ANEXO: Riesgo de crédito - Análisis, desarrollo y código

Miguel García Sánchez

2023-06-26

INICIO DEL DESARROLLO DEL TRABAJO

1. Introducción

El trabajo a desarrollar consiste en el análisis de un dataset relativo a datos de préstamos de entidades financieras, donde se quiere tratar de comprender cuales son las razones y características que llevan al llamado "default" o situación de impago, así como modelos que pueden ayudar a predecirlo y realizar un seguimiento sobre el mismo.

1.1. Desarrollo, programación y control de versiones

Se ha elegido para el desarrollo del trabajo el lenguaje de programación R (R version 4.2.2), cómo IDE de desarrollo RStudio (RStudio version 2022.12.0.353) y cómo herramienta de control de versiones GitHub (proyecto "/MDS_TFM" creado y vinculado al usuario Miguel_gs - "mgarciasanc2021").

Link del proyecto en GitHub: https://github.com/mgarciasanc2021/MDS_TFM

1.2. Paquetes R

Los paquetes de R utilizados para el desarrollo del código han sido los siguientes:

```
library(formatR)
library(readr)
library(ggplot2)
library(GGally)
library(dplyr)
library(tidyr)
library(missForest)
library(VIM)
library(formattable)
library(usmap)
library(cowplot)
library(corrplot)
library(MASS)
library(ggfortify)
library(nortest)
library(car)
library(lmtest)
library(PerformanceAnalytics)
```

```
library(Amelia)
library(ggthemes)
library(tidyverse)
library(tibble)
library(gridExtra)
library(factoextra)
library(caret)
library(ISLR)
library(rpart)
library(rpart.plot)
library(rattle)
library(tsne)
library(Rtsne)
library(class)
library(ada)
library(factoextra)
library(cluster)
library(useful)
library(mgcv)
library(xgboost)
library(randomForest)
library(kernlab)
library(pROC)
library(ggpubr)
library(ROCR)
```

2. Conjunto de datos

El conjunto de datos elegido para el desarrollo del trabajo es "Credit Risk Dataset". Este dataset incluye información de clientes de banca retail y préstamos contratados por estos en diferentes instituciones financiera.

Link del data set: https://www.kaggle.com/datasets/laotse/credit-risk-dataset.

2.1. Carga de los datos

El conjunto de datos "Credit Risk Dataset" contiene 12 columnas y 32.581 filas y se obtiene en formato .CSV.

Inicialmente se han guardado los datos en un data frame llamado "fin_credrisk" y se ha realizado un estudio preliminar sobre su contenido utilizando la función head y count.

```
fin_credrisk <- read_csv("financial_credit_risk.csv")
head(fin_credrisk)
```

```
## # A tibble: 6 x 12
##   person_age person_in~1 perso~2 perso~3 loan_~4 loan_~5 loan_~6 loan_~7 loan_~8
##        <dbl>       <dbl> <chr>     <dbl> <chr>   <chr>     <dbl>   <dbl>   <dbl>
## 1         22       59000 RENT        123 PERSON~ D         35000   16.0        1
## 2         21        9600 OWN           5 EDUCAT~ B          1000   11.1        0
## 3         25        9600 MORTGA~       1 MEDICAL C          5500   12.9        1
## 4         23       65500 RENT          4 MEDICAL C         35000   15.2        1
```

```
## 5         24       54400 RENT          8 MEDICAL C         35000   14.3        1
## 6         21        9900 OWN           2 VENTURE A          2500    7.14       1
## # ... with 3 more variables: loan_percent_income <dbl>,
## #   cb_person_default_on_file <chr>, cb_person_cred_hist_length <dbl>, and
## #   abbreviated variable names 1: person_income, 2: person_home_ownership,
## #   3: person_emp_length, 4: loan_intent, 5: loan_grade, 6: loan_amnt,
## #   7: loan_int_rate, 8: loan_status
```

```
count(fin_credrisk)
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1 32581
```

2.2. Definición de las variables

Empezando ya el análisis inicial del dataset que tenemos, se ve que las 12 variables que componen los datos pueden ser descritas como:

Input variables o Variables de entrada/predictoras:

- **person_age:** Edad de la persona que toma el crédito.
- **person_income:** Ingresos anuales en dólares de la persona que toma el crédito.
- **person_home_ownership:** Estado de la propiedad de la vivienda donde reside la persona que toma el crédito.
- **person_emp_length:** Periodo de tiempo en años desde que la persona que toma el crédito está en situación laboral activa.
- **loan_intent:** Uso del crédito concedido.
- **loan_grade:** Calidad crediticia del crédito concedido.
- **loan_amnt:** Cantidad en dólares de crédito concedido.
- **loan_int_rate:** Tipo de interés en porcentaje del crédito concedido.
- **loan_percent_income:** Porcentaje de lo que supone el préstamo sobre los ingresos anuales en dólares de la persona que toma el crédito.
- **cb_person_default_on_file:** Variable binaria que indica si la persona tomadora del crédito ha tenido antes una situación de impago o no.
- **cb_preson_cred_hist_length:** Duración en años del historial crediticio de la persona tomadora del crédito.

Output variable o Variable de salida/respuesta/objetivo:

- **loan_status:** Estado actual del crédito (suponiendo "1" como impagado o situación de default, y "0" no impagado)

Se analiza de forma preliminar estas variables utilizando la función summary.

```
summary(fin_credrisk)
```

```
##    person_age     person_income     person_home_ownership person_emp_length
##  Min.   : 20.00   Min.   :   4000   Length:32581          Min.   :  0.00
##  1st Qu.: 23.00   1st Qu.:  38500   Class :character      1st Qu.:  2.00
```

```
##  Median : 26.00   Median :  55000   Mode  :character      Median :  4.00
##  Mean   : 27.73   Mean   :  66075                         Mean   :  4.79
##  3rd Qu.: 30.00   3rd Qu.:  79200                         3rd Qu.:  7.00
##  Max.   :144.00   Max.   :6000000                         Max.   :123.00
##                                                           NA's   :895
##  loan_intent         loan_grade          loan_amnt     loan_int_rate
##  Length:32581       Length:32581       Min.   :  500   Min.   : 5.42
##  Class :character   Class :character   1st Qu.: 5000   1st Qu.: 7.90
##  Mode  :character   Mode  :character   Median : 8000   Median :10.99
##                                        Mean   : 9589   Mean   :11.01
##                                        3rd Qu.:12200   3rd Qu.:13.47
##                                        Max.   :35000   Max.   :23.22
##                                                        NA's   :3116
##   loan_status     loan_percent_income cb_person_default_on_file
##  Min.   :0.0000   Min.   :0.0000      Length:32581
##  1st Qu.:0.0000   1st Qu.:0.0900      Class :character
##  Median :0.0000   Median :0.1500      Mode  :character
##  Mean   :0.2182   Mean   :0.1702
##  3rd Qu.:0.0000   3rd Qu.:0.2300
##  Max.   :1.0000   Max.   :0.8300
##
##  cb_person_cred_hist_length
##  Min.   : 2.000
##  1st Qu.: 3.000
##  Median : 4.000
##  Mean   : 5.804
##  3rd Qu.: 8.000
##  Max.   :30.000
##
```

Sacando estos datos, se ven algunos resultados importantes a comentar, que posteriormente deberan ser analizados en mayor detalle y tratados:

- La variable “person_age” tiene un valor máximo de 144 años.
- La variable “person_emp_length” tiene un valor máximo de 123 años.
- La variable “person_emp_length” tiene 895 registros con NAs.
- La variable “loan_interest_rate” tiene 3.116 registros con NAs.

2.3. Objetivo del análisis

El objetivo final del proyecto es conseguir llegar a un modelo que permita predecir el riesgo de impago o “default” que puede tener en cartera una institución financiera.

De esta forma, se optimiza la fase de concesión de créditos y seguimiento del riesgo de crédito, logrando diferenciar si estamos ante créditos que van a poder ser devueltos por el cliente, o por si el contrario se van a quedar potencialmente como impagados y es conveniente aplicar de forma anticipada medidas preventivas de cara a la posible fase de recuperación del crédito.

ANÁLISIS INICIAL, LIMPIEZA Y PARTICIONADO DE LOS DATOS

3. Limpieza inicial del conjunto de datos

3.1. Cambio de nombres de las columnas

Se ha decidido no realizar un cambio en el nombre de las variables que aparecen en las columnas de los datos, ya que ya se sigue un patrón de nombre sin espaciones y así no habrá problemas con ello en el desarrollo del trabajo.

3.2. Cambio de tipo de variable de las columnas

Vemos que en el dataset existen variables de tipo numéricas (num - col_double()) y de tipo texto o cadena de caractéres (chr - col_character()).

```
str(fin_credrisk)
```

```
## spc_tbl_ [32,581 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ person_age                : num [1:32581] 22 21 25 23 24 21 26 24 24 21 ...
##  $ person_income             : num [1:32581] 59000 9600 9600 65500 54400 ...
##  $ person_home_ownership     : chr [1:32581] "RENT" "OWN" "MORTGAGE" "RENT" ...
##  $ person_emp_length         : num [1:32581] 123 5 1 4 8 2 8 5 8 6 ...
##  $ loan_intent               : chr [1:32581] "PERSONAL" "EDUCATION" "MEDICAL" "MEDICAL" ...
##  $ loan_grade                : chr [1:32581] "D" "B" "C" "C" ...
##  $ loan_amnt                 : num [1:32581] 35000 1000 5500 35000 35000 2500 35000 35000 35000 1600
##  $ loan_int_rate             : num [1:32581] 16 11.1 12.9 15.2 14.3 ...
##  $ loan_status               : num [1:32581] 1 0 1 1 1 1 1 1 1 1 ...
##  $ loan_percent_income       : num [1:32581] 0.59 0.1 0.57 0.53 0.55 0.25 0.45 0.44 0.42 0.16 ...
##  $ cb_person_default_on_file : chr [1:32581] "Y" "N" "N" "N" ...
##  $ cb_person_cred_hist_length: num [1:32581] 3 2 3 2 4 2 3 4 2 3 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   person_age = col_double(),
##   ..   person_income = col_double(),
##   ..   person_home_ownership = col_character(),
##   ..   person_emp_length = col_double(),
##   ..   loan_intent = col_character(),
##   ..   loan_grade = col_character(),
##   ..   loan_amnt = col_double(),
##   ..   loan_int_rate = col_double(),
##   ..   loan_status = col_double(),
##   ..   loan_percent_income = col_double(),
##   ..   cb_person_default_on_file = col_character(),
##   ..   cb_person_cred_hist_length = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Se ha decidido por ello realizar cambio en el tipo de variable de las que son cadena de caracter y pasarlas a variables categóricas.

```r
fin_credrisk$person_home_ownership = as.factor(gsub("\\$", "",
    fin_credrisk$person_home_ownership))

fin_credrisk$loan_intent = as.factor(gsub("\\$", "", fin_credrisk$loan_intent))

fin_credrisk$loan_grade = as.factor(gsub("\\$", "", fin_credrisk$loan_grade))

fin_credrisk$cb_person_default_on_file = as.factor(gsub("\\$",
    "", fin_credrisk$cb_person_default_on_file))
```

Vemos que con el cambio tenemos en el dataset variables de tipo numéricas y categóricas o de tipo factor.

```r
str(fin_credrisk)
```

```
## spc_tbl_ [32,581 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ person_age                : num [1:32581] 22 21 25 23 24 21 26 24 24 21 ...
##  $ person_income             : num [1:32581] 59000 9600 9600 65500 54400 ...
##  $ person_home_ownership     : Factor w/ 4 levels "MORTGAGE","OTHER",..: 4 3 1 4 4 3 4 4 4 3 ...
##  $ person_emp_length         : num [1:32581] 123 5 1 4 8 2 8 5 8 6 ...
##  $ loan_intent               : Factor w/ 6 levels "DEBTCONSOLIDATION",..: 5 2 4 4 4 6 2 4 5 6 ...
##  $ loan_grade                : Factor w/ 7 levels "A","B","C","D",..: 4 2 3 3 3 1 2 2 1 4 ...
##  $ loan_amnt                 : num [1:32581] 35000 1000 5500 35000 35000 2500 35000 35000 35000 1600
##  $ loan_int_rate             : num [1:32581] 16 11.1 12.9 15.2 14.3 ...
##  $ loan_status               : num [1:32581] 1 0 1 1 1 1 1 1 1 1 ...
##  $ loan_percent_income       : num [1:32581] 0.59 0.1 0.57 0.53 0.55 0.25 0.45 0.44 0.42 0.16 ...
##  $ cb_person_default_on_file : Factor w/ 2 levels "N","Y": 2 1 1 1 2 1 1 1 1 1 ...
##  $ cb_person_cred_hist_length: num [1:32581] 3 2 3 2 4 2 3 4 2 3 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   person_age = col_double(),
##   ..   person_income = col_double(),
##   ..   person_home_ownership = col_character(),
##   ..   person_emp_length = col_double(),
##   ..   loan_intent = col_character(),
##   ..   loan_grade = col_character(),
##   ..   loan_amnt = col_double(),
##   ..   loan_int_rate = col_double(),
##   ..   loan_status = col_double(),
##   ..   loan_percent_income = col_double(),
##   ..   cb_person_default_on_file = col_character(),
##   ..   cb_person_cred_hist_length = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Cabría la posibilidad de tratar de transformar la variable “loan_status” en categórica en función de si estamos ante préstamos de crédito impagados (“default”) o no.

4. Análisis preliminar del conjunto de datos

4.1. Análisis de datos faltantes

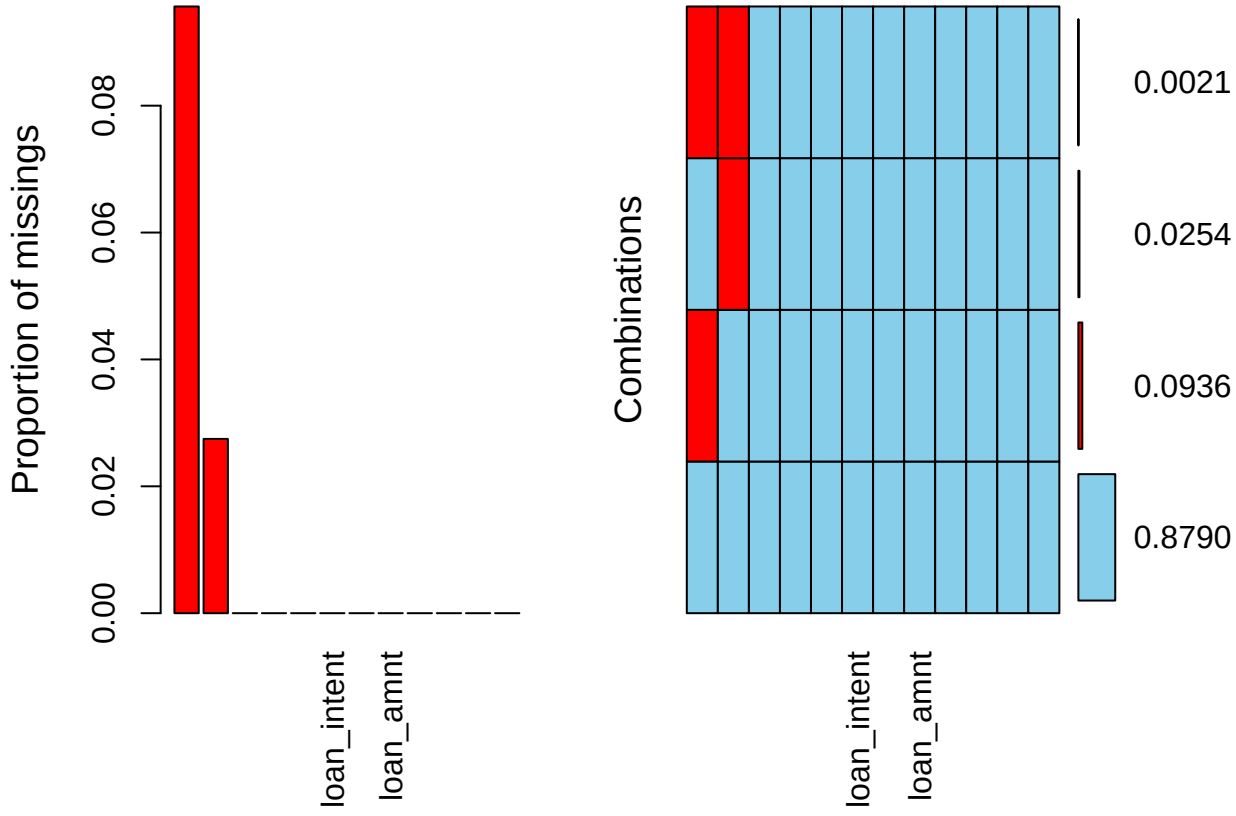
Se pasa ahora a analizar en mayor profundidad lo detectado anteriomente respecto a datos faltantes en el dataset:

- La variable "person_emp_length" tiene 895 registros con NAs.
- La variable "loan_interest_rate" tiene 3.116 registros con NAs.

Haciendo uso de la librería VIM y de la librería Amelia, analizamos la estructura que tienen los datos faltantes dentro de nuestro data set para ver y entender como se distribuyen y a que variables afecta.

Se puede comprobar que la proporción de datos faltantes es de aproximadamente un 1%. Hay 895 observaciones con datos faltantes en la variable "person_emp_length" y 3.116 observaciones con datos faltantes en "loan_int_rate".

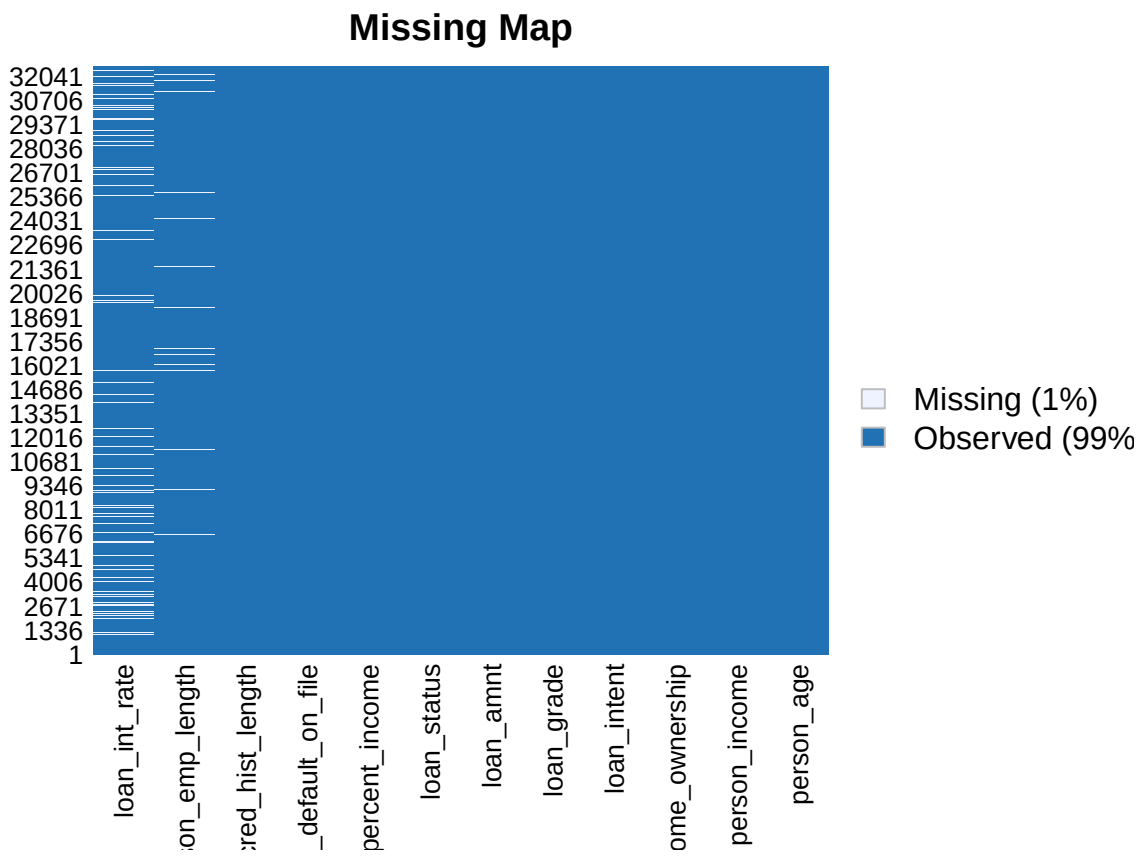
```
summary(aggr(fin_credrisk, numbers = T, sortVar = T))
```




```
##
##  Variables sorted by number of missings:
##                    Variable      Count
##               loan_int_rate 0.09563856
##           person_emp_length 0.02747000
##                  person_age 0.00000000
```

```
##               person_income 0.00000000
##       person_home_ownership 0.00000000
##                loan_intent 0.00000000
##                 loan_grade 0.00000000
##                  loan_amnt 0.00000000
##                loan_status 0.00000000
##         loan_percent_income 0.00000000
##   cb_person_default_on_file 0.00000000
##  cb_person_cred_hist_length 0.00000000

##
##  Missings per variable:
##                    Variable Count
##                  person_age     0
##               person_income     0
##       person_home_ownership     0
##           person_emp_length   895
##                 loan_intent     0
##                  loan_grade     0
##                   loan_amnt     0
##               loan_int_rate  3116
##                 loan_status     0
##         loan_percent_income     0
##   cb_person_default_on_file     0
##  cb_person_cred_hist_length     0
##
##  Missings in combinations of variables:
##             Combinations Count    Percent
##  0:0:0:0:0:0:0:0:0:0:0 28638 87.8978546
##  0:0:0:0:0:0:1:0:0:0:0  3048  9.3551456
##  0:0:0:1:0:0:0:0:0:0:0   827  2.5382892
##  0:0:0:1:0:0:0:1:0:0:0    68  0.2087106
```

```
missmap(fin_credrisk, main = "Missing Map")
```



4.1.2. Tratamiento e imputación de datos faltantes

Para realizar la imputación de datos faltantes en las columnas "person_emp_length" y "loan_int_rate", se ha decidido reemplazar todos sus NAs según los valores medianos de las mismas variables a las que se referencian.

Con la función summary se comprueba que ya no hay más datos faltantes en el data set.

```
fin_credrisk$person_emp_length[is.na(fin_credrisk$person_emp_length)] <- median(fin_credrisk$person_emp
    na.rm = TRUE)
fin_credrisk$loan_int_rate[is.na(fin_credrisk$loan_int_rate)] <- median(fin_credrisk$loan_int_rate,
    na.rm = TRUE)

summary(fin_credrisk)
```

```
##    person_age     person_income     person_home_ownership person_emp_length
##  Min.   : 20.00   Min.   :   4000   MORTGAGE:13444        Min.   :  0.000
##  1st Qu.: 23.00   1st Qu.:  38500   OTHER   :  107        1st Qu.:  2.000
##  Median : 26.00   Median :  55000   OWN     : 2584        Median :  4.000
##  Mean   : 27.73   Mean   :  66075   RENT    :16446        Mean   :  4.768
##  3rd Qu.: 30.00   3rd Qu.:  79200                         3rd Qu.:  7.000
##  Max.   :144.00   Max.   :6000000                         Max.   :123.000
##
##             loan_intent   loan_grade   loan_amnt     loan_int_rate
##  DEBTCONSOLIDATION:5212   A:10777    Min.   :  500   Min.   : 5.42
```

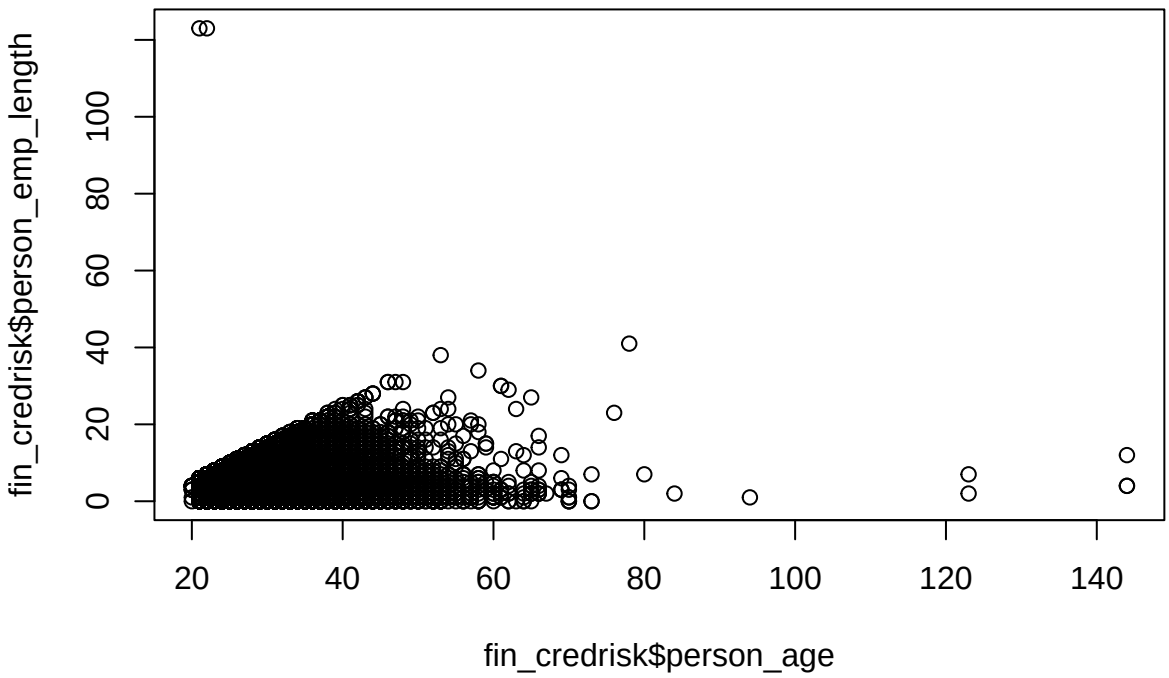
```
##  EDUCATION        :6453   B:10451    1st Qu.: 5000   1st Qu.: 8.49
##  HOMEIMPROVEMENT  :3605   C: 6458    Median : 8000   Median :10.99
##  MEDICAL          :6071   D: 3626    Mean   : 9589   Mean   :11.01
##  PERSONAL         :5521   E:  964    3rd Qu.:12200   3rd Qu.:13.11
##  VENTURE          :5719   F:  241    Max.   :35000   Max.   :23.22
##                           G:   64
##   loan_status     loan_percent_income cb_person_default_on_file
##  Min.   :0.0000   Min.   :0.0000      N:26836
##  1st Qu.:0.0000   1st Qu.:0.0900      Y: 5745
##  Median :0.0000   Median :0.1500
##  Mean   :0.2182   Mean   :0.1702
##  3rd Qu.:0.0000   3rd Qu.:0.2300
##  Max.   :1.0000   Max.   :0.8300
##
##  cb_person_cred_hist_length
##  Min.   : 2.000
##  1st Qu.: 3.000
##  Median : 4.000
##  Mean   : 5.804
##  3rd Qu.: 8.000
##  Max.   :30.000
##
```

4.2. Análisis de datos atípico o outliers

Se pasa a realizar un análisis en mayor profundidad de lo detectado anteriomente respecto a datos atípico en el dataset:

- La variable “person_age” tiene un valor máximo de 144 años.
- La variable “person_emp_length” tiene un valor máximo de 123 años.

```
plot(fin_credrisk$person_age, fin_credrisk$person_emp_length)
```



Vemos como hay algunos datos que si podemos considerar atípico. Hay 2 registros con un historial laboral de la persona contratante del préstamo ("person_emp_length") de más de 120 años y 5 registros con una edad de la persona contratante del préstamo ("person_age") superior a 120 años.

4.2.2. Tratamiento y eliminación de datos atípicos o outliers

Estos caso comentados son raros e ilógicos, y se opta finalmente por eliminarlos del dataset.

```
fin_credrisk <- fin_credrisk %>%
    filter(person_age < 120) %>%
    filter(person_emp_length < 120)
```

Se muestra el resultado de como queda ahora el dataset con esta modificación. Ya no existen ni atípicos ni datos faltantes en el dataset. Ahora el dateset mantiene las 12 columnas, pero pasamos a tener 32.574 filas o registros.

```
summary(fin_credrisk)
```

```
##    person_age    person_income     person_home_ownership person_emp_length
##  Min.   :20.00   Min.   :   4000   MORTGAGE:13441        Min.   : 0.000
##  1st Qu.:23.00   1st Qu.:  38500   OTHER   :  107        1st Qu.: 2.000
##  Median :26.00   Median :  55000   OWN     : 2584        Median : 4.000
##  Mean   :27.72   Mean   :  65879   RENT    :16442        Mean   : 4.761
##  3rd Qu.:30.00   3rd Qu.:  79200                         3rd Qu.: 7.000
##  Max.   :94.00   Max.   :2039784                         Max.   :41.000
```

```
##
##             loan_intent   loan_grade   loan_amnt     loan_int_rate
##  DEBTCONSOLIDATION:5212   A:10776    Min.   :  500   Min.   : 5.42
##  EDUCATION        :6451   B:10448    1st Qu.: 5000   1st Qu.: 8.49
##  HOMEIMPROVEMENT  :3605   C: 6456    Median : 8000   Median :10.99
##  MEDICAL          :6071   D: 3625    Mean   : 9588   Mean   :11.01
##  PERSONAL         :5519   E:  964    3rd Qu.:12200   3rd Qu.:13.11
##  VENTURE          :5716   F:  241    Max.   :35000   Max.   :23.22
##                           G:   64
##   loan_status     loan_percent_income cb_person_default_on_file
##  Min.   :0.0000   Min.   :0.0000      N:26830
##  1st Qu.:0.0000   1st Qu.:0.0900      Y: 5744
##  Median :0.0000   Median :0.1500
##  Mean   :0.2182   Mean   :0.1702
##  3rd Qu.:0.0000   3rd Qu.:0.2300
##  Max.   :1.0000   Max.   :0.8300
##
##  cb_person_cred_hist_length
##  Min.   : 2.000
##  1st Qu.: 3.000
##  Median : 4.000
##  Mean   : 5.804
##  3rd Qu.: 8.000
##  Max.   :30.000
##
```

```
count(fin_credrisk)
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1 32574
```

```
plot(fin_credrisk$person_age, fin_credrisk$person_emp_length)
```

5. Partición del conjunto de datos: data set training y data set test

Una vez vista por encima la estructura general de los datos, se procede a dividir el conjunto de datos en dos para diferenciar los que se usarán de entrenamiento de los que usarán de test (viendo la cantidad de datos de los que se dispone, la distribución elegida ha sido: 20% test y 80% training). Además, se establece una semilla que guarde de forma permanente la división realizada para que la distribución de los datos sea siempre la misma y no sufra variaciones.

Se procede a guardar también la partición de datos de test para ser utilizada a futuro para la validación del modelo final. De aquí en adelante, se pasa a trabajar con la partición de training.

```
set.seed(101)

sample <- sample.int(n = nrow(fin_credrisk), size = floor(0.8 *
    nrow(fin_credrisk)), replace = F)
fcr_train <- fin_credrisk[sample, ]
fcr_test <- fin_credrisk[-sample, ]
fcr_train
```

```
## # A tibble: 26,059 x 12
##    person_age person_i~1 perso~2 perso~3 loan_~4 loan_~5 loan_~6 loan_~7 loan_~8
##         <dbl>      <dbl> <fct>     <dbl> <fct>   <fct>     <dbl>   <dbl>   <dbl>
##  1         28      44000 RENT          2 MEDICAL C         10000   13.5        1
##  2         21      35000 OWN           5 VENTURE B          8000    9.91       0
```

```
##  3         25      96000 MORTGA~       6 HOMEIM~ C         21000   14.6        0
##  4         22      67000 OWN           5 EDUCAT~ D          7500   16.3        0
##  5         24      52800 RENT          8 PERSON~ A          9000    7.49       0
##  6         27      50004 RENT         12 DEBTCO~ B          3200   11.5        0
##  7         23      55488 RENT          4 MEDICAL D          5000   15.2        1
##  8         28      70000 RENT          2 DEBTCO~ B          6000   10.4        0
##  9         22      55000 MORTGA~       6 PERSON~ C         13000   13.8        0
## 10         26      43200 RENT          5 EDUCAT~ C          3200   14.4        0
## # ... with 26,049 more rows, 3 more variables: loan_percent_income <dbl>,
## #   cb_person_default_on_file <fct>, cb_person_cred_hist_length <dbl>, and
## #   abbreviated variable names 1: person_income, 2: person_home_ownership,
## #   3: person_emp_length, 4: loan_intent, 5: loan_grade, 6: loan_amnt,
## #   7: loan_int_rate, 8: loan_status
```

```
fcr_test
```

```
## # A tibble: 6,515 x 12
##    person_age person_i~1 perso~2 perso~3 loan_~4 loan_~5 loan_~6 loan_~7 loan_~8
##         <dbl>      <dbl> <fct>     <dbl> <fct>   <fct>     <dbl>   <dbl>   <dbl>
##  1         23      65500 RENT          4 MEDICAL C         35000   15.2        1
##  2         21       9900 OWN           2 VENTURE A          2500    7.14       1
##  3         23      95000 RENT          2 VENTURE A         35000    7.9        1
##  4         23     500000 MORTGA~       7 DEBTCO~ B         30000   10.6        0
##  5         25     137000 RENT          9 PERSON~ E         34800   16.8        0
##  6         24      10980 OWN           0 PERSON~ A          1500    7.29       0
##  7         22      48000 RENT          1 EDUCAT~ E         30000   18.4        1
##  8         24      12000 OWN           4 VENTURE B          2500   12.7        1
##  9         23     300000 OWN           1 EDUCAT~ F         24250   19.4        0
## 10         23      78000 RENT          7 DEBTCO~ F         30000   18.6        1
## # ... with 6,505 more rows, 3 more variables: loan_percent_income <dbl>,
## #   cb_person_default_on_file <fct>, cb_person_cred_hist_length <dbl>, and
## #   abbreviated variable names 1: person_income, 2: person_home_ownership,
## #   3: person_emp_length, 4: loan_intent, 5: loan_grade, 6: loan_amnt,
## #   7: loan_int_rate, 8: loan_status
```

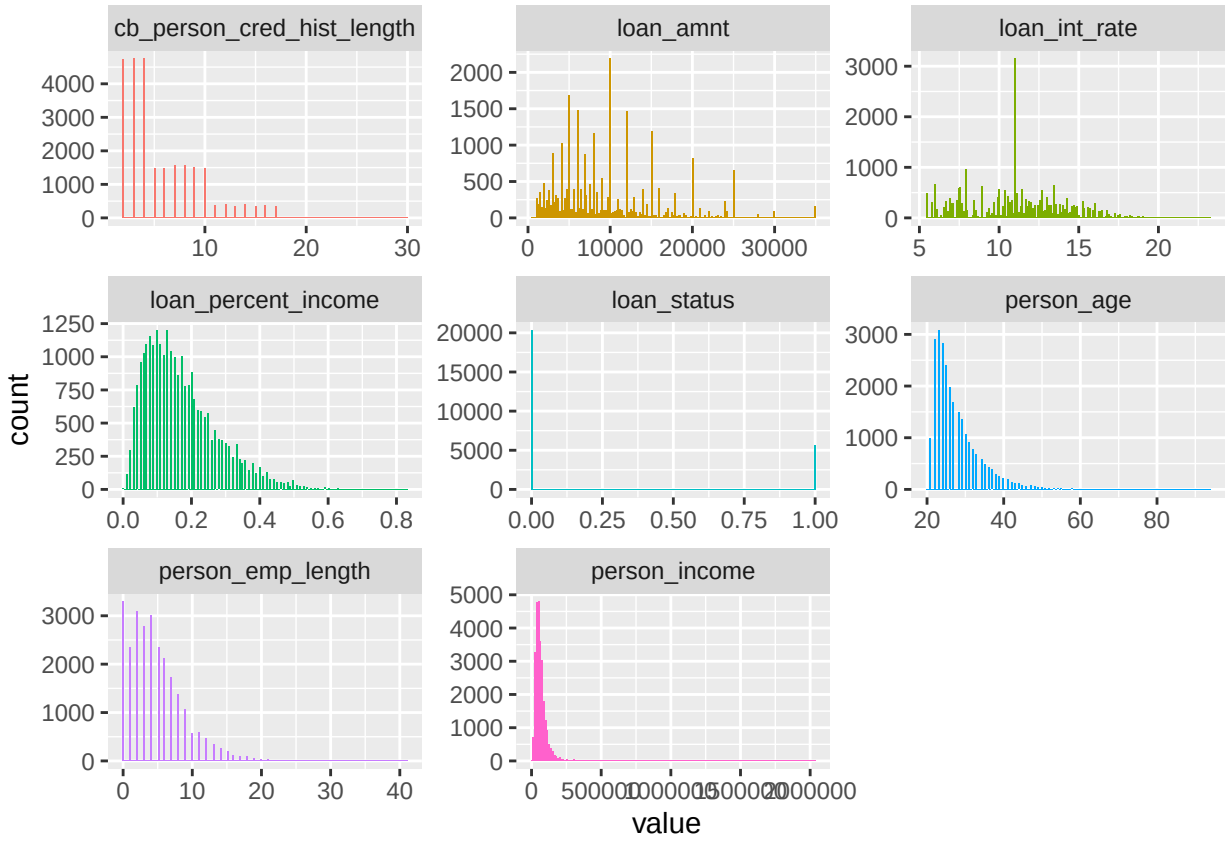
ANÁLISIS EN PROFUNDIDAD DE LOS DATOS

6. EDA - Análisis exploratorio de datos

6.1. Análisis de la distribución de las variables

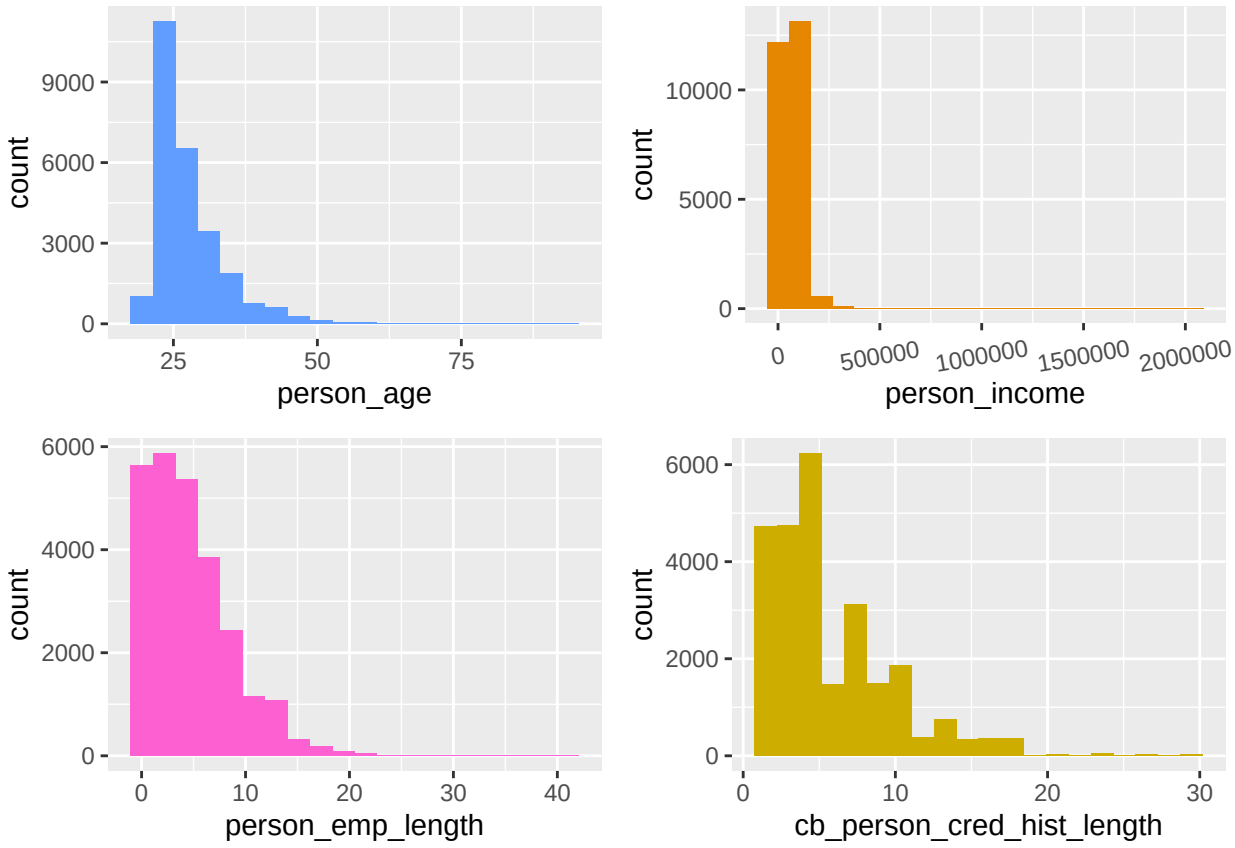
Se analizan como se distribuyen las diferentes variables de entrada del dataset:

```
fcr_train %>%
    keep(is.numeric) %>%
    gather() %>%
    ggplot(aes(value, fill = key)) + facet_wrap(~key, scales = "free") +
    geom_histogram(bins = sqrt(nrow(fcr_train))) + theme(legend.position = "none")
```



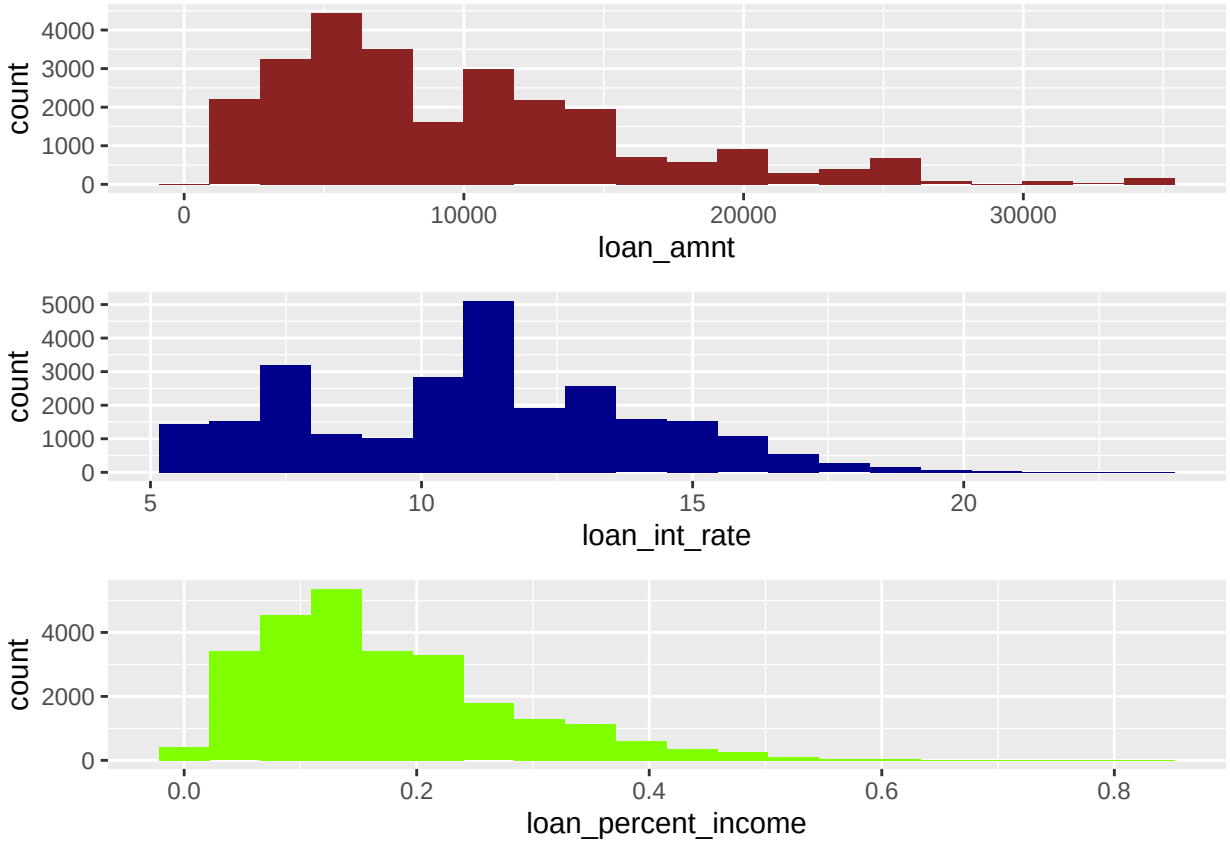
- **Variables numéricas relativas a la persona**

```
ga <- fcr_train %>%
    ggplot(aes(x = person_age)) + geom_histogram(bins = 20, fill = "#619CFF")

gb <- fcr_train %>%
    ggplot(aes(x = person_income)) + geom_histogram(bins = 20,
    fill = "#E58700") + theme(axis.text.x = element_text(angle = 10,
    hjust = 0.5, vjust = 0.5))

gd <- fcr_train %>%
    ggplot(aes(x = person_emp_length)) + geom_histogram(bins = 20,
    fill = "#FD61D1")

gl <- fcr_train %>%
    ggplot(aes(x = cb_person_cred_hist_length)) + geom_histogram(bins = 20,
    fill = "#CDAD00")

grid.arrange(ga, gb, gd, gl)
```



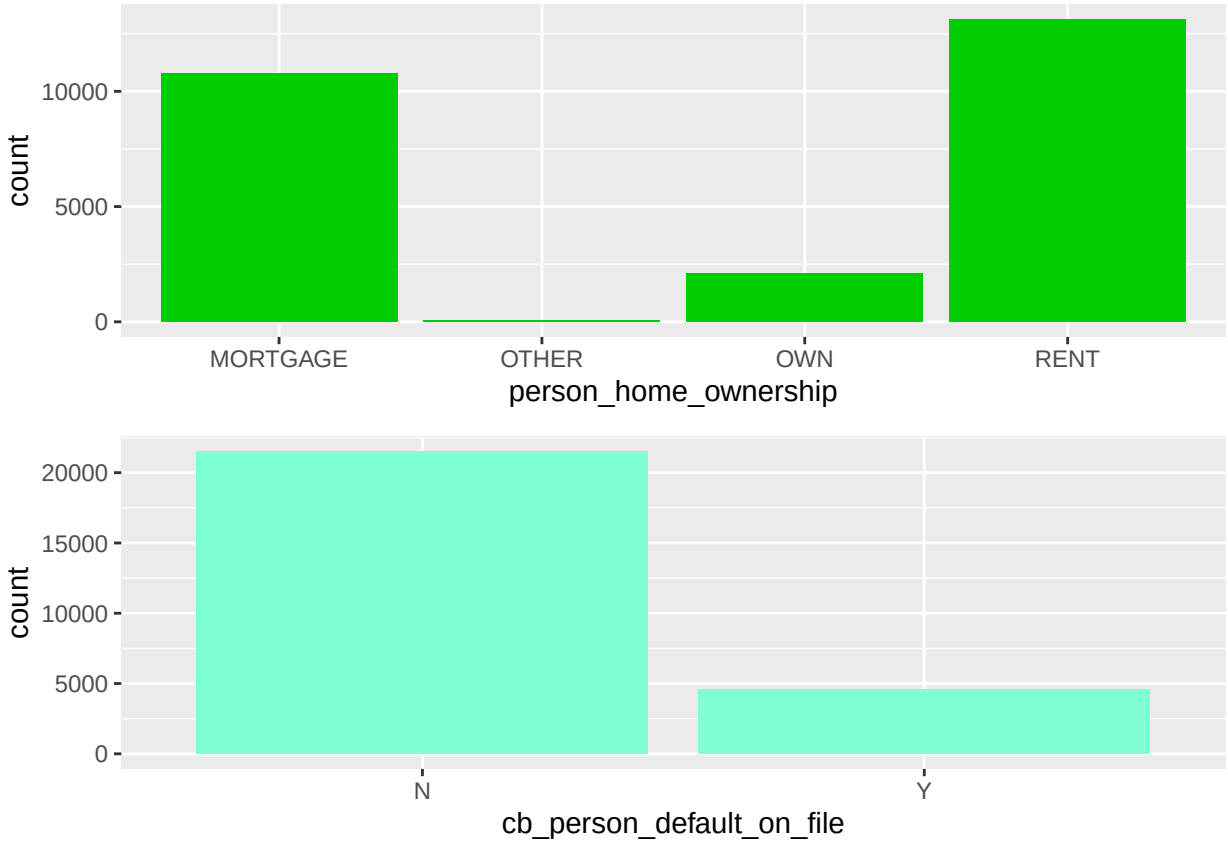
- **Variables numéricas relativas al préstamo**

```
gg <- fcr_train %>%
    ggplot(aes(x = loan_amnt)) + geom_histogram(bins = 20, fill = "#8B2323")

gh <- fcr_train %>%
    ggplot(aes(x = loan_int_rate)) + geom_histogram(bins = 20,
    fill = "#00008B")

gj <- fcr_train %>%
    ggplot(aes(x = loan_percent_income)) + geom_histogram(bins = 20,
    fill = "#7FFF00")

grid.arrange(gg, gh, gj)
```



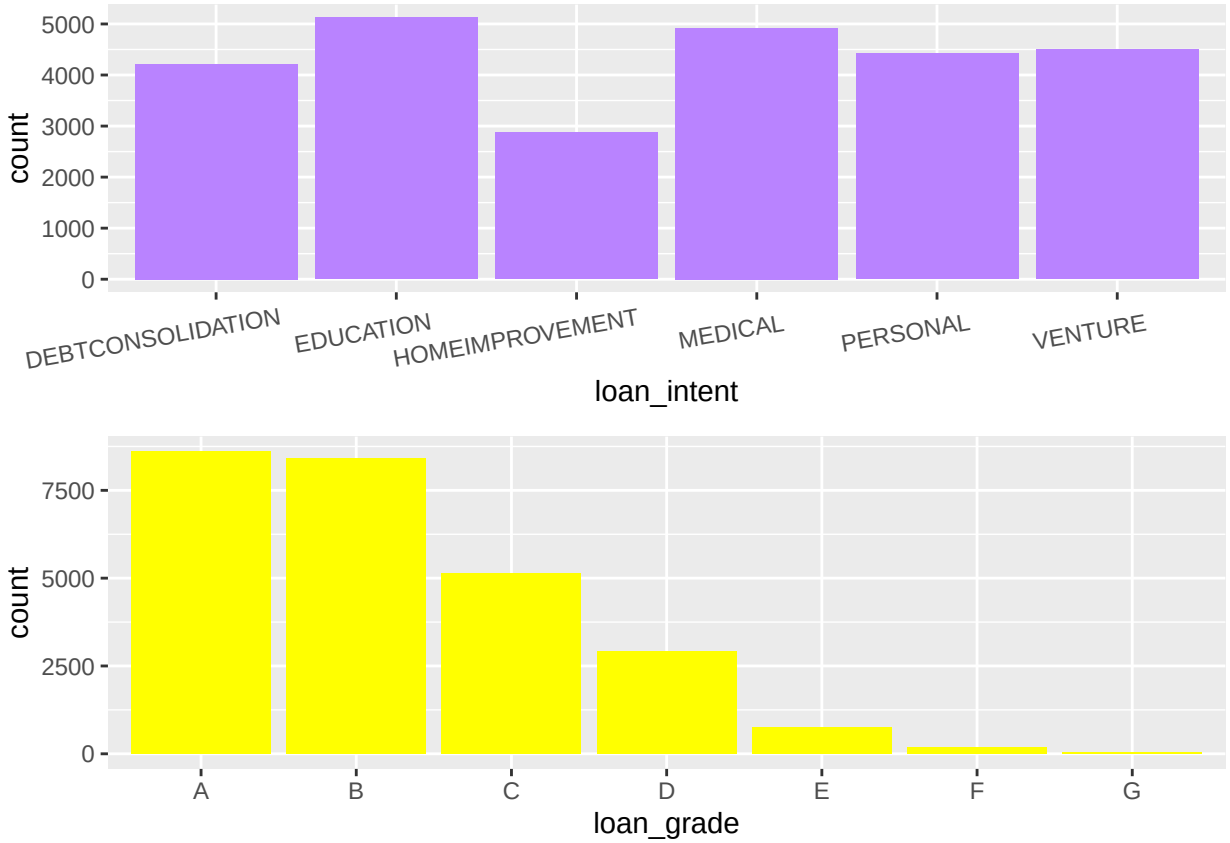
- **Variables categóricas relativas a la persona**

```
gc <- fcr_train %>%
    ggplot(aes(x = person_home_ownership)) + geom_histogram(bins = 20,
    fill = "#00CD00", stat = "count")

gk <- fcr_train %>%
    ggplot(aes(x = cb_person_default_on_file)) + geom_histogram(bins = 20,
    fill = "#7FFFD4", stat = "count")

grid.arrange(gc, gk)
```



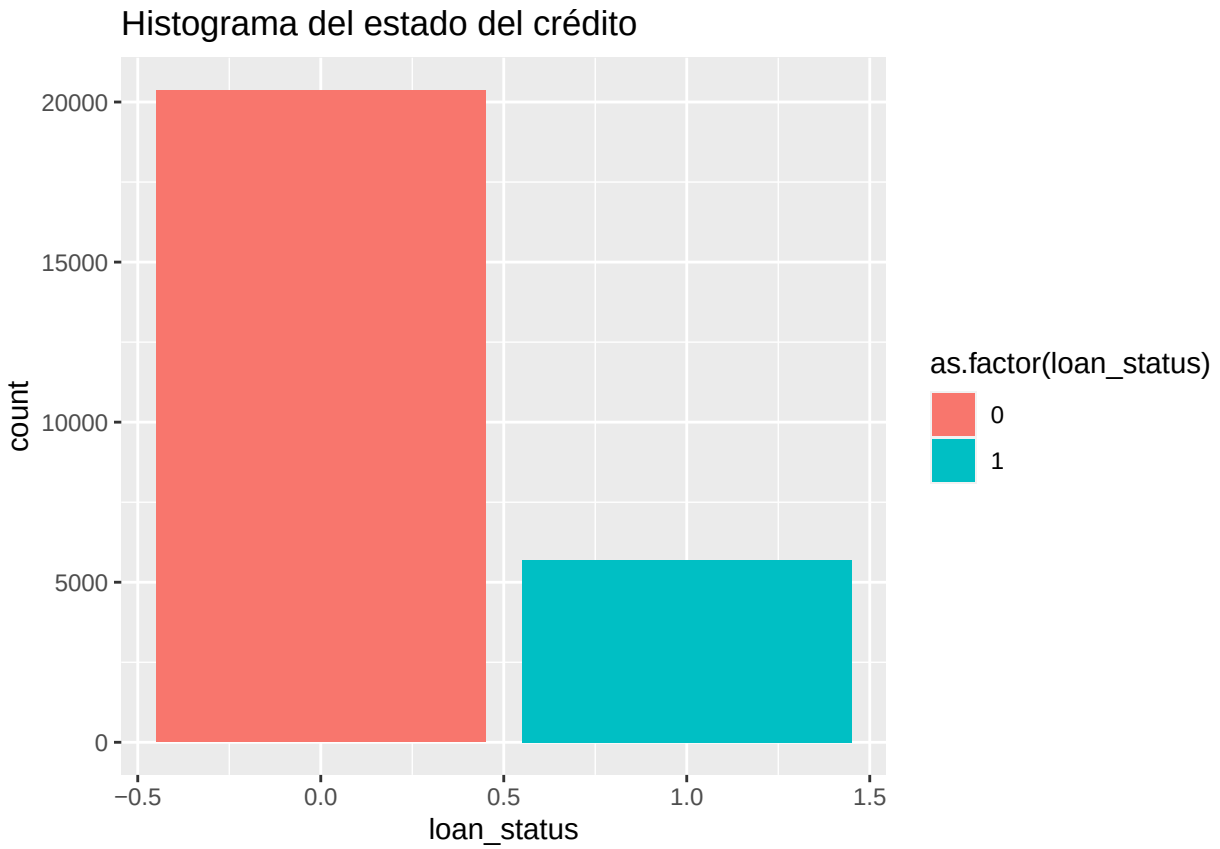
- **Variables categóricas relativas al préstamo**

```
ge <- fcr_train %>%
    ggplot(aes(x = loan_intent)) + geom_histogram(bins = 20,
    fill = "#B983FF", stat = "count") + theme(axis.text.x = element_text(angle = 10,
    hjust = 0.75, vjust = 0.75))

gf <- fcr_train %>%
    ggplot(aes(x = loan_grade)) + geom_histogram(bins = 20, fill = "#FFFF00",
    stat = "count")

grid.arrange(ge, gf)
```



Se analiza como se distribuye la variable objetivo del dataset:

```
ggplot(data = fcr_train) + geom_bar(mapping = aes(x = loan_status,
    fill = as.factor(loan_status))) + labs(title = "Histograma del estado del crédito")
```




```
table(fcr_train$loan_status)
```

```
##
##     0     1
## 20365  5694
```

```
prop.table(table(fcr_train$loan_status))
```

```
##
##         0         1
## 0.7814958 0.2185042
```

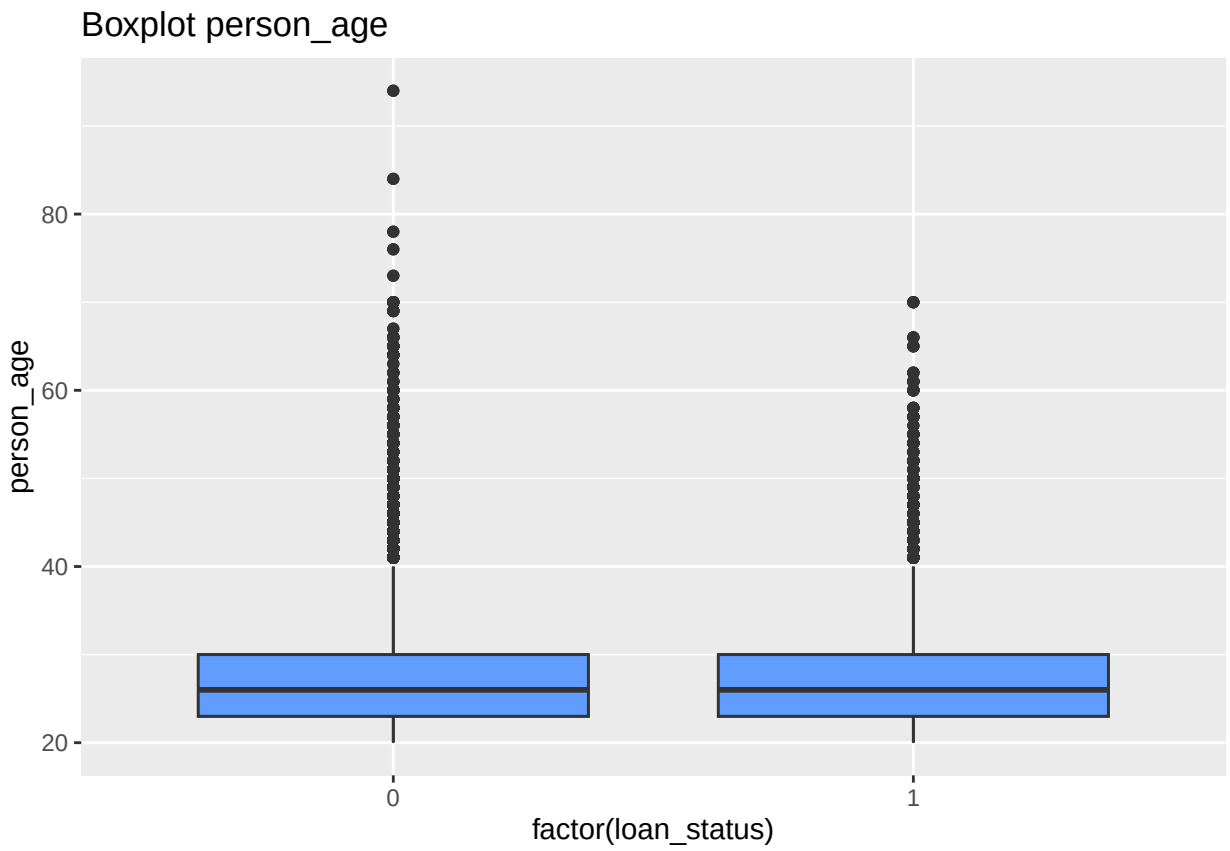
El 78.15% de los registros en el dataset de train (20.365 registros) tiene valor 0 correspondiente a créditos que no han entrado en default, y el 21.85% (5.694 registros) tienen valor 1 correspondiente a créditos que por el contrario si han sido impagados.

6.2. Boxplot - análisis de la variables de relevancia y de los atípicos observados

Se analiza si las variables tienen valores atípicos, cuales son sus valores medios y se ven sus intervalos de confianza, a través de gráficos de tipo Boxplot.

- **Boxplot variable person_age**

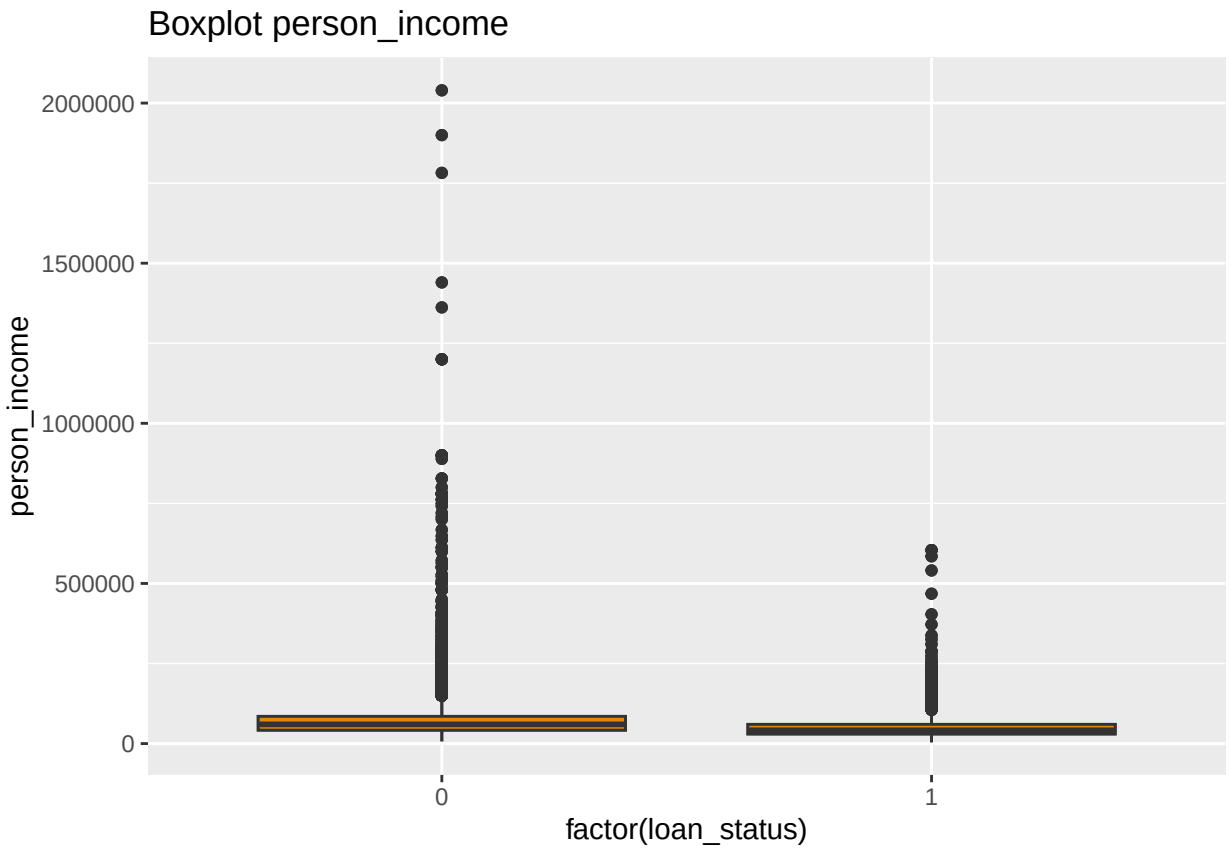
```
BoxPlot_person_age <- ggplot(fcr_train, aes(x = factor(loan_status),
    y = person_age)) + geom_boxplot() + geom_boxplot(fill = "#619CFF") +
    ggtitle("Boxplot person_age")
BoxPlot_person_age
```



Se aprecia como la variable "person_age" relativa a la edad del contratante del préstamo se mantiene bastante igual en los casos donde hay default o no hay default.

- **Boxplot variable person_income**

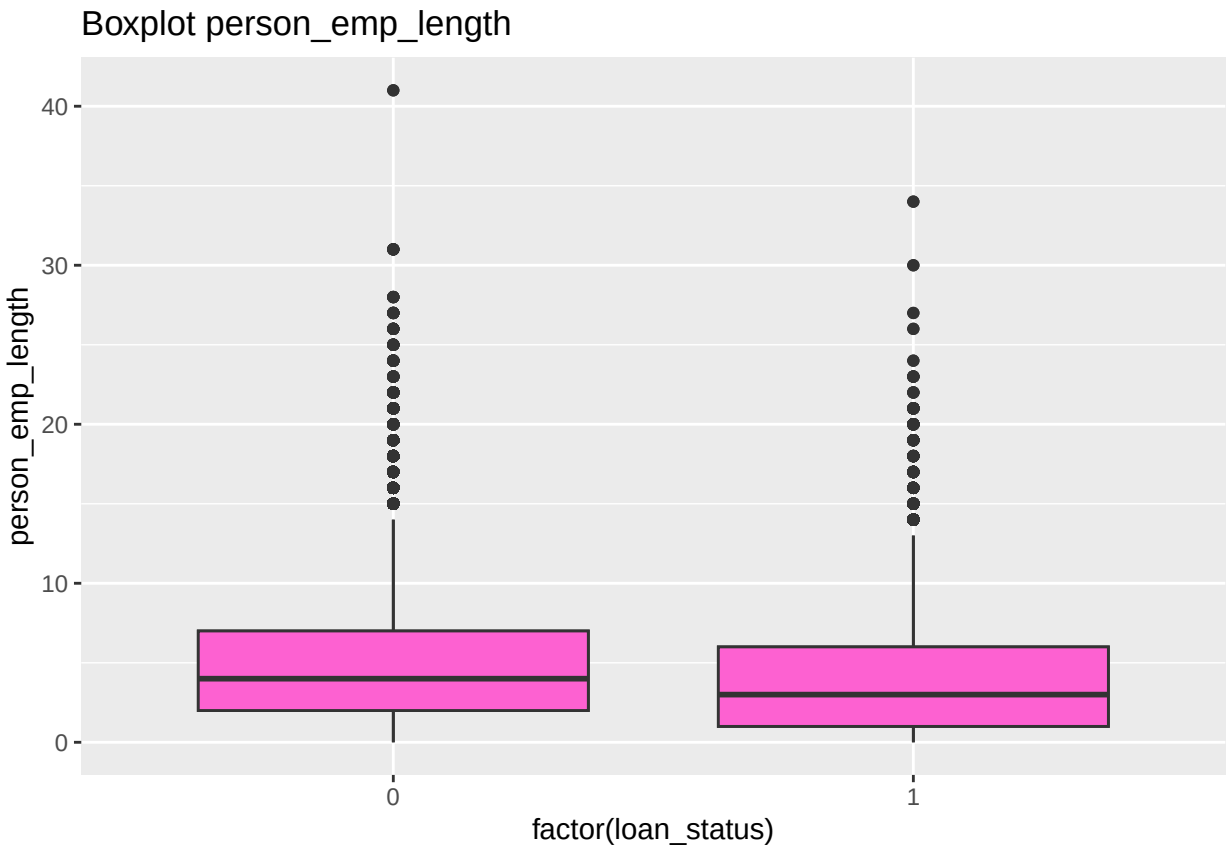
```
BoxPlot_person_income <- ggplot(fcr_train, aes(x = factor(loan_status),
    y = person_income)) + geom_boxplot() + geom_boxplot(fill = "#E58700") +
    ggtitle("Boxplot person_income")
BoxPlot_person_income
```



Se aprecia ligeramente como la variable "person_income" relativa a los ingresos anuales del contratante del préstamo, es inferior en media en los casos donde se da el impago del crédito.

- **Boxplot variable person_emp_length**

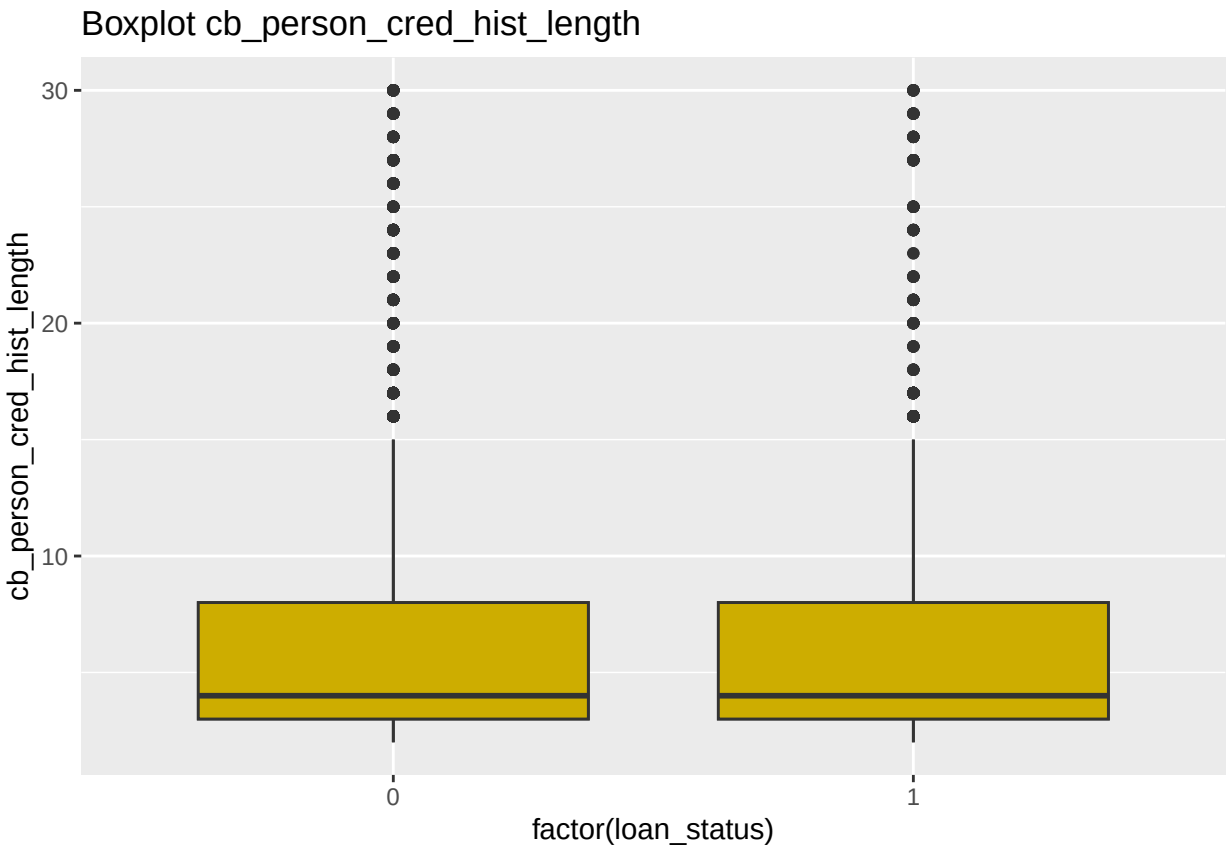
```
BoxPlot_person_emp_length <- ggplot(fcr_train, aes(x = factor(loan_status),
    y = person_emp_length)) + geom_boxplot() + geom_boxplot(fill = "#FD61D1") +
    ggtitle("Boxplot person_emp_length")
BoxPlot_person_emp_length
```



Se aprecia ligeramente como la variable "person_emp_length" relativa al periodo de tiempo en años desde que la persona que toma el crédito está en situación laboral activa, es inferior en media en los casos donde se da el impago del crédito.

- **Boxplot variable cb_person_cred_hist_length**

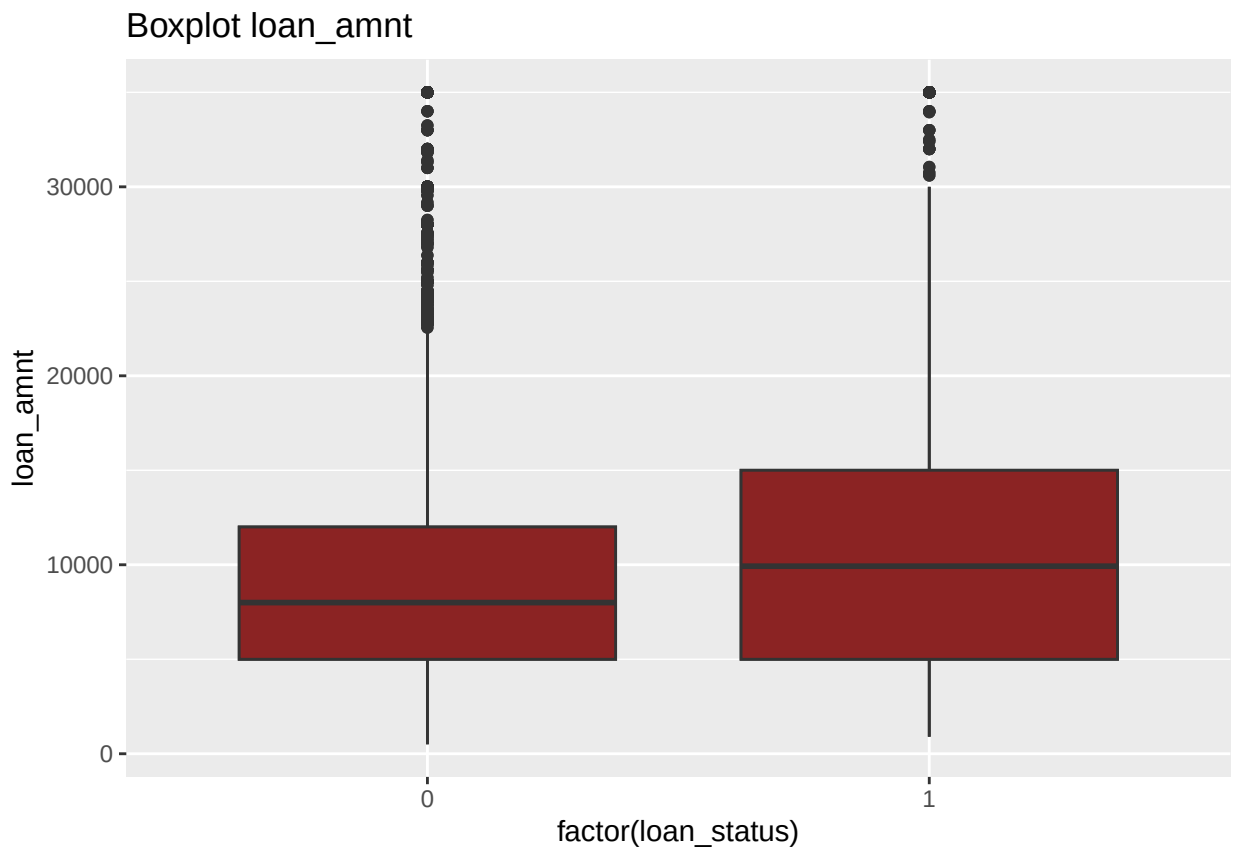
```
BoxPlot_cb_person_cred_hist_length <- ggplot(fcr_train, aes(x = factor(loan_status),
    y = cb_person_cred_hist_length)) + geom_boxplot() + geom_boxplot(fill = "#CDAD00") +
    ggtitle("Boxplot cb_person_cred_hist_length")
BoxPlot_cb_person_cred_hist_length
```



Se aprecia como la variable "person_cred_hist_length" relativa a la duración en años del historial crediticio de la persona tomadora del crédito, se mantiene bastante igual en los casos donde hay default o no hay default.

- **Boxplot variable loan_amnt**

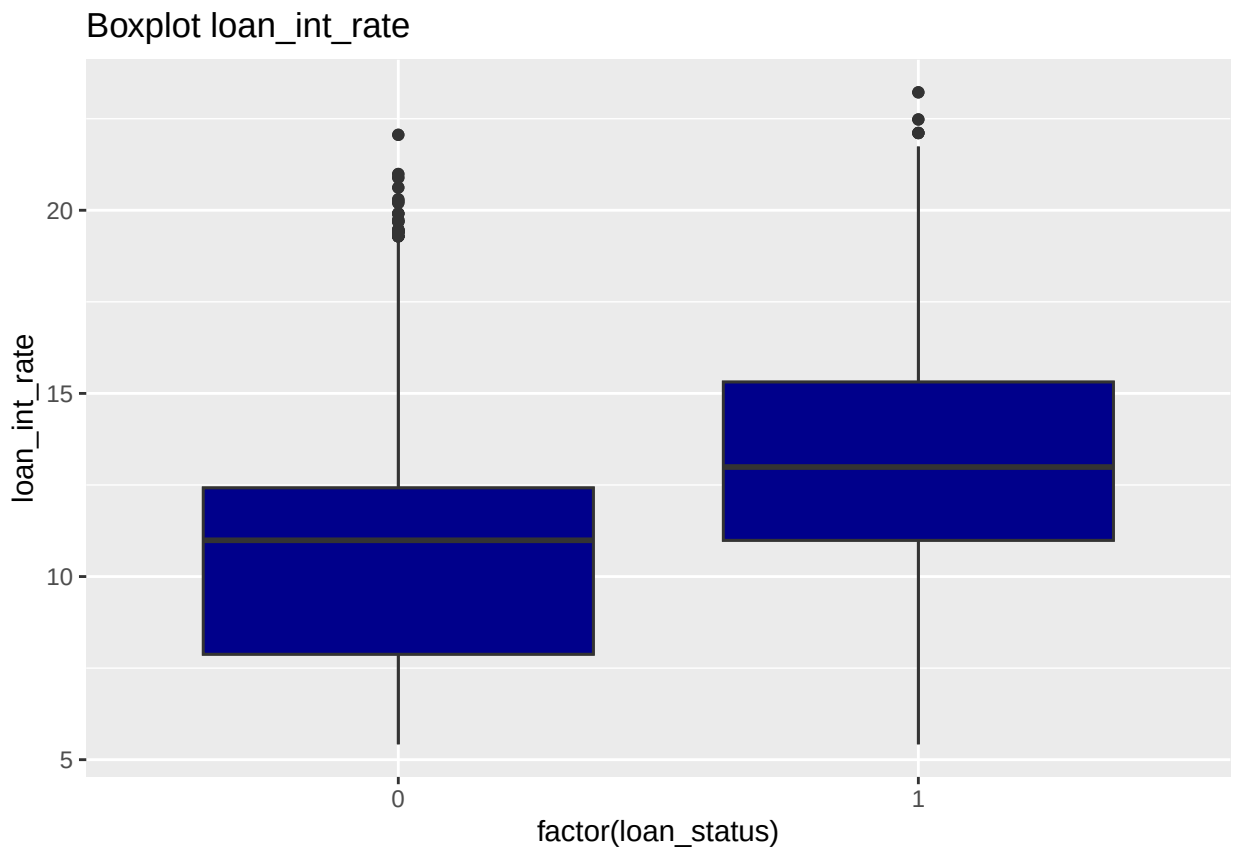
```
BoxPlot_loan_amnt <- ggplot(fcr_train, aes(x = factor(loan_status),
    y = loan_amnt)) + geom_boxplot() + geom_boxplot(fill = "#8B2323") +
    ggtitle("Boxplot loan_amnt")
BoxPlot_loan_amnt
```



Se aprecia como la variable "loan_amount" relativa a la cantidad en dólares de crédito concedido, es superior en media en los casos donde se da el impago del crédito.

- **Boxplot variable loan_int_rate**

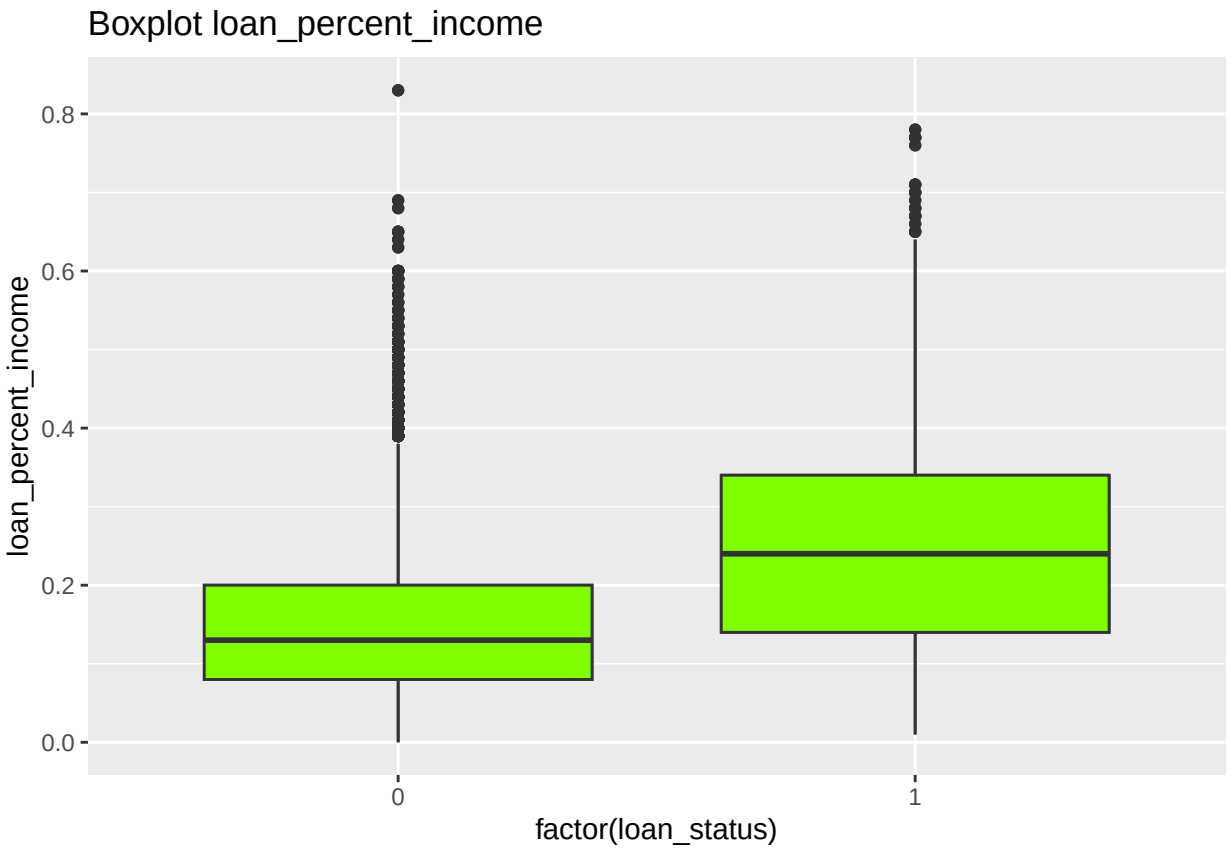
```
BoxPlot_loan_int_rate <- ggplot(fcr_train, aes(x = factor(loan_status),
    y = loan_int_rate)) + geom_boxplot() + geom_boxplot(fill = "#00008B") +
    ggtitle("Boxplot loan_int_rate")
BoxPlot_loan_int_rate
```



Se aprecia como la variable "loan_int_rate" relativa al tipo de interés en porcentaje del crédito concedido, es superior en media en los casos donde se da el impago del crédito.

- **Boxplot variable loan_percent_income**

```
BoxPlot_loan_percent_income <- ggplot(fcr_train, aes(x = factor(loan_status),
    y = loan_percent_income)) + geom_boxplot() + geom_boxplot(fill = "#7FFF00") +
    ggtitle("Boxplot loan_percent_income")
BoxPlot_loan_percent_income
```



Se aprecia como la variable "loan_percent_income" relativa al porcentaje de lo que supone el préstamo sobre los ingresos anuales en dólares de la persona que toma el crédito, es superior en media en los casos donde se da el impago del crédito.

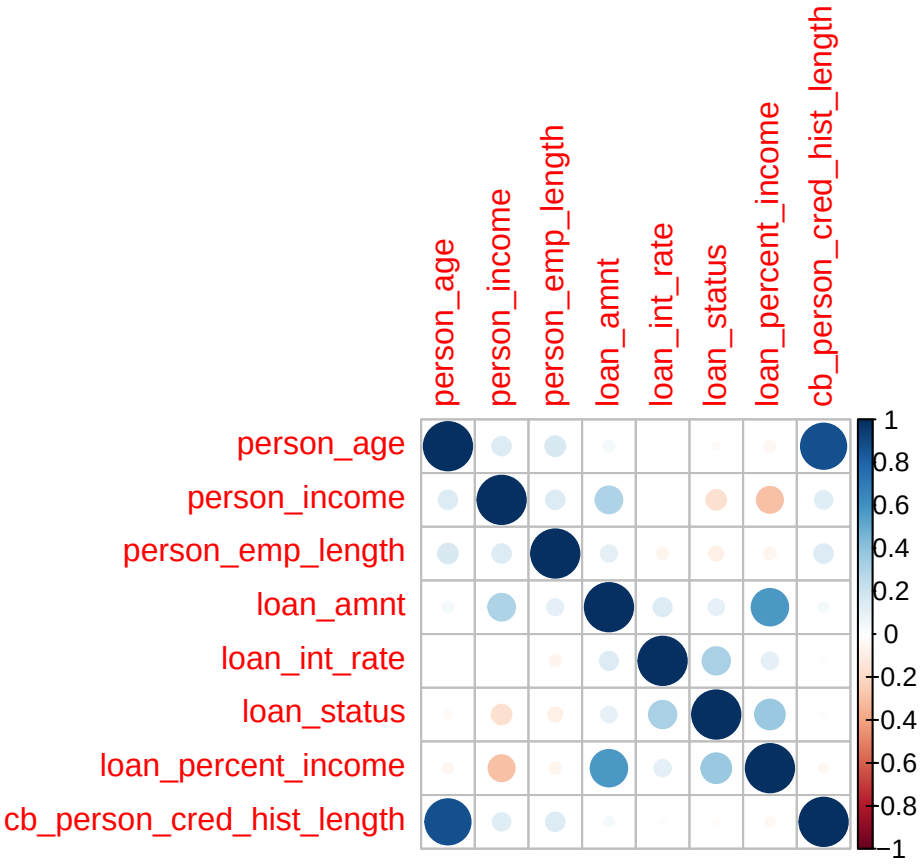
Como conclusión lógica de todo esto, normalmente el la situación de impago es más frecuente en personas con ingresos más bajos, menor número de años de actividad laboral y con préstamos de cantidades más altas y con tipos de interés más elevados.

6.3. Correlación entre variables

Continuando con en análisis de las distintas variables del dataset y el estudio de como se relacionan entre si, se quiere analizar de forma global como se correlacionan las variables numéricas que nos pueden llegar a servir para el modelo de predicción objetivo.

6.3.1. Análisis de la correlación global del conjunto de variables

```
corrplot(cor(fcr_train %>%
    mutate(loan_status = as.numeric(loan_status)) %>%
    keep(is.numeric)))
```




```
res <- cor(fcr_train %>%
    mutate(loan_status = as.numeric(loan_status)) %>%
    keep(is.numeric))
round(res, 2)
```

```
##                             person_age person_income person_emp_length loan_amnt
## person_age                        1.00          0.15              0.17      0.05
## person_income                     0.15          1.00              0.15      0.31
## person_emp_length                 0.17          0.15              1.00      0.11
## loan_amnt                         0.05          0.31              0.11      1.00
## loan_int_rate                     0.01         -0.01             -0.05      0.14
## loan_status                      -0.02         -0.17             -0.09      0.11
## loan_percent_income              -0.05         -0.29             -0.06      0.57
## cb_person_cred_hist_length        0.88          0.13              0.15      0.04
##                             loan_int_rate loan_status loan_percent_income
## person_age                           0.01       -0.02                -0.05
## person_income                       -0.01       -0.17                -0.29
## person_emp_length                   -0.05       -0.09                -0.06
## loan_amnt                            0.14        0.11                 0.57
## loan_int_rate                        1.00        0.32                 0.12
## loan_status                          0.32        1.00                 0.38
## loan_percent_income                  0.12        0.38                 1.00
## cb_person_cred_hist_length           0.02       -0.02                -0.04
##                             cb_person_cred_hist_length
## person_age                                        0.88
```

```
## person_income                          0.13
## person_emp_length                      0.15
## loan_amnt                              0.04
## loan_int_rate                          0.02
## loan_status                           -0.02
## loan_percent_income                   -0.04
## cb_person_cred_hist_length             1.00
```

Se aprecia que las variables que más estan correlacionadas con la variable respuesta "loan_status" son: "person_income", "loan_amnt", "loan_int_rate" y "loan_percent_income".

6.3.2. Análisis de la correlación bivariante

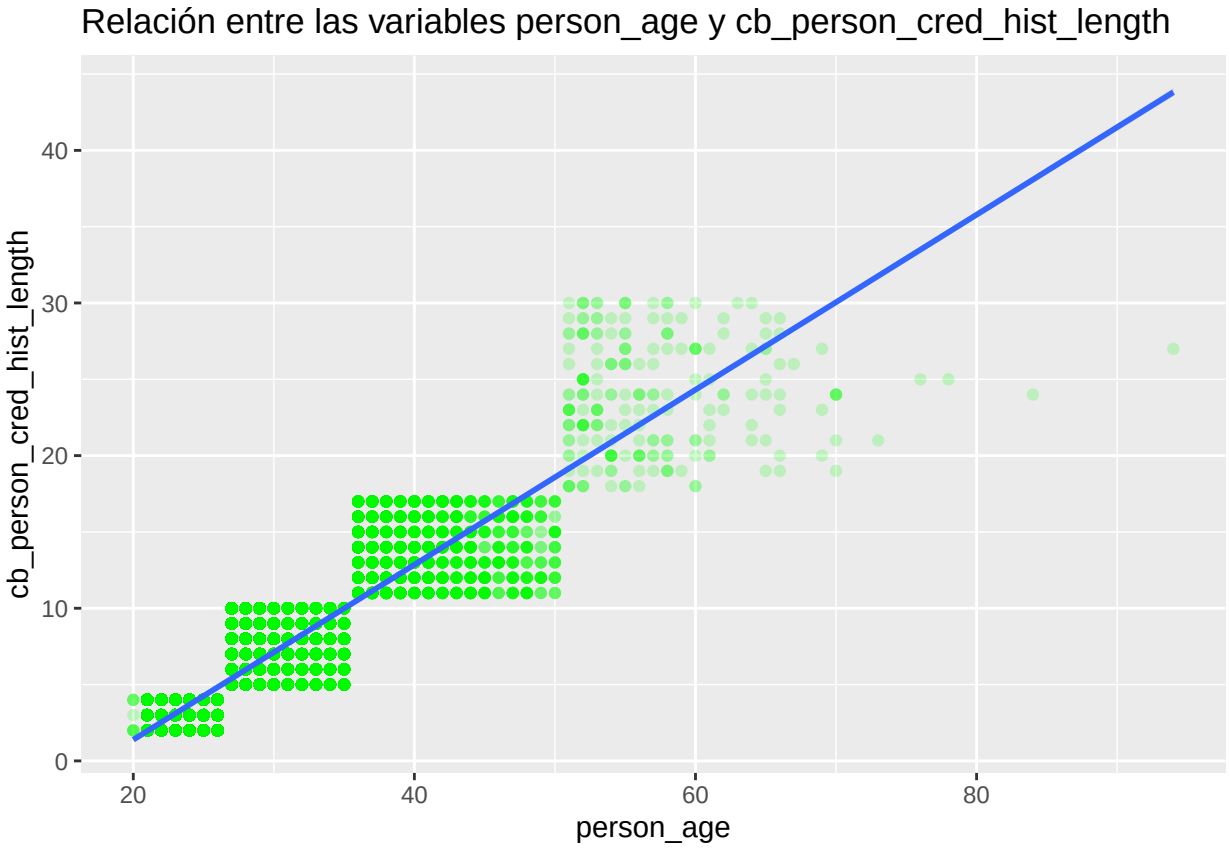
Se pasa a realizar un análisis bivariante para ver que variables están más correlacionadas, positiva o negativamente, entre si.

- **Correlación: person_age y cb_person_cred_hist_length:**

```
cor(x = fcr_train$person_age, y = fcr_train$cb_person_cred_hist_length)
```

```
## [1] 0.8775082
```

```
fcr_train %>%
    ggplot(aes(person_age, cb_person_cred_hist_length)) + geom_point(alpha = 0.2,
    colour = "green") + geom_smooth(formula = "y ~ x", method = "lm") +
    labs(title = "Relación entre las variables person_age y cb_person_cred_hist_length",
        x = "person_age", y = "cb_person_cred_hist_length")
```

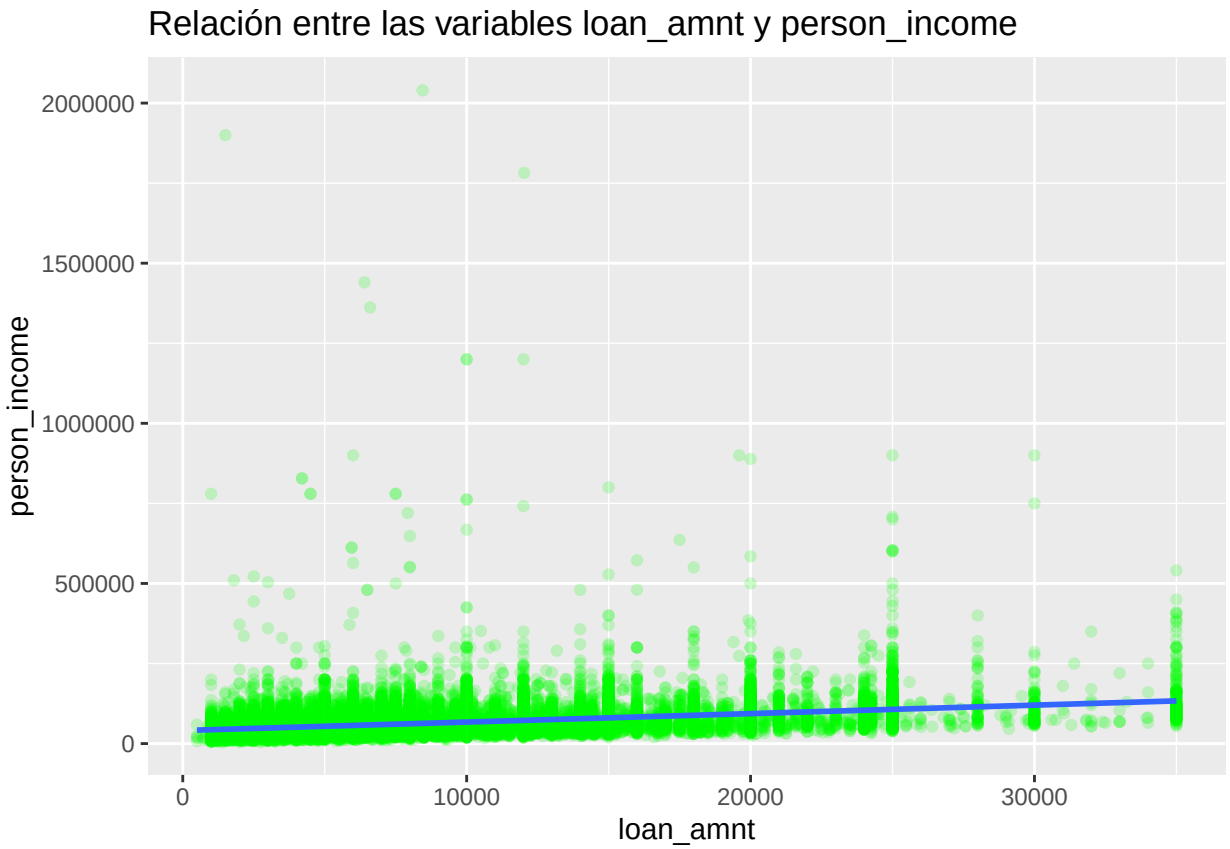


- **Correlación: loan_amnt y person_income:**

```
cor(x = fcr_train$loan_amnt, y = fcr_train$person_income)
```

```
## [1] 0.3084375
```

```
fcr_train %>%
    ggplot(aes(loan_amnt, person_income)) + geom_point(alpha = 0.2,
    colour = "green") + geom_smooth(formula = "y ~ x", method = "lm") +
    labs(title = "Relación entre las variables loan_amnt y person_income",
        x = "loan_amnt", y = "person_income")
```

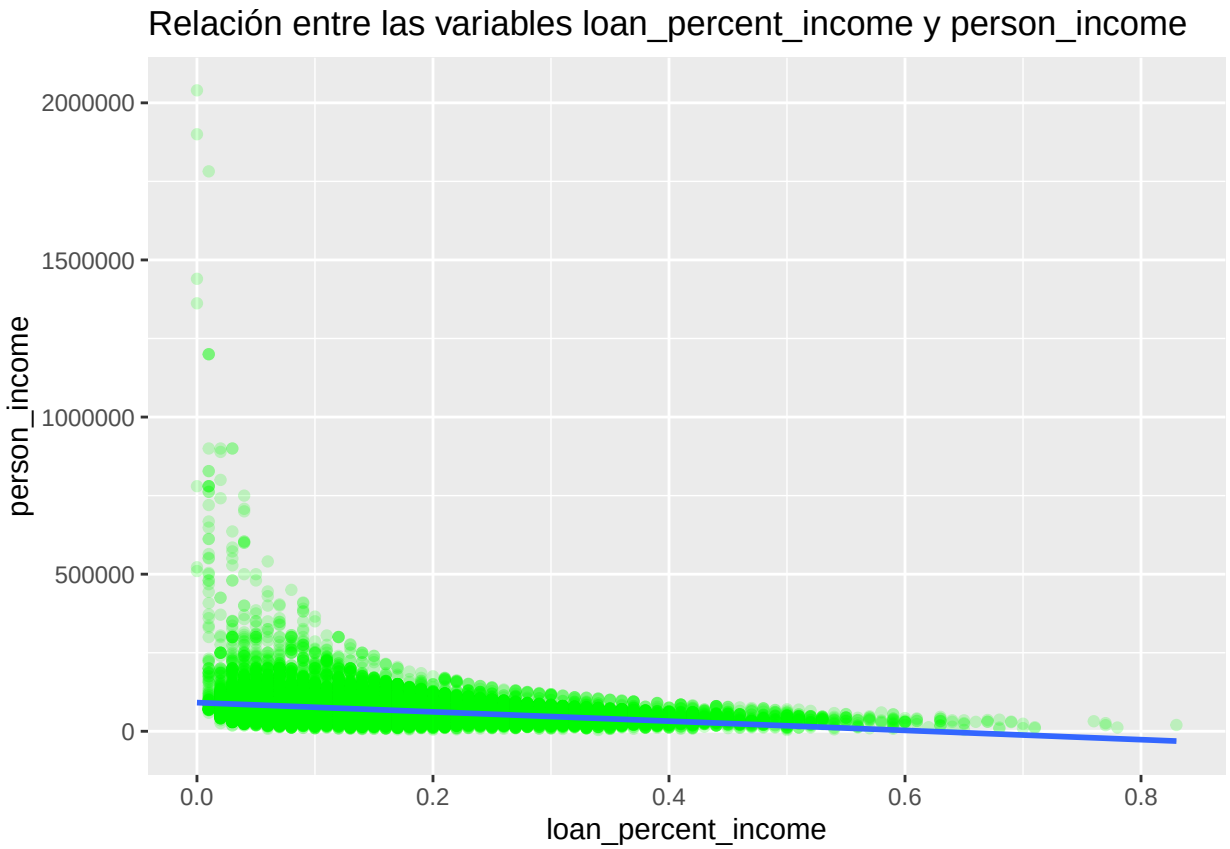


- **Correlación: loan_percent_income y person_income:**

```
cor(x = fcr_train$loan_percent_income, y = fcr_train$person_income)
```

```
## [1] -0.2900526
```

```
fcr_train %>%
    ggplot(aes(loan_percent_income, person_income)) + geom_point(alpha = 0.2,
    colour = "green") + geom_smooth(formula = "y ~ x", method = "lm") +
    labs(title = "Relación entre las variables loan_percent_income y person_income",
        x = "loan_percent_income", y = "person_income")
```



- **Correlación: loan_percent_income y loan_amnt:**

```
cor(x = fcr_train$loan_percent_income, y = fcr_train$loan_amnt)
```

```
## [1] 0.5707228
```

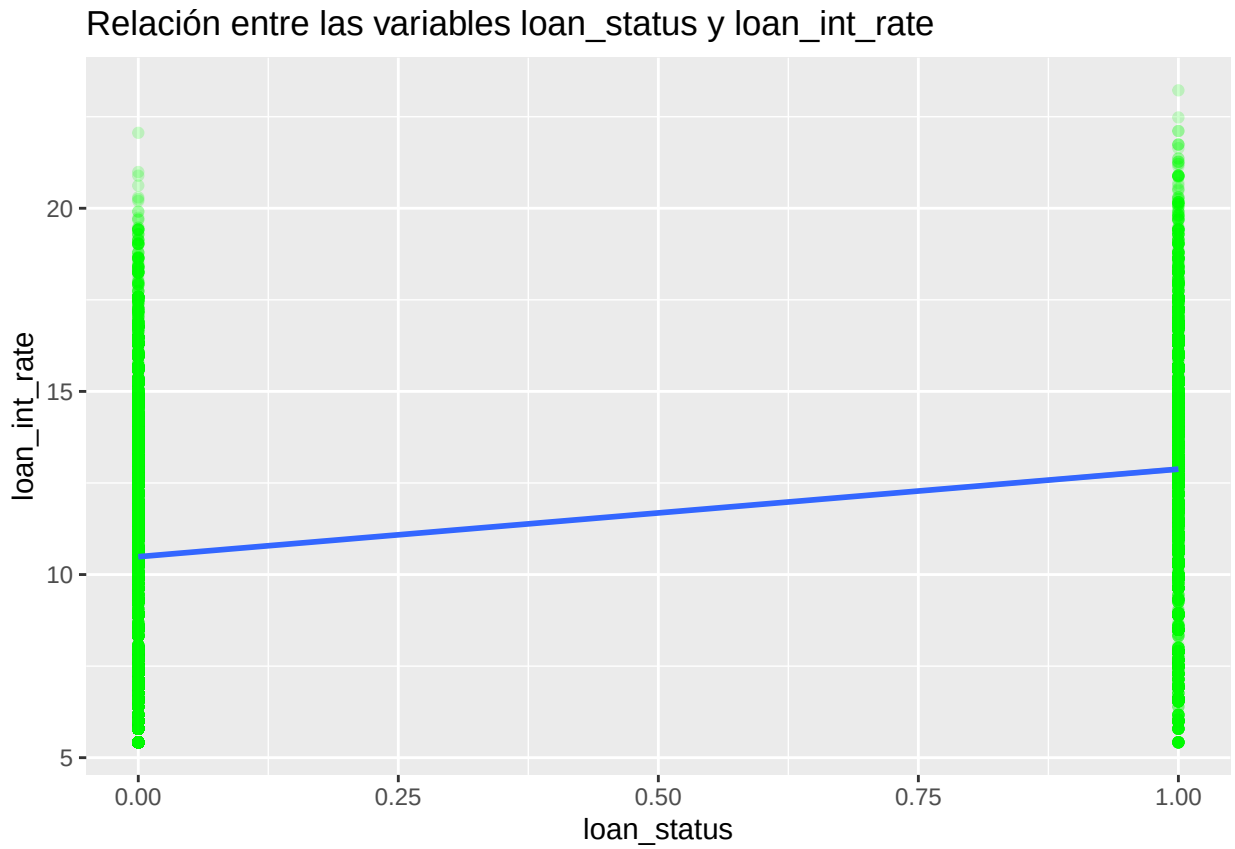
```
fcr_train %>%
    ggplot(aes(loan_percent_income, loan_amnt)) + geom_point(alpha = 0.2,
    colour = "green") + geom_smooth(formula = "y ~ x", method = "lm") +
    labs(title = "Relación entre las variables loan_percent_income y loan_amnt",
        x = "loan_percent_income", y = "loan_amnt")
```

- **Correlación: loan_status y loan_int_rate:**

```
cor(x = fcr_train$loan_status, y = fcr_train$loan_int_rate)
```

```
## [1] 0.3211891
```

```
fcr_train %>%
    ggplot(aes(loan_status, loan_int_rate)) + geom_point(alpha = 0.2,
    colour = "green") + geom_smooth(formula = "y ~ x", method = "lm") +
    labs(title = "Relación entre las variables loan_status y loan_int_rate",
        x = "loan_status", y = "loan_int_rate")
```

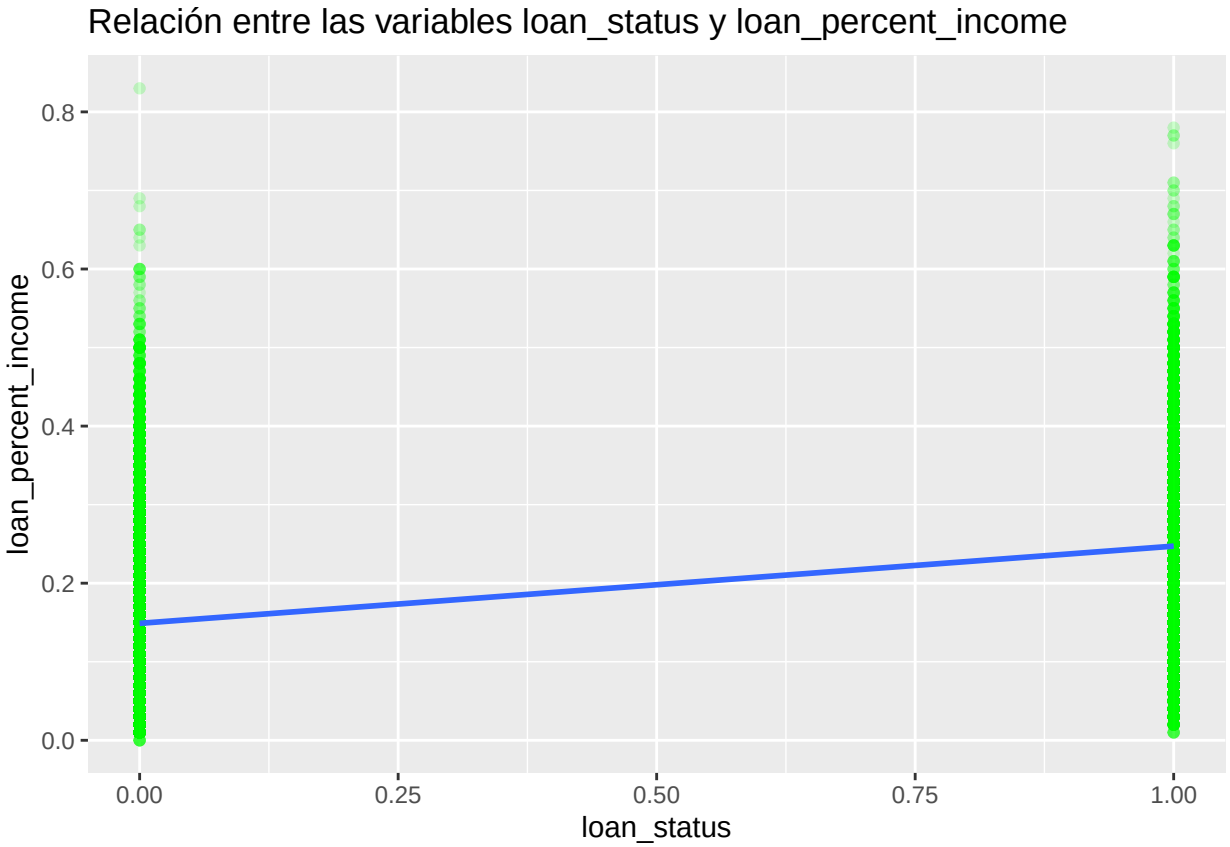


- **Correlación: loan_status y loan_percent_income:**

```
cor(x = fcr_train$loan_status, y = fcr_train$loan_percent_income)
```

```
## [1] 0.3796514
```

```
fcr_train %>%
    ggplot(aes(loan_status, loan_percent_income)) + geom_point(alpha = 0.2,
    colour = "green") + geom_smooth(formula = "y ~ x", method = "lm") +
    labs(title = "Relación entre las variables loan_status y loan_percent_income",
        x = "loan_status", y = "loan_percent_income")
```

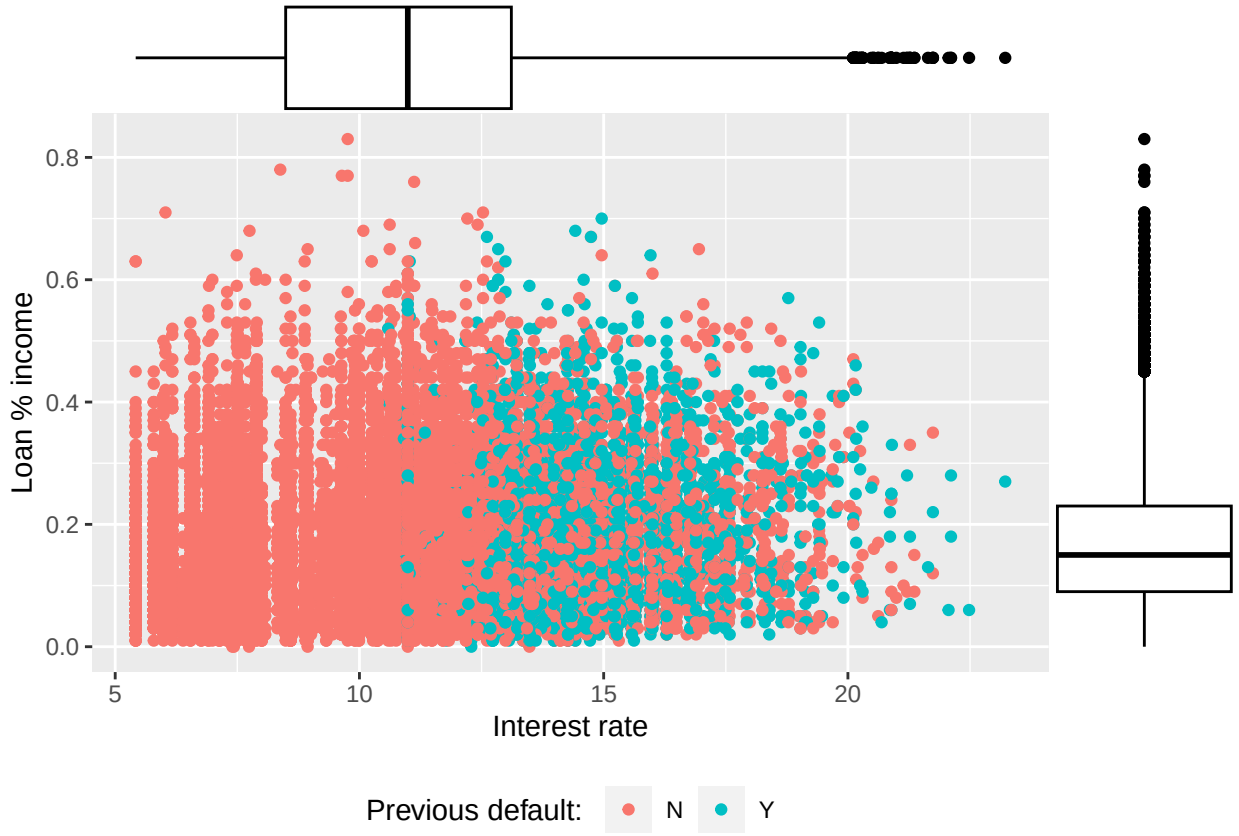


6.4. Análisis de las variables categóricas

Se realiza un análisis de las variables categóricas en relación con el resto de variables numéricas del dataset. Se trata de entender las características personales de los clientes de banca retail y su actividad crediticia para tratar de sacar un cierto análisis preliminar.

- **Variable cb_person_default_on_file en función de las variables loan_int_rate y person_income:**

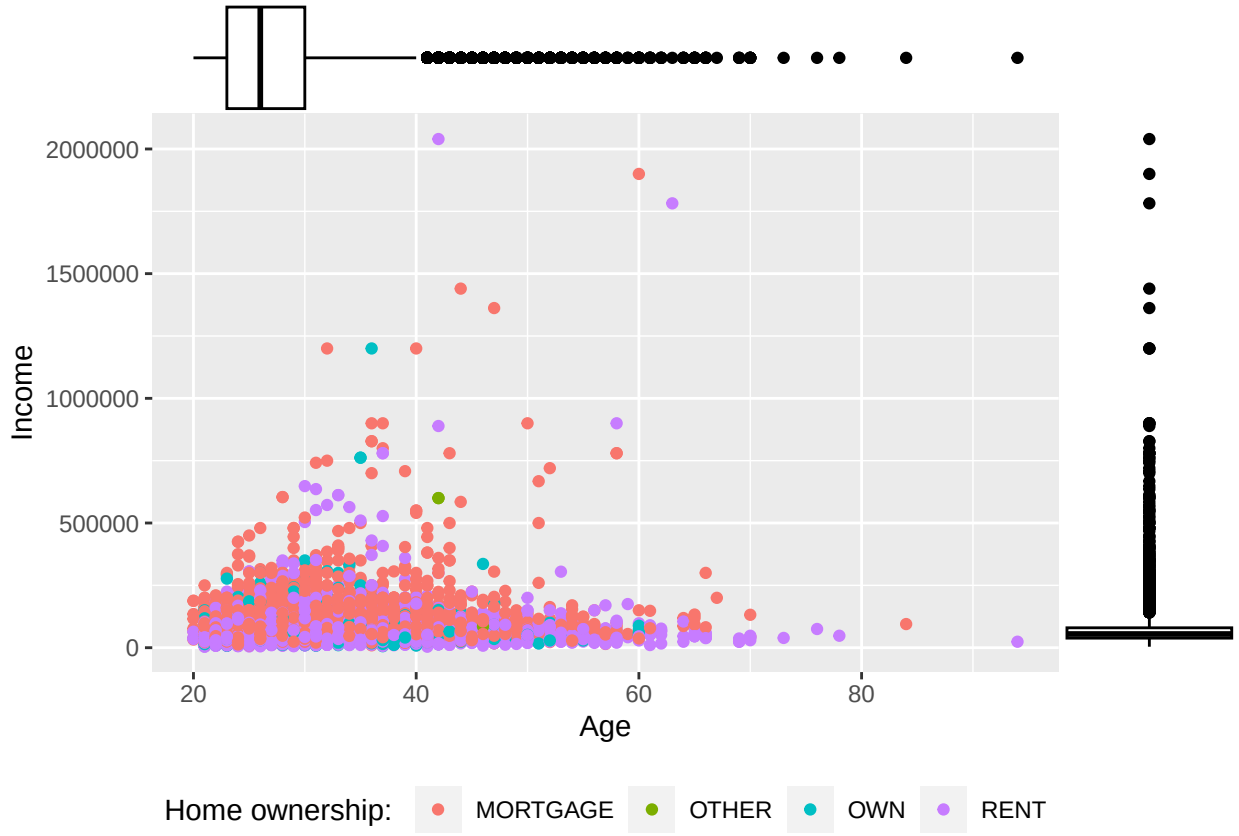
```
plot1 <- ggplot(fcr_train, aes(loan_int_rate, loan_percent_income,
    col = cb_person_default_on_file)) + geom_point() + ylab("Loan % income") +
    xlab("Interest rate") + labs(color = "Previous default: ") +
    theme(legend.position = "bottom")

ggExtra::ggMarginal(plot1, type = "boxplot")
```



Con este gráfico se puede entender que los tipos de interés aplicados por las entidades bancarias en los préstamos que conceden, vienen influidos por si el cliente ha registrado algún impago anterior en su historial crediticio. Además, se ve como el porcentaje de lo que supone el préstamo sobre los ingresos anuales en dólares de la persona que toma el crédito, no es algo definitivo para establecer el tipo de interés que se aplica en la operación.

- **Variable person_home_ownership en función de las variables person_age y person_income:**

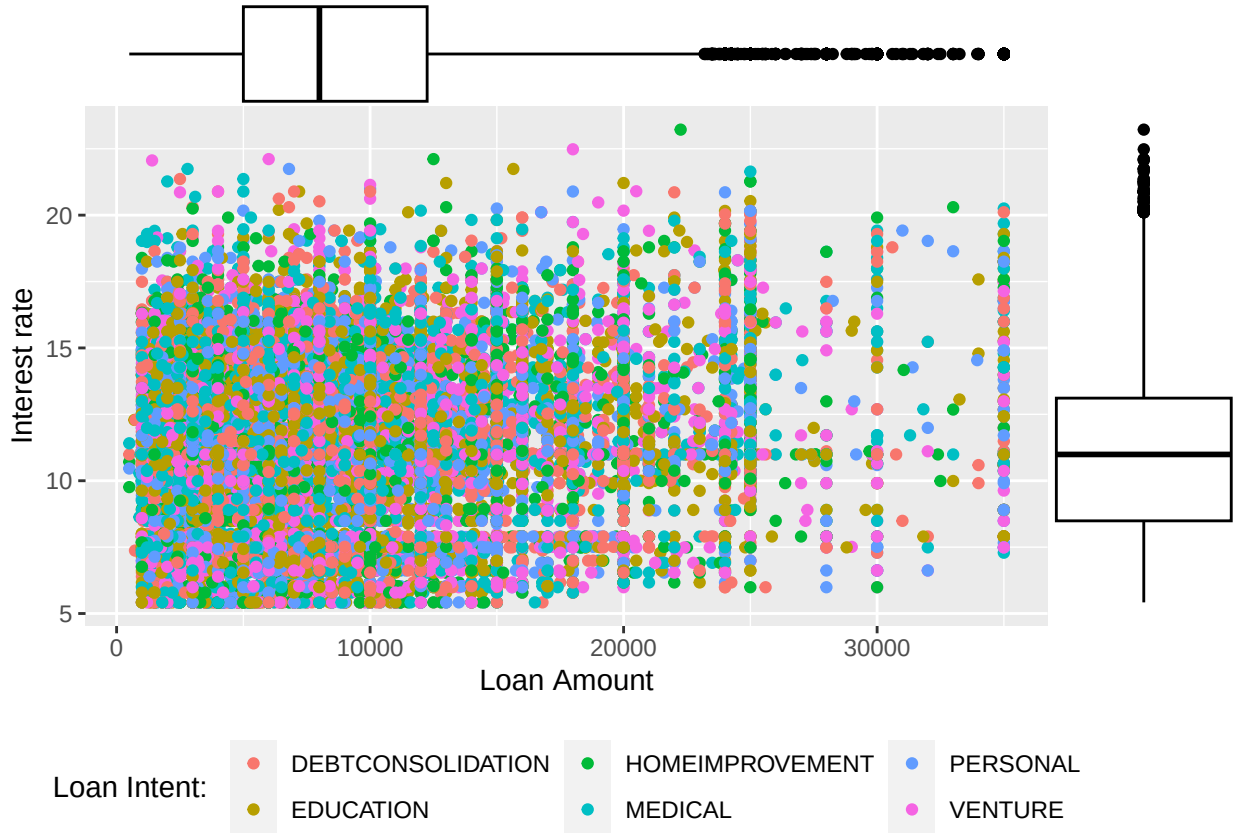
```
plot1 <- ggplot(fcr_train, aes(person_age, person_income, col = person_home_ownership)) +
    geom_point() + ylab("Income") + xlab("Age") + labs(color = "Home ownership: ") +
    theme(legend.position = "bottom")

ggExtra::ggMarginal(plot1, type = "boxplot")
```



En base a este gráfico se puede sacar como conclusión que: existe un sesgo en la edad (el dataset tiene más registros de gente joven), existe un sesgo en los ingresos (el dataset no tiene bien balanceada la variable "person_income", estando está desequilibrada y siendo más frecuente los ingresos bajos) y parece que la gente con hipoteca tiene por lo general más ingresos que la gente que vive de alquiler.

- **Variable loan_intent en función de las variables loan_amnt y loan_int_rate:**

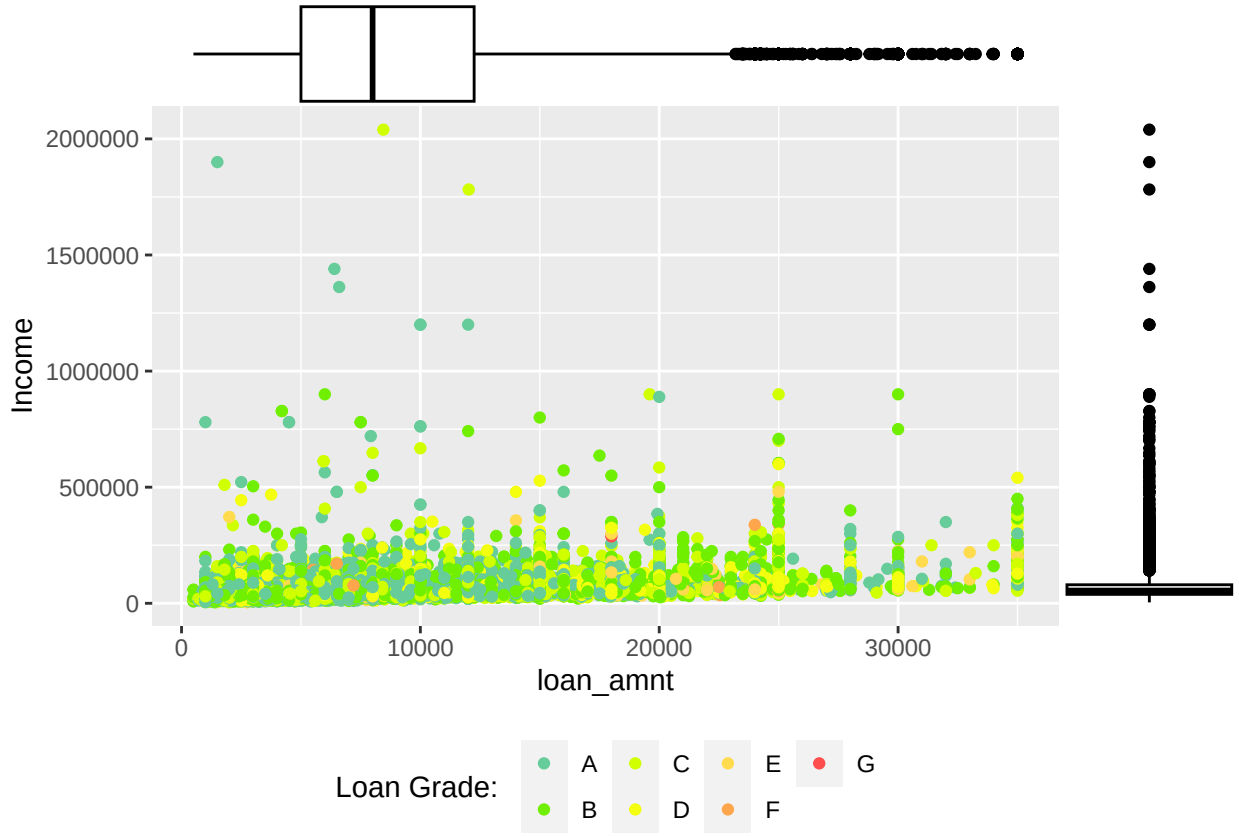
```
plot2 <- ggplot(fcr_train, aes(loan_amnt, loan_int_rate, col = loan_intent)) +
    geom_point() + ylab("Interest rate") + xlab("Loan Amount") +
    labs(color = "Loan Intent: ") + theme(legend.position = "bottom")

ggExtra::ggMarginal(plot2, type = "boxplot")
```



Este tercer gráfico obtenido, representa la relación entre el importe del préstamo solicitado y el tipo de interés aplicado. Aquí se puede ver que no hay una correlación clara entre las dos variables, por lo que se puede decir que no dependen la una de la otra. Además, parece que la intención del préstamo se distribuye por igual en la muestra, lo que significa que la intención no influye en absoluto en los tipos de interés ni en el importe del préstamo. En este caso, las distribuciones son gaussianas normales con un ligero sesgo.

- **Variable loan_grade en función de las variables loan_amnt y person_income:**

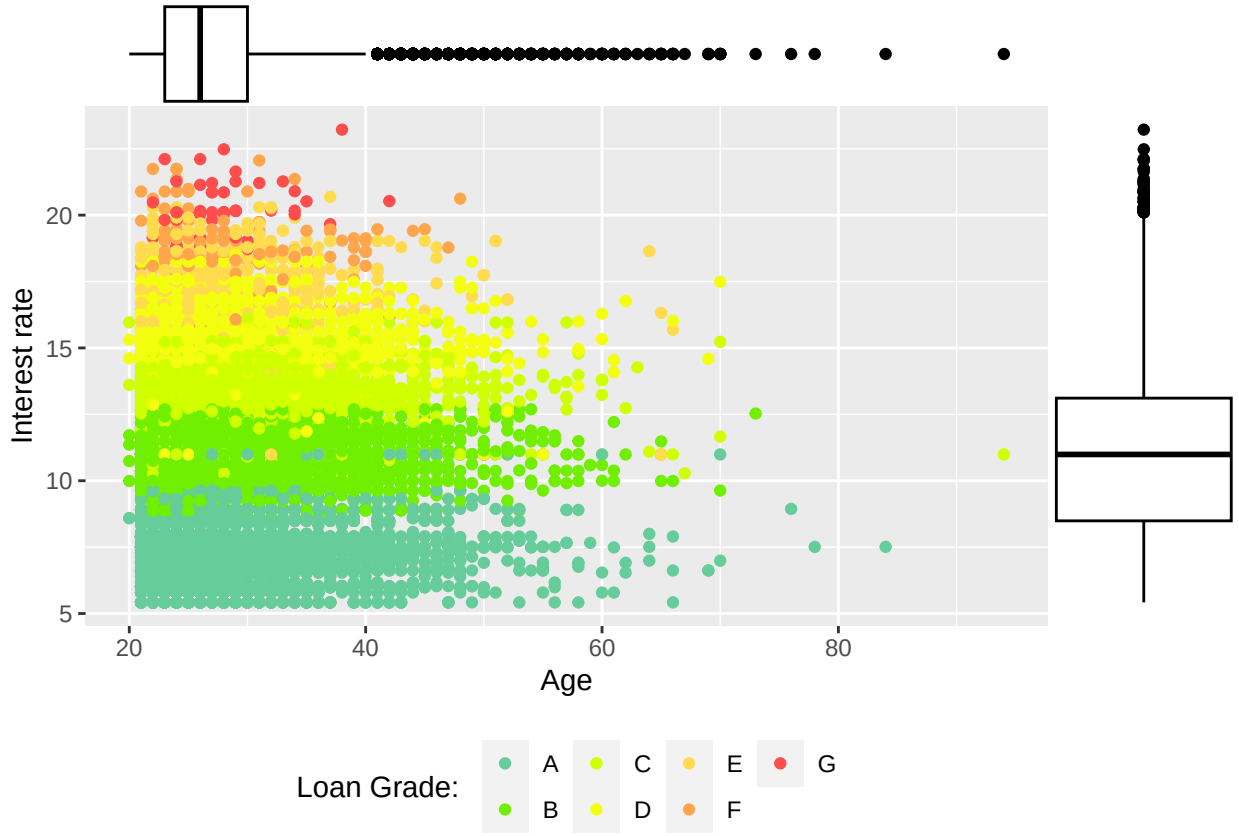
```
plot3 <- ggplot(fcr_train, aes(loan_amnt, person_income, col = loan_grade)) +
    geom_point() + ylab("Income") + labs(color = "Loan Grade: ") +
    theme(legend.position = "bottom") + scale_color_manual(values = c("#66cc99",
    "#70F000", "#D0FF00", "#F3FF0F", "#FFDB4D", "#FFA64D", "#FF4D4D"))

ggExtra::ggMarginal(plot3, type = "boxplot")
```



En este cuarto gráfico obtenido, tampoco existe una relación clara entre los ingresos y el importe del préstamo. Al añadir diferentes colores para las calificaciones de los préstamos, se observa una pequeña tendencia a que las personas con ingresos bajos y con un préstamo elevado tengan una calificación crediticia más baja. Pero no se aprecia una relación clara.

- **Variable loan_grade en función de las variables person_age y loan_int_rate:**

```
plot4 <- ggplot(fcr_train, aes(person_age, loan_int_rate, col = loan_grade)) +
    geom_point() + ylab("Interest rate") + xlab("Age") + labs(color = "Loan Grade: ") +
    theme(legend.position = "bottom") + scale_color_manual(values = c("#66cc99",
    "#70F000", "#D0FF00", "#F3FF0F", "#FFDB4D", "#FFA64D", "#FF4D4D"))

ggExtra::ggMarginal(plot4, type = "boxplot")
```



En este quinto gráfico se muestra la relación entre la edad, el tipo de interés y el grado del préstamo. Aquí se puede ver claramente que la calificación del préstamo depende del tipo de interés, que divide el gráfico en varias secciones. Básicamente se puede apreciar que cuanto más alto es el tipo de interés, mayor es el riesgo percibido. Esto podría deberse al método de evaluación del grado de riesgo de las personas. Además, se observa que las personas mayores tienden a tener tipos de interés más bajos por término medio que los jóvenes.

6.5. Análisis de la variable de interés

Ahora se pasa a analizar la variable de interes "loan_status", para ver la probabilidad de impago de un crédito o préstamo dadas ciertas características en el cliente o en el propio préstamos.

- **Variable loan_status en función de las variables loan_percent_income y loan_int_rate:**

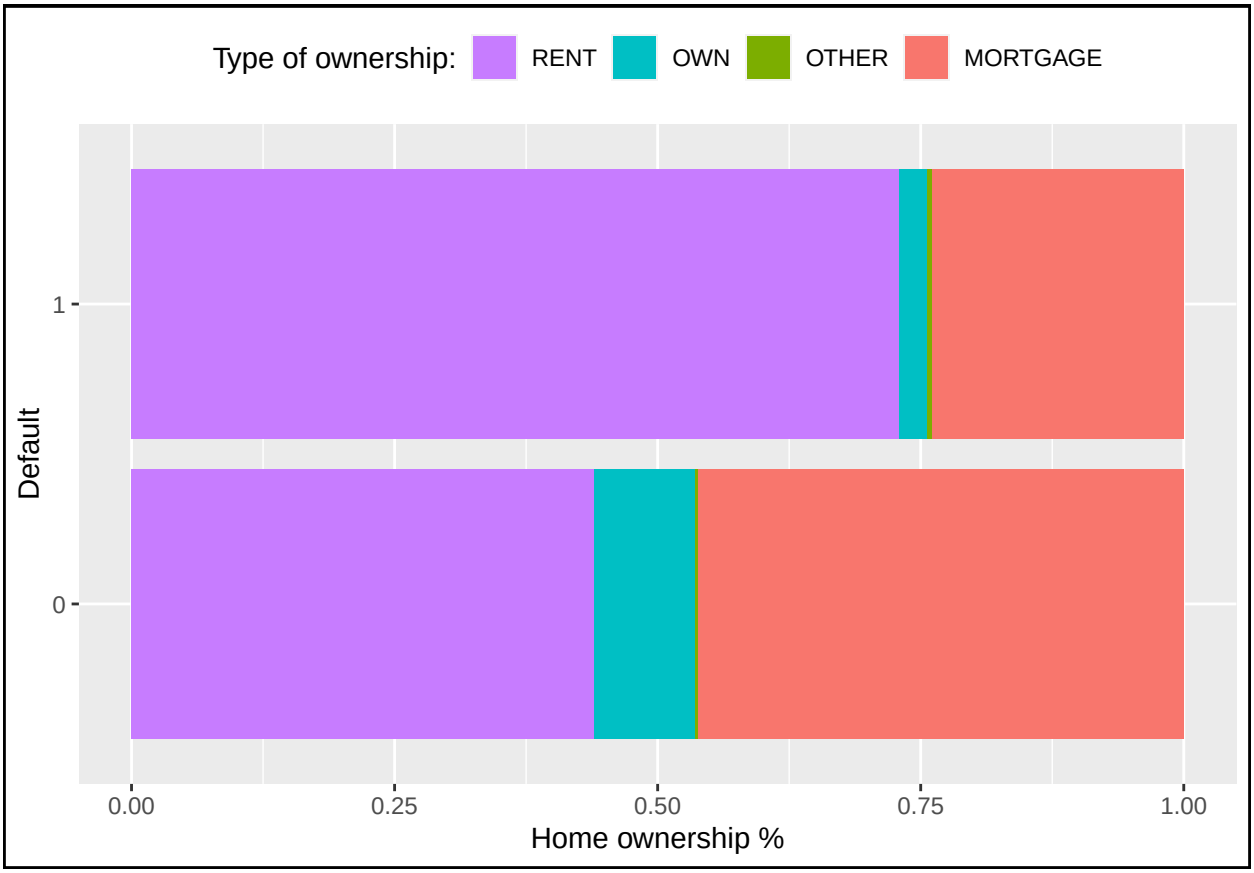
```
plot5 <- ggplot(fcr_train, aes(loan_percent_income, loan_int_rate,
    col = factor(loan_status, labels = c("Not default", "Default")))) +
    geom_point() + ylab("Interest rate") + xlab("Loan percent income") +
    labs(color = "Loan Status: ") + theme(legend.position = "bottom") +
    scale_color_manual(values = c("#66cc99", "#ff3366"))

ggExtra::ggMarginal(plot5, type = "boxplot")
```

En el siguiente gráfico se vuelve a considerar el nivel de los tipos de interés, en relación con la proporción entre el importe del préstamo y los ingresos. La forma de los puntos sugiere que no hay una relación clara entre estas dos variables, lo que significa que los tipos de interés del préstamo no dependen en absoluto del Porcentaje de lo que supone el préstamo sobre los ingresos del cliente. Sin embargo, cuanto más nos acercamos a un ratio más alto, más bajos son los tipos de interés.

Si se analiza a las personas que han impagado (en rojo) y a las personas que no han impagado (en verde), podemos ver claramente niveles de umbral hacia el 12% para los tipos de interés, y 0,3 para la variable de porcentaje de lo que supone el préstamo sobre los ingresos del cliente. Esta zona podría considerarse menos arriesgada tanto para los prestamistas como para los prestatarios. Este patrón parece razonable, ya que unos tipos de interés más altos y una mayor relación ingresos/préstamo significa que es menos probable que los clientes reembolsen su deuda. Es decir, a mayor nivel de tipo de interés del préstamo y un mayor nivel de endeudamiento en relación con los ingresos, es más probable que un cliente de una entidad bancaria termine por impagar su deuda y entrar en default.

- **Variable loan_status en función de la variable person_home_ownership:**

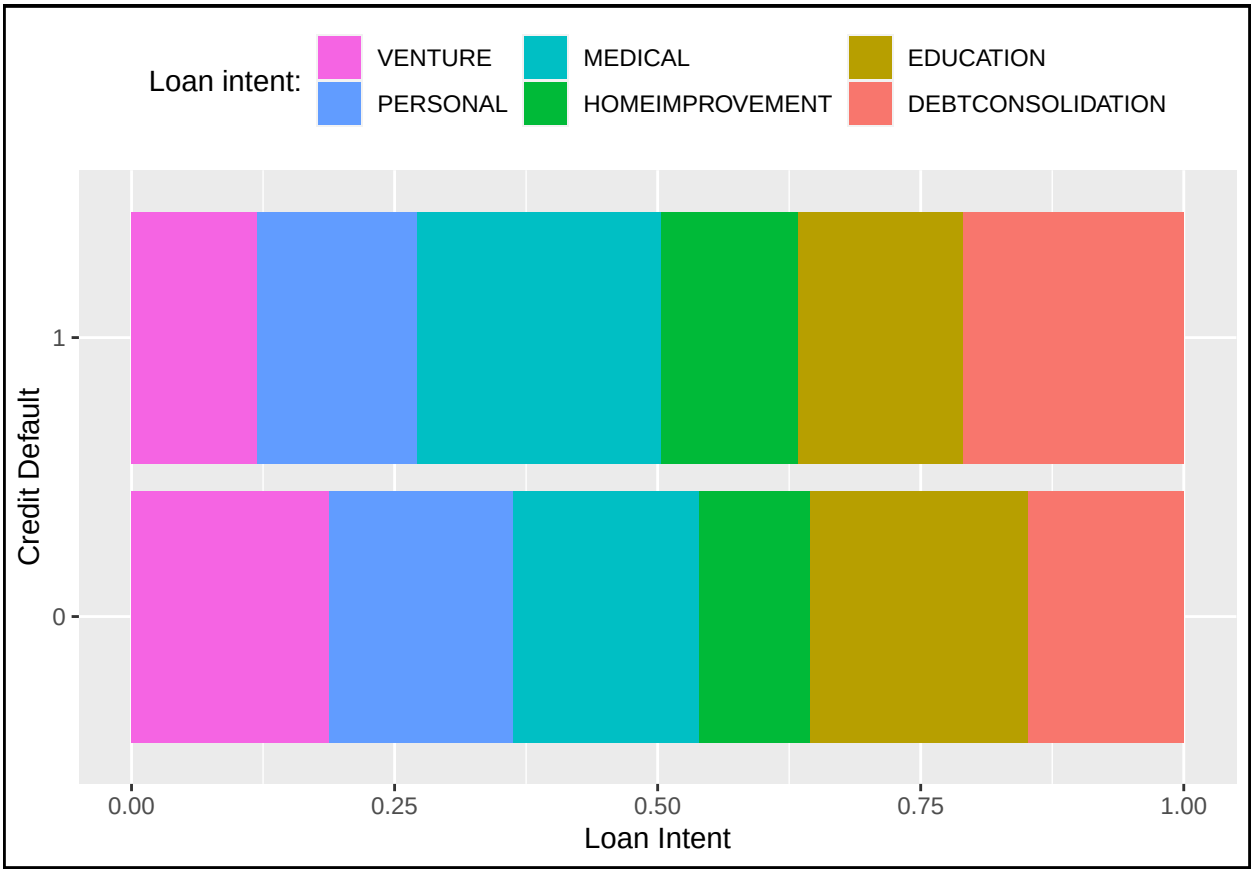
```
ggplot(fcr_train, aes(x = factor(loan_status), fill = factor(person_home_ownership))) +
    geom_bar(position = "fill") + ylab("Home ownership %") +
    xlab("Default") + labs(fill = "Type of ownership:") + theme(legend.position = "top",
    plot.background = element_rect(colour = "black", size = 1)) +
    guides(fill = guide_legend(reverse = TRUE)) + coord_flip()
```



Con este gráfico se muestran las frecuencias absolutas del tipo de propiedad de la vivienda con respecto a las personas que han incurrido en impago del crédito. Se puede observar que típicamente quien paga un alquiler es más propenso al impago considerando todas las demás clases.

- **Variable loan_status en función de la variable loan_intent:**

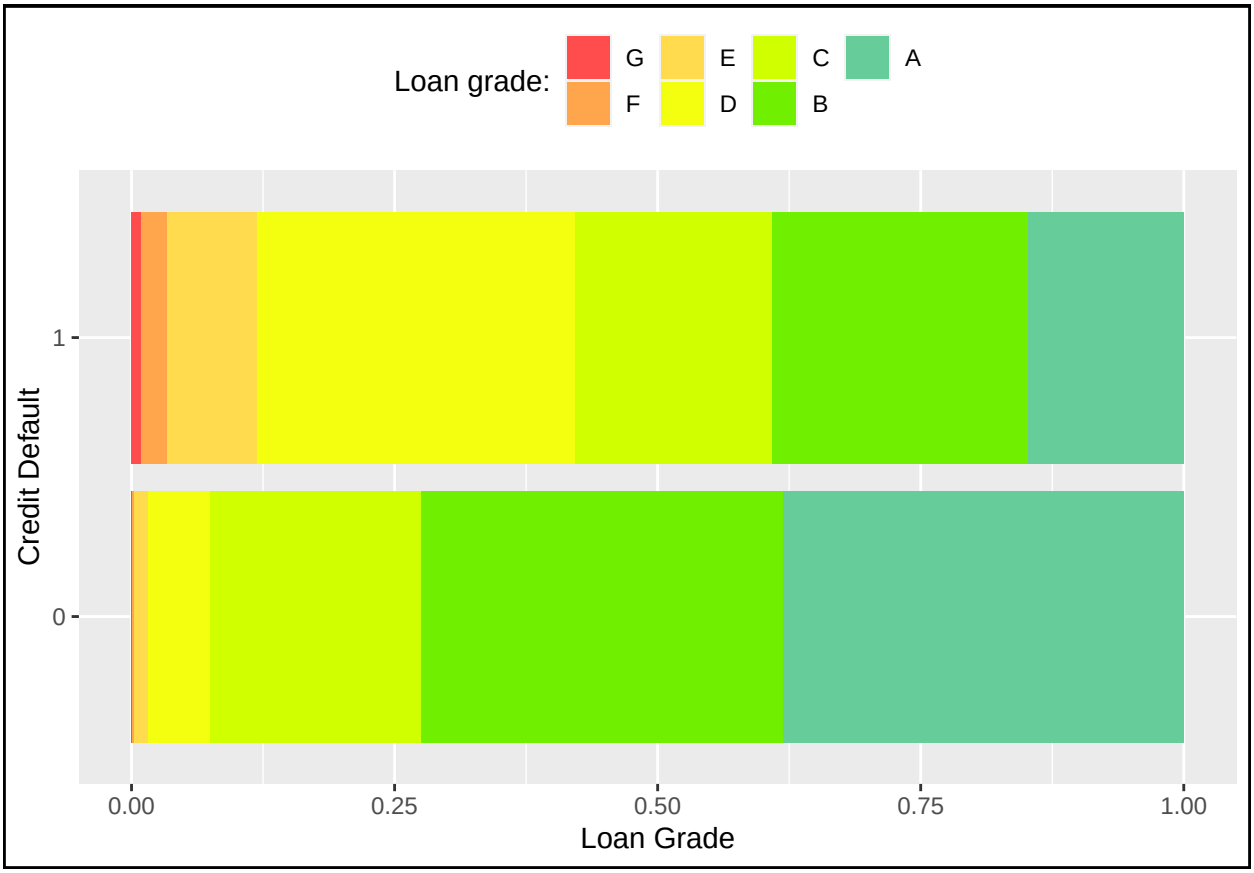
```
ggplot(fcr_train, aes(x = factor(loan_status), fill = factor(loan_intent))) +
    geom_bar(position = "fill") + ylab("Loan Intent") + xlab("Credit Default") +
    labs(fill = "Loan intent:") + theme(legend.position = "top",
    plot.background = element_rect(colour = "black", size = 1)) +
    guides(fill = guide_legend(reverse = TRUE)) + coord_flip()
```



Con este gráfico se pasa a comparar a las personas que han incumplido con el pago del préstamo y las que no, con la finalidad del préstamo solicitado. Como hemos señalado antes, la variable de la finalidad del préstamo está bien distribuida, sin embargo, cuando el préstamo se dedica a la consolidación de deudas y con fines médicos, parece más probable que se produzca un impago de la deuda.

- **Variable loan_status en función de la variable loan_grade:**

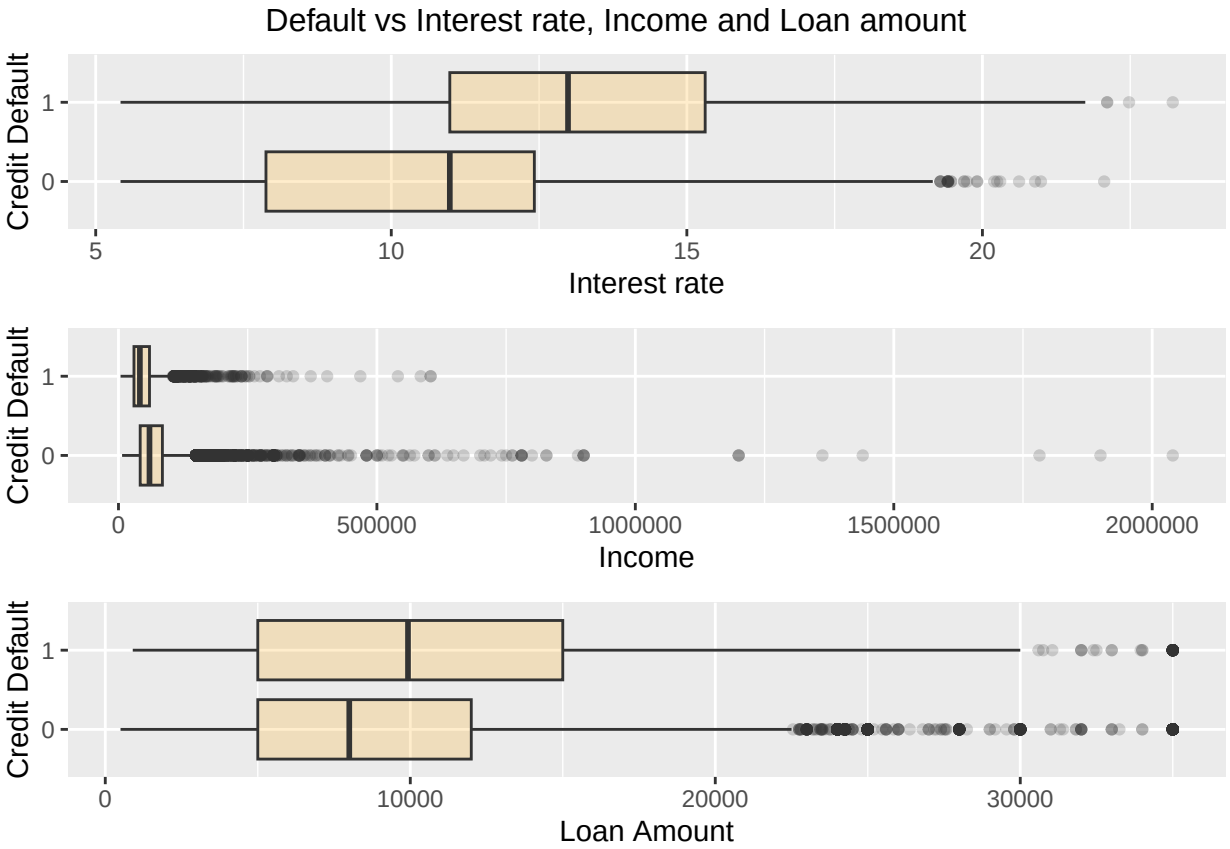
```
ggplot(fcr_train, aes(x = factor(loan_status), fill = factor(loan_grade))) +
    geom_bar(position = "fill") + scale_fill_manual(values = c("#66cc99",
    "#70F000", "#D0FF00", "#F3FF0F", "#FFDB4D", "#FFA64D", "#FF4D4D")) +
    ylab("Loan Grade") + xlab("Credit Default") + labs(fill = "Loan grade:") +
    theme(legend.position = "top", plot.background = element_rect(colour = "black",
        size = 1)) + guides(fill = guide_legend(reverse = TRUE)) +
    coord_flip()
```



Utilizando de nuevo la paleta de colores para el grado de riesgo, es posible observar que esta variable tiene cierto poder predictivo de los impagos de los tomadores de crédito. Según los datos, cuando el grado de riesgo (scoring asignado a cada cliente en función del riesgo) es más bajo (calificación crediticia "A"), es menos probable que se produzca un impago. Lo contrario ocurre con los grados inferiores ("D", "E", "F" y "G").

- **Variable loan_status en función de las variables loan_int_rate, person_income y loan_amnt:**

```
g1 <- ggplot(fcr_train, aes(x = factor(loan_status), y = loan_int_rate)) +
    geom_boxplot(fill = "orange", alpha = 0.2) + ylab("Interest rate") +
    xlab("Credit Default") + coord_flip()

g2 <- ggplot(fcr_train, aes(x = factor(loan_status), y = person_income)) +
    geom_boxplot(fill = "orange", alpha = 0.2) + ylab("Income") +
    xlab("Credit Default") + coord_flip()

g3 <- ggplot(fcr_train, aes(x = factor(loan_status), y = loan_amnt)) +
    geom_boxplot(fill = "orange", alpha = 0.2) + ylab("Loan Amount") +
    xlab("Credit Default") + coord_flip()


grid.arrange(g1, g2, g3, ncol = 1, nrow = 3, top = "Default vs Interest rate, Income and Loan amount")
```



En este último gráfico se compara el riesgo de crédito con tres variables continuas: los tipos de interés, los ingresos anuales y el importe del préstamo. Este gráfico es interesante porque muestra claramente que unos tipos de interés más altos, unos ingresos más bajos y un crédito más elevado aumentan la posibilidad de impago.

APLICACIÓN DE LAS TÉCNICAS Y MODELOS DE MACHINE LEARNING

En esta parte del trabajo, se trata de aplicar diferentes técnicas y algoritmos sobre los datos, de forma que se pueda llegar a la mejor predición y clasificación posible del riesgo de crédito. Para la comparación y selección del mejor algoritmo utilizaremos como métricas de evaluación: "Accuracy", "Precision", "Recall" y "F1 Score", así como la "curva ROC", y trataremos de establecer el mejor punto de corte posible que maximice la misma.

En este apartado se analizan diferentes algoritmos supervisados (aprendizaje supervisado, con datos etiquetados para su predicción o clasificación) sobre nuestro conjunto de datos sobre créditos bancarios.

7. GLM - Generalized Lineal Model

7.1. Variable de interés y análisis de relaciones entre variables

Primero se crean unos datos de train específicos para ser usados en el desarrollo del modelo GLM, y así mantener los originales sin modificar.

```
fcr_train_glm <- fcr_train
fcr_train_glm
```

```
## # A tibble: 26,059 x 12
##    person_age person_i~1 perso~2 perso~3 loan_~4 loan_~5 loan_~6 loan_~7 loan_~8
##         <dbl>      <dbl> <fct>     <dbl> <fct>   <fct>     <dbl>   <dbl>   <dbl>
##  1         28      44000 RENT          2 MEDICAL C         10000   13.5        1
##  2         21      35000 OWN           5 VENTURE B          8000    9.91       0
##  3         25      96000 MORTGA~       6 HOMEIM~ C         21000   14.6        0
##  4         22      67000 OWN           5 EDUCAT~ D          7500   16.3        0
##  5         24      52800 RENT          8 PERSON~ A          9000    7.49       0
##  6         27      50004 RENT         12 DEBTCO~ B          3200   11.5        0
##  7         23      55488 RENT          4 MEDICAL D          5000   15.2        1
##  8         28      70000 RENT          2 DEBTCO~ B          6000   10.4        0
##  9         22      55000 MORTGA~       6 PERSON~ C         13000   13.8        0
## 10         26      43200 RENT          5 EDUCAT~ C          3200   14.4        0
## # ... with 26,049 more rows, 3 more variables: loan_percent_income <dbl>,
## #   cb_person_default_on_file <fct>, cb_person_cred_hist_length <dbl>, and
## #   abbreviated variable names 1: person_income, 2: person_home_ownership,
## #   3: person_emp_length, 4: loan_intent, 5: loan_grade, 6: loan_amnt,
## #   7: loan_int_rate, 8: loan_status
```

```
table(fcr_train_glm$loan_status)
```

```
##
##     0     1
## 20365  5694
```

```
prop.table(table(fcr_train_glm$loan_status))
```

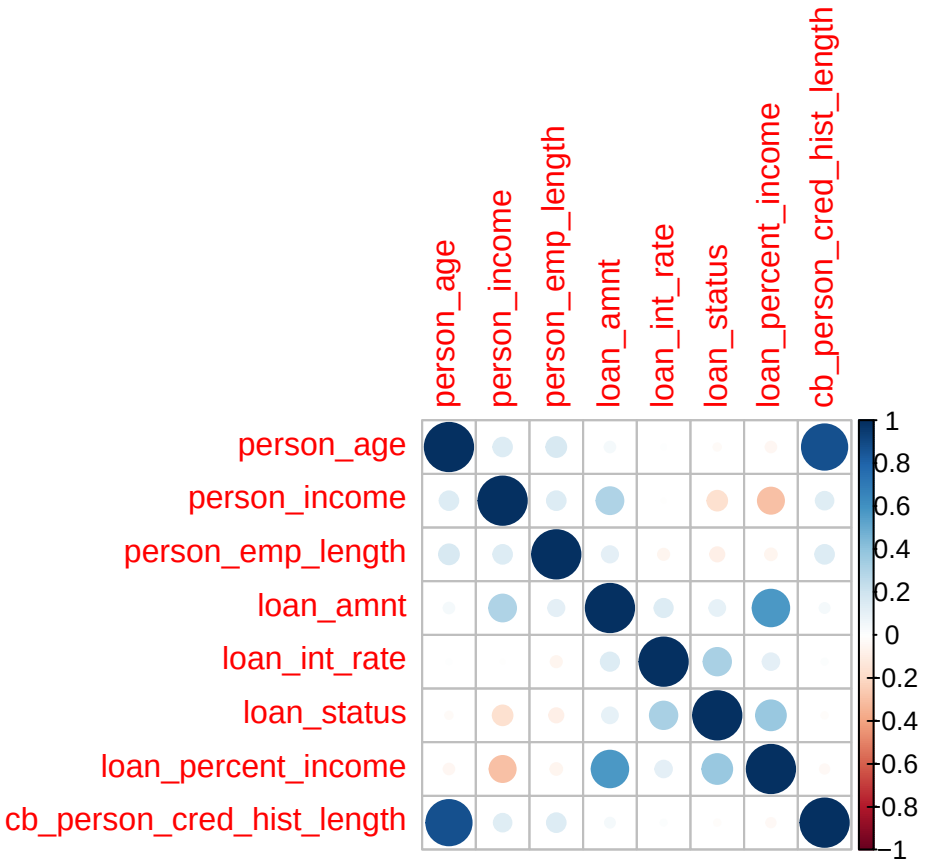
```
##
##         0         1
## 0.7814958 0.2185042
```

```
str(fcr_train_glm)
```

```
## tibble [26,059 x 12] (S3: tbl_df/tbl/data.frame)
##  $ person_age                : num [1:26059] 28 21 25 22 24 27 23 28 22 26 ...
##  $ person_income             : num [1:26059] 44000 35000 96000 67000 52800 ...
##  $ person_home_ownership     : Factor w/ 4 levels "MORTGAGE","OTHER",..: 4 3 1 3 4 4 4 4 1 4 ...
##  $ person_emp_length         : num [1:26059] 2 5 6 5 8 12 4 2 6 5 ...
##  $ loan_intent               : Factor w/ 6 levels "DEBTCONSOLIDATION",..: 4 6 3 2 5 1 4 1 5 2 ...
##  $ loan_grade                : Factor w/ 7 levels "A","B","C","D",..: 3 2 3 4 1 2 4 2 3 3 ...
##  $ loan_amnt                 : num [1:26059] 10000 8000 21000 7500 9000 3200 5000 6000 13000 3200 ..
##  $ loan_int_rate             : num [1:26059] 13.48 9.91 14.65 16.29 7.49 ...
##  $ loan_status               : num [1:26059] 1 0 0 0 0 0 1 0 0 0 ...
##  $ loan_percent_income       : num [1:26059] 0.23 0.23 0.22 0.11 0.17 0.06 0.08 0.09 0.24 0.07 ...
##  $ cb_person_default_on_file : Factor w/ 2 levels "N","Y": 1 1 1 2 1 1 1 1 1 2 ...
##  $ cb_person_cred_hist_length: num [1:26059] 8 3 3 3 4 7 2 9 3 4 ...
```

Analizando la distinción entre créditos impagados y no impagados, se puede ver que la distibución entre ambos grupos está poco balanceada, con 20.365 no impagados (78.15%) y 5.694 (21.85%) impagados en los datos de train.

```
corrplot(cor(fcr_train_glm %>%
    mutate(loan_status = as.numeric(loan_status)) %>%
    keep(is.numeric)))
```




```
res <- cor(fcr_train_glm %>%
    mutate(loan_status = as.numeric(loan_status)) %>%
    keep(is.numeric))
round(res, 2)
```

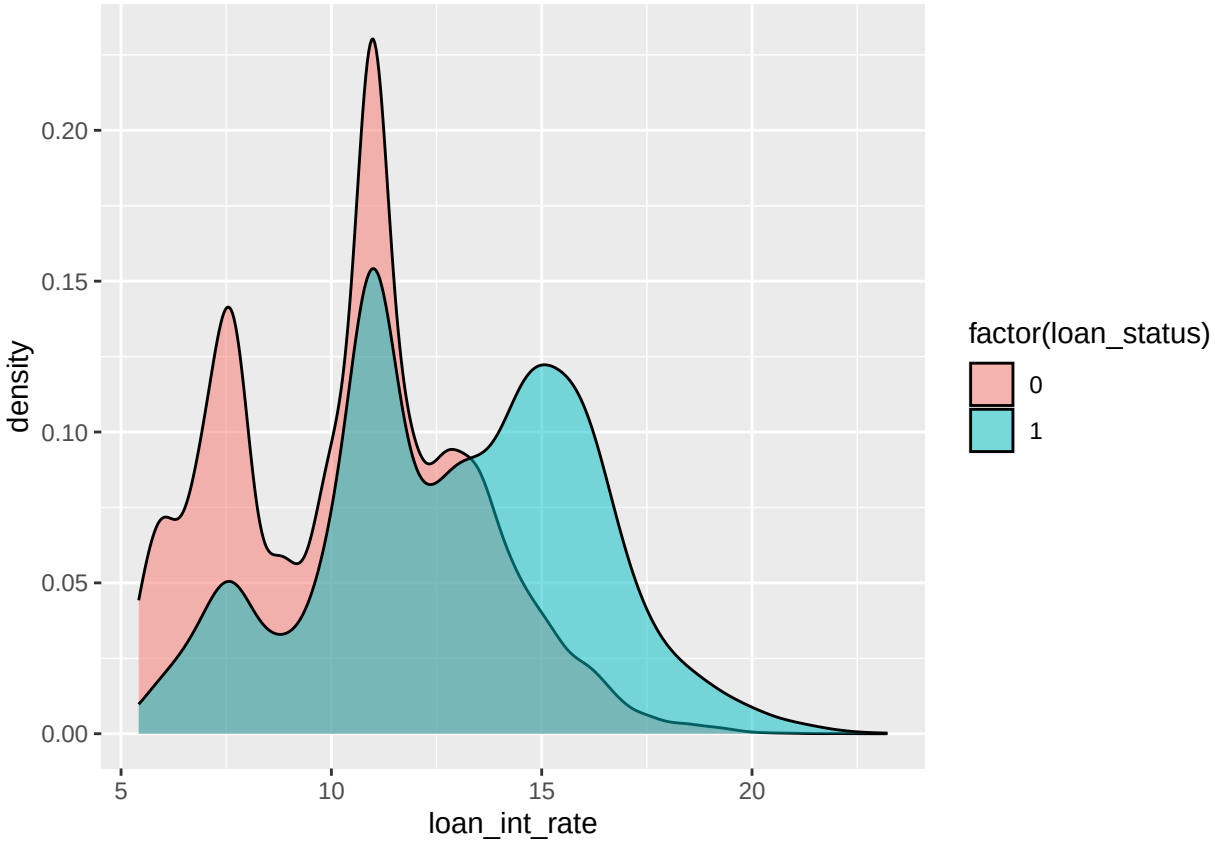
```
##                             person_age person_income person_emp_length loan_amnt
## person_age                        1.00          0.15              0.17      0.05
## person_income                     0.15          1.00              0.15      0.31
## person_emp_length                 0.17          0.15              1.00      0.11
## loan_amnt                         0.05          0.31              0.11      1.00
## loan_int_rate                     0.01         -0.01             -0.05      0.14
## loan_status                      -0.02         -0.17             -0.09      0.11
## loan_percent_income              -0.05         -0.29             -0.06      0.57
## cb_person_cred_hist_length        0.88          0.13              0.15      0.04
##                             loan_int_rate loan_status loan_percent_income
## person_age                           0.01       -0.02                -0.05
## person_income                       -0.01       -0.17                -0.29
```

```
## person_emp_length                   -0.05       -0.09               -0.06
## loan_amnt                            0.14        0.11                0.57
## loan_int_rate                        1.00        0.32                0.12
## loan_status                          0.32        1.00                0.38
## loan_percent_income                  0.12        0.38                1.00
## cb_person_cred_hist_length           0.02       -0.02               -0.04
##                            cb_person_cred_hist_length
## person_age                                       0.88
## person_income                                    0.13
## person_emp_length                                0.15
## loan_amnt                                        0.04
## loan_int_rate                                    0.02
## loan_status                                     -0.02
## loan_percent_income                             -0.04
## cb_person_cred_hist_length                       1.00
```

Se procede a analizar de forma bivariante las variables:

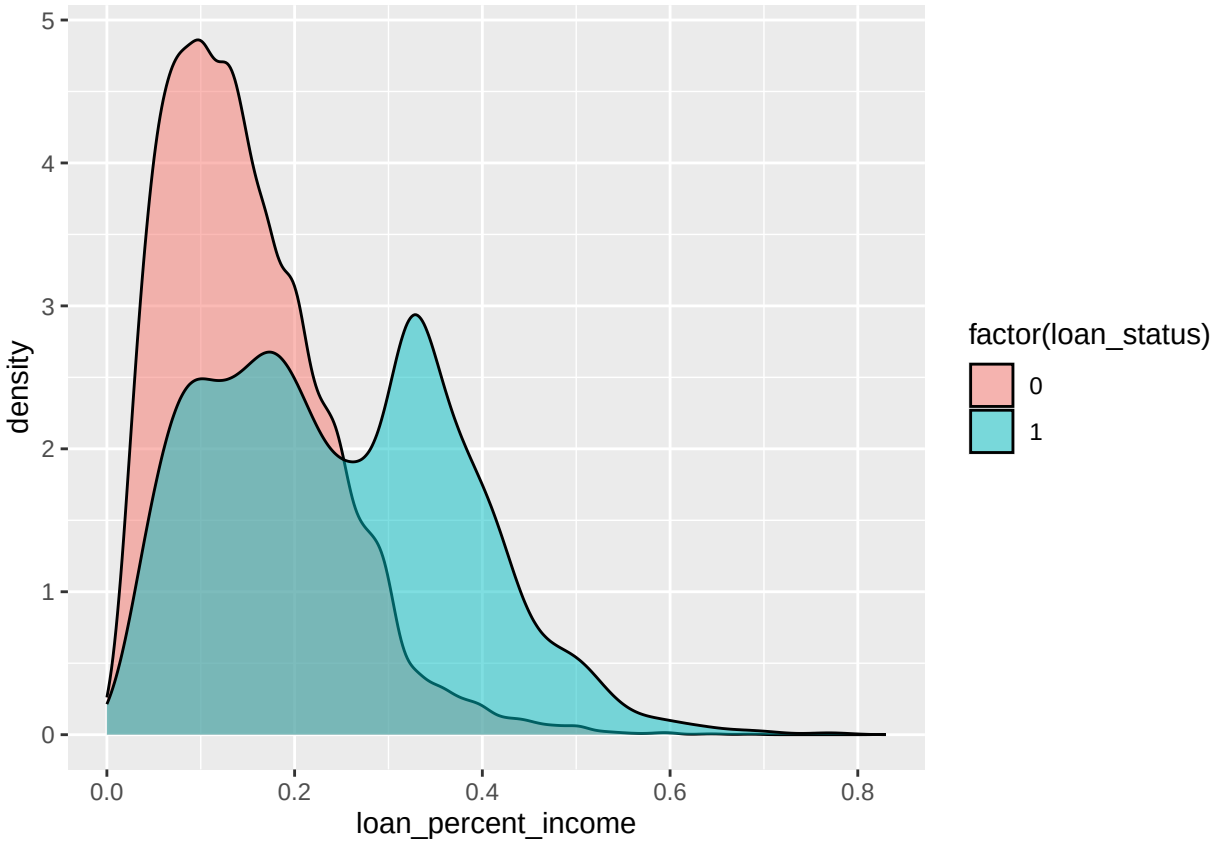
- **Variable loan_int_rate**

```
fcr_train_glm %>%
    ggplot(aes(x = loan_int_rate, fill = factor(loan_status))) +
    geom_density(alpha = 0.5)
```



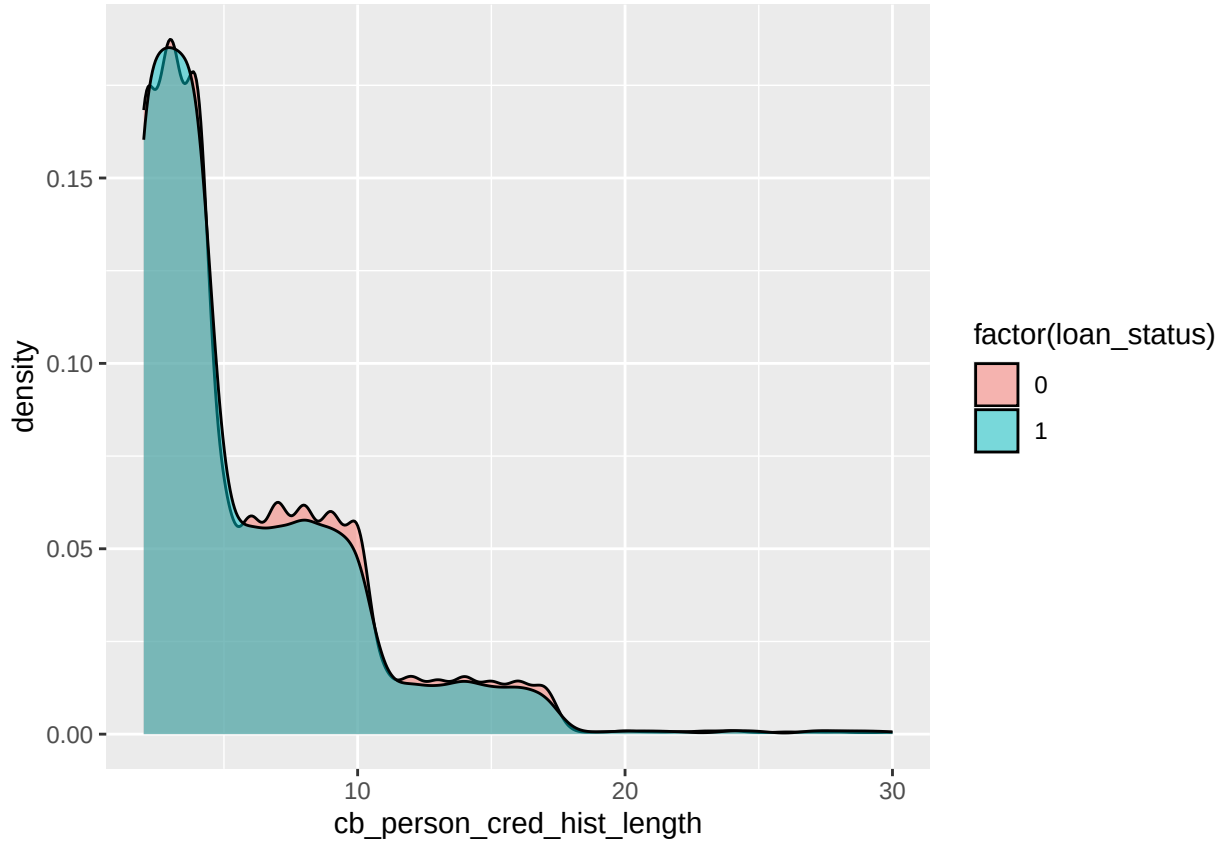
- **Variable loan_percent_income**

```
fcr_train_glm %>%
    ggplot(aes(x = loan_percent_income, fill = factor(loan_status))) +
    geom_density(alpha = 0.5)
```



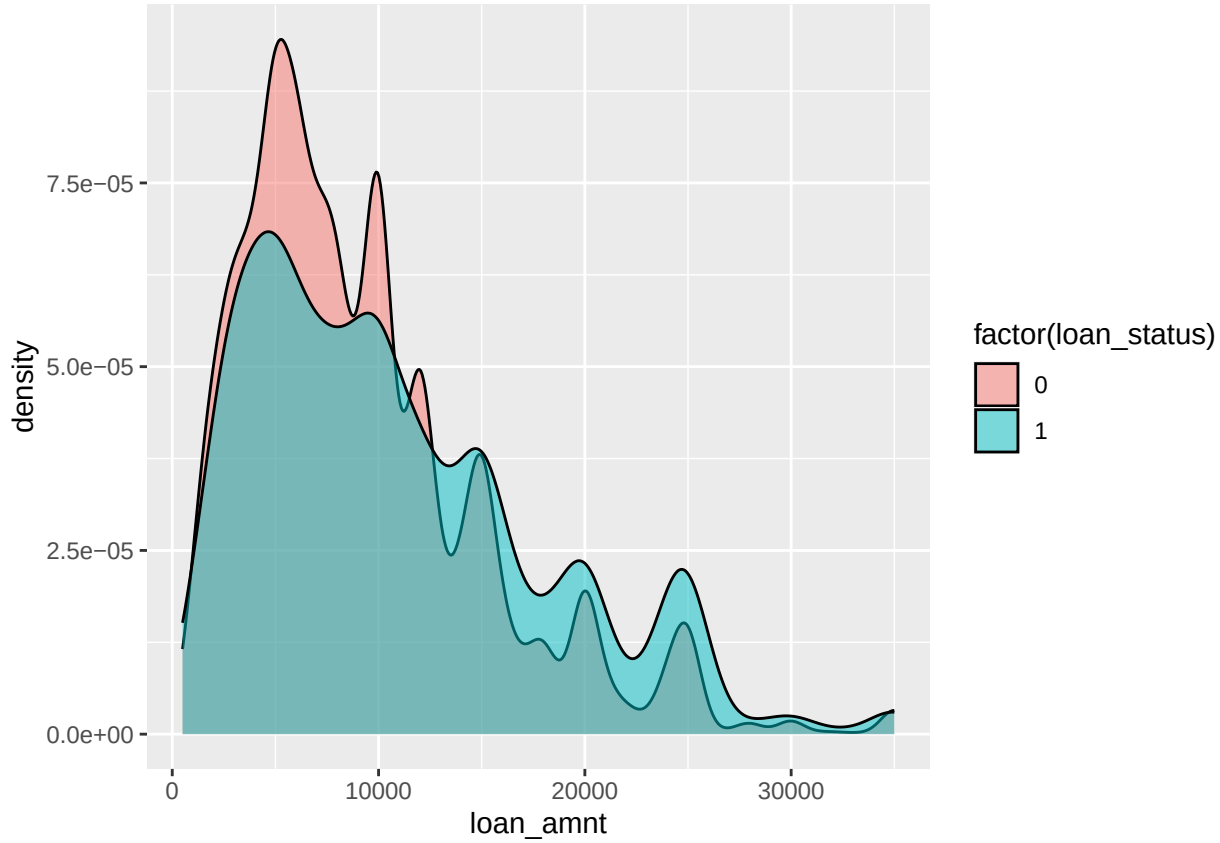
En términos generales vemos como los créditos analizados que están en la categoría de impagados, tienen en general un mayor valor de "loan_int_rate" y de "loan_percent_income".

```
fcr_train_glm %>%
    ggplot(aes(x = cb_person_cred_hist_length, fill = factor(loan_status))) +
    geom_density(alpha = 0.5)
```



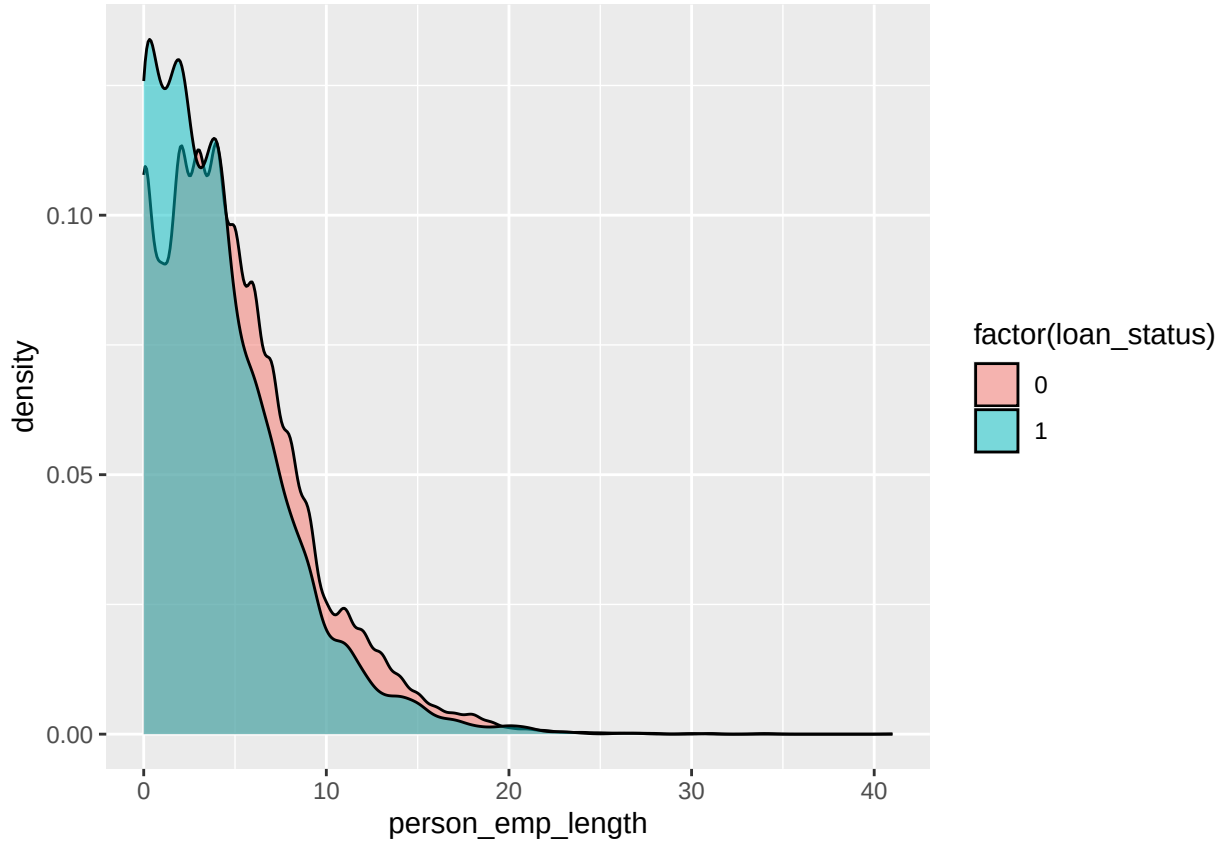
- **Variable loan_amnt**

```
fcr_train_glm %>%
    ggplot(aes(x = loan_amnt, fill = factor(loan_status))) +
    geom_density(alpha = 0.5)
```



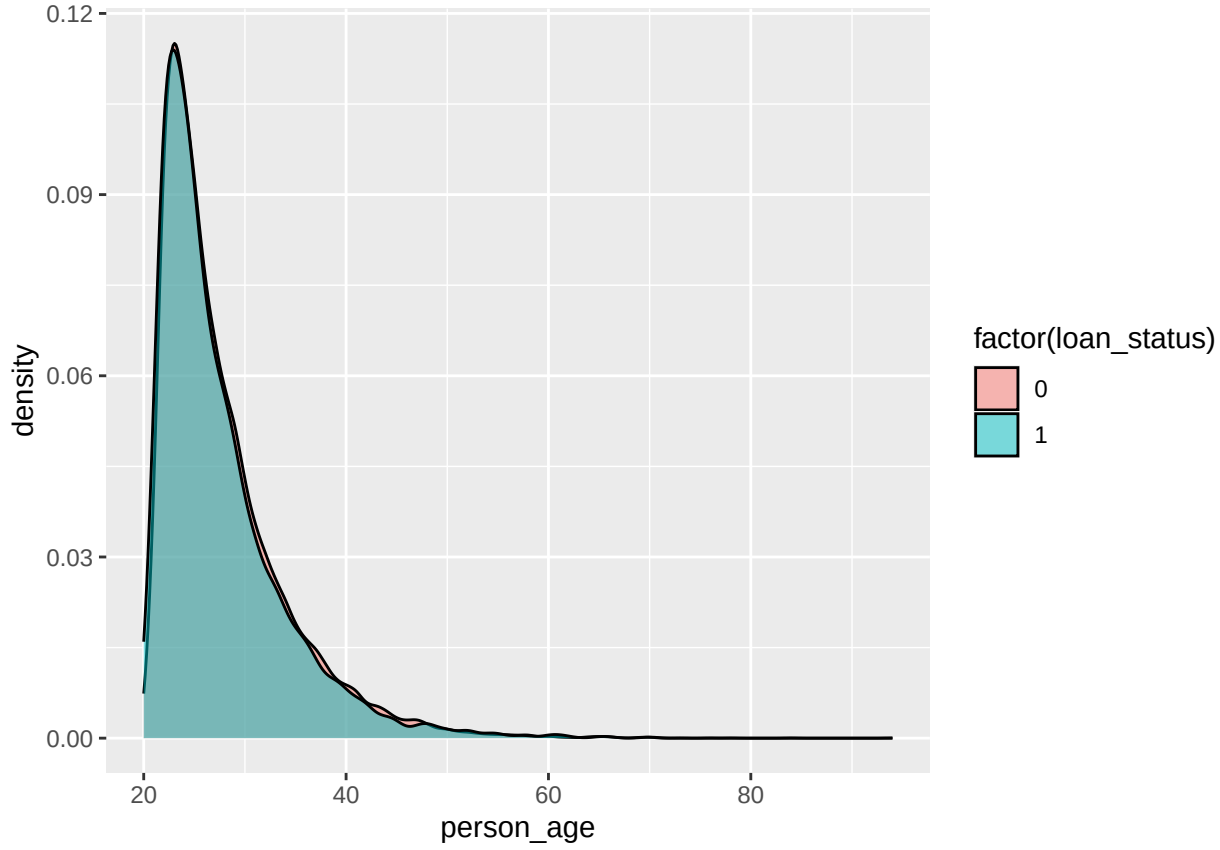
- **Variable person_emp_length**

```
fcr_train_glm %>%
    ggplot(aes(x = person_emp_length, fill = factor(loan_status))) +
    geom_density(alpha = 0.5)
```



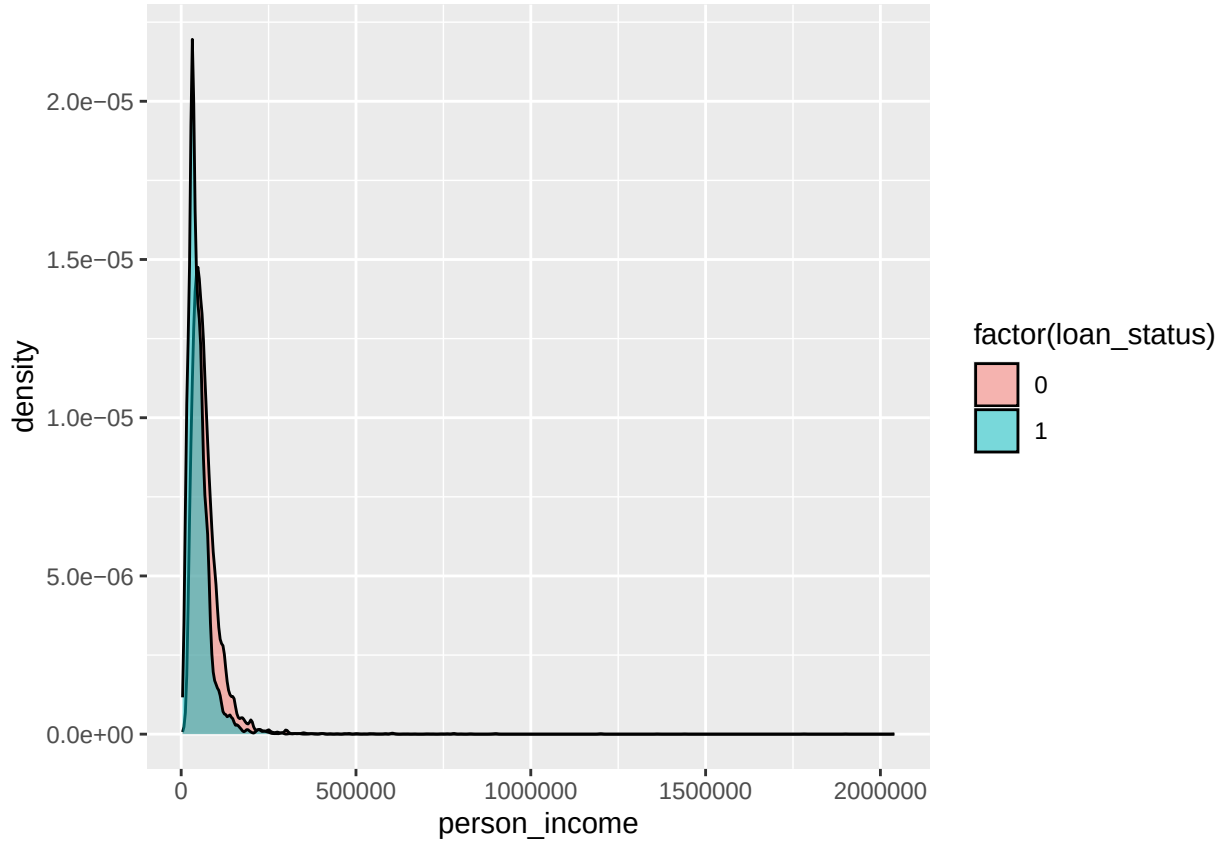
- **Variable person_age**

```
fcr_train_glm %>%
    ggplot(aes(x = person_age, fill = factor(loan_status))) +
    geom_density(alpha = 0.5)
```



- **Variable person_income**

```
fcr_train_glm %>%
    ggplot(aes(x = person_income, fill = factor(loan_status))) +
    geom_density(alpha = 0.5)
```



En los casos de las variables "cb_person_cred_hist_length", "loan_amnt", "person_emp_length", "person_age" y "person_income", cuesta más distinguir en el gráfico de densidad entre créditos impagados o no impagados, ya que no son características tan definitivas de un grupo u otro.

7.2. Creación del modelo de Regresión Logística

Se genera un modelo de regresión logística en base a las variables de nuestro dataset que sirva como predictor de la variable binaria creada.

```
modelo_glm <- glm(loan_status ~ ., data = fcr_train_glm, family = binomial(link = "logit"))
modelo_glm
```

```
##
## Call:  glm(formula = loan_status ~ ., family = binomial(link = "logit"),
##     data = fcr_train_glm)
##
## Coefficients:
##                (Intercept)                  person_age
##                  -4.097e+00                  -4.897e-03
##              person_income  person_home_ownershipOTHER
##                   1.543e-06                   4.270e-01
##   person_home_ownershipOWN   person_home_ownershipRENT
##                  -1.750e+00                   8.360e-01
##          person_emp_length        loan_intentEDUCATION
##                  -1.351e-02                  -8.218e-01
```

```
## loan_intentHOMEIMPROVEMENT          loan_intentMEDICAL
##                   7.747e-02                  -1.494e-01
##        loan_intentPERSONAL          loan_intentVENTURE
##                  -5.980e-01                  -1.069e+00
##                loan_gradeB                 loan_gradeC
##                  2.463e-01                   4.670e-01
##                loan_gradeD                 loan_gradeE
##                  2.585e+00                   2.844e+00
##                loan_gradeF                 loan_gradeG
##                  3.396e+00                   6.754e+00
##                  loan_amnt               loan_int_rate
##                 -1.084e-04                   5.562e-02
##        loan_percent_income  cb_person_default_on_fileY
##                   1.344e+01                   2.107e-02
## cb_person_cred_hist_length
##                   2.282e-03
##
## Degrees of Freedom: 26058 Total (i.e. Null);  26036 Residual
## Null Deviance:       27360
## Residual Deviance: 17660     AIC: 17700
```

```
summary(modelo_glm)
```

```
##
## Call:
## glm(formula = loan_status ~ ., family = binomial(link = "logit"),
##     data = fcr_train_glm)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2983  -0.5243  -0.3028  -0.1234   3.4420
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -4.097e+00  1.938e-01 -21.143  < 2e-16 ***
## person_age                  -4.897e-03  6.501e-03  -0.753 0.451275
## person_income                1.543e-06  5.560e-07   2.775 0.005516 **
## person_home_ownershipOTHER   4.270e-01  3.168e-01   1.348 0.177707
## person_home_ownershipOWN    -1.750e+00  1.120e-01 -15.620  < 2e-16 ***
## person_home_ownershipRENT    8.360e-01  4.490e-02  18.621  < 2e-16 ***
## person_emp_length           -1.351e-02  5.387e-03  -2.509 0.012121 *
## loan_intentEDUCATION        -8.218e-01  6.364e-02 -12.913  < 2e-16 ***
## loan_intentHOMEIMPROVEMENT   7.747e-02  7.085e-02   1.093 0.274220
## loan_intentMEDICAL          -1.494e-01  5.993e-02  -2.493 0.012651 *
## loan_intentPERSONAL         -5.980e-01  6.508e-02  -9.189  < 2e-16 ***
## loan_intentVENTURE          -1.069e+00  6.926e-02 -15.436  < 2e-16 ***
## loan_gradeB                  2.463e-01  7.122e-02   3.458 0.000544 ***
## loan_gradeC                  4.670e-01  1.016e-01   4.598 4.26e-06 ***
## loan_gradeD                  2.585e+00  1.248e-01  20.715  < 2e-16 ***
## loan_gradeE                  2.844e+00  1.617e-01  17.590  < 2e-16 ***
## loan_gradeF                  3.396e+00  2.459e-01  13.811  < 2e-16 ***
## loan_gradeG                  6.754e+00  1.044e+00   6.470 9.80e-11 ***
## loan_amnt                   -1.084e-04  4.996e-06 -21.701  < 2e-16 ***
## loan_int_rate                5.562e-02  1.446e-02   3.845 0.000120 ***
```

```
## loan_percent_income         1.344e+01  2.895e-01  46.421  < 2e-16 ***
## cb_person_default_on_fileY  2.107e-02  5.581e-02   0.377 0.705835
## cb_person_cred_hist_length  2.282e-03  9.923e-03   0.230 0.818104
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27362  on 26058  degrees of freedom
## Residual deviance: 17659  on 26036  degrees of freedom
## AIC: 17705
##
## Number of Fisher Scoring iterations: 6
```

Como observamos, nos quedamos solo con las variables significativas que relamente afectan a "loan_status", y creamos un nuevo modelo exclusivamente con ellas. De esta forma simplificamos el modelo, nos quedamos con las varibales realmente importantes para el modelo predictor y creamos el mejor modelo de regresión logística posible para nuestro conjunto de datos. En principio las variables "cb_person_cred_hist_length", "cb_person_default_on_file" y "person_age", no parecen ser estadisticamente significativas.

Se podría también pensar en la exclusión del modelo, de la variable "loan_percent_income", ya que de forma lógica y como se oberva en las correlaciones, es una variable que viene calculada, relaciona y explicada con las variables "person_income" y "loan_amnt". Esto nos puede llegar a generar multicolinealidad y crear redundancia en los cálculos que hagamos para el modelo.

Finalmente se establece el siguiente modelo definitivo:

```
modelo_glm2 <- glm(loan_status ~ person_income + person_emp_length +
    loan_amnt + loan_int_rate + person_home_ownership + loan_intent +
    loan_grade + loan_percent_income, data = fcr_train_glm, family = binomial)
modelo_glm2
```

```
##
## Call:  glm(formula = loan_status ~ person_income + person_emp_length +
##     loan_amnt + loan_int_rate + person_home_ownership + loan_intent +
##     loan_grade + loan_percent_income, family = binomial, data = fcr_train_glm)
##
## Coefficients:
##                 (Intercept)               person_income
##                   -4.212e+00                   1.494e-06
##           person_emp_length                  loan_amnt
##                   -1.423e-02                  -1.084e-04
##               loan_int_rate  person_home_ownershipOTHER
##                    5.570e-02                   4.303e-01
##    person_home_ownershipOWN   person_home_ownershipRENT
##                   -1.750e+00                   8.350e-01
##       loan_intentEDUCATION  loan_intentHOMEIMPROVEMENT
##                   -8.187e-01                   7.264e-02
##         loan_intentMEDICAL         loan_intentPERSONAL
##                   -1.515e-01                  -5.987e-01
##         loan_intentVENTURE                 loan_gradeB
##                   -1.068e+00                   2.462e-01
##                 loan_gradeC                 loan_gradeD
##                    4.767e-01                   2.595e+00
```

```
##                loan_gradeE                 loan_gradeF
##                  2.852e+00                   3.403e+00
##                loan_gradeG         loan_percent_income
##                  6.762e+00                   1.344e+01
##
## Degrees of Freedom: 26058 Total (i.e. Null);  26039 Residual
## Null Deviance:       27360
## Residual Deviance: 17660     AIC: 17700
```

```
summary(modelo_glm2)
```

```
##
## Call:
## glm(formula = loan_status ~ person_income + person_emp_length +
##     loan_amnt + loan_int_rate + person_home_ownership + loan_intent +
##     loan_grade + loan_percent_income, family = binomial, data = fcr_train_glm)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2910  -0.5238  -0.3031  -0.1232   3.4357
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -4.212e+00  1.426e-01 -29.527  < 2e-16 ***
## person_income               1.494e-06  5.540e-07   2.697 0.006994 **
## person_emp_length          -1.423e-02  5.343e-03  -2.663 0.007736 **
## loan_amnt                  -1.084e-04  4.996e-06 -21.708  < 2e-16 ***
## loan_int_rate               5.570e-02  1.446e-02   3.852 0.000117 ***
## person_home_ownershipOTHER  4.303e-01  3.162e-01   1.361 0.173558
## person_home_ownershipOWN   -1.750e+00  1.120e-01 -15.620  < 2e-16 ***
## person_home_ownershipRENT   8.350e-01  4.489e-02  18.602  < 2e-16 ***
## loan_intentEDUCATION       -8.187e-01  6.358e-02 -12.877  < 2e-16 ***
## loan_intentHOMEIMPROVEMENT  7.264e-02  7.071e-02   1.027 0.304260
## loan_intentMEDICAL         -1.515e-01  5.991e-02  -2.529 0.011446 *
## loan_intentPERSONAL        -5.987e-01  6.506e-02  -9.203  < 2e-16 ***
## loan_intentVENTURE         -1.068e+00  6.924e-02 -15.426  < 2e-16 ***
## loan_gradeB                 2.462e-01  7.121e-02   3.457 0.000547 ***
## loan_gradeC                 4.767e-01  9.753e-02   4.888 1.02e-06 ***
## loan_gradeD                 2.595e+00  1.213e-01  21.390  < 2e-16 ***
## loan_gradeE                 2.852e+00  1.594e-01  17.891  < 2e-16 ***
## loan_gradeF                 3.403e+00  2.448e-01  13.898  < 2e-16 ***
## loan_gradeG                 6.762e+00  1.044e+00   6.479 9.22e-11 ***
## loan_percent_income         1.344e+01  2.894e-01  46.437  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27362  on 26058  degrees of freedom
## Residual deviance: 17660  on 26039  degrees of freedom
## AIC: 17700
##
## Number of Fisher Scoring iterations: 6
```

Los residuos de desviación tienen buen aspecto, aunque los valores no están completamente centrados en cero y no son simétricos. Las estimaciones son los parámetros de nuestro interés, mientras que la columna Pr(>|z|) muestra los valores p de dos colas que prueban la hipótesis nula de que el coeficiente es igual a cero. En otras palabras, muestra la importancia de los efectos de cada variable independiente. En nuestro modelo parece que la mayoría de las variables son estadísticamente significativas, sin embargo, para confirmar esta hipótesis procedemos a utilizar la prueba anova, observando la tabla de desviación.

```
anova(modelo_glm2, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: loan_status
##
## Terms added sequentially (first to last)
##
##
##                       Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                  26058      27362
## person_income          1  1528.86     26057      25834 < 2.2e-16 ***
## person_emp_length      1    43.34     26056      25790 4.603e-11 ***
## loan_amnt              1  1637.16     26055      24153 < 2.2e-16 ***
## loan_int_rate          1  2409.06     26054      21744 < 2.2e-16 ***
## person_home_ownership  3   918.28     26051      20826 < 2.2e-16 ***
## loan_intent            5   393.83     26046      20432 < 2.2e-16 ***
## loan_grade             6  1294.56     26040      19137 < 2.2e-16 ***
## loan_percent_income    1  1477.25     26039      17660 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La diferencia entre la desviación nula y la desviación residual muestra cómo se comporta nuestro modelo frente al modelo nulo, es decir, un modelo con sólo el intercepto. La prueba anova dice que añadir estas variables mejora el modelo de forma significativa, de hecho, los valores p bajos significan que las variables explican una gran parte de la variabilidad.

Ahora se puede analizar el ajuste e interpretar lo que nos dice el modelo. Como se ha analizado, el modelo sugiere que la mayoría de los p-valores muestran relevancia estadística en los resultados. Los únicos que no serían de la categoría HOMEIMPROVEMENT de la variable loan_intent y la categoría OTHER de la variable person_home_ownership, que no son significativos.

```
cbind(Estimate = round(coef(modelo_glm2), 5), OR = round(exp(coef(modelo_glm2)),
    5))
```

```
##                              Estimate          OR
## (Intercept)                 -4.21199     0.01482
## person_income                0.00000     1.00000
## person_emp_length           -0.01423     0.98587
## loan_amnt                   -0.00011     0.99989
## loan_int_rate                0.05570     1.05728
## person_home_ownershipOTHER   0.43033     1.53776
## person_home_ownershipOWN    -1.74974     0.17382
## person_home_ownershipRENT    0.83499     2.30478
```

```
## loan_intentEDUCATION       -0.81872      0.44100
## loan_intentHOMEIMPROVEMENT  0.07264      1.07535
## loan_intentMEDICAL         -0.15149      0.85943
## loan_intentPERSONAL        -0.59874      0.54950
## loan_intentVENTURE         -1.06814      0.34365
## loan_gradeB                 0.24616      1.27910
## loan_gradeC                 0.47674      1.61082
## loan_gradeD                 2.59458     13.39095
## loan_gradeE                 2.85180     17.31889
## loan_gradeF                 3.40255     30.04054
## loan_gradeG                 6.76201    864.37871
## loan_percent_income        13.43799 685561.75079
```

La columna Estimate calculada representa los coeficientes en forma de log-odds. Esto significa que cuando aumentamos la variable loan_int_rate en una unidad, podríamos esperar un cambio en las probabilidades logarítmicas de aproximadamente 0.05570. También podríamos utilizar esta información observando el signo de cada estimación, para entender si el efecto del predictor es positivo o negativo.

Según se puede observar en la tabla calculada, variables numéricas como person_emp_length o loan_amnt disminuyen las probabilidades de default, sin embargo, variables numéricas como loan_int_rate o loan_percent_income las incrementan.

Las variables categóricas deben interpretarse de forma un poco diferente. En este caso, la estimación representa el cambio en las probabilidades logarítmicas tomando como referencia una categoría base como, por ejemplo, en la regresión person_home_ownership-MORTAGE, loan_intent-DEBTCONSOLIDATION y loan_grade-A. En otras palabras, estar, por ejemplo, en la categoría loan_grade-B aumenta la log-odds de impago en 0.24616, con respecto a estar en loan_grade-A.

Según los resultados que obtenemos, cuando, por ejemplo, el parámetro del tipo de interés (variable loan_int_rate) aumenta una unidad (manteniendo constantes todos los demás predictores), las probabilidades de y = 1 (impago) son 1.05728 más altas o, dicho de otro modo, aumentan aproximadamente un 5.728%. La misma lógica se aplica a todas las demás variables, por lo que, por ejemplo, el aumento de la duración del empleo (variable person_emp_length) en una unidad conduce a una disminución de las probabilidades de impago de 0.98587.

Un elemento a tener en cuenta en el análisis es la variable loan_grade. Se puede observar que a medida que nos acercamos a grados inferiores, las probabilidades de impago aumentan enormemente. Como se dijo posteriormente, las variables categóricas deben interpretarse con respecto a la categoría base, así que en este caso comparando otros niveles con el grado A. Esta variable parece ser muy influyente y se podría decir que la graduación es adecuada y resulta útil para identificar posibles riesgos de impago para prestatarios y prestamistas.

Una interpretación de los coeficientes similar a la realizada sería:

```
round(exp(cbind(Estimate = coef(modelo_glm2), confint(modelo_glm2))),
    2)
```

```
##                              Estimate     2.5 %     97.5 %
## (Intercept)                      0.01      0.01       0.02
## person_income                    1.00      1.00       1.00
## person_emp_length                0.99      0.98       1.00
## loan_amnt                        1.00      1.00       1.00
## loan_int_rate                    1.06      1.03       1.09
## person_home_ownershipOTHER       1.54      0.82       2.83
## person_home_ownershipOWN         0.17      0.14       0.22
## person_home_ownershipRENT        2.30      2.11       2.52
```

```
## loan_intentEDUCATION            0.44      0.39       0.50
## loan_intentHOMEIMPROVEMENT      1.08      0.94       1.23
## loan_intentMEDICAL              0.86      0.76       0.97
## loan_intentPERSONAL             0.55      0.48       0.62
## loan_intentVENTURE              0.34      0.30       0.39
## loan_gradeB                     1.28      1.11       1.47
## loan_gradeC                     1.61      1.33       1.95
## loan_gradeD                    13.39     10.57      17.00
## loan_gradeE                    17.32     12.68      23.69
## loan_gradeF                    30.04     18.68      48.82
## loan_gradeG                   864.38    170.97   15822.07
## loan_percent_income        685561.75 385886.22 1204041.39
```

Los intervalos de confianza no se basan en un test de Wald (como en regresión tradicional), sino en un perfilado (profiling) de la log-likelihood, que es más preciso.

Predicción de valores del modelo:

```
head(predict(modelo_glm2))
```

```
##          1          2          3          4          5          6
## -0.2574225 -4.0274530 -2.1096365 -2.5847977 -2.2851013 -2.1276994
```

Probabilidad en escala de la salida:

```
head(predict(modelo_glm2, type = "response"))
```

```
##          1          2          3          4          5          6
## 0.43599742 0.01750768 0.10816373 0.07012325 0.09236441 0.10643359
```

A la hora de desarrollar modelos de predicción, la métrica más importante es determinar la eficacia del modelo para predecir la variable objetivo en observaciones fuera de la muestra. Para ello, podemos comparar la variable objetivo predicha con los valores observados mediante la matriz de confusión.

Evaluación del rendimiento predictivo del modelo GLM presentado con las datos de train:

```
fcr_train_glm$y_pred_probs <- predict(modelo_glm2, fcr_train_glm,
    type = "response")
fcr_train_glm$y_pred <- ifelse(fcr_train_glm$y_pred_probs > 0.5,
    1, 0)

# fcr_train_glm$y_pred_probs fcr_train_glm$y_pred
```

```
cm_train <- confusionMatrix(as.factor(fcr_train_glm$y_pred),
    as.factor(fcr_train_glm$loan_status), positive = "1")
cm_train$table
```

```
##            Reference
## Prediction     0     1
##          0 19406  2503
##          1   959  3191
```

La matriz de confusión es una tabla que describe el rendimiento de clasificación de cada modelo en los datos de prueba. En nuestro caso, los "1" y "0" de las filas representan si las personas han incumplido o no, mientras que las columnas "FALSO" y "VERDADERO" indican si predijimos que las personas incumplirían o no. La tabla siguiente sólo muestra las proporciones.

Más concretamente:

- **Verdaderos positivos (cuadrante inferior derecho):** son casos en los que predijimos que la gente incumpliría y lo hizo.
- **Verdaderos negativos (cuadrante superior izquierdo):** Predijimos que no habría impago y la gente no lo hizo.
- **Falsos positivos (cuadrante superior derecho):** Predijimos un impago, pero en realidad no se produjo. (Error de tipo I)
- **Falsos negativos (cuadrante inferior izquierdo):** Predijimos que no habría impago, pero sí lo hubo. (Error de tipo II)

Se utilizan diferentes métricas de evaluación del modelo:

```
# result
accuracy_modelo_glm2 <- cm_train$overall["Accuracy"] %>%
    round(4)
accuracy_modelo_glm2
```

```
## Accuracy
##   0.8671
```

```
# result
recall_modelo_glm2 <- cm_train$byClass["Recall"] %>%
    round(4)
recall_modelo_glm2
```

```
## Recall
## 0.5604
```

```
# result
precision_modelo_glm2 <- cm_train$byClass["Precision"] %>%
    round(4)
precision_modelo_glm2
```

```
## Precision
##    0.7689
```

```
# result
F1Score_modelo_glm2 <- ((2 * precision_modelo_glm2 * recall_modelo_glm2)/(precision_modelo_glm2 +
    recall_modelo_glm2)) %>%
    round(4)
F1Score_modelo_glm2
```

```
## Precision
##    0.6483
```

Viendo el valor de las metricas obtenidas (con un punto de corte en 0.5), el valor de Accuracy (número de predicciones correctas/número total de predicciones) se situa en el 87%, el de Precision (positivos verdaderos/(positivos verdaderos + falsos positivos)) se situa en un 77%, el de Recall o Sensitividad (positivos verdaderos/(positivos verdaderos/falsos negativos)) en un 56% y el F1 Score (considerado como una media armónica que combina los valores de la precisión o precision y de la exhaustividad o recall) en un 65%.

Con estos datos se entiende que con el modelo desarrollado, en alrededor del 87% de los casos este será capaz de predecir si un crédito va a ser impagado o no.

7.3. GLM - Cross Validation, Hiperparámetros y Evaluación del modelo

Una vez desarrollado el modelo, se trata de aplicar Cross Validation sobre el modelo de GLM y realizar una selección de hiperparámetros (se busca tener un modelo robusto, generalizable y comparable con el resto para la posterior selección del mejor):

Primero se analizan cuales son las posibles variables que tienes el modelo para tratar de configurar. Cómo se puede ver, el modelo GLM no tiene la posibilidad de ajustar hiperparámetros.

```
## https://machinelearningmastery.com/how-to-estimate-model-accuracy-in-r-using-the-caret-package/?mscl
## https://daviddalpiaz.github.io/r4sl/the-caret-package.html#classification

# Vemos hiperparámetros que se pueden configurar

modelLookup("glm")
```

```
##   model parameter     label forReg forClass probModel
## 1   glm parameter parameter   TRUE     TRUE      TRUE
```

Se crea el modelo con las variables seleccionadas como relevantes y haciendo Cross Validation on 5 particiones del dataset de train.

```
caret.glm <- train(as.factor(loan_status) ~ person_income + person_emp_length +
    loan_amnt + loan_int_rate + person_home_ownership + loan_intent +
    loan_grade + loan_percent_income, method = "glm", family = "binomial",
    data = fcr_train_glm, trControl = trainControl(method = "cv",
        number = 5, search = "grid", returnResamp = "final"))
caret.glm
```

```
## Generalized Linear Model
##
## 26059 samples
##     8 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 20847, 20848, 20847, 20847, 20847
## Resampling results:
##
##   Accuracy   Kappa
##   0.8668023  0.5677015
```

```
summary(caret.glm)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2910  -0.5238  -0.3031  -0.1232   3.4357
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -4.212e+00  1.426e-01 -29.527  < 2e-16 ***
## person_income                1.494e-06  5.540e-07   2.697 0.006994 **
## person_emp_length           -1.423e-02  5.343e-03  -2.663 0.007736 **
## loan_amnt                   -1.084e-04  4.996e-06 -21.708  < 2e-16 ***
## loan_int_rate                5.570e-02  1.446e-02   3.852 0.000117 ***
## person_home_ownershipOTHER   4.303e-01  3.162e-01   1.361 0.173558
## person_home_ownershipOWN    -1.750e+00  1.120e-01 -15.620  < 2e-16 ***
## person_home_ownershipRENT    8.350e-01  4.489e-02  18.602  < 2e-16 ***
## loan_intentEDUCATION        -8.187e-01  6.358e-02 -12.877  < 2e-16 ***
## loan_intentHOMEIMPROVEMENT   7.264e-02  7.071e-02   1.027 0.304260
## loan_intentMEDICAL          -1.515e-01  5.991e-02  -2.529 0.011446 *
## loan_intentPERSONAL         -5.987e-01  6.506e-02  -9.203  < 2e-16 ***
## loan_intentVENTURE          -1.068e+00  6.924e-02 -15.426  < 2e-16 ***
## loan_gradeB                  2.462e-01  7.121e-02   3.457 0.000547 ***
## loan_gradeC                  4.767e-01  9.753e-02   4.888 1.02e-06 ***
## loan_gradeD                  2.595e+00  1.213e-01  21.390  < 2e-16 ***
## loan_gradeE                  2.852e+00  1.594e-01  17.891  < 2e-16 ***
## loan_gradeF                  3.403e+00  2.448e-01  13.898  < 2e-16 ***
## loan_gradeG                  6.762e+00  1.044e+00   6.479 9.22e-11 ***
## loan_percent_income          1.344e+01  2.894e-01  46.437  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27362  on 26058  degrees of freedom
## Residual deviance: 17660  on 26039  degrees of freedom
## AIC: 17700
##
## Number of Fisher Scoring iterations: 6
```

Con estos datos se puede entender que con el modelo desarrollado, en alrededor del 86/87% de los casos este será capaz de predecir si un crédito va a ser impagado o no.

```
confusionMatrix(caret.glm)
```

```
## Cross-Validated (5 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
```

```
##            Reference
## Prediction    0    1
##          0 74.4  9.6
##          1  3.7 12.2
##
##  Accuracy (average) : 0.8668
```

Evaluación del rendimiento predictivo del modelo Decision Tree presentado con las datos de train (metrica de evaluación utilizada de referencia: "Accuracy", "Recall", "Precision", "F1" y "ROC", y punto de corte utilizado: 0.5):

```
fcr_train_glm$y_pred_probs2 <- predict(caret.glm, fcr_train_glm,
    type = "prob")
fcr_train_glm$y_pred_probs2 <- ifelse(fcr_train_glm$y_pred_probs2$`1` >
    0.5, fcr_train_glm$y_pred_probs2$`1`, 1 - fcr_train_glm$y_pred_probs2$`0`)
fcr_train_glm$y_pred2 <- ifelse(fcr_train_glm$y_pred_probs2 >
    0.5, 1, 0)

# fcr_train_glm$y_pred_probs2 fcr_train_glm$y_pred2
```

Se reproduce la matriz de confusión y las métricas de evaluación sobre el modelo final de GLM obtenido:

```
cm_train2 <- confusionMatrix(as.factor(fcr_train_glm$y_pred2),
    as.factor(fcr_train_glm$loan_status), positive = "1")
cm_train2$table
```

```
##            Reference
## Prediction     0     1
##          0 19406  2503
##          1   959  3191
```

```
# result
accuracy_modelo_glm2_tune <- cm_train2$overall["Accuracy"] %>%
    round(4)
accuracy_modelo_glm2_tune
```

```
## Accuracy
##   0.8671
```

```
# result
recall_modelo_glm2_tune <- cm_train2$byClass["Recall"] %>%
    round(4)
recall_modelo_glm2_tune
```

```
## Recall
## 0.5604
```

```
# result
precision_modelo_glm2_tune <- cm_train2$byClass["Precision"] %>%
    round(4)
precision_modelo_glm2_tune
```

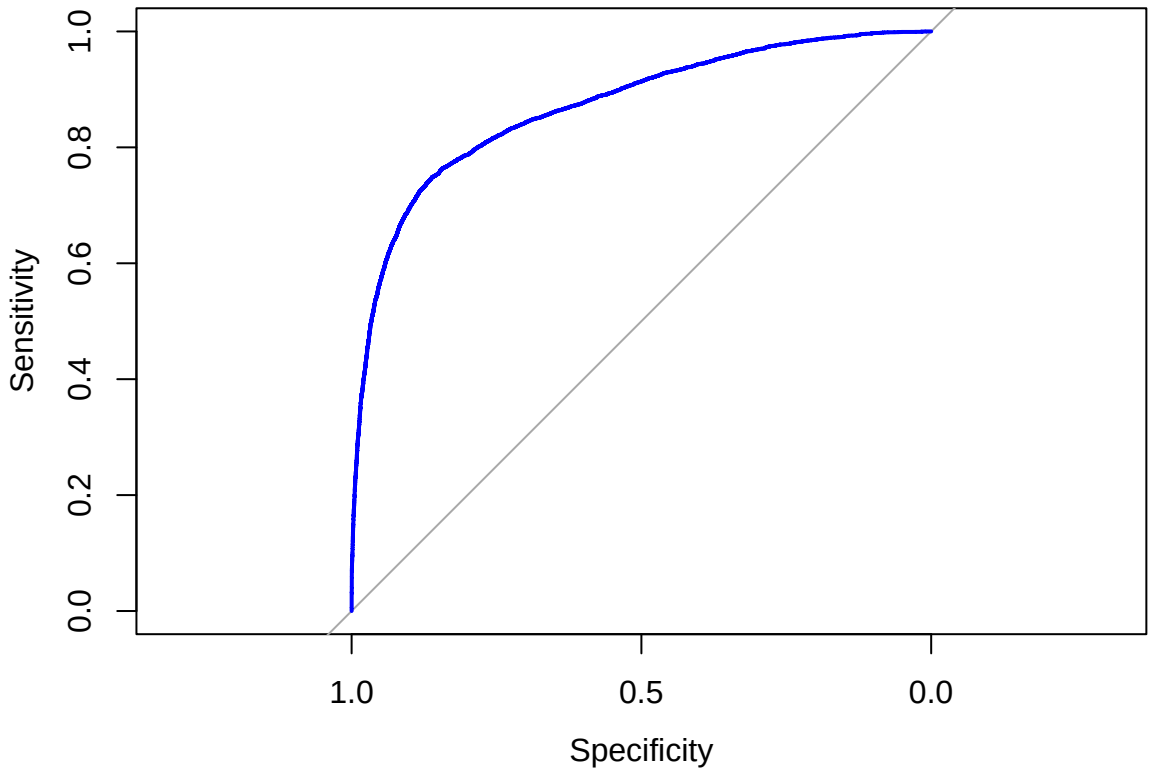
```
## Precision
##    0.7689
```

```
# result
F1Score_modelo_glm2_tune <- (2 * (precision_modelo_glm2_tune *
    recall_modelo_glm2_tune)/(precision_modelo_glm2_tune + recall_modelo_glm2_tune)) %>%
    round(4)
F1Score_modelo_glm2_tune
```

```
## Precision
##    0.6483
```

Se reproduce la curva ROC sobre el modelo final de GLM obtenido:

```
roc_glm <- plot.roc(as.numeric(fcr_train_glm$loan_status), as.numeric(fcr_train_glm$y_pred_probs2),
    col = "blue")
```




```
auc(roc_glm)
```

```
## Area under the curve: 0.8703
```

Se obtiene alrededor de un 87% de área bajo la curva.

El ROC es una curva que se genera trazando la tasa de verdaderos positivos (TPR) frente a la tasa de falsos positivos (FPR) en varios umbrales, mientras que el AUC es el área bajo la curva ROC. Dicho de otro modo,

el ROC traza el porcentaje de verdaderos positivos predichos con exactitud por el modelo a medida que el umbral de probabilidad de predicción se reduce de 1 a 0. Muestra la compensación entre la tasa a la que se puede predecir correctamente algo con la tasa de predecir incorrectamente algo.

Por lo tanto, para un buen modelo, la curva debería aumentar de forma pronunciada, indicando que el TPR (eje Y) aumenta más rápido que el FPR (eje X) a medida que disminuye la puntuación de corte. Cuanto mayor sea el área bajo la curva ROC, mejor será la capacidad predictiva del modelo. Esa métrica oscila entre 0,50 y 1,00, y los valores superiores a 0,80 indican que el modelo hace un buen trabajo al discriminar entre las dos categorías que componen nuestra variable objetivo.

8. DECISION TREE

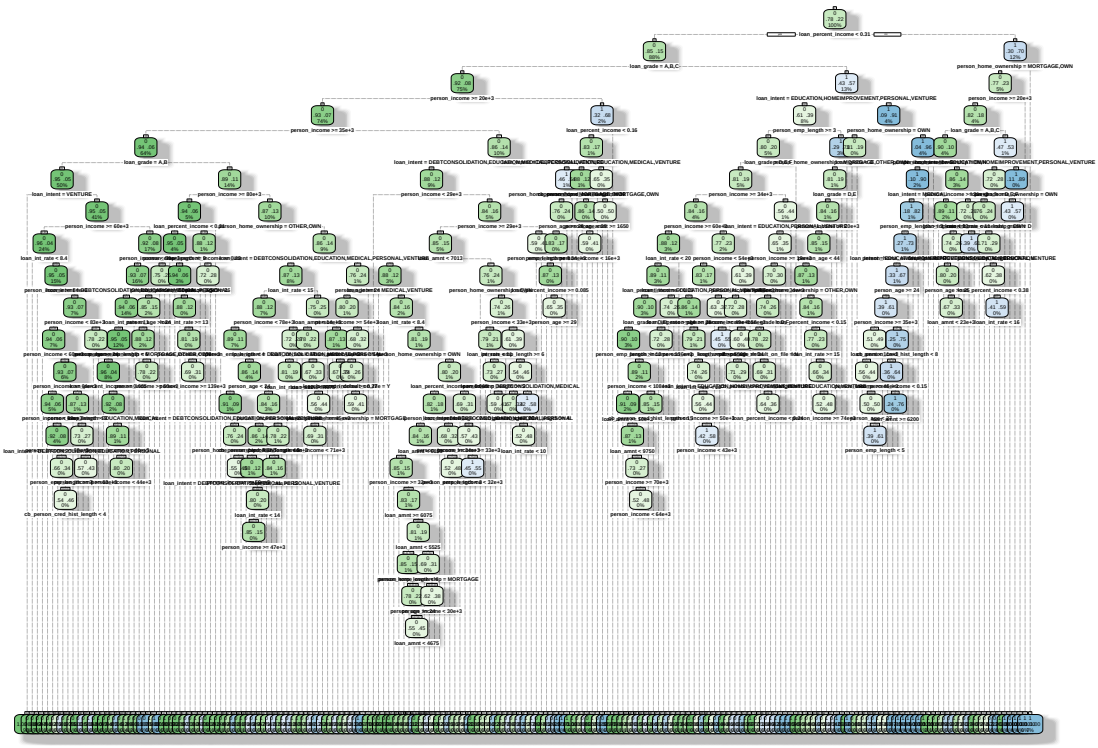
Primero se crean unos datos de train específicos para ser usados en el desarrollo del modelo de árbol de decisión, y así mantener los originales sin modificar.

```
fcr_train_tree <- fcr_train
fcr_train_tree
```

```
## # A tibble: 26,059 x 12
##    person_age person_i~1 perso~2 perso~3 loan_~4 loan_~5 loan_~6 loan_~7 loan_~8
##         <dbl>      <dbl> <fct>     <dbl> <fct>   <fct>     <dbl>   <dbl>   <dbl>
##  1         28      44000 RENT          2 MEDICAL C         10000   13.5        1
##  2         21      35000 OWN           5 VENTURE B          8000    9.91       0
##  3         25      96000 MORTGA~       6 HOMEIM~ C         21000   14.6        0
##  4         22      67000 OWN           5 EDUCAT~ D          7500   16.3        0
##  5         24      52800 RENT          8 PERSON~ A          9000    7.49       0
##  6         27      50004 RENT         12 DEBTCO~ B          3200   11.5        0
##  7         23      55488 RENT          4 MEDICAL D          5000   15.2        1
##  8         28      70000 RENT          2 DEBTCO~ B          6000   10.4        0
##  9         22      55000 MORTGA~       6 PERSON~ C         13000   13.8        0
## 10         26      43200 RENT          5 EDUCAT~ C          3200   14.4        0
## # ... with 26,049 more rows, 3 more variables: loan_percent_income <dbl>,
## #   cb_person_default_on_file <fct>, cb_person_cred_hist_length <dbl>, and
## #   abbreviated variable names 1: person_income, 2: person_home_ownership,
## #   3: person_emp_length, 4: loan_intent, 5: loan_grade, 6: loan_amnt,
## #   7: loan_int_rate, 8: loan_status
```

Se crea un modelo de árbol de decisión inicial básico y sin podar utilizando las 11 variables predictoras que tenemos a nuestra disposición y utilizando el método de "Gini" como medida de la impureza:

```
# árbol de clasificación con las opciones por defecto (cp =
# 0.0001 y split = 'gini') con el comando:
tree = rpart(as.factor(loan_status) ~ ., data = fcr_train_tree,
    cp = 1e-04)
rpart.plot(tree, nn = TRUE, extra = 104, box.palette = "GnBu",
    branch.lty = 3, shadow.col = "gray")
```



```
tree
```

```
## n= 26059
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##       1) root 26059 5694 0 (0.781495836 0.218504164)
##         2) loan_percent_income< 0.305 22949 3525 0 (0.846398536 0.153601464)
##           4) loan_grade=A,B,C 19672 1666 0 (0.915311102 0.084688898)
##             8) person_income>=19996 19191 1340 0 (0.930175603 0.069824397)
##              16) person_income>=34850 16613  970 0 (0.941611991 0.058388009)
##                32) loan_grade=A,B 12942  583 0 (0.954952867 0.045047133)
##                  64) loan_intent=VENTURE 2275    0 0 (1.000000000 0.000000000) *
##                  65) loan_intent=DEBTCONSOLIDATION,EDUCATION,HOMEIMPROVEMENT,MEDICAL,PERSONAL 10667
##                   130) person_income>=59704 6364  255 0 (0.959930861 0.040069139)
##                     260) loan_int_rate< 8.435 2525   50 0 (0.980198020 0.019801980) *
##                     261) loan_int_rate>=8.435 3839  205 0 (0.946600677 0.053399323)
##                       522) person_income>=83602 1919   74 0 (0.961438249 0.038561751) *
##                       523) person_income< 83602 1920  131 0 (0.931770833 0.068229167)
##                        1046) person_income< 82725 1880  122 0 (0.935106383 0.064893617)
##                          2092) person_income< 60054 266    3 0 (0.988721805 0.011278195) *
##                          2093) person_income>=60054 1614  119 0 (0.926270136 0.073729864)
##                            4186) person_income>=64026 1347   85 0 (0.936896808 0.063103192)
##                              8372) person_income< 68843 362   10 0 (0.972375691 0.027624309) *
##                              8373) person_income>=68843 985   75 0 (0.923857868 0.076142132)
```

```
##                             16746) person_income>=69430 944   61 0 (0.935381356 0.064618644) *
##                             16747) person_income< 69430 41   14 0 (0.658536585 0.341463415)
##                               33494) person_emp_length>=6.5 13    1 0 (0.923076923 0.076923077) *
##                               33495) person_emp_length< 6.5 28   13 0 (0.535714286 0.464285714)
##                                 66990) cb_person_cred_hist_length< 3.5 11    3 0 (0.727272727 0.27
##                                 66991) cb_person_cred_hist_length>=3.5 17    7 1 (0.411764706 0.58
##                           4187) person_income< 64026 267   34 0 (0.872659176 0.127340824)
##                             8374) person_emp_length< 8.5 219   21 0 (0.904109589 0.095890411) *
##                             8375) person_emp_length>=8.5 48   13 0 (0.729166667 0.270833333)
##                               16750) loan_intent=DEBTCONSOLIDATION,EDUCATION,PERSONAL 27    4 0 (0.85
##                               16751) loan_intent=HOMEIMPROVEMENT,MEDICAL 21    9 0 (0.571428571 0.42
##                                 33502) person_income>=62951.5 7    0 0 (1.000000000 0.000000000) *
##                                 33503) person_income< 62951.5 14    5 1 (0.357142857 0.642857143) *
##                         1047) person_income>=82725 40    9 0 (0.775000000 0.225000000)
##                           2094) person_age>=23.5 31    3 0 (0.903225806 0.096774194) *
##                           2095) person_age< 23.5 9    3 1 (0.333333333 0.666666667) *
##                       131) person_income< 59704 4303  328 0 (0.923774111 0.076225889)
##                         262) person_income< 58629 4204  303 0 (0.927925785 0.072074215)
##                           524) loan_intent=DEBTCONSOLIDATION,EDUCATION,MEDICAL,PERSONAL 3771  238 0 (0.93
##                           1048) loan_int_rate< 11.41 3225  176 0 (0.945426357 0.054573643)
##                             2096) person_emp_length< 5.5 2147   95 0 (0.955752212 0.044247788)
##                               4192) loan_percent_income>=0.085 1686   60 0 (0.964412811 0.035587189) *
##                               4193) loan_percent_income< 0.085 461   35 0 (0.924078091 0.075921909)
##                                 8386) loan_intent=EDUCATION,MEDICAL 255   12 0 (0.952941176 0.047058824
##                                 8387) loan_intent=DEBTCONSOLIDATION,PERSONAL 206   23 0 (0.888349515 0.1
##                                 16774) person_income>=44484 124    7 0 (0.943548387 0.056451613) *
##                                 16775) person_income< 44484 82   16 0 (0.804878049 0.195121951)
##                                   33550) person_income< 43750 75   10 0 (0.866666667 0.133333333) *
##                                   33551) person_income>=43750 7    1 1 (0.142857143 0.857142857) *
##                             2097) person_emp_length>=5.5 1078   81 0 (0.924860853 0.075139147) *
##                           1049) loan_int_rate>=11.41 546   62 0 (0.886446886 0.113553114) *
##                           525) loan_intent=HOMEIMPROVEMENT 433   65 0 (0.849884527 0.150115473)
##                           1050) person_age>=23.5 416   48 0 (0.884615385 0.115384615)
##                             2100) person_home_ownership=MORTGAGE,OTHER,OWN 289   20 0 (0.930795848 0.069
##                             2101) person_home_ownership=RENT 127   28 0 (0.779527559 0.220472441)
##                               4202) person_income>=49500 109   10 0 (0.908256881 0.091743119) *
##                               4203) person_income< 49500 18    0 1 (0.000000000 1.000000000) *
##                           1051) person_age< 23.5 17    0 1 (0.000000000 1.000000000) *
##                         263) person_income>=58629 99   25 0 (0.747474747 0.252525253)
##                           526) person_income>=58653 92   18 0 (0.804347826 0.195652174) *
##                           527) person_income< 58653 7    0 1 (0.000000000 1.000000000) *
##                     33) loan_grade=C 3671  387 0 (0.894579134 0.105420866)
##                       66) person_income>=79996 1190   69 0 (0.942016807 0.057983193)
##                       132) loan_percent_income< 0.205 1054   53 0 (0.949715370 0.050284630)
##                         264) person_emp_length>=8.5 235    4 0 (0.982978723 0.017021277) *
##                         265) person_emp_length< 8.5 819   49 0 (0.940170940 0.059829060)
##                           530) loan_percent_income>=0.045 691   34 0 (0.950795948 0.049204052) *
##                           531) loan_percent_income< 0.045 128   15 0 (0.882812500 0.117187500)
##                           1062) loan_int_rate>=13.29 75    5 0 (0.933333333 0.066666667) *
##                           1063) loan_int_rate< 13.29 53   10 0 (0.811320755 0.188679245)
##                             2126) person_income< 107500 21    0 0 (1.000000000 0.000000000) *
##                             2127) person_income>=107500 32   10 0 (0.687500000 0.312500000)
##                               4254) person_income>=139002 20    2 0 (0.900000000 0.100000000) *
##                               4255) person_income< 139002 12    4 1 (0.333333333 0.666666667) *
```

```
##                   133) loan_percent_income>=0.205 136   16 0 (0.882352941 0.117647059)
##                     266) loan_percent_income>=0.215 111    9 0 (0.918918919 0.081081081) *
##                     267) loan_percent_income< 0.215 25    7 0 (0.720000000 0.280000000)
##                       534) person_age>=26 17    2 0 (0.882352941 0.117647059) *
##                       535) person_age< 26 8    3 1 (0.375000000 0.625000000) *
##                  67) person_income< 79996 2481  318 0 (0.871825877 0.128174123)
##                   134) person_home_ownership=OTHER,OWN 193    0 0 (1.000000000 0.000000000) *
##                   135) person_home_ownership=MORTGAGE,RENT 2288  318 0 (0.861013986 0.138986014)
##                     270) loan_intent=DEBTCONSOLIDATION,EDUCATION,MEDICAL,PERSONAL,VENTURE 2044  259 0
##                       540) loan_int_rate< 15.025 1915  227 0 (0.881462141 0.118537859)
##                         1080) person_income< 77951.5 1869  214 0 (0.885500268 0.114499732)
##                           2160) person_emp_length< 3.5 924   84 0 (0.909090909 0.090909091) *
##                           2161) person_emp_length>=3.5 945  130 0 (0.862433862 0.137566138)
##                             4322) person_age< 23.5 274   24 0 (0.912408759 0.087591241)
##                               8644) loan_intent=DEBTCONSOLIDATION,EDUCATION,PERSONAL,VENTURE 228   13
##                               8645) loan_intent=MEDICAL 46   11 0 (0.760869565 0.239130435)
##                                 17290) person_home_ownership=RENT 26    2 0 (0.923076923 0.076923077) *
##                                 17291) person_home_ownership=MORTGAGE 20    9 0 (0.550000000 0.450000000
##                                   34582) person_income>=59000 11    2 0 (0.818181818 0.181818182) *
##                                   34583) person_income< 59000 9    2 1 (0.222222222 0.777777778) *
##                             4323) person_age>=23.5 671  106 0 (0.842026826 0.157973174)
##                               8646) person_income>=44423.5 515   72 0 (0.860194175 0.139805825)
##                                 17292) cb_person_cred_hist_length< 7.5 320   37 0 (0.884375000 0.115625
##                                   34584) loan_intent=DEBTCONSOLIDATION,MEDICAL,PERSONAL,VENTURE 251   1
##                                   34585) loan_intent=EDUCATION 69   14 0 (0.797101449 0.202898551)
##                                     69170) loan_int_rate< 14.31 62    9 0 (0.854838710 0.145161290)
##                                      138340) person_income>=46900 55    5 0 (0.909090909 0.090909091) *
##                                      138341) person_income< 46900 7    3 1 (0.428571429 0.571428571) *
##                                     69171) loan_int_rate>=14.31 7    2 1 (0.285714286 0.714285714) *
##                                 17293) cb_person_cred_hist_length>=7.5 195   35 0 (0.820512821 0.17948
##                               8647) person_income< 44423.5 156   34 0 (0.782051282 0.217948718)
##                                 17294) person_income< 43606.5 137   22 0 (0.839416058 0.160583942)
##                                   34588) loan_int_rate>=11.94 120   13 0 (0.891666667 0.108333333) *
##                                   34589) loan_int_rate< 11.94 17    8 1 (0.470588235 0.529411765) *
##                                 17295) person_income>=43606.5 19    7 1 (0.368421053 0.631578947) *
##                         1081) person_income>=77951.5 46   13 0 (0.717391304 0.282608696)
##                           2162) loan_int_rate< 13.82 37    7 0 (0.810810811 0.189189189)
##                             4324) loan_int_rate>=11.875 30    3 0 (0.900000000 0.100000000) *
##                             4325) loan_int_rate< 11.875 7    3 1 (0.428571429 0.571428571) *
##                           2163) loan_int_rate>=13.82 9    3 1 (0.333333333 0.666666667) *
##                       541) loan_int_rate>=15.025 129   32 0 (0.751937984 0.248062016)
##                         1082) loan_amnt< 14300 120   26 0 (0.783333333 0.216666667)
##                           2164) loan_intent=DEBTCONSOLIDATION,MEDICAL,PERSONAL 72   10 0 (0.861111111
##                           2165) loan_intent=EDUCATION,VENTURE 48   16 0 (0.666666667 0.333333333)
##                             4330) loan_amnt< 4675 14    1 0 (0.928571429 0.071428571) *
##                             4331) loan_amnt>=4675 34   15 0 (0.558823529 0.441176471)
##                               8662) person_income>=44800 26    8 0 (0.692307692 0.307692308)
##                                 17324) person_income< 71000 18    3 0 (0.833333333 0.166666667) *
##                                 17325) person_income>=71000 8    3 1 (0.375000000 0.625000000) *
##                               8663) person_income< 44800 8    1 1 (0.125000000 0.875000000) *
##                         1083) loan_amnt>=14300 9    3 1 (0.333333333 0.666666667) *
##                     271) loan_intent=HOMEIMPROVEMENT 244   59 0 (0.758196721 0.241803279)
##                       542) person_age>=23.5 230   45 0 (0.804347826 0.195652174)
##                         1084) person_income>=54131.5 148   19 0 (0.871621622 0.128378378)
```

```
##                         2168) loan_amnt< 15950 127   12 0 (0.905511811 0.094488189) *
##                         2169) loan_amnt>=15950 21    7 0 (0.666666667 0.333333333)
##                           4338) loan_percent_income>=0.265 13    2 0 (0.846153846 0.153846154) *
##                           4339) loan_percent_income< 0.265 8    3 1 (0.375000000 0.625000000) *
##                       1085) person_income< 54131.5 82   26 0 (0.682926829 0.317073171)
##                         2170) person_income< 53685 72   19 0 (0.736111111 0.263888889)
##                           4340) cb_person_default_on_file=Y 35    4 0 (0.885714286 0.114285714) *
##                           4341) cb_person_default_on_file=N 37   15 0 (0.594594595 0.405405405)
##                             8682) person_home_ownership=MORTGAGE 23    7 0 (0.695652174 0.304347826)
##                             8683) person_home_ownership=RENT 14    6 1 (0.428571429 0.571428571) *
##                         2171) person_income>=53685 10    3 1 (0.300000000 0.700000000) *
##                       543) person_age< 23.5 14    0 1 (0.000000000 1.000000000) *
##                 17) person_income< 34850 2578  370 0 (0.856477890 0.143522110)
##                   34) loan_intent=DEBTCONSOLIDATION,EDUCATION,MEDICAL,PERSONAL,VENTURE 2410  279 0 (0.88
##                     68) person_income< 28984 1180   86 0 (0.927118644 0.072881356) *
##                     69) person_income>=28984 1230  193 0 (0.843089431 0.156910569)
##                      138) person_income>=29002 1186  175 0 (0.852445194 0.147554806)
##                        276) loan_amnt< 7012.5 912  110 0 (0.879385965 0.120614035)
##                          552) loan_intent=MEDICAL,VENTURE 388   26 0 (0.932989691 0.067010309) *
##                          553) loan_intent=DEBTCONSOLIDATION,EDUCATION,PERSONAL 524   84 0 (0.839694656 0
##                           1106) loan_int_rate< 8.405 136   11 0 (0.919117647 0.080882353) *
##                           1107) loan_int_rate>=8.405 388   73 0 (0.811855670 0.188144330)
##                             2214) person_home_ownership=OWN 23    0 0 (1.000000000 0.000000000) *
##                             2215) person_home_ownership=MORTGAGE,RENT 365   73 0 (0.800000000 0.2000000
##                               4430) loan_percent_income>=0.065 314   57 0 (0.818471338 0.181528662)
##                                 8860) person_income< 33840.5 280   46 0 (0.835714286 0.164285714)
##                                  17720) loan_amnt< 6850 270   41 0 (0.848148148 0.151851852)
##                                    35440) person_income>=32448 51    3 0 (0.941176471 0.058823529) *
##                                    35441) person_income< 32448 219   38 0 (0.826484018 0.173515982)
##                                      70882) loan_amnt>=6075 18    0 0 (1.000000000 0.000000000) *
##                                      70883) loan_amnt< 6075 201   38 0 (0.810945274 0.189054726)
##                                       141766) loan_amnt< 5525 156   24 0 (0.846153846 0.153846154)
##                                         283532) person_emp_length< 3.5 93   10 0 (0.892473118 0.1075268
##                                         283533) person_emp_length>=3.5 63   14 0 (0.777777778 0.2222222
##                                           567066) person_age>=23.5 43    5 0 (0.883720930 0.116279070) *
##                                           567067) person_age< 23.5 20    9 0 (0.550000000 0.450000000)
##                                            1134134) loan_amnt< 4675 12    2 0 (0.833333333 0.166666667)
##                                            1134135) loan_amnt>=4675 8    1 1 (0.125000000 0.875000000) *
##                                       141767) loan_amnt>=5525 45   14 0 (0.688888889 0.311111111)
##                                         283534) person_home_ownership=MORTGAGE 8    0 0 (1.000000000 0.0
##                                         283535) person_home_ownership=RENT 37   14 0 (0.621621622 0.3783
##                                           567070) person_income< 30450 24    6 0 (0.750000000 0.2500000
##                                           567071) person_income>=30450 13    5 1 (0.384615385 0.6153846
##                                  17721) loan_amnt>=6850 10    5 0 (0.500000000 0.500000000) *
##                                 8861) person_income>=33840.5 34   11 0 (0.676470588 0.323529412)
##                                  17722) person_income>=34076 13    1 0 (0.923076923 0.076923077) *
##                                  17723) person_income< 34076 21   10 0 (0.523809524 0.476190476)
##                                    35446) person_emp_length< 2.5 13    4 0 (0.692307692 0.307692308) *
##                                    35447) person_emp_length>=2.5 8    2 1 (0.250000000 0.750000000) *
##                               4431) loan_percent_income< 0.065 51   16 0 (0.686274510 0.313725490)
##                                 8862) person_age< 23.5 21    3 0 (0.857142857 0.142857143) *
##                                 8863) person_age>=23.5 30   13 0 (0.566666667 0.433333333)
##                                  17726) person_income>=32820 10    2 0 (0.800000000 0.200000000) *
##                                  17727) person_income< 32820 20    9 1 (0.450000000 0.550000000)
```

```
##                                35454) person_income< 31500 11    4 0 (0.636363636 0.363636364) *
##                                35455) person_income>=31500 9    2 1 (0.222222222 0.777777778) *
##                     277) loan_amnt>=7012.5 274   65 0 (0.762773723 0.237226277)
##                       554) person_home_ownership=OWN 25    0 0 (1.000000000 0.000000000) *
##                       555) person_home_ownership=MORTGAGE,RENT 249   65 0 (0.738955823 0.261044177)
##                        1110) person_income< 33319.5 175   36 0 (0.794285714 0.205714286)
##                          2220) loan_int_rate< 10.85 69    7 0 (0.898550725 0.101449275) *
##                          2221) loan_int_rate>=10.85 106   29 0 (0.726415094 0.273584906)
##                            4442) person_emp_length>=1.5 74   16 0 (0.783783784 0.216216216) *
##                            4443) person_emp_length< 1.5 32   13 0 (0.593750000 0.406250000)
##                              8886) loan_intent=DEBTCONSOLIDATION,MEDICAL,PERSONAL 23    6 0 (0.73913
##                              8887) loan_intent=EDUCATION,VENTURE 9    2 1 (0.222222222 0.777777778) *
##                        1111) person_income>=33319.5 74   29 0 (0.608108108 0.391891892)
##                          2222) person_emp_length>=5.5 13    1 0 (0.923076923 0.076923077) *
##                          2223) person_emp_length< 5.5 61   28 0 (0.540983607 0.459016393)
##                            4446) loan_intent=DEBTCONSOLIDATION,MEDICAL 30   10 0 (0.666666667 0.3333
##                              8892) person_age< 27.5 22    5 0 (0.772727273 0.227272727) *
##                              8893) person_age>=27.5 8    3 1 (0.375000000 0.625000000) *
##                            4447) loan_intent=EDUCATION,PERSONAL,VENTURE 31   13 1 (0.419354839 0.580
##                              8894) cb_person_cred_hist_length>=3.5 21   10 0 (0.523809524 0.47619047
##                               17788) loan_int_rate< 9.96 9    2 0 (0.777777778 0.222222222) *
##                               17789) loan_int_rate>=9.96 12    4 1 (0.333333333 0.666666667) *
##                              8895) cb_person_cred_hist_length< 3.5 10    2 1 (0.200000000 0.800000000
##                   139) person_income< 29002 44   18 0 (0.590909091 0.409090909)
##                     278) person_emp_length>=2.5 26    7 0 (0.730769231 0.269230769) *
##                     279) person_emp_length< 2.5 18    7 1 (0.388888889 0.611111111) *
##                  35) loan_intent=HOMEIMPROVEMENT 168   77 1 (0.458333333 0.541666667)
##                    70) person_home_ownership=MORTGAGE,OWN 101   24 0 (0.762376238 0.237623762)
##                     140) person_age>=23.5 93   16 0 (0.827956989 0.172043011)
##                       280) person_income< 33800 86   11 0 (0.872093023 0.127906977)
##                         560) loan_percent_income>=0.085 66    4 0 (0.939393939 0.060606061) *
##                         561) loan_percent_income< 0.085 20    7 0 (0.650000000 0.350000000)
##                          1122) person_age>=28.5 13    2 0 (0.846153846 0.153846154) *
##                          1123) person_age< 28.5 7    2 1 (0.285714286 0.714285714) *
##                       281) person_income>=33800 7    2 1 (0.285714286 0.714285714) *
##                     141) person_age< 23.5 8    0 1 (0.000000000 1.000000000) *
##                    71) person_home_ownership=RENT 67    0 1 (0.000000000 1.000000000) *
##                9) person_income< 19996 481  155 1 (0.322245322 0.677754678)
##                 18) loan_percent_income< 0.155 186   31 0 (0.833333333 0.166666667)
##                   36) loan_intent=DEBTCONSOLIDATION,EDUCATION,MEDICAL,VENTURE 149   18 0 (0.879194631 0
##                     72) cb_person_cred_hist_length>=9.5 24    0 0 (1.000000000 0.000000000) *
##                     73) cb_person_cred_hist_length< 9.5 125   18 0 (0.856000000 0.144000000)
##                      146) person_age< 27.5 103    9 0 (0.912621359 0.087378641) *
##                      147) person_age>=27.5 22    9 0 (0.590909091 0.409090909)
##                        294) person_income< 15828 10    2 0 (0.800000000 0.200000000) *
##                        295) person_income>=15828 12    5 1 (0.416666667 0.583333333) *
##                   37) loan_intent=HOMEIMPROVEMENT,PERSONAL 37   13 0 (0.648648649 0.351351351)
##                     74) person_home_ownership=MORTGAGE,OWN 11    0 0 (1.000000000 0.000000000) *
##                     75) person_home_ownership=RENT 26   13 0 (0.500000000 0.500000000)
##                      150) loan_amnt>=1650 14    4 0 (0.714285714 0.285714286) *
##                      151) loan_amnt< 1650 12    3 1 (0.250000000 0.750000000) *
##                 19) loan_percent_income>=0.155 295    0 1 (0.000000000 1.000000000) *
##               5) loan_grade=D,E,F,G 3277 1418 1 (0.432712847 0.567287153)
##                10) loan_intent=EDUCATION,HOMEIMPROVEMENT,PERSONAL,VENTURE 2161  848 0 (0.607589079 0.392
```

```
##               20) person_emp_length>=2.5 1352  275 0 (0.796597633 0.203402367)
##                 40) loan_grade=D,E,F 1332  256 0 (0.807807808 0.192192192)
##                   80) person_income>=34240 1172  185 0 (0.842150171 0.157849829)
##                    160) person_income>=59913.5 748   89 0 (0.881016043 0.118983957)
##                      320) loan_int_rate< 19.58 740   85 0 (0.885135135 0.114864865)
##                        640) loan_percent_income< 0.255 686   71 0 (0.896501458 0.103498542)
##                         1280) loan_grade=D,E 657   63 0 (0.904109589 0.095890411)
##                           2560) person_emp_length>=11.5 79    2 0 (0.974683544 0.025316456) *
##                           2561) person_emp_length< 11.5 578   61 0 (0.894463668 0.105536332)
##                             5122) person_income< 107640 423   37 0 (0.912529551 0.087470449)
##                               10244) loan_amnt>=10412.5 210    9 0 (0.957142857 0.042857143) *
##                               10245) loan_amnt< 10412.5 213   28 0 (0.868544601 0.131455399)
##                                 20490) loan_amnt< 9750 158   13 0 (0.917721519 0.082278481) *
##                                 20491) loan_amnt>=9750 55   15 0 (0.727272727 0.272727273)
##                                   40982) person_income>=69500 34    5 0 (0.852941176 0.147058824) *
##                                   40983) person_income< 69500 21   10 0 (0.523809524 0.476190476)
##                                     81966) person_income< 63500 10    3 0 (0.700000000 0.300000000) *
##                                     81967) person_income>=63500 11    4 1 (0.363636364 0.636363636) *
##                             5123) person_income>=107640 155   24 0 (0.845161290 0.154838710)
##                               10246) cb_person_cred_hist_length< 14.5 148   20 0 (0.864864865 0.135135
##                               10247) cb_person_cred_hist_length>=14.5 7    3 1 (0.428571429 0.57142857
##                         1281) loan_grade=F 29    8 0 (0.724137931 0.275862069)
##                           2562) person_income< 135500 22    3 0 (0.863636364 0.136363636) *
##                           2563) person_income>=135500 7    2 1 (0.285714286 0.714285714) *
##                        641) loan_percent_income>=0.255 54   14 0 (0.740740741 0.259259259)
##                         1282) loan_int_rate>=13.725 47    9 0 (0.808510638 0.191489362) *
##                         1283) loan_int_rate< 13.725 7    2 1 (0.285714286 0.714285714) *
##                      321) loan_int_rate>=19.58 8    4 0 (0.500000000 0.500000000) *
##                    161) person_income< 59913.5 424   96 0 (0.773584906 0.226415094)
##                      322) person_income< 53817 309   51 0 (0.834951456 0.165048544)
##                        644) loan_intent=EDUCATION,PERSONAL,VENTURE 274   38 0 (0.861313869 0.13868613
##                         1288) person_age>=24.5 143   10 0 (0.930069930 0.069930070) *
##                         1289) person_age< 24.5 131   28 0 (0.786259542 0.213740458)
##                           2578) person_emp_length>=6.5 31    2 0 (0.935483871 0.064516129) *
##                           2579) person_emp_length< 6.5 100   26 0 (0.740000000 0.260000000)
##                             5158) loan_int_rate< 15.97 68   12 0 (0.823529412 0.176470588) *
##                             5159) loan_int_rate>=15.97 32   14 0 (0.562500000 0.437500000)
##                               10318) person_income>=49500 8    0 0 (1.000000000 0.000000000) *
##                               10319) person_income< 49500 24   10 1 (0.416666667 0.583333333)
##                                 20638) person_income< 43000 13    5 0 (0.615384615 0.384615385) *
##                                 20639) person_income>=43000 11    2 1 (0.181818182 0.818181818) *
##                        645) loan_intent=HOMEIMPROVEMENT 35   13 0 (0.628571429 0.371428571)
##                         1290) person_income>=49000 13    1 0 (0.923076923 0.076923077) *
##                         1291) person_income< 49000 22   10 1 (0.454545455 0.545454545)
##                           2582) loan_amnt>=5500 15    6 0 (0.600000000 0.400000000) *
##                           2583) loan_amnt< 5500 7    1 1 (0.142857143 0.857142857) *
##                      323) person_income>=53817 115   45 0 (0.608695652 0.391304348)
##                        646) person_income>=54005.5 89   25 0 (0.719101124 0.280898876)
##                         1292) person_income< 56002 32    2 0 (0.937500000 0.062500000) *
##                         1293) person_income>=56002 57   23 0 (0.596491228 0.403508772)
##                           2586) person_age>=23.5 41   12 0 (0.707317073 0.292682927)
##                             5172) loan_intent=EDUCATION,HOMEIMPROVEMENT,VENTURE 31    6 0 (0.80645161
##                             5173) loan_intent=PERSONAL 10    4 1 (0.400000000 0.600000000) *
##                           2587) person_age< 23.5 16    5 1 (0.312500000 0.687500000) *
```

```
##                       647) person_income< 54005.5 26    6 1 (0.230769231 0.769230769) *
##                   81) person_income< 34240 160   71 0 (0.556250000 0.443750000)
##                    162) loan_intent=EDUCATION,PERSONAL,VENTURE 136   48 0 (0.647058824 0.352941176)
##                      324) person_income>=19098 121   36 0 (0.702479339 0.297520661)
##                        648) person_income< 33998 109   28 0 (0.743119266 0.256880734)
##                         1296) loan_grade=D,F 96   21 0 (0.781250000 0.218750000)
##                           2592) cb_person_default_on_file=N 47    6 0 (0.872340426 0.127659574) *
##                           2593) cb_person_default_on_file=Y 49   15 0 (0.693877551 0.306122449)
##                             5186) cb_person_cred_hist_length>=9.5 7    0 0 (1.000000000 0.000000000) *
##                             5187) cb_person_cred_hist_length< 9.5 42   15 0 (0.642857143 0.357142857)
##                              10374) loan_percent_income< 0.235 34   10 0 (0.705882353 0.294117647) *
##                              10375) loan_percent_income>=0.235 8    3 1 (0.375000000 0.625000000) *
##                         1297) loan_grade=E 13    6 1 (0.461538462 0.538461538) *
##                        649) person_income>=33998 12    4 1 (0.333333333 0.666666667) *
##                      325) person_income< 19098 15    3 1 (0.200000000 0.800000000) *
##                    163) loan_intent=HOMEIMPROVEMENT 24    1 1 (0.041666667 0.958333333) *
##                41) loan_grade=G 20    1 1 (0.050000000 0.950000000) *
##              21) person_emp_length< 2.5 809  236 1 (0.291718171 0.708281829)
##                42) person_home_ownership=MORTGAGE,OTHER,OWN 291   55 0 (0.810996564 0.189003436)
##                  84) loan_grade=D,E 274   45 0 (0.835766423 0.164233577)
##                   168) person_income>=19600 262   38 0 (0.854961832 0.145038168)
##                     336) person_age< 43.5 250   33 0 (0.868000000 0.132000000)
##                       672) person_home_ownership=OTHER,OWN 56    2 0 (0.964285714 0.035714286) *
##                       673) person_home_ownership=MORTGAGE 194   31 0 (0.840206186 0.159793814)
##                        1346) loan_percent_income< 0.145 90    7 0 (0.922222222 0.077777778) *
##                        1347) loan_percent_income>=0.145 104   24 0 (0.769230769 0.230769231)
##                          2694) loan_int_rate>=15.005 63   10 0 (0.841269841 0.158730159) *
##                          2695) loan_int_rate< 15.005 41   14 0 (0.658536585 0.341463415)
##                            5390) loan_intent=EDUCATION,VENTURE 20    4 0 (0.800000000 0.200000000) *
##                            5391) loan_intent=HOMEIMPROVEMENT,PERSONAL 21   10 0 (0.523809524 0.47619
##                             10782) person_income>=74500 13    4 0 (0.692307692 0.307692308) *
##                             10783) person_income< 74500 8    2 1 (0.250000000 0.750000000) *
##                     337) person_age>=43.5 12    5 0 (0.583333333 0.416666667) *
##                   169) person_income< 19600 12    5 1 (0.416666667 0.583333333) *
##                 85) loan_grade=F,G 17    7 1 (0.411764706 0.588235294) *
##               43) person_home_ownership=RENT 518    0 1 (0.000000000 1.000000000) *
##             11) loan_intent=DEBTCONSOLIDATION,MEDICAL 1116  105 1 (0.094086022 0.905913978)
##               22) person_home_ownership=OWN 73   14 0 (0.808219178 0.191780822)
##                 44) loan_grade=D 61    2 0 (0.967213115 0.032786885) *
##                 45) loan_grade=E,F,G 12    0 1 (0.000000000 1.000000000) *
##               23) person_home_ownership=MORTGAGE,OTHER,RENT 1043   46 1 (0.044103547 0.955896453)
##                 46) person_income< 48128.5 433   45 1 (0.103926097 0.896073903)
##                   92) loan_intent=MEDICAL 255   45 1 (0.176470588 0.823529412)
##                    184) person_emp_length>=2.5 169   45 1 (0.266272189 0.733727811)
##                      368) person_home_ownership=RENT 138   45 1 (0.326086957 0.673913043)
##                        736) person_age>=23.5 116   45 1 (0.387931034 0.612068966)
##                         1472) person_income>=34950 61   30 0 (0.508196721 0.491803279)
##                           2944) loan_amnt< 10625 54   24 0 (0.555555556 0.444444444)
##                             5888) person_income>=45531.5 8    1 0 (0.875000000 0.125000000) *
##                             5889) person_income< 45531.5 46   23 0 (0.500000000 0.500000000)
##                              11778) person_age< 26.5 18    6 0 (0.666666667 0.333333333) *
##                              11779) person_age>=26.5 28   11 1 (0.392857143 0.607142857)
##                                23558) person_emp_length< 4.5 9    3 0 (0.666666667 0.333333333) *
##                                23559) person_emp_length>=4.5 19    5 1 (0.263157895 0.736842105) *
```

```
##                         2945) loan_amnt>=10625 7    1 1 (0.142857143 0.857142857) *
##                       1473) person_income< 34950 55   14 1 (0.254545455 0.745454545)
##                         2946) cb_person_cred_hist_length< 7.5 36   13 1 (0.361111111 0.638888889)
##                           5892) loan_percent_income< 0.145 15    7 0 (0.533333333 0.466666667) *
##                           5893) loan_percent_income>=0.145 21    5 1 (0.238095238 0.761904762)
##                            11786) loan_amnt>=6200 7    3 0 (0.571428571 0.428571429) *
##                            11787) loan_amnt< 6200 14    1 1 (0.071428571 0.928571429) *
##                         2947) cb_person_cred_hist_length>=7.5 19    1 1 (0.052631579 0.947368421) *
##                       737) person_age< 23.5 22    0 1 (0.000000000 1.000000000) *
##                     369) person_home_ownership=MORTGAGE,OTHER 31    0 1 (0.000000000 1.000000000) *
##                   185) person_emp_length< 2.5 86    0 1 (0.000000000 1.000000000) *
##                  93) loan_intent=DEBTCONSOLIDATION 178    0 1 (0.000000000 1.000000000) *
##                47) person_income>=48128.5 610    1 1 (0.001639344 0.998360656) *
##           3) loan_percent_income>=0.305 3110  941 1 (0.302572347 0.697427653)
##             6) person_home_ownership=MORTGAGE,OWN 1216  275 0 (0.773848684 0.226151316)
##              12) person_income>=19900 1148  207 0 (0.819686411 0.180313589)
##                24) loan_grade=A,B,C 933   94 0 (0.899249732 0.100750268)
##                  48) person_home_ownership=OWN 252    0 0 (1.000000000 0.000000000) *
##                  49) person_home_ownership=MORTGAGE 681   94 0 (0.861967695 0.138032305)
##                    98) person_income>=31100 572   63 0 (0.889860140 0.110139860)
##                     196) loan_int_rate< 12.855 491   42 0 (0.914460285 0.085539715) *
##                     197) loan_int_rate>=12.855 81   21 0 (0.740740741 0.259259259)
##                       394) loan_intent=EDUCATION,HOMEIMPROVEMENT,MEDICAL,PERSONAL,VENTURE 71   14 0 (0
##                         788) person_age>=24.5 41    4 0 (0.902439024 0.097560976) *
##                         789) person_age< 24.5 30   10 0 (0.666666667 0.333333333)
##                          1578) loan_amnt< 23000 23    5 0 (0.782608696 0.217391304) *
##                          1579) loan_amnt>=23000 7    2 1 (0.285714286 0.714285714) *
##                       395) loan_intent=DEBTCONSOLIDATION 10    3 1 (0.300000000 0.700000000) *
##                    99) person_income< 31100 109   31 0 (0.715596330 0.284403670)
##                     198) loan_int_rate< 11.35 81   14 0 (0.827160494 0.172839506) *
##                     199) loan_int_rate>=11.35 28   11 1 (0.392857143 0.607142857)
##                       398) loan_intent=DEBTCONSOLIDATION,EDUCATION 10    4 0 (0.600000000 0.400000000)
##                       399) loan_intent=HOMEIMPROVEMENT,MEDICAL,PERSONAL,VENTURE 18    5 1 (0.277777778
##                25) loan_grade=D,E,F,G 215  102 1 (0.474418605 0.525581395)
##                  50) loan_intent=EDUCATION,HOMEIMPROVEMENT,PERSONAL,VENTURE 130   37 0 (0.715384615 0.1
##                   100) loan_grade=D,E,F 123   30 0 (0.756097561 0.243902439)
##                     200) person_home_ownership=OWN 23    1 0 (0.956521739 0.043478261) *
##                     201) person_home_ownership=MORTGAGE 100   29 0 (0.710000000 0.290000000)
##                       402) person_emp_length>=6.5 34    4 0 (0.882352941 0.117647059) *
##                       403) person_emp_length< 6.5 66   25 0 (0.621212121 0.378787879)
##                         806) loan_percent_income< 0.375 44   12 0 (0.727272727 0.272727273) *
##                         807) loan_percent_income>=0.375 22    9 1 (0.409090909 0.590909091)
##                          1614) loan_int_rate< 16.18 14    6 0 (0.571428571 0.428571429) *
##                          1615) loan_int_rate>=16.18 8    1 1 (0.125000000 0.875000000) *
##                   101) loan_grade=G 7    0 1 (0.000000000 1.000000000) *
##                  51) loan_intent=DEBTCONSOLIDATION,MEDICAL 85    9 1 (0.105882353 0.894117647)
##                   102) person_home_ownership=OWN 21    9 1 (0.428571429 0.571428571)
##                     204) loan_grade=D 9    0 0 (1.000000000 0.000000000) *
##                     205) loan_grade=E,F,G 12    0 1 (0.000000000 1.000000000) *
##                   103) person_home_ownership=MORTGAGE 64    0 1 (0.000000000 1.000000000) *
##              13) person_income< 19900 68    0 1 (0.000000000 1.000000000) *
##             7) person_home_ownership=OTHER,RENT 1894    0 1 (0.000000000 1.000000000) *
```

Se analizan los resultados obtenidos de forma numérica:

```r
rpart.rules(tree, style = "tall")
```

```
## as.factor(loan_status) is 0.00 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income >= 34850
##     loan_intent is VENTURE
##
## as.factor(loan_status) is 0.00 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 62952 to 64026
##     loan_intent is HOMEIMPROVEMENT or MEDICAL
##     person_emp_length >= 9
##     loan_int_rate >= 8.4
##
## as.factor(loan_status) is 0.00 when
##     loan_percent_income < 0.045
##     loan_grade is C
##     person_income is 79996 to 107500
##     person_emp_length < 9
##     loan_int_rate < 13.3
##
## as.factor(loan_status) is 0.00 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 34850 to 79996
##     person_home_ownership is OTHER or OWN
##
## as.factor(loan_status) is 0.00 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 29002 to 34850
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_home_ownership is OWN
##     loan_int_rate >= 8.4
##     loan_amnt < 7013
##
## as.factor(loan_status) is 0.00 when
##     loan_percent_income is 0.065 to 0.305
##     loan_grade is A or B or C
##     person_income is 29002 to 32448
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_home_ownership is MORTGAGE or RENT
##     loan_int_rate >= 8.4
##     loan_amnt is 6075 to 6850
##
## as.factor(loan_status) is 0.00 when
##     loan_percent_income is 0.065 to 0.305
##     loan_grade is A or B or C
##     person_income is 29002 to 32448
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_home_ownership is MORTGAGE
```

```
##     loan_int_rate >= 8.4
##     loan_amnt is 5525 to 6075
##
## as.factor(loan_status) is 0.00 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 29002 to 34850
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL or VENTURE
##     person_home_ownership is OWN
##     loan_amnt >= 7013
##
## as.factor(loan_status) is 0.00 when
##     loan_percent_income < 0.155
##     loan_grade is A or B or C
##     person_income < 19996
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or VENTURE
##     cb_person_cred_hist_length >= 10
##
## as.factor(loan_status) is 0.00 when
##     loan_percent_income < 0.155
##     loan_grade is A or B or C
##     person_income < 19996
##     loan_intent is HOMEIMPROVEMENT or PERSONAL
##     person_home_ownership is MORTGAGE or OWN
##
## as.factor(loan_status) is 0.00 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F
##     person_income is 49500 to 53817
##     loan_intent is EDUCATION or PERSONAL or VENTURE
##     person_emp_length is 3 to 7
##     loan_int_rate >= 16.0
##     person_age < 25
##
## as.factor(loan_status) is 0.00 when
##     loan_percent_income < 0.305
##     loan_grade is D or F
##     person_income is 19098 to 33998
##     loan_intent is EDUCATION or PERSONAL or VENTURE
##     person_emp_length >= 3
##     cb_person_cred_hist_length >= 10
##     cb_person_default_on_file is Y
##
## as.factor(loan_status) is 0.00 when
##     loan_percent_income >= 0.305
##     loan_grade is A or B or C
##     person_income >= 19900
##     person_home_ownership is OWN
##
## as.factor(loan_status) is 0.00 when
##     loan_percent_income >= 0.305
##     loan_grade is D
##     person_income >= 19900
##     loan_intent is DEBTCONSOLIDATION or MEDICAL
```

```
##     person_home_ownership is OWN
##
## as.factor(loan_status) is 0.01 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 59704 to 60054
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or HOMEIMPROVEMENT or MEDICAL or PERSONAL
##     loan_int_rate >= 8.4
##
## as.factor(loan_status) is 0.02 when
##     loan_percent_income < 0.205
##     loan_grade is C
##     person_income >= 79996
##     person_emp_length >= 9
##
## as.factor(loan_status) is 0.02 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income >= 59704
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or HOMEIMPROVEMENT or MEDICAL or PERSONAL
##     loan_int_rate < 8.4
##
## as.factor(loan_status) is 0.03 when
##     loan_percent_income < 0.255
##     loan_grade is D or E
##     person_income >= 59914
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_emp_length >= 12
##     loan_int_rate < 19.6
##
## as.factor(loan_status) is 0.03 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 64026 to 68843
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or HOMEIMPROVEMENT or MEDICAL or PERSONAL
##     loan_int_rate >= 8.4
##
## as.factor(loan_status) is 0.03 when
##     loan_percent_income < 0.305
##     loan_grade is D
##     loan_intent is DEBTCONSOLIDATION or MEDICAL
##     person_home_ownership is OWN
##
## as.factor(loan_status) is 0.04 when
##     loan_percent_income is 0.085 to 0.305
##     loan_grade is A or B
##     person_income is 34850 to 58629
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL
##     person_emp_length < 6
##     loan_int_rate < 11.4
##
## as.factor(loan_status) is 0.04 when
##     loan_percent_income < 0.305
##     loan_grade is D or E
```

```
##     person_income >= 19600
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_home_ownership is OTHER or OWN
##     person_emp_length < 3
##     person_age < 44
##
## as.factor(loan_status) is 0.04 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income >= 83602
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or HOMEIMPROVEMENT or MEDICAL or PERSONAL
##     loan_int_rate >= 8.4
##
## as.factor(loan_status) is 0.04 when
##     loan_percent_income < 0.255
##     loan_grade is D or E
##     person_income is 59914 to 107640
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_emp_length is 3 to 12
##     loan_int_rate < 19.6
##     loan_amnt >= 10413
##
## as.factor(loan_status) is 0.04 when
##     loan_percent_income >= 0.305
##     loan_grade is D or E or F
##     person_income >= 19900
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_home_ownership is OWN
##
## as.factor(loan_status) is 0.05 when
##     loan_percent_income < 0.085
##     loan_grade is A or B
##     person_income is 34850 to 58629
##     loan_intent is EDUCATION or MEDICAL
##     person_emp_length < 6
##     loan_int_rate < 11.4
##
## as.factor(loan_status) is 0.05 when
##     loan_percent_income is 0.045 to 0.205
##     loan_grade is C
##     person_income >= 79996
##     person_emp_length < 9
##
## as.factor(loan_status) is 0.06 when
##     loan_percent_income < 0.085
##     loan_grade is A or B
##     person_income is 44484 to 58629
##     loan_intent is DEBTCONSOLIDATION or PERSONAL
##     person_emp_length < 6
##     loan_int_rate < 11.4
##
## as.factor(loan_status) is 0.06 when
##     loan_percent_income < 0.305
##     loan_grade is C
```

```
##     person_income is 34850 to 77952
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length >= 4
##     loan_int_rate < 15.0
##     person_age < 24
##
## as.factor(loan_status) is 0.06 when
##     loan_percent_income is 0.065 to 0.305
##     loan_grade is A or B or C
##     person_income is 32448 to 33841
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_home_ownership is MORTGAGE or RENT
##     loan_int_rate >= 8.4
##     loan_amnt < 6850
##
## as.factor(loan_status) is 0.06 when
##     loan_percent_income is 0.085 to 0.305
##     loan_grade is A or B or C
##     person_income is 19996 to 33800
##     loan_intent is HOMEIMPROVEMENT
##     person_home_ownership is MORTGAGE or OWN
##     person_age >= 24
##
## as.factor(loan_status) is 0.06 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F
##     person_income is 54006 to 56002
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_emp_length >= 3
##
## as.factor(loan_status) is 0.06 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F
##     person_income is 34240 to 53817
##     loan_intent is EDUCATION or PERSONAL or VENTURE
##     person_emp_length >= 7
##     person_age < 25
##
## as.factor(loan_status) is 0.06 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 69430 to 82725
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or HOMEIMPROVEMENT or MEDICAL or PERSONAL
##     loan_int_rate >= 8.4
##
## as.factor(loan_status) is 0.07 when
##     loan_percent_income < 0.045
##     loan_grade is C
##     person_income >= 79996
##     person_emp_length < 9
##     loan_int_rate >= 13.3
##
## as.factor(loan_status) is 0.07 when
```

```
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 29002 to 34850
##     loan_intent is MEDICAL or VENTURE
##     loan_amnt < 7013
##
## as.factor(loan_status) is 0.07 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 34850 to 58629
##     loan_intent is HOMEIMPROVEMENT
##     person_home_ownership is MORTGAGE or OTHER or OWN
##     person_age >= 24
##
## as.factor(loan_status) is 0.07 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F
##     person_income is 34240 to 53817
##     loan_intent is EDUCATION or PERSONAL or VENTURE
##     person_emp_length >= 3
##     person_age >= 25
##
## as.factor(loan_status) is 0.07 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 34850 to 79996
##     loan_intent is EDUCATION or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     loan_int_rate >= 15.0
##     loan_amnt < 4675
##
## as.factor(loan_status) is 0.07 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 19996 to 28984
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL or VENTURE
##
## as.factor(loan_status) is 0.08 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 34850 to 58629
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL
##     person_emp_length >= 6
##     loan_int_rate < 11.4
##
## as.factor(loan_status) is 0.08 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 68843 to 69430
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or HOMEIMPROVEMENT or MEDICAL or PERSONAL
##     person_emp_length >= 7
##     loan_int_rate >= 8.4
##
## as.factor(loan_status) is 0.08 when
```

```
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 34850 to 77952
##     loan_intent is MEDICAL
##     person_home_ownership is RENT
##     person_emp_length >= 4
##     loan_int_rate < 15.0
##     person_age < 24
##
## as.factor(loan_status) is 0.08 when
##     loan_percent_income is 0.065 to 0.305
##     loan_grade is A or B or C
##     person_income is 34076 to 34850
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_home_ownership is MORTGAGE or RENT
##     loan_int_rate >= 8.4
##     loan_amnt < 7013
##
## as.factor(loan_status) is 0.08 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 33320 to 34850
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length >= 6
##     loan_amnt >= 7013
##
## as.factor(loan_status) is 0.08 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F
##     person_income is 49000 to 53817
##     loan_intent is HOMEIMPROVEMENT
##     person_emp_length >= 3
##
## as.factor(loan_status) is 0.08 when
##     loan_percent_income < 0.145
##     loan_grade is D or E
##     person_income >= 19600
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE
##     person_emp_length < 3
##     person_age < 44
##
## as.factor(loan_status) is 0.08 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 29002 to 34850
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     loan_int_rate < 8.4
##     loan_amnt < 7013
##
## as.factor(loan_status) is 0.08 when
##     loan_percent_income is 0.215 to 0.305
##     loan_grade is C
```

```
##     person_income >= 79996
##
## as.factor(loan_status) is 0.08 when
##     loan_percent_income < 0.255
##     loan_grade is D or E
##     person_income is 59914 to 107640
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_emp_length is 3 to 12
##     loan_int_rate < 19.6
##     loan_amnt < 9750
##
## as.factor(loan_status) is 0.09 when
##     loan_percent_income >= 0.305
##     loan_grade is A or B or C
##     person_income >= 31100
##     person_home_ownership is MORTGAGE
##     loan_int_rate < 12.9
##
## as.factor(loan_status) is 0.09 when
##     loan_percent_income < 0.155
##     loan_grade is A or B or C
##     person_income < 19996
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or VENTURE
##     person_age < 28
##     cb_person_cred_hist_length < 10
##
## as.factor(loan_status) is 0.09 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 34850 to 77952
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length < 4
##     loan_int_rate < 15.0
##
## as.factor(loan_status) is 0.09 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 46900 to 77952
##     loan_intent is EDUCATION
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length >= 4
##     loan_int_rate < 14.3
##     person_age >= 24
##     cb_person_cred_hist_length < 8
##
## as.factor(loan_status) is 0.09 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 44424 to 77952
##     loan_intent is DEBTCONSOLIDATION or MEDICAL or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length >= 4
##     loan_int_rate < 15.0
```

```
##     person_age >= 24
##     cb_person_cred_hist_length < 8
##
## as.factor(loan_status) is 0.09 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 49500 to 58629
##     loan_intent is HOMEIMPROVEMENT
##     person_home_ownership is RENT
##     person_age >= 24
##
## as.factor(loan_status) is 0.09 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 54132 to 79996
##     loan_intent is HOMEIMPROVEMENT
##     person_home_ownership is MORTGAGE or RENT
##     person_age >= 24
##     loan_amnt < 15950
##
## as.factor(loan_status) is 0.10 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 60054 to 64026
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or HOMEIMPROVEMENT or MEDICAL or PERSONAL
##     person_emp_length < 9
##     loan_int_rate >= 8.4
##
## as.factor(loan_status) is 0.10 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 82725 to 83602
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or HOMEIMPROVEMENT or MEDICAL or PERSONAL
##     loan_int_rate >= 8.4
##     person_age >= 24
##
## as.factor(loan_status) is 0.10 when
##     loan_percent_income >= 0.305
##     loan_grade is A or B or C
##     person_income >= 31100
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or MEDICAL or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE
##     loan_int_rate >= 12.9
##     person_age >= 25
##
## as.factor(loan_status) is 0.10 when
##     loan_percent_income < 0.045
##     loan_grade is C
##     person_income >= 139002
##     person_emp_length < 9
##     loan_int_rate < 13.3
##
## as.factor(loan_status) is 0.10 when
##     loan_percent_income < 0.305
```

```
##     loan_grade is C
##     person_income is 77952 to 79996
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     loan_int_rate is 11.9 to 13.8
##
## as.factor(loan_status) is 0.10 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 29002 to 33320
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     loan_int_rate < 10.9
##     loan_amnt >= 7013
##
## as.factor(loan_status) is 0.11 when
##     loan_percent_income is 0.065 to 0.305
##     loan_grade is A or B or C
##     person_income is 29002 to 32448
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length < 4
##     loan_int_rate >= 8.4
##     loan_amnt < 5525
##
## as.factor(loan_status) is 0.11 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 34850 to 43607
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length >= 4
##     loan_int_rate is 11.9 to 15.0
##     person_age >= 24
##
## as.factor(loan_status) is 0.11 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 34850 to 58629
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL
##     loan_int_rate >= 11.4
##
## as.factor(loan_status) is 0.11 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 34850 to 53685
##     loan_intent is HOMEIMPROVEMENT
##     person_home_ownership is MORTGAGE or RENT
##     person_age >= 24
##     cb_person_default_on_file is Y
##
## as.factor(loan_status) is 0.12 when
##     loan_percent_income is 0.065 to 0.305
##     loan_grade is A or B or C
```

```
##     person_income is 29002 to 32448
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length >= 4
##     loan_int_rate >= 8.4
##     person_age >= 24
##     loan_amnt < 5525
##
## as.factor(loan_status) is 0.12 when
##     loan_percent_income is 0.205 to 0.215
##     loan_grade is C
##     person_income >= 79996
##     person_age >= 26
##
## as.factor(loan_status) is 0.12 when
##     loan_percent_income >= 0.305
##     loan_grade is D or E or F
##     person_income >= 19900
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE
##     person_emp_length >= 7
##
## as.factor(loan_status) is 0.12 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F or G
##     person_income is 45532 to 48129
##     loan_intent is MEDICAL
##     person_home_ownership is RENT
##     person_emp_length >= 3
##     person_age >= 24
##     loan_amnt < 10625
##
## as.factor(loan_status) is 0.13 when
##     loan_percent_income < 0.305
##     loan_grade is D or F
##     person_income is 19098 to 33998
##     loan_intent is EDUCATION or PERSONAL or VENTURE
##     person_emp_length >= 3
##     cb_person_default_on_file is N
##
## as.factor(loan_status) is 0.13 when
##     loan_percent_income < 0.085
##     loan_grade is A or B
##     person_income is 34850 to 43750
##     loan_intent is DEBTCONSOLIDATION or PERSONAL
##     person_emp_length < 6
##     loan_int_rate < 11.4
##
## as.factor(loan_status) is 0.14 when
##     loan_percent_income < 0.255
##     loan_grade is D or E
##     person_income >= 107640
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_emp_length is 3 to 12
```

```
##     loan_int_rate < 19.6
##     cb_person_cred_hist_length < 15
##
## as.factor(loan_status) is 0.14 when
##     loan_percent_income < 0.255
##     loan_grade is F
##     person_income is 59914 to 135500
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_emp_length >= 3
##     loan_int_rate < 19.6
##
## as.factor(loan_status) is 0.14 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 34850 to 79996
##     loan_intent is DEBTCONSOLIDATION or MEDICAL or PERSONAL
##     person_home_ownership is MORTGAGE or RENT
##     loan_int_rate >= 15.0
##     loan_amnt < 14300
##
## as.factor(loan_status) is 0.14 when
##     loan_percent_income < 0.065
##     loan_grade is A or B or C
##     person_income is 29002 to 34850
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_home_ownership is MORTGAGE or RENT
##     loan_int_rate >= 8.4
##     person_age < 24
##     loan_amnt < 7013
##
## as.factor(loan_status) is 0.15 when
##     loan_percent_income < 0.255
##     loan_grade is D or E
##     person_income is 69500 to 107640
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_emp_length is 3 to 12
##     loan_int_rate < 19.6
##     loan_amnt is 9750 to 10413
##
## as.factor(loan_status) is 0.15 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 60054 to 64026
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_emp_length >= 9
##     loan_int_rate >= 8.4
##
## as.factor(loan_status) is 0.15 when
##     loan_percent_income is 0.265 to 0.305
##     loan_grade is C
##     person_income is 54132 to 79996
##     loan_intent is HOMEIMPROVEMENT
##     person_home_ownership is MORTGAGE or RENT
##     person_age >= 24
```

```
##     loan_amnt >= 15950
##
## as.factor(loan_status) is 0.15 when
##     loan_percent_income < 0.085
##     loan_grade is A or B or C
##     person_income is 19996 to 33800
##     loan_intent is HOMEIMPROVEMENT
##     person_home_ownership is MORTGAGE or OWN
##     person_age >= 29
##
## as.factor(loan_status) is 0.16 when
##     loan_percent_income is 0.145 to 0.305
##     loan_grade is D or E
##     person_income >= 19600
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE
##     person_emp_length < 3
##     loan_int_rate >= 15.0
##     person_age < 44
##
## as.factor(loan_status) is 0.17 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 44800 to 71000
##     loan_intent is EDUCATION or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     loan_int_rate >= 15.0
##     loan_amnt is 4675 to 14300
##
## as.factor(loan_status) is 0.17 when
##     loan_percent_income is 0.065 to 0.305
##     loan_grade is A or B or C
##     person_income is 29002 to 32448
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length >= 4
##     loan_int_rate >= 8.4
##     person_age < 24
##     loan_amnt < 4675
##
## as.factor(loan_status) is 0.17 when
##     loan_percent_income >= 0.305
##     loan_grade is A or B or C
##     person_income is 19900 to 31100
##     person_home_ownership is MORTGAGE
##     loan_int_rate < 11.4
##
## as.factor(loan_status) is 0.18 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F
##     person_income is 34240 to 53817
##     loan_intent is EDUCATION or PERSONAL or VENTURE
##     person_emp_length is 3 to 7
##     loan_int_rate < 16.0
```

```
##     person_age < 25
##
## as.factor(loan_status) is 0.18 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 44424 to 77952
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length >= 4
##     loan_int_rate < 15.0
##     person_age >= 24
##     cb_person_cred_hist_length >= 8
##
## as.factor(loan_status) is 0.18 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 59000 to 77952
##     loan_intent is MEDICAL
##     person_home_ownership is MORTGAGE
##     person_emp_length >= 4
##     loan_int_rate < 15.0
##     person_age < 24
##
## as.factor(loan_status) is 0.19 when
##     loan_percent_income is 0.255 to 0.305
##     loan_grade is D or E or F
##     person_income >= 59914
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_emp_length >= 3
##     loan_int_rate is 13.7 to 19.6
##
## as.factor(loan_status) is 0.19 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F
##     person_income is 56002 to 59914
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or VENTURE
##     person_emp_length >= 3
##     person_age >= 24
##
## as.factor(loan_status) is 0.20 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 58653 to 59704
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or HOMEIMPROVEMENT or MEDICAL or PERSONAL
##
## as.factor(loan_status) is 0.20 when
##     loan_percent_income < 0.065
##     loan_grade is A or B or C
##     person_income is 32820 to 34850
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_home_ownership is MORTGAGE or RENT
##     loan_int_rate >= 8.4
##     person_age >= 24
##     loan_amnt < 7013
```

```
##
## as.factor(loan_status) is 0.20 when
##     loan_percent_income < 0.155
##     loan_grade is A or B or C
##     person_income < 15828
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or VENTURE
##     person_age >= 28
##     cb_person_cred_hist_length < 10
##
## as.factor(loan_status) is 0.20 when
##     loan_percent_income is 0.145 to 0.305
##     loan_grade is D or E
##     person_income >= 19600
##     loan_intent is EDUCATION or VENTURE
##     person_home_ownership is MORTGAGE
##     person_emp_length < 3
##     loan_int_rate < 15.0
##     person_age < 44
##
## as.factor(loan_status) is 0.22 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 29002 to 33320
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length >= 2
##     loan_int_rate >= 10.9
##     loan_amnt >= 7013
##
## as.factor(loan_status) is 0.22 when
##     loan_percent_income >= 0.305
##     loan_grade is A or B or C
##     person_income >= 31100
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or MEDICAL or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE
##     loan_int_rate >= 12.9
##     person_age < 25
##     loan_amnt < 23000
##
## as.factor(loan_status) is 0.22 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 33320 to 34850
##     loan_intent is EDUCATION or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length < 6
##     loan_int_rate < 10.0
##     loan_amnt >= 7013
##     cb_person_cred_hist_length >= 4
##
## as.factor(loan_status) is 0.23 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 33320 to 34850
```

```
##     loan_intent is DEBTCONSOLIDATION or MEDICAL
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length < 6
##     person_age < 28
##     loan_amnt >= 7013
##
## as.factor(loan_status) is 0.25 when
##     loan_percent_income is 0.065 to 0.305
##     loan_grade is A or B or C
##     person_income is 29002 to 30450
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_home_ownership is RENT
##     loan_int_rate >= 8.4
##     loan_amnt is 5525 to 6075
##
## as.factor(loan_status) is 0.26 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 29002 to 33320
##     loan_intent is DEBTCONSOLIDATION or MEDICAL or PERSONAL
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length < 2
##     loan_int_rate >= 10.9
##     loan_amnt >= 7013
##
## as.factor(loan_status) is 0.27 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 28984 to 29002
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL or VENTURE
##     person_emp_length >= 3
##
## as.factor(loan_status) is 0.27 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 68843 to 69430
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or HOMEIMPROVEMENT or MEDICAL or PERSONAL
##     person_emp_length < 7
##     loan_int_rate >= 8.4
##     cb_person_cred_hist_length < 4
##
## as.factor(loan_status) is 0.27 when
##     loan_percent_income is 0.305 to 0.375
##     loan_grade is D or E or F
##     person_income >= 19900
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE
##     person_emp_length < 7
##
## as.factor(loan_status) is 0.29 when
##     loan_percent_income < 0.155
##     loan_grade is A or B or C
##     person_income < 19996
##     loan_intent is HOMEIMPROVEMENT or PERSONAL
```

```
##     person_home_ownership is RENT
##     loan_amnt >= 1650
##
## as.factor(loan_status) is 0.29 when
##     loan_percent_income < 0.235
##     loan_grade is D or F
##     person_income is 19098 to 33998
##     loan_intent is EDUCATION or PERSONAL or VENTURE
##     person_emp_length >= 3
##     cb_person_cred_hist_length < 10
##     cb_person_default_on_file is Y
##
## as.factor(loan_status) is 0.30 when
##     loan_percent_income < 0.255
##     loan_grade is D or E
##     person_income is 59914 to 63500
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_emp_length is 3 to 12
##     loan_int_rate < 19.6
##     loan_amnt is 9750 to 10413
##
## as.factor(loan_status) is 0.30 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 34850 to 53685
##     loan_intent is HOMEIMPROVEMENT
##     person_home_ownership is MORTGAGE
##     person_age >= 24
##     cb_person_default_on_file is N
##
## as.factor(loan_status) is 0.31 when
##     loan_percent_income is 0.065 to 0.305
##     loan_grade is A or B or C
##     person_income is 33841 to 34076
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length < 3
##     loan_int_rate >= 8.4
##     loan_amnt < 7013
##
## as.factor(loan_status) is 0.31 when
##     loan_percent_income is 0.145 to 0.305
##     loan_grade is D or E
##     person_income >= 74500
##     loan_intent is HOMEIMPROVEMENT or PERSONAL
##     person_home_ownership is MORTGAGE
##     person_emp_length < 3
##     loan_int_rate < 15.0
##     person_age < 44
##
## as.factor(loan_status) is 0.33 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F or G
##     person_income is 34950 to 45532
```

```
##     loan_intent is MEDICAL
##     person_home_ownership is RENT
##     person_emp_length >= 3
##     person_age is 24 to 27
##     loan_amnt < 10625
##
## as.factor(loan_status) is 0.33 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F or G
##     person_income is 34950 to 45532
##     loan_intent is MEDICAL
##     person_home_ownership is RENT
##     person_emp_length is 3 to 5
##     person_age >= 27
##     loan_amnt < 10625
##
## as.factor(loan_status) is 0.36 when
##     loan_percent_income < 0.065
##     loan_grade is A or B or C
##     person_income is 29002 to 31500
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_home_ownership is MORTGAGE or RENT
##     loan_int_rate >= 8.4
##     person_age >= 24
##     loan_amnt < 7013
##
## as.factor(loan_status) is 0.38 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F
##     person_income is 34240 to 43000
##     loan_intent is EDUCATION or PERSONAL or VENTURE
##     person_emp_length is 3 to 7
##     loan_int_rate >= 16.0
##     person_age < 25
##
## as.factor(loan_status) is 0.40 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F
##     person_income is 34240 to 49000
##     loan_intent is HOMEIMPROVEMENT
##     person_emp_length >= 3
##     loan_amnt >= 5500
##
## as.factor(loan_status) is 0.40 when
##     loan_percent_income >= 0.305
##     loan_grade is A or B or C
##     person_income is 19900 to 31100
##     loan_intent is DEBTCONSOLIDATION or EDUCATION
##     person_home_ownership is MORTGAGE
##     loan_int_rate >= 11.4
##
## as.factor(loan_status) is 0.42 when
##     loan_percent_income < 0.305
##     loan_grade is D or E
```

```
##     person_income >= 19600
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or OTHER or OWN
##     person_emp_length < 3
##     person_age >= 44
##
## as.factor(loan_status) is 0.43 when
##     loan_percent_income is 0.145 to 0.305
##     loan_grade is D or E or F or G
##     person_income < 34950
##     loan_intent is MEDICAL
##     person_home_ownership is RENT
##     person_emp_length >= 3
##     person_age >= 24
##     loan_amnt >= 6200
##     cb_person_cred_hist_length < 8
##
## as.factor(loan_status) is 0.43 when
##     loan_percent_income >= 0.375
##     loan_grade is D or E or F
##     person_income >= 19900
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE
##     person_emp_length < 7
##     loan_int_rate < 16.2
##
## as.factor(loan_status) is 0.47 when
##     loan_percent_income < 0.145
##     loan_grade is D or E or F or G
##     person_income < 34950
##     loan_intent is MEDICAL
##     person_home_ownership is RENT
##     person_emp_length >= 3
##     person_age >= 24
##     cb_person_cred_hist_length < 8
##
## as.factor(loan_status) is 0.50 when
##     loan_percent_income is 0.065 to 0.305
##     loan_grade is A or B or C
##     person_income is 29002 to 33841
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_home_ownership is MORTGAGE or RENT
##     loan_int_rate >= 8.4
##     loan_amnt is 6850 to 7013
##
## as.factor(loan_status) is 0.50 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F
##     person_income >= 59914
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_emp_length >= 3
##     loan_int_rate >= 19.6
##
## as.factor(loan_status) is 0.53 when
```

```
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 34850 to 43607
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length >= 4
##     loan_int_rate < 11.9
##     person_age >= 24
##
## as.factor(loan_status) is 0.54 when
##     loan_percent_income < 0.305
##     loan_grade is E
##     person_income is 19098 to 33998
##     loan_intent is EDUCATION or PERSONAL or VENTURE
##     person_emp_length >= 3
##
## as.factor(loan_status) is 0.57 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 44424 to 46900
##     loan_intent is EDUCATION
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length >= 4
##     loan_int_rate < 14.3
##     person_age >= 24
##     cb_person_cred_hist_length < 8
##
## as.factor(loan_status) is 0.57 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 77952 to 79996
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     loan_int_rate < 11.9
##
## as.factor(loan_status) is 0.57 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 34850 to 53685
##     loan_intent is HOMEIMPROVEMENT
##     person_home_ownership is RENT
##     person_age >= 24
##     cb_person_default_on_file is N
##
## as.factor(loan_status) is 0.57 when
##     loan_percent_income < 0.255
##     loan_grade is D or E
##     person_income >= 107640
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_emp_length is 3 to 12
##     loan_int_rate < 19.6
##     cb_person_cred_hist_length >= 15
##
## as.factor(loan_status) is 0.58 when
```

```
##     loan_percent_income < 0.155
##     loan_grade is A or B or C
##     person_income is 15828 to 19996
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or VENTURE
##     person_age >= 28
##     cb_person_cred_hist_length < 10
##
## as.factor(loan_status) is 0.58 when
##     loan_percent_income < 0.305
##     loan_grade is D or E
##     person_income < 19600
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or OTHER or OWN
##     person_emp_length < 3
##
## as.factor(loan_status) is 0.59 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 68843 to 69430
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or HOMEIMPROVEMENT or MEDICAL or PERSONAL
##     person_emp_length < 7
##     loan_int_rate >= 8.4
##     cb_person_cred_hist_length >= 4
##
## as.factor(loan_status) is 0.59 when
##     loan_percent_income < 0.305
##     loan_grade is F or G
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or OTHER or OWN
##     person_emp_length < 3
##
## as.factor(loan_status) is 0.60 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F
##     person_income is 56002 to 59914
##     loan_intent is PERSONAL
##     person_emp_length >= 3
##     person_age >= 24
##
## as.factor(loan_status) is 0.61 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 28984 to 29002
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL or VENTURE
##     person_emp_length < 3
##
## as.factor(loan_status) is 0.62 when
##     loan_percent_income is 0.065 to 0.305
##     loan_grade is A or B or C
##     person_income is 30450 to 32448
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_home_ownership is RENT
##     loan_int_rate >= 8.4
##     loan_amnt is 5525 to 6075
```

```
##
## as.factor(loan_status) is 0.62 when
##     loan_percent_income is 0.205 to 0.215
##     loan_grade is C
##     person_income >= 79996
##     person_age < 26
##
## as.factor(loan_status) is 0.62 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 71000 to 79996
##     loan_intent is EDUCATION or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     loan_int_rate >= 15.0
##     loan_amnt is 4675 to 14300
##
## as.factor(loan_status) is 0.62 when
##     loan_percent_income < 0.265
##     loan_grade is C
##     person_income is 54132 to 79996
##     loan_intent is HOMEIMPROVEMENT
##     person_home_ownership is MORTGAGE or RENT
##     person_age >= 24
##     loan_amnt >= 15950
##
## as.factor(loan_status) is 0.62 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 33320 to 34850
##     loan_intent is DEBTCONSOLIDATION or MEDICAL
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length < 6
##     person_age >= 28
##     loan_amnt >= 7013
##
## as.factor(loan_status) is 0.62 when
##     loan_percent_income is 0.235 to 0.305
##     loan_grade is D or F
##     person_income is 19098 to 33998
##     loan_intent is EDUCATION or PERSONAL or VENTURE
##     person_emp_length >= 3
##     cb_person_cred_hist_length < 10
##     cb_person_default_on_file is Y
##
## as.factor(loan_status) is 0.63 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 43607 to 44424
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length >= 4
##     loan_int_rate < 15.0
##     person_age >= 24
##
```

```
## as.factor(loan_status) is 0.64 when
##     loan_percent_income < 0.255
##     loan_grade is D or E
##     person_income is 63500 to 69500
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_emp_length is 3 to 12
##     loan_int_rate < 19.6
##     loan_amnt is 9750 to 10413
##
## as.factor(loan_status) is 0.64 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 60054 to 62952
##     loan_intent is HOMEIMPROVEMENT or MEDICAL
##     person_emp_length >= 9
##     loan_int_rate >= 8.4
##
## as.factor(loan_status) is 0.67 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 82725 to 83602
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or HOMEIMPROVEMENT or MEDICAL or PERSONAL
##     loan_int_rate >= 8.4
##     person_age < 24
##
## as.factor(loan_status) is 0.67 when
##     loan_percent_income < 0.045
##     loan_grade is C
##     person_income is 107500 to 139002
##     person_emp_length < 9
##     loan_int_rate < 13.3
##
## as.factor(loan_status) is 0.67 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 77952 to 79996
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     loan_int_rate is 13.8 to 15.0
##
## as.factor(loan_status) is 0.67 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 34850 to 79996
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or MEDICAL or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     loan_int_rate >= 15.0
##     loan_amnt >= 14300
##
## as.factor(loan_status) is 0.67 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 33320 to 34850
##     loan_intent is EDUCATION or PERSONAL or VENTURE
```

```
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length < 6
##     loan_int_rate >= 10.0
##     loan_amnt >= 7013
##     cb_person_cred_hist_length >= 4
##
## as.factor(loan_status) is 0.67 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F
##     person_income is 33998 to 34240
##     loan_intent is EDUCATION or PERSONAL or VENTURE
##     person_emp_length >= 3
##
## as.factor(loan_status) is 0.69 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F
##     person_income is 56002 to 59914
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_emp_length >= 3
##     person_age < 24
##
## as.factor(loan_status) is 0.70 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 53685 to 54132
##     loan_intent is HOMEIMPROVEMENT
##     person_home_ownership is MORTGAGE or RENT
##     person_age >= 24
##
## as.factor(loan_status) is 0.70 when
##     loan_percent_income >= 0.305
##     loan_grade is A or B or C
##     person_income >= 31100
##     loan_intent is DEBTCONSOLIDATION
##     person_home_ownership is MORTGAGE
##     loan_int_rate >= 12.9
##
## as.factor(loan_status) is 0.71 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 44424 to 77952
##     loan_intent is EDUCATION
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length >= 4
##     loan_int_rate is 14.3 to 15.0
##     person_age >= 24
##     cb_person_cred_hist_length < 8
##
## as.factor(loan_status) is 0.71 when
##     loan_percent_income < 0.085
##     loan_grade is A or B or C
##     person_income is 19996 to 33800
##     loan_intent is HOMEIMPROVEMENT
##     person_home_ownership is MORTGAGE or OWN
```

```
##     person_age is 24 to 29
##
## as.factor(loan_status) is 0.71 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 33800 to 34850
##     loan_intent is HOMEIMPROVEMENT
##     person_home_ownership is MORTGAGE or OWN
##     person_age >= 24
##
## as.factor(loan_status) is 0.71 when
##     loan_percent_income < 0.255
##     loan_grade is F
##     person_income >= 135500
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_emp_length >= 3
##     loan_int_rate < 19.6
##
## as.factor(loan_status) is 0.71 when
##     loan_percent_income is 0.255 to 0.305
##     loan_grade is D or E or F
##     person_income >= 59914
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_emp_length >= 3
##     loan_int_rate < 13.7
##
## as.factor(loan_status) is 0.71 when
##     loan_percent_income >= 0.305
##     loan_grade is A or B or C
##     person_income >= 31100
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or MEDICAL or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE
##     loan_int_rate >= 12.9
##     person_age < 25
##     loan_amnt >= 23000
##
## as.factor(loan_status) is 0.72 when
##     loan_percent_income >= 0.305
##     loan_grade is A or B or C
##     person_income is 19900 to 31100
##     loan_intent is HOMEIMPROVEMENT or MEDICAL or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE
##     loan_int_rate >= 11.4
##
## as.factor(loan_status) is 0.74 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F or G
##     person_income is 34950 to 45532
##     loan_intent is MEDICAL
##     person_home_ownership is RENT
##     person_emp_length >= 5
##     person_age >= 27
##     loan_amnt < 10625
##
```

```
## as.factor(loan_status) is 0.75 when
##     loan_percent_income is 0.065 to 0.305
##     loan_grade is A or B or C
##     person_income is 33841 to 34076
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length >= 3
##     loan_int_rate >= 8.4
##     loan_amnt < 7013
##
## as.factor(loan_status) is 0.75 when
##     loan_percent_income < 0.155
##     loan_grade is A or B or C
##     person_income < 19996
##     loan_intent is HOMEIMPROVEMENT or PERSONAL
##     person_home_ownership is RENT
##     loan_amnt < 1650
##
## as.factor(loan_status) is 0.75 when
##     loan_percent_income is 0.145 to 0.305
##     loan_grade is D or E
##     person_income is 19600 to 74500
##     loan_intent is HOMEIMPROVEMENT or PERSONAL
##     person_home_ownership is MORTGAGE
##     person_emp_length < 3
##     loan_int_rate < 15.0
##     person_age < 44
##
## as.factor(loan_status) is 0.77 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F
##     person_income is 53817 to 54006
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_emp_length >= 3
##
## as.factor(loan_status) is 0.78 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 34850 to 59000
##     loan_intent is MEDICAL
##     person_home_ownership is MORTGAGE
##     person_emp_length >= 4
##     loan_int_rate < 15.0
##     person_age < 24
##
## as.factor(loan_status) is 0.78 when
##     loan_percent_income < 0.065
##     loan_grade is A or B or C
##     person_income is 31500 to 32820
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_home_ownership is MORTGAGE or RENT
##     loan_int_rate >= 8.4
##     person_age >= 24
##     loan_amnt < 7013
```

```
##
## as.factor(loan_status) is 0.78 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 29002 to 33320
##     loan_intent is EDUCATION or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length < 2
##     loan_int_rate >= 10.9
##     loan_amnt >= 7013
##
## as.factor(loan_status) is 0.80 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 33320 to 34850
##     loan_intent is EDUCATION or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length < 6
##     loan_amnt >= 7013
##     cb_person_cred_hist_length < 4
##
## as.factor(loan_status) is 0.80 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F
##     person_income < 19098
##     loan_intent is EDUCATION or PERSONAL or VENTURE
##     person_emp_length >= 3
##
## as.factor(loan_status) is 0.82 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F
##     person_income is 43000 to 49500
##     loan_intent is EDUCATION or PERSONAL or VENTURE
##     person_emp_length is 3 to 7
##     loan_int_rate >= 16.0
##     person_age < 25
##
## as.factor(loan_status) is 0.86 when
##     loan_percent_income < 0.085
##     loan_grade is A or B
##     person_income is 43750 to 44484
##     loan_intent is DEBTCONSOLIDATION or PERSONAL
##     person_emp_length < 6
##     loan_int_rate < 11.4
##
## as.factor(loan_status) is 0.86 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F
##     person_income is 34240 to 49000
##     loan_intent is HOMEIMPROVEMENT
##     person_emp_length >= 3
##     loan_amnt < 5500
##
## as.factor(loan_status) is 0.86 when
```

```
##     loan_percent_income < 0.305
##     loan_grade is D or E or F or G
##     person_income is 34950 to 48129
##     loan_intent is MEDICAL
##     person_home_ownership is RENT
##     person_emp_length >= 3
##     person_age >= 24
##     loan_amnt >= 10625
##
## as.factor(loan_status) is 0.88 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 34850 to 44800
##     loan_intent is EDUCATION or VENTURE
##     person_home_ownership is MORTGAGE or RENT
##     loan_int_rate >= 15.0
##     loan_amnt is 4675 to 14300
##
## as.factor(loan_status) is 0.88 when
##     loan_percent_income is 0.065 to 0.305
##     loan_grade is A or B or C
##     person_income is 29002 to 32448
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or PERSONAL
##     person_home_ownership is MORTGAGE or RENT
##     person_emp_length >= 4
##     loan_int_rate >= 8.4
##     person_age < 24
##     loan_amnt is 4675 to 5525
##
## as.factor(loan_status) is 0.88 when
##     loan_percent_income >= 0.375
##     loan_grade is D or E or F
##     person_income >= 19900
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE
##     person_emp_length < 7
##     loan_int_rate >= 16.2
##
## as.factor(loan_status) is 0.93 when
##     loan_percent_income is 0.145 to 0.305
##     loan_grade is D or E or F or G
##     person_income < 34950
##     loan_intent is MEDICAL
##     person_home_ownership is RENT
##     person_emp_length >= 3
##     person_age >= 24
##     loan_amnt < 6200
##     cb_person_cred_hist_length < 8
##
## as.factor(loan_status) is 0.95 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F or G
##     person_income < 34950
##     loan_intent is MEDICAL
```

```
##     person_home_ownership is RENT
##     person_emp_length >= 3
##     person_age >= 24
##     cb_person_cred_hist_length >= 8
##
## as.factor(loan_status) is 0.95 when
##     loan_percent_income < 0.305
##     loan_grade is G
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_emp_length >= 3
##
## as.factor(loan_status) is 0.96 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F
##     person_income < 34240
##     loan_intent is HOMEIMPROVEMENT
##     person_emp_length >= 3
##
## as.factor(loan_status) is 1.00 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F or G
##     person_income >= 48129
##     loan_intent is DEBTCONSOLIDATION or MEDICAL
##     person_home_ownership is MORTGAGE or OTHER or RENT
##
## as.factor(loan_status) is 1.00 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 34850 to 49500
##     loan_intent is HOMEIMPROVEMENT
##     person_home_ownership is RENT
##     person_age >= 24
##
## as.factor(loan_status) is 1.00 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 34850 to 58629
##     loan_intent is HOMEIMPROVEMENT
##     person_age < 24
##
## as.factor(loan_status) is 1.00 when
##     loan_percent_income < 0.305
##     loan_grade is A or B
##     person_income is 58629 to 58653
##     loan_intent is DEBTCONSOLIDATION or EDUCATION or HOMEIMPROVEMENT or MEDICAL or PERSONAL
##
## as.factor(loan_status) is 1.00 when
##     loan_percent_income < 0.305
##     loan_grade is C
##     person_income is 34850 to 79996
##     loan_intent is HOMEIMPROVEMENT
##     person_home_ownership is MORTGAGE or RENT
##     person_age < 24
##
```

```
## as.factor(loan_status) is 1.00 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 19996 to 34850
##     loan_intent is HOMEIMPROVEMENT
##     person_home_ownership is MORTGAGE or OWN
##     person_age < 24
##
## as.factor(loan_status) is 1.00 when
##     loan_percent_income < 0.305
##     loan_grade is A or B or C
##     person_income is 19996 to 34850
##     loan_intent is HOMEIMPROVEMENT
##     person_home_ownership is RENT
##
## as.factor(loan_status) is 1.00 when
##     loan_percent_income is 0.155 to 0.305
##     loan_grade is A or B or C
##     person_income < 19996
##
## as.factor(loan_status) is 1.00 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F or G
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_home_ownership is RENT
##     person_emp_length < 3
##
## as.factor(loan_status) is 1.00 when
##     loan_percent_income < 0.305
##     loan_grade is E or F or G
##     loan_intent is DEBTCONSOLIDATION or MEDICAL
##     person_home_ownership is OWN
##
## as.factor(loan_status) is 1.00 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F or G
##     person_income < 48129
##     loan_intent is MEDICAL
##     person_home_ownership is RENT
##     person_emp_length >= 3
##     person_age < 24
##
## as.factor(loan_status) is 1.00 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F or G
##     person_income < 48129
##     loan_intent is MEDICAL
##     person_home_ownership is MORTGAGE or OTHER
##     person_emp_length >= 3
##
## as.factor(loan_status) is 1.00 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F or G
##     person_income < 48129
```

```
##     loan_intent is MEDICAL
##     person_home_ownership is MORTGAGE or OTHER or RENT
##     person_emp_length < 3
##
## as.factor(loan_status) is 1.00 when
##     loan_percent_income < 0.305
##     loan_grade is D or E or F or G
##     person_income < 48129
##     loan_intent is DEBTCONSOLIDATION
##     person_home_ownership is MORTGAGE or OTHER or RENT
##
## as.factor(loan_status) is 1.00 when
##     loan_percent_income >= 0.305
##     loan_grade is G
##     person_income >= 19900
##     loan_intent is EDUCATION or HOMEIMPROVEMENT or PERSONAL or VENTURE
##     person_home_ownership is MORTGAGE or OWN
##
## as.factor(loan_status) is 1.00 when
##     loan_percent_income >= 0.305
##     loan_grade is E or F or G
##     person_income >= 19900
##     loan_intent is DEBTCONSOLIDATION or MEDICAL
##     person_home_ownership is OWN
##
## as.factor(loan_status) is 1.00 when
##     loan_percent_income >= 0.305
##     loan_grade is D or E or F or G
##     person_income >= 19900
##     loan_intent is DEBTCONSOLIDATION or MEDICAL
##     person_home_ownership is MORTGAGE
##
## as.factor(loan_status) is 1.00 when
##     loan_percent_income >= 0.305
##     person_income < 19900
##     person_home_ownership is MORTGAGE or OWN
##
## as.factor(loan_status) is 1.00 when
##     loan_percent_income >= 0.305
##     person_home_ownership is OTHER or RENT
```

- **Modelo Decision Tree sin poda:**

```
obs_tree1 <- as.factor(fcr_train_tree$loan_status)
head(predict(tree, newdata = fcr_train_tree))
```

```
##           0          1
## 1 0.9090909 0.09090909
## 2 1.0000000 0.00000000
## 3 0.9189189 0.08108108
## 4 0.9177215 0.08227848
## 5 0.9248609 0.07513915
## 6 0.8864469 0.11355311
```

```r
pred_tree1 <- predict(tree, newdata = fcr_train_tree, type = "class")
table(obs_tree1, pred_tree1)
```

```
##          pred_tree1
## obs_tree1     0     1
##         0 20174   191
##         1  1292  4402
```

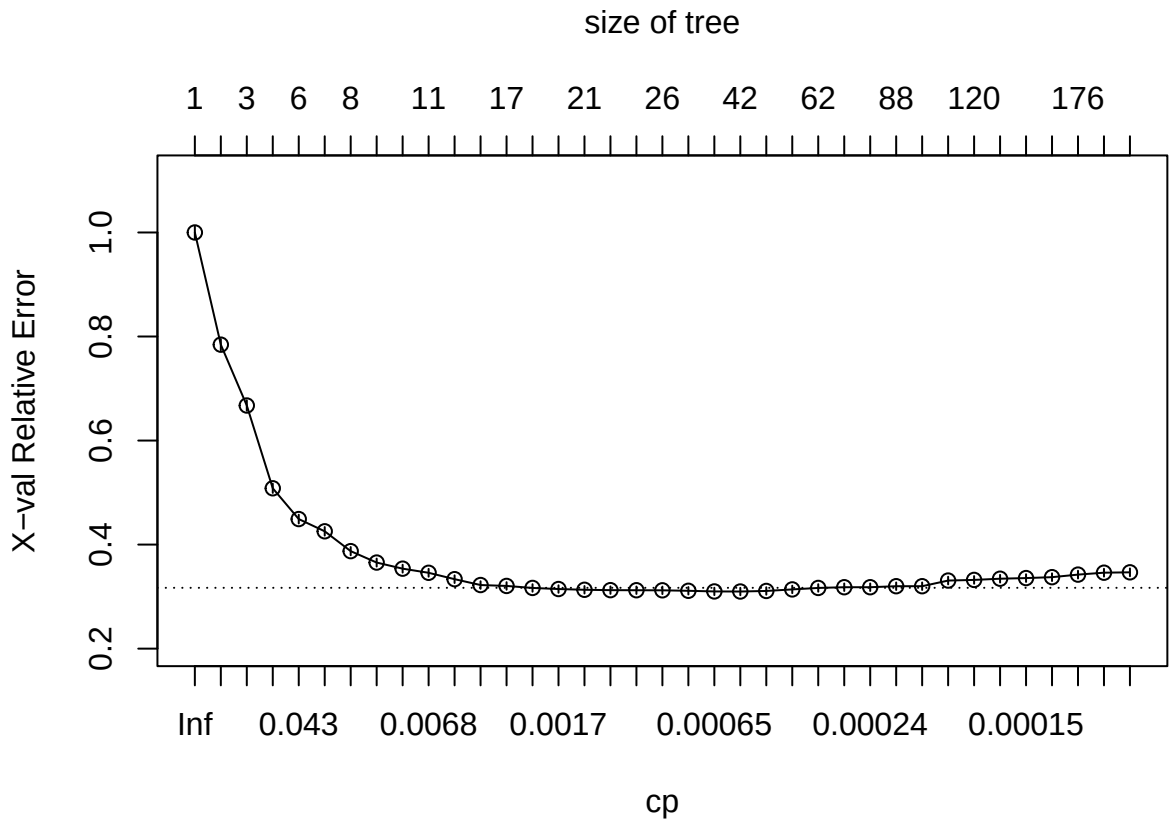
```r
caret::confusionMatrix(pred_tree1, obs_tree1)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 20174  1292
##          1   191  4402
##
##                Accuracy : 0.9431
##                  95% CI : (0.9402, 0.9459)
##     No Information Rate : 0.7815
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8209
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9906
##             Specificity : 0.7731
##          Pos Pred Value : 0.9398
##          Neg Pred Value : 0.9584
##              Prevalence : 0.7815
##          Detection Rate : 0.7742
##    Detection Prevalence : 0.8237
##       Balanced Accuracy : 0.8819
##
##        'Positive' Class : 0
##
```

Se obtiene un valor del 94.31% para la precisión del modelo, con el incoveniente de tener un modelo sin poda, demasiado complejo y que puede tender al sobreajuste.

Se realiza por ello la valoración para una posible poda del modelo que permita simplificarlo y hacerlo más explicativo sin perder capacidad predictora. Para ello vemos el CP o "Parámetro de complejidad" con el cual buscamos el árbol menos profundo que además minimice la tasa de error.

```r
plotcp(tree)
```




```
# CP - PARÁMETRO DE COMPLEJIDAD: Buscamos el árbol menos
# profundo que además minimiza la tasa de error
```

```
printcp(tree)
```

```
##
## Classification tree:
## rpart(formula = as.factor(loan_status) ~ ., data = fcr_train_tree,
##     cp = 1e-04)
##
## Variables actually used in tree construction:
##  [1] cb_person_cred_hist_length cb_person_default_on_file
##  [3] loan_amnt                  loan_grade
##  [5] loan_int_rate              loan_intent
##  [7] loan_percent_income        person_age
##  [9] person_emp_length          person_home_ownership
## [11] person_income
##
## Root node error: 5694/26059 = 0.2185
##
## n= 26059
##
##            CP nsplit rel error  xerror      xstd
## 1  0.21566561      0   1.00000 1.00000 0.0117153
## 2  0.11696523      1   0.78433 0.78433 0.0106836
```

```
## 3  0.07955743      2   0.66737 0.66737 0.0100057
## 4  0.05918511      4   0.50825 0.50825 0.0089078
## 5  0.03178785      5   0.44907 0.44907 0.0084338
## 6  0.03003161      6   0.41728 0.42554 0.0082332
## 7  0.02177731      7   0.38725 0.38743 0.0078918
## 8  0.01194240      8   0.36547 0.36547 0.0076850
## 9  0.00790306      9   0.35353 0.35371 0.0075709
## 10 0.00588339     10   0.34563 0.34580 0.0074928
## 11 0.00392226     12   0.33386 0.33351 0.0073691
## 12 0.00316122     15   0.32209 0.32227 0.0072535
## 13 0.00210748     16   0.31893 0.32051 0.0072352
## 14 0.00193186     17   0.31682 0.31665 0.0071947
## 15 0.00158061     19   0.31296 0.31454 0.0071725
## 16 0.00140499     20   0.31138 0.31314 0.0071576
## 17 0.00122936     21   0.30998 0.31243 0.0071502
## 18 0.00081958     22   0.30875 0.31226 0.0071483
## 19 0.00079031     25   0.30629 0.31208 0.0071464
## 20 0.00070249     27   0.30471 0.31120 0.0071371
## 21 0.00060670     28   0.30400 0.30998 0.0071240
## 22 0.00052687     41   0.29417 0.30980 0.0071222
## 23 0.00035125     45   0.29206 0.31085 0.0071334
## 24 0.00029271     58   0.28750 0.31384 0.0071650
## 25 0.00026344     61   0.28662 0.31665 0.0071947
## 26 0.00025089     73   0.28311 0.31805 0.0072094
## 27 0.00023416     80   0.28135 0.31805 0.0072094
## 28 0.00021953     87   0.27959 0.31999 0.0072297
## 29 0.00021075     98   0.27661 0.31999 0.0072297
## 30 0.00017562    103   0.27555 0.33087 0.0073422
## 31 0.00015806    119   0.27239 0.33210 0.0073548
## 32 0.00015053    138   0.26923 0.33439 0.0073780
## 33 0.00014635    148   0.26660 0.33562 0.0073905
## 34 0.00013172    154   0.26572 0.33737 0.0074083
## 35 0.00011708    175   0.26291 0.34229 0.0074578
## 36 0.00010036    190   0.26115 0.34598 0.0074946
## 37 0.00010000    197   0.26045 0.34668 0.0075015
```

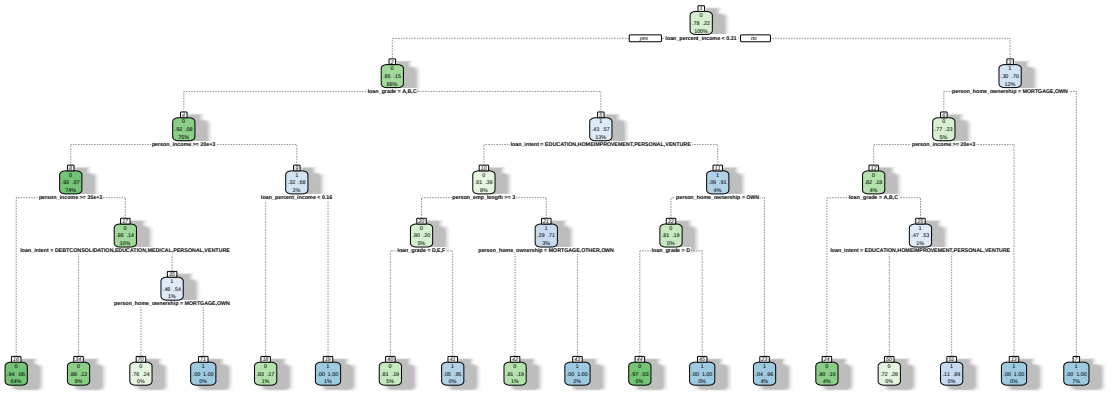
Finalmente se decide proceder a realizar la poda y crear un modelo alternativo más simplificado y con menor sobreajuste:

```
xerror <- tree$cptable[, "xerror"]
imin.xerror <- which.min(xerror)
upper.xerror <- xerror[imin.xerror] + tree$cptable[imin.xerror,
    "xstd"]
icp <- min(which(xerror <= upper.xerror))
cp <- tree$cptable[icp, "CP"]
cp
```

```
## [1] 0.001931858
```

```
tree_2 <- prune(tree, cp = cp)
# tree summary(tree) caret::varImp(tree) importance <-
# tree$variable.importance importance <-
# round(100*importance/sum(importance), 1)
```

```
# importance[importance >= 1]
rpart.plot(tree_2, nn = TRUE, extra = 104, box.palette = "GnBu",
    branch.lty = 3, shadow.col = "gray")  #, main='Classification tree winetaste'
```



- **Modelo Decision Tree con poda:**

```
obs_tree2 <- as.factor(fcr_train_tree$loan_status)
head(predict(tree_2, newdata = fcr_train_tree))
```

```
##           0          1
## 1 0.9416120 0.05838801
## 2 0.9416120 0.05838801
## 3 0.9416120 0.05838801
## 4 0.8078078 0.19219219
## 5 0.9416120 0.05838801
## 6 0.9416120 0.05838801
```

```
pred_tree2 <- predict(tree_2, newdata = fcr_train_tree, type = "class")
table(obs_tree2, pred_tree2)
```

```
##          pred_tree2
## obs_tree2     0     1
##         0 20309    56
##         1  1748  3946
```

```
caret::confusionMatrix(pred_tree2, obs_tree2)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 20309  1748
##          1    56  3946
##
##                Accuracy : 0.9308
##                  95% CI : (0.9276, 0.9338)
##     No Information Rate : 0.7815
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.773
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9973
##             Specificity : 0.6930
##          Pos Pred Value : 0.9208
##          Neg Pred Value : 0.9860
##              Prevalence : 0.7815
##          Detection Rate : 0.7793
##    Detection Prevalence : 0.8464
##       Balanced Accuracy : 0.8451
##
##        'Positive' Class : 0
##
```

Aplicando la poda a nuestro árbol se ontiene un modelo mas limpio, simple, explicativo y generalizable a otro conjunto de datos, evitando el posible sobreajuste del modelo y solo reduciendo su capacidad predictora a un valor de precisión del 93.08%. Se entiende por lo tanto que este modelo podado será el óptimo en este caso.

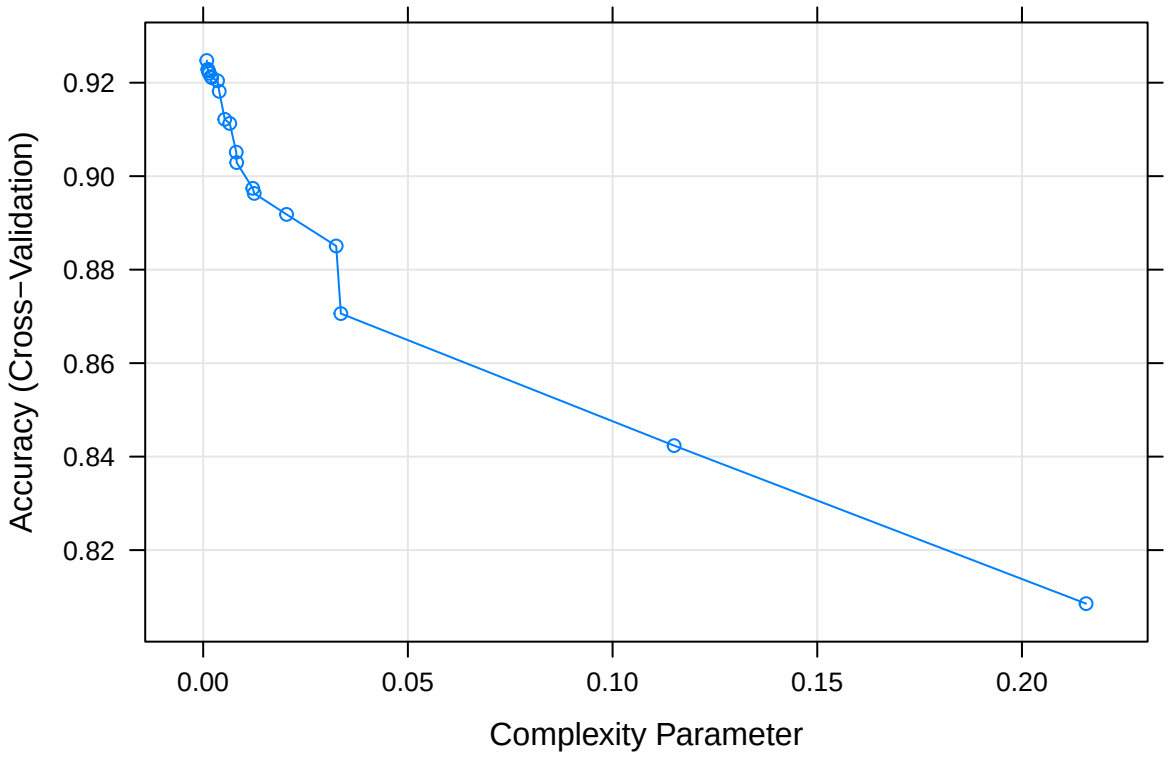
8.1. DECISION TREE - Cross Validation, Hiperparámetros y Evaluación del modelo

Se trata de aplicar Cross Validation sobre el modelo de árbol de decisión y realizar una selección de hiperparámetros (se busca tener un modelo robusto, generalizable y comparable con el resto para la posterior selección del mejor):

De cara a obtener el mejor modelo posible se realizará validación cruzada de 5 folds y se tratará de ajustar hiperparámetros (el "cp" óptimo para un modelo ya validado). Se utiliza además las variables que hemos vito como más representativas y explicativas de la variable respuesta "loan_status".

```
# Fit the model on the training set
set.seed(1234)
caret.tree <- train(as.factor(loan_status) ~ person_income +
    person_emp_length + loan_amnt + loan_int_rate + person_home_ownership +
    loan_intent + loan_grade + loan_percent_income, data = fcr_train_tree,
    method = "rpart", trControl = trainControl("cv", number = 5,
```

```
        search = "grid", returnResamp = "final"), tuneLength = 20)
# Plot model accuracy vs different values of cp (complexity
# parameter)
plot(caret.tree)
```




```
caret.tree
```

```
## CART
##
## 26059 samples
##     8 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 20847, 20847, 20848, 20847, 20847
## Resampling results across tuning parameters:
##
##   cp            Accuracy   Kappa
##   0.0008781173  0.9247479  0.7544905
##   0.0011415525  0.9228676  0.7478875
##   0.0012293642  0.9225606  0.7467760
##   0.0012586348  0.9225606  0.7467760
##   0.0014635289  0.9220234  0.7447789
##   0.0019318581  0.9213326  0.7420910
```

```
##   0.0021074816  0.9210640  0.7411880
##   0.0035124693  0.9204116  0.7394080
##   0.0039222573  0.9181475  0.7299874
##   0.0052687039  0.9121608  0.7072620
##   0.0064980681  0.9112782  0.7032548
##   0.0080786793  0.9051384  0.6749117
##   0.0081840534  0.9029127  0.6648955
##   0.0121180190  0.8973868  0.6497519
##   0.0124692659  0.8963123  0.6474650
##   0.0203723217  0.8918224  0.6348701
##   0.0324903407  0.8850681  0.6067465
##   0.0336318932  0.8706014  0.5279197
##   0.1150333685  0.8423573  0.4161644
##   0.2156656129  0.8085488  0.2372636
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.0008781173.
```

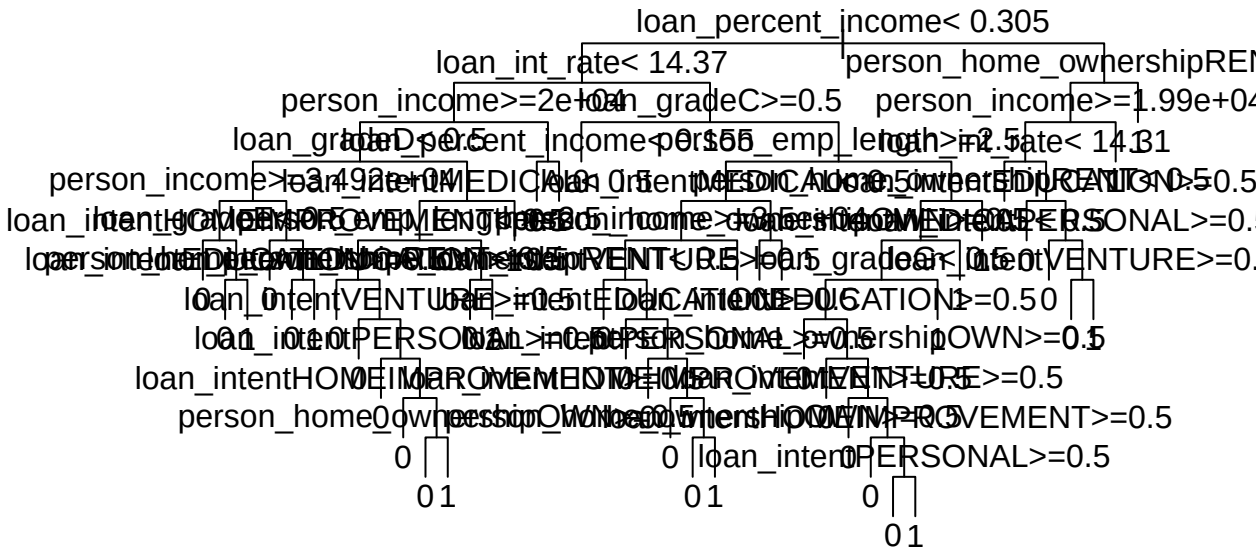
```
caret.tree$bestTune
```

```
##             cp
## 1 0.0008781173
```

Realizando la validación cruzada vemos que el CP óptimo para nuestro modelo de árbol de decisión se encuentra en 0.0008781173.

Visualizamos graficamente el árbol obtenido:

```
# Plot the final tree model
par(xpd = NA)  # Avoid clipping the text in some device
plot(caret.tree$finalModel, uniform = TRUE)
text(caret.tree$finalModel, digits = 10)
```




```
get_best_result = function(caret_fit) {
    best = which(rownames(caret_fit$results) == rownames(caret_fit$bestTune))
    best_result = caret_fit$results[best, ]
    rownames(best_result) = NULL
    best_result
}

get_best_result(caret.tree)
```

```
##             cp  Accuracy     Kappa  AccuracySD    KappaSD
## 1 0.0008781173 0.9247479 0.7544905 0.004037143 0.01365907
```

Se obtiene finalmente haciendo validación cruzada una precisión del 92/93%, con un modelo que ha sido comprobado como robusto y generalizable para funcionar previsiblemente en otro conjunto de datos diferente.

Evaluación del rendimiento predictivo del modelo Decision Tree presentado con las datos de train (metrica de evaluación utilizada de referencia: "Accuracy", "Recall", "Precision", "F1" y "ROC", y punto de corte utilizado: 0.5):

```
fcr_train_tree$y_pred_probs2 <- predict(caret.tree, newdata = fcr_train_tree,
    type = "prob")
fcr_train_tree$y_pred_probs2 <- ifelse(fcr_train_tree$y_pred_probs2$`1` >
    0.5, fcr_train_tree$y_pred_probs2$`1`, 1 - fcr_train_tree$y_pred_probs2$`0`)

fcr_train_tree$y_pred2 <- ifelse(fcr_train_tree$y_pred_probs2 >
```

```
    0.5, 1, 0)

# fcr_train_tree$y_pred_probs2 fcr_train_tree$y_pred2
```

Se reproduce la matriz de confusión y las métricas de evaluación sobre el modelo final de Decision Tree obtenido:

```
cm_train_tree <- confusionMatrix(as.factor(fcr_train_tree$y_pred2),
    as.factor(fcr_train_tree$loan_status), positive = "1")
cm_train_tree$table
```

```
##           Reference
## Prediction     0     1
##          0 20209  1746
##          1   156  3948
```

```
# result
accuracy_modelo_tree_tune <- cm_train_tree$overall["Accuracy"] %>%
    round(4)
accuracy_modelo_tree_tune
```

```
## Accuracy
##    0.927
```

```
# result
recall_modelo_tree_tune <- cm_train_tree$byClass["Recall"] %>%
    round(4)
recall_modelo_tree_tune
```

```
## Recall
## 0.6934
```

```
# result
precision_modelo_tree_tune <- cm_train_tree$byClass["Precision"] %>%
    round(4)
precision_modelo_tree_tune
```

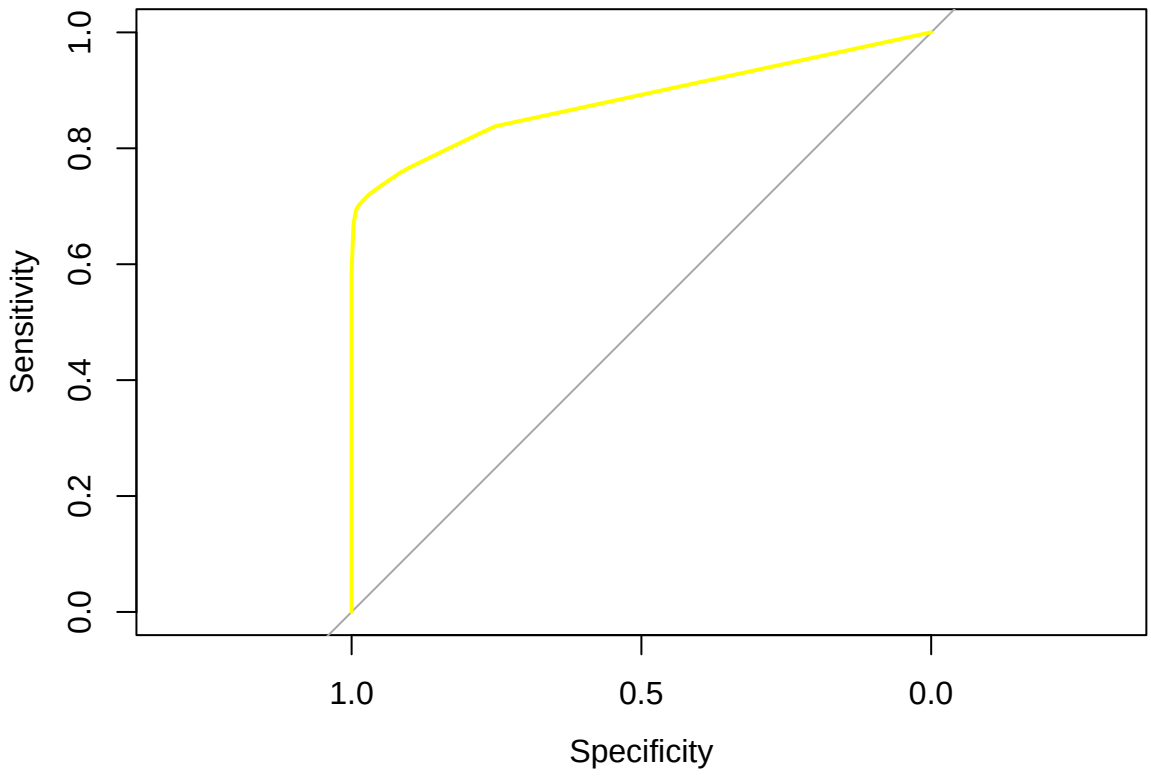
```
## Precision
##     0.962
```

```
# result
F1Score_modelo_tree_tune <- (2 * (precision_modelo_tree_tune *
    recall_modelo_tree_tune)/(precision_modelo_tree_tune + recall_modelo_tree_tune)) %>%
    round(4)
F1Score_modelo_tree_tune
```

```
## Precision
##    0.8059
```

Se reproduce la curva ROC sobre el modelo final de árbol de decisión obtenido:

```
roc_tree <- plot.roc(as.numeric(fcr_train_tree$loan_status),
    as.numeric(fcr_train_tree$y_pred_probs2), col = "yellow")
```




```
auc(roc_tree)
```

```
## Area under the curve: 0.8829
```

Se obtiene alrededor de un 88.29% de área bajo la curva.

9. RANDOM FOREST

Primero se crean unos datos de train específicos para ser usados en el desarrollo del modelo de random forest, y así mantener los originales sin modificar.

```
fcr_train_forest <- fcr_train
fcr_train_forest
```

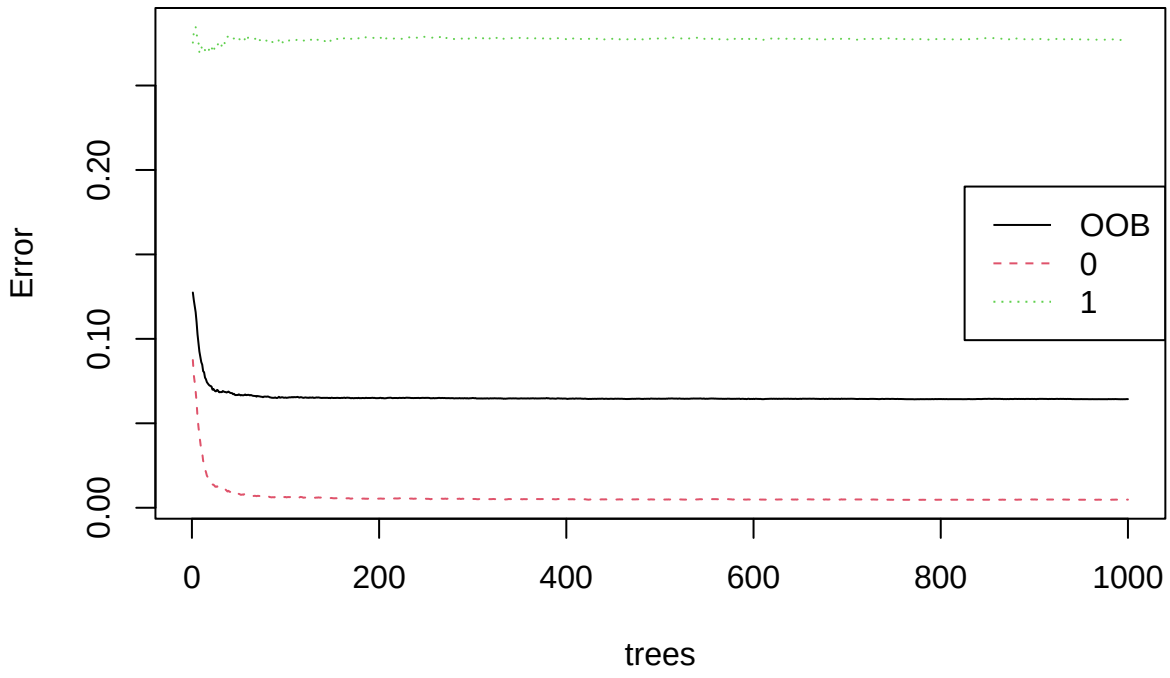
```
## # A tibble: 26,059 x 12
##    person_age person_i~1 perso~2 perso~3 loan_~4 loan_~5 loan_~6 loan_~7 loan_~8
##         <dbl>      <dbl> <fct>     <dbl> <fct>   <fct>     <dbl>   <dbl>   <dbl>
##  1         28      44000 RENT          2 MEDICAL C         10000   13.5        1
##  2         21      35000 OWN           5 VENTURE B          8000    9.91       0
```

```
##  3         25      96000 MORTGA~       6 HOMEIM~ C         21000   14.6        0
##  4         22      67000 OWN           5 EDUCAT~ D          7500   16.3        0
##  5         24      52800 RENT          8 PERSON~ A          9000    7.49       0
##  6         27      50004 RENT         12 DEBTCO~ B          3200   11.5        0
##  7         23      55488 RENT          4 MEDICAL D          5000   15.2        1
##  8         28      70000 RENT          2 DEBTCO~ B          6000   10.4        0
##  9         22      55000 MORTGA~       6 PERSON~ C         13000   13.8        0
## 10         26      43200 RENT          5 EDUCAT~ C          3200   14.4        0
## # ... with 26,049 more rows, 3 more variables: loan_percent_income <dbl>,
## #   cb_person_default_on_file <fct>, cb_person_cred_hist_length <dbl>, and
## #   abbreviated variable names 1: person_income, 2: person_home_ownership,
## #   3: person_emp_length, 4: loan_intent, 5: loan_grade, 6: loan_amnt,
## #   7: loan_int_rate, 8: loan_status
```

Se crea el modelo base de random forest realizando una simulación con 1000 árboles:

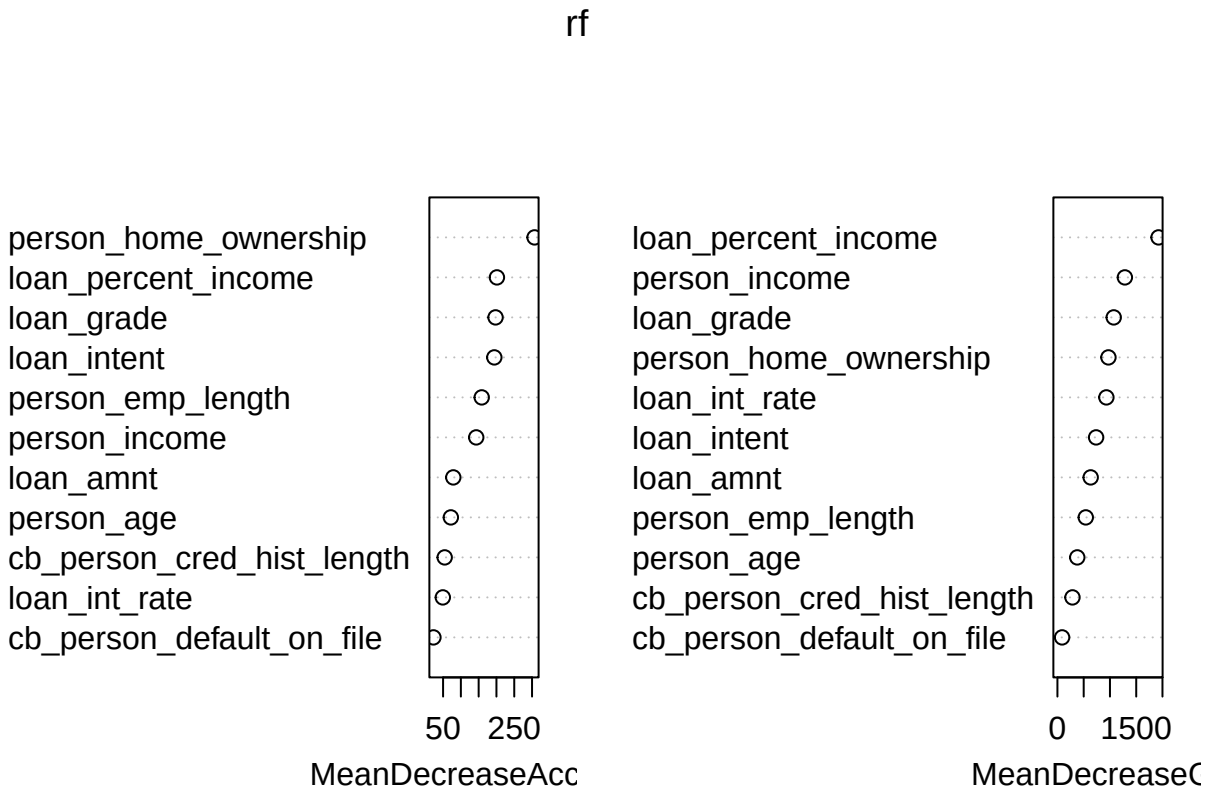
Se examina la convergencia del error en las muestras:

```
plot(rf, main = "")
legend("right", colnames(rf$err.rate), lty = 1:5, col = 1:6)
```



Se analiza la relevancia de las variables en el modelo:

```
varImpPlot(rf)
```



9.1. RANDOM FOREST - Cross Validation, Hiperparámetros y Evaluación del modelo

Se trata de aplicar Cross Validation sobre el modelo de random forest y realizar una selección de hiperparámetros (se busca tener un modelo robusto, generalizable y comparable con el resto para la posterior selección del mejor):

Vemos que el principal parámetro a configurar es el número de predictores al azar que toma el modelo.

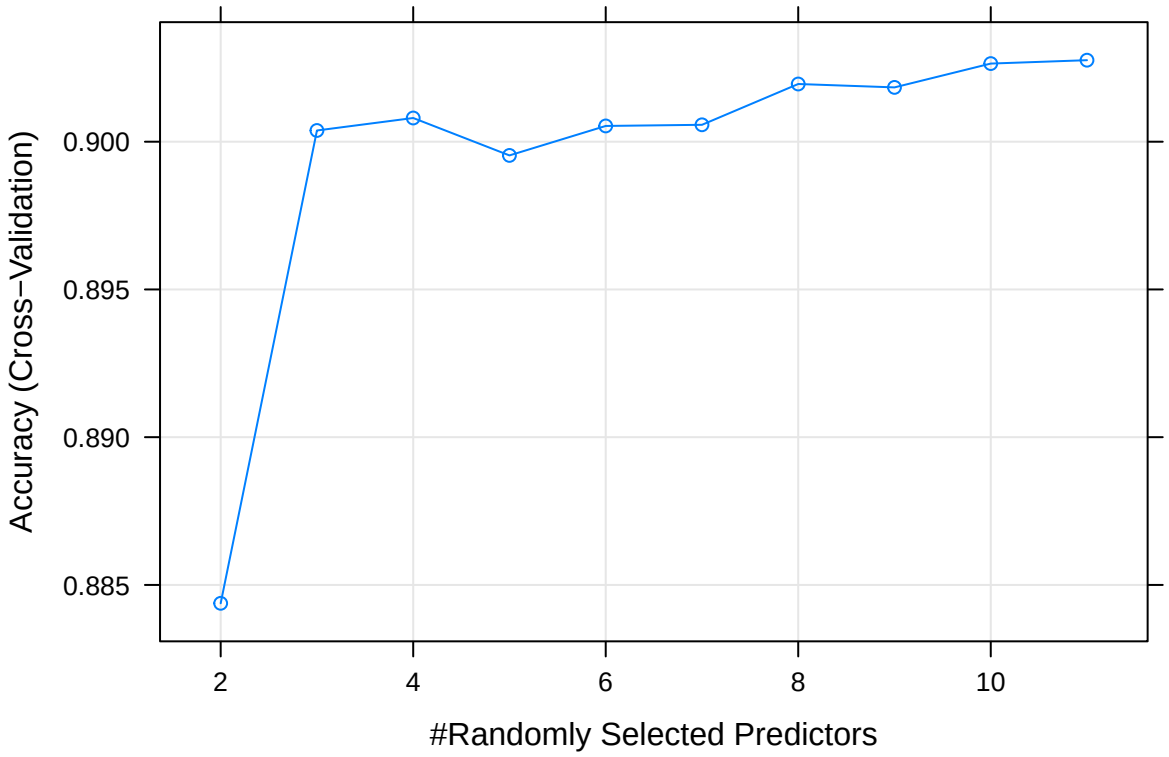
```
modelLookup("rf")
```

```
##   model parameter                          label forReg forClass probModel
## 1    rf      mtry #Randomly Selected Predictors   TRUE     TRUE      TRUE
```

Se crea un modelo aplicando la validación cruzada y ajustando hiperparámetros (mtry, número de árboles y el tamaño de los nodos para regular su profundidad) de tal forma que creemos un modelo robusto y generalizable. Se toma como base las variables de mayor relevancia que hemos observado:

```
# Fit the model on the training set
set.seed(12345)

caret.rf <- train(as.factor(loan_status) ~ loan_percent_income +
    person_income + loan_grade + person_home_ownership, data = fcr_train_forest,
    method = "rf", ntree = 20, importance = TRUE, metric = "Accuracy",
```

```
    trControl = trainControl("cv", number = 5, search = "grid",
        returnResamp = "final"), nodesize = 30, tuneLength = 10)

plot(caret.rf)
```




```
caret.rf
```

```
## Random Forest
##
## 26059 samples
##     4 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 20847, 20848, 20847, 20847, 20847
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.8843781  0.6196875
##    3    0.9003799  0.6870654
##    4    0.9008020  0.6890992
##    5    0.8995356  0.6844077
##    6    0.9005334  0.6882959
##    7    0.9005717  0.6884403
```

```
##    8    0.9019532  0.6920956
##    9    0.9018382  0.6899826
##   10    0.9026440  0.6922755
##   11    0.9027592  0.6920294
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 11.
```

```
caret.rf$bestTune
```

```
##    mtry
## 10   11
```

```
get_best_result = function(caret_fit) {
    best = which(rownames(caret_fit$results) == rownames(caret_fit$bestTune))
    best_result = caret_fit$results[best, ]
    rownames(best_result) = NULL
    best_result
}

get_best_result(caret.rf)
```

```
##   mtry  Accuracy     Kappa  AccuracySD     KappaSD
## 1   11 0.9027592 0.6920294 0.002483855 0.006137814
```

Evaluación del rendimiento predictivo del modelo Decision Tree presentado con las datos de train (metrica de evaluación utilizada de referencia: “Accuracy”, “Recall”, “Precision”, “F1” y “ROC”, y punto de corte utilizado: 0.5):

```
fcr_train_forest$y_pred_probs2 <- predict(caret.rf, newdata = fcr_train_forest,
    type = "prob")
fcr_train_forest$y_pred_probs2 <- ifelse(fcr_train_forest$y_pred_probs2$`1` >
    0.5, fcr_train_forest$y_pred_probs2$`1`, 1 - fcr_train_forest$y_pred_probs2$`0`)

fcr_train_forest$y_pred2 <- ifelse(fcr_train_forest$y_pred_probs2 >
    0.5, 1, 0)

# fcr_train_forest$y_pred_probs2 fcr_train_forest$y_pred2
# fcr_train_forest
```

Se reproduce la matriz de confusión y las métricas de evaluación sobre el modelo final de Decision Tree obtenido:

```
cm_train_forest <- confusionMatrix(as.factor(fcr_train_forest$y_pred2),
    as.factor(fcr_train_forest$loan_status), positive = "1")
cm_train_forest$table
```

```
##            Reference
## Prediction     0     1
##          0 19924  1669
##          1   441  4025
```

```r
# result
accuracy_modelo_forest_tune <- cm_train_forest$overall["Accuracy"] %>%
    round(4)
accuracy_modelo_forest_tune
```

```
## Accuracy
##    0.919
```

```r
# result
recall_modelo_forest_tune <- cm_train_forest$byClass["Recall"] %>%
    round(4)
recall_modelo_forest_tune
```

```
## Recall
## 0.7069
```

```r
# result
precision_modelo_forest_tune <- cm_train_forest$byClass["Precision"] %>%
    round(4)
precision_modelo_forest_tune
```

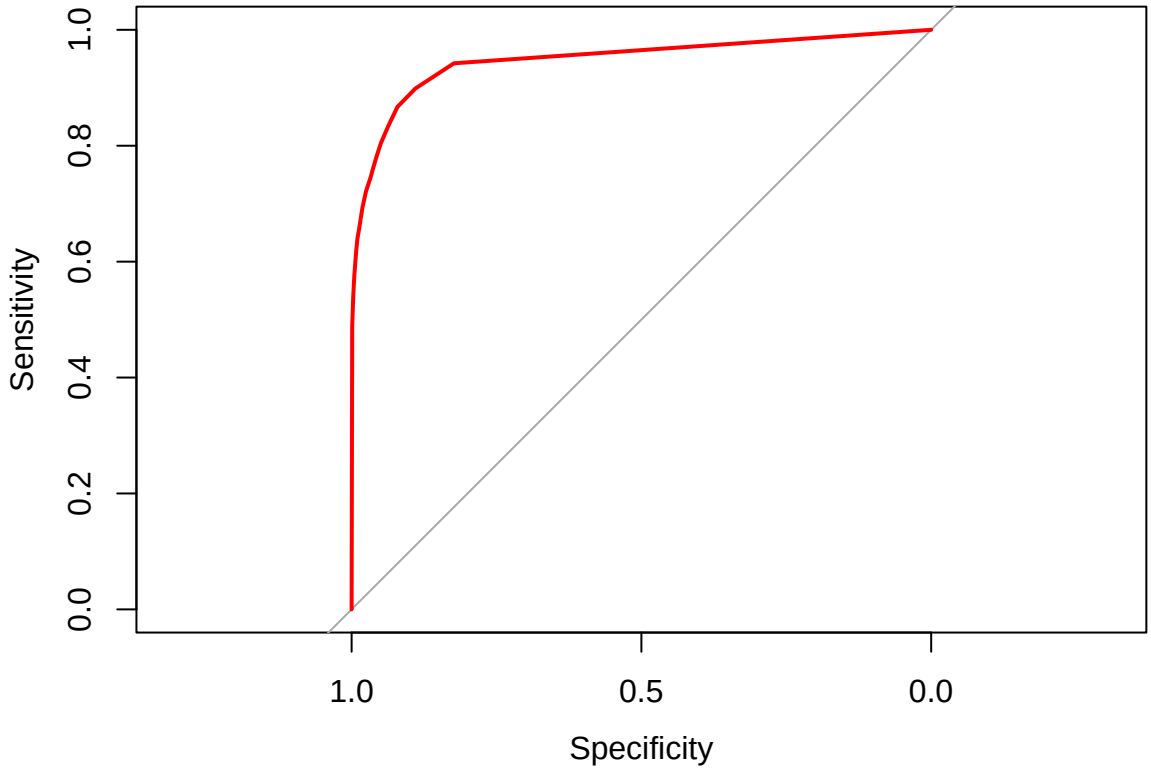
```
## Precision
##    0.9013
```

```r
# result
F1Score_modelo_forest_tune <- (2 * (precision_modelo_forest_tune *
    recall_modelo_forest_tune)/(precision_modelo_forest_tune +
    recall_modelo_forest_tune)) %>%
    round(4)
F1Score_modelo_forest_tune
```

```
## Precision
##    0.7924
```

Se reproduce la curva ROC sobre el modelo final de random forest obtenido:

```r
roc_forest <- plot.roc(as.numeric(fcr_train_forest$loan_status),
    as.numeric(fcr_train_forest$y_pred_probs2), col = "red")
```




```
auc(roc_forest)
```

```
## Area under the curve: 0.9473
```

Se obtiene alrededor de un 94.73% de área bajo la curva.

10. ADABoost - Boosted Classification Tree

Primero se crean unos datos de train específicos para ser usados en el desarrollo del modelo de ADABoost, y así mantener los originales sin modificar.

```
fcr_train_ADABoost <- fcr_train
fcr_train_ADABoost
```

```
## # A tibble: 26,059 x 12
##    person_age person_i~1 perso~2 perso~3 loan_~4 loan_~5 loan_~6 loan_~7 loan_~8
##         <dbl>      <dbl> <fct>     <dbl> <fct>   <fct>     <dbl>   <dbl>   <dbl>
##  1         28      44000 RENT          2 MEDICAL C         10000   13.5        1
##  2         21      35000 OWN           5 VENTURE B          8000    9.91       0
##  3         25      96000 MORTGA~       6 HOMEIM~ C         21000   14.6        0
##  4         22      67000 OWN           5 EDUCAT~ D          7500   16.3        0
##  5         24      52800 RENT          8 PERSON~ A          9000    7.49       0
##  6         27      50004 RENT         12 DEBTCO~ B          3200   11.5        0
```

```
##  7         23      55488 RENT          4 MEDICAL D          5000   15.2        1
##  8         28      70000 RENT          2 DEBTCO~ B          6000   10.4        0
##  9         22      55000 MORTGA~       6 PERSON~ C         13000   13.8        0
## 10         26      43200 RENT          5 EDUCAT~ C          3200   14.4        0
## # ... with 26,049 more rows, 3 more variables: loan_percent_income <dbl>,
## #   cb_person_default_on_file <fct>, cb_person_cred_hist_length <dbl>, and
## #   abbreviated variable names 1: person_income, 2: person_home_ownership,
## #   3: person_emp_length, 4: loan_intent, 5: loan_grade, 6: loan_amnt,
## #   7: loan_int_rate, 8: loan_status
```

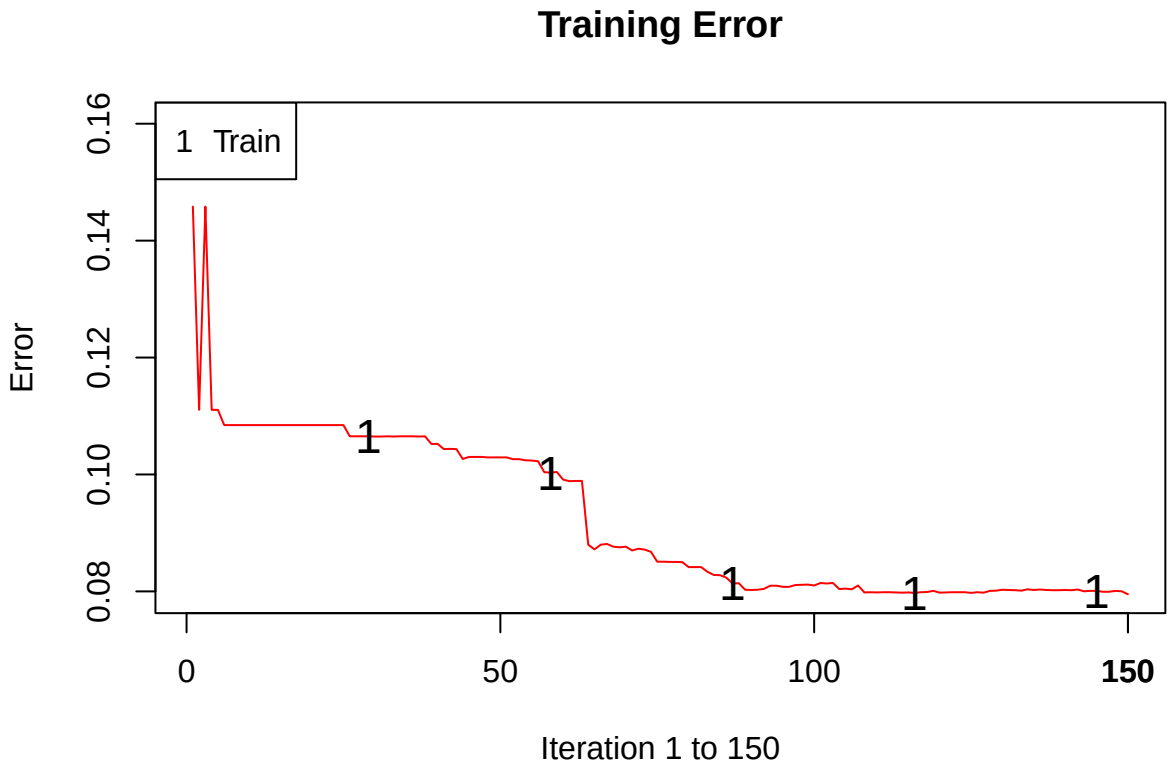
Se crea el modelo de boosting con una configuración inicial básica de parámetros:

```
ada.boost <- ada(as.factor(loan_status) ~ ., data = fcr_train_ADABoost,
    type = "real", control = rpart.control(maxdepth = 2, cp = 0,
        minsplit = 10, xval = 0), iter = 150, nu = 0.05)
ada.boost
```

```
## Call:
## ada(as.factor(loan_status) ~ ., data = fcr_train_ADABoost, type = "real",
##     control = rpart.control(maxdepth = 2, cp = 0, minsplit = 10,
##         xval = 0), iter = 150, nu = 0.05)
##
## Loss: exponential Method: real   Iteration: 150
##
## Final Confusion Matrix for Data:
##           Final Prediction
## True value     0     1
##          0 20147   218
##          1  1854  3840
##
## Train Error: 0.08
##
## Out-Of-Bag Error:  0.087  iteration= 150
##
## Additional Estimates of number of iterations:
##
## train.err1 train.kap1
##        150        150
```

Se analiza la evolución decreciente del error al aumentar el número de iteraciones en el modelo

```
plot(ada.boost)
```



Se pasa a evaluar la precisión del modelo en la muestra de train:

```
set.seed(123)
pred_ada <- predict(ada.boost, newdata = fcr_train_ADABoost)
caret::confusionMatrix(pred_ada, as.factor(fcr_train_ADABoost$loan_status),
    positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 20147  1854
##          1   218  3840
##
##                Accuracy : 0.9205
##                  95% CI : (0.9171, 0.9237)
##     No Information Rate : 0.7815
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7403
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.6744
##             Specificity : 0.9893
##          Pos Pred Value : 0.9463
```

```
##          Neg Pred Value : 0.9157
##              Prevalence : 0.2185
##          Detection Rate : 0.1474
##    Detection Prevalence : 0.1557
##       Balanced Accuracy : 0.8318
##
##        'Positive' Class : 1
##
```

Con la configuración de parámetros realizada en el modelo ada de booting se obtiene un valor de accuracy del 92% para el caso de algoritmos de clasificación.

Para optimizar los resultados del modelo creado y la generalización del modelo, se puede realizar un ajuste de hiperparámetros y validación cruzada:

```
modelLookup("ada")
```

```
##   model parameter          label forReg forClass probModel
## 1   ada      iter         #Trees  FALSE     TRUE      TRUE
## 2   ada  maxdepth Max Tree Depth  FALSE     TRUE      TRUE
## 3   ada        nu  Learning Rate  FALSE     TRUE      TRUE
```

Se ven los parámetros de "iter", "maxdepth" y "nu" que tiene el modelo ada de boosting para árboles de decisión en problemas de clasificación.

```
set.seed(123)
caret.ada <- train(as.factor(loan_status) ~ ., method = "ada",
    data = fcr_train_ADABoost, trControl = trainControl(method = "cv",
        number = 5, search = "grid", returnResamp = "final"))
caret.ada
```

```
## Boosted Classification Trees
##
## 26059 samples
##    11 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 20848, 20847, 20847, 20847, 20847
## Resampling results across tuning parameters:
##
##   maxdepth  iter  Accuracy   Kappa
##   1          50   0.8378294  0.3706354
##   1         100   0.8556738  0.4675745
##   1         150   0.8621974  0.5059552
##   2          50   0.8708702  0.5361573
##   2         100   0.8843396  0.6036405
##   2         150   0.8900958  0.6314373
##   3          50   0.8948157  0.6459253
##   3         100   0.9014163  0.6776206
##   3         150   0.9036804  0.6875708
##
```

```
## Tuning parameter 'nu' was held constant at a value of 0.1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were iter = 150, maxdepth = 3 and nu = 0.1.
```

Se obtiene una configuración óptima de los hiperparámetros del modelo en "iter" = 150, "maxdepth" = 3 y "nu" = 0.1.

```
caret.ada
```

```
## Boosted Classification Trees
##
## 26059 samples
##    11 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 20848, 20847, 20847, 20847, 20847
## Resampling results across tuning parameters:
##
##   maxdepth  iter  Accuracy   Kappa
##   1          50   0.8378294  0.3706354
##   1         100   0.8556738  0.4675745
##   1         150   0.8621974  0.5059552
##   2          50   0.8708702  0.5361573
##   2         100   0.8843396  0.6036405
##   2         150   0.8900958  0.6314373
##   3          50   0.8948157  0.6459253
##   3         100   0.9014163  0.6776206
##   3         150   0.9036804  0.6875708
##
## Tuning parameter 'nu' was held constant at a value of 0.1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were iter = 150, maxdepth = 3 and nu = 0.1.
```

```
caret.ada$bestTune
```

```
##   iter maxdepth  nu
## 9  150        3 0.1
```

Con el modelo de base se obtiene un accuracy del 90% con los datos de train.

```
get_best_result = function(caret_fit) {
    best = which(rownames(caret_fit$results) == rownames(caret_fit$bestTune))
    best_result = caret_fit$results[best, ]
    rownames(best_result) = NULL
    best_result
}

get_best_result(caret.ada)
```

```
##    nu maxdepth iter  Accuracy     Kappa  AccuracySD    KappaSD
## 1 0.1        3  150 0.9036804 0.6875708 0.004402523 0.01426331
```

Evaluación del rendimiento predictivo del modelo Ada Boost presentado con las datos de train (metrica de evaluación utilizada de referencia: “Accuracy”, “Recall”, “Precision”, “F1” y “ROC”, y punto de corte utilizado: 0.5):

```
fcr_train_ADABoost$y_pred_probs2 <- predict(caret.ada, newdata = fcr_train_ADABoost,
    type = "prob")
fcr_train_ADABoost$y_pred_probs2 <- ifelse(fcr_train_ADABoost$y_pred_probs2$`1` >
    0.5, fcr_train_ADABoost$y_pred_probs2$`1`, 1 - fcr_train_ADABoost$y_pred_probs2$`0`)

fcr_train_ADABoost$y_pred2 <- ifelse(fcr_train_ADABoost$y_pred_probs2 >
    0.5, 1, 0)

# fcr_train_ADABoost$y_pred_probs2
# fcr_train_ADABoost$y_pred2 fcr_train_ADABoost
```

Se reproduce la matriz de confusión y las métricas de evaluación sobre el modelo final de ADABoost obtenido:

```
cm_train_ada <- confusionMatrix(as.factor(fcr_train_ADABoost$y_pred2),
    as.factor(fcr_train_ADABoost$loan_status), positive = "1")
cm_train_ada$table
```

```
##           Reference
## Prediction     0     1
##          0 19893  2037
##          1   472  3657
```

```
# result
accuracy_modelo_ada_tune <- cm_train_ada$overall["Accuracy"] %>%
    round(4)
accuracy_modelo_ada_tune
```

```
## Accuracy
##   0.9037
```

```
# result
recall_modelo_ada_tune <- cm_train_ada$byClass["Recall"] %>%
    round(4)
recall_modelo_ada_tune
```

```
## Recall
## 0.6423
```

```
# result
precision_modelo_ada_tune <- cm_train_ada$byClass["Precision"] %>%
    round(4)
precision_modelo_ada_tune
```

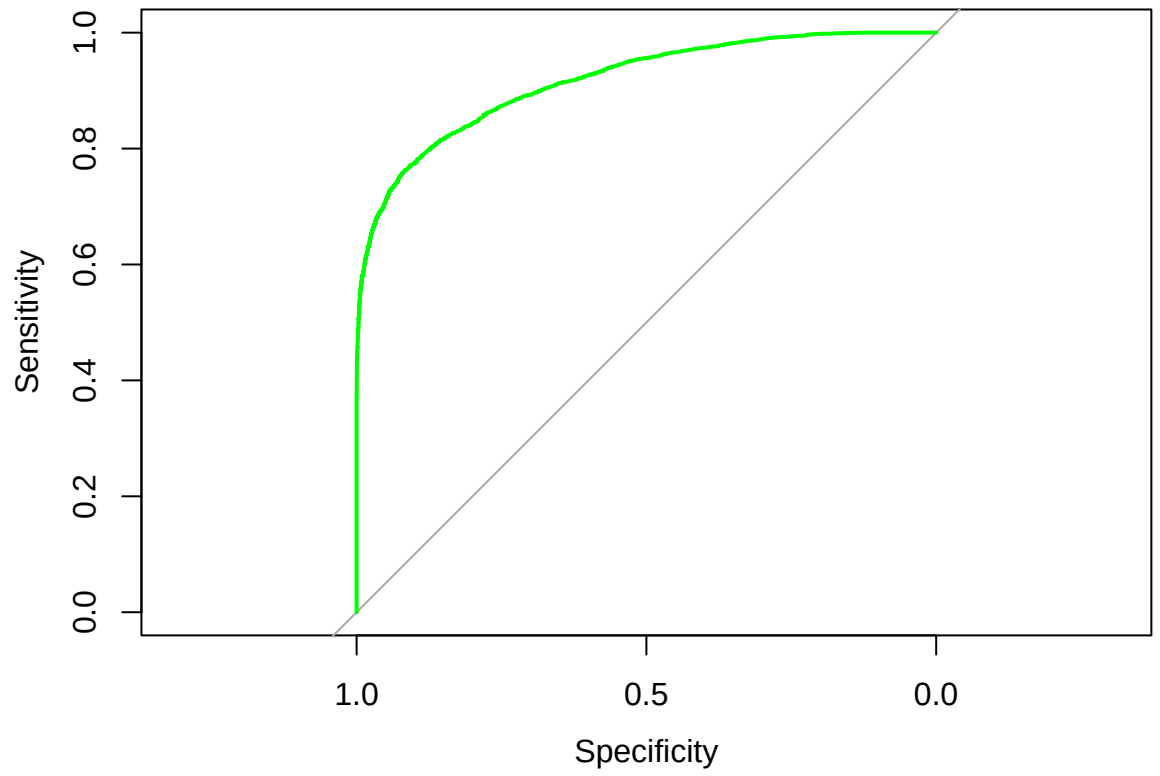
```
## Precision
##    0.8857
```

```
# result
F1Score_modelo_ada_tune <- (2 * (precision_modelo_ada_tune *
    recall_modelo_ada_tune)/(precision_modelo_ada_tune + recall_modelo_ada_tune)) %>%
    round(4)
F1Score_modelo_ada_tune
```

```
## Precision
##    0.7446
```

Se reproduce la curva ROC sobre el modelo final de ADABoost obtenido:

```
roc_ada <- plot.roc(as.numeric(fcr_train_ADABoost$loan_status),
    as.numeric(fcr_train_ADABoost$y_pred_probs2), col = "green")
```




```
auc(roc_ada)
```

```
## Area under the curve: 0.9169
```

Se obtiene alrededor de un 91.69% de área bajo la curva.

11. XGBoost - Extreme Gradient Boosting

Primero se crean unos datos de train específicos para ser usados en el desarrollo del modelo de XGBoost, y así mantener los originales sin modificar.

```
fcr_train_XGBoost <- fcr_train
fcr_train_XGBoost
```

```
## # A tibble: 26,059 x 12
##    person_age person_i~1 perso~2 perso~3 loan_~4 loan_~5 loan_~6 loan_~7 loan_~8
##         <dbl>      <dbl> <fct>     <dbl> <fct>   <fct>     <dbl>   <dbl>   <dbl>
##  1         28      44000 RENT          2 MEDICAL C         10000   13.5        1
##  2         21      35000 OWN           5 VENTURE B          8000    9.91       0
##  3         25      96000 MORTGA~       6 HOMEIM~ C         21000   14.6        0
##  4         22      67000 OWN           5 EDUCAT~ D          7500   16.3        0
##  5         24      52800 RENT          8 PERSON~ A          9000    7.49       0
##  6         27      50004 RENT         12 DEBTCO~ B          3200   11.5        0
##  7         23      55488 RENT          4 MEDICAL D          5000   15.2        1
##  8         28      70000 RENT          2 DEBTCO~ B          6000   10.4        0
##  9         22      55000 MORTGA~       6 PERSON~ C         13000   13.8        0
## 10         26      43200 RENT          5 EDUCAT~ C          3200   14.4        0
## # ... with 26,049 more rows, 3 more variables: loan_percent_income <dbl>,
## #   cb_person_default_on_file <fct>, cb_person_cred_hist_length <dbl>, and
## #   abbreviated variable names 1: person_income, 2: person_home_ownership,
## #   3: person_emp_length, 4: loan_intent, 5: loan_grade, 6: loan_amnt,
## #   7: loan_int_rate, 8: loan_status
```

Para optimizar los resultados del modelo creado, se puede realizar un ajuste de hiperparámetros con validación cruzada:

```
modelLookup("xgbTree")
```

```
##     model        parameter                          label forReg forClass
## 1 xgbTree          nrounds          # Boosting Iterations   TRUE     TRUE
## 2 xgbTree        max_depth                 Max Tree Depth   TRUE     TRUE
## 3 xgbTree              eta                     Shrinkage   TRUE     TRUE
## 4 xgbTree            gamma         Minimum Loss Reduction   TRUE     TRUE
## 5 xgbTree colsample_bytree     Subsample Ratio of Columns   TRUE     TRUE
## 6 xgbTree min_child_weight Minimum Sum of Instance Weight   TRUE     TRUE
## 7 xgbTree        subsample           Subsample Percentage   TRUE     TRUE
##   probModel
## 1      TRUE
## 2      TRUE
## 3      TRUE
## 4      TRUE
## 5      TRUE
## 6      TRUE
## 7      TRUE
```

Se crea el modelo de boosting con una configuración inicial dada de parámetros:

```
set.seed(2)
caret.xgb <- train(as.factor(loan_status) ~ ., method = "xgbTree",
    data = fcr_train_XGBoost, trControl = trainControl(method = "cv",
        number = 5, search = "grid", returnResamp = "final"))
```

```
## [20:51:55] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:55] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:56] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:56] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:56] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:56] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:57] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:57] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:57] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:57] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:57] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:57] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:58] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:58] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:58] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:58] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:59] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:59] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:59] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:51:59] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:00] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:00] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:00] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:00] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:01] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:01] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:01] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:01] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:02] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:02] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:03] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:03] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:03] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:03] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:04] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:04] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:04] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:04] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:05] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:05] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:05] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:05] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:06] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:06] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:06] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:06] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:06] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:06] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:07] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:07] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:07] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:07] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:08] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:08] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
```

```
## [20:52:08] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:08] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:09] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:09] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:09] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:09] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:10] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:10] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:10] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:10] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:11] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:11] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:12] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:12] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:13] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:13] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:14] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:14] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:14] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:14] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:15] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:15] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:15] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:15] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:16] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:16] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:16] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:16] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:17] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:17] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:17] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:17] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:18] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:18] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:19] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:19] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:19] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:19] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:20] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:20] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:20] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:20] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:21] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:21] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:22] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:22] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:22] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:22] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:23] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:23] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:24] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:24] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:24] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [20:52:24] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
```

```
## [20:52:25] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:25] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:25] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:25] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:25] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:25] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:26] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:26] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:26] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:26] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:26] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:26] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:27] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:27] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:27] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:27] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:28] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:28] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:28] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:28] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:29] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:29] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:29] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:29] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:30] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:30] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:31] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:31] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:31] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:31] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:32] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:32] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:33] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:33] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:33] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:33] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:34] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:34] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:34] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:34] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:34] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:34] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:35] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:35] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:35] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:35] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:35] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:35] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:36] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:36] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:36] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:36] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:37] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:37] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
```

```
## [20:52:37] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:37] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:38] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:38] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:38] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:38] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:39] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:39] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:40] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:40] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:40] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:40] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:41] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:41] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:42] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:42] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:42] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:42] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:43] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:43] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:43] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:43] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:43] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:43] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:44] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:44] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:44] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:44] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:45] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:45] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:45] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:45] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:46] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:46] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:47] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:47] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:47] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:47] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:48] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:48] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:49] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:49] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:49] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:49] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:50] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:50] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:51] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:51] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:52] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:52] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:53] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:53] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:53] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:53] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
```

```
## [20:52:54] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:54] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:54] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:54] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:55] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:55] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:55] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:55] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:55] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:55] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:56] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:56] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:56] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:56] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:57] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:57] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:57] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:57] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:58] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:58] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:58] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:58] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:58] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:58] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:59] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:52:59] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:00] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:00] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:00] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:00] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:01] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:01] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:02] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:02] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:02] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:02] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:03] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:03] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:03] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:03] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:03] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:03] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:04] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:04] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:04] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:04] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:05] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:05] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:05] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:05] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:05] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:05] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:06] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:06] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
```

```
## [20:53:06] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:06] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:07] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:07] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:07] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:07] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:08] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:08] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:09] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:09] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:09] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:09] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:10] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:10] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:11] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:11] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:11] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:11] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:12] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:12] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:12] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:12] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:12] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:12] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:13] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:13] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:13] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:13] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:13] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:13] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:14] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:14] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:14] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:14] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:15] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:15] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:15] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:15] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:16] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:16] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:16] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:16] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:17] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:17] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:18] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:18] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:18] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:18] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:19] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:19] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:20] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:20] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:20] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:20] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
```

```
## [20:53:21] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:21] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:21] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:21] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:21] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:21] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:22] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:22] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:22] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:22] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:22] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:22] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:23] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:23] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:23] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:23] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:24] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:24] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:24] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:24] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:25] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:25] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:25] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:25] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:26] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:26] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:27] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:27] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:28] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:28] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:29] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:29] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:30] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:30] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:31] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
## [20:53:31] WARNING: src/c_api/c_api.cc:935: ‘ntree_limit‘ is deprecated, use ‘iteration_range‘ instea
```

```
caret.xgb
```

```
## eXtreme Gradient Boosting
##
## 26059 samples
##    11 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 20847, 20847, 20848, 20847, 20847
## Resampling results across tuning parameters:
##
##   eta  max_depth  colsample_bytree  subsample  nrounds  Accuracy   Kappa
##   0.3  1          0.6               0.50        50      0.8846845  0.6293761
##   0.3  1          0.6               0.50       100      0.8876393  0.6462293
##   0.3  1          0.6               0.50       150      0.8884836  0.6510768
```

```
##   0.3  1          0.6              0.75         50       0.8834950  0.6266359
##   0.3  1          0.6              0.75        100       0.8876394  0.6473287
##   0.3  1          0.6              0.75        150       0.8880999  0.6493912
##   0.3  1          0.6              1.00         50       0.8841473  0.6284229
##   0.3  1          0.6              1.00        100       0.8874091  0.6451641
##   0.3  1          0.6              1.00        150       0.8880231  0.6492723
##   0.3  1          0.8              0.50         50       0.8847996  0.6308522
##   0.3  1          0.8              0.50        100       0.8875626  0.6466938
##   0.3  1          0.8              0.50        150       0.8891360  0.6523035
##   0.3  1          0.8              0.75         50       0.8829193  0.6250569
##   0.3  1          0.8              0.75        100       0.8876777  0.6467031
##   0.3  1          0.8              0.75        150       0.8890975  0.6525131
##   0.3  1          0.8              1.00         50       0.8827275  0.6247079
##   0.3  1          0.8              1.00        100       0.8873708  0.6458530
##   0.3  1          0.8              1.00        150       0.8882534  0.6496125
##   0.3  2          0.6              0.50         50       0.9103187  0.7125871
##   0.3  2          0.6              0.50        100       0.9198356  0.7437571
##   0.3  2          0.6              0.50        150       0.9227906  0.7536219
##   0.3  2          0.6              0.75         50       0.9113931  0.7156087
##   0.3  2          0.6              0.75        100       0.9216009  0.7488730
##   0.3  2          0.6              0.75        150       0.9246708  0.7589696
##   0.3  2          0.6              1.00         50       0.9098968  0.7116771
##   0.3  2          0.6              1.00        100       0.9206800  0.7461793
##   0.3  2          0.6              1.00        150       0.9249780  0.7599866
##   0.3  2          0.8              0.50         50       0.9109328  0.7142251
##   0.3  2          0.8              0.50        100       0.9204496  0.7457767
##   0.3  2          0.8              0.50        150       0.9238650  0.7567777
##   0.3  2          0.8              0.75         50       0.9112781  0.7156776
##   0.3  2          0.8              0.75        100       0.9218696  0.7502092
##   0.3  2          0.8              0.75        150       0.9254384  0.7614074
##   0.3  2          0.8              1.00         50       0.9133888  0.7216716
##   0.3  2          0.8              1.00        100       0.9227520  0.7525232
##   0.3  2          0.8              1.00        150       0.9250547  0.7601430
##   0.3  3          0.6              0.50         50       0.9229440  0.7531913
##   0.3  3          0.6              0.50        100       0.9290840  0.7729617
##   0.3  3          0.6              0.50        150       0.9303120  0.7778724
##   0.3  3          0.6              0.75         50       0.9232509  0.7537459
##   0.3  3          0.6              0.75        100       0.9291606  0.7725715
##   0.3  3          0.6              0.75        150       0.9310793  0.7797916
##   0.3  3          0.6              1.00         50       0.9233277  0.7536732
##   0.3  3          0.6              1.00        100       0.9295061  0.7741745
##   0.3  3          0.6              1.00        150       0.9316551  0.7812893
##   0.3  3          0.8              0.50         50       0.9237114  0.7549548
##   0.3  3          0.8              0.50        100       0.9290840  0.7733105
##   0.3  3          0.8              0.50        150       0.9298131  0.7759563
##   0.3  3          0.8              0.75         50       0.9243254  0.7570661
##   0.3  3          0.8              0.75        100       0.9304654  0.7776318
##   0.3  3          0.8              0.75        150       0.9324226  0.7841987
##   0.3  3          0.8              1.00         50       0.9247093  0.7578303
##   0.3  3          0.8              1.00        100       0.9307341  0.7773397
##   0.3  3          0.8              1.00        150       0.9322307  0.7832326
##   0.4  1          0.6              0.50         50       0.8853752  0.6360333
##   0.4  1          0.6              0.50        100       0.8883301  0.6496788
##   0.4  1          0.6              0.50        150       0.8895581  0.6553219
```

```
##   0.4  1          0.6              0.75        50      0.8836485  0.6320133
##   0.4  1          0.6              0.75       100      0.8873708  0.6471157
##   0.4  1          0.6              0.75       150      0.8887522  0.6525672
##   0.4  1          0.6              1.00        50      0.8867184  0.6398936
##   0.4  1          0.6              1.00       100      0.8875627  0.6473209
##   0.4  1          0.6              1.00       150      0.8891360  0.6535855
##   0.4  1          0.8              0.50        50      0.8867184  0.6412765
##   0.4  1          0.8              0.50       100      0.8882151  0.6500226
##   0.4  1          0.8              0.50       150      0.8898267  0.6556014
##   0.4  1          0.8              0.75        50      0.8864497  0.6385784
##   0.4  1          0.8              0.75       100      0.8880615  0.6488683
##   0.4  1          0.8              0.75       150      0.8889057  0.6525744
##   0.4  1          0.8              1.00        50      0.8857207  0.6370359
##   0.4  1          0.8              1.00       100      0.8874859  0.6475630
##   0.4  1          0.8              1.00       150      0.8893662  0.6544535
##   0.4  2          0.6              0.50        50      0.9153458  0.7286560
##   0.4  2          0.6              0.50       100      0.9225604  0.7527931
##   0.4  2          0.6              0.50       150      0.9244791  0.7594369
##   0.4  2          0.6              0.75        50      0.9145399  0.7275913
##   0.4  2          0.6              0.75       100      0.9232126  0.7553891
##   0.4  2          0.6              0.75       150      0.9260524  0.7644168
##   0.4  2          0.6              1.00        50      0.9123908  0.7201479
##   0.4  2          0.6              1.00       100      0.9225601  0.7525713
##   0.4  2          0.6              1.00       150      0.9263210  0.7644369
##   0.4  2          0.8              0.50        50      0.9155761  0.7298653
##   0.4  2          0.8              0.50       100      0.9225987  0.7524881
##   0.4  2          0.8              0.50       150      0.9258989  0.7635358
##   0.4  2          0.8              0.75        50      0.9158832  0.7305039
##   0.4  2          0.8              0.75       100      0.9231743  0.7541328
##   0.4  2          0.8              0.75       150      0.9270885  0.7670440
##   0.4  2          0.8              1.00        50      0.9149238  0.7269784
##   0.4  2          0.8              1.00       100      0.9235581  0.7552415
##   0.4  2          0.8              1.00       150      0.9265897  0.7658694
##   0.4  3          0.6              0.50        50      0.9251698  0.7595352
##   0.4  3          0.6              0.50       100      0.9288922  0.7727747
##   0.4  3          0.6              0.50       150      0.9304655  0.7789727
##   0.4  3          0.6              0.75        50      0.9245173  0.7585563
##   0.4  3          0.6              0.75       100      0.9300816  0.7772867
##   0.4  3          0.6              0.75       150      0.9308108  0.7802000
##   0.4  3          0.6              1.00        50      0.9254383  0.7614990
##   0.4  3          0.6              1.00       100      0.9308108  0.7791370
##   0.4  3          0.6              1.00       150      0.9325761  0.7850926
##   0.4  3          0.8              0.50        50      0.9261675  0.7644666
##   0.4  3          0.8              0.50       100      0.9300049  0.7770532
##   0.4  3          0.8              0.50       150      0.9299282  0.7779567
##   0.4  3          0.8              0.75        50      0.9263595  0.7640630
##   0.4  3          0.8              0.75       100      0.9308109  0.7789249
##   0.4  3          0.8              0.75       150      0.9317703  0.7830458
##   0.4  3          0.8              1.00        50      0.9270885  0.7663427
##   0.4  3          0.8              1.00       100      0.9319620  0.7823425
##   0.4  3          0.8              1.00       150      0.9330749  0.7865927
##
## Tuning parameter 'gamma' was held constant at a value of 0
## Tuning
```

```
##  parameter 'min_child_weight' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were nrounds = 150, max_depth = 3, eta
##  = 0.4, gamma = 0, colsample_bytree = 0.8, min_child_weight = 1 and subsample
##  = 1.
```

Se obtiene una configuración óptima de los hiperparámetros del modelo en:

```
get_best_result = function(caret_fit) {
    best = which(rownames(caret_fit$results) == rownames(caret_fit$bestTune))
    best_result = caret_fit$results[best, ]
    rownames(best_result) = NULL
    best_result
}

get_best_result(caret.xgb)
```

```
##   eta max_depth gamma colsample_bytree min_child_weight subsample nrounds
## 1 0.4         3     0              0.8                1         1     150
##    Accuracy     Kappa AccuracySD     KappaSD
## 1 0.9330749 0.7865927 0.00189919 0.006102938
```

Se analiza la relevancia de cada variable en el modelo:

```
varImp(caret.xgb)
```

```
## xgbTree variable importance
##
##   only 20 most important variables shown (out of 22)
##
##                              Overall
## loan_percent_income        100.0000
## person_income               78.7401
## loan_int_rate               59.3389
## person_home_ownershipRENT   45.6475
## loan_gradeD                 31.4561
## person_home_ownershipOWN    14.4366
## person_emp_length           13.3073
## loan_intentHOMEIMPROVEMENT  11.3466
## loan_intentMEDICAL           6.9108
## loan_amnt                    6.4238
## loan_intentEDUCATION         6.3647
## loan_gradeE                  5.4250
## loan_intentVENTURE           5.1817
## person_age                   4.6696
## loan_gradeC                  2.5971
## loan_gradeG                  2.0848
## loan_intentPERSONAL          1.8264
## loan_gradeF                  1.5411
## cb_person_cred_hist_length   0.7322
## cb_person_default_on_fileY   0.5910
```

Evaluación del rendimiento predictivo del modelo XG Boost presentado con las datos de train (metrica de evaluación utilizada de referencia: "Accuracy", "Recall", "Precision", "F1" y "ROC", y punto de corte utilizado: 0.5):

```
fcr_train_XGBoost$y_pred_probs2 <- predict(caret.xgb, newdata = fcr_train_XGBoost,
    type = "prob")
fcr_train_XGBoost$y_pred_probs2 <- ifelse(fcr_train_XGBoost$y_pred_probs2$`1` >
    0.5, fcr_train_XGBoost$y_pred_probs2$`1`, 1 - fcr_train_XGBoost$y_pred_probs2$`0`)

fcr_train_XGBoost$y_pred2 <- ifelse(fcr_train_XGBoost$y_pred_probs2 >
    0.5, 1, 0)

# fcr_train_XGBoost$y_pred_probs2 fcr_train_XGBoost$y_pred2
# fcr_train_XGBoost
```

Se reproduce la matriz de confusión y las métricas de evaluación sobre el modelo final de XG Boost obtenido:

```
cm_train_xgb <- confusionMatrix(as.factor(fcr_train_XGBoost$y_pred2),
    as.factor(fcr_train_XGBoost$loan_status), positive = "1")
cm_train_xgb$table
```

```
##           Reference
## Prediction     0     1
##          0 20252  1430
##          1   113  4264
```

```
# result
accuracy_modelo_xgb_tune <- cm_train_xgb$overall["Accuracy"] %>%
    round(4)
accuracy_modelo_xgb_tune
```

```
## Accuracy
##   0.9408
```

```
# result
recall_modelo_xgb_tune <- cm_train_xgb$byClass["Recall"] %>%
    round(4)
recall_modelo_xgb_tune
```

```
## Recall
## 0.7489
```

```
# result
precision_modelo_xgb_tune <- cm_train_xgb$byClass["Precision"] %>%
    round(4)
precision_modelo_xgb_tune
```

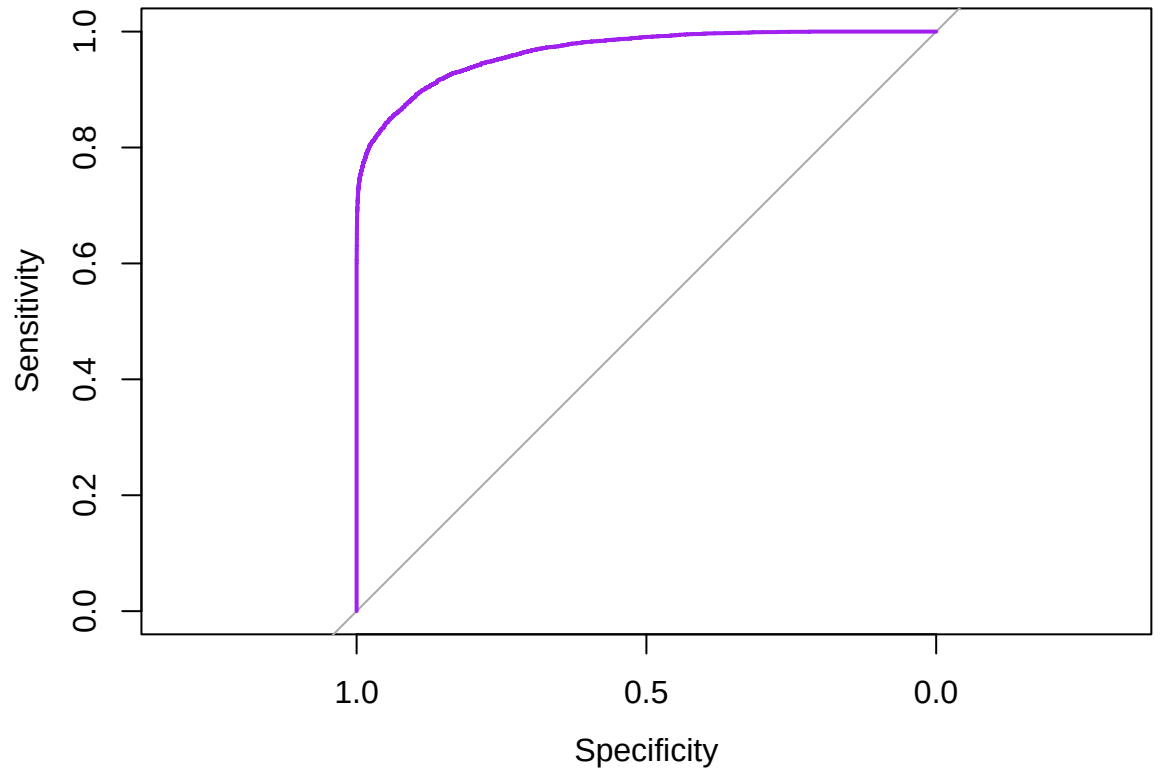
```
## Precision
##    0.9742
```

```
# result
F1Score_modelo_xgb_tune <- (2 * (precision_modelo_xgb_tune *
    recall_modelo_xgb_tune)/(precision_modelo_xgb_tune + recall_modelo_xgb_tune)) %>%
    round(4)
F1Score_modelo_xgb_tune
```

```
## Precision
##    0.8468
```

Se reproduce la curva ROC sobre el modelo final de XGBoost obtenido:

```
roc_xgb <- plot.roc(as.numeric(fcr_train_XGBoost$loan_status),
    as.numeric(fcr_train_XGBoost$y_pred_probs2), col = "purple")
```




```
auc(roc_xgb)
```

```
## Area under the curve: 0.9654
```

Se obtiene alrededor de un 96.54% de área bajo la curva.

COMPARACIÓN DE LAS TÉCNICAS Y MODELOS DE MACHINE LEARNING

12. Comparación entre los diferentes modelos

Los modelo que han sido evaluados en este trabajo para los datos analizados han sido:

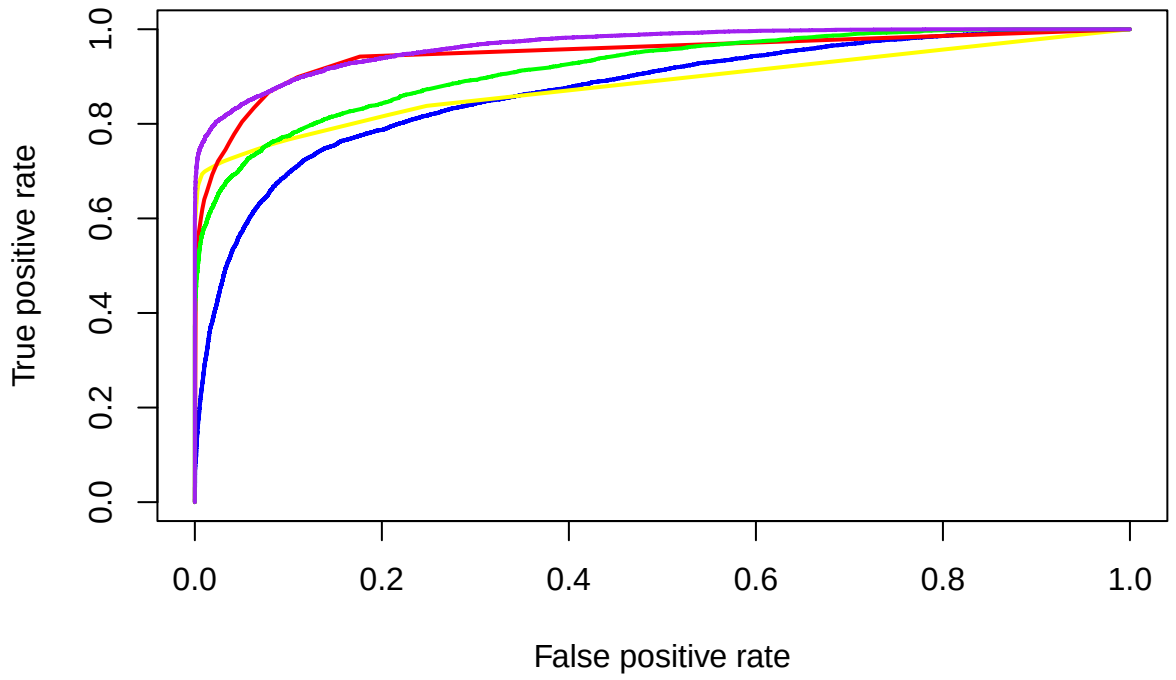
- **Modelo de regresión logística (GLM)**
- **Modelo de árbol de decisión (Decision Tree)**
- **Modelo de bosque (Random Forest)**
- **Modelo de métodos de ensamble (ADABoost y XGBoost)**

```
accuracy <- c(accuracy_modelo_glm2_tune, accuracy_modelo_tree_tune,
    accuracy_modelo_forest_tune, accuracy_modelo_ada_tune, accuracy_modelo_xgb_tune)

recall <- c(recall_modelo_glm2_tune, recall_modelo_tree_tune,
    recall_modelo_forest_tune, recall_modelo_ada_tune, recall_modelo_xgb_tune)

precision <- c(precision_modelo_glm2_tune, precision_modelo_tree_tune,
    precision_modelo_forest_tune, precision_modelo_ada_tune,
    precision_modelo_xgb_tune)

F1Score <- c(F1Score_modelo_glm2_tune, F1Score_modelo_tree_tune,
    F1Score_modelo_forest_tune, F1Score_modelo_ada_tune, F1Score_modelo_xgb_tune)

ROC <- c(auc(roc_glm), auc(roc_tree), auc(roc_forest), auc(roc_ada),
    auc(roc_xgb))

modelo <- c("GLM", "Decision Tree", "Random Forest", "ADABoost",
    "XGBoost")

datos <- data.frame(modelo = modelo, accuracy = accuracy, recall = recall,
    precision = precision, F1Score = F1Score, ROC = ROC)

head(datos)
```

```
##          modelo accuracy recall precision F1Score       ROC
## 1           GLM   0.8671 0.5604    0.7689  0.6483 0.8703091
## 2 Decision Tree   0.9270 0.6934    0.9620  0.8059 0.8829187
## 3 Random Forest   0.9190 0.7069    0.9013  0.7924 0.9473037
## 4      ADABoost   0.9037 0.6423    0.8857  0.7446 0.9169275
## 5       XGBoost   0.9408 0.7489    0.9742  0.8468 0.9653826
```

```
# GLM
pred1 <- predict(modelo_glm2, fcr_train_glm, type = "response")
pred.glm <- prediction(pred1, fcr_train_glm$loan_status)
perf.glm <- performance(pred.glm, "tpr", "fpr")
plot(perf.glm, col = "blue", lwd = 2)

# DECISION TREE
pred3 <- predict(caret.tree, newdata = fcr_train_tree, type = "prob")
```

```r
pred33 <- ifelse(pred3$`1` > 0.5, pred3$`1`, 1 - pred3$`0`)
pred.tree <- prediction(pred33, fcr_train_tree$loan_status)
perf.tree <- performance(pred.tree, "tpr", "fpr")
plot(perf.tree, add = TRUE, col = "yellow", lwd = 2)

# RANDOM FOREST
pred4 <- predict(caret.rf, newdata = fcr_train_forest, type = "prob")
pred44 <- ifelse(pred4$`1` > 0.5, pred4$`1`, 1 - pred4$`0`)
pred.rf <- prediction(pred44, fcr_train_forest$loan_status)
perf.rf <- performance(pred.rf, "tpr", "fpr")
plot(perf.rf, add = TRUE, col = "red", lwd = 2)

# ADA BOOST
pred5 <- predict(caret.ada, newdata = fcr_train_ADABoost, type = "prob")
pred55 <- ifelse(pred5$`1` > 0.5, pred5$`1`, 1 - pred5$`0`)
pred.ada <- prediction(pred55, fcr_train_ADABoost$loan_status)
perf.ada <- performance(pred.ada, "tpr", "fpr")
plot(perf.ada, add = TRUE, col = "green", lwd = 2)

# XG BOOST
pred6 <- predict(caret.xgb, newdata = fcr_train_XGBoost, type = "prob")
pred66 <- ifelse(pred6$`1` > 0.5, pred6$`1`, 1 - pred6$`0`)
pred.xgb <- prediction(pred66, fcr_train_XGBoost$loan_status)
perf.xgb <- performance(pred.xgb, "tpr", "fpr")
plot(perf.xgb, add = TRUE, col = "purple", lwd = 2)
```



Con los datos obtenidos, el algoritmo que mejores resultados ha arrojado es el de XGBoost, y será el que utilicemos en la posterior validación de los resultado.

VALIDACIÓN DE LOS RESULTADOS

13. Comprobación de los resultados con los datos de test

13.1. Cambios, modificaciones y transformaciones sobre el dataset de test

Se realizan los cambios y modificaciones necesarias sobre el conjunto de datos de test, aplicados previamente sobre nuestro dataset de train:

- Ya se han realizado anteriormente la imputación de NAs del data set de test al valor de la mediana de la variable de referencia.
- Se ha optado finalmente en el trabajo por no tranformar ninguna variable a logaritmos, por lo que en el data set de test no habría que realizar nada al respecto.

13.2. Comprobación del mejor modelo con datos de test

Comprobación del modelo de XG Boost con los datos de test:

Se procede ahora a validar la capacidad predictora del modelo desarrollado de XG Boost con el conjunto de datos de test.

```
fcr_test$y_pred_probs2 <- predict(caret.xgb, newdata = fcr_test,
    type = "prob")
fcr_test$y_pred_probs2 <- ifelse(fcr_test$y_pred_probs2$`1` >
    0.5, fcr_test$y_pred_probs2$`1`, 1 - fcr_test$y_pred_probs2$`0`)

fcr_test$y_pred2 <- ifelse(fcr_test$y_pred_probs2 > 0.5, 1, 0)

# fcr_test$y_pred_probs2 fcr_test$y_pred2 fcr_test
```

Se reproduce la matriz de confusión y las métricas de evaluación sobre el modelo final de XG Boost obtenido:

```
cm_test_xgb <- confusionMatrix(as.factor(fcr_test$y_pred2), as.factor(fcr_test$loan_status),
    positive = "1")
cm_test_xgb$table
```

```
##           Reference
## Prediction    0    1
##          0 5045  380
##          1   57 1033
```

```
# result
accuracy_modelo_xgb_tune_test <- cm_test_xgb$overall["Accuracy"] %>%
    round(4)
accuracy_modelo_xgb_tune_test
```

```
## Accuracy
##   0.9329
```

```
# result
recall_modelo_xgb_tune_test <- cm_test_xgb$byClass["Recall"] %>%
    round(4)
recall_modelo_xgb_tune_test
```

```
## Recall
## 0.7311
```

```
# result
precision_modelo_xgb_tune_test <- cm_test_xgb$byClass["Precision"] %>%
    round(4)
precision_modelo_xgb_tune_test
```

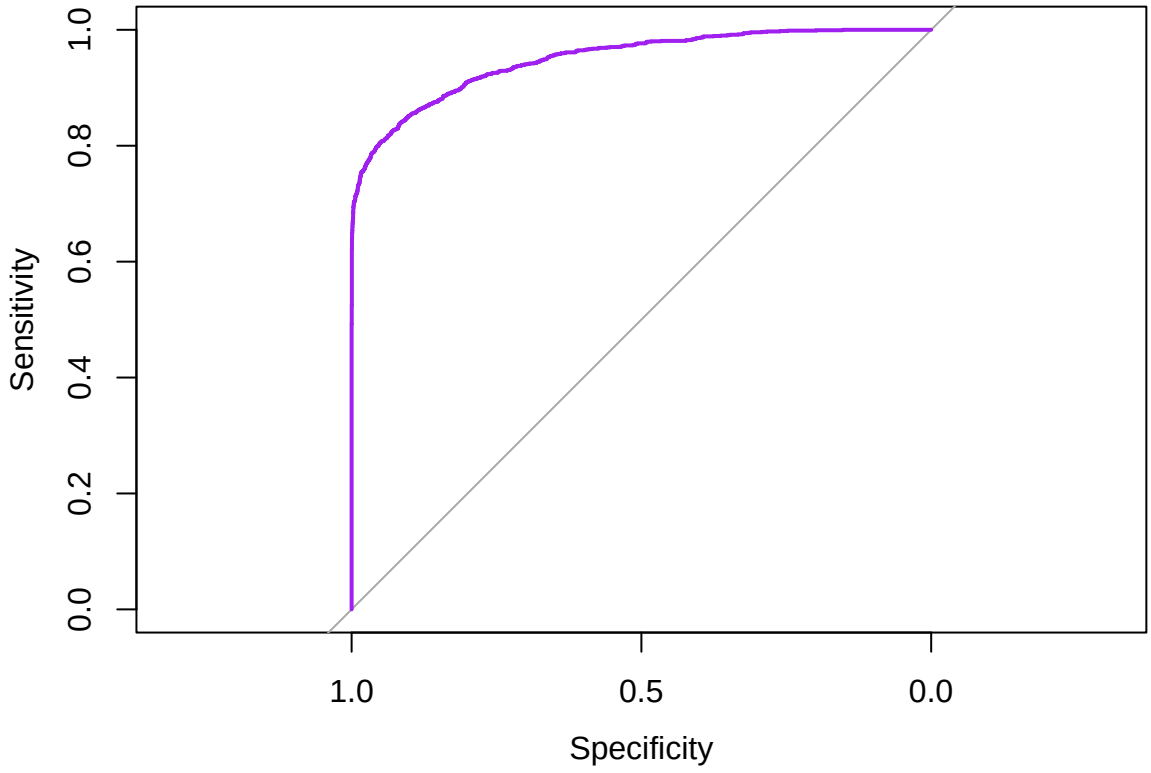
```
## Precision
##    0.9477
```

```
# result
F1Score_modelo_xgb_tune_test <- (2 * (precision_modelo_xgb_tune_test *
    recall_modelo_xgb_tune_test)/(precision_modelo_xgb_tune_test +
    recall_modelo_xgb_tune_test)) %>%
    round(4)
F1Score_modelo_xgb_tune_test
```

```
## Precision
##    0.8254
```

Se comprueba el área bajo la curva para el modelo de XG Boost en el conjunto de datos de test:

```
roc_xgb_test <- plot.roc(as.numeric(fcr_test$loan_status), as.numeric(fcr_test$y_pred_probs2),
    col = "purple")
```




```
auc(roc_xgb_test)
```

```
## Area under the curve: 0.9492
```

Se obtiene alrededor de un 94.92% de área bajo la curva.

13.2. Comparativa del mejor modelo con datos de train vs datos de test

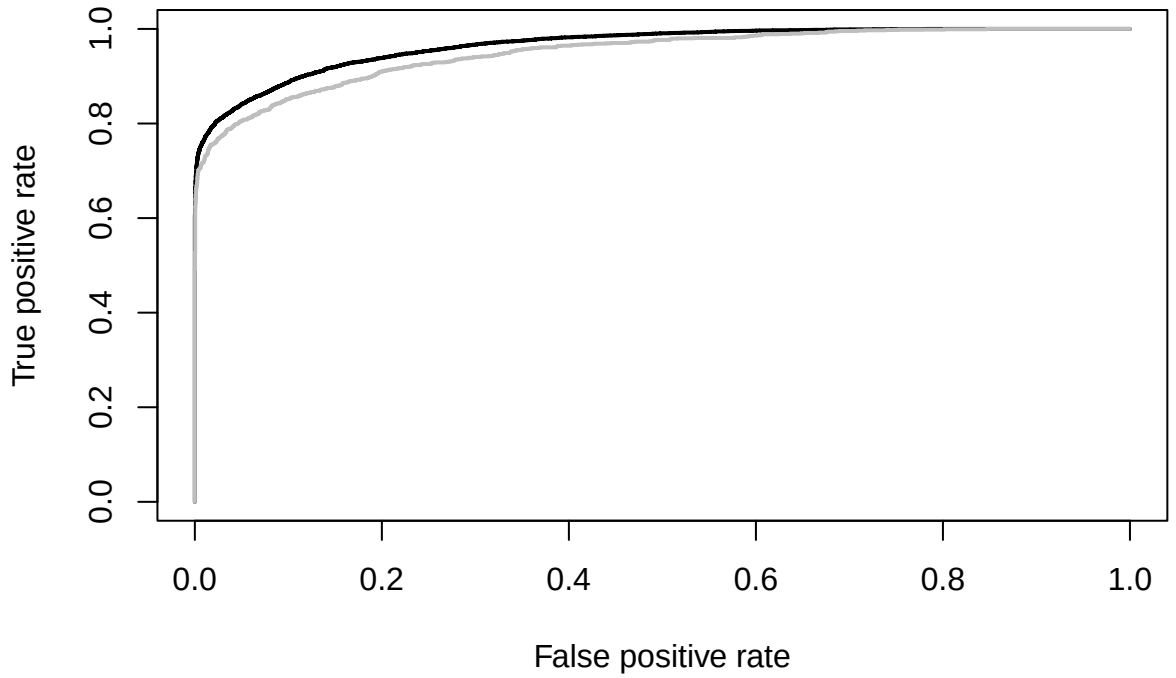
Comparativa del modelo de XG Boost en train y test:

```
accuracy2 <- c(accuracy_modelo_xgb_tune, accuracy_modelo_xgb_tune_test)

recall2 <- c(recall_modelo_xgb_tune, recall_modelo_xgb_tune_test)

precision2 <- c(precision_modelo_xgb_tune, precision_modelo_xgb_tune_test)

F1Score2 <- c(F1Score_modelo_xgb_tune, F1Score_modelo_xgb_tune_test)

ROC2 <- c(auc(roc_xgb), auc(roc_xgb_test))

datosreferencia <- c("train", "test")

modelo2 <- c("XGBoost", "XGBoost")
```

```r
datos2 <- data.frame(modelo = modelo2, datos = datosreferencia,
    accuracy = accuracy2, recall = recall2, precision = precision2,
    F1Score = F1Score2, ROC = ROC2)

head(datos2)
```

```
##    modelo datos accuracy recall precision F1Score       ROC
## 1 XGBoost train   0.9408 0.7489    0.9742  0.8468 0.9653826
## 2 XGBoost  test   0.9329 0.7311    0.9477  0.8254 0.9491626
```

```r
# XG BOOST TRAIN
pred6 <- predict(caret.xgb, newdata = fcr_train_XGBoost, type = "prob")
pred66 <- ifelse(pred6$`1` > 0.5, pred6$`1`, 1 - pred6$`0`)
pred.xgb <- prediction(pred66, fcr_train_XGBoost$loan_status)
perf.xgb <- performance(pred.xgb, "tpr", "fpr")
plot(perf.xgb, col = "black", lwd = 2)

# XG BOOST TEST
pred7 <- predict(caret.xgb, newdata = fcr_test, type = "prob")
pred77 <- ifelse(pred7$`1` > 0.5, pred7$`1`, 1 - pred7$`0`)
pred.xgb2 <- prediction(pred77, fcr_test$loan_status)
perf.xgb2 <- performance(pred.xgb2, "tpr", "fpr")
plot(perf.xgb2, add = TRUE, col = "grey", lwd = 2)
```



Realizando la comprobación del modelo con los datos de test al final del análisis, se llega a la conclusión

de que es un modelo generalizable, robusto y con resultados relativamente similares tanto en train como en test, siendo esto un resultado final del trabajo positivo.