Raport - Klasyfikacja na bazie regresji liniowej i porównanie metod klasyfikacji

Filip Michewicz 282239
Wiktor Niedźwiedzki 258882

28 maja 2025 Anno Domini

Spis treści

1 Klasyfikacja na bazie modelu regresji liniowej **6**

1.1 Charakterystyka danych . . . 6
1.2 Budowa klasyfikatora . . . 7
1.3 Ewaluacja jakości modelu . . . 8
1.4 Budowa klasyfikatora w rozszerzonej przestrzeni cech . . . 11
1.5 Ewaluacja jakości modelu . . . 12
1.6 Podsumowanie . . . 15

2 Porównanie metod klasyfikacji **16**

2.1 Charaktetystyka danych . . . 16
2.2 Budowa i ewaluacja klasyfikatorów . . . 20
2.2.1 Klasyfikator k-NN . . . 21
2.2.2 Drzewa klasyfikacyjne . . . 39
2.2.3 Naiwny klasyfikator bayesowski . . . 55
2.3 Porównanie metod klasyfikacji . . . 61
2.4 Wnioski . . . 61

Spis wykresów

1 Częstości etykiet w zbiorze iris . . . 7
2 Wartości predykcyjne dla każdej obserwacji . . . 11
3 Wartości predykcyjne dla każdej obserwacji - przestrzeń rozszerzona . . . 14
4 Porównanie przewidywanych i rzeczywistych klas . . . 15
5 Liczba występowania gatunków win w zbiorze wine . . . 17
6 Rozkłady zmiennych ilościowych z podziałem na gatunek . . . 18

7 Wykresy pudełkowe zmiennych numerycznych . . . 19
8 Wykresy pudełkowe zmiennych numerycznych - bez Magnezu i Proliny . . . 20
9 Porównanie przewidywanych i rzeczywistych klas - KNN . . . 39
10 Nieprzycięte drzewo klasyfikacyjne . . . 46
11 Przycięte drzewo klasyfikacyjne . . . 46
12 Porównanie klasyfikacji całego zbioru dla konkretnych metod . . . 61

Spis tabel

1 Opis zmiennych w zbiorze danych iris . . . 6
2 Wartości współczynników regresji liniowej dla poszczególnych cech i kategorii . . . 8
3 Macierz pomyłek - zbiór treningowy . . . 9
4 Metryki klasyfikacji dla każdego gatunku - zbiór treningowy . . . 10
5 Macierz pomyłek - zbiór testowy . . . 10
6 Metryki klasyfikacji dla każdego gatunku . . . 10
7 Wartości współczynników regresji liniowej dla poszczególnych cech i kategorii - przestrzeń rozszerzona . . . 12
8 Macierz pomyłek - zbiór treningowy . . . 13
9 Metryki klasyfikacji dla każdego gatunku . . . 13
10 Macierz pomyłek - zbiór testowy . . . 14
11 Metryki klasyfikacji dla każdego gatunku - przestrzeń rozszerzona . . . 14
12 Podsumowanie metryk klasyfikacji w różnych przestrzeniach i zbiorach . . . 16
13 Opis zmiennych w zbiorze danych wine . . . 16
14 Średnia dokładność klasyfikacji dla kombinacji parametrów rozproszenia i lokalizacji - wielokrotny podział - KNN . . . 23
15 Średni F1-score dla kombinacji parametrów rozproszenia i lokalizacji - wielokrotny podział - KNN . . . 24
16 Średnia dokładność klasyfikacji dla kombinacji parametrów rozproszenia i lokalizacji - cross-validation - KNN . . . 25
17 Średni F1-score dla kombinacji parametrów rozproszenia i lokalizacji - wielokrotny podział - KNN . . . 25
18 Średnia dokładność klasyfikacji dla kombinacji parametrów rozproszenia i lokalizacji - bootstrap - KNN . . . 27
19 Średni F1-score dla kombinacji parametrów rozproszenia i lokalizacji - wielokrotny podział - KNN . . . 27
20 Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - wielokrotny podział - zbiór treningowy - KNN . . . 28
21 Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - wielokrotny podział - zbiór testowy - KNN . . . 28

22 Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - wielokrotny podział - zbiór treningowy - KNN . . . 29
23 Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - wielokrotny podział - zbiór testowy - KNN . . . 29
24 Średnia dokładność klasyfikacji dla kombinacji parametrów liczby podzbiorów oraz liczby sąsiadów - cross-validation - zbiór treningowy - KNN . . . 29
25 Średnia dokładność klasyfikacji dla kombinacji parametrów liczby podzbiorów oraz liczby sąsiadów - cross-validation - zbiór testowy - KNN . . . 30
26 Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - cross-validation - zbiór treningowy - KNN . . . 30
27 Średni F1-score dla kombinacji parametrów liczby podzbiorów oraz liczby sąsiadów - cross-validation - zbiór testowy - KNN . . . 31
28 Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - bootstrap - zbiór treningowy - KNN . . . 31
29 Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - bootstrap - zbiór testowy - KNN . . . 32
30 Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - bootstrap - zbiór treningowy - KNN . . . 32
31 Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - bootstrap - zbiór testowy - KNN . . . 32
32 Skuteczność dyskryminacyjna zmiennych po dyskretyzacji metodą k-średnich . . . 33
33 Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - wielokrotny podział - zbiór testowy - KNN . . 34
34 Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - wielokrotny podział - zbiór testowy - KNN . . . 34
35 Średnia dokładność klasyfikacji dla kombinacji parametrów liczby podzbiorów oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - cross-validation - zbiór testowy - KNN . 35
36 Średni F1-score dla kombinacji parametrów liczby podzbiorów oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - cross-validation - zbiór testowy - KNN . . . 35
37 Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - bootstrap - zbiór testowy - KNN . . . 36
38 Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - bootstrap - zbiór testowy - KNN . . . 36
39 Macierz pomyłek — zbiór treningowy — KNN . . . 37
40 Metryki klasyfikacji dla każdego gatunku - zbiór treningowy - KNN . . . 37
41 Macierz pomyłek — zbiór testowy — KNN . . . 38
42 Metryki klasyfikacji dla każdego gatunku - zbiór testowy - KNN . . . 38
43 Średnie dokładność klasyfikacji oraz F1-score dla drzew przyciętych oraz nieprzyciętych, z podziałem na metody oceny . . . 45
44 Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentów cp i minsplit - zbiór treningowy - nieprzycięte drzewo klasyfikacyjne . . . 47
45 Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentów cp i minsplit - zbiór testowy - nieprzycięte drzewo klasyfikacyjne . . . 48

46 Średni F1-score klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentów cp i minsplit - zbiór treningowy - nieprzycięte drzewo klasyfikacyjne 48

47 Średni F1-score klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentów cp i minsplit - zbiór testowy - nieprzycięte drzewo klasyfikacyjne 49

48 Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentów cp i minsplit - zbiór treningowy - przycięte drzewo klasyfikacyjne 49

49 Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentów cp i minsplit - zbiór testowy - przycięte drzewo klasyfikacyjne 50

50 Średni F1-score klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentów cp i minsplit - zbiór treningowy - przycięte drzewo klasyfikacyjne 50

51 Średni F1-score klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentów cp i minsplit - zbiór testowy - przycięte drzewo klasyfikacyjne 51

52 Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - wielokrotny podział - zbiór testowy - przycięte drzewo klasyfikacyjne 51

53 Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - wielokrotny podział - zbiór testowy - przycięte drzewo klasyfikacyjne 52

54 Średni dokładność klasyfikacji dla kombinacji parametrów liczby podzbiorów zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - cross-validation - zbiór testowy - przycięte drzewo klasyfikacyjne 53

55 Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - cross-validation - zbiór testowy - przycięte drzewo klasyfikacyjne 53

56 Średni dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - bootstrap - zbiór testowy - przycięte drzewo klasyfikacyjne 54

57 Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - bootstrap - zbiór testowy - przycięte drzewo klasyfikacyjne . . . 54

58 Średnie dokładność klasyfikacji oraz F1-score dla naiwnego klasyfikatora bayesowskiego, z podziałem na metody oceny oraz argument laplace 55

59 Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentu laplace - zbiór treningowy - naiwny klasyfikator bayesowski 55

60 Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentu laplace - zbiór testowy - naiwny klasyfikator bayesowski 56

61 Średni F1-score dla kombinacji parametrów podziału zbioru oraz argumentu laplace - zbiór treningowy - naiwny klasyfikator bayesowski 56

62 Średni F1-score dla kombinacji parametrów podziału zbioru oraz argumentu laplace - zbiór testowy - naiwny klasyfikator bayesowski 57

63 Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - wielokrotny podział - zbiór testowy - naiwny klasyfikator bayesowski (laplace=0) 58

64 Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - wielokrotny podział - zbiór testowy - naiwny klasyfikator bayesowski (laplace=0) 58

65 Średnia dokładność klasyfikacji dla kombinacji parametrów liczba podzbiorów oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - cross-validation - zbiór testowy - naiwny klasyfikator bayesowski (laplace=0) 59

66 Średni F1-score dla kombinacji parametrów liczby podzbiorów oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - cross-validation - zbiór testowy - naiwny klasyfikator bayesowski (laplace=0) 59

67 Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - bootstrap - zbiór testowy - naiwny klasyfikator bayesowski (laplace=0) 59

68 Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - bootstrap - zbiór testowy - naiwny klasyfikator bayesowski (laplace=0) 60

1 Klasyfikacja na bazie modelu regresji liniowej

W tym zadaniu zostanie przeprowadzona klasyfikacja na bazie modelu regresji liniowej zbioru danych *iris* z pakietu *datasets* w R i zawiera wyniki pomiarów dotyczących trzech gatunków irysów:

- **Setosa**
- **Versicolor**
- **Virginica**

Dane zostały udostępnione przez znanego brytyjskiego statystyka, Ronalda Fishera, w 1936 roku i od tego czasu stały się klasycznym przykładem w wielu analizach statystycznych oraz w kontekście algorytmów klasyfikacyjnych.

1.1 Charakterystyka danych

Zbiór danych *Iris* zawiera **150** przypadków, po 50 dla każdego z trzech gatunków oraz **5** cech. Liczba brakujących danych wynosi **0**.

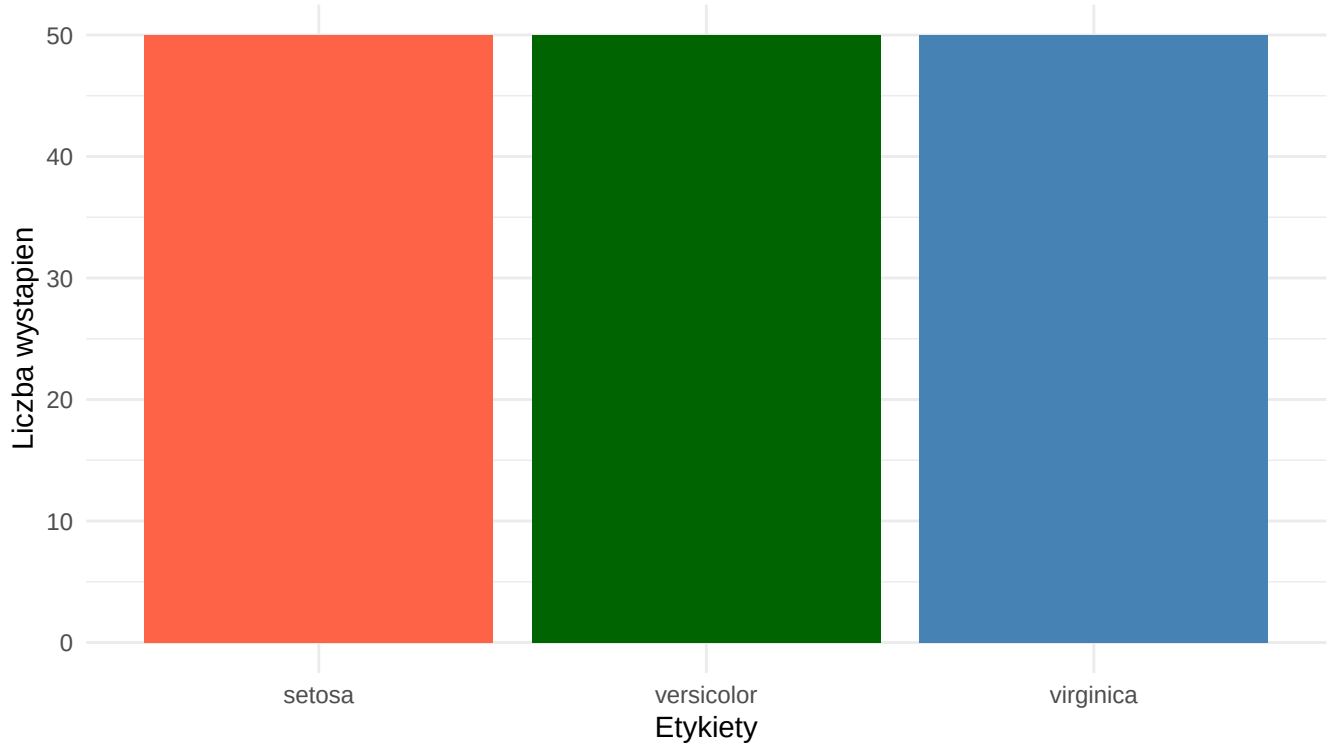
Znaczenie poszczególnych cech oraz ich typ przedstawiono w Tabeli 1.

Tabela 1: Opis zmiennych w zbiorze danych iris

Zmienna	Typ	Opis
Sepal.Length	numeric	Długość działki kielicha (cm)
Sepal.Width	numeric	Szerokość działki kielicha (cm)
Petal.Length	numeric	Długość płatka (cm)
Petal.Width	numeric	Szerokość płatka (cm)
Species	factor	Gatunek (setosa, versicolor, virginica)

Sprawdzono częstość występowania poszczególnych wartości zmiennych klasyfikacyjnych. Wyniki zaprezentowano na Wykresie 1.

```
etykiety <- iris$Species
n <- length(etykiety)
K <- length(unique(etykiety))

ggplot(data.frame(etykiety), aes(x = etykiety, fill = etykiety)) +
  geom_bar() +
  labs(x = "Etykiety", y = "Liczba wystąpień") +
  scale_fill_manual(values = c("tomato", "darkgreen", "steelblue")) +
  theme_minimal() +
  theme(legend.position = "none")
```



Wykres 1: Częstości etykiet w zbiorze iris

Wszystkie zmienne klasyfikacyjne występują z jednokową częstotliwością.

Wszystkie cechy dotyczące kwiatów są miarami ciągłymi (choć zbierane w sposób dyskretny) i różnią się w zależności od gatunku irysa, który jest zmienną jakościową. Celem analizy będzie zrozumienie, w jaki sposób te cechy mogą pomóc w klasyfikacji irysów do odpowiednich gatunków oraz zastosowanie różnych technik dyskretyzacji i analizy, aby lepiej zrozumieć właściwości tych danych.

1.2 Budowa klasyfikatora

W tej części przeprowadzono konstrukcję klasyfikatora opartego na regresji liniowej. Zbiór danych podzielono w stosunku 2:1 na zbiór treningowy oraz testowy.

```
idx_train <- sort(sample(n, size = floor(2/3 * n)))
train <- iris[idx_train, ]
test  <- iris[-idx_train, ]

train_cechy <- train[, -5]
etykiety <- train$Species
N <- nrow(train_cechy)
K <- length(unique(etykiety))

X.train <- as.matrix(cbind(rep(1, N), train_cechy))

Y.train <- matrix(0, nrow=N, ncol=K)
etykiety.num <- as.numeric(etykiety)
```

```r
for (k in 1:K) {
  Y.train[etykiety.num==k, k] <- 1
}

B.hat <- solve(t(X.train)%*%X.train) %*% t(X.train) %*% Y.train
```

W Tabeli 2. zestawiono wartości współczynników regresji liniowej przypisanych poszczególnym cechom dla kategorii klasyfikacyjnych w zbiorze danych.

Tabela 2: Wartości współczynników regresji liniowej dla poszczególnych cech i kategorii

Kombinacja Cech	Setosa	Versicolor	Virginica
1	-0.154	2.210	-1.056
Sepal.Length	0.133	-0.150	0.017
Sepal.Width	0.242	-0.471	0.229
Petal.Length	-0.254	0.275	-0.021
Petal.Width	-0.064	-0.481	0.545

1.3 Ewaluacja jakości modelu

W tej części dokonano ewaluacji jakości stworzonego modelu regresji liniowej. Do oceny modelu wykorzystano miarę F1-score, która stanowi harmoniczną średnią precyzji oraz czułości.

F1-score jest szczególnie użyteczna w przypadkach, gdy istnieje nierównowaga między klasami, ponieważ łączy informacje o dokładności klasyfikacji pozytywnej oraz jej kompletności, zapewniając zrównoważoną ocenę skuteczności modelu.

Matematycznie definiuje się ją jako:

$$F_1 = 2 \times \frac{\text{Precyzja} \times \text{Czułość}}{\text{Precyzja} + \text{Czułość}}$$

gdzie:

$$\text{Precyzja} = \frac{TP}{TP + FP}, \quad \text{Czułość} = \frac{TP}{TP + FN}$$

a TP oznacza liczbę prawdziwie pozytywnych klasyfikacji, FP – fałszywie pozytywnych, FN – fałszywie negatywnych.

Stworzono funkcję, która zwraca najważniejsze miary jakości klasyfikacji, takie jak czułość, precyzja oraz F1-score.

```r
metryki <- function(conf) {
  TP <- diag(conf)
  FP <- colSums(conf) - TP
  FN <- rowSums(conf) - TP
  TN <- sum(conf) - (TP + FP + FN)

  precision <- ifelse((TP + FP) == 0, 0, TP / (TP + FP))
  recall <- ifelse((TP + FN) == 0, 0, TP / (TP + FN))
  f1 <- ifelse((precision + recall) == 0, 0,
```

```
              2 * precision * recall / (precision + recall))

  accuracy <- sum(TP) / sum(conf)
  accuracy_per_class <- (TP + TN) / (TP + TN + FP + FN)

  results <- data.frame(
    Accuracy = accuracy_per_class,
    Precision = precision,
    Recall = recall,
    F1_Score = f1
  )

  avg_metrics <- c(
    Accuracy = mean(accuracy_per_class),
    Precision = mean(precision),
    Recall = mean(recall),
    F1_Score = mean(f1)
  )

  results <- rbind(results, avg_metrics)
  rownames(results)[nrow(results)] <- "Średnie"

  return(results)
}
```

Ewaluacja modelu na zbiorze treningowym.

```
### Treningowy

Y.hat.train <- X.train %*% B.hat
pred_train <- apply(Y.hat.train, 1, which.max)

conf.matrix <- table(pred_train, train$Species)

accuracy_a <- sum(diag(conf.matrix))/sum(conf.matrix)
```

Wyniki klasyfikacji na zbiorze treningowym przedstawiono w macierzy pomyłek zawartej w Tabeli 3.

Tabela 3: Macierz pomyłek - zbiór treningowy

Gatunek	setosa	versicolor	virginica
setosa	30	0	0
versicolor	1	28	4
virginica	0	7	30

Błąd klasyfikacyjny na zbiorze treningowym wynosi 12%. nic nie wskazuje na maskowanie klas.

W Tabeli 4. przedstawiono najwazniejsze metryki klasyfikacyjne dla każdego gatunku.

Tabela 4: Metryki klasyfikacji dla każdego gatunku - zbiór treningowy

Prawdziwy gatunek	Dokładność	Precyzja	Czułośc	F1-score
setosa	0.99	0.97	1.00	0.98
versicolor	0.88	0.80	0.85	0.82
virginica	0.89	0.88	0.81	0.85
Średnie	0.92	0.88	0.89	0.88

Średnia dokładność dla każdego gatunku wynosi 92%, a F1-score 88%, co świadczy o dobrej skuteczności modelu w klasyfikacji poszczególnych kategorii na zbiorze treningowym.

Sprawdzono zdolnosć klasyfikacyjną modelu na zbiorze testowym.

```
### Testowy
test_cechy <- test[, -5]
N <- nrow(test_cechy)

X.test <- as.matrix(cbind(rep(1, N), test_cechy))

Y.hat.test <- X.test %*% B.hat
pred_test <- apply(Y.hat.test, 1, which.max)

conf.matrix <- table(pred_test, test$Species)

accuracy_b <- sum(diag(conf.matrix))/sum(conf.matrix)
```

Wyniki klasyfikacji na zbiorze testowym przedstawiono w macierzy pomyłek zawartej w Tabeli 5.

Tabela 5: Macierz pomyłek - zbiór testowy

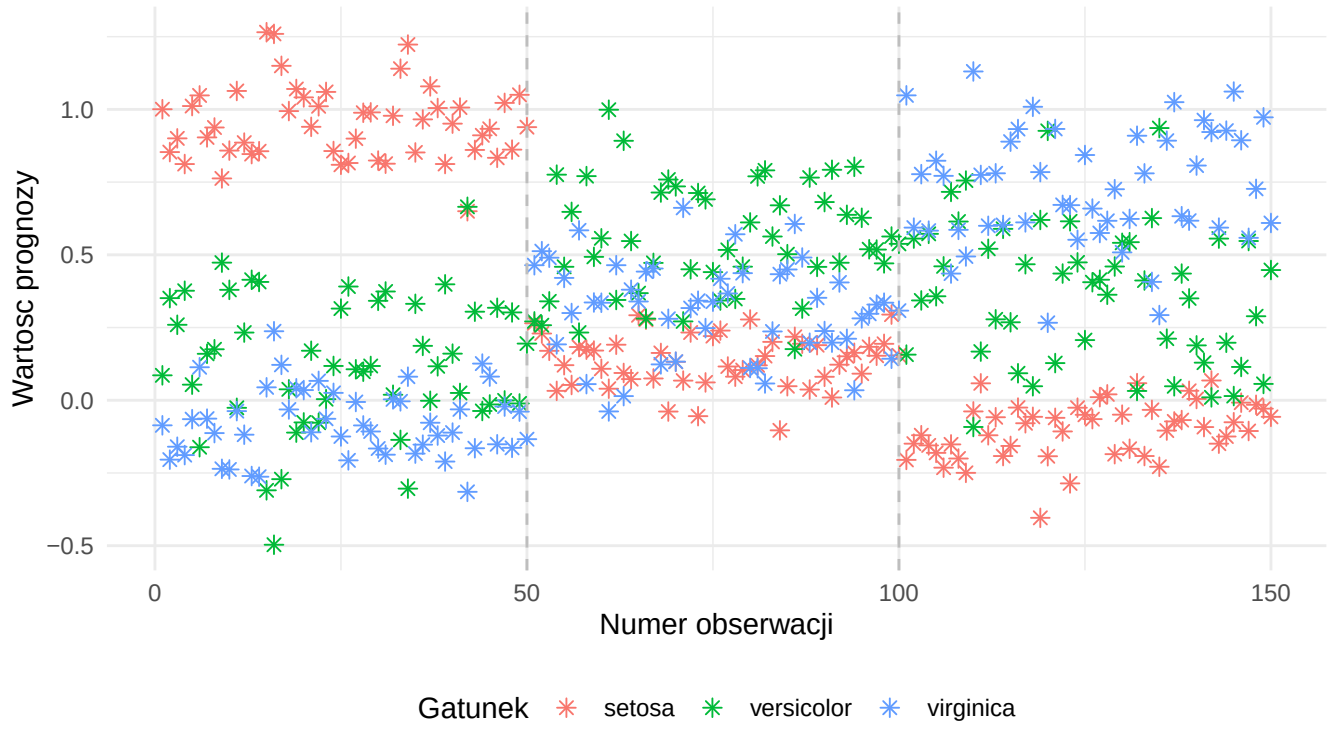
Prawdziwy gatunek	setosa	versicolor	virginica
setosa	19	0	0
versicolor	0	11	3
virginica	0	4	13

Błąd klasyfikacyjny na zbiorze testowym wynosi 14%. Nic nie wskazuje na maskowanie klas.

Tabela 6: Metryki klasyfikacji dla każdego gatunku

Prawdziwy gatunek	Dokładność	Precyzja	Czułośc	F1-score
setosa	1.00	1.00	1.00	1.00
versicolor	0.86	0.73	0.79	0.76
virginica	0.86	0.81	0.76	0.79
Średnie	0.91	0.85	0.85	0.85

Na Wykresie 2. przedstawiono wartości predykcyjne dla każdej obserwacji. Podobnie widać że nie występuje maskowanie klas, jednakże można zauważyć, podobne wartości prognozy dla obserwacji z klasy `versicolor` oraz `virginica`.



Wykres 2: Wartości predykcyjne dla każdej obserwacji

Średnia dokładność dla każdego gatunku wynosi 91%, a F1-score 85%. Jest to wartość mniejsza niż na zbiorze treningowym, lecz nadal wysoka, co wskazuje na dobrą zdolność modelu do generalizacji na nowych danych.

1.4 Budowa klasyfikatora w rozszerzonej przestrzeni cech

W tej części stworzono klasyfikator oparty na modelu regresji liniowej w rozszerzonej przestrzeni cech, uzyskanej poprzez tworzenie iloczynów par zmiennych.

```
iris1 <- iris

names(iris1) <- c("SL","SW","PL","PW","Species")

cechy <- names(iris1)[1:4]

grid <- expand.grid(cechy, cechy, stringsAsFactors = FALSE)
pairs <- grid[ grid$Var1 <= grid$Var2, ]

for (k in seq_len(nrow(pairs))) {
  i <- pairs$Var1[k]
  j <- pairs$Var2[k]
  nazwa <- paste(i, j, sep = ".")
  iris1[[nazwa]] <- iris1[[i]] * iris1[[j]]
}
```

Do trenowania i testowania modelu wykorzystano zbiory danych utworzone we wcześniejszym etapie poprzez podział w stosunku 2:1.

```r
train <- iris1[idx_train, ]
test  <- iris1[-idx_train, ]

etykiety <- train$Species
train_cechy <- train[, -5]
N <- nrow(train_cechy)
K <- length(unique(etykiety))

X.train <- as.matrix(cbind(rep(1, N), train_cechy))

Y.train <- matrix(0, nrow=N, ncol=K)
etykiety.num <- as.numeric(etykiety)

for (k in 1:K) {
  Y.train[etykiety.num==k, k] <- 1
}

B.hat <- solve(t(X.train)%*%X.train) %*% t(X.train) %*% Y.train
```

W Tabeli 7. zestawiono wartości współczynników regresji liniowej przypisanych poszczególnym cechom dla kategorii klasyfikacyjnych w zbiorze danych, uwzględniając rozszerzoną przestrzeń cech.

Tabela 7: Wartości współczynników regresji liniowej dla poszczególnych cech i kategorii - przestrzeń rozszerzona

Kombinacja Cech	Gatunki		
	Setosa	Versicolor	Virginica
1	-1.358	4.874	-2.516
SL	0.793	-2.660	1.867
SW	0.626	-0.027	-0.598
PL	-0.901	3.220	-2.320
PW	0.139	-4.054	3.915
SL.SL	-0.031	0.302	-0.272
PL.SL	-0.050	-0.034	0.083
PW.SL	0.056	-0.131	0.075
SL.SW	-0.088	-0.175	0.263
SW.SW	-0.044	0.181	-0.136
PL.SW	0.117	-0.404	0.287
PW.SW	-0.191	1.312	-1.121
PL.PL	0.099	-0.370	0.271
PL.PW	-0.080	0.886	-0.805
PW.PW	0.136	-1.189	1.053

1.5 Ewaluacja jakości modelu

W tej części dokonano ewaluacji jakości stworzonego modelu regresji liniowej.

Ewaluacja modelu na zbiorze treningowym.

```r
### Treningowy
Y.hat.train <- X.train %*% B.hat
pred_train <- apply(Y.hat.train, 1, which.max)

conf.matrix <- table(pred_train, train$Species)

accuracy_c <- sum(diag(conf.matrix))/sum(conf.matrix)
```

Wyniki klasyfikacji na zbiorze treningowym przedstawiono w macierzy pomyłek zawartej w Tabeli 8.

Tabela 8: Macierz pomyłek - zbiór treningowy

Prawdziwy gatunek	setosa	versicolor	virginica
setosa	31	0	0
versicolor	0	35	1
virginica	0	0	33

Tym razem błąd klasyfikacyjny na zbiorze treningowym wynosi jedynie 1%. Jest to bardzo dobry wynik, świadczący o wysokiej skuteczności modelu. Nic nie wskazuje na maskowanie klas.

Tabela 9: Metryki klasyfikacji dla każdego gatunku

Prawdziwy gatunek	Dokładność	Precyzja	Czułośc	F1-score
setosa	1.00	1.00	1.00	1.00
versicolor	0.99	1.00	0.97	0.99
virginica	0.99	0.97	1.00	0.99
Średnie	0.99	0.99	0.99	0.99

Średnia dokładność dla każdego gatunku wynosi 99%, a F1-score 99%. Są to znakomite wyniki, świadczące o bardzo wysokiej skuteczności klasyfikatora w rozróżnianiu poszczególnych klas.

Ewaluzacja modelu na zbiorze testowym.

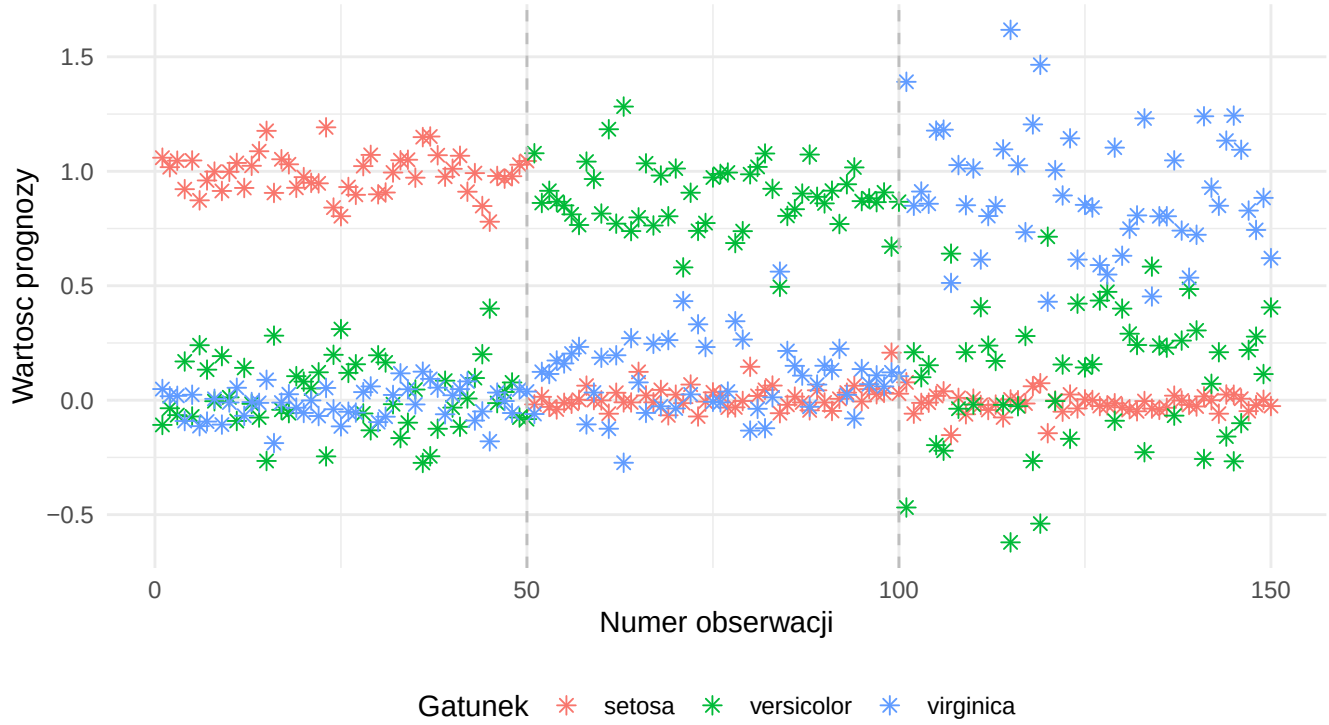
```r
### Testowy
test_cechy <- test[, -5]
N <- nrow(test_cechy)

X.test <- as.matrix(cbind(rep(1, N), test_cechy))

Y.hat.test <- X.test %*% B.hat
pred_test <- apply(Y.hat.test, 1, which.max)

conf.matrix <- table(pred_test, test$Species)

accuracy_d <- sum(diag(conf.matrix))/sum(conf.matrix)
```

Tabela 10: Macierz pomyłek - zbiór testowy

Prawdziwy gatunek	setosa	versicolor	virginica
setosa	19	0	0
versicolor	0	14	2
virginica	0	1	14

Tym razem błąd klasyfikacyjny na zbiorze treningowym wynosi jedynie 6%. Nic nie wskazuje na maskowanie klas.

Na Wykresie 3. przedstawiono wartości predykcyjne dla każdej obserwacji w przestrzeni rozszerzoenj. Podobnie widać że nie występuje maskowanie klas, jednak teraz wyraźniej widać jak dobrze model rozdziela klase.



Wykres 3: Wartości predykcyjne dla każdej obserwacji - przestrzeń rozszerzona

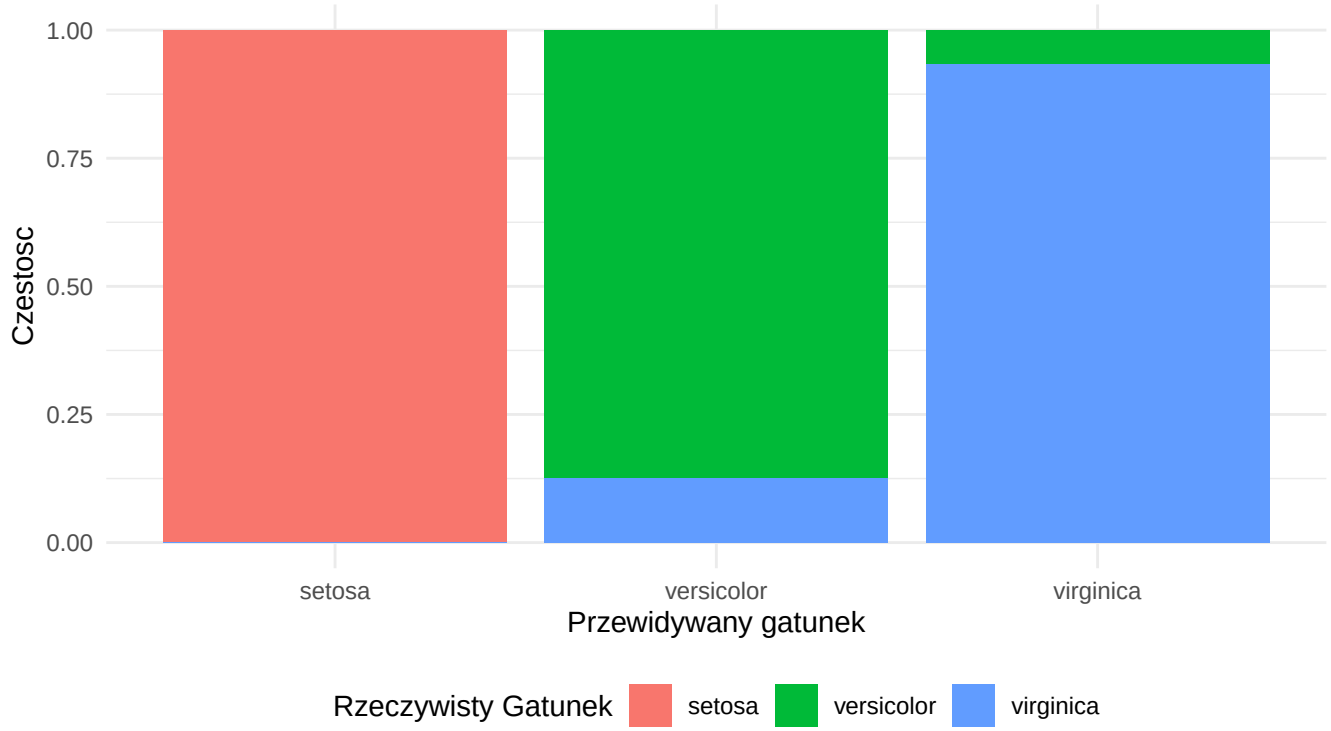
Tabela 11: Metryki klasyfikacji dla każdego gatunku - przestrzeń rozszerzona

Prawdziwy gatunek	Dokładność	Precyzja	Czułośc	F1-score
setosa	1.00	1.00	1.00	1.00
versicolor	0.94	0.93	0.88	0.90
virginica	0.94	0.88	0.93	0.90
Średnie	0.96	0.94	0.94	0.94

Średnia dokładność dla każdego gatunku wynosi 96%, a F1-score 94%. Wyniki te są lepsze niż w przypadku klasyfikacji bez zastosowania rozszerzonej przestrzeni cech.

Przedstawiono macierz pomyłek w formie wykresu słupkowego, aby umożliwić przejrzystą wizualizację zdolności klasyfikacyjnej modelu oraz lepsze zobrazowanie rozkładu poprawnych i błędnych predykcji dla poszczególnych klas.

Wyniki przewidywać przedstawiono na Wykresie 4.



Wykres 4: Porównanie przewidywanych i rzeczywistych klas

1.6 Podsumowanie

Regresja liniowa pozwoliła nam na modelowanie zależności między zmiennymi oraz przypisanie obserwacjom wartości wyjściowych, które można interpretować jako **przybliżone prawdopodobieństwo przynależności do klasy**.

Mimo że do trenowania modelu używaliśmy etykiet binarnych (100% lub 0% przynależności do danej klasy), przewidywane przez model wyniki mogą przyjmować wartości spoza przedziału $[0, 1]$ oraz nie sumują się do 1, przez co nie można ich interpretować bezpośrednio jako prawdopodobieństw. W celu uzyskania poprawnej interpretacji probabilistycznej, stosuje się funkcję **softmax**, która przekształca wyjścia modelu na prawdopodobieństwa przypisania do poszczególnych klas.

W Tabeli 12. przedstawiono zmiany najważniejszych miar klasyfikacji po zastosowaniu rozszerzonej przestrzeni cech.

Tabela 12: Podsumowanie metryk klasyfikacji w różnych przestrzeniach i zbiorach

	Zbiór treningowy		Zbiór testowy	
	Przestrzeń podstawowa	Przestrzeń rozszerzona	Przestrzeń podstawowa	Przestrzeń rozszerzona
Błąd klasyfikacyjny	12.00	1.00	14.00	6.00
Średnia dokładność	92.00	99.33	90.67	96.00
Średni F1 - score	88.41	99.03	84.88	93.55

Dzięki wprowadzeniu rozszerzonych cech wszystkie miary jakości klasyfikacji uległy wyraźnej poprawie.

2 Porównanie metod klasyfikacji

W tej części dokonamy porównania wybranych metod klasyfikacyjnych: k-NN, drzew decyzyjnych oraz naiwnego klasyfikatora bayesowskiego, analizując ich skuteczność przy różnych konfiguracjach parametrów. Oceny modeli dokonamy na podstawie miar takich jak dokładność oraz F1-score, stosując techniki wielokrotnego podziału na zbiór treningowy i testowy, cross-validation oraz bootstrap.

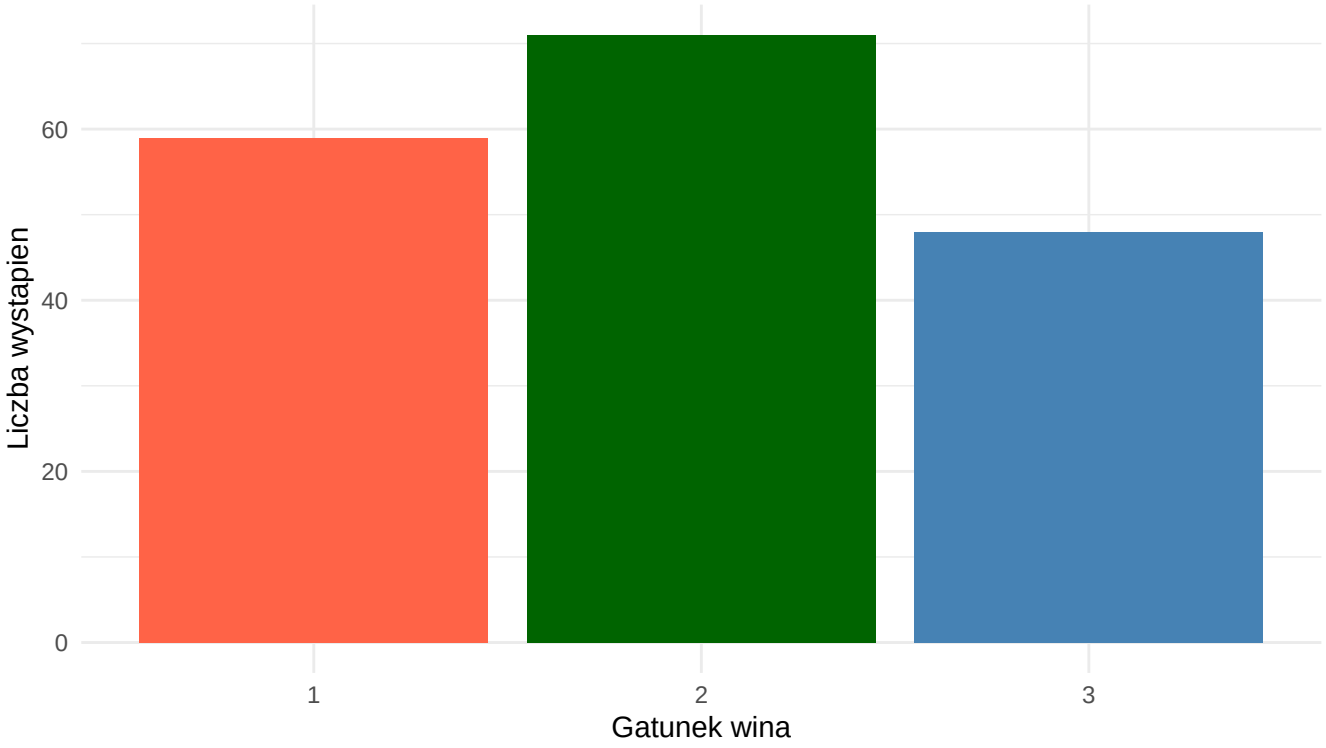
2.1 Charaktetystyka danych

Zbiór danych *wine* zawiera **178** przypadków trzech rodzajów włoskich win oraz **14** cech. Liczba brakujących danych wynosi **0**.

Znaczenie poszczególnych cech oraz ich typ przedstawiono w tabeli 13.

Tabela 13: Opis zmiennych w zbiorze danych wine

Zmienna	Typ	Opis
Gatunek	factor	Klasa (typ wina: 1, 2, 3)
Alkohol	numeric	Zawartość alkoholu [%]
Kwas_mlekowy	numeric	Kwas jabłkowy [g/l]
Popiół	numeric	Zawartość popiołu [g/l]
Zasadowość_popiołu	numeric	Zasadowość pozostałości popiołu
Magnez	numeric	Zawartość magnezu [mg/l]
Fenole_ogółem	numeric	Fenole ogółem [g/l]
Flawonoidy	numeric	Zawartość flawonoidów [g/l]
Fenole_nieflawonoidowe	numeric	Fenole niefalwonoidowe [g/l]
Proantocyjanidyny	numeric	Proantocyjanidyny [g/l]
Koloru	numeric	Intensywność koloru
Barwa	numeric	Odcień barwy
OD280_OD315	numeric	Stosunek absorbancji przy 280/315 nm dla rozcieńczonych win
Prolina	numeric	Zawartość proliny [mg/l]

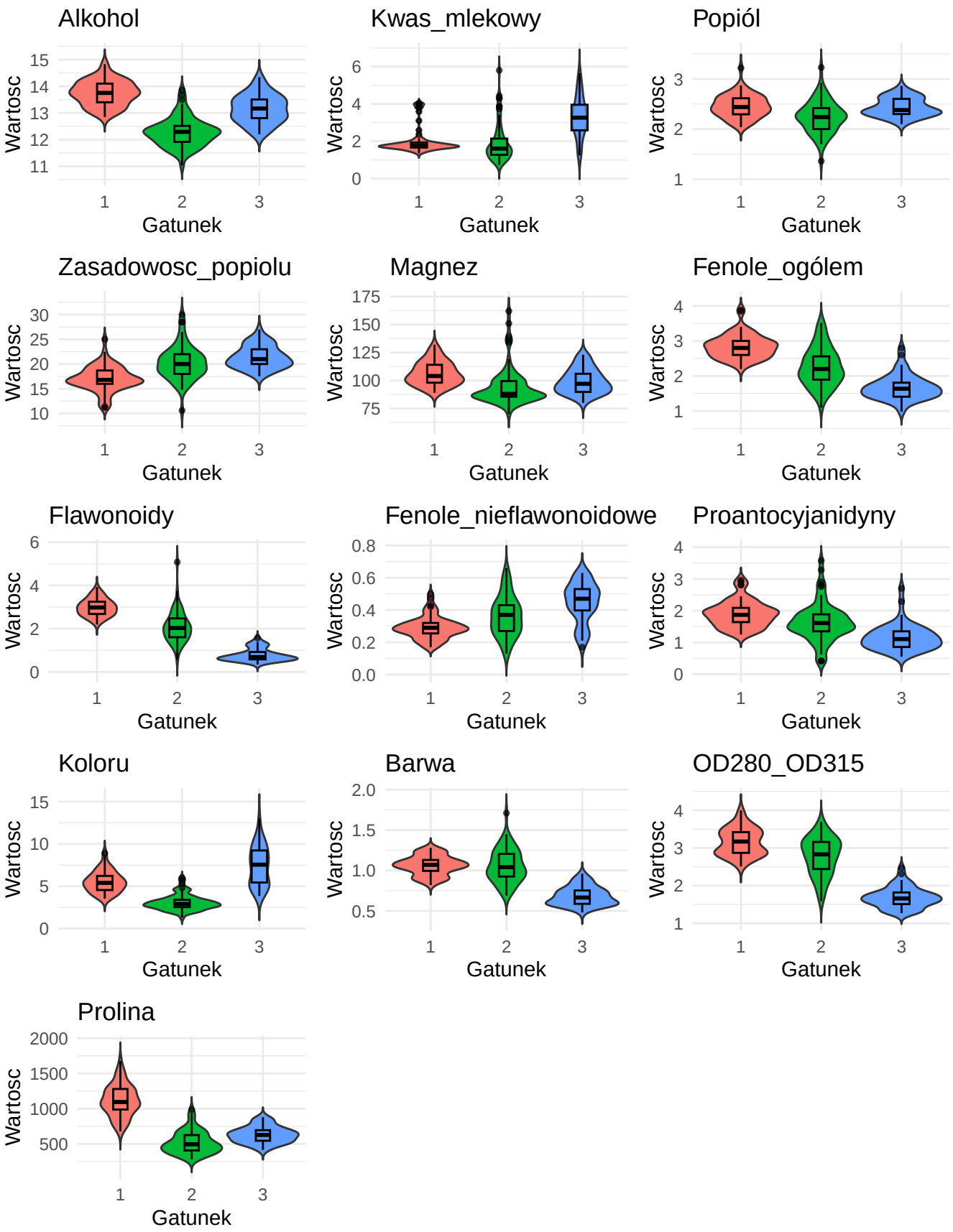


Wykres 5: Liczba występowania gatunków win w zbiorze wine

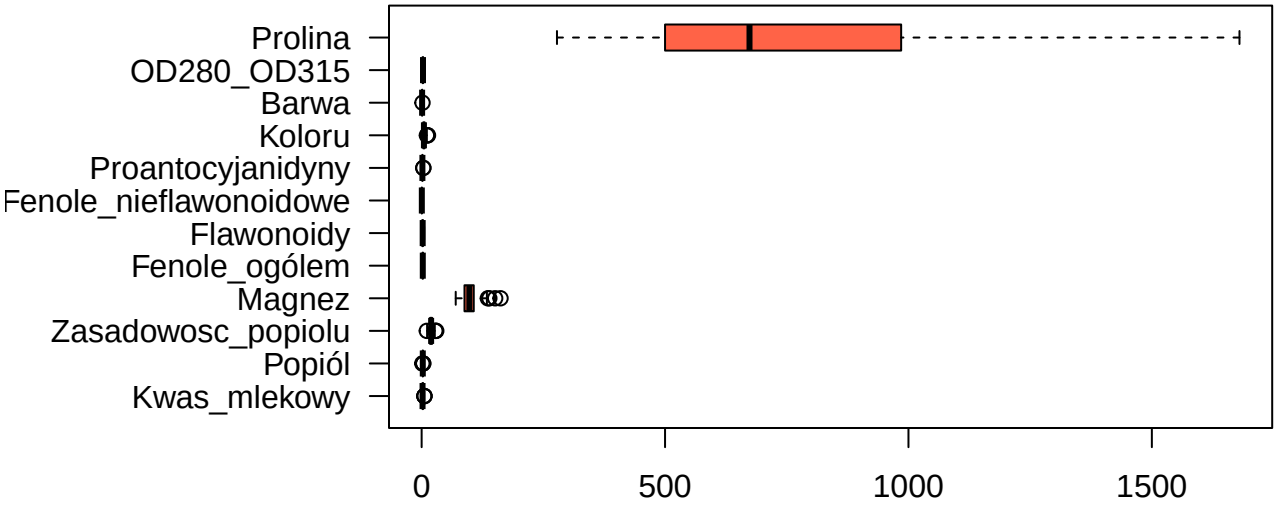
Z Wykresu 5. można odczytać, że liczba obserwacji poszczególnych klas w zbiorze danych jest zróżnicowana. Najwięcej jest win z gatunku drugiego, a najmniej z gatunku trzeciego.

Gdybyśmy przypisali wszystkie obserwacje do najczęściej występującej klasy, uzyskalibyśmy dokładność na poziomie 39.89%.

Na Wykresie 6. przedstawiono rozkłady zmiennych ilościowych z podziałem na gatunek. W zbiorze danych nie występuje zmienna, która jednoznacznie pozwalałaby na rozróżnienie klas, jednak wstępna analiza sugeruje, że zmienna `Flawonoidy` cechuje się dobrą zdolnością dyskryminacyjną. Szczegółowy wybór najlepszych cech klasyfikacyjnych zostanie omówiony w dalszej części raportu.

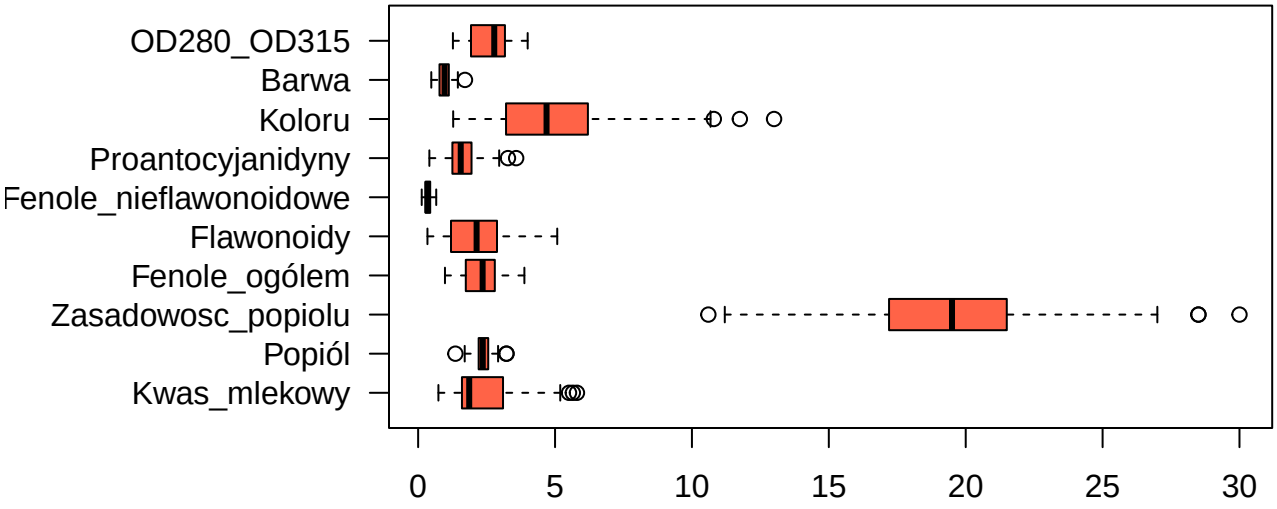


Wykres 6: Rozkłady zmiennych ilościowych z podziałem na gatunek



Wykres 7: Wykresy pudełkowe zmiennych numerycznych

Z Wykresu 7. można odczytać, że zmienne Magnez i Prolina przyjmują znacznie większe wartości w porównaniu do pozostałych cech. Wynika to z ich naturalnego zakresu oraz jednostek miary, co wpływa na różnice w skali danych. W związku z tym, w przypadku metody k-NN, niezbędne będzie zastosowanie odpowiedniej standaryzacji cech.



Wykres 8: Wykresy pudełkowe zmiennych numerycznych - bez Magnezu i Proliny

Po usunięciu tych dwóch cech (Wykres 8) zauważalna jest nadal istotna zmienność wśród pozostałych zmiennych.

2.2 Budowa i ewaluacja klasyfikatorów

W tej części zajęto się konstrukcją oraz oceną skuteczności wybranych modeli klasyfikacyjnych. Ze względu na charakterystykę działania klasyfikatora rozważano różne liczby sąsiadów, odmienne proporcje podziału danych na zbiory treningowe i testowe oraz wykorzystanie jedynie wybranych zmiennych wejściowych. Ponadto omówiono potrzebę standaryzacji danych

2.2.1 Klasyfikator k-NN

W przypadku klasyfikatora k-NN istotną rolę odgrywa sposób obliczania odległości pomiędzy punktami, dlatego należy rozważyć konieczność przeprowadzenia standaryzacji zmiennych.

2.2.1.1 Najlepsza standaryzacja

W celu wyboru względnie najlepszego wariantu klasyfikacji przeprowadzimy ocenę zdolności klasyfikacyjnych modelu, dzieląc dane na zbiór uczący i testowy w stosunku 2:1 oraz przyjmując liczbę sąsiadów równą 5, a w przypadku metody cross-validation zbiór podzielono na 10 cześci. Walidacji dokonano jedynie na **zbiorze testowym**.

Standaryzację przeprowadzimy, testując różne warianty przekształceń danych:

Ze względu na parametr lokalizacji:

- brak standaryzacji,
- odjęcie średniej,
- odjęcie mediany,
- odjęcie wartości minimalnej.

Ze względu na parametr rozproszenia:

- brak standaryzacji,
- podzielenie przez odchylenie standardowe,
- podzielenie przez rozstęp,
- podzielenie przez rozstęp międzykwartylowy.

```
scale_plus <- function(data, shift = "none", scale = "none") {
  scale_columns <- function(x) {
    if (shift == "mean") x <- x - mean(x)
    else if (shift == "median") x <- x - median(x)
    else if (shift == "min") x <- x - min(x)

    if (scale == "sd") x <- x / sd(x)
    else if (scale == "range") x <- x / (max(x) - min(x))
    else if (scale == "iqr") x <- x / IQR(x)

    return(x)
  }
  as.data.frame(lapply(data, scale_columns))
}
```

```
wielokrotny_podział <- function(data = wine,
                                K = 100,
                                podzial = 2/3,
                                sasiedzi = 5) {
  # Wielokrotny podział
  n <- nrow(data)

  sum_acc_train <- 0
  sum_acc_test <- 0
  sum_metrics_train <- NULL
  sum_metrics_test <- NULL
```

```
  for (i in seq_len(K)) {
    train_idx <- sample(seq_len(n), size = floor(podzial * n))
    train.set <- data[train_idx, ]
    test.set <- data[-unique(train_idx), ]

    pred_train <- knn(
      train = train.set[, setdiff(names(train.set), "Gatunek"), drop = FALSE],
      test  = train.set[, setdiff(names(train.set), "Gatunek"), drop = FALSE],
      cl    = train.set$Gatunek,
      k     = sasiedzi
    )
    cm_train <- table(Prawdziwa = train.set$Gatunek, Przewidziana = pred_train)
    acc_train <- sum(diag(cm_train)) / sum(cm_train)
    metrics_train <- metryki(cm_train)

    pred_test <- knn(
      train = train.set[, setdiff(names(train.set), "Gatunek"), drop = FALSE],
      test  = test.set[, setdiff(names(test.set), "Gatunek"), drop = FALSE],
      cl    = train.set$Gatunek,
      k     = sasiedzi
    )
    cm_test <- table(Prawdziwa = test.set$Gatunek, Przewidziana = pred_test)
    acc_test <- sum(diag(cm_test)) / sum(cm_test)
    metrics_test <- metryki(cm_test)

    sum_acc_train <- sum_acc_train + acc_train
    sum_acc_test <- sum_acc_test + acc_test

    if (is.null(sum_metrics_train)) {
      sum_metrics_train <- metrics_train
      sum_metrics_test <- metrics_test
    } else {
      sum_metrics_train <- sum_metrics_train + metrics_train
      sum_metrics_test <- sum_metrics_test + metrics_test
    }
  }

  avg_acc_train <- sum_acc_train / K
  avg_acc_test <- sum_acc_test / K
  avg_metrics_train <- sum_metrics_train / K
  avg_metrics_test <- sum_metrics_test / K

  return(list(
    accuracy_train = avg_acc_train,
    accuracy_test  = avg_acc_test,
    metrics_train  = avg_metrics_train,
    metrics_test   = avg_metrics_test
  ))
}
```

```
shifts <- c("none", "mean", "median", "min")
scales <- c("none", "sd", "range", "iqr")
```

```r
accuracy_matrix <- matrix(NA,
                      nrow = 4,
                      ncol = 4,
                      dimnames = list(Shift = shifts, Scale = scales))

f1_matrix <- matrix(NA,
                      nrow = 4,
                      ncol = 4,
                      dimnames = list(Shift = shifts, Scale = scales))

for (sh in shifts) {
  for (sc in scales) {
    wine_scale <- scale_plus(wine[-1], scale = sc, shift = sh)
    wine_scale$Gatunek <- wine$Gatunek

    results <- wielokrotny_podział(wine_scale)

    avg_accuracy <- results$accuracy_test
    accuracy_matrix[sh, sc] <- avg_accuracy

    avg_f1 <- results$metrics_test$F1_Score[4]
    f1_matrix[sh, sc] <- avg_f1
  }
}
```

W Tabeli 14. przedstawiono średnie wartości dokładności dla różnych kombinacji parametrów rozproszenia i lokalizacji, obliczone metodą wielokrotnego podziału Okazuje się że najlepsze wyniki osiąga standaryzacja minimum i odchyleniem standardowym, lecz podobne wyniki osiągją również kombinacje ze średnią, medianą i rozstępem. Również standaryzacja rozporoszeniem daje lepsze wyniki, niż ta lokalizacją.

Tabela 14: Średnia dokładność klasyfikacji dla kombinacji parametrów rozproszenia i lokalizacji - wielokrotny podział - KNN

Parametr lokalizacji	Parametr rozproszenia			
	Brak	Odch. standardowe	Rozstęp	Rozstęp międzykwartylowy
Brak	69.17	96.18	95.42	94.43
Średnia	69.12	95.65	95.65	94.77
Mediana	68.72	95.52	95.82	94.33
Minimum	69.67	95.92	95.05	94.68

W Tabeli 15. przedstawiono średnie wartości F1-score dla różnych kombinacji parametrów rozproszenia i lokalizacji, obliczone metodą wielokrotnego podziału. Tym razem brak standaryzacji lokalizacją i standaryzacja odchyleniem standardowym przynosi najlepsze wyniki.

Tabela 15: Średni F1-score dla kombinacji parametrów rozproszenia i lokalizacji - wielokrotny podział - KNN

Parametr lokalizacji	Parametr rozproszenia			
	Brak	Odch. standardowe	Rozstęp	Rozstęp międzykwartylowy
Brak	67.67	96.24	95.36	94.50
Średnia	67.66	95.68	95.59	94.82
Mediana	67.23	95.52	95.82	94.43
Minimum	68.24	95.91	94.95	94.78

```
cross_validation <- function(data = wine, K = 10, sasiedzi = 5) {
  # Cross-validation
  n <- nrow(data)
  sum_acc_train <- 0
  sum_acc_test  <- 0
  sum_metrics_train <- NULL
  sum_metrics_test  <- NULL

  folds <- sample(rep(seq_len(K), length.out = n))

  for (i in seq_len(K)) {
    train.set <- data[folds != i, ]
    test.set  <- data[folds == i, ]

    pred_train <- knn(
      train = train.set[, setdiff(names(train.set), "Gatunek"), drop = FALSE],
      test  = train.set[, setdiff(names(train.set), "Gatunek"), drop = FALSE],
      cl    = train.set$Gatunek,
      k     = sasiedzi
    )
    cm_train <- table(Prawdziwa = train.set$Gatunek, Przewidziana = pred_train)
    acc_train <- sum(diag(cm_train)) / sum(cm_train)
    metrics_train <- metryki(cm_train)

    pred_test <- knn(
      train = train.set[, setdiff(names(train.set), "Gatunek"), drop = FALSE],
      test  = test.set[, setdiff(names(test.set), "Gatunek"), drop = FALSE],
      cl    = train.set$Gatunek,
      k     = sasiedzi
    )
    cm_test <- table(Prawdziwa = test.set$Gatunek, Przewidziana = pred_test)
    acc_test <- sum(diag(cm_test)) / sum(cm_test)
    metrics_test <- metryki(cm_test)

    sum_acc_train <- sum_acc_train + acc_train
    sum_acc_test  <- sum_acc_test  + acc_test
    if (is.null(sum_metrics_train)) {
      sum_metrics_train <- metrics_train
      sum_metrics_test  <- metrics_test
    } else {
      sum_metrics_train <- sum_metrics_train + metrics_train
```

```
      sum_metrics_test  <- sum_metrics_test  + metrics_test
    }
  }

  avg_acc_train    <- sum_acc_train / K
  avg_acc_test     <- sum_acc_test  / K
  avg_metrics_train <- sum_metrics_train / K
  avg_metrics_test  <- sum_metrics_test  / K

  return(list(
    accuracy_train = avg_acc_train,
    accuracy_test  = avg_acc_test,
    metrics_train  = avg_metrics_train,
    metrics_test   = avg_metrics_test
  ))
}
```

W Tabeli 16. przedstawiono średnie wartości dokładności dla różnych kombinacji parametrów rozproszenia i lokalizacji, obliczone metodą cross-validation. Analiza wykazała, że najlepsze wyniki uzyskuje standaryzacja oparta na minimum oraz odchyleniu standardowemu.

Tabela 16: Średnia dokładność klasyfikacji dla kombinacji parametrów rozproszenia i lokalizacji - cross-validation - KNN

Parametr lokalizacji	Parametr rozproszenia			
	Brak	Odch. standardowe	Rozstęp	Rozstęp międzykwartylowy
Brak	66.27	95.46	95.52	94.97
Średnia	69.61	96.63	95.52	96.08
Mediana	72.55	95.52	94.90	94.90
Minimum	68.63	97.16	94.90	95.46

W Tabeli 17. przedstawiono średnie wartości F1-score dla różnych kombinacji parametrów rozproszenia i lokalizacji, obliczone metodą cross-validation. Podobnie najlepsze wyniki dla standaryzacji minimum oraz dochyleniem standardowym

Tabela 17: Średni F1-score dla kombinacji parametrów rozproszenia i lokalizacji - wielokrotny podział - KNN

Parametr lokalizacji	Parametr rozproszenia			
	Brak	Odch. standardowe	Rozstęp	Rozstęp międzykwartylowy
Brak	63.26	95.06	95.09	94.69
Średnia	67.50	96.50	95.14	95.28
Mediana	68.56	95.52	94.06	93.72
Minimum	65.10	97.13	93.69	95.47

```
bootstrap <- function(data = wine, K = 100, podzial = 2/3, sasiedzi = 5) {
  # Bootstrap
```

```r
n <- nrow(data)

sum_acc_train <- 0
sum_acc_test <- 0
sum_metrics_train <- NULL
sum_metrics_test <- NULL

for (i in seq_len(K)) {
  train_idx <- sample(seq_len(n), size = floor(podzial * n), replace = TRUE)
  train.set <- data[train_idx, ]
  test.set <- data[-unique(train_idx), ]

  pred_train <- knn(
    train = train.set[, setdiff(names(train.set), "Gatunek"), drop = FALSE],
    test  = train.set[, setdiff(names(train.set), "Gatunek"), drop = FALSE],
    cl    = train.set$Gatunek,
    k     = sasiedzi
  )
  cm_train <- table(Prawdziwa = train.set$Gatunek, Przewidziana = pred_train)
  acc_train <- sum(diag(cm_train)) / sum(cm_train)
  metrics_train <- metryki(cm_train)

  pred_test <- knn(
    train = train.set[, setdiff(names(train.set), "Gatunek"), drop = FALSE],
    test  = test.set[, setdiff(names(test.set), "Gatunek"), drop = FALSE],
    cl    = train.set$Gatunek,
    k     = sasiedzi
  )
  cm_test <- table(Prawdziwa = test.set$Gatunek, Przewidziana = pred_test)
  acc_test <- sum(diag(cm_test)) / sum(cm_test)
  metrics_test <- metryki(cm_test)

  sum_acc_train <- sum_acc_train + acc_train
  sum_acc_test <- sum_acc_test + acc_test

  if (is.null(sum_metrics_train)) {
    sum_metrics_train <- metrics_train
    sum_metrics_test <- metrics_test
  } else {
    sum_metrics_train <- sum_metrics_train + metrics_train
    sum_metrics_test <- sum_metrics_test + metrics_test
  }
}

avg_acc_train <- sum_acc_train / K
avg_acc_test <- sum_acc_test / K
avg_metrics_train <- sum_metrics_train / K
avg_metrics_test <- sum_metrics_test / K

return(list(
  accuracy_train = avg_acc_train,
  accuracy_test  = avg_acc_test,
  metrics_train  = avg_metrics_train,
```

```
    metrics_test   = avg_metrics_test
  ))
}
```

W Tabeli 18. przedstawiono średnie wartości dokładności dla różnych kombinacji parametrów rozproszenia i lokalizacji, obliczone metodą bootstrap. Okazuje się że najlepsze wyniki osiąga tym razem standarzyacja średnią i rozstępem, lecz podobne wyniki osiągją również kombinacje z medianą i odchyleniem standardowym.

Tabela 18: Średnia dokładność klasyfikacji dla kombinacji parametrów rozproszenia i lokalizacji - bootstrap - KNN

Parametr lokalizacji	Parametr rozproszenia			
	Brak	Odch. standardowe	Rozstęp	Rozstęp międzykwartylowy
Brak	68.12	94.75	94.60	93.83
Średnia	67.72	94.63	94.77	93.32
Mediana	68.80	94.75	94.72	93.35
Minimum	68.36	94.26	94.72	93.90

W Tabeli 19. przedstawiono średnie wartości F1-score dla różnych kombinacji parametrów rozproszenia i lokalizacji, obliczone metodą bootstrap. Najlepsze wyniki wystepują dla standaryzacji mediana i odchyleniem standardowym, lecz jest on podobny do innych kombinacji standaryzacji.

Tabela 19: Średni F1-score dla kombinacji parametrów rozproszenia i lokalizacji - wielokrotny podział - KNN

Parametr lokalizacji	Parametr rozproszenia			
	Brak	Odch. standardowe	Rozstęp	Rozstęp międzykwartylowy
Brak	66.71	94.82	94.63	93.97
Średnia	66.40	94.72	94.80	93.45
Mediana	67.48	94.86	94.78	93.49
Minimum	66.99	94.38	94.77	94.02

Na podstawie danych przedstawionych w Tabelach 14., 15., 16., 17., 18. i 19. podjęto decyzję, że do dalszej analizy zdolności klasyfikacyjnych metody k-NN zostanie zastosowana standaryzacja oparta na średniej oraz odchyleniu standardowym. Zdaje się jednak sprawę, że zastosowanie rozstępu lub mediany mogłoby potencjalnie przynieść lepsze wyniki, niemniej rozszerzenie analizy o te metody byłoby nieuzasadnione, gdyż prawdopodobnie nie wpłynęłoby znacząco na poprawę efektywności klasyfikacji.

2.2.1.2 Najlepszy model Po dokonaniu standaryzacji przeprowadzono ocenę zdolności klasyfikacyjnej modelu dla różnych proporcji podziału zbioru danych oraz różnych wartości liczby sąsiadów. Wybrano następujące udziały zbioru uczącego: 1/20, 1/10, 1/5, 1/3, 1/2, 2/3, 4/5, 9/10 oraz 19/20, a także liczby sąsiadów: 1, 2, 3, 4, 5, 10 i 20 dla metody wielokrotnego podziału oraz bootstrap. Dla metody cross-validation zbiór podzielono na 2, 5, 10, 20, 50 i 178 (Leave-One-Out) części.

Na podstawie wyników postarano się wybrać najlepszy sposób podziału. Jako kryterium dobrego wybrano największe wartości średniej dokładności klasyfikacji oraz średniego F1-score.

W pierwszej kolejności dokonano sprawdzenia modelu za pomocą metody wielokrotnego podziału na zbiór uczący i testowy.

Tabela 20: Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - wielokrotny podział - zbiór treningowy - KNN

Podział zbioru	Liczba sąsiadów						
	1	2	3	4	5	10	20
1/20	100	91.12	88.88	83.12	74.38	49.25	51.62
1/10	100	95.00	95.24	93.12	92.00	75.24	45.18
1/5	100	96.23	97.43	96.00	96.63	95.06	78.74
1/3	100	96.98	97.44	97.20	97.00	96.25	95.14
1/2	100	97.42	97.36	96.65	97.37	96.78	96.88
2/3	100	97.64	97.09	96.99	97.60	96.91	97.01
4/5	100	97.54	96.58	96.91	97.78	97.26	97.01
9/10	100	97.62	96.45	96.82	97.76	97.60	97.36
19/20	100	97.86	96.18	96.67	97.79	97.73	97.39

Z Tabeli 20. wynika, że dokładność na zbiorze treningowym maleje wraz ze wzrostem liczby sąsiadów i zmniejszeniem rozmiaru zbioru treningowego. Jest to zgodne z intuicją, gdyż większa liczba sąsiadów uśrednia decyzje modelu, a mniejszy zbiór dostarcza mniej informacji do nauki. Dodatkowo, wzrost liczby sąsiadów zwiększa czas predykcji oraz obciążenie pamięciowe modelu.

Tabela 21: Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - wielokrotny podział - zbiór testowy - KNN

Podział zbioru	Liczba sąsiadów						
	1	2	3	4	5	10	20
1/20	84.09	74.65	73.98	62.41	55.87	34.26	34.67
1/10	90.09	87.12	88.86	86.36	87.24	62.75	35.19
1/5	92.83	91.29	93.87	93.20	93.74	91.66	69.72
1/3	93.99	92.98	94.81	94.01	94.66	95.16	93.74
1/2	94.40	93.93	95.06	94.58	95.69	95.64	96.12
2/3	95.00	94.42	95.75	95.25	96.20	96.02	96.72
4/5	95.11	94.69	96.14	95.72	96.03	96.28	96.53
9/10	95.28	95.39	95.06	95.94	97.72	97.00	96.83
19/20	96.78	93.78	96.33	96.11	96.67	96.78	96.89

Z Tabeli 21. wynika, że ogólny trend obserwowany na zbiorze treningowym został zachowany także na zbiorze testowym. Najlepsze wyniki model osiąga przy jednej liczbie sąsiadów oraz przy większym udziale danych przeznaczonych do treningu (podział 19:20), co sugeruje dobrą separowalność klas w przestrzeni cech. Może to również sugerować, że model nie ulega overfittingowi, a wręcz przeciwnie – czerpie korzyść z większej liczby przykładów uczących.

Tabela 22: Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - wielokrotny podział - zbiór treningowy - KNN

Podział zbioru	Liczba sąsiadów						
	1	2	3	4	5	10	20
1/20	96	80.10	75.25	63.72	51.84	23.94	24.72
1/10	100	93.55	93.67	89.73	87.83	60.42	22.18
1/5	100	96.16	97.42	95.93	96.48	93.97	68.67
1/3	100	96.99	97.50	97.27	97.04	96.34	94.93
1/2	100	97.45	97.40	96.72	97.41	96.83	96.96
2/3	100	97.71	97.15	97.04	97.66	96.95	97.08
4/5	100	97.62	96.67	96.97	97.83	97.31	97.06
9/10	100	97.70	96.54	96.89	97.80	97.66	97.42
19/20	100	97.93	96.27	96.73	97.83	97.80	97.47

Tabela 23: Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - wielokrotny podział - zbiór testowy - KNN

Podział zbioru	Liczba sąsiadów						
	1	2	3	4	5	10	20
1/20	82.83	72.39	69.18	54.60	47.15	19.54	20.15
1/10	90.26	87.16	88.67	85.99	86.04	55.62	18.90
1/5	92.99	91.51	94.03	93.37	93.90	91.52	64.52
1/3	94.13	93.15	94.89	94.16	94.79	95.27	93.91
1/2	94.49	94.01	95.12	94.68	95.73	95.70	96.20
2/3	95.02	94.46	95.75	95.25	96.20	96.01	96.70
4/5	95.17	94.76	96.02	95.65	95.88	96.29	96.49
9/10	94.76	94.72	94.73	95.48	97.80	96.71	96.63
19/20	94.48	90.13	94.26	94.15	93.39	93.96	94.92

Dane z Tabeli 22. oraz Tabeli 23. potwierdzają poprzednie obserwacje.

W drugiej kolejności dokonano sprawdzenia modelu za pomocą cross-validation.

Tabela 24: Średnia dokładność klasyfikacji dla kombinacji parametrów liczby podzbiorów oraz liczby sąsiadów - cross-validation - zbiór treningowy - KNN

Liczba podzbiorów	Liczba sąsiadów						
	1	2	3	4	5	10	20
2	100	97.19	96.63	96.07	95.51	97.19	96.07
5	100	97.89	96.49	96.49	97.47	97.33	97.19
10	100	97.94	96.38	96.44	97.94	97.56	97.25

Tabela 24: Średnia dokładność klasyfikacji dla kombinacji parametrów liczby podzbiorów oraz liczby sąsiadów - cross-validation - zbiór treningowy - KNN (kontynuacja)

Liczba podzbiorów	Liczba sąsiadów						
	1	2	3	4	5	10	20
20	100	97.87	96.24	96.72	97.75	97.96	97.43
50	100	97.76	96.12	96.78	97.76	97.79	97.44
178	100	97.77	96.08	96.62	97.76	97.77	97.50

Z Tabeli 24 wynikają podobne wnioski jak z analizy przeprowadzonej metodą wielokrotnego podziału. Warto jednak zauważyć, że porównywalne wyniki uzyskiwane są zarówno przy podziale zbioru na 5 części, jak i przy podziale na 178 części (walidacja Leave-One-Out). Świadczy to o stabilności modelu oraz niewielkiej wrażliwości na sposób podziału danych, co sugeruje dobrą zdolność generalizacji i ograniczoną podatność na nadmierne dopasowanie.

Tabela 25: Średnia dokładność klasyfikacji dla kombinacji parametrów liczby podzbiorów oraz liczby sąsiadów - cross-validation - zbiór testowy - KNN

Liczba podzbiorów	Liczba sąsiadów						
	1	2	3	4	5	10	20
2	94.94	92.70	94.94	93.26	96.63	93.82	94.94
5	95.56	94.95	94.95	96.65	96.60	96.67	97.19
10	95.52	95.52	95.46	96.08	95.49	96.11	96.67
20	95.42	94.93	95.49	94.44	97.22	97.78	97.22
50	95.33	94.67	95.33	96.33	97.17	97.33	97.00
178	95.51	94.94	95.51	95.51	97.19	97.75	97.19

Z Tabeli 25. wynikają podobne wnioski jak w poprzedniej części sprawozdania. Warto jednak zauważyć, że użycie 3 sąsiadów często przynosi lepsze wyniki niż zastosowanie 2 lub 4 sąsiadów, podobnie jak w przypadku 20 sąsiadów. Mimo to, najlepsze rezultaty uzyskuje się przy zastosowaniu jednego sąsiada.

Tabela 26: Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - cross-validation - zbiór treningowy - KNN

Liczba podzbiorów	Liczba sąsiadów						
	1	2	3	4	5	10	20
2	100	97.06	96.61	96.11	95.48	97.40	96.14
5	100	97.93	96.59	96.54	97.52	97.40	97.26
10	100	98.01	96.47	96.48	97.97	97.64	97.31
20	100	97.94	96.34	96.78	97.80	98.03	97.51
50	100	97.85	96.22	96.85	97.81	97.86	97.53
178	100	97.85	96.18	96.68	97.80	97.85	97.59

Z Tabeli 26. nie wynikają żadne nowe wnioski.

Tabela 27: Średni F1-score dla kombinacji parametrów liczby podzbiorów oraz liczby sąsiadów - cross-validation - zbiór testowy - KNN

Liczba podzbiorów	Liczba sąsiadów						
	1	2	3	4	5	10	20
2	95.12	92.66	94.97	93.45	96.68	94.19	95.04
5	95.75	94.97	95.20	96.83	96.49	96.69	97.29
10	95.73	95.04	95.11	96.01	95.54	96.33	96.24
20	93.66	92.83	93.86	90.42	94.18	94.37	95.56
50	69.81	76.38	67.82	67.28	74.00	74.71	80.31
178	31.84	31.65	31.84	31.84	32.40	32.58	32.40

Tabela 27. prezentuje zupełnie odmienne wyniki. Średni F1-score zasadniczo spada wraz ze wzrostem liczby podzbiorów oraz liczby sąsiadów. W przypadku większej liczby podziałów, mimo że każdy zbiór treningowy jest niemal pełny (np. w walidacji Leave-One-Out), model jest oceniany na bardzo niewielkich zbiorach testowych, co może powodować większą wariancję wyników i niższą stabilność estymacji skuteczności. Z kolei wzrost liczby sąsiadów prowadzi do silniejszego uśredniania decyzji klasyfikatora, co może skutkować utratą zdolności do wychwycenia lokalnych wzorców, a tym samym spadkiem dokładności.

Jako ostatnią dokonano sprawdzenia modelu za pomocą metody bootstrap.

Tabela 28: Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - bootstrap - zbiór treningowy - KNN

Podział zbioru	Liczba sąsiadów						
	1	2	3	4	5	10	20
1/20	100	90.38	89.38	81.75	76.38	52.12	52.00
1/10	100	95.47	96.06	93.82	93.41	76.53	45.59
1/5	100	96.51	97.29	96.09	96.06	94.40	78.57
1/3	100	97.66	97.59	96.80	96.46	95.90	94.54
1/2	100	97.72	97.57	96.81	96.40	96.44	96.64
2/3	100	98.20	97.74	96.94	96.98	96.55	96.50
4/5	100	98.42	97.60	97.13	97.24	96.99	96.55
9/10	100	98.51	97.76	97.34	97.24	96.70	96.52
19/20	100	98.53	97.81	97.19	96.99	96.88	96.56

Tabela 29: Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - bootstrap - zbiór testowy - KNN

Podział zbioru	Liczba sąsiadów						
	1	2	3	4	5	10	20
1/20	85.07	75.45	71.35	62.43	54.66	33.44	34.67
1/10	90.76	87.97	87.91	84.65	85.31	63.11	34.19
1/5	92.51	91.46	92.67	92.11	92.99	91.02	69.84
1/3	93.66	92.51	93.45	93.84	94.01	94.47	92.07
1/2	94.43	93.56	94.18	93.86	94.12	95.12	95.26
2/3	94.60	93.44	94.13	94.62	94.31	95.02	95.72
4/5	94.53	94.25	94.30	94.85	94.60	94.91	95.94
9/10	94.86	94.47	94.52	94.48	94.33	95.24	95.84
19/20	94.61	94.45	94.22	94.58	94.57	95.05	95.85

Tabela 30: Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - bootstrap - zbiór treningowy - KNN

Podział zbioru	Liczba sąsiadów						
	1	2	3	4	5	10	20
1/20	97.33	79.55	73.70	62.58	53.93	25.35	24.64
1/10	99.67	94.40	93.27	89.92	89.60	61.61	22.74
1/5	100.00	96.35	97.28	95.91	95.91	93.10	67.99
1/3	100.00	97.69	97.61	96.77	96.42	95.86	93.93
1/2	100.00	97.75	97.57	96.87	96.40	96.50	96.71
2/3	100.00	98.21	97.76	96.99	97.03	96.55	96.56
4/5	100.00	98.45	97.63	97.18	97.28	97.04	96.62
9/10	100.00	98.54	97.79	97.39	97.27	96.78	96.59
19/20	100.00	98.56	97.85	97.23	97.03	96.94	96.64

Tabela 31: Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - bootstrap - zbiór testowy - KNN

Podział zbioru	Liczba sąsiadów						
	1	2	3	4	5	10	20
1/20	84.28	73.37	65.70	54.95	45.89	19.14	19.64
1/10	90.80	88.03	87.12	83.41	84.19	56.07	19.00
1/5	92.67	91.64	92.90	92.31	93.21	90.73	64.31
1/3	93.81	92.67	93.62	94.04	94.17	94.61	92.15
1/2	94.52	93.69	94.29	93.94	94.26	95.20	95.38
2/3	94.68	93.53	94.24	94.68	94.41	95.11	95.81
4/5	94.64	94.31	94.36	94.91	94.70	95.00	95.98
9/10	94.94	94.51	94.58	94.52	94.42	95.28	95.87

Tabela 31: Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby sąsiadów - bootstrap - zbiór testowy - KNN (kontynuacja)

Podział zbioru	Liczba sąsiadów						
	1	2	3	4	5	10	20
19/20	94.69	94.50	94.25	94.64	94.58	95.15	95.91

Tabele 28, 29, 30 oraz 31, które prezentują wyniki oceny modelu przy użyciu metody bootstrap, potwierdzają podobne wnioski jak te uzyskane metodą wielokrotnego podziału. Zarówno w przypadku bootstrapu, jak i wielokrotnego podziału danych, obserwujemy spójność wyników dotyczących wpływu liczby sąsiadów oraz proporcji podziału zbioru na skuteczność klasyfikacji. Metoda bootstrap, dzięki wielokrotnemu próbkowaniu z zastępowaniem, pozwala dodatkowo ocenić stabilność i wariancję estymatorów modelu, co wzmacnia wiarygodność uzyskanych rezultatów. W efekcie, wyniki obu metod wskazują na podobne zachowanie modelu, co zwiększa pewność co do jego jakości i zdolności do generalizacji.

Podsumowując:

- Dane charakteryzują się wystarczającą różnorodnością i dobrą separowalnością klas, co sprawia, że najlepsze wyniki osiąga model k-NN wykorzystujący jednego sąsiada.
- Nie zaobserwowano wyraźnych sygnałów świadczących o nadmiernym dopasowaniu modelu na zbiorze treningowym i kiepskim dopasowaniu na zbiorze testowym (overfittingu), co potwierdza stabilność i zdolność generalizacji klasyfikatora.

Autorzy rekomendują, aby przy zastosowaniu metody k-NN do zbioru danych *wine* po standaryzacji wykorzystać pełny zbiór treningowy oraz ustawić parametr liczby sąsiadów na wartość jeden, co zapewnia optymalną skuteczność modelu.

2.2.1.3 Najlepszy model na podstawie najlepszych cech dyskryminujących Pojawia się jednak istotny problem związany z czasem obliczeń oraz wymaganiami pamięciowymi podczas klasyfikacji nowych obserwacji, wynikający ze złożoności metody k-NN. W związku z tym, podobne analizy jak dla całego zbioru przeprowadzono również na podzbiorze najlepszych cech dyskryminujących, co pozwala na zmniejszenie kosztów obliczeniowych przy zachowaniu skuteczności modelu.

Do wybrania cech o najlepszej zdolności dyskrymijacyjnej posłuży metoda k-średnich.

W Tabeli 32. przedstawiono wyniki zdolności dyskriminacyjnych zmiennych.

Tabela 32: Skuteczność dyskryminacyjna zmiennych po dyskretyzacji metodą k-średnich

Zmienna	Dopasowanie (%)
Flawonoidy	79.78
Koloru	75.84
Prolina	70.22
Alkohol	68.54
Fenole_ogółem	62.36
OD280_OD315	62.36
Barwa	54.81
Kwas_mlekowy	49.81

Tabela 32: Skuteczność dyskryminacyjna zmiennych po dyskretyzacji metodą k-średnich (kontynuacja)

Zmienna	Dopasowanie (%)
Fenole_nieflawonoidowe	47.85
Zasadowość_popiołu	47.77
Proantocyjanidyny	45.91
Magnez	45.29
Popiół	33.33

W poprzedniej części wykazano, że model osiąga najlepsze wyniki przy zastosowaniu jednego sąsiada. Dlatego też parametr ten zostanie przyjęty jako stały. W dalszej analizie zbadano, jak zmienia się skuteczność klasyfikacji w zależności od liczby wykorzystywanych cech (rozpoczynając od tych najlepiej dyskryminujących) oraz od proporcji podziału danych na zbiór treningowy i testowy. Ze względu na fakt, że dla jednego sąsiada model może osiągać niemal idealne wyniki na zbiorze treningowym, w analizie pominięto ocenę jakości klasyfikacji na tym zbiorze.

Tabela 33: Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - wielokrotny podział - zbiór testowy - KNN

Podział zbioru	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1/20	67.41	80.81	85.78	88.26	84.34	85.20	87.18	85.88	83.60	84.38	85.15	81.52	84.11
1/10	71.39	87.17	91.87	92.43	91.77	92.13	92.55	91.98	91.42	91.23	91.48	90.96	90.61
1/5	73.51	89.90	93.69	94.22	93.78	93.77	94.01	94.10	93.26	93.40	93.36	93.22	93.10
1/3	72.82	90.87	93.80	95.08	94.37	94.86	94.97	94.89	94.75	94.24	94.47	94.42	94.03
1/2	72.37	90.89	94.49	95.08	94.53	94.97	95.91	95.76	95.54	95.01	95.33	95.12	94.17
2/3	70.80	92.17	95.03	95.27	94.30	95.38	96.12	96.10	96.50	94.88	95.60	95.80	95.03
4/5	69.64	91.92	95.72	94.69	94.61	95.00	95.75	96.69	96.83	95.75	96.00	96.22	94.69
9/10	71.39	92.00	96.39	95.39	93.94	95.50	95.89	97.22	96.33	96.61	96.44	96.22	95.50
19/20	68.22	92.00	97.22	94.89	95.56	95.56	96.11	96.89	96.00	96.89	96.67	95.44	96.56

W Tabeli 33. widać, że już przy 3–4 najbardziej dyskryminujących cechach dokładność klasyfikacji przekracza 90%, a dalsze zwiększanie liczby cech powyżej przynosi jedynie minimalne korzyści. Zwiększenie udziału danych treningowych do około 2/3–9/10 istotnie podnosi skuteczność, przy czym najlepszy wynik otrzymano dla 3 cech i udziału danych treningowych 19/20.

Tabela 34: Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - wielokrotny podział - zbiór testowy - KNN

Podział zbioru	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1/20	65.90	79.42	83.73	86.87	82.08	83.34	85.87	84.45	82.04	83.03	84.32	79.32	82.64
1/10	71.70	87.56	92.14	92.49	91.83	92.18	92.69	92.21	91.61	91.37	91.69	91.15	90.73

Tabela 34: Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - wielokrotny podział - zbiór testowy - KNN (kontynuacja)

Podział zbioru	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1/5	74.45	90.46	93.96	94.34	93.86	93.88	94.21	94.29	93.42	93.53	93.54	93.40	93.26
1/3	73.65	91.39	94.05	95.17	94.45	94.93	95.10	95.02	94.87	94.36	94.64	94.55	94.14
1/2	73.20	91.33	94.72	95.14	94.56	95.04	96.02	95.85	95.61	95.08	95.44	95.21	94.27
2/3	71.59	92.56	95.18	95.29	94.28	95.50	96.21	96.14	96.51	94.91	95.64	95.84	95.05
4/5	70.37	92.05	95.79	94.64	94.50	95.00	95.71	96.73	96.82	95.74	96.00	96.28	94.71
9/10	71.04	91.92	96.41	95.11	93.43	94.94	95.40	97.13	95.96	96.22	95.82	95.93	95.19
19/20	65.10	86.34	94.95	90.68	92.62	92.68	93.36	95.68	94.49	91.61	93.81	92.53	93.07

W Tabeli 34. można zaobserwować przeuczenie modelu. Najlepsze wyniki osiągnięto przy udziale danych treningowych w przedziale od 2/3 do 9/10 oraz przy doborze 7–10 cech, przy czym najwyższą dokładność odnotowano dla 8 cech i podziału danych na 90% treningowych i 10% testowych.

Tabela 35: Średnia dokładność klasyfikacji dla kombinacji parametrów liczby podzbiorów oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - cross-validation - zbiór testowy - KNN

Liczba podzbiorów	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
2	69.10	92.70	96.63	96.63	97.19	95.51	96.63	95.51	94.38	93.26	93.82	95.51	94.38
5	70.27	92.68	95.51	94.38	93.25	94.89	96.05	96.06	96.60	94.37	96.03	96.03	95.00
10	70.92	91.57	95.52	95.00	93.27	95.00	96.63	96.05	96.60	94.97	96.67	95.49	95.49
20	69.65	92.71	97.22	95.49	93.68	95.49	96.04	97.22	96.04	95.56	95.56	96.11	96.04
50	68.67	92.50	96.50	95.00	94.67	95.33	96.00	97.33	96.50	96.17	95.83	96.17	95.83
178	69.10	92.70	96.63	94.94	94.38	95.51	96.07	97.19	96.63	96.07	96.07	96.07	95.51

W Tabeli 35. widać, że przy zastosowaniu walidacji krzyżowej już od 3–4 najlepiej dyskryminujących cech dokładność klasyfikacji przekracza 95%, a jej wartość jest relatywnie stabilna niezależnie od liczby podziałów. Najwyższą średnią dokładność uzyskano dla 8 cech przy 50-krotnej walidacji. Optymalny zakres konfiguracji to 7–9 cech i co najmniej 20–50 podziałów cross-validation.

Tabela 36: Średni F1-score dla kombinacji parametrów liczby podzbiorów oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - cross-validation - zbiór testowy - KNN

Liczba podzbiorów	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
2	69.68	93.17	96.71	96.74	97.18	95.55	96.48	95.54	94.53	93.16	93.94	95.84	94.55
5	70.52	92.61	95.82	94.63	92.86	94.83	95.85	95.95	96.71	94.41	96.14	95.97	95.27
10	71.08	90.86	94.57	94.14	90.22	92.45	95.03	96.25	95.42	93.89	96.90	95.64	95.48

Tabela 36: Średni F1-score dla kombinacji parametrów liczby podzbiorów oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - cross-validation - zbiór testowy - KNN (kontynuacja)

Liczba podzbiorów	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
20	63.82	87.93	94.23	89.08	88.64	92.79	93.00	95.23	92.61	91.76	88.35	90.97	96.04
50	52.69	70.13	71.90	69.73	69.04	68.49	76.20	76.08	74.62	73.91	73.67	74.68	72.99
178	23.03	30.90	32.21	31.65	31.46	31.84	32.02	32.40	32.21	32.02	32.02	32.02	31.84

W Tabeli 36. widać, że najwyższy średni F1-score uzyskano dla 2-krotnej walidacji krzyżowej z wykorzystaniem pięciu najlepiej dyskryminujących zmiennych. Zauważalny spadek miary przy rosnącej liczbie podziałów (50-, 178-fold) wskazuje, że optymalną konfiguracją jest stosowanie umiarkowanej liczby podziałów (2–10) oraz selekcja od trzech do pięciu najsilniej dyskryminujących cech.

Tabela 37: Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - bootstrap - zbiór testowy - KNN

Podział zbioru	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1/20	66.99	79.64	85.04	86.17	84.63	86.75	85.67	85.59	84.44	81.66	82.61	84.43	83.67
1/10	71.89	86.52	91.02	92.26	91.43	92.75	92.05	91.60	90.14	90.97	90.08	90.45	89.38
1/5	72.41	89.17	93.32	94.28	93.18	93.67	93.44	93.46	93.44	92.07	92.42	92.99	92.88
1/3	72.54	90.35	93.90	94.94	94.46	94.40	94.75	94.51	94.31	94.00	93.91	94.31	93.35
1/2	72.15	91.12	94.04	94.93	94.47	95.16	95.61	95.33	95.12	94.28	94.96	94.42	94.20
2/3	72.43	91.38	94.71	94.85	94.52	95.19	95.66	95.43	95.94	94.98	95.10	95.45	94.82
4/5	71.06	91.77	94.77	94.84	94.58	95.48	95.66	96.47	95.93	94.93	95.61	95.43	94.50
9/10	72.00	91.79	95.12	94.97	94.55	95.75	96.03	96.44	96.21	95.48	95.33	95.76	95.01
19/20	71.61	92.10	94.76	95.11	94.58	94.70	96.02	96.38	96.25	95.03	95.71	95.71	95.04

W Tabeli 37. widać, że już przy trzech najlepiej dyskryminujących cechach dokładność przekracza 85%, a po dodaniu czwartej–piątej cechy zbliża się do 94%. Najwyższe wartości uzyskano dla podziałów bootstrapowych z udziałem 4/5–9/10 danych treningowych i 8–9 cechami. Dalsze zwiększanie liczby cech powyżej 9 oraz udziału danych treningowych ponad 90% przynosi jedynie marginalne korzyści. Optymalną konfiguracją jest więc wykorzystanie około 8–9 kluczowych zmiennych przy zastosowaniu bootstrapu z próbkowaniem rzędu 9/10 na dane treningowe.

Tabela 38: Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - bootstrap - zbiór testowy - KNN

Podział zbioru	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1/20	65.14	78.49	83.31	84.77	82.34	85.52	83.68	84.03	83.18	79.48	80.85	83.27	81.79

Tabela 38: Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - bootstrap - zbiór testowy - KNN (kontynuacja)

Podział zbioru	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1/10	72.27	86.75	91.22	92.21	91.49	92.87	92.20	91.49	90.15	91.21	90.09	90.47	89.46
1/5	73.26	89.70	93.58	94.37	93.29	93.76	93.59	93.57	93.61	92.23	92.62	93.17	93.04
1/3	73.46	90.86	94.15	95.01	94.52	94.50	94.94	94.66	94.43	94.14	94.08	94.42	93.48
1/2	73.10	91.57	94.31	95.00	94.55	95.23	95.74	95.44	95.22	94.37	95.09	94.58	94.31
2/3	73.27	91.81	94.91	94.94	94.56	95.26	95.77	95.53	95.99	95.08	95.22	95.53	94.89
4/5	71.91	92.17	94.94	94.91	94.58	95.53	95.75	96.51	95.96	94.98	95.67	95.52	94.59
9/10	72.85	92.15	95.31	95.06	94.58	95.81	96.11	96.53	96.22	95.55	95.42	95.82	95.07
19/20	72.29	92.47	94.97	95.17	94.63	94.74	96.08	96.46	96.25	95.06	95.81	95.79	95.11

W Tabeli 38. widać, że już przy trzech najlepiej dyskryminujących cechach średni F1-score przekracza 83 %, a przy czterech–pięciu cechach zbliża się do 94–95 %. Najwyższe wartości osiągnięto dla udziału danych treningowych w przedziale 4/5–9/10 oraz przy wykorzystaniu 8 cech, co wskazuje na optymalność tej konfiguracji. Dalsze zwiększanie liczby cech powyżej 9 czy udziału bootstrapowego powyżej 90 % nie przynosi istotnych korzyści.

Podsumowując:

- Dla tradycyjnego podziału optymalny udział danych treningowych wynosi około 80–90% (podziały 4/5–9/10), przy czym najlepsze rezultaty osiąga się, wykorzystując 8 najlepiej dyskryminujących zmiennych.
- W przypadku walidacji krzyżowej (cross-validation) wystarczy stosować umiarkowaną liczbę podziałów (2–5 foldów) oraz wybór około 5 najsilniej dyskryminujących cech.

Autorzy stwierdzili, że optymalny model uzyskuje się, stosując klasyfikator k-NN z jednym sąsiadem, wykorzystując maksymalny możliwy udział danych w zbiorze treningowym oraz **osiem** najlepiej dyskryminujących cech, co zapewnia odpowiedni kompromis między rozmiarem modelu (przechowującego wszystkie dane) a jego zdolnością klasyfikacyjną.

Tabela 39: Macierz pomyłek — zbiór treningowy — KNN

Gatunek	1	2	3
1	55	0	0
2	0	61	0
3	0	0	44

W Tabeli 39. przedstawiono macierz pomyłek dla wyżej przedstawionego modelu przy udziale danych treningowych 9/10.

Tabela 40: Metryki klasyfikacji dla każdego gatunku - zbiór treningowy - KNN

Prawdziwy gatunek	Dokładność	Precyzja	Czułośc	F1-score
1	1	1	1	1

Tabela 40: Metryki klasyfikacji dla każdego gatunku - zbiór treningowy - KNN (kontynuacja)

Prawdziwy gatunek	Dokładność	Precyzja	Czułośc	F1-score
2	1	1	1	1
3	1	1	1	1
Średnie	1	1	1	1

Średnia dokładność dla każdego gatunku wynosi 100%, a F1-score 100%. Są to znakomite wyniki, świadczące o bardzo wysokiej skuteczności klasyfikatora w rozróżnianiu poszczególnych klas.

Tabela 41: Macierz pomyłek — zbiór testowy — KNN

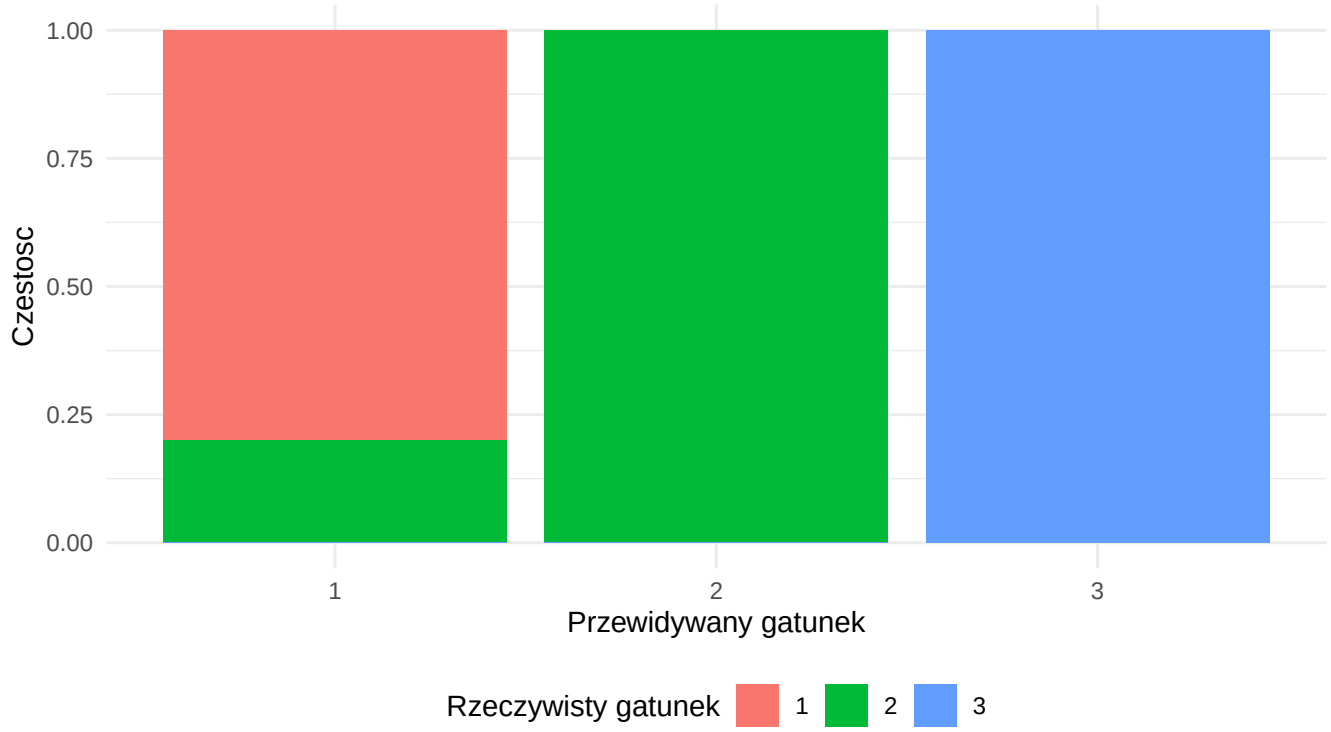
Gatunek	1	2	3
1	4	0	0
2	1	9	0
3	0	0	4

W Tabeli 31. przedstawiono macierz pomyłek modelu dla zbioru testowego.

Tabela 42: Metryki klasyfikacji dla każdego gatunku - zbiór testowy - KNN

Prawdziwy gatunek	Dokładność	Precyzja	Czułośc	F1-score
1	0.94	0.80	1.00	0.89
2	0.94	1.00	0.90	0.95
3	1.00	1.00	1.00	1.00
Średnie	0.96	0.93	0.97	0.95

Średnia dokładność dla każdego gatunku wynosi 96%, a F1-score 95%. Są to znakomite wyniki, świadczące o bardzo wysokiej skuteczności klasyfikatora w rozróżnianiu poszczególnych klas. Są to fenomenalne wyniki.



Wykres 9: Porównanie przewidywanych i rzeczywistych klas - KNN

Na Wykresie 9. graficznie widać jak dobrze model sobie radzi.

Trzeba jednak pamiętać, że model zajmuje 13.6 Kb, czyli tyle co zbiór treningowy.

2.2.2 Drzewa klasyfikacyjne

W tej części zajmiemy się porównaniem wyników uzyskiwanych poprzez nieobcięte drzewa, oraz ich optymalnie przycięte odpowiedniki. Następnie najlepsze drzewo zostanie zbadane pod kątem ilości najlepszych cech dyskryminujących, które pozwolą na maksymalizację dokładności oraz wskaźnika F1.

2.2.2.1 Obcięte VS nieobcięte drzewo W celu ustalenia, czy warto obcinać drzewo klasyfikacyjne, aby zwiększyć optymalizację kosztem dokładności, porównamy wyniki dla obu rodzajów. Dane podzielono na zbiór treningowy i testowy w skali 2:1, a w przypadku metody cross-validation zbiór podzielono na 10 części. Ustalono również dwa wymienione parametry:

- $cp = 0.01$ - parametr złożoności, który określa minimalną poprawę (zmniejszenie błędu) wymaganą, aby wykonać podział w drzewie,
- $minsplit = 5$ - minimalna liczba obserwacji, które muszą znajdować się w węźle, aby możliwe było jego dalsze dzielenie.

```
wielokrotny_podział <- function(data = wine, K = 100, podzial = 2/3,
                                cp = 0.01, minsplit = 5) {
  # Wielokrotny podział
  n <- nrow(data)
```

```r
sum_acc_train <- 0
sum_acc_test <- 0
sum_metrics_train <- NULL
sum_metrics_test <- NULL

sum_acc_train.pruned <- 0
sum_acc_test.pruned <- 0
sum_metrics_train.pruned <- NULL
sum_metrics_test.pruned <- NULL

for (i in seq_len(K)) {
  train_idx <- sample(seq_len(n), size = floor(podzial * n))
  train.set <- data[train_idx, ]
  test.set <- data[-unique(train_idx), ]

  wine.tree <- rpart(Gatunek ~ .,
                     data = train.set,
                     method = "class",
                     control = rpart.control(cp = cp, minsplit = minsplit))

  #Przycinanie drzewa
  frame <- printcp(wine.tree)
  SE <- frame[nrow(frame), 4] + frame[nrow(frame), 5]

  index <- min(which(frame[, 4] < SE))
  if(index == Inf) index <- nrow(frame)

  cp.optimal <- frame[index, 1]
  wine.tree.pruned <- prune(wine.tree, cp = cp.optimal)



  #Nieprzycięte drzewo
  pred_train <- predict(wine.tree, newdata = train.set, type = "class")
  cm_train <- table(Prawdziwa = train.set$Gatunek, Przewidziana = pred_train)
  acc_train <- sum(diag(cm_train)) / sum(cm_train)
  metrics_train <- metryki(cm_train)

  pred_test <- predict(wine.tree, newdata = test.set, type = "class")
  cm_test <- table(Prawdziwa = test.set$Gatunek, Przewidziana = pred_test)
  acc_test <- sum(diag(cm_test)) / sum(cm_test)
  metrics_test <- metryki(cm_test)

  sum_acc_train <- sum_acc_train + acc_train
  sum_acc_test <- sum_acc_test + acc_test

  if (is.null(sum_metrics_train)) {
    sum_metrics_train <- metrics_train
    sum_metrics_test <- metrics_test
  } else {
    sum_metrics_train <- sum_metrics_train + metrics_train
    sum_metrics_test <- sum_metrics_test + metrics_test
  }
```

```
    #Przycięte drzewo
    pred_train <- predict(wine.tree.pruned, newdata = train.set, type = "class")
    cm_train <- table(Prawdziwa = train.set$Gatunek,Przewidziana = pred_train)
    acc_train <- sum(diag(cm_train)) / sum(cm_train)
    metrics_train <- metryki(cm_train)

    pred_test <- predict(wine.tree.pruned,newdata = test.set, type = "class")
    cm_test <- table(Prawdziwa = test.set$Gatunek,Przewidziana = pred_test)
    acc_test <- sum(diag(cm_test)) / sum(cm_test)
    metrics_test <- metryki(cm_test)

    sum_acc_train.pruned <- sum_acc_train.pruned + acc_train
    sum_acc_test.pruned <- sum_acc_test.pruned + acc_test

    if (is.null(sum_metrics_train.pruned)) {
      sum_metrics_train.pruned <- metrics_train
      sum_metrics_test.pruned <- metrics_test
    } else {
      sum_metrics_train.pruned <- sum_metrics_train.pruned + metrics_train
      sum_metrics_test.pruned <- sum_metrics_test.pruned + metrics_test
    }
  }

  avg_acc_train <- sum_acc_train / K
  avg_acc_test <- sum_acc_test / K
  avg_metrics_train <- sum_metrics_train / K
  avg_metrics_test <- sum_metrics_test / K

  avg_acc_train.pruned <- sum_acc_train.pruned / K
  avg_acc_test.pruned <- sum_acc_test.pruned / K
  avg_metrics_train.pruned <- sum_metrics_train.pruned / K
  avg_metrics_test.pruned <- sum_metrics_test.pruned / K

  return(list(
    list(accuracy_train = avg_acc_train,
         accuracy_test  = avg_acc_test,
         metrics_train  = avg_metrics_train,
         metrics_test   = avg_metrics_test),
    list(accuracy_train.pruned = avg_acc_train.pruned,
         accuracy_test.pruned  = avg_acc_test.pruned,
         metrics_train.pruned  = avg_metrics_train.pruned,
         metrics_test.pruned   = avg_metrics_test.pruned)
  ))
}
```

```
cross_validation <- function(data = wine, K = 10,
                             cp = 0.01, minsplit = 5) {
  # Cross-validation
  n <- nrow(data)
  sum_acc_train <- 0
  sum_acc_test  <- 0
```

```r
sum_metrics_train <- NULL
sum_metrics_test  <- NULL

sum_acc_train.pruned <- 0
sum_acc_test.pruned <- 0
sum_metrics_train.pruned <- NULL
sum_metrics_test.pruned <- NULL

folds <- sample(rep(seq_len(K), length.out = n))

for (i in seq_len(K)) {
  train.set <- data[folds != i, ]
  test.set  <- data[folds == i, ]

  wine.tree <- rpart(Gatunek ~ .,
                     data = train.set,
                     method = "class",
                     control = rpart.control(cp = cp, minsplit = minsplit))

  frame <- printcp(wine.tree)
  SE <- frame[nrow(frame), 4] + frame[nrow(frame), 5]

  index <- min(which(frame[, 4] < SE))
  if(index == Inf) index <- nrow(frame)

  cp.optimal <- frame[index, 1]
  wine.tree.pruned <- prune(wine.tree, cp = cp.optimal)

  #Nieprzycięte drzewo
  pred_train <- predict(wine.tree, newdata = train.set, type = "class")
  cm_train <- table(Prawdziwa = train.set$Gatunek, Przewidziana = pred_train)
  acc_train <- sum(diag(cm_train)) / sum(cm_train)
  metrics_train <- metryki(cm_train)

  pred_test <- predict(wine.tree, newdata = test.set, type = "class")
  cm_test <- table(Prawdziwa = test.set$Gatunek, Przewidziana = pred_test)
  acc_test <- sum(diag(cm_test)) / sum(cm_test)
  metrics_test <- metryki(cm_test)

  sum_acc_train <- sum_acc_train + acc_train
  sum_acc_test <- sum_acc_test + acc_test

  if (is.null(sum_metrics_train)) {
    sum_metrics_train <- metrics_train
    sum_metrics_test <- metrics_test
  } else {
    sum_metrics_train <- sum_metrics_train + metrics_train
    sum_metrics_test <- sum_metrics_test + metrics_test
  }



  #Przycięte drzewo
```

```
    pred_train <- predict(wine.tree.pruned, newdata = train.set, type = "class")
    cm_train <- table(Prawdziwa = train.set$Gatunek,Przewidziana = pred_train)
    acc_train <- sum(diag(cm_train)) / sum(cm_train)
    metrics_train <- metryki(cm_train)

    pred_test <- predict(wine.tree.pruned,newdata = test.set, type = "class")
    cm_test <- table(Prawdziwa = test.set$Gatunek,Przewidziana = pred_test)
    acc_test <- sum(diag(cm_test)) / sum(cm_test)
    metrics_test <- metryki(cm_test)

    sum_acc_train.pruned <- sum_acc_train.pruned + acc_train
    sum_acc_test.pruned <- sum_acc_test.pruned + acc_test

    if (is.null(sum_metrics_train.pruned)) {
      sum_metrics_train.pruned <- metrics_train
      sum_metrics_test.pruned <- metrics_test
    } else {
      sum_metrics_train.pruned <- sum_metrics_train.pruned + metrics_train
      sum_metrics_test.pruned <- sum_metrics_test.pruned + metrics_test
    }
  }

  avg_acc_train <- sum_acc_train / K
  avg_acc_test <- sum_acc_test / K
  avg_metrics_train <- sum_metrics_train / K
  avg_metrics_test <- sum_metrics_test / K

  avg_acc_train.pruned <- sum_acc_train.pruned / K
  avg_acc_test.pruned <- sum_acc_test.pruned / K
  avg_metrics_train.pruned <- sum_metrics_train.pruned / K
  avg_metrics_test.pruned <- sum_metrics_test.pruned / K

  return(list(
    list(accuracy_train = avg_acc_train,
         accuracy_test  = avg_acc_test,
         metrics_train  = avg_metrics_train,
         metrics_test   = avg_metrics_test),
    list(accuracy_train.pruned = avg_acc_train.pruned,
         accuracy_test.pruned  = avg_acc_test.pruned,
         metrics_train.pruned  = avg_metrics_train.pruned,
         metrics_test.pruned   = avg_metrics_test.pruned)
  ))
}
```

```
bootstrap <- function(data = wine, K = 100, podzial = 2/3,
                      cp = 0.01, minsplit = 5) {
  # Bootstrap
  n <- nrow(data)

  sum_acc_train <- 0
  sum_acc_test <- 0
  sum_metrics_train <- NULL
  sum_metrics_test <- NULL
```

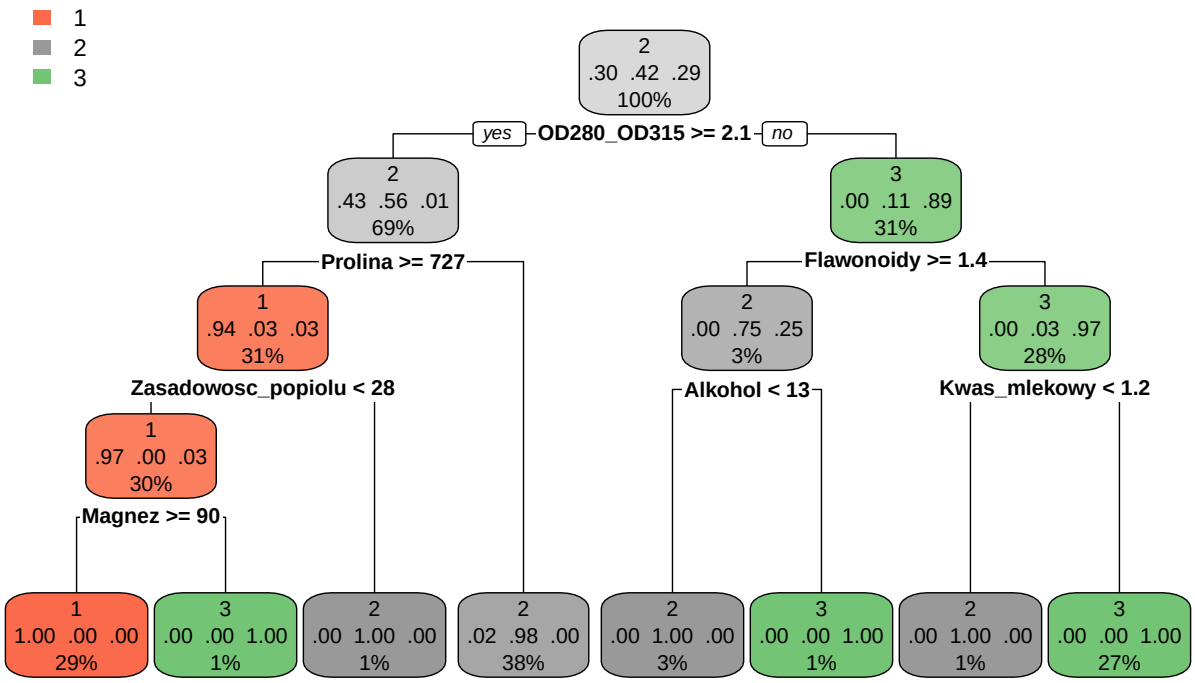
```r
sum_acc_train.pruned <- 0
sum_acc_test.pruned <- 0
sum_metrics_train.pruned <- NULL
sum_metrics_test.pruned <- NULL

for (i in seq_len(K)) {
  train_idx <- sample(seq_len(n), size = floor(podzial * n), replace = TRUE)
  train.set <- data[train_idx, ]
  test.set <- data[-unique(train_idx), ]

  wine.tree <- rpart(Gatunek ~ .,
                     data = train.set,
                     method = "class",
                     control = rpart.control(cp = cp, minsplit = minsplit))

  frame <- printcp(wine.tree)
  SE <- frame[nrow(frame), 4] + frame[nrow(frame), 5]

  index <- min(which(frame[, 4] < SE))
  if(index == Inf) index <- nrow(frame)

  cp.optimal <- frame[index, 1]
  wine.tree.pruned <- prune(wine.tree, cp = cp.optimal)

  #Nieprzycięte drzewo
  pred_train <- predict(wine.tree, newdata = train.set, type = "class")
  cm_train <- table(Prawdziwa = train.set$Gatunek, Przewidziana = pred_train)
  acc_train <- sum(diag(cm_train)) / sum(cm_train)
  metrics_train <- metryki(cm_train)

  pred_test <- predict(wine.tree, newdata = test.set, type = "class")
  cm_test <- table(Prawdziwa = test.set$Gatunek, Przewidziana = pred_test)
  acc_test <- sum(diag(cm_test)) / sum(cm_test)
  metrics_test <- metryki(cm_test)

  sum_acc_train <- sum_acc_train + acc_train
  sum_acc_test <- sum_acc_test + acc_test

  if (is.null(sum_metrics_train)) {
    sum_metrics_train <- metrics_train
    sum_metrics_test <- metrics_test
  } else {
    sum_metrics_train <- sum_metrics_train + metrics_train
    sum_metrics_test <- sum_metrics_test + metrics_test
  }



  #Przycięte drzewo
  pred_train <- predict(wine.tree.pruned, newdata = train.set, type = "class")
  cm_train <- table(Prawdziwa = train.set$Gatunek,Przewidziana = pred_train)
  acc_train <- sum(diag(cm_train)) / sum(cm_train)
  metrics_train <- metryki(cm_train)
```

```
    pred_test <- predict(wine.tree.pruned,newdata = test.set, type = "class")
    cm_test <- table(Prawdziwa = test.set$Gatunek,Przewidziana = pred_test)
    acc_test <- sum(diag(cm_test)) / sum(cm_test)
    metrics_test <- metryki(cm_test)

    sum_acc_train.pruned <- sum_acc_train.pruned + acc_train
    sum_acc_test.pruned <- sum_acc_test.pruned + acc_test

    if (is.null(sum_metrics_train.pruned)) {
      sum_metrics_train.pruned <- metrics_train
      sum_metrics_test.pruned <- metrics_test
    } else {
      sum_metrics_train.pruned <- sum_metrics_train.pruned + metrics_train
      sum_metrics_test.pruned <- sum_metrics_test.pruned + metrics_test
    }
  }

  avg_acc_train <- sum_acc_train / K
  avg_acc_test <- sum_acc_test / K
  avg_metrics_train <- sum_metrics_train / K
  avg_metrics_test <- sum_metrics_test / K

  avg_acc_train.pruned <- sum_acc_train.pruned / K
  avg_acc_test.pruned <- sum_acc_test.pruned / K
  avg_metrics_train.pruned <- sum_metrics_train.pruned / K
  avg_metrics_test.pruned <- sum_metrics_test.pruned / K

  return(list(
    list(accuracy_train = avg_acc_train,
         accuracy_test  = avg_acc_test,
         metrics_train  = avg_metrics_train,
         metrics_test   = avg_metrics_test),
    list(accuracy_train.pruned = avg_acc_train.pruned,
         accuracy_test.pruned  = avg_acc_test.pruned,
         metrics_train.pruned  = avg_metrics_train.pruned,
         metrics_test.pruned   = avg_metrics_test.pruned)
  ))
}
```

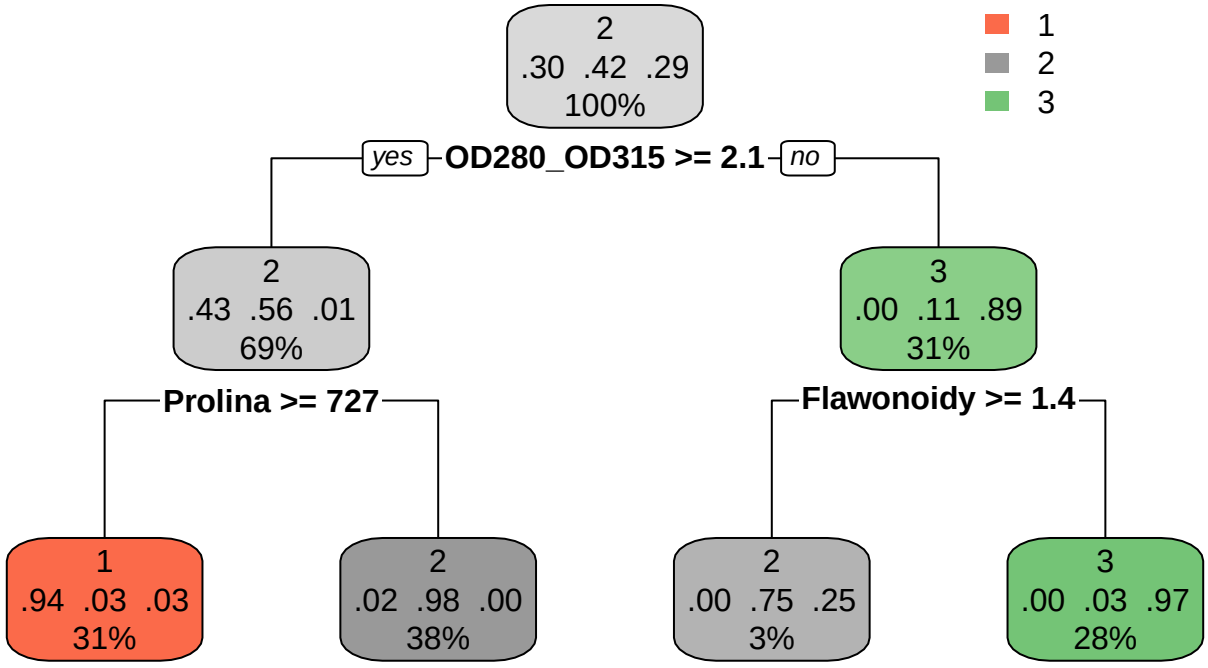
W Tabeli 43. przedstawiono średnie wartości dokładności oraz wskaźnika F1, dla drzewa odpowiednio przyciętego lub nieprzyciętego. Jak widać, otrzymane wyniki różnią się w zależności od metody, jednak średnio odpowiednie parametry różnią się o mniej niż 2%, z korzyścią dla bardziej szczegółowego drzewka.

Tabela 43: Średnie dokładność klasyfikacji oraz F1-score dla drzew przyciętych oraz nieprzyciętych, z podziałem na metody oceny

	Wielokrotny podział	Cross-validation	Bootstrap	Średnio
Dokładność	90.88	91.63	89.49	90.67
Dokładność (pruned)	88.72	89.97	87.86	88.85
F1 score	90.88	92.38	89.51	90.92
F1 score (pruned)	88.76	90.55	87.82	89.04



Wykres 10: Nieprzycięte drzewo klasyfikacyjne



Wykres 11: Przycięte drzewo klasyfikacyjne

Na Wykresach 10. oraz 11. znajdują się odpowiednio nieprzycięte oraz przycięte drzewo decyzyjne. Zauważyć można, że uciętych zostało łącznie aż 8 gałęzi. To kosztem mniejszej o ok. 2% dokładności zapewniło nam nie tylko lepszą czytelność drzewa, ale przede wszystkim mniej zużytej pamięci oraz mniejszą złożoność obliczeniową, a co za tym idzie szybszą klasyfikację przyszłych zbiorów danych.

2.2.2.2 Najlepszy model W tej części zostanie dokonana ocena klasyfikacji modelu dla różnych proporcji podziału zbioru danych oraz różnych wartości liczby sąsiadów. Podobnie jak przy k-NN, wybrano odpowiednie udziały zbioru uczącego: 1/20, 1/10, 1/5, 1/3, 1/2, 2/3, 4/5, 9/10 oraz 19/20, dla metody wielokrotnego podziału oraz bootstrap. Dla metody cross-validation ponownie zbiór podzielono na 2, 5, 10, 20, 50 i 178 (Leave-One-Out) części.

Jako badane parametry wybrano kombinacje *cp*: 0.1, 0.005, oraz *minsplit*: 3, 7, 12. Dodatkowo, w ramach badania overfittingu, dobrano dodatkową parę parametrów(0.001, 1) (w tym przypadku model stworzy oddzielny liść dla każdego przypadku z modelu uczącego).

NIEPRZYCIĘTE DRZEWO

Tabela 44: Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentów cp i minsplit - zbiór treningowy - nieprzycięte drzewo klasyfikacyjne

Podział zbioru	Kombinacje cp i minsplit						
	cp=0.1 minsplit=3	cp=0.1 minsplit=7	cp=0.1 minsplit=12	cp=0.005 minsplit=3	cp=0.005 minsplit=7	cp=0.005 minsplit=12	cp=0.001 minsplit=1
1/20	99.75	82.12	50.88	99.12	82.62	51.88	100
1/10	99.65	98.88	80.65	100.00	98.41	82.00	100
1/5	97.20	97.17	96.29	99.77	98.00	96.66	100
1/3	94.25	94.54	94.92	99.85	98.39	96.20	100
1/2	93.36	93.73	92.97	99.82	98.06	96.46	100
2/3	92.31	92.21	92.78	99.88	97.84	96.68	100
4/5	91.58	91.60	91.42	99.99	98.04	96.41	100
9/10	91.11	91.01	90.38	99.96	98.08	95.91	100
19/20	89.81	90.06	90.04	99.94	97.87	95.63	100

Tabela 44. pokazuje nam, że dla małych zbiorów uczących (stosunek niższy lub równy 1/5) daje bardzo podobne wyniki. Dla reszty widzimy już spore różnice dla odpowiednich *cp* - mniejsze daje lepszą dokładność, rosnącą wraz ze stosunkiem., druga daje natomiast malejącą. Dla drugiej również *minsplit* daje większe zróżnicowanie między poszczególnymi wynikami. Naturalnie, w ostatniej kolumnie mamy po 1 elemencie w liściu, zatem dokładność wynosi 100%.

Tabela 45: Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentów cp i minsplit - zbiór testowy - nieprzycięte drzewo klasyfikacyjne

Podział zbioru	Kombinacje cp i minsplit						
	cp=0.1 minsplit=3	cp=0.1 minsplit=7	cp=0.1 minsplit=12	cp=0.005 minsplit=3	cp=0.005 minsplit=7	cp=0.005 minsplit=12	cp=0.001 minsplit=1
1/20	64.12	54.39	34.64	65.01	55.47	35.22	64.32
1/10	77.20	76.00	60.91	76.53	76.01	62.11	76.48
1/5	83.10	83.55	82.73	84.64	83.76	82.82	83.67
1/3	84.85	85.12	84.63	87.07	86.79	85.50	87.57
1/2	85.39	85.31	84.99	89.49	88.03	87.39	89.78
2/3	84.78	84.90	84.53	90.68	90.50	88.25	90.37
4/5	84.33	83.86	83.28	90.17	90.94	89.31	91.36
9/10	83.22	82.83	83.17	90.22	90.72	89.22	91.33
19/20	83.33	82.11	79.44	88.89	88.33	88.22	87.44

Tabela 45. zachowuje trendy wskazane wcześniej. NMależy jednak zauważyć, że najlepsze wyniki osiąga stosunek 9/10 dla $cp = 0.005$, $minsplit = 7$. Ostatnia kolumna oraz występujące w niej różnice sugerują wystąpienie zjawiska overfittingu (dlatego też trudna uznać wyniki ponad 91% jako lepsze niż te wskazane wcześniej).

Tabela 46: Średni F1-score klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentów cp i minsplit - zbiór treningowy - nieprzycięte drzewo klasyfikacyjne

Podział zbioru	Kombinacje cp i minsplit						
	cp=0.1 minsplit=3	cp=0.1 minsplit=7	cp=0.1 minsplit=12	cp=0.005 minsplit=3	cp=0.005 minsplit=7	cp=0.005 minsplit=12	cp=0.001 minsplit=1
1/20	94.78	59.48	22.28	92.56	59.39	22.54	95.67
1/10	99.61	97.88	62.61	100.00	95.88	64.17	100.00
1/5	97.15	96.98	96.06	99.79	97.92	96.39	100.00
1/3	94.03	94.47	94.86	99.85	98.34	96.16	100.00
1/2	93.22	93.59	92.79	99.82	98.04	96.40	100.00
2/3	92.28	92.10	92.71	99.88	97.85	96.69	100.00
4/5	91.54	91.56	91.36	99.99	98.07	96.44	100.00
9/10	91.00	90.90	90.30	99.97	98.11	95.98	100.00
19/20	89.62	89.91	89.91	99.95	97.90	95.70	100.00

W Tabeli 46. widzimy od stosunku 1/5 niewielki wpływ parametru *minsplit* na wartości wskaźnika F1. Lepsze wyniki osiąga mniejsze *cp*, z czego najlepsze drugiego argumentu również najniższego. Większy wpływ *minsplitu* widać dla mniejszych stosunków, gdzie różnice wynoszą ponad 30%. Ponownie, ostatnia kolumna zawiera idealne wręcz wyniki.

Tabela 47: Średni F1-score klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentów cp i minsplit - zbiór testowy - nieprzycięte drzewo klasyfikacyjne

Podział zbioru	Kombinacje cp i minsplit						
	cp=0.1 minsplit=3	cp=0.1 minsplit=7	cp=0.1 minsplit=12	cp=0.005 minsplit=3	cp=0.005 minsplit=7	cp=0.005 minsplit=12	cp=0.001 minsplit=1
1/20	59.70	43.28	17.09	61.04	44.35	17.31	60.77
1/10	76.36	74.69	50.81	75.80	74.69	52.50	75.61
1/5	82.87	83.28	82.60	84.50	83.48	82.61	83.42
1/3	84.61	85.02	84.63	87.00	86.71	85.45	87.57
1/2	85.30	85.14	84.80	89.41	88.02	87.32	89.77
2/3	84.49	84.77	84.31	90.73	90.48	88.29	90.44
4/5	83.86	83.39	82.79	90.09	90.95	89.17	91.26
9/10	81.87	81.82	82.23	89.59	90.28	88.47	90.83
19/20	79.30	78.59	74.18	86.34	84.78	83.99	83.95

W Tabeli 47. widzimy podobny trend do poprzedniej tabeli: wraz ze wzrostem stosunku zbioru uczącego, zmniejsza sie wpływ *misplit* na wyniki, zwiększa się natomiast dla *cp*. Lepsze wyniki na ogół osiągają niższe wartości argumentów, a ostatnia kolumna ponownie nasuwa na myśl wystąpienie overfittingu.

PRZYCIĘTE DRZEWO

Tabela 48: Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentów cp i minsplit - zbiór treningowy - przycięte drzewo klasyfikacyjne

Podział zbioru	Kombinacje cp i minsplit						
	cp=0.1 minsplit=3	cp=0.1 minsplit=7	cp=0.1 minsplit=12	cp=0.005 minsplit=3	cp=0.005 minsplit=7	cp=0.005 minsplit=12	cp=0.001 minsplit=1
1/20	80.50	71.25	52.12	82.12	70.12	51.25	80.62
1/10	90.06	91.00	67.53	91.24	89.47	71.24	91.06
1/5	94.94	94.63	92.80	95.54	94.69	94.71	95.46
1/3	94.05	93.88	94.51	95.68	95.10	94.51	96.14
1/2	92.62	93.17	92.91	96.11	95.45	94.63	96.78
2/3	92.23	91.81	91.75	96.83	95.89	95.26	96.65
4/5	91.25	91.31	90.97	96.92	96.04	95.18	96.94
9/10	90.39	90.54	90.45	96.69	95.89	95.18	96.35
19/20	89.82	89.54	90.01	95.55	94.75	94.14	95.67

Tabela 49: Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentów cp i minsplit - zbiór testowy - przycięte drzewo klasyfikacyjne

Podział zbioru	Kombinacje cp i minsplit						
	cp=0.1 minsplit=3	cp=0.1 minsplit=7	cp=0.1 minsplit=12	cp=0.005 minsplit=3	cp=0.005 minsplit=7	cp=0.005 minsplit=12	cp=0.001 minsplit=1
1/20	51.97	45.09	35.21	51.99	46.08	34.31	51.61
1/10	68.32	69.55	49.32	69.57	67.42	52.07	67.76
1/5	80.73	79.86	79.71	82.56	82.27	81.19	82.13
1/3	83.48	84.20	84.54	85.04	84.32	84.39	85.03
1/2	84.10	84.36	84.98	86.76	86.27	85.39	87.15
2/3	84.15	84.47	83.87	88.58	87.57	86.93	88.67
4/5	83.22	83.61	84.47	89.08	87.78	87.92	89.14
9/10	81.33	82.22	83.17	88.00	89.67	87.72	88.28
19/20	81.00	78.11	81.89	87.00	88.00	87.00	87.44

Tabela 50: Średni F1-score klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentów cp i minsplit - zbiór treningowy - przycięte drzewo klasyfikacyjne

Podział zbioru	Kombinacje cp i minsplit						
	cp=0.1 minsplit=3	cp=0.1 minsplit=7	cp=0.1 minsplit=12	cp=0.005 minsplit=3	cp=0.005 minsplit=7	cp=0.005 minsplit=12	cp=0.001 minsplit=1
1/20	60.07	43.02	22.63	60.61	42.73	22.39	57.66
1/10	80.32	80.18	44.89	81.81	77.84	49.57	81.57
1/5	92.57	91.59	88.71	93.89	92.49	92.47	93.40
1/3	93.42	93.05	94.28	95.61	94.86	94.33	96.08
1/2	92.64	93.03	92.85	96.11	95.42	94.59	96.79
2/3	92.17	91.76	91.72	96.84	95.89	95.28	96.65
4/5	91.24	91.32	90.94	96.95	96.08	95.17	96.99
9/10	90.26	90.51	90.43	96.74	95.93	95.22	96.40
19/20	89.67	89.06	89.94	95.60	94.79	94.17	95.72

Tabela 51: Średni F1-score klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentów cp i minsplit - zbiór testowy - przycięte drzewo klasyfikacyjne

Podział zbioru	Kombinacje cp i minsplit						
	cp=0.1 minsplit=3	cp=0.1 minsplit=7	cp=0.1 minsplit=12	cp=0.005 minsplit=3	cp=0.005 minsplit=7	cp=0.005 minsplit=12	cp=0.001 minsplit=1
1/20	41.52	31.12	17.30	41.11	31.33	16.96	40.21
1/10	62.50	63.74	35.91	64.22	60.84	39.51	62.27
1/5	79.23	77.85	76.92	81.42	81.08	79.58	80.66
1/3	82.98	83.59	84.35	84.96	84.03	84.17	84.93
1/2	83.89	84.01	84.88	86.74	86.15	85.28	87.12
2/3	83.93	84.12	83.62	88.53	87.53	86.57	88.59
4/5	82.57	83.10	84.15	88.95	87.79	87.61	89.08
9/10	80.17	81.09	82.16	87.30	89.38	86.85	87.46
19/20	77.11	73.60	77.05	82.35	83.66	82.18	82.94

Tabele 48., 49., 50. oraz 51. ukazują nam identyczne zależności do tych uzyskanych przy badaniu nieprzyciętych drzew. Na ogół wyniki są niższe (naturalnie wynika ze "zmniejszenia" dokładności poprzez przycięcie gałęzi), jednak różnice wynoszą ok. 1%-2%. Wyjątek stanowi jedynie ostatnia kolumna - "skracanie" liści spowodowało, że przestały się w nich znajdować pojedyncze obserwacje. Poprawa wyników (najlepsze osiągane dla stosunków 2/3 oraz 4/5) sugeruje nam, że w ten sposób niejako pozbylismy się potencjalnego overfittingu.

2.2.2.3 Najlepszy model na podstawie najlepszych cech dyskryminujących W przypadku drzew klasyfikacyjnych większa ilość cech wpływa na złożoność obliczeniową mniej niż w przypadku metody k-NN. Dzieje się tak, gdyż na odległość między punktami wpływa każda wartość, natomiast algorytm tworzenia drzew wybiera tylko te istotne cechy. Te nieistotne mogą zostać nawet zignorowane. Jednak w ramach optymalizacji nadal warto rozważyć wybranie podzbioru najistotniejszych cech.

Cechy zastosowane są w tej samej kolejności, co w tabeli @ref(tab:zmienne).

Dodatkowo, w ramach optymalizacji rozważane są jedynie drzewka obcięte, gdyż zajmują mniej miejsca oraz działają szybciej (widoczne w szczególności przy badaniu dużych zbiorów danych), a osiągają te wyniki całkiem małym kosztem na dokładności oraz wskaźniku F1. Drzewo to najlepsze wyniki osiągnęło przy parametrach: $cp = 0.005$, $minsplit = 3$.

Pomijamy również badanie parametrów dla zbioru uczącego, gdyż najważniejsze jest jak klasyfikator zachowa się dla nowych zbiorów danych.

Tabela 52: Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - wielokrotny podział - zbiór testowy - przycięte drzewo klasyfikacyjne

Podział zbioru	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1/20	56.53	47.59	69.06	60.71	62.24	56.76	62.35	50.88	67.00	60.53	60.29	61.41	53.35
1/10	66.52	77.70	79.25	71.68	85.96	81.93	75.84	71.93	73.23	76.21	83.48	83.98	81.74

Tabela 52: Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - wielokrotny podział - zbiór testowy - przycięte drzewo klasyfikacyjne (kontynuacja)

Podział zbioru	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1/5	70.07	83.22	86.01	84.83	84.55	81.61	85.10	86.08	80.49	82.80	87.06	82.45	88.60
1/3	75.13	87.73	86.72	87.56	88.40	87.06	88.91	87.98	88.82	85.63	87.39	85.71	88.74
1/2	72.70	90.34	89.55	89.21	90.00	87.87	89.21	89.10	88.54	92.36	91.12	89.33	88.65
2/3	80.50	89.83	86.33	88.83	91.00	87.00	89.83	90.67	88.83	89.00	89.50	91.17	91.17
4/5	78.33	89.72	90.56	91.39	91.11	90.00	91.11	91.67	89.72	90.56	90.28	90.28	92.78
9/10	81.11	90.56	96.67	95.00	85.56	93.89	89.44	90.56	87.22	88.33	86.67	90.00	90.00
19/20	82.22	94.44	94.44	86.67	94.44	83.33	90.00	85.56	88.89	94.44	86.67	92.22	94.44

W Tabeli 52. możemy zauważyć, że już od udziału 1/2 danych treningowych pozwala osiągnąć dokładność w okolicach 90%. Do osiągnięcia tej dokładności potrzebne są też tylko 2 cechy. Co ciekawe, dodawanie kolejnych cech w niektórych przypadkach nie poprawia, a wręcz trochę pogarsza poprawną klasyfikację. Najlepszy wynik został osiągnięty dla 6 cech dyskryminacyjnych oraz udziale danych treningowych 9/10.

Tabela 53: Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - wielokrotny podział - zbiór testowy - przycięte drzewo klasyfikacyjne

Podział zbioru	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1/20	47.49	35.87	62.67	52.69	54.21	46.38	55.08	41.46	61.78	53.96	49.97	52.71	41.76
1/10	62.47	74.93	79.32	67.09	86.11	79.45	72.24	67.22	69.27	73.44	82.36	84.00	80.76
1/5	67.45	83.48	85.97	83.21	84.21	80.00	85.17	85.99	79.15	80.30	87.33	80.83	88.76
1/3	73.52	88.03	86.73	87.51	88.28	86.97	88.83	87.96	88.82	85.45	87.26	85.60	88.75
1/2	71.18	90.91	89.37	89.68	89.89	87.71	89.27	89.06	88.36	92.39	91.47	89.27	88.62
2/3	80.60	89.86	86.00	88.58	90.75	86.84	89.99	90.80	88.54	89.27	89.59	91.11	91.15
4/5	79.15	89.92	90.37	91.23	91.15	90.16	91.36	91.77	89.34	90.02	90.39	90.35	92.76
9/10	80.37	90.32	96.45	95.28	85.10	93.66	88.81	88.13	85.88	87.32	85.38	89.36	89.76
19/20	73.90	86.49	90.65	81.41	93.57	79.66	87.71	85.46	84.18	89.14	83.03	87.02	90.69

W Tabeli 53. wyniki są podobne do poprzednich, z tym że dorby wskaźnik F1 (>85%) otrzymujemy już dla udziału zbioru testowego 1/3, czyli mniej niż poprzednio. Ponownie najwyższy wynik osiągnęło 6 cech oraz udział danych treningowych 9/10.

Tabela 54: Średni dokładność klasyfikacji dla kombinacji parametrów liczby podzbiorów zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - cross-validation - zbiór testowy - przycięte drzewo klasyfikacyjne

Liczba podzbiorów	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
2	81.46	84.83	93.26	93.26	84.83	90.45	86.52	89.33	84.83	89.33	89.33	88.76	85.96
5	77.54	89.27	89.38	88.79	89.90	92.68	92.16	90.46	93.25	92.13	92.73	91.03	90.43
10	80.82	92.16	91.01	90.98	92.75	90.52	89.35	89.90	91.67	87.65	91.08	91.54	88.04
20	79.10	88.33	93.26	93.26	89.79	88.68	89.44	89.86	89.93	87.78	86.60	88.06	87.78
50	78.83	88.67	89.67	90.50	90.17	89.50	85.17	88.17	90.00	85.17	87.33	87.83	88.50
178	79.21	90.45	91.57	91.01	90.45	88.76	85.39	85.96	88.20	86.52	86.52	87.08	85.96

W Tabeli 54. zaobserwować możemy, że dobre wyniki uzyskujemy już dla 2-krotnej walidacji oraz 2 cech dyskryminujących. Ponownie zaobserwować można spadek dokładności dla większej liczby cech dyskryminacyjnych (w tym istotny: 1.13%, przy 5-krotnej walidacji i przejściu z 9 na 10 cech). Najlepszy wynik osiągnęła 5-krotna walidacja dla 9 cech, najlepsze skupisko natomiast znajduje się w okolicy punktu 10-krotna walidacja, 3 cechy.

Tabela 55: Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - cross-validation - zbiór testowy - przycięte drzewo klasyfikacyjne

Liczba podzbiorów	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
2	81.46	85.64	93.48	93.36	85.53	90.29	86.70	88.33	84.31	89.65	89.36	88.20	86.21
5	76.88	90.04	89.58	88.93	90.37	92.59	91.88	90.02	93.52	92.31	92.88	91.30	90.16
10	77.36	91.92	90.39	90.85	91.57	89.63	86.99	88.83	90.58	87.36	91.02	91.11	88.39
20	73.16	86.05	92.13	89.88	84.90	86.21	87.79	88.93	83.64	87.48	81.84	84.54	84.37
50	56.06	63.64	70.02	66.15	65.59	73.71	64.86	64.48	63.51	61.71	70.49	67.47	66.78
178	26.40	30.15	30.52	30.34	30.15	29.59	28.46	28.65	29.40	28.84	28.84	29.03	28.65

W Tabeli 55. zauważyć można, że niezależnie od ilości cech 50 oraz 178-krotna walidacja wypada bardzo słabo (możliwe przeuczenie modelu), oraz 20-krotna, choć akceptowalna, wypada gorzej na tle mniejszej ilości. Najlepsze wyniki osiągane są okolicy 10-krotnej walidacji oraz 3 i 4 cech (dla 4 cech mamy również najwyższy wynik). Wynik powyżej 94% osiągamy również dla wcześniej wymienionych 9 cech - 5-krotnej walidacji.

Tabela 56: Średni dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - bootstrap - zbiór testowy - przycięte drzewo klasyfikacyjne

Podział zbioru	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1/20	55.69	70.22	58.97	54.20	49.88	70.63	65.23	64.11	59.16	62.25	66.07	61.30	68.39
1/10	62.77	80.30	80.94	72.06	74.38	81.99	68.18	78.45	83.06	77.65	64.28	82.28	76.75
1/5	70.37	84.99	84.62	84.58	86.51	80.78	84.72	85.58	85.47	83.58	81.96	79.96	85.45
1/3	78.12	86.70	86.20	86.21	87.98	89.02	89.47	85.01	86.56	83.81	86.01	87.25	87.68
1/2	77.28	88.28	93.22	90.12	88.78	90.42	86.65	90.45	89.54	91.19	87.57	88.51	87.65
2/3	76.12	88.60	92.58	89.14	92.18	89.69	89.26	89.28	92.36	90.86	90.50	90.93	91.80
4/5	78.55	90.72	90.90	92.47	90.57	91.13	89.78	88.23	90.10	91.58	90.74	90.27	91.23
9/10	75.59	88.68	93.22	93.71	93.54	92.01	93.39	90.73	90.10	90.52	90.43	91.65	89.18
19/20	75.64	91.30	92.19	92.65	91.12	92.54	90.04	92.11	91.54	89.72	89.33	90.71	90.83

Tabela 56. pokazuje nam, że przy zastosowaniu metody bootstrap wyniki z udziałem zbioru uczącego 1/3 lub niżej osiągają dokładność poniżej 90%, w przypadku 1/2 pojedyncze minimalnie przekraczają ten próg. Najlepszy wynik osiąga nauka na 19/20 przypadków zbioru, przy wykorzystaniu 4 najlepszych cech dyskryminacyjnych. Duże wartości osiągane są również w okolicy tego punktu.

Tabela 57: Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - bootstrap - zbiór testowy - przycięte drzewo klasyfikacyjne

Liczba podzbiorów	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1/20	45.29	65.23	50.43	43.16	40.38	64.98	58.14	57.91	48.91	53.74	57.61	52.39	62.62
1/10	56.38	79.93	80.18	66.91	70.15	79.52	62.61	76.58	81.80	75.77	55.80	82.11	75.21
1/5	67.33	85.43	84.76	84.81	86.54	79.39	84.56	85.72	84.87	82.07	80.70	77.84	85.48
1/3	78.55	87.05	86.20	86.16	88.07	89.18	89.54	85.14	86.70	83.70	86.04	87.01	87.84
1/2	77.49	88.63	93.40	90.49	88.77	90.68	86.69	90.50	89.52	91.22	87.54	88.59	87.78
2/3	76.42	88.89	92.66	88.93	92.06	89.80	89.33	89.44	92.28	90.93	90.77	90.90	91.91
4/5	78.79	91.17	91.08	92.65	90.51	91.11	89.72	88.51	90.10	91.63	90.84	90.34	91.31
9/10	76.19	89.23	93.30	93.76	93.59	92.38	93.46	90.80	89.61	90.55	90.48	91.53	89.16
19/20	76.26	91.53	92.39	92.83	91.27	92.13	90.17	92.29	91.50	90.11	89.03	90.68	90.87

W Tabeli 57. możemy zaobserwować wyniki podobne do poprzednich. W tym przypadku wskazują one jednak na wysoką wydajność predykcyjną modelu. Najlepszy wynik ponownie uzyskujemy dla udziału zbioru uczącego 9/10 oraz 4 cech, oraz bardzo dobre w jego okolicach.

Podsumowując:

- Walidacja krzyżowa (cross-validation) najlepsze wyniki osiąga dla 3-5 cech oraz liczby podziałów poniżej 10.
- W przypadku tradycyjnego podziału wystarczy zastosować co najmniej 2 najlepiej dyskryminujące cechy, stosunek zbioru uczącego natomiast powinien wynieść co najmniej 2/3.

2.2.3 Naiwny klasyfikator bayesowski

W tej części zajmiemy się porównaniem wyników uzyskiwanych poprzez nieobcięte drzewa, oraz ich optymalnie przycięte odpowiedniki. Następnie najlepsze drzewo zostanie zbadane pod kątem ilości najlepszych cech dyskryminujących, które pozwolą na maksymalizację dokładności oraz wskaźnika F1.

Ta część sprawozdania skupi się na ocenie metody klasyfikacyjnej w postaci naiwnego klasyfikatora bayesowskiego (naiwnego, gdyż zakłada on niezależność zmiennych, tymczasem niekoniecznie jest to prawdą). Następnie za pomocą metod bootstrap, cross-validation oraz wielokrotny podział, ocenimy skuteczność NB w zależności od liczby cech dyskryminujących.

2.2.3.1 *Laplace*: lepsze 0 czy 1? W ramach badania najlepszego naiwnego klasyfikatora bayesowskiego, na początku skupimy się na argumencie *laplace*. W trakcie liczenia odpowiednich prawdopodobieństw warunkowych dodaje ona liczbę do licznika. Głównie powoduje to, że wcześniejsze prawdopodobieństwa zerowe stają się liczbami bliskimi zera, co z kolei może wpływać na wyniki. Ustalone to zostanie przy stosunku 2:1 zbioru uczącego do zbioru testowego dla bootstrap i wielokrotnego podziału, oraz przy 10 częściach dla cross-validation.

Tabela 58: Średnie dokładność klasyfikacji oraz F1-score dla naiwnego klasyfikatora bayesowskiego, z podziałem na metody oceny oraz argument laplace

	Wielokrotny podział	Cross-validation	Bootstrap	Średnio
Dokładność (laplace = 0)	97.08	97.16	96.85	97.03
Dokładność (laplace = 1)	97.12	97.78	97.15	97.35
F1 score (laplace = 0)	97.11	97.24	96.94	97.10
F1 score (laplace = 1)	97.14	97.99	97.22	97.45

W tabeli 58. widać, że argument *laplace* niewiele wpływa zarówno na dokładność, jak i F1-score. Minimalnie większe wyniki osiąga naiwny klasyfikator bayesowski dla *laplace* = 0, z wyjątkiem wskaźnika F1 dla wielokrotnego podziału.

2.2.3.2 Najlepszy model Przy ocenie najlepszego modelu ponownie zostanie zastosowany odpowiedni udział zbioru uczącego dla metod wielokrotnego podziału i bootstrap: 1/20, 1/10, 1/5, 1/3, 1/2, 2/3, 4/5, 9/10 oraz 19/20, oraz stosowny podział dla walidacji krzyżowej: 2, 5, 10, 20, 50, 178 (Leave-One-Out). W ramach badania parametru ponownie zobaczymy wpływ zmiennej *laplace* wynoszącej 0 lub 1.

Tabela 59: Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentu laplace - zbiór treningowy - naiwny klasyfikator bayesowski

Podział zbioru	Laplace	
	0	1
1/20	94.50	94.12
1/10	99.65	99.59
1/5	99.54	99.57
1/3	98.85	98.76
1/2	98.55	98.60
2/3	98.37	98.47

Tabela 59: Średnia dokładność klasyfikacji dla kombinacji paramet (kontynuacja)

Podział zbioru	Laplace	
	0	1
4/5	98.45	98.53
9/10	98.42	98.56
19/20	98.60	98.60

W Tabeli 59. widać, że dla zbiorów uczących otrzymujemy bardzo dobrą dokładność dla obu parametrów (powyżej 94%). Różnice między poszczególnymi parametrami przy odpowiednich stosunkach zbioru uczącego są zróżniocowane, jednak niewielkie (poniżej 1%).

Tabela 60: Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz argumentu laplace - zbiór testowy - naiwny klasyfikator bayesowski

Podział zbioru	Laplace	
	0	1
1/20	58.50	57.79
1/10	83.96	84.43
1/5	94.22	94.71
1/3	96.34	96.07
1/2	96.79	97.09
2/3	97.70	97.30
4/5	96.97	97.25
9/10	96.89	96.83
19/20	97.56	96.89

Tabela 60. pokazuje, że wcześniejszy trend został jedynie częściowo zachowany. Dla stosunków 1/20 oraz 1/10 zbiór treningowy jest za mały, przez co dostajemy odpowiednio słabą oraz nienajlepszą kategoryzację. Natomiast pozostałe wyniki wynoszą ponad 94% i ponownie, różnice są bardzo małe.

Tabela 61: Średni F1-score dla kombinacji parametrów podziału zbioru oraz argumentu laplace - zbiór treningowy - naiwny klasyfikator bayesowski

Podział zbioru	Laplace	
	0	1
1/20	78.10	78.91
1/10	98.79	98.44
1/5	99.53	99.59
1/3	98.87	98.79
1/2	98.58	98.62
2/3	98.40	98.51

Tabela 61: Średni F1-score dla kombinacji parametrów podziału zbioru oraz argumentu laplace - zbiór treningowy - naiwny klasyfikator bayesowski (kontynuacja)

Podział zbioru	Laplace	
	0	1
4/5	98.49	98.56
9/10	98.46	98.60
19/20	98.63	98.64

Tabela 61. wskazuje na to, że oprócz stosunku zbioru uczącego 1/20, otrzymujemy bardzo wysokie wskaźniki F1 (powyżej 98%). Mówi to nam, że zbiorów treningowych naiwny klasyfikator bayesowski bardzo dobrze radzi sobie z rozróżnianiem nierównowagi klas.

Tabela 62: Średni F1-score dla kombinacji parametrów podziału zbioru oraz argumentu laplace - zbiór testowy - naiwny klasyfikator bayesowski

Podział zbioru	Laplace	
	0	1
1/20	49.24	49.46
1/10	82.54	83.40
1/5	94.40	94.90
1/3	96.46	96.21
1/2	96.88	97.15
2/3	97.72	97.34
4/5	96.92	97.17
9/10	96.72	96.61
19/20	92.04	92.25

W Tabeli 62.widzimy, podobnie jak między dokładnością w zbiorach uczącym i treningowym, prawie zachowany trend. Dla stosunku zbioru uczącego 1/20 otrzymujemy bardzo słaby wynik, a dla 1/10 - raczej słaby. Dla pozostałych F1-score jest wysoki (powyżej 94%). Istnieje również zgodność z dokładnoscią - parametr, który lepiej zaklasyfikował gatunki win, miał również lepszy wskaźnik F1. Wyjątkiem jest stosunek 1/20.

2.2.3.3 Najlepszy model na podstawie najlepszych cech dyskryminujących Podobnie jak wcześniej, cechy zastosowane są w tej samej kolejności, co w tabeli @ref(tab:zmienne). Zastosowany zostanie jedynie parametr *laplace* = 0, jednak jako iż wcześniejsze badania to wykazały, wyniki otrzymane dla 1 byłyby bardzo zbliżone, i osiągnięte różnice byłyby niewielkie. I tak samo jak wcześniej, rozważymy tylko wyniki dla zbioru testowego.

Tabela 63: Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - wielokrotny podział - zbiór testowy - naiwny klasyfikator bayesowski (laplace=0)

Podział zbioru	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1/20	59.91	60.88	66.32	65.44	65.62	64.55	63.19	63.51	59.23	58.04	57.37	60.38	59.01
1/10	73.76	83.91	86.05	88.56	88.17	87.11	88.78	87.59	82.58	84.43	85.40	83.34	83.93
1/5	78.32	88.91	93.36	94.64	94.81	94.02	95.04	94.28	93.62	93.86	93.60	94.26	94.12
1/3	79.25	89.62	94.39	95.36	96.27	94.71	95.64	95.25	94.75	95.49	95.67	95.82	96.38
1/2	79.78	89.85	94.29	95.84	96.34	95.11	96.53	95.61	95.28	95.82	95.93	96.53	96.82
2/3	79.22	89.57	95.02	95.85	96.33	95.43	96.13	95.87	95.20	96.32	96.13	96.55	97.28
4/5	80.03	90.44	94.86	96.03	97.00	95.33	96.08	95.25	94.94	96.50	95.89	95.92	97.36
9/10	80.56	90.67	94.89	95.33	96.83	95.17	96.06	95.44	95.50	97.06	96.00	95.83	97.17
19/20	78.56	89.00	95.44	94.78	96.78	94.78	96.11	96.22	94.78	97.00	95.33	96.44	97.56

W Tabeli 63. widzimy niezadowalające wyniki dla stosunku zbioru uczącego 1/20, oraz dla tylko 1 cechy dyskryminującej. Oprócz pojedynczych przypadków zauważyć można, że dla każdej liczby stosunków liczba cech odpowiednio zwiększa lub zmniejsza dokładność. "Stabilne" wyniki, pozostające stale powyżej 90%, otrzymujemy dla 3 cech, z wyjątkiem stosunku 1/10, który oscyluje między 80% a 90%. Najwyższe wyniki uzyskujemy dla wszystkich 13 cech.

Tabela 64: Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - wielokrotny podział - zbiór testowy - naiwny klasyfikator bayesowski (laplace=0)

Podział zbioru	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1/20	53.59	52.49	58.30	57.64	57.91	57.48	56.16	55.92	50.19	49.05	48.75	53.34	51.57
1/10	73.16	83.62	84.98	87.49	87.04	86.02	87.95	86.59	81.30	83.15	84.17	81.77	82.30
1/5	78.98	89.45	93.52	94.72	94.90	94.12	95.12	94.51	93.73	94.05	93.69	94.44	94.27
1/3	80.06	90.13	94.60	95.48	96.41	94.77	95.73	95.42	94.92	95.68	95.81	95.95	96.50
1/2	80.49	90.45	94.49	95.95	96.47	95.16	96.63	95.72	95.42	95.98	96.04	96.61	96.89
2/3	79.86	90.08	95.18	95.94	96.44	95.38	96.19	95.89	95.26	96.40	96.13	96.61	97.34
4/5	80.39	90.66	94.99	96.07	96.94	95.19	96.01	95.26	94.90	96.52	95.85	95.93	97.31
9/10	79.91	90.29	94.82	94.75	96.13	94.79	95.70	94.44	95.34	96.95	95.44	95.66	97.01
19/20	75.38	85.06	92.21	91.61	94.36	92.11	91.91	94.15	91.68	94.36	91.93	92.57	95.32

W Tabeli 64. wielokrotny podział zapewnie trend podobny do tego z tabeli 63.: stosunek 1/20 oraz 1 cecha dają słaby wskaźnik F1, 1/10 daje jedynie zadowalający, od 3 cech otrzymujemy stabilne F1-score powyżej 90%, oraz najlepsze dostajemy dla 13 cech (z wyjątkiem stosunku 1/5, tam 12 cech daje niewiele, ale lepszy wynik).

Tabela 65: Średnia dokładność klasyfikacji dla kombinacji parametrów liczba podzbiorów oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - cross-validation - zbiór testowy - naiwny klasyfikator bayesowski (laplace=0)

Liczba podzbiorów	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
2	76.40	87.08	94.94	96.63	96.63	94.94	95.51	96.63	96.07	94.94	95.51	93.82	97.19
5	80.83	89.94	93.84	96.05	97.19	94.95	96.10	95.49	94.95	96.65	94.92	95.52	96.11
10	80.20	89.77	95.56	94.93	97.19	94.97	96.60	95.98	95.00	96.63	96.05	96.67	97.75
20	80.83	89.93	95.00	95.49	96.67	94.86	96.04	96.11	94.86	96.60	96.04	95.49	97.78
50	80.83	89.00	94.67	95.50	97.33	95.00	96.17	96.00	95.50	95.83	95.67	96.50	98.00
178	80.90	89.89	94.38	95.51	97.19	94.94	96.07	96.07	94.94	96.63	95.51	96.63	97.19

Tabela 65. pokazuje bardzo dobre wyniki wyznaczone przez krzyżową walidację. Dla 1 i 2 cech otrzymujemy jedynie wyniki poniżej 90% (w tym poniżej 80% tylko dla 1 cechy i 2 podziałów). Podobnie jak wcześniej, najlepsze wyniki otrzymaliśmy dla 13 cech (z wyjątkiem 5 podziałów, gdzie wystarczyło 5 cech).

Tabela 66: Średni F1-score dla kombinacji parametrów liczby podzbiorów oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - cross-validation - zbiór testowy - naiwny klasyfikator bayesowski (laplace=0)

Liczba podzbiorów	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
2	76.93	87.52	95.17	96.73	96.81	94.99	95.55	96.75	96.15	95.14	95.62	93.75	97.19
5	80.96	90.15	94.02	96.06	97.40	94.91	95.81	95.61	94.95	96.81	94.93	95.50	96.11
10	79.28	89.03	95.61	94.98	97.19	95.02	96.55	95.66	95.03	96.54	95.85	96.56	97.49
20	79.69	87.34	93.04	89.67	93.47	93.25	92.36	91.31	89.05	95.10	87.80	93.16	94.47
50	60.41	65.58	69.38	74.40	74.84	70.44	70.71	68.84	71.55	71.16	76.36	72.70	70.31
178	26.97	29.96	31.46	31.84	32.40	31.65	32.02	32.02	31.65	32.21	31.84	32.21	32.40

W Tabeli 66. widzimy, że wskaźniki F1 zachowują się wręcz identycznie jak pomiary dokładności, z wyjątkiem podziałów 50 oraz 178. Dla 50 wyniki są poniżej 80%, zatem nienajlepiej radzi sobie z wykrywaniem pozytywnych przypadków, lub poprawnym rozpoznawaniem negatywnych przypadków. Dla 178 wyniki są poniżej 33%, czyli modele te są bardzo słabymi modelami (pomimo wysokiej dokładności).

Tabela 67: Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - bootstrap - zbiór testowy - naiwny klasyfikator bayesowski (laplace=0)

Podział zbioru	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1/20	59.79	62.23	64.25	65.03	63.91	62.63	63.84	62.01	59.56	60.81	59.27	57.70	58.62

Tabela 67: Średnia dokładność klasyfikacji dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - bootstrap - zbiór testowy - naiwny klasyfikator bayesowski (laplace=0) (kontynuacja)

Podział zbioru	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1/10	73.62	83.63	86.60	88.20	85.77	87.79	87.23	85.14	84.65	83.16	83.41	82.33	81.06
1/5	77.75	88.60	93.01	93.95	94.11	93.75	94.89	93.40	92.93	93.75	93.06	92.53	93.44
1/3	79.21	89.42	94.43	95.12	95.13	94.50	95.52	95.40	94.34	95.10	95.12	94.90	95.63
1/2	79.67	89.87	94.07	95.96	95.98	94.74	95.88	95.90	95.15	95.91	95.68	96.13	96.48
2/3	79.47	89.96	94.54	95.72	96.23	94.89	95.99	95.82	95.41	96.43	95.81	96.25	96.32
4/5	79.51	89.80	94.13	96.07	96.41	94.84	96.09	95.71	95.58	96.03	96.13	96.17	96.94
9/10	80.06	89.76	94.49	95.89	96.41	95.60	96.26	96.28	95.38	96.43	96.21	96.59	97.25
19/20	79.65	90.05	95.07	95.99	95.85	94.78	96.22	95.78	95.60	96.74	95.97	96.51	97.15

W Tabeli 67. widać, że dokładność osiągana przy metodzie boostrap jest bardzo podobna do tej osiąganej przez wielokrotny podział, nie wnoszą zatem nic nowego.

Tabela 68: Średni F1-score dla kombinacji parametrów podziału zbioru oraz liczby zmiennych o najlepszej zdolności dyskryminacyjnej - bootstrap - zbiór testowy - naiwny klasyfikator bayesowski (laplace=0)

Podział zbioru	Liczba zmiennych o najlepszych zdolnościach dyskryminacyjnych												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1/20	53.27	53.98	56.12	56.95	55.42	53.64	56.91	53.88	50.73	53.99	51.62	50.11	49.49
1/10	72.99	83.08	85.62	87.03	84.65	86.80	85.86	84.32	83.60	81.58	81.57	81.17	79.28
1/5	78.36	89.12	93.22	94.01	94.17	93.80	95.02	93.54	92.98	93.82	93.27	92.62	93.60
1/3	79.97	89.99	94.60	95.25	95.28	94.56	95.61	95.53	94.50	95.30	95.29	95.02	95.78
1/2	80.36	90.37	94.30	96.07	96.11	94.80	95.97	96.01	95.29	96.05	95.81	96.24	96.58
2/3	80.19	90.50	94.75	95.85	96.38	94.94	96.07	95.93	95.55	96.54	95.93	96.34	96.44
4/5	80.19	90.27	94.32	96.14	96.49	94.85	96.18	95.81	95.70	96.14	96.24	96.24	97.03
9/10	80.67	90.18	94.67	95.98	96.49	95.58	96.34	96.33	95.49	96.56	96.30	96.67	97.29
19/20	80.26	90.54	95.25	96.05	96.00	94.83	96.29	95.88	95.73	96.81	96.02	96.60	97.19

Tabele 67. oraz 68., wykonane metodą boostrap, dają wyniki bardzo zbliżone do tych osiągniętych wielokrotnym podziałem, zatem nie wnoszą nic nowego.

Podsumowując:

- Dla tradycyjnego podziału optymalny udział danych treningowych wynosi ponad 20% (podział 1/5), przy czym najlepsze rezultaty uzyskujemy wykorzystując wszystkie cechy (jednak już dla 4 osiągamy bardzo wysokie dokładność oraz F1-score).
- W przypadku walidacji krzyżowej (cross-validation) wystarczy stosować liczbę podziałów poniżej 50 (najbardziej optymalnie - 10 foldów) oraz wybór co najmniej 5 najsilniej dyskryminujących cech.

2.3 Porównanie metod klasyfikacji

Najważniejszymi parametrami porównywania metod klasyfikacji są przede wszystkim czas działania oraz dokładność metody.

Porównamy to dla najlepszych parametrów każdej metody, czyli:

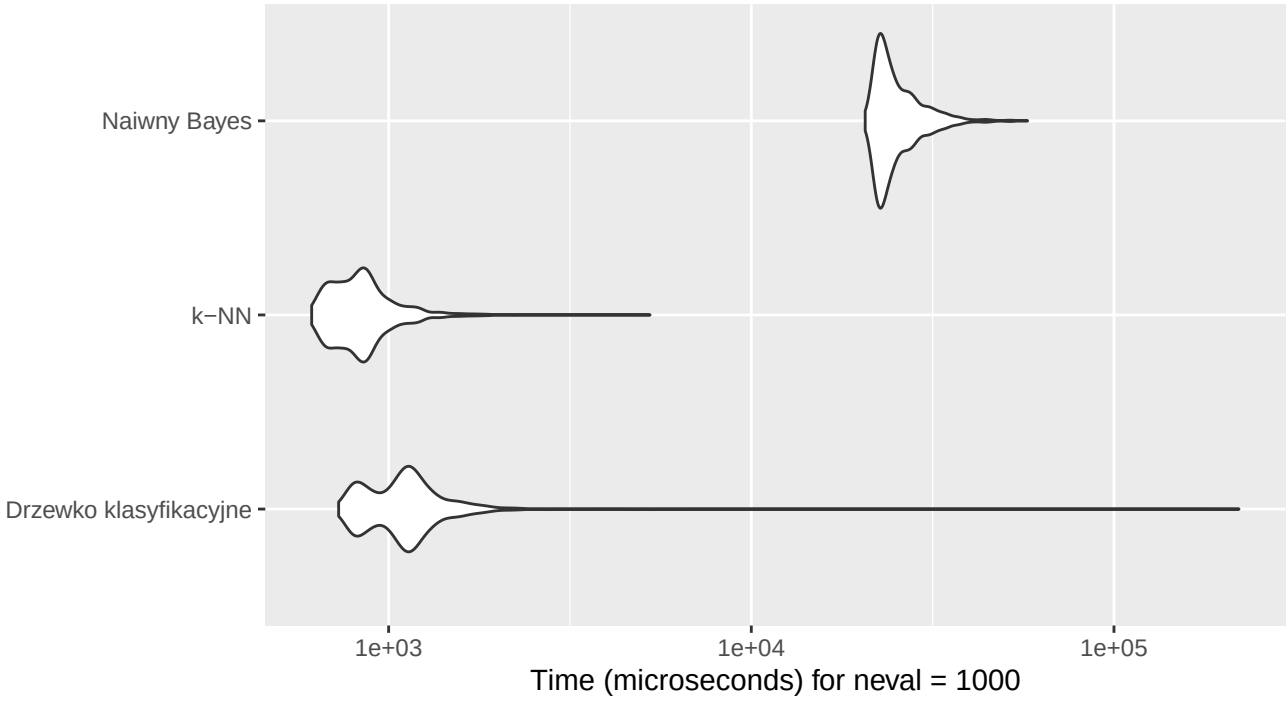
- *liczba sąsiadów* = 1 oraz 8 najlepszych cech - k-NN,
- *cp* = 0.005, *minsplit* = 3 oraz 3 najlepsze cechy - drzewo klasyfikacyjne,
- *laplace* = 0 oraz 5 najlepszych cech- naiwny klasyfikator bayesowski.

Każdy zbiór będzie uczony na tym samym modelu, wynoszącym 9/10 danych, klasyfikować będzie natomiast cały zbiór.

Najwyższy F1-score uzyskał **Naive Bayes** (97,29 %), nieznacznie przewyższając **k-NN** (96,53 %). **Drzewo klasyfikacyjne** osiągnęło nieco niższą maksymalną wartość (93,76 %). Oznacza to, że przy optymalnych parametrach Naive Bayes zapewnia minimalnie lepszą precyzję i czułość niż k-NN, podczas gdy drzewo decyzyjne, choć wciąż skuteczne, nie dorównuje obu pozostałym metodom pod względem czystej dokładności.

Z wcześniejszych badań otrzymaliśmy, że najgorszą zdolność dyskryminacyjną otrzymaliśmy dla drzew, najlepsze zaś dla naiwnego Bayesa.

Na Wykresie 12. widać czas działania najlepszych wersji metod do klasyfikacji. Widać że najszybciej radzi sobie k-NN, a jajdłużej Naiwny Bayes.



Wykres 12: Porównanie klasyfikacji całego zbioru dla konkretnych metod

2.4 Wnioski

- Rozszerzenie przestrzeni cech w regresji liniowej znacząco poprawia jakość klasyfikacji – zarówno na zbiorze treningowym, jak i testowym odnotowano wyraźny spadek błędu oraz wzrost dokładności i F1-

score. Pokazuje to, że skuteczność regresji liniowej jako metody klasyfikacji silnie zależy od odpowiedniej reprezentacji danych wejściowych.

- Naiwny Bayes (laplace = 0, 5 cech) osiągnął najwyższy F1-score (97,29 %), co dowodzi jego przewagi w precyzji i czułości przy optymalnych ustawieniach.
- k-NN (k = 1, 8 cech) był najszybszy w klasyfikacji pełnego zbioru, co czyni go doskonałym wyborem w aplikacjach czasu rzeczywistego, pod warunkiem zapewnienia wystarczającej pamięci na przechowanie danych treningowych.
- Najmniej wrażliwy na przeuczenie okazał się Naiwny Bayes – wąski zakres wyników F1 przy różnych konfiguracjach potwierdza stabilność tego modelu.
- Wszystkie metody znacząco zyskują przy wzroście udziału danych treningowych od 20 % do 90 %, jednak k-NN i Bayes odnotowały największy wzrost skuteczności.
- Drzewo decyzyjne (cp = 0,005, minsplit = 3, 3 cechy) oferuje najczytelniejszy model reguł, z akceptowalnym F1-score (93,76 %), co czyni je rekomendowanym wyborem tam, gdzie wymagana jest pełna przejrzystość decyzji.
- Rekomendacje praktyczne:

    1. **Gdy priorytetem jest maksymalna dokładność i stabilność** – wybór pada na Naiwny Bayes.
    2. **Gdy kluczowa jest szybkość klasyfikacji** – najlepszym rozwiązaniem jest k-NN.
    3. **Gdy potrzebna jest interpretowalność i kontrola złożoności** – optymalne okażą się drzewa decyzyjne.

Ogólny wniosek: Dobór metody powinien zależeć od specyfiki projektu — jeżeli wymagane są maksymalne wyniki i odporność na zmienność danych, rekomendujemy Naiwny Bayes; jeżeli liczy się czas, preferowany jest k-NN; w przypadku konieczności transparentności modelu i łatwej analizy reguł optymalnym wyborem pozostają drzewa decyzyjne.

PS. Czas wykonywania kodu wynosi 46 minut i 30 sekund.