Credit card defaults prediction

Andrea

21 June 2020

Contents

Introduction

In this project, using a dataset containing six months information, we want to create a model that predicts credit card defaults. From a risk management perspective, an accurate prediction and assessment of the credit risk is of relevant importance for the bank issuing the credit card. The dataset we are going to use can be downloaded at the following url:

"https://archive.ics.uci.edu/ml/machine-learning-databases/00350/default%20of%20credit%20card%20clients.xls".

We have downloaded and manually saved a version of the *.xlsx* file in the "data" subfolder you can find within the github repo link we have provided together with the three files or at the following ulr:

"https://github.com/andrearoetti/Choose-Your-Own/tree/master/data".

Before getting started, we load the necessary packages for our work, after installing them in case they are not installed yet.

```
if (!require(tidyverse)) install.packages('tidyverse')
if (!require(lattice)) install.packages('lattice')
if (!require(caret)) install.packages('caret')
if (!require(readxl)) install.packages('readxl')
library(tidyverse)
library(lattice)
library(caret)
library(readxl)
```

We therefore use here below a relative path in order to read the excel file in R: for the sake of simplicity, we will simply call the dataset **credit_card**.

```
credit_card <- read_xlsx("data/default of credit card clients.xlsx")
```

Data cleaning and exploratory analysis

We first take a look of the first six rows to get a flavour of how the dataset is made.

```
head(credit_card)
```

```
## # A tibble: 6 x 25
##   ...1  X1     X2    X3    X4    X5    X6    X7    X8    X9    X10   X11   X12
##   <chr> <chr>  <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 ID    LIMIT~ SEX   EDUC~ MARR~ AGE   PAY_0 PAY_2 PAY_3 PAY_4 PAY_5 PAY_6 BILL~
## 2 1     20000  2     2     1     24    2     2     -1    -1    -2    -2    3913
## 3 2     120000 2     2     2     26    -1    2     0     0     0     2     2682
## 4 3     90000  2     2     2     34    0     0     0     0     0     0     29239
## 5 4     50000  2     2     1     37    0     0     0     0     0     0     46990
## 6 5     50000  1     2     1     57    -1    0     -1    0     0     0     8617
## # ... with 12 more variables: X13 <chr>, X14 <chr>, X15 <chr>, X16 <chr>,
## #   X17 <chr>, X18 <chr>, X19 <chr>, X20 <chr>, X21 <chr>, X22 <chr>,
## #   X23 <chr>, Y <chr>
```

We immediately notice that