**SPRAWOZDANIE**

Zajęcia: Analiza Procesów Uczenia

Prowadzący: prof. dr hab. Vasyl Martsenuk

**Laboratorium** 5

Data: 20.04.2020r

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

**Wariant** 1

Zadanie 2

Aleksander Słodczyk

Informatyka II stopień

Stacjonarne

1 semestr

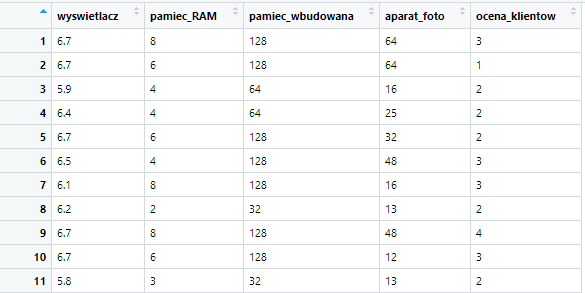
Grupa 2

1. **Polecenie**:

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.

1. **Wprowadzane dane**

Dane to zbiór smartphonów. Został przedstawiony na poniższym obrazku. Na podstawie parametrów smartphona należy określić ocenę klienta. **smartphones** =



1. **Wykorzystane komendy:**

* **makeResampleDesc**() – Wybiera algorytm próbkujący, w tym przypadku wybrałem algorytm Cross-Validation

rdesc = **makeResampleDesc**(method = "CV", stratify = F)

* **makeRegrTask**() – stworzenie zadania dotyczącego regresji, ponieważ chcę przewidzieć wartość na podstawie innych wartości.

regr.task = **makeRegrTask**(id = "Smartphones ocena klientow MSE", data = smartphones, target = "ocena\_klientow")

* **makeLearners() –** Tworzy nowych learnerów, których wykorzystam do stworzenia modelu.

regr.learners = **makeLearners**(c("lm", "randomForest", "nnet", "ctree", "rpart"), type = "regr")

* **benchmark**() – porównuje różne algorytmy uczące na podstawie jednego lub więcej tasków, w tym przypadku regr.task

regr.comparison = **benchmark**(learners = regr.learners, tasks = regr.task, resamplings = rdesc)

* **getBMRAggrPerformances**(), **getBMRPerformances**(), **plotBMRBoxplots**(), służą do wyświetlania wyników, ostatnia metoda tworzy wykres pudełkowy.

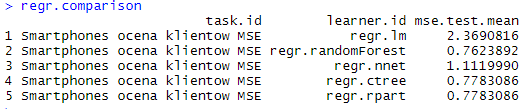
getBMRAggrPerformances(regr.comparison)

getBMRPerformances(regr.comparison)

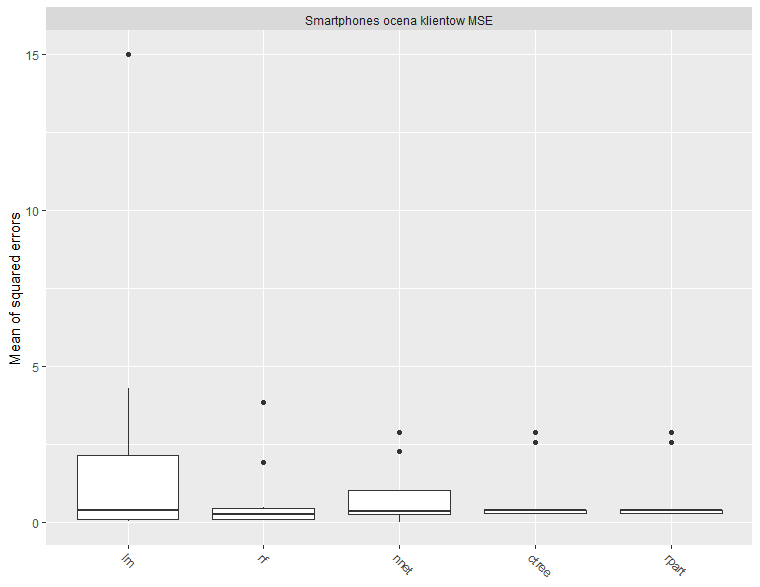
plotBMRBoxplots(regr.comparison)

1. **Wynik działania**

Poniżej przedstawiłem wynik porównania learnerów, których wybrałem do stworzenia modelu.



Na poniższym obrazku widać wykresy pudełkowe błędu średniokwadratowego poszczególnych learnerów.

****

Na podstawie błędu średniokwadratowego można stwierdzić, że największy błąd pojawia się przy zastosowaniu zwykłej regresji liniowej (lm). MSE jest największy, jak i wartości odstające. Drugim mniej pewnym algorytmem jest nnet, który ma o połowę mniejszy błąd niż regresja liniowa. Po jednakowych wynikach ctree i rpart można stwierdzić, że te dwa algorytmy są do siebie bardzo podobne.

1. **Wnioski:**

* Z użytych 5 algorytmów regresji, najpewniejszymi są rpart, ctree lub randomForest.
* Jako najlepszy algorytm wybrałbym randomForest, ponieważ jego błąd średniokwadratowy jest najmniejszy.
* Najprostszy algorytm regresji liniowej dał najgorsze wyniki, czego można się było spodziewać.
* Wszystkie algorytmy mają błąd mniejszy niż 5%

1. **Link do githuba:**

https://github.com/allo97/Analiza-procesow-uczenia-Programming-in-R/tree/master/lab5

Skrypt do algorytmów mlr znajduje się w pliku zadanie/**skrypt3Regr.R**

Wykres pudełkowy znajduje się w pliku zadanie/**smartphones\_ocena\_klientow\_MSE.png**

Wyniki z konsoli znajdują się w pliku zadanie/**skrypt3Regr\_console\_log**