POLITECHNIKA WARSZAWSKA

METODY EKSPLOATACJI DANYCH

PROJEKT

„Porównanie algorytmu grupowania CURE i CLARANS”

Autor:

Jakub Kuczmarski

Warszawa 2023/24

1. Temat projektu

Tematem projektu jest implementacja algorytmów PAM, CLARA, CLARANS oraz CURE dla zadania grupowania, w celu porównania szybkości i efektywności ich działania.

2. Wstęp teoretyczny

2.1 Grupowanie

Grupowanie danych, znane również jako analiza skupień lub klastryzacja, to proces organizowania zbioru danych na podstawie ich podobieństwa. Celem grupowania jest znalezienie naturalnych struktur lub wzorców w danych, które umożliwią lepsze zrozumienie ich charakterystyki lub pomogą w podejmowaniu decyzji.

Podczas grupowania dane są dzielone na różne grupy lub klastry, gdzie obiekty wewnątrz każdego klastra są podobne do siebie pod względem pewnych cech. Grupowanie polega na minimalizacji podobieństwa między klastrami i maksymalizacji podobieństwa wewnątrz klastrów. Może ono pomóc w odkrywaniu ukrytych wzorców, identyfikowaniu podobnych grup obiektów oraz wszędzie tam, gdzie istnieje potrzeba organizacji i zrozumienia dużych zbiorów danych.

Istnieje wiele różnych metod grupowania danych, takich jak k-means, grupowanie hierarchiczne lub gęstościowe i wiele innych. Każda z tych metod ma swoje własne zasady i założenia dotyczące tworzenia klastrów. Algorytmy grupowania danych są szeroko stosowane w dziedzinach takich jak analiza danych, eksploracja danych, rozpoznawanie wzorców i wiele innych.

2.2 Algorytmy partycjonujące

Algorytmy partycjonujące dzielą zbiór danych na nienachodzące na siebie klastry. Są one oparte na idei przypisania obiektów do pewnej liczby klastrów w sposób optymalizujący określony kryterium. Przykładami algorytmów partycjonujących są:

- K-means: Jest to jeden z najpopularniejszych algorytmów partycjonujących. Działa na zasadzie iteracyjnego przypisywania obiektów do klastrów, minimalizując sumę kwadratów odległości między obiektami a centroidami klastrów.
- K-medoids: Jest podobny do K-means, ale zamiast centroidów używa obiektów rzeczywistych jako medoidów, które są najbardziej reprezentatywne dla klastra.

Algorytmy partycjonujące są często używane w przypadkach, gdy z góry określona liczba klastrów jest znana lub przypuszczana. Są one bardziej skuteczne dla dużych zbiorów danych, ale mogą być wrażliwe na początkowe ustawienia i odchylenia od normy.

Do grupu tego typu algorytmów należy m.in. algorytm PAM (Partitioning Around Medoids), CLARA (Clustering Large Applications), i CLARANS (Clustering Large Applications based on RANdomized Search). Wszystkie te trzy algorytmy, opierają się na medoidach jako punktach reprezentujących klastry i mają na celu znalezienie optymalnych konfiguracji klastrów, minimalizujących koszt (odległość między obiektami, a medoidami). Wybór konkretnego algorytmu zależy od rozmiaru danych, złożoności obliczeniowej i oczekiwanych wyników.

2.2.1 Algorytm PAM (Partitioning Around Medoids)

Algorytm PAM jest jednym z algorytmów partycjonujących, który wykorzystuje medoidy jako punkty reprezentatywne dla klastrów. Medoid to obiekt w klastrze, który jest najbardziej centralny lub reprezentatywny dla reszty obiektów w klastrze.

Schemat działania algorytmu PAM prezentuje się następująco:

1. Inicjalizacja: Wybierz początkowo k obiektów jako medoidy klastrów.
2. Przypisanie: Przypisz każdy obiekt do najbliższego medoidu.
3. Optymalizacja: Dla każdego klastra, spróbuj zamienić medoid na inny obiekt w klastrze i obliczaj koszt, czyli sumę odległości między obiektami a medoidami klastrów. Wybierz medoid, dla którego koszt jest najmniejszy. Powtarzaj ten krok, aż koszt nie ulegnie poprawie.

2.2.2 Algorytm CLARA (Clustering Large Applications)

Algorytm CLARA jest modyfikacją algorytmu PAM opracowaną dla dużych zbiorów danych. Ze względu na ograniczenia wydajnościowe, tradycyjny PAM może być trudny do zastosowania na dużych zbiorach danych. CLARA próbuje rozwiązać ten problem, stosując próbkowanie.

Schemat działania algorytmu CLARA prezentuje się następująco:

1. Losowe próbkowanie: Wybierz losową próbkę zawierającą n obiektów z oryginalnego zbioru danych.
2. PAM: Zastosuj kroki algorytmu PAM do próbki danych.
3. Ocena: Oblicz koszt klastrów dla próbki danych.
4. Powtarzanie: Powtórz kroki próbkowania, PAM i oceny wielokrotnie, a następnie wybierz wynik z najniższym kosztem klastrów.

2.2.3 Algorytm CLARANS ((Clustering Large Applications based on RANdomized Search)

Algorytm CLARANS to kolejna modyfikacja algorytmu PAM, która stosuje losową eksplorację przestrzeni rozwiązań w celu znalezienia lepszych medoidów. CLARANS jest bardziej elastyczny i eksploruje różne kombinacje medoidów, co prowadzi do lepszych wyników, ale może być bardziej czasochłonny.

Schemat działania algorytmu CLARANS prezentuje się następująco:

1. Wybór losowego obiektu: Wybierz losowo obiekt jako medoid dla klastra
2. Eksploracja: Wykonaj losowe zamiany medoidów w sąsiedztwie obecnej konfiguracji i obliczaj koszt klastrów dla każdej zmiany.
3. Akceptacja lub odrzucenie: Sprawdź, czy nowa konfiguracja medoidów prowadzi do mniejszego kosztu klastrów. Jeśli tak, zaakceptuj zmianę. W przeciwnym razie odrzuć zmianę
4. Powtarzanie: Powtórz kroki eksploracji i akceptacji/odrzucenia przez określoną liczbę iteracji lub do momentu znalezienia satysfakcjonującego wyniku.

2.3 Algorytmy hierarchiczne

Algorytmy hierarchiczne tworzą hierarchię klastrów, gdzie klastry mogą być podzielane na podklastry, tworząc strukturę drzewa. W zależności od sposobu konstruowania hierarchii, algorytmy hierarchiczne mogą być aglomeracyjne lub deglomeracyjne. Algorytmy hierarchiczne dzielą się na:

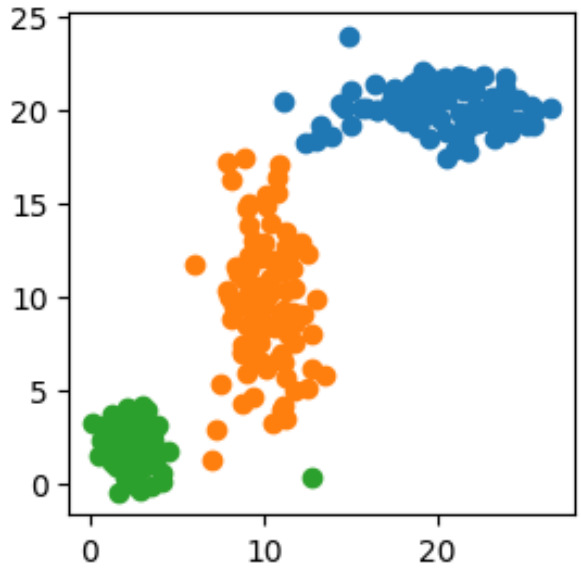
- Aglomeracyjne - każdy punkt ma na początku własny klaster i w kolejnych krokach klastry są łączone.
- Deaglomeracyjne – na początku wszystkie punkty są przypisane do jednego klastra. Następnie klaster ten jest dzielony na mniejsze klastry.

Algorytmy hierarchiczne oferują bardziej elastyczną strukturę hierarchii, ale mogą być bardziej złożone obliczeniowo i trudniejsze do interpretacji.

Algorytm CURE należy do grupy algorytmów hierarchicznych aglomeracyjnych.

2.3.1 Algorytm CURE

Algorytm CURE (Clustering Using Representatives) to algorytm grupowania danych, który łączy w sobie techniki hierarchicznego grupowania i partycjonowania. CURE został zaproponowany jako rozwiązanie dla danych o wysokiej wymiarowości, gdzie tradycyjne algorytmy grupowania mogą mieć trudności. W założeniu algorytm CURE dobrze sprawdza się w przypadku występowania punktów odstających i dla przykładów, w których klastry są rozproszone w różny sposób. Dla przykładu, na poniższym rysunku została przedstawiona klasteryzacja algorytmem CURE dla trzech klastrów na losowych danych wygenerowanych przez numpy. Można zauważyć, że klastry zostały stworzone w prawidłowy sposób.



Opis kroków algorytmu CURE:

1. Dla każdego punktu tworzony jest oddzielny klaster
2. Tworzone jest drzewo zakresowe zawierające wszystkie punkty
3. Dopóki nie zostanie osiągnięta wymagana liczba klastrów wykonywane są kroki:

3.4 Wybierany jest klaster z najmniejszą odległością do innego klastra(biorąc pod uwagę odległości pomiędzy punktami reprezentatywnymi

3.5 Wybierany jest najbliższy sąsiad tego klastra i klastry te są łączone

Opis składowych algorytmu:

- Obiekty reprezentatywne – dla każdego klastra wybierana jest określona liczba obiektów reprezentatywnych. Liczba ta jest taka sama dla każdego klastra z wyjątkiem początkowej fazy algorytmu, w której ustalona liczba jest większa od liczby punktów w klastrach.

- Min Heap – kopiec przechowujący minimalną wartość w korzeniu. Struktura wykorzystywana do przechowywania najmniejszych odległości punktów reprezentatywnych danego klastra do punktów reprezentatywnych innych klastrów.
- KD Tree – drzewo zakresowe używane do pobierania odległości pomiędzy punktami reprezentatywnymi.

Algorytm CURE ma na celu zwiększenie wydajności grupowania dla danych o wysokiej wymiarowości. Wykorzystuje kombinację partycjonowania i hierarchicznego grupowania, przy jednoczesnym wykorzystaniu reprezentantów klastrów, skalowania i korekty. CURE może być skutecznym narzędziem do grupowania dużych i złożonych zbiorów danych.

2.4 Miara jakości grupowania – „sylwetka" (ang. silhouette)

Jest to miara jakości grupowania, która ocenia, jak dobrze punkty danych są przypisane do swoich klastrów oraz jak dobrze klastry są oddzielone od siebie.

Miara silhouette jest obliczana dla każdego punktu danych i jest zdefiniowana następującym wzorem:

$$S_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

gdzie:

- a_i to średnia odległość punktu i do innych punktów w tym samym klastrze,
- b_i to średnia odległość punktu i do punktów w najbliższym sąsiednim klastrze.

Wartości silhouette mieszczą się w zakresie od -1 do 1. Wyższe wartości oznaczają lepsze przypisanie punktów do klastrów, gdzie wartość bliska 1 wskazuje na dobrze zdefiniowane klastry, a wartość bliska -1 sugeruje, że punkty mogą być lepiej przypisane do innego klastra. Wartość bliska 0 oznacza, że punkt jest blisko granicy między klastrami.

2.4 Odległość między punktami

Miara Euklidesowa i Manhattan są dwiema popularnymi miarami służącymi do obliczania odległości między dwoma punktami w przestrzeni.

2.5.1 Odległość Euklidesowa

Jest to odległość geometryczna w przestrzeni wielowymiarowej.

Wyraża się wzorem:

$$d(A,B) = \sqrt{(x_{1A} - x_{1B})^2 + (x_{2A} - x_{2B})^2 + \ldots + (x_{nA} - x_{nB})^2} = \sqrt{\sum_{i=1}^{n}((x_{iA} - x_{iB})^2)}$$

2.5.3 Odległość Manhattan

Miara ta tłumi wpływ pojedynczych dużych różnic (przypadków odstających) ponieważ nie podnosi się ich do kwadratu.

Wyraża się wzorem

$$d(A,B) = \sum_{i}^{n} |x_{iA} - x_{iB}|$$

3 Opis zbioru danych

Do realizacji projektu wybrano ogólnodostępny zbiór danych zawierający informację na temat sieci Starbucks działających na świecie:

Nazwa	Liczba punktów - współrzędnych (x,y)	Liczba atrybutów
Starbucks Locations Worldwide	25 600	13

Zbiór danych zawiera wiersz dla każdej lokalizacji punktu Starbucks. Każdy wiersz zawiera szerokość i długość geograficzną danego punktu. Zbiór zawiera m.in. informacje na temat nazwy danego sklepu, miasta, regionu, kraju, w którym się znajduje, a także strefy czasowej. W projekcie zostały wykorzystane kolumny z szerokością i długością geograficzną.

4 Przeprowadzone eksperymenty

W ramach eksperymentów dokonane zostanie porównanie szybkości oraz dokładności działania (miara silhouette) algorytmu CURE oraz CLARANS w zależności od wielkości zbioru danych i liczby klastrów. Dodatkowo zbadany zostanie wpływ sposobu wyznaczania odległości między punktami (odległość euklidesowa lub Manhattan) oraz doboru poszczególnych parametrów algorytmów PAM, CLARA i CLARANS na otrzymane wyniki grupowania.

Eksperymenty dokonano na odpowiednio zmodyfikowanym zbiorze danych „Starbucks Locations Worldwide". Do grupowania danych wykorzystano ostatnie dwie kolumny - longitude (długość geograficzna) i latitude (szerokość geograficzna) – są to dwie współrzędne używane do określania położenia geograficznego danej restauracji na powierzchni Ziemi.

4.1 Rodzaj odległości między punktami

W tej części projektowej zbadano wpływ rodzaju wyznaczanej odległości między punktami podczas procesu grupowania, na otrzymane wyniki dla algorytmów PAM, CLARA i CLARANS. Eksperymenty przeprowadzono dla 10 000 przykładów i 3 klastrów.

Algorytm PAM

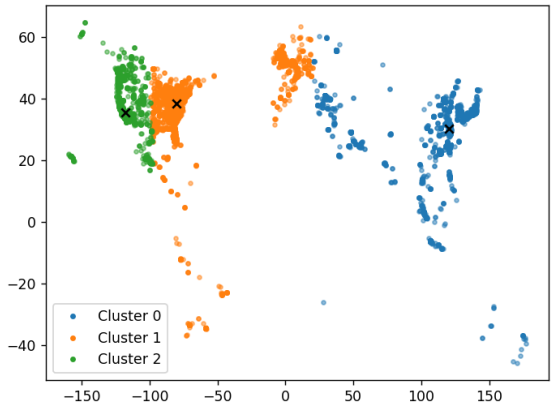
Otrzymane wyniki dla algorytmu PAM prezentuje tabela 1.

Wizualizacja wyników grupowania dla odległości euklidesowej prezentuje rysunek 1, a dla odległości Manhattan rysunek 2.

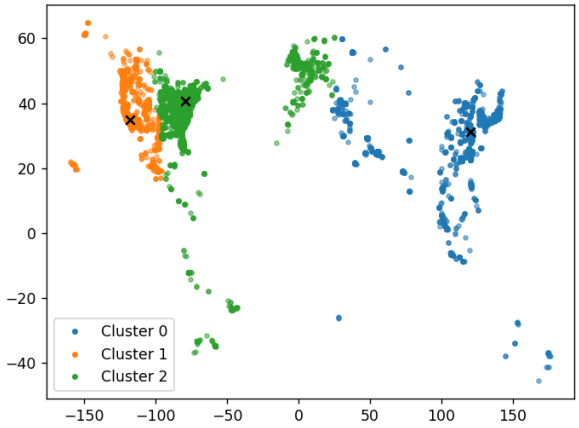
Tabela 1 Wyniki grupowania dla algorytmu PAM w zależności od rodzaju wyliczanej odległości między punktami.

Odległość	Miara Silhouette	Czas grupowania [s]	Koszt grupowania	Uzyskane medoidy	Liczba klastrów
Euklidesowa	0.4744	618.42	190623.43	[120.12, 30.34] [-80.0, 38.4] [-117.67, 35.63]	3
Manhattan	0.4597	534.28	229269.32	[120.52, 31.26] [-117.93, 34.92]	

				[-79.18, 40.61]	



Rys. 1 Grupowanie algorytmem PAM przy pomocy odległości euklidesowej



Rys. 2 Grupowanie algorytmem PAM przy pomocy odległości Manhattan

Algorytm CLARA

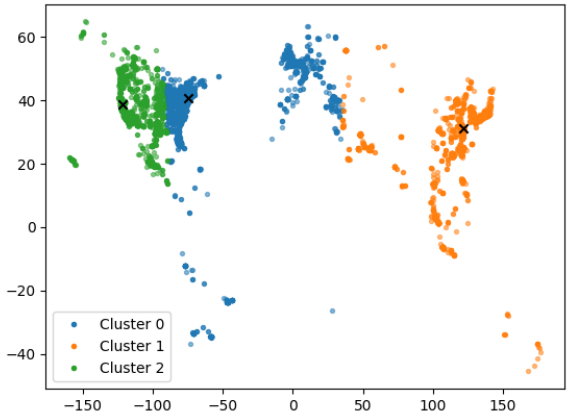
Nastawy parametrów: repeats = 2, num_representative_sample = 40*2*k

Otrzymane wyniki dla algorytmu CLARA prezentuje tabela 2.

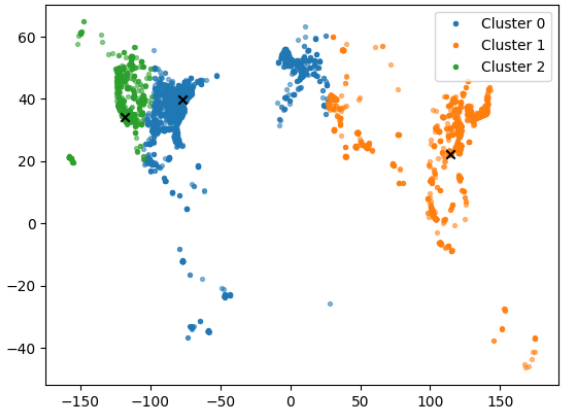
Wizualizacja wyników grupowania dla odległości euklidesowej prezentuje rysunek 3, a dla odległości Manhattan rysunek 4.

Tabela 2 Wyniki grupowania dla algorytmu CLARA w zależności od rodzaju wyliczanej odległości między punktami.

Odległość	Miara Silhouette	Czas grupowania [s]	Koszt grupowania	Uzyskane medoidy	Liczba klastrów
Euklidesowa	0.6993	0.03	233962,57	[121.41, 31.28] [-121.22, 38.79] [-74.61, 40.58]	3
Manhattan	0.6704	0.03	278436.03	[114.22, 22.28] [-118.35, 34.2] [-76.73, 39.98]	



Rys. 3 Grupowanie algorytmem CLARA przy pomocy odległości euklidesowej



Rys. 4 Grupowanie algorytmem CLARA przy pomocy odległości Manhattan

Algorytm CLARANS

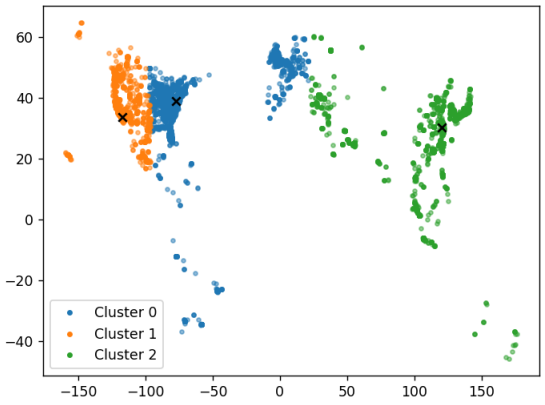
Nastawy parametrów: numlocal = 10, maxneightbor = 10

Otrzymane wyniki dla algorytmu CLARANS prezentuje tabela 3.

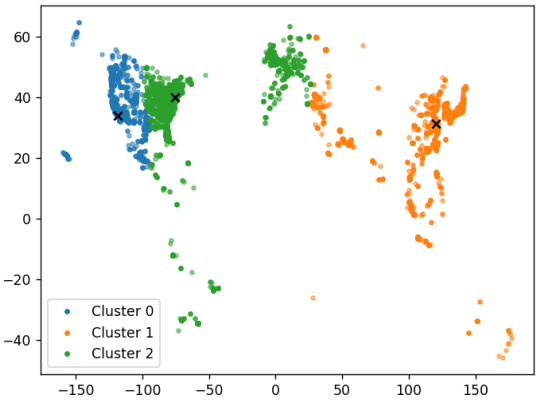
Wizualizacja wyników grupowania dla odległości euklidesowej prezentuje rysunek 5, a dla odległości Manhattan rysunek 6.

Tabela 3 Wyniki grupowania dla algorytmu CLARANS w zależności od rodzaju wyliczanej odległości między punktami.

Odległość	Miara Silhouette	Czas grupowania [s]	Koszt grupowania	Uzyskane medoidy	Liczba klastrów
Euklidesowa	0.6929	2.2	194128.3	[-77.17, 39.11] [-116.97, 33.8] [120.18, 30.33]	3
Manhattan	0.6590	2.86	234406.82	[-118.38, 33.93] [120.42, 31.34] [-75.17, 39.95]	



Rys. 5 Grupowanie algorytmem CLARANS przy pomocy odległości euklidesowej



Rys. 6 Grupowanie algorytmem CLARA przy pomocy odległości Manhattan

Zestawienie wyników dla wszystkich algorytmów zamieszczono w tabeli 4.

Tabela 4 Wyniki grupowania dla poszczególnych algorytmów w zależności od rodzaju wyliczanej odległości między punktami.

Odległość	Miara Silhouette	Czas grupowania [s]	Koszt grupowania	Algorytm grupowania
Euklidesowa	0.4744	618.42	190623.43	PAM
	0.6993	0.03	233962,57	CLARA
	0.6929	2.2	194128.3	CLARANS
Manhattan	0.4597	534.28	229269.32	PAM
	0.6704	0.03	278436.03	CLARA
	0.6590	2.86	234406.82	CLARANS

Algorytm CURE

W ramach testów nie zostały przetestowane inne miary odległości dla algorytmu CURE.

WNIOSKI

Analizując otrzymane wyniki związane ze sposobem wyliczania odległości między punktami widać dobrze, że zastosowanie odległości euklidesowej daje lepsze wyniki (pozwala lepiej pogrupować dane) w porównaniu z odległością Manhattan.

W związku z tym do dalszych testów wykorzystana zostanie odległość euklidesowa.

4.2 Liczba klastrów

W tej części projektowej zbadano wpływ liczby klastrów na proces grupowania i otrzymane wyniki dla algorytmów PAM, CLARA, CLARANS i CURE. Eksperymenty przeprowadzono dla 10 000 wartości, odległość euklidesowa

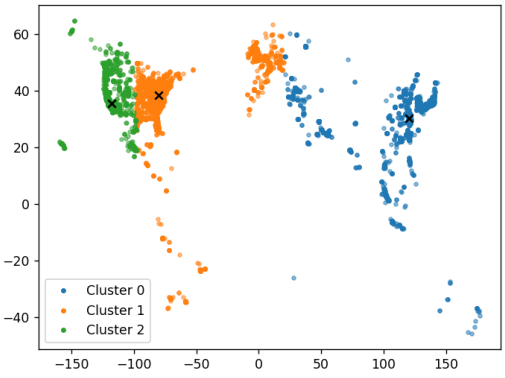
Algorytm PAM

Otrzymane wyniki dla algorytmu PAM prezentuje tabela 5.

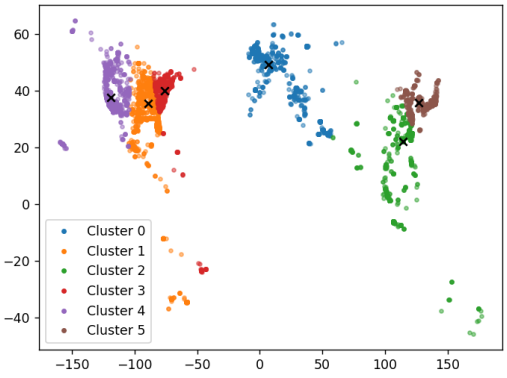
Wizualizacja wyników grupowania dla określonej liczby klastrów prezentuje rysunek 7,8 i 9.

Tabela 5 Wyniki grupowania dla algorytmu PAM w zależności od rodzaju wyliczanej odległości między punktami.

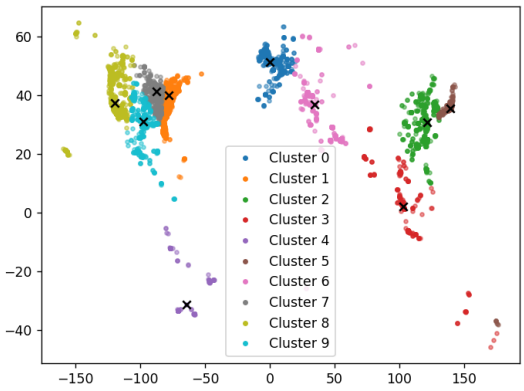
Liczba klastrów	Miara Silhouette	Czas grupowania [s]	Koszt grupowania	Uzyskane medoidy
3	0.4744	618.42	190623.43	[120.12, 30.34] [-80.0, 38.4] [-117.67, 35.63]
6	0.6227	1242.05	104214.33	[6.99, 49.24] [-88.86, 35.68] [114.13, 22.26] [-76.3, 40.13] [-118.99, 37.65] [127.12, 35.83]
10	0.7646	2384.36	70060.38	[-0.12, 51.53] [-78.09, 40.06] [121.33, 30.76] [103.32, 2.04] [-64.19, -31.41] [139.38, 35.66] [34.58, 36.81] [-87.73, 41.5] [-119.66, 37.33] [-97.41, 31.12]



Rys. 7 Grupowanie algorytmem PAM dla k=3



Rys. 8 Grupowanie algorytmem PAM dla k-6



Rys. 9 Grupowanie algorytmem PAM dla k=10

Algorytm CLARA

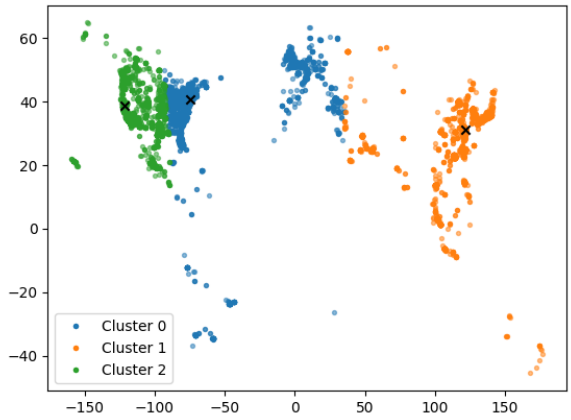
Nastawy parametrów: repeats = 2, num_representative_sample = 40*2*k

Otrzymane wyniki dla algorytmu CLARA prezentuje tabela 6.

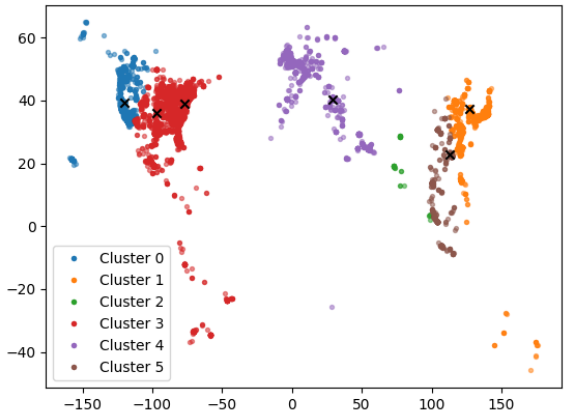
Wizualizacja wyników grupowania dla określonej liczby klastrów prezentuje rysunek 10, 11 i 12.

Tabela 6 Wyniki grupowania dla algorytmu CLARA w zależności od rodzaju wyliczanej odległości między punktami.

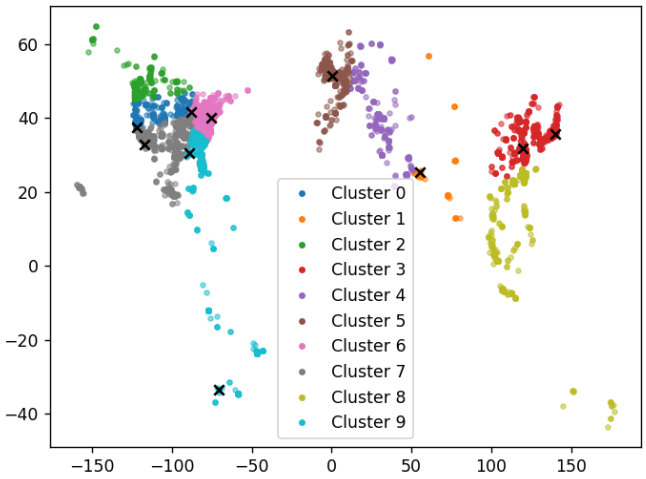
Liczba klastrów	Miara Silhouette	Czas grupowania [s]	Koszt grupowania	Uzyskane medoidy
3	0.6993	0.03	233962.57	[121.41, 31.28] [-121.22, 38.79] [-74.61, 40.58]
6	0.5092	0.06	176297.42	[29.06, 40.19] [-77.08, 38.81] [-97.06, 36.13] [-120.04, 39.37] [127.11, 37.37] [113.08, 22.97]
10	0.3937	0.14	129855.28	[119.98, 31.81] [55.35, 25.25] [-70.57, -33.4] [-88.88, 30.44] [139.74, 35.63] [-122.04, 37.53] [-87.66, 41.68] [-75.23, 40.03] [0.27, 51.49] [-117.02, 32.78]



Rys. 10 Grupowanie algorytmem CLARA dla k=3



Rys. 11 Grupowanie algorytmem CLARA dla k-6



Rys. 12 Grupowanie algorytmem CLARA dla k=10

Algorytm CLARANS

Nastawy parametrów: numlocal = 10, maxneightbor = 10

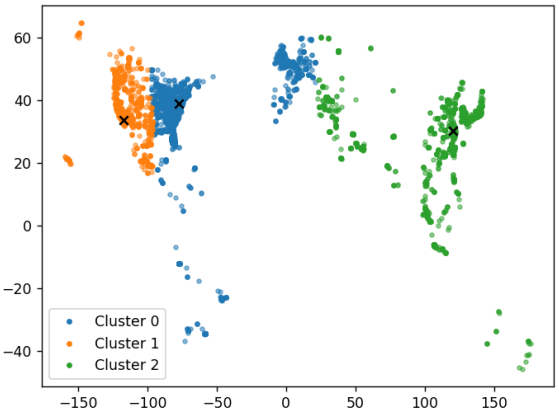
Otrzymane wyniki dla algorytmu CLARANS prezentuje tabela 7.

Wizualizacja wyników grupowania dla określonej liczby klastrów prezentuje rysunek 13, 14 i 15.

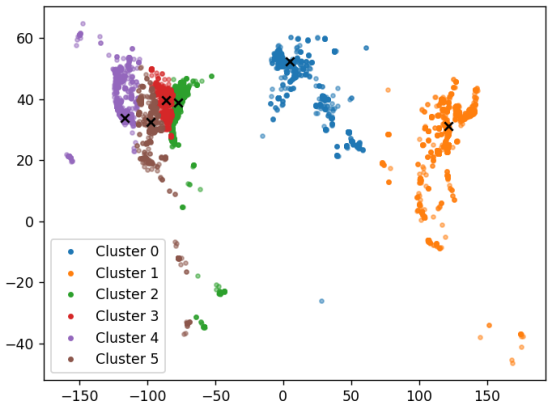
Tabela 7 Wyniki grupowania dla algorytmu CLARANS w zależności od rodzaju wyliczanej odległości między punktami.

Liczba klastrów	Miara Silhouette	Czas grupowania [s]	Koszt grupowania	Uzyskane medoidy
3	0.6929	2.2	194128.3	[-77.17, 39.11] [-116.97, 33.8]

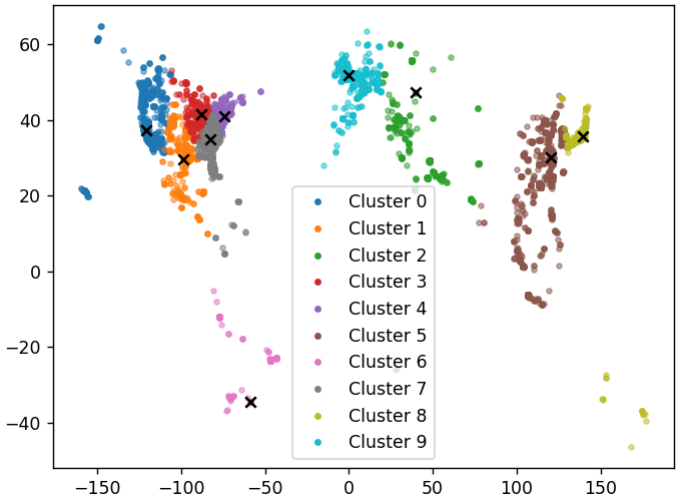
				[120.18, 30.33]
6	0.6486	2.7	110391.83	[4.9, 52.38] [121.45, 31.23] [-77.35, 38.98] [-86.16, 39.62] [-116.53, 33.85] [-97.73, 32.46]
10	0.7231	3.3	78184.55	[-120.48, 37.32] [-98.57, 29.49] [39.72, 47.26] [-87.85, 41.62] [-73.96, 41.1] [120.16, 30.24] [-58.41, -34.6] [-82.27, 34.95] [139.7, 35.66] [-0.38, 51.88]



Rys. 13 Grupowanie algorytmem CLARANS dla k=3



Rys. 14 Grupowanie algorytmem CLARANS dla k-6



Rys. 15 Grupowanie algorytmem CLARANS dla k=10

Zestawienie wyników dla wszystkich algorytmów zamieszczono w tabeli 8.

Tabela 8 Wyniki grupowania dla poszczególnych algorytmów w zależności od liczby klastrów.

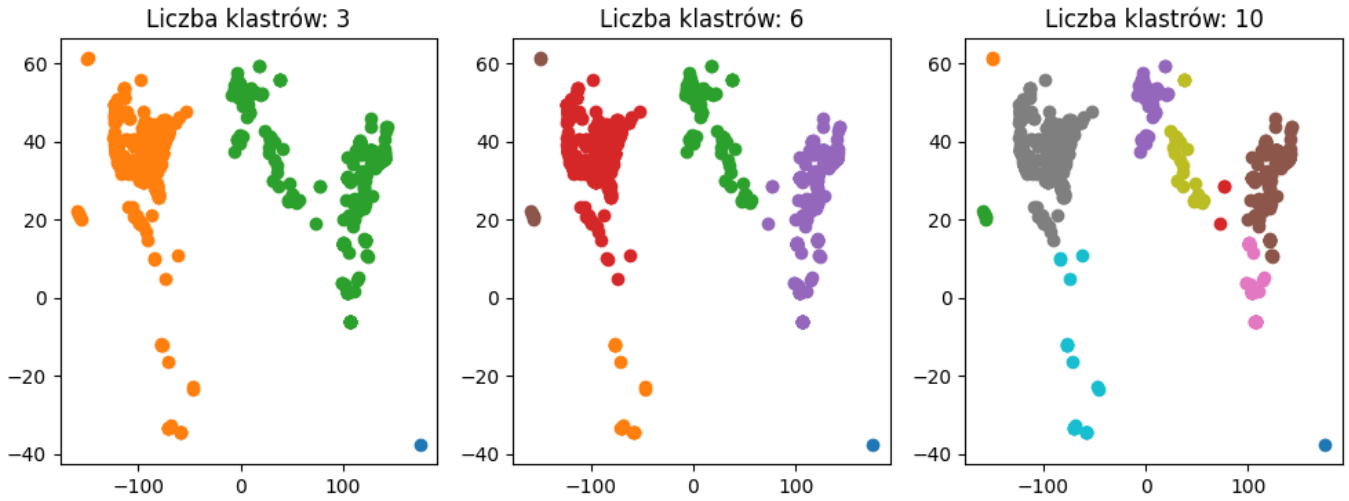
Liczba klastrów	**Miara Silhouette**	**Czas grupowania [s]**	**Koszt grupowania**	**Algorytm**
3	0.4744	618.42	190623.43	PAM
	0.6993	0.03	233962.57	CLARA
	0.6929	2.2	194128.3	CLARANS
6	0.5092	0.06	176297.42	PAM
	0.5092	0.06	176297.42	CLARA
	0.6486	2.7	110391.83	CLARANS
10	0.7646	2384.36	70060.38	PAM
	0.3937	0.14	129855.28	CLARA
	0.7231	3.3	78184.55	CLARANS

Algorytm CURE

Na potrzeby tego eksperymentu zostało wykorzystane 1000 punktów, liczbę reprezentantów c równą 4 i wartość alpha równą 0,4. Poniższa tabela przedstawia czas działania algorytmu CURE i miary silhouette w zależności od liczby klastrów. Czas działania powinien spadać w miarę zwiększania liczby klastrów, w przypadku algorytmu hierarchicznego, w którym algorytm zatrzymuje się wcześniej, jeżeli wymagana jest większa liczba klastrów. W tym przypadku czas nie zmienia się ze względu na zbyt małe różnice pomiędzy liczbą wybieranych klastrów.

Liczba klastrów	Czas działania [s]	Miara silhouette
3	19.192270	0.684903
6	18.788851	0.639432
10	19.220457	0.554329

Na poniższym rysunku zostały przedstawione znalezione klastry:



WNIOSKI

Analizując otrzymane wyniki ze względu na liczbę klastrów widać, że dla algorytmu PAM i CLARANS wraz ze wzrostem liczby klastrów rosła w większości przypadków miara Silhouette, a medoidy zaznaczone na powyższych wizualizacjach wydają się być rozmieszczone w sposób poprawny dla każdego z klastrów. Przeciwna zależność dostrzegana jest w przypadku algorytmu CLARA, dla którego jakość grupowania maleje wraz ze wzrostem parametru k, a patrząc na wykresy widać, że niektóre z medoidów nie są zaznaczone poprawnie wewnątrz danego klastra.

Wraz ze wzrostem liczby szukanych klastrów wzrasta czas potrzebny na ich zlokalizowanie. W przypadku algorytmu PAM i k=10 jest to aż 2384.36 sekund, czyli około 40 min.

Dla algorytmu CURE wraz ze wzrostem liczby klastrów czas działania pozostawał stabilny, a miara silhouette malała. Można zauważyć, że algorytm CURE nie jest podatny na obserwacje odstające i dobrze dzieli punkty na klastry.

4.3 Wielkość zbioru

W tej części projektowej zbadano wpływ wielkości zbioru danych na proces grupowania i otrzymane wyniki dla algorytmów PAM, CLARA, CLARANS i CURE. Eksperymenty przeprowadzono dla podziału na 3 klastry i odległość euklidesową.

Algorytm PAM

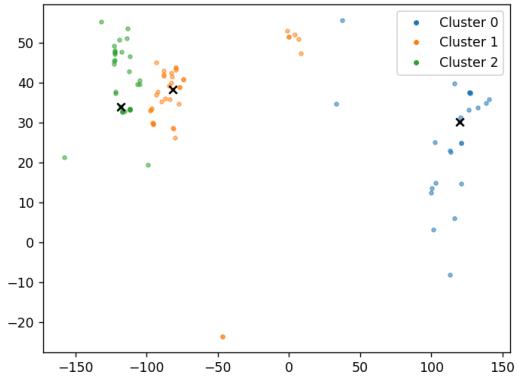
Otrzymane wyniki dla algorytmu PAM prezentuje tabela 9.

Wizualizacja wyników grupowania dla określonej liczby klastrów prezentuje rysunek 16,17 i 18.

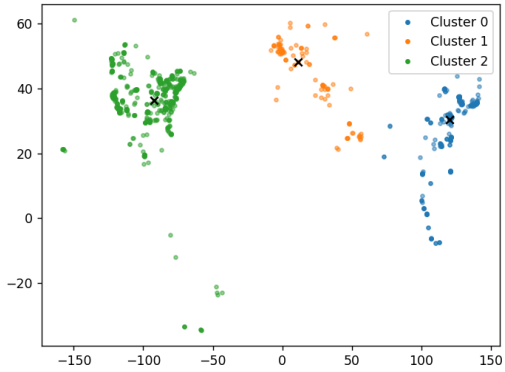
Tabela9 Wyniki grupowania dla algorytmu PAM w zależności od rodzaju wielkości zbioru.

Wielkość zbioru	Miara Silhouette	Czas grupowania [s]	Koszt grupowania	Uzyskane medoidy
100	0.7049	0.03	1863.58	[120.19, 30.23] [-81.64, 38.35] [-118.4, 33.96]
1000	0.7659	5.94	18101.01	[120.14, 30.32]

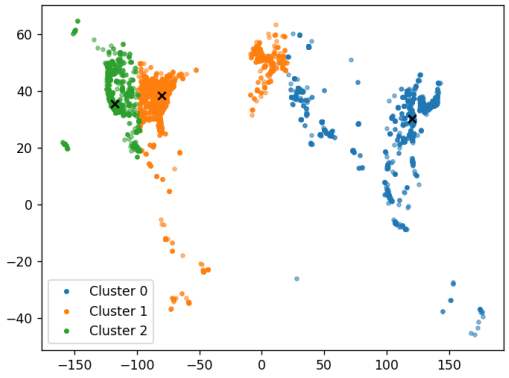
				[11.59, 48.16] [-92.37, 36.35]
10000	0.4744	618.42	190623.43	[120.12, 30.34] [-80.0, 38.4] [-117.67, 35.63]



Rys. 16 Grupowanie algorytmem PAM dla zbioru 100 elementów



Rys. 17 Grupowanie algorytmem PAM dla zbioru 1000 elementów



Rys. 18 Grupowanie algorytmem PAM dla zbioru 10000 elementów

Algorytm CLARA

Nastawy parametrów: repeats = 2, num_representative_sample = 40*2*k.

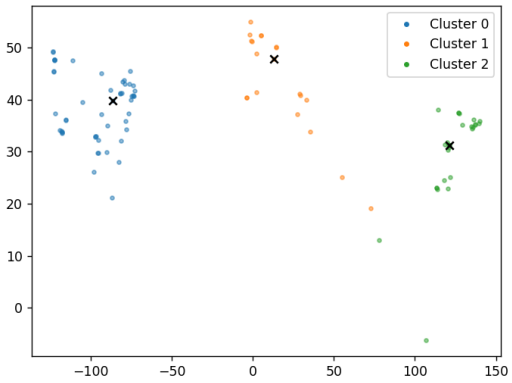
Otrzymane wyniki dla algorytmu CLARA prezentuje tabela 10.

Wizualizacja wyników grupowania dla określonej liczby klastrów prezentuje rysunek 19,20 i 21.

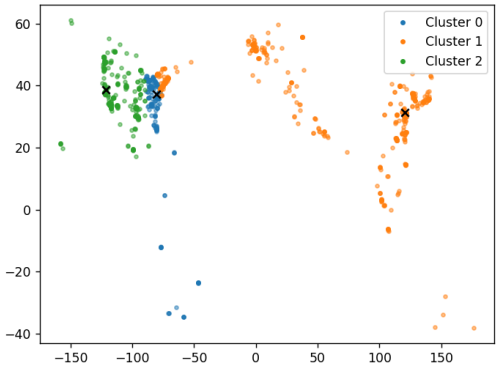
Tabela 10. Wyniki grupowania dla algorytmu PAM w zależności od rodzaju wielkości zbioru.

Wielkość zbioru	Miara Silhouette	Czas grupowania [s]	Koszt grupowania	Uzyskane medoidy
100	0.7957	0.03	2373.87	[121.44, 31.19]

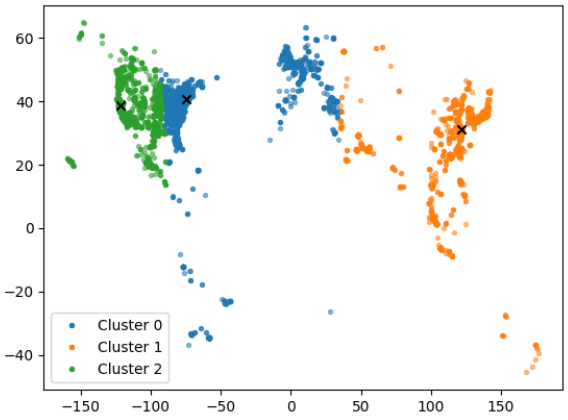
				[13.04, 47.8] [-86.33, 39.76]
1000	0.3013	0.03	71968.42	[120.62, 31.37] [-121.33, 38.71] [-80.42, 37.16]
10000	0.6993	0.03	233962.57	[121.41, 31.28] [-121.22, 38.79] [-74.61, 40.58]



Rys. 19 Grupowanie algorytmem CLARA dla zbioru 100 elementów



Rys. 20 Grupowanie algorytmem CLARA dla zbioru 1000 elementów



Rys. 21 Grupowanie algorytmem CLARA dla zbioru 10000 elementów

Algorytm CLARANS

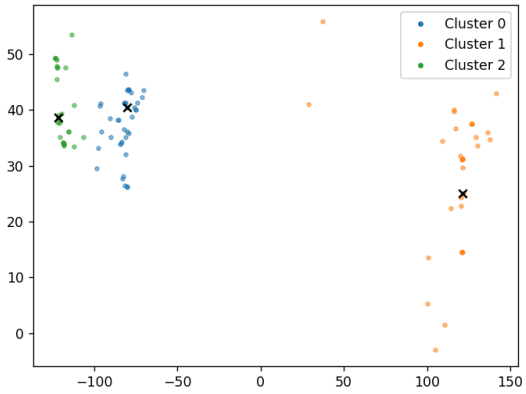
Nastawy parametrów: numlocal = 10, maxneightbor = 10.

Otrzymane wyniki dla algorytmu CLARANS prezentuje tabela 11.

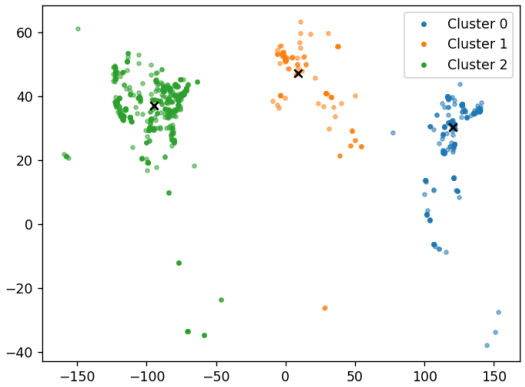
Wizualizacja wyników grupowania dla określonej liczby klastrów prezentuje rysunek 22,23 i 24.

Tabela 11. Wyniki grupowania dla algorytmu PAM w zależności od rodzaju wielkości zbioru.

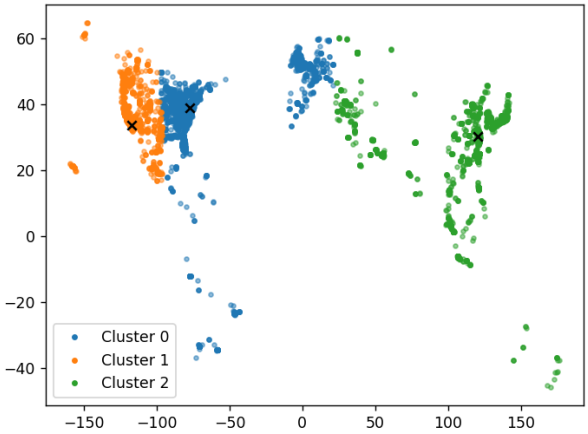
Wielkość zbioru	Miara Silhouette	Czas grupowania [s]	Koszt grupowania	Uzyskane medoidy
100	0.8290	0.03	1080.33	[-80.0, 40.44] [121.41, 25.02] [-121.35, 38.66]
1000	0.7685	0.23	17952.84	[120.18, 30.33] [8.82, 47.23] [-94.53, 37.07]
10000	0.6929	2.2	194128.3	[-77.17, 39.11] [-116.97, 33.8] [120.18, 30.33]



Rys. 22 Grupowanie algorytmem CLARANS dla zbioru 100 elementów



Rys. 23 Grupowanie algorytmem CLARANS dla zbioru 1000 elementów



Rys. 24 Grupowanie algorytmem CLARANS dla zbioru 10000 elementów

Zestawienie wyników dla wszystkich algorytmów zamieszczono w tabeli 12.

Tabela 12 Wyniki grupowania dla poszczególnych algorytmów w zależności od wielkości zbioru danych.

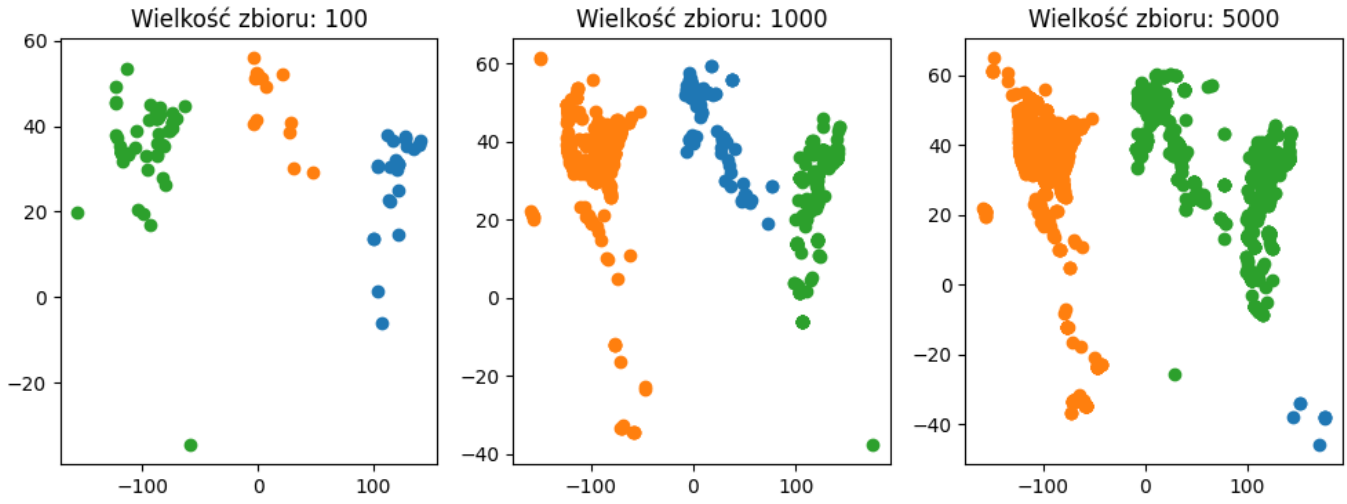
Wielkość zbioru	Miara Silhouette	Czas grupowania [s]	Koszt grupowania	Algorytm
100	0.7049	0.03	1863.58	PAM
	0.7957	0.03	2373.87	CLARA
	0.8290	0.03	1080.33	CLARANS
1000	0.7659	5.94	18101.01	PAM
	0.3013	0.03	71968.42	CLARA
	0.7685	0.23	17952.84	CLARANS
10000	0.4744	618.42	190623.43	PAM
	0.6993	0.03	233962.57	CLARA
	0.6929	2.2	194128.3	CLARANS

Algorytm CURE

Na potrzeby tego eksperymentu została wykorzystana liczba klastrów równa 3, liczba reprezentantów c równa 4 i wartość alpha równa 0,4. Poniższa tabela pokazuje wzrost czasu działania algorytmu CURE w miarę zwiększania liczby wykorzystywanych danych.

Wielkość zbioru	Czas działania [s]	Miara silhouette
100	0.263034	0.777437
1000	20.132657	0.769053
5000	526.687790	0.675028

Na poniższym rysunku zostały pokazane znalezione klastry dla podanych wielkości zbiorów:



WNIOSKI

Analizując otrzymane wyniki ze względu na wielkość zbioru danych widać, że wraz ze wzrostem liczby elementów zbioru jakość grupowania maleje, gdyż mamy więcej wartości na granicy klastrów. Najlepiej statystycznie radził sobie algorytm CLARANS – optymalizował czas, z jakością grupowania.

Widać również, że czas grupowania algorytmu PAM znacznie odbiegał w porównaniu z dwoma pozostałymi algorytmami. Na podstawie zamieszczonych wizualizacji można wywnioskować, że medoidy dla poszczególnych algorytmów i wielkości zbiorów danych wyznaczane są poprawnie.

Dla algorytmu CURE w miarę zwiększania liczby punktów do przetworzenia w naturalny sposób zwiększał się czas potrzebny na ich przetworzenie. Miara silhouette zmniejszyła się.

4.4 Algorytm CLARANS

W tej części projektowej zbadano wpływ poszczególnych parametrów na proces grupowania i otrzymane wyniki dla algorytmów CLARANS. Eksperymenty przeprowadzono dla zbioru o wielkości 1000 wartosci, podziału na 3 klastry i odległość euklidesową.

Parametr max_neightbour

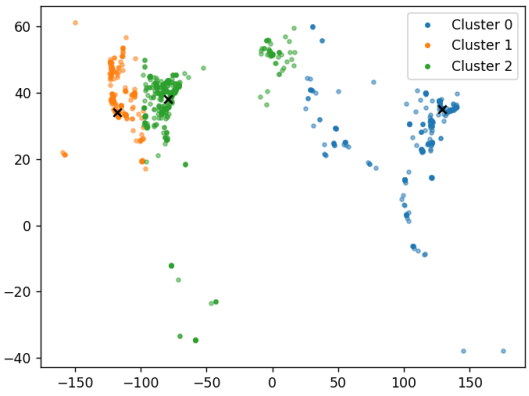
Nastawy parametrów: numLocal = 10.

Otrzymane wyniki dla algorytmu CLARANS prezentuje tabela 13.

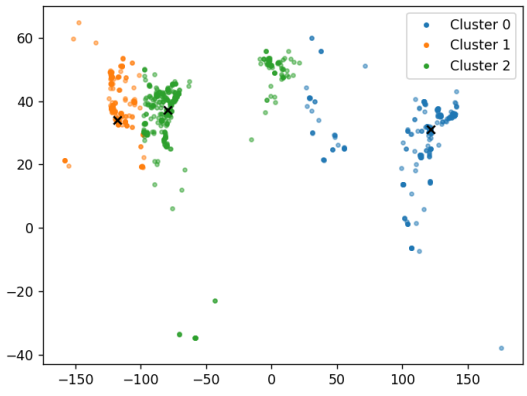
Wizualizacja wyników grupowania dla określonej liczby klastrów prezentuje rysunek 25, 26 i 27.

Tabela 13. Wyniki grupowania dla algorytmu CLARANC w zależności od parametru max_neightbour.

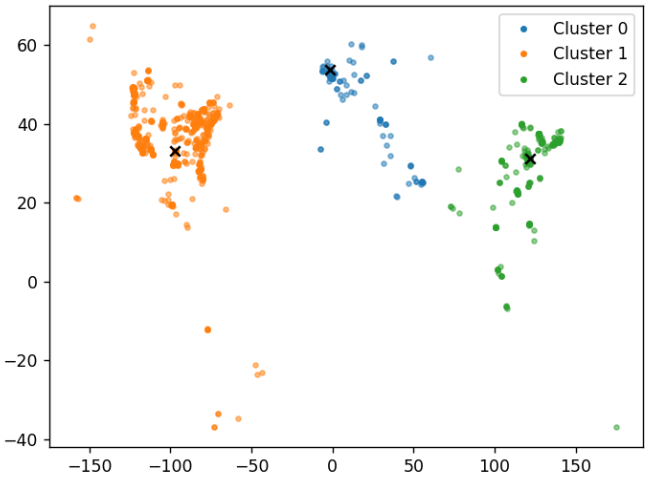
Parametr max_neightbour	Miara Silhouette	Czas grupowania [s]	Koszt grupowania	Uzyskane medoidy
3	0.6977	0.06	20096.86	[129.14, 35.17] [-118.13, 34.09] [-79.04, 38.13]
10	0.7017	0.23	19311.06	[121.42, 31.22] [-118.05, 34.13] [-79.24, 37.33]
20	0.7715	0.39	18117.85	[-1.59, 53.73] [-97.14, 33.23] [121.52, 31.29]



Rys. 25 Grupowanie algorytmem CLARANS dla max_neightbour = 3



Rys. 26 Grupowanie algorytmem CLARANS dla max_neightbour = 10



Rys. 27 Grupowanie algorytmem CLARANS dla max_neightbour = 20

Parametr numLocal

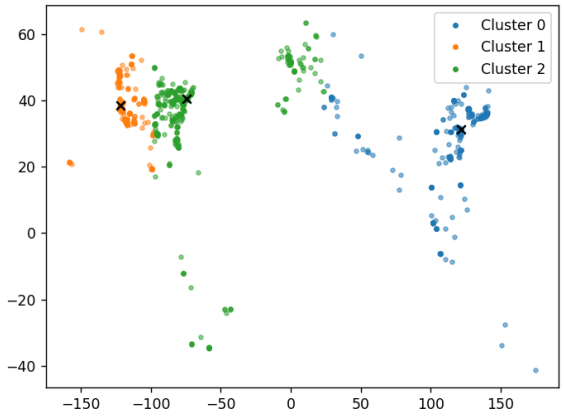
Nastawy parametrów: max_neightbour = 10.

Otrzymane wyniki dla algorytmu CLARANS prezentuje tabela 14.

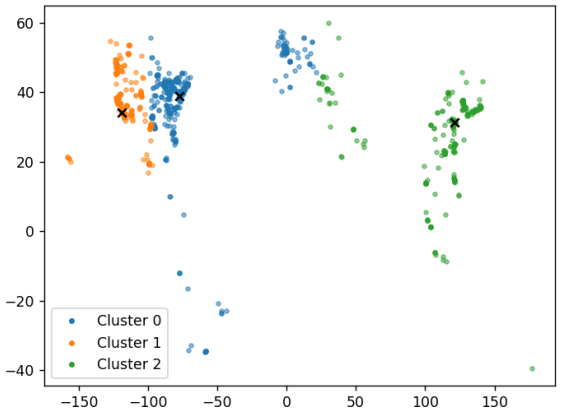
Wizualizacja wyników grupowania dla określonej liczby klastrów prezentuje rysunek 28,29 i 30.

Tabela 14. Wyniki grupowania dla algorytmu PAM w zależności od parametru numLocal

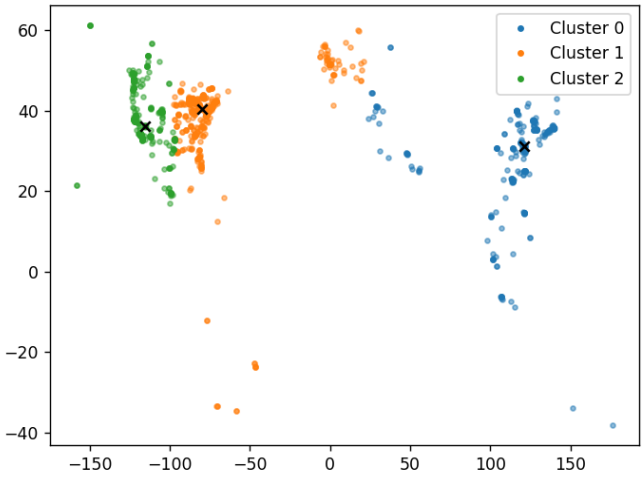
Parametr numLocal	Miara Silhouette	Czas grupowania [s]	Koszt grupowania	Uzyskane medoidy
3	0.6937	0.09	20921.56	[121.59, 31.28] [-121.43, 38.47] [-74.44, 40.55]
10	0.7030	0.23	18842.36	[-77.02, 38.9] [-118.91, 34.19] [121.45, 31.17]
20	0.7150	0.33	17717.99	[121.36, 31.22] [-79.96, 40.44] [-115.15, 36.08]



Rys. 25 Grupowanie algorytmem CLARANS dla numLocal = 3



Rys. 26 Grupowanie algorytmem CLARANS dla numLocal = 10



Rys. 27 Grupowanie algorytmem CLARANS dla numLocal = 20

Zestawienie wyników dla wszystkich algorytmów zamieszczono w tabeli 15.

Tabela 15 Wyniki grupowania dla algorytmu CALRANS w zależności od wartości określonych parametrów.

Parametr	Wartość parametru	Miara Silhouette	Czas grupowania [s]	Koszt grupowania
numLocal	3	0.6937	0.09	20921.56
	10	0.7030	0.23	18842.36
	20	0.7150	0.33	17717.99
max_neightbour	3	0.6977	0.06	20096.86
	10	0.7017	0.23	19311.06
	20	0.7715	0.39	18117.85

WNIOSKI

Analizując otrzymane wyniki widać, że wzrost parametry numLocal (a tym samym liczby poszukiwanych minimów) wpływa na czas i jakość grupowania. Wraz ze wzrostem wartości tego parametru rośnie dokładność (miara Silhouette) oraz czas grupowania, zaś koszt grupowania maleje. Podobne zależności widoczne są w przypadku parametru max_ neightbour. Dla tego parametru można zauważyć, że przy wartości równej 20 podział danych na klastry jest inny niż niż dla dwóch pozostałych badanych wartości.

Taki rodzaj zależności wraz ze wzrostem poszczególnych parametrów wynika z następujących czynników:

- Większa przestrzeń eksploracji rozwiązań: Zwiększanie liczby lokalnych poszukiwań (numLocal) i maksymalnej liczby sąsiadów (max_neighbour) powoduje szerszą eksplorację przestrzeni rozwiązań, a tym samym poprawę jakości grupowania.
  Algorytm wykonuje większą liczba iteracji, co pozwala na bardziej dogłębne badanie przestrzeni rozwiązań, co prowadzi do lepszego podziału danych na klastry, a co za tym idzie wyższej jakości grupowania (miara Silhouette) .
- Większa złożoność obliczeniowa: Zwiększenie wartości parametrów numLocal i max_neighbour oznacza przeprowadzenie większej liczby obliczeń. To z kolei prowadzi do dłuższego czasu grupowania, ponieważ algorytm musi przeprowadzić więcej iteracji i analizować większą liczbę sąsiadów i możliwych rozwiazań.
- Redukcja kosztu grupowania: Zwiększenie liczby lokalnych poszukiwań i maksymalnej liczby sąsiadów może prowadzić do znalezienia lepszych medoidów (punktów reprezentujących klastry) i optymalniejszego podziału klastrów. To z kolei prowadzi do redukcji kosztu grupowania, który mierzy odległość między punktami,a medoidami. Niższy koszt grupowania wskazuje na bardziej skoncentrowane i lepiej oddzielone klastry.

Warto jednak zauważyć, że wzrost wartości parametrów numLocal i max_neighbour nie jest bezkresny, aż do osiągnięcia najlepszych wyników. Istnieje pewien punkt, po którym dalsze zwiększanie tych parametrów może nieznacznie poprawić wyniki lub nawet spowodować pogorszenie, przy jednoczesnym wzroście czasu grupowania i kosztu obliczeń. Optymalne wartości tych parametrów zależą od charakterystyki danych, problemu grupowania oraz dostępnych zasobów obliczeniowych.

4.4 Algorytm CURE

W tej części został zbadany wpływ poszczególnych parametrów na proces grupowania i otrzymane wyniki dla algorytmów CURE. Eksperymenty przeprowadzono dla zbioru o wielkości 1000 wartości.

Parametr c – liczba reprezentantów

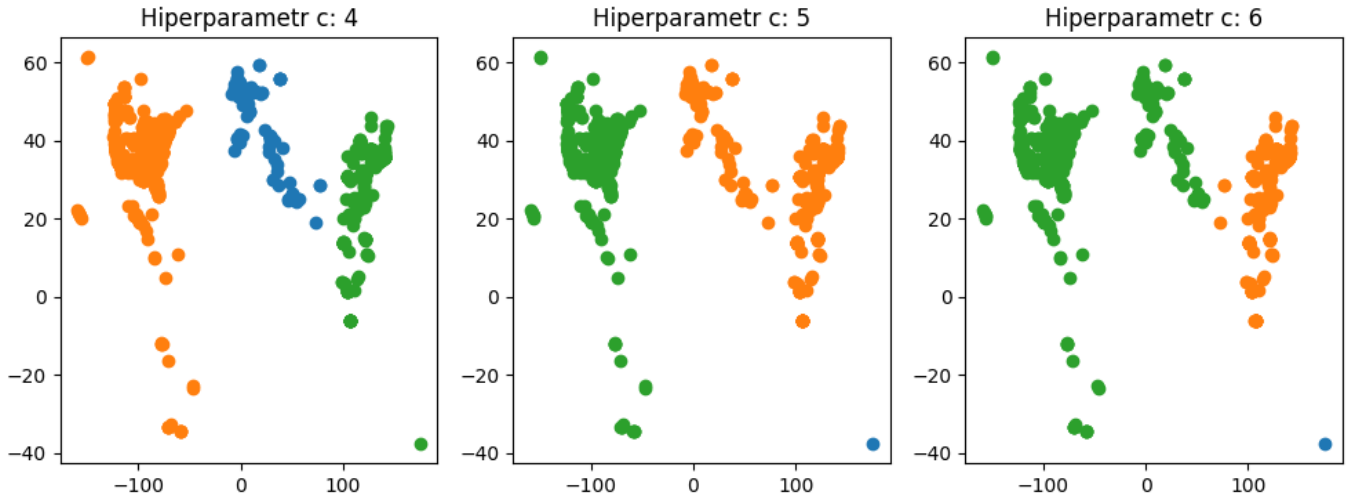
Parametr c określa liczbę reprezentantów klastra używanych przy obliczaniu odległości pomiędzy klastrami. W tym eksperymencie została wykorzystana wartość parametru alpha równa 0,4 i liczba klastrów równa 3.

Poniższa tabela pokazuje, że w ramach zwiększania liczby punktów reprezentacyjnych c zwiększa się czas działania algorytmu. Prawdopodobnie jest to związane z koniecznością obliczenia większej liczby punktów reprezentatywnych. Miara silhouette przy zwiększeniu parametru c z 4 na 5 zmniejszyła się, a przy zwiększeniu z 5 do 6 zwiększyła się.

Hiperparametr c	Czas działania [s]	Miara silhouette

4	20.293543	0.769053
5	24.763016	0.684903
6	29.672089	0.733626

Na poniższych rysunkach zostały przedstawione znalezione klastry:



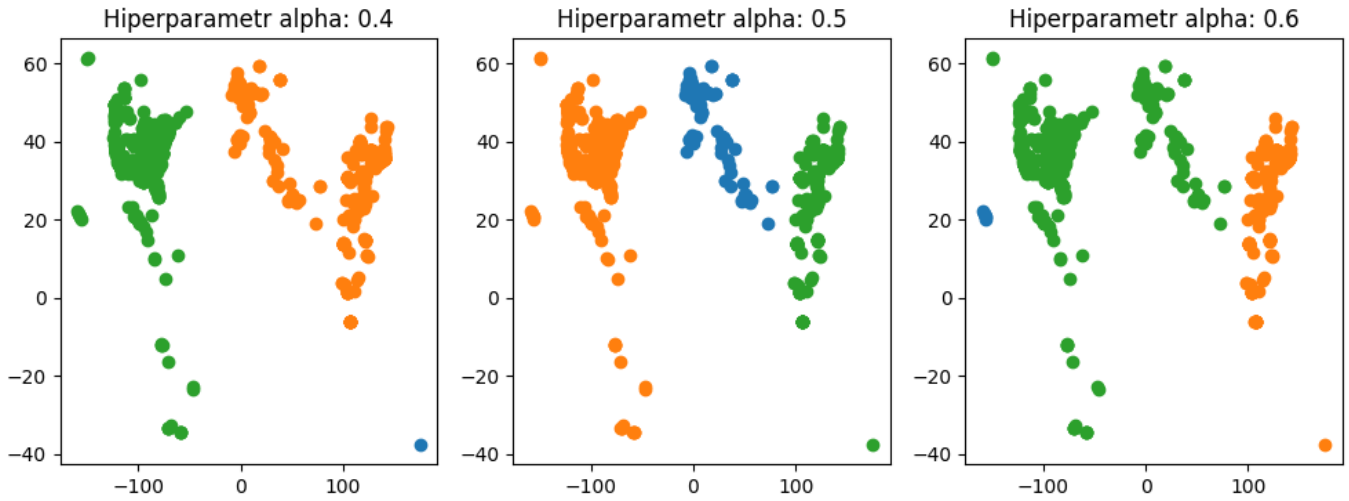
Parametr alpha – współczynnik przeniesienia reprezentantów w stronę centroidu

W algorytmie CURE punkty reprezentatywne nie są wykorzystywane w sposób bezpośredni. Zamiast tego są obliczane współrzędne punktów przesuniętych o współczynnik alpha pomnożony przez odległość do centroidu w stronę tego centroidu. Przygotowane w taki sposób punkty są wykorzystywane jako reprezentanci.

Parametr alpha nie wpływa na czas działania algorytmu, ponieważ nie zmienia liczby wykonywanych operacji. Jednak parametr ten wpływa na jakość wyników. W przypadku danych przedstawionych poniżej danych zwiększanie parametru alpha pogorszyło wyniki biorąc pod uwagę miarę silhouette.

Hiperparametr alpha	Czas działania [s]	Miara silhouette
0.4	21.509777	0.684903
0.5	20.096045	0.769053
0.6	21.744629	0.475786

Na poniższych rysunkach zostały przedstawione znalezione klastry. W przypadkach, w których alpha jest równe 0,4 i 0,6 został znaleziony klaster ze znacznie mniejszą liczbą punktów niż w przypadku alpha równego 0,5. To pokazuje, że najlepszą wartością alpha dla wykorzystywanych danych jest 0,5.



5 Wnioski końcowe

Otrzymane wyniki jakości i szybkości grupowania dla poszczególnych algorytmów z grupy – PAM, CLARA, CLARANS potwierdzają założenia dotyczące działania tych algorytmów zawarte w paragrafie „Wstęp ogólny”. Mianowicie pod względem szybkości działania najlepiej wypadł algorytm CLARA, który działa na próbce danych, najwolniej zaś działa algorytm PAM, który przechodzi po wszystkich punktach w przestrzeni w celu znalezienia optymalnego sposobu grupowania. Czas działania tego algorytmu odbiegał znacznie od czasu działania pozostałych algorytmów. Algorytm CLARANS przeprowadza losowe poszukiwanie w przestrzeni medoidów, co pozwala na odkrywanie różnych rozwiązań i znalezienie lepszych konfiguracji klastrów (wyższa miara Silhouette), jednak jest on wolniejszy od algorytmu CLARA.

Algorytm CLARANS z odpowiednio dobranymi parametrami łączy szybkość działania algorytmu CLARA i jakość grupowania PAM, nie powielając ich wad. Z tego powodu doskonale nadaje się do grupowania danych geograficznych.

Wnioski CURE vs CLARANS

CURE i CLARANS to dwa różne algorytmy używane w klasteryzacji (grupowaniu danych). Na podstawie eksperymentów można powiedzieć, że algorytm CLARANS jest znacznie szybszy niż algorytm CURE. Biorąc pod uwagę jakość grupowania na podstawie miary silhouette dla większości przypadków wyniki dla algorytmów CURE i CLARANS są podobne. Znaczną różnicę można zauważyć przy testach liczby klastrów. Dla liczby klastrów równej 10 wyniki dla algorytmu CURE są znacznie gorsze niż dla algorytmu CLARANS. Dla CURE miara silhouette wynosi około 0,55 a dla CLARANS około 0,72.

Wybrane podobieństwa i różnice między algorytmami CURE i CLARANS:

Podobieństwa:

- Obie metody wykorzystują techniki grupowania punktów danych na podstawie odległości między nimi.
- Zarówno CURE, jak i CLARANS wykorzystują punkty reprezentatywne.

Różnice:

- CURE to algorytm klasteryzacji hierarchicznej, podczas gdy CLARANS to algorytm klasteryzacji stochastycznej.
- CURE jako reprezentantów klastra wykorzystuje zmodyfikowane losowe punkty przeniesione w stronę centroidu, a CLARANS bierze pod uwagę samych reprezentantów.

6 Instrukcja użycia i pliki projektu

- Plik jupyter notebook CURE.ipynb zawiera implementację algorytmu CURE wraz z wynikami przeprowadzonych testów. W ramach notebooka można uruchomić kolejno wszystkie komórki.
- Plik CLARANS.py zawiera implementacje algorytmów PAM, CLARA i CLARANS
- Plik csv directory.csv zawiera dane wykorzystywane w projekcie w formie tabelarycznej.
- Plik requirements.txt - plik z bibliotekami python potrzebnymi do uruchomienia programów.

7 Bibliografia

1. Zbiór danych Starbucks - https://www.kaggle.com/datasets/starbucks/store-locations?select=directory.csv
2. CLARA- https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/CLARA
3. PAM - https://towardsdatascience.com/k-medoid-clustering-pam-algorithm-in-python-with-solved-example-c0dcb35b3f46
4. CLARANS – Raymond T.Ng, J.Han: CLARANS: A method for clustering objects for spatial data mining 2002 IEEE Transactions on Knowledge and Data Engineering
5. CURE - Sudipto Guha, Rajeev Rastogi, Kyuscok Shim, CURE: An Efficient Clustering Algorithm for Large Databases, 1998 (citeseer.ist.psu.edu)
6. CURE algorithm Wikipedia - https://en.wikipedia.org/wiki/CURE_algorithm