Katzper Michno, 31.05.2022, MPUM

Miniprojekt 4

Celem projektu była implementacja algorytmu grupowania hierarchicznego, algorytmu k-średnich i klasteryzacji spektralnej oraz porównanie ich działania. Dane wejściowe pochodzą z kilku plików:

- *rp.data* - każda obserwacja ma 9 cech i przypisaną klasę 2 lub 4
- *dane_2D_n (*$1 \leq n \leq 8$*)* - punkty na płaszczyźnie z przypisaną klasą (różna liczba klas w zależności od pliku)
- *dane_9D* - każda obserwacja ma 9 cech bez przypisanej klasy

Rozwiązania zostały zaimplementowane w języku Python z pomocą biblioteki NumPy i znajdują się w plikach *main.py*, *hierarchical.py*, *kmeans.py*, *spectral.py*. Wykresy użyte w raporcie zostały wygenerowane z pomocą modułu Matplotlib.

Obróbka danych

Przed uruchomieniem dowolnego algorytmu dane ze wszystkich plików zostały ustandaryzowane, czyli przesunięte o średnią arytmetyczną i podzielone przez odchylenie standardowe.

Miara jakości algorytmów klasteryzacji

Wszystkie zestawy danych poza plikiem *dane_9D* posiadają informację o klasach do których przydzielone są poszczególne obserwacje, co pozwala nam wprowadzić miarę jakości klasteryzacji którą nazywać będziemy *błędem klasyfikacji*. Formalnie, jeśli algorytm klasteryzacji podzielił nasz zbiór danych $\{(x^{(i)}, y^{(i)}) : i \in \{0, .., m-1\}\}$ na klastry $C_0, C_1, ..., C_k \subseteq \{0, ..., m-1\}$, to *błąd klasyfikacji* wynosi:

$$e(C_0, ..., C_k) = 1 - \frac{1}{\binom{m}{2}} \left[\sum_{0 \leq a \leq b \leq k} \sum_{i \in C_a} \sum_{i \neq j \in C_b} 1[(y^{(i)} = y^{(j)} \wedge a = b) \vee (y^{(i)} \neq y^{(j)} \wedge a \neq b)] \right]$$

Intuicyjnie, jest to frakcja par obserwacji które zostały źle przypisane do klastrów względem siebie, czyli są w różnych klastrach pomimo że należą do tej samej klasy, lub są w tym samym klastrze mimo różnej klasy. Oczywiście im niższy błąd klasyfikacji, tym lepiej.

Grupowanie hierarchiczne

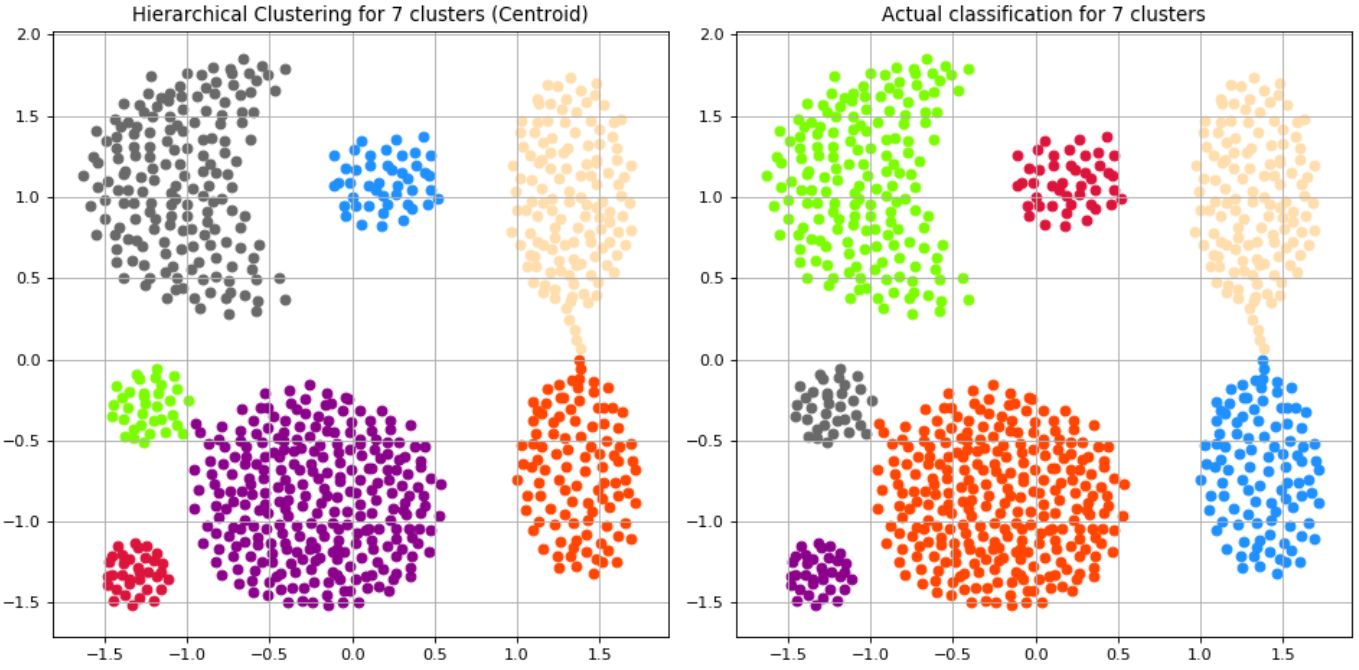
Przy implementacji grupowania hierarchicznego została wykorzystana metryka euklidesowa. Przetestowane zostały różne sposoby łączenia klastrów:

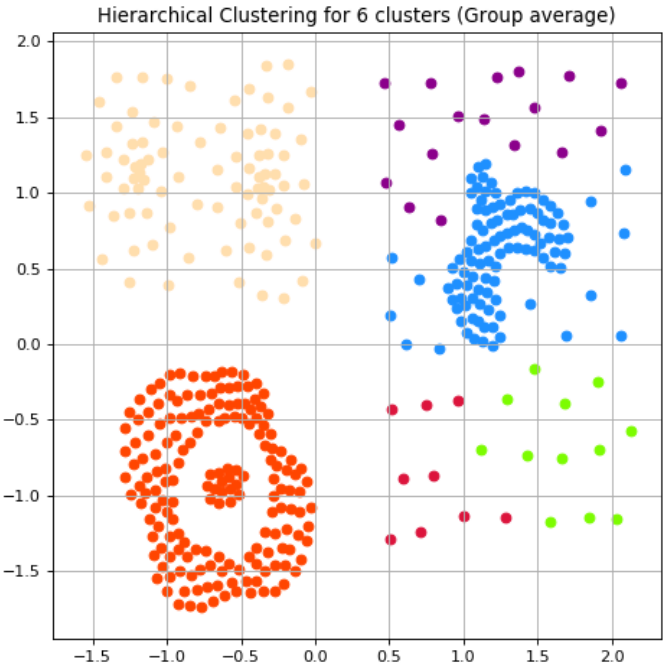
- pojedyncze
- pełne
- średnie
- centroidalne
- Warda

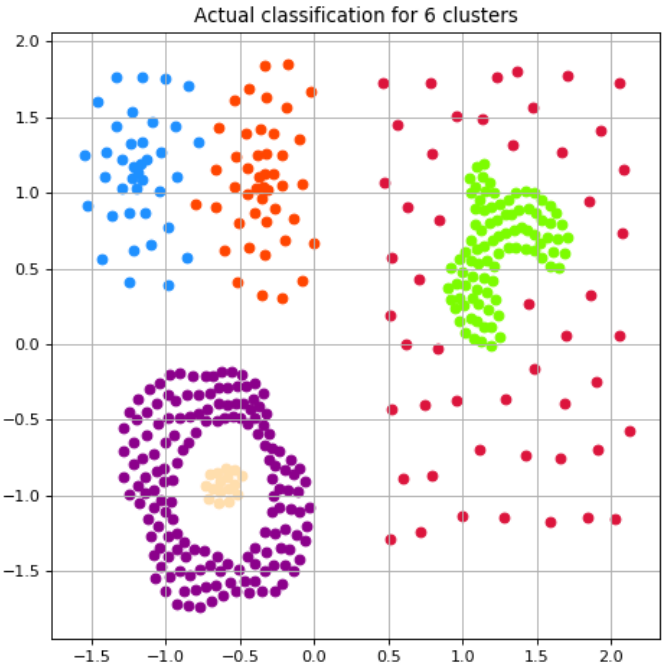
Klasteryzacja została zaimplementowana z pomocą algorytmu Lance'a–Williamsa. Poniżej zostały przedstawione wartości błędu klasyfikacji dla różnych zestawów danych i metod łączenia klastrów przy narzuceniu algorytmowi liczby klastrów równej liczbie klas w danym pliku.

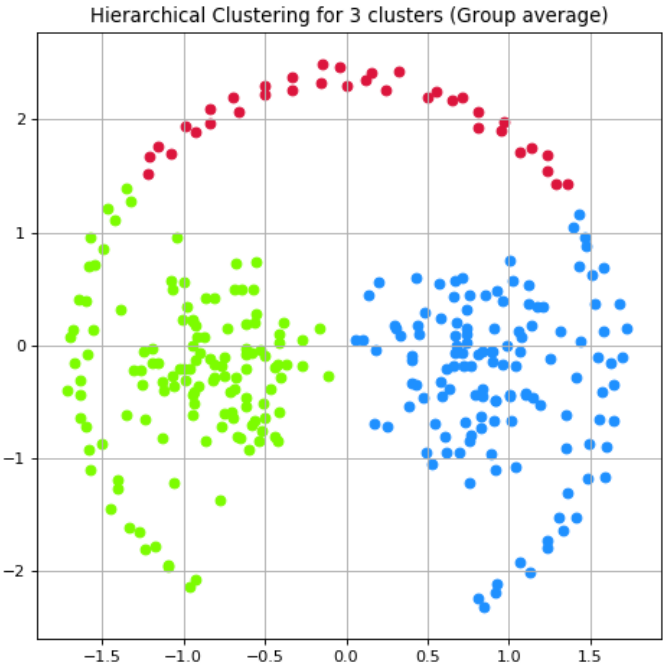
	łączenie pojedyncze	łączenie pełne	łączenie średnie	łączenie centroidalne	łączenie Warda
rp.data	0.4547	0.2585	0.4183	0.4547	0.1258
dane_2D_1	0.0737	0.0228	0.0029	0.0	0.0623
dane_2D_2	0.1012	0.1066	0.0788	0.0842	0.1704
dane_2D_3	0.6598	0.2718	0.2463	0.5121	0.2547
dane_2D_4	0.0	0.4677	0.4434	0.5086	0.4448
dane_2D_5	0.2461	0.0079	0.0045	0.0111	0.0054
dane_2D_6	0.0902	0.0026	0.0013	0.0021	0.0021
dane_2D_7	0.3832	0.2405	0.2405	0.3378	0.3378
dane_2D_8	0.4593	0.3547	0.3213	0.4593	0.3408

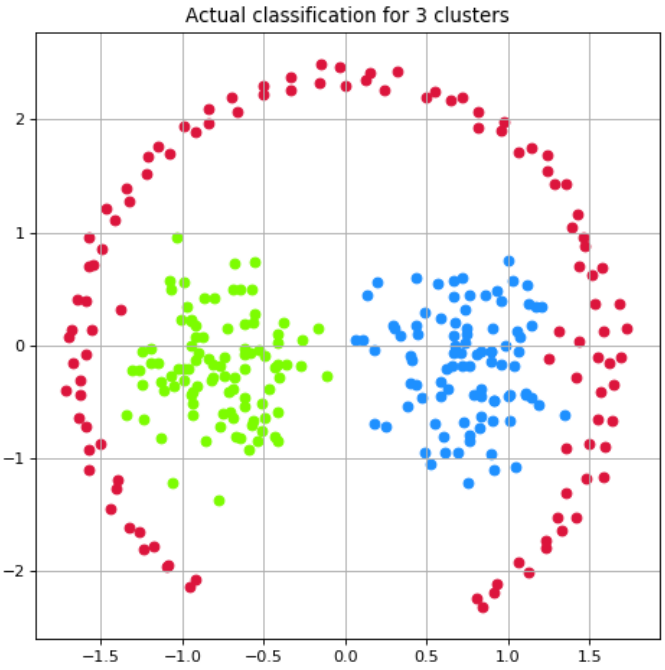
Poniżej dla każdego zastawu danych dwuwymiarowych zostały przedstawione graficznie podziały dokonane przez grupowanie hierarchiczne dla metody osiągającej najmniejszy błąd klasyfikacji. Obok, dla porównania, został zwizualizowany podział obserwacji na klasy.



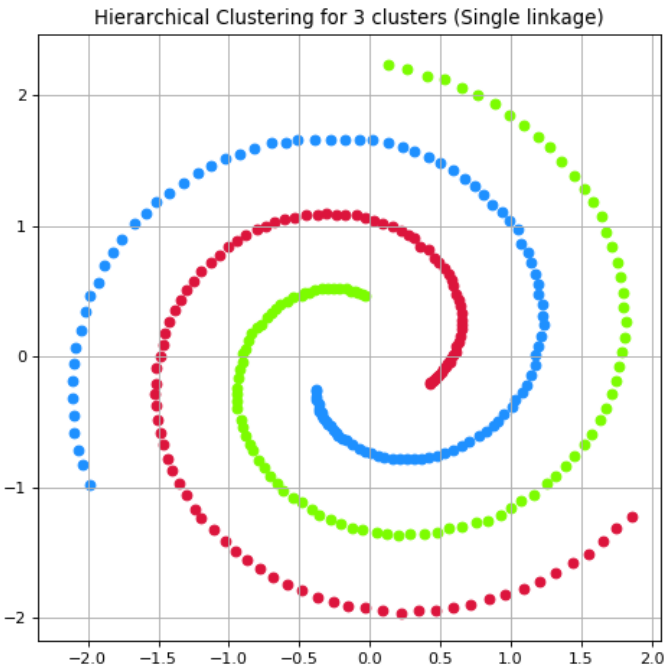

Hierarchical Clustering for 6 clusters (Group average)

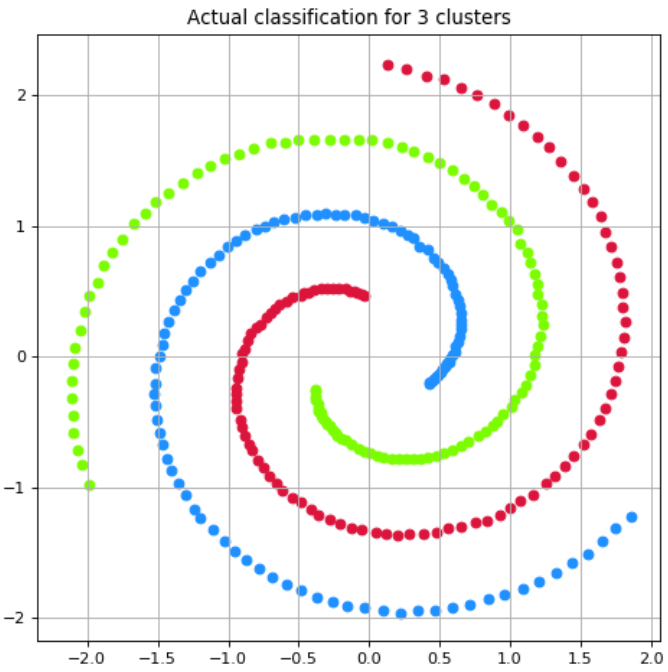

Actual classification for 6 clusters

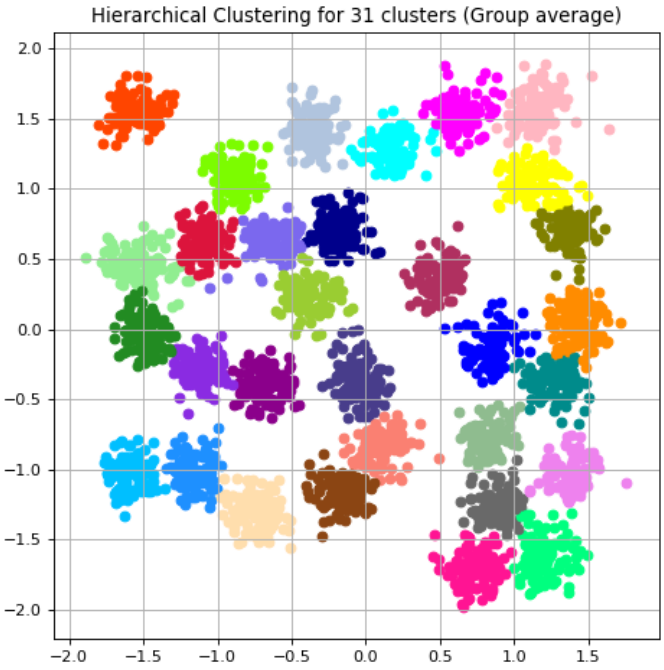

Hierarchical Clustering for 3 clusters (Group average)

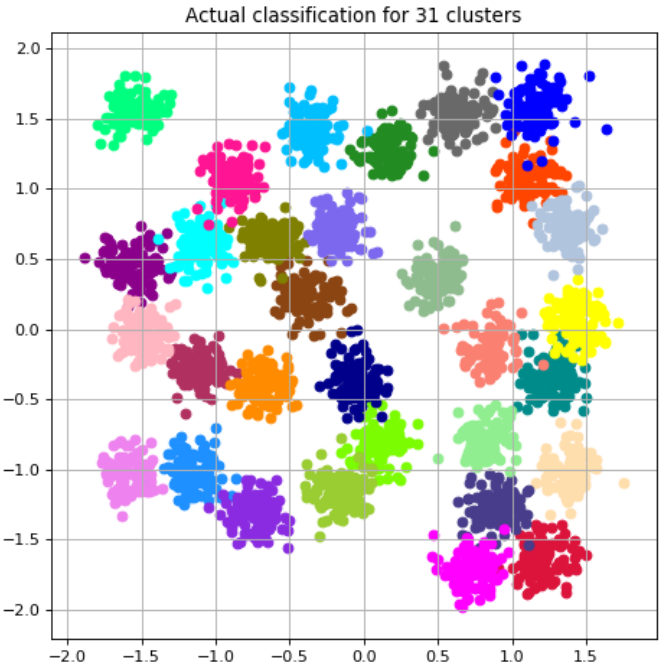

Actual classification for 3 clusters

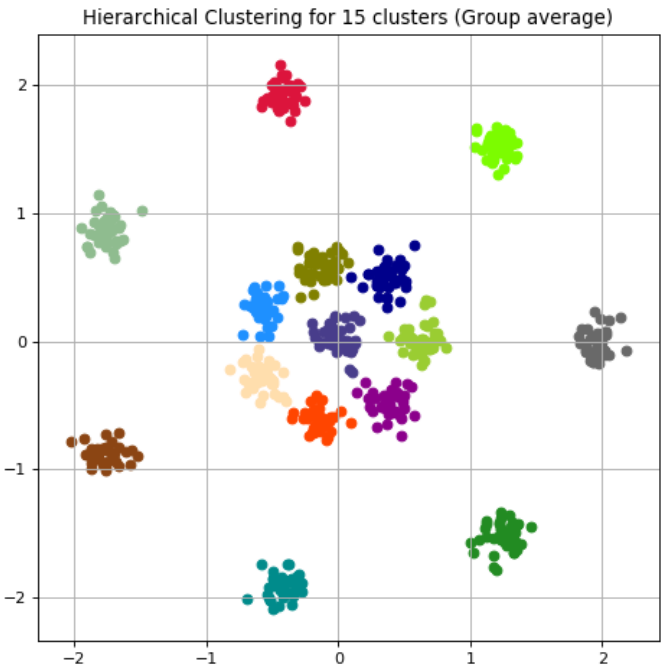
Katzper Michno, 31.05.2022, MPUM

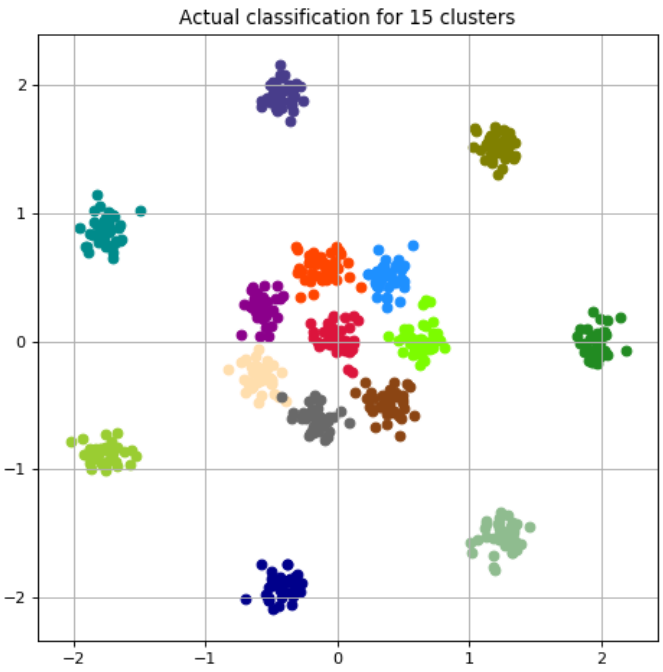


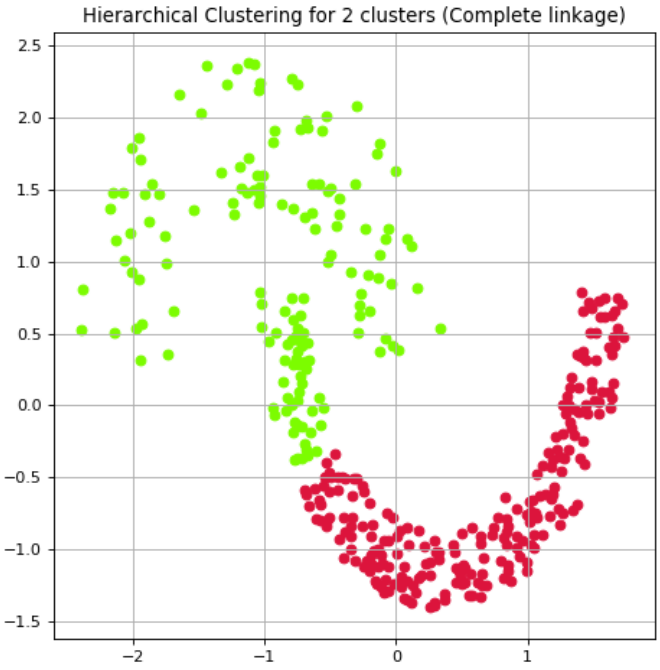


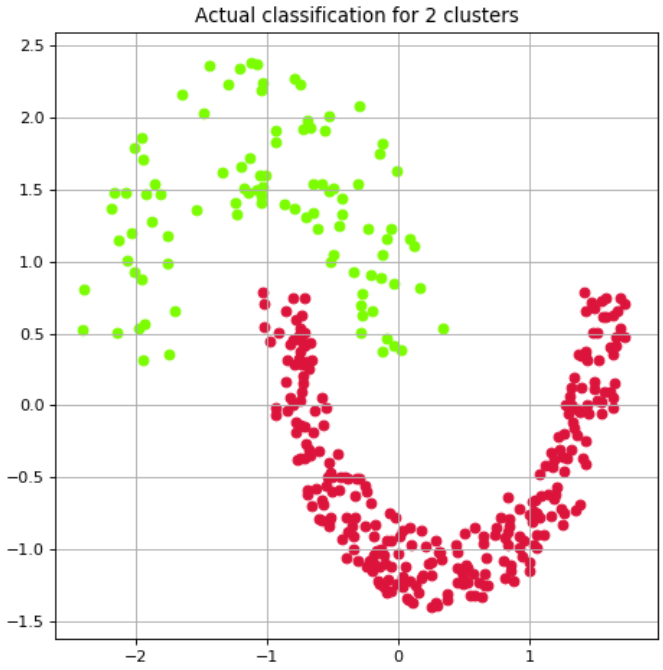


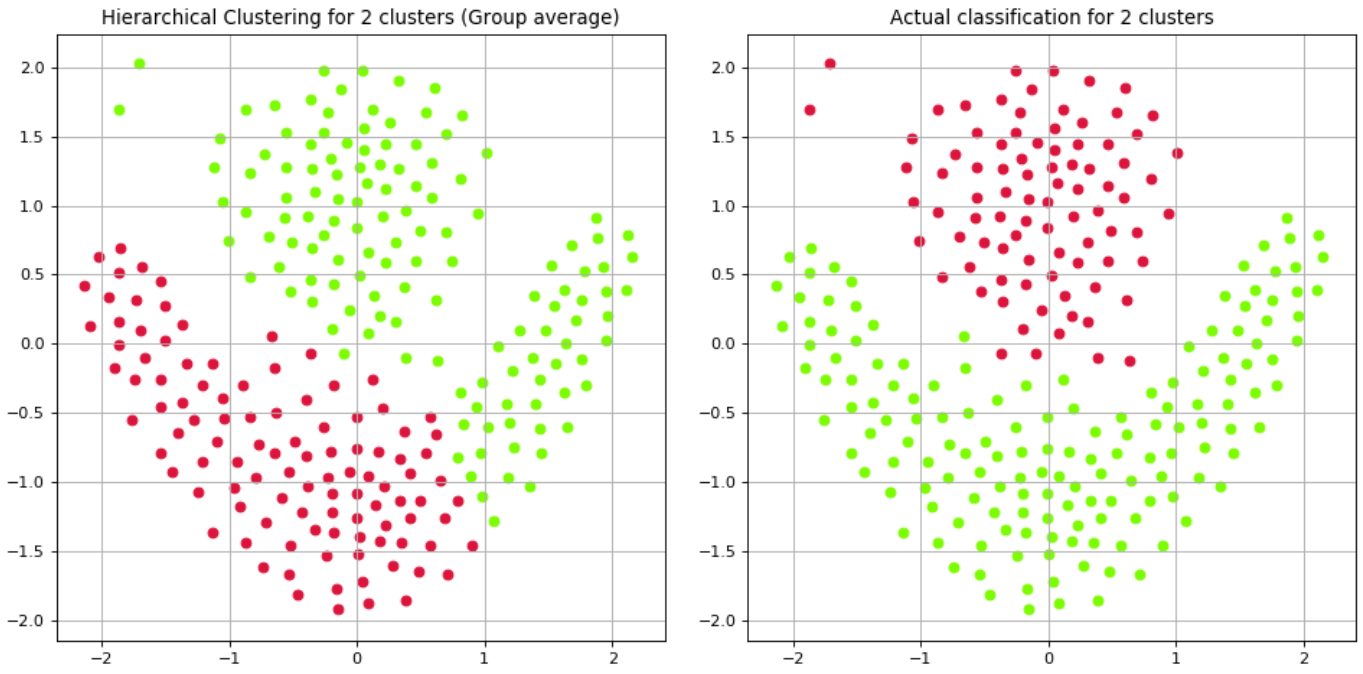



Hierarchical Clustering for 15 clusters (Group average)


Actual classification for 15 clusters


Hierarchical Clustering for 2 clusters (Complete linkage)


Actual classification for 2 clusters



Algorytm k-średnich

W przypadku metody k-średnich także działaliśmy na metryce euklidesowej. Zaimplementowany został algorytm *k-means++*, który w lepszy sposób wybiera początkowe centroidy. Algorytm k-średnich dąży do minimalizacji funkcji błędu:

$$f_k(C_1, ..., C_k) = \sum_{j=1}^{k} \sum_{i \in C_j} d(x^{(i)}, \mu(C_j))^2$$

Dla podziału danych na k klastrów $C_1, ..., C_k$ z centroidami $\mu(C_1), ..., \mu(C_k)$.

Jako, że algorytm k-średnich jest zależny od początkowego wyboru centroidów, najlepszy (na podstawie wyżej zdefiniowanego błędu) podział dla ustalonego k został wybrany spośród 10 przebiegów programu. Następnie dla wybranego podziału został obliczony tzw. *wskaźnik sylwetkowy* S_k:

$$a_i = \frac{1}{|C_I| - 1} \sum_{j \in C_I, j \neq i} d(x^{(i)}, x^{(j)})$$

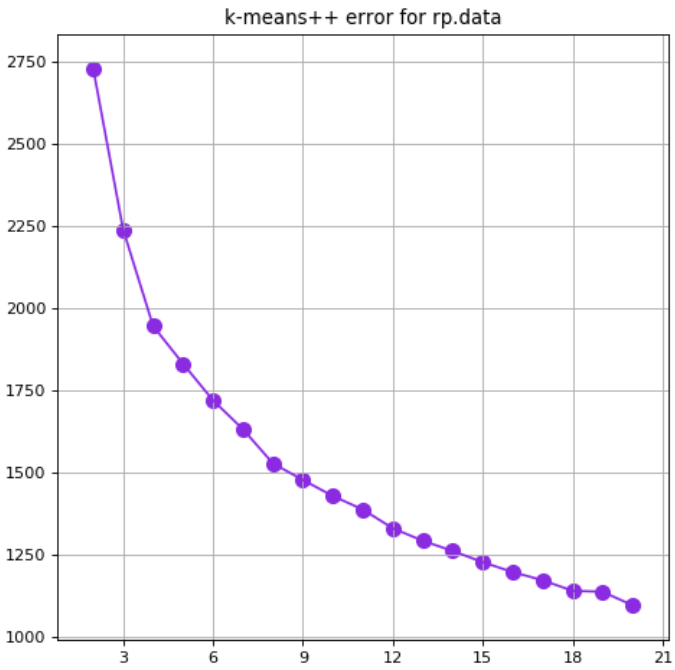
$$b_i = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i, j)$$

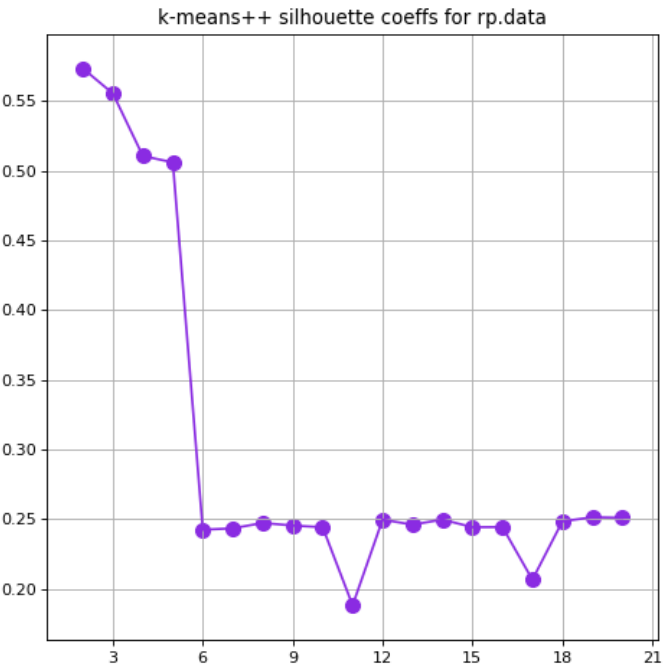
$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

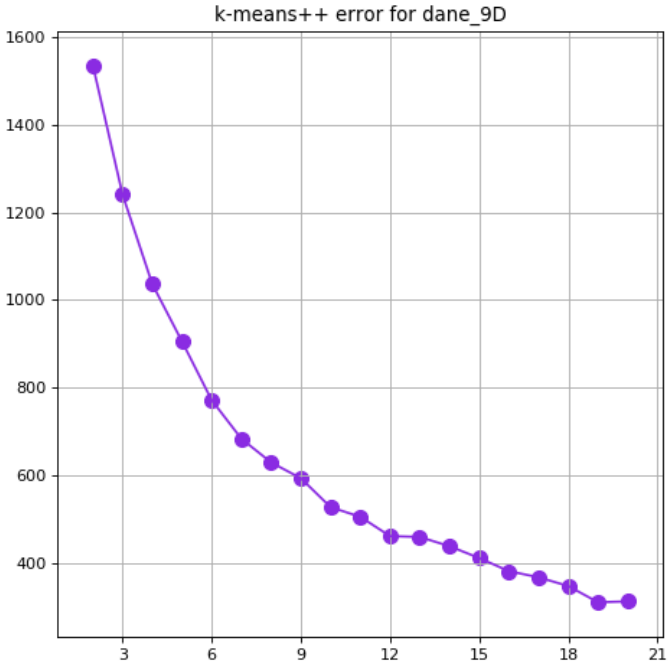
$$S_k = \frac{1}{m} \sum_{i=0}^{m-1} s_i$$

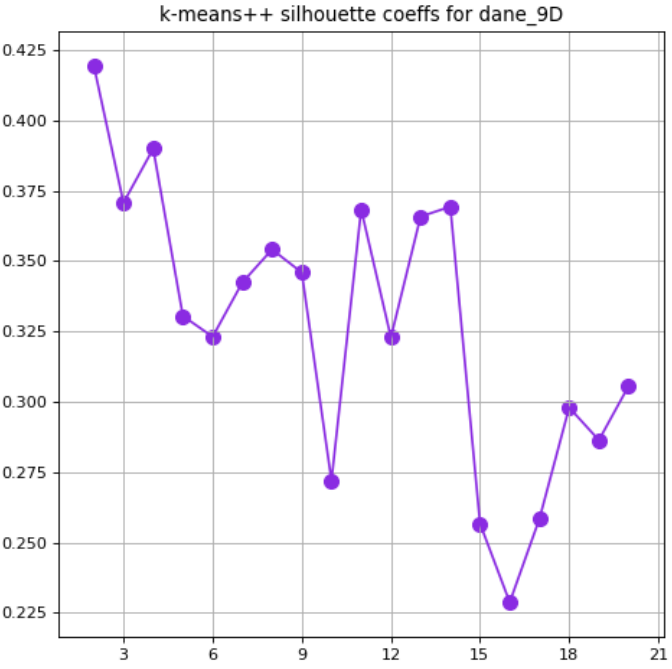
Dla $i \in C_I$ oraz $|C_I| > 1$. Jeśli $|C_I| = 1$, to $s_i = 0$. Wskaźnik sylwetkowy przyjmuje wartość z przedziału $[-1, 1]$ i mówi jak dobrze punkty pasują do swoich klastrów (im wskaźnik jest wyższy, tym lepiej). Został on wykorzystany do określenia optymalnej liczby klastrów w algorytmie k-średnich, jako że metoda łokcia jest ciężka do sformalizowania i niejednoznaczna. Algorytm k-średnich został uruchomiony dla każdego zestawu danych dla pewnych przedziałów wartości k. Następnie wybrane zostało k maksymalizujące współczynnik sylwetkowy. Dla takiego optymalnego k został potem policzony błąd klasyfikacji. Wyniki powyższego eksperymentu zostały przedstawione w poniższej tabeli i na wykresach. Na wykresach znajdują się wskaźniki sylwetkowe i błąd algorytmu k-średnich (tak jak zdefiniowany w tej sekcji - nie błąd klasyfikacji!). Przy wizualizacji dwuwymiarowych wyników poprzez plusy zostały zaznaczone centroidy wynikowych klastrów.

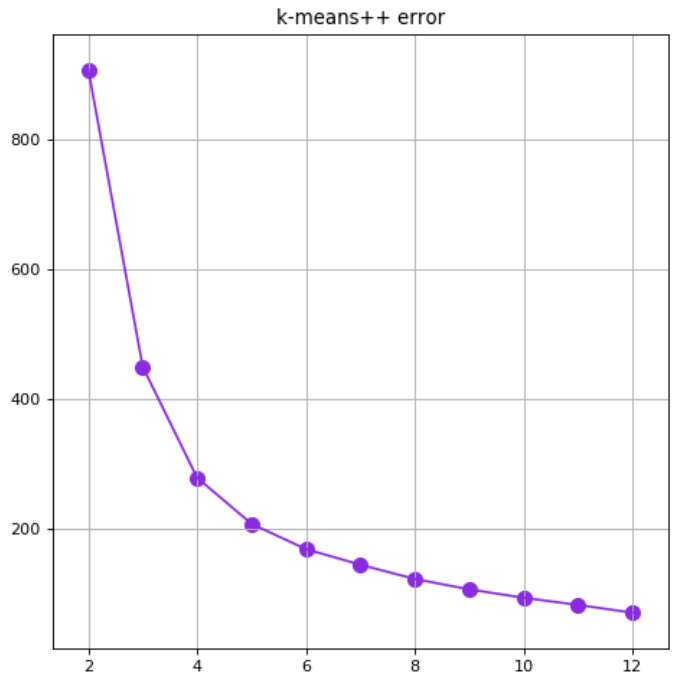
	przedział rozważanych k	optymalne k	błąd klasyfikacji
rp.data	[2, 7]	2	0.0867
dane_9D	[2, 20]	2	-
dane_2D_1	[2, 12]	4	0.0971
dane_2D_2	[2, 11]	3	0.1197
dane_2D_3	[2, 8]	3	0.2419
dane_2D_4	[2, 8]	3	0.4462
dane_2D_5	[2, 36]	30	0.0053
dane_2D_6	[2, 20]	15	0.0008
dane_2D_7	[2, 7]	2	0.2327
dane_2D_8	[2, 7]	4	0.2967

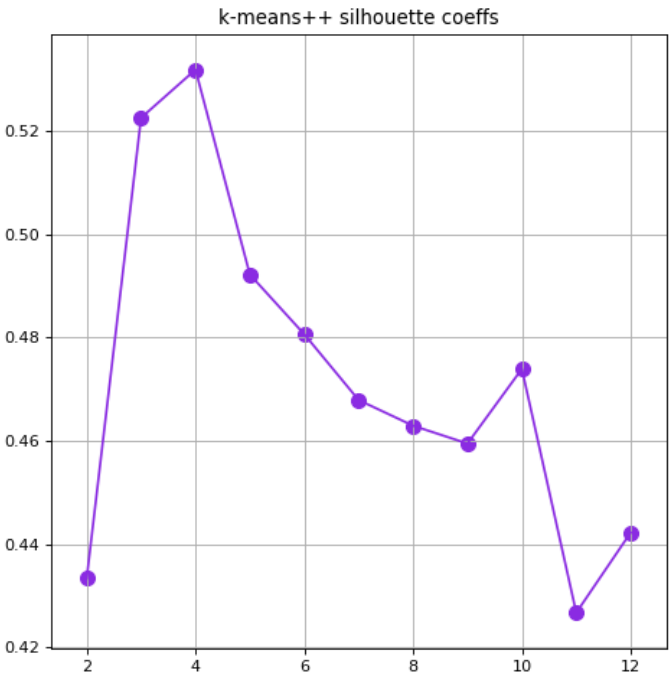

k-means++ error for rp.data
2750
2500
2250
2000
1750
1500
1250
1000
3
6
9
12
15
18
21

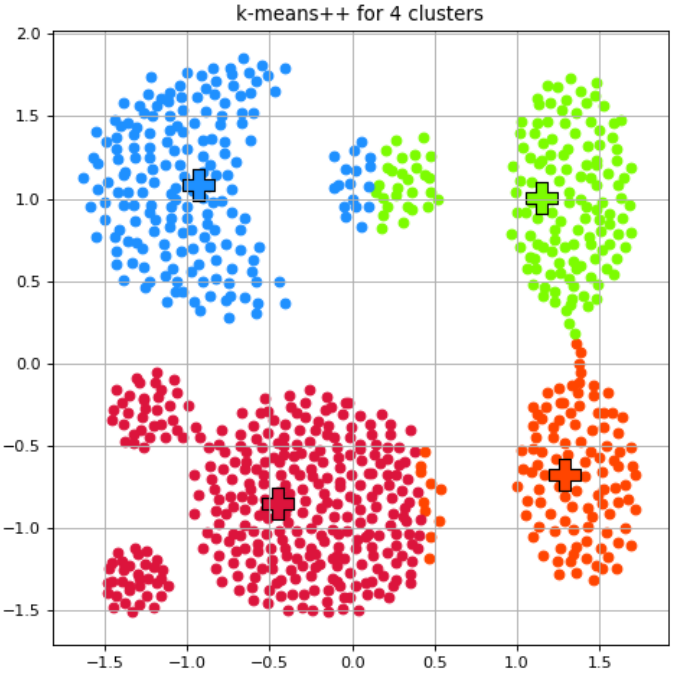

k-means++ silhouette coeffs for rp.data
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
3
6
9
12
15
18
21

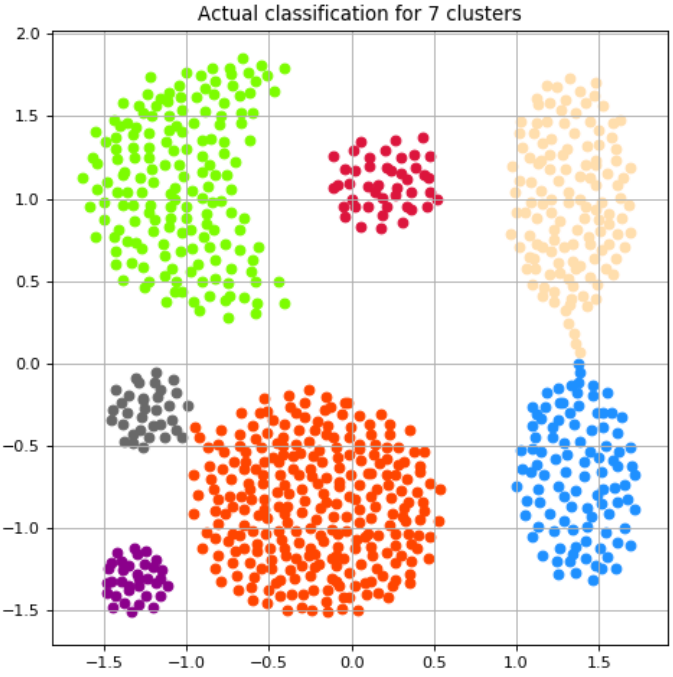

k-means++ error for dane_9D
1600
1400
1200
1000
800
600
400
3
6
9
12
15
18
21

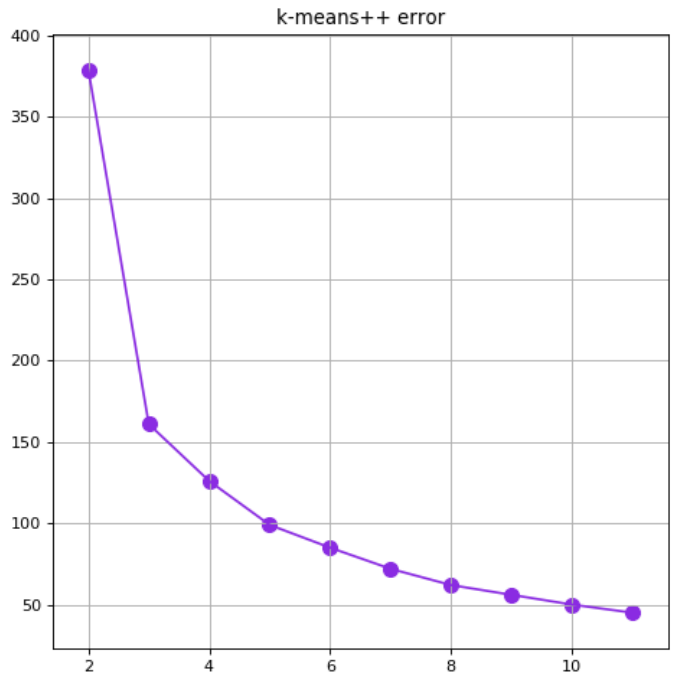

k-means++ silhouette coeffs for dane_9D
0.425
0.400
0.375
0.350
0.325
0.300
0.275
0.250
0.225
3
6
9
12
15
18
21

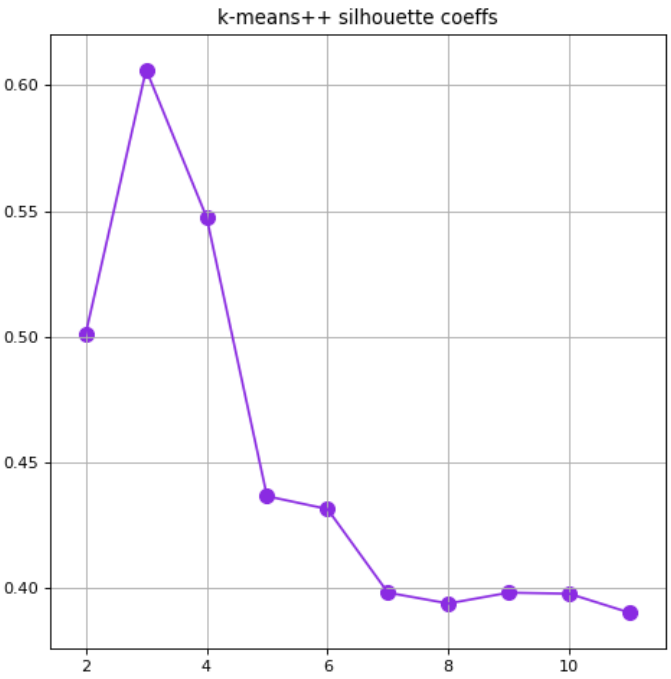

k-means++ error
800
600
400
200
2
4
6
8
10
12

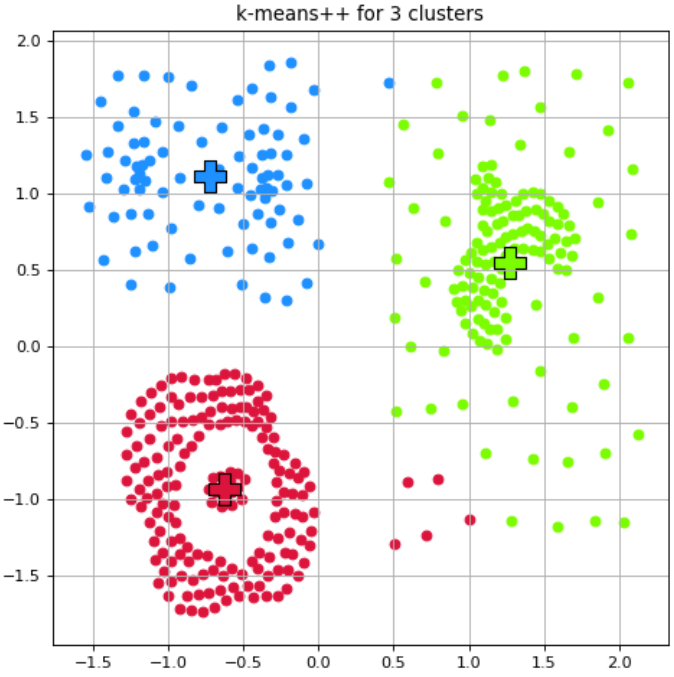

k-means++ silhouette coeffs
0.52
0.50
0.48
0.46
0.44
0.42
2
4
6
8
10
12

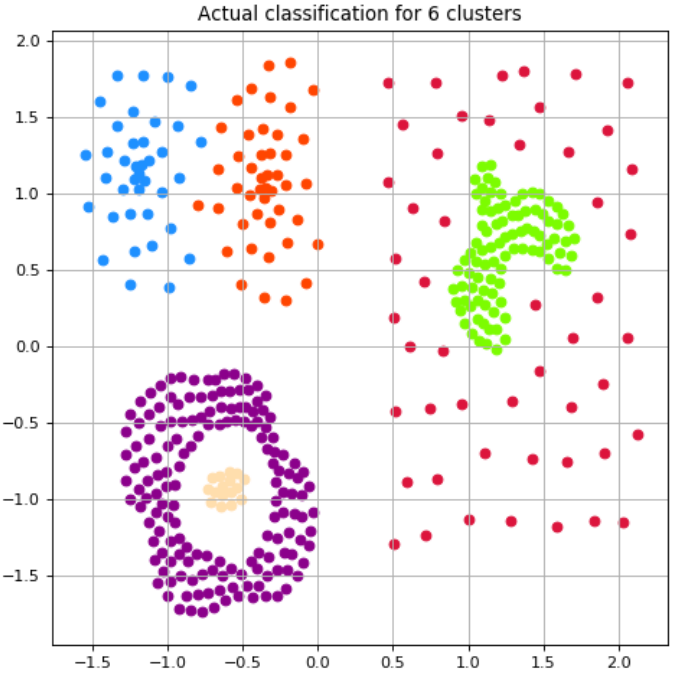

k-means++ for 4 clusters
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

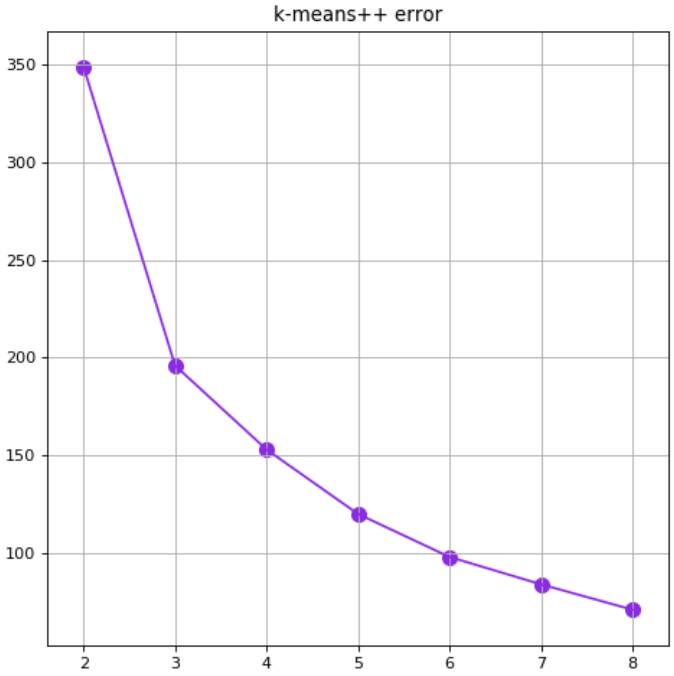

Actual classification for 7 clusters
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

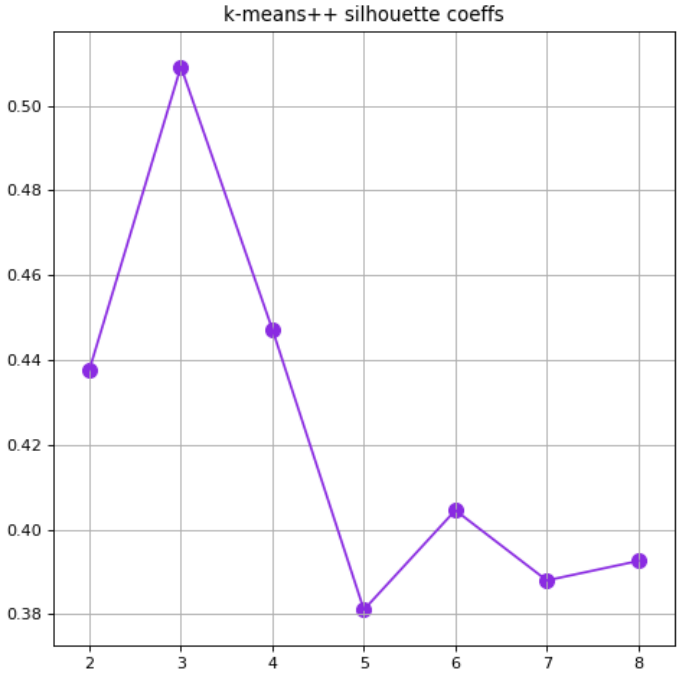

k-means++ error
400
350
300
250
200
150
100
50
2
4
6
8
10

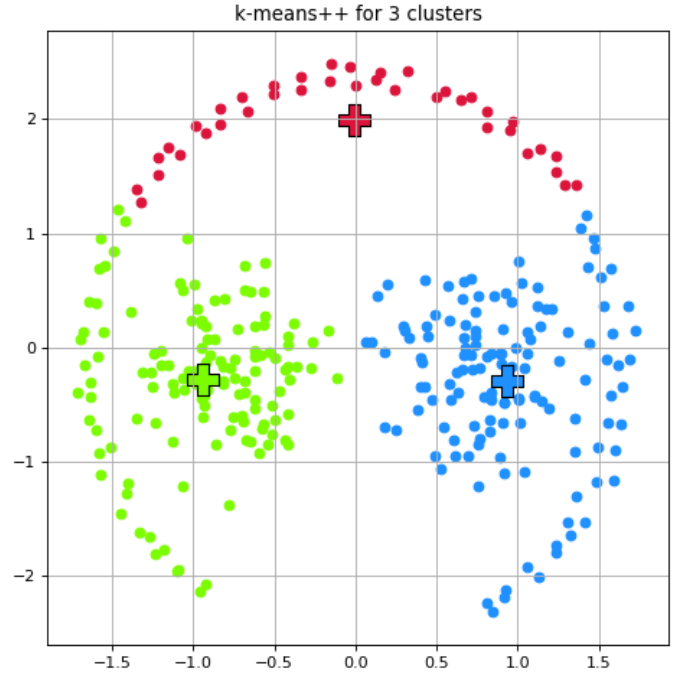

k-means++ silhouette coeffs
0.60
0.55
0.50
0.45
0.40
2
4
6
8
10

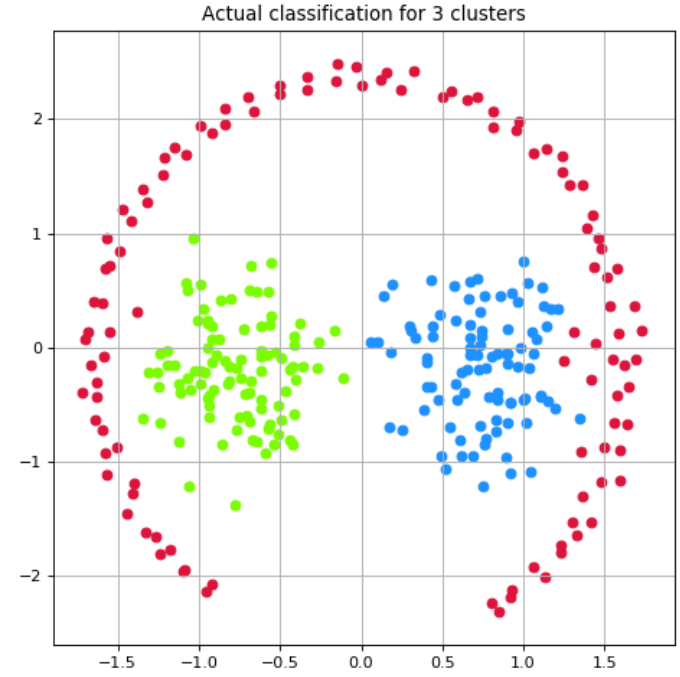

k-means++ for 3 clusters
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Actual classification for 6 clusters
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

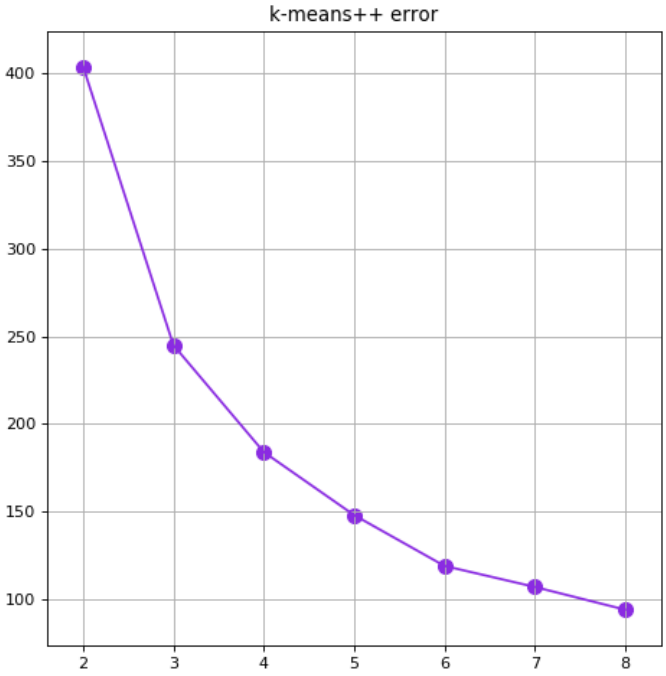

k-means++ error
350
300
250
200
150
100
2
3
4
5
6
7
8

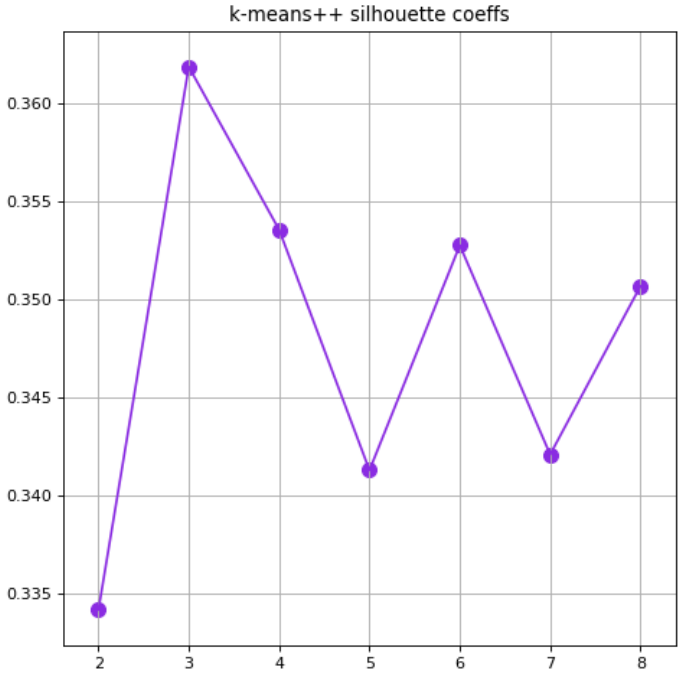

k-means++ silhouette coeffs
0.50
0.48
0.46
0.44
0.42
0.40
0.38
2
3
4
5
6
7
8

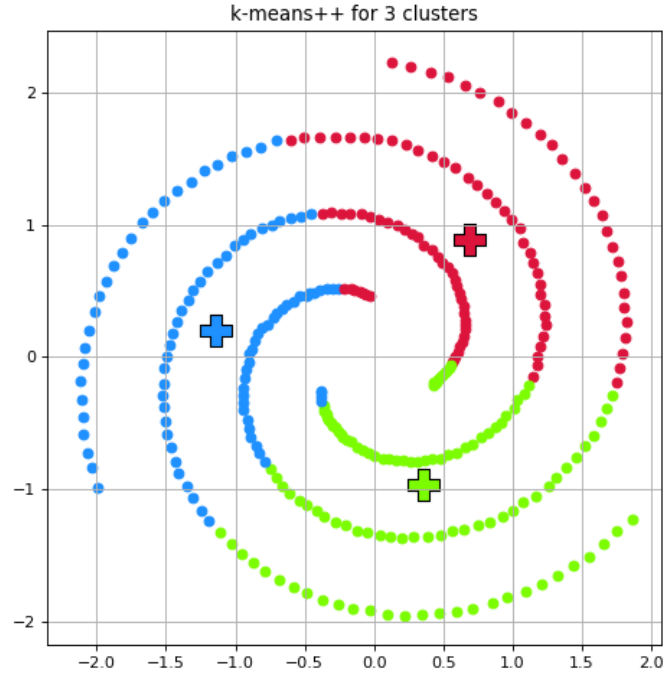

k-means++ for 3 clusters
2
1
0
−1
−2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

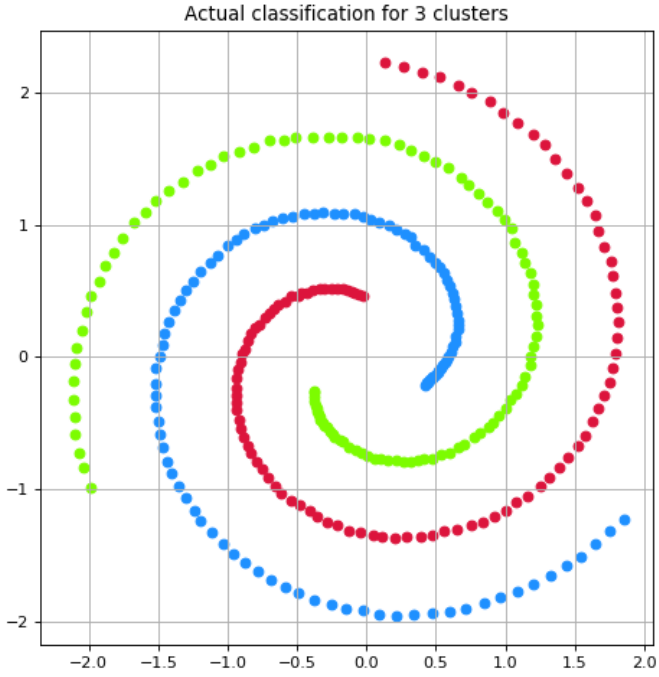

Actual classification for 3 clusters
2
1
0
−1
−2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

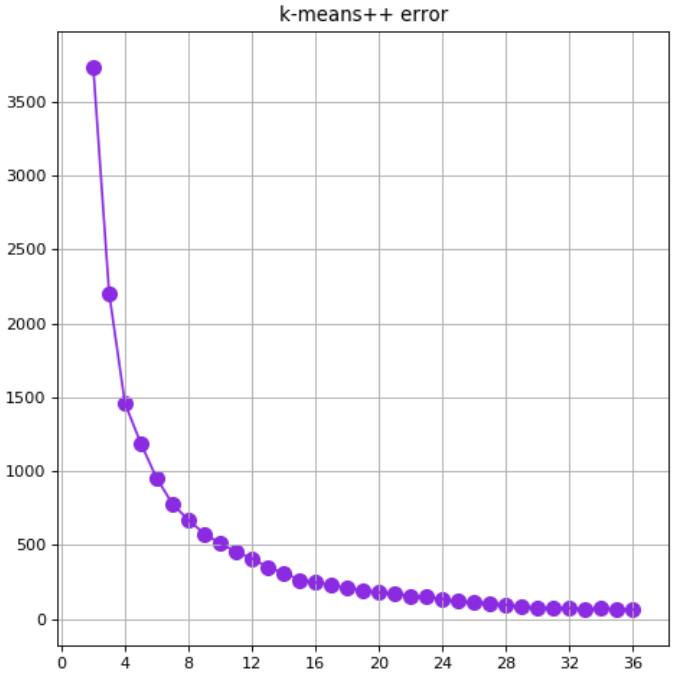
Katzper Michno, 31.05.2022, MPUM

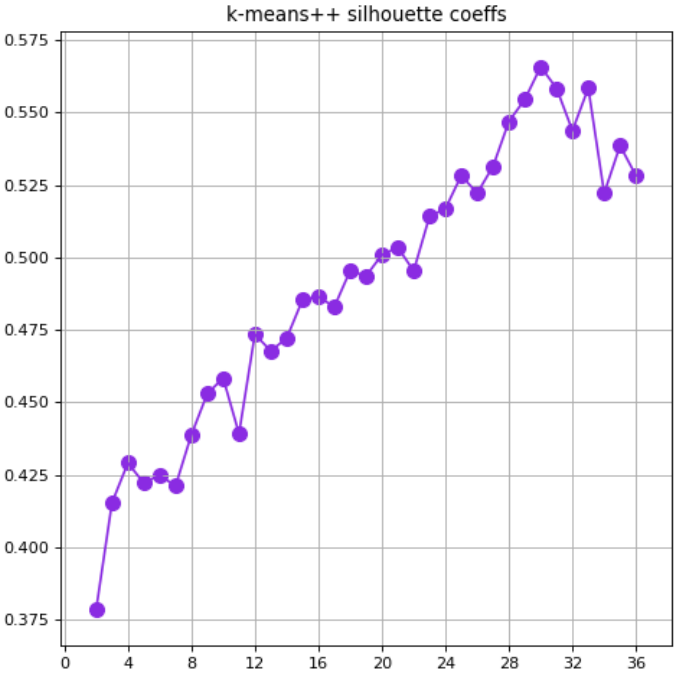


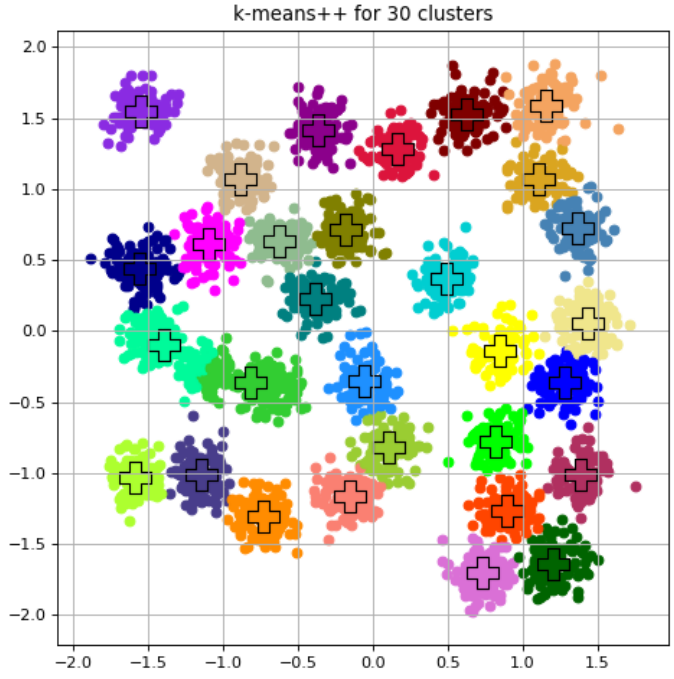


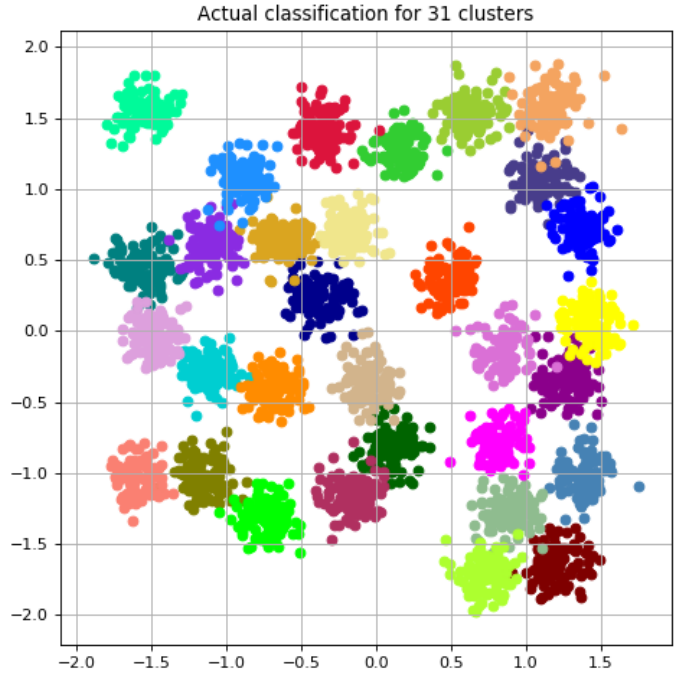


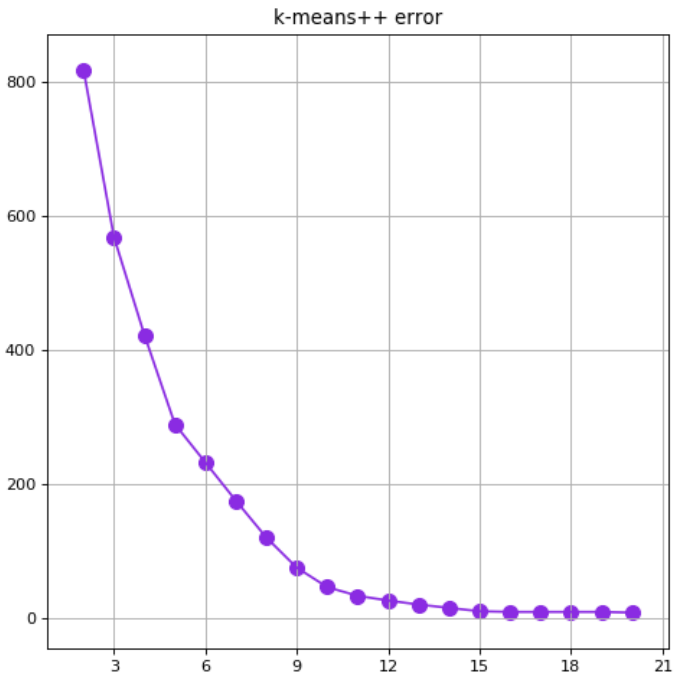


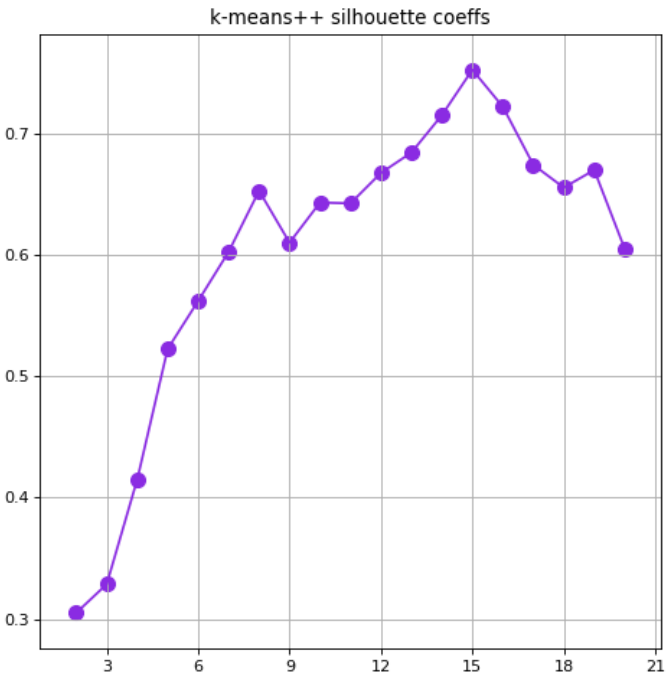

k-means++ error
3500
3000
2500
2000
1500
1000
500
0
0
4
8
12
16
20
24
28
32
36

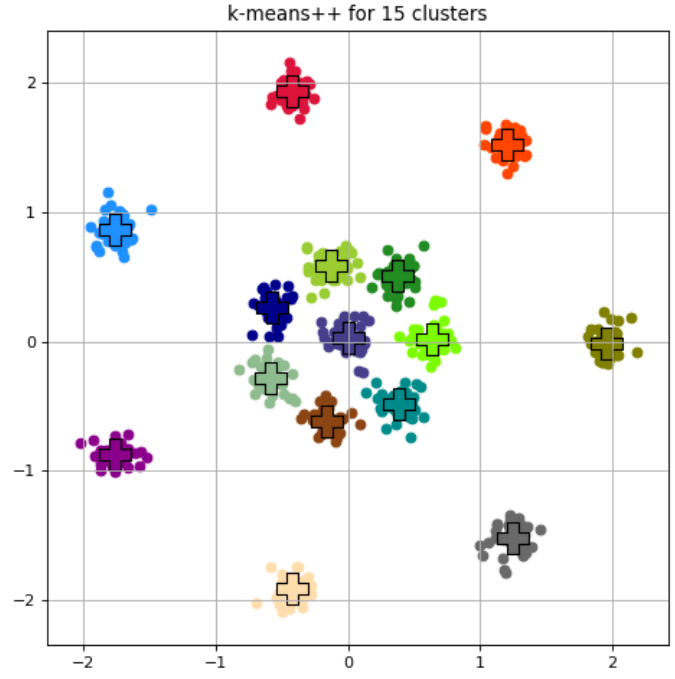

k-means++ silhouette coeffs
0.575
0.550
0.525
0.500
0.475
0.450
0.425
0.400
0.375
0
4
8
12
16
20
24
28
32
36

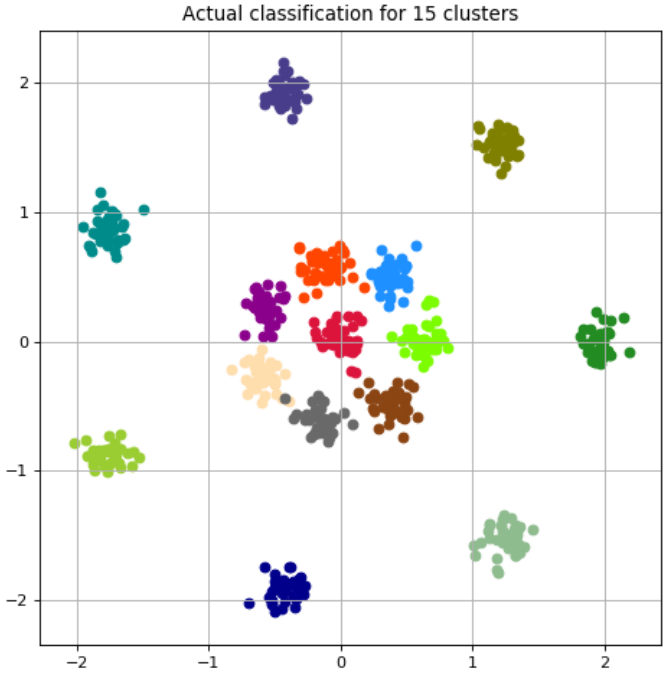

k-means++ for 30 clusters
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

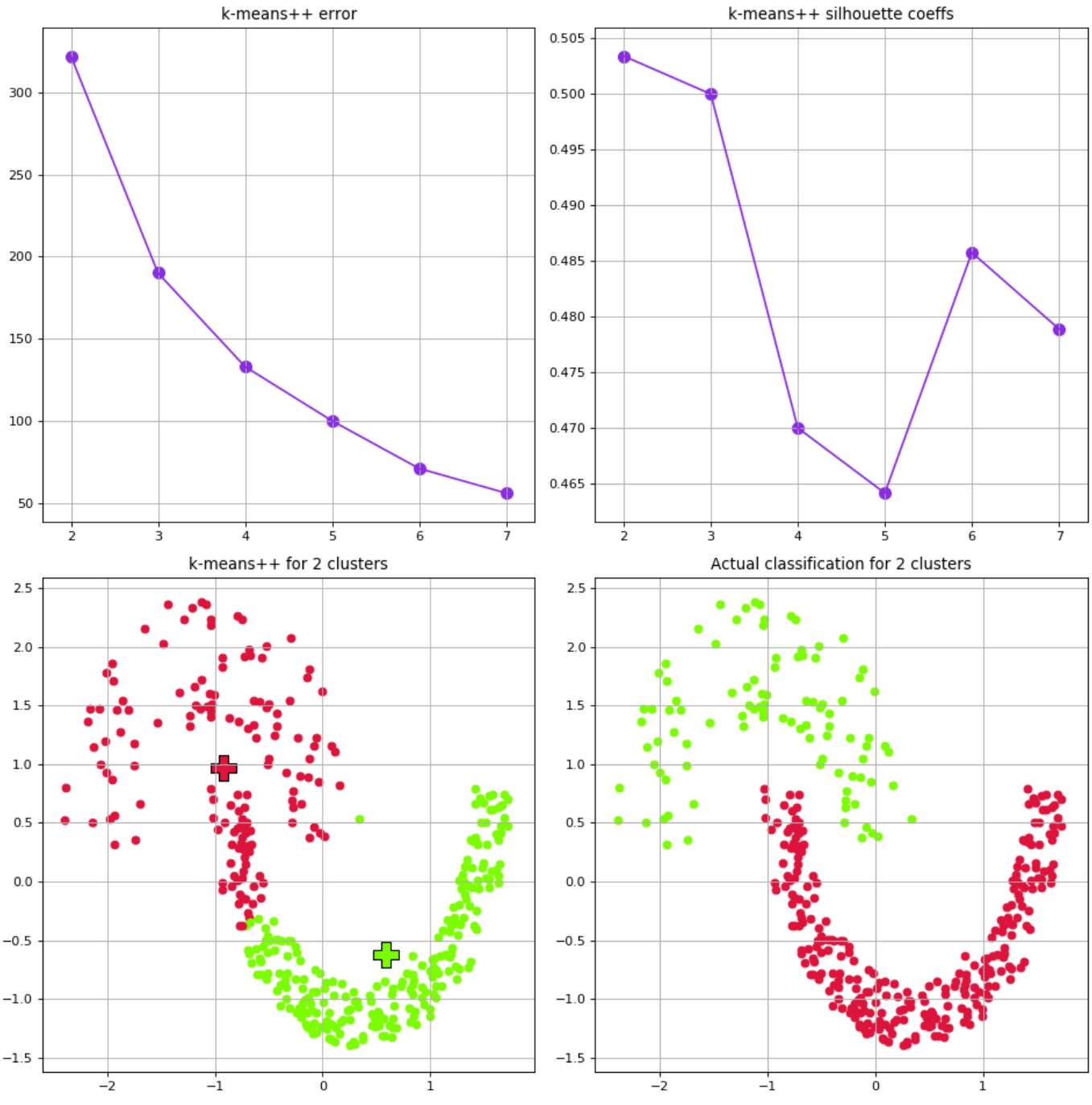

Actual classification for 31 clusters
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

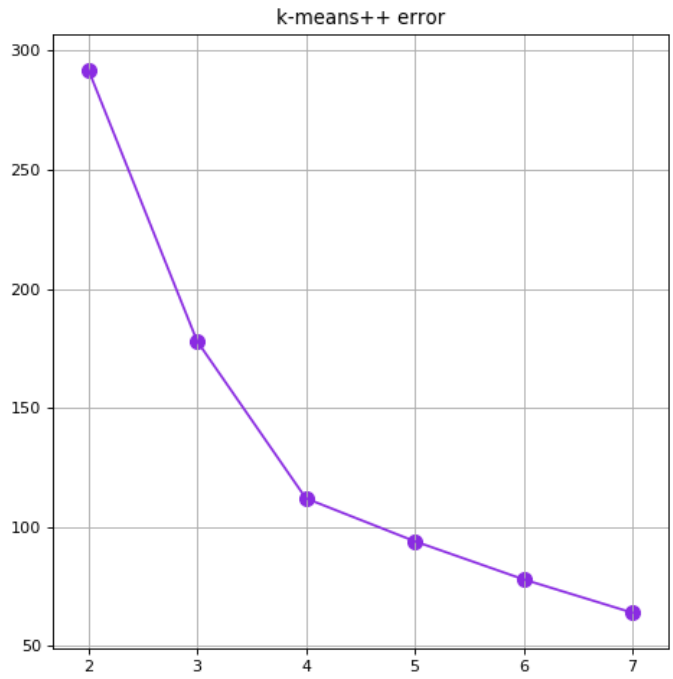

k-means++ error
800
600
400
200
0
3
6
9
12
15
18
21

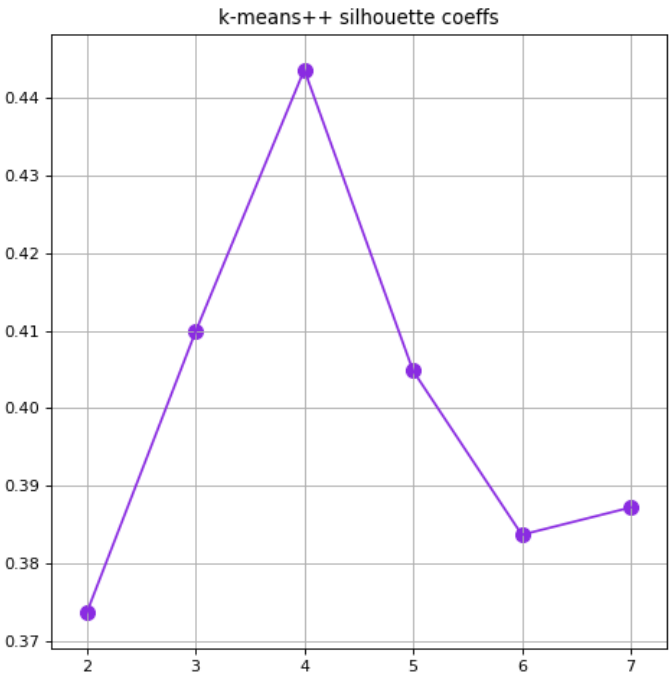

k-means++ silhouette coeffs
0.7
0.6
0.5
0.4
0.3
3
6
9
12
15
18
21

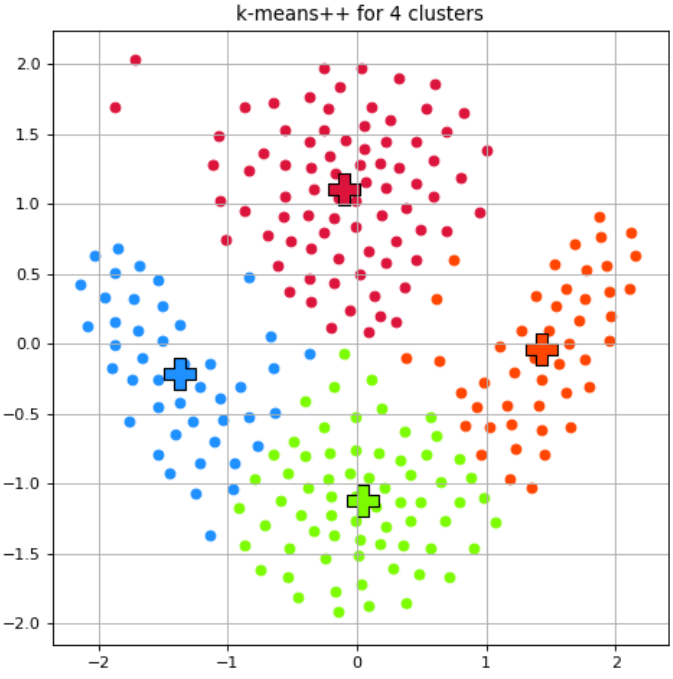

k-means++ for 15 clusters
2
1
0
−1
−2
−2
−1
0
1
2

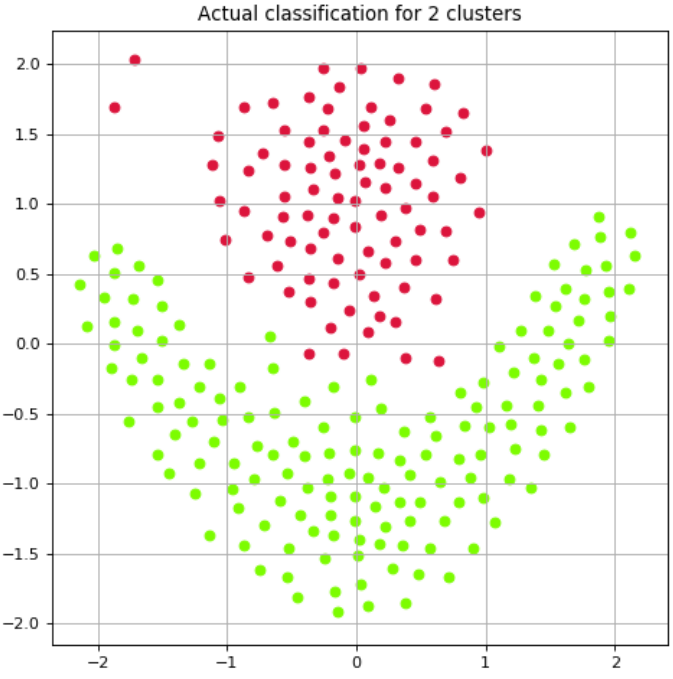

Actual classification for 15 clusters
2
1
0
−1
−2
−2
−1
0
1
2


k-means++ error
k-means++ silhouette coeffs
k-means++ for 2 clusters
Actual classification for 2 clusters


k-means++ error
300
250
200
150
100
50
2
3
4
5
6
7


k-means++ silhouette coeffs
0.44
0.43
0.42
0.41
0.40
0.39
0.38
0.37
2
3
4
5
6
7


k-means++ for 4 clusters
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2
−1
0
1
2


Actual classification for 2 clusters
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2
−1
0
1
2

Katzper Michno, 31.05.2022, MPUM

Klasteryzacja spektralna

W problemie klasteryzacji spektralnej zaczynamy od stworzenia pewnego grafu podobieństwa dla zbioru wierzchołków będącego zbiorem obserwacji. Zakładamy, że graf ten jest nieskierowany i ważony (czasami wszystkie wagi krawędzi będą równe 1). Zadaniem grafu podobieństwa jest w pewien sposób przedstawić, które pary obserwacji są do siebie podobne. Rozważamy trzy sposoby konstrukcji grafu podobieństwa:

- otoczka epsilonowa - dla ustalonego $\epsilon > 0$ łączymy krawędzią o wadze 1 obserwacje odległe od siebie o mniej niż ϵ w metryce euklidesowej
- k najbliższych sąsiadów - łączymy każdą obserwację z jej k najbliższymi obserwacjami w metryce euklidesowej, krawędziom nadajemy wagi według pewnej funkcji wagi zależnej od odległości pomiędzy punktami
- graf pełny - łączymy każdą parę obserwacji krawędzią o wadze wyznaczoną przez pewną funkcję wagi zależną od odległości w metryce euklidesowej

Będziemy używać dwóch typów funkcji wag:

- odwrotność odległości: $(x^{(i)}, x^{(j)}) \longrightarrow \frac{1}{||x^{(i)} - x^{(j)}||}$
- funkcja gaussa dla parametru τ: $(x^{(i)}, x^{(j)}) \longrightarrow \exp(-||x^{(i)} - x^{(j)}||^2/\tau^2)$

Następnym krokiem jest obliczenie znormalizowanego laplasjanu grafu i znalezienie jego wartości i wektorów własnych. Następnie, jeśli chcemy podzielić obserwacje na k klastrów, ustawiamy pierwsze k wektorów własnych (odpowiadające k najmniejszym wartościom własnym) w kolumny nowej macierzy $M \in R^{m \times k}$. Na końcu pozostaje zastosować na macierzy M inny algorytm klasteryzacji, gdzie nowymi obserwacjami są wiersze macierzy M. W naszym wypadku po znalezieniu macierzy M, stosujemy algorytm k średnich opisany w poprzednim rozdziale.

Przy rozważaniu klasteryzacji spektralnej, ze względu na to że jest ona w naszym przypadku metodą na poprawę działania algorytmu k średnich, pomijamy zestawy danych *dane_2D_5* oraz *dane_2D_6*, z którymi bardzo dobrze już poradziła sobie zwykła klasteryzacja k średnich, a są to zestawy kosztowne obliczeniowo, zwłaszcza w momencie obliczania wektorów własnych macierzy. Zamiast tego, skupimy się na poprawieniu rezultatów poprzednich algorytmów w miejscach, gdzie nie poradziły sobie zbyt dobrze.

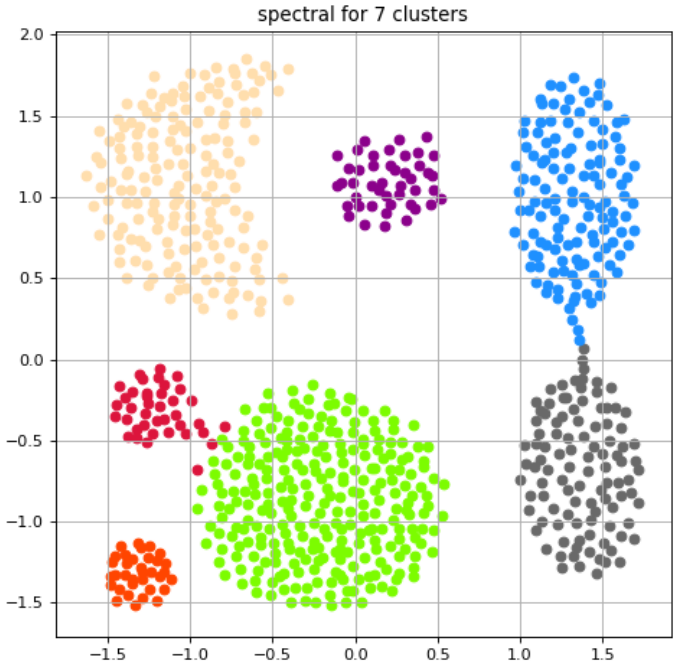
Dla wszystkich zestawów danych z przypisanymi klasami do obserwacji klasteryzacja spektralna została przetestowana na kilku zestawach parametrów:

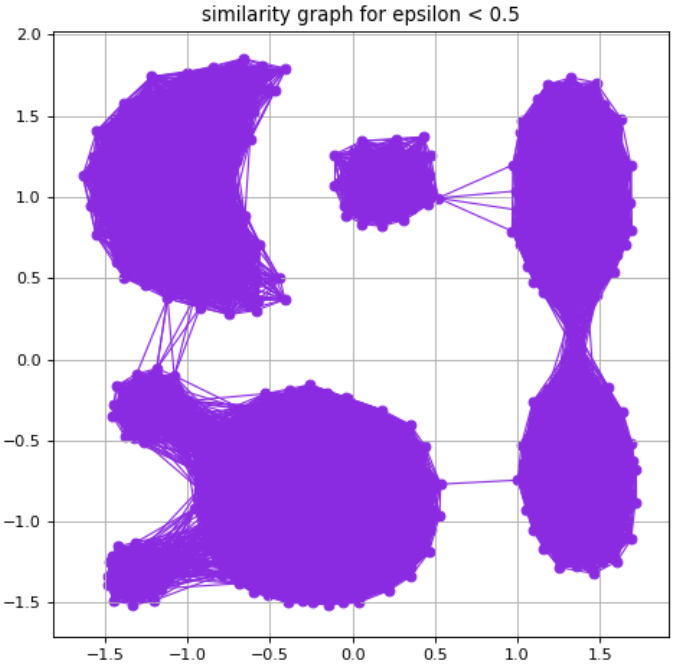
1. graf pełny z odwrotnością odległości
2. graf pełny z gaussem $\tau = 0.01$
3. graf pełny z gaussem $\tau = 0.1$
4. graf pełny z gaussem $\tau = 0.5$
5. graf pełny z gaussem $\tau = 1$
6. graf pełny z gaussem $\tau = 10$
7. otoczka epsilonowa z $\epsilon = 0.1$
8. otoczka epsilonowa z $\epsilon = 0.2$

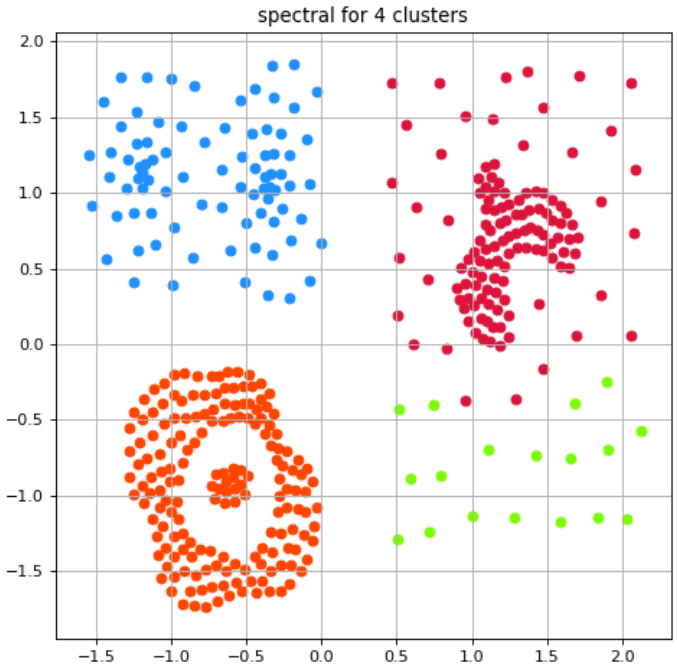
9. otoczka epsilonowa z $\epsilon = 0.3$
10. otoczka epsilonowa z $\epsilon = 0.35$
11. otoczka epsilonowa z $\epsilon = 0.5$
12. otoczka epsilonowa z $\epsilon = 0.7$
13. 3 najbliższych sąsiadów z odwrotnością odległości
14. 5 najbliższych sąsiadów z odwrotnością odległości
15. 10 najbliższych sąsiadów z odwrotnością odległości
16. 3 najbliższych sąsiadów z gaussem $\tau = 0.5$
17. 5 najbliższych sąsiadów z gaussem $\tau = 0.5$
18. 10 najbliższych sąsiadów z gaussem $\tau = 0.5$

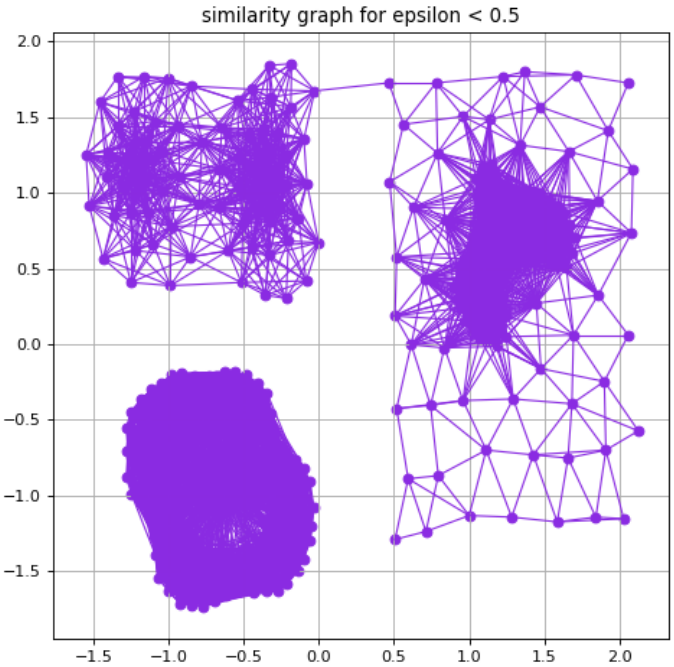
W poniższej tabeli i zostały przedstawione błędy klasyfikacji dla każdego zestawu danych dla najlepszego z powyższych zestawów parametrów. Poniżej znajdują się wizualizacje działania algorytmu dla najlepszych parametrów dla zestawów z danymi dwuwymiarowymi. Przedstawiony jest podział na klastry i graf podobieństwa, w którym grubość krawędzi symbolizuje jej wagę (cieńsza linia - mniejsza waga). W przypadku grafów pełnych pominięte zostały krawędzie o bardzo małych wagach.

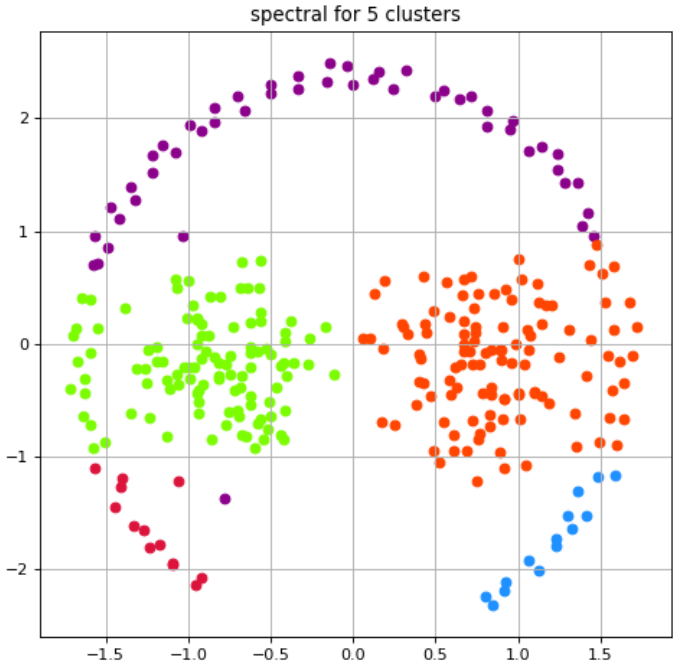
	przedział rozważanych k	najlepszy zestaw parametrów	optymalne k	błąd klasyfikacji
rp.data	[2, 7]	4	2	0.0841
dane_2D_1	[2, 12]	9	7	0.0046
dane_2D_2	[2, 11]	9	4	0.0967
dane_2D_3	[2, 8]	9	5	0.1894
dane_2D_4	[2, 8]	7	3	0.0
dane_2D_7	[2, 7]	10	2	0.0
dane_2D_8	[2, 7]	3	2	0.0489

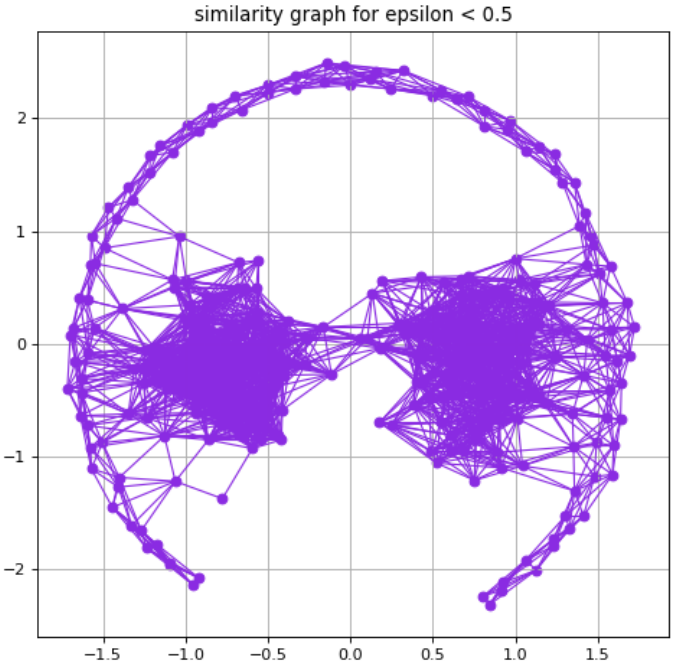

spectral for 7 clusters

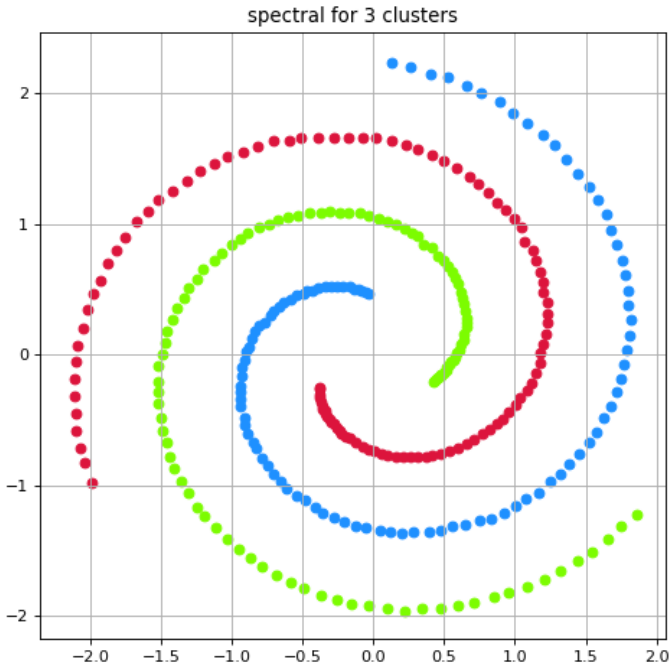

similarity graph for epsilon < 0.5

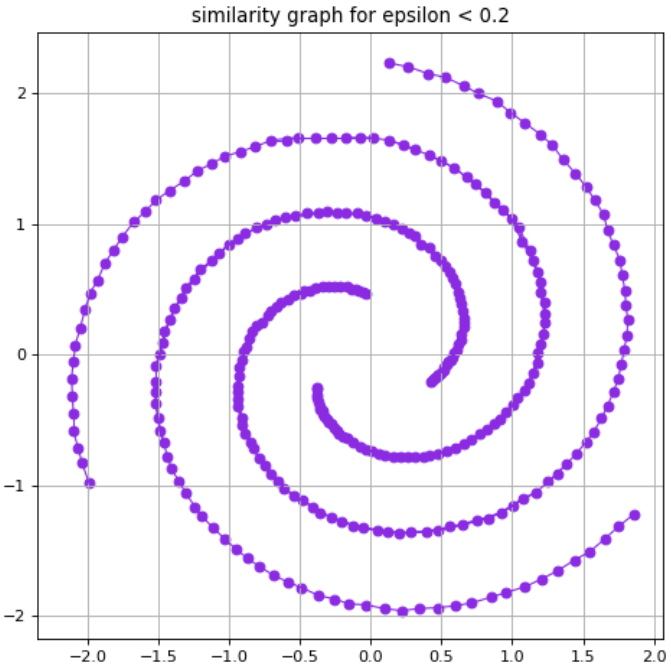

spectral for 4 clusters

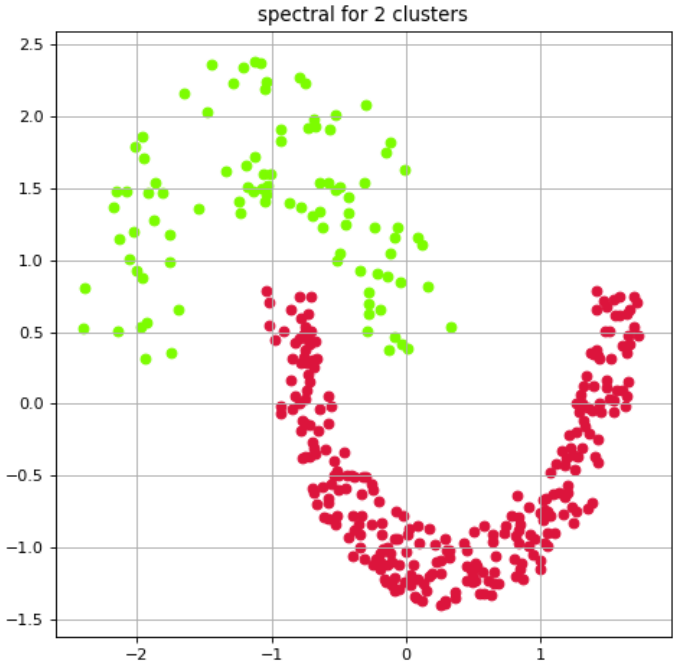

similarity graph for epsilon < 0.5

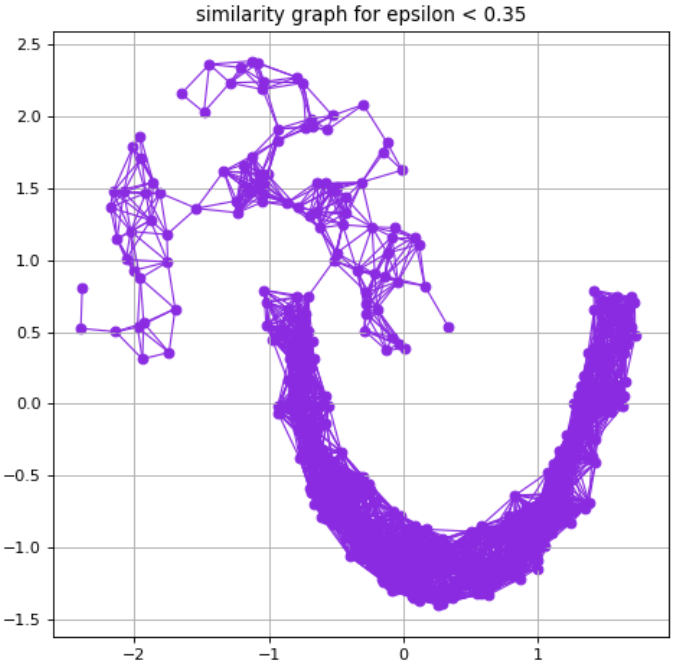

spectral for 5 clusters

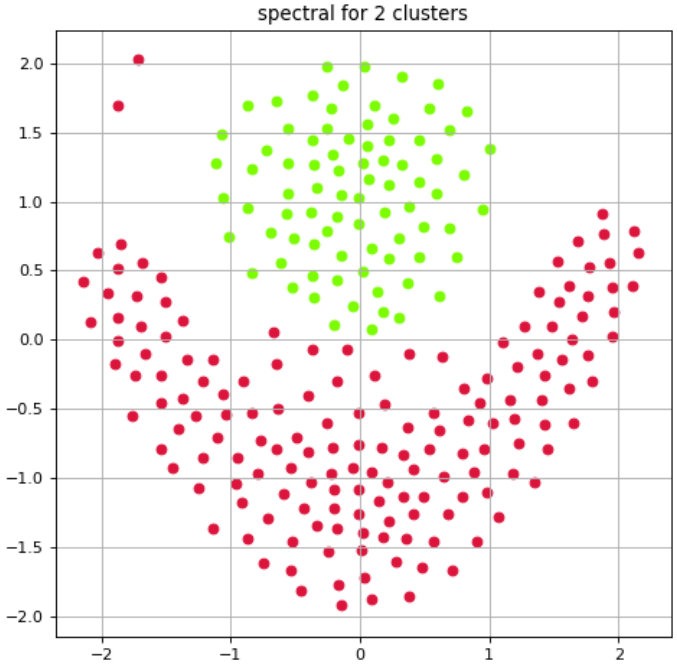

similarity graph for epsilon < 0.5

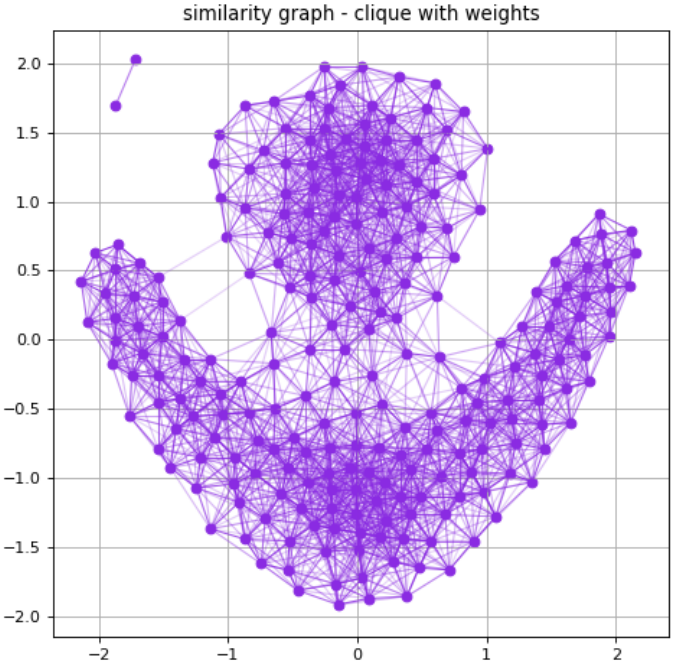

spectral for 3 clusters


similarity graph for epsilon < 0.2


spectral for 2 clusters


similarity graph for epsilon < 0.35

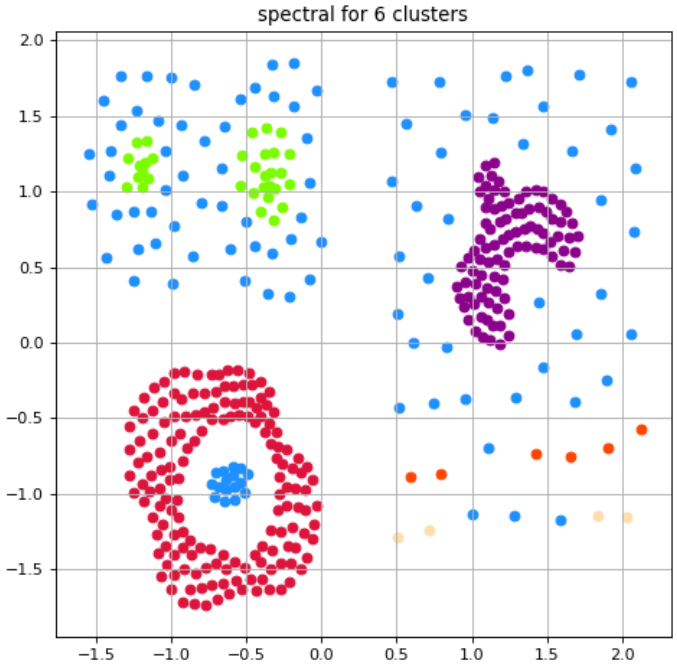

spectral for 2 clusters

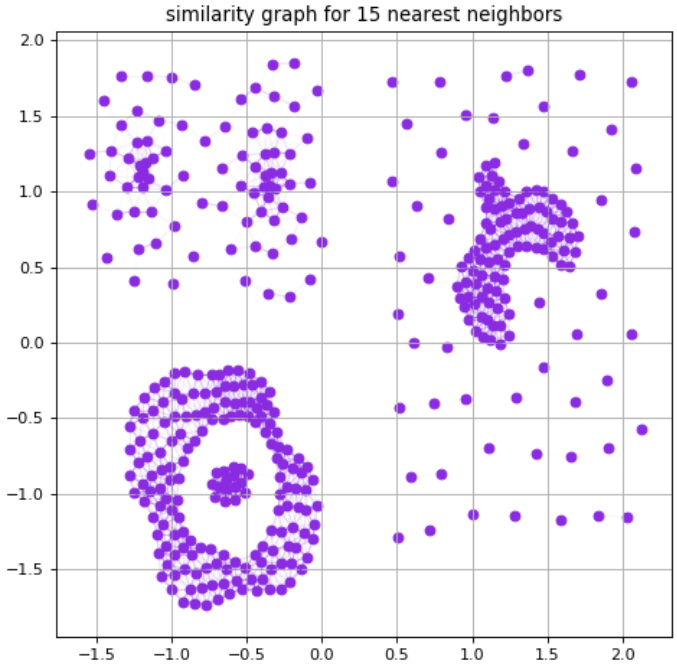

similarity graph - clique with weights

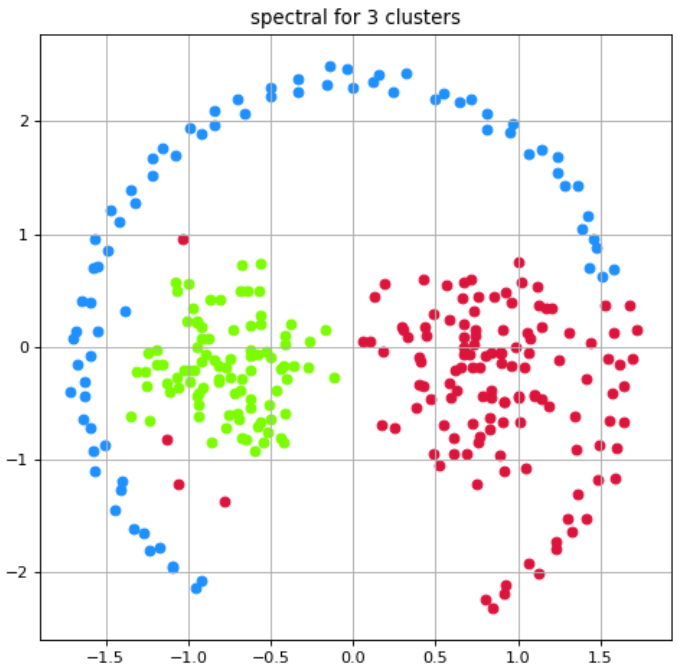
W poszukiwaniu jeszcze lepszych parametrów

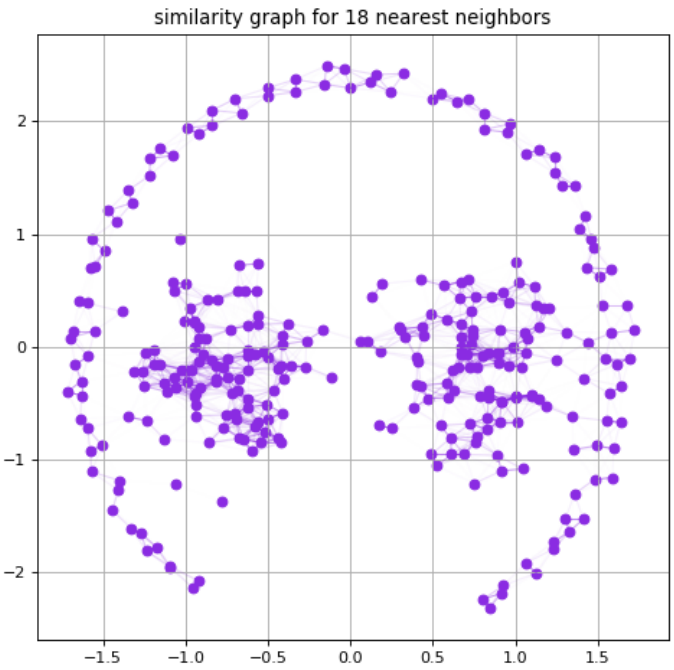
W większości zestawów danych przedstawione algorytmy dokonały satysfakcjonujących podziałów na liczbę klastrów różniącą się o nie więcej niż 1 od liczby klas. Pozostały dwa przykłady, dla których się to nie powiodło: *dane_2D_2* oraz *dane_2D_3*. Poniżej zostały przedstawione wyniki poszukiwania lepszych parametrów klasteryzacji spektralnej dla tych dwóch przypadków, przy ustaleniu liczby klastrów równej liczbie klas. Dodatkowo udało się nieznacznie poprawić błąd klasyfikacji w zestawie *dane_2D_8* przy ustaleniu liczby klastrów równej 2.

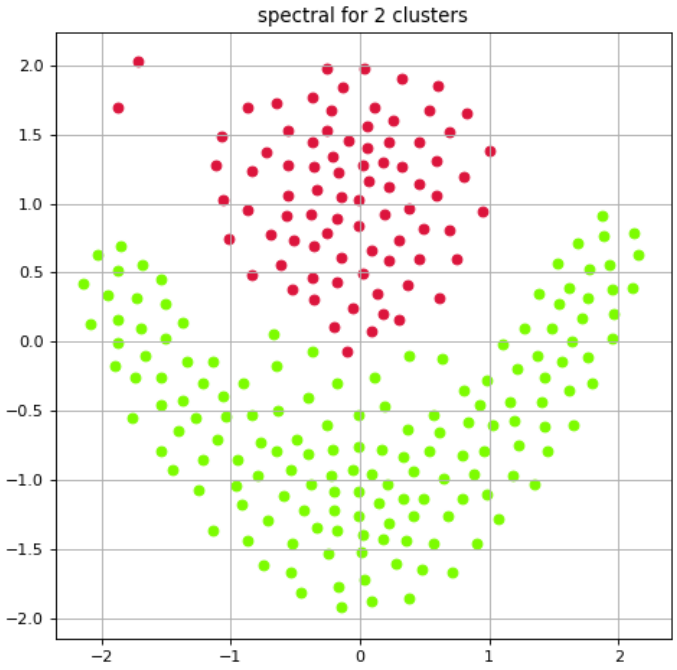
	typ grafu podobieństwa	parametry	błąd klasyfikacji
dane_2D_2	k najbliższych sąsiadów	$k = 15$, gauss $\tau = 0.1$	0.0705
dane_2D_3	k najbliższych sąsiadów	$k = 18$, gauss $\tau = 0.14$	0.1366
dane_2D_8	graf pełny	gauss $\tau = 0.35$	0.0247

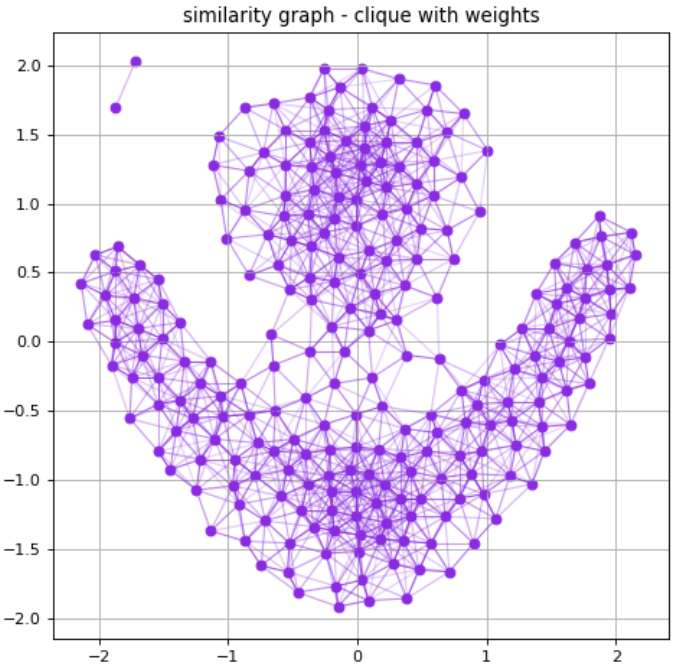





spectral for 3 clusters


similarity graph for 18 nearest neighbors


spectral for 2 clusters


similarity graph - clique with weights

dane_9D

Problem z zestawem danych *dane_9D* polega na tym, że nie zawiera on informacji o klasach obserwacji, a dodatkowo ze względu na swoją wielowymiarową naturę, rezultaty algorytmów klasteryzacji są trudne do wizualizacji. To co możemy zrobić, to oszacować z mniejszą lub większą dozą pewności, na ile klas powinny zostać podzielone dane z tego zestawu. Jak wyżej już opisaliśmy, algorytm k średnich na podstawie wskaźnika sylwetkowego stwierdził że najlepszą liczbą klastrów jest 2. Aby zweryfikować ten rezultat, dane te zostały sprawdzone przez algorytm klasteryzacji spektralnej przy użyciu wszystkich zestawów parametrów opisanych wcześniej. Okazało się, że w każdym przypadku współczynnik sylwetkowy osiągał maksimum w $k = 2$. Z tego możemy wnioskować że z dużym prawdopodobieństwem dane te pochodzą z dwóch klas.

Wnioski

Algorytm grupowania hierarchicznego osiągnął niski błąd klasyfikacji w zestawach danych z kulistymi klastrami oraz w zestawie *dane_2D_1*, gdzie jako jedyny osiągnął błąd równy 0. Także zestaw *dane_2D_4* okazał się być dobrze przystosowany do metody łączenia pojedynczego klastrów.
Algorytm k średnich radził sobie dobrze tylko przy klastrach kulistych, jednak udostępniał sposób na zautomatyzowanie poszukiwania optymalnej liczby klastrów bez znajomości klas obserwacji. W większości przypadków udało mu się zgadnąć odpowiednią liczbę klastrów, jednak nadal pozostało kilka zestawów z którymi musieliśmy sobie poradzić inaczej.
Pozostałe zestawy danych udało się podzielić na klastry w sposób satysfakcjonujący z pomocą klasteryzacji spektralnej i użyciu odpowiednich parametrów grafu podobieństwa. Najlepsze wyniki w porównaniu do pozostałych algorytmów udało się uzyskać w przypadku zestawu *dane_2D_2*, *dane_2D_3*, *dane_2D_7* oraz *dane_2D_8*
Poniżej znajduje się tabela podsumowująca najniższe wartości błędu klasyfikacji osiągnięte przez wszystkie trzy algorytmy.

	grupowanie hierarchiczne	k średnich	klasteryzacja spektralna
rp.data	0.1258	0.0857	0.0841
dane_2D_1	0.0	0.0971	0.0046
dane_2D_2	0.0788	0.1197	0.0705
dane_2D_3	0.2463	0.2419	0.1366
dane_2D_4	0.0	0.4462	0.0
dane_2D_5	0.0045	0.0053	—
dane_2D_6	0.0013	0.0008	—
dane_2D_7	0.2405	0.2327	0.0
dane_2D_8	0.3547	0.2967	0.0247