

Laboratorium Analizy Procesów Uczenia.

Data wykonania ćwiczenia:

26.04.2024

Rok studiów:

1

Semestr:

1

Grupa studencka:

1b

Grupa laboratoryjna:

-

Ćwiczenie nr

3.2

Temat: Modelowanie procesów uczenia maszynowego w pakiecie mlr.
Trenowanie, ocena i porównywanie modeli w pakiecie mlr.

Osoby wykonujące ćwiczenia:

1. Gracjan Wackermann

Katedra Informatyki i Automatyki

1. Cel ćwiczenia:

Celem ćwiczenia było uczenie maszynowe za pomocą pakietu mlr.

2. Zadanie do wykonania:

Zadanie dotyczy prognozowania oceny klientów, w skali 5-punktowej, $\text{error} < 5\%$ urządzeń RTV AGD, określonych na zajęciach nr.1. Rozwiązanie polega na użyciu pakietu mlr. Należy wybrać najlepszą metodę.

Uzyskany kod:

```
# Instalacja i załadowanie potrzebnych pakietów
install.packages("mlr")
install.packages("dplyr")
install.packages("ggplot2")
install.packages("randomForest")
install.packages("e1071")
install.packages("nnet")
install.packages("kernlab")

library(mlr)
library(dplyr)
library(ggplot2)
library(randomForest)
library(e1071)
library(nnet)
library(kernlab)

# Przygotowanie danych
nazwa <- c("aparat1", "aparat2", "aparat3", "aparat4",
"aparat5", "aparat6", "aparat7", "aparat8", "aparat9",
"aparat10", "aparat11", "aparat12")
rozdzielczosc <- c(20, 24, 18, 26, 30, 20, 22, 24, 26, 28, 32,
34)
zakres_czulosci <- c(100, 200, 100, 300, 200, 400, 100, 200,
300, 100, 200, 300)
cena <- c(5000, 6000, 5500, 7000, 7500, 8000, 8500, 9000, 6500,
6000, 7000, 8000)
liczba_opinii <- c(100, 200, 150, 300, 250, 400, 350, 450, 300,
200, 100, 150)
ocena_klientow <- c(4.5, 4.7, 4.2, 4.8, 4.9, 4.3, 4.6, 4.7,
4.1, 4.4, 4.6, 4.5)

data <- data.frame(rozdzielczosc, zakres_czulosci, cena,
```

```

liczba_opinii, ocena_klientow)

# Zdefiniowanie zadania
regr_task <- makeRegrTask(id = "RTV_AGD_Prediction", data =
data, target = "ocena_klientow")

# Wybór learnerów
learners <- list(
  makeLearner("regr.lm"),
  makeLearner("regr.randomForest"),
  makeLearner("regr.ksvm"),
  makeLearner("regr.nnet")
)

# Wykonanie benchmarkingu
rdesc <- makeResampleDesc("CV", iters = 5)
bmr <- benchmark(learners, regr_task, rdsc, measures =
list(rmse))

# Wizualizacja wyników
# Boxplot
plotBMRBoxplots(bmr, style = "box") + ggtitle("Porównanie metod
pod względem RMSE") + theme_bw(base_size = 16)

# Violin plot
plotBMRBoxplots(bmr, style = "violin") + ggtitle("Porównanie
metod pod względem RMSE") + theme_bw(base_size = 16)

# Ranking
plotBMRRanksAsBarChart(bmr, pos = "stack") + ggtitle("Ranking
metod") + theme_bw(base_size = 16)

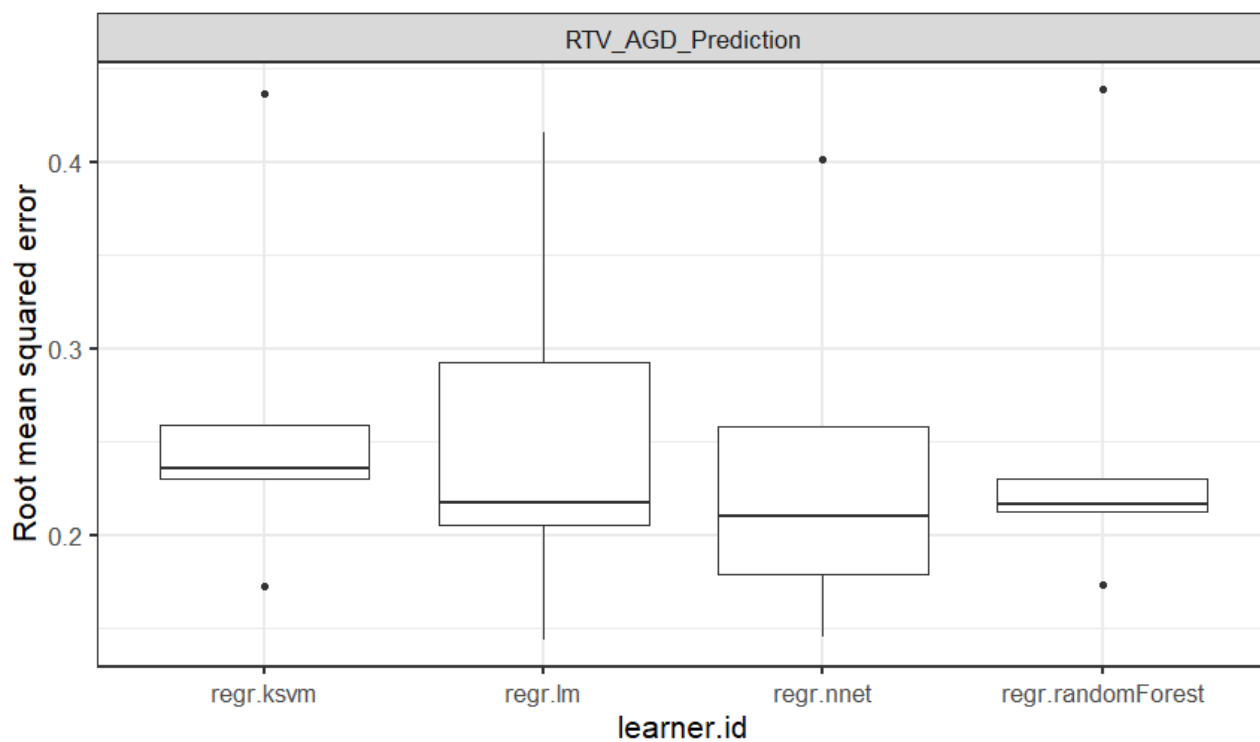
# Uzyskanie wyników benchmarkingu
getBMRAggrPerformances(bmr)

# Użycie wyników dla wizualizacji i analizy
perf <- getBMRPerformances(bmr, as.df = TRUE)
ggplot(perf, aes(x = learner.id, y = rmse, color = learner.id))
+
  geom_boxplot() +
  ggtitle("Porównanie RMSE różnych metod") +
  theme_bw(base_size = 16)

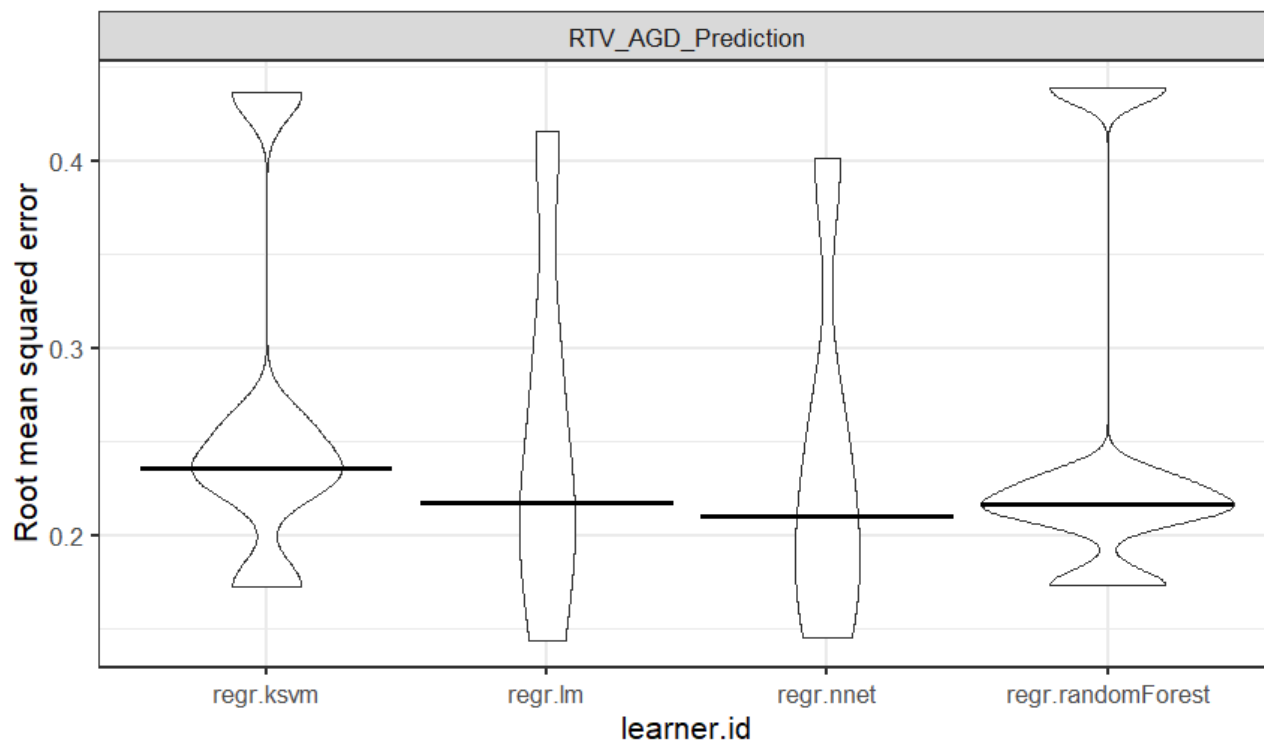
```

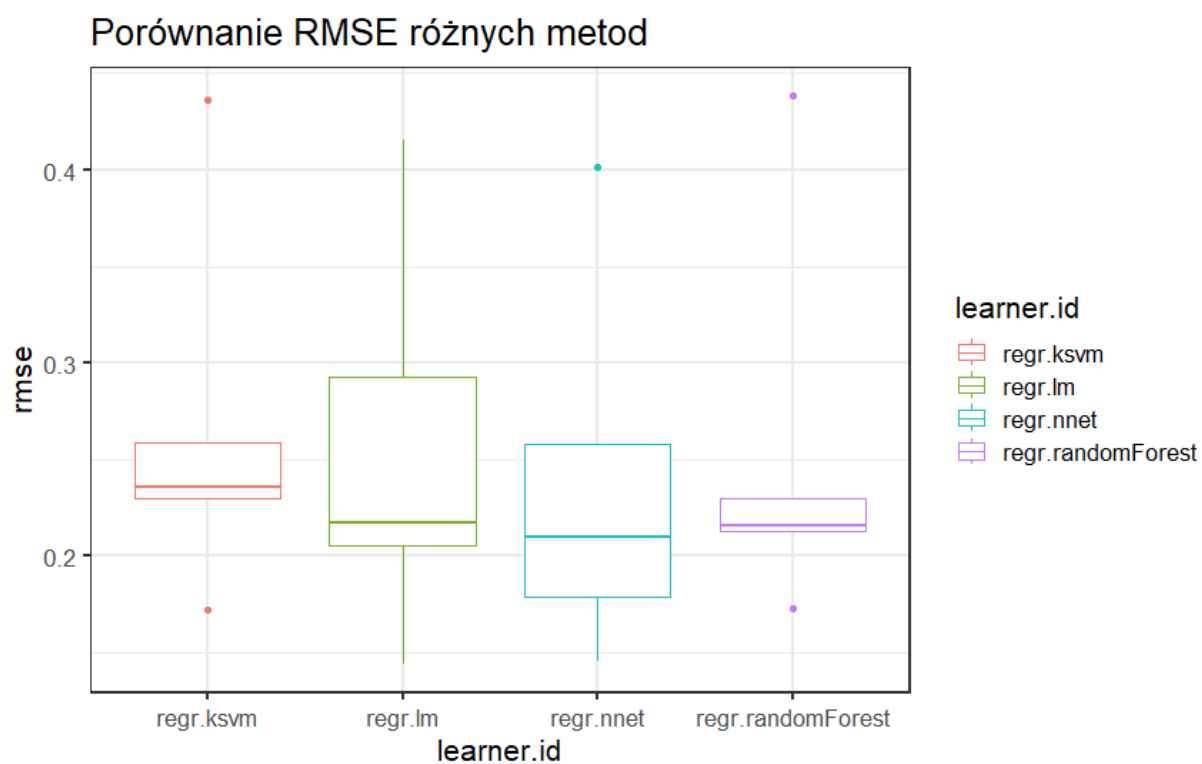
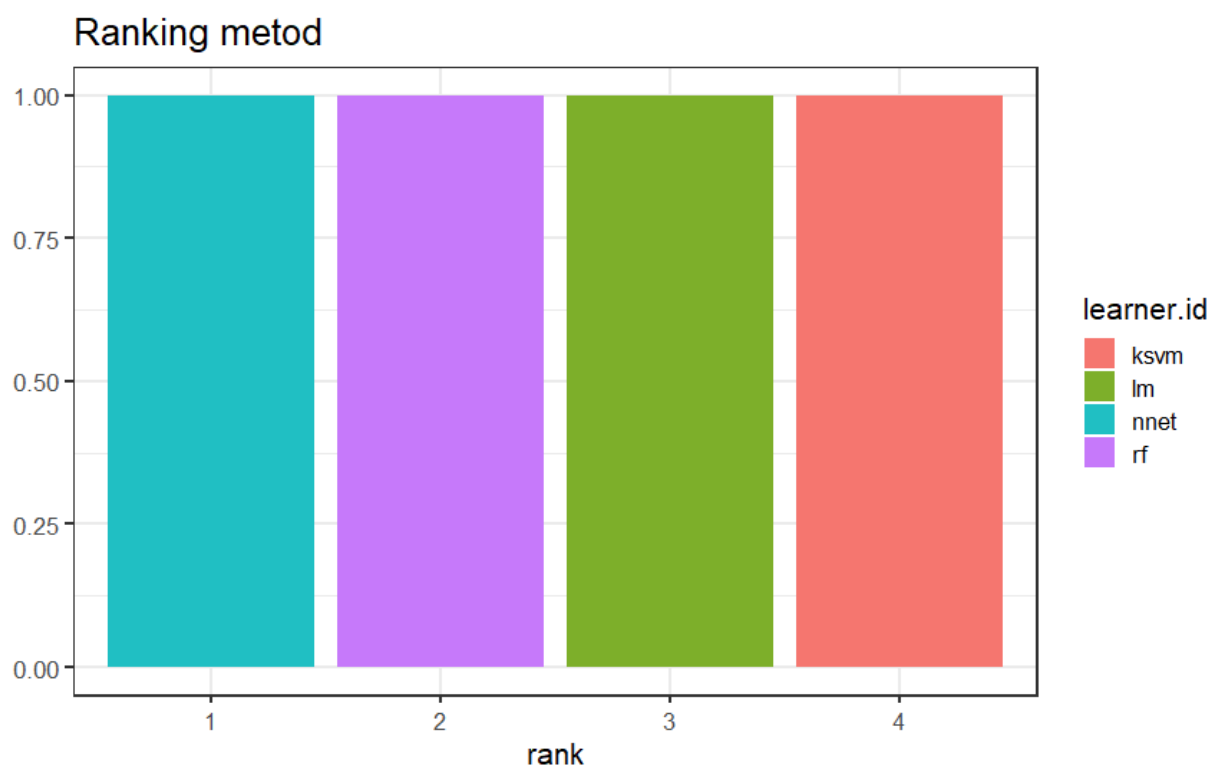
Wyniki z konsoli:

Porównanie metod pod względem RMSE



Porównanie metod pod względem RMSE





3. Wnioski:

- Wyniki dla poszczególnych modeli:

2. Model liniowy (`regr.lm`):

Średni błąd kwadratowy (RMSE) z 5-krotnej walidacji krzyżowej wynosi: **0.2714976**

4. Random Forest (`regr.randomForest`):

Średni błąd kwadratowy (RMSE) z 5-krotnej walidacji krzyżowej wynosi: **0.2710008**

1. Support Vector Machine z jądrem radialnym (`regr.ksvm`):

Średni błąd kwadratowy (RMSE) z 5-krotnej walidacji krzyżowej wynosi: **0.281168**

3. Sieć neuronowa (`regr.nnet`):

Średni błąd kwadratowy (RMSE) z 5-krotnej walidacji krzyżowej wynosi: **0.2548279**

Na podstawie wyników RMSE, model sieci neuronowej (`regr.nnet`) osiągnął najniższy średni błąd kwadratowy, co sugeruje, że jest on najbardziej precyzyjny w przewidywaniu oceny klientów spośród rozważanych modeli. Model ten uzyskał RMSE na poziomie 0.2548279, co jest najniższą wartością w porównaniu do pozostałych metod.

Link do repozytorium: <https://github.com/fireinx/apu>