

# PD8 Interpretable Machine Learning

*Daniel Ponikowski*

*16 maja 2019*

## Dane:

Wybrany zbiorem danych jest zbiór **boston**. A black-boxami do testowania (oraz przekształcania zmiennych) będzie las losowy oraz SVM.

## Wczytanie danych:

```
df <- read.csv(file = "boston.csv") %>% mutate(CHAS = factor(CHAS), RAD = factor(RAD))
```

W pracy domowej 8 przyjąłem konwencję, że jeżeli zmienna nominalna ma mniej poziomów niż liczba grup na które chce je podzielić to ta zmienna nie będzie dzielona. Black-box na podstawie którego będę agregować zmienne, będzie modelem który otrzyma mniejsze RMSE na zbiorze testowym.

## PD8:

```
best_model <- function(model1, model2, model1_rmse, model2_rmse){
  if (model1_rmse < model2_rmse ) {
    model1
  }
  else model2
}

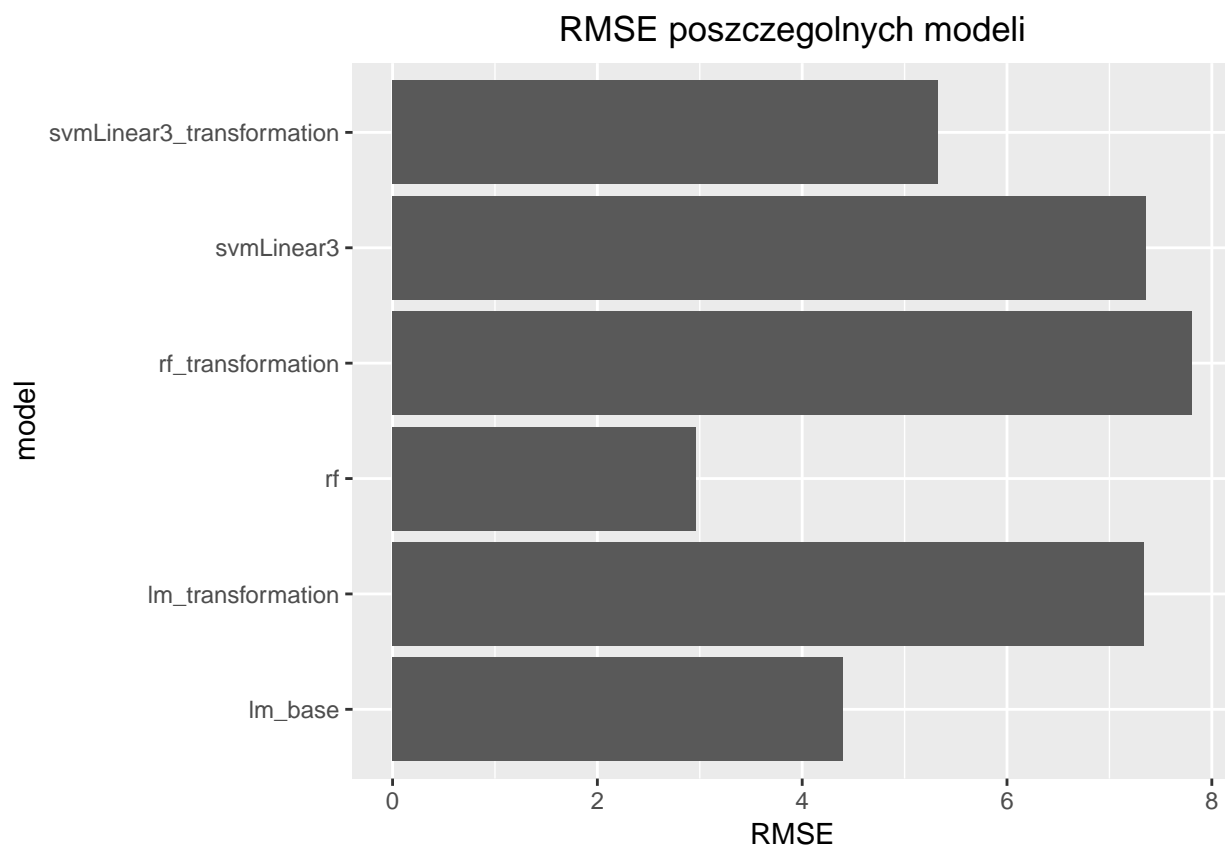
PD8 <- function(method1, method2, X, y, nominal, k = 3, p = 0.8){
  unique_values <- apply(X = X, MARGIN = 2, function(x) length(unique(x)))
  unique_values <- unique_values[unique_values > k]
  train_num <- createDataPartition(y = y, list = FALSE, p = p)
  X_train <- X[train_num,]
  X_test <- X[-train_num,]
  y_train <- y[train_num]
  y_test <- y[-train_num]
  model1 <- train(x = X_train, y = y_train, method = method1)
  model2 <- train(x = X_train, y = y_train, method = method2)
  lmbase <- train(x = X_train, y = y_train, method = "lm")
  lmbase_rmse <- rmse(predict(lmbase, X_test), y_test)
  model1_rmse <- rmse(predict(model1, X_test), y_test)
  model2_rmse <- rmse(predict(model2, X_test), y_test)
  model_final <- best_model(model1, model2, model1_rmse, model2_rmse)
  for (zmienna in names(unique_values)){
    mv1 <- partial(model_final, pred.var = zmienna, class = TRUE)
    if (zmienna %in% nominal) {
      clust <- hclust(dist(mv1$yhat), method = "complete", members = NULL)
      mv1$pred <- cutree(clust, k)
      levels(X_train[[zmienna]]) <- as.character(mv1$pred)
      levels(X_test[[zmienna]]) <- as.character(mv1$pred)
    }
    else{
      change_point <- cpt.meanvar(c(mv1$yhat), pen.value = 0.01)
    }
  }
}
```

```

minimum <- min(X_train[[zmienna]])
X_train[[zmienna]] <- cut(X_train[[zmienna]],c(minimum,mv1[attr(change_point,"cpts"),1])
                        ,include.lowest = TRUE)
X_test[[zmienna]] <- cut(X_test[[zmienna]],c(minimum,mv1[attr(change_point,"cpts"),1])
                        ,include.lowest = TRUE)
}
}
data2 <- cbind(X_train, y = y_train)
lm_transf <- train(y~.,data2,"lm")
lm_transf_rmse <- rmse(predict(lm_transf,X_test),y_test)
model1_transformation <- train(y~.,data2,method1)
model2_transformation <- train(y~.,data2,method2)
model1_transformation_rmse <- rmse(predict(model1_transformation,X_test),y_test)
model2_transformation_rmse <- rmse(predict(model2_transformation,X_test),y_test)
result <- c(model1_rmse,model1_transformation_rmse,
            model2_rmse,model2_transformation_rmse,lm_transf_rmse,lmbase_rmse)
data.frame(RMSE = result, model = c(method1,paste0(method1,"_transformation"),
            paste0(method2,"_transformation"),method2,"lm_transformation","lm_base"))
}

y <- df$MEDV
df$MEDV <- NULL
X <- df
result <- PD8(method1 = "rf",method2 = "svmLinear3",X = X,y = y,
              nominal = c("RAD"),k = 3, p = 0.8)
ggplot(data = result,aes(x = model,y = RMSE)) + geom_bar(stat="identity") + coord_flip() +
  ggtitle("RMSE poszczegolnych modeli") + theme(plot.title = element_text(hjust = 0.5))

```



Wnioski :

1. model regresji liniowej na oryginalnych zmiennych otrzymał lepszy wynik od black-boxu (SVM)
2. przekształcenie zmiennych znacznie pogorszyło predykcje lasu losowego
3. przekształcenie zmiennych polepszyło predykcje SVM
4. wynik regresji liniowej na przekształconych zmiennych jest gorszy od wyniku na oryginalnych danych