Raport - Lista 3

Paweł Karwecki 282249

2025-06-03

Spis treści

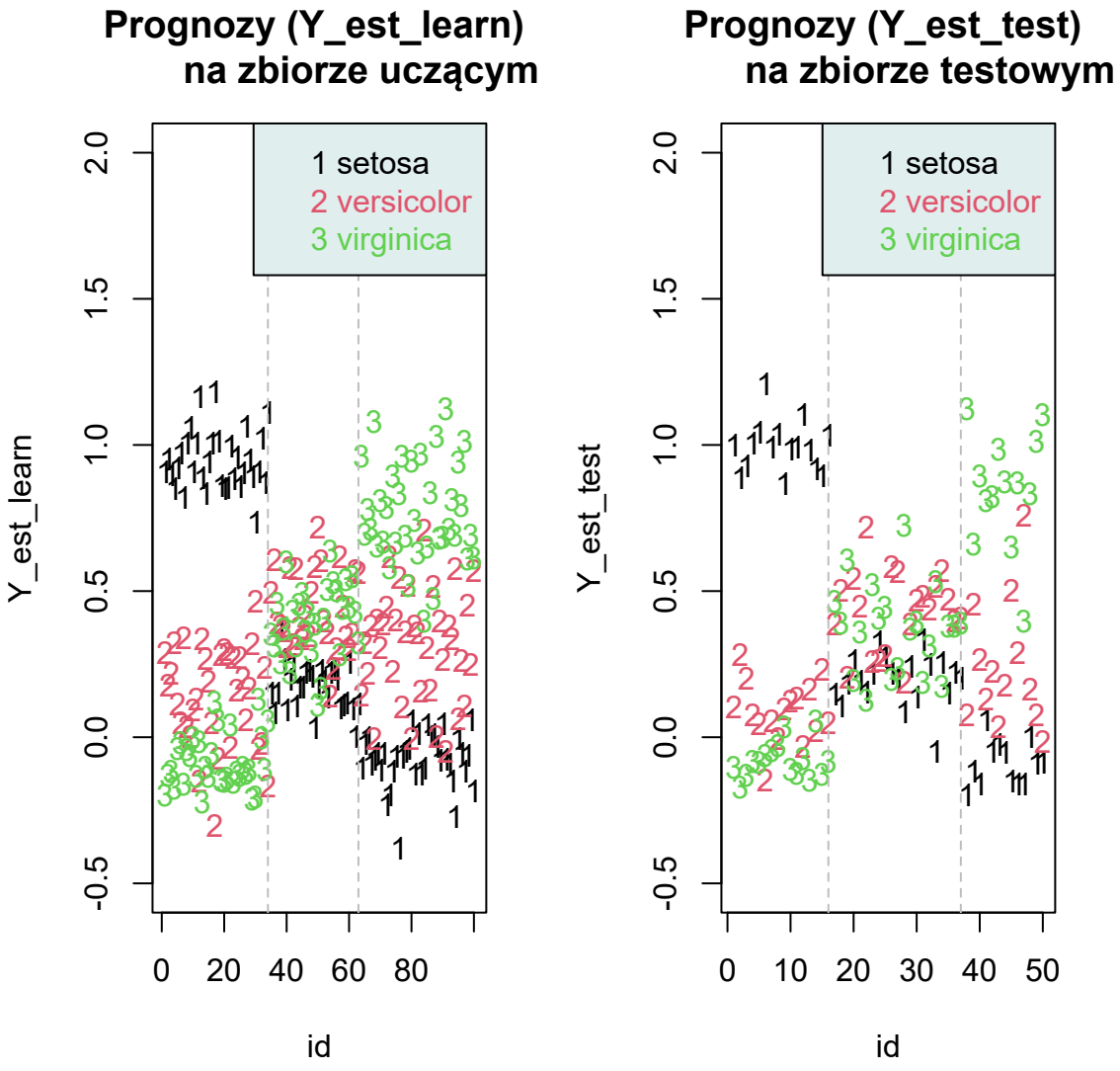
1 Zadanie 1 - Klasyfikacja na podstawie modelu regresji

Tabela 1: Rodzaj zmiennych i wytłumaczone nazwy cech

Nazwa zmiennej	Typ zmiennej	Opis zmiennej
Sepal.Length	numeric	Długość kielicha
Sepal.Width	numeric	Szerokość kielicha
Petal.Length	numeric	Długość płatka
Petal.Width	numeric	Szerokość płatka
Species	factor	Gatunek

Tabela 1 zawiera wytłumaczone nazwy kolumn zbioru danych `iris`.

1.1 Konstrukacja klasyfikatora i wzynaczenie prognoz



Rysunek 1: Prawdopodobieństwa przynależności do danej klasy wyznaczone na podstawie modelu regresji

Na rysunku 1 widać, że zarówno w zbiorze uczącym, jak i zbiorze testowym jest 100% skuteczność poprawnego przypisania klasy `setosa` do obiektu. Problemy zaczynają się w przypadku gatunków `versicolor` i `viriginca`. Między 1. i 2. szarą, przerywaną linią są punkty, które mają w rzeczywistości etykietę `versicolor`, widać, że prognozowana etykieta nie zawsze na to wskazuje. Jest duże zagęszczenie na poziomie `Y_est_learn` $= 0.5$ i `Y_est_test` $= 0.5$ punktów z tych dwóch gatunków, co wskazuje na brak jednoznaczności w przypisaniu klasy.

1.2 Ocena jakości modelu

Tabela 2: Macierz pomyłek dla zbioru uczącego

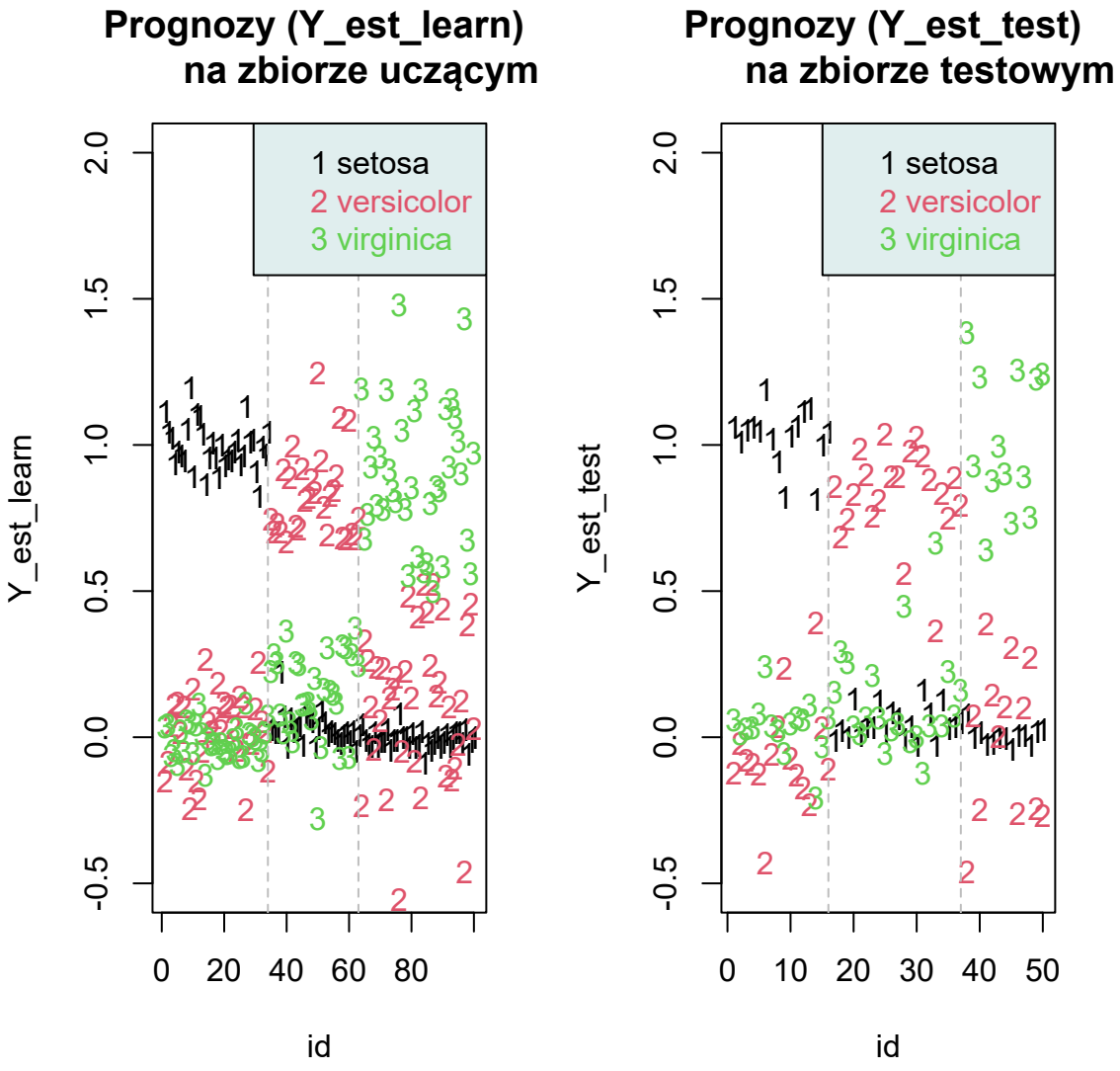
rzecz_etyk	prog_etyk_test		
	setosa	versicolor	virginica
setosa	34	0	0
versicolor	0	14	15
virginica	0	4	33

Tabela 3: Macierz pomyłek dla zbioru testowego

rzecz_etyk	prog_etyk_test		
	setosa	versicolor	virginica
setosa	16	0	0
versicolor	0	14	7
virginica	0	1	12

Tabela 2 to macierz pomyłek dla zbioru uczącego. Na jej podstawie wnioskujemy, że błąd klasyfikacji wynosi 0.19. Tabela 3 to też macierz pomyłek, ale dla zbioru testowego. Z niej wynika, że błąd klasyfikacji dla tego zbioru wynosi 0.16. W obu tabelach można zauważyć problem z przypisaniem właściwej klasy punktom z gatunku `versicolor`. W tabeli związanej ze zbiorem uczącym, ponad 50% obiektów z gatunku `versicolor` otrzymało etykietę `virginica`. Co prawda zbiór testowy poradził sobie lepiej z klasyfikacją, ale i tak 33% obiektów klasy `versicolor` przypisano zły gatunek. Klasyfikator dobrze by sobie poradził, gdybyśmy mieli dwie klasy: `setosa` i `non-setosa`. Trzy klasy powodują, że dochodzi do efektu maskowania.

1.3 Budowa modelu liniowego dla rozszerzonej przestrzeni cech



Rysunek 2: Prawdopodobieństwa przynależności do danej klasy wyznaczone na podstawie modelu regresji dla rozszerzonej przestrzeni cech

Z wykresu 2 można wywnioskować, że uzupełnienie wyjściowych cech o składniki wielomianowe stopnia 2 znacznie ulepszyło klasyfikację. Niezależnie od zbioru, klasyfikator dobrze przypisuje etykiety obiektom z danej klasy.

Tabela 4: Macierz pomyłek dla zbioru uczącego

rzecz_etyk	prog_etyk_test		
	setosa	versicolor	virginica
setosa	34	0	0
versicolor	0	29	0
virginica	0	1	36

Tabela 5: Macierz pomyłek dla zbioru testowego

rzecz_etyk	prog_etyk_test		
	setosa	versicolor	virginica
setosa	16	0	0
versicolor	0	20	1
virginica	0	0	13

Macierze z tabel 4 i 5 potwierdzają wnioski z wykresu 2. Zbiór uczący ma błąd klasyfikacyjny równy 0.01, a zbiór testowy 0.02.

Podsumowując, w przypadku liczby klas równej 3, lepiej zastosować składniki wielomianowe stopnia 2 do zbudowania klasyfikatora, co spowoduje zmniejszenie błędu klasyfikacji.

2 Zadanie 2 - Porównanie metod klasyfikacji

2.1 Opis i podstawowe cechu zbioru danych

Porównanie zostanie przeprowadzone na zbiorze danych `Vehicle` z pakietu `mlbench`.

Tabela 6: Rodzaj zmiennych i wytłumaczone nazwy cech

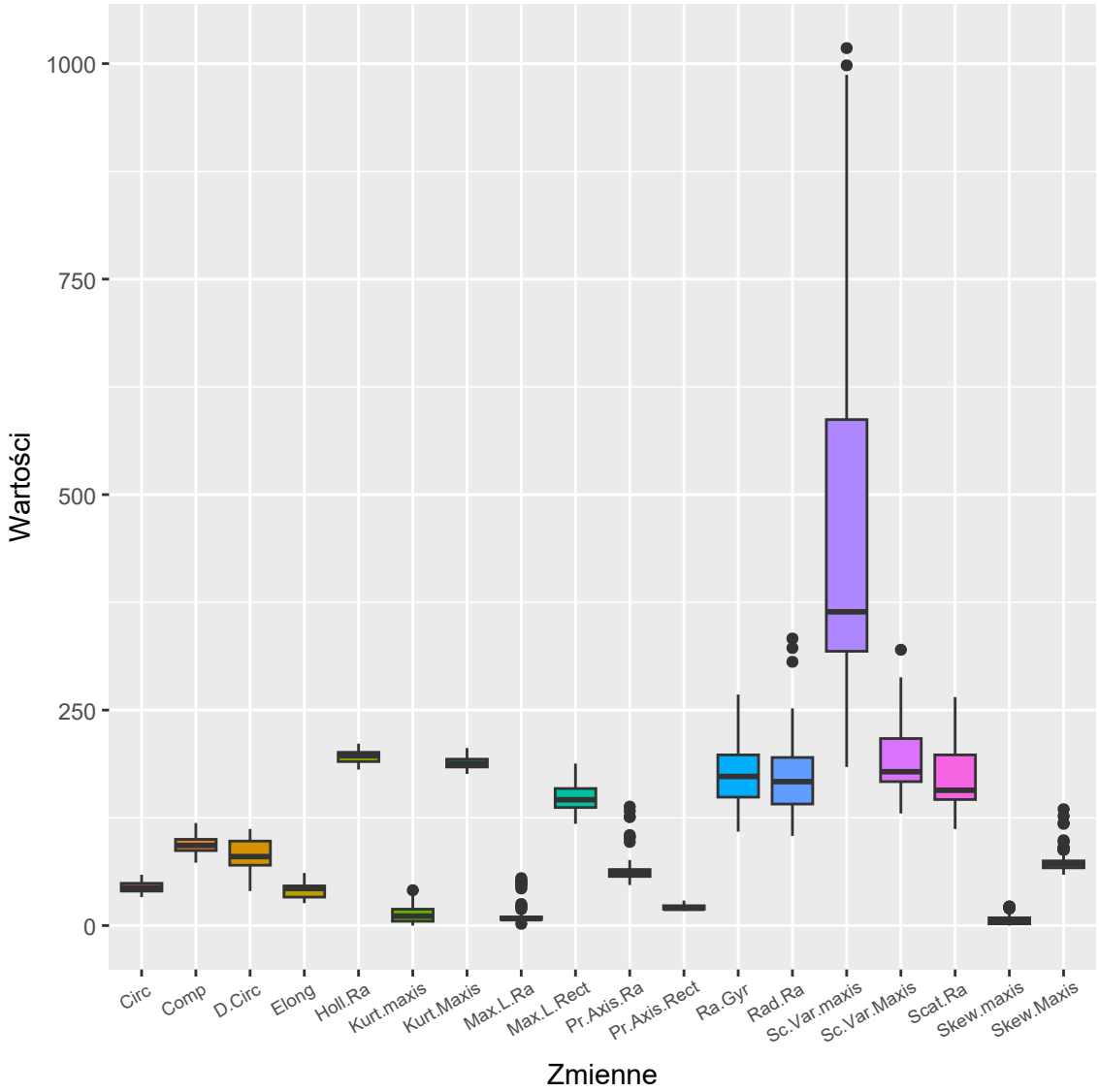
nazwy_zmiennych	typy_zmiennych	opis
Comp	numeric	Zwartość
Circ	numeric	Kolistość
D.Circ	numeric	Kolistość odległościowa
Rad.Ra	numeric	Stosunek promieni
Pr.Axis.Ra	numeric	Stosunek osi głównych
Max.L.Ra	numeric	Maksymalny stosunek długości
Scat.Ra	numeric	Stosunek rozproszenia
Elong	numeric	Wydłużenie
Pr.Axis.Rect	numeric	Prostokątność osi głównych
Max.L.Rect	numeric	Prostokątność maksymalnej długości
Sc.Var.Maxis	numeric	Skalowana wariancja wzdłuż głównej osi
Sc.Var.maxis	numeric	Skalowana wariancja wzdłuż pobocznej osi
Ra.Gyr	numeric	Skalowany promień bezwładności
Skew.Maxis	numeric	Skośność względem głównej osi
Skew.maxis	numeric	Skośność względem pobocznej osi
Kurt.maxis	numeric	Kurtoza względem pobocznej osi
Kurt.Maxis	numeric	Kurtoza względem głównej osi
Holl.Ra	numeric	Stosunek pustek
Class	factor	Typ klasy

Zbiór danych zawiera 846 przypadków i 19 cech. Mamy 4 klasy, które są w kolumnie `Class`. W ramce danych nie występują wartości brakujące. Wszystkie zmienne mają poprawnie przypisany typ, nie ma nietypowych własności danych.

2.2 Wstępna analiza danych

```
##
##  bus opel saab  van
##  218  212  217  199
```

Rozkład klas jest mniej więcej równy, wyróżnia się jedynie `van`, w którym jest mniej niż 200 obiektów. Gdybyśmy przypisywali wszystkie obiekty do najliczniejszej klasy (`bus`), to błąd klasyfikacji wynosiłby ponad 74%.

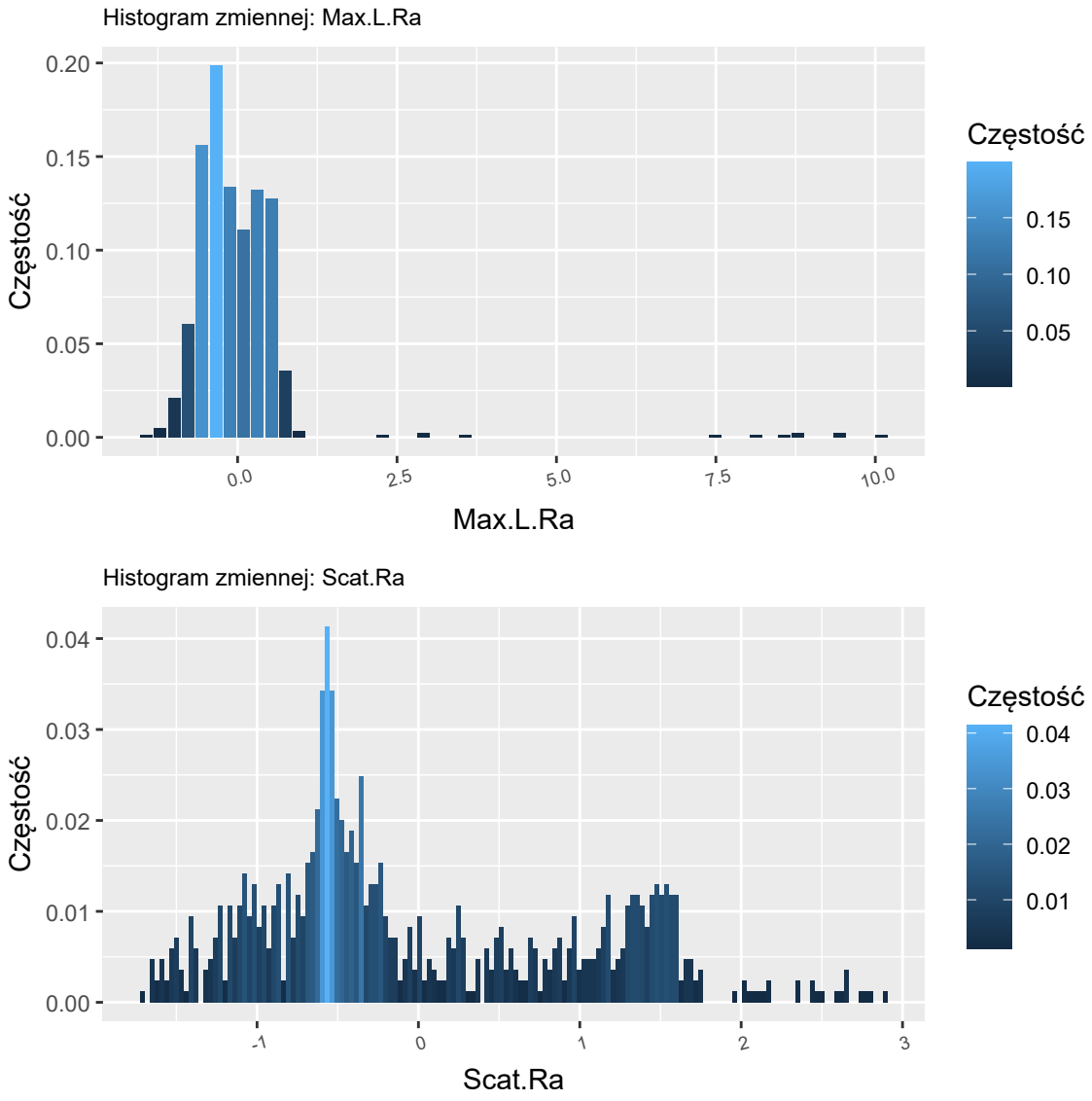


Rysunek 3: Wykresy pudełkowe poszczególnych zmiennych

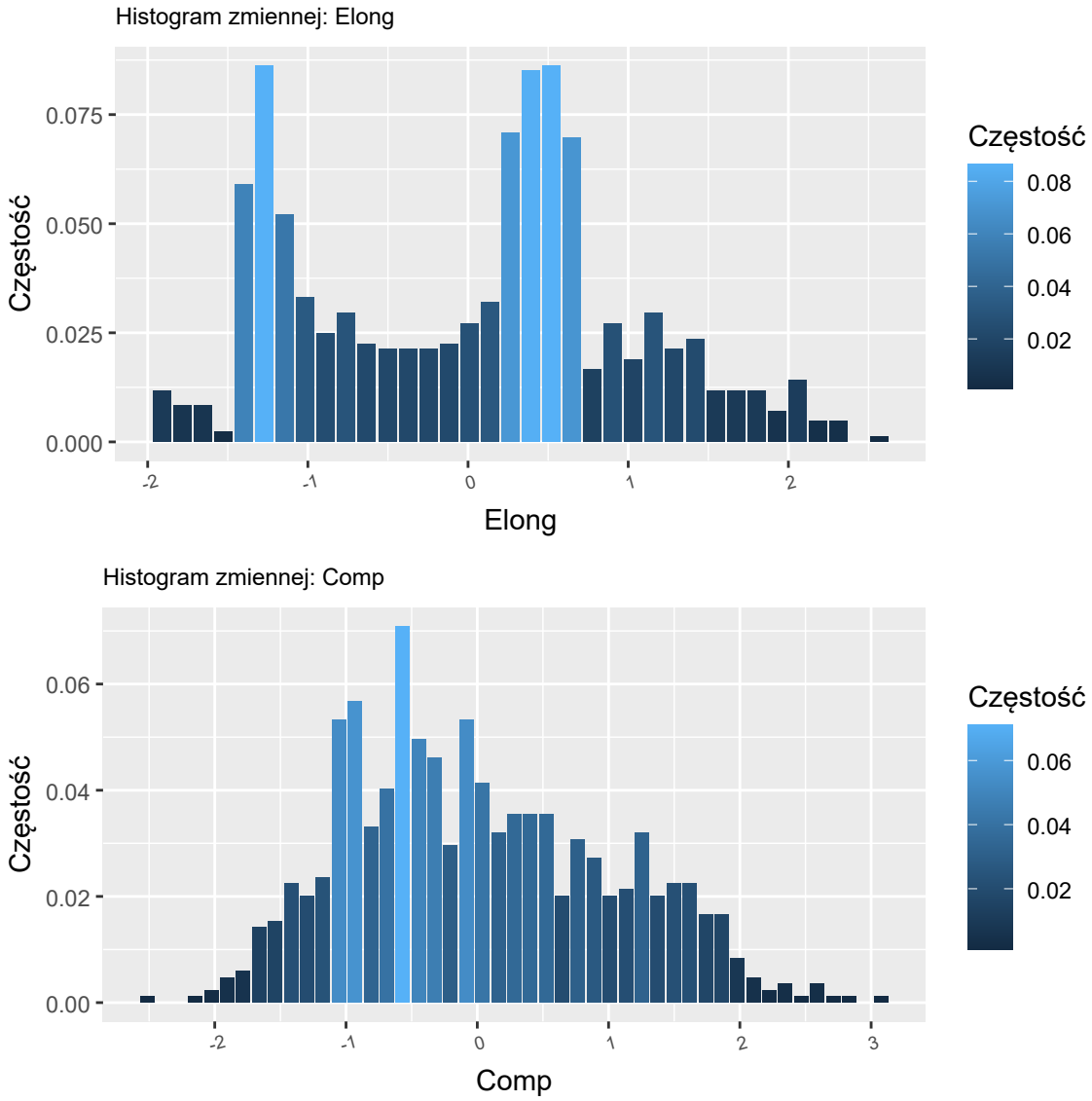
Jak widać na rysunku 3, zmienne przyjmują bardzo różne wartości i wariancję. W przypadku metod drzew klasyfikacyjnych i naiwnego klasyfikatora bayesowskiego standaryzacja nie zmieni za bardzo wyników, ale

w metodzie k-najbliższych sąsiadów będzie ona potrzebna, gdyż wartości z kolumny `Sc.Var.maxis` będą niwelować różnice z pozostałych kolumn.

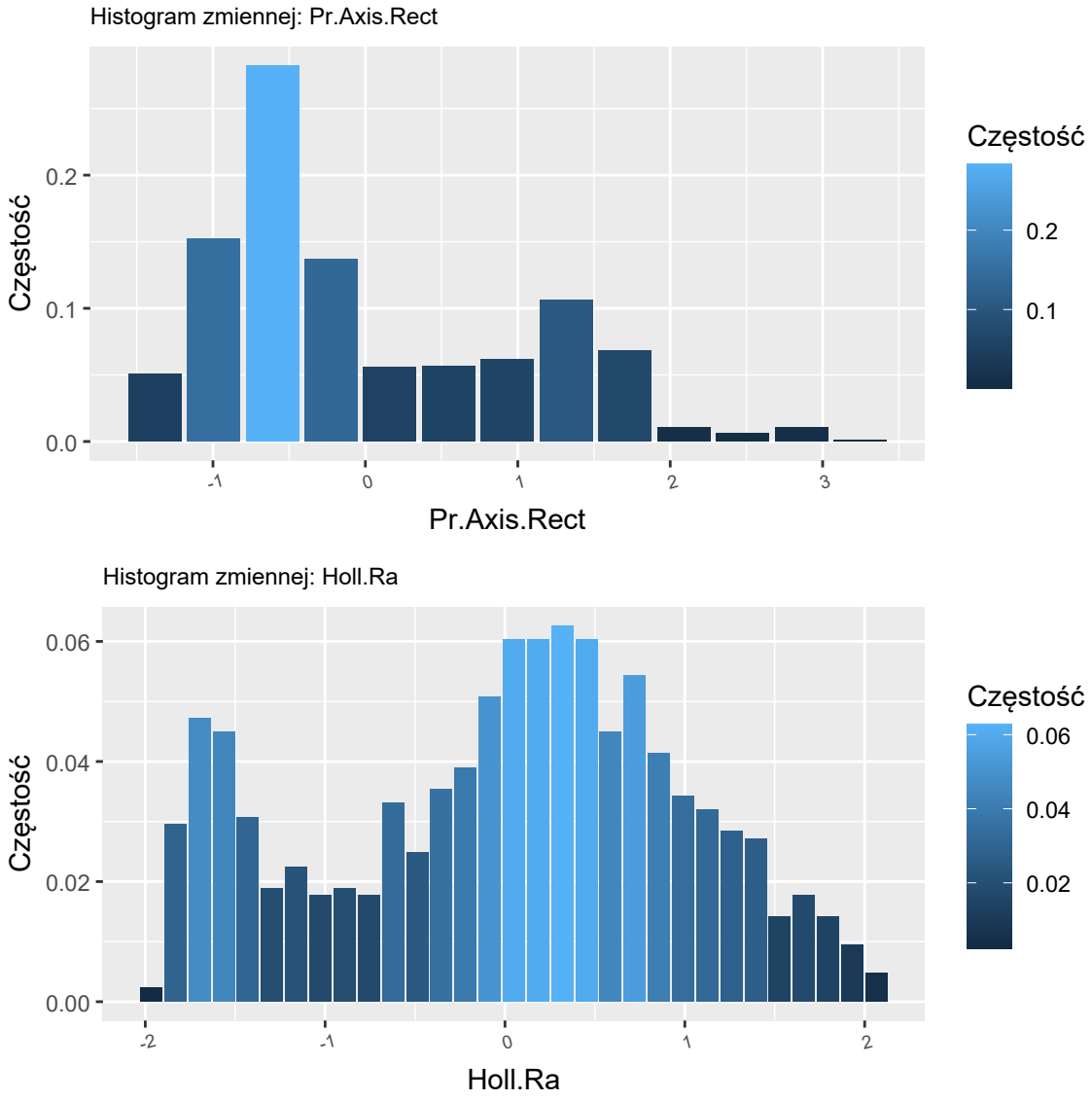
Po wcześniejszej analizie wszystkich zmiennych zdecydowałem, że zmienne o najlepszej zdolności dyskryminacyjnej to: `Max.L.Ra`, `Scat.Ra`, `Elong`, `Comp`, `Pr.Axis.Rect`, `Holl.Ra`.



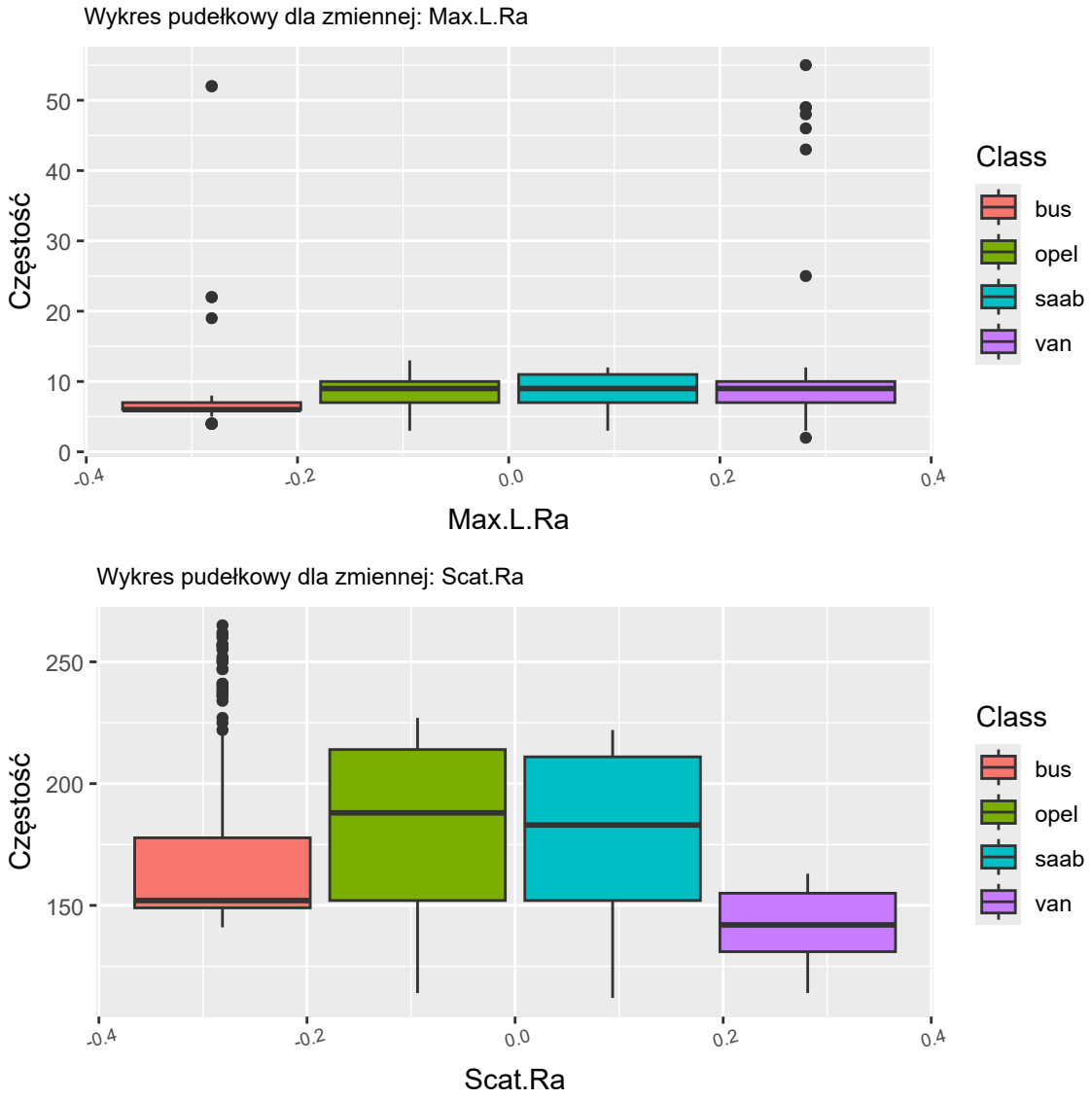
Rysunek 4: Histogramy zmiennych o najlepszej zdolności dyskryminacyjnej



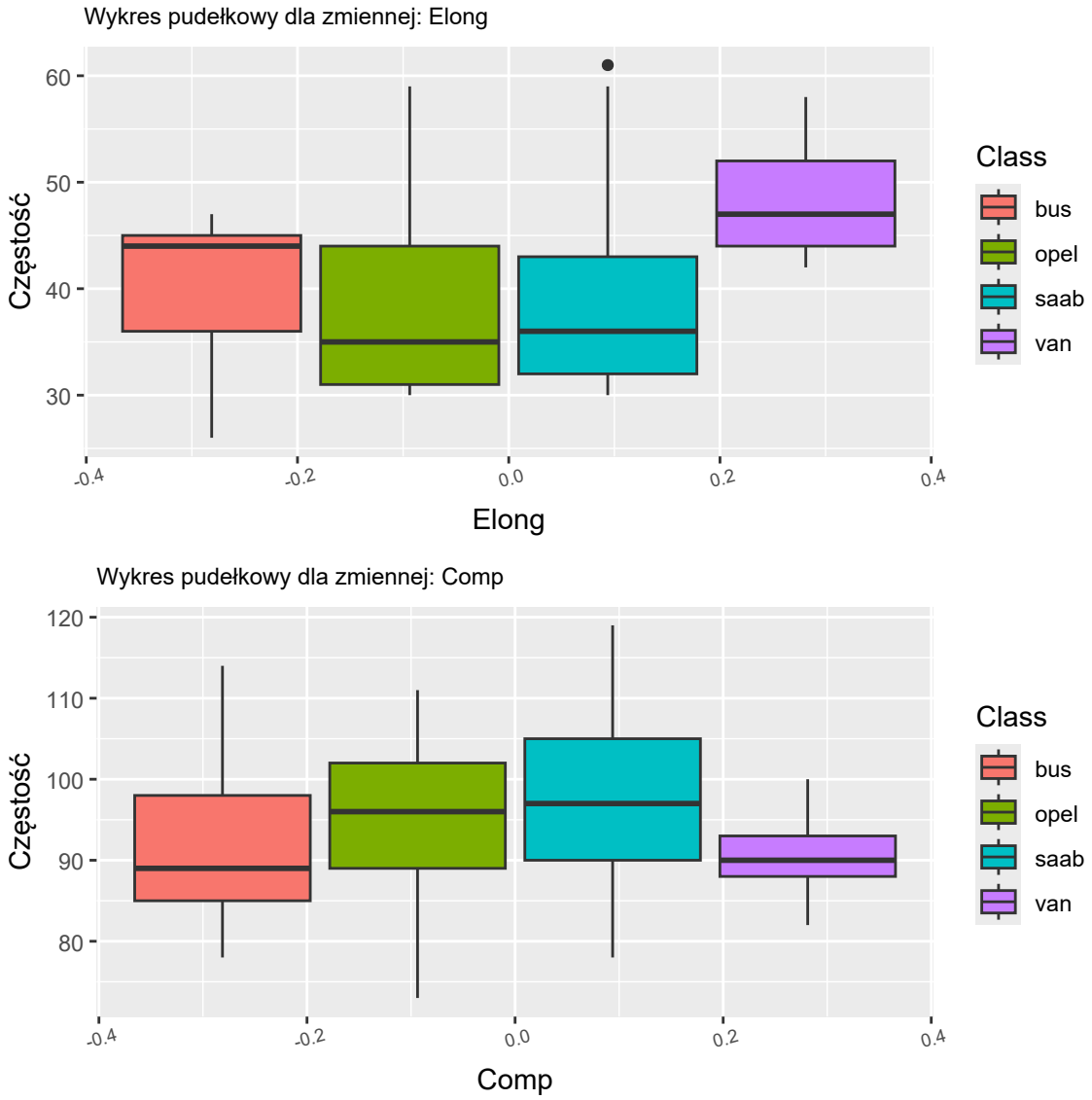
Rysunek 5: Histogramy zmiennych o najlepszej zdolności dyskryminacyjnej



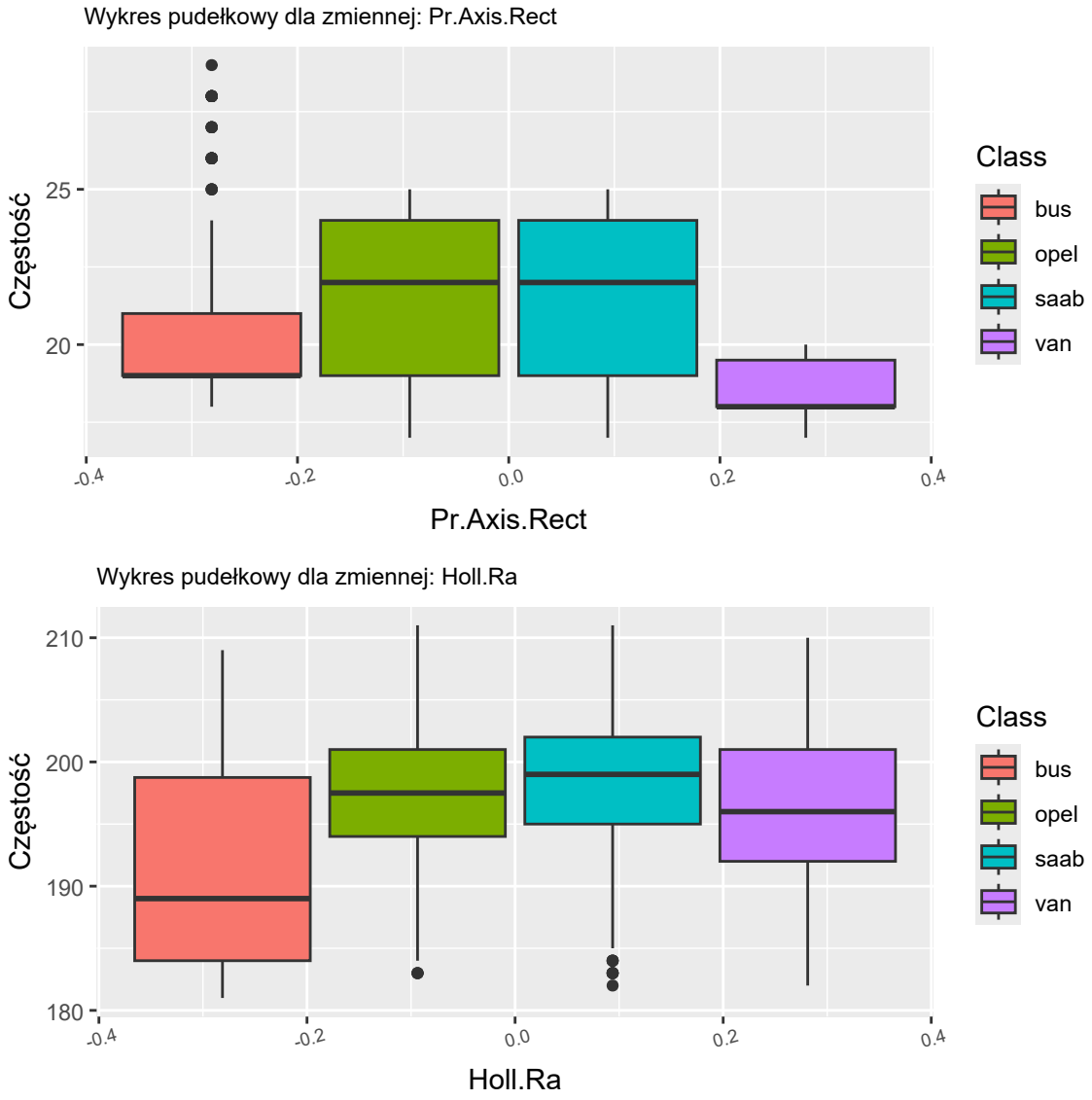
Rysunek 6: Histogramy zmiennych o najlepszej zdolności dyskryminacyjnej



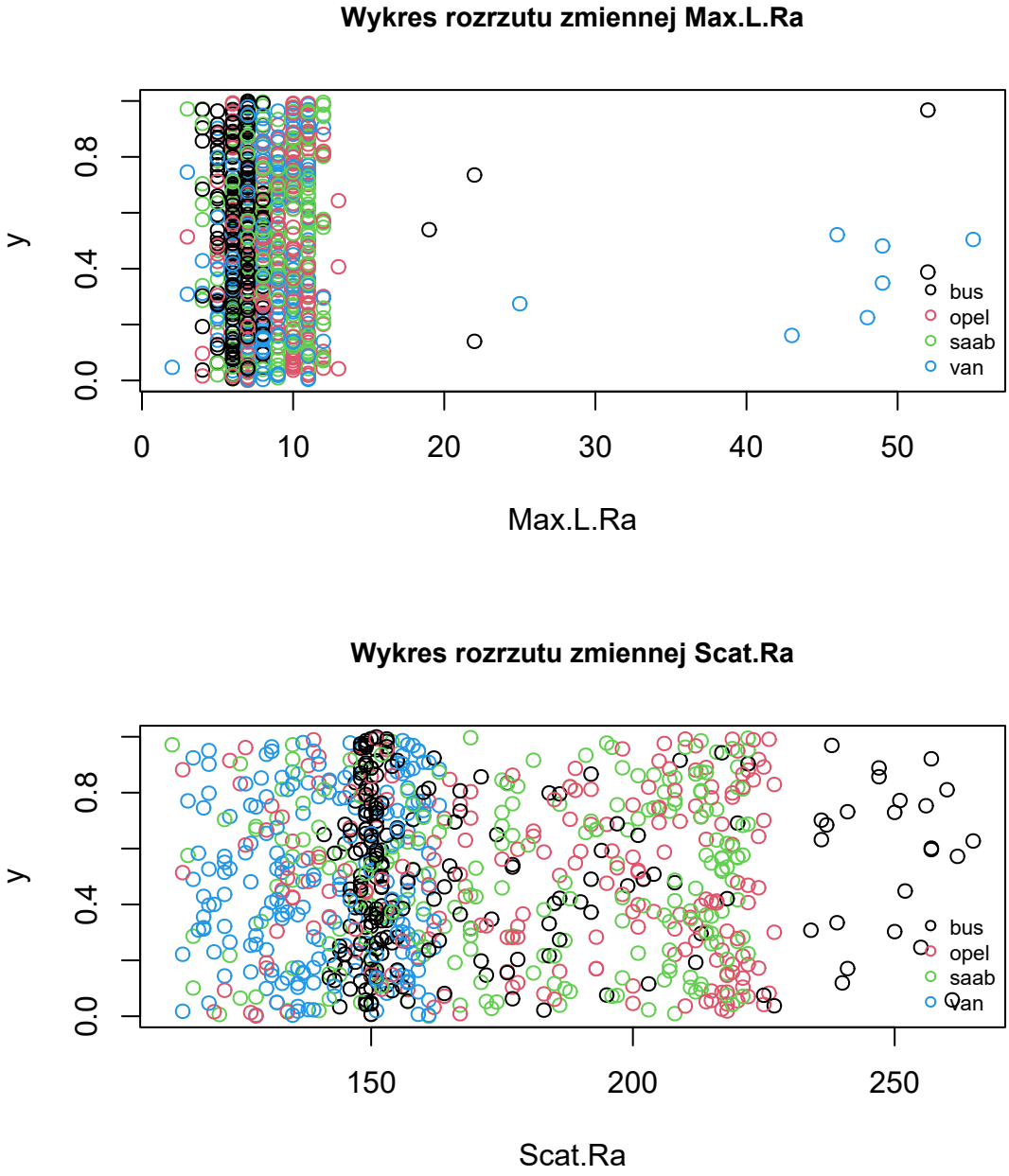
Rysunek 7: Wykresy pudełkowe zmiennych o najlepszej zdolności dyskryminacyjnej



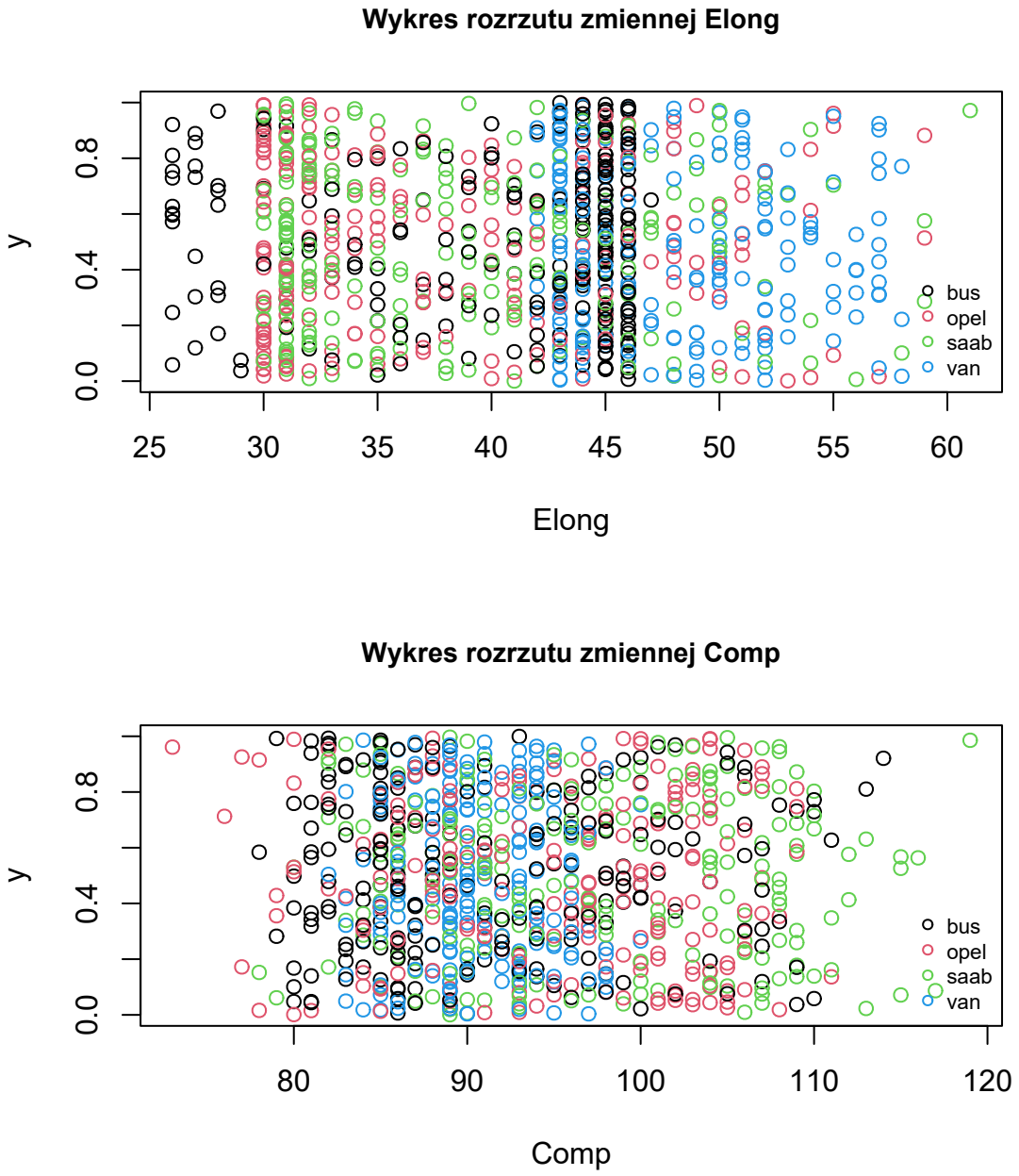
Rysunek 8: Wykresy pudełkowe zmiennych o najlepszej zdolności dyskryminacyjnej



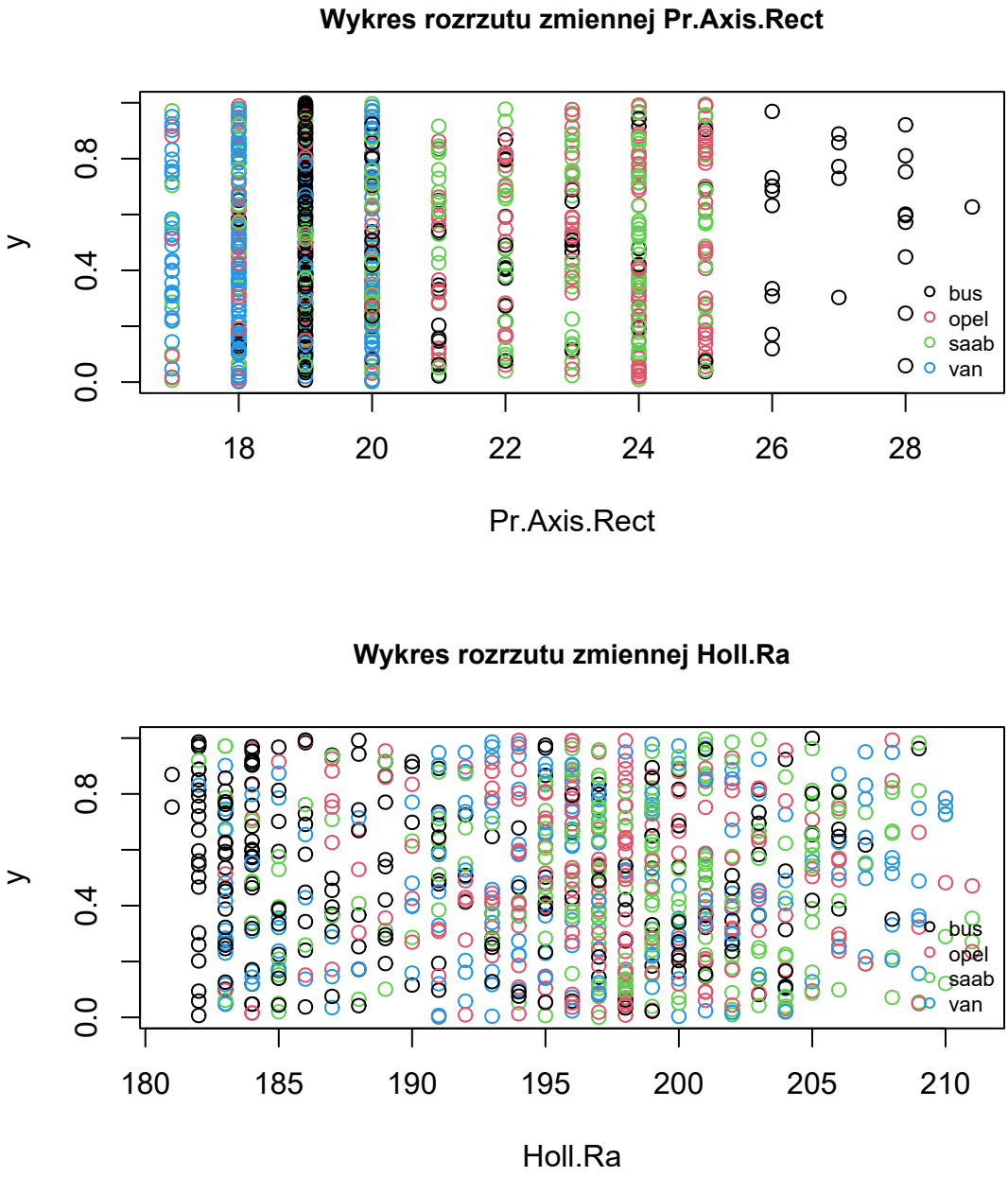
Rysunek 9: Wykresy pudełkowe zmiennych o najlepszej zdolności dyskryminacyjnej



Rysunek 10: Wykresy rorzutu zmiennych o najlepszej zdolności dyskryminacyjnej



Rysunek 11: Wykresy rorzutu zmiennych o najlepszej zdolności dyskryminacyjnej



Rysunek 12: Wykresy rorzutu zmiennych o najlepszej zdolności dyskryminacyjnej

Z rysunków 4-12 widać, że klasyfikacja w tym zbiorze danych nie będzie najłatwiejszym zadaniem. O ile samochody typu `bus` i `van` jeszcze jakoś się wyróżniają, tak pojazdy z kategorii `opel` i `saab` mają bardzo podobne rozkłady, co zdecydowanie nie pomoże w klasyfikacji.

2.3 Stosowanie algorytmów

Do porównania klasyfikacji zostaną użyte 3 zbiory:

- Wszystkie cechy
- Cechy o najlepszej zdolności dyskryminacyjnej
- Kombinacja cech o najlepszej zdolności dyskryminacyjnej, dla której naiwny klasyfikator bayesowski zwraca najmniejszy błąd

2.3.1 Naiwny klasyfikator bayesowski

Zacznijmy od wyznaczenia kombinacji cech, dla której naiwny klasyfikator bayesowski zwraca najmniejszy (teoretycznie) błąd, uwzględniając schematy oceny dokładności cross-validation i bootstrap. Do wyliczenia błędów i tworzenia klasyfikatora została użyta funkcja `NaiveBayes` z pakietu `klaR`, która pozwala na zastosowanie jądrowej estymacji gęstości, co zmniejsza (przynajmniej w tym przypadku) potencjalny błąd.

Tabela 7: Błędy z metody cross validation (5-krotna walidacja krzyżowa)

	Liczba_cech	Cechy	Błąd_CV
20	3	Max.L.Ra, Elong, Comp	0.3569740
57	6	Max.L.Ra, Scat.Ra, Elong, Comp, Pr.Axis.Rect, Holl.Ra	0.3640662
22	3	Max.L.Ra, Elong, Holl.Ra	0.3676123
17	3	Max.L.Ra, Scat.Ra, Comp	0.3699764
43	4	Max.L.Ra, Elong, Comp, Holl.Ra	0.3723404

Tabela 8: Błędy z metody Bootstrap (5 losowań bootstrapowych)

	Liczba_cech	Cechy	Błąd_Boot
20	3	Max.L.Ra, Elong, Comp	0.3697781
43	4	Max.L.Ra, Elong, Comp, Holl.Ra	0.3764143
17	3	Max.L.Ra, Scat.Ra, Comp	0.3790252
22	3	Max.L.Ra, Elong, Holl.Ra	0.3813316
23	3	Max.L.Ra, Comp, Pr.Axis.Rect	0.3873803

Jak widać z tabeli 7 i 8, najmniejszy błąd zarówno w metodzie cross-validation i schemacie bootstrap uzyskujemy dla kombinacji `Max.L.Ra, Elong, Comp`. Będzie to zatem trzeci zbiór, który uwzględnimy w porównaniu jakości algorytmów klasyfikacji.

Przechodząc do klasyfikacji metodą naiwnego klasyfikatora bayesowskiego: zbiór uczący składa się z 564 wierszy (2/3 całości), a zbiór testowy z 282 wierszy.

Tabela 9: Porównanie błędów w klasyfikacji metodą naiwnego klasyfikatora bayesowskiego

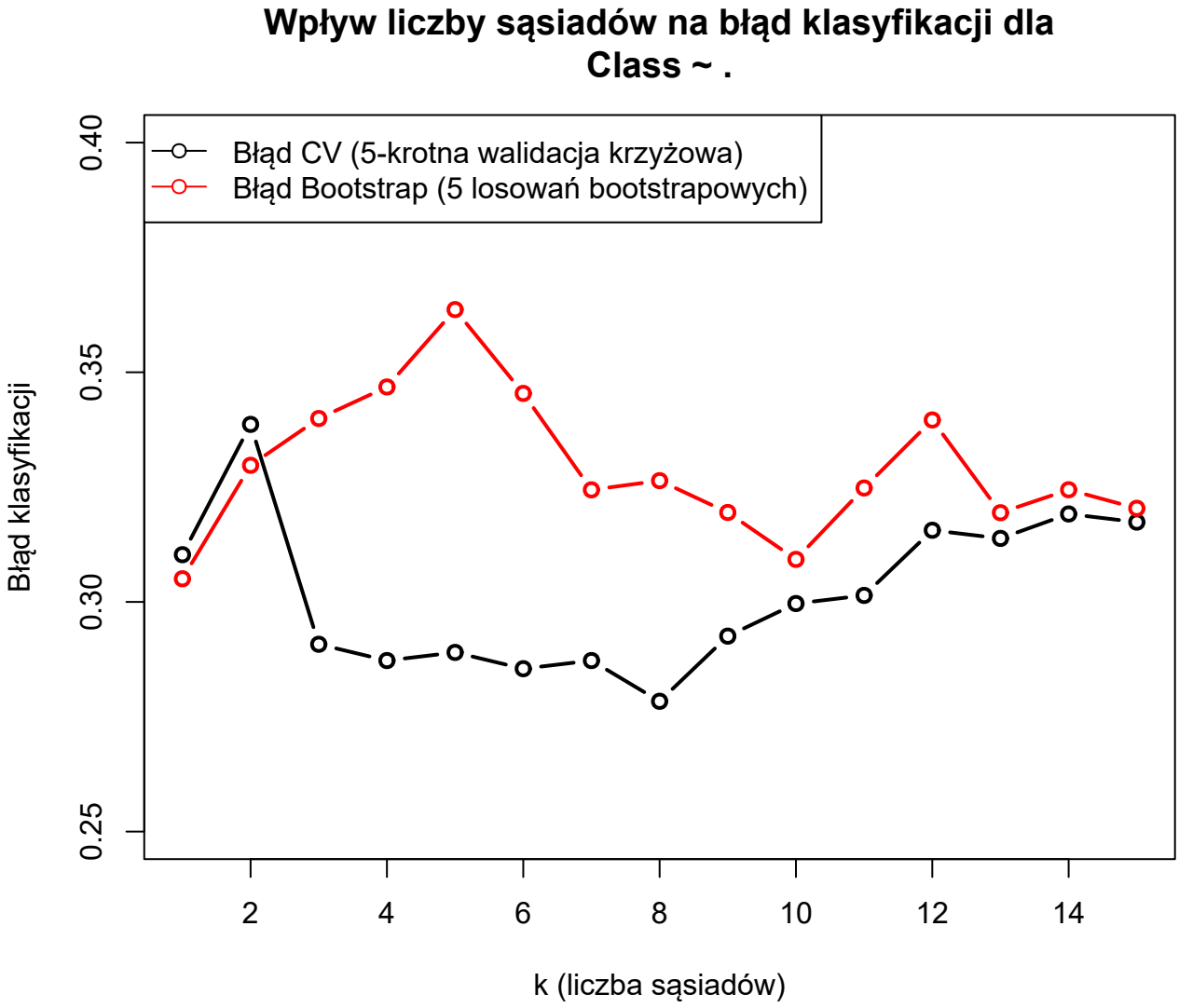
Zmienne	Zbiór	Błąd
Wszystkie	Uczący	0.3475177
Wszystkie	Testowy	0.4007092
Max.L.Ra+Scat.Ra+Elong+Comp+Pr.Axis.Rect+Holl.Ra	Uczący	0.3581560
Max.L.Ra+Scat.Ra+Elong+Comp+Pr.Axis.Rect+Holl.Ra	Testowy	0.4042553
Max.L.Ra+Elong+Comp	Uczący	0.3457447
Max.L.Ra+Elong+Comp	Testowy	0.3617021

Tabela 10: Macierz pomyłek dla zbioru testowego w zależności od cech "Max.L.Ra+Elong+Comp"

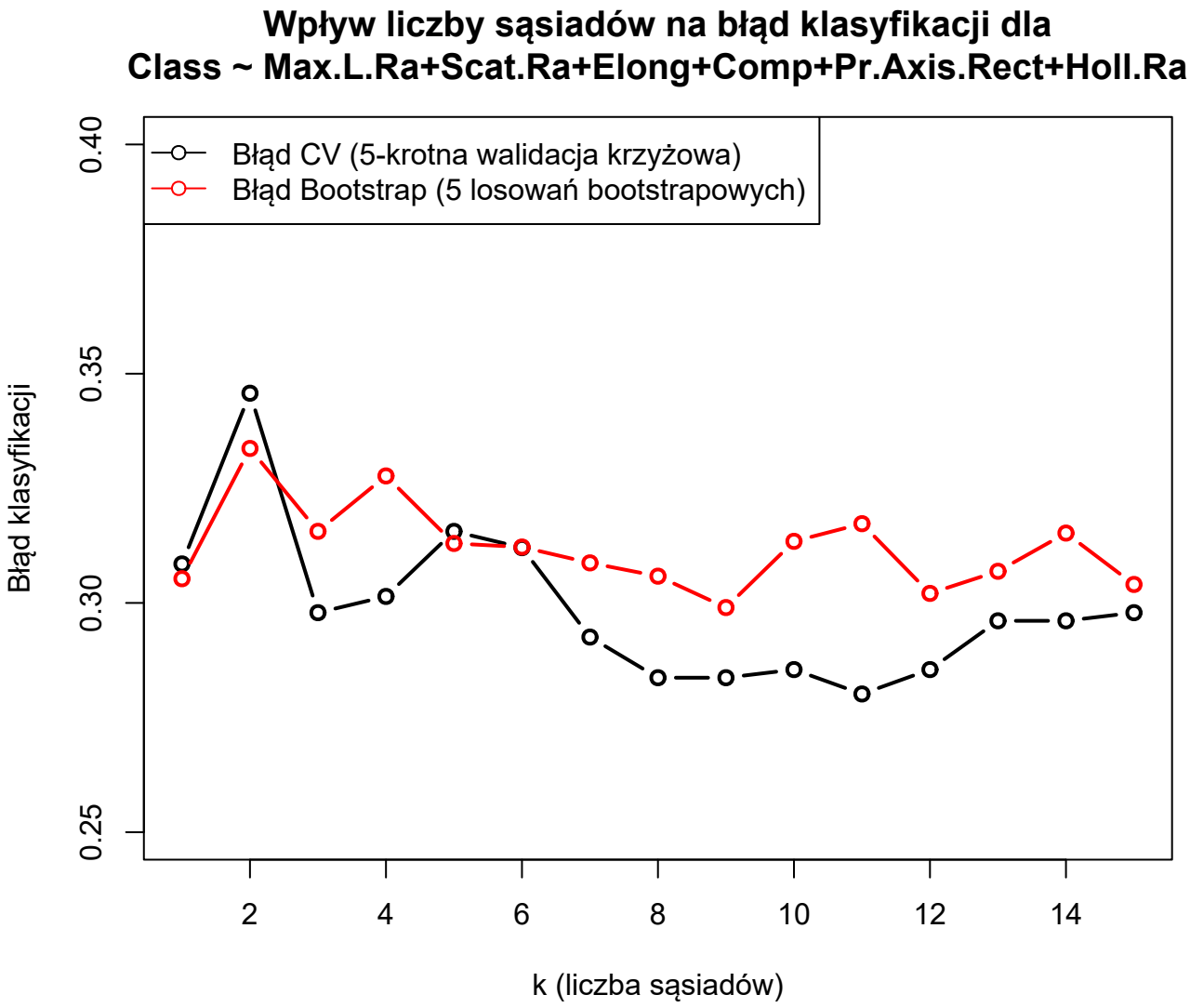
rzecz_etyk	prog_etyk_test			
	bus	opel	saab	van
bus	60	20	14	2
opel	4	39	34	1
saab	3	5	19	0
van	1	6	12	62

Tabela 9 pokazuje, że najmniejszy błąd mamy dla zbioru uczącego przy rozważeniu cech `Max.L.Ra, Elong, Comp`, a dla zbioru testowego również najlepszy wynik uzyskujemy, korzystając z tych trzech cech. Warto zauważyć, że wszystkie błędy są większe niż 0.34. Macierz pomyłek przedstawiona w tabeli 10 potwierdza, że poprawne rozpoznanie samochodów z klasy `opel` i `saab` jest trudne.

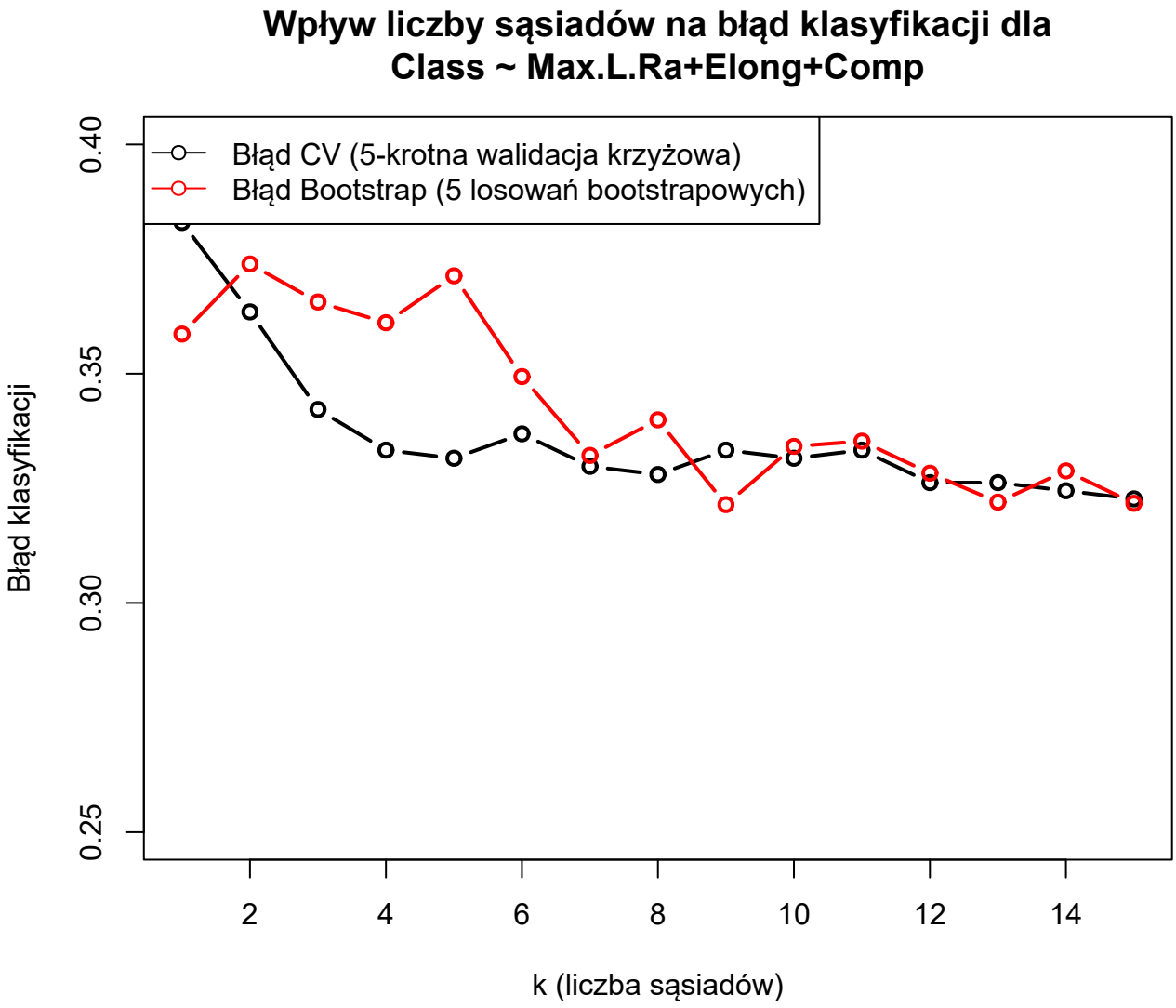
2.3.2 Algorytm k-najbliższych sąsiadów



Rysunek 13: Wpływ liczby sąsiadów na błąd klasyfikacji w zależności od wybranych cech dla algorytmu kNN



Rysunek 14: Wpływ liczby sąsiadów na błąd klasyfikacji w zależności od wybranych cech dla algorytmu kNN



Rysunek 15: Wpływ liczby sąsiadów na błąd klasyfikacji w zależności od wybranych cech dla algorytmu kNN

Na rysunkach 13-15 widzimy, jak zmienia się błąd klasyfikacji w zależności od liczby sąsiadów dla różnych schematów oceny dokładności. Z każdego wykresu weźmy punkt, dla którego cross-validation i bootstrap zwróciły najmniejsze błędy (łącznie 6 punktów).

Tabela 11: Porównanie błędów

Zmienne	Liczba_sąsiadów	Metoda	Błąd_rzeczywisty
Wszystkie	8	CV	0.3014184
Wszystkie	1	Bootstrap	0.3226950
Max.L.Ra+Scat.Ra+Elong+Comp+Pr.Axis.Rect+Holl.Ra+Max.L.Rect	11	CV	0.3085106
Max.L.Ra+Scat.Ra+Elong+Comp+Pr.Axis.Rect+Holl.Ra+Max.L.Rect	9	Bootstrap	0.3262411
Max.L.Ra+Elong+Comp	14	CV	0.2836879
Max.L.Ra+Elong+Comp	7	Bootstrap	0.2978723

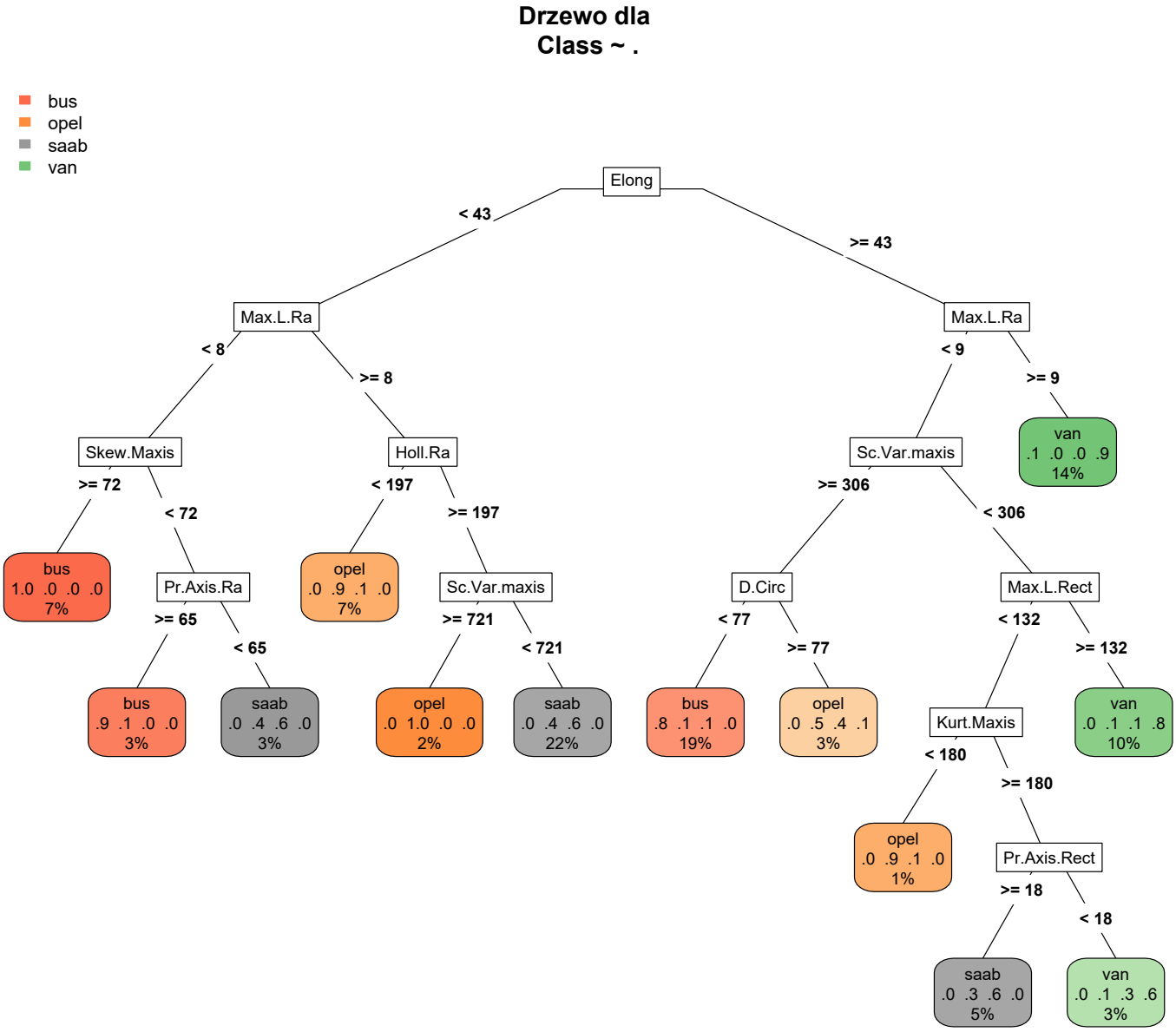
Tabela 12: Macierz pomyłek w zależności od cech 'Max.L.Ra+Elong+Comp' dla 14 sąsiadów

rzecz_etyk	prog_etyk_test			
	bus	opel	saab	van
bus	63	9	7	3
opel	2	28	16	2
saab	2	29	49	2
van	1	4	7	58

Z tabeli 11 można wyciągnąć ciekawe wnioski. Wszystkie zwrócone błędy są mniejsze niż w przypadku drzew klasyfikacyjnych. Najlepszy błąd uzyskujemy zawsze, gdy kierujemy się najmniejszym błędem metody cross-validation. Najmniejszy błąd zwraca metoda stosująca jedynie zmienne `Max.L.Ra+Elong+Comp`. Podobnie jak w przypadku drzew, macierz pomyłek z tabeli 12 pokazuje problemy z rozpoznaniem marki `opel` i `saab`.

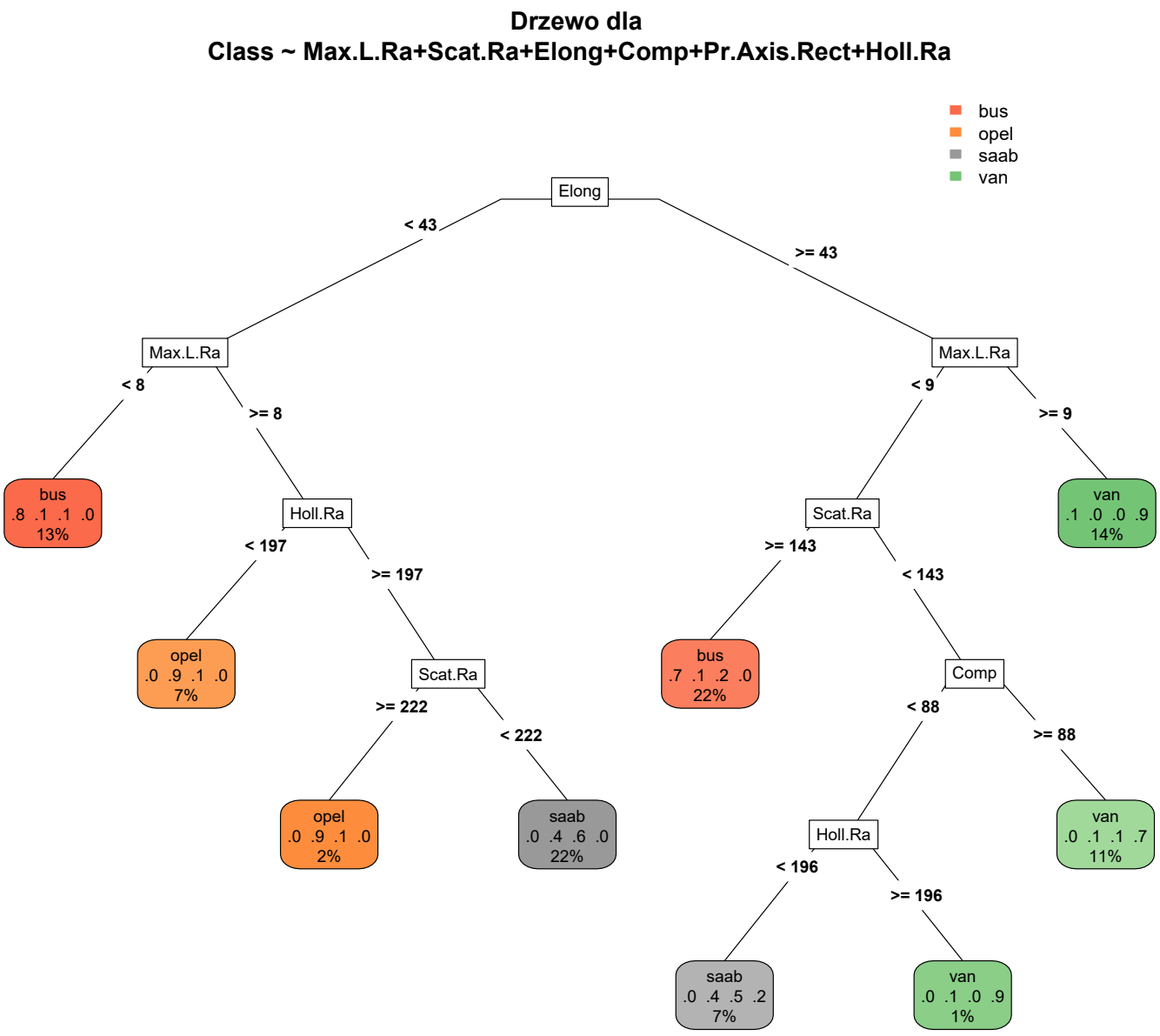
Błąd 0.2836879 jest póki co najlepszym błędem, jaki do tej pory otrzymaliśmy. Czy drzewa klasyfikacyjne przebiją ten wynik?

2.3.3 Drzewa klasyfikacyjne



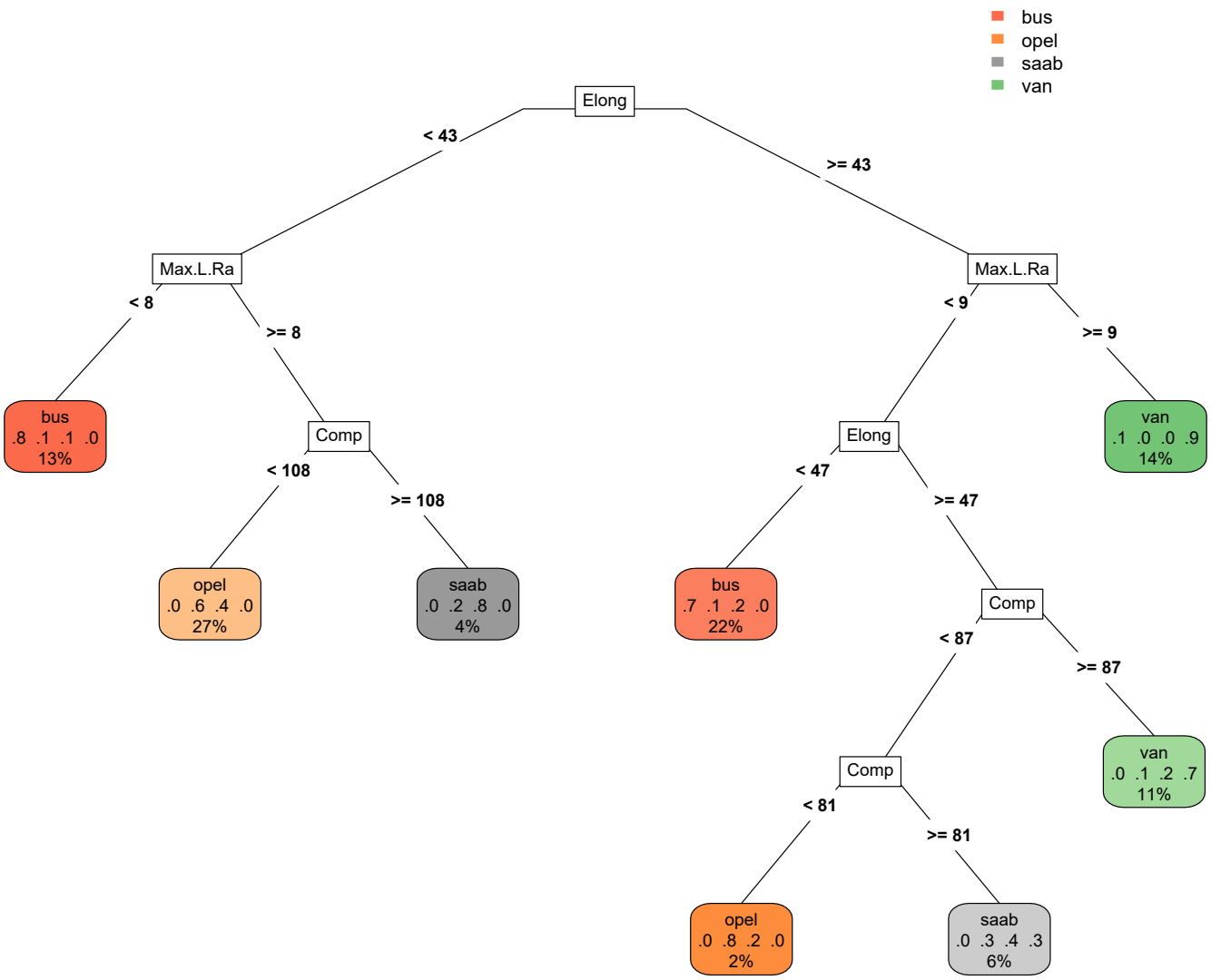
Rysunek 16: Drzewa z parametrem cp = .01, maxdepth = 10, minsplit = 0

```
## [1] "Błąd na learning set: 0.221631 Błąd na testing set: 0.297872"
```



Rysunek 17: Drzewa z parametrem cp = .01, maxdepth = 10, minsplit = 0

```
## [1] "Błąd na learning set: 0.281915 Błąd na testing set: 0.351064"
```

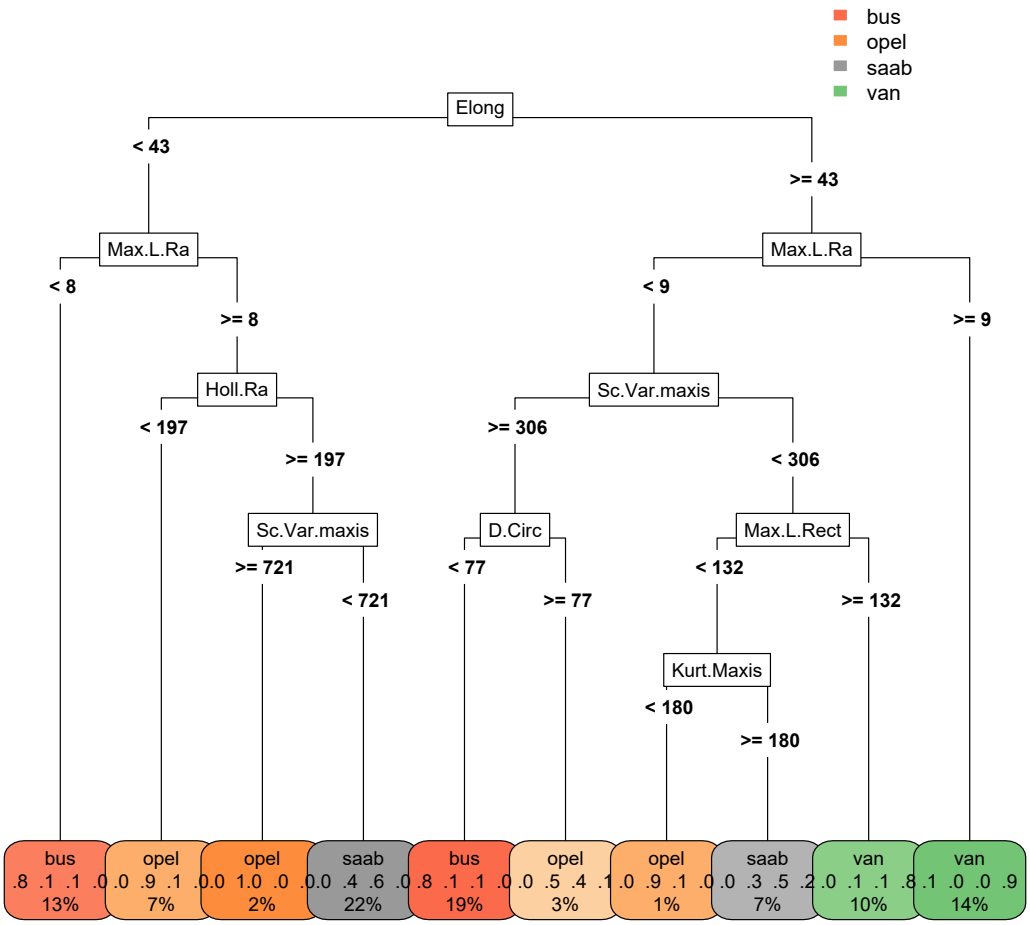


Rysunek 18: Drzewa z parametrem cp = .01, maxdepth = 10, minsplit = 0

```
## [1] "Błąd na learning set: 0.306738 Błąd na testing set: 0.361702"
```

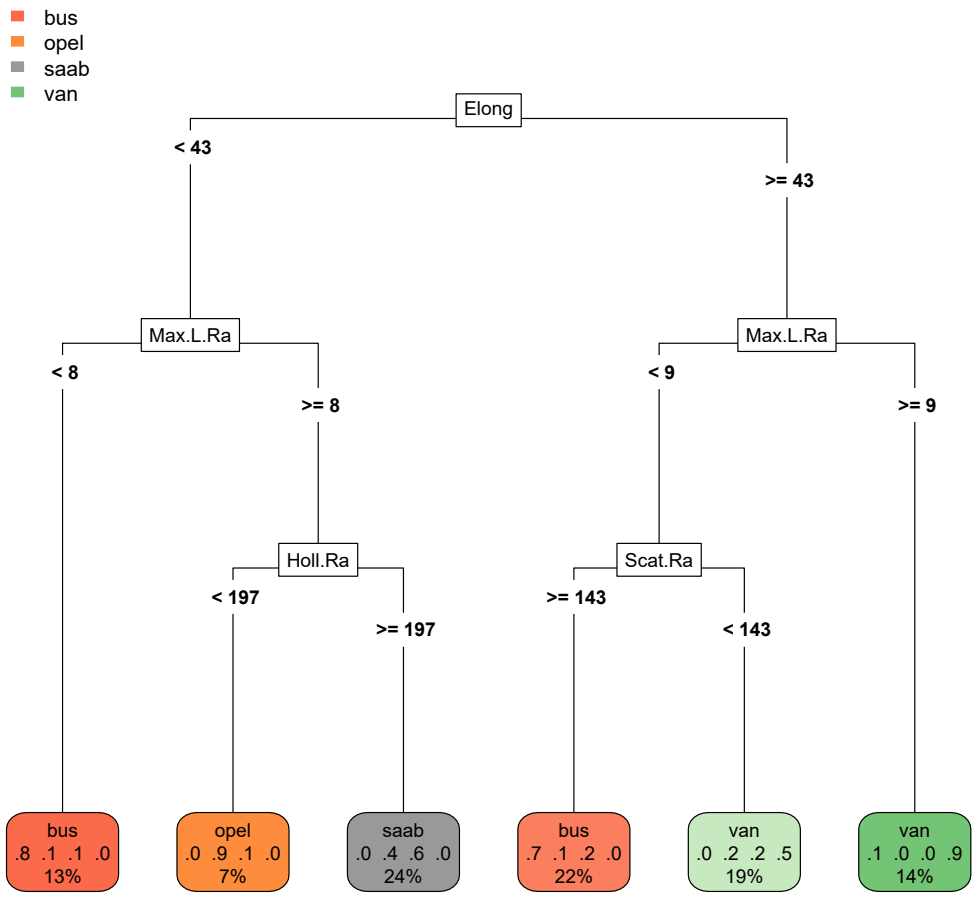
Na rysunkach 16-18 widać, że struktura drzew różni się. Mają one różną liczbę liści i maksymalną wysokość drzewa. Wszystkie mają podobne początkowe podziały, czyli ze względu na zmienną `Elong` i `Max.L.Ra`. Najmniejszy błąd, zarówno uczący jak i testowy, jest przy uwzględnieniu wszystkich cech w klasyfikacji.

Spójrzy teraz jak wyglądają drzewa po przycięciu.



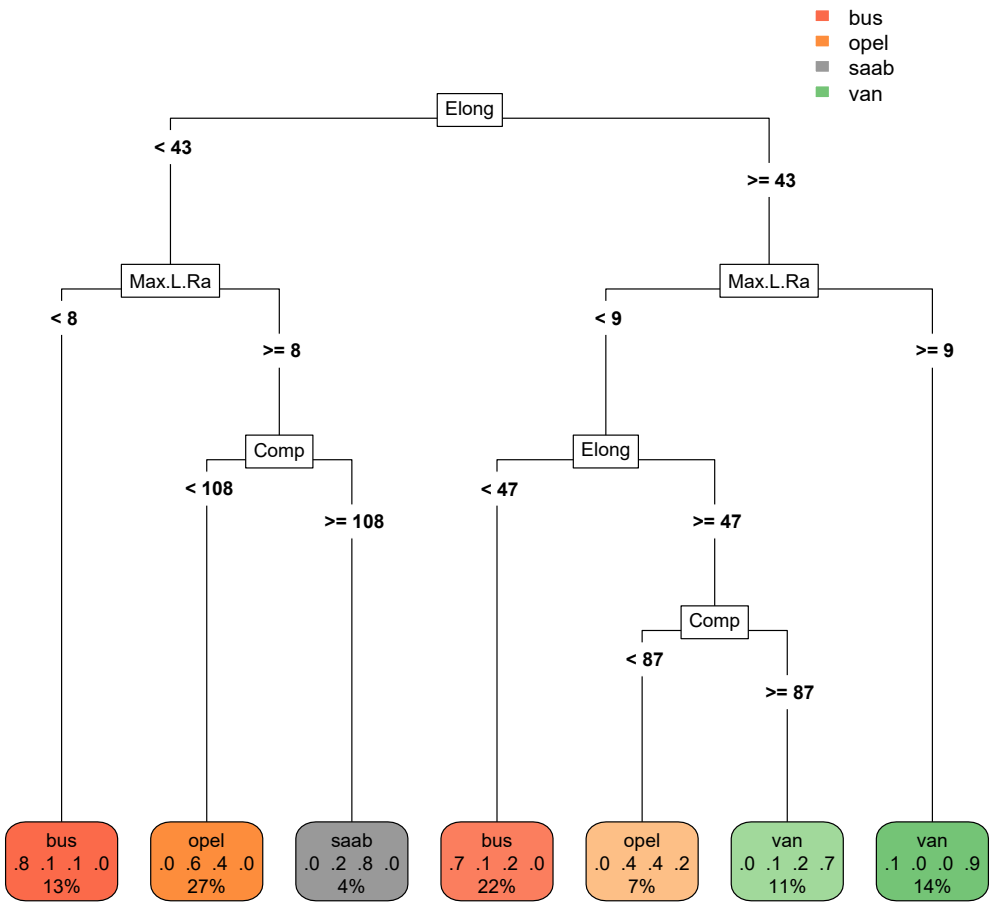
Rysunek 19: Przycięte drzewa z parametrem cp = .01, maxdepth = 10, minsplit = 0

```
## [1] "Błąd na learning set: 0.248227 Błąd na testing set: 0.326241"
```



Rysunek 20: Przycięte drzewa z parametrem cp = .01, maxdepth = 10, minsplit = 0

```
## [1] "Błąd na learning set: 0.315603 Błąd na testing set: 0.336879"
```



Rysunek 21: Przycięte drzewa z parametrem cp = .01, maxdepth = 10, minsplit = 0

```
## [1] "Błąd na learning set: 0.315603 Błąd na testing set: 0.368794"
```

Drzewa z rysunków 19-21 są już mniej skomplikowane. W większości przypadków wzrosły błędy zarówno na zbiorze uczącym, jak i testowym, więc raczej nie jest to najlepszy rozwiązanie w tym przypadku.

Zmieńmy teraz trochę parametry i spójrzmy na błędy drzew o maksymalnej wysokości drzewa równej 5 i minimalnej liczbie elementów w liściu równej 10.

```
## [1] "Błąd na learning set: 0.230496 Błąd na testing set: 0.315603"
```

```
## [1] "Błąd na learning set: 0.281915 Błąd na testing set: 0.351064"
```

```
## [1] "Błąd na learning set: 0.306738 Błąd na testing set: 0.361702"
```

Błędy za bardzo się nie zmieniły względem oryginalnego drzewa. Możemy to jeszcze poprawić, ustalając `cp` na .005.

```
## [1] "Błąd na learning set: 0.207447 Błąd na testing set: 0.322695"
```

```
## [1] "Błąd na learning set: 0.264184 Błąd na testing set: 0.35461"
```

```
## [1] "Błąd na learning set: 0.306738 Błąd na testing set: 0.361702"
```

Poraz kolejny zmalały błęd na zbiorach uczących, ale teraz mamy wzrost dla zbiorów testowych. Warto

przypomnieć, że zbiór uczący ma 564 wiersze, a drzewa klasyfikujące są bardzo podatne na przeuczenie. Zmniejszmy liczbę wierszy w zbiorze uczącym do 390.

```
## [1] "Błąd na learning set: 0.205128 Błąd na testing set: 0.307018"
```

```
## [1] "Błąd na learning set: 0.261538 Błąd na testing set: 0.313596"
```

```
## [1] "Błąd na learning set: 0.282051 Błąd na testing set: 0.29386"
```

Tabela 13: Macierz pomyłek dla zbioru testowego w zależności od cech 'Max.L.Ra+Elong+Comp'

rzecz_etyk	prog_etyk_test			
	bus	opel	saab	van
bus	112	18	22	2
opel	3	67	43	0
saab	0	13	39	0
van	2	13	18	104

We wszystkich przypadkach spadł błąd na zbiorze testowym, co można uznać za ulepszenie metody. Najlepiej sprawi się drzewo klasyfikacyjne dla zmiennych `Max.L.Ra, Elong, Comp` ze względu na podobny błąd w zbiorze uczącym i testowym oraz najmniejszy błąd testowych ze wszystkich 3 zbiorów cech. Jest to również najlepiej zbudowane drzewo ze wszystkich zbudowanych klasyfikatorów tego typu. Z kolei macierz pomyłek w tabeli 13 bardzo dobrze poradziła sobie z samochodami typu `bus` i `van`, ale nadal występuje niejednoznaczność w klasyfikacji `opel` i `saab`.

2.4 Podsumowanie

Ze wszystkich klasyfikatorów, najgorzej poradził sobie naiwny klasyfikator bayesowski, który dla każdego ze zbioru cech zwracał błąd powyżej 0.34, bez względu czy był to zbiór uczący, czy testowy. Podobna sytuacja mogła przydarzyć się w przypadku metody drzew klasyfikacyjnych, lecz przy odpowiedniej zmianie parametrów i rozmiaru zbioru uczącego i testowego, udało się osiągnąć błąd poniżej 0.3 dla zbioru uczącego i testowego w przypadku użycia do klasyfikacji cech `Max.L.Ra, Elong, Comp`. Najlepiej sobie radził algorytm k-najbliższych sąsiadów, który przy użyciu wszystkich cech o najlepszej zdolności do dyskryminacji i ustawieniu liczby sąsiadów równej 14 zwrócił błąd 0.2853535. Udało się dotrzeć do tego wyniku dzięki schematowi oceny dokładności cross-validation. Podsumowując, najlepsza metoda klasyfikacyjna dla tego zbioru danych to metoda **k-najbliższych sąsiadów**, następnie są **drzewa klasyfikacyjne**, a najgorszy okazał się **naiwny klasyfikator bayesowski**.