Data Science: Capstone - Heart Failure prediction system

Francesco Mariotti - Harvard Data Science Professional Certificate Program

05/20/2021 - Bologna, Italy

Contents

1 Executive Summary

The word prediction in machine learning refers to the output of a trained model, representing the most likely value that will be obtained for a given input. Prediction in machine learning has a variety of applications, from chatbot development to recommendation systems.

The model is trained with historical data, and then predicts a selected property of the data for new inputs.

Prediction is used in lots of different areas, since it allows us to make highly accurate guesses about many things, such as predicting what the stock markets will do on any given day, predict results in sports, or even help the medical industry predict diseases.

The algorithms for prediction are classified as supervised learning algorithms since they need a training dataset with correct examples to learn from them.

The scope of this project is creating a machine learning based prediction system which can predict survival of patients with heart failure depending on some key parameters of their clinical picture.

The Dataset used is obtained from Davide Chicco and Giuseppe Jurman.

In the first part of this report will be presented an exploratory analysis of the dataset including the data pre-processing needed for further analysis and for bulding the prediction system.

The prediction systems builted on this dataset are evaluated and choosen based on accuracy of the model. Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right.

Formally, accuracy has the following definition:

$$\text{Acc} = \frac{Number\ of\ correct\ predictions}{Total\ numbers\ of\ predictions}$$

For accomplishing this goal, the **Tuned XGBoost model** is capable to reach an accuracy of **0.87**, which is accettable.

2 Exploratory Data Analysis

2.1 Inital data Exploration

2.1.1 Background:

Cardiovascular diseases kill approximately 17 million people globally every year, and they mainly exhibit as myocardial infarctions and heart failures. Heart failure (HF) occurs when the heart cannot pump enough blood to meet the needs of the body.

Available electronic medical records of patients quantify symptoms, body features, and clinical laboratory test values, which can be used to perform biostatistics analysis aimed at highlighting patterns and correlations otherwise undetectable by medical doctors.

Machine learning, in particular, can predict patients' survival from their data and can individuate the most important features among those included in their medical records.

2.1.2 The dataset

The dataset is composed by 299 patients with heart failure collected in 2015. For every patient were collected key parameters of their clinical picture which are theoretically and realistically correlated with their status.

Age	Anaemia	CPK	Diabetes	EF	HBP	P	SC	SS	Sex	Smoking	Time	Death Event
75	0	582	0	20	1	265000	1.9	130	1	0	4	1
55	0	7861	0	38	0	263358	1.1	136	1	0	6	1
65	0	146	0	20	0	162000	1.3	129	1	1	7	1
50	1	111	0	20	0	210000	1.9	137	1	0	7	1
65	1	160	1	20	0	327000	2.7	116	0	0	8	1
90	1	47	0	40	1	204000	2.1	132	1	1	8	1

The features/variables/columns in the datasets are the following:

- Age `<integer>` that contains the age of each patient at the time of the heart failure.
- Anaemia `<factor>` binary value which reveals the absence (0) or the presence (1) of Anaemia.
- Creatinine PhosphoKinase - CPK `<integer>` that contains the level of the CPK enzyme in the blood (mcg/L)
- Diabetes `<factor>` binary value which reveals the absence (0) or the presence (1) of Diabetes.
- Ejection Fraction - EF`<numeric>` that contains the title of each movie including the year of the release.
- High Blood Presure - HBP`<factor>` binary value which reveals the absence (0) or the presence (1) of hypertension.
- Platelets - P`<integer>` that count the number of platelets.
- Serum Creatinine - SC `<integer>` that contains the level of Serum Creatinine in the blood (mg/dL).
- Serum Sodium - SS `<integer>` that contains level of Serum Sodium in the blood (mEq/L).
- Sex `<factor>` binary value which reveals the sex. 0 if female, 1 if male
- Smoking `<factor>` binary value which reveals the nicotine addiction. 0 if absent, 1 if present
- Time `<integer>` that represents the follow up period (days)
- Death Event `<factor>` binary value which reveals if the patient deceased during the follow-up period 1 or not 0;

2.1.3 Data Pre - Processing and Data Exploration

The first steps of exploratory analysis are the preprocessing of the data that must be prepared to be used for effective understanding of the data.

First of all, for all continuous variables we define discrete intervals to have more easily interpretable data.

2.1.3.1 Age trend For instance, we could group patients by age range:

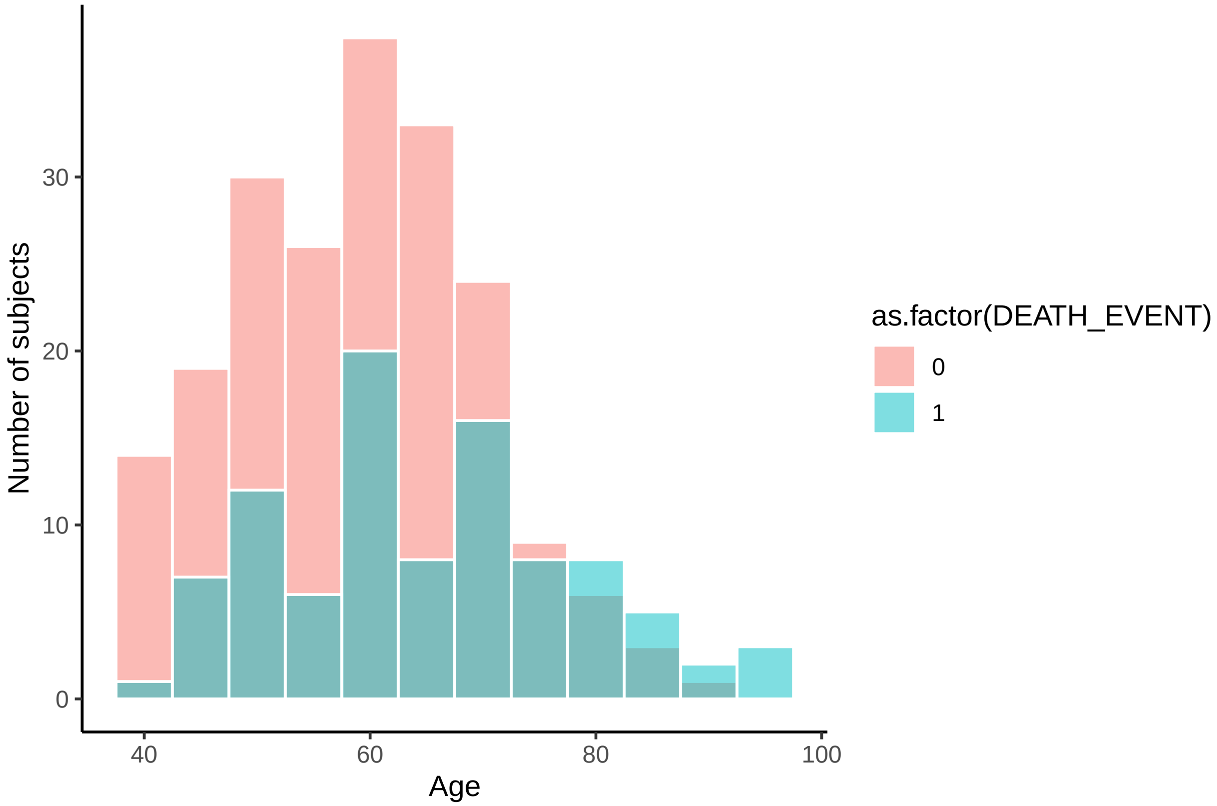
```
#Define useful variables for the data analysis
agebreaks <- c(40,45,50,55,60,65,70,75,80,85,90,200)
agelabels <- c("40-44","45-49","50-54","55-59","60-64","65-69",
               "70-74","75-79","80-84","85-89","90+")


#Set the data in order to have patients grouped by age intervals
setDT(heartfailure.dat)[ ,agegroups := cut(age,
                                           breaks = agebreaks,
                                           right = FALSE,
                                           labels = agelabels)]
```

Now that we have grouped the patients by age intervals we can plot the percentage of patients deceased per age intervals and display the data.

We immediately notice that:

- Majority of the patients are in their 50s (27.5%) and 60s (31.1%)
- As the age of a patient increases, the probabilty of death event increases.

Age Group	No. of Patients	No. of Deceased Patients	Percentage of patients who deceased (%)
40-44	18	1	5.6
45-49	29	10	34.5
50-54	48	11	22.9
55-59	34	9	26.5
60-64	55	15	27.3
65-69	38	12	31.6
70-74	36	13	36.1
75-79	16	7	43.8
80-84	11	8	72.7
85-89	8	5	62.5
90+	6	5	83.3



Age Distribution with Death Event

As seen before it is straightforward to group the data into intervals to have a better understanding and a clearer view on the distribution of patients in relation to the variables analyzed.

All the continuous variables have been grouped in discrete intervals as done before with the age. The results are presented below.

2.1.3.2 Creatinine Phosphokinase trend Creatine phosphokinase (a.k.a., creatine kinase, CPK, or CK) is an enzyme (a protein that helps to elicit chemical changes in your body) found in your heart, brain, and skeletal muscles. When muscle tissue is damaged, CPK leaks into your blood. Therefore, high levels of CPK usually indicate some sort of stress or injury to your heart or other muscles. To test CPK, blood is drawn from a vein in your arm

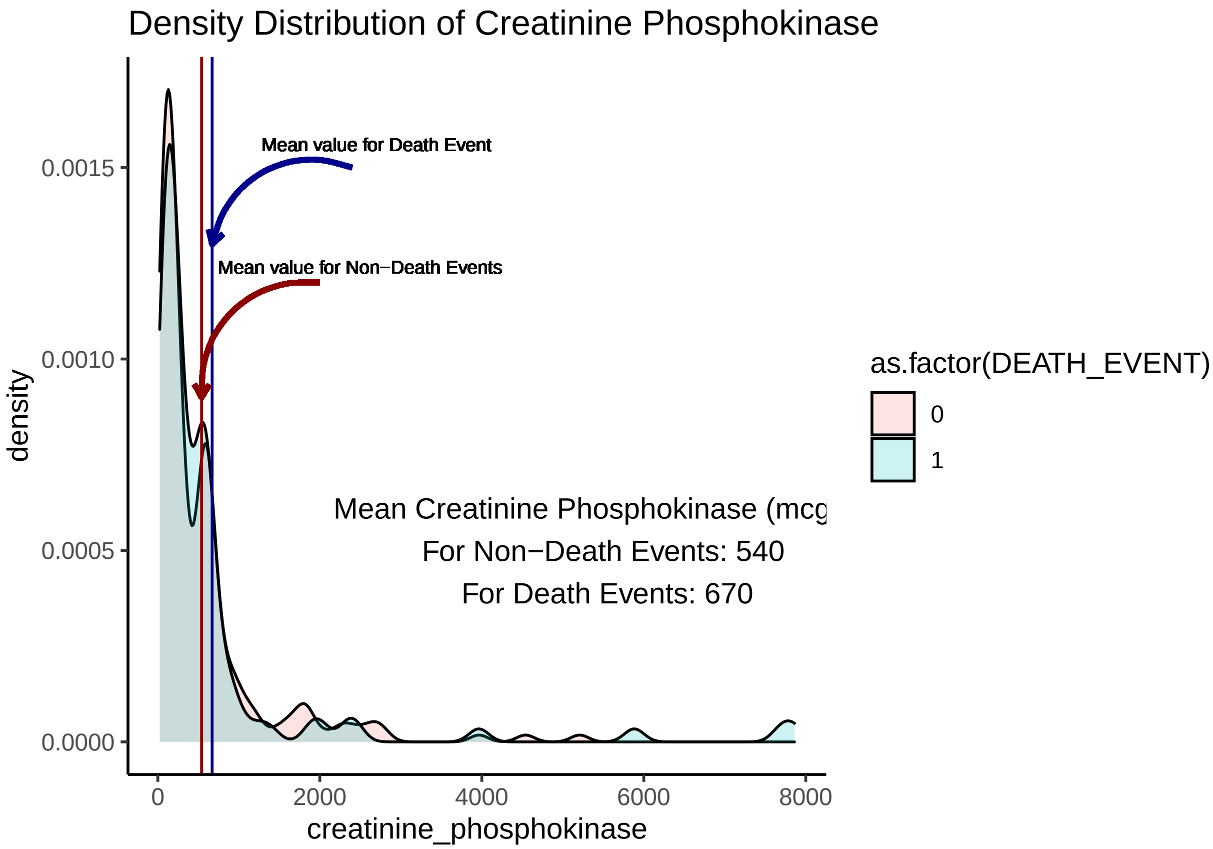
In the hospital, a person's CK-MB level is often checked when they exhibit signs of heart attack. However, in lupus treatment, an elevated CPK may suggest muscle inflammation due to disease activity or an overlapping condition. CPK levels can also be high after strenuous exercise, so your doctor may wish to recheck your CPK after several days of rest.

If your CPK is high with no exercise or remains high with rest, your doctor may order additional tests to determine which type (isoenzyme) of CPK is elevated. This information will help her/him to determine the source of the damage (skeletal muscles, heart, or brain). Certain medications, such as statins, can cause increases in CPK, so be sure to tell your doctor about any medications you currently take.

Considering the dataset:

- Levels of Creatinine Phosphokinase (in all subjects) range from 23.0 - 7861.0 (mcg/L) with mean 581.8 and median 250.0
- Mean values of Creatinine Phosphokinase level are 540.05 mcg/L for non-death events and 670.2 mcg/L for death events
- When level of Creatinine Phosphokinase level > 3500 mcg/L, the chances of death are at least 50%.

CP Range	No. of Patients	No. of Deceased Patients	Percentage of patients who deceased (%)
<500	183	59	32.2
500-999	80	27	33.8
1001-1499	9	2	22.2
1500-1999	9	1	11.1
2000-2499	7	3	42.9
2500-2999	4	0	0.0
3500-3999	2	1	50.0
4500-4999	1	0	0.0
5000-5499	1	0	0.0
5500+	3	3	100.0


Density Distribution of Creatinine Phosphokinase
Mean value for Death Event
Mean value for Non−Death Events
Mean Creatinine Phosphokinase (mcg
For Non−Death Events: 540
For Death Events: 670
density
0.0015
0.0010
0.0005
0.0000
0
2000
4000
6000
8000
creatinine_phosphokinase
as.factor(DEATH_EVENT)
0
1

2.1.3.3 Ejection Fraction trend Ejection fraction is a measurement of the percentage of blood leaving your heart each time it squeezes (contracts). It is just one of many tests your doctor may use to determine how your heart works.

The heart contracts and relaxes. When your heart contracts, it pumps out (ejects) blood from the two lower chambers (ventricles). When your heart relaxes, the ventricles refill with blood. No matter how forceful the contraction, the heart can never pump all blood out of a ventricle. The term "ejection fraction" refers to the percentage of blood that's pumped out of a filled ventricle with each heartbeat.

The ejection fraction is usually measured only in the left ventricle. The left ventricle is the heart's main pumping chamber. It pumps oxygen-rich blood up into your body's main artery (aorta) to the rest of the body.

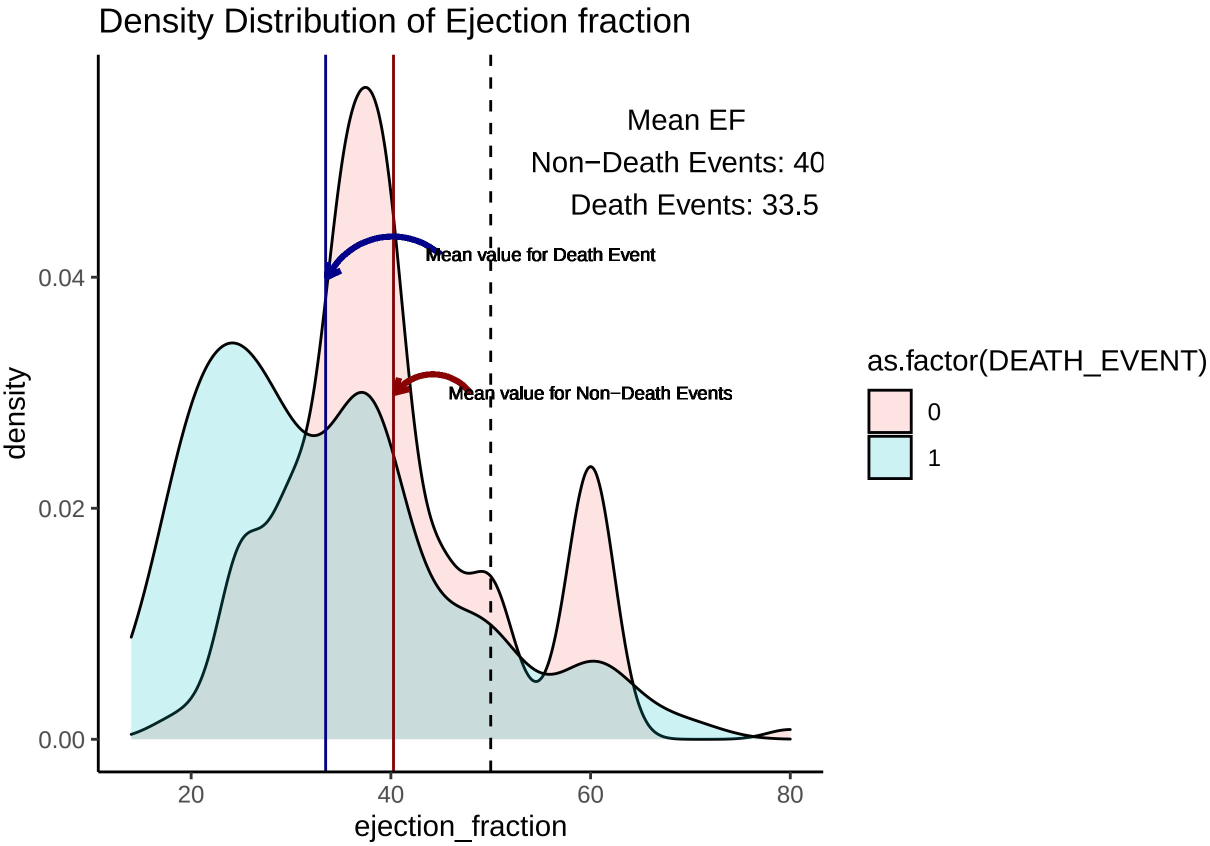
A normal ejection fraction is about 50% to 75%, according to the American Heart Association. A borderline ejection fraction can range between 41% and 50%.

Even if you have a normal ejection fraction, your overall heart function may not be normal. Talk with your doctor if you have concerns about your heart.

Some things that may cause a reduced ejection fraction are:

- Weakness of the heart muscle, such as cardiomyopathy
- Heart attack that damaged the heart muscle
- Heart valve problems
- Long-term, uncontrolled high blood pressure.

Death events usually corresponds to low values of Ejection Fraction; infact the mean values of Ejection Fracition are 40.3% for non-death events and 33.5% for death events.



2.1.3.4 Platelets trend Platelets are specialized disk-shaped cells in the blood stream that are involved in the formation of blood clots that play an important role in heart attacks, strokes, and peripheral vascular disease.

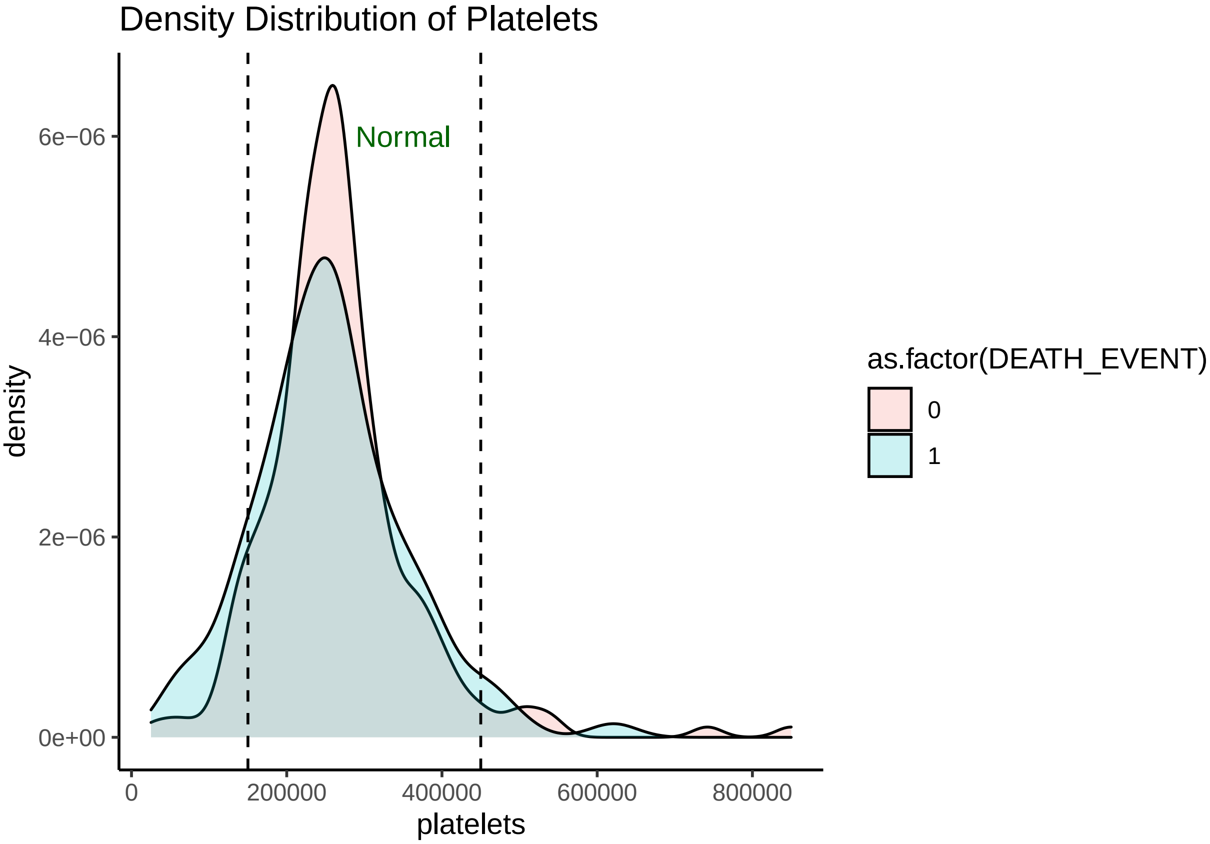
In most people, the more than 200 million platelets in a milliliter of blood act as tiny building blocks to form the basis of a clot to stop bleeding from cuts or injuries. Platelets can detect a disruption in the lining of a blood vessel and react to build a wall to stop bleeding

In cardiovascular disease, abnormal clotting occurs that can result in heart attacks or stroke. Blood vessels injured by smoking, cholesterol, or high blood pressure develop cholesterol-rich build- ups (plaques) that line the blood vessel; these plaques can rupture and cause the platelets to form a clot.

Even though no bleeding is occurring, platelets sense the plaque rupture and are confused, think- ing that an injury has taken place that will cause bleeding. Instead of sealing the vessel to prevent bleeding as would occur with a cut, a clot forms in an intact blood vessel, causing a blockage of blood flow.

A normal platelet count ranges from 150,000 to 450,000 platelets per microliter of blood.

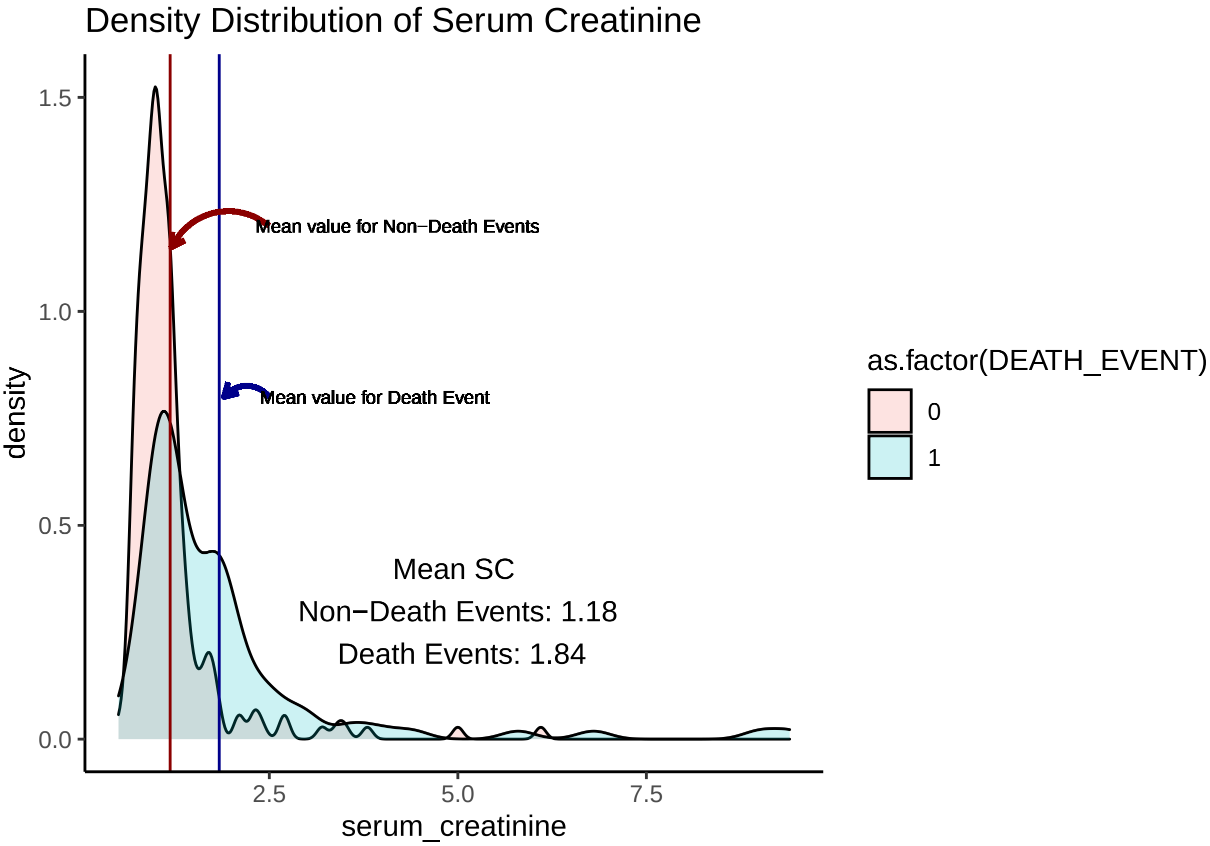
Mean values of Platelets counts are 266657 for non-death events and 256381 for death events As visible in the plot, the distributions of Platelets count in the absence or presence of death events are similar.



2.1.3.5 Serum Creatinine trend A serum creatinine test measures the level of creatinine in your blood and provides an estimate of how well your kidneys filter (glomerular filtration rate). The normal range for creatinine in the blood may be 0.84 to 1.21 mg/dL

Level of Serum Creatinine (in all subjects) range from 0.5 to 9.4 mg/dL, with mean 1.394 and median 1.1 Mean values of Serum Creatinine (mg/dL) are 1.18 for non-death events and 1.84 for death events When serum creatinine levels are greater than 2.5, chances of death > 60%.

SC Range	No. of Patients	No. of Deceased Patients	Percentage of patients who deceased (%)
<0.8	25	3	12
0.8-1.19	149	30	20.1
1.2-1.49	53	18	34
1.5-1.99	37	23	62.2
2.0-2.49	12	7	58.3
2.5-2.99	7	5	71.4
3+	16	10	62.5

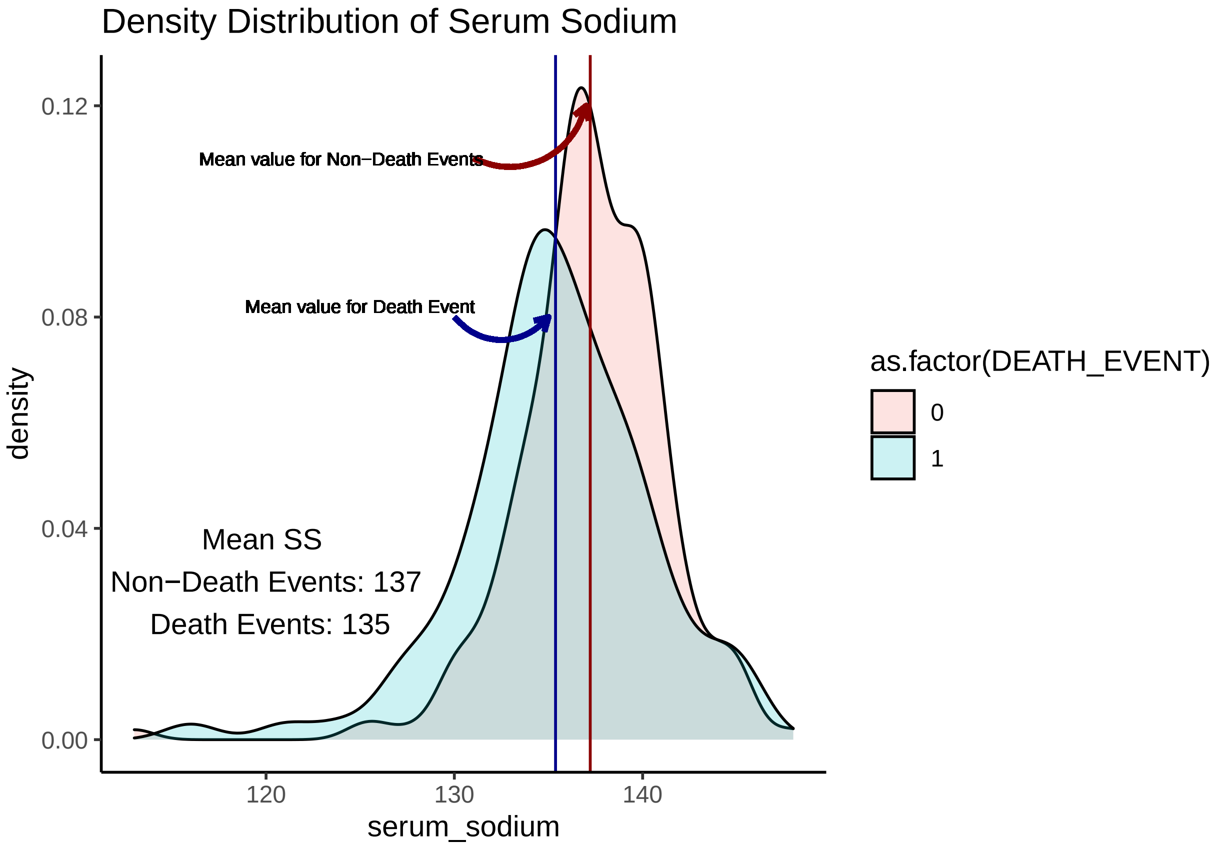


2.1.3.6 Serum Sodium trend A sodium blood test is a routine test that allows the doctors to see how much sodium is present in the blood. It's also called a serum sodium test. Sodium is an essential mineral to the body. It's also referred to as Na+.

Sodium is particularly important for nerve and muscle function. The body keeps sodium in balance through a variety of mechanisms. Sodium gets into the blood through food and drink. It leaves the blood through urine, stool, and swea. Too much sodium can raise the blood pressure resulting, in the long term, in heart fatigue/failure.

A normal blood sodium level is between 135 and 145 milliequivalents per liter (mEq/L). A serum sodium concentration of <135 mEq/L and is one of the most common biochemical disorders featured in heart failure patients.

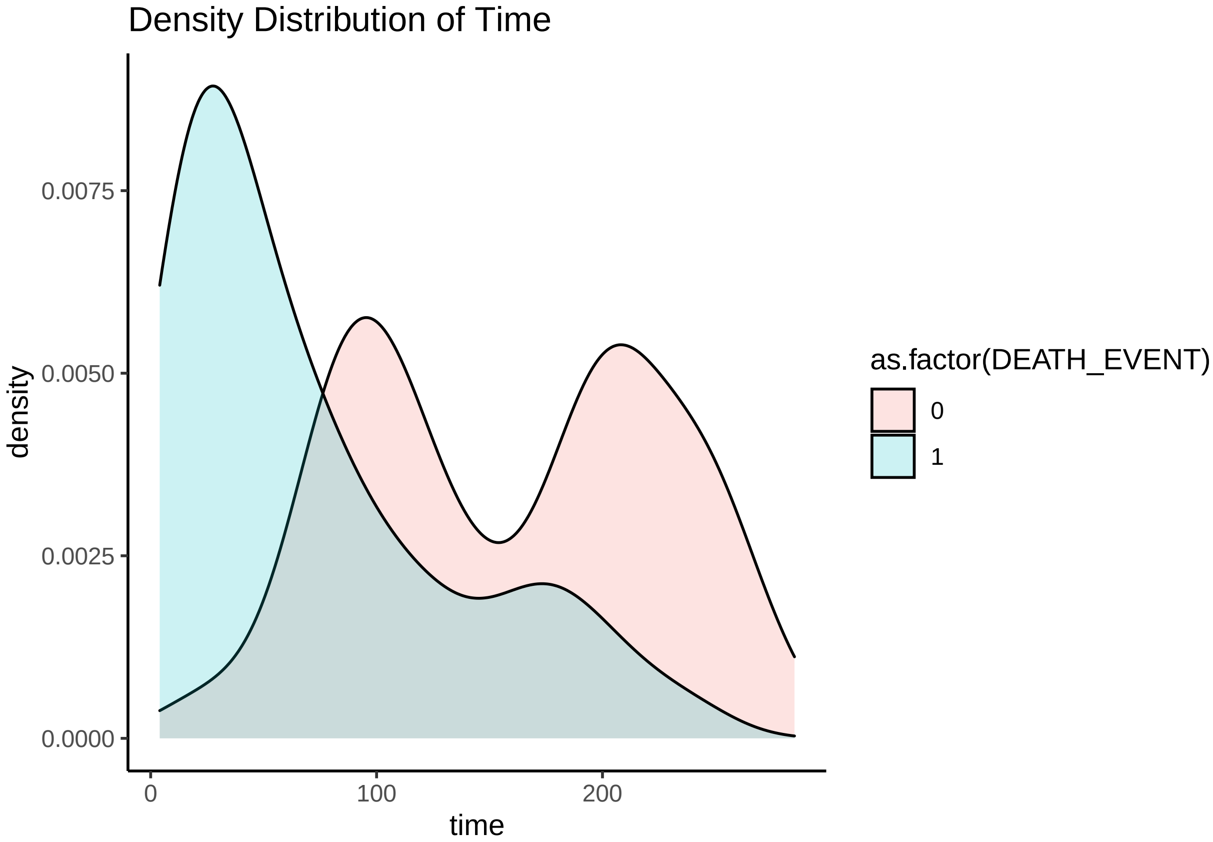
- Level of Serum Sodium (in all subjects) range from 113.0 to 148.0 mEq/L, with mean 136.6 and median 137.0
- Mean values of Serum Sodium (mEq/L) are 137.2 for non-death events and 135.4 for death events.



2.1.3.7 Time trend The data stored in the time column represent the follow-up period expressed in days.

It is logical to assume that, considering the average age of the patients in the sample under analysis, the longer the follow-up time, the greater the possibility that a death event will be required.

It is not trivial and somewhat surprising to justify a second peak in patients who died at relatively short “time” values.



In the dataframe there are “Boolean data” which means that they are data whose value can be 0 or 1 which can be interpreted, if you prefer, as False or True or vice versa.

In particular in the following plots we will have the following conditions (previosuly presented and reported here for completeness):

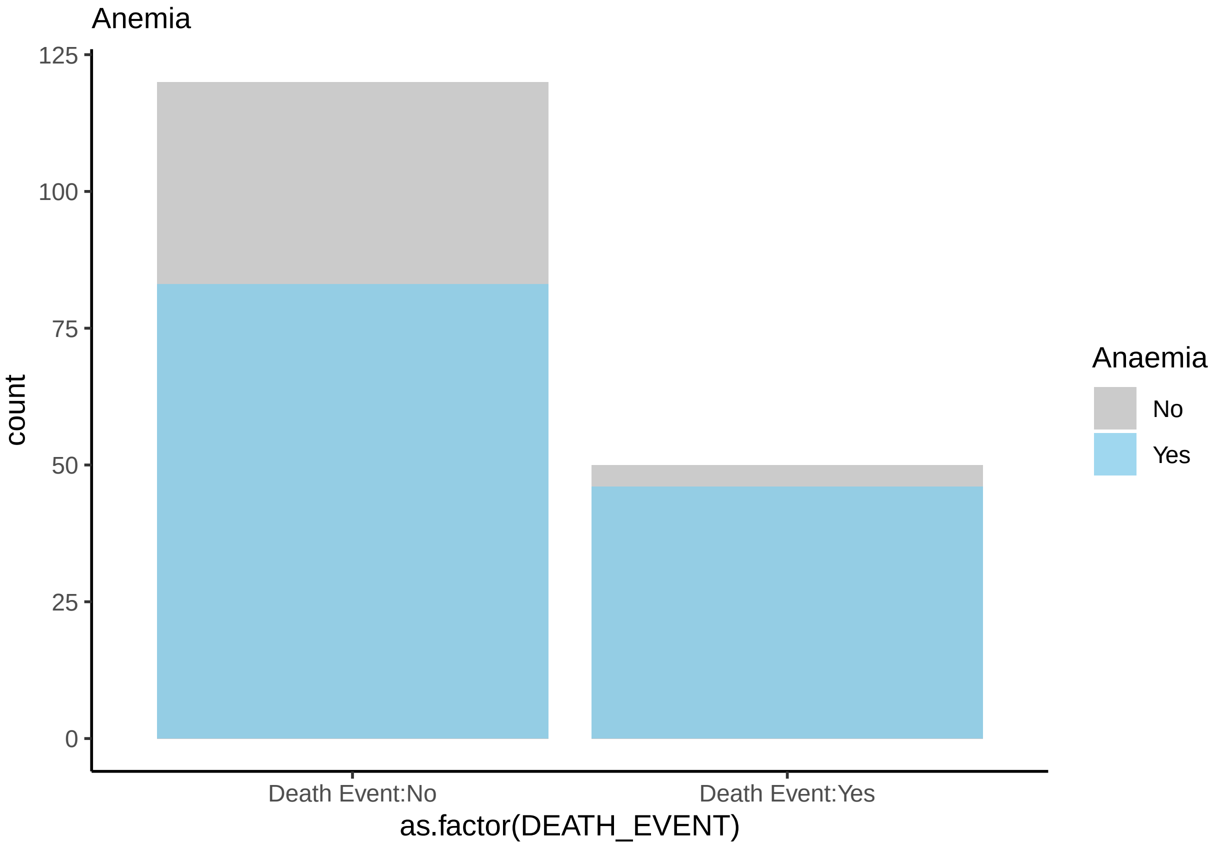
- Anaemia –> absence (0) presence (1)
- Diabetes –> absence (0) presence (1)
- High Blood Presure - HBP –> absence (0) presence (1)
- Sex –> female (0) male (1)
- Smoking addiction –> absence (0) presence (1).

It will be clearly notable that a higher proportion of patients who died had Anemia, Diabetes and High Blood Pressure.

2.1.3.8 Anaemia Effect

Anaemia is a decrease in the total amount of red blood cells (RBCs) or hemoglobin in the blood or a lowered ability of the blood to carry oxygen.

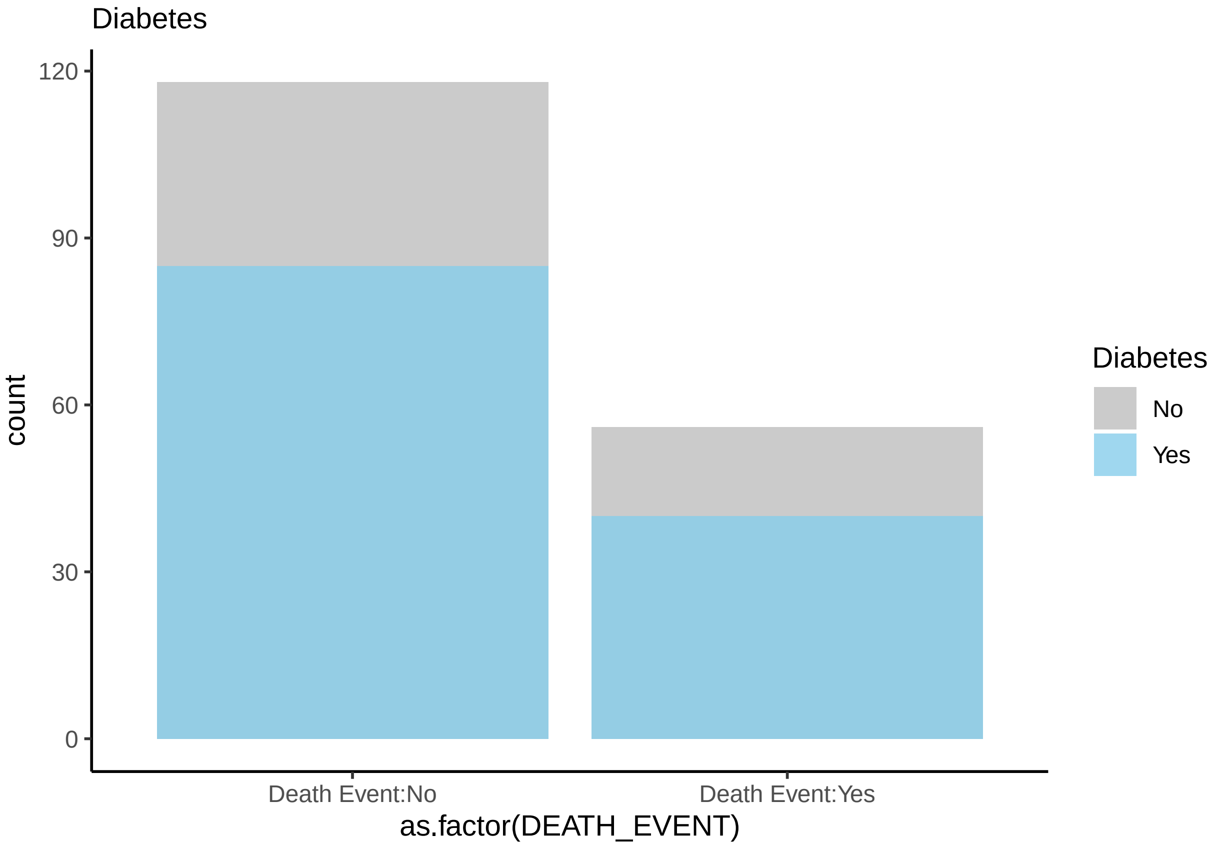
Anemia associated with heart failure is a frequent condition, which may lead to heart function deterioration by the activation of neuro-hormonal mechanisms. Therefore, a vicious circle is present in the relationship of heart failure and anemia. The consequence is reflected upon the patients’ survival, quality of life, and hospital readmissions.



2.1.3.9 Diabetes Effect Diabetes usually refers to diabetes mellitus, a group of metabolic diseases in which a person has high blood glucose levels over a prolonged period.

Over time, high blood sugar can damage blood vessels and the nerves that control your heart. People with diabetes are also more likely to have other conditions that raise the risk for heart disease:

- Too much LDL (“bad”) cholesterol in your bloodstream can form plaque on damaged artery walls.
- High triglycerides (a type of fat in your blood) and low HDL (“good”) cholesterol or high LDL cholesterol is thought to contribute to hardening of the arteries.

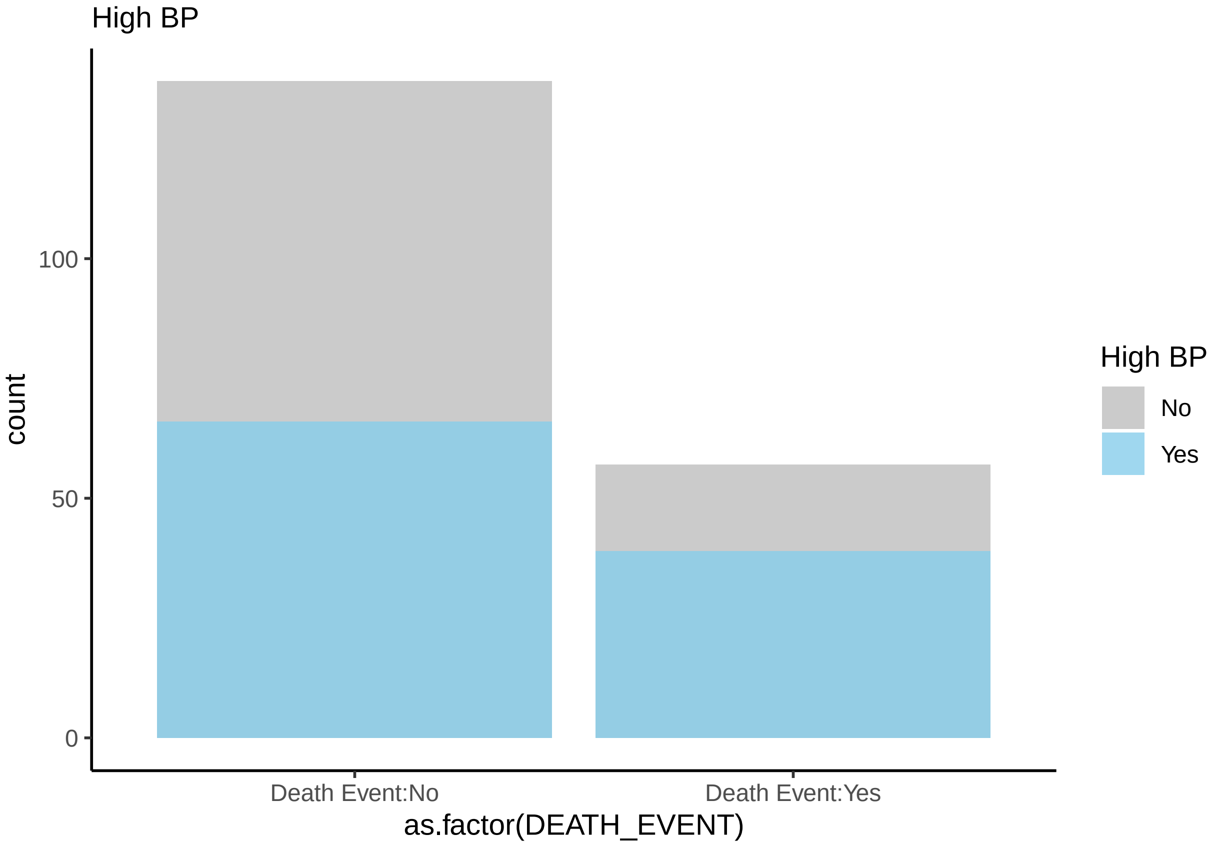


2.1.3.10 High Blood Pressure Effect

High blood pressure (also referred to as HBP, or hypertension) is when your blood pressure, the force of blood flowing through your blood vessels, is consistently too high.

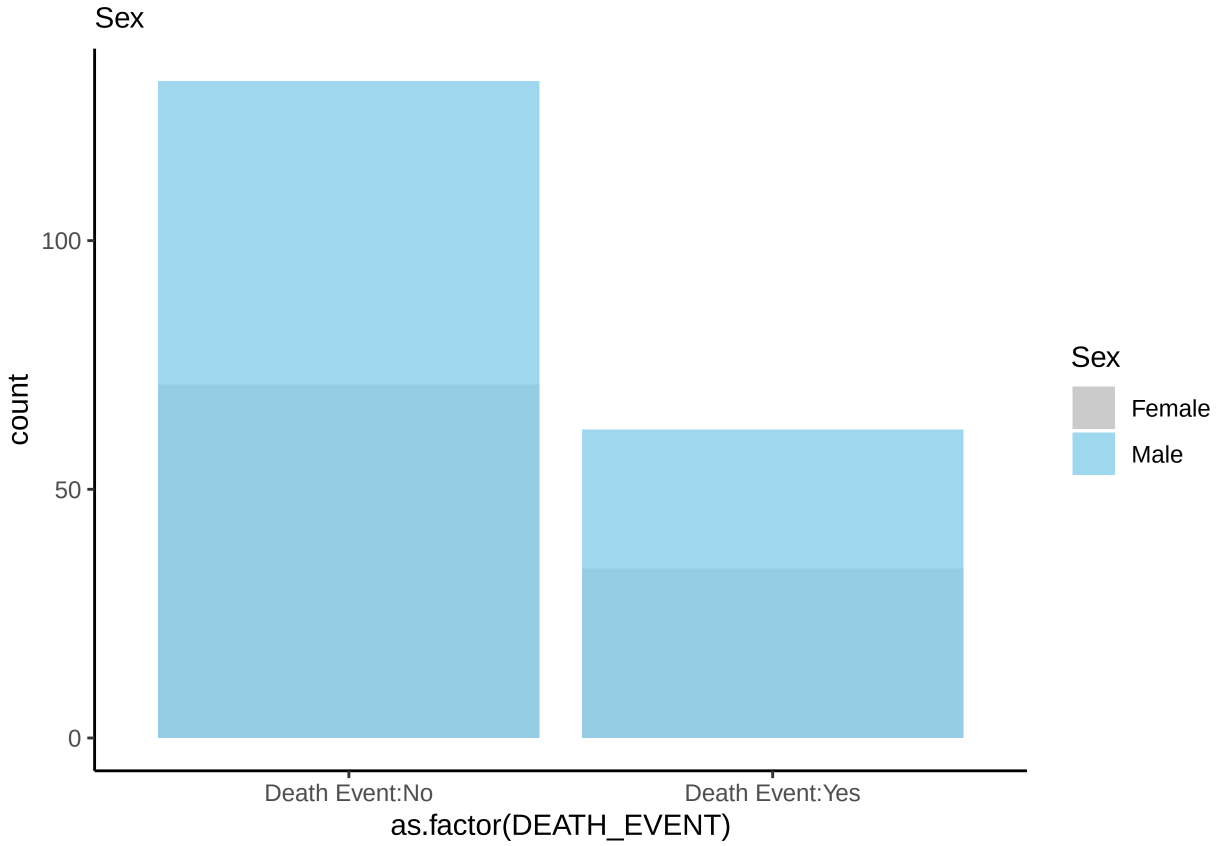
Your blood pressure changes throughout the day based on your activities. Having blood pressure measures consistently above normal may result in a diagnosis of high blood pressure (or hypertension).

The higher your blood pressure levels, the more risk you have for other health problems, such as heart disease, heart attack, and stroke.



2.1.3.11 Sex Effect Due to differences in the cardiovascular system, HF affects men and women in different ways. Currently, the mechanisms underlying these gender related differences remain unresolved, however research is underway to discover their root causes.

However the data collected in this dataframe do not show a considerable difference between men and women.

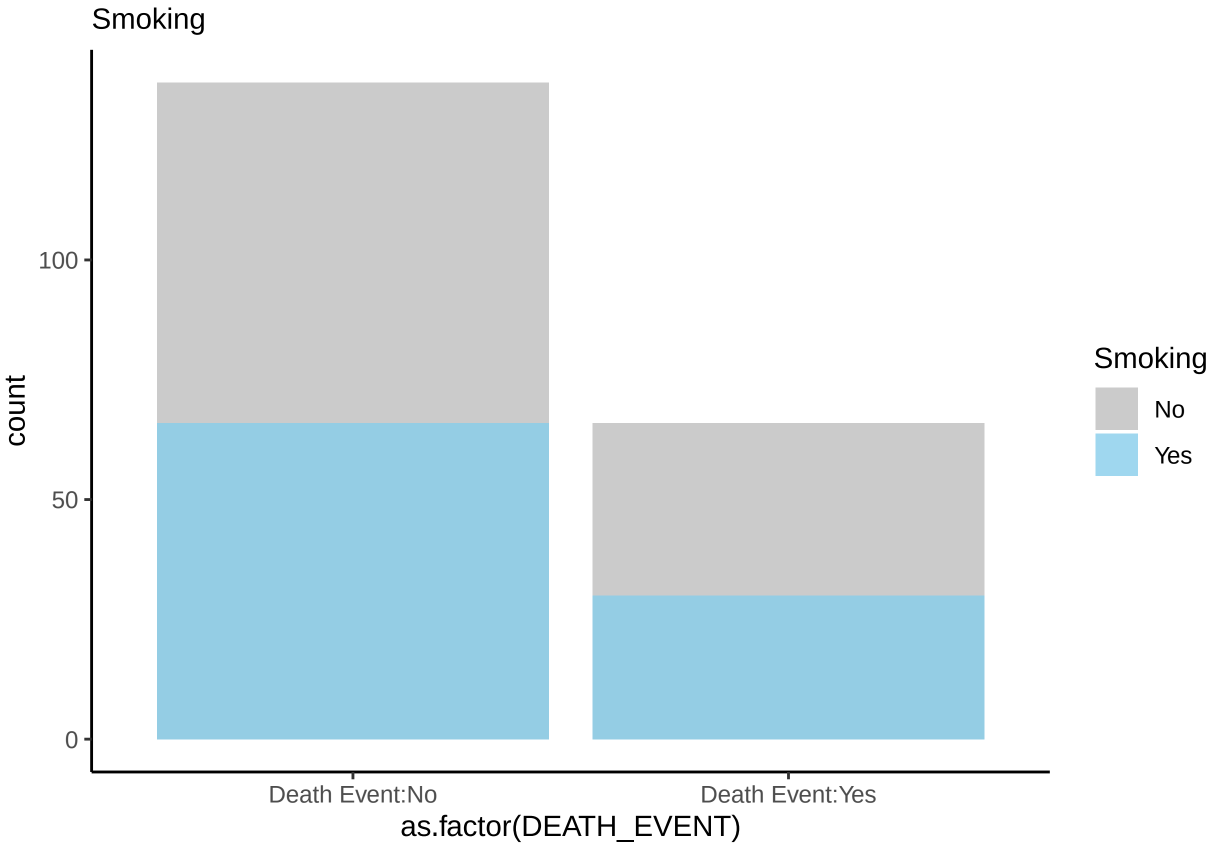


2.1.3.12 Smoking Effect Cigarette smoking is the chief cause of preventable disease and death in the United States and can harm nearly any part of the body.

Cigarette smoke is a toxic mix of more than 7,000 chemicals1 and, when inhaled, can interfere with important processes in the body that keep it functioning normally. One of these processes is the delivery of oxygen-rich blood to your heart and the rest of your body.

When you breathe in air from the atmosphere, the lungs take in oxygen and deliver it to the heart, which pumps this oxygen-rich blood to the rest of the body through the blood vessels.

But when you breathe in cigarette smoke, the blood that is distributed to the rest of the body becomes contaminated with the smoke's chemicals. These chemicals can damage to your heart and blood vessels, which can lead to cardiovascular disease.



2.2 Conclusion on the Exploratory Analysis

As a concluding remark for this first part of the report below are presented the key information obtained from the analysis of the data in the dataset.

- As the age of a patient increases, the probabilty of death event increases noticeably.
- When level of Creatinine Phosphokinase level > 3500 mcg/L, the chances of death are at least 50%.
- Death events usually corresponds to low values of Ejection Fraction.
- The contibution of platelets seems negligible or at least there are no differences between surviving and dead patients on this data.
- When serum creatinine levels are greater than 2.5, chances of death > 60%.
- Serum Sodium (mEq/L) contribute seems negligible or at least there are no evident differences between surviving and dead patients on this data.
- Time seems to be a key factor.
- Smoking does not seems to have a huge impact as the most would have expected
- higher proportion of patients who died had anemia, diabetes and high blood pressure.

3 Machine Learning Models

3.1 Data pre-processing for the Machine Learning Analysis

Let's start subdividing the dataset into the training and test dataset:

```
# STEEP 4: MACHINE LEARNING MODELS

### Preparing the DATA for ML analysis
set.seed(999)
trainIndex = sample(1:length(heartfailure.dat$DEATH_EVENT),0.8*length(heartfailure.dat$DEATH_EVENT))
train = heartfailure.dat[trainIndex,-c(14:16)]
test.x = heartfailure.dat[-trainIndex,-c(13:16)]
test.y = heartfailure.dat[-trainIndex,13]
```

The test set will be populated by the 20% of the patient of the initial dataset; as a consequence the train set will be composed by the remaninign 80% of the patients.

3.2 Logistic Regression

The first machine learning model used to build the prediction system which can predict survival of patients with heart failure is `<Logistic Regression>`.

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

Generally, logistic regression means binary logistic regression having binary target variables, but there can be two more categories of target variables that can be predicted by it. Based on those number of categories, Logistic regression can be divided into different types such as Binomial, Multinomial and Ordinal.

In this situation, the dependent variable will have only two possible types either 1 and 0.

The confusion matrix togethere with the results of the logistic regression model are printed below:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 34  9
##          1  4 13
##
##                Accuracy : 0.783
##                  95% CI : (0.658, 0.879)
##     No Information Rate : 0.633
##     P-Value [Acc > NIR] : 0.00944
##
##                   Kappa : 0.51
##
##  Mcnemar's Test P-Value : 0.26726
##
##             Sensitivity : 0.895
##             Specificity : 0.591
##          Pos Pred Value : 0.791
##          Neg Pred Value : 0.765
##              Prevalence : 0.633
##          Detection Rate : 0.567
##    Detection Prevalence : 0.717
##       Balanced Accuracy : 0.743
##
##        'Positive' Class : 0
##
```

```
## [1] "Accuracy of Logistic Regression is 0.78"
```

Method	Accuracy
Logistic Regression	0.78

3.3 Random Forest

The second machine learning model used is the `<Random Forest>`.

Random Forest is a powerful and versatile supervised machine learning algorithm that grows and combines multiple decision trees to create a "forest." It can be used for both classification and regression problems in R and Python.

A decision tree is another type of algorithm used to classify data. In very simple terms, it can be think as s flowchart that draws a clear pathway to a decision or outcome; it starts at a single point and then branches off into two or more directions, with each branch of the decision tree offering different possible outcomes.

The mtry value has been set to the square root of the number of columns in the dataframe as suggested by the theory.

```
## Random Forest
##
## 239 samples
##  12 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 215, 215, 214, 216, 215, 215, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.8588    0.6598
##
## Tuning parameter 'mtry' was held constant at a value of 4
```

3.3.1 Tuning using Caret Package

The parameters that affect the random forest model are several and it is important to try to tune them in order to avoid overfitting or others possible mistakes in the model output.

The key parameters that most likely have the biggest effect on the final accuracy are the `<mtry>` and the `<ntree>`.

Direct from the help page for the randomForest() function in R:

- `<mtry>`: Number of variables randomly sampled as candidates at each split.
- `<ntree>`: Number of trees to grow.

The caret package in R provides an excellent facility to tune machine learning algorithm parameters.

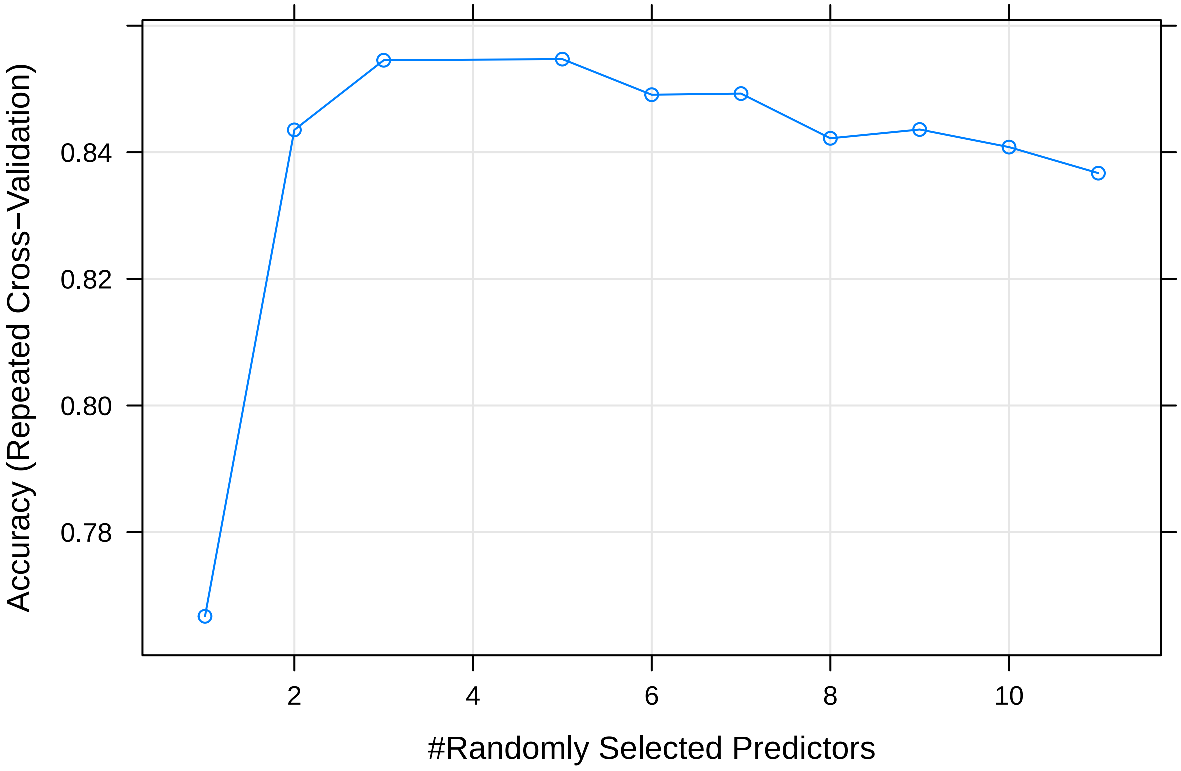
However only mtry parameter is available in caret for tuning. The reason is its effect on the final accuracy and that it must be found empirically for a dataset.

The ntree parameter is different in that it can be as large as you like, and continues to increases the accuracy up to some point. It is less difficult or critical to tune and could be limited more by compute time available more than anything.

3.3.1.1 Random Forest model with random search for mtry One search strategy that can be used is to try random values within a range.

This can be good if there is uncertnaities of what the value might be and it is wanted to overcome any biases it might be for setting the parameter.

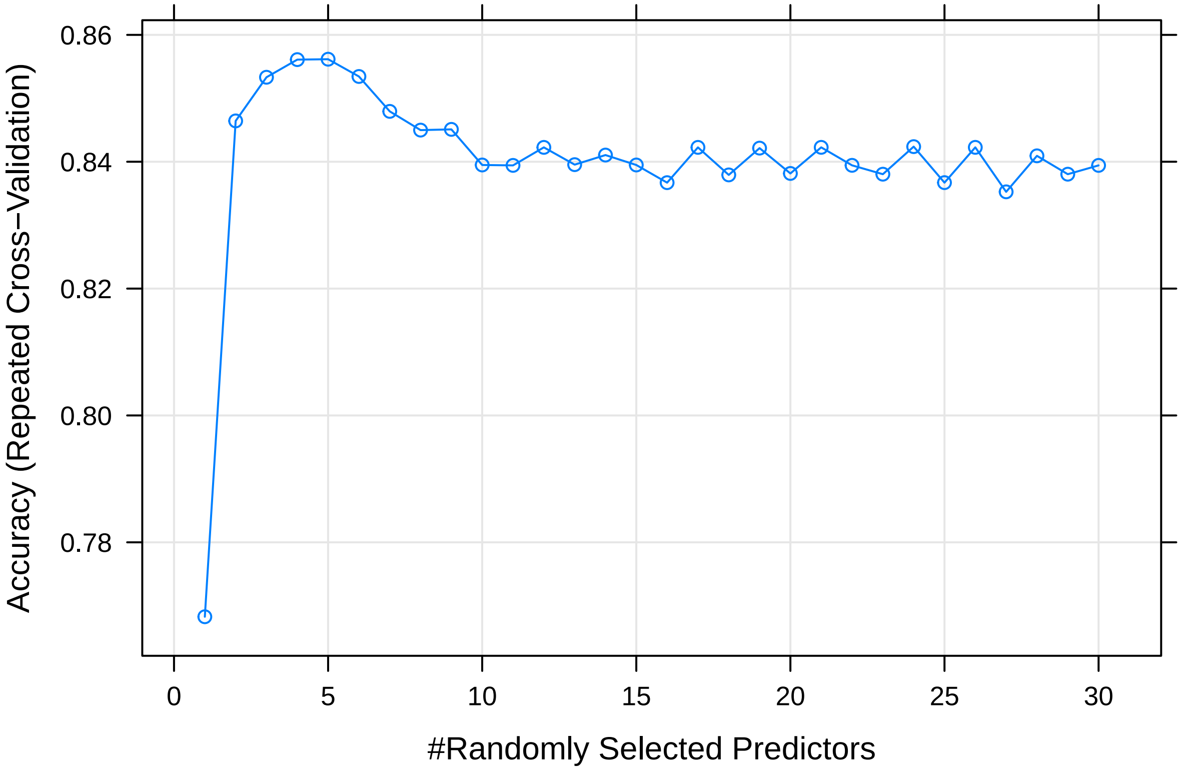
```
## Random Forest
##
## 239 samples
##  12 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 215, 215, 214, 216, 215, 215, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy  Kappa
##    1    0.7667    0.3252
##    2    0.8435    0.6110
##    3    0.8545    0.6466
##    5    0.8547    0.6485
##    6    0.8491    0.6397
##    7    0.8493    0.6408
##    8    0.8422    0.6246
##    9    0.8436    0.6274
##   10    0.8408    0.6222
##   11    0.8367    0.6129
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 5.
```



3.3.1.2 Random Forest model with grid search for mtry Another search is to define a grid of algorithm parameters to try.

Each axis of the grid is an algorithm parameter, and points in the grid are specific combinations of parameters. Since in this case only one parameter has been tuned, the grid search is a linear search through a vector of candidate values.

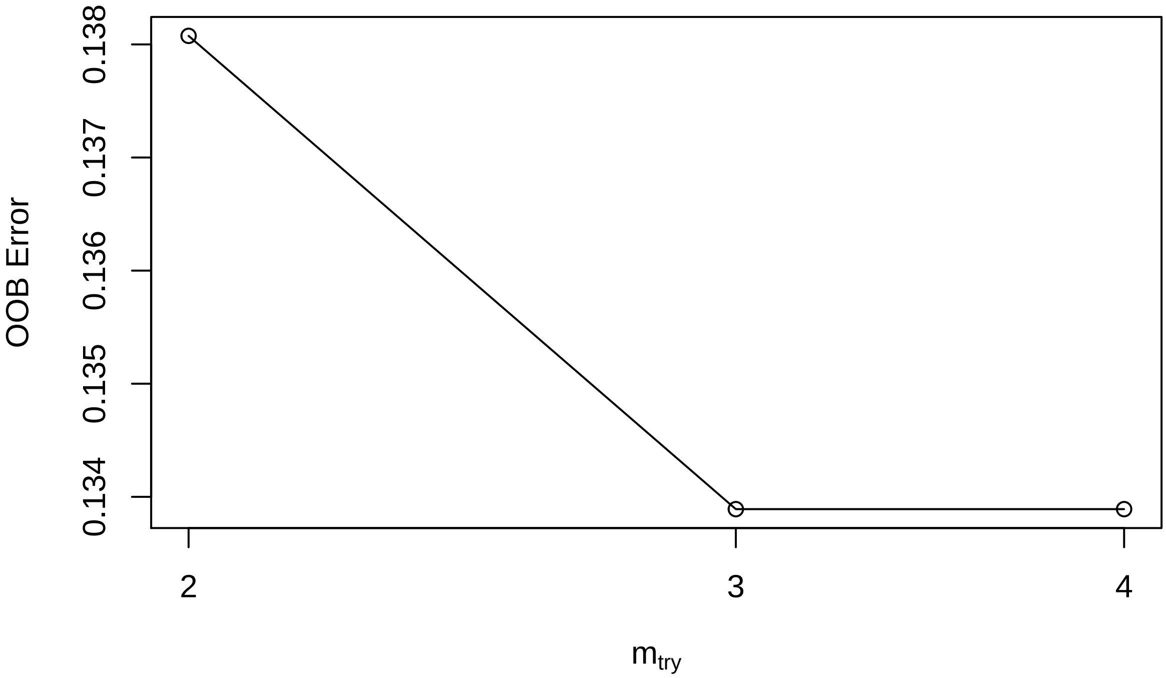
```
## Random Forest
##
## 239 samples
##  12 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 215, 215, 214, 216, 215, 215, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy  Kappa
##    1    0.7683    0.3269
##    2    0.8464    0.6230
##    3    0.8533    0.6416
##    4    0.8561    0.6529
##    5    0.8562    0.6550
##    6    0.8534    0.6508
##    7    0.8479    0.6345
##    8    0.8450    0.6306
##    9    0.8451    0.6293
##   10    0.8395    0.6194
##   11    0.8394    0.6187
##   12    0.8423    0.6246
##   13    0.8395    0.6206
##   14    0.8411    0.6240
##   15    0.8395    0.6194
##   16    0.8367    0.6135
##   17    0.8423    0.6241
##   18    0.8379    0.6162
##   19    0.8421    0.6249
##   20    0.8382    0.6164
##   21    0.8423    0.6251
##   22    0.8394    0.6181
##   23    0.8380    0.6161
##   24    0.8424    0.6250
##   25    0.8367    0.6142
##   26    0.8423    0.6253
##   27    0.8353    0.6108
##   28    0.8409    0.6217
##   29    0.8380    0.6165
##   30    0.8394    0.6184
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 5.
```



3.3.1.3 Random Forest model with auto-tuning of parameters The random forest model has been also tuned with the caret fucntion tuneRF.

This function, starting with the default value of mtry, search for the optimal value (with respect to Out-of-Bag error estimate) of mtry for randomForest.

```
## mtry = 3  OOB error = 13.39%
## Searching left ...
## mtry = 2     OOB error = 13.81%
## -0.03125 1e-05
## Searching right ...
## mtry = 4     OOB error = 13.39%
## 0 1e-05
```



```
##       mtry OOBError
## 2.OOB    2   0.1381
## 3.OOB    3   0.1339
## 4.OOB    4   0.1339
```

3.3.2 Tuning results

In the table below are reporte the value of mtry suggested by the tuning models.

Method	mtry
Random Search	5
Grid Search	5
TuneRF	4

3.3.3 Random Forest Model's results

For the sake of completeness, the results relating to the Random Forest models built on the 4 different tuning methods are presented below.

The improvement in the accuracy of the model compared to the Logistic Regression model is evident and significant.

```
## [1] "Accuracy of Random Forest with default parameters is 0.82"
```

```
## [1] "Accuracy of Random Forest with random parameters is 0.8"
```

```
## [1] "Accuracy of Random Forest with grid-tuned parameters is 0.8"
```

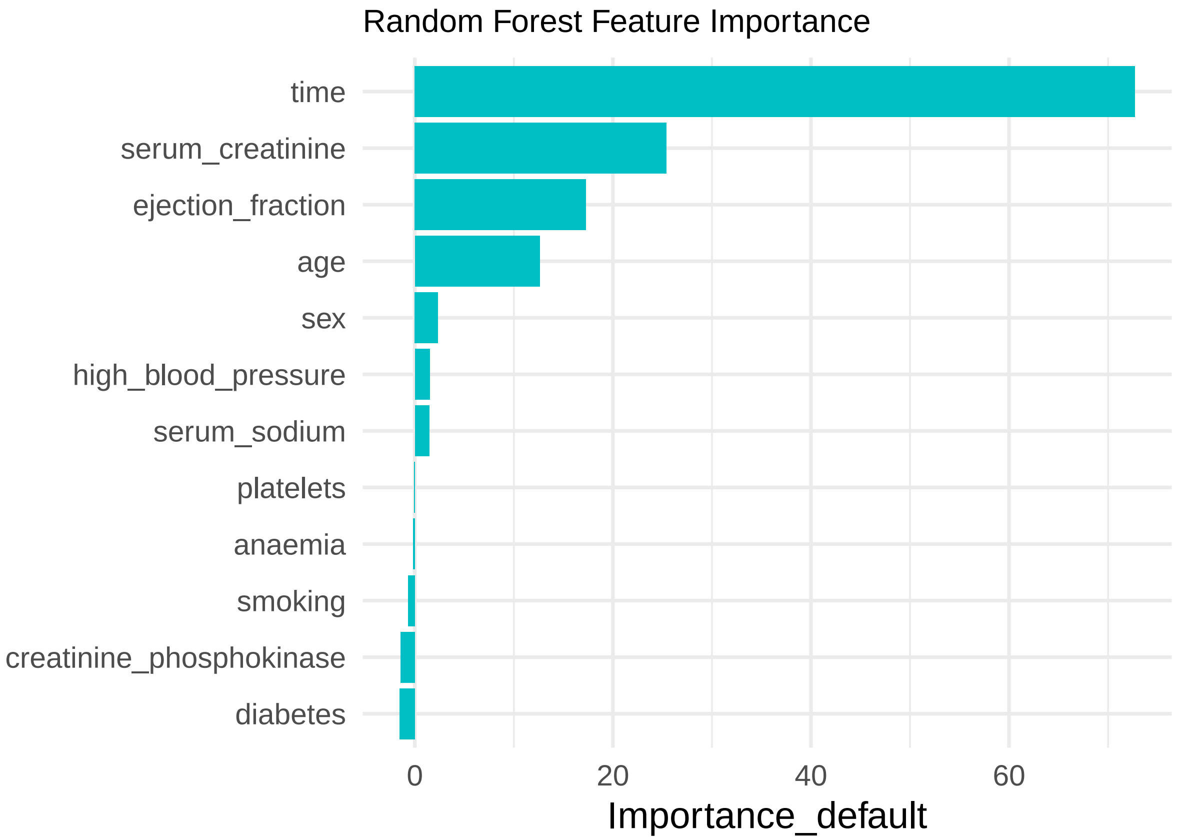
```
## [1] "Accuracy of Random Forest with tuned parameters is 0.8"
```

Method	Accuracy
Logistic Regression	0.78
Random Forest - Theroretical Parameters	0.82
Random Forest - mtry Random Tuning	0.80
Random Forest - mtry Grid Tuning	0.80
Random Forest - TuneRF Tuning	0.80

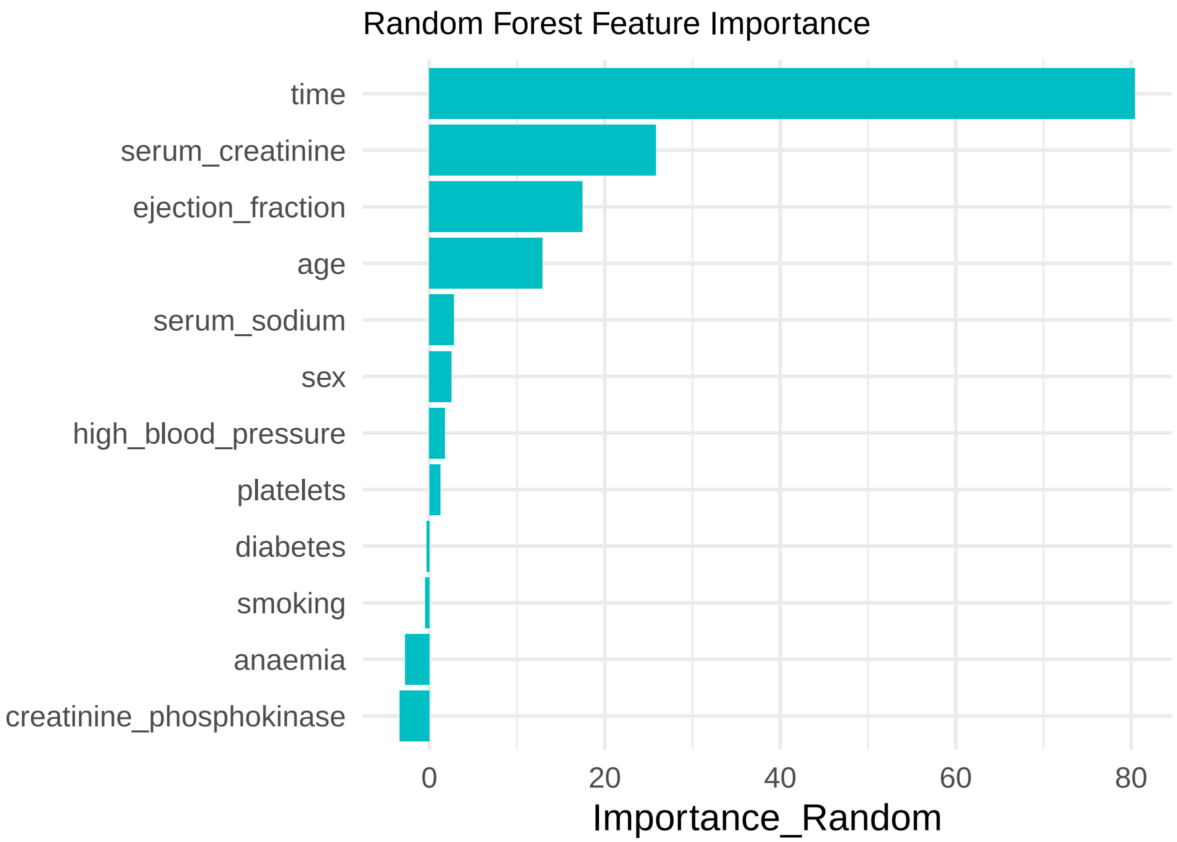
Below are shown the plots highlighting the feature importance for every Random Forest model tried.

It is interesting that there are no differences between the 4 models in the factors that have a greater influence on the algorithm while there is some difference between those that have a lesser effect.

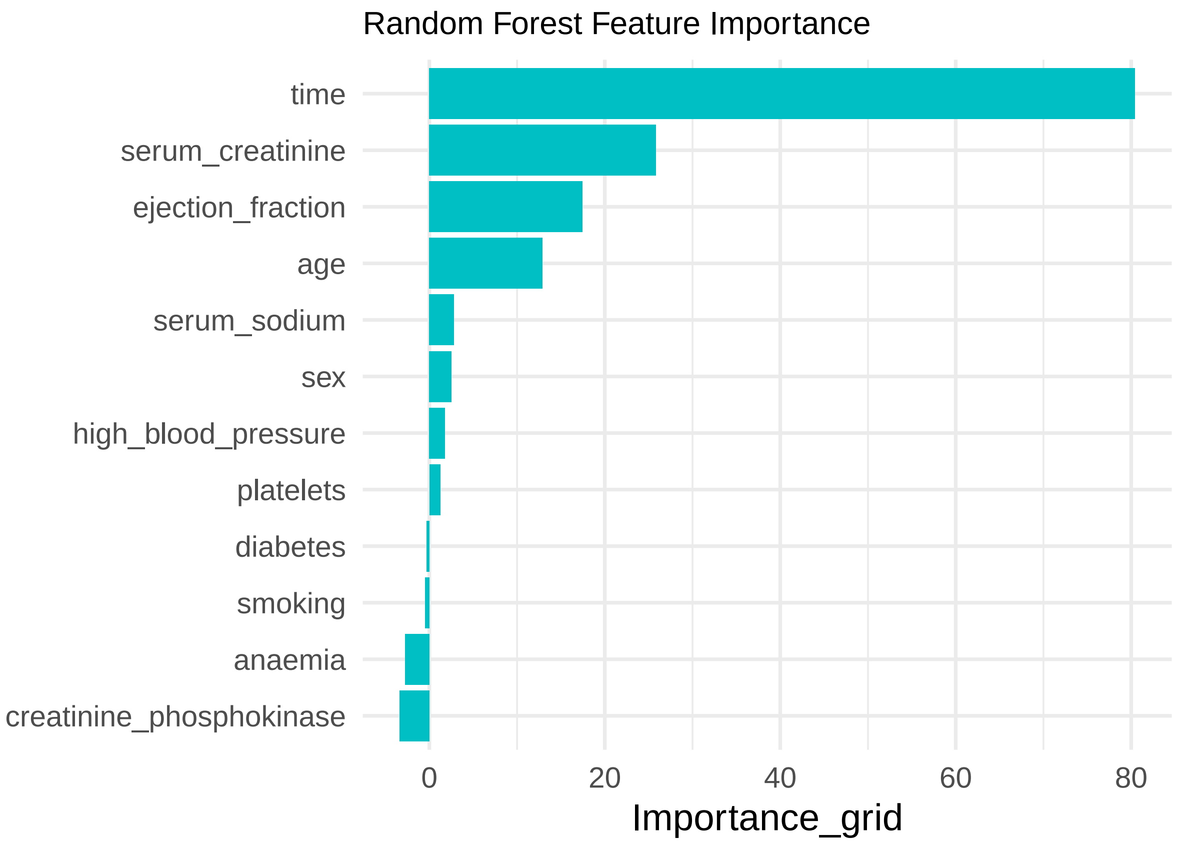
3.3.3.1 Feature Importance in Random Forest model - Theretical Parameters



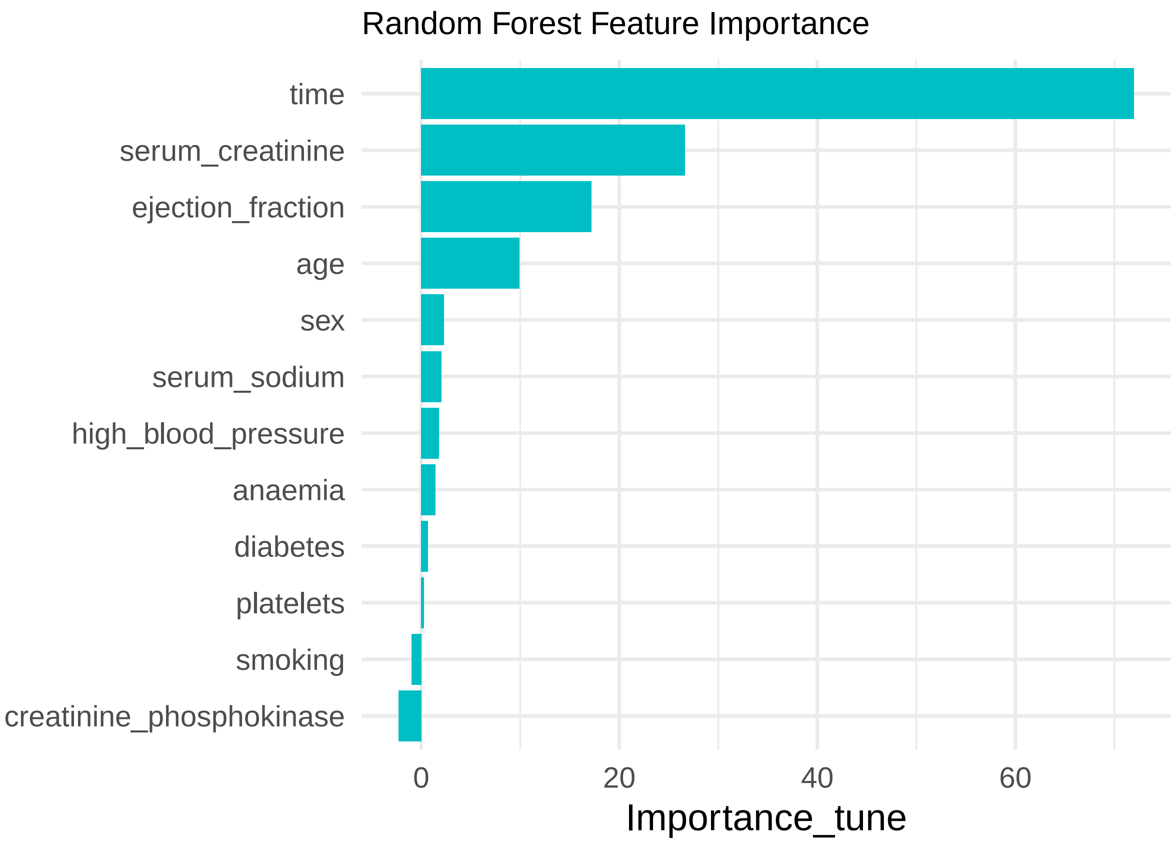
3.3.3.2 Feature Importance in Random Forest model - mtry Random Tuning



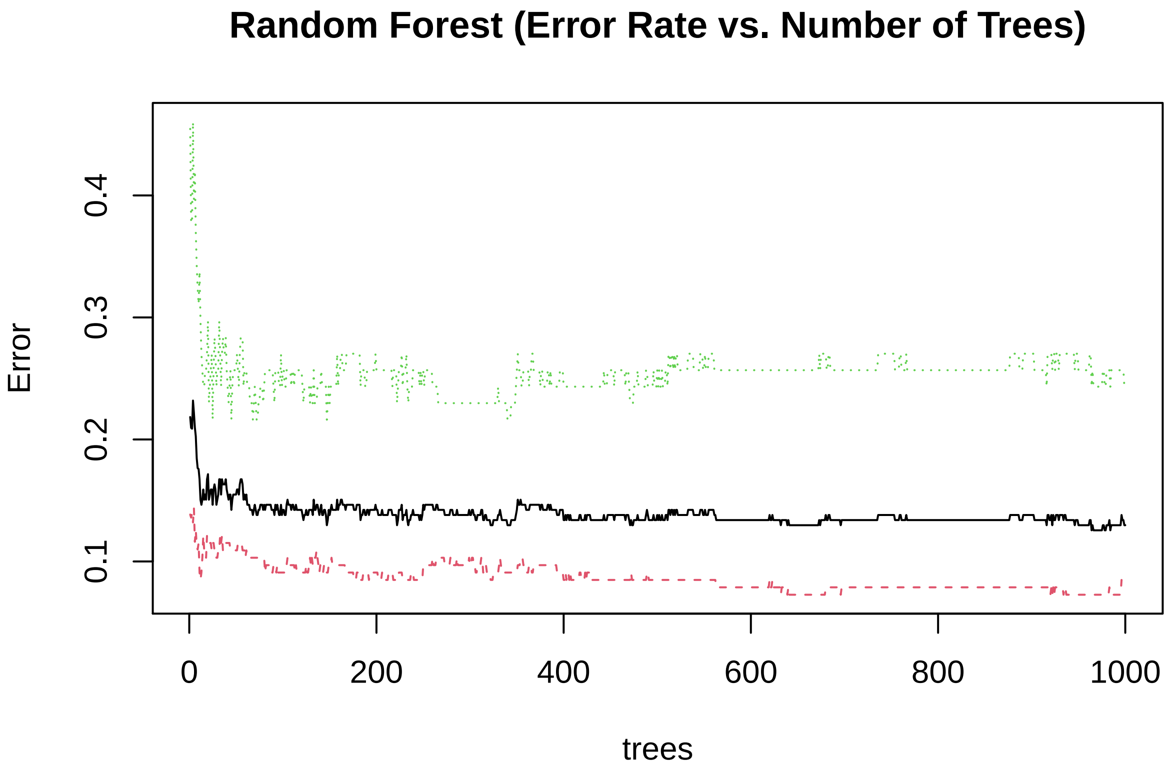
3.3.3.3 Feature Importance in Random Forest model - mtry Grid Tuning

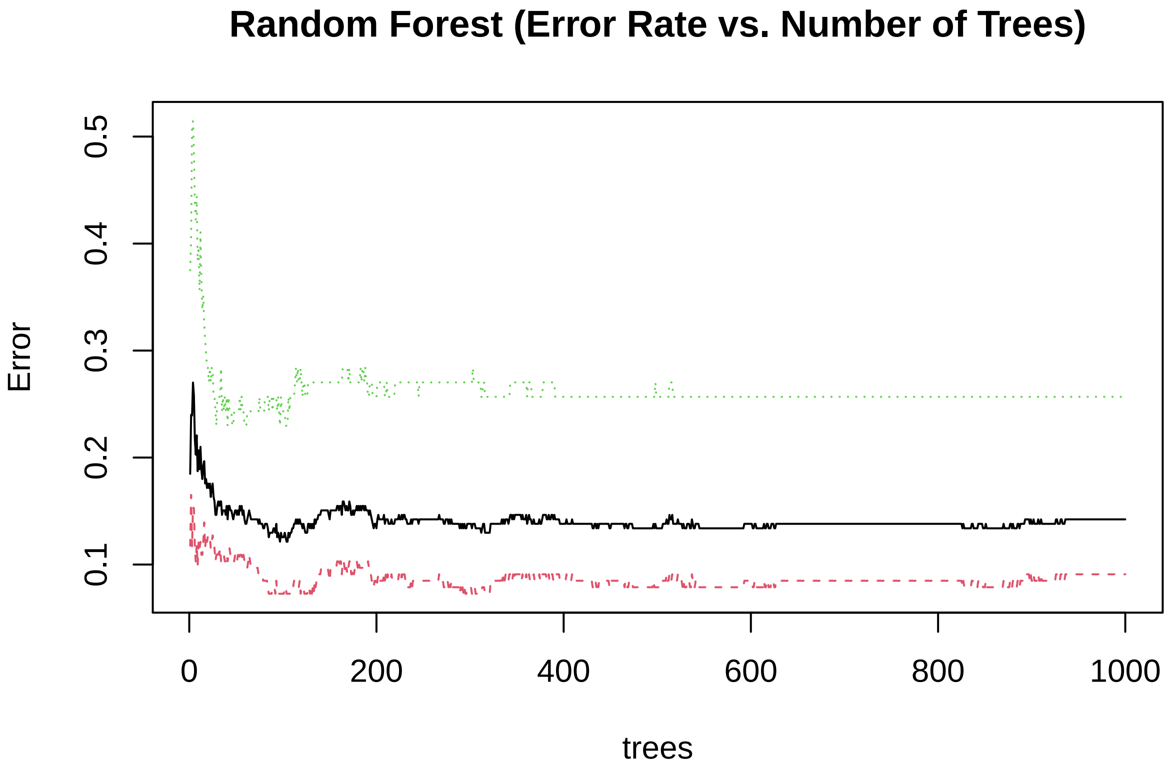


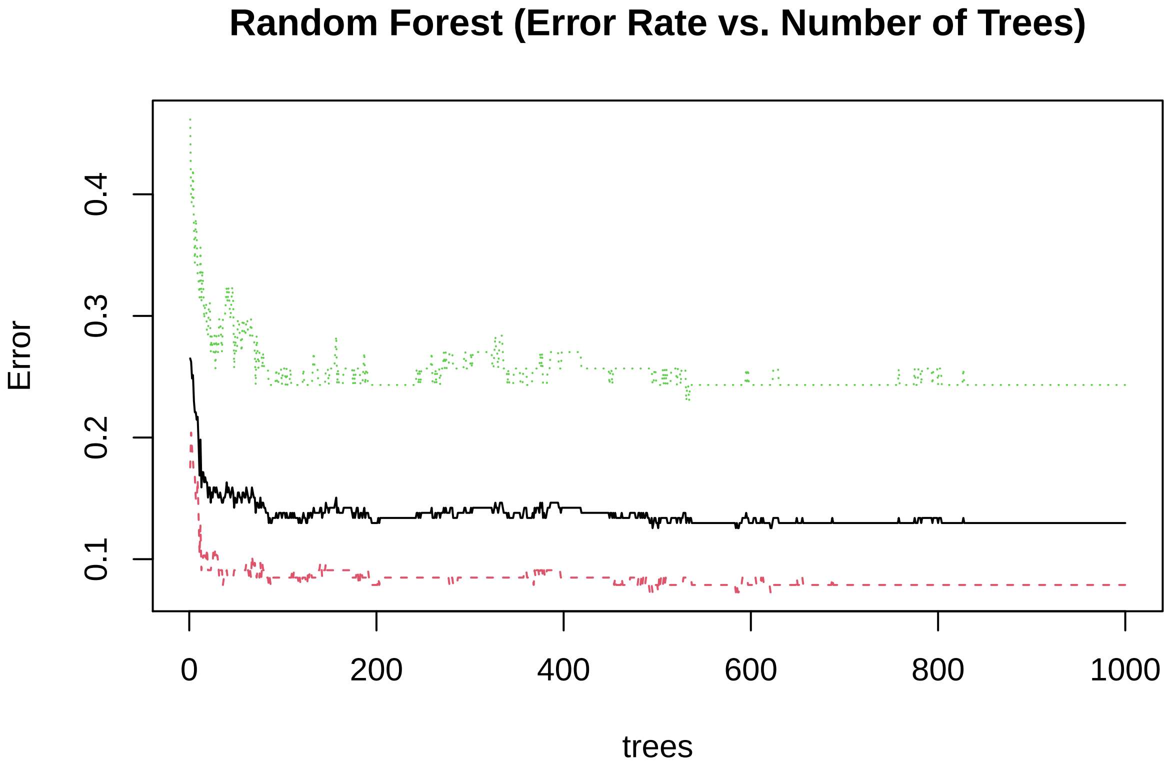
3.3.3.4 Feature Importance in Random Forest model - mtry TuneRF Tuning

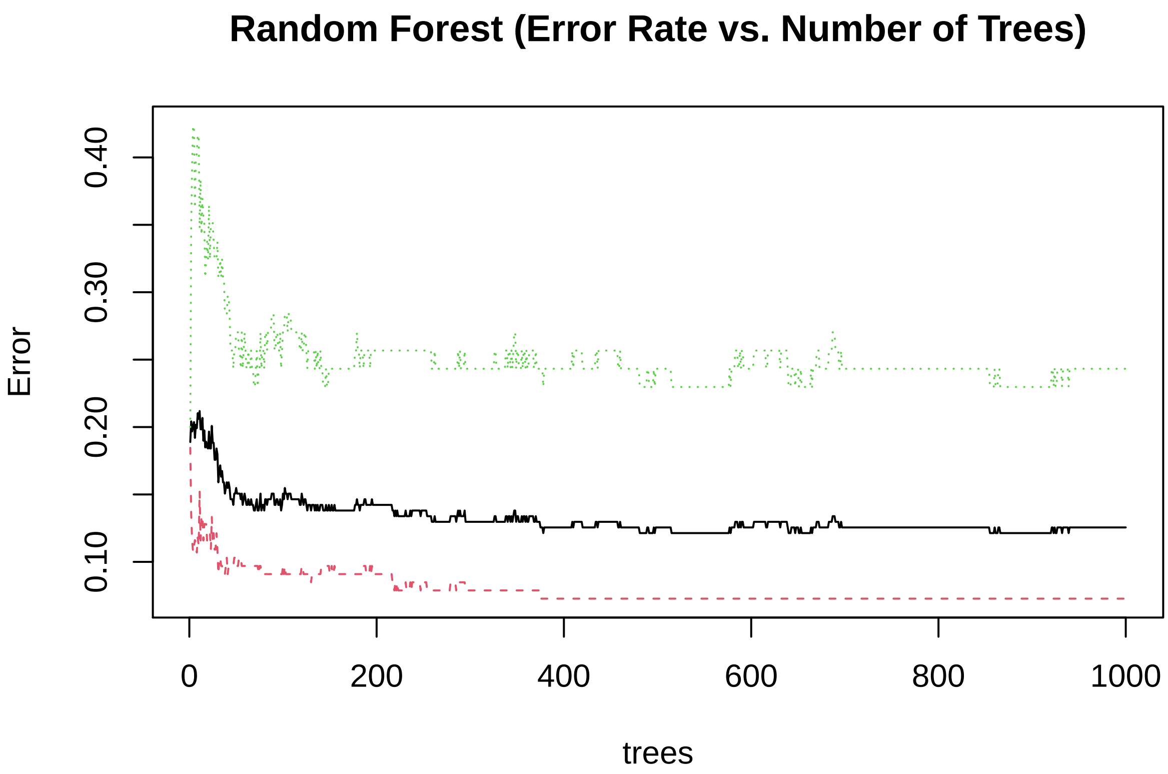


3.3.3.5 Error Rate relation with Number of trees In the plot below are represented the error rate with respect to the number of trees in each Random Forest model developed.




Random Forest (Error Rate vs. Number of Trees)
Error
0.1
0.2
0.3
0.4
0.5
0
200
400
600
800
1000
trees


Random Forest (Error Rate vs. Number of Trees)
Error
0.1
0.2
0.3
0.4
0
200
400
600
800
1000
trees



3.4 XGBoost

The third Machine Learning model implemented to improve more the accuracy of the prediction system is the `<Extreme Gradient Boost - XGB>`.

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements Machine Learning algorithms under the Gradient Boosting framework. It provides a parallel tree boosting to solve many data science problems in a fast and accurate way.

Boosting is an ensemble learning technique to build a strong classifier from several weak classifiers in series. Boosting algorithms play a crucial role in dealing with bias-variance trade-off. Unlike bagging algorithms, which only controls for high variance in a model, boosting controls both the aspects (bias & variance) and is considered to be more effective.

The key parameters which affect the most the results of the XGBoost models are the following:

- `<eta>`: is the learning rate. Step size shrinkage used in update to prevents overfitting. After each boosting step, we can directly get the weights of new features, and eta shrinks the feature weights to make the boosting process more conservative.
- `<subsample>`: is subsample ratio of the training instances. Setting it to 0.5 means that XGBoost would randomly sample half of the training data prior to growing trees. and this will prevent overfitting. Subsampling will occur once in every boosting iteration.
- `<max_depth>`: is the maximum depth of a tree. Increasing this value will make the model more complex and more likely to overfit.
- `<colsample_bytree>`: is the subsample ratio of columns when constructing each tree. Subsampling occurs once for every tree constructed.

To achieve better accuracy these 4 key parameters were tuned by selecting the combination that led to the best result.

The best combination of these parameters was found among the following values:

Parameter	Initial_Value	Final_Value	Increment
Eta	0.01	0.1	0.01
SubSample	0.50	1.0	0.10
Max Depth	3.00	10.0	1.00
Colsamle by tree	0.50	1.0	0.10

For completeness, the code of the for loop used to find the best combiantion is shown below

```
for (i in 1:NROW(mat)) {
  params <- list (eta = mat[i,1],
                  max_depth = mat[i,3],
                  min_child_weight = 2,
                  subsample = mat[i,2],
                  colsample_bytree = mat[i,4],
                  objective = "binary:logistic",
                  eval_metric = "rmse")

  xgb.model = xgb.train(params, train.xgb, nround = 10)
  xgb.predict <- predict(xgb.model, test.xgb)
  xgb.predict = ifelse(xgb.predict<0.5,0,1)
  table(test.y$DEATH_EVENT, xgb.predict)

  pred[i,1] = confusionMatrix(as.factor(xgb.predict),test.y$DEATH_EVENT)$overall[1]
  updatedmat <- cbind(mat, pred)
}
```

The tuned parameters which led to the best accuracy for the XGboost Model are the following:

Parameter	Tuned_Value
Eta	0.02
SubSample	0.60
Max Depth	10.00
Colsample by Tree	0.50

The Accuracy reached with the tuned XGBoost model is:

Parameter	Accuracy
Tuned XGBoost	0.8667

4 Conclusions and Results

The results obtained with the Machine Learning model developed are listed below.

Method	Accuracy
Logistic Regression	0.7800
Random Forest - Theroretical Parameters	0.8200
Random Forest - mtry Random Tuning	0.8000
Random Forest - mtry Grid Tuning	0.8000
Random Forest - TuneRF Tuning	0.8000
Tuned XGBoost	0.8667

The Tuned XGBoost model has evidenced an increment in the accuracy of 10% which is a considerable result. It could be interesting to try to change the number of trees in the Random forest algorithm to check if better results can be obtained.

The variable time has a considerable effect on the model; it could be also interesting to repeat the analysis discarding the time variable from the dataset.

As a conclusion I would like to express my opinion on the importance of machine learning applied to medicine.

This rreport and the work on this dataset is just a very small and trivial proof of how machine learning techniques are a support tool for medical activities of fundamental importance.

In my opinion, over time, the use of machine learning algorithms will become crucial in all medical sectors; from diagnostics to therapy to conclude with surgical intervention.