BIG DATA MODELLING TECNICS PROJECT 2021 (UE2)

3 JUILLET 2022

PRIVATE MEDICAL COMPANY

Consulting team: AGOSSOU Ines, KOUYATE Inès, MAVUA Nandy, TIATSE SOUOP Varesse

1 Discovery

LEARN THE BUSINESS DOMAIN, ASSESS THE RESOURCES AVAILABLE TO SUPPORT THE PROJECT, FRAME THE BUSINESS PROBLEM AS AN ANALYTICS CHALLENGE, FORMULATE INITIAL HYPOTHESES TO TEST

We are currently enrolled as Business Intelligence Consultants in a private medical company, which has a service in charge of making research about human diseases. For our mission, they gave us a dataset about heart failures. This is one of their main subjects now, as well as the Covid-19.

Indeed, Cardiovascular diseases (CVDs) are the number 1 cause of death globally, taking an estimated 17.9 million lives each year, which accounts for 31% of all deaths worldwide. Four out of five CVD deaths are due to heart attacks and strokes, and one-third of these deaths occur prematurely in people under 70 years of age. Heart failure is a common event caused by CVDs and this dataset contains 11 features that can be used to predict a possible heart disease.

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

The dataset given contains 11 features that can be used to predict a possible heart disease. Here they are:

Feature	Definition	Values	Type
Age	age of the patient	[28-77 years old]	Integer
Sex	sex of the patient	[M: Male, F: Female]	Factor
ChestPainType	chest pain type	[TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]	Factor
RestingBP	resting blood pressure (mm Hg)	[0-200 mm Hg]	Integer
Cholesterol	serum cholesterol	[0-603 mm/dl]	Integer
FastingBS	fasting blood sugar	[1: if FastingBS > 120 mg/dl, 0: otherwise]	Factor
RestingECG	resting electrocardiogram results	[Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]	Factor
MaxHR	maximum heart rate achieved	[60-202]	Integer
ExerciseAngina	exercise-induced angina	[Y: Yes, N: No]	Factor
Oldpeak	oldpeak = ST	Values from -2.6 to 6.2 measured in depression]	Numeric
ST_Slope	the slope of the peak exercise ST segment	[Up: upsloping, Flat: flat, Down: downsloping]	Factor
HeartDisease	output class	[1: heart disease, 0: Normal]	Factor

« We want to use the data provided to predict heart diseases and identify the parameters that influence the most. Therefore, our outcome variable will be "HeartDisease", and all the other ones will be our explanatory variables. » - Consulting team, Data Scientist

Therefore, our business problem is:

« Which individuals are more likely to develop heart diseases? »

Hypothesis: Chest Pain Type, Age and Sex have a great influence on the prediction of heart diseases. Older men with a Chest Pain Type which is not angina have a great chance to develop heart diseases.

2 Data Preparation

EXPLAIN HOW YOU DID THE EXTRACT, LOAD, AND TRANSFORM (ELT) OR EXTRACT, TRANSFORM AND LOAD (ETL) STEPS, TO GET YOUR DATA.

We used RStudio for our entire analysis. We started by **changing the variables' types** when it was necessary. Then we **checked the missing values**, there was not any. To check the missing values, we used 3 methods: the visualization of missing values, the function anyNA, and the statistic test MCAR.

```
# 1.2. Check for missing values ------------------------------------------

#----------- METHOD 1: Visualize missing values-----------
# install.packages("naniar")
par(mfrow=c(1,1))
library(naniar)
vis_miss(data)
# Conclusion: No missing values.

#----------- METHOD 2: Use the function anyNA------------
anyNA(data, recursive = FALSE)
# Conclusion: No missing values.

#----------- METHOD 3: Statistic test MCAR---------------
mcar_test(data)
# Goal: assess if data is missing completely at random (MCAR).
# The statistic test is a chi-squared value. Ho is "the data is MCAR".
# Conclusion: Given the high statistic value and low p-value, we can conclude
# that the data is not missing completely at random.
```

Figure 1: Three methods to check the missing values

The last method, the MCAR test, permitted us to check if data was missing completely at random. Thanks to the results we concluded that the data was not missing completely at random.

Also, before starting our analysis, we **checked the normality of our quantitative variables** thanks to a Shapiro test. The result is that all variables have a p-value of less than 0.5.

After this, we had to handle the outliers. Here are the boxplots of our quantitative variables:

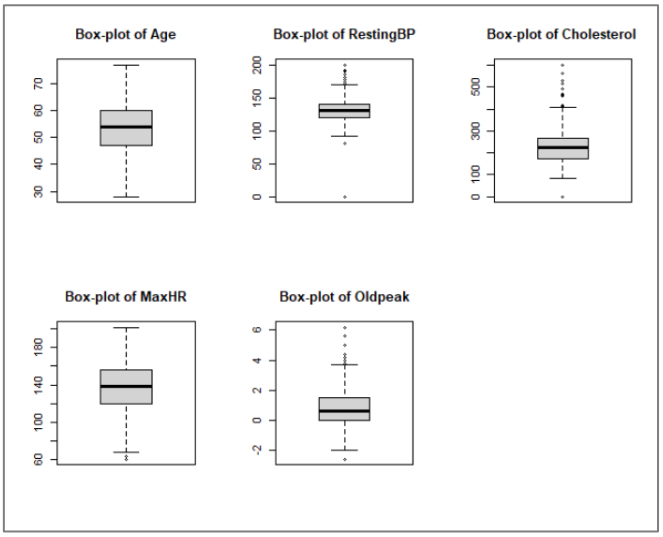

Figure 2: Box-plots of the quantitative variables

There were some outliers in the dataset, so we decided to use **Cook's Distance** using the traditional 4/n criterion. This technique allows to **check if the outliers are influential**, and if so, remove them. Our final dataset (without the influential observations) contains 91% of the original dataset. Here is how we created the function to plot the Cook's distance, it takes 3 inputs (a model, a dataset and a title):

```
plot.cooksd = function(mod, data, main){

  n = nrow(data)
  cooksd = cooks.distance(mod)
  plot(cooksd, pch="+", main = main)
  abline(h = 4/n, col="red")

}
```

Figure 3: Plot Cook's distance function

Thanks to this function we were able to **remove some influential outliers**. Here is a comparison of Cook's distance.

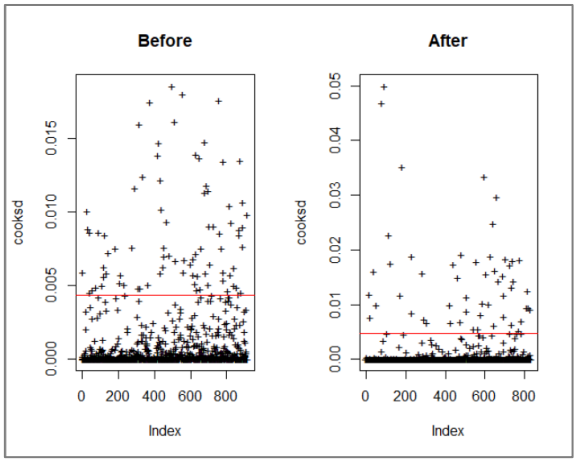

Figure 4: Comparison of Cook's distance before and after removing influential outliers (91% of the dataset kept)

Our last preprocessing step was **handling multicollinearity**. Multicollinearity is a problem because it undermines the statistical significance of an independent variable. It is also a pre-requisite to check before building logistic regression models.

Except for ST_Slope and Oldpeak (R = -0.52), there was not much correlation between the variables. Age was also correlated to MaxHR, but we can tolerate since the coefficient is only -0.39. Here are the correlations between numerical features, and between all variables.

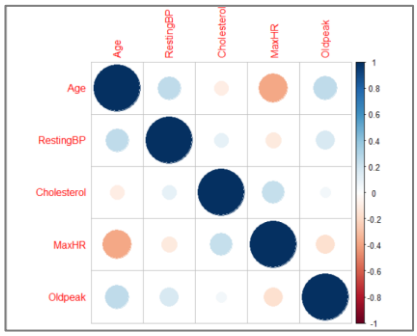

Figure 5: Correlation between numerical features

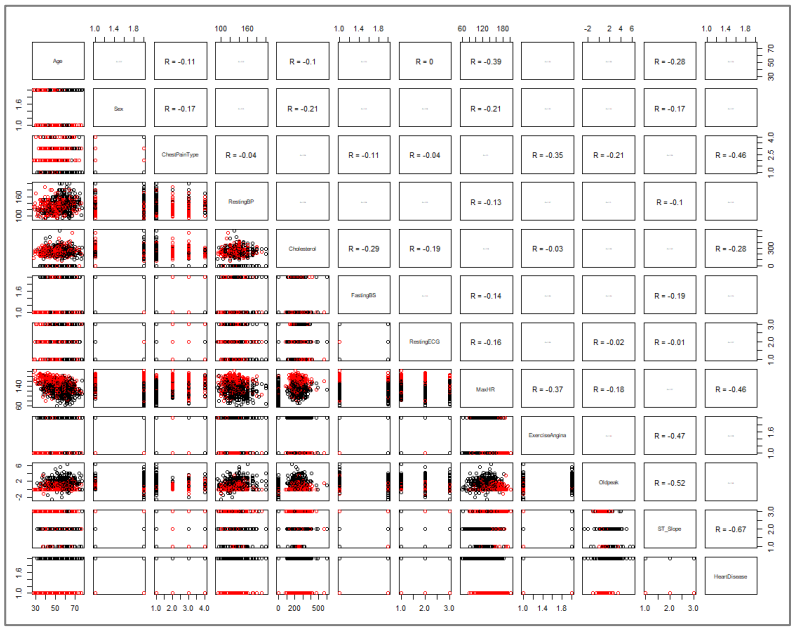

Figure 6: Correlation between all variables

3 Model planning

EXPLANATION AND JUSTIFICATION OF OUR CHOICES, METHODS, TECHNIQUES). DESCRIPTION OF HOW WE LEARNT ABOUT THE RELATIONSHIPS BETWEEN VARIABLES, AND HOW WE DID THE SELECTION OF KEY VARIABLES AND THE MOST SUITABLE MODELS.

« We chose to perform a logistic regression as it allows to study the relationship between the categorical/numerical variables (Xi) and the binary outcome variable "HeartDisease". When it comes to classification problems, this algorithm is extremely efficient, fast, and easy to interpret.

We decided then to perform Generalized Additive Models (GAM) using natural and smoothing splines for the sake of interpretability and flexibility. GAMs also allow to consider the nonlinearity of the variables, so we it may enhance our predictions… Let's see! » - Consulting team, Data Scientist

In order to select the best models, we opted for stepwise selection methods (backward and forward). We created two functions to evaluate the models faster, automatically get their AIC, accuracy, predictions, and know, thanks to the p-values, the variables to remove to enhance it step by step. We called these functions evaluate and evaluate.gam in the Rscript.

Here is the fastest method we used to get the best model for the logistic regression.

```
# 4.5. METHOD 3: Use the function stepAIC which allows to do both ---------
# forward and backward stepwise regressions at the same time.
# This method is faster and only requires one line of code.

#install.packages("MASS")
library(MASS)

full.model <- glm(HeartDisease ~., family = binomial, data = train)
step.model <- stepAIC(full.model, direction = "both",
                      trace = FALSE)
evaluate(step.model)
```

Once we got the best models for both classical logistic regression and GAMs, our strategy was to compare them using mainly their cross-validation errors and AIC. The best model must have the lowest cross-validation error and the lowest AIC, as well as a high recall.

Here is the loop that has been created to extract the models, their cross-validation errors and their AIC as a data frame.

```
# Comparison of cross-validation errors
library(boot)

cv_table = data.frame("model" = as.character(), "cv score" = as.numeric(), "AIC" = as.numeric)
for  (i in 1:length(all_models)) {

    model = all_models[[i]] #select one model
    score = cv.glm(test, model, K=10)$delta[1] #compute the error for this model
    score = round(score, 2)
    aic = model$aic #extract AIC
    aic = round(aic,2)
    name = names(all_models)[i]
    new_line = data.frame("model" = name, "cv score" = score, "AIC" = aic)
    cv_table = rbind(cv_table, new_line) #add nex line to the table
    cv_table = arrange(cv_table, -desc(cv_table$cv.score)) #order by cv error
    print(new_line)
  }
View(cv_table)
```

4 Model building

4.1. LOGISTIC REGRESSION

We have checked some assumptions before building the logistic regression model:

1. The outcome is binary
2. The observations are independent
3. The outliers are not strongly influential
4. There is enough data in the set
5. There is no multicollinearity between the variables or at least just a little
6. There is linearity of independent variables and log-odds

After checking these 6 assumptions, we created a 5-step functions to evaluate our models. This function is called “evaluate” and takes as input the model to evaluate:

```
evaluate <- function(mod){

  #------ 1st Step: Create Train/Test sets -------------------------------
  set.seed(1)
  row.number <- sample(1:nrow(data), 0.8*nrow(data))
  train <- data[row.number,]
  test <- data[-row.number,]
  # dim(train)
  # dim(test)

  #------ 2nd Step: Compare the predictions using the test set -----------
  pred.hd <- predict(mod, newdata = test, type = "response")
  # head(pred.hd)
  pred.hd <- ifelse(pred.hd>0.5,1,0) #Replace probabilities by 0/1 coding
  # head(pred.hd)


  #------ 3rd Step: Evaluate the model by checking accuracy and errors  --
  misclass.err <- mean(pred.hd != test$HeartDisease) %>% round(2)

  pred.hd <- factor(pred.hd, levels=c(0,1),
                    labels=c("Predicted Health", " Predicted Heart Disease"))
  test$HeartDisease <- factor(test$HeartDisease,
                              levels=c(0,1), labels=c("Health", "Heart Disease"))

  TN = round(prop.table(table(pred.hd,test$HeartDisease),2)[1,1]*100,0)
  TP = round(prop.table(table(pred.hd,test$HeartDisease),2)[2,2]*100,0)
  FN = round(prop.table(table(pred.hd,test$HeartDisease),2)[2,1]*100,0)
  FP = round(prop.table(table(pred.hd,test$HeartDisease),2)[1,2]*100,0)

  precision = round(TP/(TP+FP),2)
  recall = round(TP/(TP+FN),2)

  #------ 4th Step: Print the results ------------------------------------
  print(summary(mod))

  print(paste("AIC =", round(mod$aic,2)))
  print(paste("Missclassification error =", misclass.err))
  print(paste("Prediction accuracy =", 1-misclass.err))
  print(paste("Precision =", precision))
  print(paste("Recall =", recall))

  # print(table(pred.hd, test$HeartDisease))
  # print(prop.table(table(pred.hd,test$HeartDisease),2))

  print(paste(FP, "% of the predicted Healthy had indeed a Heart Disease"))
  print(paste(FN, "% of the predicted Heart Disease were indeed Healthy"))

  #--- 5th Step: Check if some variables can be removed to enhance the model----

  #Extract the pvalues
  pvalue = coef(summary(mod))[,4] %>% as.data.frame()
  colnames(pvalue) = "Pr(>|z|)"
  # pvalue

  #Extract the worst pvalue (without taking the intercept into account)
  worst_pvalue = coef(summary(mod))[-1,4] %>% max()
  # worst_pvalue

  #Extract the name of the worst pvalue
  worst_var = row.names(pvalue)[which(pvalue[,1] == worst_pvalue)]
  if (worst_pvalue > 0.05) {

    #Check if the variable to remove is dummy before choosing to remove it
    if (worst_var %in% names(mod$data)) {
      print(paste(worst_var, "is the less significant variable (Pr(>|z|) =",
                  round(worst_pvalue,2),"and can be removed"))
    } else {
      print(paste(worst_var, "is the less significant variable (Pr(>|z|) =",
                  round(worst_pvalue,2),"but can not be removed since it is a dummy variable."))
    }
  } else {
    "All variables in this model are significant."
  }
}
```

How did we enhance the models?

We have asterisks (*) next to each variable, which shows that each variable is significant:

```
Call:
glm(formula = HeartDisease ~ . - RestingECG - RestingBP - MaxHR,
    family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.6747  -0.0392   0.0077   0.0692   2.1891

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -4.704592   1.823800  -2.580  0.00989 **
Age               0.066011   0.024045   2.745  0.00604 **
SexM              4.639036   0.856585   5.416 6.10e-08 ***
ChestPainTypeATA -5.096093   0.885318  -5.756 8.60e-09 ***
ChestPainTypeNAP -5.168660   0.871232  -5.933 2.98e-09 ***
ChestPainTypeTA  -3.373314   1.093201  -3.086  0.00203 **
Cholesterol      -0.011790   0.002357  -5.003 5.66e-07 ***
FastingBS1        2.244754   0.675460   3.323  0.00089 ***
ExerciseAngina1   2.234261   0.490590   4.554 5.26e-06 ***
Oldpeak           1.094382   0.278660   3.927 8.59e-05 ***
ST_SlopeFlat      3.103693   1.167629   2.658  0.00786 **
ST_SlopeUp       -2.526734   1.187852  -2.127  0.03341 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 912.14  on 665  degrees of freedom
Residual deviance: 152.28  on 654  degrees of freedom
AIC: 176.28
```

We made predictions using several methods and enhanced the models using stepwise selection. Here are the results of the best model:

model_11 <- glm(HeartDisease~.-RestingECG -RestingBP -MaxHR , family = binomial, data = train)

```
Number of Fisher Scoring iterations: 8

[1] "AIC = 176.28"
[1] "Missclassification error = 0.04"
[1] "Prediction accuracy = 0.96"
[1] "Precision = 0.97"
[1] "Recall = 0.94"
[1] "3 % of the predicted Healthy had indeed a Heart Disease"
[1] "6 % of the predicted Heart Disease were indeed Healthy"
[1] "All variables in this model are significant."
```

5 Model building

5.2. GENERALIZED ADDITIVE MODELS

One of the disadvantages of logistic regression is that it is tough to obtain complex relationships. Interpretability, flexibility/automation, and regularization are at least three compelling reasons to utilize GAM. As a result, when your model has nonlinear effects, GAM delivers a regularized and interpretable solution, whereas other methods usually lack at least one of these three characteristics. To put it another way, GAMs strike a decent mix between the interpretable but biased linear model and the incredibly flexible "black box" learning algorithms.

To avoid overfitting, the GAM framework allows us to modify the smoothness of the predictor functions. We may directly address the bias/variance tradeoff by adjusting the wiggliness of the prediction functions.

Assumption: No assumptions are made about the distributions of the environmental variables. However, they should not be highly correlated with one another because this could cause problems with the estimation.

Process:

1. In our case, we created knots (using the quartiles) on the descriptive statistical value of the continuous variables (Age, RestingBP, Cholesterol, MaxHR and Oldpeak). This method helped us to avoid losing some information and it proved to be very useful since MaxHR and RestingBP for instance, which were not significant in the logistic regression, became significant with the GAMs.
2. We used natural splines in order to reduce variance on the boundaries of the approximation by using linear regression when we have a lack of data on the boundary, and smoothing spline to reduce the number of parameters estimated while still considering the nonlinearity of some variables.

```
# 1st try : Fit all numeric predictors using natural spline (even the ----
# predictors we removed previously using classical logistic regression).
# We'll use knots = median.

mod.gam = glm(HeartDisease ~ ns(Age, knots=c(47,54,60)) + Sex + ChestPainType +
                ns(RestingBP, knots=c(120,130,132.3)) + ns(Cholesterol, knots=c(172,221,197.3)) +
                FastingBS + RestingECG + ns(MaxHR, knots=c(119,137,155)) + ExerciseAngina +
                ns(Oldpeak, knots=c(0,0.5,0.8)) + ST_Slope, family=binomial, data = train)

evaluate.gam(mod.gam) #AIC = 181.23


# Given the results, we can get a better AIC = 178.76 using the following :
mod.gam = glm(HeartDisease ~ ns(Age, knots=c(47,54,60)) + Sex + ChestPainType +
                ns(RestingBP, knots = c(120, 130, 132.3)) +
                ns(Cholesterol, knots = c(172, 221, 197.3)) + FastingBS +
                ns(MaxHR, knots = c(119, 137, 155)) + ExerciseAngina +
                ns(Oldpeak, knots = c(0, 0.5, 0.8)) + ST_Slope,
              family=binomial, data = train)

evaluate.gam(mod.gam) #AIC = 178.76
```

In the function evaluate.gam :

•We plot the gam plot of each feature

•We compute the AIC using bidirectionnal stepwise selection

•We print significant variable and the formula selected by the model (i.e. the formula that gives the lowest AIC).

```
# 5.1. Create a function to evaluate our different GAM's model ----------

evaluate.gam = function(mod.gam){
  par(mfrow=c(3,4))
  plot.Gam(mod.gam, se=TRUE, col="red")
  step.gam = stepAIC(mod.gam, direction = "both", trace = FALSE)
  # summary(mod.gam)
  # evaluate(mod.gam) #to see the accuracy
  print(drop1(step.gam, test="Chisq")) #to see only significant predictors
  print(paste("AIC = ", summary(mod.gam)$aic %>% round(2)))

  print(paste("AIC = ", summary(step.gam)$aic %>% round(2),
              " for the folowing formula: ", summary(step.gam)$call[2]))
}
```

How did we enhance the models?

We have asterisks (*) next to each variable, which shows us that each variable is significant:

```
Single term deletions

Model:
HeartDisease ~ ns(Age, knots = c(47, 54, 60)) + Sex + ChestPainType +
    ns(RestingBP, knots = c(120, 130, 132.3)) + ns(Cholesterol,
    knots = c(172, 221, 197.3)) + FastingBS + ns(MaxHR, knots = c(119,
    137, 155)) + ExerciseAngina + ns(Oldpeak, knots = c(0, 0.5,
    0.8)) + ST_Slope
                                          Df Deviance    AIC     LRT  Pr(>Chi)
<none>                                         120.76 178.76
ns(Age, knots = c(47, 54, 60))             4   132.99 182.99  12.232 0.0157046 *
Sex                                        1   173.78 229.78  53.017 3.306e-13 ***
ChestPainType                              3   214.81 266.81  94.052 < 2.2e-16 ***
ns(RestingBP, knots = c(120, 130, 132.3))  4   130.11 180.11   9.351 0.0528954 .
ns(Cholesterol, knots = c(172, 221, 197.3)) 4  152.43 202.43  31.665 2.239e-06 ***
FastingBS                                  1   132.90 188.90  12.133 0.0004953 ***
ns(MaxHR, knots = c(119, 137, 155))        4   130.79 180.79  10.029 0.0399374 *
ExerciseAngina                             1   149.63 205.63  28.870 7.742e-08 ***
ns(Oldpeak, knots = c(0, 0.5, 0.8))        4   137.45 187.45  16.686 0.0022246 **
ST_Slope                                   2   244.43 298.43 123.666 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We made three models using GAMs. Here is the one with the lowest AIC:

mod.gam3 <- glm(HeartDisease ~ s(Age) + Sex + ChestPainType + ns(RestingBP, knots = c(120, 130, 132.3)) + ns(Cholesterol, knots = 221) + FastingBS + s(MaxHR) + ExerciseAngina + s(Oldpeak) + ST_Slope, family=binomial, data = train)

```
Number of Fisher Scoring iterations: 9

[1] "AIC = 166.96"
[1] "Missclassification error = 0.05"
[1] "Prediction accuracy = 0.95"
[1] "Precision = 0.95"
[1] "Recall = 0.96"
[1] "5 % of the predicted Healthy had indeed a Heart Disease"
[1] "4 % of the predicted Heart Disease were indeed Healthy"
[1] "ns(RestingBP, knots = c(120, 130, 132.3))4 is the less significan
t variable (Pr(>|z|) = 0.96 but can not be removed since it is a dummy
 variable."
```

6 Communicate results

		Estimated parameters	Cross-Validation error	AIC	Accuracy	Precision	Recall
Logistic Regression	model_11	**12**	**0.52**	176.28	**96%**	**97%**	94%
Natural and Smoothing Spline (GAM)	mod.gam3	18	0.54	**166.96**	95%	95%	**96%**

The best model based on AIC is **mod.gam3**. Besides, its cross-validation error (good technique to test a model on its predictive performance) is not so far from that of model_11, which tends to confirm that mod.gam3 is indeed the best model to choose.

However, one may wonder whether it is really valuable to build such a sophisticated model which requires the estimation of 18 parameters against 12 for model_11, especially since the accuracy and precision of the latter are better. The answer is simple: do not draw conclusions based on accuracy alone. Moreover, since the precision measures the quality of our predictions solely on the basis of what our predictor claims to be positive, while the recall measures quality with respect to the mistakes we have made, one can understand that the recall is more important in our case, because we are mostly interested in reducing false negative for rare medicine data modelling. Therefore, the model using GAM is the best model.

Here is the ranking in descending order of the 17 parameters estimated during the building of mod.gam3 (intercept excluded). ChestPainType has the highest influence and should be considered to be able to predict heart failure at a risk of 5%.

```
> rank_gam3 = varImp(mod.gam3)
> rank_gam3 = data.frame("Variable" = rownames(rank_gam3),
 "Overall" = rank_gam3$Overall)
> rank_gam3 = arrange(rank_gam3, desc(rank_gam3$Overall))
> rank_gam3
                                       Variable    Overall
1                                ChestPainTypeNAP 5.84600493
2                                            SexM 5.32852089
3                                ChestPainTypeATA 5.25957094
4                                 ExerciseAngina1 4.88025910
5                    ns(Cholesterol, knots = 221)1 4.52751765
6                                      s(Oldpeak) 3.50028561
7                                          s(Age) 3.39410876
8                                      FastingBS1 3.03465387
9                                ChestPainTypeTA 2.52026473
10                                  ST_SlopeFlat 2.51248529
11                                    ST_SlopeUp 2.18583077
12 ns(RestingBP, knots = c(120, 130, 132.3))3 2.17770324
13                                       s(MaxHR) 1.98718084
14 ns(RestingBP, knots = c(120, 130, 132.3))2 0.83376924
15 ns(RestingBP, knots = c(120, 130, 132.3))1 0.39845414
16                   ns(Cholesterol, knots = 221)2 0.32202685
17 ns(RestingBP, knots = c(120, 130, 132.3))4 0.05220107
```

Top 5 predictors

ChestPainType has a negative relation with HeartDisease. Having Typical Angina (TA), Atypical Angina (ATA), or Non-Anginal Pain (NAP), reduce the chance to develop heart diseases by 3 to 5 units accordingly (at a risk of 5%).

```
Coefficients:
                                        Estimate Std. Error z value Pr(>|z|)
(Intercept)                           -11.58834    3.47305  -3.337 0.000848 ***
s(Age)                                  0.10425    0.03071   3.394 0.000689 ***
SexM                                    5.14914    0.96634   5.329 9.90e-08 ***
ChestPainTypeATA                       -5.58187    1.06128  -5.260 1.44e-07 ***
ChestPainTypeNAP                       -5.91734    1.01220  -5.846 5.04e-09 ***
ChestPainTypeTA                        -3.35917    1.33287  -2.520 0.011727 *
ns(RestingBP, knots = c(120, 130, 132.3))1  0.59387    1.49043   0.398 0.690295
ns(RestingBP, knots = c(120, 130, 132.3))2  1.22681    1.47141   0.834 0.404411
ns(RestingBP, knots = c(120, 130, 132.3))3  8.63272    3.96414   2.178 0.029428 *
ns(RestingBP, knots = c(120, 130, 132.3))4  0.19508    3.73702   0.052 0.958368
ns(Cholesterol, knots = 221)1          -8.79918    1.94349  -4.528 5.97e-06 ***
ns(Cholesterol, knots = 221)2          -0.69637    2.16247  -0.322 0.747432
FastingBS1                              2.48554    0.81905   3.035 0.002408 **
s(MaxHR)                                0.02485    0.01250   1.987 0.046902 *
ExerciseAngina1                         2.98795    0.61225   4.880 1.06e-06 ***
s(Oldpeak)                              1.18671    0.33903   3.500 0.000465 ***
ST_SlopeFlat                            3.38867    1.34873   2.512 0.011988 *
ST_SlopeUp                             -3.04903    1.39491  -2.186 0.028828 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

« Yes indeed, thank you for asking […] here is an example of a diagnosis based on the coefficients of this model: At a risk of 5%, men having Asymptomatic Chest Pain (ASY), Resting blood pressure 130 mm Hg, Fasting Blood Sugar > 120 mg/dl (FastingBS1), Maximum heart rate achieved, Exercise-induced angina (ExerciseAngina1), Flat ST Slope, have a heart disease, and should receive appropriate treatments and follow-up. » - Consulting team, Data Scientist