Heart Failure Prediction

Hrithik Agarwal

04/09/2020

About the Dataset

Cardiovascular diseases (CVDs) are the number 1 cause of death globally, taking an estimated 17.9 million lives each year, which accounts for 31% of all deaths worlwide. Heart failure is a common event caused by CVDs and this dataset contains 12 features that can be used to predict mortality by heart failure.

People with cardiovascular disease or who are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension, diabetes, hyperlipidaemia or already established disease) need early detection and management wherein a machine learning model can be of great help.

Citation

Davide Chicco, Giuseppe Jurman: Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Medical Informatics and Decision Making 20, 16 (2020)

Loading the Libraries

```
library(caTools)
library(randomForest)
library(caret)
library(ggplot2)
```

Import the dataset

```
heart <- read.csv('heart.csv',header =T)
```

Observing the dataset

The dataset contains 299 observations and 13 variables.

```
dim(heart)
```

```
## [1] 299  13
```

```
summary(heart)
```

```
##       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
```

It shows the minimum, maximum, mean, median and number of missing values(if any). From this it can be infered that there are no missingvalues in the dataset.

```
str(heart)
```

```
## '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 ...
```

Now we can see from *summary* and *str* function that the following variables are binary (i.e 0,1).

1. anaemia
2. diabetes
3. high_blood_pressure
4. sex
5. smoking
6. DEATH_EVENT

Converting Binary Variables into factors

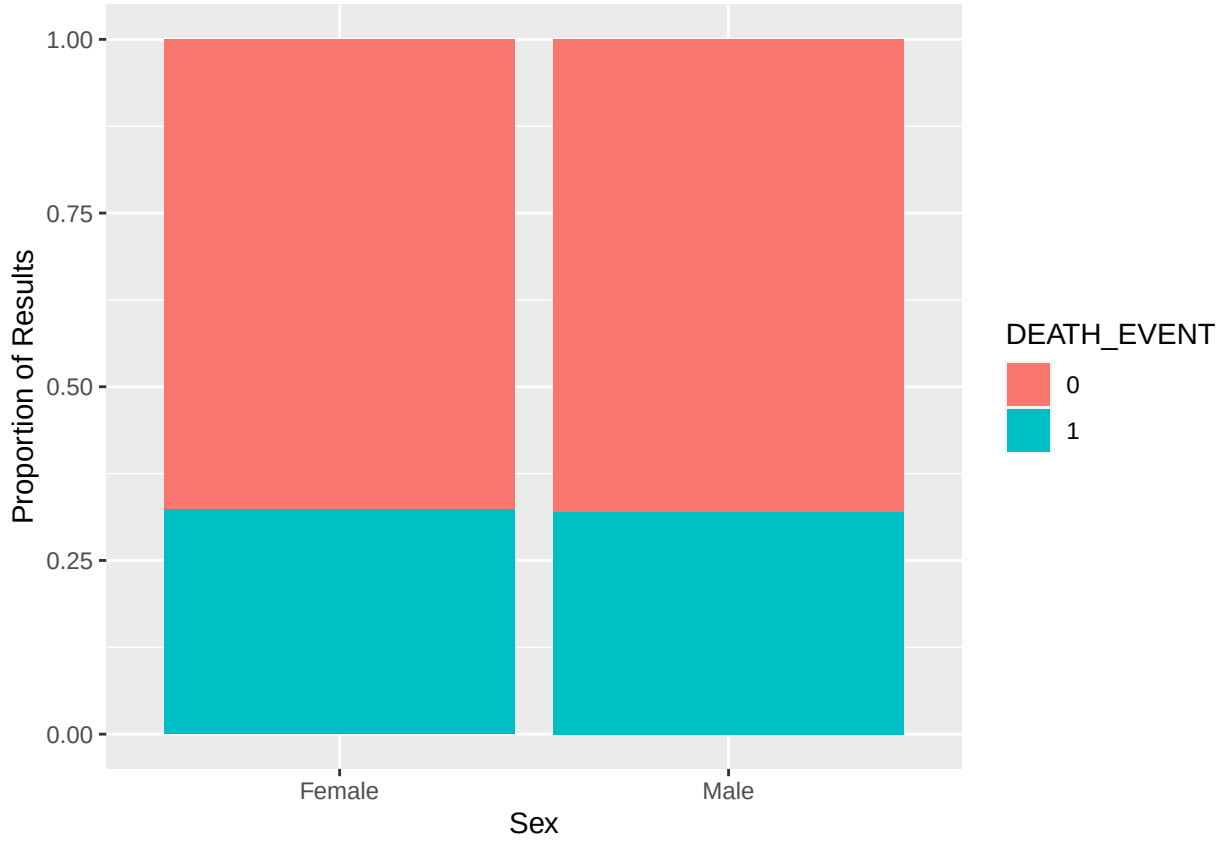
```
heart$anaemia <- as.factor(heart$anaemia)
heart$diabetes <- as.factor(heart$diabetes)
heart$high_blood_pressure <- as.factor(heart$high_blood_pressure)
heart$sex <- as.factor(heart$sex)
heart$smoking <- as.factor(heart$smoking)
heart$DEATH_EVENT <- as.factor(heart$DEATH_EVENT)
```

Now our Data has been processed for visulaization as well as model-making.

Data Visualiztion

Sex vs Death

```
ggplot(heart,aes(sex,fill = DEATH_EVENT))+
  geom_bar(position = "fill")+
  labs(y = "Proportion of Results",x = "Sex")+
  scale_x_discrete(labels = c("Female","Male"))
```

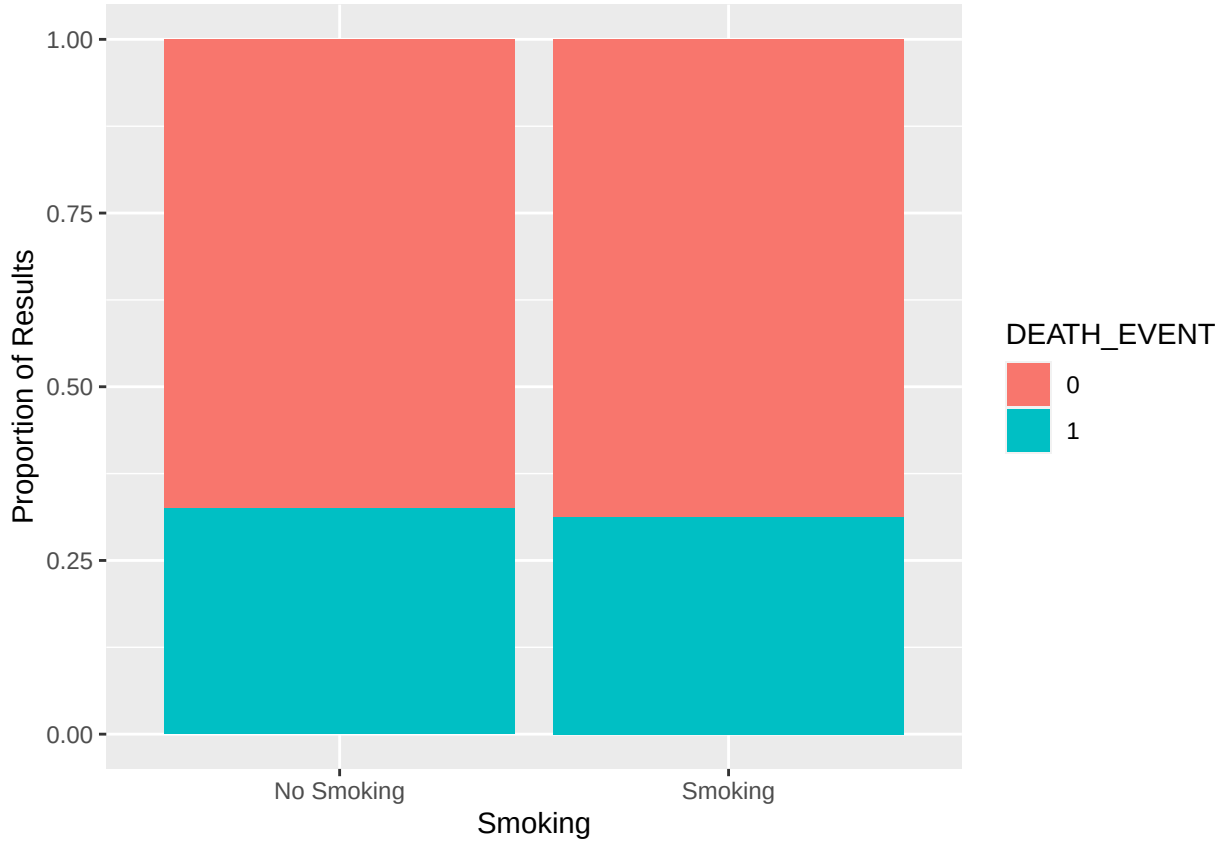


This shows that there is not any major difference between male and female who die due to heart failure.

Smoking

As we have read in litrature that smoking increases the risk of heart problems. However int the dataset the events have already occured and smoking does not seem to have any major effect.

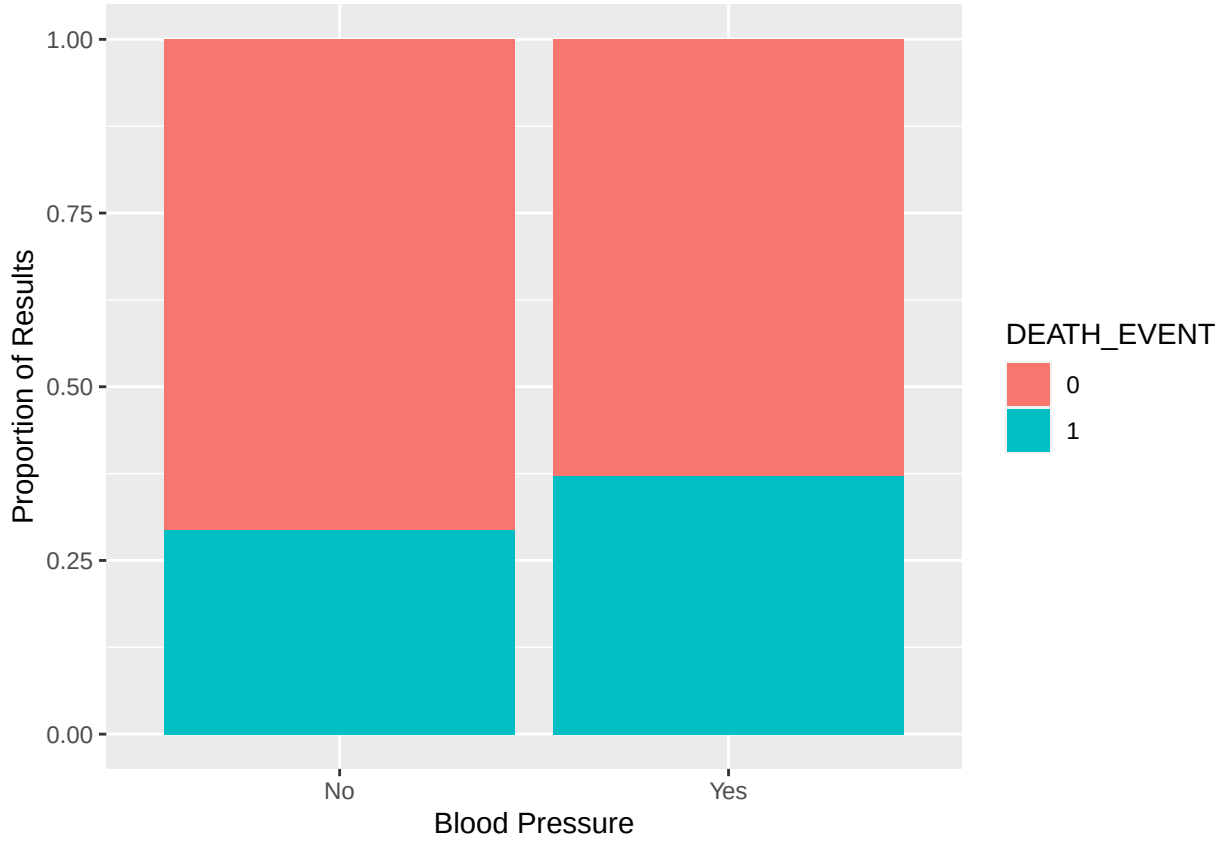
```
ggplot(heart,aes(smoking,fill = DEATH_EVENT))+
  geom_bar(position = "fill")+
  labs(y = "Proportion of Results",x = "Smoking")+
  scale_x_discrete(labels = c("No Smoking","Smoking"))
```



High Blood Pressure

It is known that high blood pressure leads to heart problems and is evident from the plot

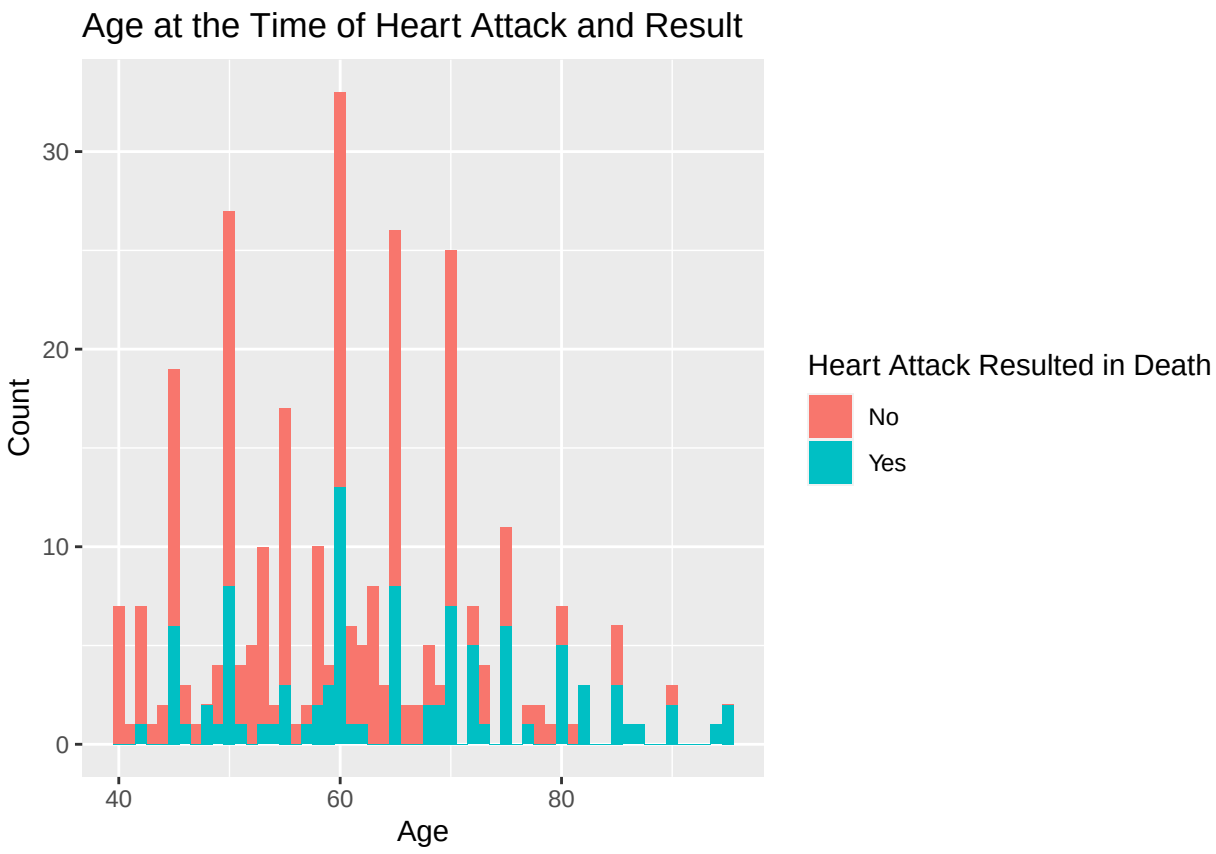
```
ggplot(heart,aes(high_blood_pressure,fill = DEATH_EVENT))+
  geom_bar(position = "fill")+
  labs(y = "Proportion of Results",x = "Blood Pressure")+
  scale_x_discrete(labels = c("No","Yes"))
```



Age

From this it is evident that probablitiy to survive a heart attack decreases with age.

```
ggplot(heart, aes(age, fill = DEATH_EVENT))+
geom_histogram(binwidth = 1)+
  labs(title = "Age at the Time of Heart Attack and Result"
       ,y = "Count",x = "Age")+
  scale_fill_discrete(name = "Heart Attack Resulted in Death",
                      labels = c("No","Yes"))
```



Data Splitting

Using caret package to split the data in 70:30 ratio.

```
set.seed(100)
split <- sample.split(heart,SplitRatio = 0.7)
tr <- subset(heart,split == T)
ts<- subset(heart,split == F)
```

Now that the dataset has been split into two parts, 1. Training dataset 2. Testing dataset

Model Making (Random Forest)

Making a different variable for formula in order to save typing later.

```
formula <-"DEATH_EVENT ~ age+anaemia+creatinine_phosphokinase+
diabetes+ejection_fraction+high_blood_pressure+
platelets+serum_creatinine+serum_sodium+
sex+smoking+time"

formula <- as.formula(formula)
```

Tuning RF

```
bestm <- tuneRF(tr,tr$DEATH_EVENT , stepFactor = 1.1,improve = 0.01,trace = T,plot = F)
```

```
## mtry = 3  OOB error = 0%
## Searching left ...
## Searching right ...
```

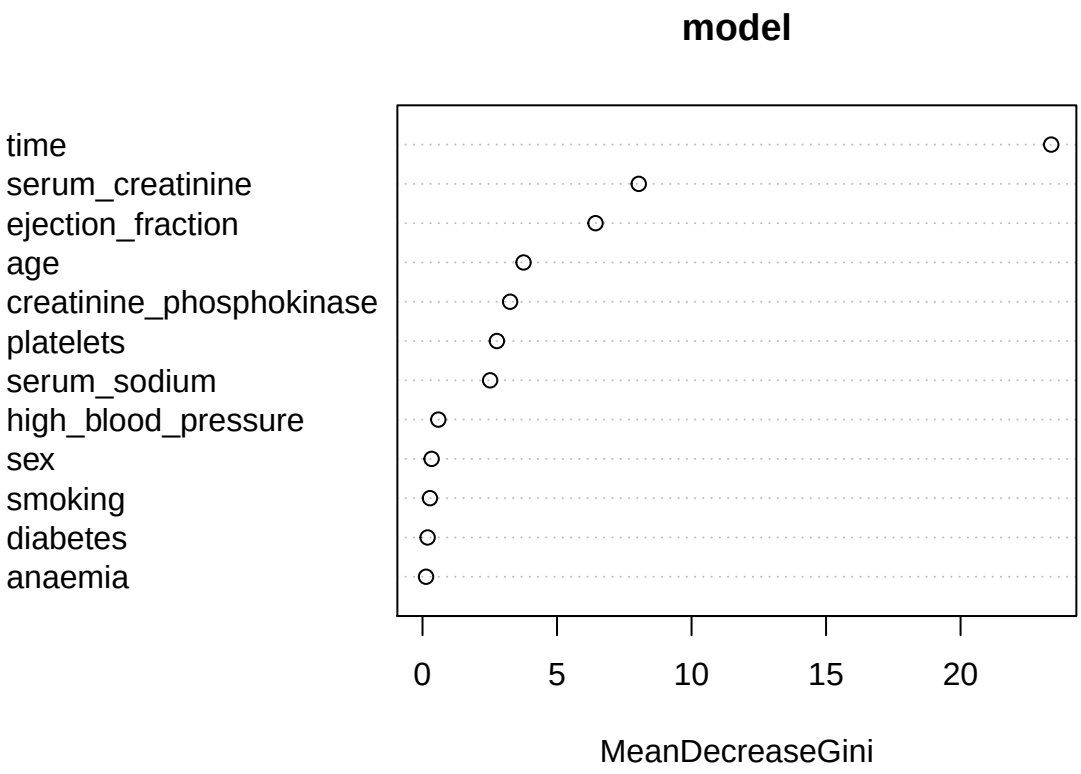
The OOB error for **mtry = 3** is zero, so we'll be using this value.

Training the Model using Random Forest

```
set.seed(111)
model <- randomForest(formula , data= tr,ntree=1000,mtry=3,nodesize = 0.1*nrow(tr))
```

After tuning the hyperparameters the values for different hyperparameters is used as above.

```
varImpPlot(model) ##The variables ae plotted according to their importance in model
```



Prediction

```
pred <- predict(model, newdata = ts)
```

Now the model has predicted the values for DEATH_EVENT using the model we have trained. Let's look at the accuracy of the model.

```
confusionMatrix(pred,ts$DEATH_EVENT)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 61  7
##          1  2 22
##
##                Accuracy : 0.9022
##                  95% CI : (0.8224, 0.9543)
##     No Information Rate : 0.6848
##     P-Value [Acc > NIR] : 7.725e-07
##
##                   Kappa : 0.7623
##
##  Mcnemar's Test P-Value : 0.1824
##
##             Sensitivity : 0.9683
##             Specificity : 0.7586
##          Pos Pred Value : 0.8971
##          Neg Pred Value : 0.9167
##              Prevalence : 0.6848
##          Detection Rate : 0.6630
##    Detection Prevalence : 0.7391
##       Balanced Accuracy : 0.8634
##
##        'Positive' Class : 0
##
```

The accuracy is **90.22 %** and is calculated using the confusion matrix generated by confusionMatrix function in caret package.