Projekt - Statistical Learning

Justyna Grapa, Sylwester Kubik, Alicja Łata

13 Styczeń 2025

1 Wczytanie i przygotowanie danych

Zbiór danych "Heart Disease" zawiera informacje dotyczące diagnozy chorób serca, zgromadzone z czterech różnych instytucji medycznych. W bazie znajdują się szczegółowe informacje o pacjentach, uwzględniające różne zmienne medyczne, które mogą mieć wpływ na występowanie i ryzyko wystąpienia chorób sercowo-naczyniowych. Dane obejmują **14 atrybutów**:

1. Wiek pacjenta
2. Płeć (1 - mężczyzna, 0 - kobieta)
3. Rodzaj bólu w klatce piersiowej
4. Ciśnienie krwi w spoczynku
5. Poziom cholesterolu
6. Poziom cukru na czczo
7. Wynik EKG w spoczynku
8. Maksymalne tętno podczas badania
9. Duszności podczas wysiłku (1 - Tak, 0 - Nie)
10. Zmiana ST w wyniku wysiłku
11. Skrócony odcinek ST
12. Liczba zwężonych naczyń krwionośnych
13. Wynik badania talu - defekt (3 - normalny wynik, 6 i 7 - obecność defektu)
14. Diagnoza choroby serca (klasa: 0 - brak choroby, 1 - obecność choroby)

	wiek	plec	b_klat_pier	cis	chol	cuk	EKG	max_tet	dusz_bol	zm_ST	skr_odc_ST	licz_nacz	defekt	klasa
1	28	1	2	130	132	0	2	185	0	0	?	?	?	0
2	29	1	2	120	243	0	0	160	0	0	?	?	?	0
3	29	1	2	140	?	0	0	170	0	0	?	?	?	0
4	30	0	1	170	237	0	1	170	0	0	?	?	6	0
5	31	0	2	100	219	0	1	150	0	0	?	?	?	0

Mamy łącznie 294 obserwacji, jednak w zbiorze danych zauważyliśmy występowanie brakujących wartości, szczególnie w kolumnach:

- Skrócony odcinek ST - brakujące wartości w 190 przypadkach
- Liczba zwężonych naczyń - brakujące wartości w 291 przypadkach
- Defekt - brakujące wartości w 266 przypadkach

Brakujące wartości w tych kolumnach są bardzo liczne i najprawdopodobniej wynikają z błędów pomiarowych lub niekompletności danych, dlatego postanowiliśmy usunąć te obserwacje. Dodatkowo, w 32 innych przypadkach brakujące były pojedyncze atrybuty – te obserwacje również zostały usunięte.
Po usunięciu braków mamy łącznie **261 obserwacji**.

1.1 Przetwarzanie danych

Pewne atrybuty w zbiorze danych były typu znakowego. Zamieniamy je na odpowiedni typ:

- Numeryczny: ciśnienie krwi w spoczynku, poziom cholesterolu, maksymalne tętno, zmiana ST w wyniku wysiłku
- Kategoryczny (czynnikowy): poziom cukru na czczo, rodzaj bólu w klatce piersiowej
- Kolumnę `klasa` zmieniliśmy na typ czynnikowy, ponieważ jest to zmienna klasyfikacyjna

2 Cel analizy

Celem naszej analizy jest określenie, czy na podstawie różnych parametrów zdrowotnych można przewidzieć obecność choroby serca (klasa 1) lub jej brak (klasa 0). Dodatkowo przeprowadziliśmy analizę zależności pomiędzy zmiennymi. Za pomocą macierzy rozrzutu która przedstawia wizualizację relacji między atrybutami oraz analizy korelacji nie zauważyliśmy istotnych zależności.

```
> cor(Choroba[,c(1,2,3,4,5,7,8,10)])
                wiek        plec  b_klat_pier         cis        chol          EKG     max_tet       zm_ST
wiek       1.00000000  0.02013288  0.142592378  0.25788877  0.09693666  0.052657432 -0.46009497  0.20863297
plec       0.02013288  1.00000000  0.217587743  0.09493728  0.05565287 -0.081372305 -0.07306168  0.12092531
b_klat_pier 0.14259238 0.21758774  1.000000000  0.07950370  0.16104871  0.006511533 -0.39012785  0.36006275
cis        0.25788877  0.09493728  0.079503704  1.00000000  0.11689042  0.022250459 -0.22070767  0.22911655
chol       0.09693666  0.05565287  0.161048707  0.11689042  1.00000000  0.056886124 -0.13629200  0.11357179
EKG        0.05265743 -0.08137231  0.006511533  0.02225046  0.05688612  1.000000000 -0.01111651  0.02345726
max_tet   -0.46009497 -0.07306168 -0.390127846 -0.22070767 -0.13629200 -0.011116507  1.00000000 -0.32720709
zm_ST      0.20863297  0.12092531  0.360062745  0.22911655  0.11357179  0.023457265 -0.32720709  1.00000000
```

3 Podział danych

Zbiór danych został losowo podzielony na dwa podzbiory:

- Zbiór treningowy: 196 obserwacji
- Zbiór testowy: 65 obserwacji

```
> prop.table(table(Choroba_test$klasa))

        0         1
0.6153846 0.3846154
> prop.table(table(Choroba_train$klasa))

        0         1
0.627551 0.372449
```

Rysunek 1: Proporcja podziału klas w zbiorze treningowym i testowym

4 Normalizacja danych

Aby odpowiednio przygotować zbiór danych, przeprowadzimy ich normalizację. Na podstawie wcześniej wykonanych testów Shapiro-Wilka ustaliliśmy, że normalizacja metodą `Z-score` będzie zastosowana dla kolumn `chol` oraz `max_tet`. Analiza testem Shapiro-Wilka potwierdziła nas w wykorzystaniu tej metody dla tych kolumn, uwzględniając również wartości odstające w zbiorze treningowym i testowym. Po zastosowaniu opisanych działań, w zbiorze treningowym pozostało **189 obserwacji**, natomiast w zbiorze testowym **61 obserwacji**.

5 Model LDA (Linear Discriminant Analysis)

Do budowy modelu użylismy analizy dyskryminacyjnej liniowej (LDA). Celem tego modelu jest przewidywanie przynależności do klasy 0 lub 1 na podstawie zestawu zmiennych niezależnych: `wiek`, `płeć`, `b_klat_pier`, `cis`, `chol`, `cuk`, `EKG`, `max_tet`. Model został stworzony za pomocą funkcji `lda()` z pakietu MASS. Wyniki modelu LDA:

- Dokładność modelu na zbiorze testowym przy progu 0.5: **80,33%**

```
             | predicted
      actual |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |        39 |         0 |        39 |
             |     0.639 |     0.000 |           |
-------------|-----------|-----------|-----------|
           1 |        12 |        10 |        22 |
             |     0.197 |     0.164 |           |
-------------|-----------|-----------|-----------|
Column Total |        51 |        10 |        61 |
-------------|-----------|-----------|-----------|

> mean(lda.class1 == Choroba_test_pred)
[1] 0.8032787
```

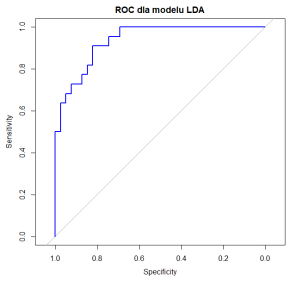
Rysunek 2: Predykcja modelu podstawowego LDA

- Zmienione progi (0.6 i 0.8) wpłynęły na wyniki, zwiększając liczbę fałszywych negatywów co prowadzi do spadku dokładności modelu, ponieważ więcej osób chorych zostaje uznanych za zdrowych (aż 17). Przy progu 0.6 oraz przy progu 0.8 dokładność wynosiła 72,13%.

actual	predicted 0	1	Row Total
0	39 0.639	0 0.000	39
1	17 0.279	5 0.082	22
Column Total	56	5	61

```
> mean(lda.class2.threshold == Choroba_test_pred)  #Trafność klasyfikacji z progiem 0.6
[1] 0.7213115
```

Rysunek 3: Predykcja LDA z progami



Rysunek 4: Krzywa ROC modelu LDA

- Tworzymy **krzywą ROC**, która daje nam możliwość przedstawienia obu rodzajów błędu dla wszystkich możliwych progów. Pole powierzchni pod tą krzywą informuje nas o wydajności naszego klasyfikatora. Ogólna jakość modelu LDA wynosi **93,59%**.

6 Model QDA (Quadratic Discriminant Analysis)

Zastosowaliśmy również model kwadratowej analizy dyskryminacyjnej (QDA), który lepiej radzi sobie z przypadkami, gdy klasy mają różne macierze kowariancji. Wyniki modelu QDA:

- Dokładność na zbiorze testowym: **77,05%**
- Przy progu 0.6: 73,77%
- Przy progu 0.8: 75,41%

actual	predicted 0	1	Row Total
0	32 0.525	7 0.115	39
1	7 0.115	15 0.246	22
Column Total	39	22	61

```
> mean(qda.class1 == Choroba_test$klasa)  #Trafność klasyfikacji
[1] 0.7704918
```

Rysunek 5: Predykcja QDA

actual	predicted 0	1	Row Total
0	33 0.541	6 0.098	39
1	10 0.164	12 0.197	22
Column Total	43	18	61

```
> mean(qda.class2.threshold == Choroba_test$klasa)
[1] 0.7377049
```

Rysunek 6: Predykcja QDA z progiem 0.6

actual	predicted 0	1	Row Total
0	35 0.574	4 0.066	39
1	11 0.180	11 0.180	22
Column Total	46	15	61

```
> mean(qda.class3.threshold == Choroba_test$klasa)
[1] 0.7540984
```

Rysunek 7: Predykcja QDA z progiem 0.8

- Ogólna jakość modelu QDA, czyli wartość współczynnika AUC, wynosi 76,92%.

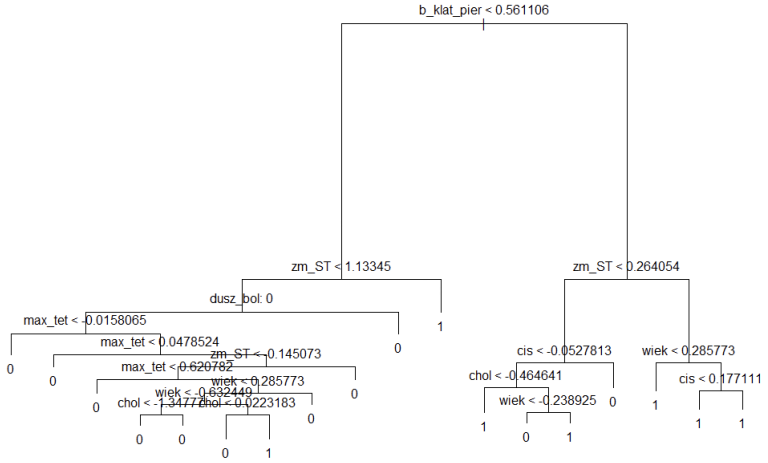
7 Porównanie modeli

Model LDA osiągnął wyższą dokładność w porównaniu do modelu QDA. Zmiana progów pozwoliła na dostosowanie modeli do różnych wymagań precyzji:

- Niższe progi zwiększają liczbę wykrytych przypadków choroby kosztem większej liczby fałszywych alarmów.
- Wyższe progi redukują liczbę fałszywych pozytywów, ale zwiększają liczbę fałszywych negatywów.

8 Drzewa decyzyjne

Teraz przejdziemy do budowy drzewa klasyfikacyjnego dla zmiennej `klasa` na podstawie wszystkich innych zmiennych.

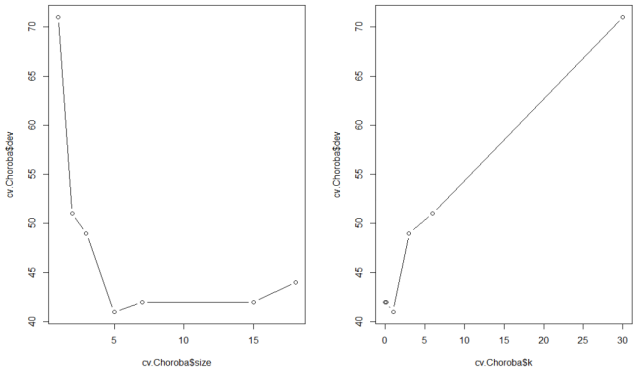


- Do budowy modelu wykorzystujemy 7 zmiennych objaśniających. Predykcja tego modelu wynosi **75,41%**. Mamy łącznie 15 niepoprawnie przypisanych osób, z czego 7 z nich jest faktycznie chorych, natomiast nasz model sklasyfikował je jako osoby zdrowe.

	predicted		
actual	0	1	Row Total
0	31	8	39
	0.508	0.131	
1	7	15	22
	0.115	0.246	
Column Total	38	23	61

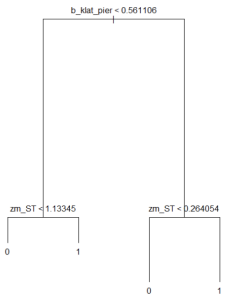
```
> mean(tree.pred_1 == Choroba_test$klasa)
[1] 0.7540984
```

Rysunek 8: Predykcja modelu drzewa



Rysunek 9: Wykresy błędów z odpowiednimi rozmiarami liści

- Drzewa decyzyjne mają tendecję do przeuczenia, ponieważ możemy nadać tyle warunków, że każdy przykład będzie należał do własnej podprzestrzeni. Zatem spróbujemy zmniejszyć w tym drzewie liczbę jego rozgałęzień. Przeprowadzimy walidację krzyżową w celu znalezienia optymalnego rozmiaru drzewa.



Rysunek 10: Drzewo decyzyjne z 4 liśćmi

	predicted		
actual	0	1	Row Total
0	36	3	39
	0.590	0.049	
1	7	15	22
	0.115	0.246	
Column Total	43	18	61

```
> mean(tree.pred2 == Choroba_test$klasa)
[1] 0.8360656
```

Rysunek 11: Predykcja drzewa z 4 liśćmi

- Okazuje się, że najlepszym przyciętym drzewem może być drzewo o rozmiarze 4, które daje najmniejszy błąd klasyfikacji. Spróbujemy również zbudować drzewa o innych rozmiarach. Poniższa tabela przedstawia 4 modele drzewa z różnymi ich rozmiarami oraz z pewnymi informacjami:

Rozmiar drzewa	**Błąd źle sklasyfikowanych osób zdrowych**	**Błąd źle sklasyfikowanych osób chorych**	**Jakość modelu**
4,5	3	7	83,61%
3	3	10	78,69%
6	9	4	78,69%
7	5	7	80,33%

Tabela 1: Wyniki klasyfikacji dla różnych rozmiarów drzewa

- Model, który zasugerowała nam metoda CV jest jakościowo najlepszy, z drugiej strony model drzewa z 6 liśćmi ma najmniejszą liczbę źle sklasyfikowanych osób chorych, a jakość tego modelu jest niewiele gorsza od jakości modelu z 4 liśćmi. Dlatego w tej części możemy stwierdzić, że model drzewa przyciętego do 6 liści jest najlepszy (w tej części podrozdziału).

9 Metody grupowania

9.1 Bagging

Ta technika ma na celu zwiększenie dokładności i stabilności modeli predykcyjnych poprzez zmniejszenie wariancji modelu.

- Dla domyślnej liczby drzew `ntree`, model łącznie przypisuje błędnie 14 osób, z czego 8 jest uznanych przez model za zdrowych, mimo iż w rzeczywistości są chorzy. Dokładność modelu wynosi **77.05%**.

	predicted		
actual	0	1	Row Total
0	33	6	39
	0.541	0.098	
1	8	14	22
	0.131	0.230	
Column Total	41	20	61

```
> mean(yhat.bag1 == Choroba_test$klasa)
[1] 0.7704918
```

Rysunek 12: Predykcja Bagging z domyślną liczbą ntree

- W tym modelu możemy zmieniać współczynnik **ntree** który opisuje liczbę zbudowanych drzew w bootstrapie.

	predicted		
actual	0	1	Row Total
0	31	8	39
	0.508	0.131	
1	8	14	22
	0.131	0.230	
Column Total	39	22	61

```
> mean(yhat.bag2 == Choroba_test$klasa)
[1] 0.7377049
```

Rysunek 13: Predykcja Bagging z ntree = 10

	predicted		
actual	0	1	Row Total
0	30	9	39
	0.492	0.148	
1	9	13	22
	0.148	0.213	
Column Total	39	22	61

```
> mean(yhat.bag3 == Choroba_test$klasa)
[1] 0.704918
```

Rysunek 14: Predykcja Bagging z ntree = 20

- Spośród rozważanych modeli z nadaną liczbą `ntree`, model z domyślną liczbą ntree jest wśród wszystkich wymienionych modeli w tabeli najlepszy.

Model z pewną liczbą ntree	Błąd źle sklasyfikowanych osób zdrowych	Błąd źle sklasyfikowanych osób chorych	Jakość modelu)
domyślnie	6	8	77,05
10	8	8	73,77
20	9	9	70,50
50	9	8	73,77
100	7	8	75,41
170	6	9	75,41

Tabela 2: Wyniki klasyfikacji dla modelu Bagging z różnymi parametrami ntree

9.2 Las losowy

Jest to modyfikacja metody bagging, która zmiejsza korelację pomiędzy drzewami składowymi całego modelu. W problemie klasyfikacji najczęściej wybieramy $\sqrt{p}$ cech.

- Dla 3 losowych cech dokładność modelu wynosi **78,69%**. Mamy 8 źle sklasyfikowanych osób zdrowych.
- Dla 5 zmiennych dokładność modelu spada.

Spośród obu modeli, możemy wybrać model z 3 losowymi cechami.

actual	predicted 0	1	Row Total
0	34 0.557	5 0.082	39
1	8 0.131	14 0.230	22
Column Total	42	19	61

```
> mean(yhat.rf1 == choroba_test$klasa)
[1] 0.7868852
```

Rysunek 15: Predykcja Las Losowy z mtry = 3

actual	predicted 0	1	Row Total
0	32 0.525	7 0.115	39
1	8 0.131	14 0.230	22
Column Total	40	21	61

```
> mean(yhat.rf2 == Choroba_test$klasa)
[1] 0.7540984
```

Rysunek 16: Predykcja Las Losowy z mtry = 5

9.3 Boosting

Technika która ma na celu zmniejszenie błędu modelu poprzez sekwencyjne jego trenowanie. Będziemy wykorzystywać informacje pozyskane w poprzednich modelach drzew do budowy kolejnych drzew.

- Dokładność naszego modelu o `ntree` = 50 wynosi **79,37%**, natomiast po raz kolejny mamy dość dużą liczbę osob chorych niepoprawnie sklasyfikowanych jako zdrowe (13). Możemy zauważyć, że w tym modelu istotne znaczenie mają zmienne: `zm_ST`, `b_klat_pier`, oraz `dusz_bol`.
- Dla modelu o ntree = 20 i parametrze $\lambda = 0.2$ mamy 12 źle sklasyfikowanych osób chorych. Dla modelu o ntree = 20 i domyślnej wartości $\lambda = 0.1$ ilosć źle sklasyfikowanych osób zdrowych wynosi aż 15.

actual	predicted 0	1	Row Total
0	40 0.635	0 0.000	40
1	13 0.206	10 0.159	23
Column Total	53	10	63

```
> mean(yhat.rf.boo1 == choroba_test_boo$klasa)
[1] 0.7936508
```

Rysunek 17: ntree = 50

actual	predicted 0	1	Row Total
0	39 0.619	1 0.016	40
1	12 0.190	11 0.175	23
Column Total	51	12	63

```
> mean(yhat.rf.boo2 == Choroba_test$klasa)
[1] 0.7936508
```

Rysunek 18: ntree =20, λ =0.2

actual	predicted 0	1	Row Total
0	40 0.635	0 0.000	40
1	15 0.238	8 0.127	23
Column Total	55	8	63

```
> mean(yhat.rf.boo3 == Choroba_test$klasa)
[1] 0.7619048
```

Rysunek 19: ntree = 20

Spróbujemy użyć metody CV do wyboru modelu jakościowo (według współczynnika AUC) najlepszego względem liczby ntree oraz liczby shrinkage = λ. Według tej metody wybraliśmy trzy najlepsze modele:

- ntree = 10, $\lambda = 0.1$,
- ntree = 70, $\lambda = 0.05$,
- ntree = 110, $\lambda = 0.3$,

ntree	**Shrinkage**	**Błąd źle sklasyfikowanych osób zdrowych**	**Błąd źle sklasyfikowanych osób chorych**	**Jakość modelu**
50	domyślnie	0	13	79,37
20	0.2	1	12	79,37
20	domyślnie	0	15	76,19
10	0.1	0	15	76,19
70	0.05	0	15	76,19
110	0.3	0	14	77,78

Tabela 3: Wyniki klasyfikacji dla różnych parametrów ntree i shrinkage

Podsumowując, w tej części, najlepszym modelem okazuje się model z ntree = **20** oraz $\lambda = \mathbf{0.2}$. Modele te jednak nie są dobre, gdyż sklasyfikowały błędnie dużą liczbę osób faktycznie chorych.

9.4 XGBoost

XGBoost działa w oparciu o *boosting* drzew decyzyjnych. Jest używany do klasyfikacji, gdzie kolejne modele uczą się na błędach poprzednich. Wprowadza *regularyzację*, co pomaga uniknąć przeuczenia. W klasyfikacji przypisuje próbkom odpowiednie klasy na podstawie wyników drzew decyzyjnych.
Wybieramy najlepsze hiperparametry według współczynnika AUC:

- eta = 0.6, max_depth = 8, nround = 100,
- eta = 0.6, max_depth = 2, nround = 16
- eta = 0.2, max_depth = 2, nround = 16

Uwzględniając jeszcze zbudowany wskaźnik FPR zbudowaliśmy jeszcze dwa modele z hiperparametrami:

- eta = 0.4, max_depth = 8, nround = 13,
- eta = 0.1, max_depth = 9, nround = 20,

oraz pewny model, wprowadzając ręcznie hiperparametry: eta = 0.1, max_depth = 3, nround = 50.

Eta	**Max_depth**	**Nrounds**	**Błąd źle sklasyfikowanych osób zdrowych**	**Błąd źle sklasyfikowanych osób chorych**	**Jakość modelu**
0.6	8	100	2	10	80,95
0.6	2	16	6	7	79,37
0.2	2	16	0	11	82,54
0.4	8	13	3	10	79,37
0.1	9	20	0	11	82,54
0.1	3	50	0	11	82,54

Tabela 4: Wyniki klasyfikacji dla różnych parametrów eta, max_depth i nrounds

Wśród wszystkich zbudowanych, okazuje się, że najlepszy jest model z hiperparametrami: **eta = 0.6, max_depth = 2, nround = 16**.

9.5 BART

Jest to metoda oparta na drzewach decyzyjnych, która wykorzystuje podejście bayesowskie do przewidywania wyników. W odróżnieniu od np. XGBoost, BART generuje prognozy, które są średnią z wielu drzew decyzyjnych, przy czym każde z tych drzew może mieć różną strukturę.

- Model BART uzyskał stosunkowo dobrą jakość klasyfikacji (84.13% dokładności).
- Błędnie sklasyfikowanych przypadków osób faktycznie chorych, jako zdrowe jest łącznie 9.
- 1 zdrowa osoba została błędnie przypisana do grupy chorych.

	predicted		
actual	0	1	Row Total
0	39	1	40
	0.619	0.016	
1	9	14	23
	0.143	0.222	
Column Total	48	15	63

```
> mean(yhat.class == Choroba_test_y)
[1] 0.8412698
```

Na początku uruchomiliśmy model z domyślnymi parametrami (w przypadku funkcji pbart(), standradowo liczba drzew ustawiona jest na 200). Następnie eksperymentowaliśmy już z ręcznie dobranymi wartościami parametru `ntree` (230,330,430), aby sprawdzić, czy zwiększenie liczby drzew poprawi jakość klasyfikacji. Najlepszym jakościowo modelem okazały się modele z ntree=230 oraz ntree=430, dokładność modeli wynosi **84,13%**. Mamy 9 osób chorych źle sklasyfikowanych jako zdrowe oraz 1 zdrową zaklasyfikowaną jako osoba chora.

9.6 Podsumowanie

W projekcie przeanalizowaliśmy różne modele klasyfikacyjne, takie jak LDA, QDA, Drzewa Decyzyjne, Bagging, Boosting , XGBoost i Bart. Celem było zminimalizowanie błędów klasyfikacji zdrowych i chorych osób oraz uzyskanie najwyższej jakości modelu.

-> Błąd zdrowych - Błąd źle sklasyfikowanych osób zdrowych
-> Błąd chorych - Błąd źle sklasyfikowanych osób chorych

Model	Błąd zdrowych	Błąd chorych	Jakość modelu
Bagging	6	8	77.05%
QDA	7	7	77.05%
Drzewo decyzyjne: size = 6	9	4	78.69%
Las losowy: m=3	5	8	78.69%
XGBoost: eta = 0.6, max_depth = 2	6	7	79,37%
Boosting: ntree = 20, shrinkage = 0.2	1	12	79,37%
LDA przy progu 0.5	0	12	80.33%
BART dla ntree=430	1	9	84.13%

Tabela 5: Modele posortowane według jakości modelu w porządku rosnącym.

W oparciu o uzyskane wyniki najlepsze modele to:

- **Drzewo decyzyjne , size=6**: jakość (**78,69%**) i niskie błędy (9 zdrowych, 4 chorych).
- **XGBoost (eta=0.6, max_depth=2)**: jakość (**79,37%**), błędy (6 zdrowych, 7 chorych).
- **QDA**:jakość (**77,05%**) , błędy (7 zdrowych, 7 chorych).