Lista 12

1. 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)
```

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

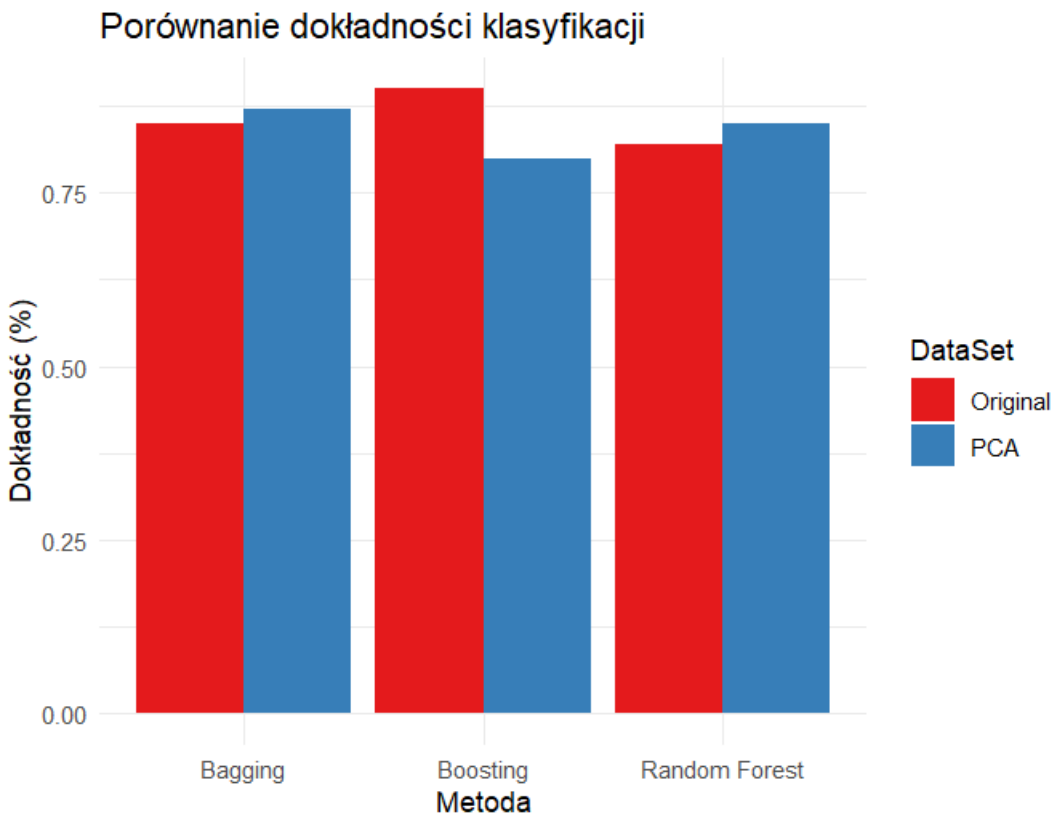
```
# Podpunkt 2
# 1. Bagging
set.seed(123)
model_bagging <- train(V1 ~ ., data = train_set, method =
"treebag")
predictions_bagging <- predict(model_bagging, test_set)
conf_matrix_bagging <-
confusionMatrix(as.factor(predictions_bagging), test_set$V1)

# 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")
```

3. 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.