Myocardial infarction complications

Akastia Christo

2022-08-30

Akastia Christo
390250
BFV3
2022-08-30
Michiel Noback

Myocardial Infraction

Analysis of myocardial infraction and using machine learning

Student: Akastia Christo
Bio-informatica
Student number: 390250
Life science en technology
Lecturer: Micheal Noback
Date: 2022-08-30

Contents

1 Abbreviations and symbols

Abbreviations	Symbols
Myocardial Infractions	MI
Coronary Heart Disease	CHD
Electrocardiogram	ECG
Exploratory Data Analysis	EDA
Principal Component Analysis	PCA
True Positive	TP
False Positive	FP
True Negative	TN
False Negative	FN
True Positive Rate	TPR
False Positive Rate	FPR
Receiver Operating Characteristics	ROC
Area Under The Curve	AUC

2 Introduction

Myocardial infraction (MI), is known as heart attacks happens when one or more areas of the heart mucles do not get enough oxygen which can happen when the blood flow of the heart muscle is blocked [1] MI is one of the most challenging problems of the modern medicine. Because the course of disease in patients with MI differs each patient. MI can happen with or without complications that do not affect the long-term prognosis. Approximately half of the patients in the acture and subactute period experience complication that results in worsening of the disease or death. Even an experienced specialist may not be able to predict the development of these complications. Acute MI is associated with high mortality in the first year after it. The incidence of MI remains high in all countries. This is especially true for the urban population of highly developed countries, which is exposed to chronic stress factors, irregular and not always balanced nutrition. [2]

2.1 Research question

The goal for this research is to answer the question, can you predict the complications of the patients after the third day of the admission based on the admission period and patients data using machine learning?

3 Materials & Methods

3.1 Materials

The data set that is used in this research contains information about the myocardial infarction complications database was collected in the Krasnoyarsk Interdistrict Clinical Hospital No20 named after I. S. Berzon (Russia) in 1992-1995. The database contains 1700 records (patients), 111 input features and 12 complications. In the database there contains 7.6% of missing values.

The codebook for this data set can be found as a pdf and is available in the repo. Because the data set is large it is difficult to make a codebook from it.

For the analysis of the myocardial infraction dataset, the software tool RStudio Version 1.4.1717 [3] is used. RStudio is an environment for using statistical computing and graphic, that uses the R programming language Version 4.0.4. In RStudio there are a few libraries that R used throughout this project these are listed in table 2 and the appendix.

Table 2: Libraries

Library	Version	Goal
utils	4.0.4	Reads and writes files in R
dplyr	1.0.7	Provides data manipulation
tidyr	1.1.3	Tool for changing shape of a dataset
kableExtra	1.3.4	Build common complex tables
ggplot2	3.3.5	System for creating graphics
stats	4.0.4	Functions for statistical calculations
pheatmap	1.0.12	Creates heatmaps
ggpubr	0.4.0	Some easy-to-use functions for creating and customizing ‘ggplot2’
ggbiplot	0.55	Biplot for Principal Components
cowplot	1.1.1	Arrange multiple plots into a grid

To analyse and create classifiers for a model that will predict the consequences of myocardial infarction, machine learning is applied. Weka Version 3.85 is the programme used to accomplish this.[4] Weka is opensource software that offers tools for processing data and implementing various machine learning algorithms into practise.

After creating and saving the best model. Java uses the model to build a Java wrapper.[6] Java wrapper is produced using IntelliJ IDEA Version 2021.2.1 software. [5] A development environment for creating Java-based applications, IntelliJ makes use of the JDK.[7] It contains resources for developing and evaluating programmes written in Java programming language that operate on the Java platform.

3.2 Methods

In RStudio the Exploratory Data Analysis (EDA) is first performed using R. This was accomplished via visualizing, transforming, and modeling the data. The myocardial infraction data set's variation, distribution, missing values, and clustering are displayed by creating numerous plots in R using the libraries provided in table 2. The libraries ggplot2, pheatmap, and ggbiplot were used to produce numerous plots, and kableExtra was used to produce tables. After visualizing which data is relevant and which isn't, the data had been cleaned. The cleaned myocardial infraction data is used for machine learning in Weka. To find the best model to categorise the instances, there were multiple algorithms applied and evaluated with 10-fold cross-validation in Weka. This was done with the explorer and experimenter in Weka. The best model is kept, and a Java wrapper is made using this model. The myocardial infraction data set and a new data set with the class in unknown are read by the Java wrapper generated in IntelliJ using Java. The programme can classify new instances that are defined in the data set with unidentified classes with the aid of the model.

Throughout this paper, the term "False-Positive" is used to refer to occurrences that were mistakenly classified as Unknown when, in fact, they are Lethal Cause, while "False-Negative" is used to refer to situations that were incorrectly labelled as Lethal cause.

Code and data for data analysis and cleaning data done with R and Weka is avaible at https://github.com/akastia/Thema09 Code and data for the Java wrapper is available at https://github.com/akastia/Wrapper

4 Results

Here we are going to show you the results of the data set Myocardial Infraction.

```
##   ID AGE SEX INF_ANAM STENOK_AN FK_STENOK IBS_POST IBS_NASL GB SIM_GIPERT
## 1  1  77   1        2         1         1        2       NA  3          0
## 2  2  55   1        1         0         0        0        0  0          0
## 3  3  52   1        0         0         0        2       NA  2          0
## 4  4  68   0        0         0         0        2       NA  2          0
## 5  5  60   1        0         0         0        2       NA  3          0
## 6  6  64   1        0         1         2        1       NA  0          0
##   DLIT_AG ZSN_A nr_11 nr_01 nr_02 nr_03 nr_04 nr_07 nr_08 np_01 np_04 np_05
## 1       7     0     0     0     0     0     0     0     0     0     0     0
## 2       0     0     0     0     0     0     0     0     0     0     0     0
## 3       2     0     0     0     0     0     0     0     0     0     0     0
## 4       3     1     0     0     0     0     0     0     0     0     0     0
## 5       7     0     0     0     0     0     0     0     0     0     0     0
## 6       0     0     0     0     0     0     0     0     0     0     0     0
##   np_07 np_08 np_09 np_10 endocr_01 endocr_02 endocr_03 zab_leg_01 zab_leg_02
## 1     0     0     0     0         0         0         0          0          0
## 2     0     0     0     0         0         0         0          0          0
## 3     0     0     0     0         0         0         0          0          0
## 4     0     0     0     0         0         0         0          1          0
## 5     0     0     0     0         0         0         0          0          0
## 6     0     0     0     0         0         0         0          0          0
##   zab_leg_03 zab_leg_04 zab_leg_06 S_AD_KBRIG D_AD_KBRIG S_AD_ORIT D_AD_ORIT
## 1          0          0          0         NA         NA       180       100
## 2          0          0          0         NA         NA       120        90
## 3          0          0          0        150        100       180       100
## 4          0          0          0         NA         NA       120        70
## 5          0          0          0        190        100       160        90
## 6          0          0          0         NA         NA       140        90
##   O_L_POST K_SH_POST MP_TP_POST SVT_POST GT_POST FIB_G_POST ant_im lat_im
## 1        0         0          0        0       0          0      1      0
## 2        0         0          0        0       0          0      4      1
## 3        0         0          0        0       0          0      4      1
## 4        0         0          0        0       0          0      0      1
## 5        0         0          0        0       0          0      4      1
## 6        0         0          0        0       0          0      1      1
##   inf_im post_im IM_PG_P ritm_ecg_p_01 ritm_ecg_p_02 ritm_ecg_p_04
## 1      0       0       0             0             0             0
## 2      0       0       0             1             0             0
## 3      0       0       0             1             0             0
## 4      1       0       0             1             0             0
## 5      0       0       0             0             0             0
## 6      0       0       0             0             0             0
##   ritm_ecg_p_06 ritm_ecg_p_07 ritm_ecg_p_08 n_r_ecg_p_01 n_r_ecg_p_02
## 1             0             1             0            0            0
## 2             0             0             0            0            0
## 3             0             0             0            0            0
## 4             0             0             0            0            0
## 5             0             1             0            0            0
## 6             0             1             0            0            0
##   n_r_ecg_p_03 n_r_ecg_p_04 n_r_ecg_p_05 n_r_ecg_p_06 n_r_ecg_p_08 n_r_ecg_p_09
## 1            0            0            1            0            0            0
## 2            0            1            0            0            0            0
## 3            1            0            0            0            0            0
## 4            0            0            0            0            0            0
## 5            0            0            0            0            0            0
## 6            0            0            0            0            0            0
##   n_r_ecg_p_10 n_p_ecg_p_01 n_p_ecg_p_03 n_p_ecg_p_04 n_p_ecg_p_05 n_p_ecg_p_06
## 1            0            0            0            0            0            0
## 2            0            0            0            0            0            0
## 3            0            0            0            0            0            0
## 4            0            0            0            0            0            0
## 5            0            0            0            0            0            0
## 6            0            0            0            0            0            0
##   n_p_ecg_p_07 n_p_ecg_p_08 n_p_ecg_p_09 n_p_ecg_p_10 n_p_ecg_p_11 n_p_ecg_p_12
## 1            0            1            0            0            0            0
## 2            0            0            0            0            0            0
## 3            0            0            0            0            0            0
## 4            0            0            0            0            0            0
## 5            0            0            0            0            0            0
## 6            0            0            0            0            0            0
##   fibr_ter_01 fibr_ter_02 fibr_ter_03 fibr_ter_05 fibr_ter_06 fibr_ter_07
## 1           0           0           0           0           0           0
## 2           0           0           0           0           0           0
## 3           0           0           0           0           0           0
## 4           0           0           0           0           0           0
## 5           0           0           0           0           0           0
## 6           0           0           0           0           0           0
##   fibr_ter_08 GIPO_K K_BLOOD GIPER_NA NA_BLOOD ALT_BLOOD AST_BLOOD KFK_BLOOD
## 1           0      0     4.7        0      138        NA        NA        NA
## 2           0      1     3.5        0      132      0.38      0.18        NA
## 3           0      0     4.0        0      132      0.30      0.11        NA
## 4           0      1     3.9        0      146      0.75      0.37        NA
## 5           0      1     3.5        0      132      0.45      0.22        NA
## 6           0     NA      NA       NA       NA      0.45      0.22        NA
##   L_BLOOD ROE TIME_B_S R_AB_1_n R_AB_2_n R_AB_3_n NA_KB NOT_NA_KB LID_KB NITR_S
## 1     8.0  16        4        0        0        1    NA        NA     NA      0
## 2     7.8   3        2        0        0        0     1         0      1      0
## 3    10.8  NA        3        3        0        0     1         1      1      0
## 4      NA  NA        2        0        0        1    NA        NA     NA      0
## 5     8.3  NA        9        0        0        0     0         0      0      0
## 6     7.2   2        2        0        0        0     0         1      0      0
##   NA_R_1_n NA_R_2_n NA_R_3_n NOT_NA_1_n NOT_NA_2_n NOT_NA_3_n LID_S_n
## 1        0        0        0          0          0          0       1
## 2        0        0        0          1          0          0       1
## 3        1        0        0          3          2          2       1
## 4        0        0        0          0          0          0       0
## 5        0        0        0          0          0          0       0
## 6        0        0        0          0          0          0       0
##   B_BLOK_S_n ANT_CA_S_n GEPAR_S_n ASP_S_n TIKL_S_n TRENT_S_n FIBR_PREDS
## 1          0          0         1       1        0         0          0
## 2          0          1         1       1        0         1          0
## 3          1          0         1       1        0         0          0
## 4          0          1         1       1        0         0          0
## 5          0          1         0       1        0         1          0
## 6          1          0         1       1        0         0          1
##   PREDS_TAH JELUD_TAH FIBR_JELUD A_V_BLOK OTEK_LANC RAZRIV DRESSLER ZSN REC_IM
## 1         0         0          0        0         0      0        0   0      0
## 2         0         0          0        0         0      0        0   0      0
## 3         0         0          0        0         0      0        0   0      0
## 4         0         0          0        0         0      0        0   1      0
## 5         0         0          0        0         0      0        0   0      0
## 6         0         0          0        0         0      0        0   0      0
##   P_IM_STEN LET_IS
## 1         0      0
## 2         0      0
## 3         0      0
## 4         0      0
## 5         0      0
## 6         0      0
```

The selected data is the blood pressures of the patients that has arrived. This violin plot shows the ages which has a higher blood pressure whether it is systolic or diastolic.

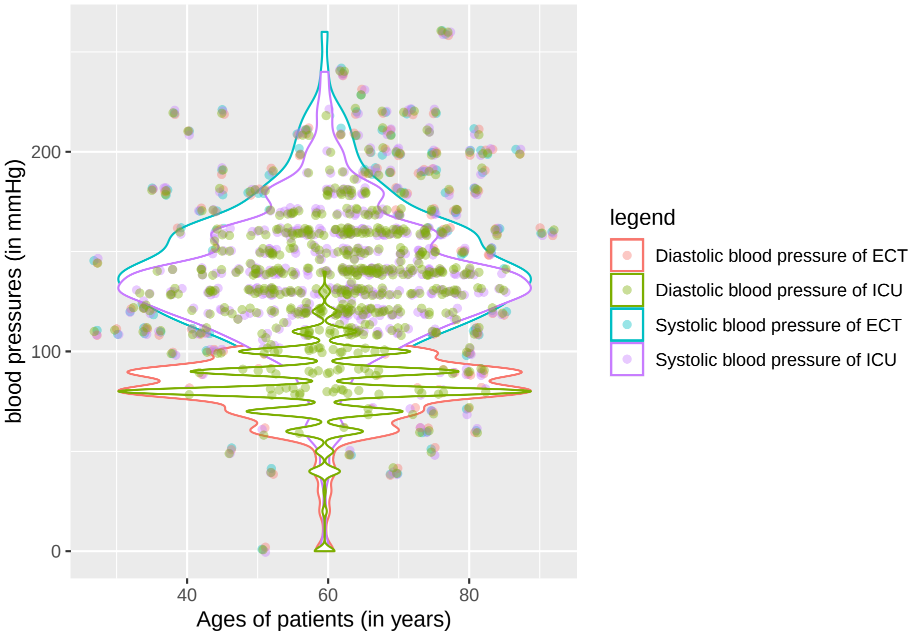


Figure 1: Blood pressures of patients according to the ECT (Emergency cardiology team) and ICU (Intensive care unit)

Figure 1 shows that there are a few outliers in this selected data set. The distribution is expected because most of the patients could have normal blood pressure. Most of the patients have normal diastolic blood pressure in this case, but the patients who measured the systolic blood pressure have a higher mmHg. This actively demonstrates that their heart pushed a lot of blood out. So when they have higher systolic blood pressure for an extended period, it can increase your risk of strokes and heart disease.

Made a bar chart to check if the time elapsed from the beginning of the attack of CHD to the hospital of all the patients relates to the risk of strokes or heart diseases.

This bar chart 2 shows that the the time elapsed is higher at the number 2. Number 2 is the time elapsed of 2-4 hours. Which could be concluded that the time elapsed is mostly around the time of 2 tot 4 hours for most of the patients that has been admitted to the hospital.

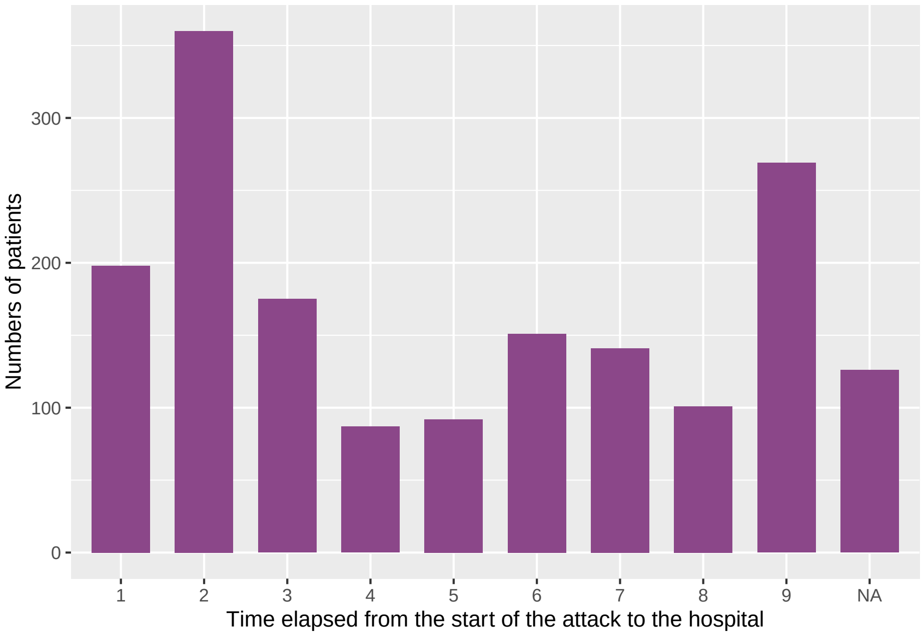


Figure 2: Time elapsed from the beginning of the attack of CHD to the hospital

This figure shows the blood pressure with different sex and ages to see if there any relation between the blood pressure and the sex and ages.

Figure 3 shows that most of the males in comparison to the females were younger than 60. And most of the females where older than 60. It could be concluded that the males have more risk to have a strokes or heart diseased at a younger age.

```
## # A tibble: 6 x 3
##     AGE sinus_with_a_heart_rate heart_rate
##   <int> <chr>                   <fct>
## 1    77 ritm_ecg_p_01           No
## 2    77 ritm_ecg_p_07           Yes
## 3    77 ritm_ecg_p_08           No
## 4    55 ritm_ecg_p_01           Yes
## 5    55 ritm_ecg_p_07           No
## 6    55 ritm_ecg_p_08           No
```

This figure shows the distribution of the myocardial infraction heart rate data set. This violin plot will show the distribution of the myocardial infraction heart rate data set of the patients at the time of admission to the hospital. Made a range of the heart rate with the ages of the patients to see if the patients fell in the category of the heart rate below 60, between 60 and 60 and above 90.

Figure 4 shows that most of the patients didn't have a heart rate below 60 at the time of admission to the hospital. And it shows that most of the patients had a normal heart rhythm at the time of admission, so it could mean they don't have a problem with their heart rate. Patients with a heart rate above 90 should be observed because it could lead to heart diseases if they have an irregular heart rate.

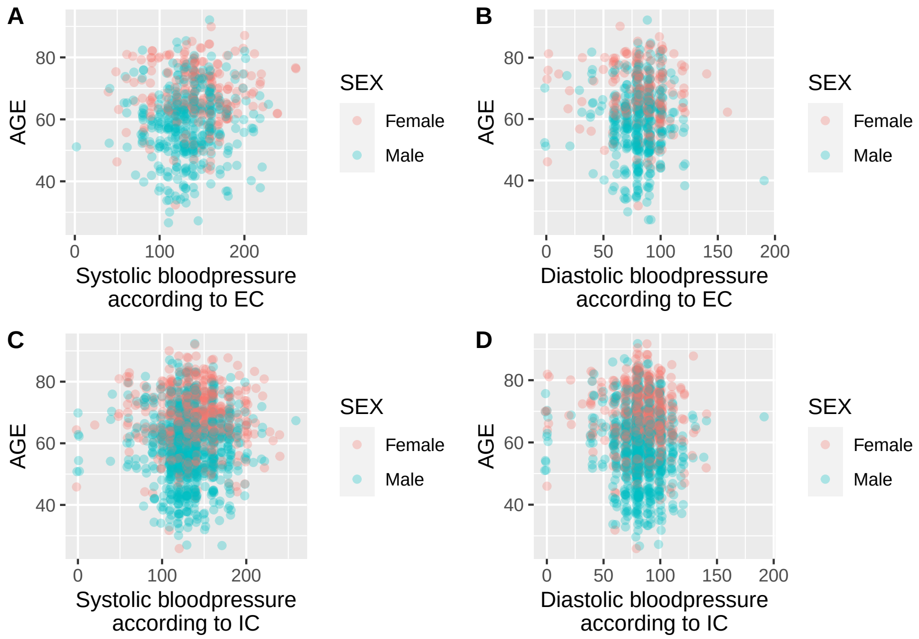


Figure 3: Blood pressure according to the Emergency Cardiology Team (EC) and Intensive Care Unit (IC) with the variables sex and ages

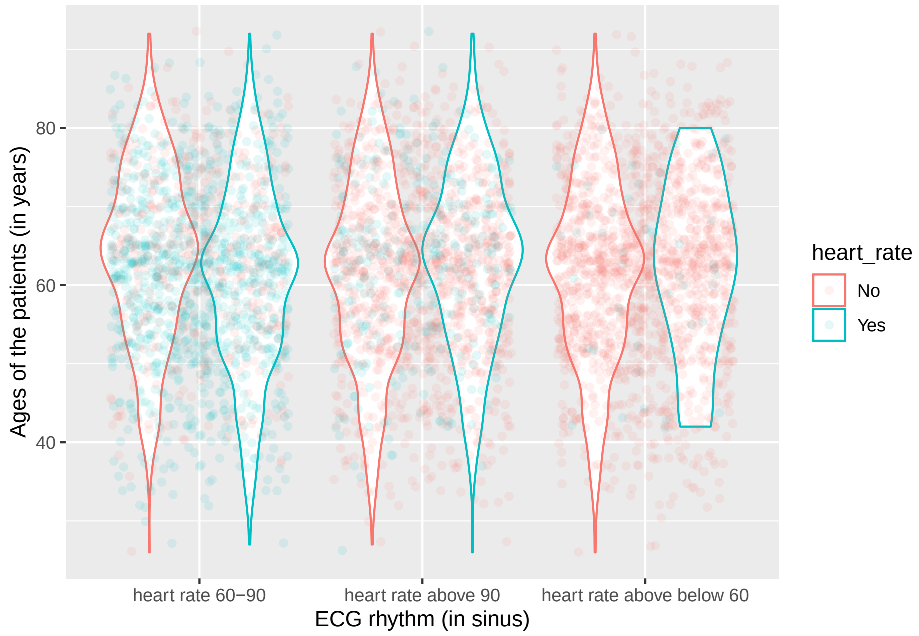


Figure 4: ECG rhythm at the time of admission to the hospital

4.0.1 Clustering of the myocardial infraction data set

Made a PCA to show the clustering of the data set.

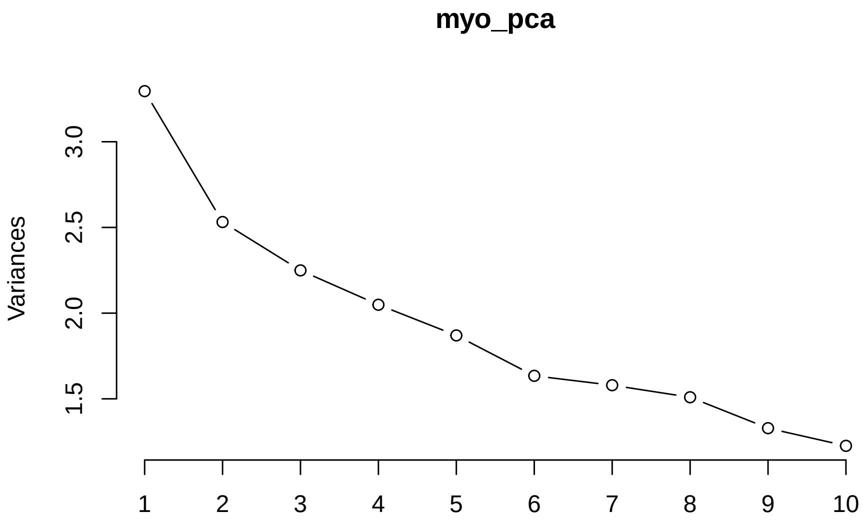


Figure 5: PCA of the classes that were important to see the cluserting of the data were taken and set in a prcomp we could see the variance of our PCA.

This figure 5 shows the variances of the PCA and shows that it is decreasing.

```
## Importance of components:
##                            PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.81525 1.59113 1.49979 1.43141 1.36747 1.27842 1.25678
## Proportion of Variance 0.09985 0.07672 0.06816 0.06209 0.05667 0.04953 0.04786
## Cumulative Proportion  0.09985 0.17657 0.24473 0.30682 0.36349 0.41301 0.46088
##                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     1.22854 1.15272 1.10719 1.06798 1.03221 0.97974 0.95108
## Proportion of Variance 0.04574 0.04027 0.03715 0.03456 0.03229 0.02909 0.02741
## Cumulative Proportion  0.50661 0.54688 0.58403 0.61859 0.65088 0.67996 0.70737
##                           PC15    PC16    PC17    PC18    PC19    PC20    PC21
## Standard deviation     0.95023 0.92301 0.91578 0.90308 0.87326 0.85924 0.77164
## Proportion of Variance 0.02736 0.02582 0.02541 0.02471 0.02311 0.02237 0.01804
## Cumulative Proportion  0.73474 0.76055 0.78597 0.81068 0.83379 0.85616 0.87421
##                           PC22    PC23   PC24    PC25   PC26    PC27    PC28
## Standard deviation     0.75589 0.73764 0.7197 0.69805 0.6473 0.56594 0.55959
## Proportion of Variance 0.01731 0.01649 0.0157 0.01477 0.0127 0.00971 0.00949
## Cumulative Proportion  0.89152 0.90801 0.9237 0.93847 0.9512 0.96087 0.97036
##                           PC29    PC30   PC31    PC32    PC33
## Standard deviation     0.51528 0.46438 0.4298 0.42410 0.36372
## Proportion of Variance 0.00805 0.00653 0.0056 0.00545 0.00401
## Cumulative Proportion  0.97841 0.98494 0.9905 0.99599 1.00000
```

This summary of PCA shows measures of the components. It shows the standard deviation, the proportion of the variance and the cumulative proportion. The proportion of the variance indicates that the data set has a low percentage of variability. This could be seen in the data set. This could be because most of the columns have only two variables, yes or no, in numeric levels, which results in low variability.

Making a PCA plot with the information of the PCA variances.

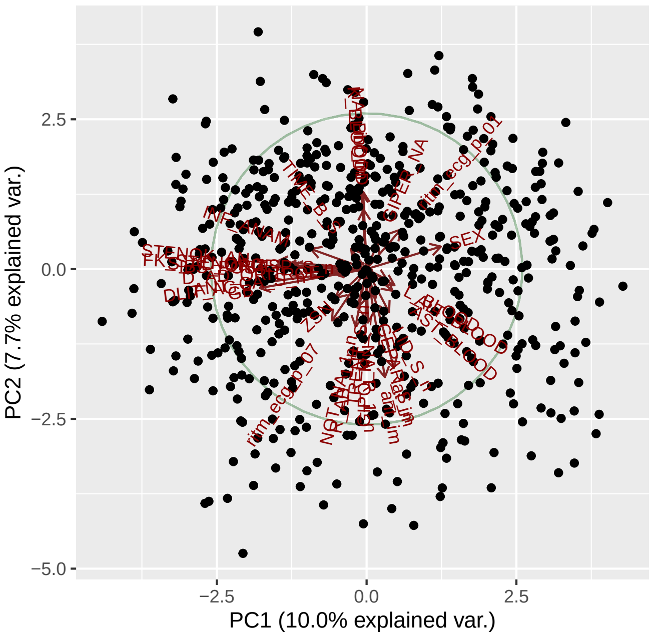


Figure 6: PCA plot of the myocardial of each complications with taking the group klasses of the causes of the myocardial infraction.

Figure 6 shows a scatter plot of the variances of all the component. IT shows a variation of each principal component of the data where the x-axis shows the number of components and the y-axis shows the amount of variations. But 6 PC1 explains only 7.7% variations and PC2 explains only 10% variations. So this actively demonstrates that it doesn't show all of the variances of the myocardial infraction data.

4.0.1.1 Machine learning

The next figure are the results of the machine learning analysis in Weka.[4]

The table 3 shows the results of the 8 classifiers from the machine learning algorithm using weka. The results are shows speed, accuracy, True Positive (TP), False Positive(FP), True Negative(TN), and False Negative(FN). The speed is shown in seconds, the accuracy is shown in percentages. The last four of the table 3 are part of the confusion matrix in Weka. Because the class that was chosen has 8 classes the classes are categorized in unknown and all the lethal causes.

Looking at the table 3 you could see that there are a few who has a higher accuracy compared to the others. OneR has 88.6% accuracy, RandomForest has 86.1% accuracy and SimpleLogistic has 86.2% accuracy. Because the data is about diseases the RandomForest and SimpleLogistic is a good algorithm to use even though the OneR scored better in accuracy.

The table 4 shows the CostSensitiveClassifier for the two algorithms that work the best for the Myocardial Infraction data set, SimpleLogistic and RandomForest. The costMatrix is adapted from 1.0 to 4.0 for each build of the CostSensitiveClassifier. The RandomeForest has for each row a different confusion matrix and accuracy. In contrast, SimpleLogistic has the same accuracy for the first two rows but, like the RandomForest different confusion matrix. The costMatrix cost of the FP is increased in this research. The reason for this is we don't want people who actually have heart diseases to be predicted with the unknown, as in other diseases

Table 3: Classification using cross validation 10-fold

Classifier	Speed	Accuracy	TP	FP	FN	TN
ZeroR	0.00	84.1	1429	271	0	0
OneR	0.11	88.6	1423	148	6	123
NaiveBayes	0.05	74.6	1195	85	234	189
SimpleLogistic	1.03	86.2	1424	235	5	49
SMO	1.18	85.5	1417	221	12	49
Ibk	0.00	80.6	1352	212	77	59
J48	0.59	80.6	1408	211	21	60
RandomForest	1.30	86.1	1428	236	1	35

Table 4: CostSensitiveClassifier with the two algorithms

Classifier	Accuracy	costMatrix	TP	FP	FN	TN
SimpleLogistic	86.2	1	1424	223	5	48
SimpleLogistic	86.2	2	1420	216	9	55
SimpleLogistic	85.6	3	1407	211	19	60
SimpleLogistic	82.4	4	1349	179	80	92
RandomForest	86.1	1	1428	236	1	35
RandomForest	86.5	2	1421	203	8	68
RandomForest	84.6	3	1358	129	71	142
RandomForest	80.3	4	1270	88	159	183

or doesn't have any diseases. In the confusion matrix, the FN is increased so that that FP decreases. The results that we want to keep with this data set are the lethal causes, the TN, and the true positives, which is the unknown. This could be patients that didn't have any lethal reason in the aspect of heart diseases. And the results of the patients are not related to heart diseases. But it can also be that those patients need more research.

In table 4 it shows that the RandomForest with the cost adapted to 2.0 the accuracy increases by approximately 0.4, but when the cost is increased to 3.0 and 4.0, the accuracy decreases. But with the algorithm SimpleLogistic, the accuracy is the same for the cost with 1.0 and 2.0, but like the RandomForest, the accuracy decreases.

4.0.1.2 ROC curve analysis

The following section visualizes Receiver Operating Characteristics (ROC) curves for both RandomForest and SimpleLogistic ROC curve is a graph showing the performance of a classification model at all threshold settings. The ROC curve is created by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR), where TPR is on the y-axis and FDR is on the x-axis.

ROC curve for the classifier RandomForest and SimpleLogistic with the curve of all the threshold setting with the class attribute cardiogenic shock, because the cost matrix shows that the true negatives are mostly the cardiogenic shock.

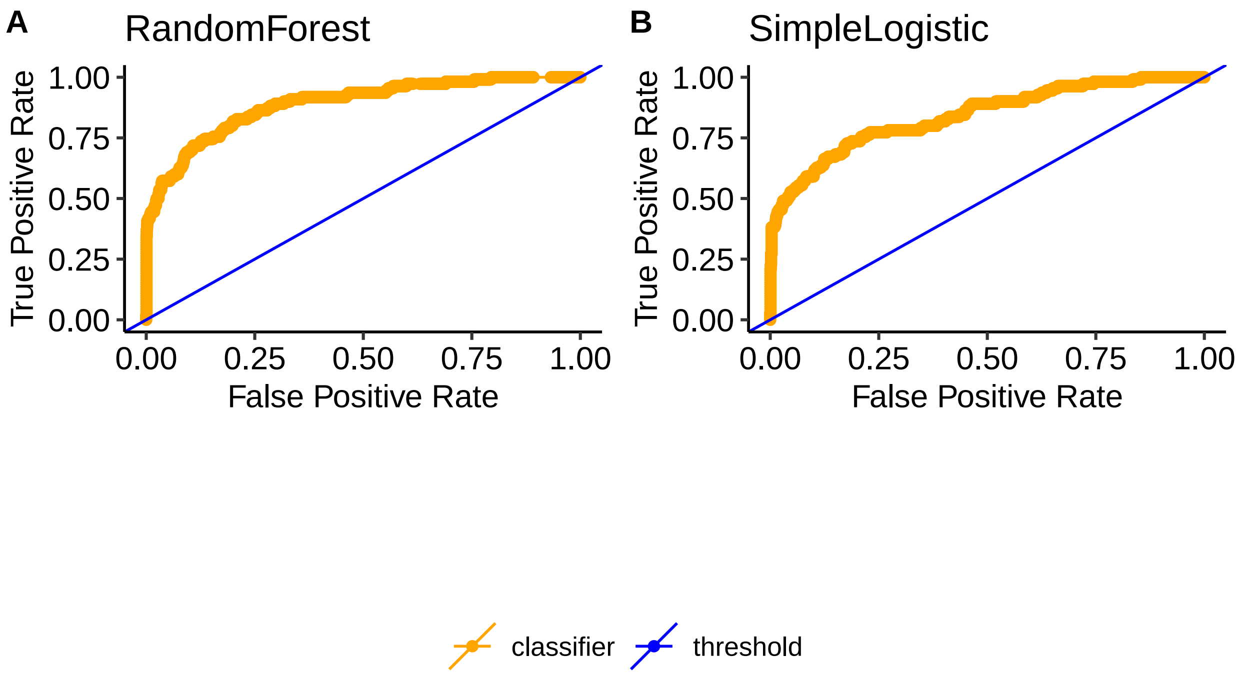
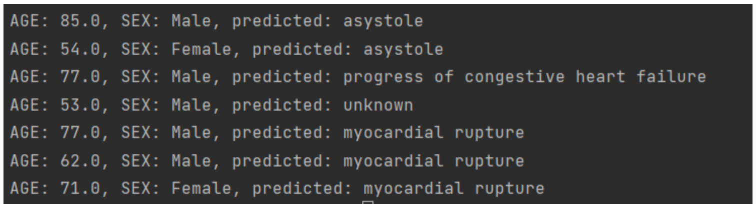


Figure 7: ROC curve for the classifiers RandomForest and SimpleLogistic at all threshold setting with the attribute cardiogenic shock

In figure 7 there are two figures with a ROC curve. The first one figure 7A shows the ROC curve from the algorithm RandomForest and the second figure 7B shows the ROC curve from the algorithm SimpleLogistic. The TPR is plotted against the FPR at various thresholds, forming a line when the dots joins. The area under the ROC curve shown in orange is called the Area Under The Curve (AUC) curve. The blue threshold line shown in the figures is straight, the predicted observation probability. Figure 7A line classifier has dots that are connected, and between the dots at the end, the dots are connected. The line has almost a line of 90 degrees and a straight angle. This shows that the curve is ideal because the two curves of TP and TN almost do not overlap. The reason for it is that it is perfectly able to distinguish between positive class and negative class. If the two curves overlap, the type 1 and type 2 errors are introduced, and the model runs in a curve or a straight line like the threshold line. The ROC curve of SimpleLogistic is shown in figure 7B, and the TPR is plotted against the FPR. Both threshold lines of the ROC curve RandomForest and SimpleLogistic are the same. The most significant difference between the two figures is that the ROC curve of the RandomForest line compared to the ROC curve of SimpleLogistic is slightly straighter. This means that the ROC curve RandomForest figure 7A has almost an ideal situation because there is virtually no overlap. This the ROC curve of RandomForest is slightly better than SimpleLogistic. In Weka, the SimpleLogistic has the AUC value of 0.842, and the AUC value of RandomForest is 0.890. This will be indicated that the ROC curve of RandomForest is better than SimpleLogistic because the AUC of RandomForest is closer to one which categories as perfect.



```
AGE: 85.0, SEX: Male, predicted: asystole
AGE: 54.0, SEX: Female, predicted: asystole
AGE: 77.0, SEX: Male, predicted: progress of congestive heart failure
AGE: 53.0, SEX: Male, predicted: unknown
AGE: 77.0, SEX: Male, predicted: myocardial rupture
AGE: 62.0, SEX: Male, predicted: myocardial rupture
AGE: 71.0, SEX: Female, predicted: myocardial rupture
```

Figure 8: Result of the classification with the Java wrapper.

The Java wrapper is modeled after the best algorithm, RandomForest. The section on materials and methods includes a link to the code. The output of the program is seen in image 8. As shown in the image the ID is not given, because it didn't add any value for the results. So the instances shows the AGE and SEX of the patient. There are 6 instances that predicted a lethal cause and 1 instance unknown. As you can see in the image 8 the 6 lethal causes are asystole, progress of congestive heart failure and myocardial rupture. And out of the 7 instances only 2 were female and the other 5 instances were male.

5 Discussion & conclusion

5.1 Discussion

In a quick overview reveals that the database contains 1700 records (patients), 111 input features and 12 complications. In the database there contains 7.6% of missing values. The values have variations between properties. Which can be seen when comparing the results of the figure 1 until figure 6. The figures demonstrate that the values are required, although there are numerous input characteristics that provided the same information or that the values were not required for this data set. As a result, this suggested that certain features were removed from the dataset.

In general is the data of myocardial infraction a good quality. This is because all of the values associated with a property have almost same variation and distribution per property. Eight different classification methods are used on the myocardial infraction data as shown in table 3. Here, it can be seen that Naive Bayes has the lowest accuracy. This is due to the Naive Bayes algorithm being less accurate because it makes the assumption that all features are independent, which is not true in real life. Looking at the table 3 you could see that there are a few who has a higher accuracy compared to the others. OneR has 88.6% accuracy, RandomForest has 86.1% accuracy and SimpleLogistic has 86.2% accuracy. Because the data is about diseases the RandomForest and SimpleLogistic is a good algorithm to use even though the OneR scored better in accuracy. The reason for this is that they have a good accuracy and for this data set a tree classifiers are a good algorithm to make a decision as if the patient is at risk. And SimpleLogistic is a good classifier to build a linear logistic regression model, so with this model it can help make a best fitting logistic model to use for deciding which lethal cause a patient could have. RandomForest has the best value for FP, because we do not want people who are actually classified as a lethal cause of myocardial infraction to be predicted with the unknown, as in other diseases or doesn't have any diseases. These algorithms are used with the CostSensitiveClassifier with different values for the costMatrix as shown in table 4. In table 4 it shows that the RandomForest with the cost adapted to 2.0 the accuracy increases by approximately 0.4, but when the cost is increased to 3.0 and 4.0, the accuracy decreases. But with the algorithm SimpleLogistic, the accuracy is the same for the cost with 1.0 and 2.0, but like the RandomForest, the accuracy decreases. The ROC curve from these two algorithms is shown in figure 7. These curves are almost the same, but RandomForest has more dots and the curve runs more at a 90 degrees angle which shows a better situation. Thus RandomForest is the best algorithm.

The classification with machine learning doe with the Java wrapper as seen in image 8 classifies the instances based on the properties that have been given. With the information of the patients as blood pressures, ECG rhythms and admission periods etc. There are a lot of values that decide if the patient classification is unknown or a lethal cause.

5.2 conclusion

As discussed there were a lot of properties removed, because of the same results and not adding any information to the data. So it lowered the variation and distrubtion of the data. From the machine learning done with Weka had RandomForest the best model with highest accuracy. This model is used for the Java wrapper which can classify new instances as seen in image 8.

The goal for this research was to answer the question, can you predict the complications of the patients after the third day of the admission based on the admission period and patients data using machine learning? As seen in image 8 that is predicts the myocardial infraction reliably be predicted with the given information of the patient. It can forecast the class, but how dependable it is must be proven. That much depends on if there are further combinations or additional properties that might have a more significant impact on the class.

Further research is RandomForest good an reliable to use for doctors instead of J48 the most widely used machine learning algorithm. The data set had a lot of input features with 12 complications. Which was hard to find what to use to predict. Additionally, there was not much information available regarding the original data and subject. This could be due to the fact that the data only dates back to 1992–1995, however the data is quite organised and clear, so there was enough of patient information.

5.2.1 Project proposl for minor

Application Design

The goal is to create an application that classifies new instances of the myocardial infraction reliably and fast. The classification will be done with an accurate model. This application should have an outcome of the class which is predicted and which of the patients have higher risk based of the class prediction. So it if unknown it should give an overview of the features that have probably influence on the class prediction. And for lethal cause it should give values that might indicate that it is a lethal cause, and categorise the indication which one of the lethal cause it is and the probability of the risk, so it could be decided if the patient have to be admitted immediately. The technology where this application can be used is on a desktop, because companies, hospitals and clinics mainly use computers to keep track of all their patients. This application targets hospitals because myocardial infraction is a disease and this should give the ability to help doctors with the classification if the information that is given is a lethal cause of myocardial or unknown. So that the doctors could find in time if the patients have another disease or a lethal cause of myocardial. The application should give after the input, if it is unknown or if it is a lethal cause of myocardial infraction. If it is unknown, it should give the features if it is beginning of a lethal cause or it is another disease that looks like a myocardial infraction. And if it is lethal cause it should give the features and the which one of the lethal cause it is. The input is an instance with the features and after running the application it gives case it is with some explanation on the screen with the overview.

References

[1] Johns Hopkins medicine: *Heart attack*, Conditions and Diseases, Retrieved from https://www.hopkinsmedicine.org/health/conditions-and-diseases/heart-attack on 4-10-2021

[2] Machine learning repository: *Machine Learning*, Myocardial infraction complications data set, Retrieved from https://archive.ics.uci.edu/ml/datasets/Myocardial+infarction+complications on 04-10-2021

[3] RStudio: *Download RStudio*, Download the RStudio IDE, Retrieved from https://www.rstudio.com/products/rstudio/download/ on 15-11-2021

[4] Weka: *Download Weka*, Downloading and installing Weka, Retrieved from https://waikato.github.io/weka-wiki/ then Download, loading and installing Weka on 15-11-2021

[5] IntelliJ IDEA: *Download IntelliJ*, Download IntelliJ IDEA, Retrieved from https://www.jetbrains.com/idea/download/ on 19-11-2021

[6] Java, Oracle: *Download Java*, Download Java, Retrieved from https://www.java.com/nl/download/ on 19-11-2021

[7] Java SE Development Kit 17.0.1: *Download JDK*, Java SE 17 Archive Downloads, Retrieved from https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html on 19-11-2021

Appendix

```
knitr::opts_chunk$set(echo = FALSE)

knitr::opts_chunk$set(cache = TRUE)
## Libraries
options(digits = 3)
library(dplyr)
library(ggplot2)
library(tidyr)
library(readr)
library(gridExtra)
library(GGally)
library(purrr)
library(psych)
library(kableExtra)
library(ggbiplot)
library(png)
library(ggpubr)
library(cowplot)

# Define the file
data_file <- "data/Myocardial_infarction_complications_Database.csv"

# Load the data
Myocardial <- read.table(data_file, sep=",", header = TRUE, na.strings = "?")

# Head of the myocardial infraction data
head(Myocardial)

tmp <- Myocardial %>% select(2, 35:38) %>% drop_na()

df1 <- data.frame(x=tmp$AGE, y=tmp$S_AD_KBRIG)
df2 <- data.frame(x=tmp$AGE, y=tmp$D_AD_KBRIG)
df3 <- data.frame(x=tmp$AGE, y=tmp$S_AD_ORIT)
df4 <- data.frame(x=tmp$AGE, y=tmp$D_AD_ORIT)

ggplot(df1, aes(x,y)) +
  geom_violin(aes(color="Systolic blood pressure of ECT")) +
  geom_jitter(aes(color="Systolic blood pressure of ECT"),
              alpha=0.4) +
  geom_violin(data=df2,aes(color="Diastolic blood pressure of ECT")) +
  geom_jitter(aes(color="Diastolic blood pressure of ECT"),
              alpha=0.4) +
  geom_violin(data=df3,aes(color="Systolic blood pressure of ICU")) +
   geom_jitter(aes(color="Systolic blood pressure of ICU"),
              alpha=0.4) +
  geom_violin(data=df4, aes(color="Diastolic blood pressure of ICU")) +
   geom_jitter(aes(color="Diastolic blood pressure of ICU"),
              alpha=0.4) +
  xlab("Ages of patients (in years)") +
  ylab("blood pressures (in mmHg)") +
  labs(color="legend")
```

```r
ggplot(Myocardial, aes(x=factor(TIME_B_S))) +
  geom_bar(width = 0.7, fill = "orchid4") +
  xlab("Time elapsed from the start of the attack to the hospital") +
  ylab("Numbers of patients")
# Mutating the sex attribute to make it readable.
Myocardial_SEX <- Myocardial %>%
  mutate(SEX = factor(SEX, labels = c("Female", "Male"), levels = c(0, 1)))
# Plots the blood pressures
SB_EC <- ggplot(Myocardial_SEX, aes(x= S_AD_KBRIG, y=AGE)) +
  geom_jitter(aes(color = SEX), alpha = 0.3) +
  xlab("Systolic bloodpressure\naccording to EC")
DB_EC <- ggplot(Myocardial_SEX, aes(x=D_AD_KBRIG, y=AGE)) +
  geom_jitter(aes(color = SEX), alpha = 0.3) +
  xlab("Diastolic bloodpressure\naccording to EC")
SB_IC <- ggplot(Myocardial_SEX, aes(x=S_AD_ORIT, y=AGE)) +
  geom_jitter(aes(color = SEX), alpha = 0.3) +
  xlab("Systolic bloodpressure\naccording to IC")
DB_IC <- ggplot(Myocardial_SEX, aes(x=D_AD_ORIT, y=AGE)) +
  geom_jitter(aes(color = SEX), alpha = 0.3) +
  xlab("Diastolic bloodpressure\naccording to IC")

# Combine the plots
plot_grid(SB_EC, DB_EC, SB_IC, DB_IC,
          labels = c("A", "B", "C", "D"),
          label_size = 12)
# Making a subset of the ecg to make a violin plot with it.
Myocardial.ECG <- Myocardial %>%
  select("AGE", "ritm_ecg_p_01", "ritm_ecg_p_07", "ritm_ecg_p_08") %>%
  pivot_longer(-AGE, names_to = "sinus_with_a_heart_rate",
               values_to = "heart_rate") %>% drop_na()

# Mutating the heart rate to yes and no, to understand it better
Myocardial.ECG <- Myocardial.ECG %>%
  mutate(heart_rate = factor(heart_rate, labels = c("No","Yes"),
                             levels = c(0, 1)))

head(Myocardial.ECG)
# Violin plot of the ECG rhythm
ggplot(Myocardial.ECG, aes(sinus_with_a_heart_rate, AGE, col = heart_rate))+
  geom_violin() +
  geom_jitter(alpha = 0.1) +
  scale_x_discrete(labels = c('heart rate 60-90','heart rate above 90',
                              'heart rate above below 60')) +
  ylab("Ages of the patients (in years)") +
  xlab("ECG rhythm (in sinus)")
# Selecting the data to make a PCA
rows <- nrow(Myocardial)

myo <- Myocardial %>%
  select(where(function(x){sum(is.na(x))/rows < 0.3})) %>%
  select(-ID) %>%
  select(where(function(x){sum(is.infinite(x)) == 0})) %>%
  select(where(function(x){(sum(x == 0, na.rm=T) / rows) < 0.8})) %>%
  drop_na()

myo_pca <- prcomp(as.matrix(myo),
scale. = T,
center = T)
# Plotting the pca
plot(myo_pca, type = "l")
#shows the summary of PCA of the myocardial infraction
summary(myo_pca)
# Plot the PCA
g <- ggbiplot(myo_pca, obs.scale = 1, var.scale = 1, ellipse = FALSE, circle = TRUE)
g <- g + scale_color_discrete(name = '')
g <- g + theme(legend.direction = 'horizontal',
legend.position = 'top')
# Show plot
g
classifiers <- read.csv(file ='Weka_Data/Classifiers.csv', sep = ",", header = TRUE, fileEncoding="UTF-8
kable(classifiers, caption = "Classification using cross validation 10-fold") %>%
  kable_styling(latex_options = 'scale_down', position = 'center')


costsensitive <- read.csv("Weka_Data/CostSensitive.csv", sep = ";",
                          header = TRUE, fileEncoding="UTF-8-BOM")
kable(costsensitive, caption = "CostSensitiveClassifier with the two algorithms" ) %>%
  kable_styling(latex_options = 'scale_down', position = 'center')

# Read the roc curve data of RandomForest and SimpleLogisitic
roc_data_randomforest <- read.table("Weka_Data/roccurve_randomforest.arff",
                                    sep = ",", comment.char = "@")
roc_data_simplelogistic <- read.table("Weka_Data/roccurve_simplelogistic.arff",
                                    sep = ",", comment.char = "@")

# Defining the names the roc data for RandomForest
names(roc_data_randomforest) <- c("Instance_number", "True_Positives", "False_Negatives",
                         "False_Positives", "True_Negatives",
                         "False_Positive_Rate", "True_Positive_Rate",
                         "Precision", "Recall", "Fallout", "FMeasure",
                         "Sample_Size", "Lift", "Threshold")


# Assign colors for the classifier and threshold
colors <- c(classifier ="orange", threshold = "blue")

# Plot the ROC Curve of RandomForest
roc_randomforest <- ggplot(data = roc_data_randomforest, mapping = aes(x = False_Positive_Rate,
                                             y = True_Positive_Rate)) +
  geom_point(mapping = aes(color = "classifier")) +
  geom_line(aes(color = "classifier")) +
  geom_abline(aes(color = "threshold", slope = 1, intercept = 0)) +
  scale_color_manual(values = colors) +
  ggtitle("RandomForest") +
  xlab("False Positive Rate") +
  ylab("True Positive Rate") +
  theme_pubr() +
  theme(legend.title = element_blank())

# Define the names for SimpleLogistic
names(roc_data_simplelogistic) <- names(roc_data_randomforest)

# Plot the ROC Curve of SimpleLogistic
roc_simplelogistic <- ggplot(data = roc_data_simplelogistic, mapping = aes(x = False_Positive_Rate,
                                             y = True_Positive_Rate)) +
  geom_point(mapping = aes(color = "classifier")) +
  geom_line(aes(color = "classifier")) +
  geom_abline(aes(color = "threshold", slope = 1, intercept = 0)) +
  scale_color_manual(values = colors) +
  ggtitle("SimpleLogistic") +
  xlab("False Positive Rate") +
  ylab("True Positive Rate") +
  theme_pubr() +
  theme(legend.title = element_blank())

# Combine ROC Curve of RandomForest and SimpleLogistic plots
combined_roccurve <- plot_grid(roc_randomforest + theme(legend.position = "none"),
                               roc_simplelogistic + theme(legend.position = "none"),
                               labels = c("A", "B"), label_size = 12)
# Create legend
legend <- get_legend(roc_simplelogistic + guides(color = guide_legend(nrow = 1)))

# Plot the combined plots
plot_grid(combined_roccurve, legend, ncol = 1)

# Loads the result image

knitr::include_graphics("images/result.png")
```