ST442 Project

Rabin Paudel

12/01/2021

Abstract

The name of data set of this project is *Heart Failure Prediction dataset*, and sorce is kiggle.com. In this data set, there are twelve prediction variable to predict death events. According to the description of data, "cardiovascular diseases (CVDs) are the number one cause of death globally taking an estimate 17.9 million lives each year which accounts for 31% of all deaths worldwise". First, I am going to analyze Multiple logistic regression to find the best model to predict death event cause by heart failure. In the model selection process, I am going to utilize forward selection, backward elimination, and stepwise selection process it terms of non significant (higher p-value) first eliminate. For my first pririty minimum value of AIC(Akaike information Criterion) and BIC(Bayesian Information Criterion). The Formula of AIC = -2(lnL) + 2k and BIC = -2(lnL) + 2k.In this project, I am going to present graphical presentation with diffrent way. We considered using a linear regression model to represent these probabilities: $p(Y_i) = \beta_0 + \beta_i X_i$. In logistic regression, we use the logistic function,

$$log(\frac{p(X_i)}{1-p(X_i)}) = \beta_0 + \beta_i X_i$$

The ligistic regression gives the assumption of the form of probability of failure to allow us to predict a plausible probability.

probability of failure simple ligistic regression or we can add multiple case more variable

$$p[failure] = [\frac{exp(\beta_0 + \beta_1 X_i)}{(1 + exp(\beta_0 + \beta_1 X_i)}]$$

In summary, logistic regression can be thought of as either.

(a) Empirical risk minimization where we replace 0-1 loss $I(g(X_i \neq Y_i))$ with logistic $log(1 + exp(-Y_i f(X_i)))$

(b) Classification where we model $\eta(x) = P[Y_i|X = x] = exp(\frac{f(x)}{1+exp(f(x))})$

(c) maximum likelihood estimation for a collection of independent Bernoulli random variables, Y_i with $pr[Y_i = 1] = \frac{exp(f(X_i))}{(1+exp(f(X_i)))}$

Data Analysis: Read Data and Visual Representation of data

```
data <- read.csv("~/Desktop/heart_failure_clinical_records_dataset.csv")
# To Know the variable and type
names(data)
```

Read the Data with csv file from the Desktop (download in Desktop).

```
##  [1] "age"                      "anaemia"
##  [3] "creatinine_phosphokinase" "diabetes"
##  [5] "ejection_fraction"        "high_blood_pressure"
##  [7] "platelets"                "serum_creatinine"
##  [9] "serum_sodium"             "sex"
## [11] "smoking"                  "time"
## [13] "DEATH_EVENT"
```

```
str(data)
```

```
## 'data.frame':    299 obs. of  13 variables:
##  $ age                     : num  75 55 65 50 65 90 75 60 65 80 ...
##  $ anaemia                 : int  0 0 0 1 1 1 1 1 0 1 ...
##  $ creatinine_phosphokinase: int  582 7861 146 111 160 47 246 315 157 123 ...
##  $ diabetes                : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ ejection_fraction       : int  20 38 20 20 20 40 15 60 65 35 ...
##  $ high_blood_pressure     : int  1 0 0 0 0 1 0 0 0 1 ...
##  $ platelets               : num  265000 263358 162000 210000 327000 ...
##  $ serum_creatinine        : num  1.9 1.1 1.3 1.9 2.7 2.1 1.2 1.1 1.5 9.4 ...
##  $ serum_sodium            : int  130 136 129 137 116 132 137 131 138 133 ...
##  $ sex                     : int  1 1 1 1 0 1 1 1 0 1 ...
##  $ smoking                 : int  0 0 1 0 0 1 0 1 0 1 ...
##  $ time                    : int  4 6 7 7 8 8 10 10 10 10 ...
##  $ DEATH_EVENT             : int  1 1 1 1 1 1 1 1 1 1 ...
```

Column Variable name:

Age:

anaemia: Decrease of red blood cells or hemoglobin (boolean)

creatinine_phosphokinase: Level of the CPK enzyme in the blood (mcg/L)

diabetes: if the patient has diabetes (boolean)

ejection_fraction: Percentage of blood leaving the heart at each contraction (percentage)

high_blood_pressure: If the patient has hypertension(boolean)

platelets: Platelets in the blood (kiloplatelets/mL)

serum_creatinine: Level of serum creatinine in the blood (mg/dL)

serum_sodium: Level of serum sodium in the blood(mEq/L)

sex: Woman or man (binary)

smoking: If the patient smokes or not (boolean)

time: Follow-up perriod (days)

DEATH_EVENT: If the patient deceased during the follow-up period (boolean)

```r
library(dplyr)
```

Refine or modified the Data Changing the name and variable type.

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
data[data$sex == 0,]$sex <- "Female"
data[data$sex == 1,]$sex <- "Male"
data[data$anaemia == 0,]$anaemia <- "No"
data[data$anaemia == 1,]$anaemia <- "Yes"
data[data$high_blood_pressure == 1,]$high_blood_pressure <- "Yes"
data[data$high_blood_pressure == 0,]$high_blood_pressure <- "No"
data[data$diabetes == 0,]$diabetes <- "No"
data[data$diabetes == 1,]$diabetes <- "Yes"
data[data$smoking == 0,]$smoking <- "No"
data[data$smoking == 1,]$smoking <- "Yes"
data$DEATH_EVENT <- ifelse(test = data$DEATH_EVENT == 0, yes = "Survived", no = "Dead")
data$DEATH_EVENT <- as.factor(data$DEATH_EVENT)
```

```r
library(ggplot2)
library(tidyverse)
```

Creating graph to compression between the factor variable with DEATH_EVENT or like a visual representation of the data. Those bar diagram (visualization figure) clearly shows that not smoking, not high blood pressure, not diabetes, and less anaemia higher survival rate.

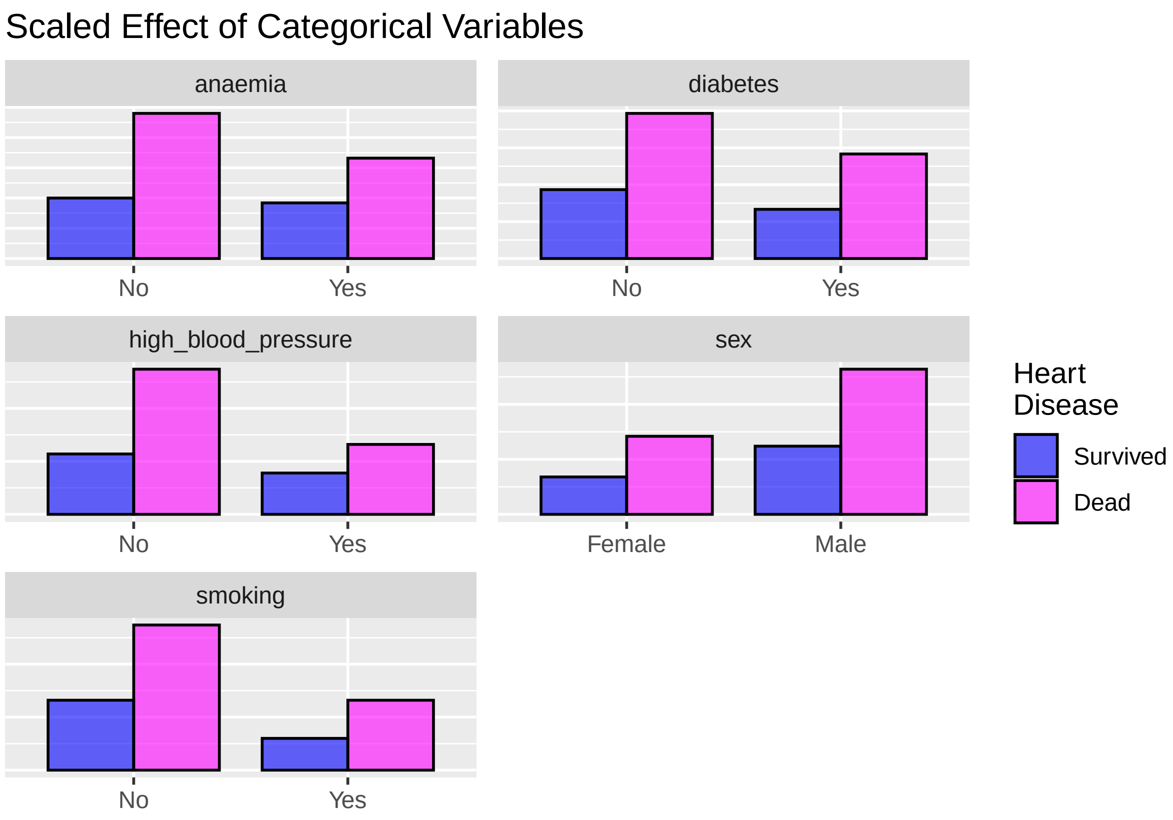
```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.6     v purrr   0.3.4
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1
```

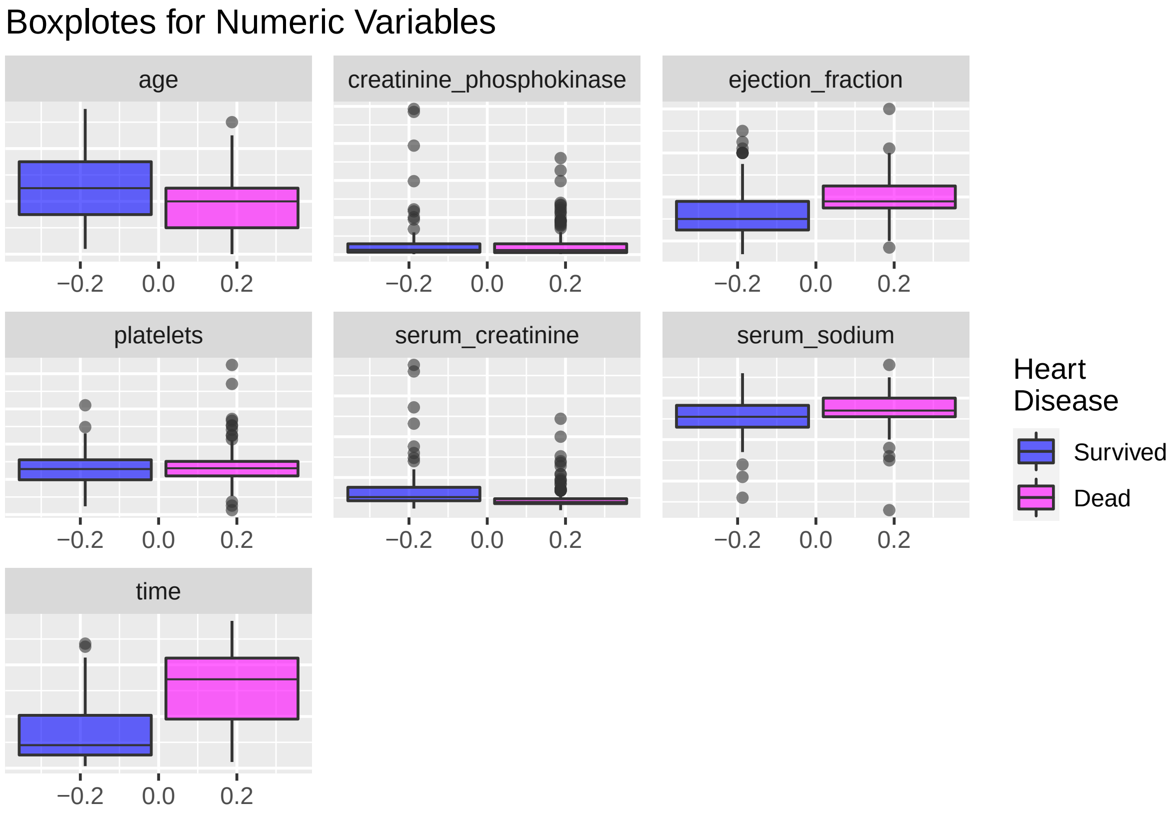
```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
data1 <- data.frame(data) %>%
  select (anaemia,diabetes,high_blood_pressure, sex, smoking,DEATH_EVENT) %>%
  gather(key = "key", value = "value", -DEATH_EVENT)
```

```r
  #Visualize with bar plot
data1 %>%
  ggplot(aes(value)) +
  geom_bar(aes(x        = value,
               fill     = DEATH_EVENT),
           alpha    = .6,
           position = "dodge",
           color    = "black",
           width    = .8
  ) +
  labs(x = "",
       y = "",
       title = "Scaled Effect of Categorical Variables") +
  theme(
    axis.text.y  = element_blank(),
    axis.ticks.y = element_blank()) +
  facet_wrap(~ key, scales = "free", nrow = 3) +

scale_fill_manual(
          values = c("#0000FF", "#FF00FF"),
         name   = "Heart\nDisease",
         labels = c("Survived", "Dead"))
```



```
#Visualize with box plot
data2 <- data %>% select(age, creatinine_phosphokinase, ejection_fraction, platelets, serum_creatinine,
         value = "value",
         -DEATH_EVENT)

data2 %>%
  ggplot(aes(y = value)) +
  geom_boxplot(aes(fill = DEATH_EVENT),
           alpha    = .6,
           fatten   = 0.7
  ) +
  labs(x = "",
       y = "",
       title = "Boxplotes for Numeric Variables") +
  theme(
    axis.text.y  = element_blank(),
    axis.ticks.y = element_blank()) +
  facet_wrap(~ key, scales = "free", nrow = 3) +

scale_fill_manual(
          values = c("#0000FF", "#FF00FF"),
         name   = "Heart\nDisease",
         labels = c("Survived", "Dead"))
```



Those box plot diagram (visualization figure) clearly shows that less age, higher ejection_fraction, higher serum_sodium, and more time follow up are higher survival rate.

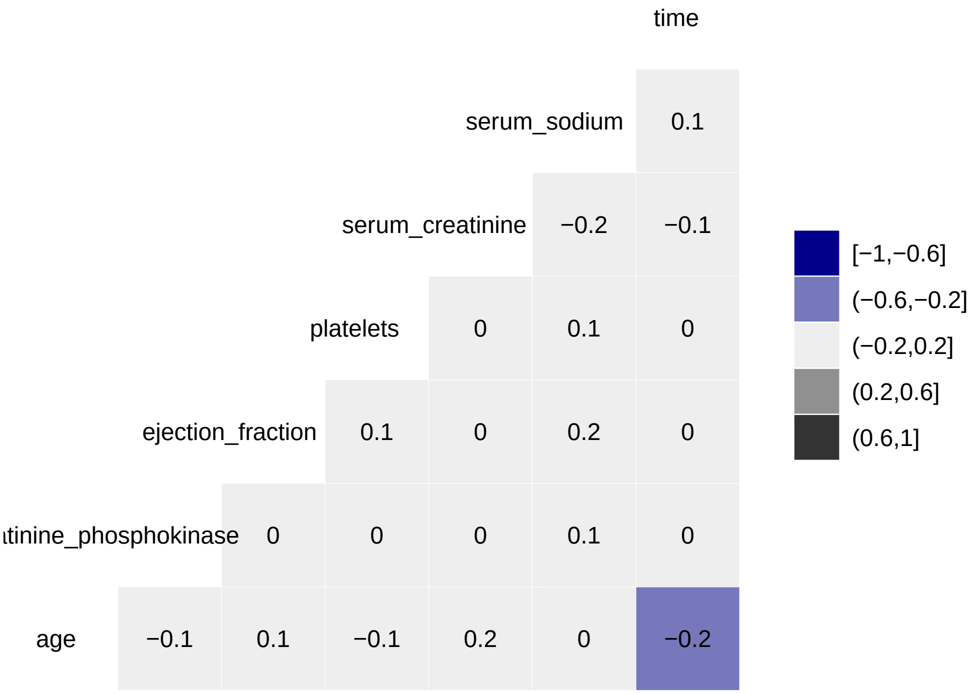
```
library(GGally)
```

Highly correlated variables can lead to overly complicated models. So, ggcorr() function from GGally package provides a nice, clean correlation matrix of the numeracic variable.

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
data %>% ggcorr(high = "gray20",
                low = "blue4",
                label      = TRUE,
                hjust      = .75,
                size       = 3,
                label_size = 3,
                nbreaks    = 5
                ) +
  labs(title = "Correlation Matrix",
  subtitle = "Pearson Method Using Pairwise Obervations")
```

```
## Warning in ggcorr(., high = "gray20", low = "blue4", label = TRUE, hjust =
## 0.75, : data in column(s) 'anaemia', 'diabetes', 'high_blood_pressure', 'sex',
## 'smoking', 'DEATH_EVENT' are not numeric and were ignored
```


Correlation Matrix

Pearson Method Using Pairwise Obervations

Way of Model Bulding:

I like to describe the way of model-building multiple Logistic regression such as Forward Selection, Backward Elimation, and Stepwise Selection Sequence. Forward Selection: Inter the variables in order of terms with highest score statistic. Backward Elimination: In this process, starts with all terms in the model and droups them out in order according to the smallest wald statistic. Stepwise: Just like forward selection except that variables can be deleted from the model if p-values are above slstay. I am going to use Model-Producing Methods such as:* Akaike Information Criterion AIC = -2(lnL) + 2k Where k = number of terms in the model (including intercept) ** Bayesian Information Criterion** BIC = -2(lnL)+(ln n)k.So, we want AIC and BIC is smaller.**

Fit multiple logistic regression

```
fit1 <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + diabetes + ejection_fraction + hig
summary(fit1)
```

```
##
```

```
## Call:
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##     diabetes + ejection_fraction + high_blood_pressure + platelets +
##     serum_creatinine + serum_sodium + sex + smoking + time, family = binomial(link = "logit"),
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6668  -0.4466   0.2401   0.5706   2.1848
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -1.018e+01  5.657e+00  -1.801 0.071774 .
## age                      -4.742e-02  1.580e-02  -3.001 0.002690 **
## anaemiaYes                7.470e-03  3.605e-01   0.021 0.983467
## creatinine_phosphokinase -2.222e-04  1.779e-04  -1.249 0.211684
## diabetesYes              -1.451e-01  3.512e-01  -0.413 0.679380
## ejection_fraction         7.666e-02  1.633e-02   4.695 2.67e-06 ***
## high_blood_pressureYes    1.027e-01  3.587e-01   0.286 0.774688
## platelets                 1.200e-06  1.889e-06   0.635 0.525404
## serum_creatinine         -6.661e-01  1.815e-01  -3.670 0.000242 ***
## serum_sodium              6.698e-02  3.974e-02   1.686 0.091855 .
## sexMale                   5.337e-01  4.139e-01   1.289 0.197299
## smokingYes                1.349e-02  4.126e-01   0.033 0.973915
## time                      2.104e-02  3.014e-03   6.981 2.92e-12 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 375.35  on 298  degrees of freedom
## Residual deviance: 219.55  on 286  degrees of freedom
## AIC: 245.55
##
## Number of Fisher Scoring iterations: 6
```

```
BIC(fit1)
```

```
## [1] 293.6599
```

```
fit2 <- glm(DEATH_EVENT ~ age + ejection_fraction + serum_sodium + time, family = binomial(link = "logi
summary(fit2)
```

The process backward elimination, stepwise selection sequence, and forward process I found the best multiple ligistic regression model is:

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_sodium +
##     time, family = binomial(link = "logit"), data = data)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1984  -0.5454   0.2625   0.6472   2.1076
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -12.497986   5.241047  -2.385  0.01710 *
## age                 -0.045038   0.014650  -3.074  0.00211 **
## ejection_fraction    0.068004   0.015265   4.455 8.39e-06 ***
## serum_sodium         0.082437   0.037262   2.212  0.02694 *
## time                 0.020331   0.002768   7.344 2.07e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 375.35  on 298  degrees of freedom
## Residual deviance: 239.56  on 294  degrees of freedom
## AIC: 249.56
##
## Number of Fisher Scoring iterations: 5
```

```r
BIC(fit2)
```

```
## [1] 268.0657
```

The value age, ejection_fraction, serum_sodium, and time are highly significant and AIC and BIC are also small. So, I found Best Model to predict is:

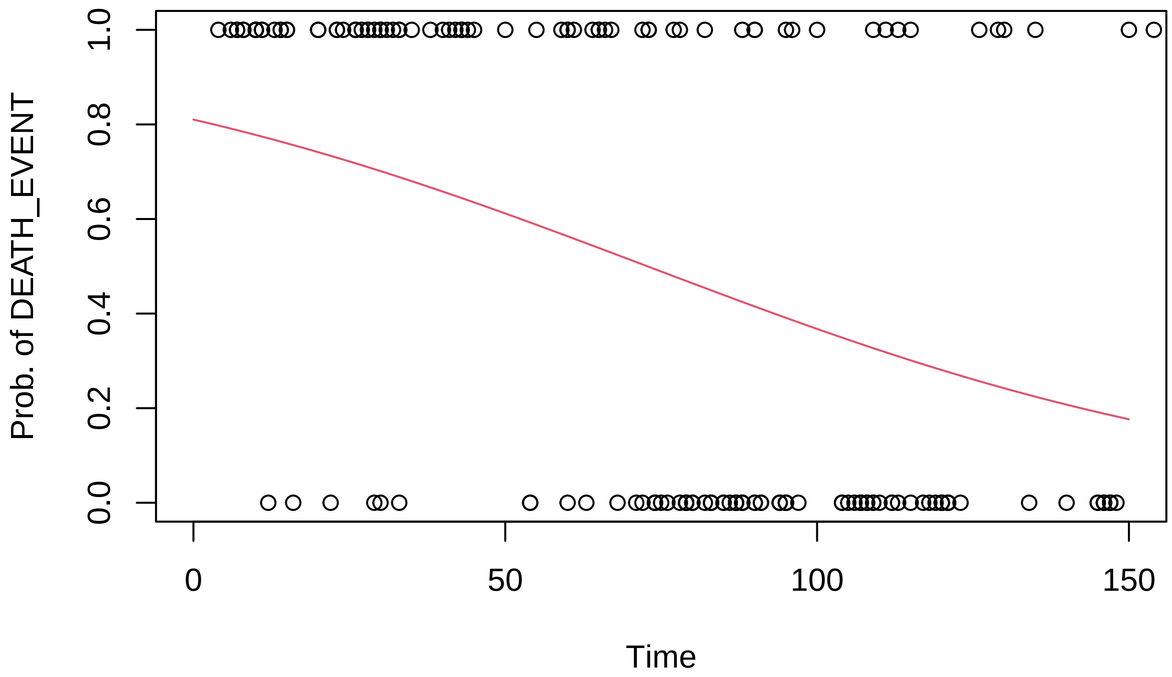
DEATH_EVENT = 12.498 + 0.045 * age - 0.068 * ejection_fraction - 0.082 * serum_sodium - 0.020 * time.

```r
library("faraway")
```

I try to find to predict prob. of DEATH_EVENT vs time with ligistic regression to look the data with ligistic curve.

```
##
## Attaching package: 'faraway'

## The following object is masked from 'package:GGally':
##
##     happy
```

```r
df <- data %>% mutate(x2 = ifelse(data$DEATH_EVENT=="Dead",1,0))
plot(x = df$time, y = df$x2, ylim = c(0, 1),xlim = c(0,150), xlab = "Time",
ylab = "Prob. of DEATH_EVENT")
x <- seq(0, 150, 1)
logitmod <- glm(cbind(df$x2) ~ time, family = binomial, data = df)
lines(x, ilogit(coef(logitmod)[1] + coef(logitmod)[2] * x), col = 2)
```



Some theorical expression of Classification and regression trees suppose we are given $(X_i, Y_i)_{i=1}^n$ with $X_i = (X_{i1}, ..., X_{ip})$ in $\mathbb{R}^p$ and $Y_i \in \mathbb{R}$. Then a regression tree for predicting Y given $X = x$ is a model of the form.

$$f(x) = \sum_{m=1}^{\mathbb{M}} {}_m\mathbb{I}(x \in \mathbb{R}_{\gtrdot})$$

Where $R_1, R_2, ..., R_M$ are partition regions of the form.

```


library(rattle)
```

To create the decision tree of the given data set and try to predict best model

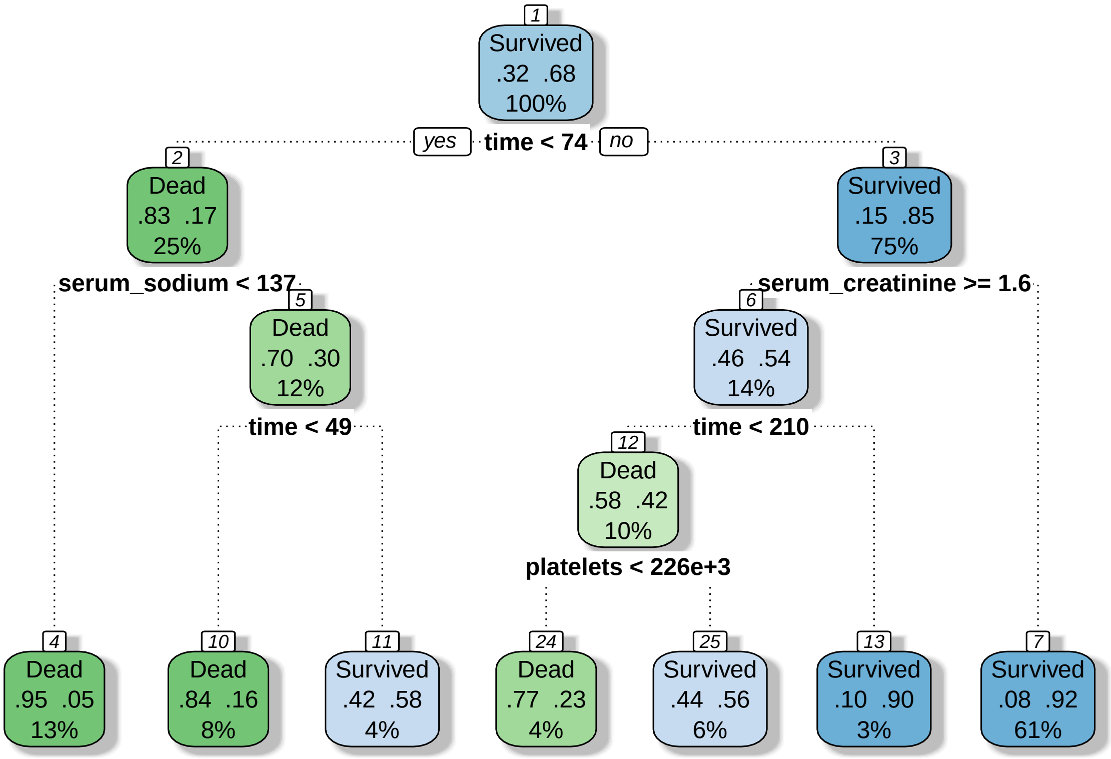
```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

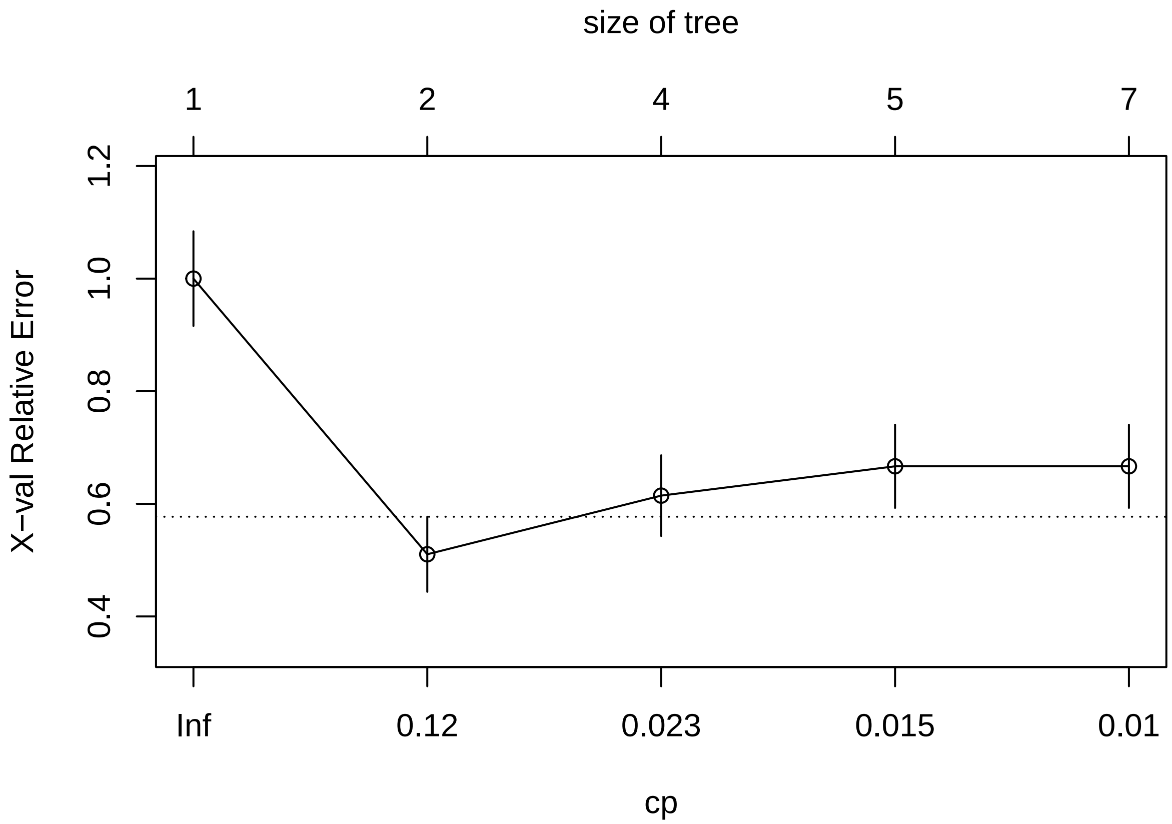
```
library(readr)
library(rpart)
```

```
##
## Attaching package: 'rpart'

## The following object is masked from 'package:faraway':
##
##     solder
```

```
heart.tree <- rpart(DEATH_EVENT ~ . , data = data)
fancyRpartPlot(heart.tree, sub = "")
```



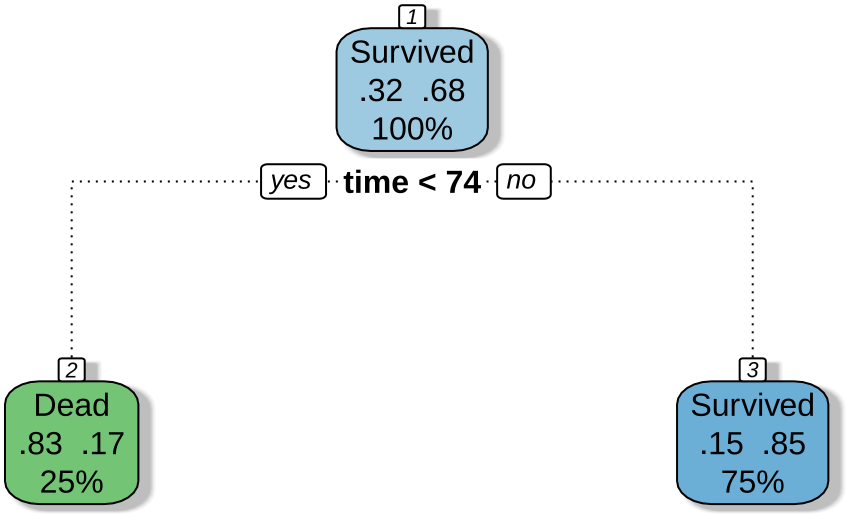
```
plotcp(heart.tree)
```



```
bestcp <- heart.tree$cptable[which.min(heart.tree$cptable[,"xerror"]),"CP"]
bestcp
```

```
## [1] 0.02604167
```

```
pruned.tree <- prune(heart.tree, cp = bestcp)
fancyRpartPlot(pruned.tree, sub = "")
```



```
conf.matrix <- table(data$DEATH_EVENT, predict(pruned.tree,type="class"))
rownames(conf.matrix) <- paste("Actual", rownames(conf.matrix), sep = ":")
colnames(conf.matrix) <- paste("Pred", colnames(conf.matrix), sep = ":")
conf.matrix
```

I am going to create the conf.matrix to analysis the data.

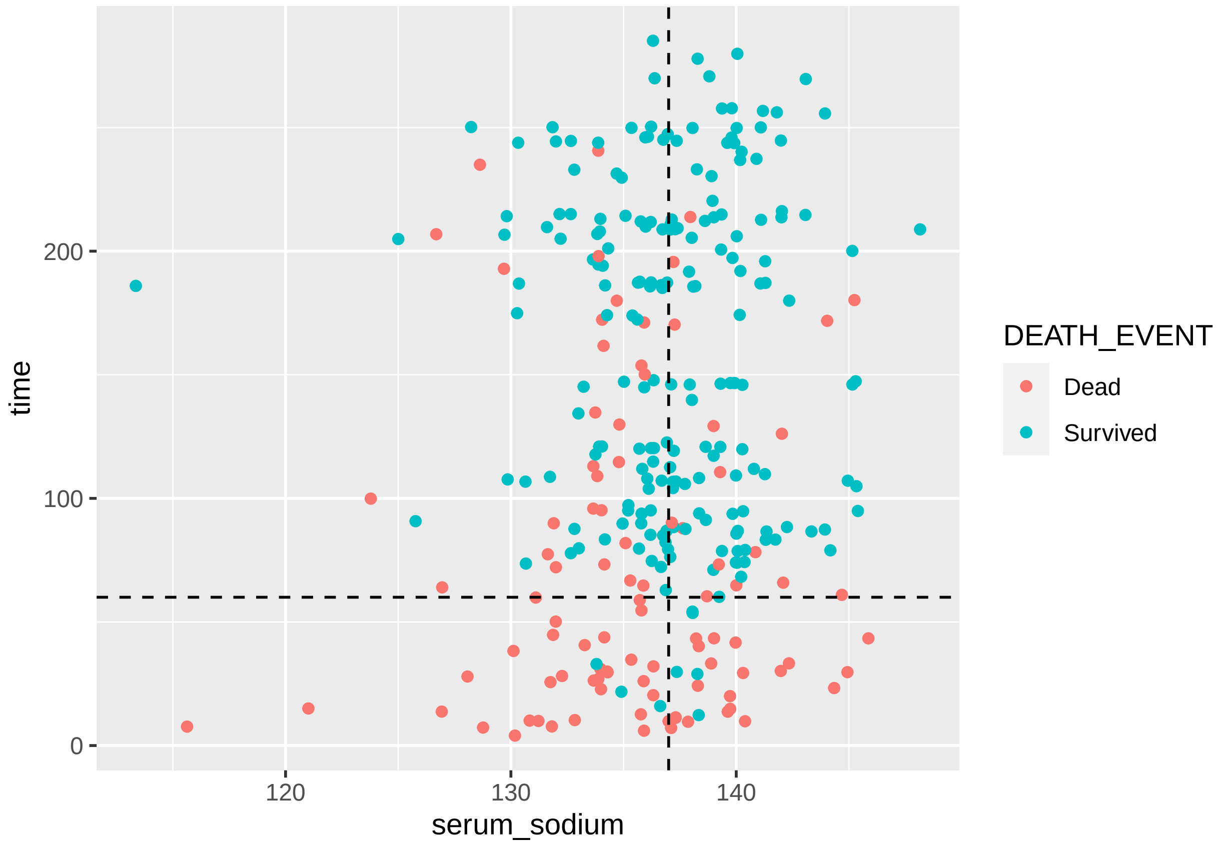
```
##
##                    Pred:Dead Pred:Survived
##   Actual:Dead             63            33
##   Actual:Survived         13           190
```

```
accuracy <- (conf.matrix[2,1] + conf.matrix[1,2])/sum(conf.matrix)
accuracy
```

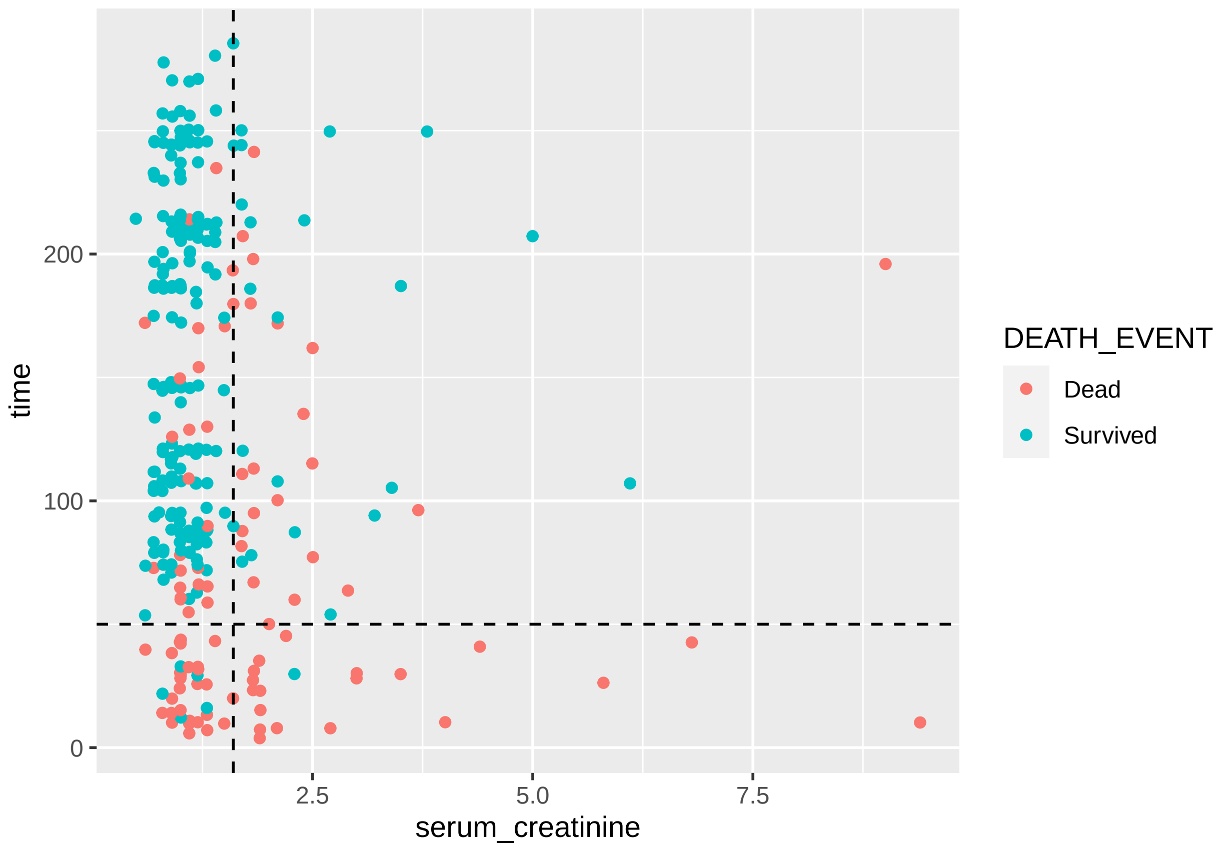
Visualized the data to using ggplot and plot geom_point(), geom_vline(), and geom_hline() to relation between time, serum_sodium, and serum_creatinine. So, these variable are highly significant to DEATH_EVENT variable.

```
## [1] 0.1538462
```

```
ggplot(data, aes(x = serum_sodium, y = time, color = DEATH_EVENT)) +
geom_point(position = "jitter") +
geom_vline(xintercept = 137, lty = 2) +
geom_hline(yintercept = 60, lty = 2)
```



```
ggplot(data, aes(x = serum_creatinine, y = time, color = DEATH_EVENT)) +
  geom_point(position = "jitter") +
  geom_vline(xintercept = 1.6, lty = 2) +
  geom_hline(yintercept = 50, lty = 2)
```

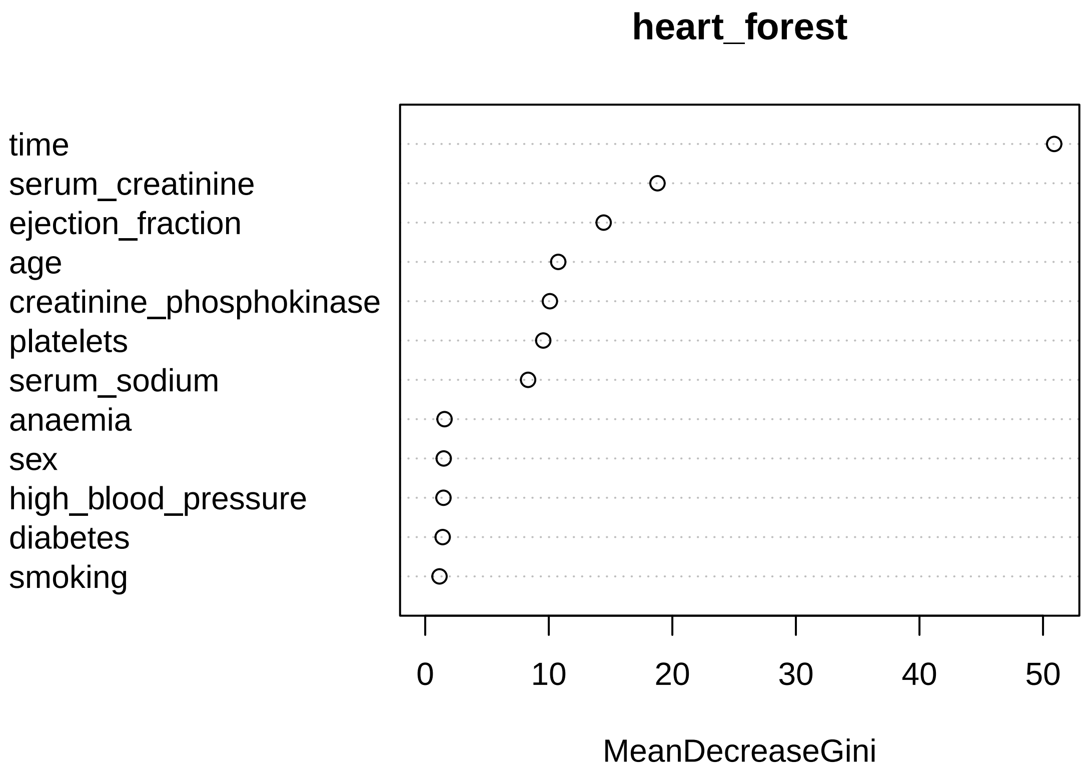


Analysis the data using the randomForest library.

```
#install.packages("randomForest")
library(randomForest)
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:rattle':
##
##     importance

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine
```

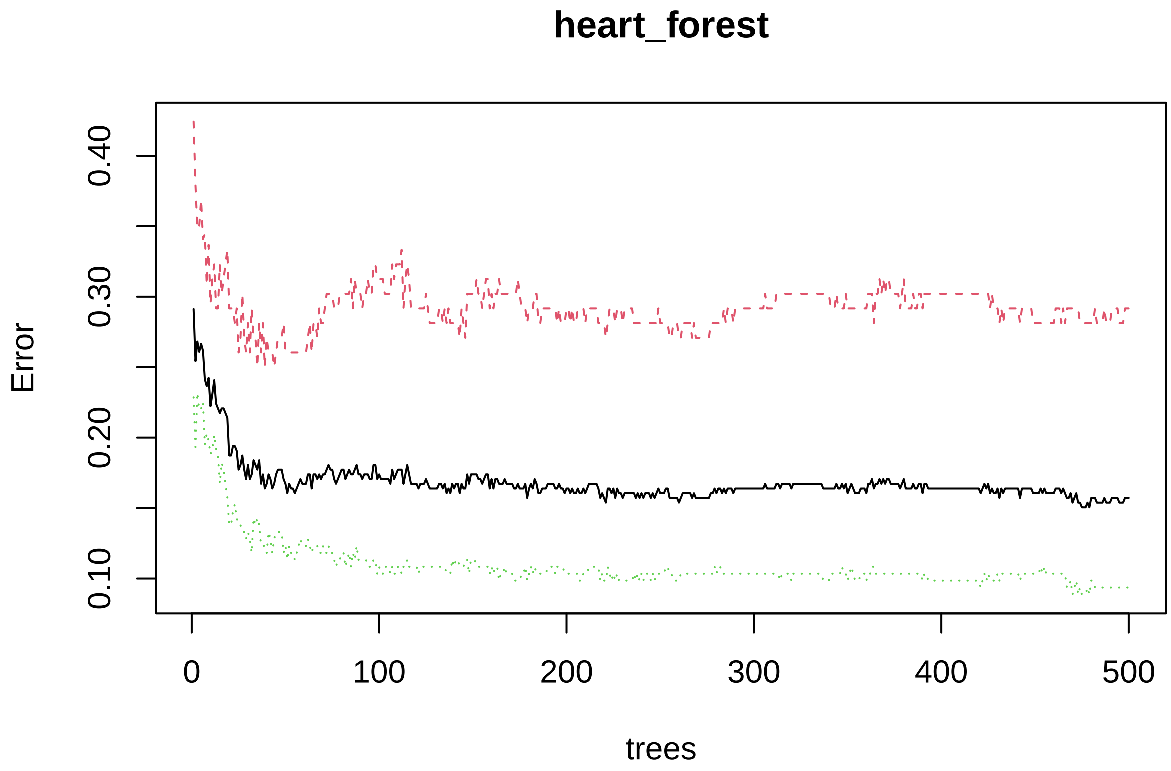
```r
heart_forest <- randomForest(DEATH_EVENT ~ ., data = data, ntree = 500, mtry = 4)
varImpPlot(heart_forest)
```



```r
heart_complete <- dplyr::mutate_if(data, is.character, as.factor)
conf.matrix <- table(heart_complete$DEATH_EVENT,predict(heart_forest, type = "class"))
rownames(conf.matrix) <- paste("Actual", rownames(conf.matrix), sep = ":")
colnames(conf.matrix) <- paste("Pred", colnames(conf.matrix), sep = ":")
conf.matrix
```

```
##
##                   Pred:Dead Pred:Survived
##   Actual:Dead            68            28
##   Actual:Survived        19           184
```

```r
plot(heart_forest)
```



Conclusion

In this project, I am try to predict best multiple logistic regression model. First, I can use glm()function and try to fit the logistic regression in the family = binomial in my data set. Some of the variable are non significant and remove those variable with the help of forward, backword, and stepwise selection method, and try to keep AIC and BIC values are small as well as null deviance and residual deviance keep small. Also, data visualize the dafferent possible way such as box-plot, bar-plot, ggplot, and decision tree. However, my overall goal is to the help of data visualization to predict best logistic regression model. So, my best fit logistic regression model is:

DEATH_EVENT = 12.498 + 0.045 * age - 0.068 * ejection_fraction - 0.082 * serum_sodium - 0.020 * time.

Reference

https://www.kaggle.com/andrewmvd/heart-failure-clinical-data?select=heart_failure_clinical_records_dataset.csv

THE END