Основи прикладного математичного моделювання в R.

Дерева рішень. Регресія

Юрій Клебан, IntelSoft Tech

# Вступ

Даний матеріал містить інформацію у межах курсу “Основи прикладного математичного моделювання в R”. Курс створено для працівників банків. У даному навчальному матеріалі використано дані про баланси кредитних карт.

Джерела:

* An Introduction to Statistical Learning with Applications in R (<http://www-bcf.usc.edu/~gareth/ISL/data.html>)
* Predicting Credit Card Balance using Regression (<https://www.kaggle.com/suzanaiacob/predicting-credit-card-balance-using-regression/notebook>)

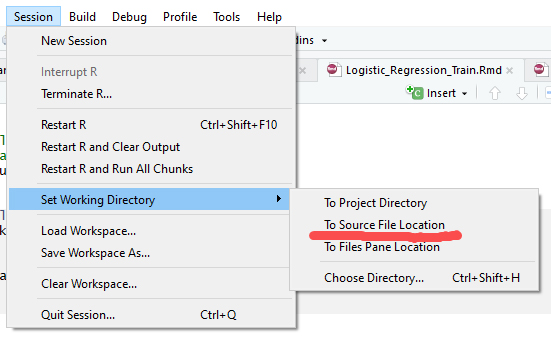
Увага! Частина значень у Вашому дослідженні може відрізнятися, адже це залежить від сформованих тренувальної та тестової вибірок.

Для виконнання завдань потрібний ряд R-пакетів:

library(ggplot2) # побудова графіків  
 library(gmodels) # побудова крос-таблиць  
 library(rpart) # дерево рішень  
 library(partykit) # дерево рішень  
 library(rpart.plot) # візуалізація дерева рішень  
 library(DMwR) # оцінка похибок моделі  
 library(rattle) # допомога у візуалізації   
 library(RColorBrewer) # допомога у візуалізації

Якщо ці пакети відсутні, то інсталюйте їх за допомогою команди install.packages(назва\_пакету).

# Етап 1. Попередній огляд даних та створення тренувальної і тестової вибірки

Для початку варто встановити робочу папку. Це можна зробити за допомогою команди setwd(шлях\_до\_каталогу) або через меню *R Studio*: 

Для очистки сесії від непотрібних даних використайте команду rm:

rm(list = ls()) #видаляє усі змінні

Імпорт даних:

data <- read.csv("credit\_card\_balance.csv")

Переглянемо структуру даних:

str(data)

## 'data.frame': 400 obs. of 12 variables:  
## $ X : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Income : num 14.9 106 104.6 148.9 55.9 ...  
## $ Limit : int 3606 6645 7075 9504 4897 8047 3388 7114 3300 6819 ...  
## $ Rating : int 283 483 514 681 357 569 259 512 266 491 ...  
## $ Cards : int 2 3 4 3 2 4 2 2 5 3 ...  
## $ Age : int 34 82 71 36 68 77 37 87 66 41 ...  
## $ Education: int 11 15 11 11 16 10 12 9 13 19 ...  
## $ Gender : Factor w/ 2 levels "Female","Male": 2 1 2 1 2 2 1 2 1 1 ...  
## $ Student : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 2 ...  
## $ Married : Factor w/ 2 levels "No","Yes": 2 2 1 1 2 1 1 1 1 2 ...  
## $ Ethnicity: Factor w/ 3 levels "African American",..: 3 2 2 2 3 3 1 2 3 1 ...  
## $ Balance : int 333 903 580 964 331 1151 203 872 279 1350 ...

head(data)

## X Income Limit Rating Cards Age Education Gender Student Married  
## 1 1 14.891 3606 283 2 34 11 Male No Yes  
## 2 2 106.025 6645 483 3 82 15 Female Yes Yes  
## 3 3 104.593 7075 514 4 71 11 Male No No  
## 4 4 148.924 9504 681 3 36 11 Female No No  
## 5 5 55.882 4897 357 2 68 16 Male No Yes  
## 6 6 80.180 8047 569 4 77 10 Male No No  
## Ethnicity Balance  
## 1 Caucasian 333  
## 2 Asian 903  
## 3 Asian 580  
## 4 Asian 964  
## 5 Caucasian 331  
## 6 Caucasian 1151

Дані складають з 400 спостежень та наступник пкоазників:

* *ID* - ідентифікатор;
* *Income* - дохід у $10,0000;
* *Limit* - кредитний ліміт;
* *Rating* - кредитний рейтинг;
* *Age* - вік, роки;
* *Education* - освіта, кількість років навчання;
* *Gender* - стать (Male or Female);
* *Student* - флаг чи є студентом (Yes or No);
* *Married* - флаг чи одружений (Yes or No);
* *Ethnicity* - етнічна належність (African American, Asian or Caucasian);
* *Balance* - середній баланс по карті у $.

Задача: визначити вплив факторів на середній баланс по карті.

Оглянемо описову статистику факторів:

summary(data)

## X Income Limit Rating   
## Min. : 1.0 Min. : 10.35 Min. : 855 Min. : 93.0   
## 1st Qu.:100.8 1st Qu.: 21.01 1st Qu.: 3088 1st Qu.:247.2   
## Median :200.5 Median : 33.12 Median : 4622 Median :344.0   
## Mean :200.5 Mean : 45.22 Mean : 4736 Mean :354.9   
## 3rd Qu.:300.2 3rd Qu.: 57.47 3rd Qu.: 5873 3rd Qu.:437.2   
## Max. :400.0 Max. :186.63 Max. :13913 Max. :982.0   
## Cards Age Education Gender Student   
## Min. :1.000 Min. :23.00 Min. : 5.00 Female:207 No :360   
## 1st Qu.:2.000 1st Qu.:41.75 1st Qu.:11.00 Male :193 Yes: 40   
## Median :3.000 Median :56.00 Median :14.00   
## Mean :2.958 Mean :55.67 Mean :13.45   
## 3rd Qu.:4.000 3rd Qu.:70.00 3rd Qu.:16.00   
## Max. :9.000 Max. :98.00 Max. :20.00   
## Married Ethnicity Balance   
## No :155 African American: 99 Min. : 0.00   
## Yes:245 Asian :102 1st Qu.: 68.75   
## Caucasian :199 Median : 459.50   
## Mean : 520.01   
## 3rd Qu.: 863.00   
## Max. :1999.00

Підготуємо дані до моделювання. Перетворимо категоріальні показники до факторів:

data$X <- NULL  
data$Gender <- factor(data$Gender)  
data$Student <- factor(data$Student)  
data$Married <- factor(data$Married)  
data$Ethnicity <- factor(data$Ethnicity)

Розділимо загальну вибірку на 2 частини: \* тренувальна, 70% вибірки, для побудови регресії; \* тестова, 30% вибірки, для перевірки точності моделі.

set.seed(2019) #довільне число як точка "відправки" для генератора випадкових чисел  
  
# Згенеруємо набір чисел від 1 до кількості спостережень у вибірці і відберемо випадквоим чином 70% із них  
train\_index <- sample(1:nrow(data), size = 0.7\*nrow(data))  
  
#Запишемо по номерах відібраних рядків тренувальний набір даних  
train\_data <- data[train\_index,]  
  
#Всі інші значення, що не увійшли в тренувальну вибірку запишемо у тестову  
test\_data <- data[-train\_index,]

Переглянемо наявність зв’язків між числовими параметрами для тренувальної вибірки за допомогою матриці попарних кореляцій/ Дані на перетині рядків вказують на кореляцію між вибраними показниками.

cor(train\_data[, -c(7:10)])

## Income Limit Rating Cards Age  
## Income 1.000000000 0.80719762 0.80826490 0.008646931 0.15216962  
## Limit 0.807197620 1.00000000 0.99686783 0.040912602 0.12353888  
## Rating 0.808264901 0.99686783 1.00000000 0.082664078 0.12687122  
## Cards 0.008646931 0.04091260 0.08266408 1.000000000 0.02396783  
## Age 0.152169617 0.12353888 0.12687122 0.023967831 1.00000000  
## Education 0.021712979 0.03487428 0.02793893 -0.038762817 0.05471784  
## Balance 0.502586712 0.87232448 0.87359653 0.126673137 0.05672078  
## Education Balance  
## Income 0.02171298 0.50258671  
## Limit 0.03487428 0.87232448  
## Rating 0.02793893 0.87359653  
## Cards -0.03876282 0.12667314  
## Age 0.05471784 0.05672078  
## Education 1.00000000 0.05155195  
## Balance 0.05155195 1.00000000

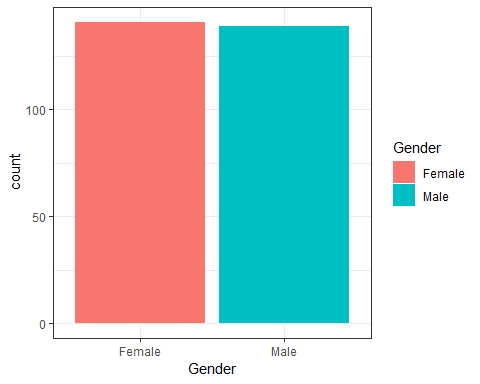
###Вивчимо показники перед моделюванням

Для початку переглянемо категоріальні змінні.

# Потбірні пакети  
 # library(ggplot2)  
 # library(gmodels)

Стать (Gender):

ggplot(train\_data, aes(Gender)) + geom\_bar(aes(fill = Gender)) + theme\_bw()

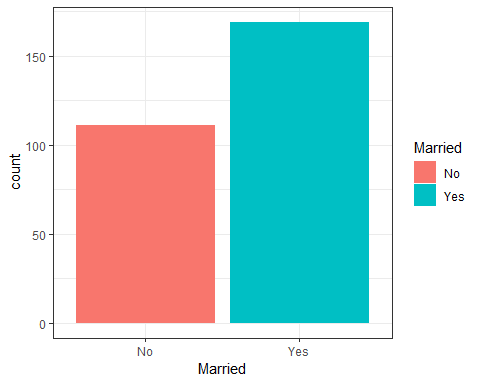


CrossTable(train\_data$Gender)

## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 280   
##   
##   
## | Female | Male |   
## |-----------|-----------|  
## | 141 | 139 |   
## | 0.504 | 0.496 |   
## |-----------|-----------|

Сімейний стан (Married):

ggplot(train\_data, aes(Married)) + geom\_bar(aes(fill = Married)) + theme\_bw()

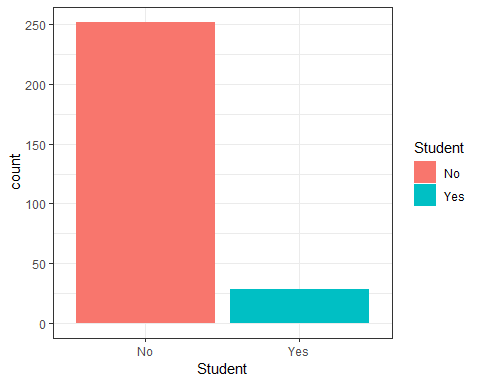


CrossTable(train\_data$Married)

## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 280   
##   
##   
## | No | Yes |   
## |-----------|-----------|  
## | 111 | 169 |   
## | 0.396 | 0.604 |   
## |-----------|-----------|

Студент:

ggplot(train\_data, aes(Student)) + geom\_bar(aes(fill = Student)) + theme\_bw()

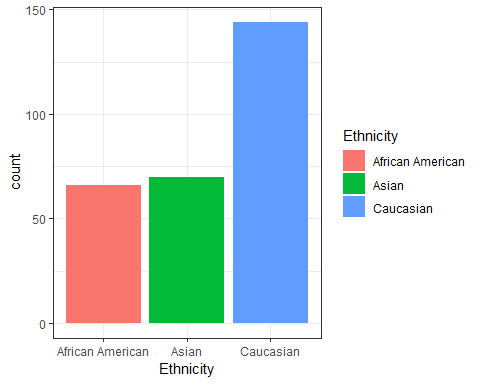


CrossTable(train\_data$Student)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 280   
##   
##   
## | No | Yes |   
## |-----------|-----------|  
## | 252 | 28 |   
## | 0.900 | 0.100 |   
## |-----------|-----------|  
##   
##   
##   
##

Етнічна належність:

ggplot(train\_data, aes(Ethnicity)) + geom\_bar(aes(fill = Ethnicity)) + theme\_bw()



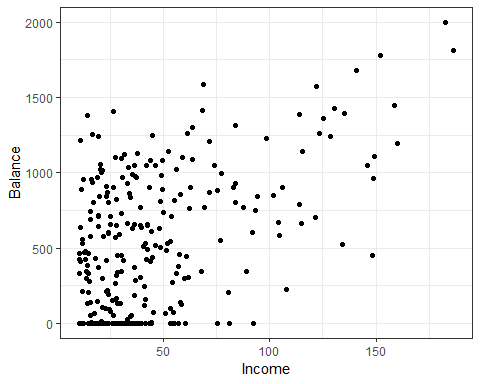
CrossTable(train\_data$Ethnicity)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 280   
##   
##   
## | African American | Asian | Caucasian |   
## |------------------|------------------|------------------|  
## | 66 | 70 | 144 |   
## | 0.236 | 0.250 | 0.514 |   
## |------------------|------------------|------------------|  
##   
##   
##   
##

Порівняємо числові змінні з показником середнього балансу.

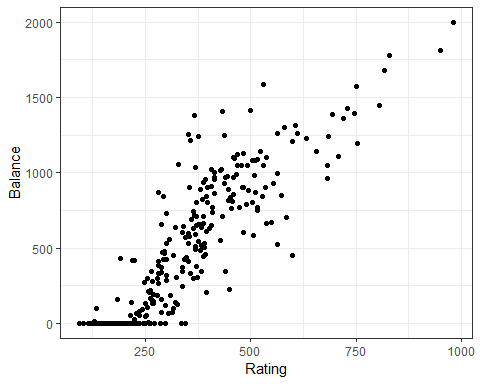
Графік залежності між доходом (Income) та середнім балансом по карті (Balance):

ggplot(train\_data, aes(Income, Balance)) + geom\_point() + theme\_bw()



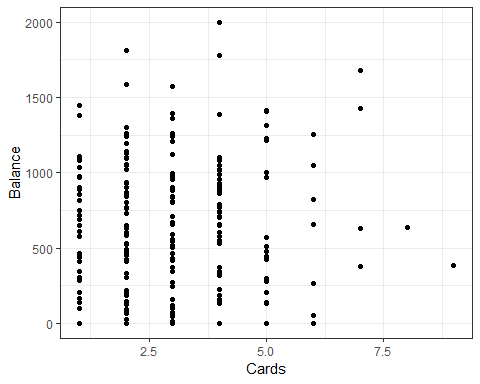
Графік залежності між доходом (Rating) та середнім балансом по карті (Balance):

ggplot(train\_data, aes(Rating, Balance)) + geom\_point() + theme\_bw()



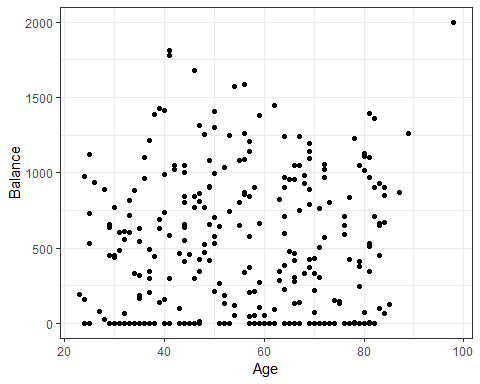
Графік залежності між кількістю карт (Cards) та середнім балансом по карті (Balance):

ggplot(train\_data, aes(Cards, Balance)) + geom\_point() + theme\_bw()



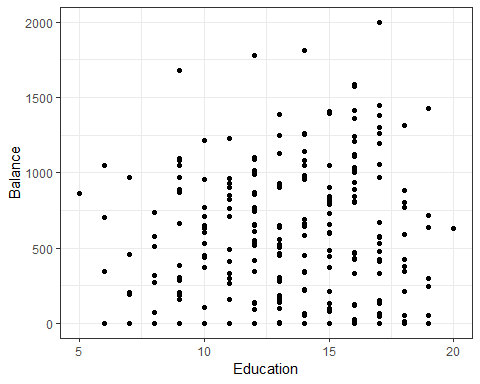
Графік залежності між віком (Age) та середнім балансом по карті (Balance):

ggplot(train\_data, aes(Age, Balance)) + geom\_point() + theme\_bw()



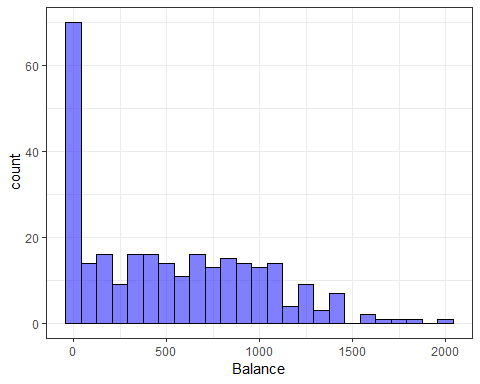
Графік залежності між роками навчання (Education) та середнім балансом по карті (Balance):

ggplot(train\_data, aes(Education, Balance)) + geom\_point() + theme\_bw()



Графік Частоти значень балансу:

ggplot(train\_data, aes(Balance)) + geom\_histogram(bins = 25, alpha = 0.5, fill = 'blue', color='black') + theme\_bw()



# Етап 3. Робота з пакетом RPart

Для побудови регресії на основі дерева рішень інсталюємо пакет rpart (install.package(rpart)).

Використаємо функцію для побудови моделі rpart().

#library(rpart)  
rpart\_model <- rpart(Balance ~ ., train\_data)

Створимо дата-фрейми для запису результатів моделювання на тестовій та тренувальній вибірках:

train\_res <- data.frame(No = c(1:nrow(train\_data)), Balance = train\_data$Balance, RPartPredictedDef = predict(rpart\_model, train\_data))  
  
test\_res <- data.frame(No = c(1:nrow(test\_data)),Balance = test\_data$Balance, RPartPredictedDef = predict(rpart\_model, test\_data))  
  
head(train\_res)

## No Balance RPartPredictedDef  
## 308 1 0 57.70000  
## 285 2 0 57.70833  
## 121 3 0 57.70833  
## 246 4 0 57.70833  
## 20 5 1048 986.40678  
## 18 6 368 459.53659

Переглянемо похибки моделі на обох вибірках:

#library(DMwR)  
regr.eval(train\_res$Balance, train\_res$RPartPredictedDef)

## mae mse rmse mape   
## 122.9285 28795.2733 169.6917 Inf

regr.eval(test\_res$Balance, test\_res$RPartPredictedDef)

## mae mse rmse mape   
## 162.0100 49397.0116 222.2544 Inf

Оцінимо точність моделі за допомогою коефіцієнта детермінації:

r\_train <- cor(train\_res$Balance, train\_res$RPartPredictedDef)^2  
r\_test <- cor(test\_res$Balance, test\_res$RPartPredictedDef)^2  
  
print(paste("R\_train =", r\_train, sep = ""))

## [1] "R\_train =0.865233862708679"

print(paste("R\_test =", r\_test, sep = ""))

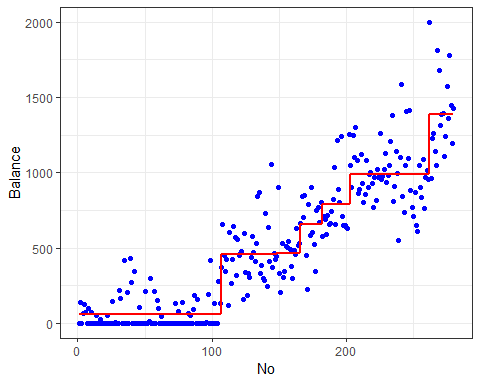
## [1] "R\_test =0.763227042391921"

Для покращення візуалізації відсортуємо результати:

# відсортуємо за зростанням значення балансів карт  
ordered\_train\_res <- train\_res[order(train\_res$Balance),]  
# відсортуємо за зростанням значення модельованого значення балансу  
ordered\_train\_res <- train\_res[order(train\_res$RPartPredictedDef),]  
# "Перепишемо" номери по порядку  
ordered\_train\_res$No <- c(1:nrow(train\_res))

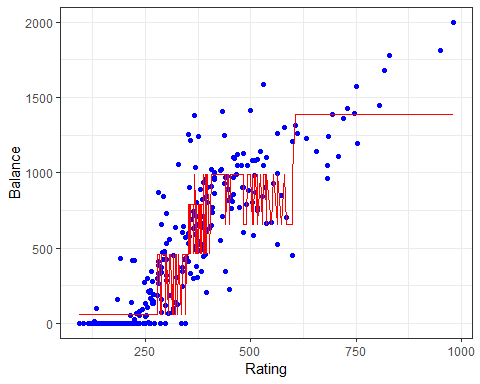
Побудуємо графік модельованих та реальних значень балансу з відсортованими показниками для наочності:

ggplot(ordered\_train\_res) +  
 geom\_point(aes(x = No, y = Balance), colour = "blue") +  
 geom\_line(aes(x = No, y = RPartPredictedDef), colour = "red", size = 1) + theme\_bw()



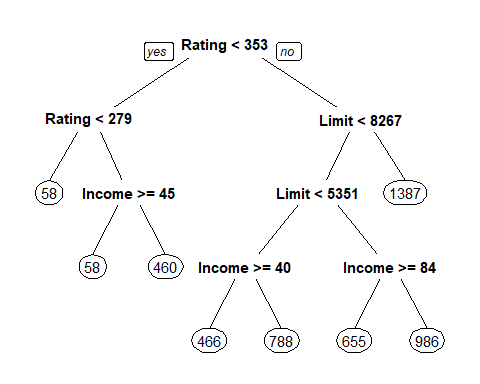
Переглянемо залежність між рейтингом та балансом клієнта, а також прогнозованих значеннях.

ggplot(train\_data) +  
 geom\_point(aes(x = Rating, y = Balance), colour = "blue") +  
 geom\_line(aes(x = Rating, y = train\_res$RPartPredictedDef), colour = "red") + theme\_bw()

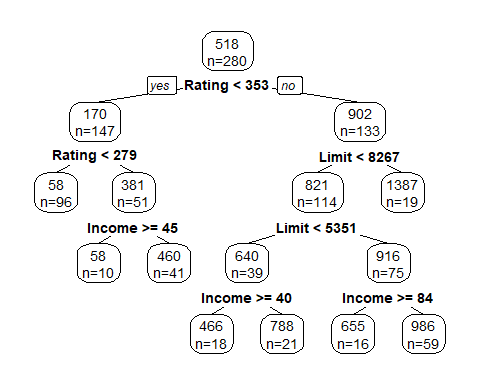


Для візуалізації дерева рішень скористаємося пакетом rpart.plot:

#library(rpart.plot)  
prp(rpart\_model)



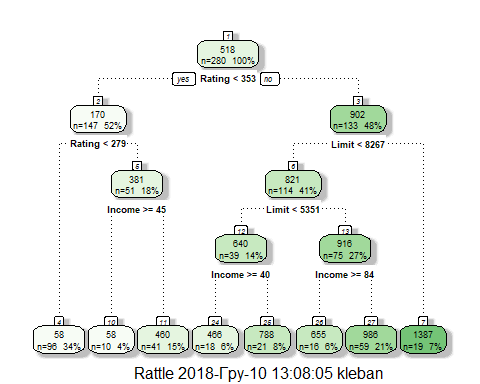
prp(rpart\_model, extra = 1, type = 2)



Додамо інтерактивності для побудованого дерева рішень: {r prp(rpart\_model, snip = TRUE).

Побудуємо також “розфарбоване” дерево рішень за допомогою пакетів rattle та RColorBrewer:

#library(rattle)  
#library(RColorBrewer)  
fancyRpartPlot(rpart\_model)



Побудуємо модель з вказанням мінімальної кількості розділень даних (minsplit):

rpart\_model2 <- rpart(Balance ~ ., train\_data, control = rpart.control(minsplit = 10))

Отримаємо прогнозовані значення для обох вибірок на основі другої моделі:

train\_res$RPartPredicted10 <- predict(rpart\_model2, train\_data)  
test\_res$RPartPredicted10 <- predict(rpart\_model2, test\_data)

Переглянемо похибки:

regr.eval(train\_res$Balance, train\_res$RPartPredicted10)

## mae mse rmse mape   
## 104.9391 18340.9943 135.4289 Inf

regr.eval(test\_res$Balance, test\_res$RPartPredicted10)

## mae mse rmse mape   
## 162.0908 47198.7466 217.2527 Inf

Оцінимо точність моделі за допомогою коефіцієнта детермінації:

r\_train <- cor(train\_res$Balance, train\_res$RPartPredicted10)^2  
r\_test <- cor(test\_res$Balance, test\_res$RPartPredicted10)^2  
  
print(paste("R\_train =", r\_train, sep = ""))

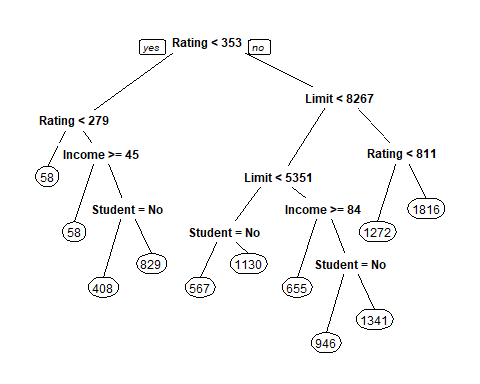
## [1] "R\_train =0.914161434483023"

print(paste("R\_test =", r\_test, sep = ""))

## [1] "R\_test =0.774778882920703"

Візуалізуємо модель:

prp(rpart\_model2)



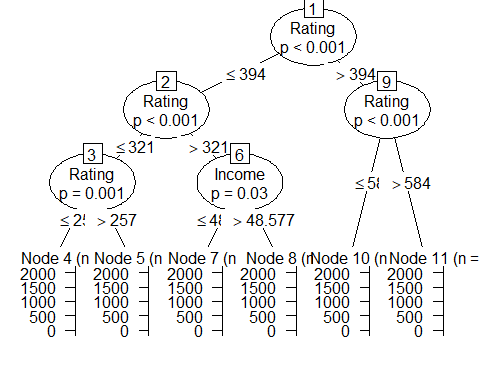
# Етап 4. Пакет partykit.

Побудуємо дерево рішень на основі partykit та функції ctree() та порівняємо з результатами роботи rpart().

#library(partykit)  
party\_model <- ctree(Balance ~ ., data = train\_data[1:100,])  
print(party\_model)

##   
## Model formula:  
## Balance ~ Income + Limit + Rating + Cards + Age + Education +   
## Gender + Student + Married + Ethnicity  
##   
## Fitted party:  
## [1] root  
## | [2] Rating <= 394  
## | | [3] Rating <= 321  
## | | | [4] Rating <= 257: 46.206 (n = 34, err = 419211.6)  
## | | | [5] Rating > 257: 213.200 (n = 15, err = 493792.4)  
## | | [6] Rating > 321  
## | | | [7] Income <= 48.577: 589.643 (n = 14, err = 296561.2)  
## | | | [8] Income > 48.577: 385.286 (n = 7, err = 93181.4)  
## | [9] Rating > 394  
## | | [10] Rating <= 584: 953.429 (n = 21, err = 544837.1)  
## | | [11] Rating > 584: 1382.111 (n = 9, err = 1036828.9)  
##   
## Number of inner nodes: 5  
## Number of terminal nodes: 6

plot(party\_model)



Обчислимо модельовані значення середнього балансу по карті:

train\_res$PredictedPartyDef <- predict(party\_model, train\_data)  
test\_res$PredictedPartyDef <- predict(party\_model, test\_data)

Переглянемо коефіцієнти детермінації:

r\_train <- cor(train\_res$Balance, train\_res$PredictedPartyDef)^2  
r\_test <- cor(test\_res$Balance, test\_res$PredictedPartyDef)^2  
  
print(paste("R\_train =", r\_train, sep = ""))

## [1] "R\_train =0.780580225804902"

print(paste("R\_test =", r\_test, sep = ""))

## [1] "R\_test =0.699772975746937"