Analiza algorytmów klasteryzacyjnych

Patryk Korczak 188618, Alicja Graczyk 188814, Łukasz Prahl 184340

Maj 2023

1 Opis problemu

Celem projektu była implementacja i analiza trzech różnych metod klasteryzacji danych, czyli tak zwanej klasyfikacji bez nadzoru, dla trzech różnych zestawów danych.

Metody, które wybraliśmy do realizacji zadania to K-means, DBSCAN i klasteryzacja hierarchiczna. Natomiast do sprawdzenia jakości ich działania wybraliśmy zestawy:

- **Wine** - 178 instancji, 13 atrybutów
- **Yeast** - 1484 instancji, 8 atrybutów
- **E-coli** - 336 instancji, 8 atrybutów

Do wczytania danych i zaimplementowania algorytmów wykorzystaliśmy język Python wraz z bibliotekami *sklearn*, *pandas* i *numpy*. Wykresy natomiast zostały utworzone za pomocą biblioteki *matplotlib*

2 Opis algorytmów

Algorytm K-means to popularny algorytm klasteryzacji, który grupuje zbiór danych na podstawie ich podobieństwa do siebie. Algorytm ten wymaga wcześniejszego określenia liczby klastrów.

Proces działania algorytmu K-means przebiega w następujący sposób:

1. Na początku losowo wybieramy K punktów jako centroidy klastrów. K oznacza liczbę klastrów, którą musimy wcześniej określić.
2. Przypisujemy każdy punkt danych do najbliższego centroidu na podstawie odległości euklidesowej.

3. Dla każdego klastra obliczamy nową pozycję centroidu, jako średnią ze wszystkich punktów należących do tego klastra.

4. Powtarzamy kroki 2-3, aż centroidy przestaną się znacząco przesuwać lub zostanie osiągnięta maksymalna liczba iteracji.

W wyniku działania algorytmu K-means otrzymujemy zbiór K klastrów, gdzie każdy punkt danych jest przypisany do jednego z klastrów. Każdy klaster jest reprezentowany przez swojego centroida.

Dobrą praktyką jest wykonanie kilku prób algorytmu K-means z różnymi losowymi początkowymi centroidami, aby wybrać najlepszy wynik.

Algorytm DBSCAN (Density-Based Spatial Clustering of Applications with Noise) to algorytm klasteryzujący, który identyfikuje gęste obszary danych w przestrzeni wielowymiarowej.

Algorytm DBSCAN nie wymaga określania liczby klastrów na początku. Zamiast tego, opiera się na dwóch parametrach: epsilon (ϵ) i minimum punktów ($minPts$). Parametr ϵ definiuje promień wokół każdego punktu, a parametr $minPts$ określa minimalną liczbę punktów, które muszą być obecne wewnątrz promienia ϵ, aby taki punkt został uznany za punkt rdzeniowy.

Proces działania algorytmu DBSCAN przebiega następująco:

1. Wybieramy losowy punkt, który nie został jeszcze przypisany do żadnego klastra.

2. Sprawdzamy, czy liczba punktów znajdujących się wewnątrz promienia ϵ wokół tego punktu jest większa lub równa $minPts$. Jeśli tak, to punkt ten jest uznawany za punkt rdzeniowy.

3. Jeśli punkt jest punktem rdzeniowym, tworzymy nowy klaster i dodajemy ten punkt do klastra. Następnie znajdujemy wszystkie sąsiadujące punkty wewnątrz promienia ϵ i dodajemy je do klastra. Powtarzamy ten krok dla każdego nowo dodanego punktu rdzeniowego, aż nie zostaną znalezione wszystkie sąsiednie punkty.

4. Jeśli punkt nie jest punktem rdzeniowym, ale znajduje się wewnątrz promienia ϵ innego punktu rdzeniowego, to jest on dodawany do tego klastra.

5. Powtarzamy kroki 1-4 dla każdego nieprzypisanego jeszcze punktu danych.

6. Algorytm kończy się, gdy wszystkie punkty zostaną przypisane do odpowiednich klastrów.

W wyniku działania algorytmu DBSCAN otrzymujemy zbiór klastrów, które reprezentują gęste obszary danych, oraz punkty odstające, które nie należą do żadnego klastra.

Hierarchiczny algorytm klasteryzujący to metoda grupowania danych, która tworzy hierarchię klastrów na podstawie podobieństwa między punktami danych. Istnieją dwa główne podejścia w hierarchicznej klasteryzacji: aglomeracyjne (dolne do góry) i deglomeracyjne (górne do dołu). W naszym projekcie korzystamy z podejścia aglomeracyjnego.

Proces działania klasteryzacji hierarchicznej aglomeracyjnej:

1. Na początku każdy punkt danych jest traktowany jako osobny klaster.
2. Obliczane jest podobieństwo między klastrami na podstawie określonej miary odległości (w naszym przypadku odległość euklidesowa).
3. Dwa najbardziej podobne klastry są łączone w jeden większy klaster.
4. Krok 2 i 3 powtarzany jest aż do utworzenia jednego globalnego klastra zawierającego wszystkie punkty danych.

W wyniku hierarchicznej klasteryzacji otrzymujemy hierarchię klastrów w postaci dendrogramu, który przedstawia kolejne poziomy połączeń klastrów. Na podstawie dendrogramu można wybrać ostateczną liczbę klastrów, obcinając dendrogram na odpowiednim poziomie. Algorytm hierarchiczny jest bardziej złożony obliczeniowo, zwłaszcza dla dużych zbiorów danych.

3 Opis realizacji zadania

Przed przystąpieniem do realizacji zadania każdy zestaw danych, który użyliśmy w projekcie, należało przygotować poprzez oddzielenie od całego pliku danych kolumny z informacją, do której klasy należy dany rząd atrybutów. W wyniku dostajemy wektor klas i tablicę wartości atrybutów.

Tablicę wartości przepuszczamy przez nasze algorytmy. Algorytm hierarchiczny oraz K-means przed uruchomieniem wymaga podania ile docelowo powinno być klastrów, natomiast algorytm DBSCAN wymaga podanie wartości ϵ do wyznaczenia samodzielnie klastrów.

Utworzone klastry porównujemy z rzeczywistymi klasami. Korzystając z redukcji wymiarów nasze rezultaty przedstawiamy na wykresie, aby łatwo dostrzec jak każdy algorytm poradził sobie z danym zestawem danych. Poza

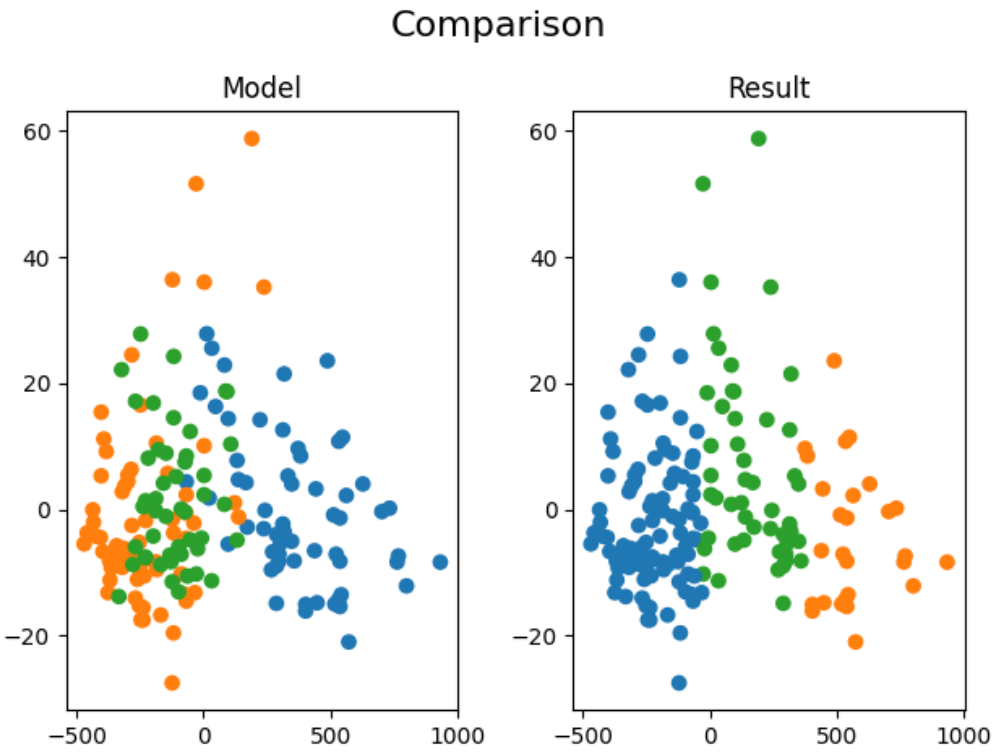
redukcją wymiarów korzystamy także z wskaźnika AMI (Adjusted Mutual Information), czyli wariacji informacji wzajmnej, dzięki której możemy porównywać wyniki klasteryzacji z realnymi przypisaniami.

$$AMI(U,V) = \frac{MI(U,V) - E\{MI(U,V)\}}{max\{H(U),H(V)\} - E\{MI(U,V)\}}$$

Gdzie U, V to zbiory klastrów, H entropia danego zbioru, MI informacja wzajemna, a E to spodziewana wartość wzajemnej informacji. Wskaźnik AMI może przyjmować wartości od 0 do 1, gdzie 0 oznacza brak zależności, a 1 oznacza doskonałą zależność.

4 Analiza uzyskanych wyników

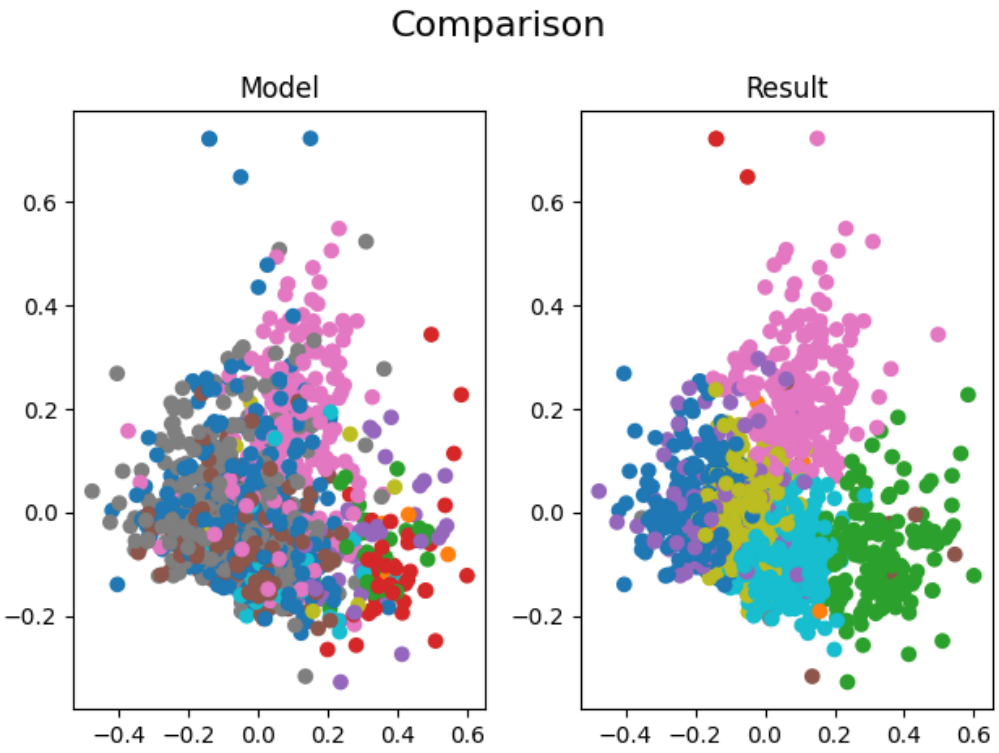
4.1 Metoda K-means



Wykres 1: Wynik klasteryzacji zbioru Wine metodą K-means

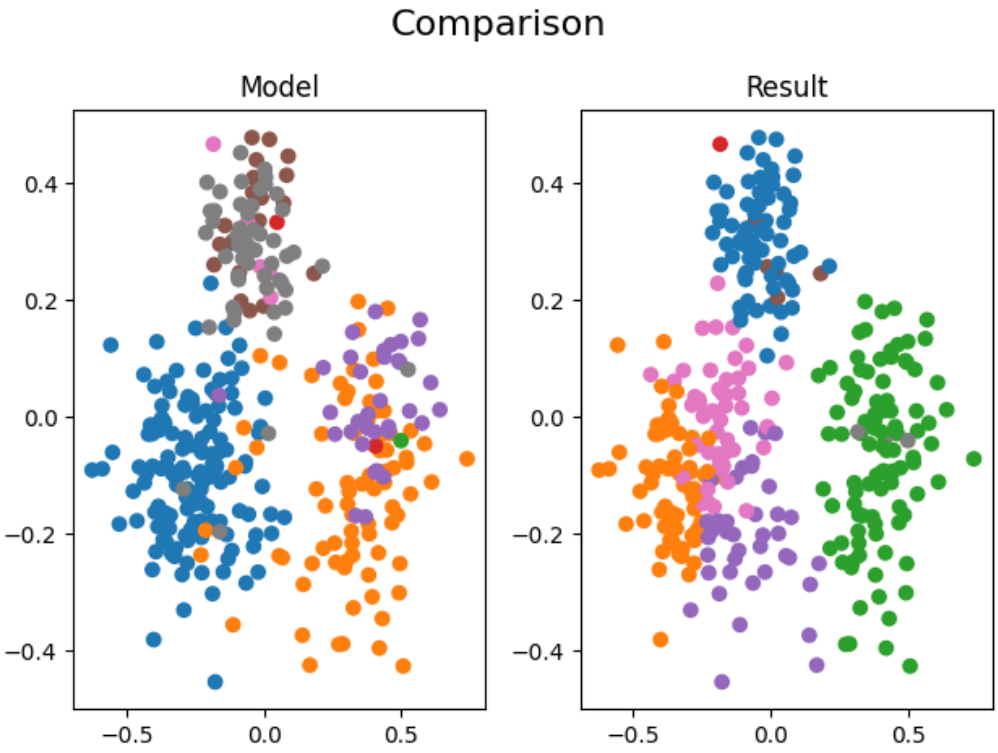
K-means utworzył wyraźnie separowalne klastry. Klaster niebieski zawiera w sobie dane modelowo należace do klastra zielonego. Podobnie klaster

pomarańczowy zawiera dane klastrea zielonego. Wynika to z przenikania przez siebie danych w modelowych klastrach bez wyraźnych granic. Wartość $AMI = 0.42$



Wykres 2: Wynik klasteryzacji zbioru Yeast metodą K-means

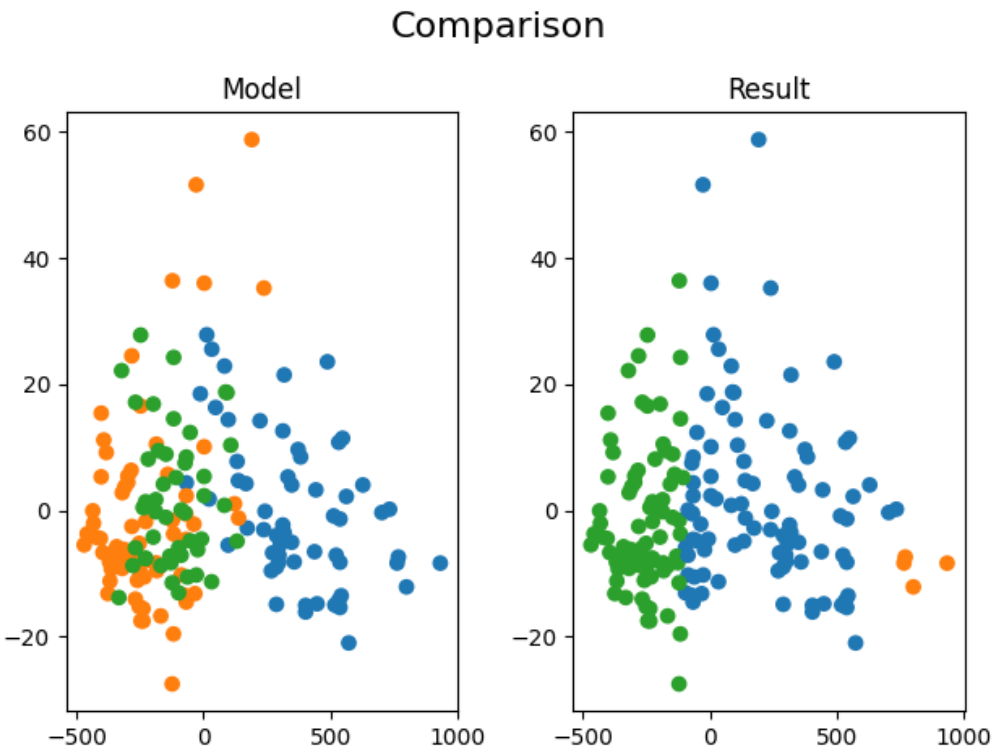
K-means ponownie tworzy wyraźnie zarysowane klastry, co zasadniczo różni się od danych modelowych. Punkty z oryginalnego szarego klastra zotsały podzielone na kilka różnych grup, za to wyjściowy zielony klaster przypisał do siebie nadmiarowe dane. Gorszy wynik jest skutkiem nieliniowych struktur modelowych klastrów. Wartość $AMI = 0.26$



Wykres 3: Wynik klasteryzacji zbioru E-coli metodą K-means

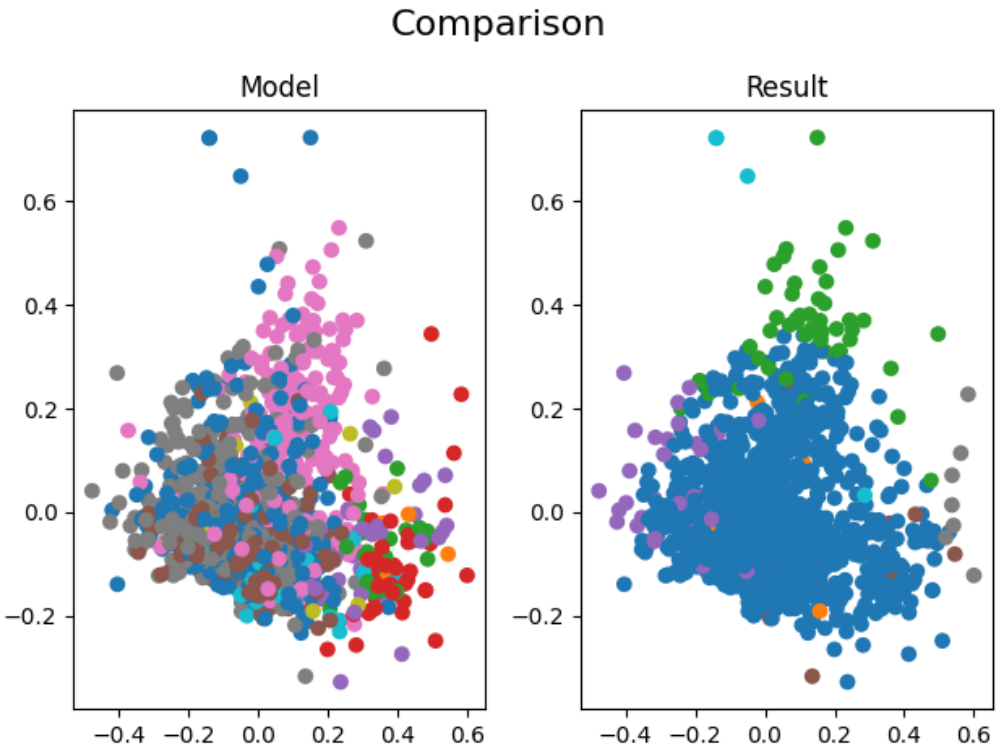
Jest to najdokładniejszy wynik w projekcie ze względu na to, że klastry zbioru E-coli są najlepiej oddzielone, co znacząco zwiększa porawność działania K-means. Przyporządkowanie danych to klastrów jest bardzo podobne z modelowym, problemem wciąż pozostają nachodzące na siebie dane jak np. oryginalne klastry pomarańczowy i fioletowy zostały błędnie zaklasyfikowane do jednego klastra zielonego. Wartość $AMI = 0.62$

4.2 Metoda DBSCAN



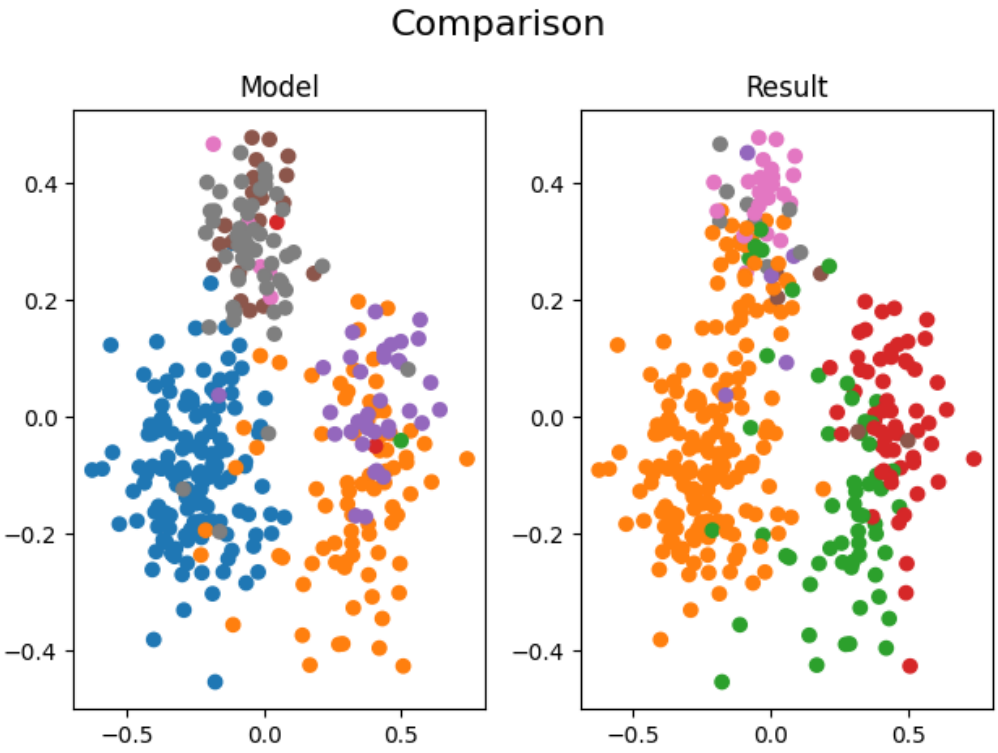
Wykres 4: Wynik klasteryzacji zbioru Wine metodą DBSCAN

Algorytm DBSCAN na zbiorze Win poprawnie utworzył trzy klastry. Zielony klaster zawiera dane z dwóch klas. Drugi, niebieski klaster zawiera dane z wszystkich klas, z czego jednej klasy tylko kilka elementów. Trzeci, najmniejszy klaster zawiera tylko kilka elementów z jednej klasy, są to elementy z tej samej klasy, z której zawierają się elementy w klastrze niebieskim. Wartość AMI dla tego zbioru wynosi 0.32.



Wykres 5: Wynik klasteryzacji zbioru Yeast metodą DBSCAN

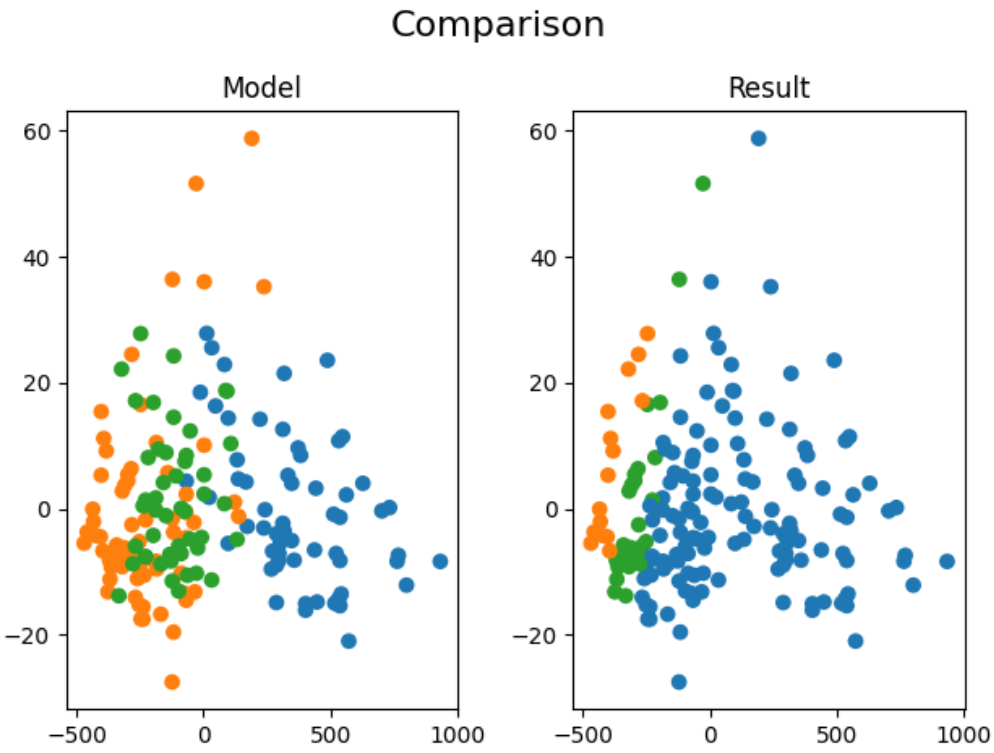
W przypadku zestawu danych Yeast algorytm DBSCAN nie poradził sobie za dobrze. Jak możemy zauważyć na wykresie, większość elementów zawiera się w klastrze niebieskim, poza nim utworzone zostały inne klastry, między innymi klaster zielony i fioletowy, które zawierają część elementów z klasy. Wartość AMI dla tego zbioru wynosi 0.076.



Wykres 6: Wynik klasteryzacji zbioru E-coli metodą DBSCAN

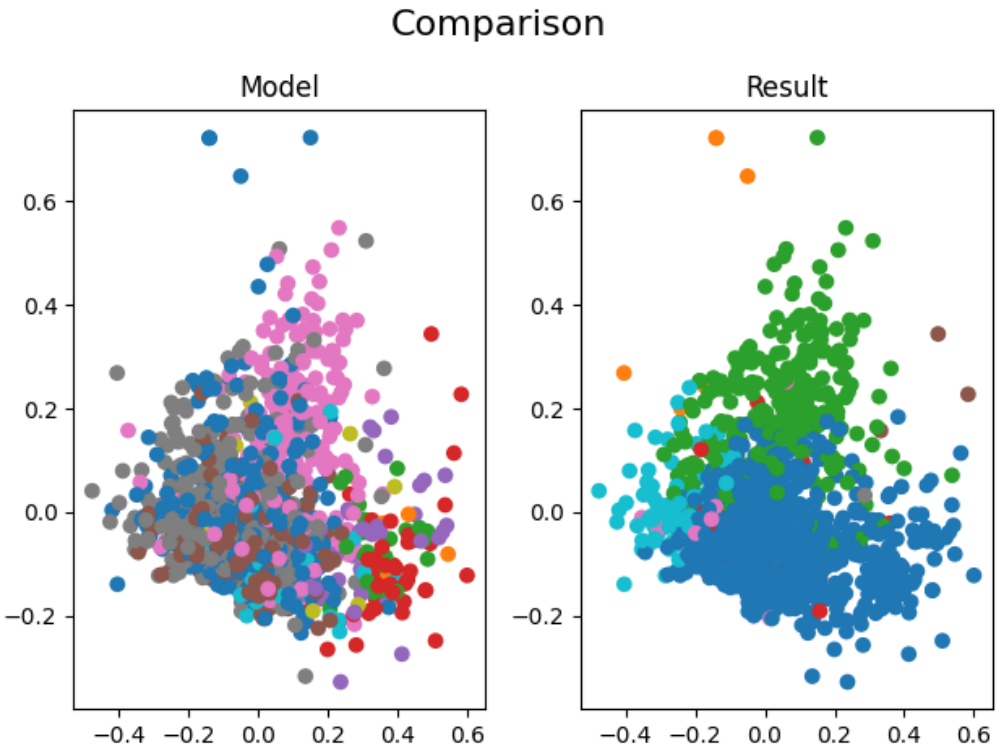
Z trzech sprawdzanych zestawów danych algorytm DBSCAN poradził sobie najlepiej z zestawem danych E-coli, powstałe klastry są bardzo podobne do rzeczywistego podziału elementów w zbiorze i jedyny problem występuje przy klastrach o kolorze różowym i szarym, tam można zauważyć że algorytm nie rozpoznał poprawnie jednego obszaru i zamiast utworzenia nowego klastru, przypisał go do istniejącego klastra pomarańczowego. Wartość AMI dla tego zbioru wynosi 0.52.

4.3 Metoda hierarchiczna aglomeracyjna



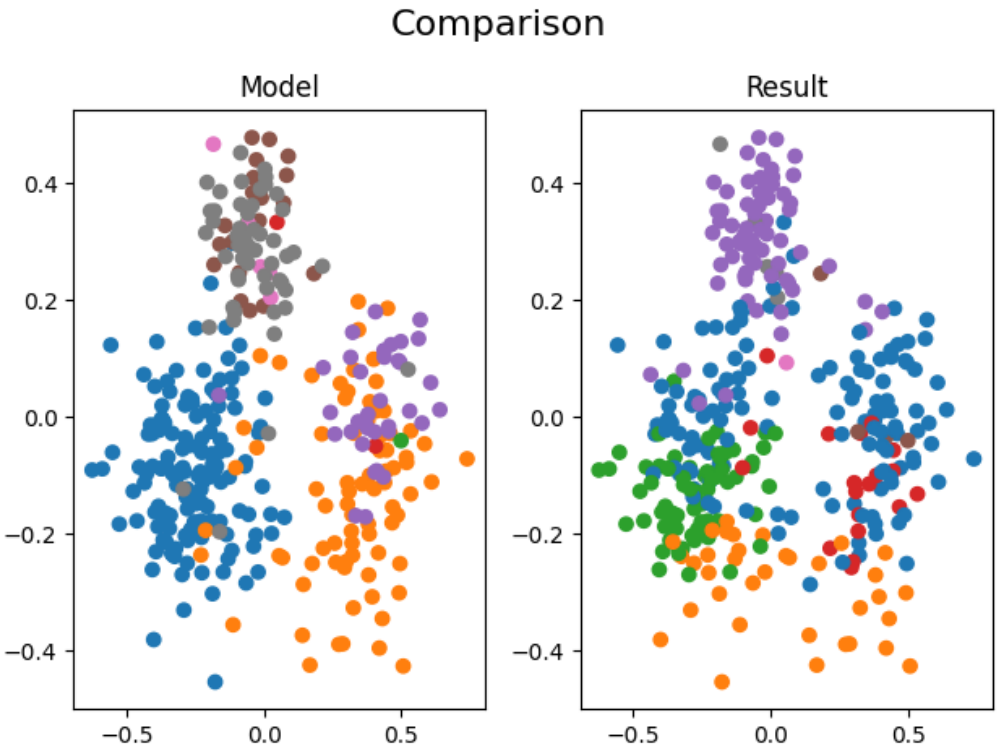
Wykres 7: Wynik klasteryzacji zbioru Wine metodą hierarchiczną

Na wykresie widać, że korzystając z metody klasteryzacji hierarchicznej powstał jeden dominujący klaster, który zawiera w sobie dane z wszystkich trzech klas win zawartych w badanym zbiorze. Pozostałe dwa klastry są zdecydowanie mniejsze i nie zawierają elementów klasy przedstawionej kolorem niebieskim. Wartość wskaźnika AMI wyniosła 0.17.



Wykres 8: Wynik klasteryzacji zbioru Yeast metodą hierarchiczną

Dla zbioru Yeast wynik klasteryzacji bardzo różni się od rzeczywistego podziału. Powstał klaster zawierający elementy z każdej klasy (kolor niebieski), dwa klastry zawierające elementy głównie z jednej klasy (kolor jasnoniebieski oraz zielony) oraz mniejsze klastry zawierające pojedyncze elementy. Taki wynik klasteryzacji może być spowodowany strukturą zbioru. Jak można zauważyć na wykresie, elementy klas przeplatają się, co znacząco utrudnia pracę algorytmu. Wartość wskaźnika AMI jest bliska wartości uzyskanej przez algorytm DBSCAN i wynosi 0.087.



Wykres 9: Wynik klasteryzacji zbioru E-coli metodą hierarchiczną

Wyniki pracy klasteryzacji hierarchicznej na zbiorze E-coli są najlepsze spośród pozyskanych tym algorytmem klasteryzacji. Możemy zaobserwować, że klasa przedstawiona kolorem szarym została przyzwoicie odwzorowana poprzez kolor fioletowy. Niefortunne jest ponowne pojawienie się klastrów o bardzo małej ilości elementów. Algorytmowi udało się stworzyć klaster zawierający 95% elementów z jednej klasy, aczkolwiek jest to niecałe 50% wszystkich elementów tej klasy zawartych w badanym zbiorze. Wartość wskaźnika AMI wyniosła 0.37. Jest to największa wartość wskaźnika uzyskana tym algorytmem.

5 Podsumowanie

Na podstawie wyników uzyskanych dla wszystkich metod można stwierdzić, że z testowanych algorytmów, K-means poradził sobie najlepiej. Osiąga on najwyższe wartości wskaźnika AMI dla wszystkich zestawów danych. Generalnie K-means jest oparty na założeniu, że klastry mają kształt hipersferyczny i są linowo separowalne, dlatego jest najbardziej skuteczny w przy-

padku danych, w których klastry są dobrze oddzielone i mają wyraźną strukturę.

Algorytm DBSCAN ma problemy w wykrywaniu klastrów o różnych gęstościach. Dobranie odpowiedniego parametru ϵ i $minPts$ poprawi wyniki działania algorytmu. Na podstawie wskaźnika AMI można zauważyć, że algorytm DBSCAN poradził sobie najlepiej ze zbiorem E-coli, a najgorzej ze zbiorem Yeast.

W przypadku algorytmu hierarchicznego wyniki były najmniej zadowalające. Algorytm posiada tendencję to tworzenia jednego dominującego klastra. Ponadto sposób, w jaki tworzone są klastry, powoduje częste występowanie zbiorów z małą ilością elementów. Zbiory te nie odpowiadają rzeczywistemu stanowi klas.

Niskie wartości wskaźnika AMI są spowodowane nachodzeniem na siebie klas, co widać na wykresach. Rezultatem tego jest problem z liczeniem odległości między elementami zbioru, ponieważ często lepszy wynik uzyskiwany jest między instancjami z różnych klas, niż z tej samej.

Kolejnym czynnikiem wpływającym na wyniki klasyfykacji jest różnica w liczności klastrów. Zbiór Wine jest najbardziej wyrównayny, natomiast w zbiorach E-coli i Yeast różnice w liczności między klastrami wynoszą dla niektórych par ponad 100. Różnice w liczności klastrów mogą wpływać na identyfikację granic klastrów. Te z większą liczbą instancji mogą mieć większy zasięg, a mniejsze mogą być bardziej związane z większymi klastrami jako punkty graniczne. W rezultacie, mniejsze klastry są błędnie uznane za część większych.