# Credit Card Defaults

Gergely Fenyvesi

4/19/2020

## I. Introduction

In the following study I will try to predict credit card defaults using a public dataset available on the Kaggle site here. This dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005. The original dataset was first presented on the UCI Machine Learning Repository here.

The dataset contains 25 variables:

- ID: ID of each client
- LIMIT_BAL: Amount of given credit in NT dollars (includes individual and family/supplementary credit
- SEX: Gender (1=male, 2=female)
- EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
- MARRIAGE: Marital status (1=married, 2=single, 3=others)
- AGE: Age in years
- PAY_0: Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, ... 8=payment delay for eight months, 9=payment delay for nine months and above)
- PAY_2: Repayment status in August, 2005 (scale same as above)
- PAY_3: Repayment status in July, 2005 (scale same as above)
- PAY_4: Repayment status in June, 2005 (scale same as above)
- PAY_5: Repayment status in May, 2005 (scale same as above)
- PAY_6: Repayment status in April, 2005 (scale same as above)
- BILL_AMT1: Amount of bill statement in September, 2005 (NT dollar)
- BILL_AMT2: Amount of bill statement in August, 2005 (NT dollar)
- BILL_AMT3: Amount of bill statement in July, 2005 (NT dollar)
- BILL_AMT4: Amount of bill statement in June, 2005 (NT dollar)
- BILL_AMT5: Amount of bill statement in May, 2005 (NT dollar)
- BILL_AMT6: Amount of bill statement in April, 2005 (NT dollar)
- PAY_AMT1: Amount of previous payment in September, 2005 (NT dollar)
- PAY_AMT2: Amount of previous payment in August, 2005 (NT dollar)
- PAY_AMT3: Amount of previous payment in July, 2005 (NT dollar)
- PAY_AMT4: Amount of previous payment in June, 2005 (NT dollar)
- PAY_AMT5: Amount of previous payment in May, 2005 (NT dollar)
- PAY_AMT6: Amount of previous payment in April, 2005 (NT dollar)
- default.payment.next.month: Default payment (1=yes, 0=no)

First I will partition the dataset into a training and test set, using 90% of the data for training purposes. We need to find the best algorithm to predict the default.payment.next.month value of the test set using the original or transformed values of the other columns. As the default is a binary event (1=yes, 0=no) the accuracy or the balanced accuracy can be used to evaluate our algorithm. We do not strictly need to use a loss function like the Residual Mean Squared Error (RMSE).

## II. Methods

First I explored the whole dataset to identify meaningful and meaningless columns in determining the default in next month. I worked with the whole dataset. Then using a 90-10 rule I partitioned the dataset into a training and a test set. I chose in an arbitrary manner 3 algorithms to work with:

1. Decision / Regression tree (rpart): the choice was made to identify the 3 most meaningful columns from my pre-selection and use them later
2. KNN: this is my personal favorite and I think that this algorithm is particularly useful in our case
3. Randomforest (rf): this is an advanced version of the regression tree, averaging regression tree values

To benchmark my selection, I executed first the 3 methods on all the columns of the dataset, then compared the obtained results with numbers obtained using the 3 most meaningful columns. With my trained models I created an ensemble. Finally to identify other models that can be added in the ensemble to further enhance results and benchmark again obtained numbers using these 3 models, I created a more general ensemble, using 8 models with the default parameters.

# III. Results

## 1. Exploring the dataset

The Credit Card Defaults dataset (data) contains 30000 client data from which 6636 clients defaulted in Sept 2005, resulting in a global default proportion of 0.2212. The number of distinct client IDs is 30000 hence one client cannot be listed twice. The structure of the dataset is the following:

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 30000 obs. of  25 variables:
##  $ ID                        : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ LIMIT_BAL                 : num  20000 120000 90000 50000 50000 50000 500000 100000 140000 20000
##  $ SEX                       : num  2 2 2 2 1 1 1 2 2 1 ...
##  $ EDUCATION                 : num  2 2 2 2 2 1 1 2 3 3 ...
##  $ MARRIAGE                  : num  1 2 2 1 1 2 2 2 1 2 ...
##  $ AGE                       : num  24 26 34 37 57 37 29 23 28 35 ...
##  $ PAY_0                     : num  2 -1 0 0 -1 0 0 0 0 -2 ...
##  $ PAY_2                     : num  2 2 0 0 0 0 0 -1 0 -2 ...
##  $ PAY_3                     : num  -1 0 0 0 -1 0 0 -1 2 -2 ...
##  $ PAY_4                     : num  -1 0 0 0 0 0 0 0 0 -2 ...
##  $ PAY_5                     : num  -2 0 0 0 0 0 0 0 0 -1 ...
##  $ PAY_6                     : num  -2 2 0 0 0 0 0 -1 0 -1 ...
##  $ BILL_AMT1                 : num  3913 2682 29239 46990 8617 ...
##  $ BILL_AMT2                 : num  3102 1725 14027 48233 5670 ...
##  $ BILL_AMT3                 : num  689 2682 13559 49291 35835 ...
##  $ BILL_AMT4                 : num  0 3272 14331 28314 20940 ...
##  $ BILL_AMT5                 : num  0 3455 14948 28959 19146 ...
##  $ BILL_AMT6                 : num  0 3261 15549 29547 19131 ...
##  $ PAY_AMT1                  : num  0 0 1518 2000 2000 ...
##  $ PAY_AMT2                  : num  689 1000 1500 2019 36681 ...
##  $ PAY_AMT3                  : num  0 1000 1000 1200 10000 657 38000 0 432 0 ...
##  $ PAY_AMT4                  : num  0 1000 1000 1100 9000 ...
##  $ PAY_AMT5                  : num  0 0 1000 1069 689 ...
##  $ PAY_AMT6                  : num  0 2000 5000 1000 679 ...
##  $ default.payment.next.month: num  1 1 0 0 0 0 0 0 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ID = col_double(),
##   ..   LIMIT_BAL = col_double(),
##   ..   SEX = col_double(),
##   ..   EDUCATION = col_double(),
##   ..   MARRIAGE = col_double(),
##   ..   AGE = col_double(),
##   ..   PAY_0 = col_double(),
##   ..   PAY_2 = col_double(),
##   ..   PAY_3 = col_double(),
##   ..   PAY_4 = col_double(),
##   ..   PAY_5 = col_double(),
##   ..   PAY_6 = col_double(),
##   ..   BILL_AMT1 = col_double(),
##   ..   BILL_AMT2 = col_double(),
##   ..   BILL_AMT3 = col_double(),
##   ..   BILL_AMT4 = col_double(),
##   ..   BILL_AMT5 = col_double(),
##   ..   BILL_AMT6 = col_double(),
##   ..   PAY_AMT1 = col_double(),
```

```
##   ..   PAY_AMT2 = col_double(),
##   ..   PAY_AMT3 = col_double(),
##   ..   PAY_AMT4 = col_double(),
##   ..   PAY_AMT5 = col_double(),
##   ..   PAY_AMT6 = col_double(),
##   ..   default.payment.next.month = col_double()
##   .. )
```

We can observe the summary statistics of each data column:

```
##        ID           LIMIT_BAL            SEX          EDUCATION
##  Min.   :    1   Min.   :  10000   Min.   :1.000   Min.   :0.000
##  1st Qu.: 7501   1st Qu.:  50000   1st Qu.:1.000   1st Qu.:1.000
##  Median :15000   Median : 140000   Median :2.000   Median :2.000
##  Mean   :15000   Mean   : 167484   Mean   :1.604   Mean   :1.853
##  3rd Qu.:22500   3rd Qu.: 240000   3rd Qu.:2.000   3rd Qu.:2.000
##  Max.   :30000   Max.   :1000000   Max.   :2.000   Max.   :6.000
##     MARRIAGE          AGE             PAY_0             PAY_2
##  Min.   :0.000   Min.   :21.00   Min.   :-2.0000   Min.   :-2.0000
##  1st Qu.:1.000   1st Qu.:28.00   1st Qu.:-1.0000   1st Qu.:-1.0000
##  Median :2.000   Median :34.00   Median : 0.0000   Median : 0.0000
##  Mean   :1.552   Mean   :35.49   Mean   :-0.0167   Mean   :-0.1338
##  3rd Qu.:2.000   3rd Qu.:41.00   3rd Qu.: 0.0000   3rd Qu.: 0.0000
##  Max.   :3.000   Max.   :79.00   Max.   : 8.0000   Max.   : 8.0000
##      PAY_3             PAY_4             PAY_5             PAY_6
##  Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000
##  1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000
##  Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.0000
##  Mean   :-0.1662   Mean   :-0.2207   Mean   :-0.2662   Mean   :-0.2911
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000
##  Max.   : 8.0000   Max.   : 8.0000   Max.   : 8.0000   Max.   : 8.0000
##    BILL_AMT1         BILL_AMT2         BILL_AMT3         BILL_AMT4
##  Min.   :-165580   Min.   :-69777   Min.   :-157264   Min.   :-170000
##  1st Qu.:   3559   1st Qu.:  2985   1st Qu.:   2666   1st Qu.:   2327
##  Median :  22382   Median : 21200   Median :  20088   Median :  19052
##  Mean   :  51223   Mean   : 49179   Mean   :  47013   Mean   :  43263
##  3rd Qu.:  67091   3rd Qu.: 64006   3rd Qu.:  60165   3rd Qu.:  54506
##  Max.   : 964511   Max.   :983931   Max.   :1664089   Max.   : 891586
##    BILL_AMT5         BILL_AMT6          PAY_AMT1          PAY_AMT2
##  Min.   :-81334   Min.   :-339603   Min.   :     0   Min.   :      0
##  1st Qu.:  1763   1st Qu.:   1256   1st Qu.:  1000   1st Qu.:    833
##  Median : 18104   Median :  17071   Median :  2100   Median :   2009
##  Mean   : 40311   Mean   :  38872   Mean   :  5664   Mean   :   5921
##  3rd Qu.: 50190   3rd Qu.:  49198   3rd Qu.:  5006   3rd Qu.:   5000
##  Max.   :927171   Max.   : 961664   Max.   :873552   Max.   :1684259
##     PAY_AMT3          PAY_AMT4          PAY_AMT5          PAY_AMT6
##  Min.   :     0   Min.   :     0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:   390   1st Qu.:   296   1st Qu.:   252.5   1st Qu.:   117.8
##  Median :  1800   Median :  1500   Median :  1500.0   Median :  1500.0
##  Mean   :  5226   Mean   :  4826   Mean   :  4799.4   Mean   :  5215.5
##  3rd Qu.:  4505   3rd Qu.:  4013   3rd Qu.:  4031.5   3rd Qu.:  4000.0
##  Max.   :896040   Max.   :621000   Max.   :426529.0   Max.   :528666.0
##  default.payment.next.month
##  Min.   :0.0000
##  1st Qu.:0.0000
```

```
##  Median :0.0000
##  Mean   :0.2212
##  3rd Qu.:0.0000
##  Max.   :1.0000
```

I examined gender, age, education and payment information of the population, trying to find signals of default.

### A. Gender

Proportionally we have more defaults in the case of men, but this information need to be trreated with caution as female prevalence is higher in the dataset:

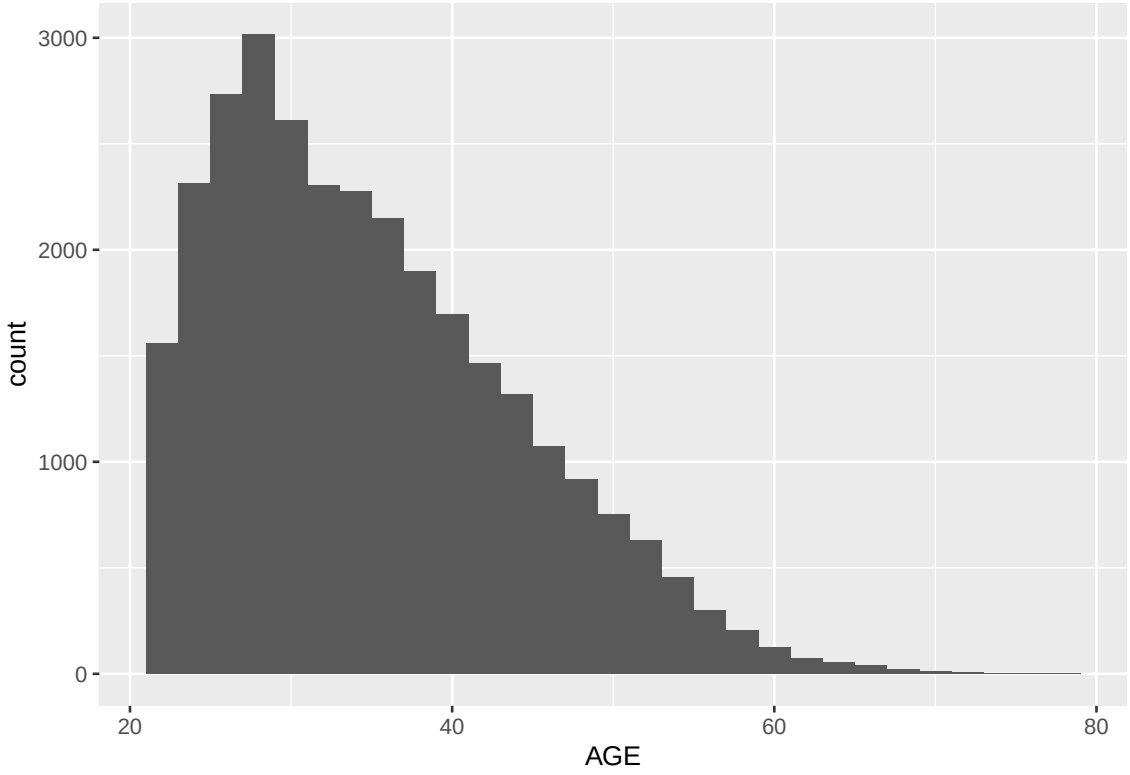
| . | Freq |
|---|---|
| Female | 18112 |
| Male | 11888 |

| | In default | No default |
|---|---|---|
| Female | 0.2077628 | 0.7922372 |
| Male | 0.2416723 | 0.7583277 |

The difference is not sigbificant enough to be qualified as a signal of more probable default.

### B. Age

Credit card debt is the most common between young, around 30 year old clients in our population data.



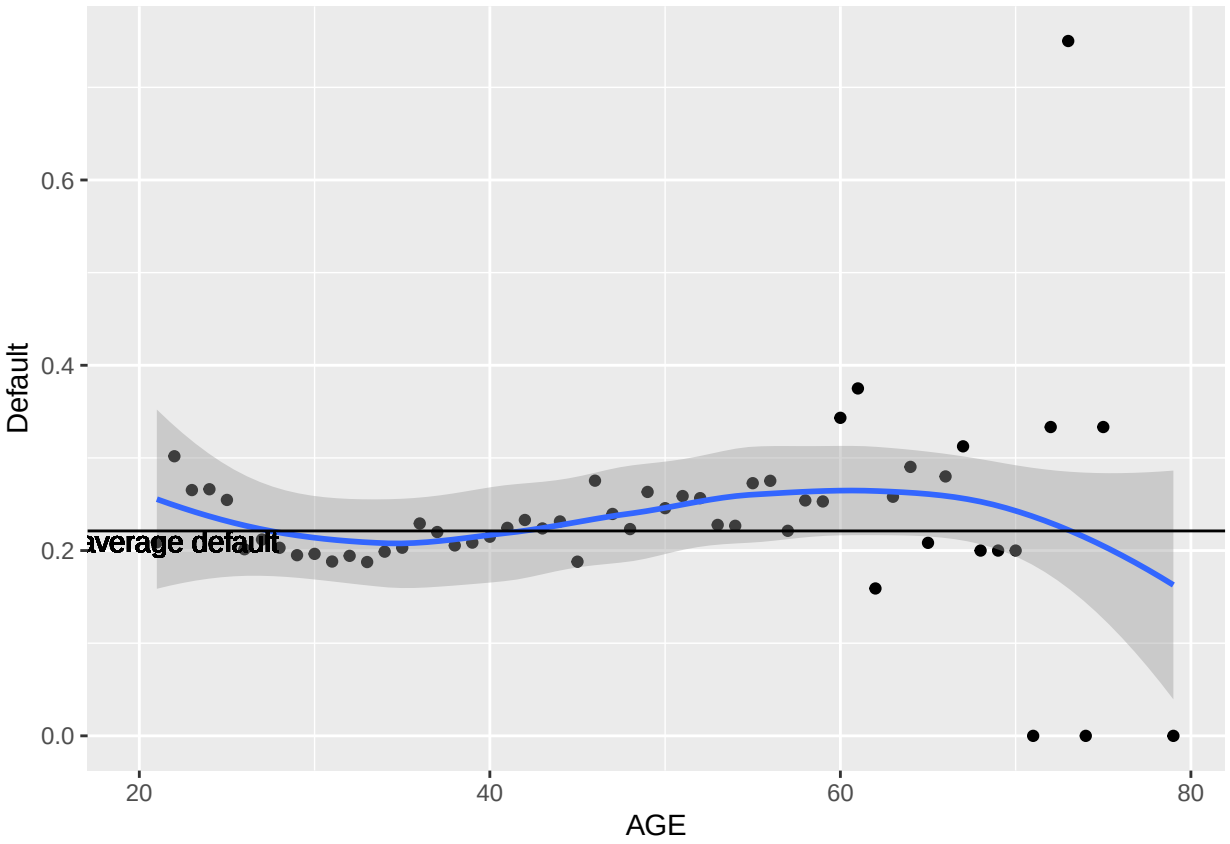
| AGE | n |
|---|---|
| 29 | 1605 |
| 27 | 1477 |
| 28 | 1409 |
| 30 | 1395 |
| 26 | 1256 |
| 31 | 1217 |
| 25 | 1186 |
| 34 | 1162 |
| 32 | 1158 |
| 33 | 1146 |

Basic statistics show a positive skew and a kurtosis close to normal. For information in the case of a standard normal distribution the mean equals 0, standard deviation is unit (1), skewness=0, kurtosis=3 amd mean=mode=median. For a positive skew we typically see that: mode < median < mean.

| AGE |
|---|
| Min. :21.00 |
| 1st Qu.:28.00 |
| Median :34.00 |
| Mean :35.49 |
| 3rd Qu.:41.00 |
| Max. :79.00 |

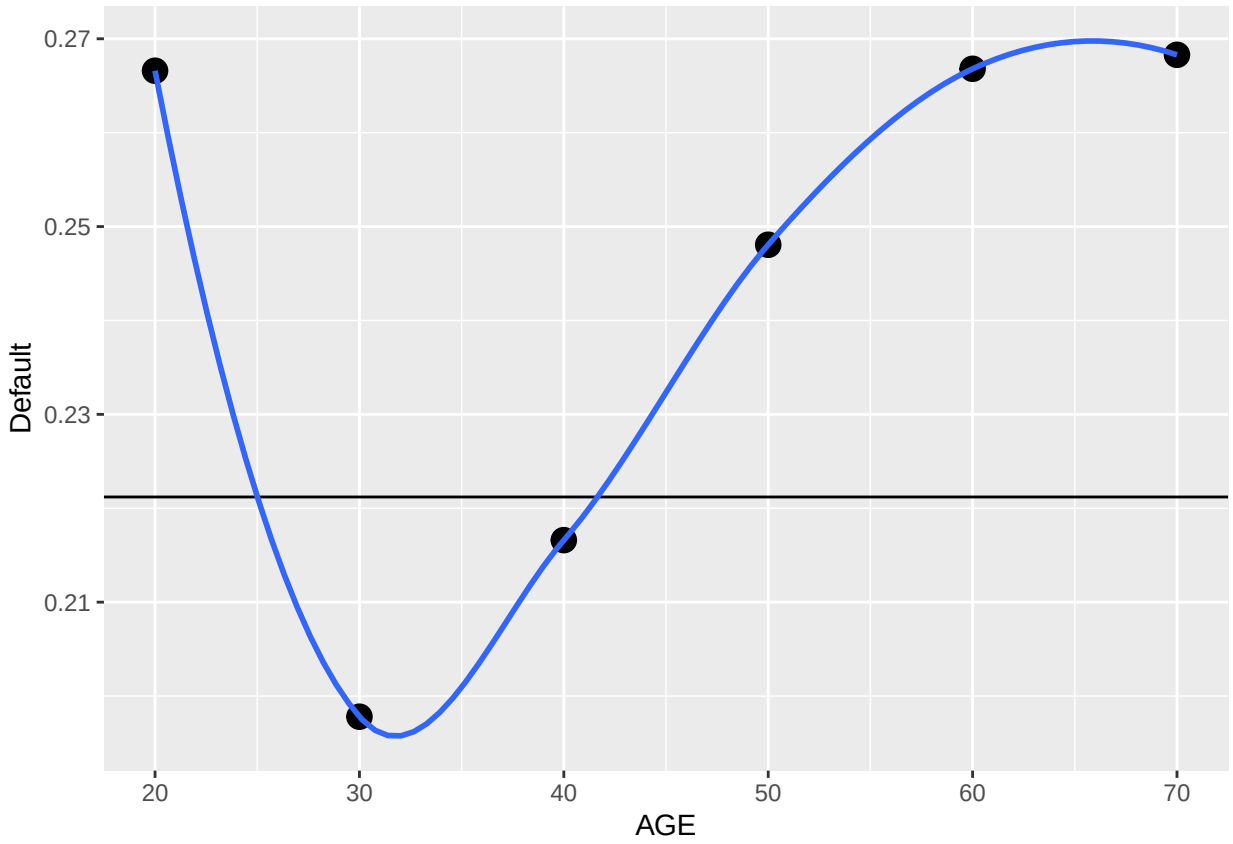
| AVG | STDEV | SKEW | KURTOSIS | MODE |
|---|---|---|---|---|
| 35.4855 | 9.217904 | 0.7322093 | 3.044096 | 29 |

When we try to visualise the default rate against the age of the clients we can see that the points are not randomly distributed around the mean. Defaults are under average around 30 and above average for younger and older generations.




If we create age groups this connection between age and default rates is even more pronounced. However it is to treat with caution as the different age groups have different number of participants. It is common to observe a regression to the mean with higher number of observations.

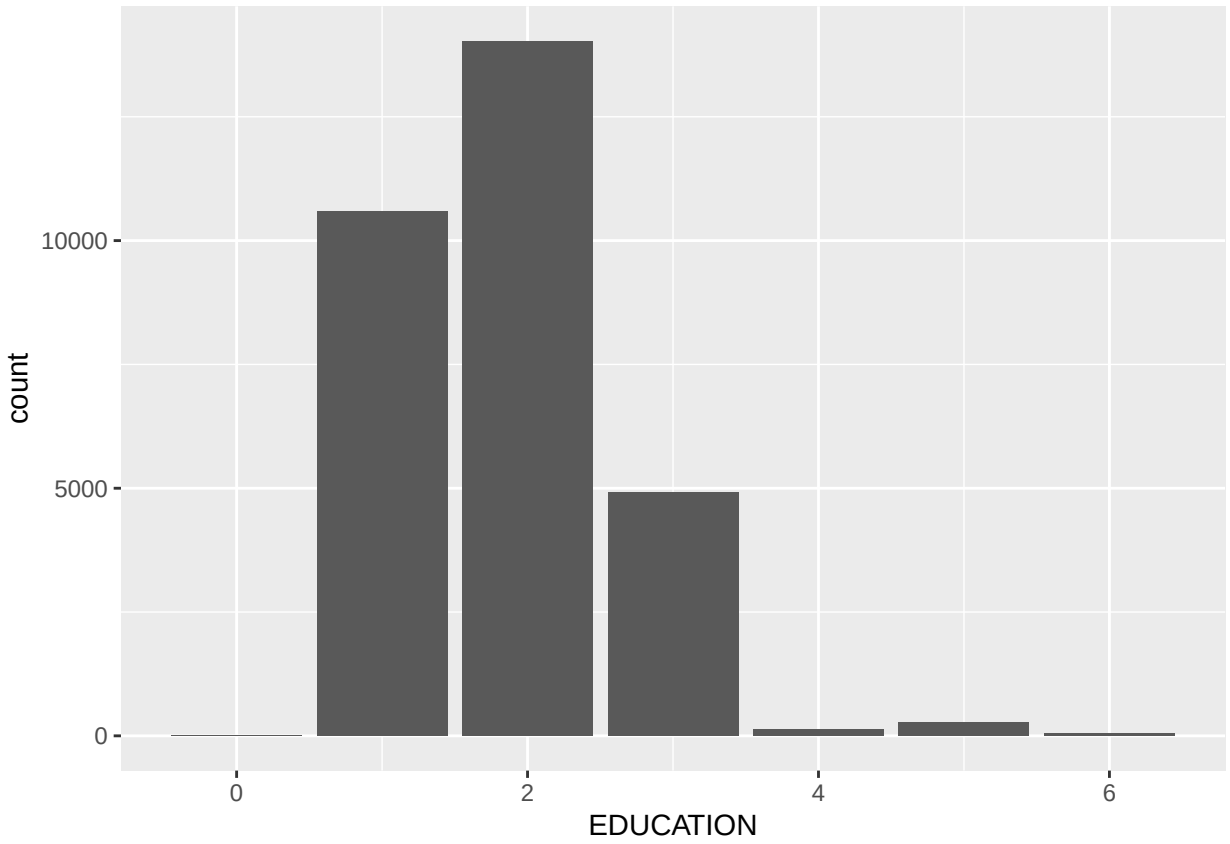
| AGE | all | default | proportion |
|---|---|---|---|
| 30 | 11825 | 2339 | 0.1978013 |
| 40 | 9635 | 2087 | 0.2166061 |
| 50 | 3616 | 897 | 0.2480642 |
| 20 | 3871 | 1032 | 0.2665978 |
| 60 | 967 | 258 | 0.2668046 |
| 70 | 82 | 22 | 0.2682927 |




**C. Education**

Most clients are graduates and undergraduates (1 and 2), and a smaller proportion earned a high school degree. Other education is marginal and can be excluded.

| . | Freq |
|---|---|
| 0 | 14 |
| 1 | 10585 |
| 2 | 14030 |
| 3 | 4917 |
| 4 | 123 |
| 5 | 280 |
| 6 | 51 |




Defaults are lower for higher education clients.

| EDUCATION | all | default | proportion |
|---|---|---|---|
| 1 | 10585 | 2036 | 0.1923477 |
| 2 | 14030 | 3330 | 0.2373485 |
| 3 | 4917 | 1237 | 0.2515762 |

**D. Marital status**

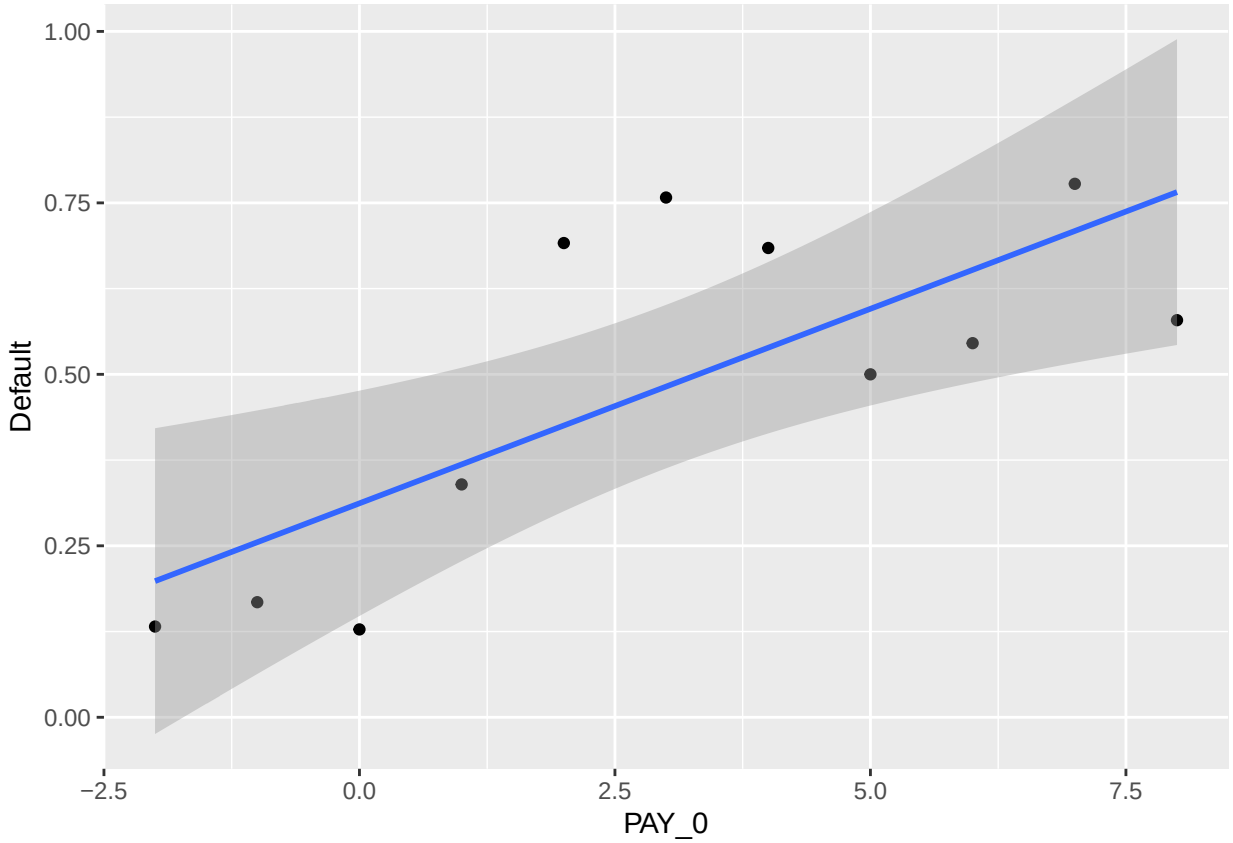
Single persons seems to make less frequently default.

| MARRIAGE | all | default | proportion |
|---|---|---|---|
| 0 | 54 | 5 | 0.0925926 |
| 2 | 15964 | 3341 | 0.2092834 |
| 1 | 13659 | 3206 | 0.2347170 |
| 3 | 323 | 84 | 0.2600619 |

**E. Repayment status in September, 2005 (PAY_0)**

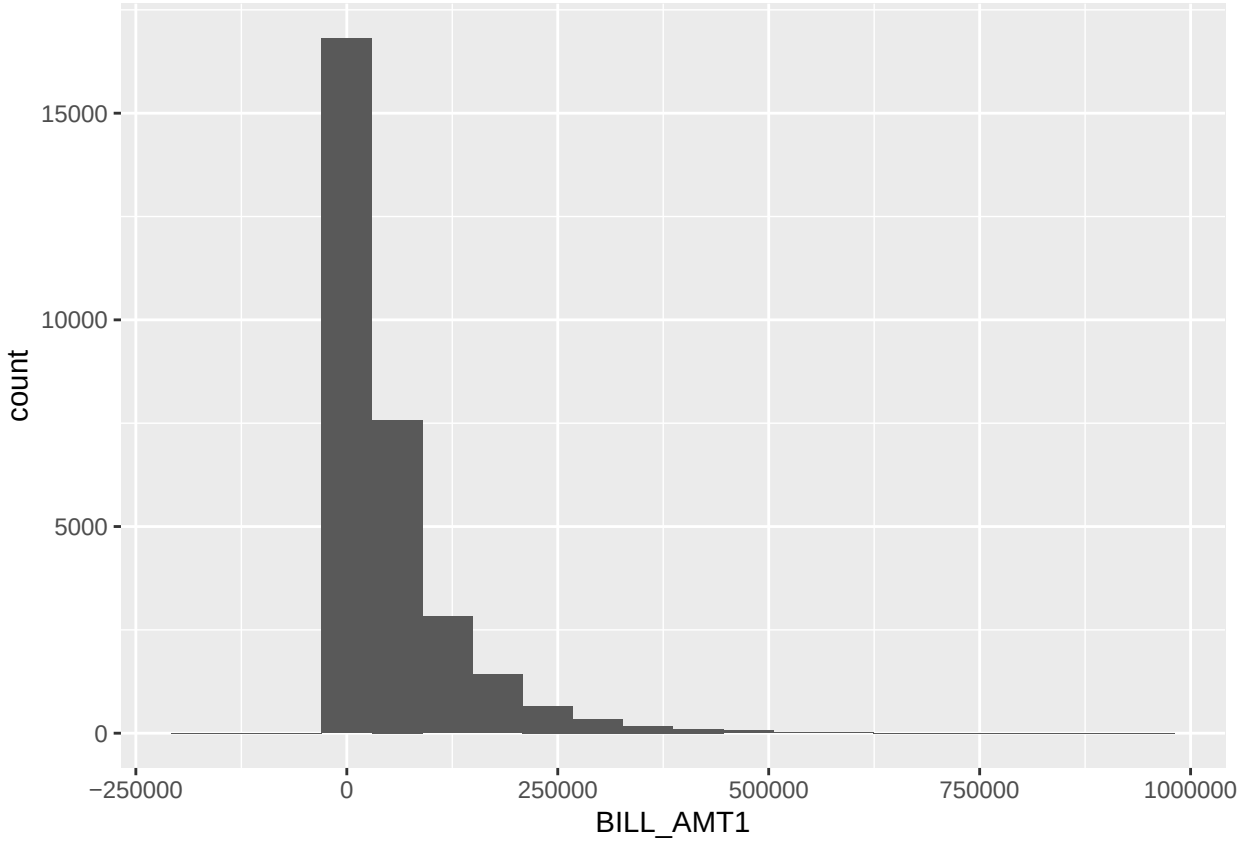
We can observe a positive relationship between late payments and defaults. The higher the number, the higher the delay of payment of credit card debt.

| PAY_0 | all | default | proportion |
|---|---|---|---|
| 0 | 14737 | 1888 | 0.1281129 |
| -2 | 2759 | 365 | 0.1322943 |
| -1 | 5686 | 954 | 0.1677805 |
| 1 | 3688 | 1252 | 0.3394794 |
| 5 | 26 | 13 | 0.5000000 |
| 6 | 11 | 6 | 0.5454545 |
| 8 | 19 | 11 | 0.5789474 |
| 4 | 76 | 52 | 0.6842105 |
| 2 | 2667 | 1844 | 0.6914136 |
| 3 | 322 | 244 | 0.7577640 |
| 7 | 9 | 7 | 0.7777778 |

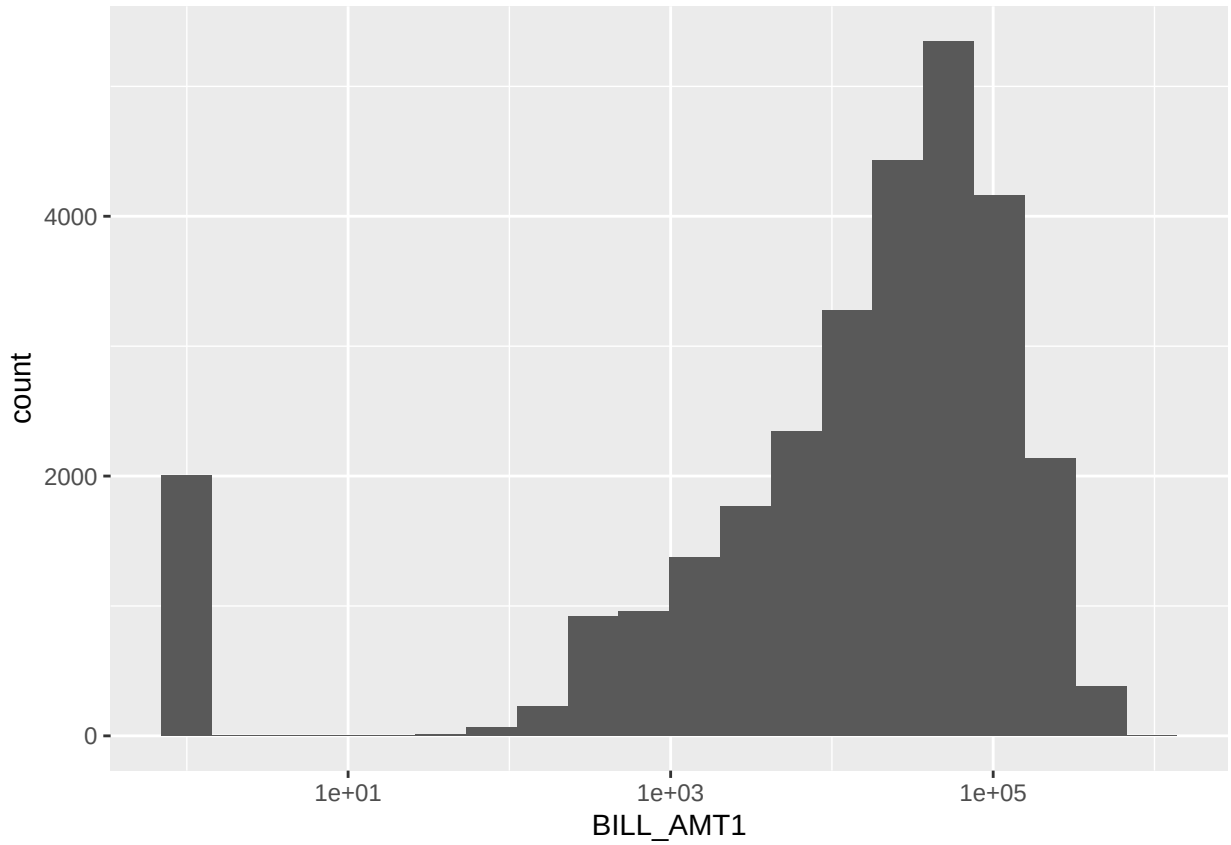


**F. Amount of bill statement in September, 2005 (BILL_AMT1)**

Most people have a low bill amount, but we cannot clearly see details on a simple linear x-axis.




We could be tempted to use a log transformation of the x-axis, but in this case we will lose 0 values (log10 of 0 = -Inf). The correct way is to transorm 0 amounts to 1 before the log transformation.



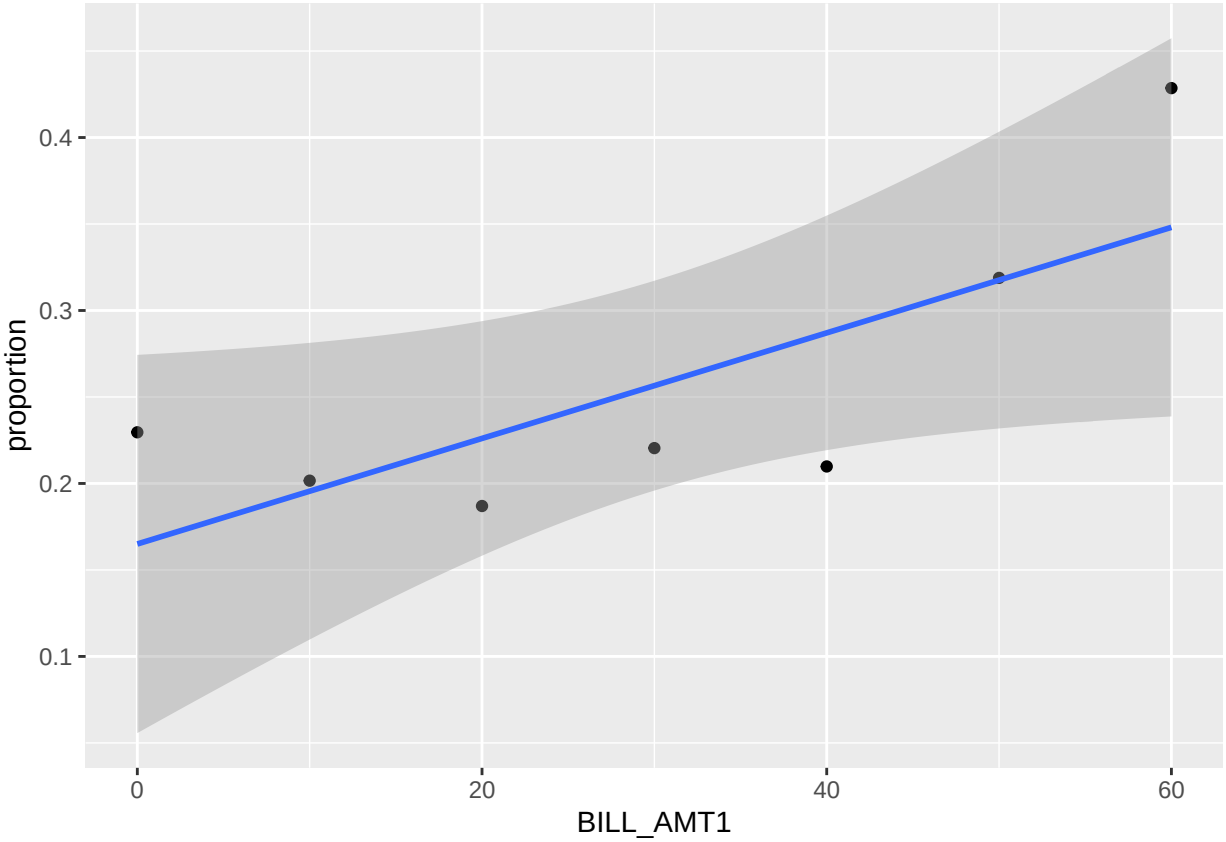

To show whether there is a relationship between bill amount and default probability I created two groups, higher and lower then the median value and examined default probability. With only two groups we see no evidence.

| | 0 | 1 |
|---|---|---|
| Less | 0.7703333 | 0.2296667 |
| More | 0.7872667 | 0.2127333 |

By rounding bill amounts to the nearest 10k NT dollars I created more groups. We see some evidence of relationship.

| BILL_AMT1 | all | default | proportion |
|---|---|---|---|
| 60 | 21 | 9 | 0.4285714 |
| 50 | 69 | 22 | 0.3188406 |
| 0 | 21274 | 4884 | 0.2295760 |
| 30 | 499 | 110 | 0.2204409 |
| 40 | 224 | 47 | 0.2098214 |
| 10 | 5823 | 1174 | 0.2016143 |
| 20 | 2086 | 390 | 0.1869607 |
| -20 | 2 | 0 | 0.0000000 |
| 80 | 1 | 0 | 0.0000000 |
| 100 | 1 | 0 | 0.0000000 |

By excluding classes with too few observations then plotting them we can see that higher amount means higher risk of default.

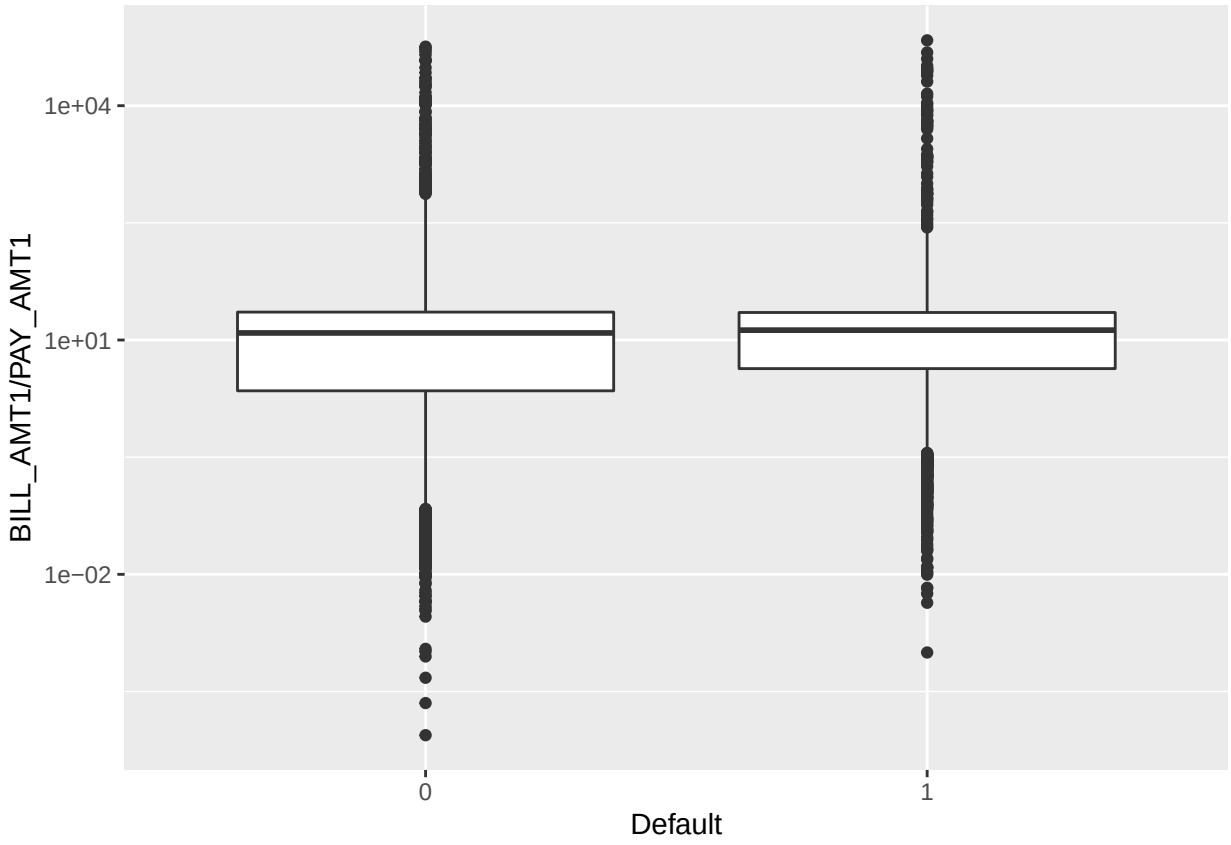



**G. Amount of previous payment in September, 2005 (PAY_AMT1)**

Instead of examining the payment amount I checked its proportion to the bill amount (BILL_AMT1/PAY_AMT1). I filtered out where PAY_AMT1 equals to 0, then compared the proportion for defaults and no defaults. For defaults this proportion was significantly higher (> 2x). This could be treated as a signal but with caution as the standard deviation is huge compared to the average values.

| default.payment.next.month | avg | stdev |
|---|---|---|
| 0 | 59.98545 | 1113.151 |
| 1 | 135.95177 | 1844.633 |

Visually we cannot clearly distinguish the two boxes as one includes the other.

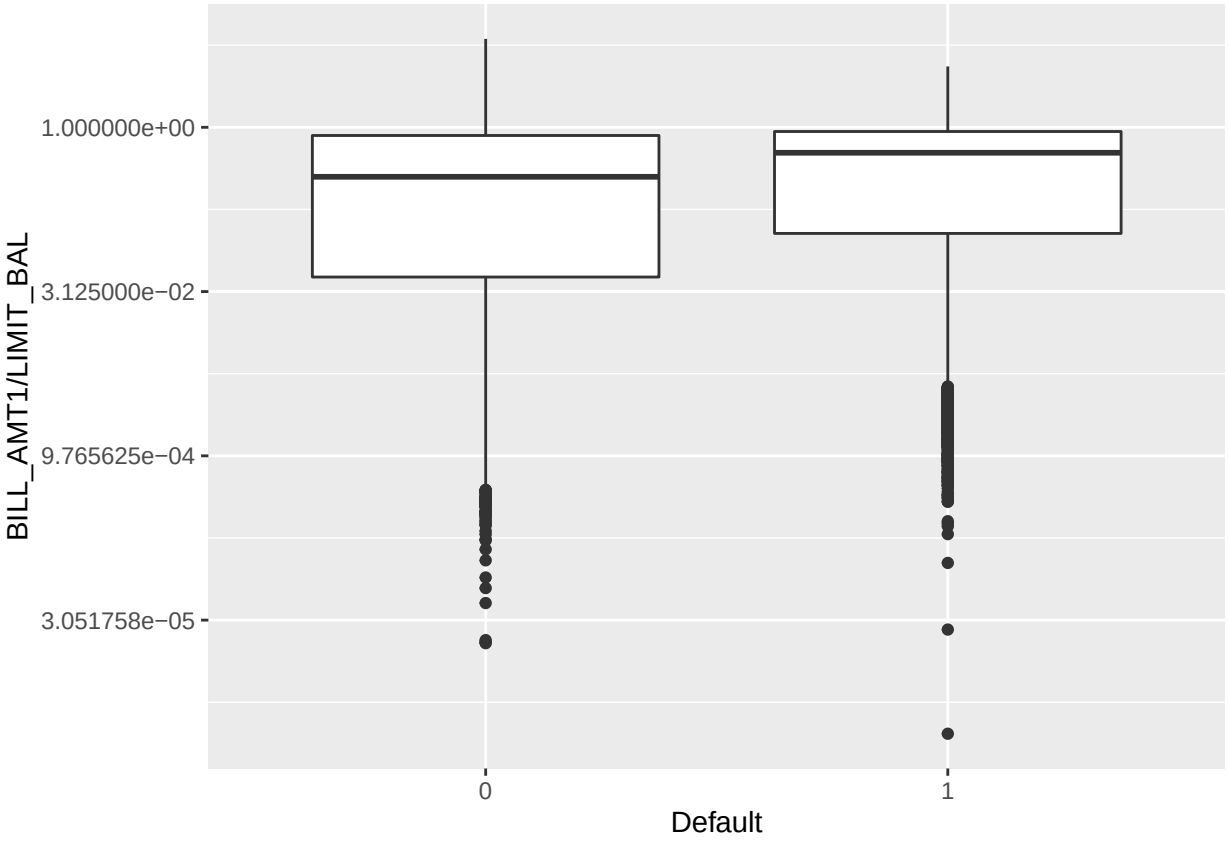


### H. Amount of given credit (LIMIT_BAL)

Again instead of checking the amount I checked the proportion to the bill amount (BILL_AMT1/LIMIT_BAL). Logically a bill amount closer to the limit means higher risk and higher default probability. In this case we don't need to filter out where LIMIT_BAL = 0 because simply it has no meaning. For defaults this proportion is a bit higher (+20%). This could be a signal, but it is to treat with caution as the standard deviation is significant compared to the average values.

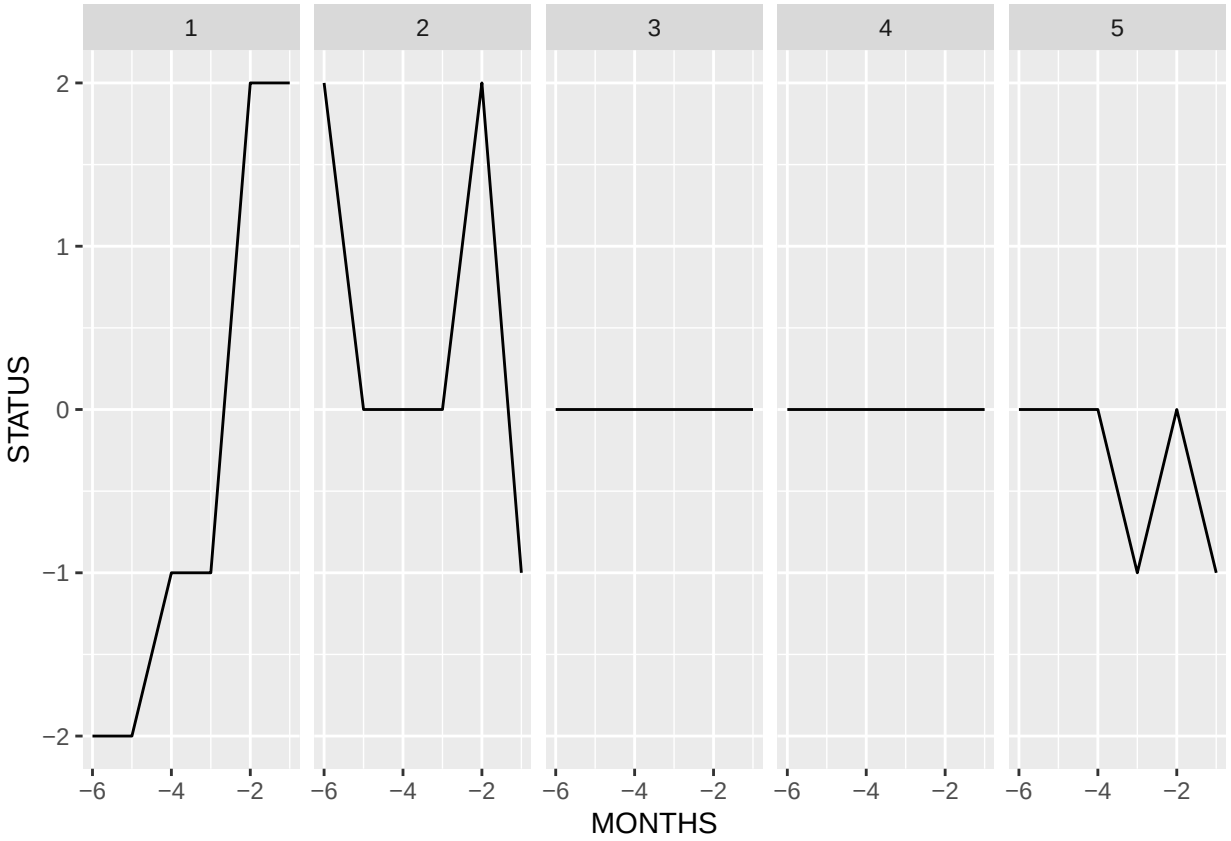
| default.payment.next.month | avg | stdev |
|---|---|---|
| 0 | 0.4048764 | 0.4082873 |
| 1 | 0.4902972 | 0.4157019 |

Visually again we cannot clearly distinguish the two boxes.



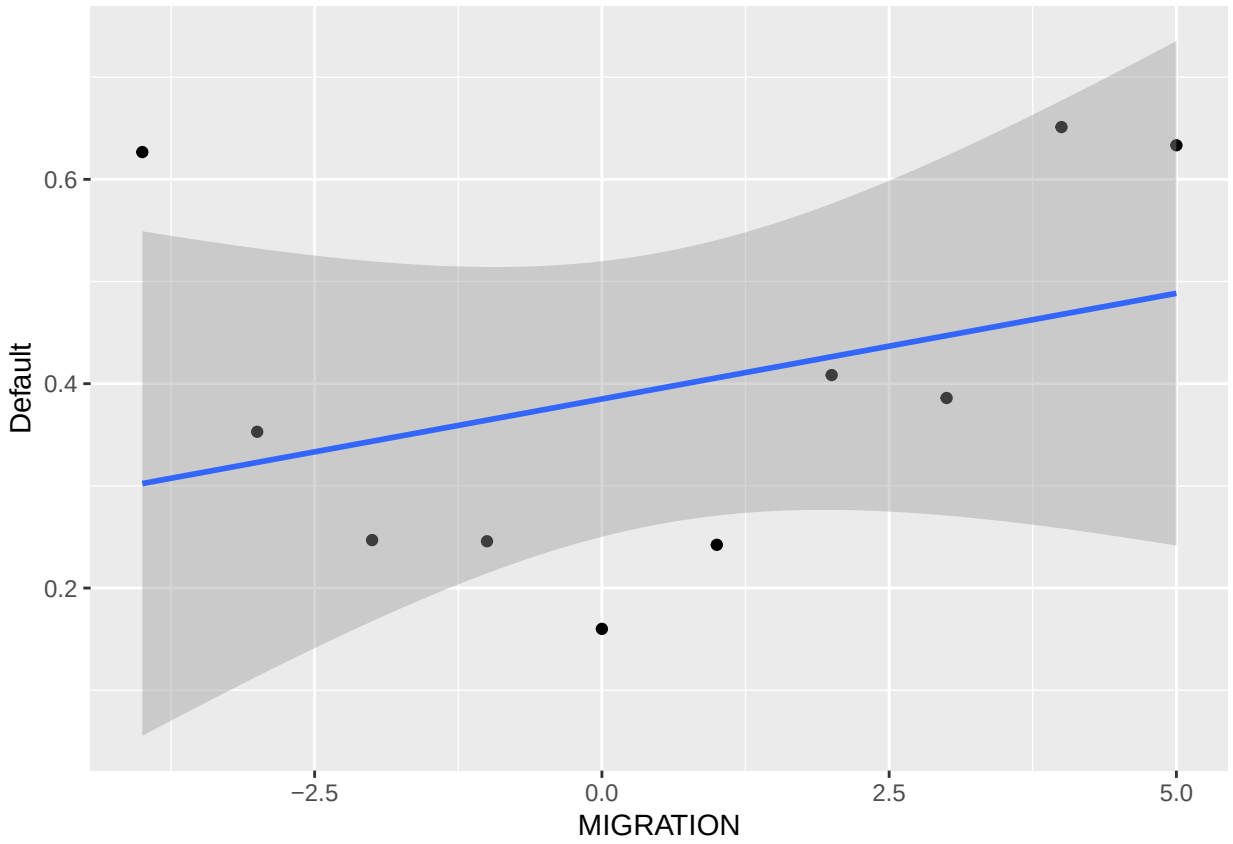
**I. Credit migration**

I had the idea to check the deterioration of credit quality over time, by exmining the difference between Sept, 2005 and April, 2005 payments. When we check the payments for the first 5 users we see that it is not stable over time: it can change in the two directions.



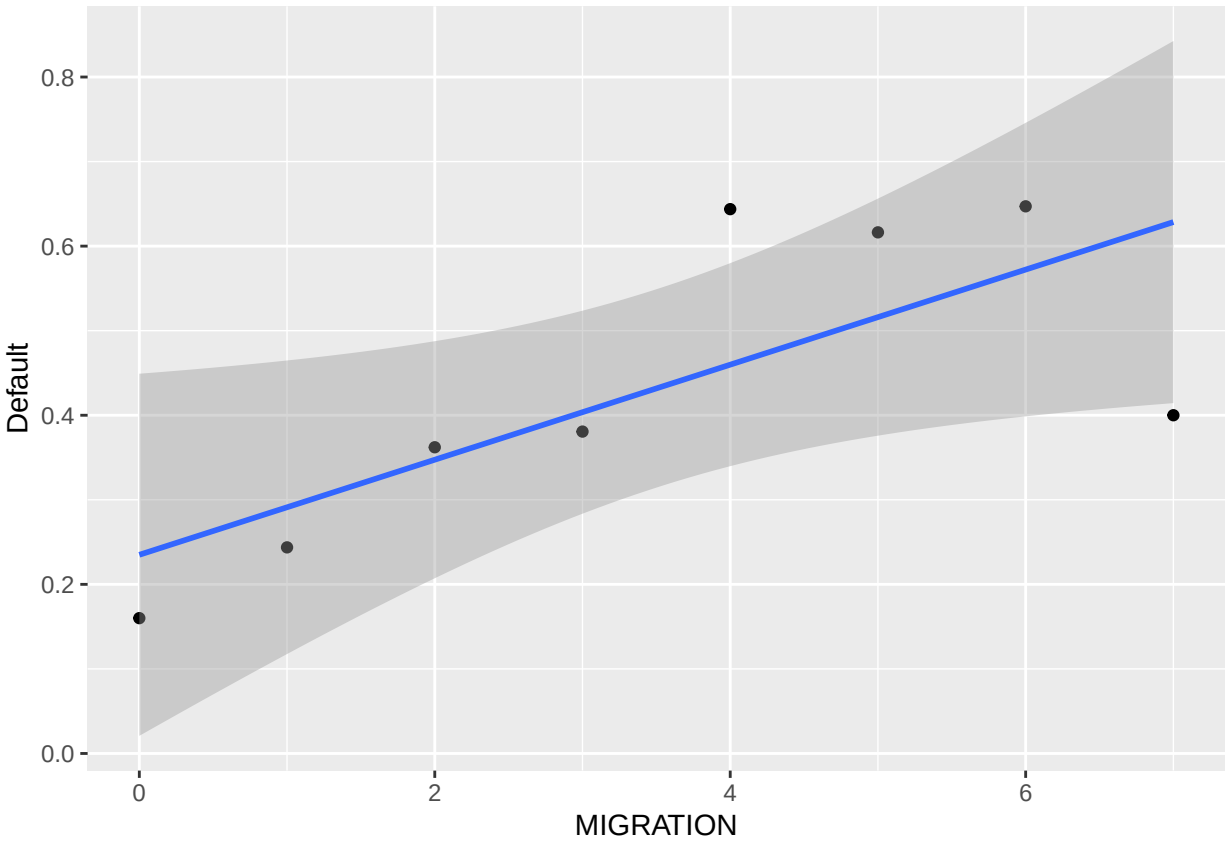

The proper solution would be to fit a line for each user and use the slope as a sign to determine whether the financial status became better or worse. For simplicity I took the delta between PAY_6 and PAY_0 (MIGRATION = PAY_0-PAY_6). I filtered out classes with too few observations, but evidence were still weak.

| MIGRATION | all | default | proportion |
|---|---|---|---|
| 0 | 18164 | 2907 | 0.1600418 |
| 1 | 3384 | 820 | 0.2423168 |
| -1 | 2368 | 582 | 0.2457770 |
| -2 | 1219 | 301 | 0.2469237 |
| -3 | 238 | 84 | 0.3529412 |
| 3 | 1241 | 479 | 0.3859790 |
| 2 | 3031 | 1238 | 0.4084461 |
| -4 | 75 | 47 | 0.6266667 |
| 5 | 60 | 38 | 0.6333333 |
| 4 | 172 | 112 | 0.6511628 |




Stable ratings (where MIGRATION is close to 0) seems to make less frequently default. I tried to apply abs as transformation then examnined and plotted results again.

| MIGRATION | all | default | proportion |
|---|---|---|---|
| 0 | 18164 | 2907 | 0.1600418 |
| 1 | 5752 | 1402 | 0.2437413 |
| 2 | 4250 | 1539 | 0.3621176 |
| 3 | 1479 | 563 | 0.3806626 |
| 7 | 5 | 2 | 0.4000000 |
| 5 | 86 | 53 | 0.6162791 |
| 4 | 247 | 159 | 0.6437247 |
| 6 | 17 | 11 | 0.6470588 |




The evidence was much stronger, the line fits better on the points. I created a transfomed column using the formula MIGRATION = abs(PAY_0-PAY_6).

## 2. Creation of training and test dataset

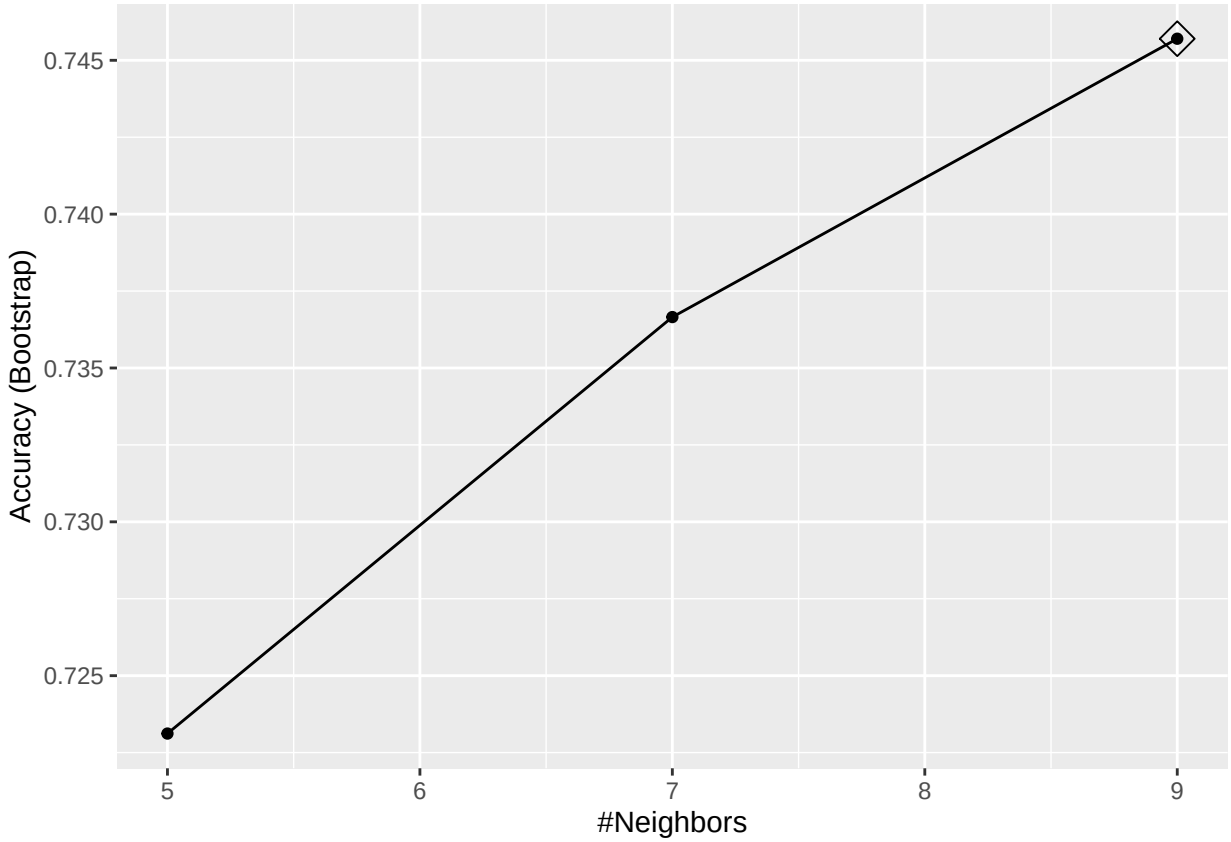
I partitioned the dataset into 2 groups randomly: train and test dataset, using 90% of the data for training purposes and 10% for testing. I used set seed to guarantee the same result at each execution.

```
################################
# division to train + test set
################################
set.seed(1, sample.kind="Rounding")
test_index <- createDataPartition(y = data$default.payment.next.month, times = 1, p = 0.1, list = FALSE)
train <- data[-test_index,]
test <- data[test_index,]
```

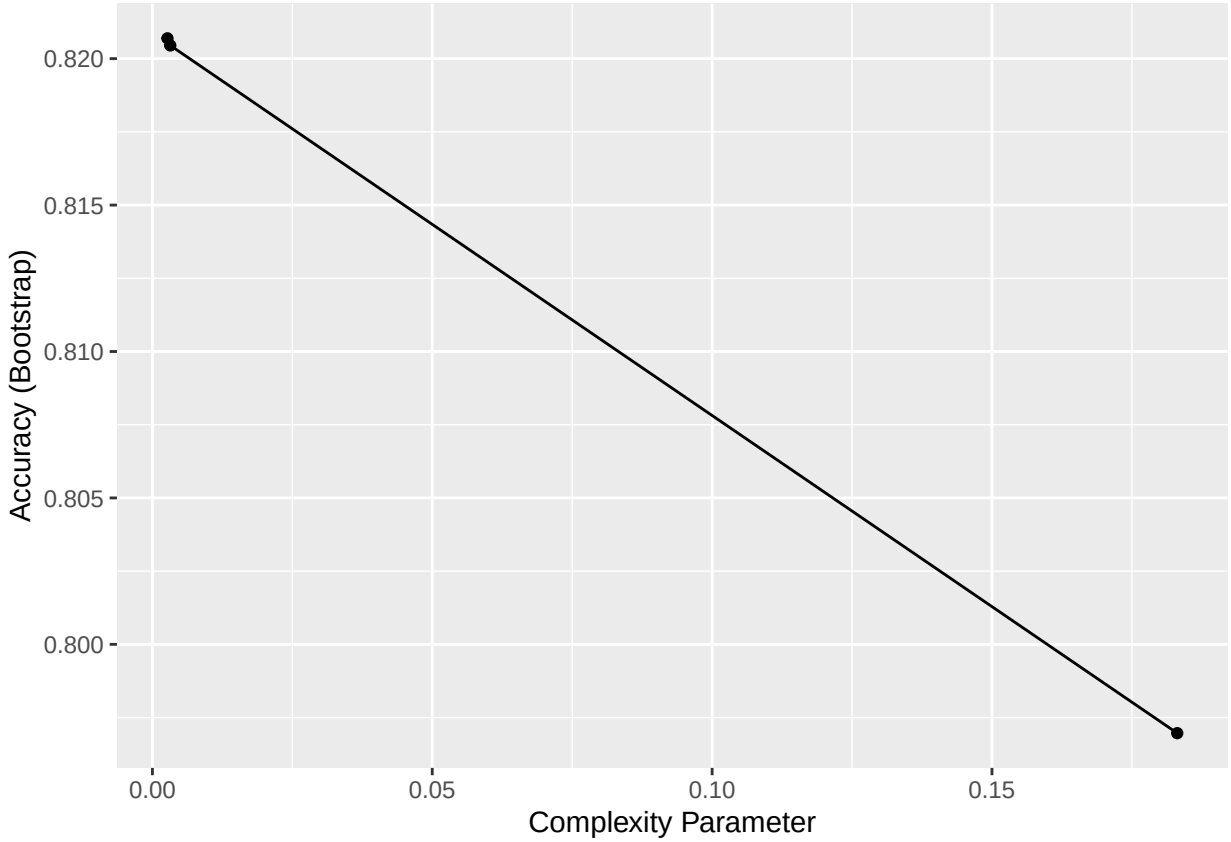
## 3. Test of the 3 algorithms using all columns

To benchmark results I started by guessing the default, then testing the 3 algorithms using all the columns of the dataset except the customer id. To avoid automatic selection of only a subset of columns by the regression tree, I executed the rpart training once with default complexity parameter and a second time with cp forced to 0. This parameter sets a minimum for how much the RSS must improve for another partition to be added and 0 value gives the highest flexibility and the maximum number of branches created.

With the default parameters R tunes the number of k to 9 using knn:

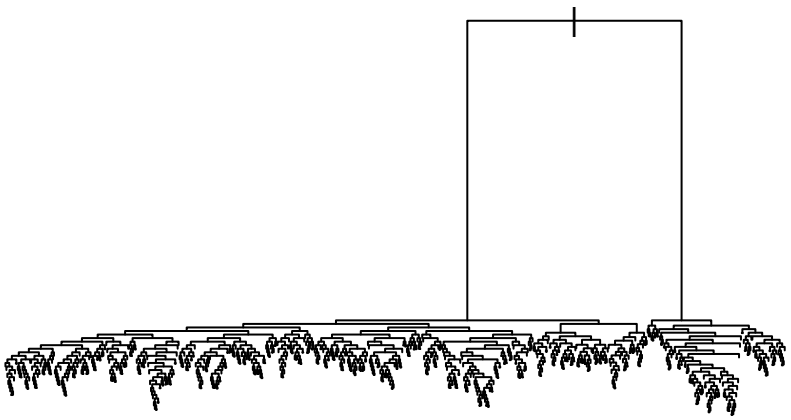


The default cp used is 0.00268772047707038:




We can visaulize the regression tree to see which columns were used:


It is not surprising to see on first place the repayment status of the last month. The PAY_3 variable on the second place is more just the consequence of overtraining, it has no more explanatory power then the PAY_0 column. By setting the cp to 0 we can force the usage of more (possibly all) columns in the data.



The variable importance can change but PAY_0 is still the most important variable:

| var | importance |
|---|---|
| PAY_0 | 1461.7717 |
| PAY_2 | 578.3466 |
| PAY_AMT3 | 545.0979 |
| PAY_3 | 543.1454 |
| LIMIT_BAL | 526.0349 |
| BILL_AMT1 | 478.8250 |
| BILL_AMT2 | 468.5834 |
| PAY_6 | 383.1489 |
| LIMIT_BAL | 283.9511 |
| PAY_AMT4 | 263.3690 |

The global results of the 3 algorithms are the following:

| method | Accuracy | Sensitivity | Specificity | Balanced_Accuracy |
|---|---|---|---|---|
| guess | 0.6516667 | 0.7665084 | 0.2620791 | 0.5142937 |
| knn | 0.7636667 | 0.9499353 | 0.1317716 | 0.5408534 |
| rpart | 0.8160000 | 0.9633146 | 0.3162518 | 0.6397832 |
| rpart cp=0 | 0.7596667 | 0.8739750 | 0.3718887 | 0.6229318 |
| rforest | 0.8156667 | 0.9438930 | 0.3806735 | 0.6622832 |

Accuracy is quite high, but we should not stop here. Specificity is especially poor in the case of knn. Other algorithms failed as well when we tried to predict defaults. The high accuracy is just the consequence of the high sensitivity. Prevalence of no default in our data is 4 times higher then defaults. Our results are not much better then guessing using the global average default rate. The table part of the confusionmatrix shows that in the case of knn we almost missed predicting default in 5/6th of the cases:

| | 0 | 1 |
|---|---|---|
| 0 | 2201 | 593 |
| 1 | 116 | 90 |

## 4. Working with selected columns

### A. Choosing the 3 most important variables

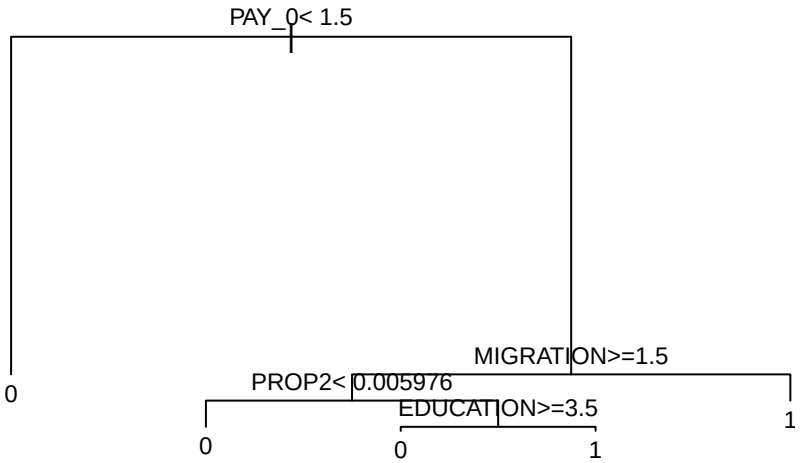
I started working with a subset of columns and the regression tree to identify the 3 most important variables that I used later:

- SEX
- EDUCATION
- MARRIAGE
- AGE
- PAY_0
- PROP1 = BILL_AMT1/PAY_AMT1
- PROP2 = BILL_AMT1/LIMIT_BAL
- MIGRATION = abs(PAY_0-PAY_6)

I removed NA, Inf, -Inf obtained during the transformation of data.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2232  464
##          1   85  219
##
##                Accuracy : 0.817
##                  95% CI : (0.8027, 0.8307)
##     No Information Rate : 0.7723
##     P-Value [Acc > NIR] : 1.288e-09
##
##                   Kappa : 0.353
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9633
##             Specificity : 0.3206
##          Pos Pred Value : 0.8279
##          Neg Pred Value : 0.7204
##              Prevalence : 0.7723
##          Detection Rate : 0.7440
##    Detection Prevalence : 0.8987
##       Balanced Accuracy : 0.6420
##
##        'Positive' Class : 0
##
```

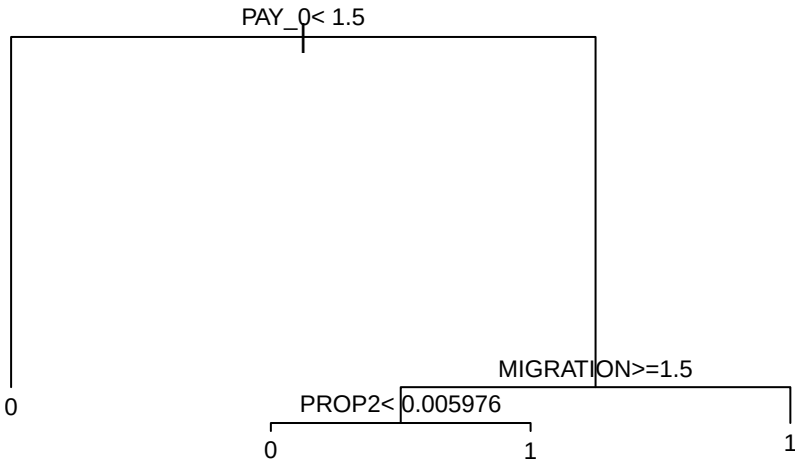
Results are slightly better but we still predict with a relatively low specificity, failing to predict default in 2/3 of the cases. Forcing the cp close to 0 created a regression tree with PAY_0, MIGRATION, PROP2 and EDUCATION columns. As we could expect, the PAY_0 (the payment status of the last month) column had still the most impact:



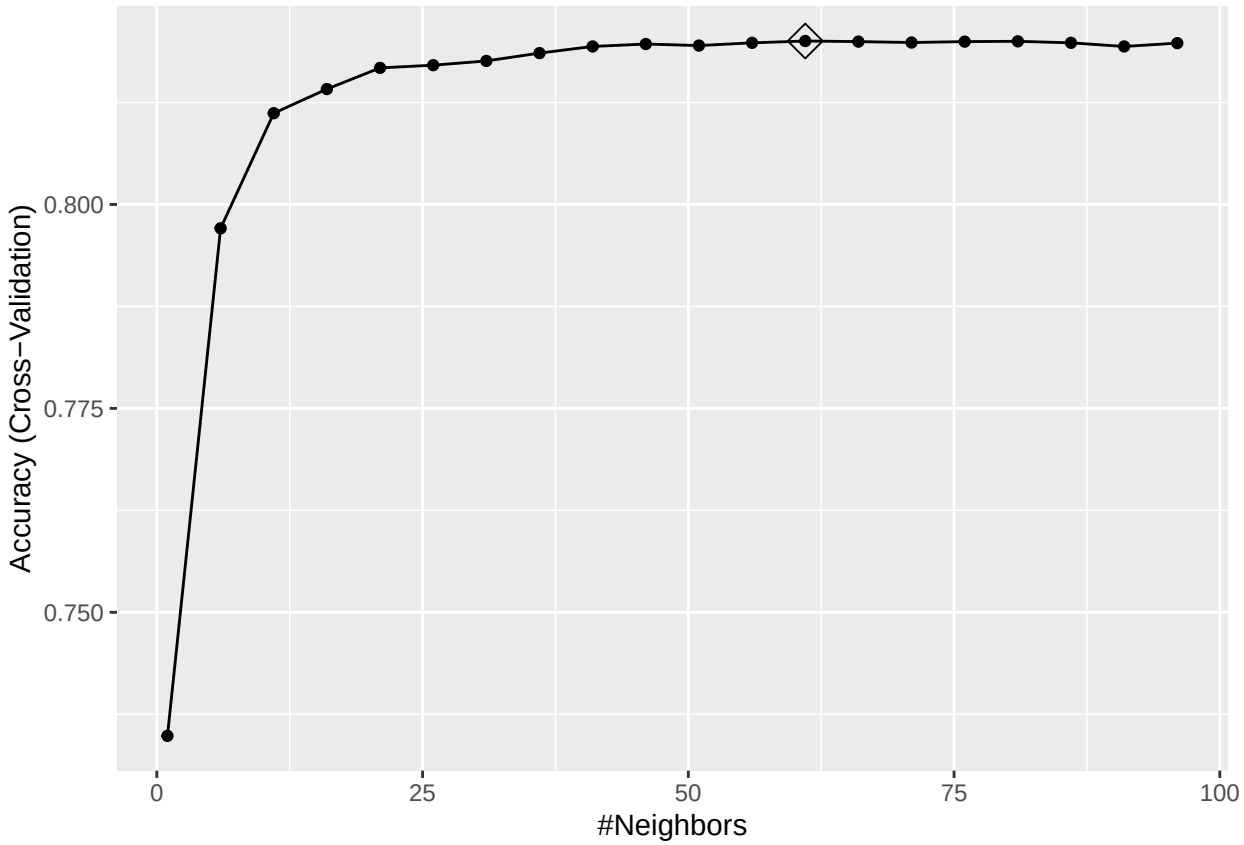

For the final algorithm I retained only 3 variables: PAY_0, MIGRATION and PROP2. I tried to predict defaults using the latest payment status, its evolution during the observed period and the distance from the accorded limit by the bank for each client. Using less variables helped to reduce computing time as well. For bigger datasets it could be a real advantage.

**B. Results using only 3 columns**

For the rpart algorithm I forced cp to be close to 0, by this way I guaranteed using all columns instead of making the decision tree only on PAY_0.




To train knn for the k value, I used cross validation. The final k that resulted the highest accuracy was 61.




Finally I created an ensemble with the 3 results:

| method | Accuracy | Sensitivity | Specificity | Balanced_Accuracy |
|---|---|---|---|---|
| rpart | 0.8166667 | 0.9624514 | 0.3221083 | 0.6422799 |
| knn | 0.8133333 | 0.9546828 | 0.3338214 | 0.6442521 |
| rforest | 0.8170000 | 0.9620199 | 0.3250366 | 0.6435282 |
| ensemble | 0.8163333 | 0.9615883 | 0.3235725 | 0.6425804 |

The ensemble could not beat the randomforest. The rf can be already seen as an ensemble, it is obtained by averaging regression trees. To improve results further I created a bigger ensemble with more models. This helps to identify other algorithms that we could use instead of choosing in an arbitrary way. Of course the proper method would be partitioning again the training set and not use the test set when we make a choice.

```
## note: only 2 unique complexity parameters in default grid. Truncating the grid to 2 .
##
## note: only 2 unique complexity parameters in default grid. Truncating the grid to 2 .
```

| | glm | lda | naive_bayes | svmLinear | knn | gamLoess | Rborist | qda |
|---|---|---|---|---|---|---|---|---|
| Accuracy | 0.8023333 | 0.8016667 | 0.8000000 | 0.7723333 | 0.8063333 | 0.8170000 | 0.7513333 | 0.8010000 |
| Sensitivity | 0.9736729 | 0.9723781 | 0.9266293 | 1.0000000 | 0.9430298 | 0.9598619 | 0.8636167 | 0.9192922 |
| Specificity | 0.2210835 | 0.2225476 | 0.3704246 | 0.0000000 | 0.3426061 | 0.3323572 | 0.3704246 | 0.3997072 |
| Balanced Accuracy | 0.5973782 | 0.5974628 | 0.6485269 | 0.5000000 | 0.6428180 | 0.6461096 | 0.6170207 | 0.6594997 |

Naive Bayes, GamLoess and qda seems to fit well to our results. By majority vote I used the results of this ensemble of 8 models as well. It is comparable to our initial model using the 3 algorithms.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2224  455
##          1   93  228
##
##                Accuracy : 0.8173
##                  95% CI : (0.803, 0.831)
##     No Information Rate : 0.7723
##     P-Value [Acc > NIR] : 9.717e-10
##
##                   Kappa : 0.3612
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9599
##             Specificity : 0.3338
##          Pos Pred Value : 0.8302
##          Neg Pred Value : 0.7103
##              Prevalence : 0.7723
##          Detection Rate : 0.7413
##    Detection Prevalence : 0.8930
##       Balanced Accuracy : 0.6468
##
##        'Positive' Class : 0
##
```

# IV. Conclusion

By using only 3 variables, we were able to predict default with around 80% global accuracy. This result seems really good, but in reality banks are more intereseted in predicting default then no default. Specificity is even more important in our model then sensitivity (if the positive value is the 0 = "no default" as in our case). To take this into account, we could train our algorithms to maximize the balanced accuracy, or even give more weight (beta) to specificity when computing the F-score.