Sprawozdanie 4

Jakub Markowiak
album 255705

15 czerwca 2021

Spis treści

1 Krótki opis zagadnienia

W tym sprawozdaniu będziemy kontynuować porównywanie metod klasyfikacji dla danych `Glass` z pakietu `mlbench`. Tym razem, w celu poprawienia dokładności klasyfikacji, wykorzystamy algorytmy bagging, boosting oraz random forest, a następnie postaramy się rozstrzygnąć, który poradził sobie najlepiej z naszym zagadnieniem. Następnie zastosujemy metody analizy skupień, również dla danych `Glass`, oraz porównamy algorytmy k-means, PAM, AGNES i DIANA w celu wyłonienia najbardziej optymalnego. Postaramy się również wybrać optymalną liczbę skupień, a następnie zinterpretować otrzymany podział na klastry.

2 Opis eksperymentów/analiz

Przeprowadzimy następujące analizy i eksperymenty:

1. porównanie metod klasyfikacji dla danych `Glass` z pakietu `mlbench` (cd.),
2. ocena oraz porównanie jakości grupowania dla różnych algorytmów analizy skupień.

3 Wyniki

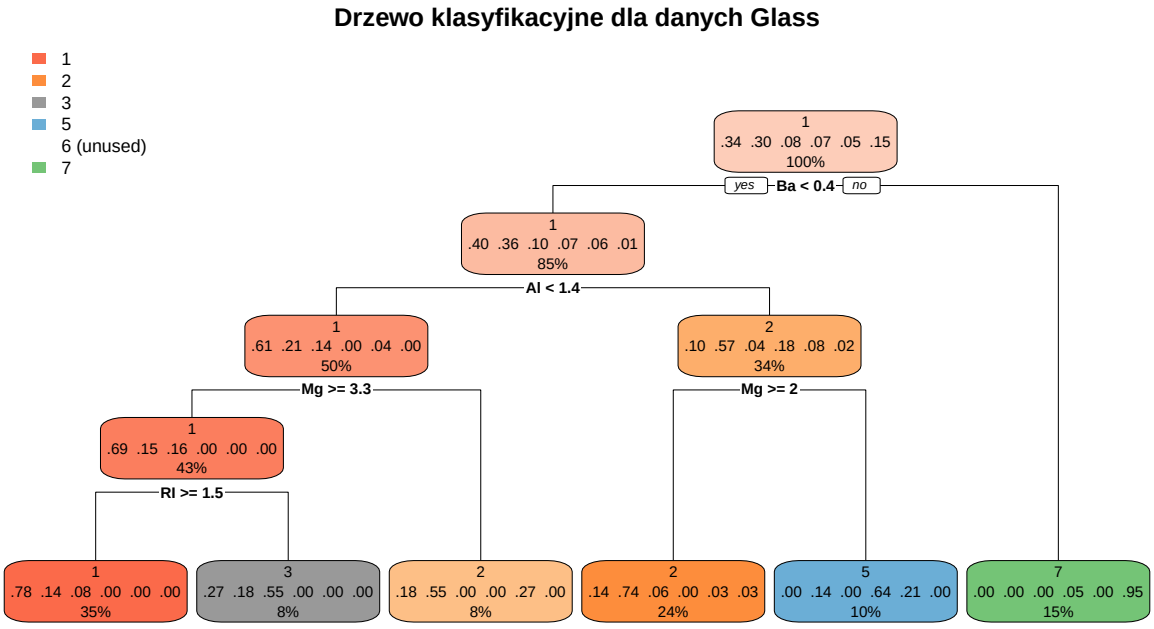
3.1 Porównanie metod klasyfikacji dla danych Glass z pakietu mlbench cd.

Porównywanie metod klasyfikacji będziemy kontynuować wykorzystując dane `Glass`, które zawierają informacje o współczynniku załamania światła oraz zawartości poszczególnych pierwiastków chemicznych dla badanych szkieł. Tak jak poprzednio przygotowujemy zbiór uczący i zbiór testowy (w proporcji 2 : 1).

	l. obserwacji	1	2	3	5	6	7
learn set	143	49	43	12	10	7	22
test set	71	21	33	5	3	2	7

Tabela 1: Podział na zbiór uczący i testowy

Za klasyfikator bazowy będzie nam służyło najlepsze drzewo klasyfikacyjne skonstruowane w poprzednim sprawozdaniu.



Rysunek 1: Drzewo klasyfikacyjne dla danych Glass

	5-cv	bootstrap	.632+	predykcje na zb. testowym
błąd klasyfikacji (dk)	0.346	0.339	0.288	0.324

Tabela 3: Błędy klasyfikacji dla drzewa klas.

Analizę przeprowadzimy w celu porównania następujących metod:

1. bagging,

Tabela 2: Macierz pomyłek dla zbioru testowego (dk)

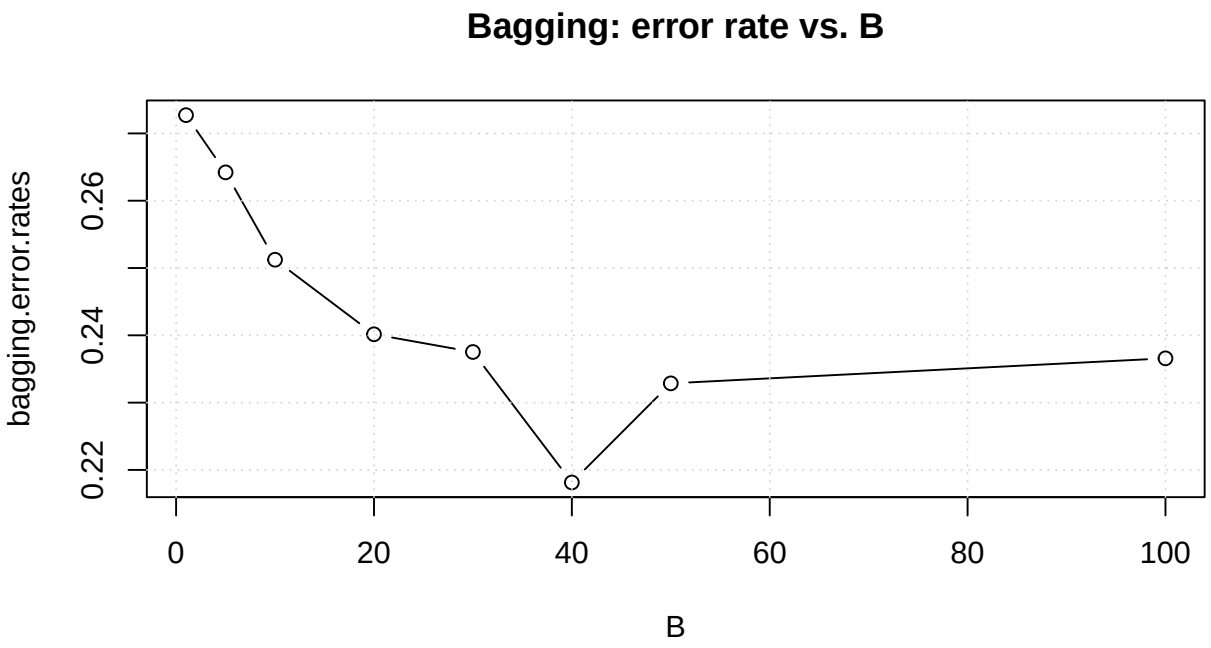
Rzeczywiste etykietki	Prognozowane etkietki					
	1	2	3	5	6	7
1	17	8	2	0	0	1
2	3	23	2	1	2	1
3	0	1	1	0	0	0
5	0	0	0	2	0	0
6	0	0	0	0	0	0
7	1	1	0	0	0	5

2. boosting,

3. random forest.

Metoda bagging

Rozpoczniemy od metody **bagging**. Najpierw spróbujemy wyłonić optymalną liczbę replikacji `B`.



Rysunek 2: Błąd w zależności od B

Widzimy, że optymalną wartością `B` jest 20. Przeprowadzimy zatem bagging dla 20 replikacji.

	5-cv	bootstrap	.632+	predykcje na zb. testowym
błąd klasyfikacji (bagging)	0.262	0.283	0.211	0.254

Tabela 5: Ocena dokładności klasyfikacji

Tabela 4: Macierz pomyłek dla zbioru testowego (bagging)

Rzeczywiste etykietki	Prognozowane etkietki					
	1	2	3	5	6	7
1	19	7	2	0	0	1
2	1	24	2	1	0	1
3	0	1	1	0	0	0
5	0	0	0	2	0	0
6	0	0	0	0	2	0
7	1	1	0	0	0	5

Metoda boosting (AdaBoost)

Skonstruujemy teraz rodzinę klasyfikatorów z wykorzystaniem metody Adaptive Boosting. Ustalamy liczbę potwórzeń jako 100.

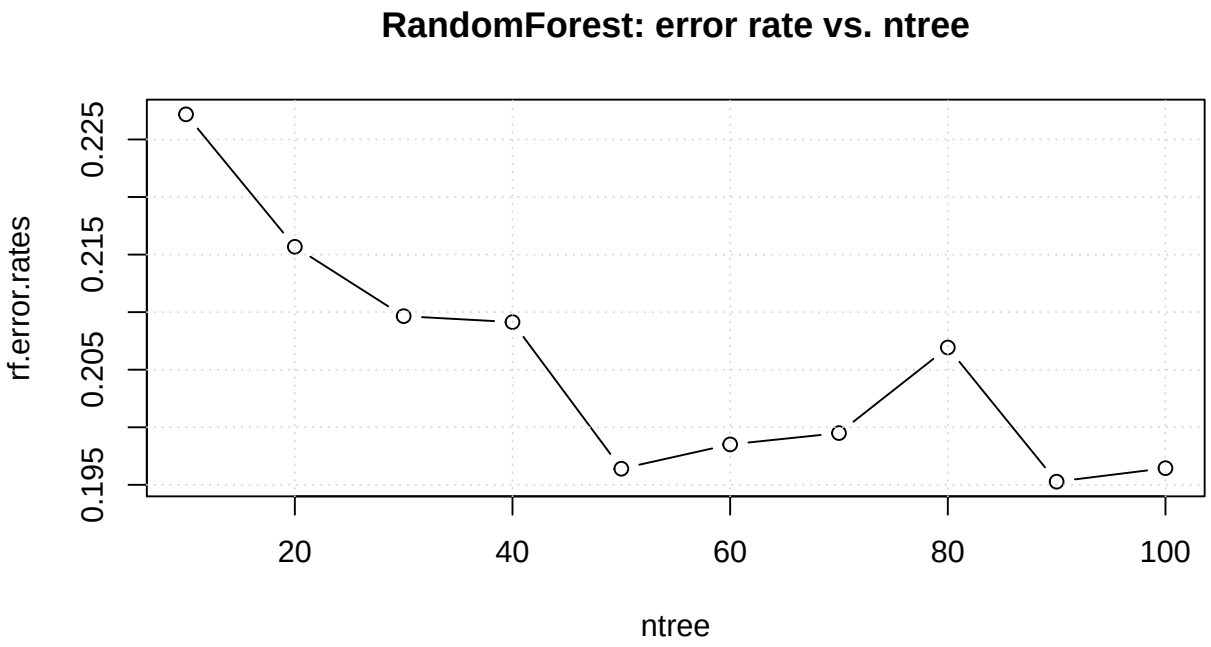
Tabela 6: Macierz pomyłek dla zbioru testowego (boosting)

Rzeczywiste etykietki	Prognozowane etkietki					
	1	2	3	5	6	7
1	18	6	3	0	0	1
2	2	24	1	3	0	0
3	1	0	1	0	0	0
5	0	0	0	0	0	1
6	0	2	0	0	2	0
7	0	1	0	0	0	5

	5-cv	bootstrap	.632+	predykcje na zb. testowym
błąd klasyfikacji (boosting)	0.257	0.275	0.201	0.296

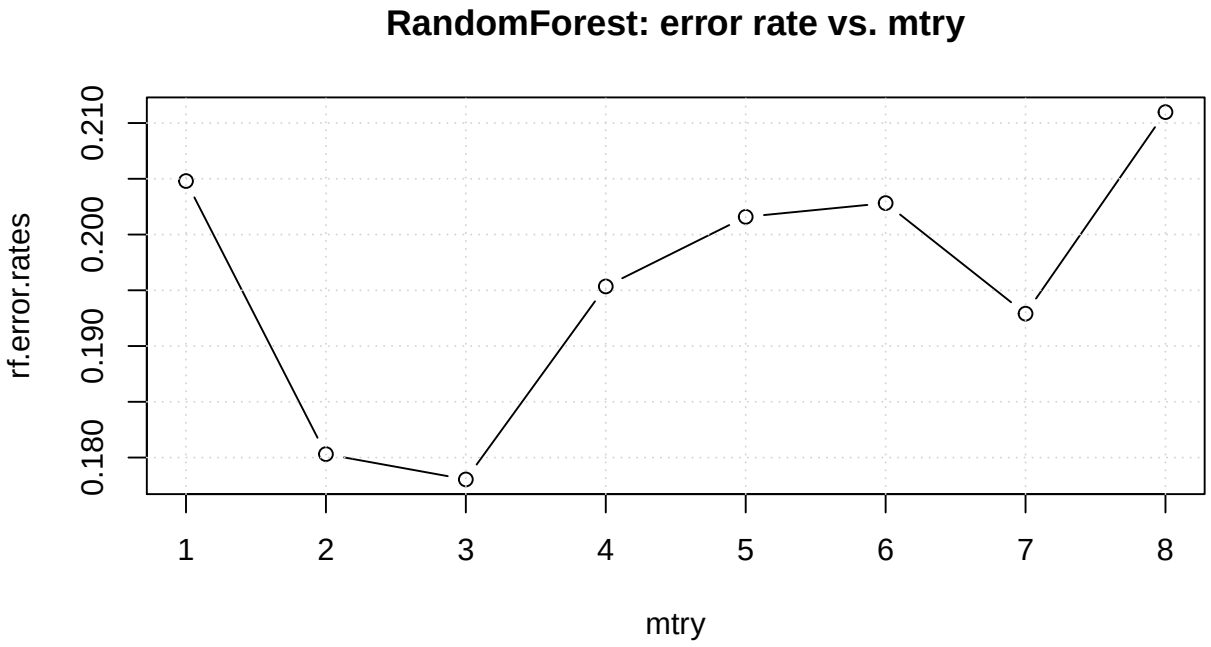
Tabela 7: Ocena dokłądności klasyfikacji

Metoda Random Forest



Rysunek 3: Błąd klasyfikacji w zależności od liczby drzew.

Ustalamy liczbę drzew jako 50.



Rysunek 4: Błąd klasyfikacji w zależności od liczby losowo wybranych cech

Ustalamy liczbę losowo wybranych cech jako 3.

Tabela 8: Macierz pomyłek dla zbioru testowego (randomForest)

Rzeczywiste etykietki	Prognozowane etkietki					
	1	2	3	5	6	7
1	18	5	2	0	0	1
2	2	26	2	1	0	1
3	0	1	1	0	0	0
5	0	0	0	2	0	0
6	0	0	0	0	2	0
7	1	1	0	0	0	5

	5-cv	bootstrap	.632+	predykcje na zb. testowym
błąd klasyfikacji (randomForest)	0.234	0.253	0.180	0.239

Tabela 9: Ocena dokładności klasyfikacji

Porównanie wyników

Porównajmy teraz otrzymane wyniki i sprawdźmy, czy zastosowanie rodzin klasyfikatorów zauważalnie zredukowały błąd klasyfikacji.

	5-cv	bootstrap	.632+	predykcje na zb. testowym
1 drzewo	0.346	0.339	0.288	0.324
bagging	0.262	0.283	0.211	0.254
boosting	0.257	0.275	0.201	0.296
randomForest	0.234	0.253	0.180	0.239

Tabela 10: Błąd klasyfikacji dla analizowanych modeli

Zauważamy, że zastosowanie rodzin klasyfikatorów znacznie zredukowało błąd klasyfikacji. Przyjrzyjmy się jeszcze błędom względnym.

	5-cv	bootstrap	.632+	predykcje na zb. testowym
bagging	24.393	16.302	26.894	21.739
boosting	25.743	18.723	30.090	8.696
randomForest	32.494	25.432	37.471	26.087

Tabela 11: Błąd klasyfikacji dla analizowanych modeli (względem 1 drzewa) [%]

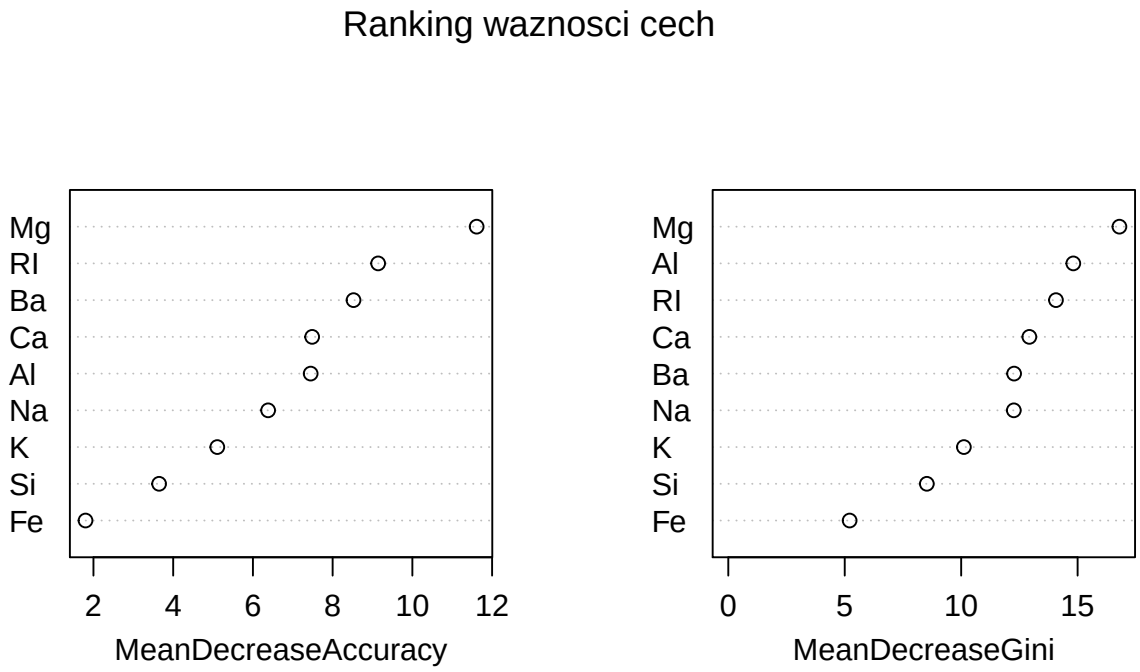
Widzimy, że najlepiej wypadł algorytm `RandomForest`. Sprawdzimy jeszcze jedno kryterium – czas potrzebny na skonstruowanie modelu.

	Bagging	Boosting	randomForest
czas konstruowania modelu [s]	0.045	6.298	0.227

Tabela 12: Czas konstruowania modelu [s]

Wyraźnie więcej czasu zajmuje realizacja jednej funkcji `boosting`. Wnioskujemy stąd, że rodzina klasyfikatorów skonstruowana metodą `RandomForest` jest najlepsza spośród badanej

trójki – skonstruowany model charakteryzuje się najniższym błędem klasyfikacji oraz najkrótszym czasem realizacji. Korzystając z metody `RandomForest` sporządzimy teraz rankng cech.

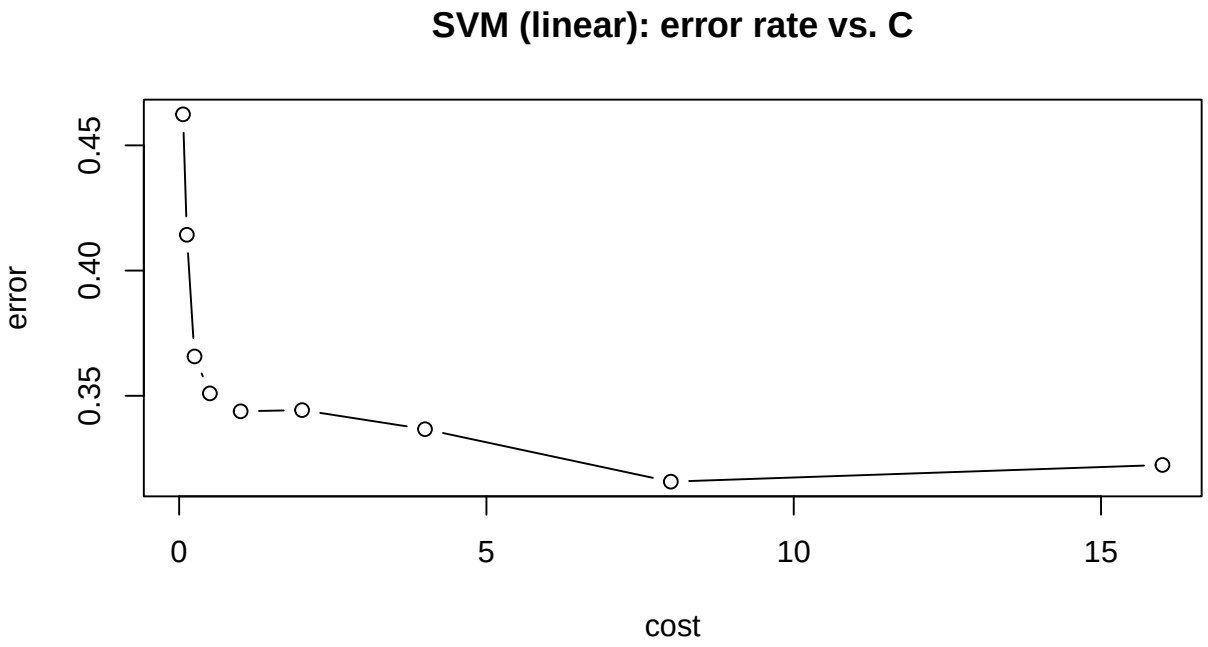


Rysunek 5: Ranking ważności cech

Przypomnijmy, że wykorzystując jedynie metody analizy opisowej wytypowaliśmy żelazo `Fe` oraz potas `Si` jako zmienne o najgorszych zdolnościach dyskryminacyjnych. Ranking ważności cech potwierdził jedynie wysnute wcześniej wnioski.

3.2 Zastosowanie metody wektorów nośnych

Wykorzystamy teraz metodę wektorów nośnych w celu skonstruowania klasyfikatora dla jądra liniowego. Sprawdzimy najpierw, jak na błąd klasyfikacji wpływa parametr `C`.



Rysunek 6: Błąd klasyfikacji w zależności od parametru C

Widzimy, że optymalną wartością C dla jądra liniowego jest 8.

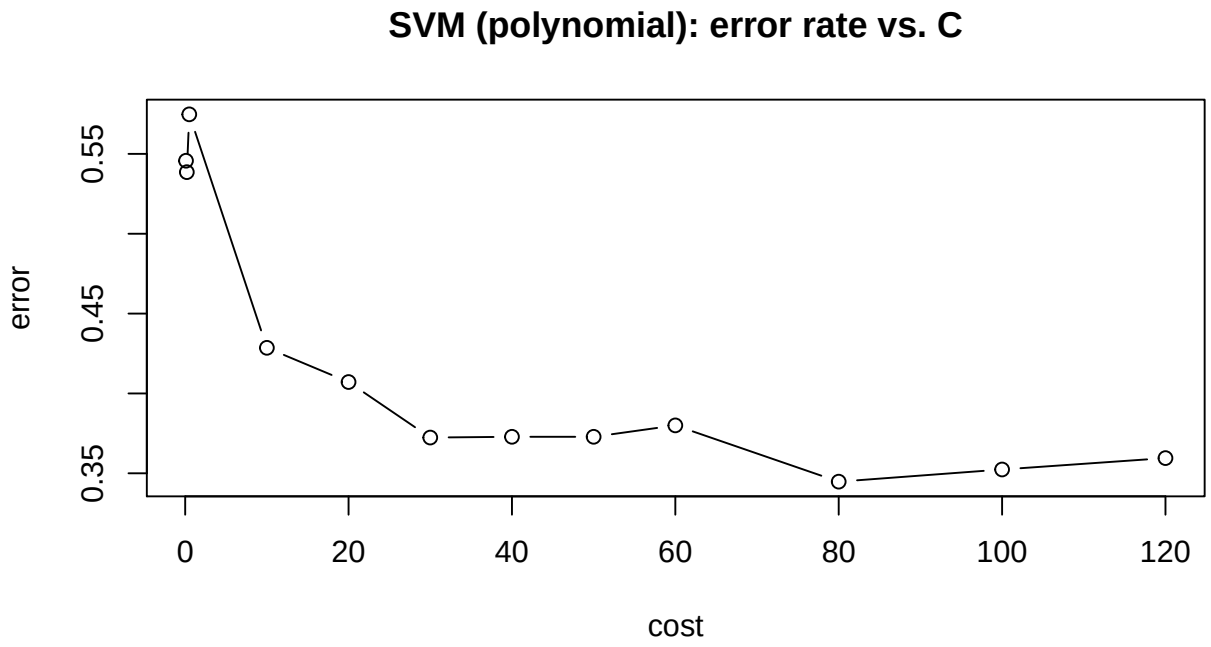
Tabela 13: Macierz pomyłek dla zbioru testowego (SVM (linear))

Rzeczywiste etykietki	Prognozowane etkietki					
	1	2	3	5	6	7
1	15	12	1	0	0	1
2	4	17	2	0	0	0
3	2	0	2	0	0	0
5	0	3	0	3	0	1
6	0	1	0	0	2	0
7	0	0	0	0	0	5

	5-cv	bootstrap	.632+	predykcje na zb. testowym
błąd klasyfikacji (SVM (linear))	0.374	0.390	0.357	0.380

Tabela 14: Ocena jakości klasyfikacji

Widzimy, że otrzymaliśmy gorsze wyniki w porównaniu z np. metodą `RandomForest`. Sprawdzimy, jakie wyniki otrzymamy przy zastosowaniu innego jądra – rozpoczniemy od jądra wielomianowego (polynomial).



Rysunek 7: Błąd klasyfikacji w zależności od parametru C

Widzimy, że optymalną wartością C dla jądra wielomianowego jest 80.

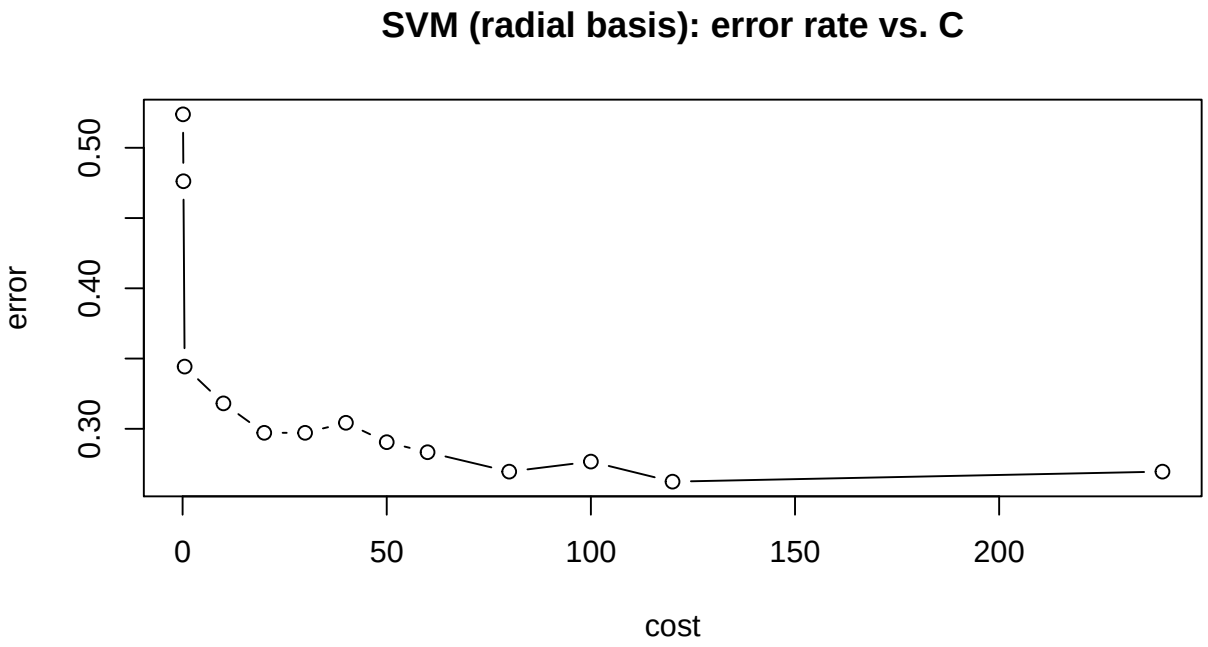
Tabela 15: Macierz pomyłek dla zbioru testowego (SVM (polynomial))

Rzeczywiste etykietki	Prognozowane etkietki					
	1	2	3	5	6	7
1	18	11	4	0	0	0
2	2	16	1	2	1	1
3	1	0	0	0	0	0
5	0	5	0	1	0	1
6	0	1	0	0	1	0
7	0	0	0	0	0	5

	5-cv	bootstrap	.632+	predykcje na zb. testowym
błąd klasyfikacji (SVM (polynomial))	0.346	0.360	0.312	0.423

Tabela 16: Ocena jakości klasyfikacji

Również otrzymaliśmy dość słabe rezultaty – szczególnie wysoki jest błąd predykcji na zbiorze testowym. Analogicznie postępujemy dla jądra `radialnego` oraz `sigmoid`.



Rysunek 8: Błąd klasyfikacji w zależności od parametru C

Widzimy, że optymalną wartością C dla jądra radialnego jest 120.

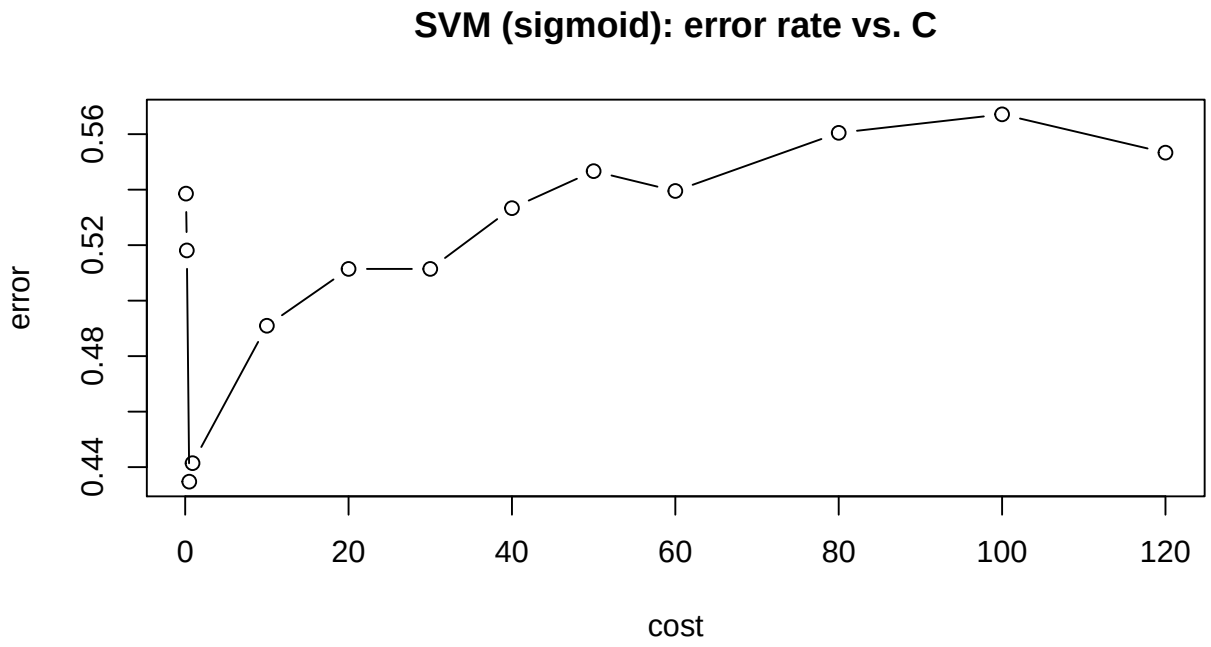
Tabela 17: Macierz pomyłek dla zbioru testowego (SVM (radial))

Rzeczywiste etykietki	Prognozowane etkietki					
	1	2	3	5	6	7
1	16	7	1	0	0	0
2	3	22	1	2	0	1
3	2	1	3	0	0	0
5	0	1	0	1	0	1
6	0	1	0	0	2	0
7	0	1	0	0	0	5

	5-cv	bootstrap	.632+	predykcje na zb. testowym
błąd klasyfikacji (SVM (radial))	0.299	0.292	0.239	0.310

Tabela 18: Ocena jakości klasyfikacji

Pozostało jeszcze zbadać jądro `sigmoid`.



Rysunek 9: Błąd klasyfikacji w zależności od parametru C

Widzimy, że optymalną wartością C dla jądra sigmoid jest 0.5.

Tabela 19: Macierz pomyłek dla zbioru testowego (SVM (sigmoid))

Rzeczywiste etykietki	Prognozowane etkietki					
	1	2	3	5	6	7
1	21	28	3	0	0	1
2	0	4	2	2	2	0
3	0	0	0	0	0	0
5	0	1	0	1	0	1
6	0	0	0	0	0	0
7	0	0	0	0	0	5

	5-cv	bootstrap	.632+	predykcje na zb. testowym
błąd klasyfikacji (SVM (sigmoid))	0.505	0.492	0.490	0.563

Tabela 20: Ocena jakości klasyfikacji

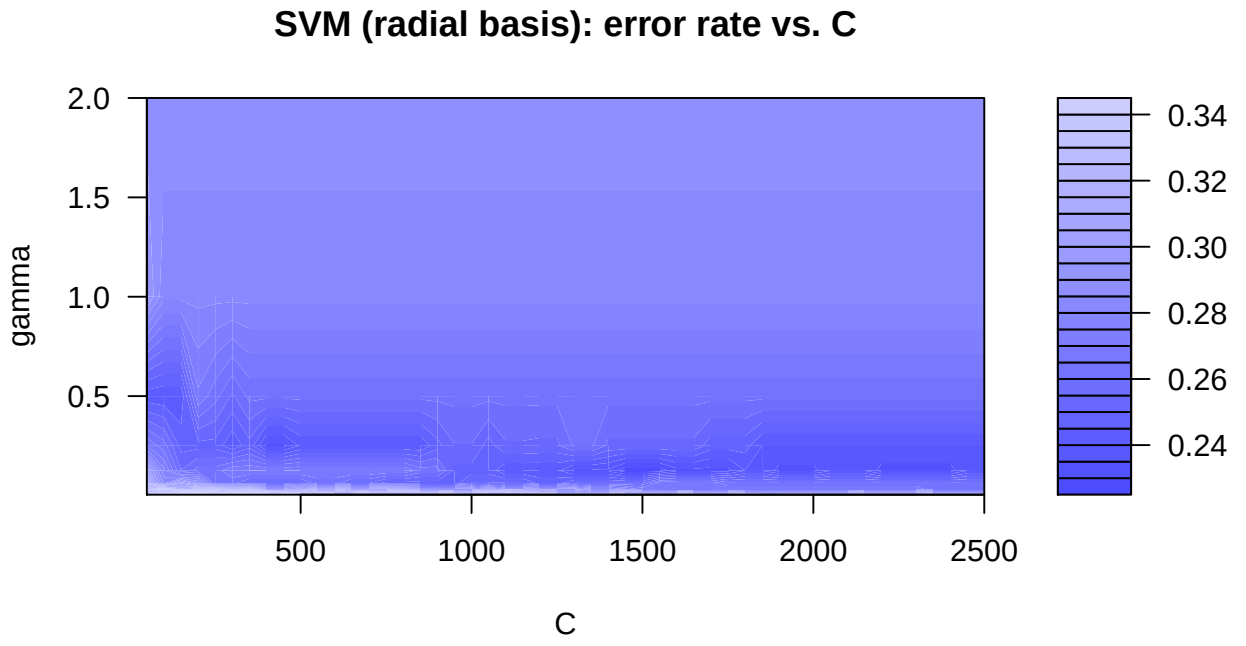
Dla jądra sigmoid otrzymaliśmy rezultaty przypominające wyniki rzutu monetą – błędy klasyfikacji oscylują wokół 0.5. Porównajmy teraz ze sobą wszystkie jądra.

	5-cv	bootstrap	.632+	predykcje na zb. testowym
SVM linear	0.374	0.390	0.357	0.380
SVM polynomial	0.346	0.360	0.312	0.423
SVM radial	0.299	0.292	0.239	0.310
SVM sigmoid	0.505	0.492	0.490	0.563

Tabela 21: Błąd klasyfikacji dla różnych jąder SVM

Widzimy, że wybór jądra ma kluczowy wpływ na błąd klasyfikacji. Porównując jądro radialne z jądrem sigmoid widzimy, że w tym drugim wystąpił niemal dwa razy większy błąd klasyfikacji.

Możemy również wysnuć wniosek, że dla naszych danych najbardziej odpowiednim jądrem jest jądro radialne. W związku z tym postaramy się „dostroić" otrzymany model z tym jądrem w celu zmniejszenia błędu klasyfikacji. Zoptymalizujemy w tym celu zarówno parametr `C`, jak i parametr γ.



Rysunek 10: Błąd klasyfikacji w zależności od parametru C oraz gamma

Otrzymujemy, że najlepszą parą parametrów jest C równe 1450 oraz γ równa 0.125.

	5-cv	bootstrap	.632+	predykcje na zb. testowym
błąd klasyfikacji (SVM (radial optimized))	0.276	0.300	0.234	0.282

Tabela 23: Ocena jakości klasyfikacji

Zbadajmy teraz względny błąd klasyfikacji.

Tabela 22: Macierz pomyłek dla zbioru testowego (SVM (radial optimized))

Rzeczywiste etykietki	Prognozowane etkietki					
	1	2	3	5	6	7
1	17	5	2	0	0	0
2	2	24	1	2	0	1
3	2	2	2	0	0	0
5	0	1	0	1	0	1
6	0	0	0	0	2	0
7	0	1	0	0	0	5

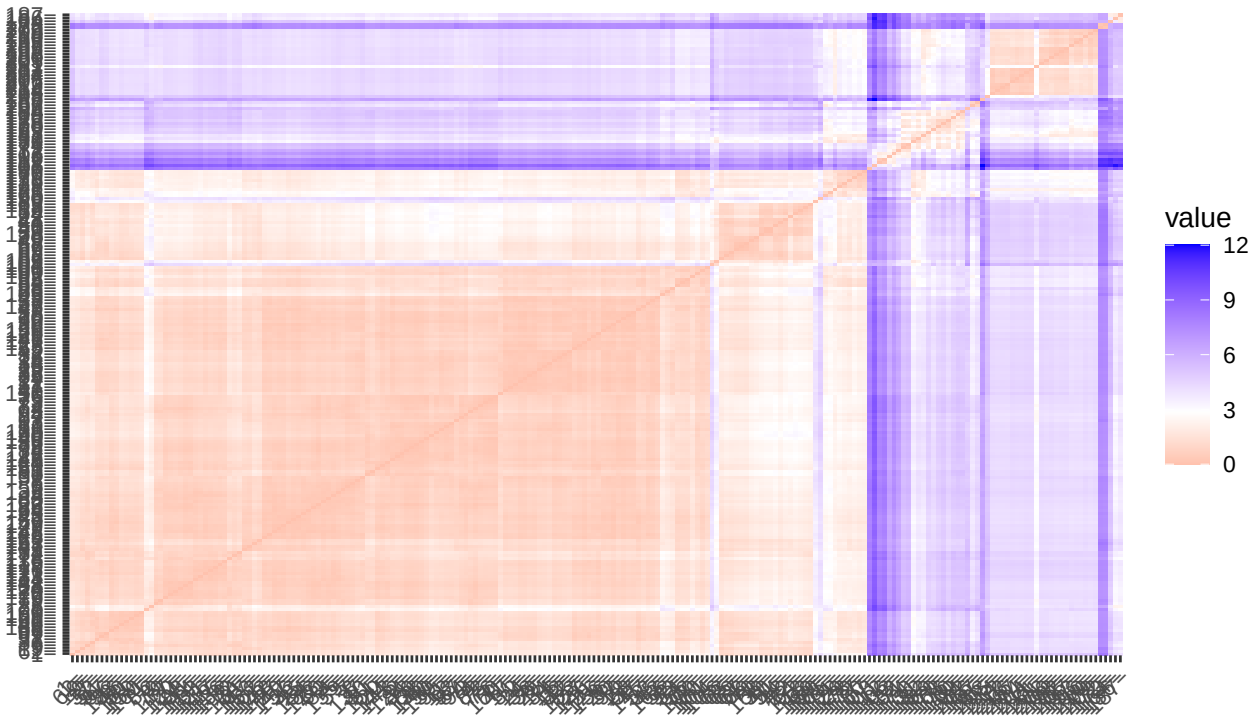
	5-cv	bootstrap	.632+	predykcje na zb. testowym
SVM (optimized)	7.813	-2.525	1.936	9.091

Tabela 24: Błąd klasyfikacji dla analizowanych modeli (radial optimized względem radial) [%]

Widzimy, że otrzymaliśmy zauważalnie mniejszy błąd predykcji na zbiorze testowym, a także błąd zmierzony przy użyciu metody 5 cross validation. Nieznaczne różnice pojawiły się natomiast badając błąd metodą bootstrap oraz .632+.

3.3 Ocena oraz porównanie jakości grupowania dla różnych algorytmów analizy skupień

W tym segmencie, ponownie z wykorzystaniem danych `Glass` z pakietu `mlbench`, weźmiemy pod lupę zagadnienie analizy skupień. Postaramy się porównać algorytmy oraz wyłonić ten, który najlepiej poradził sobie z naszymi danymi. Przygotujemy dane do analizy – usuwamy z ramki danych zmienną grupującą `Type`. Ponieważ wartości zmiennych są mierzone tą samą jednostką, nie ma potrzeby standaryzacji danych.



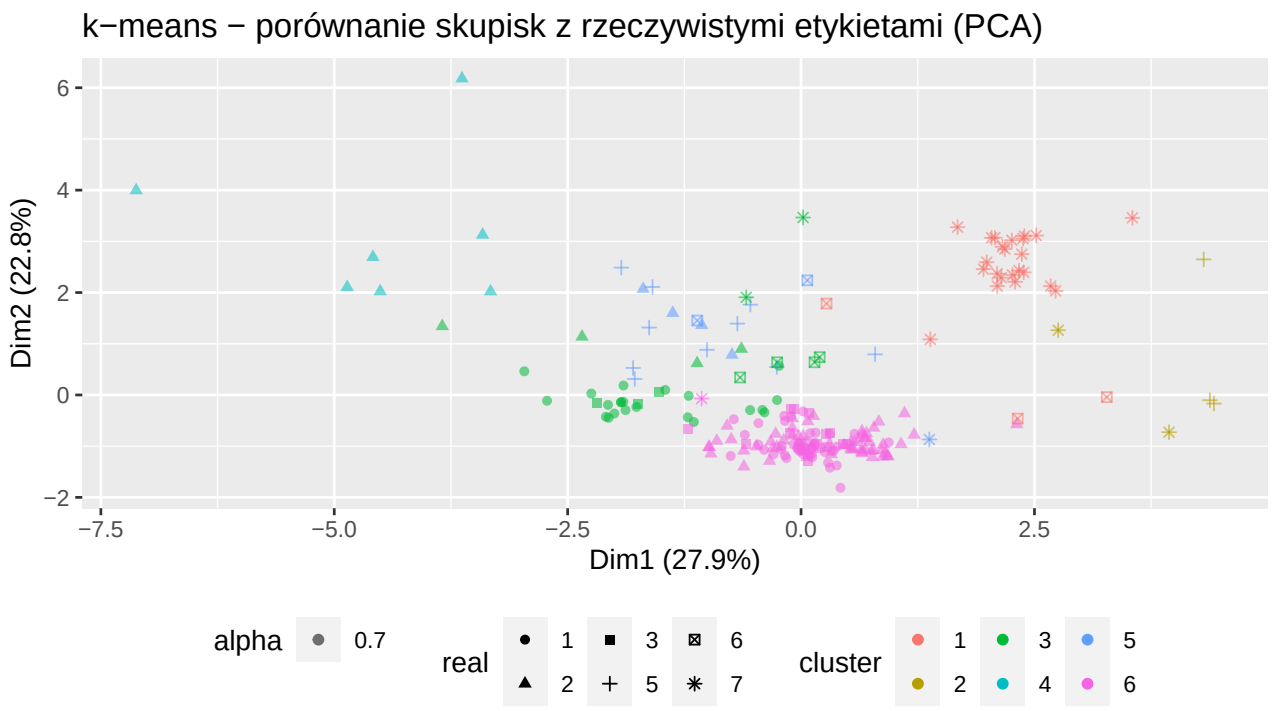
Rysunek 11: Macierz odległości dla danych Glass

Analizę przeprowadzimy dla następujących algorytmów:

1. k-means,
2. PAM,
3. AGNES,
4. DIANA.

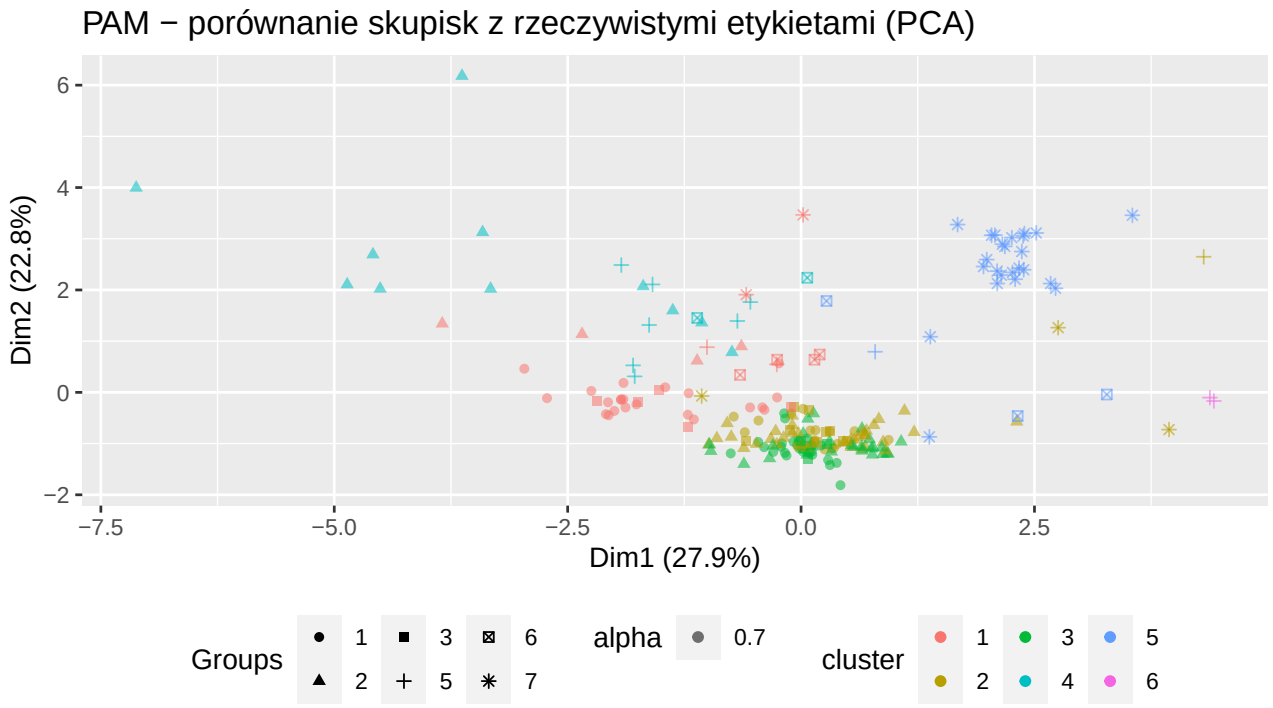
Przyjmujemy liczbę skupień jako 6 – rzeczywistą liczbę klas.

Algorytm k-means



Rysunek 12: k-means - wizualizacja wyników z wykorzystaniem PCA

Otrzymaliśmy dobrze odseparowane skupiska, natomiast dość poprawnie została wykryta tylko klasa 7, a więc ta najbardziej różniąca się od pozostałych zmiennych, dla których błąd jest wyraźnie większy.

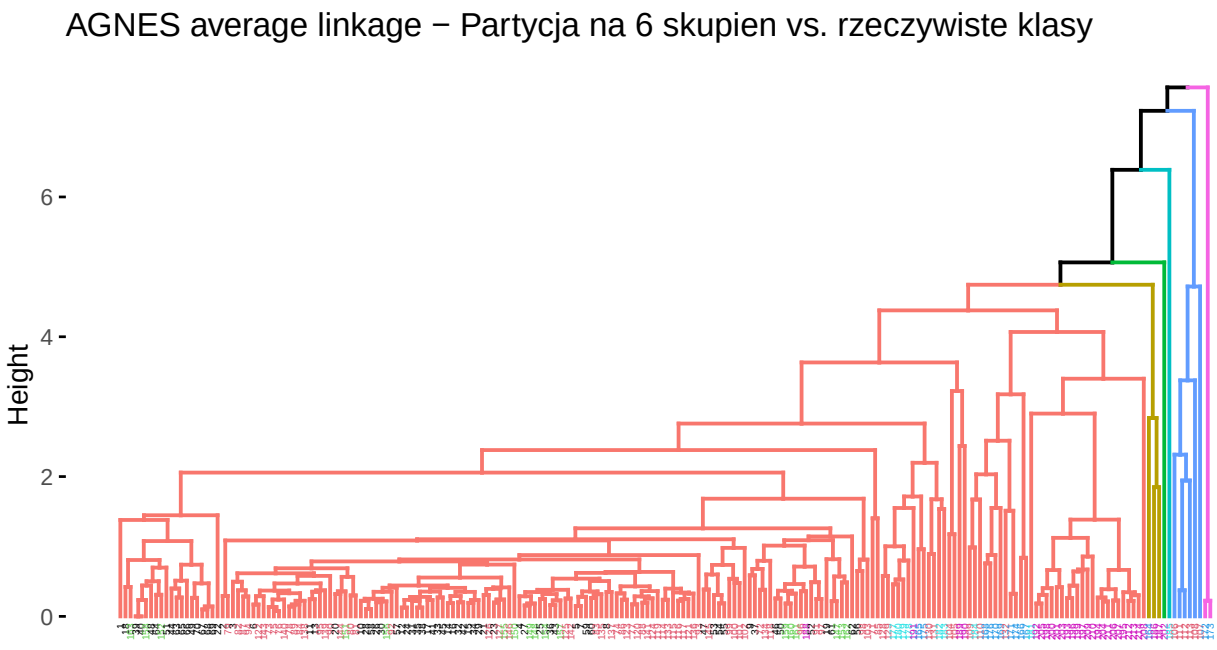


Rysunek 13: PAM - wizualizacja wyników z wykorzystaniem PCA

Ponownie otrzymujemy widocznie odseparowane skupiska z wyjątkiem klastrów 2 i 3, które

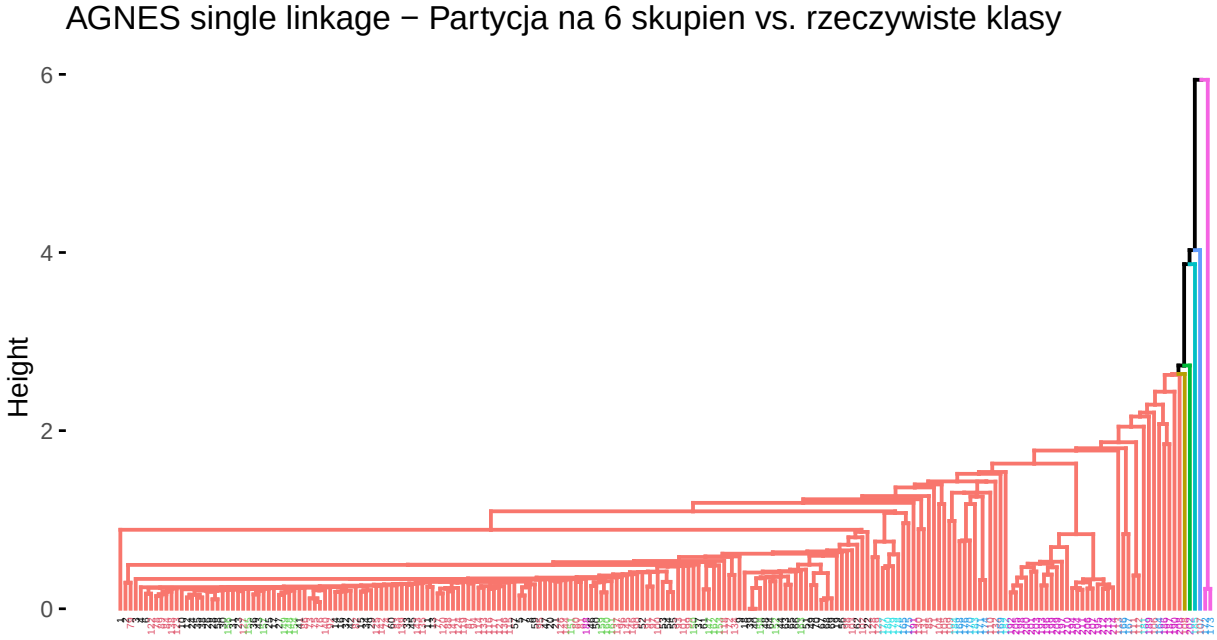
wyraźnie się przenikają. Ponownie otrzymujemy dość dobre wyłonienie tylko typu 7.

Sprawdźmy teraz, jak radzą sobie metody hierarchiczne – AGNES oraz DIANA.



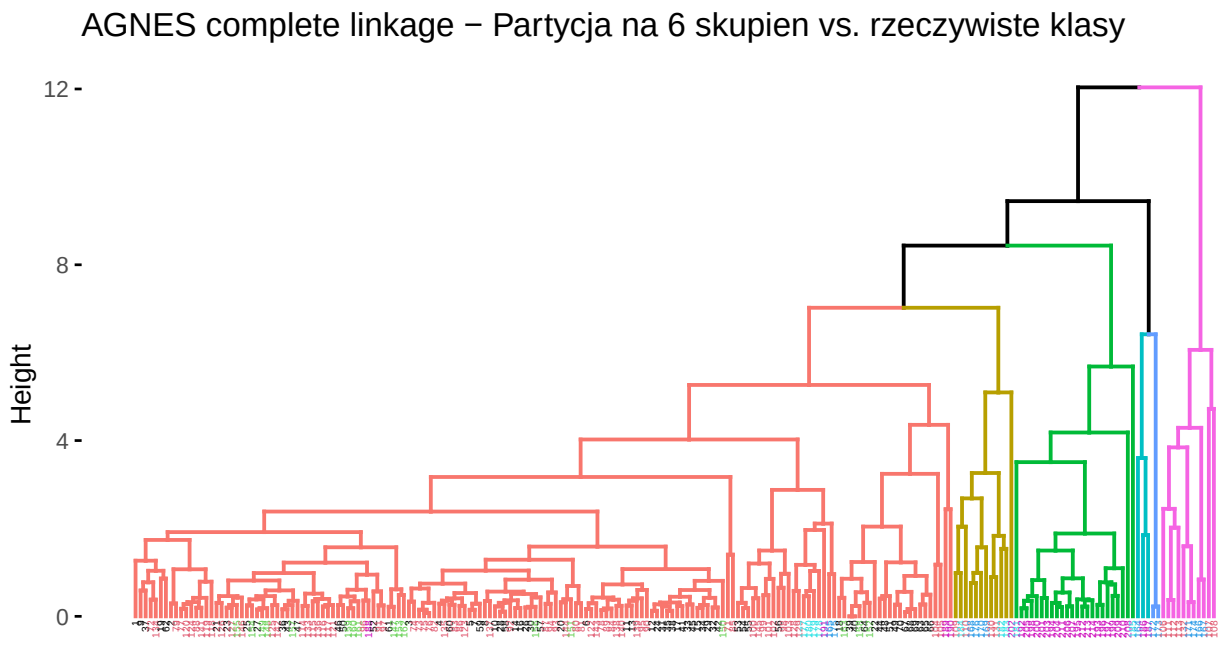
Rysunek 14: AGNES average linkage - dendrogram

Większość obserwacji została objęta w ramach jednego skupiska, zaledwie 13 obserwacji natomiast zostało podzielone na aż 5 grup.



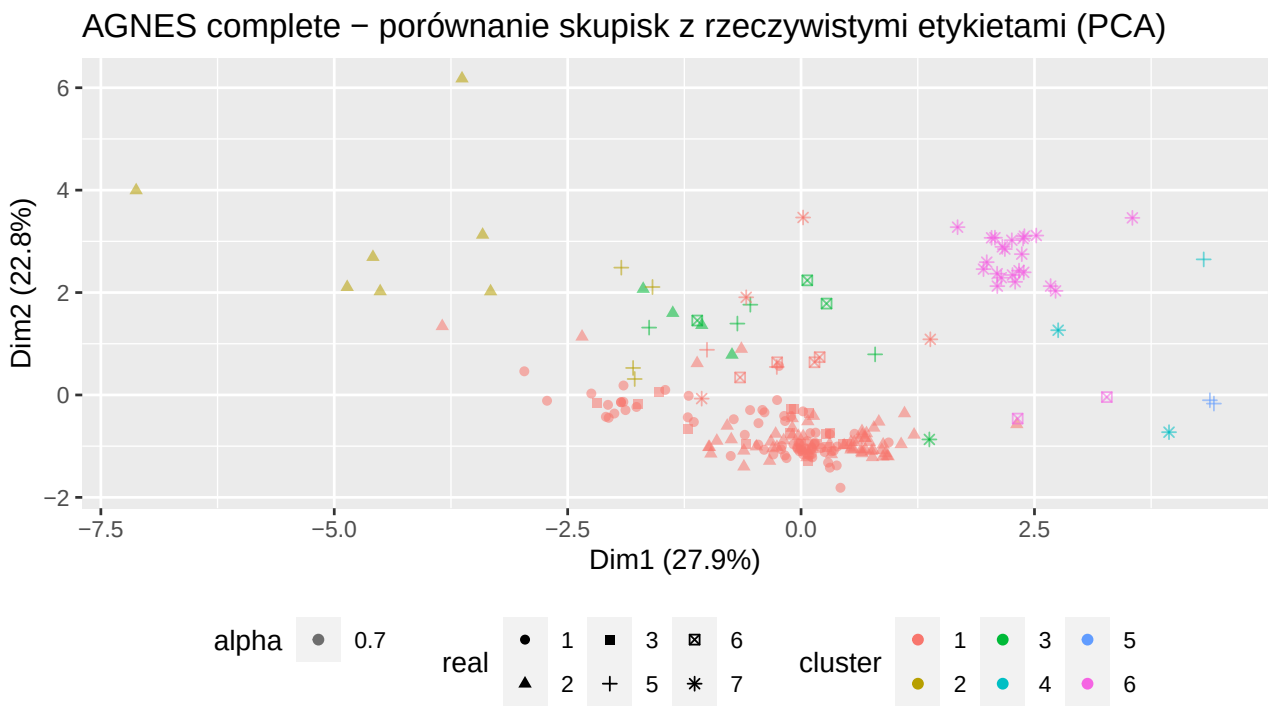
Rysunek 15: AGNES single linkage - dendrogram

Otrzymaliśmy bardzo podobny dendrogram – natomiast tym razem przed wcieleniem do najliczniejszej grupy uchroniło się jedynie 6 obserwacji.



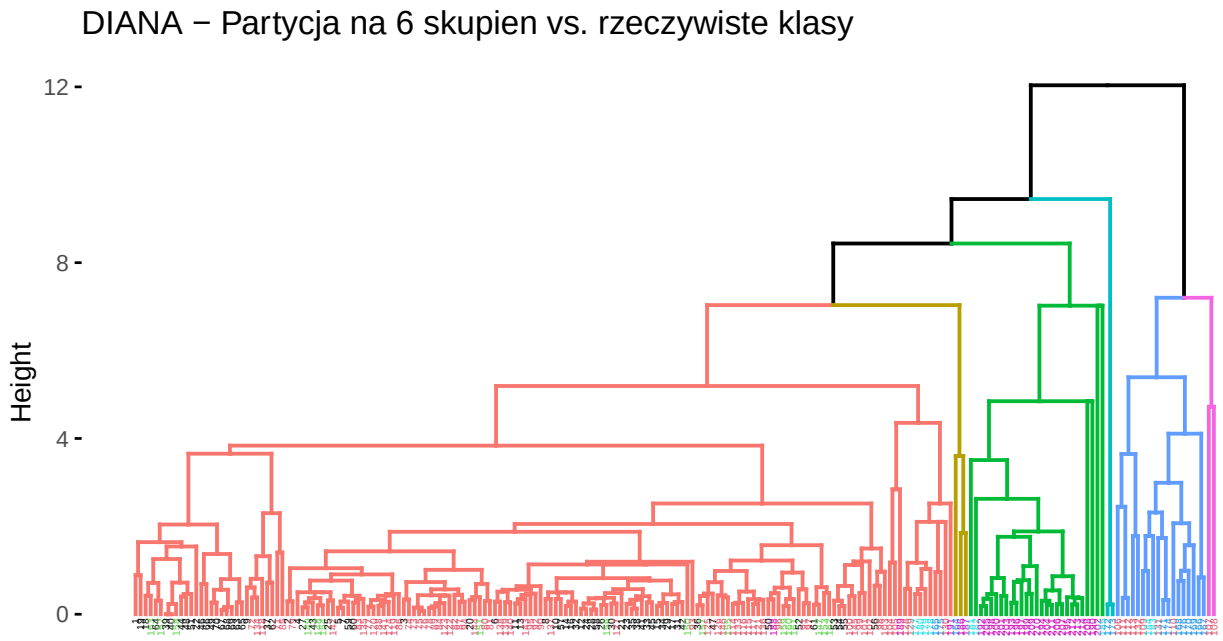
Rysunek 16: AGNES complete linkage - dendrogram

Dendogram dla metody `complete linkage` wyraźnie różni się od poprzednich. Również mamy jedno dominujące skupisko, ale w pozostałych klastrach znalazła się zdecydowanie większa liczba obserwacji.



Rysunek 17: AGNES - wizualizacja wyników z wykorzystaniem PCA

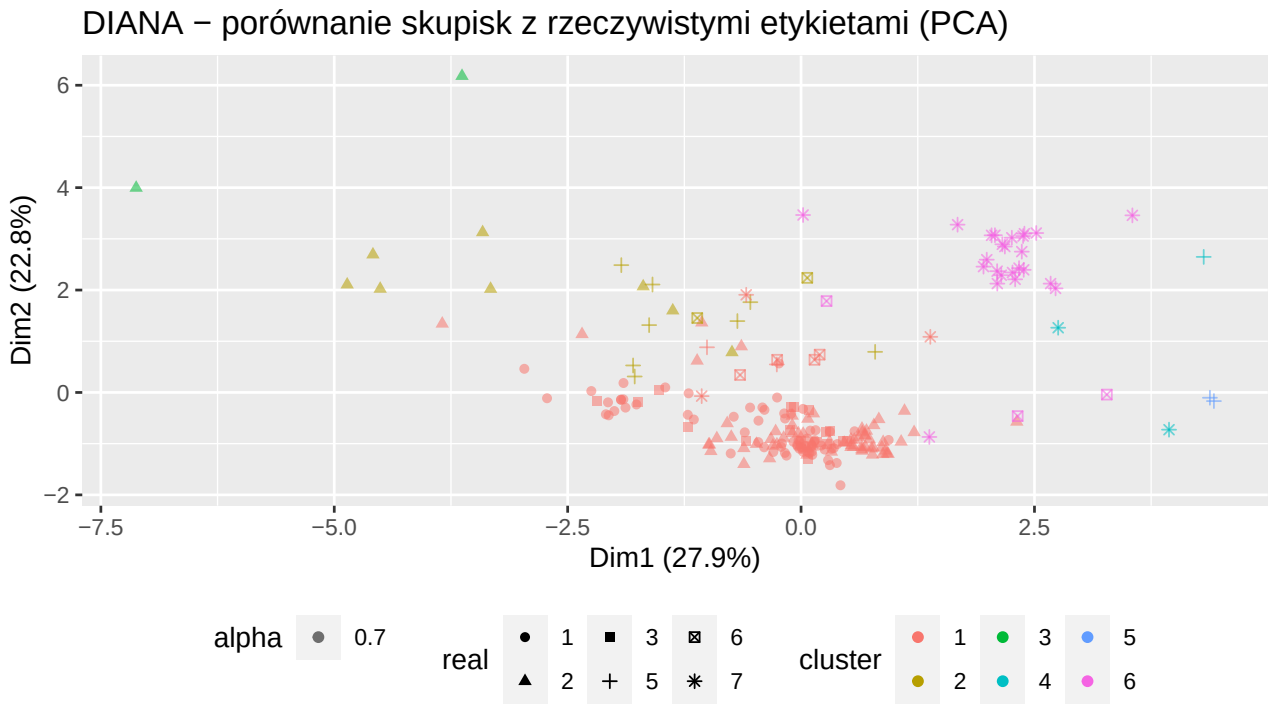
Również wyraźnie widać, że skupienia zostały wyraźnie odseparowane oraz tak samo jak w przypadku metod grupujących jedynie typ 7 został wyłoniony w sposób przypominający wyjściowe dane.



Rysunek 18: DIANA - dendrogram

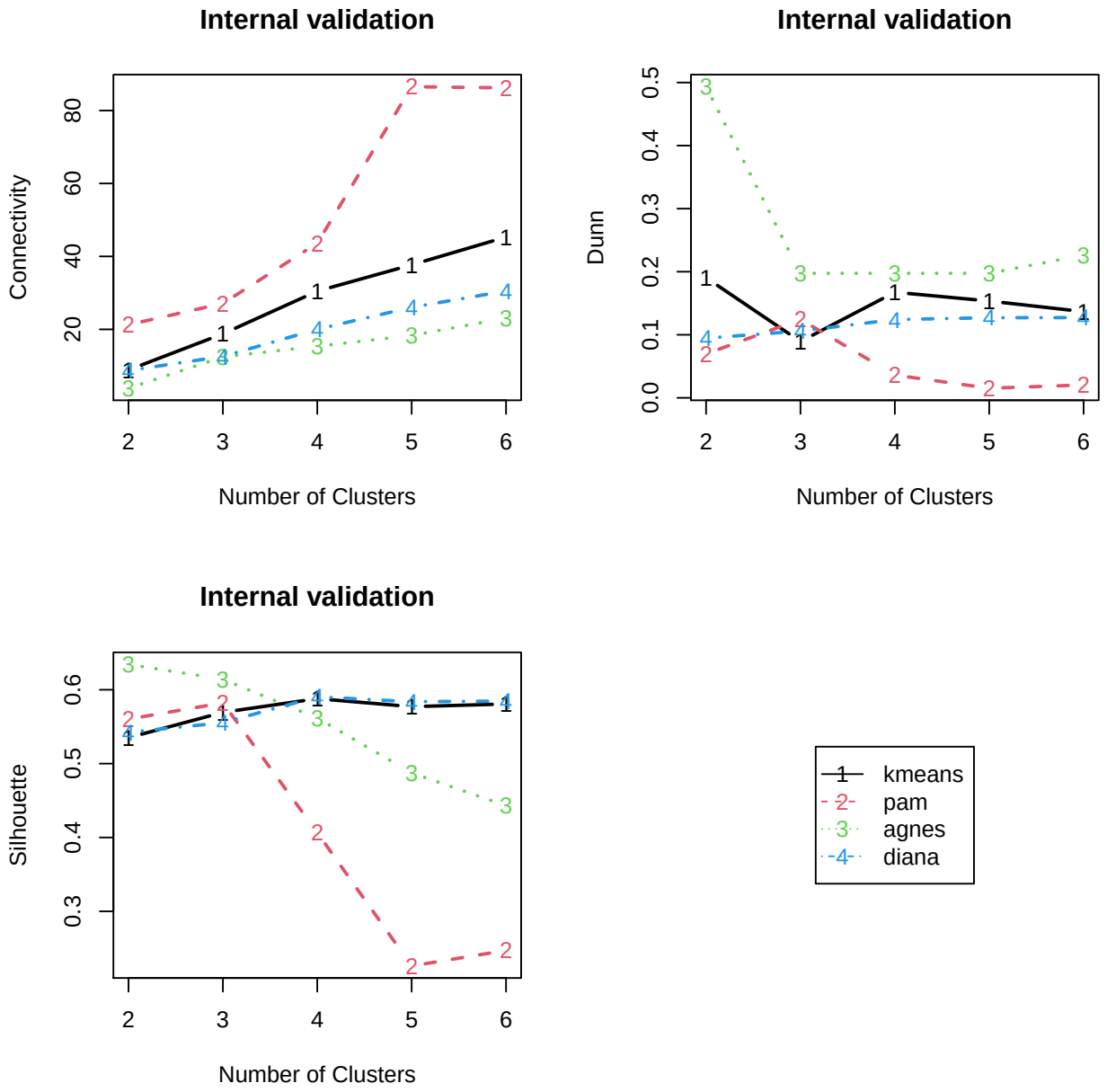
Dla algorytmu `DIANA` dendogram dość wyraźnie wskazuje na dominację trzech klastrów,

ale ponownie tylko typ 7 został wyłoniony dość dokładnie. Podobne wnioski uzyskamy przy wizualizacji z wykorzystaniem PCA.



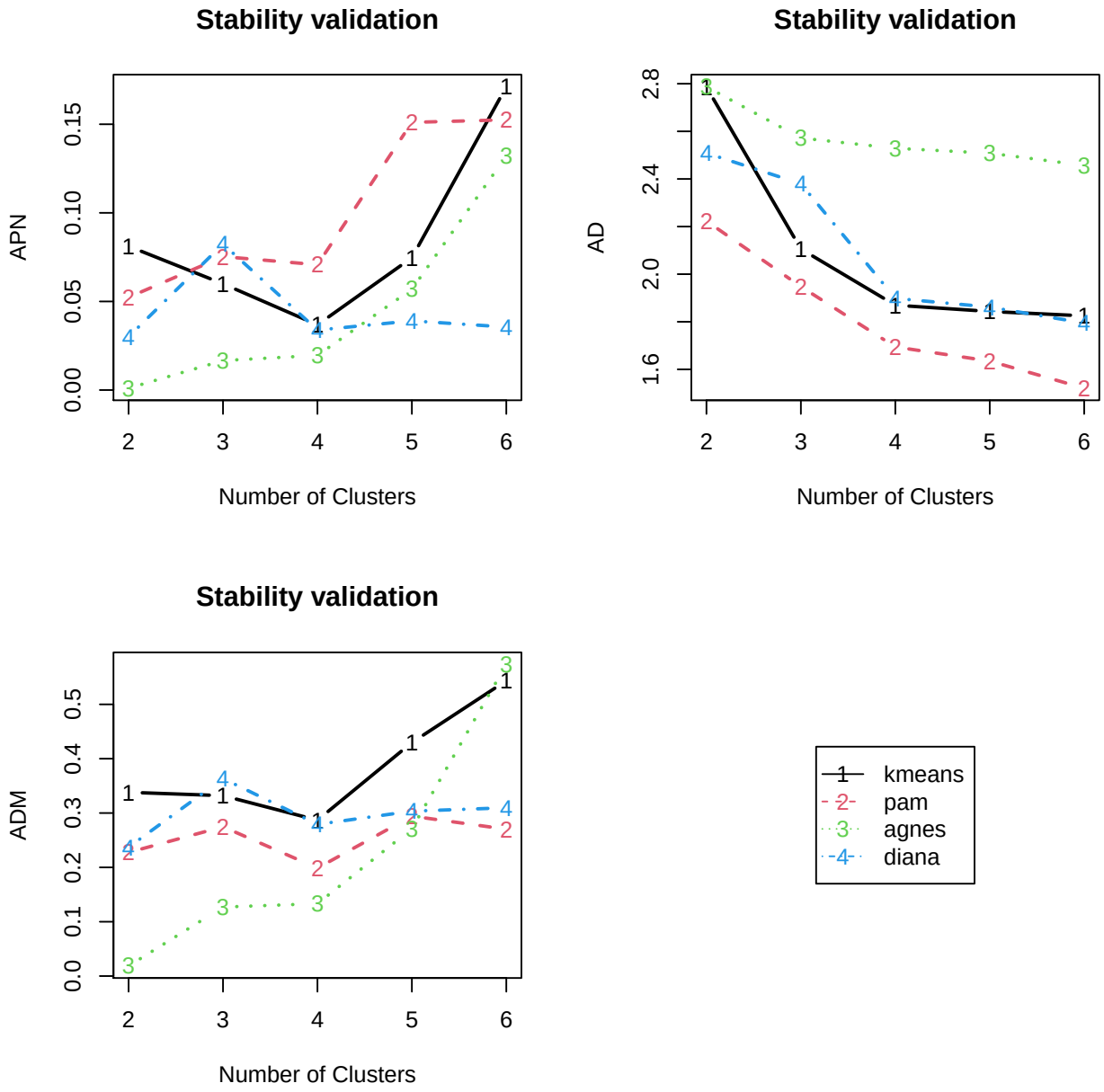
Rysunek 19: DIANA - wizualizacja wyników z wykorzystaniem PCA

Zajmiemy się teraz oceną jakości grupowania i postaramy się wyłonić najlepszy algorytm oraz najbardziej optymalną liczbę klastrów. Wykorzystamy w tym celu wskaźniki wewnętrzne.



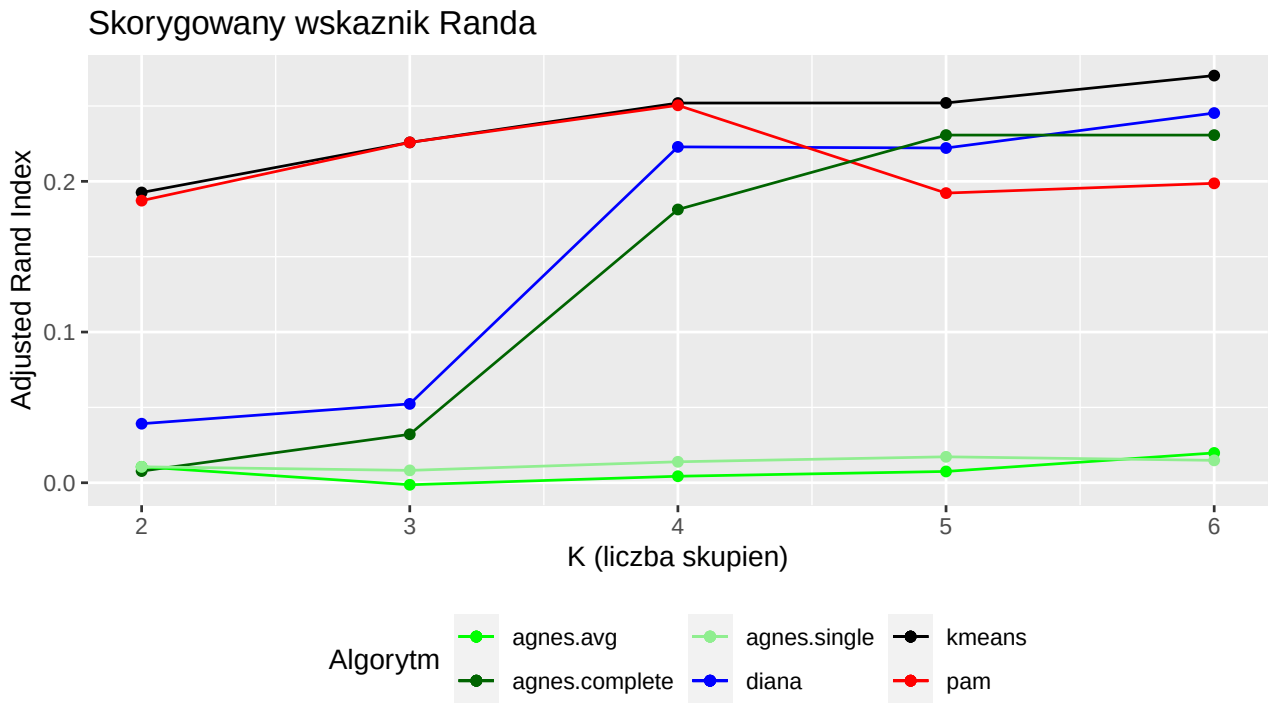
Rysunek 20: Wskaźniki wewnętrzne

Widzimy, że algorytm `AGNES` osiąga najniższą wartość wskaźnika Connectivity oraz najwyższą wartość wskaźnika Dunn'a, a także najwyższą wartość wskaźnika Silhouette dla $k \in \{2, 3\}$.



Rysunek 21: Badanie stabilności algorytmów

W badaniu stabilności również najlepiej wypadł algorytm `AGNES` – ma najniższy wskaźnik APN oraz ADM. Uzyskał natomiast najwyższy wskaźnik AD, jednak, ponieważ we wszystkich pozostałych testach wypadł najlepiej, wyciągamy wniosek, że ten algorytm jest najlepszy z badanej czwórki. Wykorzystamy teraz wskaźnik zewnętrzny w celu określenie optymalnej liczby klastrów – skorygowany wskaźnik Randa.



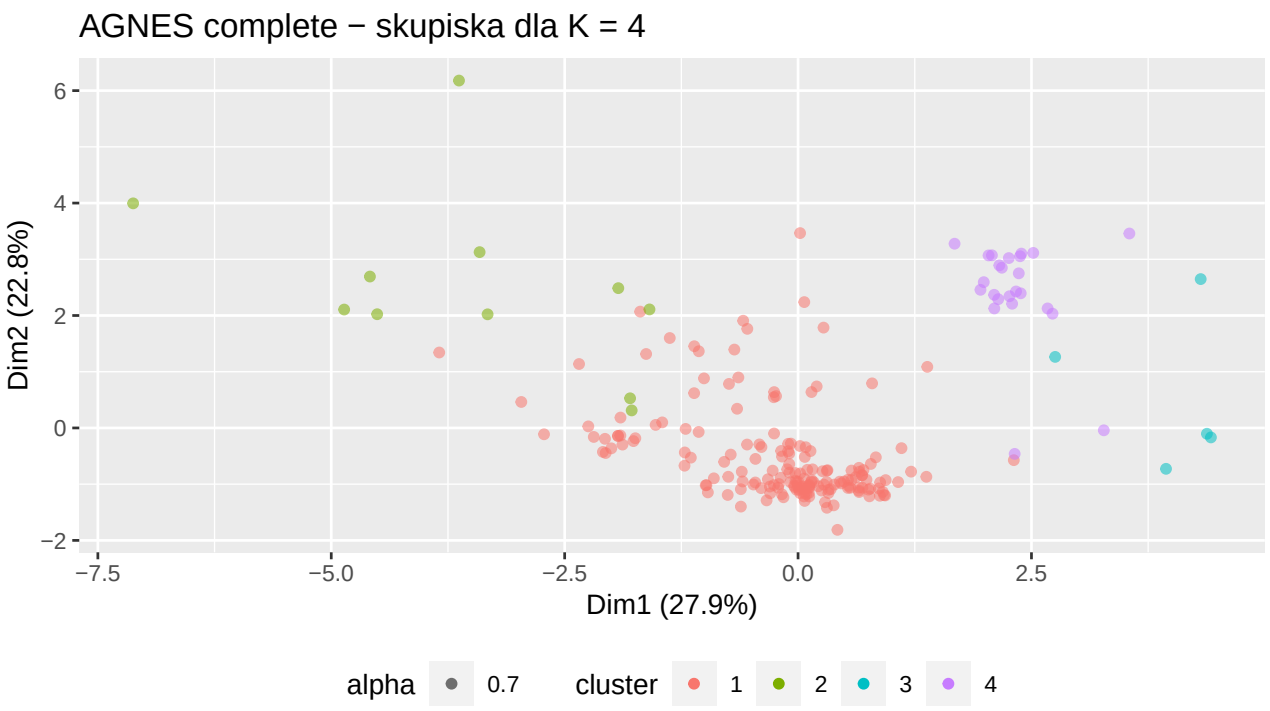
Rysunek 22: Skorygowany wskaźnik Randa dla różnej liczby skupień

Dla algorytmu `AGNES (complete)` widzimy duży wzrost wskaźnika Randa dla liczby skupień równej 4. Porównując z wynikami otrzymanymi z użyciem wskaźników wewnętrznych możemy wywnioskować, że to właśnie liczba klastrów równa 4 jest dobrym kompromisem (nieznaczny wzrost wskaźników Connectivity, ADM, APN, a także maleje wskaźnik AD oraz wskaźnik Dunna, jednak wciąż jest on najwyższy spośród badanych algorytmów).

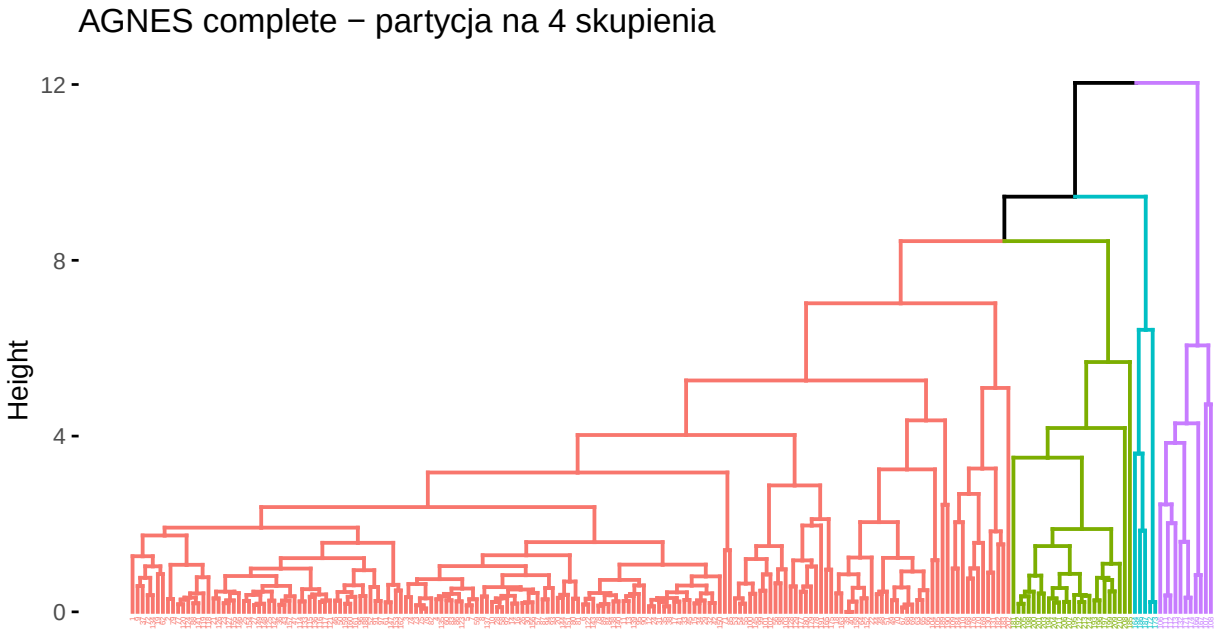
	K-MEANS	PAM	AGNES	DIANA
diameter	7.02	9.45	7.02	7.20
separation	1.18	0.34	1.26	0.89
size C.1	5.00	40.00	174.00	165.00
size C.2	159.00	124.00	11.00	20.00
size C.3	21.00	20.00	5.00	2.00
size C.4	29.00	30.00	24.00	27.00

Tabela 25: Własności skupisk, k = 4

Ostatecznie wyciągnęliśmy wniosek, że najbardziej optymalnym algorytmem jest `AGNES (complete)` z liczbą klastrów 4. Zwizualizujmy wyniki dla tego algorytmu przy użyciu metody PCA oraz sporządzając dendrogram.

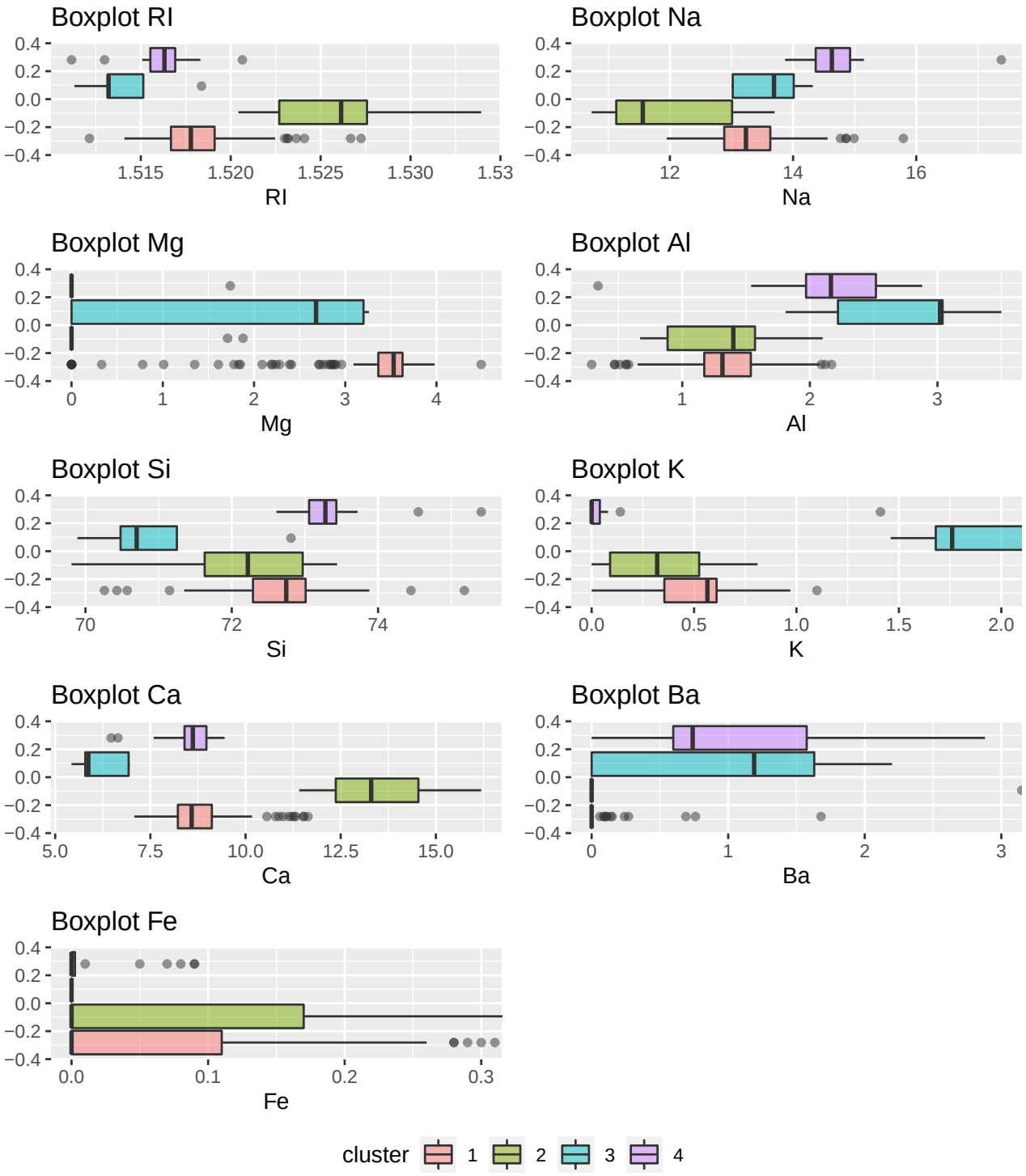


Rysunek 23: AGNES - wizualizacja wyników z wykorzystaniem PCA, K = 4

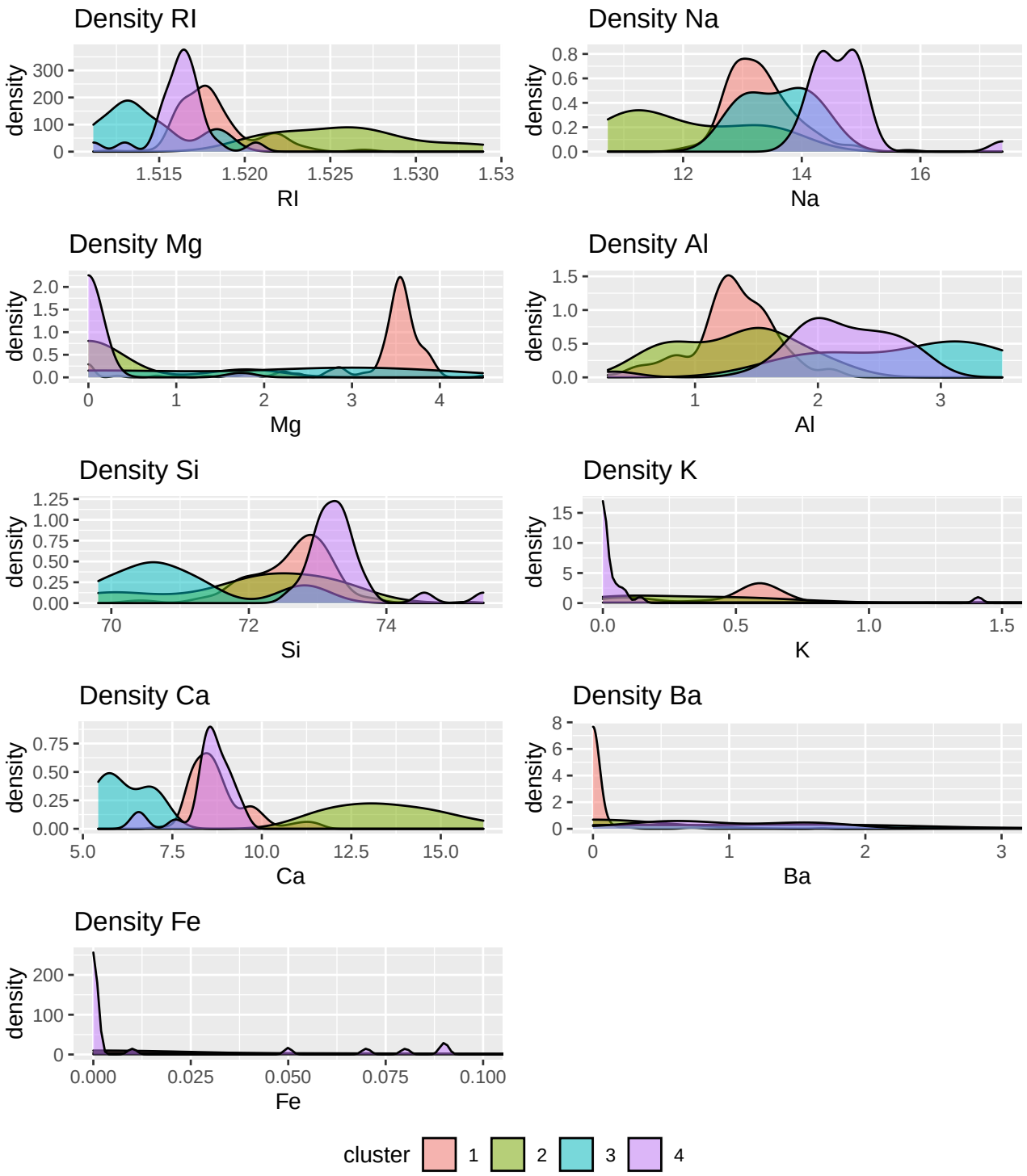


Rysunek 24: AGNES - wizualizacja wyników z wykorzystaniem dendogramu, K = 4

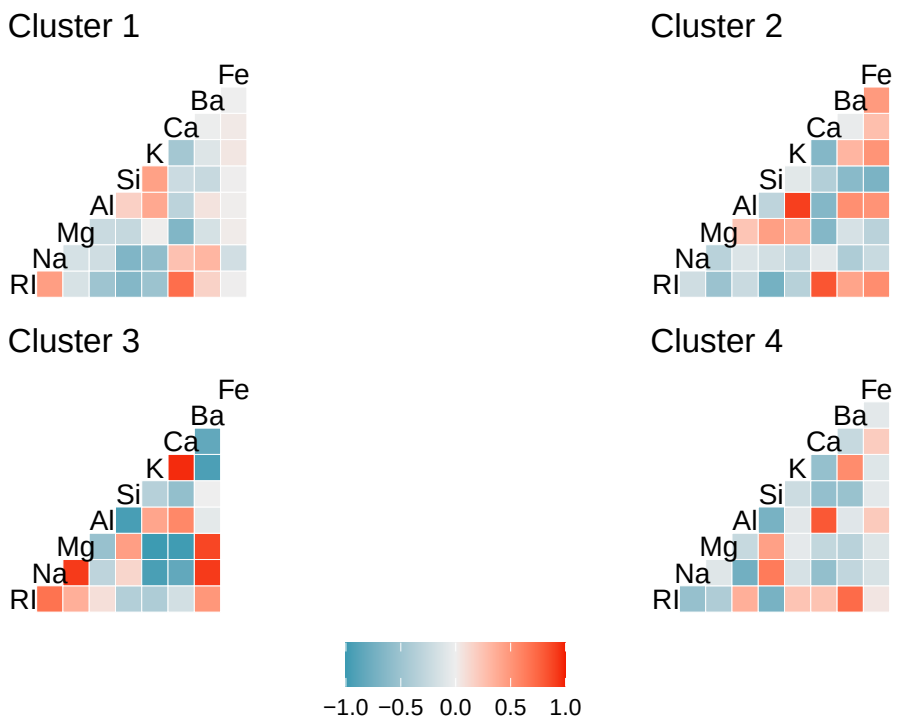
Spróbujmy teraz scharakteryzować obserwacje, które znalazły się w kolejnych skupieniach.



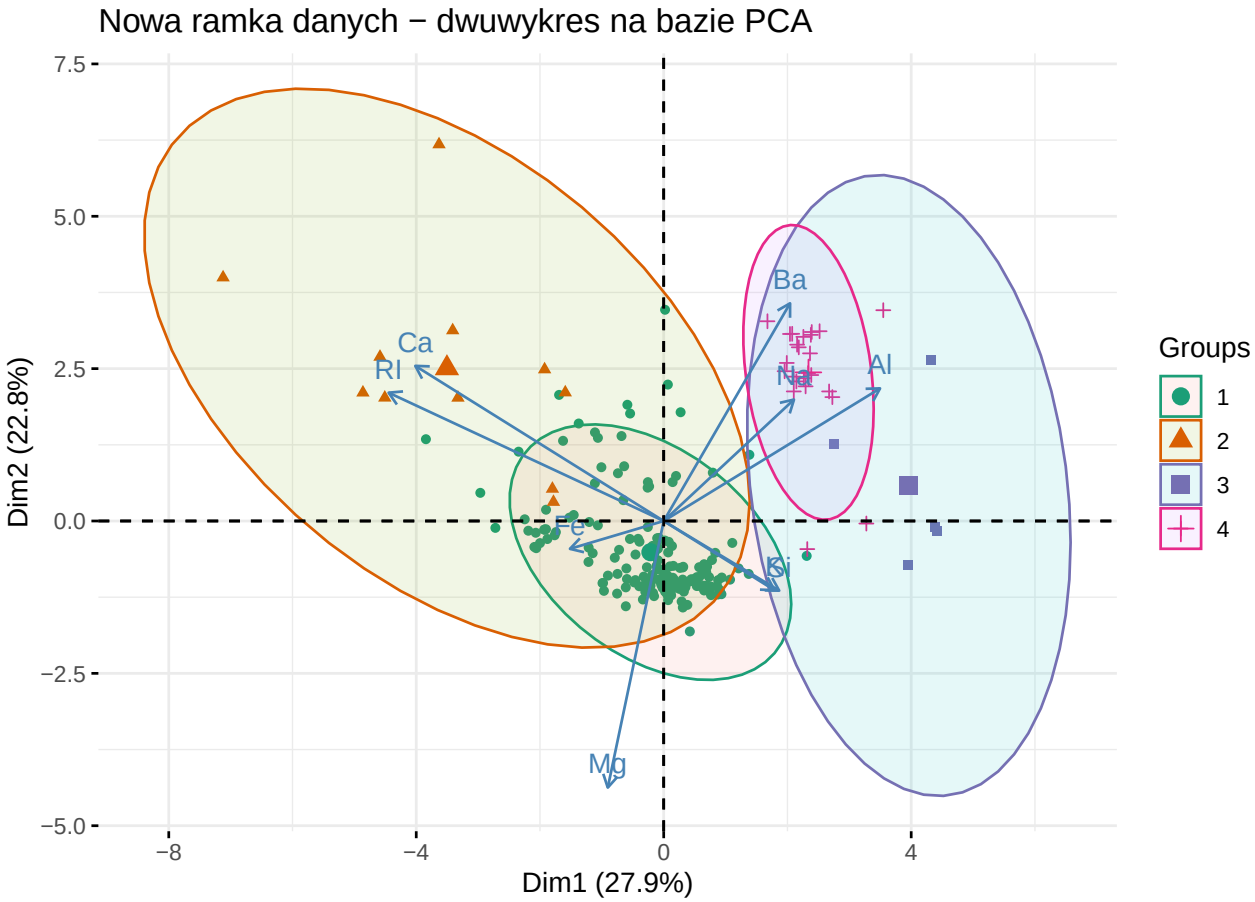
Rysunek 25: Nowa ramka danych - wykresy pudełkowe



Rysunek 26: Nowa ramka danych - wykresy gęstości



Rysunek 27: Nowa ramka danych - macierze korelacji



Rysunek 28: Nowa ramka danych - PCA

Spoglądając na powyższe wykresy, możemy wyciągnąć następujące wnioski dotyczące uzyskanych skupisk:

1. typ 1 charakteryzuje się większą zawartością magnezu oraz mniejszą zawartością baru,
2. typ 2 charakteryzuje się większą zawartością wapienia oraz wyższym współczynnikiem załamania (a także wysoką korelacją między tymi dwiema cechami) oraz niższą zawartością krzemu,
3. typ 3 charakteryzuje się niską korelacją pomiędzy zawartością wapienia oraz współczynnikiem załamania, wysoką korelacją pomiędzy zawartością wapienia i potasu oraz zawartością sodu i magnezu, a także wysoką zawartością baru oraz potasu,
4. typ 4 charakteryzuje się dużą zawartością baru i krzemu, niewielką zawartością potasu oraz większą zawartością glinu niż pozostałe typy.

4 Podsumowanie

Poniżej wypunktujemy najważniejsze wnioski, jakie można wyciągnąć z przeprowadzanych analiz:

- najlepszym algorytmem rodzin klasyfikatorów dla naszych danych okazał się być `RandomForest` – najlepiej zwiększył dokładność klasyfikacji oraz najkrócej trwa realizacja jednego użycia funkcji,
- korzystając z metody wektorów nośnych, wybór jądra ma kluczowy wpływ na dokładność klasyfikacji,
- dostrojenie modelu uzyskanego metodą wektorów nośnych pozwala na zauważalne zmniejszenie błędu klasyfikacji,
- w zagadnieniu analizy skupień wykorzystanie wskaźników wewnętrznych i zewnętrznych pozwoliło na wybór optymalnej liczby klastrów oraz najlepszego algorytmu,
- klastry uzyskane z wykorzystaniem metod analizy skupień znacznie różniły się od grup w wyjściowych danych.