**SPRAWOZDANIE**

Zajęcia: Analiza Procesów Uczenia

Prowadzący: prof. dr hab. Vasyl Martsenyuk

**Laboratorium 5**

Data 28.04.2023

**Temat:**​ Modelowanie procesów uczenia maszynowego w pakiecie mlr.

**Wariant 1**

Rafał Klinowski

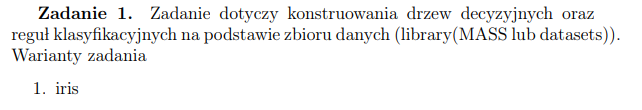
Informatyka II stopień,

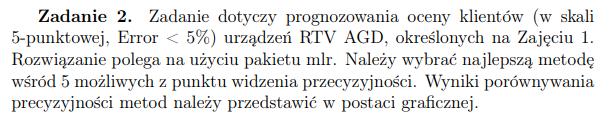
Stacjonarne,

1 semestr,

Gr. a

1. **Polecenie: Wariant 1**





1. **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’.

1. **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)

# Zaladowanie danych 'iris'

data(iris)

# Utworzenie klasyfikatora na podstawie gatunku 'species'

iris\_task <- makeClassifTask(data = iris, target = "Species")

# Podzial 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.rFerns")

smartfony\_model <- train(smartfony\_learner, smartfony\_task)

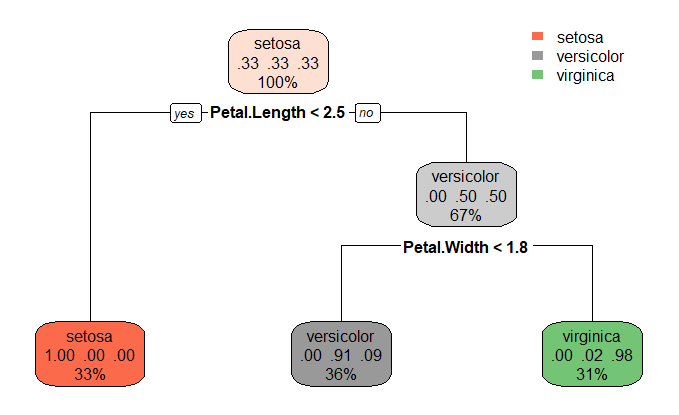
smartfony\_predictions <- predict(smartfony\_model, newdata = smartfony\_test)

performance <- performance(smartfony\_predictions, measures = list(acc))

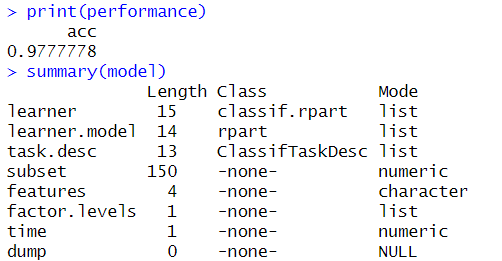
print(performance)

**4. Wynik działania:**

Wyniki poleceń w konsoli można znaleźć w pliku „wyniki\_z\_konsoli.txt”, link do repozytorium poniżej.



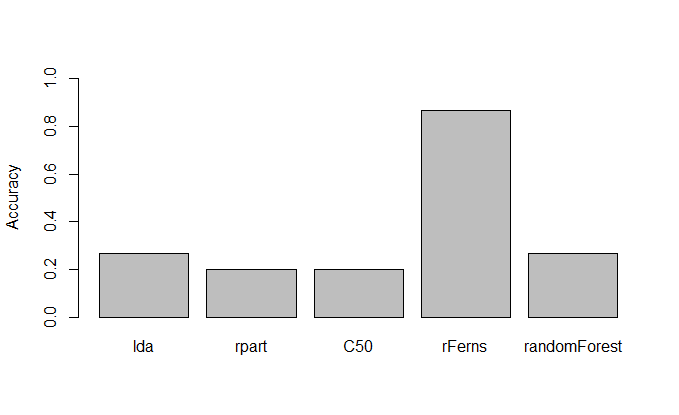
Rysunek . Drzewo będące wynikiem zadania 1.



Rysunek . 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 . Wizualizacja dokładności learnerów.

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

1. **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 możliwie spowodowane niską ilością danych wejściowych (jedynie 15 smartfonów) oraz niską korelacją między atrybutami, jak również niskim przełożeniem wartości atrybutów na (subiektywne) oceny klientów.