final_project

QuocNguyen

2023-11-10

Introduction

Logistic regression analysis is a commonly used model when the response variable is a binary dependent variables. When there are more than two classes, we would prefer the multinomial logistic regression. Logistic regression not only can predict the possibility of one observation based on predictors, but it is also helpful in measuring the relationship between the dependent variable and other independent variables.

In this project, I will utilize the logistic regression model not only to identify the relationship between medical predictors and heart failure but also to predict the survival of a patient based on these healthcare variables.

Dataset

Dataset used in this project contains the health records of 299 heart failure patients at the Faisalabad Institute of Cardiology and at the Allied Hospital in Faisalabad (Punjab, Pakistan), during April–December 2015. The patients consisted of 105 women and 194 men, and their ages range between 40 and 95 years old . All 299 patients had left ventricular systolic dysfunction and had previous heart failures that put them in classes III or IV of New York Heart Association (NYHA) classification of the stages of heart failure. (Download data here: https://archive.ics.uci.edu/dataset/519/heart+failure+clinical+records).

Dataset contains 12 variables and one dependent variables. All the variables will be described below:

- **age**: age of the patient (Years)
- **anaemia**: decrease of red blood cells or hemoglobin. 0 as no anaemia; 1 otherwise
- **creatinine_phosphokinase**: level of the CPK enzyme in the blood (mcg/L)
- **diabetes**: if the patient has diabetes. 0 if patient has no diabetes; 1 otherwise
- **ejection_fraction**: percentage of blood leaving the heart at each contraction (Percentage)
- **high_blood_pressure**: if the patient has hypertension. 0 as no high blood; 1 otherwise
- **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. 0 as woman; 1 as man
- **smoking**: if the patient smokes or not. 0 as no-smoking; 1 otherwise
- **time**: follow-up period
- **DEATH_EVENT**: if the patient died during the follow-up period. 0 as survived; 1 as dead

EDA

Before looking for the best logistic regression model, we need to explore our dataset.

```r
library(readr)

df=read.csv("heart_failure_clinical_records_dataset.csv")

head(df,2)
```

```
##   age anaemia creatinine_phosphokinase diabetes ejection_fraction
## 1  75       0                      582        0                20
## 2  55       0                     7861        0                38
##   high_blood_pressure platelets serum_creatinine serum_sodium sex smoking time
## 1                   1    265000              1.9          130   1       0    4
## 2                   0    263358              1.1          136   1       0    6
##   DEATH_EVENT
## 1           1
## 2           1
```

```r
str(df)
```

```
## ’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 ...
```

```r
summary(df)
```

```
##       age           anaemia       creatinine_phosphokinase    diabetes
##  Min.   :40.00   Min.   :0.0000   Min.   :  23.0           Min.   :0.0000
##  1st Qu.:51.00   1st Qu.:0.0000   1st Qu.: 116.5           1st Qu.:0.0000
##  Median :60.00   Median :0.0000   Median : 250.0           Median :0.0000
##  Mean   :60.83   Mean   :0.4314   Mean   : 581.8           Mean   :0.4181
##  3rd Qu.:70.00   3rd Qu.:1.0000   3rd Qu.: 582.0           3rd Qu.:1.0000
##  Max.   :95.00   Max.   :1.0000   Max.   :7861.0           Max.   :1.0000
##  ejection_fraction high_blood_pressure   platelets      serum_creatinine
##  Min.   :14.00     Min.   :0.0000      Min.   : 25100   Min.   :0.500
##  1st Qu.:30.00     1st Qu.:0.0000      1st Qu.:212500   1st Qu.:0.900
##  Median :38.00     Median :0.0000      Median :262000   Median :1.100
##  Mean   :38.08     Mean   :0.3512      Mean   :263358   Mean   :1.394
##  3rd Qu.:45.00     3rd Qu.:1.0000      3rd Qu.:303500   3rd Qu.:1.400
##  Max.   :80.00     Max.   :1.0000      Max.   :850000   Max.   :9.400
##   serum_sodium        sex            smoking            time
##  Min.   :113.0   Min.   :0.0000   Min.   :0.0000   Min.   :  4.0
##  1st Qu.:134.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 73.0
##  Median :137.0   Median :1.0000   Median :0.0000   Median :115.0
##  Mean   :136.6   Mean   :0.6488   Mean   :0.3211   Mean   :130.3
##  3rd Qu.:140.0   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:203.0
##  Max.   :148.0   Max.   :1.0000   Max.   :1.0000   Max.   :285.0
##   DEATH_EVENT
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.3211
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

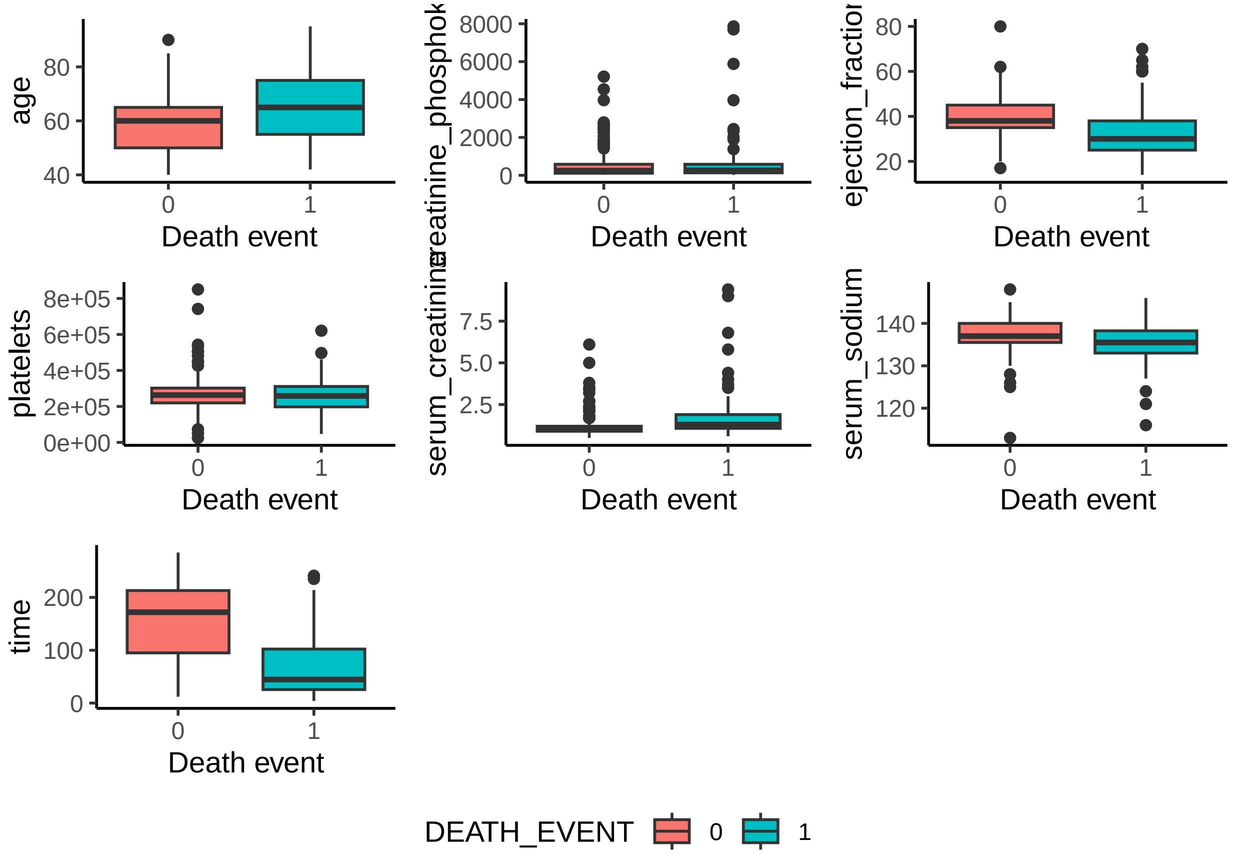
Dataset does not include any missing values.

```
print("Total missing values in data -")
```

```
## [1] "Total missing values in data -"
```

```
sum(is.na(df))
```

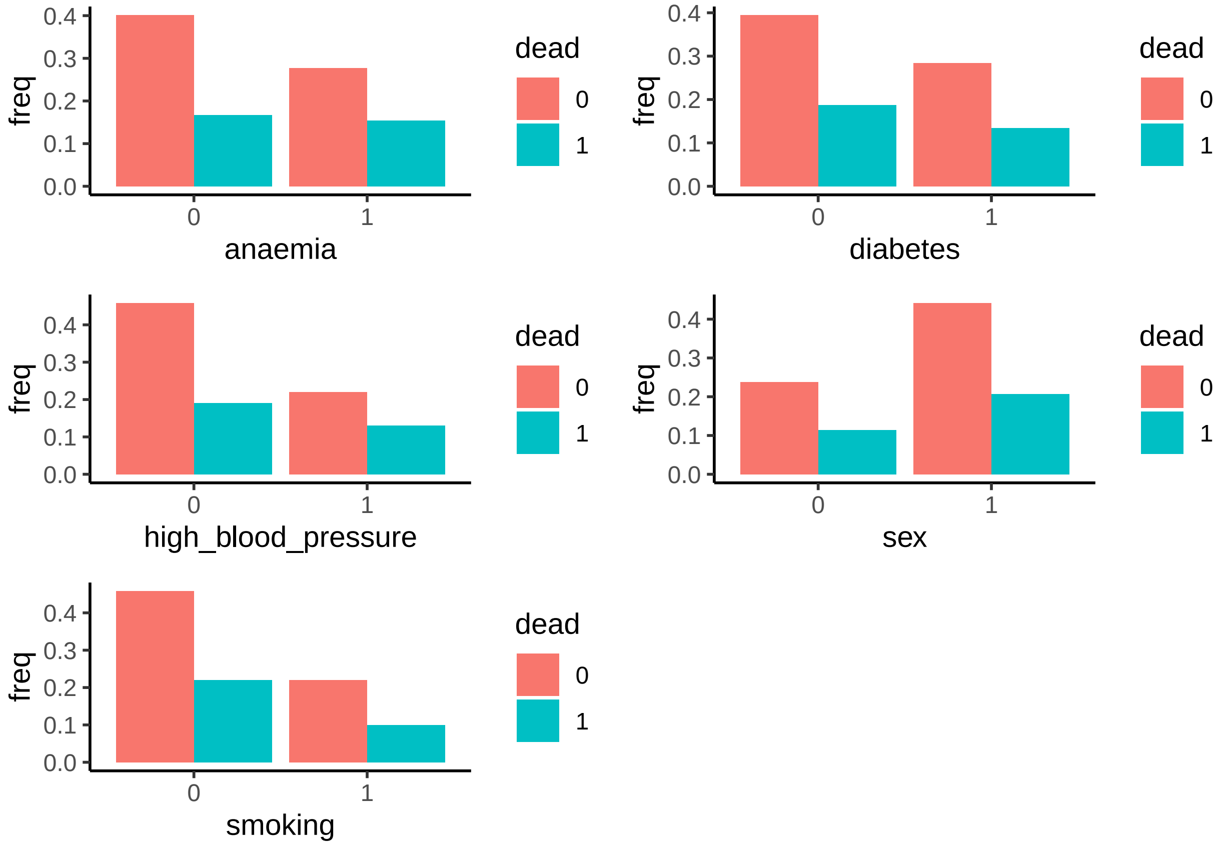
```
## [1] 0
```



We have some interesting points based on the boxplot of dead and survived patients:

- For platelets, level of serum creatinine, CPK enzyme and serum sodium in the blood, there are no significant differences between dead and survived patients.

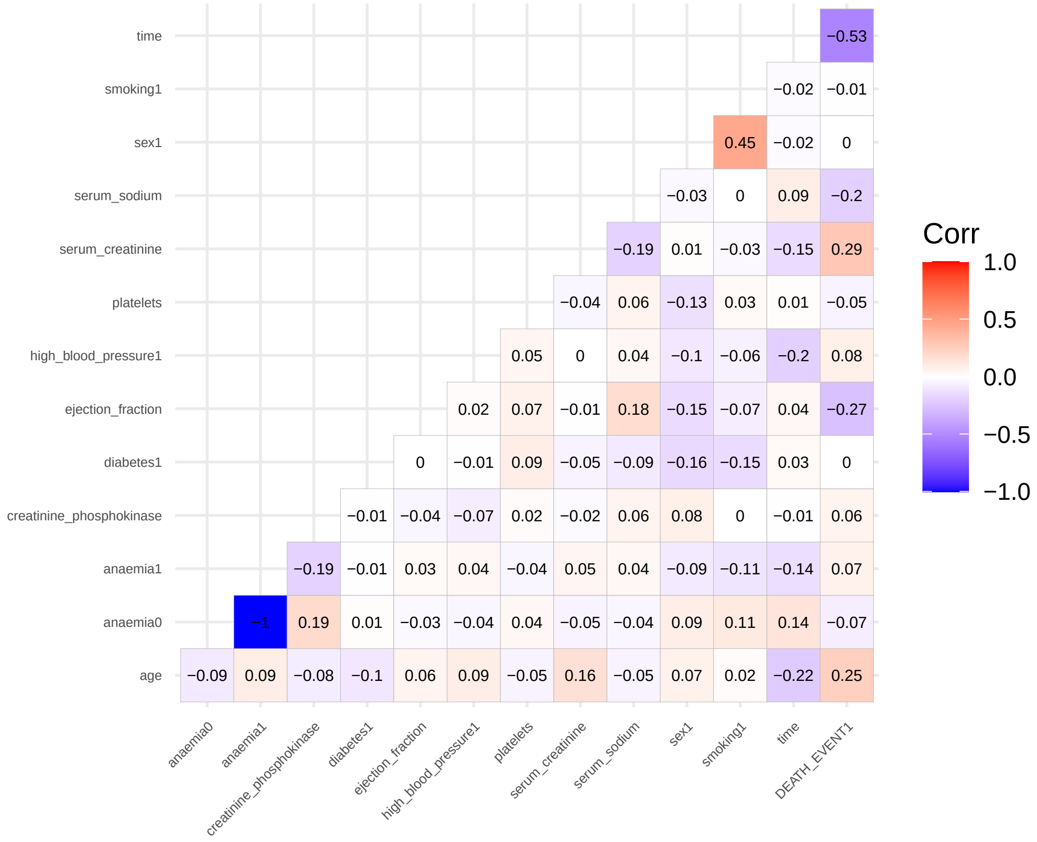
- Period followed-up of survived patients tends to much higher than deceased cases.
- Dead patients are older than the survived since the minimum age of dead patients are nearly equal the average age of the survived patients.
- Patients who survived would have the higher percentage of blood leaving the heart at each contraction. The average rate of the survived is 40.27%, and only 33.47% for the dead.



In general, the bar plots above hardly to show any important indicators of the dependent variables. We can figure out the importance of these categorical variables later by the logistic regression model.

Next, we might want if there is any cases of multicollinearity in this dataset since multicollinearity could affect on the results of the logistic regression models. To identify it, I will use the correlation matrix.

```
library(ggcorrplot)
model.matrix(~0+., data=df) %>%
  cor(use="pairwise.complete.obs") %>%
  ggcorrplot(show.diag=FALSE, type="lower", lab=TRUE, lab_size=2,tl.cex=5)
```



The correlation matrix shows that the multicollinearity is not the problem in our dataset since there are no strong relations between each variables. The period follow-up has a moderate negative correlation with the dependent variables, with the coefficient of -0.53. Other variables are almost uncorrelated with the target variable.

Methodology

Logistic regression model is the main model. Moreover, in this project, we would apply some criteria for the model selection. First, I will create the original model that include all predictors of dataset

```
original_model<-glm(DEATH_EVENT~.,family = binomial,data=df)
summary(original_model)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ ., family = binomial, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1848  -0.5706  -0.2401   0.4466   2.6668
##
## 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 **
## anaemia1                   -7.470e-03  3.605e-01  -0.021 0.983467
## creatinine_phosphokinase  2.222e-04  1.779e-04   1.249 0.211684
## diabetes1                   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_pressure1       -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 .
## sex1                       -5.337e-01  4.139e-01  -1.289 0.197299
## smoking1                   -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
```

```
original_model$deviance
```

```
## [1] 219.5541
```

In this original one, p-value shows that only age, ejection_fraction, serum_creatinine and time are statistically significant. Furthermore, no categorical variable has the statistical effect on the dependent variable. And the deviance of the original model is 219.5541 with degree of freedom of 298, which shows the original model is not good.

I will transform the platelets and creatinine_phosphokinase using the logarithm function.

```
#Add log_platelets, log_creatinine_phosphokinase
df$logplatelets<-log(df$platelets)
df$logcreatinine_phosphokinase<-log(df$creatinine_phosphokinase)
model1<-glm(DEATH_EVENT~.-creatinine_phosphokinase-platelets,family = binomial,data=df)
summary(model1)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ . - creatinine_phosphokinase - platelets,
##     family = binomial, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2569  -0.5376  -0.2308   0.4774   2.5678
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  15.25224    8.02512   1.901 0.057359 .
## age                           0.04879    0.01609   3.032 0.002429 **
## anaemia1                     0.01505    0.36440   0.041 0.967052    
## diabetes1                    0.12577    0.35372   0.356 0.722169    
## ejection_fraction           -0.07745    0.01663  -4.656 3.22e-06 ***
## high_blood_pressure1        -0.06467    0.36333  -0.178 0.858737    
## serum_creatinine             0.67542    0.18341   3.682 0.000231 ***
## serum_sodium                -0.06554    0.04034  -1.625 0.104195    
## sex1                        -0.59950    0.41797  -1.434 0.151483    
## smoking1                     0.01398    0.41577   0.034 0.973167    
## time                        -0.02202    0.00311  -7.079 1.45e-12 ***
## logplatelets                -0.58193    0.44364  -1.312 0.189619    
## logcreatinine_phosphokinase  0.31939    0.15956   2.002 0.045318 *  
## ---
## 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: 215.93  on 286  degrees of freedom
## AIC: 241.93
## 
## Number of Fisher Scoring iterations: 6
```

```
model1$deviance
```

```
## [1] 215.9345
```

By transforming these two variables, we can see that our new variable *logcreatinine_phosphokinase* has the p-value <0.05, which is statistically significant. And the deviance of this new model is also lower than the original.

AIC Criteria

AIC criteria is also one of the way to compare the models. Here, I will use the forward AIC and backward AIC. In Forward selection, we start with the logistic regression model includes nothing. And for each step, one variable will be added in the model until we get the lowest AIC. On the other hand for Backward selection, we start with the model included full predictors, and each variable will be respectively removed out of the model until we get the model with the lowest AIC.

First, we use the Backward AIC, we start with the *model1*.

```
#Backwards AIC


backwards = step(model1) # Backwards selection is the default
```

```
## Start:  AIC=241.93
## DEATH_EVENT ~ (age + anaemia + creatinine_phosphokinase + diabetes + 
##     ejection_fraction + high_blood_pressure + platelets + serum_creatinine + 
##     serum_sodium + sex + smoking + time + logplatelets + logcreatinine_phosphokinase) - 
##     creatinine_phosphokinase - platelets
## 
##                               Df Deviance    AIC
## - smoking                      1   215.94 239.94
## - anaemia                      1   215.94 239.94
## - high_blood_pressure          1   215.97 239.97
## - diabetes                     1   216.06 240.06
## - logplatelets                 1   217.65 241.64
## <none>                             215.93 241.93
## - sex                          1   218.03 242.03
## - serum_sodium                 1   218.59 242.59
## - logcreatinine_phosphokinase  1   220.09 244.09
## - age                          1   226.01 250.01
## - serum_creatinine             1   230.55 254.55
## - ejection_fraction            1   242.66 266.66
## - time                         1   295.72 319.72
##
## Step:  AIC=239.94
## DEATH_EVENT ~ age + anaemia + diabetes + ejection_fraction +
##     high_blood_pressure + serum_creatinine + serum_sodium + sex +
##     time + logplatelets + logcreatinine_phosphokinase
##
##                               Df Deviance    AIC
## - anaemia                      1   215.94 237.94
## - high_blood_pressure          1   215.97 237.97
## - diabetes                     1   216.06 238.06
## - logplatelets                 1   217.65 239.65
## <none>                             215.94 239.94
## - sex                          1   218.36 240.36
## - serum_sodium                 1   218.59 240.59
## - logcreatinine_phosphokinase  1   220.09 242.09
## - age                          1   226.02 248.02
## - serum_creatinine             1   230.71 252.71
## - ejection_fraction            1   242.67 264.67
## - time                         1   295.87 317.87
##
## Step:  AIC=237.94
## DEATH_EVENT ~ age + diabetes + ejection_fraction + high_blood_pressure +
##     serum_creatinine + serum_sodium + sex + time + logplatelets +
##     logcreatinine_phosphokinase
##
##                               Df Deviance    AIC
## - high_blood_pressure          1   215.97 235.97
## - diabetes                     1   216.06 236.06
## - logplatelets                 1   217.66 237.66
## <none>                             215.94 237.94
## - sex                          1   218.43 238.43
## - serum_sodium                 1   218.61 238.61
## - logcreatinine_phosphokinase  1   220.19 240.19
## - age                          1   226.04 246.04
## - serum_creatinine             1   230.71 250.71
## - ejection_fraction            1   242.69 262.69
## - time                         1   297.58 317.58
##
## Step:  AIC=235.97
## DEATH_EVENT ~ age + diabetes + ejection_fraction + serum_creatinine +
##     serum_sodium + sex + time + logplatelets + logcreatinine_phosphokinase
```

```
##
##                               Df Deviance    AIC
## - diabetes                     1   216.10 234.10
## - logplatelets                 1   217.74 235.74
## <none>                             215.97 235.97
## - sex                          1   218.44 236.44
## - serum_sodium                 1   218.63 236.63
## - logcreatinine_phosphokinase  1   220.27 238.27
## - age                          1   226.05 244.05
## - serum_creatinine             1   231.10 249.10
## - ejection_fraction            1   242.70 260.70
## - time                         1   299.69 317.69
##
## Step:  AIC=234.1
## DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + serum_sodium +
##     sex + time + logplatelets + logcreatinine_phosphokinase
##
##                               Df Deviance    AIC
## - logplatelets                 1   217.83 233.83
## <none>                             216.10 234.10
## - sex                          1   218.71 234.71
## - serum_sodium                 1   218.93 234.93
## - logcreatinine_phosphokinase  1   220.46 236.46
## - age                          1   226.05 242.05
## - serum_creatinine             1   231.15 247.15
## - ejection_fraction            1   242.78 258.77
## - time                         1   299.86 315.86
##
## Step:  AIC=233.83
## DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + serum_sodium +
##     sex + time + logcreatinine_phosphokinase
##
##                               Df Deviance    AIC
## <none>                             217.83 233.83
## - sex                          1   220.12 234.12
## - serum_sodium                 1   220.86 234.86
## - logcreatinine_phosphokinase  1   222.04 236.04
## - age                          1   227.96 241.96
## - serum_creatinine             1   233.46 247.46
## - ejection_fraction            1   244.89 258.89
## - time                         1   300.22 314.22
```

```
summary(backwards)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##     serum_sodium + sex + time + logcreatinine_phosphokinase,
##     family = binomial, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2702  -0.5452  -0.2163   0.4842   2.6463
##
```

```
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  8.37772    5.63388   1.487 0.137008
## age                          0.04781    0.01569   3.047 0.002312 **
## ejection_fraction           -0.07688    0.01639  -4.689 2.74e-06 ***
## serum_creatinine             0.68795    0.18100   3.801 0.000144 ***
## serum_sodium                -0.06798    0.03919  -1.734 0.082843 .
## sex1                        -0.55601    0.36932  -1.506 0.132193
## time                        -0.02157    0.00299  -7.216 5.36e-13 ***
## logcreatinine_phosphokinase  0.31601    0.15717   2.011 0.044360 *
## ---
## 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: 217.83  on 291  degrees of freedom
## AIC: 233.83
##
## Number of Fisher Scoring iterations: 6
```

After dropping steps, the final model includes age, ejection_fraction, serum_creatinine, serum_sodium, sex, time and logcreatinine_phosphokinase.

Next, we use the Forward AIC, which starts at the model with nothing to the *model1*.

```
#Toward AIC
#Model with nothing
nothing<-glm(DEATH_EVENT~ 1,family=binomial,data=df)
#Forward AIC
forwards=step(nothing,direction="forward",scope=list(upper=model1,lower=nothing))
```

```
## Start:  AIC=377.35
## DEATH_EVENT ~ 1
##
##                               Df Deviance    AIC
## + time                         1   279.07 283.07
## + serum_creatinine             1   347.25 351.25
## + ejection_fraction            1   351.97 355.97
## + age                          1   355.99 359.99
## + serum_sodium                 1   364.02 368.02
## <none>                             375.35 377.35
## + high_blood_pressure          1   373.49 377.49
## + logplatelets                 1   373.98 377.98
## + anaemia                      1   374.04 378.04
## + logcreatinine_phosphokinase  1   375.04 379.04
## + smoking                      1   375.30 379.30
## + sex                          1   375.34 379.34
## + diabetes                     1   375.35 379.35
##
## Step:  AIC=283.07
## DEATH_EVENT ~ time
##
##                               Df Deviance    AIC
## + ejection_fraction            1   256.08 262.08
## + serum_creatinine             1   259.64 265.64
## + serum_sodium                 1   269.83 275.83
## + age                          1   271.46 277.46
## + logcreatinine_phosphokinase  1   275.90 281.90
## + logplatelets                 1   275.93 281.93
## <none>                             279.07 283.07
## + smoking                      1   278.81 284.81
## + high_blood_pressure          1   278.96 284.96
## + sex                          1   279.02 285.02
## + diabetes                     1   279.06 285.06
## + anaemia                      1   279.07 285.07
##
## Step:  AIC=262.08
## DEATH_EVENT ~ time + ejection_fraction
##
##                               Df Deviance    AIC
## + serum_creatinine             1   235.41 243.41
## + age                          1   244.51 252.51
## + serum_sodium                 1   249.73 257.73
## + logplatelets                 1   253.62 261.62
## + logcreatinine_phosphokinase  1   254.04 262.04
## <none>                             256.08 262.08
## + sex                          1   254.98 262.98
## + smoking                      1   255.20 263.20
## + high_blood_pressure          1   255.93 263.93
## + diabetes                     1   256.05 264.05
## + anaemia                      1   256.08 264.08
##
## Step:  AIC=243.41
## DEATH_EVENT ~ time + ejection_fraction + serum_creatinine
##
##                               Df Deviance    AIC
## + age                          1   226.30 236.30
## + serum_sodium                 1   232.02 242.02
## + logcreatinine_phosphokinase  1   232.77 242.77
## <none>                             235.41 243.41
## + logplatelets                 1   233.62 243.62
## + sex                          1   234.69 244.69
## + smoking                      1   235.20 245.20
## + diabetes                     1   235.33 245.33
## + high_blood_pressure          1   235.41 245.41
## + anaemia                      1   235.41 245.41
##
## Step:  AIC=236.3
## DEATH_EVENT ~ time + ejection_fraction + serum_creatinine + age
##
##                               Df Deviance    AIC
## + logcreatinine_phosphokinase  1   222.85 234.85
## + serum_sodium                 1   223.49 235.49
## <none>                             226.30 236.30
## + logplatelets                 1   224.67 236.67
## + sex                          1   225.08 237.08
## + diabetes                     1   225.87 237.87
## + smoking                      1   225.95 237.95
## + high_blood_pressure          1   226.27 238.27
## + anaemia                      1   226.28 238.28
##
## Step:  AIC=234.86
## DEATH_EVENT ~ time + ejection_fraction + serum_creatinine + age +
##     logcreatinine_phosphokinase
##
##                       Df Deviance    AIC
## + serum_sodium         1   220.12 234.12
## <none>                     222.85 234.85
## + sex                  1   220.86 234.86
## + logplatelets         1   221.24 235.24
## + diabetes             1   222.41 236.41
## + smoking              1   222.44 236.44
## + anaemia              1   222.82 236.82
## + high_blood_pressure  1   222.85 236.85
##
## Step:  AIC=234.12
## DEATH_EVENT ~ time + ejection_fraction + serum_creatinine + age +
##     logcreatinine_phosphokinase + serum_sodium
##
##                       Df Deviance    AIC
## + sex                  1   217.83 233.83
## <none>                     220.12 234.12
## + logplatelets         1   218.71 234.71
## + smoking              1   219.64 235.64
## + diabetes             1   219.91 235.91
## + anaemia              1   220.02 236.02
## + high_blood_pressure  1   220.12 236.12
##
## Step:  AIC=233.83
## DEATH_EVENT ~ time + ejection_fraction + serum_creatinine + age +
##     logcreatinine_phosphokinase + serum_sodium + sex
##
##                       Df Deviance    AIC
## <none>                     217.83 233.83
## + logplatelets         1   216.10 234.10
## + diabetes             1   217.74 235.74
## + high_blood_pressure  1   217.75 235.75
## + anaemia              1   217.82 235.82
## + smoking              1   217.82 235.82
```

```
summary(forwards)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ time + ejection_fraction + serum_creatinine +
##     age + logcreatinine_phosphokinase + serum_sodium + sex, family = binomial,
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2702  -0.5452  -0.2163   0.4842   2.6463
```

```
## 
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)    
## (Intercept)                   8.37772    5.63388   1.487 0.137008    
## time                         -0.02157    0.00299  -7.216 5.36e-13 ***
## ejection_fraction            -0.07688    0.01639  -4.689 2.74e-06 ***
## serum_creatinine              0.68795    0.18100   3.801 0.000144 ***
## age                           0.04781    0.01569   3.047 0.002312 ** 
## logcreatinine_phosphokinase   0.31601    0.15717   2.011 0.044360 *  
## serum_sodium                 -0.06798    0.03919  -1.734 0.082843 .  
## sex1                         -0.55601    0.36932  -1.506 0.132193    
## ---
## 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: 217.83  on 291  degrees of freedom
## AIC: 233.83
## 
## Number of Fisher Scoring iterations: 6
```

```
(forwards$deviance)
```

```
## [1] 217.8297
```

By adding variables respectively, we have the final model using Forwards AIC. And we can see that both backwards and forwards give us the model with the same predictors. From this final model, we can develop by add the intersection between sex and ejection fraction and the intersection between sex and logcreatinine_phosphokinase

```
#Add the intersection between gender and other variables
model3<-glm(DEATH_EVENT~age + ejection_fraction + serum_creatinine +
              serum_sodium + sex + time + logcreatinine_phosphokinase+
              sex*ejection_fraction+
 sex*logcreatinine_phosphokinase,
            family = binomial, data = df)
summary(model3)
```

```
## 
## Call:
## glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + 
##     serum_sodium + sex + time + logcreatinine_phosphokinase + 
##     sex * ejection_fraction + sex * logcreatinine_phosphokinase, 
##     family = binomial, data = df)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.1839  -0.5354  -0.1796   0.3803   2.6382  
## 
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        7.401046   6.001262   1.233 0.217483
## age                               0.051536   0.016283   3.165 0.001550 **
## ejection_fraction                -0.048110   0.021370  -2.251 0.024371 *
## serum_creatinine                  0.644550   0.188665   3.416 0.000635 ***
## serum_sodium                     -0.093534   0.042616  -2.195 0.028177 *
## sex1                              6.927031   2.417865   2.865 0.004171 **
## time                             -0.024343   0.003372  -7.220  5.2e-13 ***
## logcreatinine_phosphokinase       0.947896   0.304189   3.116 0.001832 **
## ejection_fraction:sex1           -0.070378   0.032867  -2.141 0.032250 *
## sex1:logcreatinine_phosphokinase -0.886444   0.357707  -2.478 0.013207 *
## ---
## 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: 207.18  on 289  degrees of freedom
## AIC: 227.18
##
## Number of Fisher Scoring iterations: 6
```

```
(model3$deviance)
```

```
## [1] 207.1825
```

```
model3$coefficients
```

```
##                      (Intercept)                              age
##                       7.40104589                       0.05153625
##                ejection_fraction                 serum_creatinine
##                      -0.04810975                       0.64455038
##                     serum_sodium                             sex1
##                      -0.09353394                       6.92703093
##                             time      logcreatinine_phosphokinase
##                      -0.02434349                       0.94789551
##           ejection_fraction:sex1 sex1:logcreatinine_phosphokinase
##                      -0.07037788                      -0.88644446
```

The model with the intersections has the deviance is smaller than the backwards model.

```
#Chisquare test between two models
anova(backwards,model3,test='Chisq')
```

```
## Analysis of Deviance Table
##
## Model 1: DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + serum_sodium +
##     sex + time + logcreatinine_phosphokinase
## Model 2: DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + serum_sodium +
##     sex + time + logcreatinine_phosphokinase + sex * ejection_fraction +
##     sex * logcreatinine_phosphokinase
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       291     217.83
## 2       289     207.18  2   10.647 0.004875 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Instead of comparing the deviance, we use Chi-square test to compare two models. The p-value is statistically significant, which implies that the significantly effect of the intersection.

Model interpretation

The final model is expressed as below:

$$\log(\frac{\pi_i}{1-\pi_i}) = 7.40{+}0.051.Age{-}0.048.EjectionFraction{+}0.644.SerumCreatinine{-}0.093.Sodium{+}6.927.Sex+$$

$$-0.024.Time + 0.947.Phosphokinase + 0.07.(Ejection \times sex) - 0.886.(Phosphokinase \times sex)$$

Coefficients

Interpretations

We can interpret model as follows:

- Holding other variables as constant, the odds of dead by heart failure are predicted to grow about 1.05 times for an increase of one year-old in age of the patient.
- Holding other variables as constant, the odds of dead by heart failure are predicted to decrease about 0.953 times for one ejection rate increase.

In our problem, we have Gender variable as the binary variable, where 0 denoted as Female and 1 as False

When the patient is Male, we have:

$$\log(\frac{\pi_i}{1-\pi_i}) = 7.40{+}0.051.Age{+}({-}0.07{-}0.048).EjectionFraction{+}0.644.SerumCreatinine{-}0.093.Sodium+$$

$$+6.927 - 0.024.Time + (0.947 - 0.886).Phosphokinase$$

and when a patient is Female, we have:

$$\log(\frac{\pi_i}{1-\pi_i}) = 7.40 + 0.051.Age - 0.048.EjectionFraction + 0.644.SerumCreatinine - 0.093.Sodium+$$

$$-0.024.Time + 0.947.Phosphokinase$$

The difference of log odds ratio between Female and Male patients is the sum of $B_{sex}{+}B_{(Ejection\times sex)}.Eject+$ $B_{(Phosphokinase\times sex)}.Phosphokinase.$

When a patient is Male, one unit increase in ejection fraction would decrease the odds of dead 0.8887 times. And one-unit increase in logPhosphokinase, we expect the odds of dead to grow by 6%.

Confidence interval of coefficients

```
confint(model3)
```

```
##                                          2.5 %       97.5 %
## (Intercept)                         -4.32922524 19.480284755
## age                                  0.02071008  0.084841073
## ejection_fraction                   -0.09241062 -0.007920487
## serum_creatinine                     0.29110205  1.059140178
## serum_sodium                        -0.17995299 -0.010783081
## sex1                                 2.28634792 11.816744810
## time                                -0.03147119 -0.018186920
## logcreatinine_phosphokinase          0.36261294  1.563590179
## ejection_fraction:sex1              -0.13627523 -0.006603186
## sex1:logcreatinine_phosphokinase    -1.60807831 -0.197645320
```

We can find the 95% confidence interval of each coefficient by the givien Standard error. For example, we are 95% confident that the *Age* coefficient lies between 0.02 and 0.08.

p-values

We see that p-values of all coefficients are smaller than 0.05, which indicates that age of the patient, percentage of blood leaving the heart at each contraction, level of serum creatinine and sodium in the blood, the time followed-up, the gender of the patient, log of level of the CPK enzyme in the blood and the intersections have significant effects on the survival rate. Later, we will would rank the these features to indicate which is the most important factor that affect on the response variable.

Deviance

The smaller the deviance of the model, the better the model is. In our model, we have the deviance is the smallest deviance compared to other models that we tested, which implies we possibly got the good logistic regression model.

```
#Deviance
model3$deviance
```

```
## [1] 207.1825
```

```
#Degree of freedom
model3$df.residual
```

```
## [1] 289
```

Goodness of Fit Test

To confirm our selection, we need to check its GOF to know if this model does fit well on our dataset. The Hosmer-Lemeshow test and residual deviance test will be used to check the GOF.

```
library(ResourceSelection)
#Hosmer-Lemeshow
hoslem.test(df$DEATH_EVENT, fitted(model3))
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  df$DEATH_EVENT, fitted(model3)
## X-squared = 299, df = 8, p-value < 2.2e-16
```

```
#Test residual deviance
1-pchisq(model3$deviance,model3$df.residual)
```

```
## [1] 0.9999158
```

Since p-value of both tests are larger than 0.5, we conclude that lack-of-fit does not appear in our model.

Prediction accuracy

To avoid the bias in prediction, dataset will be splitted into training (70%) and testing (30%) dataset.

```
#make this example reproducible
set.seed(100)
df$DEATH_EVENT<-as.factor(df$DEATH_EVENT)
#use 70% of dataset as training set and 30% as test set
sample <- sample(c(TRUE, FALSE), nrow(df), replace=TRUE, prob=c(0.7,0.3))
train  <- df[sample, ]
test   <- df[!sample, ]
```

Now, we build the model using the training dataset. After that, we will test the accuracy of our model using the testing dataset. This will avoid the bias in predicted results.

```
library(pROC)
# Train on training dataset
model3<-glm(DEATH_EVENT~age + ejection_fraction + serum_creatinine +
              serum_sodium + sex + time + logcreatinine_phosphokinase+
              sex*ejection_fraction+sex*logcreatinine_phosphokinase,
            family = binomial,
            data = train)

# Run the model on test dataset
test_prob_logistic<-predict(model3,test,type ="response")
# Choose threshold=0.5
test_pred_logistic<-ifelse(test_prob_logistic > 0.5,"1","0")
#Compare
cat("Confusion matrix of logistic regression on test data")
```

```
## Confusion matrix of logistic regression on test data
```

```
table(test_pred_logistic,test$DEATH_EVENT)
```

```
##
## test_pred_logistic  0  1
##                  0 57  3
##                  1 15 16
```

```r
#Accuracy on testing dataset

accuracy_logistic<-round(mean(test_pred_logistic==test$DEATH_EVENT),4)

cat("Accuracy of Logistic regression model on test data is:",accuracy_logistic)
```

```
## Accuracy of Logistic regression model on test data is: 0.8022
```

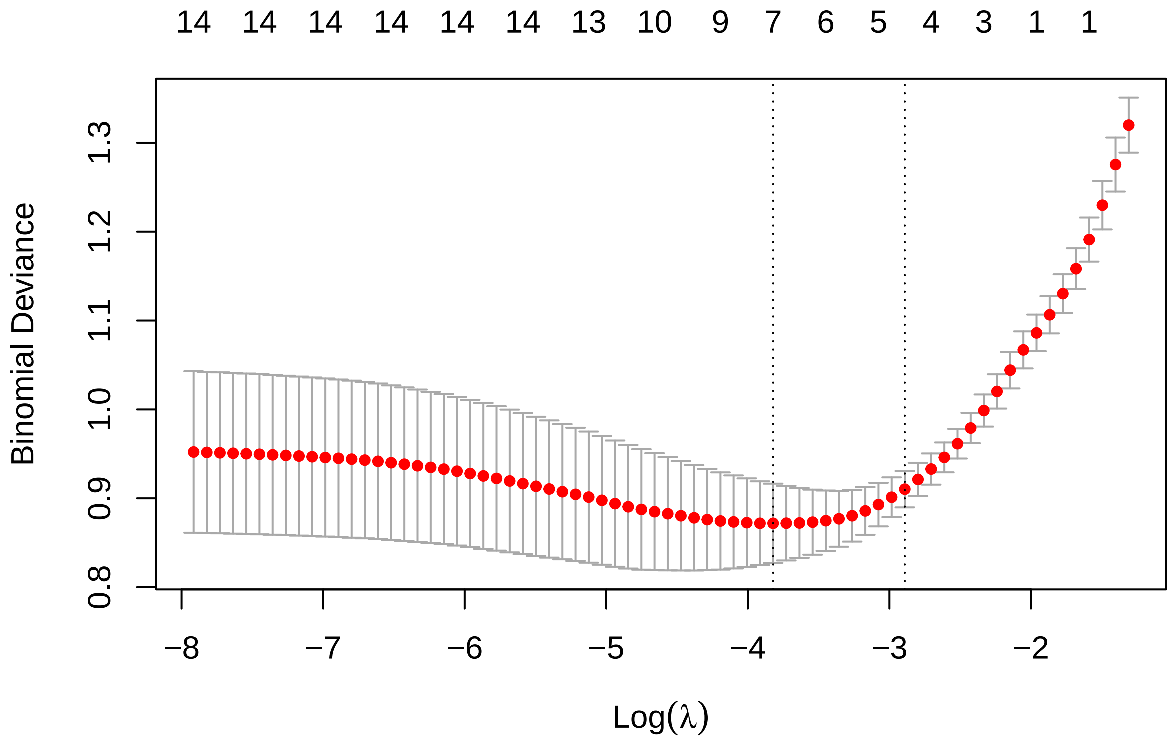
```r
# Run the model on train dataset

train_prob<-predict(model3,type ="response")

train_pred<-ifelse(train_prob > 0.5,"1","0")

#Accuracy on training dataset
cat("Accuracy of Logistic regression model on training data",mean(train_pred==train$DEATH_EVENT))
```

```
## Accuracy of Logistic regression model on training data 0.8605769
```

The accuracy of the logistic regression model on testing dataset is 80.2%, which is pretty well. We also get the accuracy of 86.05% of the model on the training dataset.

However, this shows that we migh be facing the problem of overfitting, when the accuracy on the training dataset is larger than the accuracy on the testing dataset nearly 6%. To solve this problem, we can use Lasso technique.

```r
library(glmnet)
#Data prepare

levels(train$DEATH_EVENT) <- c("survived", "died")
levels(test$DEATH_EVENT) <- c("survived", "died")

#Train matrix data
x<-model.matrix(DEATH_EVENT~.,train)[,-1]
y<-train$DEATH_EVENT

#Test matrix data
x_test<-model.matrix(DEATH_EVENT~.,test)[,-1]
y_test<-test$DEATH_EVENT

#Lasso
cv.lasso<-cv.glmnet(x,y,alpha=1,family='binomial')

plot(cv.lasso)
```



```
# Fit the final model on the training data
lasso_model <- glmnet(x, y, alpha = 1, family = "binomial",
                lambda = cv.lasso$lambda.min)






test_prob_lasso=predict(lasso_model,x_test ,type ="response")

test_classes_lasso<-ifelse(test_prob_lasso>0.5,"died","survived")

accuracy_lasso<-round(mean(test_classes_lasso==y_test),4)
cat("Accuracy of Lasso model on test data is:",accuracy_lasso)
```

```
## Accuracy of Lasso model on test data is: 0.8462
```

```
#roc_lasso_model <- roc(y_test, test_prob)
```

Evaluate Logistic regression and Lasso regression model

To evaluate the predictive model, we not only use the accuracy rate, but also other measurements, such as the sensitivity or specificity. Sensitivity measurement shows the ability of the model to predict correctly the survival cases; and the specificity tells us the ability of the model to predict correctly the dead cases. Usually, when the sensitivity increases, the specificity decreases and otherwise.

Moreover, the area under the ROC curve is called AUC. AUC is also a important measurement to evaluate one model. The higher the AUC, the better the model distinguish between survived class and dead class.

```
## AUC of Logistic Regression model is: 0.913

## Sensitivity of Logistic Regression model is: 0.7917

## Specificity of Logistic Regression model is: 0.8421

## AUC of Lasso Regression model is: 0.8918

## Sensitivity of Lasso Regression model is: 0.8611

## Specificity of Lasso Regression model is: 0.7895
```

While the our final logistic regression model has the lower accuracy but the percentage the logistic regression model can distinguish between the dead cases and the survival cases up to 91%. On the other hand, Lasso regression model has higher accuracy on test data but it's AUC is lower than Logistic model's AUC.

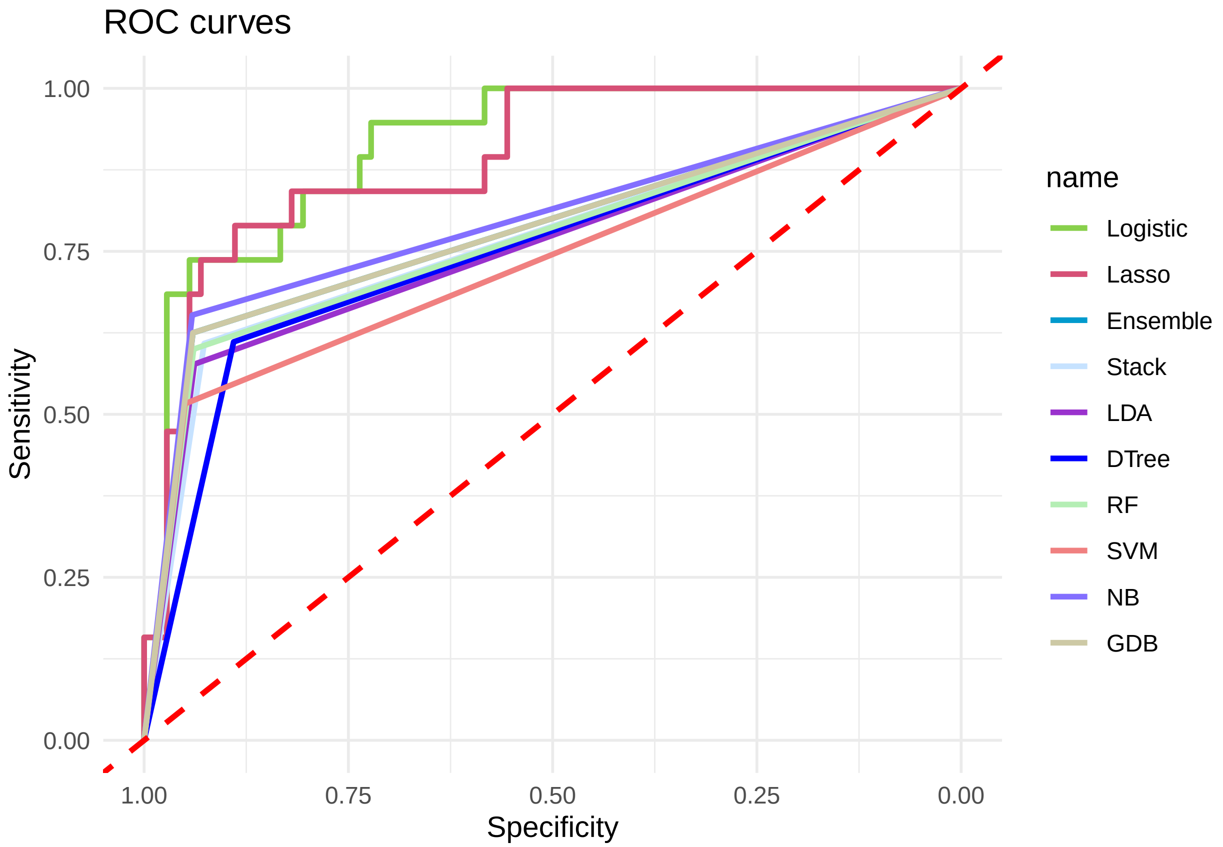
At the threshold =0.5, the Lasso model may have the higher accuracy rate than our final model. However, we can adjust this threshold to another value, the Logistic regression model can distinguish two classes better than the Lasso model.

Furthermore, at threshold =0.5, Lasso regression model predicts correctly the survival patient better than the Logistic regression, and otherwise.

Compare with other machine learning models

In this section, I will compare the two our logistic regression with other classification models (Naive Bayes, Random Forest, Gradient Boosting, Decision Tree, SVM and also Ensemble model and Stack mode of these machine learning models).

Model	Accuracy	Sensitivity	Specificity	AUC
Logistic regression	0.8022	0.7917	0.8421	0.9130000
Lasso regression	0.8462	0.8611	0.7895	0.8918000
Ensemble model	0.8571	0.8750	0.7895	0.7826493
Stack model	0.8462	0.8750	0.7368	0.7675831
LDA	0.8352	0.8472	0.7895	0.7576923
Decision Tree	0.8352	0.9028	0.5789	0.7507610
Random Forest	0.8462	0.8611	0.7895	0.7696970
SVM	0.8022	0.7917	0.8421	0.7330645
Naive Bayes	0.8681	0.8889	0.7895	0.7966752
Gradient Boosting	0.8571	0.8750	0.7895	0.7826493



Look at the table above, we can see that compared to others models, the accuracy and sensitivity of Logistic regression is not really good. However, with the highest specificity means that our final logistic regression model works really well on predicting the dead cases. And Logistic regression model has the highest AUC, while other machine learning models have much lower AUC rate.

Conclusion

After different steps of selection, we eventually have our final Logistic regression. We keep *Age*, *Ejection_fraction*, *serum_creatinine*, *serum_sodium*, *sex*, *time* and add into model new variables *logcreatinine_phosphokinase*. Moreover, we also include the intersection between gender and two other variables (eject_fraction and logcreatinine_phosphokinase). P-values show that every factors in the final model have significant effect on the dependent variable. By knowing these significant factors, patients who suffered from heart failure can be treated in time when there are some unusual changes in these heal factors.

Moreover, we see the ability to distinguish two classes is much better than other models. And it also works very well on predicting correctly the dead cases.