Praca Domowa 2 Raport

Jan Borowski

Contents

Wstęp

(W całym raporcie używam określenia mój oczywiście chodzi o moją implementację algorytmu spektralnego) W poniższym raporcie dokonam porówania mojej implementacij algorymtu spektralnego (z parametrem M = k,k+2,k+6 gdzie k = liczba skupień) z nastepujacymi algorytmami :

- Wszystkimi algorytmami hierarchicznymi z funkcji `hclust()`.
- Algorytmem *Genie* z pakietu `genie`.
- Algorytmem *hdbscan* (z parametrem minPts=5,10,20,50,100) z pakietu `dbscan`.

Porównania będą dotyczyły różnych aspektów działania algorytmów takich jak dokładność i czas. Dane będą przedstawaine w postaci różnego rodzaju wykresów. Surowe informacje znajdują się w przesłanych plikach .csv. Do zbierania danych używałem zbiorów benchmarkowych oraz dołączonego do nich zestawu moich zbiorów przedstawionych w pliku Testy.pdf.

Analiza hclust

Zacznę od analizy poszczególnych metod funkcij **hclust** aby wyznaczyć najlepsze metody do dalszych testów. Testy wykonywałem na 13 losowo wybranych zbiorach.
Najpierw przeprowadzę analizę średniego czasu działania poszczególnych metod :

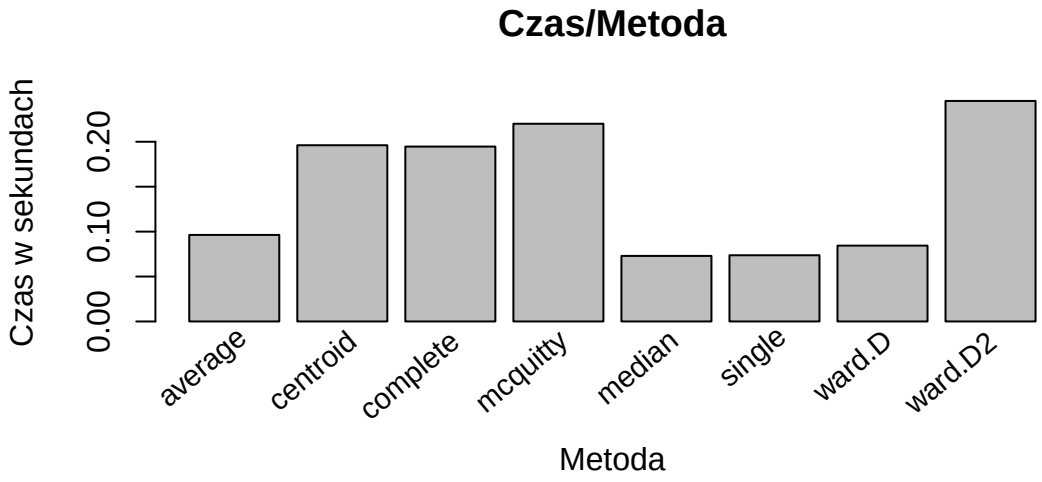

Figure 1: Zależność czasu od metody

Teraz dokonam analizy działania algorytmów porównując indeks Randa i Fowlkesa–Mallowsa :

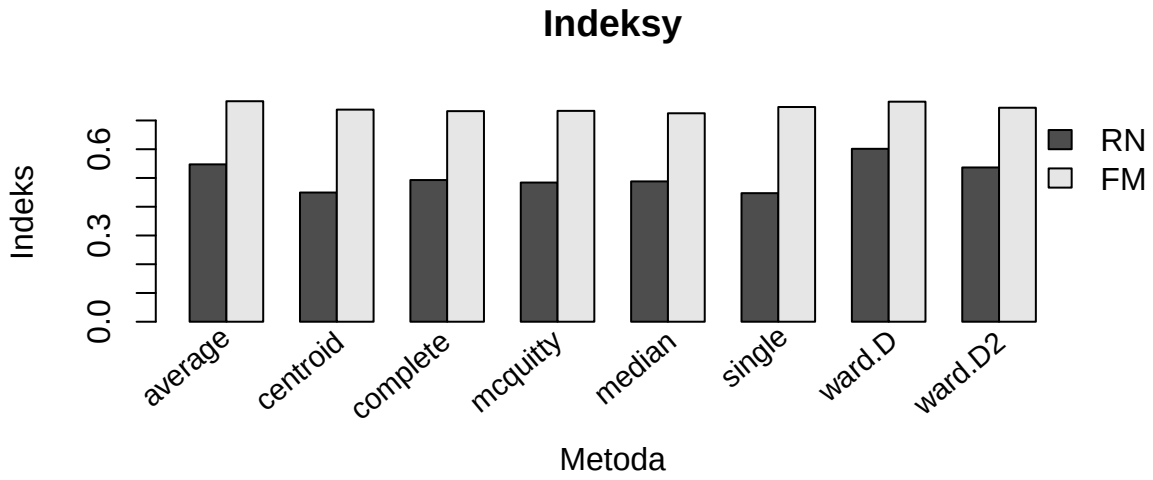

Figure 2: Zależność parametrów jakościowych od metody

Jak widać najszybsza metdoa to **median** a najwolniesza **ward.D2**. Te dwie metody wybrałem do dalszych testów wydajnościowych i porówania z innymi algorytamai . Należy jednak zauważyć , że choć w przypadku indeksu Fowlkesa–Mallowsa nie widać znaczących róznic. Metoda **ward.D** uzuskała najwyższy indeks Randa.

Analiza parametrów dodatkowych

Najpierw sprawdzę dla jakiego M moja implementacja działała najkuteczniej (Testy przeprowadzanę na 46 zbiorach):

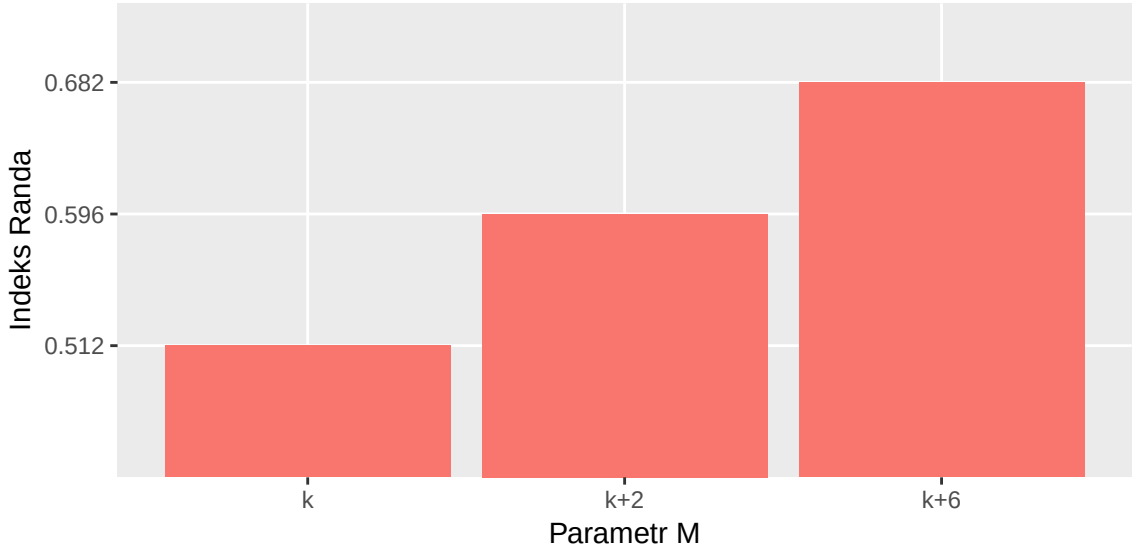

Figure 3: Zależność dokładności od parametru M

Jak widać moja implementacja uzyskała najlepszą średnią skuteczność przy M=k+6.
Sprawdzę teraz funkciję **hdbscan**:

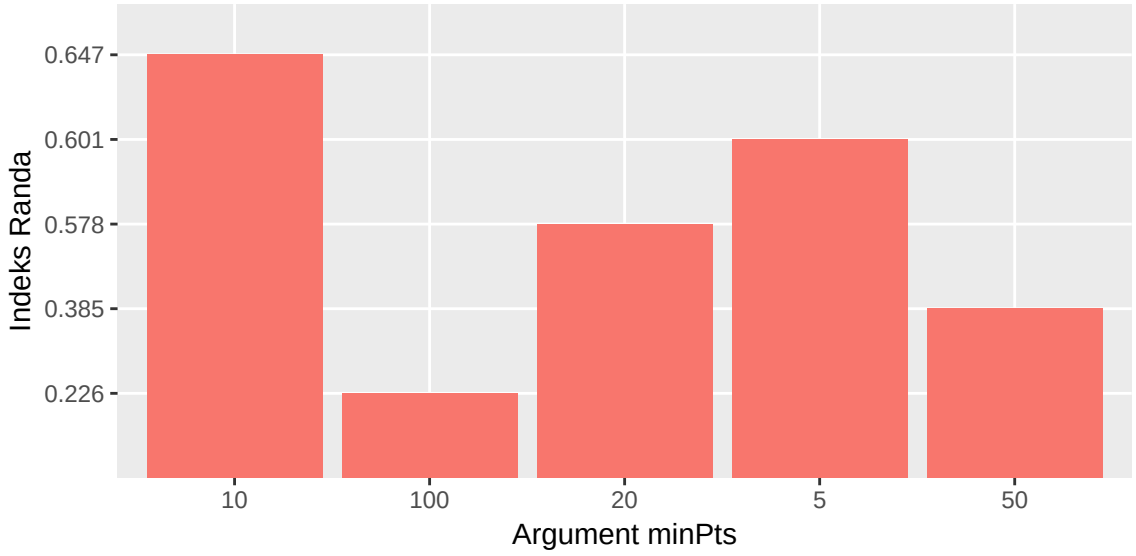

Figure 4: Zależność dokładności od parametru minPts

Jak widać funkcja **hdbscan** działa średnio najdokładniej przy parametrze minPts = 10.

Porównanie wydajności

Przetstawię średnie czasy wykonania algorytmów dla **Genie** z parametrem **thresholdGini** = 0.3 oraz **Genie** z **thresholdGini** = 1 , wybranych metod funkcij **hclust** ,funkcij **hdbscan** oraz mojej implementacij algorytmu spektralnego. W przypadku funkcij **hdbscan** oraz mojej implementacij gdzie występuję dodatkowy parametry wybierany jest ten dla którego czas wykonania był najmniejszy(Testy wykonywane na 46 zbiorach testowych):

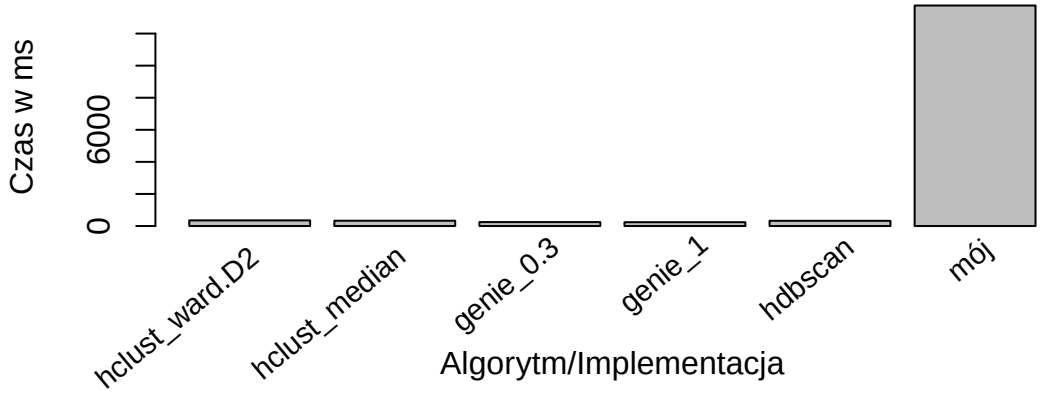

Figure 5: Zależność czasu od algorytmu/implementacij

Jak widać zdecydowanie najwolniejsza jest moja implementacja więc narazie odrzucę ją z dalszej analizy wydajności.
Przedstawię teraz analizę czasów działania pozostalych algorytmów:

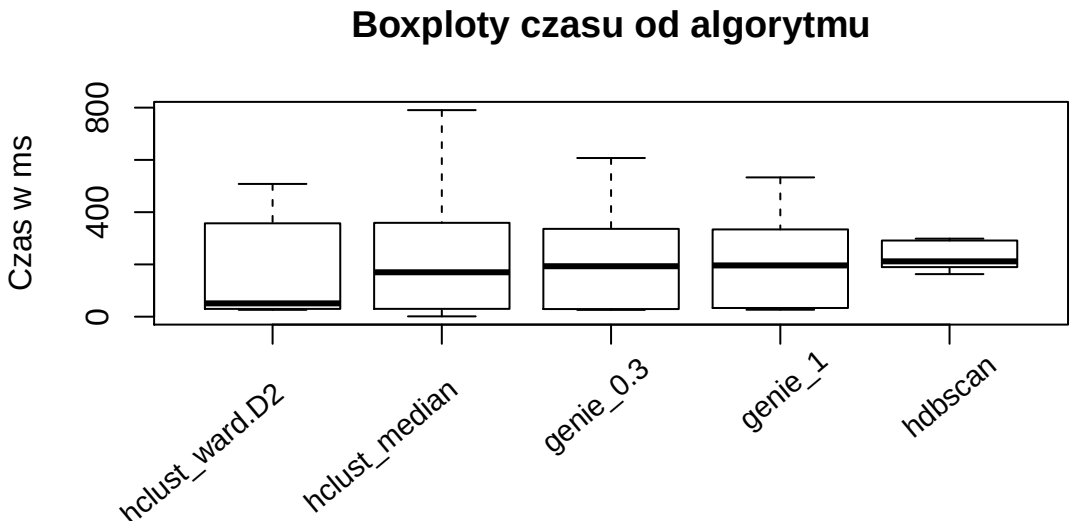

Figure 6: Zależność czasu od algorytmu

Jak widać po rozpatrzeniu danych można po piersze zauważyć ,że najwolniejsza metoda **hclust** okazała się najszybsza na większej próbce (benchmarki wykonywane z argumentem times = 10) . Potem mniej wiecej równo jeśli chodzi o medianę **genie** i metodę **median** z funckij **hclust** najwolniejszy okazał się algorytm **hbdscan** ale w jego wypatku mamy doczynienia z bardzo małym rozrzutem. Tym niemniej można ułożyć algorytmy w kolejności od najszybszego do najwolniejszego :
moja implementacja >hdbscan >hclust_median>genie>hclust_ward.D2

Analiza dokładności

W tej sekcij pod uwagę będzie brana metoda **ward_D** z fuknkcij **hclust** .Najpierw sprawdzę średni indeks Randa w zależności od algorytmu. Jak poprzednio w przypadku metod z dodatkowymi parametrami wybiorę ten o najwyższym indeksie (Testy wykonywane na 46 zbiorach testowych) :

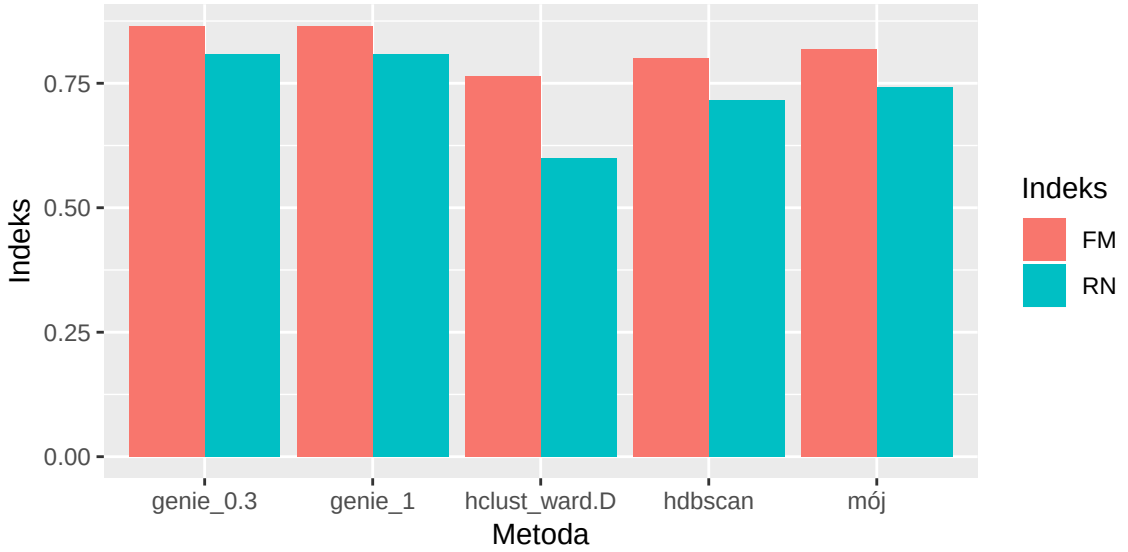

Figure 7: Zależność indeksów od metody

W tym wypadku wyraźnie widać że najlepiej działajacy algorytm to **genie**. Co oznacza ,że dokonuje on najlepszych podziałów.

Przedstawię teraz zależność średniego indeksu Randa od średniego czasu wykonania algorytmu (czas w ms) :

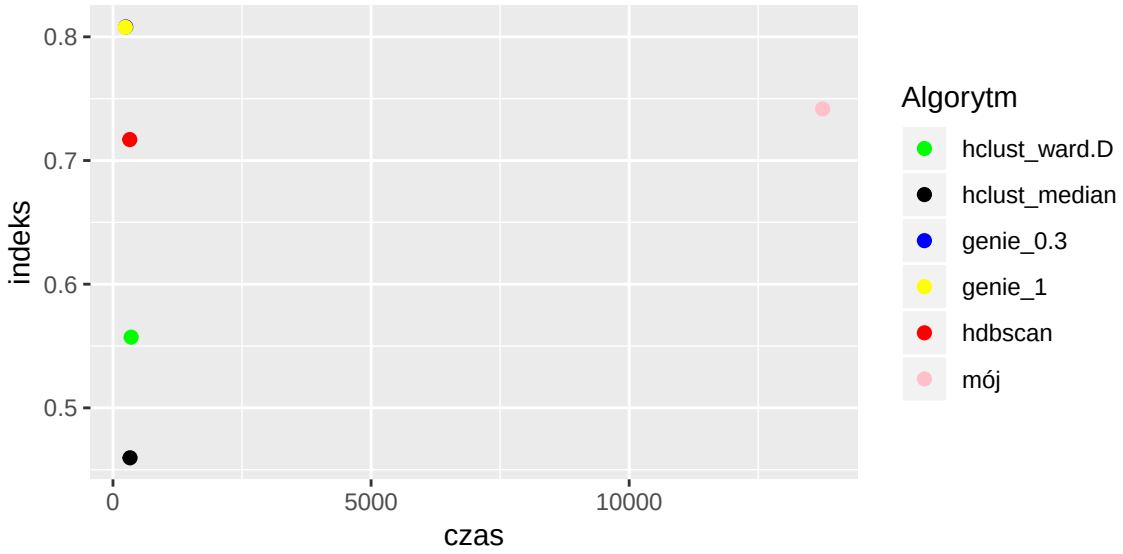

Figure 8: Zależność jakości od czasu 1

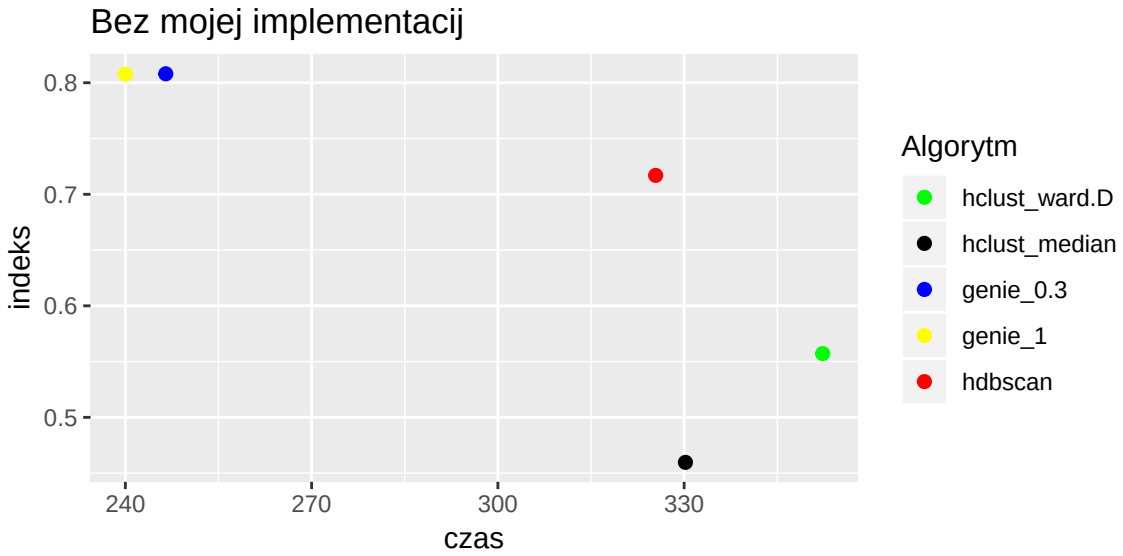

Figure 9: Zależność jakości od czasu 2

Dopiero ten wykres pokazuje skuteczność **genie** które uzyskuję bardzo wysoką jakość w krótkim czasie .

Wpływ standaryzacij

W tej części sprawdzę wpływ standaryzacij na dokładność działania algorytmów (testy przeprowadanę na 14 losowo wybranych zbiorach).

Najpierw sprawdzę wpływ standaryzacji na Indeks Randa:

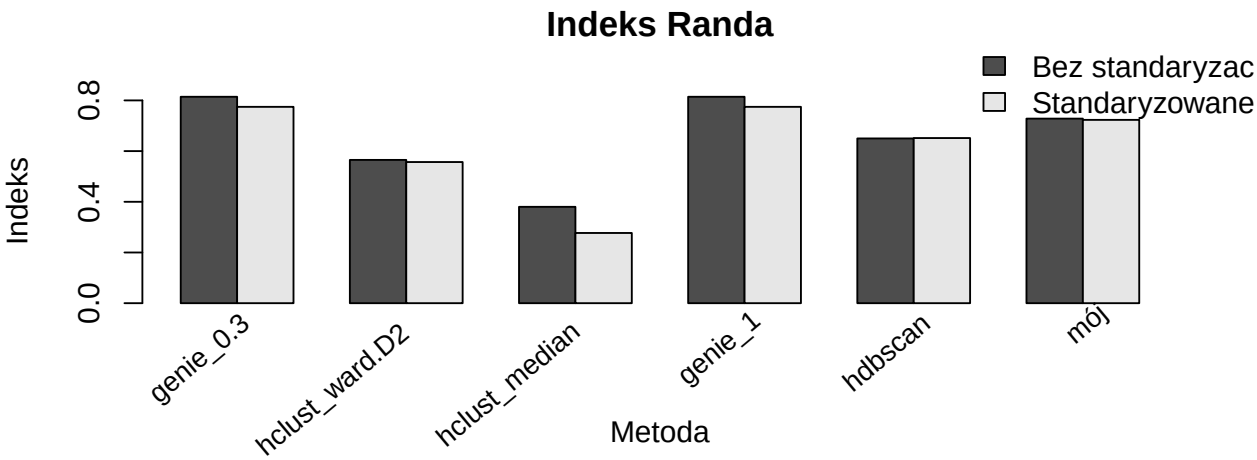

Figure 10: Zależność Indeksu Randa od Standaryzacji

Jak widać w przypadku niektórych algorytmów wpływ jest niezauważalny a w innych wypadkach negatywny. Teraz indeks Fowlkesa–Mallowsa:

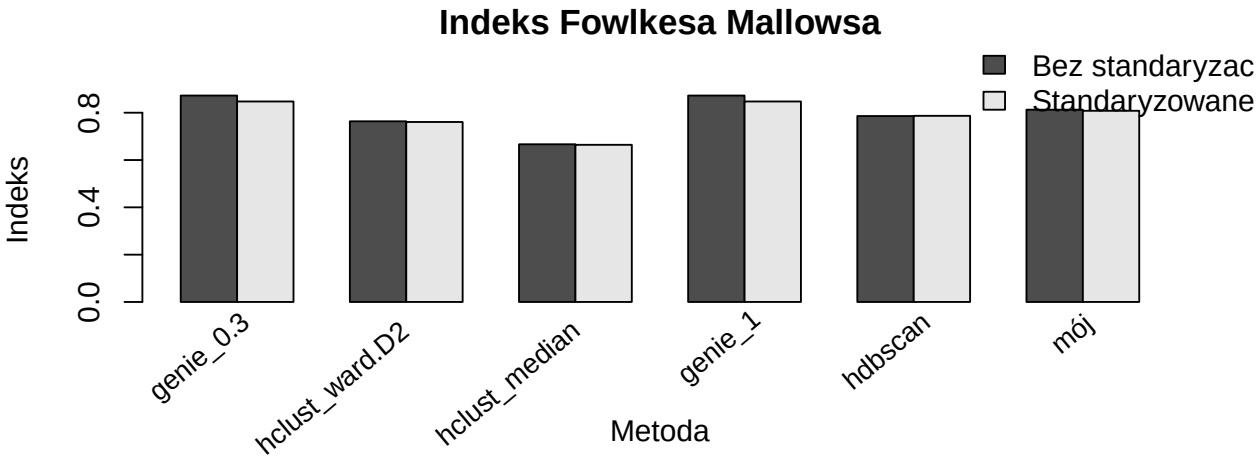

Figure 11: Zależność Indeksu Fowlkesa–Mallowsa od Standaryzacji

Jak widać drugi parametr jakościwy również wskazuje na to samo chociaż w mniejszym stopniu . Mimo to myślę ,że można wyciągnąć wniosek o negatywnym wpływię standaryzac przy analizie skupień.

Zbiory

Teraz zaprezentuję z którym zbiorem najgorzej poradził sobie dany algorytm:

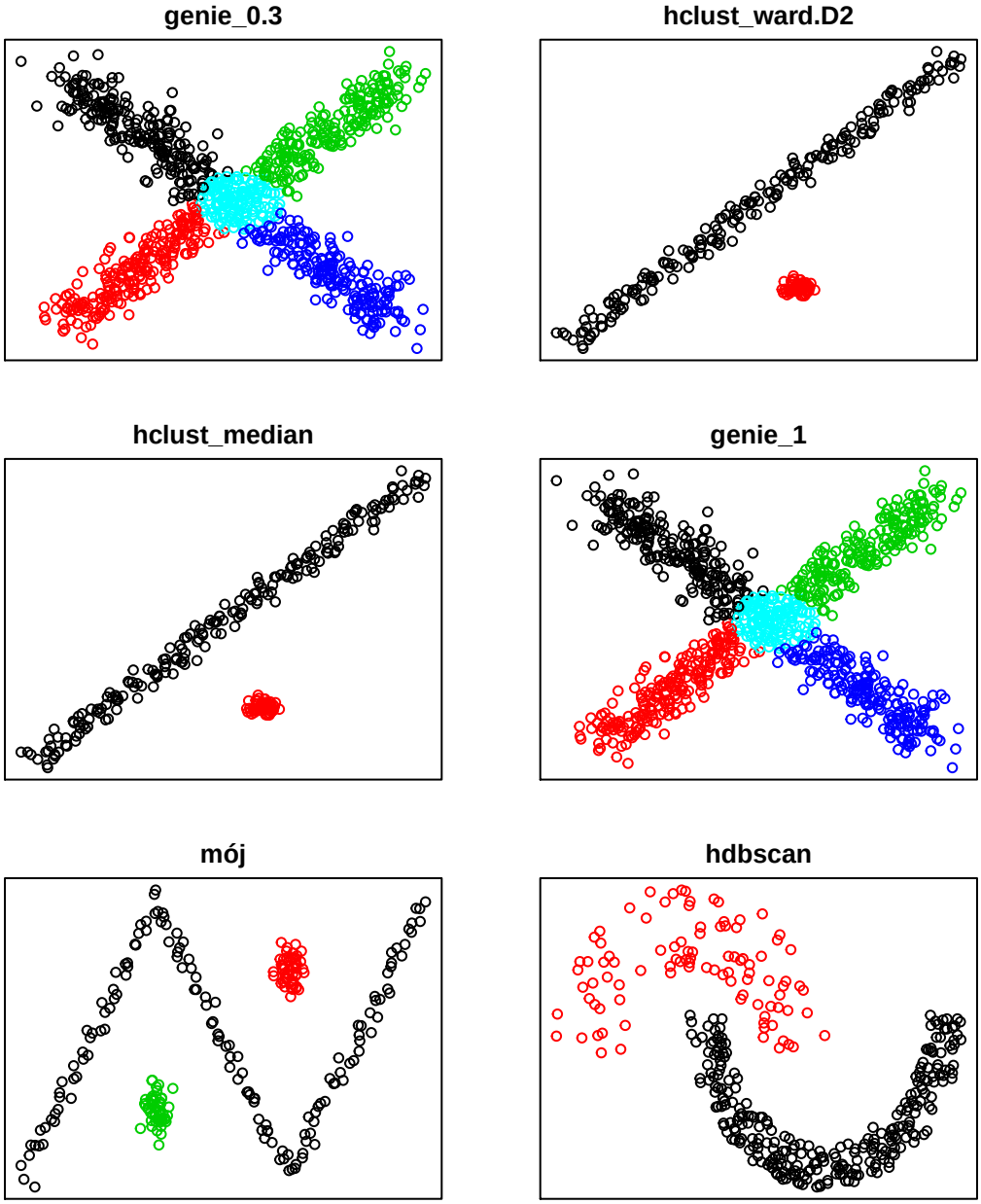

Figure 12: Najgorzej podzielone zbiory

Podsumowanie

Wszystkie przytoczone przeze mnie wykesy wskazują na wyraźną dominację algorytmu **genie** który nie okazał się najszybszy . Pomimo tege przy bardzo wyskojej prędkości działania zapenia najlepszą skuteczność z badanych algorytmów/implementacij . Moja własna implementacja okazała się około 100 razy wolniejsza od **genie** ale nie wiele od niej mniej dokładna. Algorytm **hdbscan** pomimo działania bez podanej odgórnie liczby skupień okazał sie dokładniejszy niż metody funkcij **hclust**.