Raport 3

Eksploracja danych

Olaf Masłowski, album 277543

2025-07-09

Spis treści

1 Wstęp

Ponieważ klasyfikacja danych Iris na bazie modelu regresji liniowej była wykonana w jednym z plików na eportalu - jak mniemam plik ten został udostępniony już po terminie oddania raportu - zdecydowałem się znaleźć podobne dane i wykonać zadanie dla nich. Dane seeds zawierają informacje o ziarnach trzech odmian pszenicy takie jak:

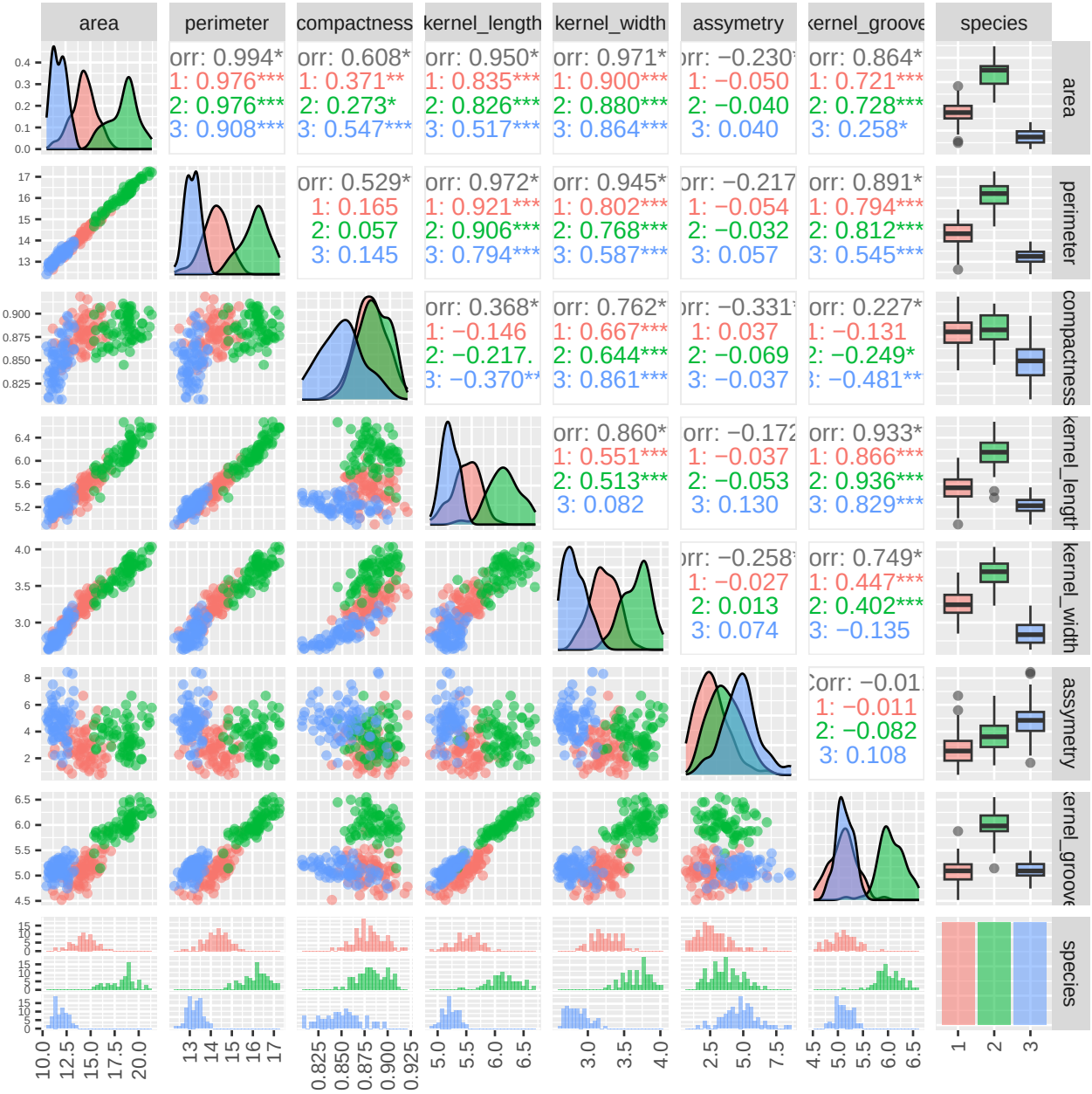
- Powierzhnia
- Obwód
- Ścisłość
- Długość ziarna
- Szerokość ziarna
- Współczynnik asymetrii
- Długość rowka ziarna (?)

Wszystkie cechy są ciągłe.

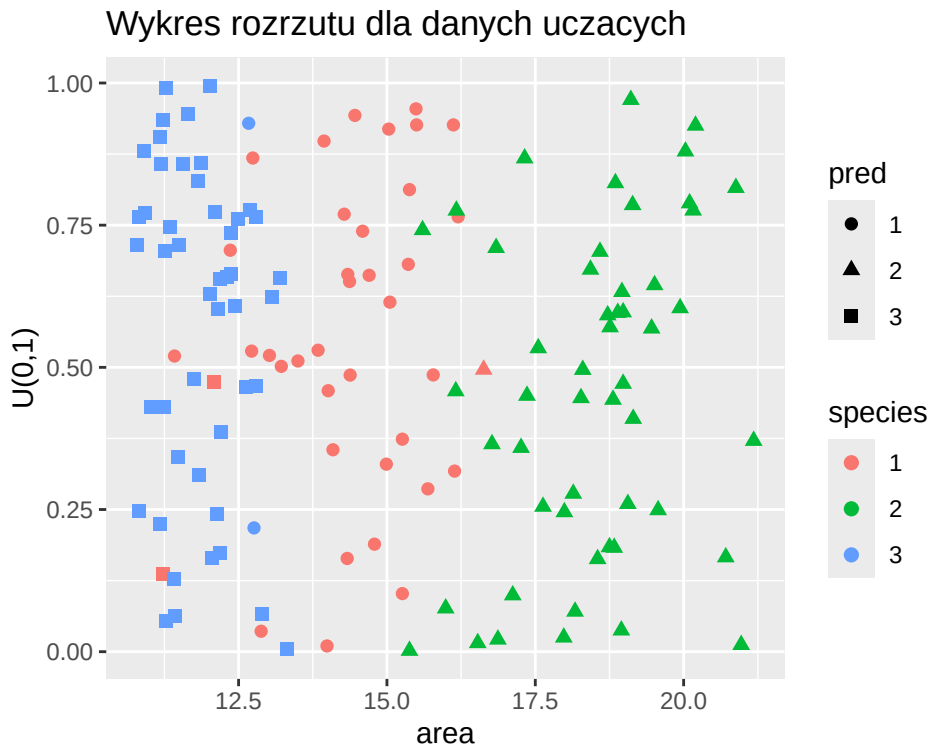
2 Klasyfikacja na bazie modelu regresji liniowej

Wykorzystamy regresję liniową do wyznaczenia "prawdopodobieństw", że dany przypadek należy do danej kategorii, a następnie dla każdego przypadku wybierzemy tę kategorię, którą przyjmuje z największym prawdopodobieństwem.

2.1 Model z podstawowymi czynnikami



Rysunek 1: Zestawienie istotnych wykresów



Rysunek 2: wykres rozrzutu U(0,1) area z podziałem na kategorie. pred - klasyfikacja przez model. species - prawdziwa etykietka

	1	2	3
1	36	0	2
2	1	51	0
3	2	0	47

Tabela 1: Macierz pomyłek na zbiorze uczącym

	1	2	3
1	31	0	0
2	0	19	0
3	0	0	21

Tabela 2: Macierz pomyłek na zbiorze testowym

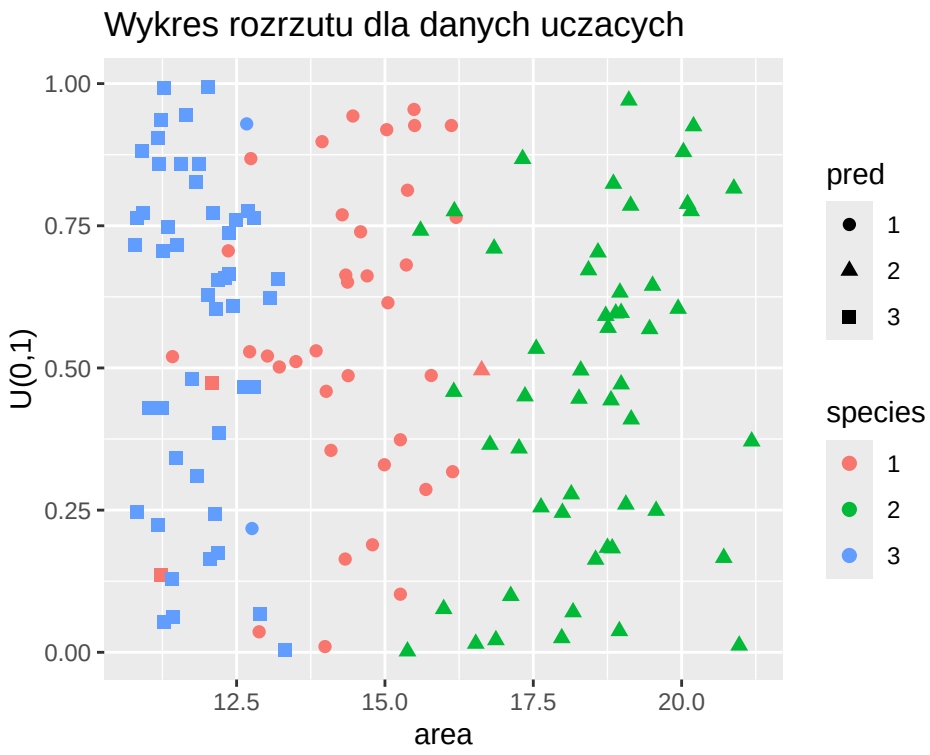
Wobec wykresów (rys. 1) zdolnością dyskryminacyjną wyróżnia się zmienna area. Błąd klasyfikacji wyniósł 0.0359712 na zbiorze uczącym i 0 na zbiorze testowym. Zjawisko maskowania klas nie wystąpiło w istotnym stopniu - klasy mają podobną liczność i dostatecznie się od siebie różnią.

2.2 Model z czynnikami wielomianowymi drugiego stopnia

Do utworzenia ramki danych z czynnikami wielomianowymi napisałem poniższą funkcję:

```
# Z wielomianami

wielomian <- function(df) {
  cechy <- dim(df)[[2]]
  df2 <- df
  n=1
  for (x in 1:cechy) {
    for (y in x:cechy) {
      W <- df[,x] * df[,y]
      df2 <- cbind(df2,W)
      names(df2)[cechy + n] <- paste0("w",as.character(n))
      n = n + 1
    }
  }
  return(df2)
}
```



	1	2	3
1	37	0	0
2	0	51	0
3	2	0	49

Tabela 3: Macierz pomyłek na zbiorze uczącym

	1	2	3
1	29	1	0
2	1	18	0
3	1	0	21

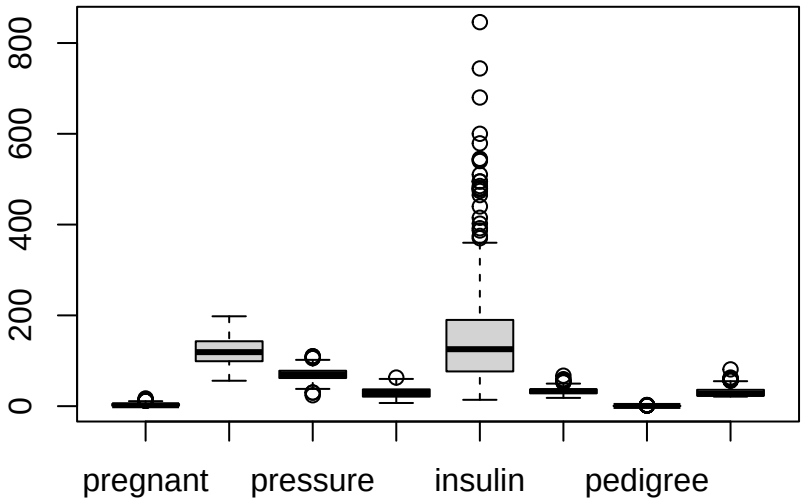
Tabela 4: Macierz pomyłek na zbiorze testowym

Błąd klasyfikacji na zbiorze uczącym wyniósł 0.014, na zbiorze testowym 0.042. Moje przypuszczenia dotyczące dokładności powyższych algorytmów są następujące: ze względu na większą ilość zmiennych objaśniających o dobrych zdolnościach dykryminacyjnych (w porównaniu z danymi Iris), standardowa regresja była dostatecznie dobra i czynniki wielomianowe wprowadziły jedynie dodatkowy szum objawiający się przede wszystkim na zbiorze testowym.

3 Zadanie 2 - porównanie metod klasyfikacji

Zastosujemy poznane metody klasyfikacji - k-najbliższych sąsiadów, drzewa klasyfikacyjne i naiwny klasyfikator bayesowski - do danych PimaIndiansDiabetes (Nie zwróciłem uwagi na "2", ale według mej najlepszej wiedzy i tak doprowadziłem dane do stanu jak w "drugiej edycji").

3.1 Zapoznanie się z danymi i wstępna analiza danych



Rysunek 3: wykresy pudełkowe zmiennych ilościowych

Rysunek 4: Zestawienie istotnych wykresów dla PimaIndianDiabetes

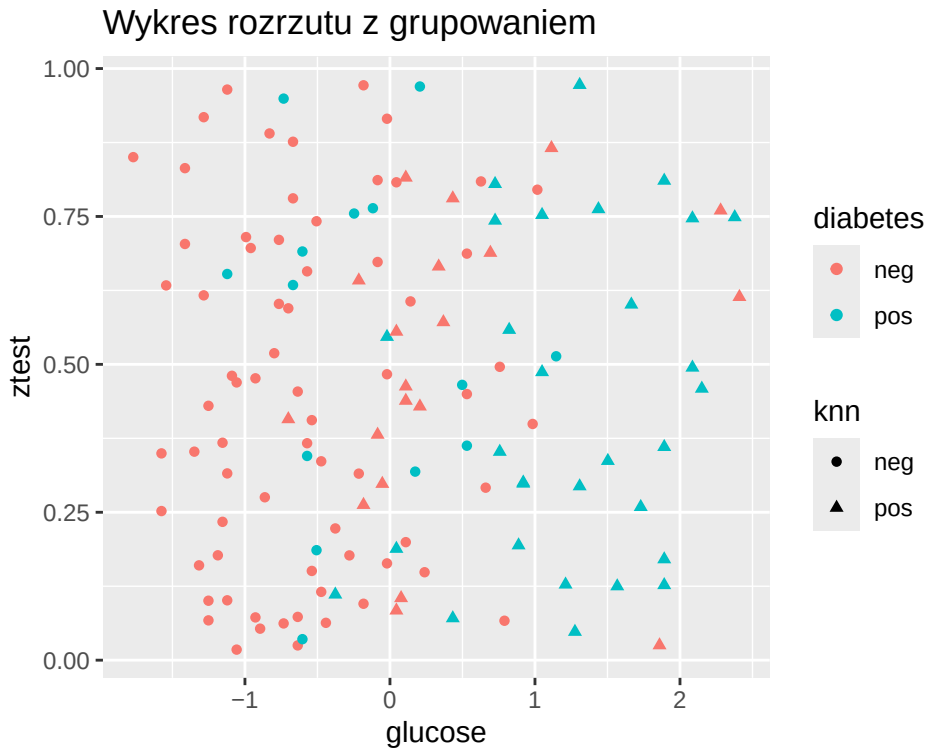
Dane zawierają następujące informacje:

- Ilość przebytych ciąż
- Poziom glukozy
- Ciśnienie krwi
- Grubość skóry na tricepsie
- Poziom insuliny
- BMI
- DPF (diabetes pedigree function)

- Wiek
- Cukrzyca (wynik testu)

Przypisanie wszystkich przypadków do największyej klasy poskutkowałoby błędem 0.332. Wykresy pudełkowe (rys. 3) sugerują konieczność przeprowadzenia standaryzacji. Najlepszą zdolność dyskryminacyjną zdają się mieć cechy *glucose* i *age* (rys. 4).

3.2 Metoda k-najbliższych sąsiadów



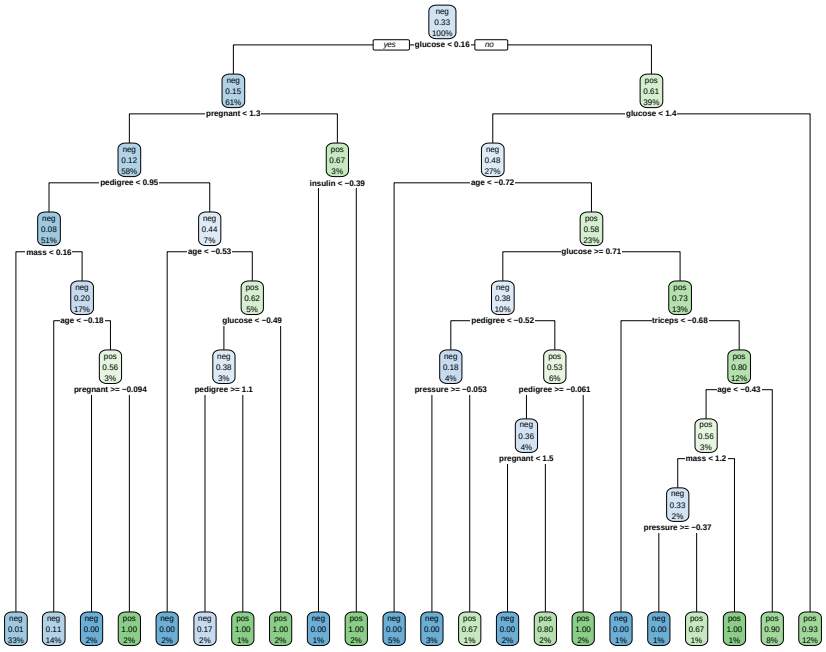
Rysunek 5: Wykres rozrzutu dla k=15, zmiennych glucose i age. Grupowanie wg rzeczywistej etykietki i klasyfikacji k-nn

	parametry	bledy
1	train, k=1, all	0.000
2	train, k=5, all	0.162
3	train, k=15, all	0.201
4	train, k=30, all	0.224
5	test, k=1, all	0.271
6	test, k=5, all	0.271
7	test, k=15, all	0.263
8	test, k=30, all	0.218
9	test, k=15, g+a	0.256
10	test, k=15, g + a + i	0.218
11	crossvalidation, k=15,g+a	0.232
12	crossvalidation, k=15, all	0.230

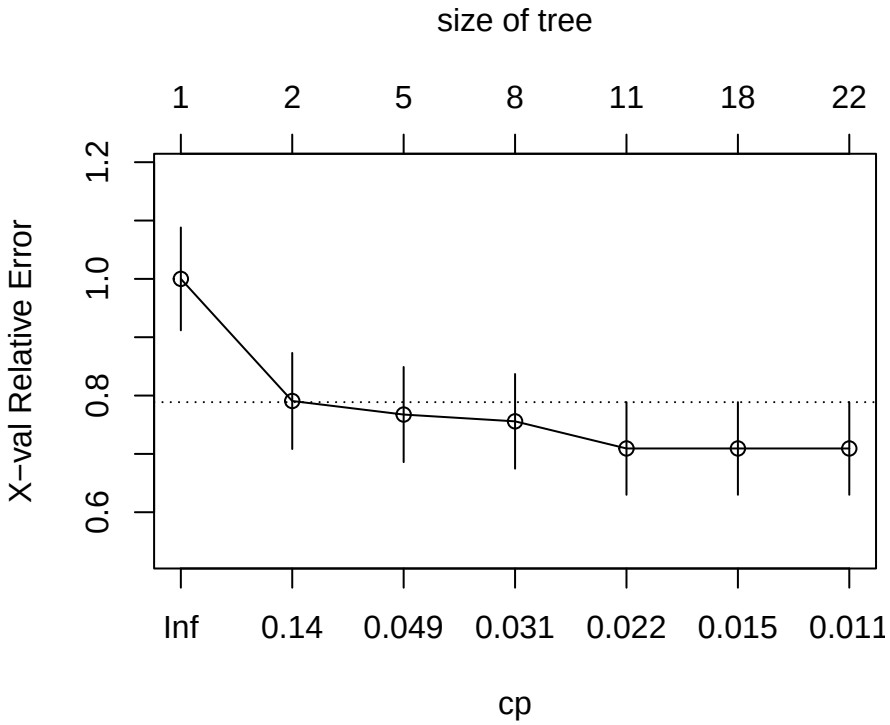
Tabela 5: Zestawienie błądów klasyfikacji. W pierwszej kolumnie informacje zakodowane: test/train - zbiór testowy/uczący, k - parametr algorytmu, all - wykorzystano wszystkie cechy, g - glucose, a - age, i - insulin

Ze względu na próby dla wielu parametrów, macierze pomyłek nie są uwzględnione w raporcie - można je znaleźć w pliku .Rmd. Błędy dla małych k są zauważalnie niższe na zbiorze testowym, zatem algorytm może w takich wypadkach być podatny na przeuczenie.

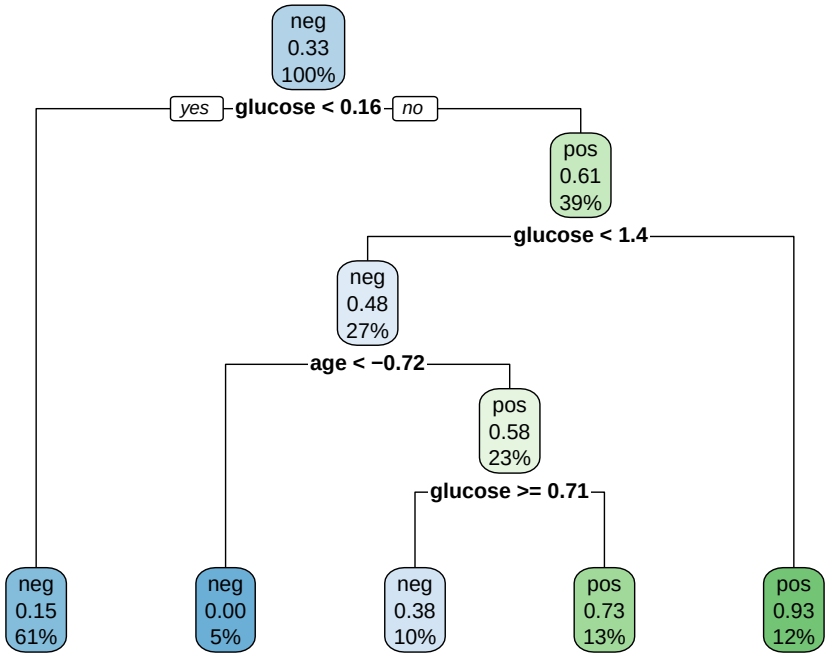
3.3 Drzewa klasyfikacyjne



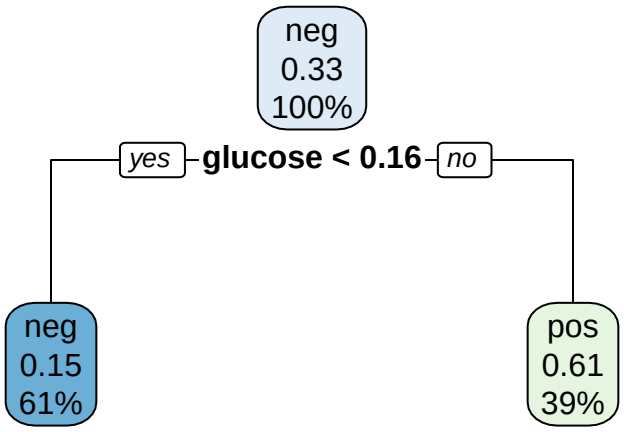
Rysunek 6: Drzewo klasyfikacyjne dla cp = .01 i wszystkich zmiennych. Zachęcam do przybliżenia



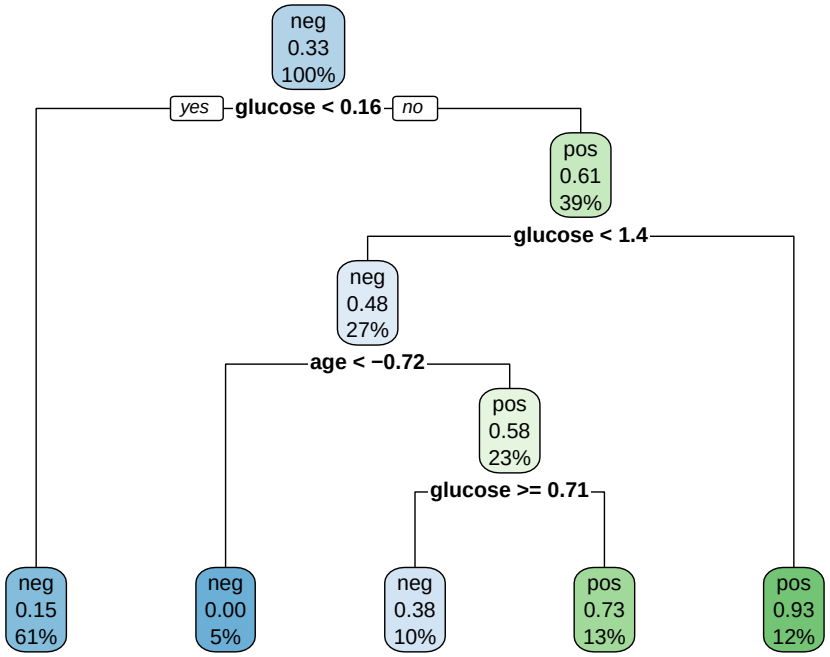
Rysunek 7: Wykres xerror-rozmiar drzewa. Jego zadaniem jest pomoc w ustaleniu optymalnego rozmiaru drzewa zgodnie z regułą 1SE



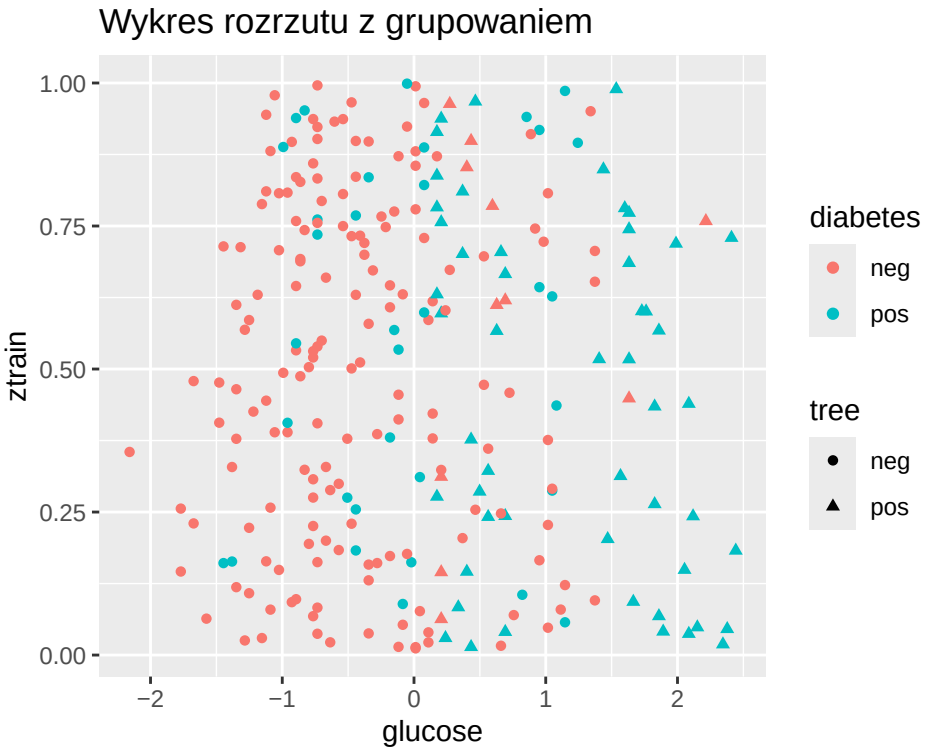
Rysunek 8: Optymalnie przycięte drzewo



Rysunek 9: Optymalnie przycięte drzewo dla cp = .02 i wszystkich zmiennych



Rysunek 10: Optymalnie przycięte drzewo dla cp = .02 i zmiennych glucose, oraz age (akurat takie samo jak pierwsze)



Rysunek 11: Wykres rozrzutu dla cp=.02, , minsplit=5, maxdepth=20, zmiennych glucose i age. Grupowanie wg rzeczywistej etykietki i klasyfikacji metodą drzew klasyfikacyjnych

	parametry	bledy
1	train, cp .01, ms 5, md 20	0.174
2	train, cp .02, ms 10, md 20	0.243
3	train, cp .02, ms 5, md 20, g+a	0.174
4	test, cp .01, ms 5, md 20	0.271
5	test, cp .02, ms 10, md 20	0.226
6	test, cp .02, ms 5, md 20, g+a	0.271
7	bootstrap 50, cp .02, ms 5, md 20, g+a	0.232
8	bootstrap 50, cp .01, ms 5, md 20	0.264

Tabela 6: Zestawienie błądów klasyfikacji. W pierwszej kolumnie informacje zakodowane: test/train - zbiór testowy/uczący, ms - minsplit, md - maxdepth, g - glucose, a - age, ”” - wszystkie cechy

Ponownie wyniki na zbiorze uczącym odbiegają od tych na zbiorze testowym, czy z wykorzystaniem bootstrapu. Przy odpowiednich parametrach wyniki są porównywalne z tymi osiągniętymi za pomocą k-najbliższych sąsiadów, mając zaletę w postaci większej interpretowalności.
Poniżej zaprezentuję kod wykorzystany do znalezienia optymalnej wielkości drzewa:

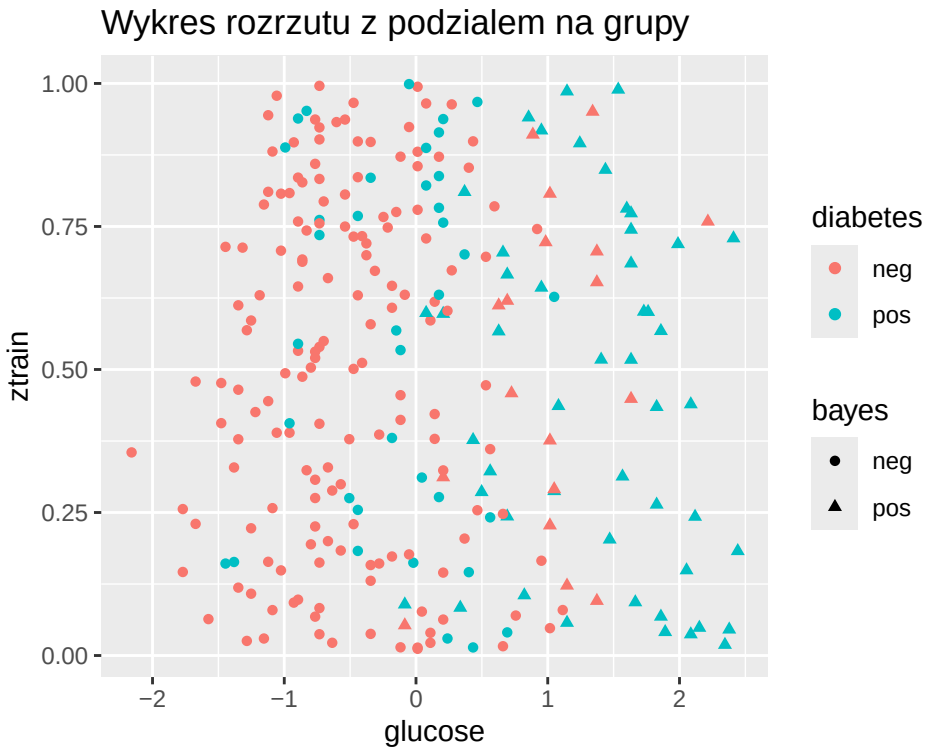
```
# ustalam cp dla najmniejszego błędu kroswalidacyjnego

bestcp1 <- tree1$cptable[which.min(tree1$cptable[,"xerror"]),"CP"]
val1 <- tree1$cptable[which.min(tree1$cptable[,"xerror"]),"xerror"] +
  tree1$cptable[which.min(tree1$cptable[,"xerror"]),"xstd"]

# znajduję najmniejsze drzewo zgodnie z regułą 1SE

s1SE1 <- which.min((tree1$cptable[tree1$cptable[,"xerror"] <= val1,])[,"nsplit"])
valid1 <- as.data.frame(tree1$cptable[tree1$cptable[,"xerror"] <= val1,])
cp1 <- valid1[which.min(valid1$nsplit),"CP"]
tree1pruned <- prune(tree1,cp=cp1)
```

3.4 Naiwny klasyfikator bayesowski



Rysunek 12: Wykres rozrzutu dla danych uczących z wykorzystaniem zmiennych glucose i age, bez wykorzystania jądrowego estymatora gęstości. Grupowanie wg rzeczywistej etykietki i klasyfikacji metodą drzew klasyfikacyjnych

	parametry	bledy
1	train, kernel	0.170
2	train, no kernel	0.220
3	train, kernel, b + a	0.193
4	train, no kernel, b + a	0.212
5	test, kernel	0.286
6	test, no kernel	0.248
7	test, kernel, b + a	0.241
8	test, no kernel, b + a	0.195
9	crossvalidation, kernel	0.242
10	crossvalidation, kernel, b + a	0.209
11	crossvalidation, no kernel	0.230
12	crossvalidation, no kernel, b + a	0.214

Tabela 7: Zestawienie błądów klasyfikacji. W pierwszej kolumnie informacje zakodowane: test/train - zbiór testowy/uczący, g - glucose, a - age, "" - wszystkie cechy

Naiwny klasyfikator bayesowski osiągnął bardzo dobrą dokładność, szczególnie gdy nie korzystaliśmy z jądrowego estymatora gęstości, który oferował lepsze rezultaty na zbiorze uczącym, znacznie pogarszając je na zbiorze testowym. Jednak przy zastosowaniu kroswalidacji wersja z jądrowym estymatorem nie odbiegała tak mocno. Istotnie lepsze rezultaty otrzymywaliśmy konstruując klasyfikator na podstawie zmiennych o dużej zdolności dyskryminacyjnej.

3.5 Podsumowanie

Poniżej zestawimy błędy wyznaczone przy pomocy zaawansowanych metod - bootstrap i kroswalidacji.

	metoda	error
1	KNN k=15 g + a	0.232
2	KNN k=15 all	0.230
3	tree cp=0.02 g + a	0.232
4	tree cp=0.01 all	0.264
5	bayes kernel all	0.242
6	bayes kernel g + a	0.209
7	bayes all	0.230
8	bayes g + a	0.214

Tabela 8: Zestawienie błądów klasyfikacji dla różnych metod i parametrów

W klasyfikacji analizowanych danych najlepiej sprawdził się naiwny klasyfikator bayesowski. Zarówno w przypadku tego klasyfikatora, jak i drzew klasyfikacyjnych ograniczenie się do zmiennych *glucose* i *age* przynosiło lepsze rezultaty. Przy testach dla wielu parametrów, zbiorów i zmiennych zazwyczaj można było wyciągnąć podobne wnioski, jak przy korzystaniu z

zaawansowanych metod oceny dokładności. W przypadku metod k-nn, drzew klasyfikacyjnych i naiwnego estymatora bayesowskiego z jądrowym estymatorem gęstości wyniki na zbiorze uczącym i testowym istotnie się różniły, zatem można było przypuszczać, że "prawdziwa dokładność" jest gdzieś pomiędzy. Takie rozumowanie nie sprawdziłoby się jednak w przypadku chociażby naiwnego estymatora bayesowskiego bez jądrowego estymatora gęstości dla zmiennych *glucose* i *age*.