

SPRAWOZDANIE

Zajęcia: Analiza Procesów Ucznia

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium: 5.1

Data: 29.05.2020

Temat: „Modelowanie procesów uczenia maszynowego w pakiecie MLR
Trenowanie, ocena i porównanie modeli w pakiecie MLR”

Wariant: 1, zadanie 2

Jacek Adamczyk

Informatyka II stopień,

Stacjonarne,

1 semestr,

Gr. A

<https://github.com/jacekaGIT/ATH-1g>

1. Polecenia dla wariantu 1 (Zadanie 1):

Celem ćwiczenia jest ocena i wybór najlepszej metody prognozowania oceny klientów na bazie zbioru danych opracowanego na Zajęciu 1.

Polecenie:

Zadanie 2. Zadanie dotyczy prognozowania oceny klientów (w skali 5-punktowej, Error < 5%) urządzeń RTV AGD, określonych na Zajęciu 1. Rozwiązanie polega na użyciu pakietu mlr. Należy wybrać najlepszą metodę wśród 5 możliwych z punktu widzenia precyzyjności. Wyniki porównywania precyzyjności metod należy przedstawić w postaci graficznej.

2. Skrypt:

Po uruchomieniu skryptu zostaną wykonane kolejno wszystkie polecenia zadania. Instalacja pakietów została wyłączona (ustawiona jako komentarz) żeby niepotrzebnie nie instalować pakietów przy każdym uruchomieniu skryptu.

```
#..... Jacek Adamczyk, sem 1, II st .....
#..... Lab 5, gr A, wariant 1 .....
# Modelowanie procesów uczenia maszynowego ...
# w pakiecie MLR. Trenowanie, ocena .....
# i porównywanie modeli w pakiecie MLR .....
#.....

# Zadanie 2

#install.packages("mlr")
#install.packages("rFems")
#install.packages("randomForestSRC")

library("mlr")
library("rFems")
library("randomForestSRC")

# Funkcja normalizująca dane w zakr min do max
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

setwd("C:/Users/jacek/Dysk Google/ATH 1/APU/Lab5/Zad2")
smartfony <- read.csv("JacekAdamczyk_APU_Lab1.csv")
smartfony
head(smartfony)

smartfony$ocena <- factor(smartfony$ocena)
smartfony$ocena

smartfony <- smartfony[,-6]
smartfony <- smartfony[,-1]
```

```

smartfony <- smartfony[,-5]
smartfony <- smartfony[,-7]
smartfony <- smartfony[,-5]
smartfony

smartfony$wyswietlacz <- normalize(smartfony$wyswietlacz)
smartfony$pamiec_RAM <- normalize(smartfony$pamiec_RAM)
smartfony$pamiec_wbudowana <- normalize(smartfony$pamiec_wbudowana)
smartfony <- data.frame(smartfony)

smartfony

zadanie = makeClassifTask(id = deparse(substitute(smartfony)),
                          smartfony,
                          "ocena",
                          weights = NULL,
                          blocking = NULL,
                          coordinates = NULL,
                          positive = NA_character_,
                          fixup.data = "warn",
                          check.data = TRUE)

ponowne_probkowanie = makeResampleDesc("CV", iters = 5)

metody_uczenia <- makeLearners(c("rpart",
                                "C50",
                                "rFerns",
                                "randomForestSRC"),
                              type = "classif")

porownnanie_metod <- benchmark(learners = metody_uczenia,
                              tasks = zadanie,
                              resamplings = ponowne_probkowanie)

porownnanie_metod
porownnanie_metod$results

plotBMRBoxplots(porownnanie_metod, measure = mmce,
                order.lrn = getBMRLearnerIds(porownnanie_metod))
plotBMRSummary(porownnanie_metod)
plotBMRRanksAsBarChart(porownnanie_metod, pos = "dodge",
                      order.lrn = getBMRLearnerIds(porownnanie_metod))

```

3. Wyniki działania (Test 1 – 5 iteracji):

Poniżej przedstawiono wyniki działania skryptu. Pełne wydruki z konsoli można znaleźć w repozytorium GitHub (adres na stronie tytułowej).

- Dane po przygotowaniu (usunięcie kolumn, normalizacja):

```
> smartfony
      nazwy  wyswietlacz  pamiec_RAM  pamiec_wbudowana  ocena
1 Galaxy A Quantum    0.9714286      0.6      0.4285714      5
2   Galaxy A21S      0.8571429      0.2      0.1428571      4
3   Galaxy A41      0.6285714      0.2      0.1428571      4
4   Galaxy S20      0.9714286      1.0      0.4285714     3.5
5   Galaxy A71      0.9714286      0.4      0.4285714      3
6   Galaxy M21      0.8000000      0.4      0.1428571     3.5
7   Galaxy Z Flip      0.9714286      0.6      1.0000000      4
8 Galaxy S10 Lite      0.9714286      0.6      0.4285714      5
9   Galaxy A51      0.8571429      0.2      0.4285714     4.5
10  Galaxy Note10     1.0000000      1.0      1.0000000      5
11 Galaxy Xcover4S     0.0000000      0.1      0.0000000     4.5
12 Galaxy XcoverPro    0.7428571      0.2      0.1428571      5
13   Galaxy A10      0.6857143      0.0      0.0000000      4
14   Galaxy A80      0.9714286      0.6      0.4285714     4.5
15   Galaxy A40      0.4000000      0.2      0.1428571      5
>
```

- Wyniki porównania metod:

```
> porownnanie_metod <- benchmark(learners = metody_uczenia,
+                               tasks = zadanie,
+                               resamplings = ponowne_probkowanie)
Task: smartfony, Learner: classif.rpart
Resampling: cross-validation
Measures:      mmce
[Resample] iter 1: 0.6666667
[Resample] iter 2: 1.0000000
[Resample] iter 3: 0.6666667
[Resample] iter 4: 1.0000000
[Resample] iter 5: 1.0000000
```

Aggregated Result: mmce.test.mean=0.8666667

```
Task: smartfony, Learner: classif.C50
Resampling: cross-validation
Measures:      mmce
[Resample] iter 1: 1.0000000
[Resample] iter 2: 1.0000000
[Resample] iter 3: 0.6666667
[Resample] iter 4: 1.0000000
[Resample] iter 5: 1.0000000
```

Aggregated Result: mmce.test.mean=0.9333333

```
Task: smartfony, Learner: classif.rFerns
```

```

Resampling: cross-validation
Measures:          mmce
[Resample] iter 1:  1.0000000
[Resample] iter 2:  1.0000000
[Resample] iter 3:  0.6666667
[Resample] iter 4:  0.3333333
[Resample] iter 5:  1.0000000

```

Aggregated Result: mmce.test.mean=0.8000000

```

Task: smartfony, Learner: classif.randomForestSRC
Resampling: cross-validation
Measures:          mmce
[Resample] iter 1:  1.0000000
[Resample] iter 2:  1.0000000
[Resample] iter 3:  1.0000000
[Resample] iter 4:  1.0000000
[Resample] iter 5:  1.0000000

```

Aggregated Result: mmce.test.mean=1.0000000

- Wyniki porównania metod 2:

```

> porownnanie_metod
  task.id          learner.id mmce.test.mean
1 smartfony      classif.rpart    0.8666667
2 smartfony      classif.C50      0.9333333
3 smartfony      classif.rFerns    0.8000000
4 smartfony classif.randomForestSRC 1.0000000
> porownnanie_metod$results
$smartfony
$smartfony$classif.rpart
Resample Result
Task: smartfony
Learner: classif.rpart
Aggr perf: mmce.test.mean=0.8666667
Runtime: 0.0499969

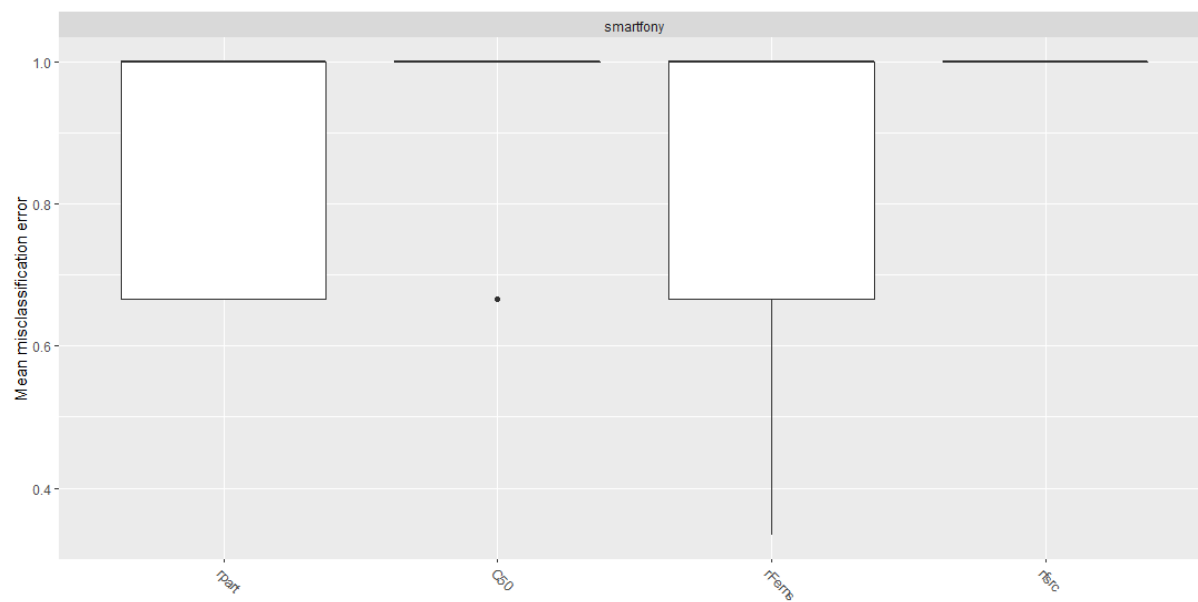
$smartfony$classif.C50
Resample Result
Task: smartfony
Learner: classif.C50
Aggr perf: mmce.test.mean=0.9333333
Runtime: 0.0550001

$smartfony$classif.rFerns
Resample Result
Task: smartfony
Learner: classif.rFerns
Aggr perf: mmce.test.mean=0.8000000
Runtime: 0.0469992

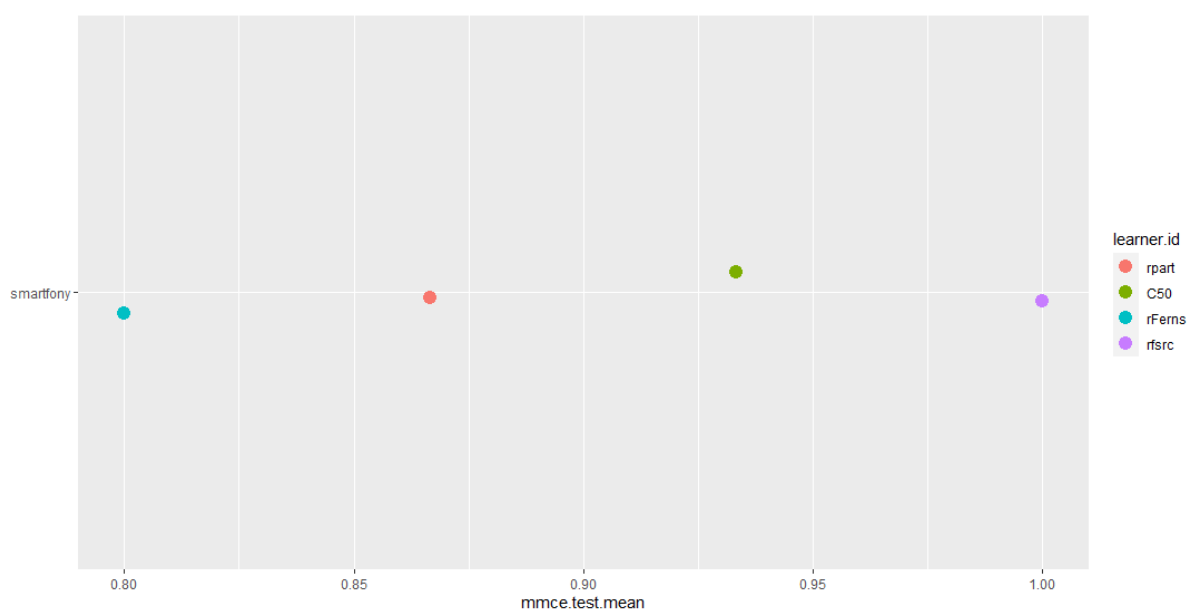
$smartfony$classif.randomForestSRC
Resample Result
Task: smartfony
Learner: classif.randomForestSRC
Aggr perf: mmce.test.mean=1.0000000
Runtime: 0.34

```

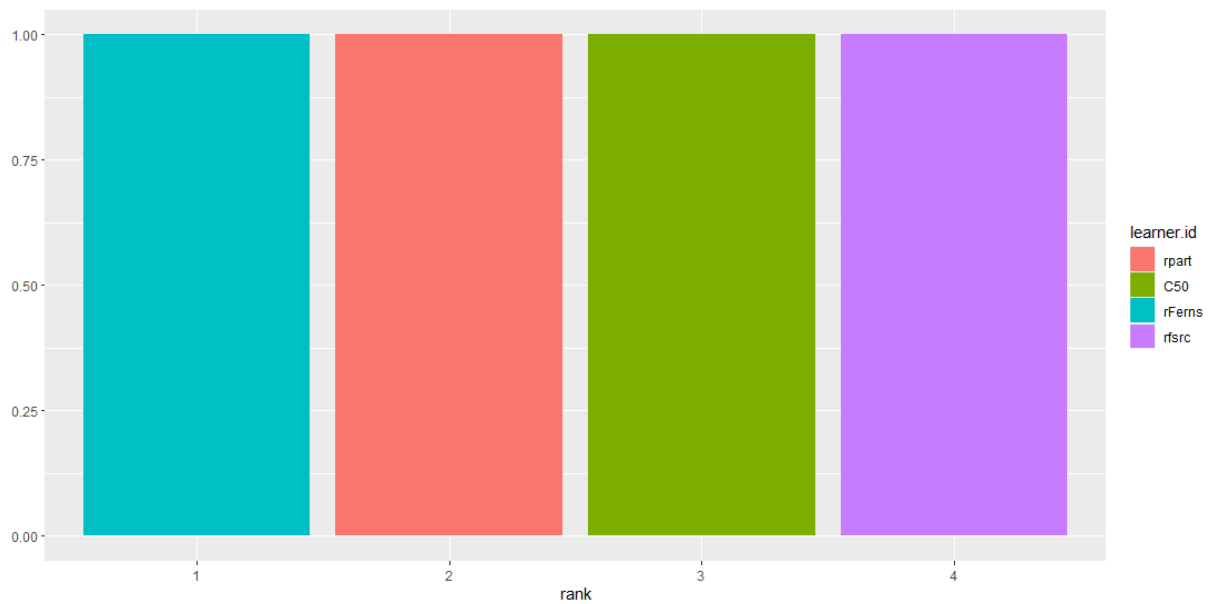
- Reprezentacja graficzna:



Rysunek 1. BMRBoxplots



Rysunek 1. BMRSummary

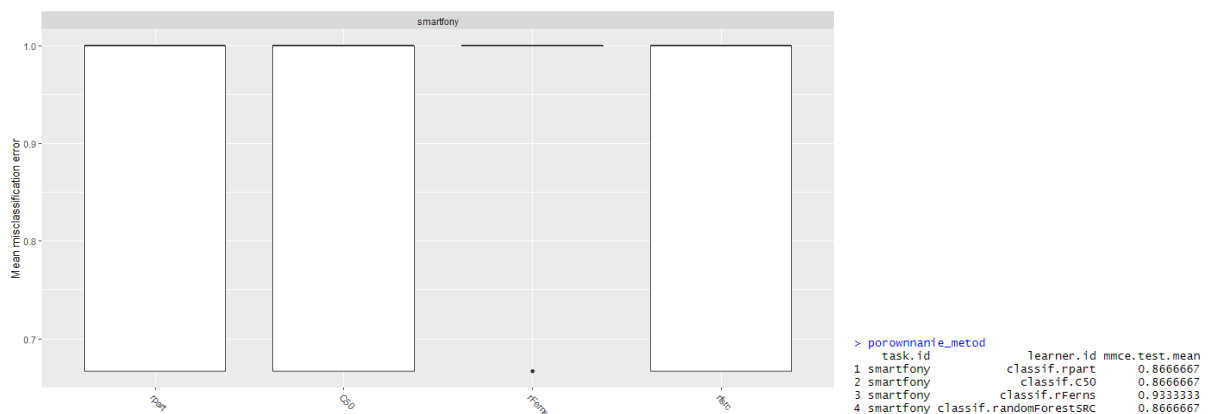


Rysunek 2. *BMRRanksAsBarChart*

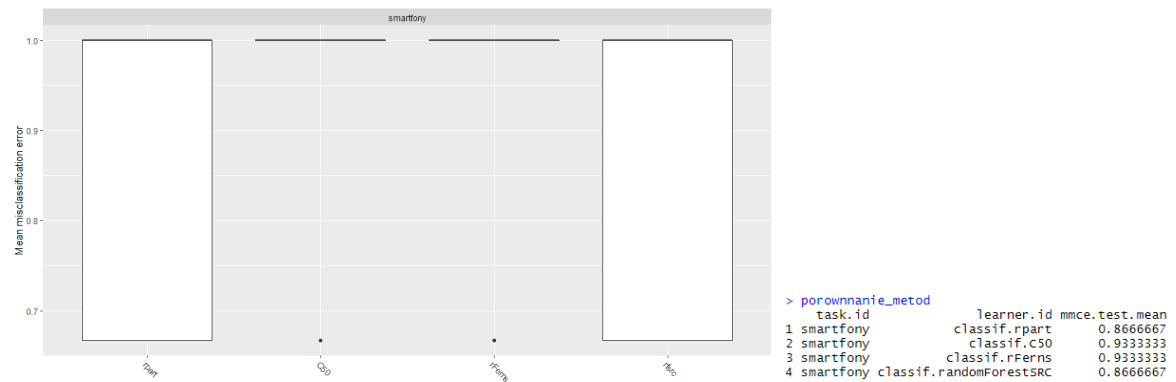
4. Wnioski:

Podczas testów wykonano kilkadziesiąt prób. Z początku za każdym razem wyniki poszczególnych prób różniły się dosyć znacznie i wyglądały na dosyć przypadkowe. Po wprowadzeniu parametru **iters** z różnymi wartościami sytuacja zaczęła się poprawiać. Przy wartości **iters** = 5 wyniki parametru **mmce.test.mean** zaczęły być powtarzalne i mieściły się w granicach od 0,8 do 1 (testy 1 do 4). Jednak nie było widać wyraźnych zwycięzców. Sytuacja znacznie się poprawiła po wprowadzeniu **iters** = 10 (testy 5 do 8). Wynik **mmce.test.mean** zaczął być zazwyczaj $\geq 0,9$, a metody **rpart** i **C50** uzyskiwały najlepsze rezultaty. Poniżej przedstawiono wyniki i wykresy z porównania metod dla testów 2-8. Metoda **rpart** ostatecznie daje najlepsze wyniki i zwycięża w rankingu.

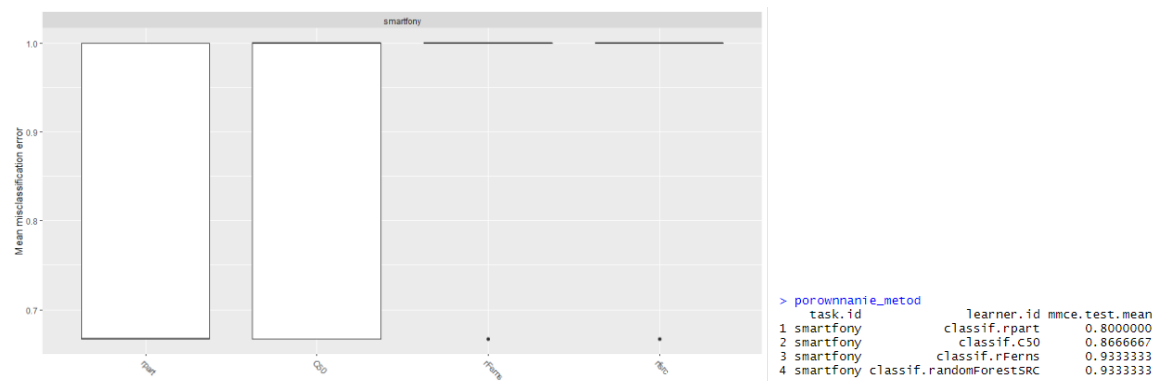
- Test 2 – 5 iteracji:



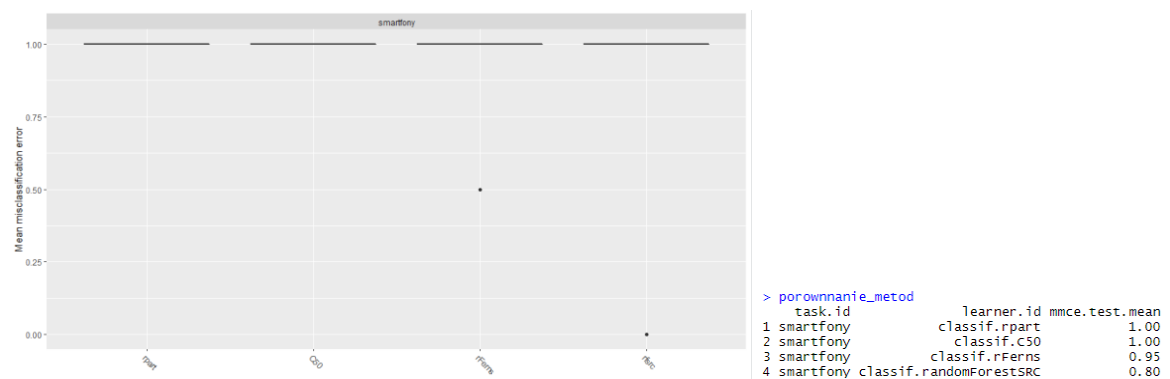
- Test 3 – 5 iteracji:



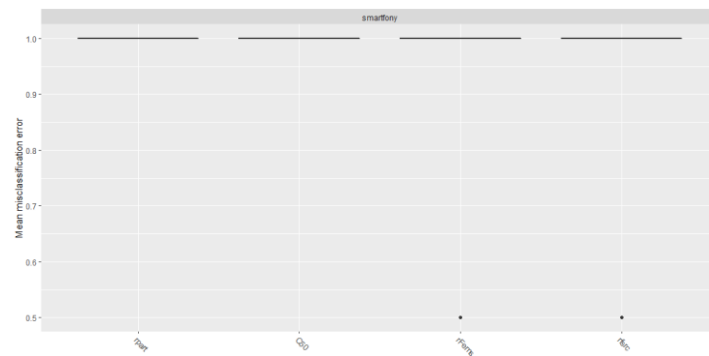
- Test 4 – 5 iteracji:



- Test 5 – zmiana liczby iteracji z 5 na 10:

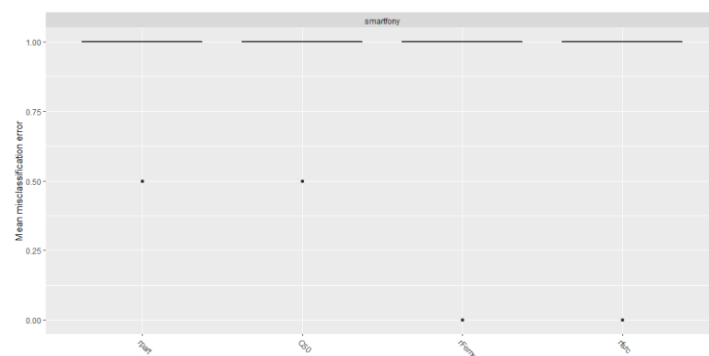


- Test 6 – zmiana liczby iteracji z 5 na 10:



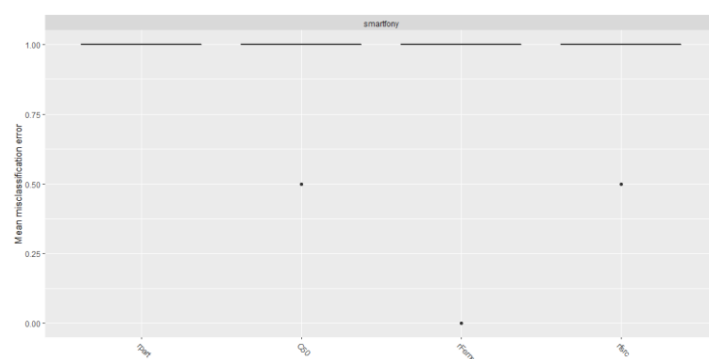
```
> porownnanie_metod
  task.id learner.id mmce.test.mean
1 smartfony   classif.rpart      1.00
2 smartfony   classif.c50      1.00
3 smartfony   classif.rFems      0.95
4 smartfony classif.randomForestSRC 0.95
```

- Test 7 – zmiana liczby iteracji z 5 na 10:



```
> porownnanie_metod
  task.id learner.id mmce.test.mean
1 smartfony   classif.rpart      0.95
2 smartfony   classif.c50      0.95
3 smartfony   classif.rFems      0.90
4 smartfony classif.randomForestSRC 0.90
```

- Test 8 – zmiana liczby iteracji z 5 na 10:



```
> porownnanie_metod
  task.id learner.id mmce.test.mean
1 smartfony   classif.rpart      1.00
2 smartfony   classif.c50      0.95
3 smartfony   classif.rFems      0.90
4 smartfony classif.randomForestSRC 0.95
```

End Lab