

SPRAWOZDANIE

Zajęcia: Analiza Procesów Ucznienia

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium 5

Data 28.04.2023

Temat: Modelowanie procesów uczenia maszynowego w pakiecie mlr.

Wariant 1

Rafał Klinowski
Informatyka II stopień,
Stacjonarne,
1 semestr,
Gr. a

1. Polecenie: Wariant 1

Zadanie 1. Zadanie dotyczy konstruowania drzew decyzyjnych oraz reguł klasyfikacyjnych na podstawie zbioru danych (library(MASS lub datasets)).
Warianty zadania

1. iris

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. Wprowadzane dane:

Dane do zadania 2. zostały zaczerpnięte z pierwszego z laboratoriów (plik CSV zawierający opis 15 smartfonów).

Ponadto wykorzystano dataset 'iris' zawarty w pakiecie 'datasets'.

3. Wykorzystane komendy:

Poniżej można znaleźć wszystkie wykorzystane komendy:

Autor: Rafal Klinowski, wariant: 1.

```
setwd('C:\\Users\\klino\\Pulpit\\Studia magisterskie\\APU\\Lab5')
```

```
# Instalacja mlr
install.packages("mlr")
library(mlr)
```

```
# Zadanie 1.
# Dataset: iris
library(datasets)
```

```
# Załadowanie danych 'iris'
data(iris)
```

```
# Utworzenie klasyfikatora na podstawie gatunku 'species'
iris_task <- makeClassifTask(data = iris, target = "Species")
```

```
# Podział danych na set treningowy i testowy (70% / 30%)
indices <- sample(1:nrow(iris), size = nrow(iris), replace = FALSE)
train_ratio <- 0.7
train_size <- floor(train_ratio * nrow(iris))
```

```

train_data <- iris[indices[1:train_size], ]
test_data <- iris[indices[(train_size + 1):nrow(iris)], ]

# Jakie sa dostepne learnery
# [1] "classif" "cluster" "multilabel" "regr" "surv"
levels(factor(listLearners()$type))

# Utworzenie learnera i trening
learner <- makeLearner("classif.rpart")
model <- train(learner, iris_task)

# Ocena jakosci modelu
predictions <- predict(model, newdata = test_data)

performance <- performance(predictions, measures = list(acc))
print(performance)

summary(model)

# Wykres
library(rpart.plot)
rpart.plot(getLearnerModel(model))

# Reguły klasyfikacyjne
ruleModel <- C5.0(Species ~ ., data=train_data, rules=TRUE)
ruleModel
summary(ruleModel)

# Zadanie 2.
smartfony <- read.csv('smartfony.csv')
smartfony_reduced <- smartfony[,
  c('pamiec_ram', 'pamiec_wbudowana', 'aparat_foto', 'cena', 'oceny_klientow')]

# Utworzenie klasyfikatora
smartfony_task <- makeClassifTask(data = smartfony_reduced, target = "oceny_klientow")

# Stworzenie 5 learnerow
install.packages("rFerns")
install.packages("randomForest")

lrns <- makeLearners(c("lda", "rpart", "C50", "rFerns",
  "randomForest"), type = "classif")

porownanie <- benchmark(learners = lrns,
  tasks = smartfony_task,
  resampling = cv5)

# Wizualizacja oceny
learner <- c("lda", "rpart", "C50", "rFerns", "randomForest")
accuracy <- c(0.267, 0.2, 0.2, 0.867, 0.267)

data <- data.frame(learner, accuracy)

barplot(data$accuracy, names.arg = data$learner, ylim = c(0, 1), ylab = "Accuracy")

```

```
# Utworzenie finalnego modelu
smartfony_task <- makeClassifTask(data = smartfony_reduced, target = "oceny_klientow")

smartfony_train <- smartfony_reduced[1:12, ]
smartfony_test <- smartfony_reduced[13:15, ]

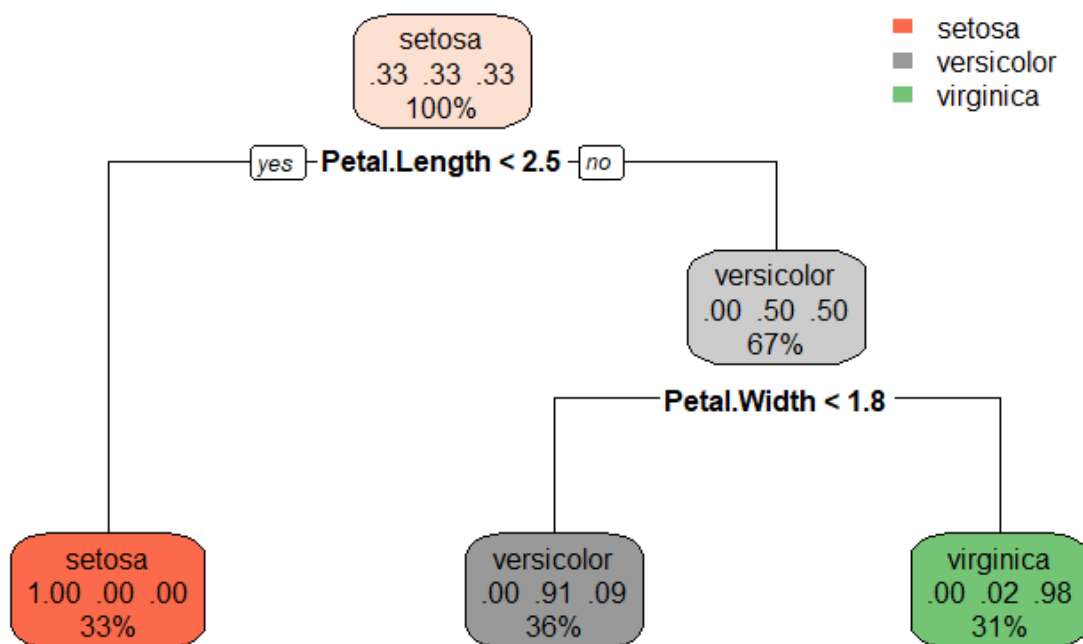
smartfony_learner <- makeLearner("classif.rFems")
smartfony_model <- train(smartfony_learner, smartfony_task)

smartfony_predictions <- predict(smartfony_model, newdata = smartfony_test)

performance <- performance(smartfony_predictions, measures = list(acc))
print(performance)
```

4. Wynik działania:

Wyniki poleceń w konsoli można znaleźć w pliku „wyniki_z_konsoli.txt”, link do repozytorium poniżej.



Rysunek 1. Drzewo będące wynikiem zadania 1.

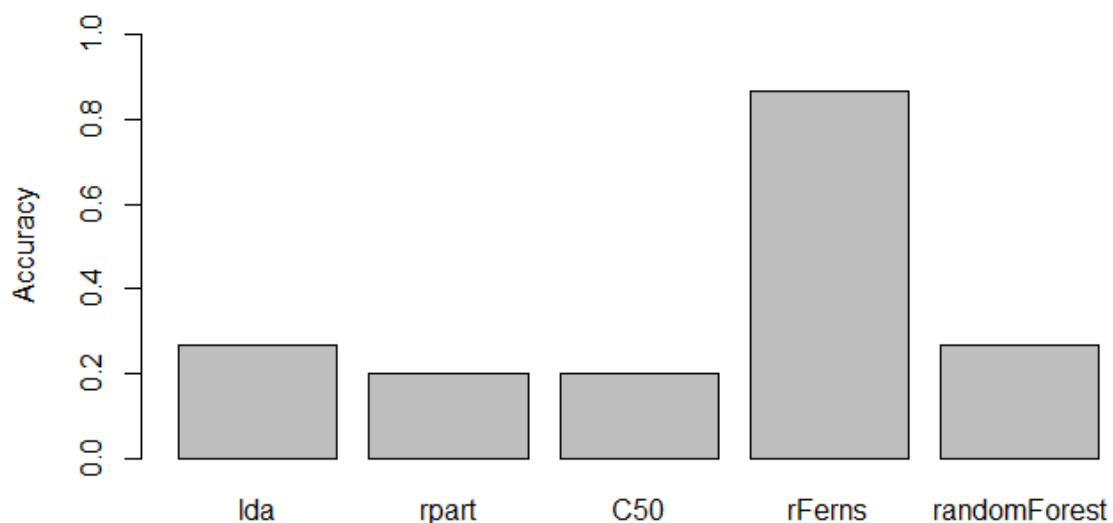
```
> print(performance)
      acc
0.9777778
> summary(model)
```

	Length	Class	Mode
learner	15	classif.rpart	list
learner.model	14	rpart	list
task.desc	13	ClassifTaskDesc	list
subset	150	-none-	numeric
features	4	-none-	character
factor.levels	1	-none-	list
time	1	-none-	numeric
dump	0	-none-	NULL

Rysunek 2. Dane o modelu uzyskanym w ramach zadania 1.

Reguły klasyfikacyjne można znaleźć w pliku zad1_rules.txt.

W zadaniu 2., porównanie learnerów można znaleźć w pliku zad2_porownanie.txt. Na jego podstawie określono, że najlepiej radzi sobie learner 'rFerns', dlatego wykorzystano go do stworzenia finałowego modelu.



Rysunek 3. Wizualizacja dokładności learnerów.

Link do repozytorium: https://github.com/Stukeley/APU_Lab5

5. Wnioski:

W zadaniu pierwszym udało się uzyskać dokładność rzędu 98%, co jest wartością zadawalającą dla ilości danych $n=150$ i bez dokładniejszego

porównania poszczególnych modeli klasyfikujących (wykorzystano tylko model `classif.rpart`).

W przypadku drugiego, finałowego modelu, wbrew wstępnej analizie learnerów (która wykazała, że 'rFerns' radzi sobie całkiem dobrze), nie udało się uzyskać dobrych wyników. Jest to możliwe spowodowane niską ilością danych wejściowych (jedynie 15 smartfonów) oraz niską korelacją między atrybutami, jak również niskim przełożeniem wartości atrybutów na (subiektywne) oceny klientów.