CREDIT SCORECARD

ABSTRACT

A credit scorecard is a three-digit number that depicts a customer’s creditworthiness. Credit score card is based on credit history; number of open accounts, total level of debts, and repayment history. Lenders use credit scores to evaluate the probability that an individual will repay loans in a timely manner.

Your credit score is the number one that cost or save you a lot of money. An excellent score can land you lower interest rates, but it’s up to you the borrower, to make sure your credit score is strong.

This project on credit score helps you check your scores from time to time so you can keep track and improve your score to avail benefits.

INTRODUCTION

The methodologies and the processes of developing credit scorecards for lending purposes have been a long-held property of financial institutions and consulting firms.

Practical guidance and demonstration on the hands-on aspect of credit score cards development have been in short supply. This could be due to the following reason- availability of data is a common hinderance. Due to regulations and customer privacy, financial institutions cannot share data with external parties.

This project seeks to fill a gap by providing a practical hand-on demonstration on how to develop a credit score card, as close to reality as possible.

A credit score card development package, called **scorecard**, has been built in R by Scichen Xie. This package makes credit score card development a breeze and is used through the project.

R PACKAGES USED

* scorecard- for credit score card development
* tidyverse- for data import and manipulations
* Hmisc- for summary statistics generation
* ClusOfvar- for variable clustering

DATA SET

The data set contains **30,000 credit card holders** in Taiwan. In addition to the usual demographic information, payments and outstanding balances, the delinquency status of the credit card holders were tracked over a period of time (6 months). In total, there are **24 variables** in the data set.

CODE

1. DATA
   1. IMPORT DATA INTO R

#load the tidyverse package

library(tidyverse)

#download data from github

Data= read.csv(“dataset.csv”)

#print first few lines of data

head(data)

#print out structure of the data

str(data)

* 1. DATA EXPLORATION

library(Hmisc)

describe(data)

* 1. DATA MANIPULATION

data<-data5>%

#rename variables in lower case

rename\_with(str\_to\_lower, everything())%>%

#rename “pay\_0” to “pay\_1” and “default payment next month” to “ gb\_flag”

rename(“pay\_1=”pay\_0”,”gb\_flag”=”default.payment.next.month”)

#set up a function to recoil delinquency status

new\_deq<-function(x)

{

ifelse(x %in% c(-2,-1), 0, x)

}

#recode delinquency status

dat<-data%>%

mutate(across (pay\_1:pay\_6, new\_deq, .names=”{col}\_recode”))

#check

table(data$pay\_1, data$pay\_1\_recode)

* 1. VARIABLE GENERATION

#deliquency status

#un-needed “pay\_x” variables removed

var\_data<-data%>%

select(-(pay\_1:pay\_6))

#identify position of variables

deq\_pos\_l3m<-which(str\_detect(manes(var\_data),”pay\_(1|2|3)\_recode”))

deq\_pos\_l6m<- which(str\_detect(manes(var\_data),”pay\_(1|2|3|4|5|6)\_recode”))

#generate new variables

var\_data<-var\_data%>%

mutate(

#average delinquency

avg\_deq\_lm3=apply(var\_data[,deq\_lm3], 1, mean, na.rm=true),

avg\_deq\_lm6=apply(var\_data[,deq\_lm6], 1, mean, na.rm=true),

#maximum delinquency

max\_deq\_lm3=apply(var\_data[,deq\_lm3], 1, max, na.rm=true),

max\_deq\_lm6=apply(var\_data[,deq\_lm6], 1, max, na.rm=true),

#minimum delinquency

min\_deq\_lm3=apply(var\_data[,deq\_lm3], 1, min, na.rm=true),

min\_deq\_lm6=apply(var\_data[,deq\_lm6], 1, min, na.rm=true)

)

#utilization rate

#function to compute utilization rate

uti\_rate<-function(x)

{

ifelse(x<0, 0, x/var\_data$limit\_bal)

}

#compute utilization rate

var\_data<-var\_data%>%

mutate(across(contains(“bill\_amt”), .fns=list(util=uti\_rate)))

#identify position of variable

util\_pos\_l3m<-which(str\_detect(names(var\_data),”bill\_amt(1|2|3)\_util”))

util\_pos\_l6m<-which(str\_detect(names(var\_data),”bill\_amt(1|2|3|4|5|6)\_util”))

#generate new variables

var\_data<-var\_dat%>%

mutate(

#average utilization

avg\_util\_l3m=apply(var\_data[, util\_pos\_l3m], 1, mean, na.rm=true),

avg\_util\_l6m=apply(var\_data[, util\_pos\_l6m], 1, mean, na.rm=true),

#maximun utilization

max\_util\_l3m=apply(var\_data[, util\_pos\_l3m], 1, max, na.rm=true),

max\_util\_l6m=apply(var\_data[, util\_pos\_l6m], 1, max, na.rm=true),

#minimum utilization

min\_util\_l3m=apply(var\_data[, util\_pos\_l3m], 1, min, na.rm=true),

min\_util\_l6m=apply(var\_data[, util\_pos\_l6m], 1, min, na.rm=true)

)

* 1. SAMPLING

#set seed for random sample

set.seed(1234)

#randomly select from id column

dev\_ind<-sample(var\_data$id, 24000, replace=false)

#use random select id to form development sample

dev<-var\_data[dev\_ind,]

#use non select id to form validation sample

oot<-var\_data[-dev\_ind,]

2. UNIVARIATE ANALYSIS

* 1. FINE CLASSING

library(scorecard)

#select all variables for computation

var\_list<-dev%>%

select(-id, -gb\_flag)%>%

names()

#invoke woebin function in package scorecard

fine\_class<-woebin(

dev,

y= ”gb\_flag”,

x= var\_list,

positive= 1,

method= ”freq”,

bin\_num\_limit= 20

)

#collection of variables

iv<-map\_df(fine\_class, ~pluck(.x, 10, 1))%>%

pivot\_longer(everything(), names\_to=”var”, values\_to=”iv”)

* 1. INITAIL VARIABLES REMOVAL

#keep relavant variables

dev<-dev%>%

select(id, age, gb\_flag:max\_util\_l6m, avg\_deq\_l3m:min\_deq\_l3m, bill\_amt6\_util)

* 1. VARIABLE CLUSTERING

library(ClusOfVar)

#perform variable clustering

tree<-dev%>%

select(-id, -gb\_flag)%>%

hclustvar()

#stability plot

set.seed(345)

stab<-stability(tree, B=30)

boxplot(stab$matCR, main= “dispersion of the adjusted rand index)

#generate the final cluster output

clus<-cutree(tree, 4)

* 1. VARIABLE REDUCTION

3. COARSE CLASSING

plot<-woebin\_plot(fine\_class\_final)

plot[[1]]

plot[[2]]

plot[[3]]

plot[[4]]

plot[[5]]

plot[[6]]

plot[[7]]

breaks\_list<-list(

age= c(“25”, “45”),

avd\_deq\_l3m= c(“0.67”, “2”),

pay\_1\_recode= c(“1”, “2”),

pay\_4\_recode= c(“1”),

pay\_5\_recode= c(“2”),

avg\_util\_l6m= c(“0.45”, “0.83”),

max\_util\_l6m= c(“o,43”, “1”)

)

#set positive= 0

coarse\_class<-woebin(

dev,

y= “gb\_flag”,

x= short\_var\_list,

positive= 0,method= “freq”,

break\_list= break\_list

)

#transform variable values to woe values

dev\_woe<-woebin\_ply(dev, coarse\_class)

4. REGRESSION ANALYSIS

#logistic regression

logistic<-glm(

I(gb\_flag==0)~.,

family= binomial(),

data= dev\_woe%>% select(-id)

)

#setwise regression

logistic\_step<-step(logistic, direction= “both”, trace= false)

#print output

summary(logistic\_step)

#generate VIF

vif(logistic\_step, merge\_coef= true)

5. SCORECARD CREATION, SCALING NAD VALIDATION

#select variables in final regression

dev\_final<- dev%>%

select (id, age, gb\_flag, avg\_deq\_l3m, pay\_1\_recode, pay\_4\_recode, pay\_5\_recode, avg\_util\_l6m)

var\_select<-c(“age”, “avg\_deq\_l3m”, “pay\_1\_recode”, “pay\_4\_recode”, “pay\_5\_recode”, “avg\_util\_l6m”)

break\_list<-list(

age= c(“25”, “45”),

avd\_deq\_l3m= c(“0.67”, “2”),

pay\_1\_recode= c(“1”, “2”),

pay\_4\_recode= c(“1”),

pay\_5\_recode= c(“2”),

avg\_util\_l6m= c(“0.45”, “0.83”)

)

bins<-woebin(

dev\_final,

y= “gb\_flag),

x= var\_select,

positive= 0,

method= “freq”,

breaks\_list= breaks\_list

)

score\_card<-score\_card(

bins,

logistic\_step,

points0= 500,

pdo= -30,

oods0= 100

basepoints\_eq0= true

)

#display results

score\_card

#compute score

score<-scorecard\_ply(dev\_final, score\_card, only\_total\_score= f)

#validation

#select variables from oot sample

oot\_final<-oot%>%

select(id, age, gb\_flag, avg\_deq\_l3m, pay\_1\_recode, pay\_4\_recode, pay\_5\_recode, avg\_util\_l6m)

#generate reports

#report(

list(dt1= dev\_final, dt\_2= oot\_final),

y= “gb\_flag”,

x= var\_select,

breaks\_list= break\_list,

seed= nul,

basepoints\_eq0= true,

method= “freq”,

positive= 0,

points0= 500,

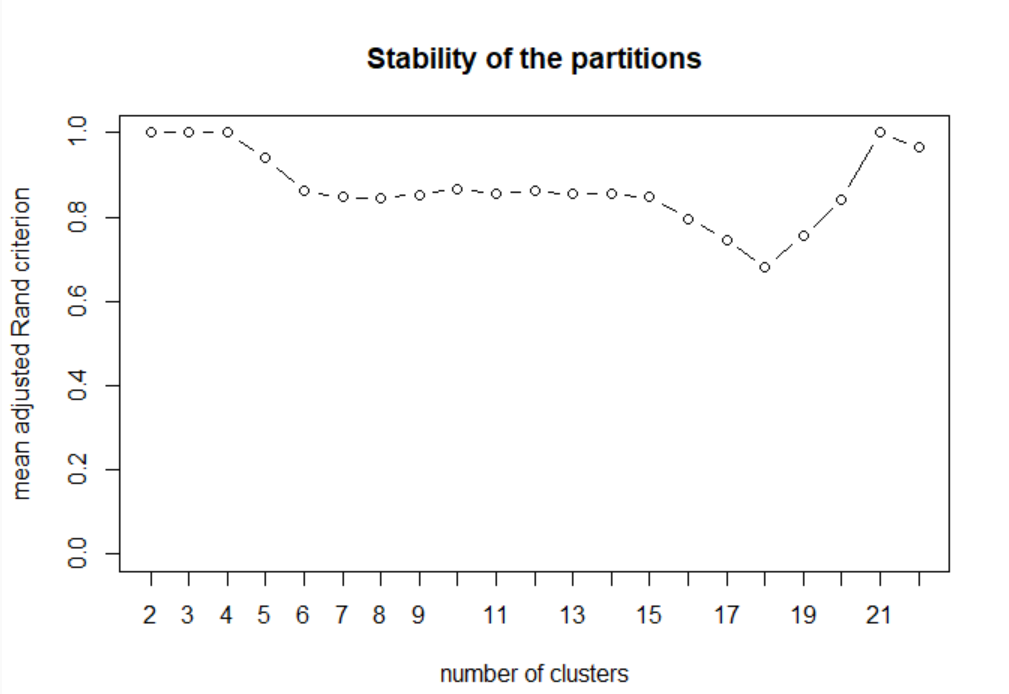
oods0= 100,

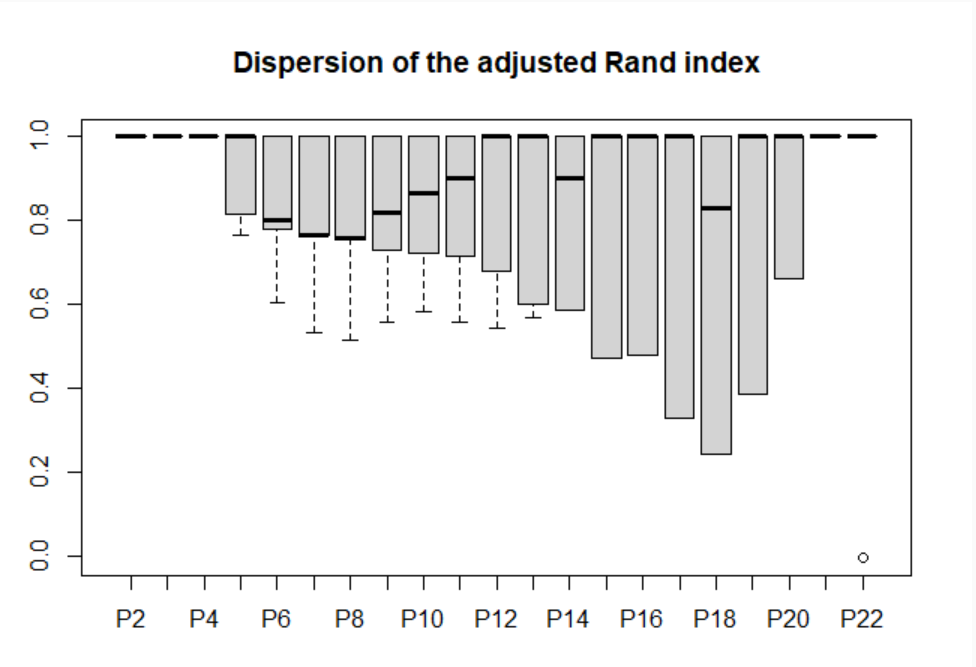
pdo= -30

)

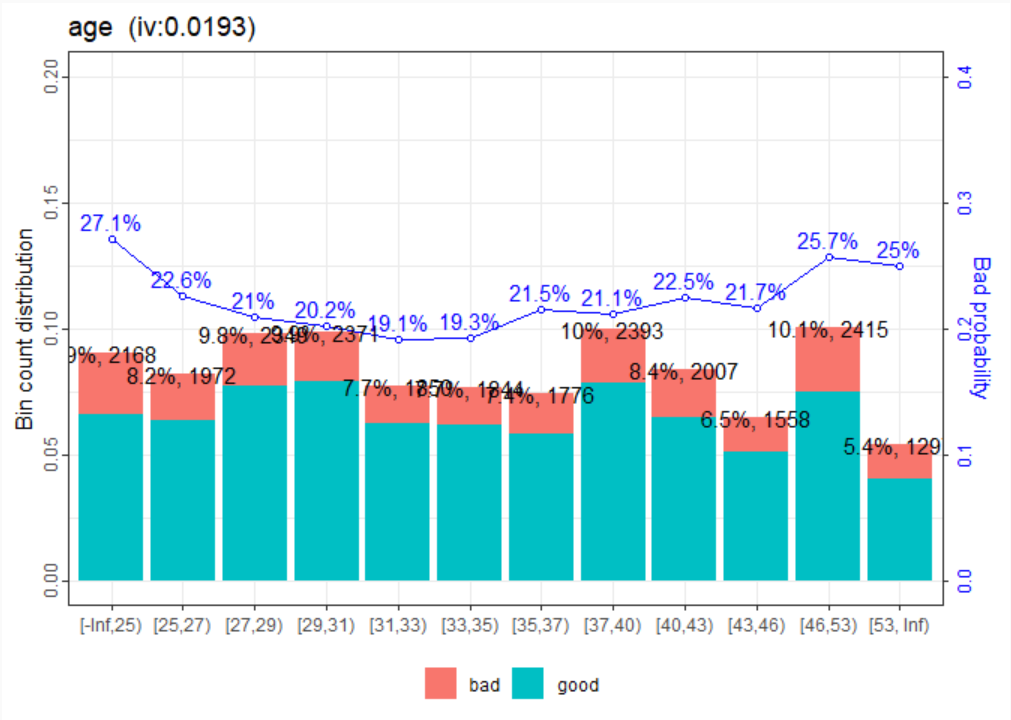
PLOTS

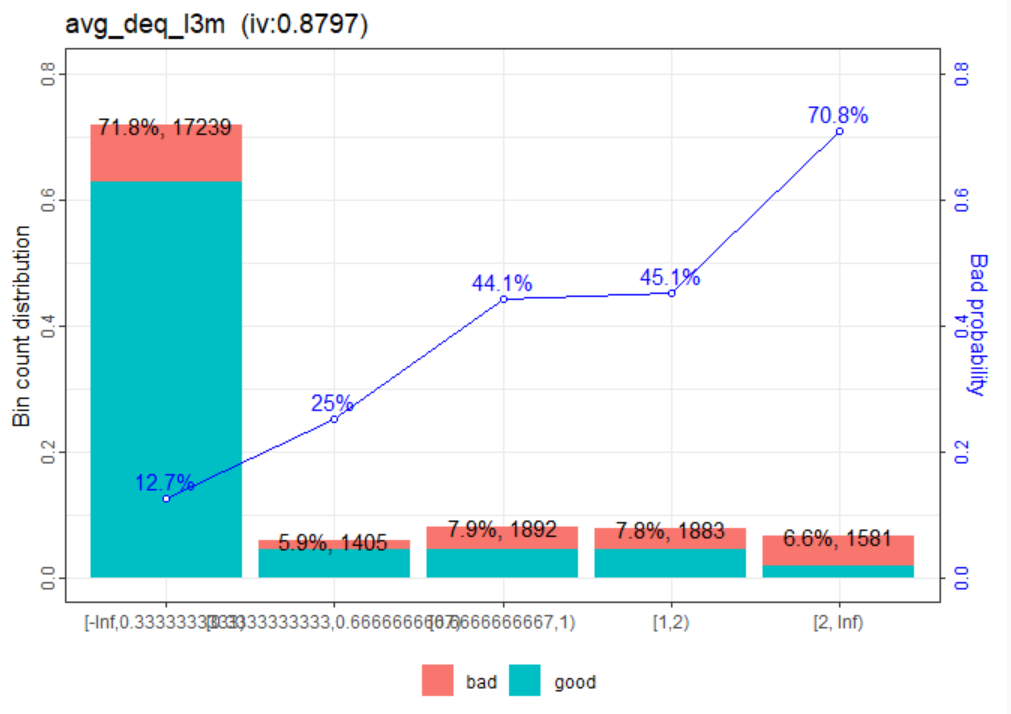
* 1. VARIABLE CLUSTERING

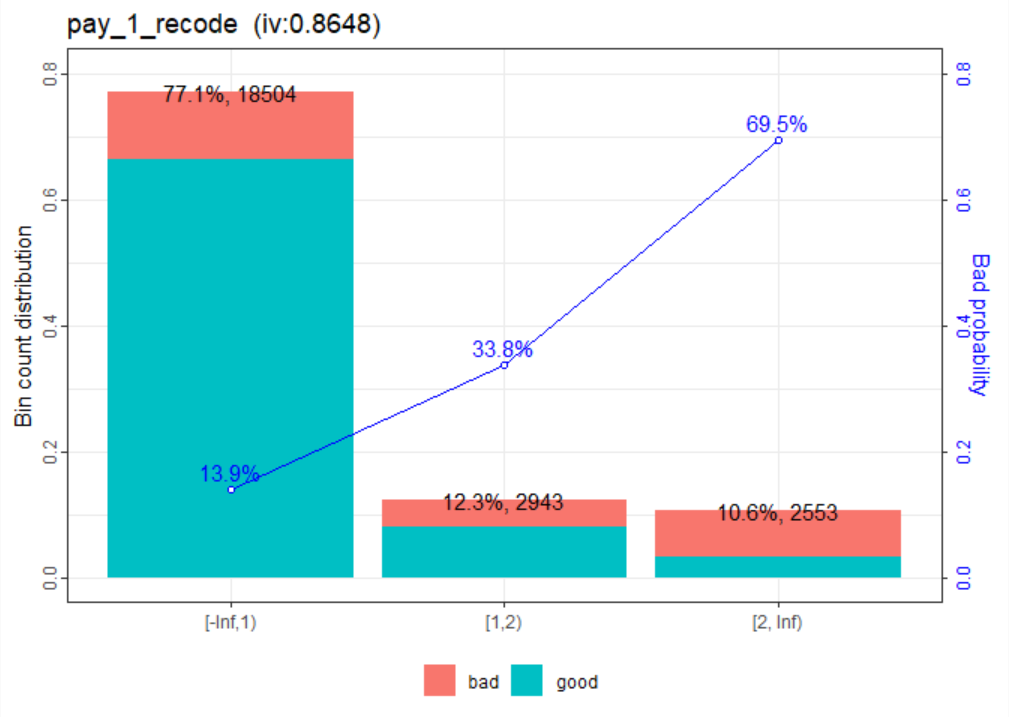


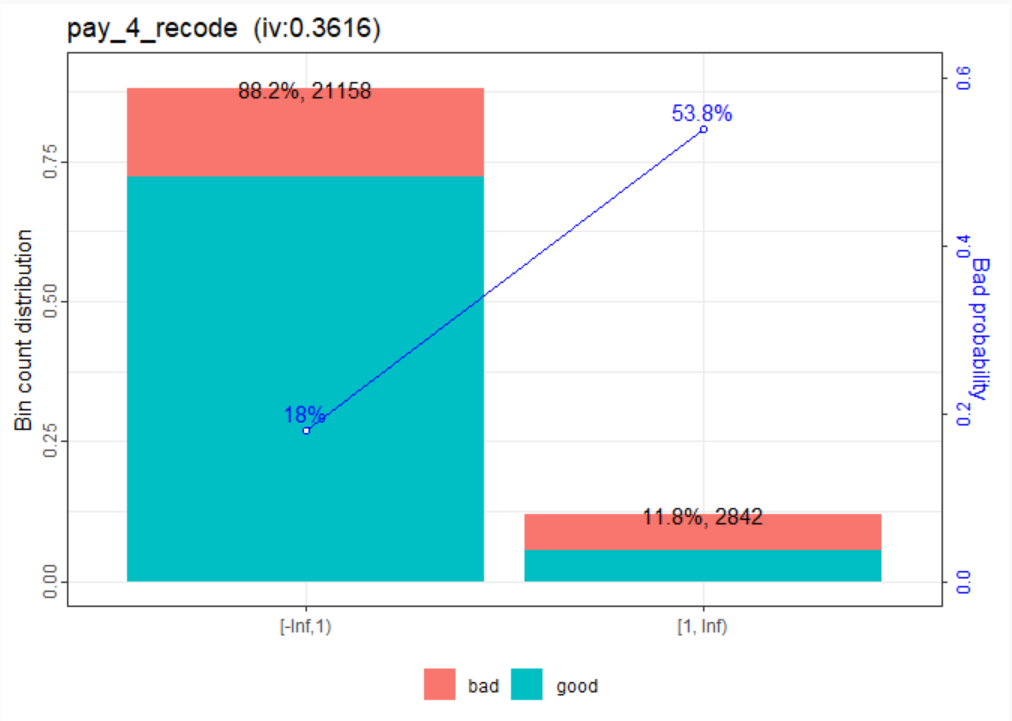


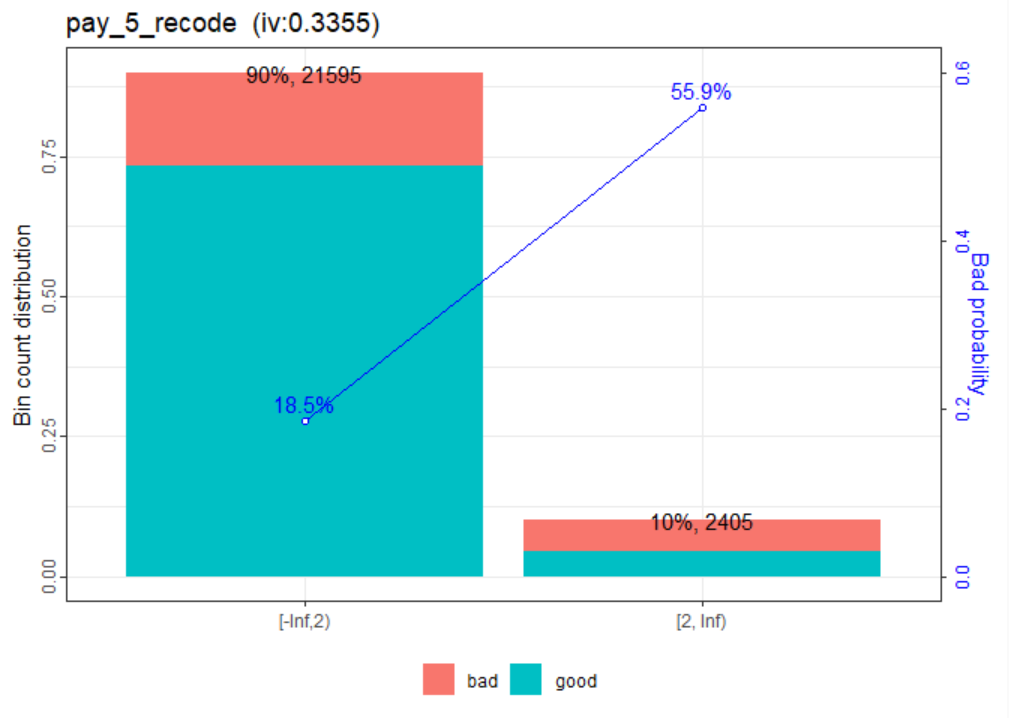
3. COARSE CLASSING

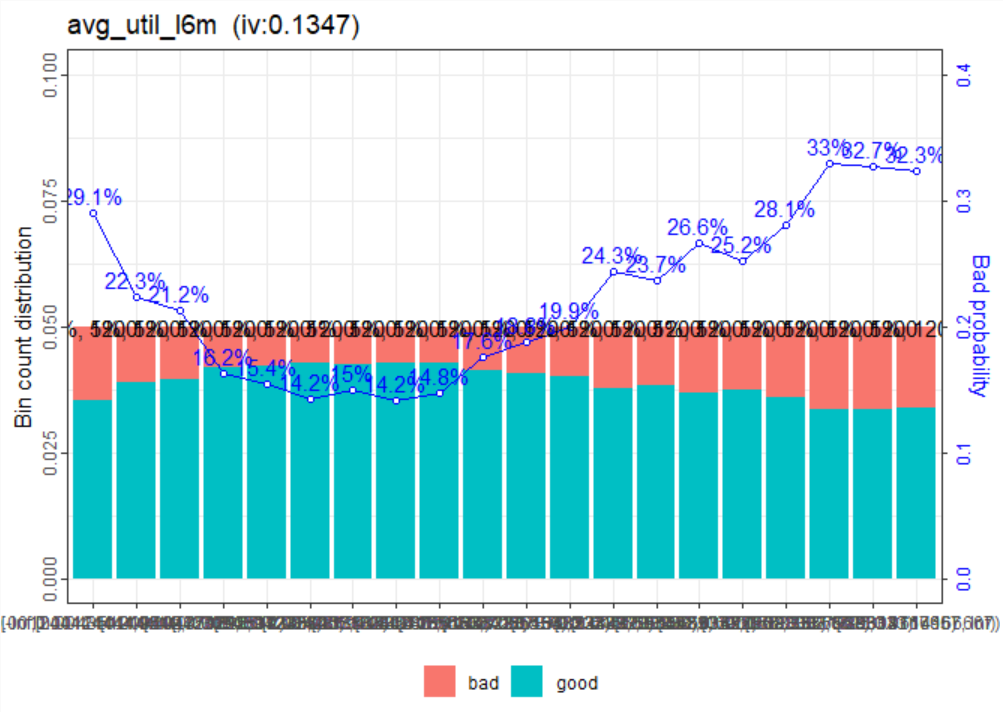


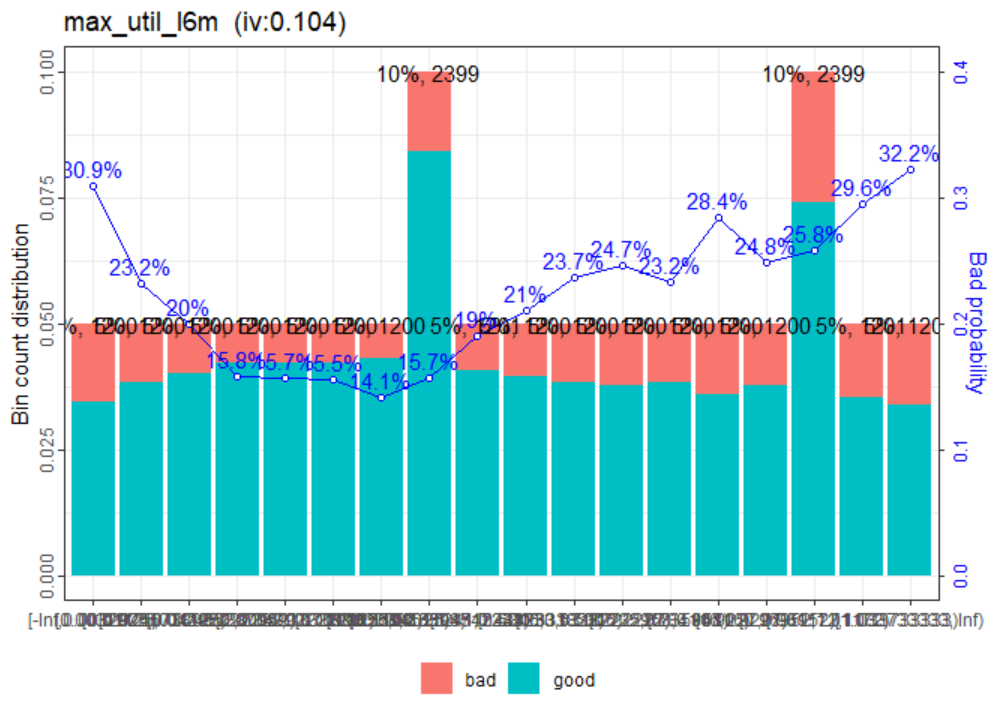




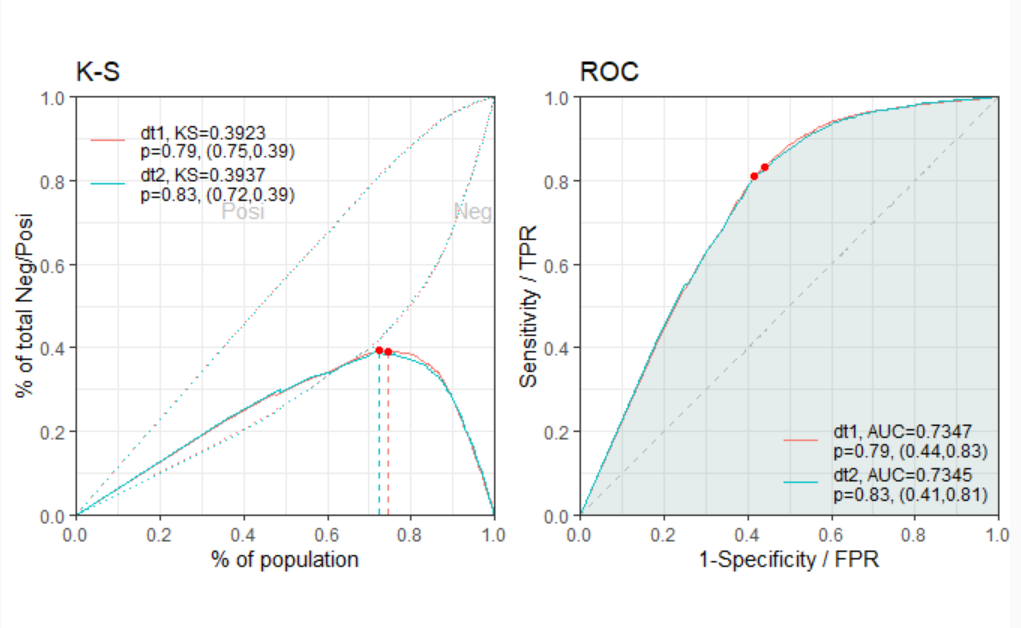


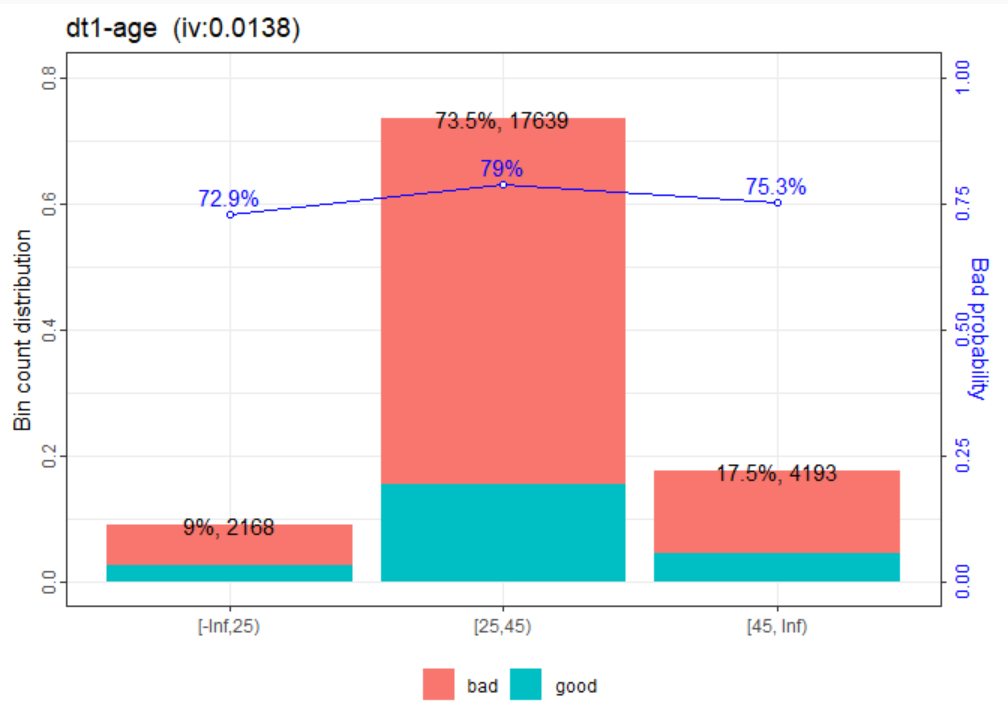


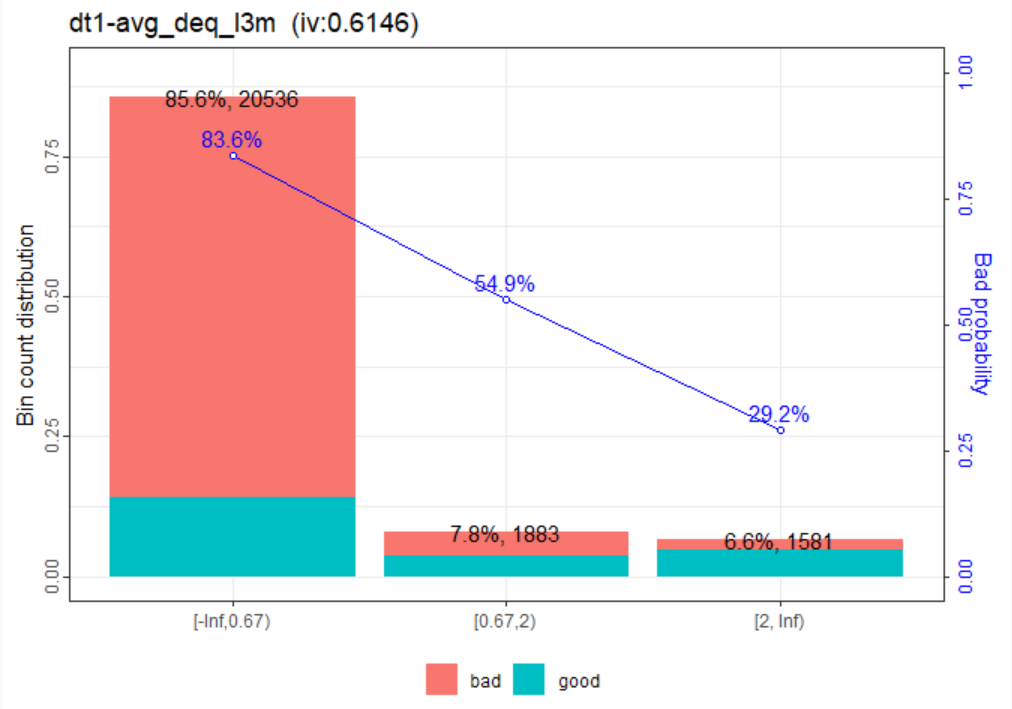


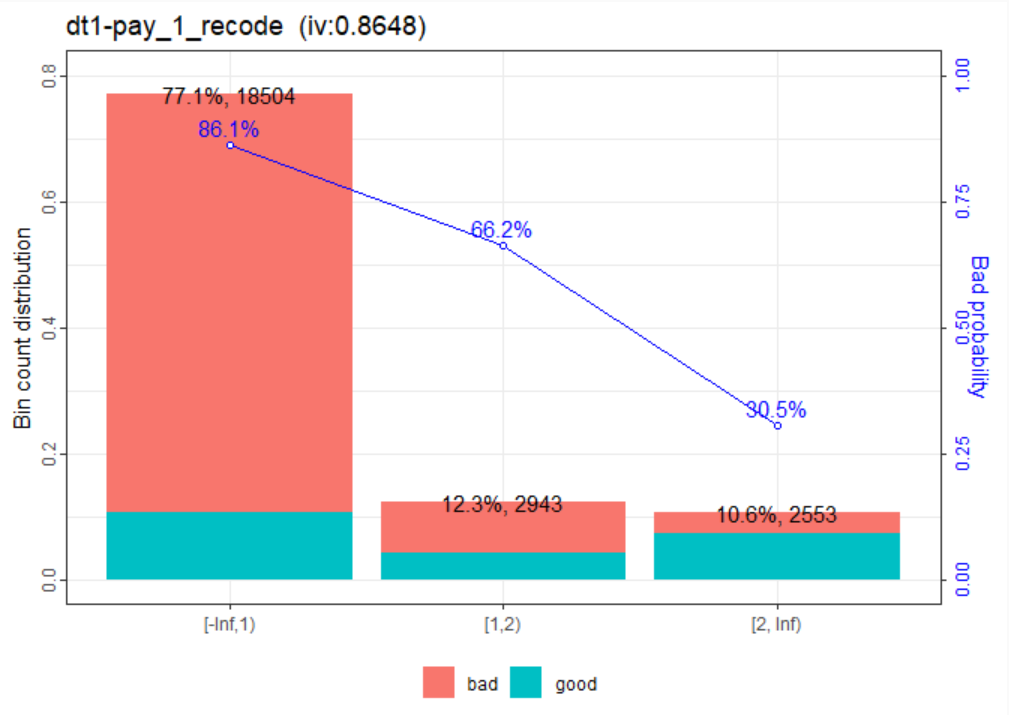


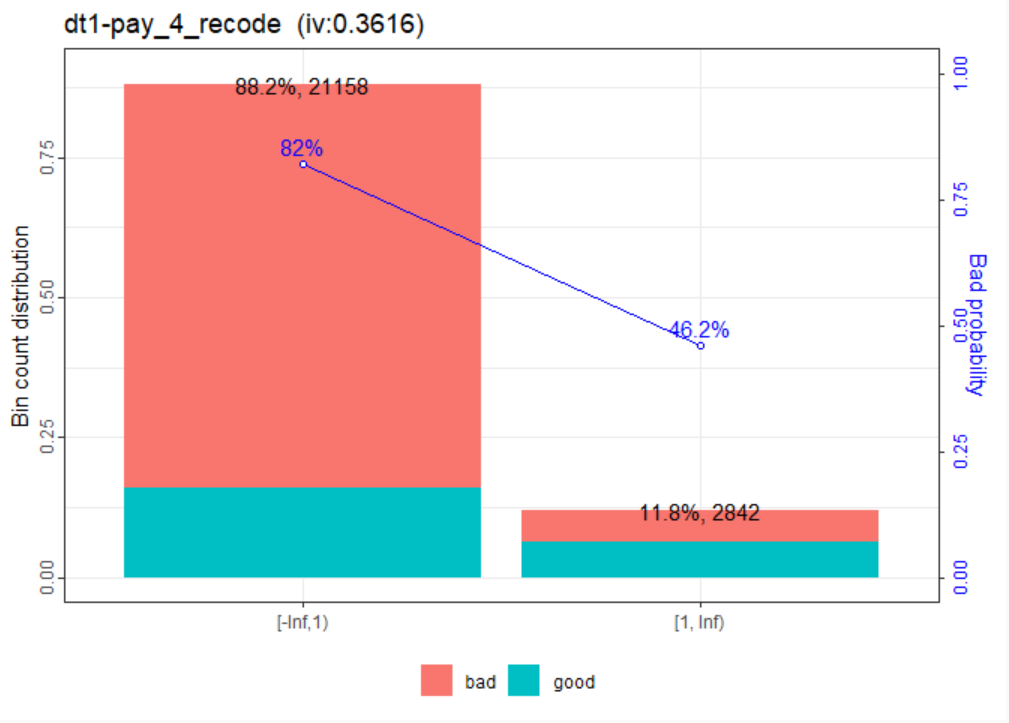
5. SCORECARD CREATION, SCALLING AND VALIDATION

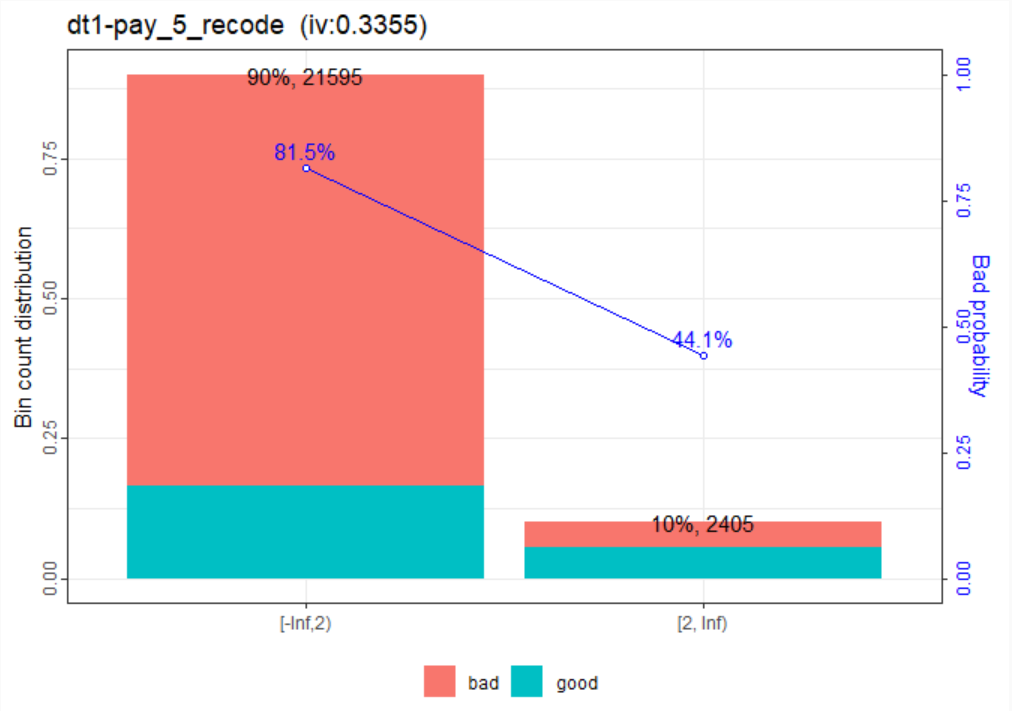


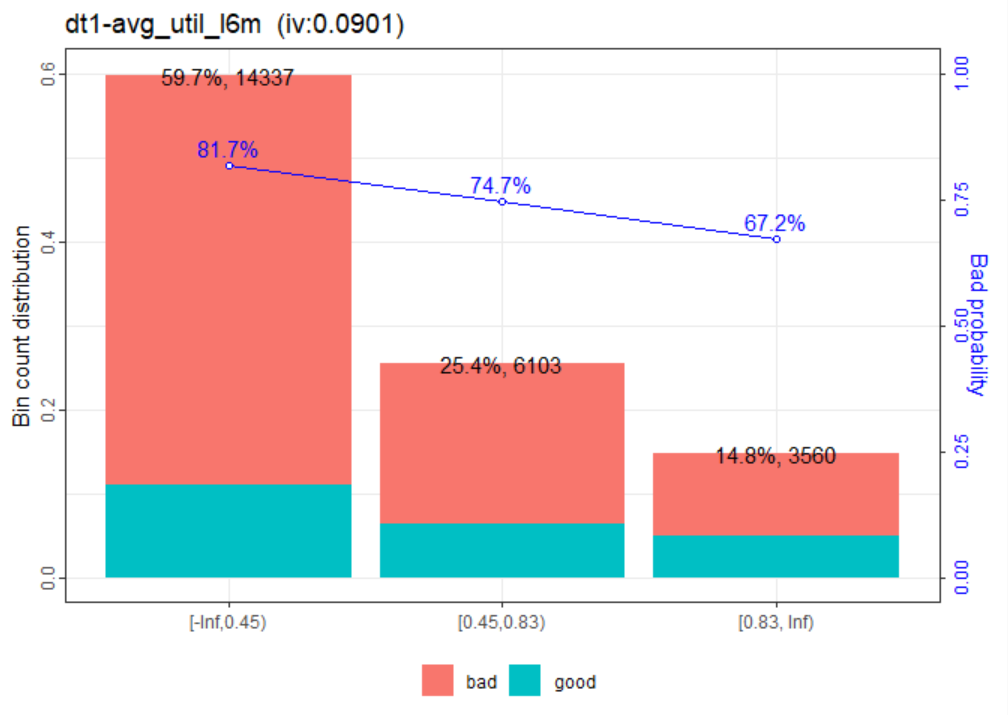


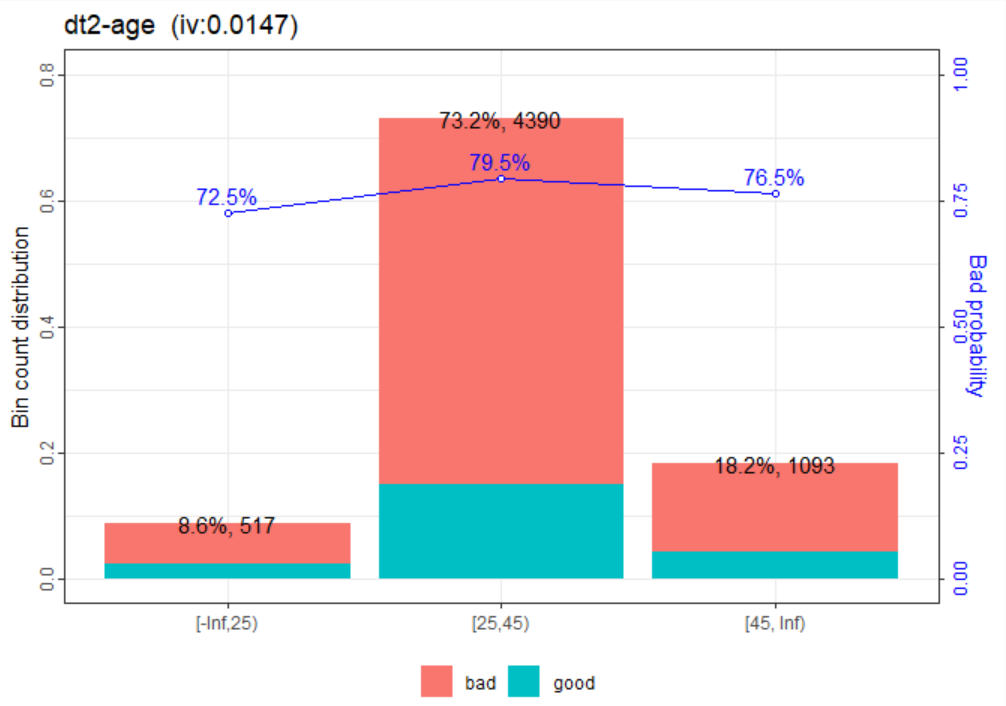


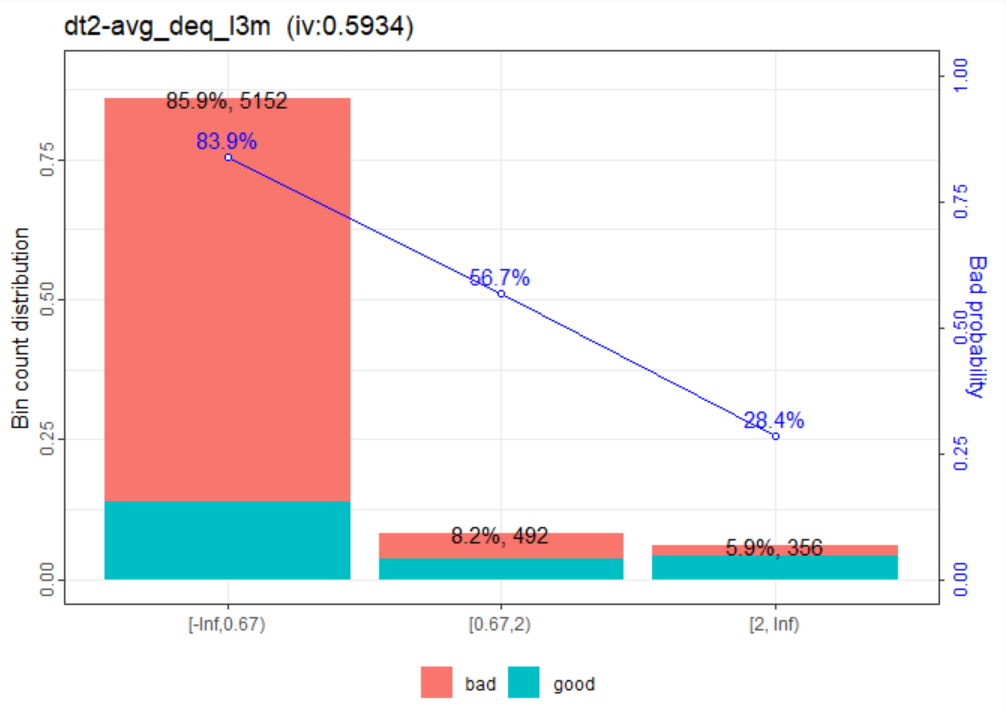


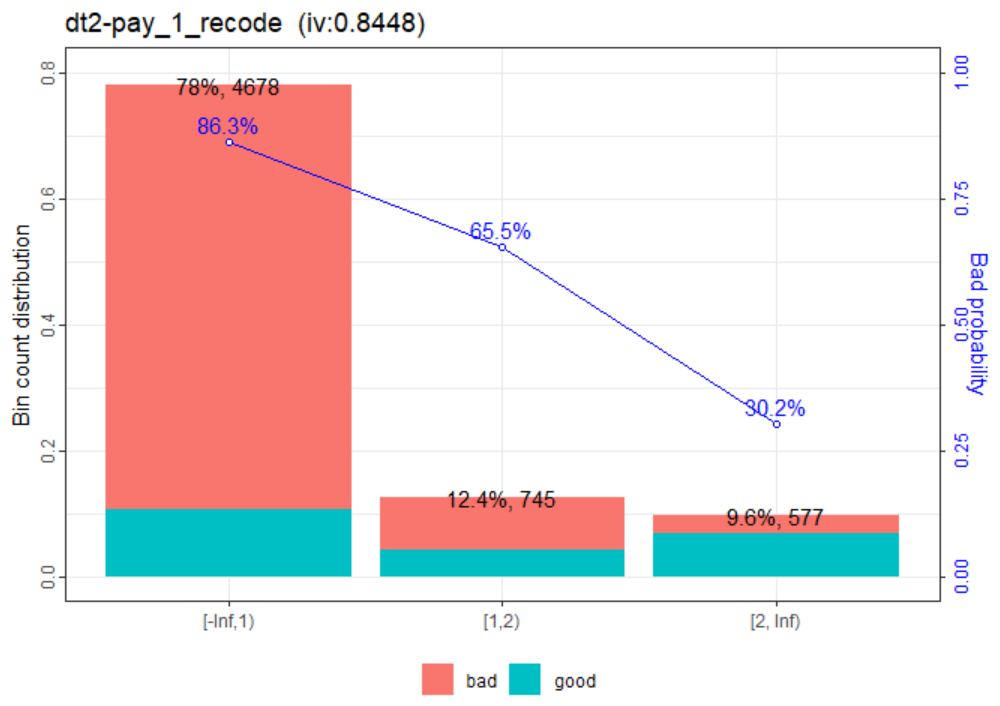


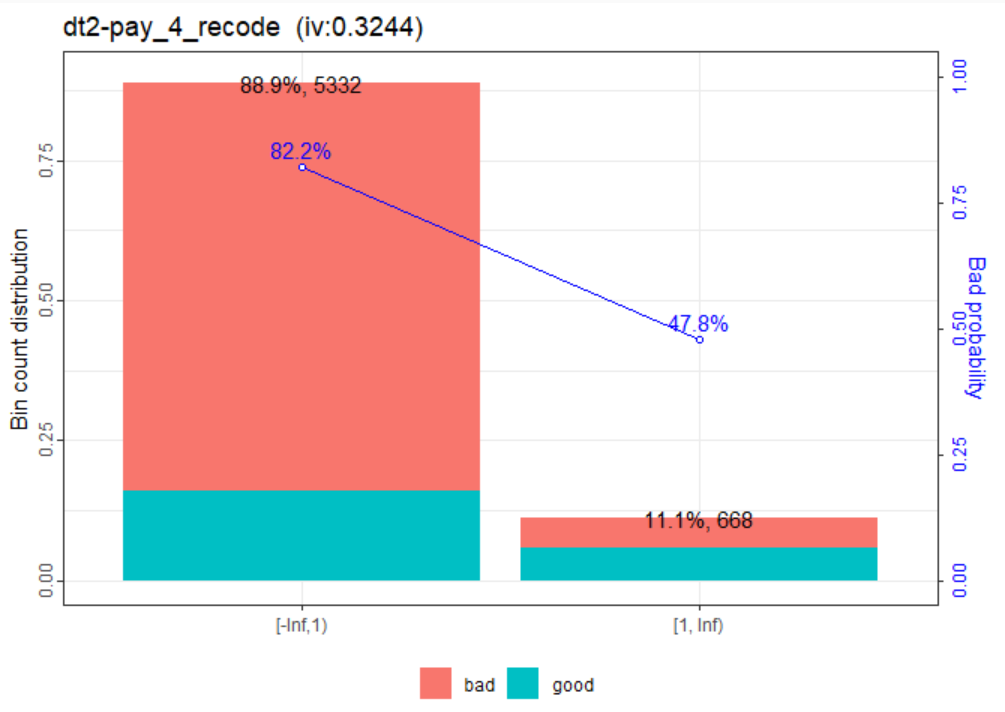


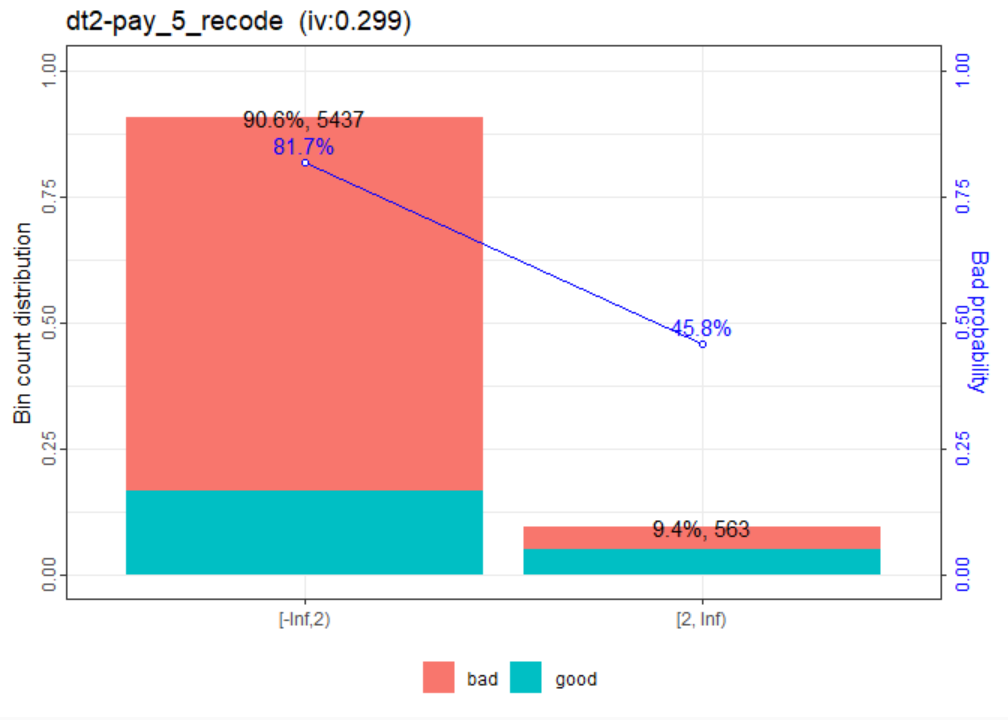


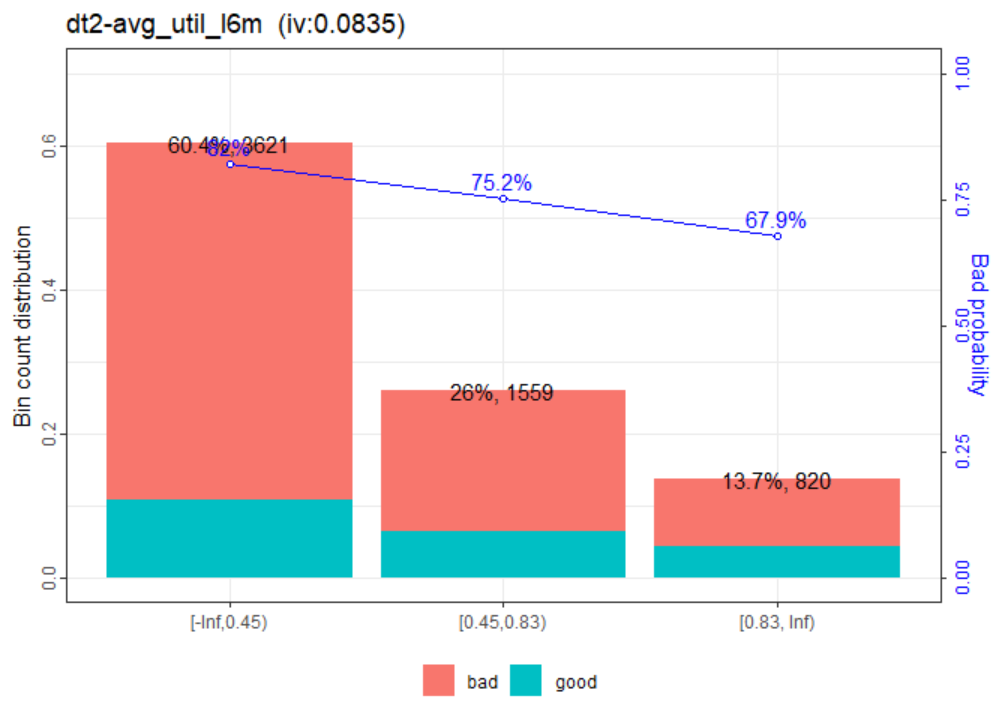


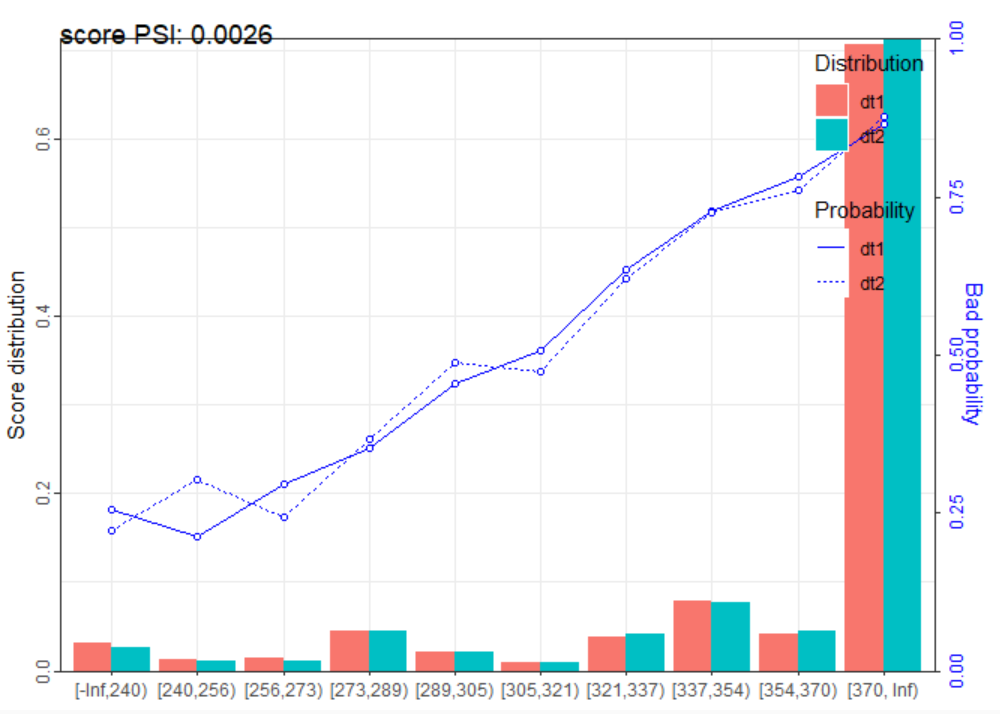






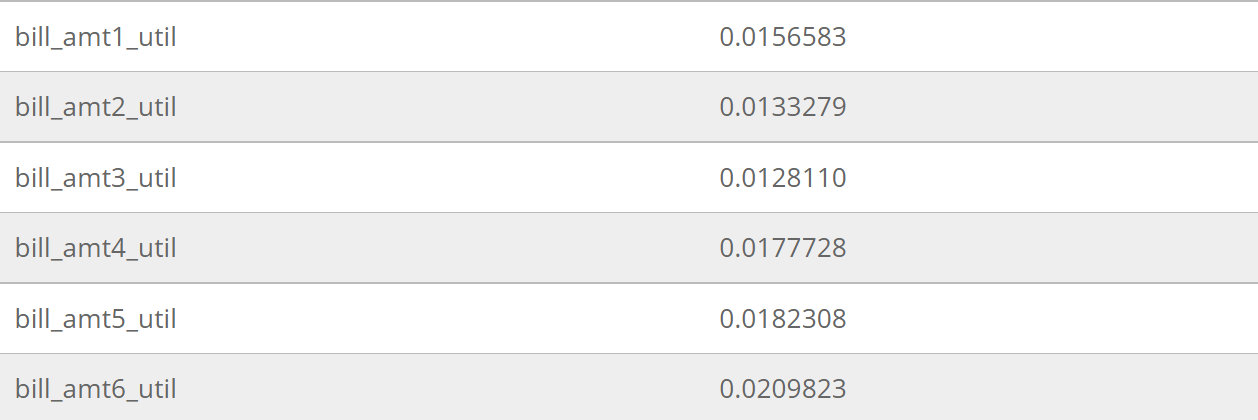
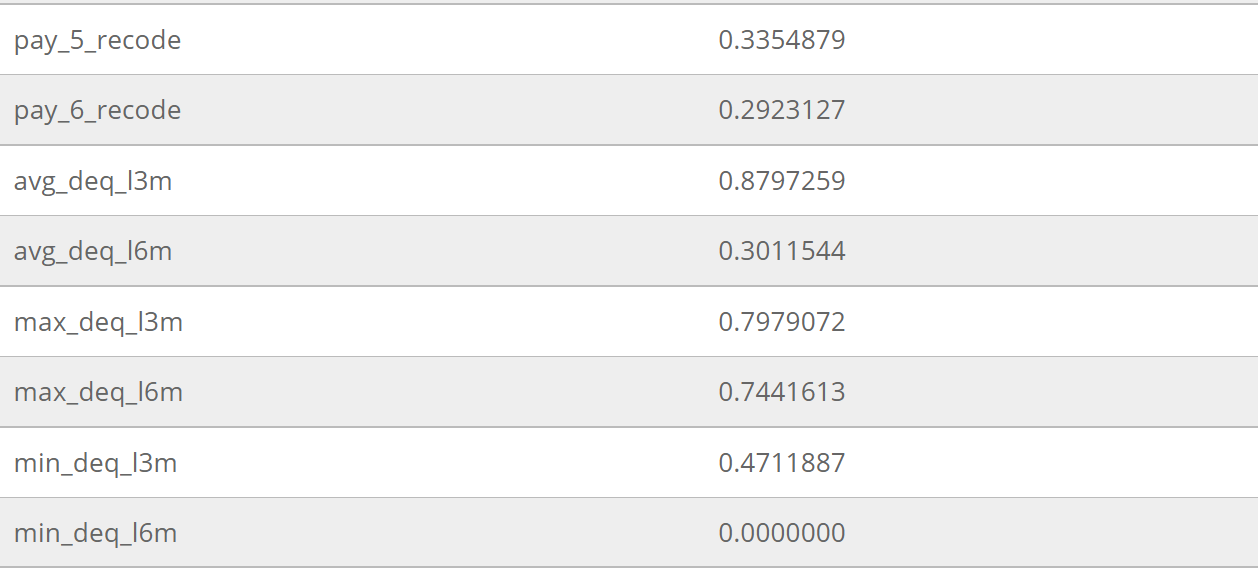
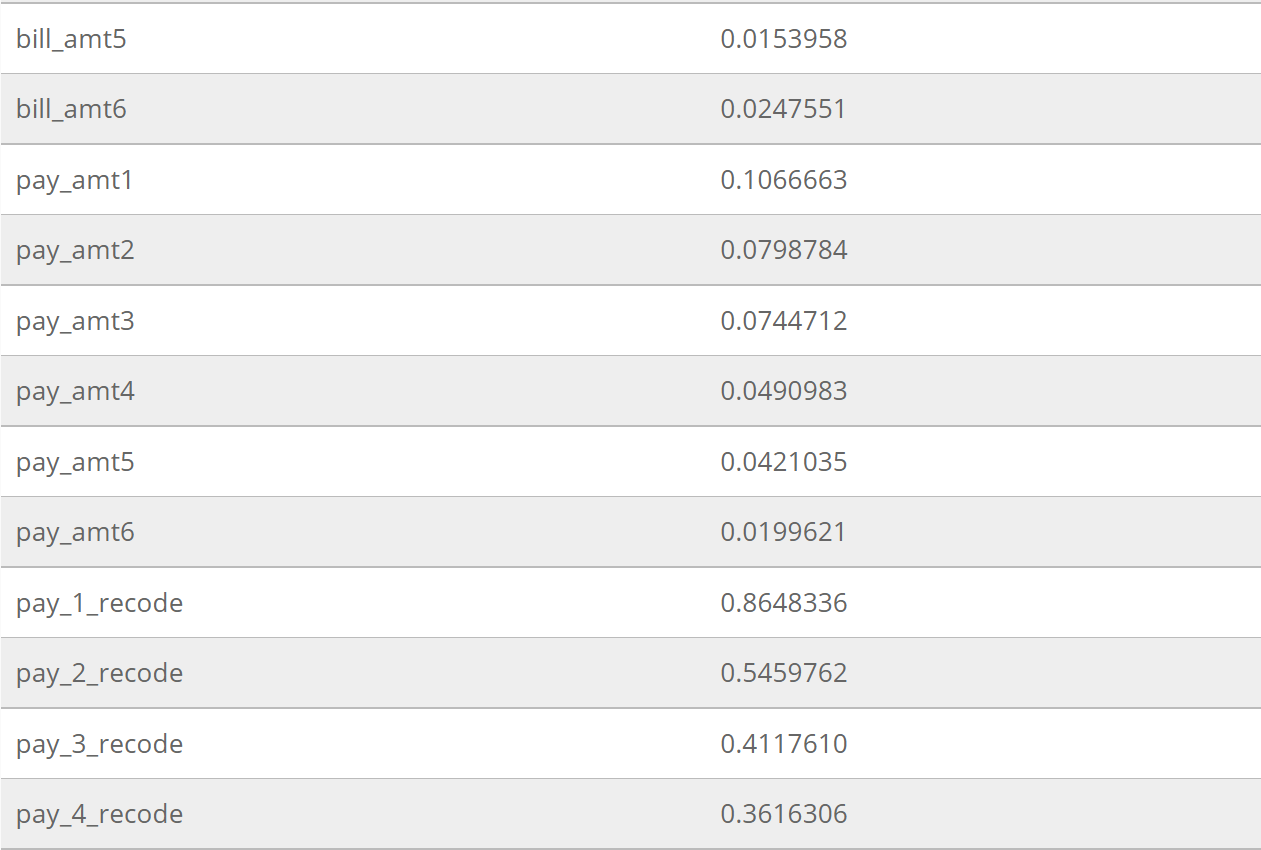
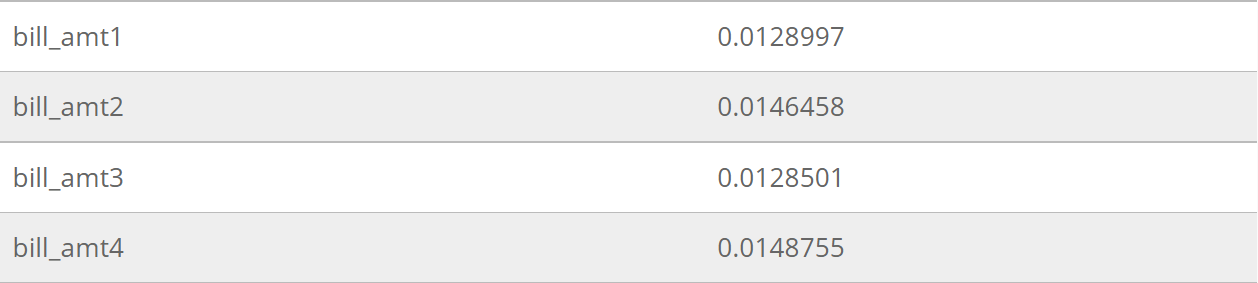
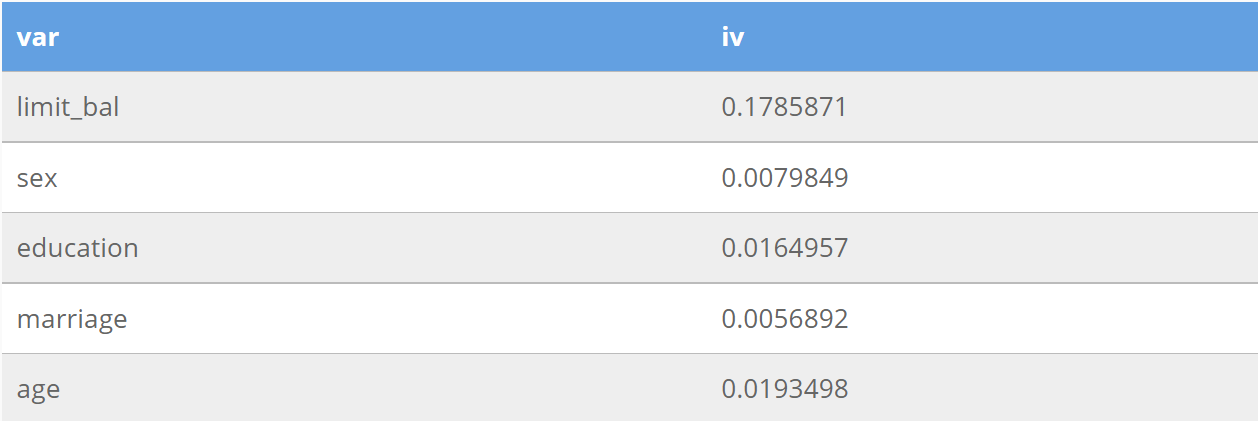


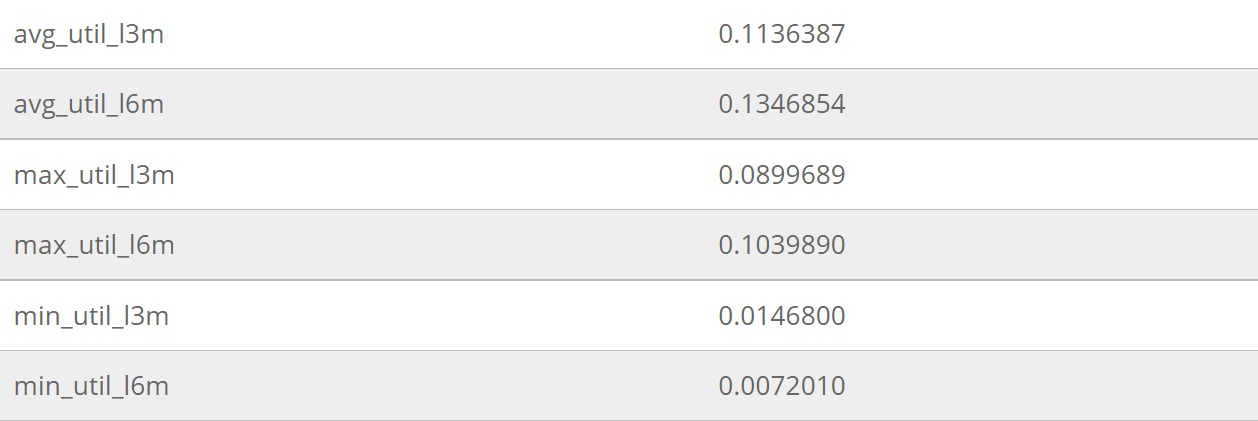




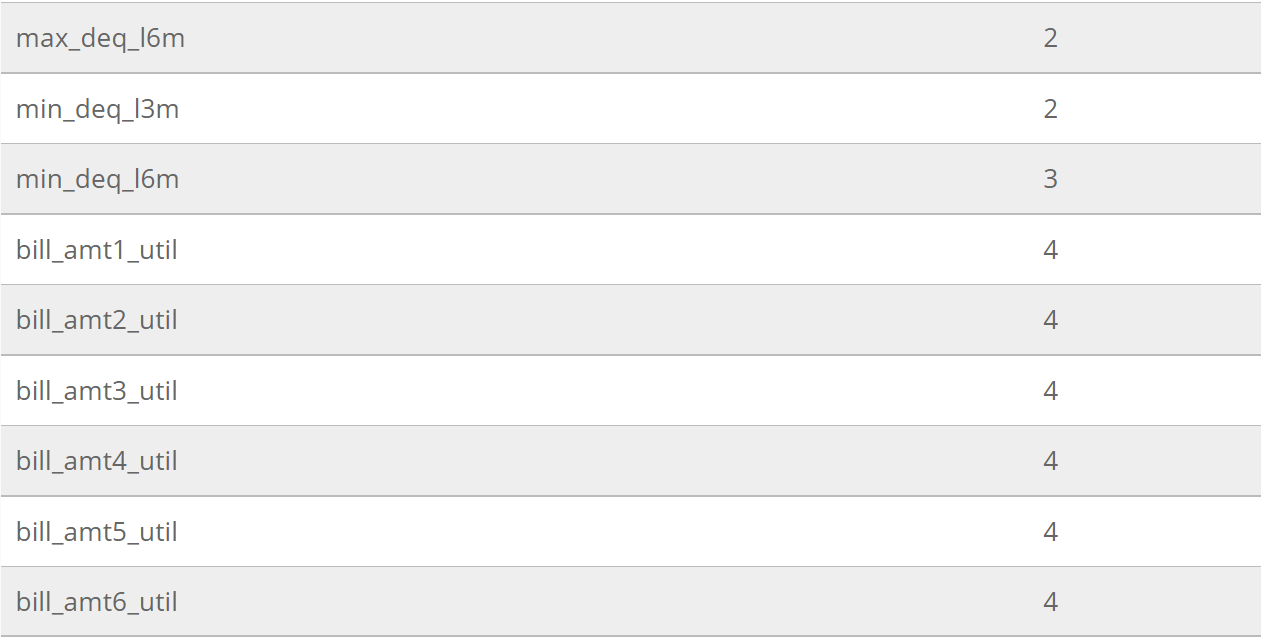
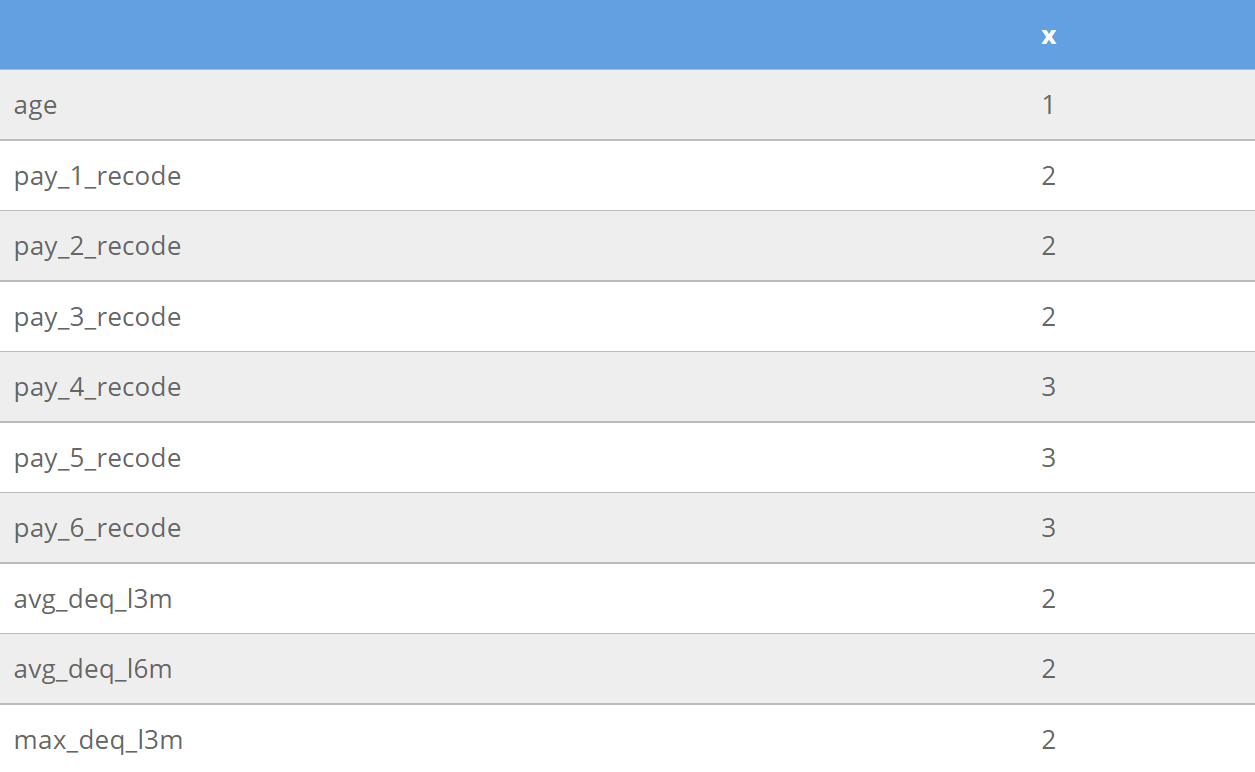
TABLES

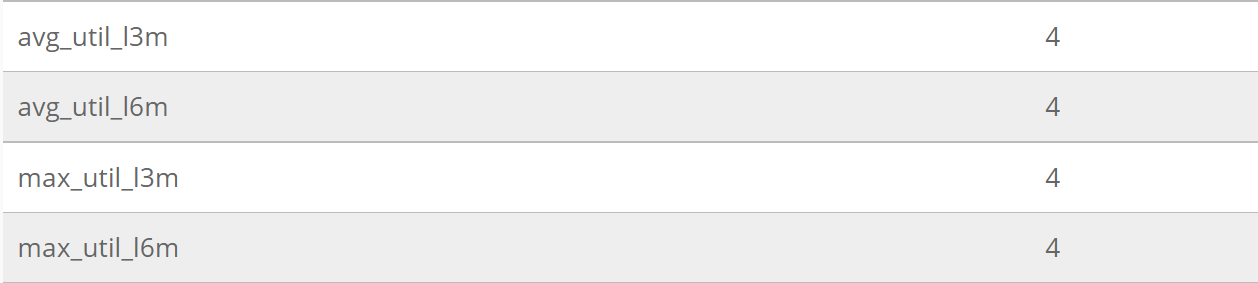
2.1 FINE CLASSING



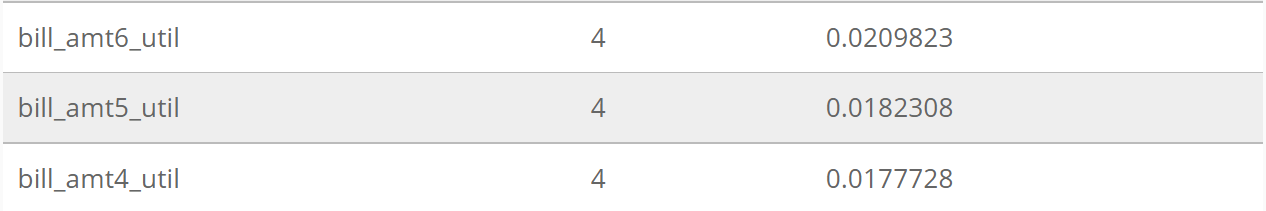
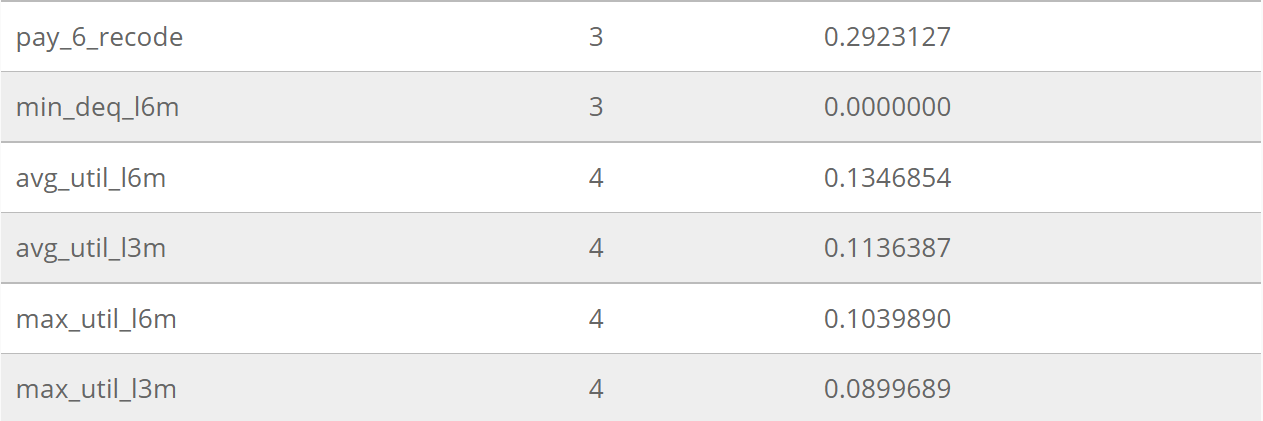


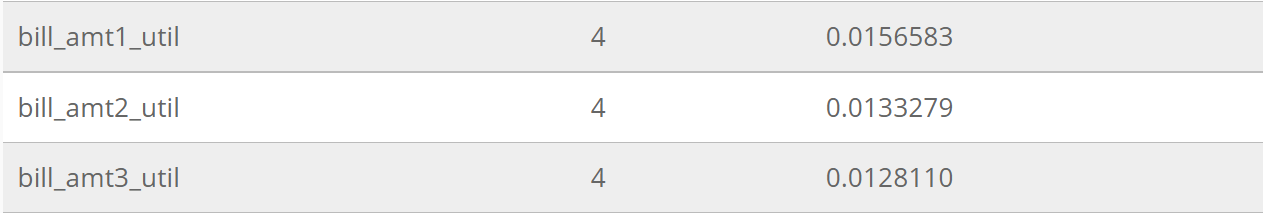
2.3 VARIABLE CLUSTERING





2.4 VARIABLE REDUCTION





APPENDIX

1. Abstract
2. Introduction
3. R Packages Used
4. Data Set
5. Code
6. Data
   1. Import Data From R
   2. Data Exploration
   3. Data Manipulation
   4. Variable Generation
   5. Sampling
7. Univariate Analysis
   1. Fine Classing
   2. Initial Variable Removal
   3. Variable Clustering
   4. Variable Reduction
8. Coarse Classing
9. Regression Analysis
10. Scorecard Creation, Scaling And Validation
11. Plots

1. Variabe Clustering

2. Coarse Classing

3. Scorecard Creation, Scaling And Validation

1. Tables

1. Fine Classing

2. Variable Clustering

3. Variable Reduction

CONCLUSION

In this project we develop credit scorecards with ease using the scorecard package available in R programming language. We learn various aspects of R language like data exploration, data manipulation, variable generation, univariate analysis ( variable clustering & reduction) and regression analysis through the course of the project.

Consumers, purchasers and business ventures alike are highly dependent on credit in the present day. The entire system of the availability of credit, the time period of the acceptance of an offer is dependent on your credit score.

REFERENCES

1. Data set- <https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients>
2. <https://rstudio.github.io/rstudio-extensions/index.html>
3. [https://www.upgrad.com/blog/rstudio-projects-for-beginners](https://www.upgrad.com/blog/rstudio-projects-for-beginners/)
4. [https://stackoverflow.com/questions](https://stackoverflow.com/questions/)
5. <https://aip.scitation.org/doi/abs/10.1063/1.5112333>
6. [https://www.kaggle.com](https://www.kaggle.com/)