Harvardx Capstone Project 2

Daisuke Ohnuki

6/16/2021

## Introduction

Heart failure is the one of the most crucial matters in hospitalizing. We will find how it is caused by related factors in seeing the “heart\_failure\_clinical\_records\_dataset.csv”, provided by Larxel at Kaggle.[1] We use machine learning technique in R to predict the accuracy of the models including Desicion Tree, k-Nearest neighbour and Random forest model. To facilitate this project, we will look through the dataset with visualization first. Second, we brush up and select variables for machine learning models we described above. Then we build up the modelings to find the highest accuracy. We conclude with our outcome for the results of the accuracy, with limitations of this project and possibilities for future works.

### Load libraries

# We will install libraries for our analysis and modeling.  
knitr::opts\_chunk$set(echo = TRUE, warning = FALSE)  
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")

## Loading required package: tidyverse

## ─ Attaching packages ──────────────────── tidyverse 1.3.0 ─

## ✓ ggplot2 3.3.3 ✓ purrr 0.3.4  
## ✓ tibble 3.1.0 ✓ dplyr 1.0.5  
## ✓ tidyr 1.1.3 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.1

## ─ Conflicts ───────────────────── tidyverse\_conflicts() ─  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

if(!require(e1071)) install.packages("e1071", repos = "http://cran.us.r-project.org")

## Loading required package: e1071

if(!require(randomForest)) install.packages("randomForest", repos = "http://cran.us.r-project.org")

## Loading required package: randomForest

## randomForest 4.6-14

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

##   
## Attaching package: 'randomForest'

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

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

if(!require(rsample)) install.packages("rsample", repos = "http://cran.us.r-project.org")

## Loading required package: rsample

##   
## Attaching package: 'rsample'

## The following object is masked from 'package:e1071':  
##   
## permutations

if(!require(tinytex)) install.packages("tinytex", repos = "http://cran.us.r-project.org")

## Loading required package: tinytex

if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")

## Loading required package: data.table

##   
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':  
##   
## between, first, last

## The following object is masked from 'package:purrr':  
##   
## transpose

if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")

## Loading required package: caret

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

if(!require(ggplot2)) install.packages("ggplot2", repos = "http://cran.us.r-project.org")  
if(!require(corrplot)) install.packages("corrplot", repos = "http://cran.us.r-project.org")

## Loading required package: corrplot

## corrplot 0.84 loaded

if(!require(latexpdf)) install.packages("latexpdf", repos = "http://cran.us.r-project.org")

## Loading required package: latexpdf

library(dplyr)  
library(tidyverse)  
library(tinytex)  
library(e1071)  
library(randomForest)  
library(rsample)  
library(data.table)  
library(caret)  
library(ggplot2)  
library(corrplot)  
library(latexpdf)

### Data setting

Then we set the data. We download the dataset, “heart\_failure\_clinical\_records\_dataset.csv”, from the Kaggle site. The data is provided by Larxel.

# Dawnload the dataset from the website;  
#https://www.kaggle.com/andrewmvd/heart-failure-clinical-data The data is provided by Larxel.  
data<- read.csv ("heart\_failure\_clinical\_records\_dataset.csv",   
 header = TRUE)

### Summary of the dataset

#We can see the summary of the dataset.  
#The data set has 299 rows with 13 variables.  
summary(data)

## 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

### Explanation of the variables

The DEATH\_EVENT variables will be the dependent variable. 1.age = Age of patient

2.anaemia = Decrease of red blood cells or hemoglobin (0=False, 1=True)

3.creatinine\_phosphokinase = Creatine phosphokinase, or CPK,is an enzyme in the body. This variabe shows the level of the CPK enzyme in the blood. (in mcg/L)

4.diabetes - It implies whether the patient has diabetes. (0=False, 1=True)

5.ejection\_fraction - Ejection fraction is a measurement of how much blood the left ventricle pumps out with each contraction. (in percentage)

6.high\_blood\_pressure - It shows whether the patient has hypertension. (0=False, 1=True)

7.platelets - Platelets, also called thrombocytesl, are a component of blood whose function is to react to bleeding from blood vessel injury by clumping, thereby initiating a blood clot.(kiloplatelets/mL)

8.serum\_creatinine - Level of serum creatinine in the blood (in mg/dL)

9.serum\_sodium - Level of serum sodium in the blood (in mEq/L)

10.sex - Female= 0, Male = 1

11.smoking - If the patient smokes, it returns 1.

12.time - Follow-up period of the patient in days.

13.DEATH\_EVENT - If the patient deceased during the follow-up period, it returnes 1. Or, survived, 0.

### Structure of the dataset

Also, it seems effective to see the structure of the dataset. It suggests that “age”, “platelets” and “serum\_creatinine” are numerical. Others are intergers.

### Head of the dataset

## Exploratory Data Analysis

#Copy the data as "heartd" for later modeling.  
heartd <- data

### Check any missing value.

There is no missing value on the dataset.

#There is no missing value on the dataset.  
anyNA(data)

## [1] FALSE

### Data visualization

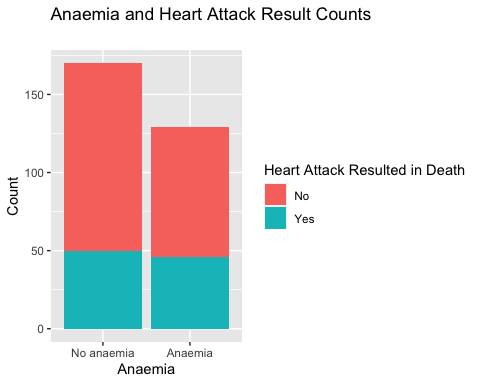
#For visualization, convert numeric to factor.  
data$DEATH\_EVENT <- as.factor(data$DEATH\_EVENT)  
data$anaemia <- as.factor(data$anaemia)  
data$diabetes <- as.factor(data$diabetes)  
data$high\_blood\_pressure <- as.factor(data$high\_blood\_pressure)  
data$sex <- as.factor(data$sex)  
data$smoking <- as.factor(data$smoking)

### Distribution of binary variables

#### Anaemia and Heart Attack

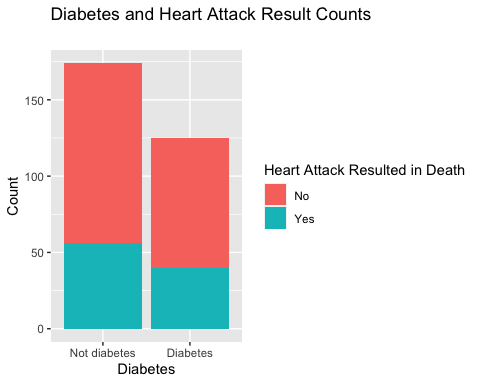
In the first half of this section, we show the distribution of numeric variables with the heart attack in death. We suspect that there would be no significant difference between the number of the death of “No anaemia” and “Anaemia”.

#1.Anaemia and Heart Attack in death  
f1 <- ggplot(data,aes(anaemia,fill = DEATH\_EVENT))+  
 geom\_bar()+  
 labs(title = "Anaemia and Heart Attack Result Counts\n",  
 y = "Count",  
 x = "Anaemia")+  
 theme(legend.position = "right")+   
 scale\_fill\_discrete(name = "Heart Attack Resulted in Death", labels = c("No","Yes"))+  
 scale\_x\_discrete(labels = c("No anaemia","Anaemia"))  
f1



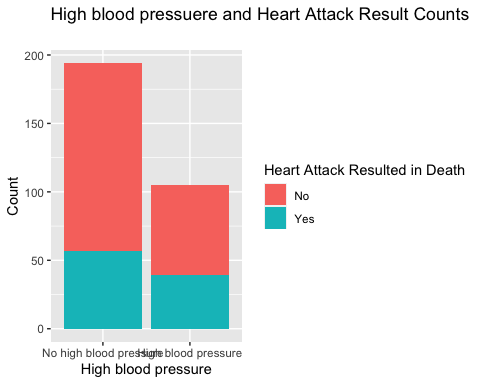
#### Diabetes and Heart Attack in death

#2.Diabetes and Heart Attack in death  
f2 <- ggplot(data,aes(diabetes,fill = DEATH\_EVENT))+  
 geom\_bar()+  
 labs(title = "Diabetes and Heart Attack Result Counts\n",  
 y = "Count",  
 x = "Diabetes")+  
 theme(legend.position = "right")+   
 scale\_fill\_discrete(name = "Heart Attack Resulted in Death", labels = c("No","Yes"))+  
 scale\_x\_discrete(labels = c("Not diabetes","Diabetes"))  
f2



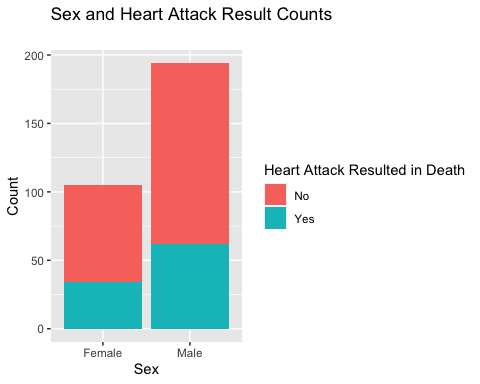
#### High blood pressuere and Heart Attack

#3.High blood pressuere and Heart Attack in death  
f3 <- ggplot(data,aes(high\_blood\_pressure,fill = DEATH\_EVENT))+  
 geom\_bar()+  
 labs(title = "High blood pressuere and Heart Attack Result Counts\n",  
 y = "Count", x = "High blood pressure")+  
 theme(legend.position = "right")+   
 scale\_fill\_discrete(name = "Heart Attack Resulted in Death", labels = c("No","Yes"))+  
 scale\_x\_discrete(labels = c("No high blood pressure","High blood pressure"))  
f3



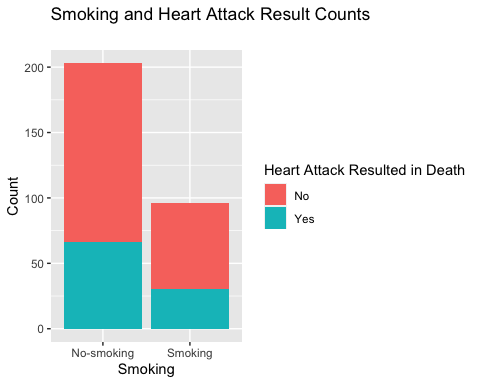
#### Sex and Heart Attack

#4.Sex and Heart Attack in death  
f4 <- ggplot(data,aes(sex,fill = DEATH\_EVENT))+  
 geom\_bar()+  
 labs(title = "Sex and Heart Attack Result Counts\n",  
 y = "Count", x = "Sex")+  
 theme(legend.position = "right")+   
 scale\_fill\_discrete(name = "Heart Attack Resulted in Death", labels = c("No","Yes"))+  
 scale\_x\_discrete(labels = c("Female","Male"))  
f4



#### Age and heart attack in death

#5.Smoking and Heart Attack in death  
f5 <- ggplot(data,aes(smoking,fill = DEATH\_EVENT))+  
 geom\_bar()+  
 labs(title = "Smoking and Heart Attack Result Counts\n",  
 y = "Count",  
 x = "Smoking")+  
 theme(legend.position = "right")+   
 scale\_fill\_discrete(name = "Heart Attack Resulted in Death", labels = c("No","Yes"))+  
 scale\_x\_discrete(labels = c("No-smoking","Smoking"))  
f5

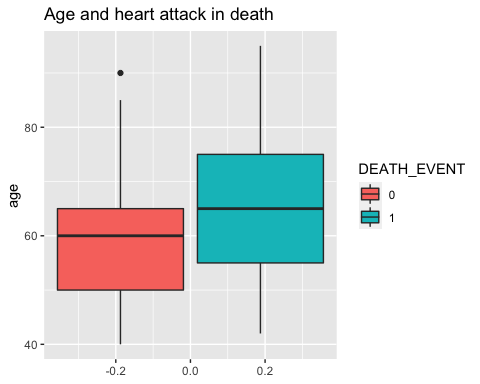


### Distribution of numeric variables

#### Age and heart attack in death

As the age goes up from 60, the total death event increase.

#6.Age and heart attack in death  
f6 <- data %>%  
 select(age, DEATH\_EVENT) %>%  
 ggplot(aes(x = age, fill = DEATH\_EVENT)) +  
 geom\_boxplot(show.legend = TRUE) +  
 coord\_flip() +  
 theme(legend.position = "right")+  
 ggtitle("Age and heart attack in death")  
f6

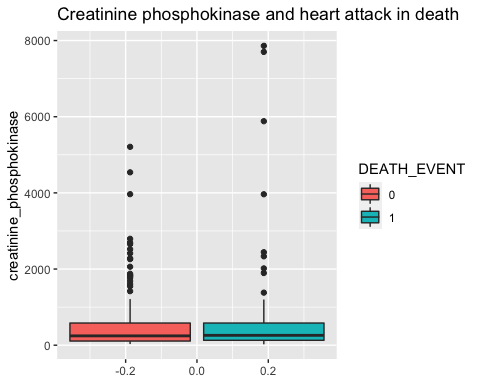


summary(data$age)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 40.00 51.00 60.00 60.83 70.00 95.00

#### Creatinine phosphokinase and heart attack in death

#7. Creatinine phosphokinase and heart attack in death  
f7 <- data %>%  
 select(creatinine\_phosphokinase, DEATH\_EVENT) %>%  
 ggplot(aes(x = creatinine\_phosphokinase, fill = DEATH\_EVENT)) +  
 geom\_boxplot(show.legend = TRUE) +  
 coord\_flip() +  
 theme(legend.position = "right")+  
 ggtitle("Creatinine phosphokinase and heart attack in death")  
f7

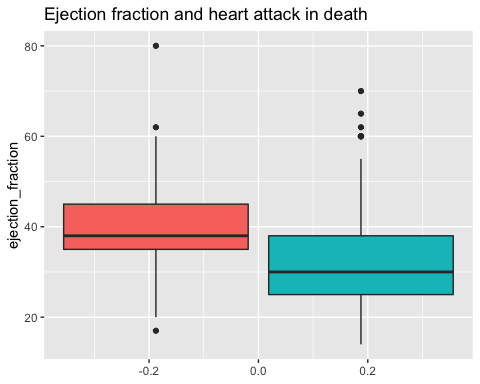


summary(data$creatinine\_phosphokinase)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 23.0 116.5 250.0 581.8 582.0 7861.0

#### Ejection fraction and heart attack in death

#8.Ejection fraction and heart attack in death  
f8 <- data %>%  
 select(ejection\_fraction, DEATH\_EVENT) %>%  
 ggplot(aes(x = ejection\_fraction, fill = DEATH\_EVENT)) +  
 geom\_boxplot(show.legend = FALSE) +  
 coord\_flip() +  
 theme(legend.position = "right")+  
 ggtitle("Ejection fraction and heart attack in death")  
f8

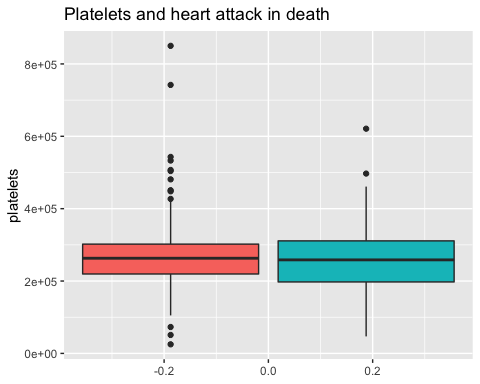


summary(data$ejection\_fraction)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 14.00 30.00 38.00 38.08 45.00 80.00

#### Platelets and heart attack in death

#9.Platelets and heart attack in death  
f9 <- data %>%  
 select(platelets, DEATH\_EVENT) %>%  
 ggplot(aes(x = platelets, fill = DEATH\_EVENT)) +  
 geom\_boxplot(show.legend = FALSE) +  
 coord\_flip() +  
 theme(legend.position = "right")+  
 ggtitle("Platelets and heart attack in death")  
f9

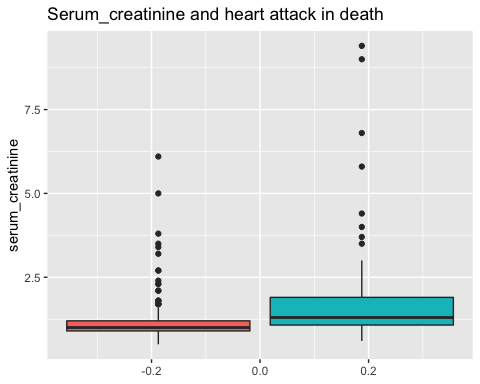


summary(data$platelets)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 25100 212500 262000 263358 303500 850000

#### Serum\_creatinine and heart attack in death

#10.Serum\_creatinine and heart attack in death  
f10 <- data %>%  
 select(serum\_creatinine, DEATH\_EVENT) %>%  
 ggplot(aes(x = serum\_creatinine, fill = DEATH\_EVENT)) +  
 geom\_boxplot(show.legend = FALSE) +  
 coord\_flip() +  
 theme(legend.position = "right")+  
 ggtitle("Serum\_creatinine and heart attack in death")  
f10

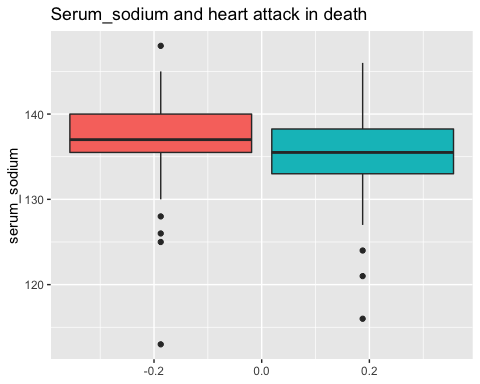


summary(data$serum\_creatinine)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.500 0.900 1.100 1.394 1.400 9.400

#### Serum sodium and heart attack in death

#11.Serum sodium and heart attack in death  
p11 <- data %>%  
 select(serum\_sodium, DEATH\_EVENT) %>%  
 ggplot(aes(x = serum\_sodium, fill = DEATH\_EVENT)) +  
 geom\_boxplot(show.legend = FALSE) +  
 coord\_flip() +  
 theme(legend.position = "right")+  
 ggtitle("Serum\_sodium and heart attack in death")  
p11

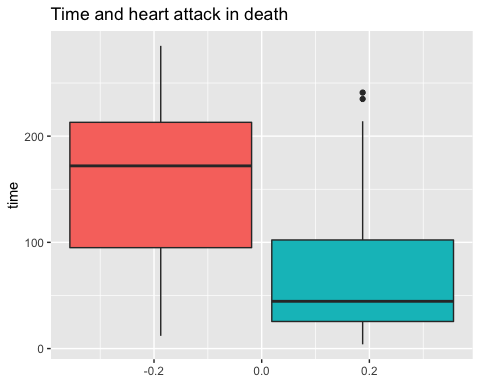


summary(data$serum\_sodium)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 113.0 134.0 137.0 136.6 140.0 148.0

#### Time and heart attack in death

#12.Time and heart attack in death  
p12 <- data %>%  
 select(time, DEATH\_EVENT) %>%  
 ggplot(aes(x = time, fill = DEATH\_EVENT)) +  
 geom\_boxplot(show.legend = FALSE) +  
 coord\_flip() +  
 theme(legend.position = "right")+  
 ggtitle("Time and heart attack in death")  
p12



summary(data$time)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 4.0 73.0 115.0 130.3 203.0 285.0

### Correlation of the variables

First, we have to prepare the data for the correlation.

#Prepare for the correlation.  
f\_features = c("anaemia", "diabetes", "high\_blood\_pressure", "sex", "smoking", "DEATH\_EVENT")  
  
heart\_n <- heartd  
heartd <- heartd %>%  
 mutate\_at(f\_features, as.factor)

We can see the p-value of the variables in the correlation map. We take the p-value which are less than 0.05, as significant parameters. It suggests that we should focus on “age”,“ejection\_fraction”, “serum\_creatinine”, “serum\_sodium” and “time”, for predicting “DEATH\_EVENT”.

#Use heart\_n data  
#We can also see the p-value of the variables in the correlation map. We take the p-value which are less than 0.05, as significant parameters. It suggests that we should focus on "age","ejection\_fraction", "serum\_creatinine", "serum\_sodium" and "time", for predicting "DEATH\_EVENT".  
cor(heart\_n) %>%  
 corrplot(method = "circle", type = "lower", tl.col = "black", tl.srt = 15,  
 p.mat = cor.mtest(heart\_n)$p,  
 insig = "p-value", sig.level = -1)



### Data cleaning

For our modeling of machine learning, we will clean the data. As the previous section suggest, we pick up five variables for the prediction for the death event.

# As we set the DEATH EVENT as the dependent variable, we focus on the five variables as follows; age,ejection\_fraction, serum\_creatinite, serum\_sodium and time.  
keep\_columns <- c("age","ejection\_fraction", "serum\_creatinine", "serum\_sodium", "time", "DEATH\_EVENT")   
cleaned\_data <- heartd[, keep\_columns]   
  
# We are now ready to select a machine learning algorithm to create a prediction   
# model for our datasets.  
cols <- c("DEATH\_EVENT" )   
cleaned\_data[cols] <- lapply(cleaned\_data[cols], factor)  
str(cleaned\_data)

## 'data.frame': 299 obs. of 6 variables:  
## $ age : num 75 55 65 50 65 90 75 60 65 80 ...  
## $ ejection\_fraction: int 20 38 20 20 20 40 15 60 65 35 ...  
## $ 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 ...  
## $ time : int 4 6 7 7 8 8 10 10 10 10 ...  
## $ DEATH\_EVENT : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...

## Modeling

Creating the Training and Testing Sets  
In order to predict heart disease in patients, we will separate the dataset into a training set, as “train\_set” and a testing set, “test\_set”. To refrain overlearning or learning shortage, we will set 80% for the train set, 20% for the test set.

# We will separate the test set as 20% from the orignal dataset.  
set.seed(1980)  
index <- createDataPartition(y = data$DEATH\_EVENT, times = 1, p = 0.2,   
 list = FALSE)  
train\_set <- cleaned\_data[-index,]  
test\_set <- cleaned\_data[index,]

summary(train\_set)

## age ejection\_fraction serum\_creatinine serum\_sodium   
## Min. :40.00 Min. :14.00 Min. :0.500 Min. :113.0   
## 1st Qu.:50.00 1st Qu.:30.00 1st Qu.:0.900 1st Qu.:134.0   
## Median :60.00 Median :38.00 Median :1.100 Median :137.0   
## Mean :60.56 Mean :38.09 Mean :1.398 Mean :136.6   
## 3rd Qu.:69.00 3rd Qu.:45.00 3rd Qu.:1.400 3rd Qu.:140.0   
## Max. :94.00 Max. :80.00 Max. :9.400 Max. :148.0   
## time DEATH\_EVENT  
## Min. : 4.0 0:162   
## 1st Qu.: 73.0 1: 76   
## Median :120.0   
## Mean :132.7   
## 3rd Qu.:205.8   
## Max. :285.0

### Naive Bayes model

First, we choose Naive Bayes model.

# Train and predict using Naive Bayes  
set.seed(1980)  
train\_nb <- train(DEATH\_EVENT ~ ., method = "nb", data = train\_set)  
y\_hat\_nb <- predict(train\_nb, test\_set)  
nb\_accuracy <- confusionMatrix(data = y\_hat\_nb, reference = test\_set$DEATH\_EVENT,   
 positive = NULL)$overall["Accuracy"]  
nb\_accuracy

## Accuracy   
## 0.8032787

### Decision tree model

Second, we set the decision tree model.

#Train a decision tree model  
set.seed(1980)   
train\_rpart <- train(DEATH\_EVENT ~ .,  
 method = "rpart",  
 tuneGrid = data.frame(cp = seq(0, 0.1, len=25)),  
 data = train\_set)  
#Use best tune code for the optimal results  
train\_rpart$bestTune

## cp  
## 25 0.1

#Compute the accuracy of our decision tree model on the testing dataset  
dt\_accuracy <- confusionMatrix(predict(train\_rpart, test\_set),  
 test\_set$DEATH\_EVENT)$overall["Accuracy"]  
dt\_accuracy

## Accuracy   
## 0.8852459

### k-Nearest Neigbour Model

Third, we train a k-nearest neighbour algorithm.

set.seed(1980)  
train\_knn <- train(DEATH\_EVENT ~ ., method = "knn",  
 data = train\_set,  
 tuneGrid = data.frame(k = seq(2, 30, 2)))  
#Use best tune code for the optimal results.   
train\_knn$bestTune

## k  
## 5 10

#Compute the accuracy of our knn model on the testing dataset  
knn\_accuracy <-confusionMatrix(predict(train\_knn, test\_set, type = "raw"),  
 test\_set$DEATH\_EVENT)$overall["Accuracy"]  
knn\_accuracy

## Accuracy   
## 0.8688525

### Random Forest Model

Lastly, we try a random forest model for our fourth one.

set.seed(1980)  
# Define train control for k-fold (5-fold) cross validation  
train\_control <- trainControl(method="cv", number=5)  
# Train and predict using Random Forest  
train\_rf <- train(DEATH\_EVENT ~ ., data = train\_set,  
 method = "rf",  
 trControl = train\_control)  
y\_hat\_rf <- predict(train\_rf, test\_set)  
rf\_accuracy <- confusionMatrix(data = y\_hat\_rf, reference = test\_set$DEATH\_EVENT,   
 positive = NULL)$overall["Accuracy"]  
rf\_accuracy

## Accuracy   
## 0.852459

## Results

We gather the accuracy for each model.

#Results  
results <- data\_frame(  
 Model=c("Model 1: Naive Bayes",  
 "Model 2: Decision Tree",   
 "Model 3: Knn",  
 "Model 4: Random Forest" ),   
 Accuracy=c(nb\_accuracy, dt\_accuracy, knn\_accuracy, rf\_accuracy))  
results

## # A tibble: 4 x 2  
## Model Accuracy  
## <chr> <dbl>  
## 1 Model 1: Naive Bayes 0.803  
## 2 Model 2: Decision Tree 0.885  
## 3 Model 3: Knn 0.869  
## 4 Model 4: Random Forest 0.852

## Conclusion

As we described, we successfully predicted the death event from the five variables;“age”,“ejection\_fraction”, “serum\_creatinine”, “serum\_sodium” and “time”. We use four different models; Naive Bayes, Decision Tree, K-nearest neighbour and Random Forest model. We found that the decision tree model performed the best of the four, with the accuracy of 0.869. The limitation of this project is derived from that we did not use other machine learning models such as Support Vector Machne(SVM), neural network model or ensemble learning. For future work, for example, we should focus on other machine learning techniques to find out which would fit the the dataset we use. In addition, we might try to strengthen the model we use in this project by modifying the variables which we selected five. For example, we might predict the death event with four variables; age“,”ejection\_fraction“,”serum\_creatinine“,”serum\_sodium“, and excluding”time", or three. We found that we should do much more tries and errors to brush up our model.

## References

[1]<https://www.kaggle.com/andrewmvd/heart-failure-clinical-data>

[2]Irizarry A. Rafael (2018) Introduction to Data Science: Data Analysis and Prediction Algorithms with R.