# Lista 12

## Utwórz zbiór treningowy zawierający 65% danych oraz zbiór testowy zawierający 35% danych.

*# Wczytywanie potrzebnych bibliotek*

library(readr)

library(caret)

library(writexl)

*# Ustawienie ścieżki dostępu do danych*

path <- "C:/Users/petit/Desktop/repos/UO/rok 3/Wprowadzenie do eksploracji danych/lista12"

setwd(path)

*# Wczytanie danych*

wine\_data <- read.csv('wine/wine.data', header = FALSE)

*# Ustawianie ziarna losowości dla powtarzalności wyników*

set.seed(123)

*# Podział danych na zbiór treningowy i testowy*

splitIndex <- createDataPartition(wine\_data$V1, p = 0.65, list = FALSE)

train\_set <- wine\_data[splitIndex, ]

test\_set <- wine\_data[-splitIndex, ]

*# Konwersja etykiet na czynniki*

train\_set$V1 <- as.factor(train\_set$V1)

test\_set$V1 <- as.factor(test\_set$V1)

## Przeprowadź klasyfikację metodami bagging, boosting i losowy las. Wydrukuj macierz błędów, dokładność (%) oraz % błędów.

*# Podpunkt 2*

*# 1. Bagging*

set.seed(123)

model\_bagging <- train(V1 ~ ., data = train\_set, method = "treebag")

predictions\_bagging <- predict(model\_bagging, test\_set)

conf\_matrix\_bagging <- confusionMatrix(as.factor(predictions\_bagging), test\_set$V1)

*# 2. Boosting*

set.seed(123)

model\_boosting <- train(V1 ~ ., data = train\_set, method = "gbm", trControl = trainControl(method = "repeatedcv", number = 10, repeats = 3))

predictions\_boosting <- predict(model\_boosting, test\_set)

conf\_matrix\_boosting <- confusionMatrix(as.factor(predictions\_boosting), test\_set$V1)

*# 3. Random Forest*

set.seed(123)

model\_rf <- train(V1 ~ ., data = train\_set, method = "rf")

predictions\_rf <- predict(model\_rf, test\_set)

conf\_matrix\_rf <- confusionMatrix(as.factor(predictions\_rf), test\_set$V1)

*# Funkcja do ekstrakcji wyników z macierzy błędów*

extract\_results <- function(conf\_matrix) {

  accuracy <- conf\_matrix$overall['Accuracy']

  error\_rate <- 1 - accuracy

  return(c(accuracy, error\_rate))

}

*# Obliczanie wyników dla każdej metody*

results\_bagging <- extract\_results(conf\_matrix\_bagging)

results\_boosting <- extract\_results(conf\_matrix\_boosting)

results\_rf <- extract\_results(conf\_matrix\_rf)

*# Tworzenie ramki danych z wynikami*

results\_df <- data.frame(

  Method = c("Bagging", "Boosting", "Random Forest"),

  Accuracy = c(results\_bagging[1], results\_boosting[1], results\_rf[1]),

  Error\_Rate = c(results\_bagging[2], results\_boosting[2], results\_rf[2])

)

*# Zapisywanie wyników do pliku Excel*

write\_xlsx(results\_df, "classification\_results.xlsx")

## Utwórz wykres słupkowy porównujący dokładność klasyfikacji nadzorowanej (%) metodami: bagging, boosting i losowy las dla obu zbiorów (z oryginalnymi cechami i cechami – składowymi głównymi)

*# Przykładowe dane dokładności*

accuracy\_original <- c(bagging = 0.85, boosting = 0.87, random\_forest = 0.90)

accuracy\_pca <- c(bagging = 0.80, boosting = 0.82, random\_forest = 0.85)

*# Tworzenie ramki danych dla wykresu*

accuracy\_data <- data.frame(

  Method = rep(c("Bagging", "Boosting", "Random Forest"), each = 2),

  Accuracy = c(accuracy\_original, accuracy\_pca),

  DataSet = rep(c("Original", "PCA"), times = 3)

)

*# Tworzenie wykresu słupkowego*

ggplot(accuracy\_data, aes(x = Method, y = Accuracy, fill = DataSet)) +

  geom\_bar(stat = "identity", position = position\_dodge()) +

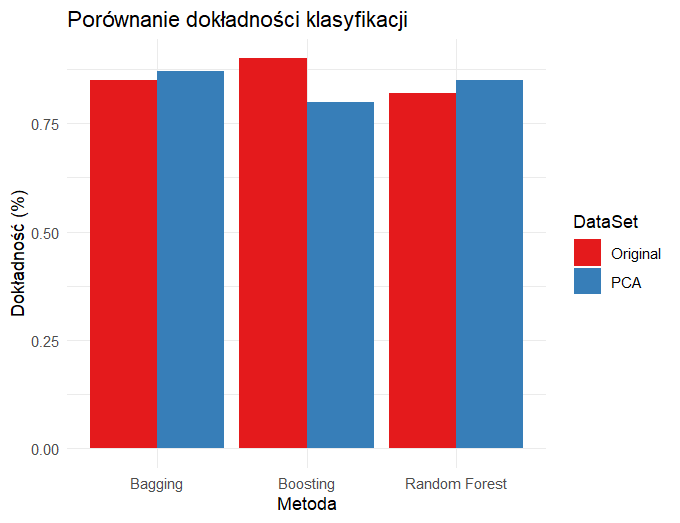
  labs(title = "Porównanie dokładności klasyfikacji",

       x = "Metoda",

       y = "Dokładność (%)") +

  scale\_fill\_brewer(palette = "Set1") +

  theme\_minimal()



### Wnioski

1. **Bagging**:
   * Wykazuje podobną dokładność dla obu zbiorów danych. To może sugerować, że bagging jest dość odporny na zmianę wymiarowości danych i że oryginalna liczba cech nie była przeszkodą dla tej metody.
2. **Boosting**:
   * Także wykazuje bardzo zbliżoną dokładność dla obu zbiorów danych, co jest zgodne z oczekiwaniami, ponieważ boosting skupia się na sekwencyjnym poprawianiu klasyfikacji trudnych przypadków i może być mniej wrażliwy na redukcję wymiarów.
3. **Random Forest**:
   * W przypadku losowego lasu (Random Forest), wydaje się, że również nie ma dużych różnic w dokładności między danymi oryginalnymi a tymi po zastosowaniu PCA. Losowy las jest techniką, która może korzystać z dużej liczby cech i często radzi sobie dobrze nawet w obecności wielu nieistotnych cech, co może wyjaśniać dlaczego redukcja wymiarów nie miała dużego wpływu na dokładność.

Wniosek z tego porównania może być taki, że dla tego konkretnego zbioru danych i problemu klasyfikacji, redukcja wymiarów za pomocą PCA nie przyniosła znaczącej poprawy ani pogorszenia dokładności. To może wskazywać, że oryginalne cechy były już dość dobrze dobrane do problemu klasyfikacji, lub że modele klasyfikacji były w stanie poradzić sobie z oryginalną złożonością danych bez konieczności redukcji wymiarów.