] 70.55748
```

```
# Confusion matrix
conf_mat <- table(dati_test$HeartDisease, knn_b)
conf_mat
```

```
##    knn_b
##         0     1
##   0 21729 51375
##   1  5035  1810
```

```
# true positive rate
TPR <- conf_mat[2, 2]/sum(conf_mat[2, ])
TPR
```

```
## [1] 0.2644266
```

```
# true negative rate
TNR <- conf_mat[1, 1]/sum(conf_mat[1, ])
TNR
```

```
## [1] 0.2972341
```

```
# precision
PREC <- conf_mat[2, 2]/sum(conf_mat[, 2])
PREC
```

```
## [1] 0.03403215
```

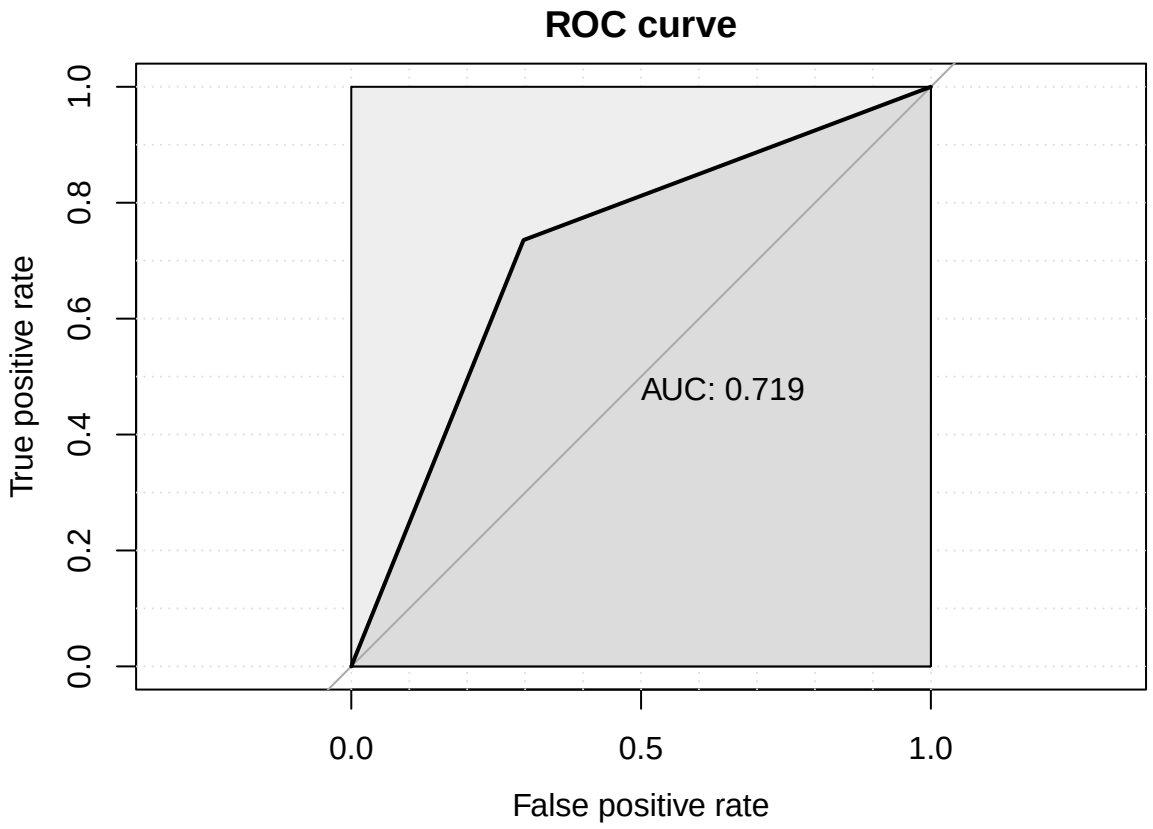
```
# recall
REC <- conf_mat[2, 2]/sum(conf_mat[2, ])
REC
```

```
## [1] 0.2644266
```

```
# F1 score
F1_knnb <- 2 * (PREC * REC)/(PREC + REC)
F1_knnb
```

```
## [1] 0.06030318
```

```
# ROC curve
library(pROC)
roc_b <- roc(dati_test$HeartDisease, as.numeric(knn_b))
auc_b <- auc(roc_b)
plot(roc_b, main = "ROC curve", print.auc = TRUE, auc.polygon = TRUE,
     max.auc.polygon = TRUE, grid = TRUE, legacy.axes = TRUE,
     xlab = "False positive rate", ylab = "True positive rate")
```



Comparing models

Unbalanced dataset

We will now compare the performances of the models on the unbalanced dataset. By the plot and the summary table we can notice how the glm model performs better than the others in terms of AUC, while the LDA and QDA perform better in terms of F1 score.

```
# compare all the models for unbalanced dataset

# ROC curve
library(dplyr)
library(pROC)
library(ggplot2)
pred.glm.best <- predict(glm10u, dati_test[,-1], type="response")
pred.lda.best <- lda_u.post[,2]
pred.qda.best <- qda_u.post[,2]
pred.knn <- knn_u
pred.ridge <- ridge_u.predmin
pred.lasso <- lasso_u.predmin

prediction <- tibble(truth=as.factor(dati_test$HeartDisease))
prediction <- prediction %>% mutate(pred=as.numeric(pred.glm.best)) %>%
  mutate(model="best GLM") %>%
  add_row(truth=as.factor(dati_test$HeartDisease), pred=as.numeric(pred.lda.best),
          model="LDA") %>%
  add_row(truth=as.factor(dati_test$HeartDisease), pred=as.numeric(pred.qda.best),
```

```
          model="QDA") %>%
  add_row(truth=as.factor(dati_test$HeartDisease), pred=as.numeric(pred.knn),
          model="KNN") %>%
  add_row(truth=as.factor(dati_test$HeartDisease), pred=as.numeric(pred.ridge),
          model="Ridge") %>%
  add_row(truth=as.factor(dati_test$HeartDisease), pred=as.numeric(pred.lasso),
          model="LASSO")

#roc <- prediction %>% group_by(model) %>%
  #roc_curve(truth, pred, event_level="second") %>%
 # autoplot()
#roc
```

Table 5: Metrics - Unbalanced models

	AUC values	F1 scores
GLM	0.8280971	0.1341228
LDA	0.8219570	0.2785238
QDA	0.8212459	0.2785238
Ridge	0.8279756	0.0987193
Lasso	0.8219570	0.1403152
KNN	0.5419710	0.1586842

Balanced dataset

For the balanced dataset we can notice how the glm model performs better in terms of AUC value while the ridge has the higher F1 score. Being the F1 scores very close to each others, we are more confident to select the glm model as the best one by also considering the AUC value.

```
# compare all the models for balanced dataset

# ROC curves

pred.glm.best_b <- predict(glm10, dati_test[,-1], type="response")
pred.lda.best_b <- lda_b.post[,2]
pred.qda.best_b <- qda_b.post[,2]
pred.knn_b <- knn_b
pred.ridge_b <- ridge_b.predmin
pred.lasso_b <- lasso_b.predmin

prediction_b <- tibble(truth=as.factor(dati_test$HeartDisease))
prediction_b <- prediction_b %>% mutate(pred=as.numeric(pred.glm.best_b)) %>%
  mutate(model="best GLM") %>%
  add_row(truth=as.factor(dati_test$HeartDisease), pred=as.numeric(pred.lda.best_b),
          model="LDA") %>%
  add_row(truth=as.factor(dati_test$HeartDisease), pred=as.numeric(pred.qda.best_b),
          model="QDA") %>%
  add_row(truth=as.factor(dati_test$HeartDisease), pred=as.numeric(pred.knn_b),
          model="KNN") %>%
  add_row(truth=as.factor(dati_test$HeartDisease), pred=as.numeric(pred.ridge_b),
          model="Ridge") %>%
  add_row(truth=as.factor(dati_test$HeartDisease), pred=as.numeric(pred.lasso_b),
          model="LASSO")
```

```
#roc_b <- prediction_b %>% group_by(model) %>%
 # roc_curve(truth, pred, event_level="second") %>%
  #autoplot()
#roc_b
```

Table 6: Metrics - Balanced models

	AUC values	F1 scores
GLM	0.8284663	0.1420924
LDA	0.8275273	0.1367604
QDA	0.8145059	0.0932506
Ridge	0.8268174	0.0538589
Lasso	0.8283280	0.0505628
KNN	0.7191697	0.0603032

Residual analysis

We are now comparing all the residuals as well as MSE and MAE of our models. Residuals provide important information about the performance and accuracy of a predictive model, as they represent the discrepancies between the predicted values and the actual values in the dataset. By analyzing the residuals, we can assess the goodness-of-fit of the model and evaluate its predictive capabilities. The minimum and maximum residuals give us insights into the range of errors made by the model. A smaller range indicates that the model is making more consistent predictions, while a larger range suggests more variability in the predictions. The mean residual provides an overall measure of the average prediction error. It represents the average discrepancy between the predicted and actual values. A mean residual close to zero indicates that, on average, the model's predictions are accurate. However, if the mean residual deviates significantly from zero, it suggests a systematic bias in the model's predictions. MSE (Mean Squared Error) and MAE (Mean Absolute Error) are metrics that quantify the accuracy of a predictive model. MSE measures the average squared difference between the predicted and actual values, providing a measure of overall prediction error. MAE, on the other hand, measures the average absolute difference, giving us a sense of the average magnitude of the errors. We take into account this error because it is less sensible to outliers and each observation contributes equally to it. Lower values of MSE and MAE indicate better model performance, as they reflect smaller prediction errors. By examining the minimum, maximum, and mean residuals, along with MSE and MAE, we gain insights into the variability, bias, and overall accuracy of the model's predictions. These measures help us assess the model's performance and determine its suitability for the intended task. When analyzing the unbalanced dataset, we observe that only the ridge and lasso models exhibit a mean residual close to zero, indicating potentially more accurate predictions compared to the other models. Despite this, the error table reveals generally low values across all models, with ridge and lasso performing particularly well in terms of precise predictions. In the case of the balanced dataset, we find that the mean residuals for all models are not as close to zero, suggesting accurate predictions only on the GLM. However, it is worth noting that the GLM model displays a larger interval of residuals, which could imply less precise predictions. By examining the MSE and MAE values, we can confirm that the GLM model appears to be less accurate, as it exhibits higher error values compared to the other models, which still have quite high values.

```
# residuals
res.glm10u <- residuals(glm10u) #glm unb
res.glm10 <- residuals(glm10) #glm bal
res.lda_u <- dati_test$HeartDisease - lda_u.post[,2] #lda unb
res.lda_b <- dati_test$HeartDisease - lda_b.post[,2] #lda bal
res.qda_u <- dati_test$HeartDisease - qda_u.post[,2] #qda unb
res.qda_b <- dati_test$HeartDisease - qda_b.post[,2] #qda bal
res.ridge_u <- dati_test$HeartDisease - ridge_u.predmin #ridge unb
res.ridge_b <- dati_test$HeartDisease - ridge_b.predmin #ridge b
```

Table 7: Residual table - Unbalanced models

	Min	Max	Mean
GLM	-1.9696994	-1.9696994	-0.1443781
LDA	-0.9989677	0.9946669	-0.0046114
QDA	-0.9989677	0.9946669	-0.0046114
Ridge	-0.8487290	0.9740942	-0.0002257
Lasso	-0.7958625	0.9969847	-0.0002648

Table 8: Residual table - Balanced models

	Min	Max	Mean
GLM	-2.9569617	2.8963514	-0.0052140
LDA	-0.9835007	0.9914764	-0.5472783
QDA	-1.0000000	1.0000000	-0.7244625
Ridge	-0.9613577	0.9944226	-0.5398900
Lasso	-0.9837261	0.9914445	-0.5497567

```
res.lasso_u <- dati_test$HeartDisease - lasso_u.predmin #lasso unb
res.lasso_b <- dati_test$HeartDisease - lasso_b.predmin #lasso unb

# error results
mse_unb <- rbind(mse_glm_u <- mean(res.glm10u^2),
  mse_lda_u <- mean(res.lda_u^2),
  mse_qda_u <- mean(res.qda_u^2),
  mse_ridge_u <- mean(res.ridge_u^2),
  mse_lasso_u <- mean(res.lasso_u^2))

mse_bal <- rbind(
  mse_glm_b <- mean(res.glm10^2),
  mse_lda_b <- mean(res.lda_b^2),
  mse_qda_b <- mean(res.qda_b^2),
  mse_ridge_b <- mean(res.ridge_b^2),
  mse_lasso_b <- mean(res.lasso_b^2))

mae_unb <- rbind(mae_glm_u <- mean(abs(res.glm10u)),
             mae_lda_u <- mean(abs(res.lda_u)),
             mae_qda_u <- mean(abs(res.qda_u)),
             mae_ridge_u <- mean(abs(res.ridge_u)),
             mae_lasso_u <- mean(abs(res.lasso_u)))

mae_bal <- rbind(
             mae_glm_b <- mean(abs(res.glm10)),
             mae_lda_b <- mean(abs(res.lda_b)),
             mae_qda_b <- mean(abs(res.qda_b)),
             mae_ridge_b <- mean(abs(res.ridge_b)),
             mae_lasso_b <- mean(abs(res.lasso_b)))
```

Table 9: Errors table - Unbalanced models

	MSE	MAE
GLM	0.4615414	0.4608556
LDA	0.0749447	0.1257165
QDA	0.0749447	0.1257165
Ridge	0.0683614	0.1396739
Lasso	0.0670936	0.1335185

Table 10: Errors table - Balanced models

	MSE	MAE
GLM	1.0095954	0.8949366
LDA	0.5052514	0.6629722
QDA	0.8047152	0.8178350
Ridge	0.4740931	0.6510354
Lasso	0.4995197	0.6629065

Conclusions

By taking into account the results of each model for the unbalanced dataset, we would choose the LDA model, as it brings the higher F1 score, a good AUC value, better residuals and error values. As of the balanced dataset, we would consider the GLM model, as it has the best F1 score and the highest AUC value. If we compare the overall results, we notice that the balancing brings an improvement only on the GLM model. We also notice how on the unbalanced dataset the precision is average and the recall is quite low, while in the balanced dataset the precision is low and the recall is quite high. The difference in precision and recall between the datasets may be due to the distribution of the input data and its effect on the model's performance. In the case of the unbalanced dataset, where the minority class (has a heart disease) is significantly underrepresented compared to the majority class (does not have a heart disease), the model may struggle to correctly detect the positive class. Consequently, the precision could be low as the model might make erroneous positive predictions, resulting in a high number of false positives. Additionally, the recall could be even lower as the model may miss many positive cases, leading to a high number of false negatives in its predictions. In the case of the balanced dataset, where the data distribution between positive and negative classes is more evenly distributed, the model can learn the distinctive patterns of the positive class more accurately. As a result, the recall could increase as the model can identify a higher number of positive cases correctly. However, due to the balanced nature of the classes, the precision may decrease as the model becomes more cautious in making positive predictions to avoid false positives. In summary, the models demonstrate a high AUC value, indicating good overall classification ability. However, they also exhibit a low F1 score, suggesting room for improvement in accurately identifying people with a heart disease and achieving a balanced classification performance.