Heart failure analysis

Jody Iabichino

16/11/2021

1. Introduction

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.

The scope of the analysis was death event prediction. We analized a dataset of 299 patients with heart failure collected in 2015.

The dataset used for the analysis came from the following link of UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/machine-learning-databases/00519/heart_failure_clinical_records_dataset.csv

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

The patients consisted of 105 women and 194 men, and their ages ranged between 40 and 95 years old. All 299 patients had left ventricular systolic dysfunction and had previous heart failures that put them in classes III or IV of New York Heart Association (NYHA) classification of the stages of heart failure.

The dataset contained 13 features, which reported clinical, body, and lifestyle information, that we briefly describe here:

- age: age of the patient (years);
- anaemia: decrease of red blood cells or hemoglobin (boolean);
- high blood pressure: if the patient has hypertension (boolean);
- creatinine phosphokinase (CPK): level of the CPK enzyme in the blood (mcg/L);
- diabetes: if the patient has diabetes (boolean);
- ejection fraction: percentage of blood leaving the heart at each contraction (percentage);
- platelets: platelets in the blood (kiloplatelets/mL);
- sex: woman or man (binary);
- serum creatinine: level of serum creatinine in the blood (mg/dL);
- serum sodium: level of serum sodium in the blood (mEq/L);
- smoking: if the patient smokes or not (boolean);
- time: follow-up period (days);

- death event (target): if the patient deceased during the follow-up period (boolean);

Like we report, some features were binary: anaemia, high blood pressure, diabetes, sex, and smoking. The hospital physician considered a patient having anaemia if haematocrit levels were lower than 36%. Unfortunately, the original dataset manuscript provided no definition of high blood pressure.

The idea behind this analysis was that a machine learning algorithm could be performed with this data to predict death event.

So, after some steps of data cleaning and data exploration, the dataset was splitted into two separate datasets: a "training set" and a "test set". According to machine learning standards, the development and training of the algorithms was made on the "training set", while the final RMSE was evaluated on the "test set".

We tried to use three kinds of different models and the model with the lowest RMSE achievable was finally chosen as the best model. RMSE is a measure of "goodness of fit", calculated as the square root of the average through all the observations of the difference squared between death event counts and predicted death event counts.

In the last chapter of the analysis, we have tried to propose some suggestions for future developments.

2. Analysis

2.1 Downloading

Before starting with the actual data exploration, the zipper file containing all the necessary data was downloaded:

```
url <- "https://archive.ics.uci.edu/ml/machine-learning-databases/00519/heart_failure_clinical_records_
path <- file.path("~", "heart_failure_clinical_records_dataset.csv")
download.file(url, path)
dataset <- read.table(path,
                      header = TRUE,
                      sep = ",",
                      stringsAsFactors = FALSE)
```

2.2 Data cleaning

At this point, we analyzed the database to understand its structure, also by analyzing its first rows:

```
str(dataset)
```

```
## 'data.frame':    299 obs. of  13 variables:
##  $ age                     : num  75 55 65 50 65 90 75 60 65 80 ...
##  $ anaemia                 : int  0 0 0 1 1 1 1 1 0 1 ...
##  $ creatinine_phosphokinase: int  582 7861 146 111 160 47 246 315 157 123 ...
##  $ diabetes                : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ ejection_fraction       : int  20 38 20 20 20 40 15 60 65 35 ...
##  $ high_blood_pressure     : int  1 0 0 0 0 1 0 0 0 1 ...
##  $ platelets               : num  265000 263358 162000 210000 327000 ...
##  $ serum_creatinine        : num  1.9 1.1 1.3 1.9 2.7 2.1 1.2 1.1 1.5 9.4 ...
##  $ serum_sodium            : int  130 136 129 137 116 132 137 131 138 133 ...
##  $ sex                     : int  1 1 1 1 0 1 1 1 0 1 ...
##  $ smoking                 : int  0 0 1 0 0 1 0 1 0 1 ...
##  $ time                    : int  4 6 7 7 8 8 10 10 10 10 ...
##  $ DEATH_EVENT             : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
head(dataset)
```

```
##   age anaemia creatinine_phosphokinase diabetes ejection_fraction
## 1  75       0                      582        0                20
## 2  55       0                     7861        0                38
## 3  65       0                      146        0                20
## 4  50       1                      111        0                20
## 5  65       1                      160        1                20
## 6  90       1                       47        0                40
##   high_blood_pressure platelets serum_creatinine serum_sodium sex smoking time
## 1                   1    265000              1.9          130   1       0    4
## 2                   0    263358              1.1          136   1       0    6
## 3                   0    162000              1.3          129   1       1    7
## 4                   0    210000              1.9          137   1       0    7
## 5                   0    327000              2.7          116   0       0    8
## 6                   1    204000              2.1          132   1       1    8
##   DEATH_EVENT
## 1           1
## 2           1
## 3           1
## 4           1
## 5           1
## 6           1
```

At the end of the first check, we have verified that there were no missing values to be dealt with:

```
sum(is.na(dataset))
```

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

And no missing value was detected.

2.3 Data exploration

Having verified that the database was clean, we explored the variables through a series of quick statistics.

a. Death_event

```
knitr::kable(table(dataset$DEATH_EVENT))
```

Var1	Freq
0	203
1	96

The survived patients (death event = 0) were 203, while the dead patients (death event = 1) were 96.

b. Serum_creatinine

```
dataset %>%
  summarize(min_serum = min(serum_creatinine),
            max_serum = max(serum_creatinine))
```

```
##   min_serum max_serum
## 1       0.5       9.4
```

Serum creatinine ranged from 0.5 to 9.4. We groupped the values of the variable into bands so that we could better analyze its behavior.

```
dataset <- dataset %>%
  mutate(serum_range = as.factor(case_when(serum_creatinine <= 1.5 ~ "<=1.5",
                                          serum_creatinine >1.5 & serum_creatinine <=3 ~ "1.5-3",
                                          serum_creatinine >3 & serum_creatinine <=4.5 ~ "3-4.5",
                                          serum_creatinine >4.5 & serum_creatinine <=6 ~ "4.5-6",
                                          serum_creatinine >6 ~ ">6"
  )))

knitr::kable(table(dataset$serum_range, dataset$DEATH_EVENT))
```

	0	1
<=1.5	179	53
>6	1	3
1.5-3	18	35
3-4.5	4	4
4.5-6	1	1

Frequencies showed that for high values of the variable, there was a higher probability that the patient would die.

c. Ejection_fraction

```
dataset %>%
  summarize(min_eje = min(ejection_fraction),
            max_eje = max(ejection_fraction))
```

```
##   min_eje max_eje
## 1      14      80
```

Ejection_fraction ranged from 14 to 80. We groupped the values of the variable into bands so that we could better analyze its behavior.

```r
dataset <- dataset %>%
  mutate(eje_range = as.factor(case_when(ejection_fraction <= 20 ~ "<=20",
                                         ejection_fraction >20 & ejection_fraction <=40 ~ "20-40",
                                         ejection_fraction >40 & ejection_fraction <=60 ~ "40-60",
                                         ejection_fraction >60 & ejection_fraction <=80 ~ "60-80",
                                         ejection_fraction >80 ~ ">80"
  )))

knitr::kable(table(dataset$eje_range, dataset$DEATH_EVENT))
```

	0	1
<=20	3	20
20-40	139	57
40-60	59	16
60-80	2	3

Frequencies showed that for low values of the variable, there was a higher probability that the patient would die.

d. Age

```r
knitr::kable(table(dataset$age, dataset$DEATH_EVENT))
```

	0	1
40	7	0
41	1	0
42	6	1
43	1	0
44	2	0
45	13	6
46	2	1
47	1	0
48	0	2
49	3	1
50	19	8
51	3	1
52	5	0
53	9	1
54	1	1
55	14	3
56	1	0
57	1	1
58	8	2
59	1	3
60	20	13
60.667	1	1
61	4	0
62	4	1

	0	1
63	8	0
64	3	0
65	18	8
66	2	0
67	2	0
68	3	2
69	1	2
70	18	7
72	2	5
73	3	1
75	5	6
77	1	1
78	2	0
79	1	0
80	2	5
81	1	0
82	0	3
85	3	3
86	0	1
87	0	1
90	1	2
94	0	1
95	0	2

The patients were between 40 and 95 years old. We groupped the values of the variable into bands so that we could better analyze its behavior.

```
dataset <- dataset %>%
  mutate(age_range = as.factor(case_when(age <= 50 ~ "<=50",
                                          age >50 & age<=60 ~ "51-60",
                                          age >60 & age<=70 ~ "61-70",
                                          age >70 & age<=80 ~ "71-80",
                                          age>=81 ~ "Over 80"
         )))

knitr::kable(table(dataset$age_range, dataset$DEATH_EVENT))
```

	0	1
<=50	55	19
51-60	63	25
61-70	64	21
71-80	16	18
Over 80	5	13

Frequencies showed that only in case of patients with more than 70 years old, there was an higher probability of die.

e. Diabetes

```
knitr::kable(table(dataset$diabetes, dataset$DEATH_EVENT))
```

	0	1
0	118	56
1	85	40

Deaths were equally distributed between patients with and without diabetes.

f. Anaemia

```
knitr::kable(table(dataset$anaemia, dataset$DEATH_EVENT))
```

	0	1
0	120	50
1	83	46

Deaths were equally distributed between patients with and without anaemia.

g. Sex

```
knitr::kable(table(dataset$sex, dataset$DEATH_EVENT))
```

	0	1
0	71	34
1	132	62

Deaths were equally distributed between women and men.

h. Creatinine_phosphokinase

```
knitr::kable(table(dataset$creatinine_phosphokinase, dataset$DEATH_EVENT))
```

	0	1
23	0	1
30	1	0
47	1	2

	0	1
52	1	0
53	1	0
54	1	0
55	1	0
56	2	0
57	1	0
58	1	0
59	3	0
60	1	2
61	2	0
62	1	0
63	1	0
64	3	0
66	3	1
68	2	1
69	2	1
70	0	1
72	1	0
75	1	0
76	0	1
78	1	0
80	2	0
81	1	1
84	3	0
86	1	0
88	1	0
90	1	0
91	0	1
92	1	0
93	1	0
94	0	1
95	1	0
96	2	0
97	1	0
99	0	1
101	1	0
102	2	0
103	1	0
104	0	1
109	2	0
110	0	1
111	0	1
112	0	1
113	1	1
115	3	0
118	1	0
119	1	0
121	1	0
122	1	1
123	0	1
124	0	1
125	0	1

	0	1
127	1	0
128	0	1
129	2	2
130	1	0
131	0	1
132	2	0
133	1	0
135	2	0
143	1	1
144	1	0
145	0	1
146	0	1
148	1	1
149	0	1
151	1	0
154	0	1
156	1	0
157	1	1
159	1	0
160	0	1
161	0	1
166	0	1
167	2	0
168	0	2
170	1	0
171	1	0
176	0	1
180	1	0
185	1	0
190	1	0
191	1	0
193	1	0
196	2	0
198	1	0
200	1	0
203	1	0
207	1	0
211	1	0
212	2	0
213	1	0
220	0	1
224	2	0
231	2	1
232	1	0
233	0	1
235	0	1
244	1	0
245	1	0
246	0	1
248	1	0
249	0	1
250	1	1

	0	1
253	1	0
257	1	0
258	0	1
260	0	1
270	1	0
280	0	1
281	1	0
291	1	0
292	1	0
298	1	0
305	1	0
308	1	0
315	0	1
318	0	1
320	1	0
326	1	0
328	0	1
335	1	0
336	1	0
337	1	0
358	1	0
364	0	1
369	1	0
371	0	1
379	0	1
395	0	1
400	1	0
418	0	1
427	0	1
446	1	0
478	1	0
482	1	0
514	1	0
553	0	1
571	0	1
572	1	0
577	0	1
582	30	17
588	0	1
607	1	0
615	1	0
618	1	0
624	1	0
646	1	0
655	1	0
675	1	0
707	1	0
719	1	0
720	1	0
737	1	0
748	1	0
754	1	0

	0	1
776	0	1
789	0	1
805	0	1
835	2	0
855	0	1
892	1	0
897	1	0
898	1	0
910	1	0
936	1	0
943	0	1
972	1	0
981	0	1
1021	1	0
1051	1	0
1082	1	0
1185	1	0
1199	0	1
1202	1	0
1211	1	0
1380	0	1
1419	1	0
1548	1	0
1610	1	0
1688	1	0
1767	1	0
1808	1	0
1820	1	0
1846	1	0
1876	1	0
1896	0	1
2017	0	1
2060	1	0
2261	1	0
2281	1	0
2334	0	1
2413	1	0
2442	0	1
2522	1	0
2656	1	0
2695	1	0
2794	1	0
3964	0	1
3966	1	0
4540	1	0
5209	1	0
5882	0	1
7702	0	1
7861	0	1

Deaths were equally distributed between patients with high and low creatinine phosphokinase levels.

i. High_blood_pressure

```
knitr::kable(table(dataset$high_blood_pressure, dataset$DEATH_EVENT))
```

	0	1
0	137	57
1	66	39

Deaths were equally distributed between patients with high blood pressure (37%) and patients with low blood pressure (29%).

l. Platelets

```
knitr::kable(table(dataset$platelets, dataset$DEATH_EVENT))
```

	0	1
25100	1	0
47000	0	1
51000	1	0
62000	0	1
70000	0	1
73000	1	0
75000	0	1
87000	0	1
105000	1	0
119000	0	1
122000	1	0
126000	0	1
127000	1	1
130000	1	0
132000	1	0
133000	2	0
136000	0	1
140000	1	1
141000	1	0
147000	2	0
149000	2	1
150000	1	0
151000	0	1
153000	0	3
155000	1	0
160000	1	0
162000	1	1
164000	1	0
166000	0	2
172000	2	0
173000	2	0

	0	1
174000	1	0
176000	1	0
179000	1	0
181000	1	0
184000	1	0
185000	1	1
186000	1	0
188000	0	1
189000	3	0
192000	0	1
194000	2	1
196000	0	2
198000	0	1
2e+05	0	1
201000	1	0
203000	3	0
204000	0	2
208000	1	0
210000	1	2
211000	1	0
212000	1	0
213000	0	1
215000	2	0
216000	1	1
217000	0	1
218000	2	0
219000	1	1
220000	3	0
221000	3	1
222000	2	0
223000	2	1
224000	0	1
225000	0	1
226000	3	1
227000	1	0
228000	3	1
229000	1	0
231000	2	0
232000	1	0
233000	1	0
235000	3	1
236000	1	0
237000	3	1
241000	1	0
242000	2	0
243000	0	1
244000	1	2
246000	1	0
248000	1	0
249000	3	0
250000	1	0
252000	1	0

	0	1
253000	1	1
254000	2	1
255000	3	1
257000	1	0
259000	1	0
260000	1	0
262000	1	1
263000	1	0
263358.03	15	10
264000	1	0
265000	1	2
266000	1	1
267000	2	0
268000	1	0
270000	2	0
271000	2	2
274000	2	1
275000	1	0
276000	1	1
277000	2	0
279000	4	0
281000	1	0
282000	1	0
283000	3	0
284000	0	1
286000	1	0
289000	0	1
290000	1	0
293000	1	0
294000	1	0
295000	1	0
297000	1	1
298000	1	0
3e+05	1	0
301000	1	0
302000	2	1
303000	1	0
304000	2	0
305000	4	0
306000	0	1
308000	1	0
309000	1	0
310000	0	1
314000	0	1
317000	1	0
318000	1	0
319000	0	2
321000	0	1
324000	1	0
325000	1	0
327000	2	1
328000	1	0

	0	1
329000	1	1
330000	1	0
334000	1	1
336000	1	0
337000	1	0
338000	0	1
348000	1	0
350000	1	0
351000	0	2
358000	1	0
360000	0	1
362000	3	0
365000	1	1
368000	1	1
371000	1	0
374000	1	0
377000	1	0
382000	1	0
385000	0	1
388000	0	1
389000	1	1
390000	1	1
395000	1	1
404000	1	0
406000	2	0
418000	0	1
422000	1	0
427000	1	0
448000	1	0
451000	1	1
454000	0	1
461000	0	1
481000	1	0
497000	0	1
504000	1	0
507000	1	0
533000	1	0
543000	1	0
621000	0	1
742000	1	0
850000	1	0

Deaths were equally distributed between patients with high and low platelets levels

m. Serum_sodium

```
knitr::kable(table(dataset$serum_sodium, dataset$DEATH_EVENT))
```

	0	1
113	1	0
116	0	1
121	0	1
124	0	1
125	1	0
126	1	0
127	0	3
128	1	1
129	0	2
130	6	3
131	2	3
132	6	8
133	8	2
134	15	17
135	10	6
136	29	11
137	31	7
138	17	6
139	16	6
140	28	7
141	11	1
142	7	4
143	3	0
144	3	2
145	6	3
146	0	1
148	1	0

The most important number of survived patients were between 134 and 141

n. Smoking

```
knitr::kable(table(dataset$smoking, dataset$DEATH_EVENT))
```

	0	1
0	137	66
1	66	30

Smoking patients showed a similar probability to die (31%) respect to no-smoking patients (32%).

o. Time

```
knitr::kable(table(dataset$time, dataset$DEATH_EVENT))
```

	0	1
4	0	1
6	0	1
7	0	2
8	0	2
10	0	6
11	0	2
12	1	0
13	0	1
14	0	2
15	0	2
16	1	0
20	0	2
22	1	0
23	0	2
24	0	1
26	0	3
27	0	1
28	0	2
29	1	1
30	1	4
31	0	1
32	0	1
33	1	2
35	0	1
38	0	1
40	0	1
41	0	1
42	0	1
43	0	3
44	0	1
45	0	1
50	0	1
54	2	0
55	0	1
59	0	1
60	1	2
61	0	1
63	1	0
64	0	1
65	0	2
66	0	1
67	0	1
68	1	0
71	1	0
72	1	1
73	0	2
74	4	0
75	1	0
76	1	0
77	0	1
78	1	1
79	5	0

	0	1
80	2	0
82	1	1
83	3	0
85	2	0
86	1	0
87	5	0
88	4	1
90	2	2
91	2	0
94	3	0
95	4	1
96	0	1
97	1	0
100	0	1
104	2	0
105	1	0
106	1	0
107	6	0
108	3	0
109	2	1
110	1	0
111	0	1
112	2	0
113	1	1
115	1	1
117	1	0
118	1	0
119	1	0
120	4	0
121	4	0
123	1	0
126	0	1
129	0	1
130	0	1
134	1	0
135	0	1
140	1	0
145	2	0
146	5	0
147	4	0
148	1	0
150	0	1
154	0	1
162	0	1
170	0	1
171	0	1
172	1	2
174	3	0
175	1	0
180	1	2
185	1	0
186	6	0

	0	1
187	7	0
188	1	0
192	2	0
193	0	1
194	1	0
195	1	0
196	1	1
197	2	0
198	0	1
200	1	0
201	2	0
205	3	0
206	1	0
207	2	1
208	1	0
209	5	0
210	2	0
211	1	0
212	3	0
213	3	0
214	4	1
215	4	0
216	1	0
220	1	0
230	2	0
231	1	0
233	2	0
235	0	1
237	2	0
240	1	0
241	0	1
244	5	0
245	5	0
246	3	0
247	1	0
250	7	0
256	2	0
257	1	0
258	2	0
270	2	0
271	1	0
278	1	0
280	1	0
285	1	0

We groupped the values of the variable into bands so that we could better analyze its behavior.

```
dataset <- dataset %>%
  mutate(time_range = as.factor(case_when(time <= 50 ~ "<=50",
                                          time >50 & time <=100 ~ "50-100",
                                          time >100 & time <=150 ~ "100-150",
                                          time >150 & time <=200 ~ "150-200",
                                         time >200 & time <=250 ~ "200-250",
                                         time >250 ~ ">250"
  )))

knitr::kable(table(dataset$time_range, dataset$DEATH_EVENT))
```

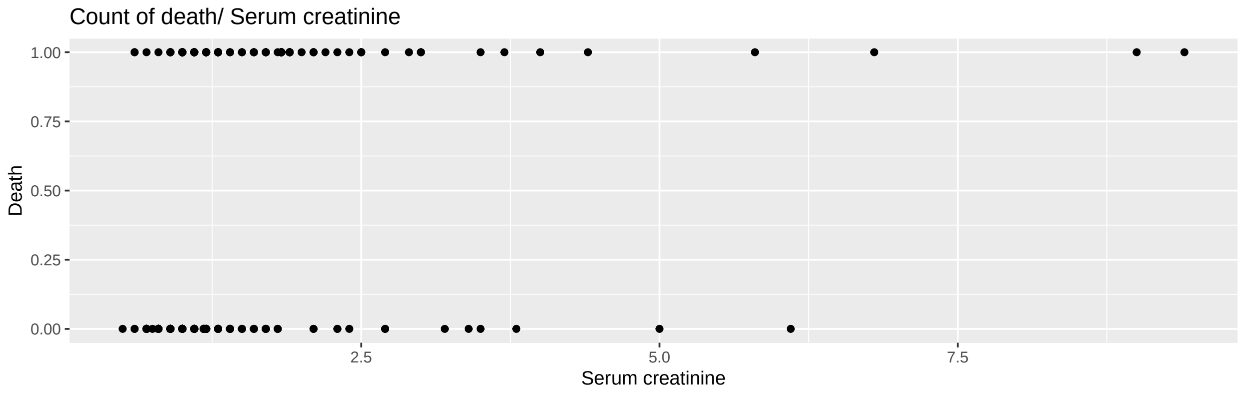
	0	1
<=50	6	50
>250	11	0
100-150	46	9
150-200	29	11
200-250	62	4
50-100	49	22

Analyzing the time variable, no particular trends were found.

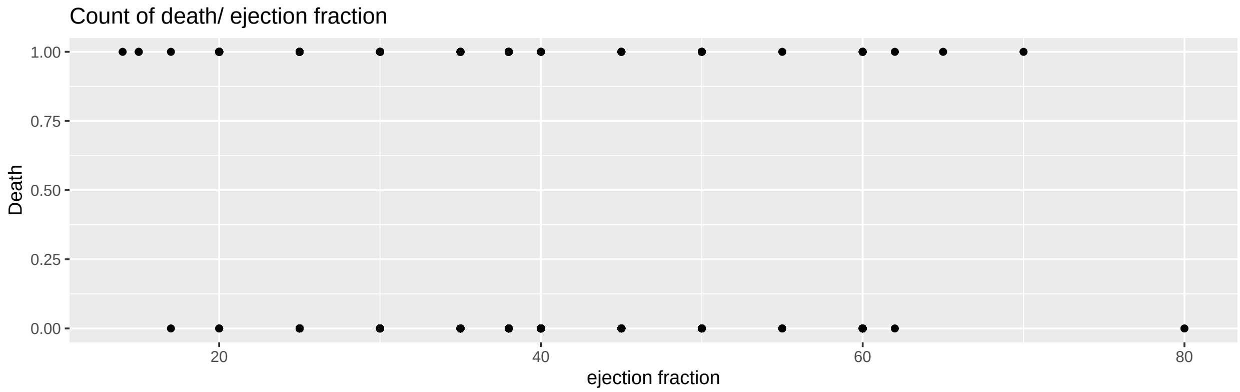
2.4 Data visualization

Serum creatinine and ejection fraction showed a key role in predicting death events. In the following graphs, we tried to visualize the behavior of the two variables, and in particular their relationship with the target variable (death event).

```
ggplot(data = dataset, aes(x = serum_creatinine, y = DEATH_EVENT)) +
  geom_point() +
  labs(x = "Serum creatinine",
       y = "Death") +
  ggtitle("Count of death/ Serum creatinine")
```



```
ggplot(data = dataset, aes(x = ejection_fraction, y = DEATH_EVENT)) +
  geom_point() +
  labs(x = "ejection fraction",
       y = "Death") +
  ggtitle("Count of death/ ejection fraction")
```



3. Machine learning models

Our results showed that serum creatinine and ejection fraction were sufficient to predict survived and dead patients from records. So, we tried to select some machine learning models that could be used with these two factors. In particular, we selected a KNN (k-Nearest Neighbors) and a random forest, among the most popular models applied in machine learning; moreover, given the nature of the target variable, we decided to try also a logit regression model, whose output is always between 0 and 1.

3.1 Data partition

The application of a machine learning model always starts with the subdivision of the database in two parts: a training set and a test set on which then proceed with the validation. Typically 80% of the database becomes the training set, while 20% is used as a test set, but given the choice of a random forest, subject to over-fitting, we preferred a split of 70%-30%.

```
dataset_selection <- dataset %>%
  select(DEATH_EVENT, serum_creatinine, ejection_fraction)
set.seed(1)
index<-createDataPartition(dataset$DEATH_EVENT,times=1,p=0.3,list=FALSE)
dataset_selection_train <- dataset_selection[index,]
dataset_selection_test <- dataset_selection[-index,]
```

3.2 The three models

We started fitting the KNN model on the "training set" and determining the predictions on the "test set":

```
knn_fit <- knn3(DEATH_EVENT ~ .,
                data = dataset_selection_train, k=5)

y_hat_knn <- predict(knn_fit, dataset_selection_test, type = "prob")
```

Then we fit the random forest model on the "training set" and determined the predictions on the "test set"

```
rf_fit <- randomForest(DEATH_EVENT ~ .,data = dataset_selection_train)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
y_hat_rf <- predict(rf_fit, newdata = dataset_selection_test)
```

We finally fit the logistic regression on the “training set” and determined the predictions on the “test set”.

```
log_fit <-glm(DEATH_EVENT ~ .,data = dataset_selection_train, family = "binomial")
y_hat_log <- predict(log_fit, newdata = dataset_selection_test, type = "response")
```

4. Models results

At the end we could analized the models results. The three models have typical features that distinguish them. The KNN, for example, estimates conditional probabilities in a manner similar to bin smoothing, and is also easily adaptable to multiple dimensions. Random forests, on the other hand, are popular machine learning approaches in which multiple decision trees are averaged (a forest of trees constructed with randomness). Finally, logistic regression is an extension of linear regression that ensures that the estimated conditional probabilities are between 0 and 1. This approach, which uses a logistic transformation, has the limitation of failing to capture the possible nonlinear nature of the true conditional probabilities.

To be able to understand which model performed better, we looked at the RMSE of each model, evaluating it on the “test set”. In order to calculate the RMSE, we took the residuals of each model calculated on the “test set”; then we squared each of these and then we took the mean and then the square root of this value.

The lowest the RMSE, the better performance we had from the model.

```
rmses <- dataset_selection_test %>%
  mutate(residual_knn = y_hat_knn - DEATH_EVENT,
         residual_rf = y_hat_rf - DEATH_EVENT,
         residual_log = y_hat_log - DEATH_EVENT) %>%
  summarize(rmse_knn_model = sqrt(mean(residual_knn^2)),
            rmse_rf_model = sqrt(mean(residual_rf^2)),
            rmse_log_model = sqrt(mean(residual_log^2)),
            )

knitr::kable(rmses)
```

rmse_knn_model	rmse_rf_model	rmse_log_model
0.5741121	0.4215237	0.4157527

Logistic regression showed the lowest RMSE: it was the best model.

5. Conclusions

In this analysis we tried to make a death event prediction. We analized a dataset of 299 patients with heart failure collected in 2015 and we found that a good machine learning model for death event was the logistic regression.

Our results of this two-feature model showed that serum creatinine and ejection fraction were sufficient to predict survival of heart failure patients from medical records.

This discovery has the potential to impact on clinical practice, becoming a new supporting tool for physicians when predicting if a heart failure patient will survive or not. Indeed, medical doctors aiming at understanding if a patient will survive after heart failure may focus mainly on serum creatinine and ejection fraction.

For future work, other models could be tested, maybe doing an ensembling able to aggregate the property of different models.