Grupowanie danych

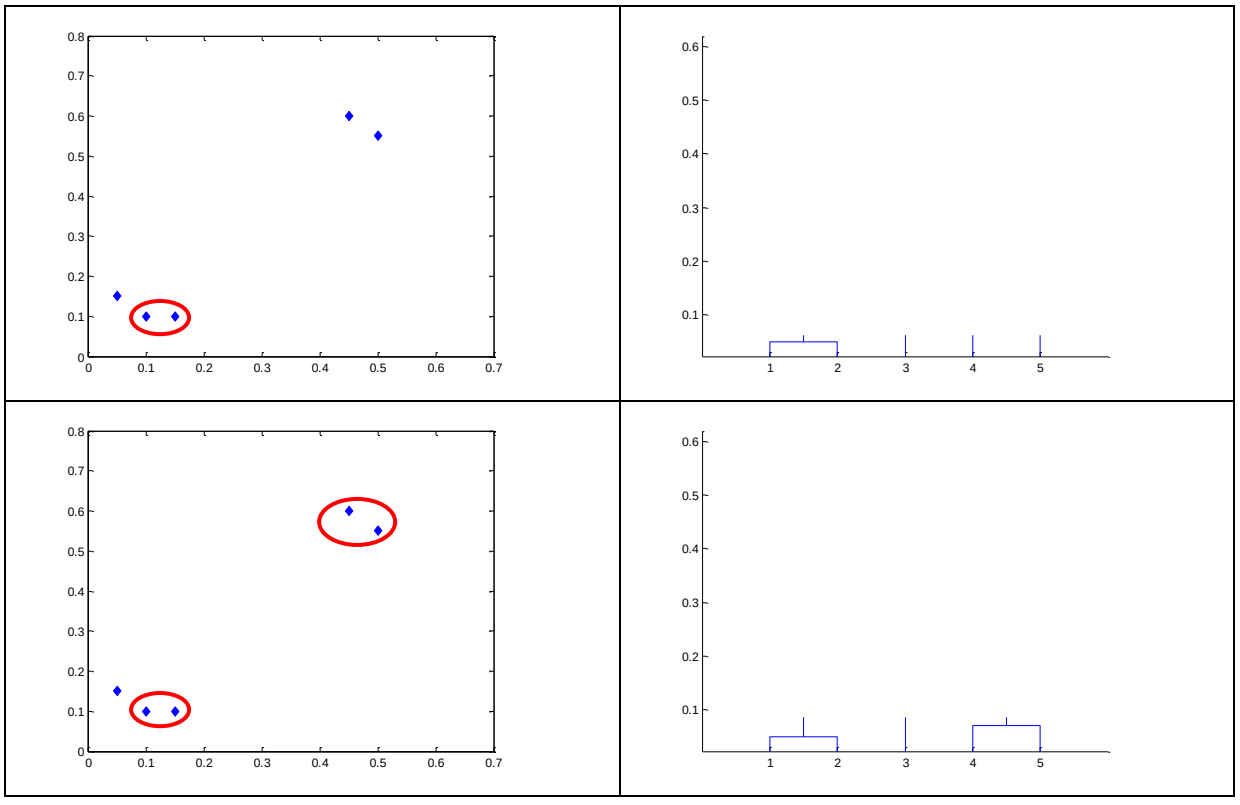
Algorytm grupowania hierarchicznego

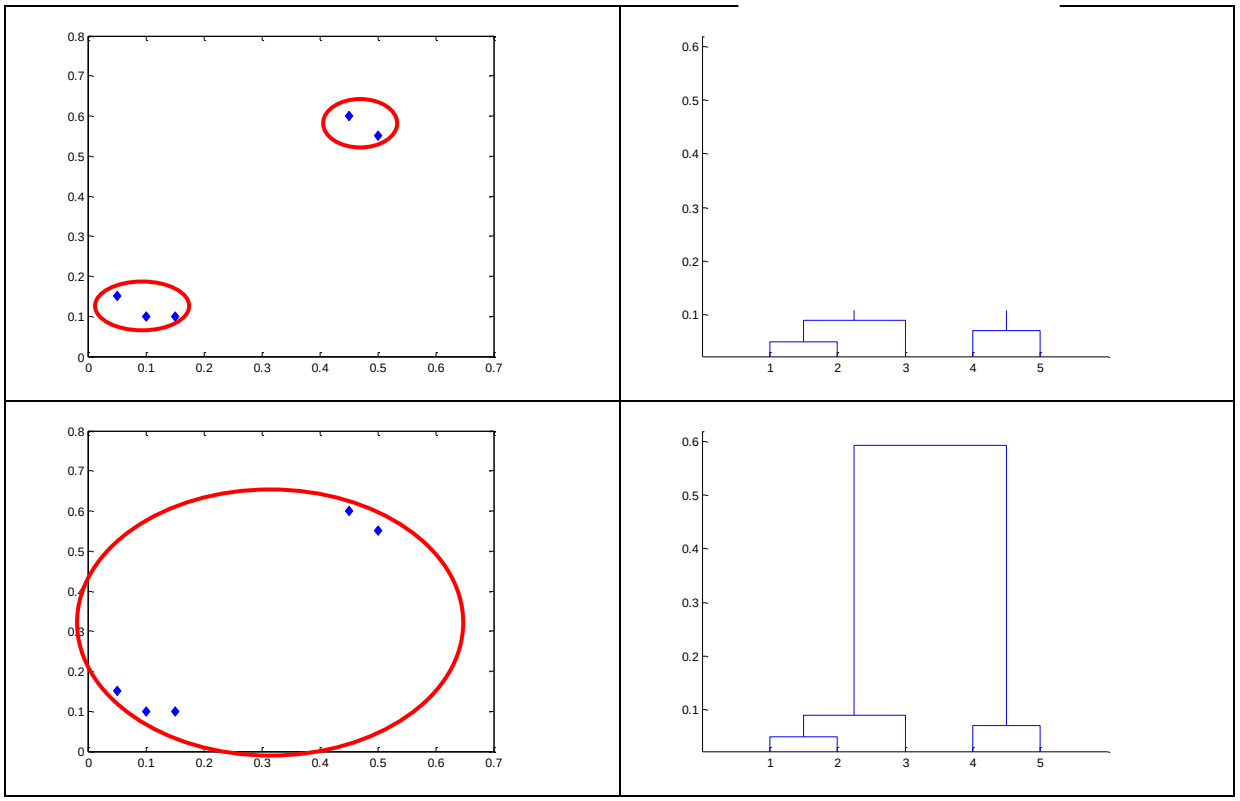
Opis

Grupowanie hierarchiczne jest odmienną grupą metod grupowania danych w stosunku do metod bazujących na minimalizacji skalarnego współczynnika jakości jak np. algorytm k-means. Metoda ta bazuje na budowie grafu w postaci drzewa. Algorytm ten jest typem algorytmów z dołu do góry „buttom – top" gdzie zakłada się, iż każdy wektor stanowi oddzielny klaster, a następnie łączy się małe klastry w coraz to większe. Proces łączenia realizowany jest na zasadzie poszukiwania klasterów leżących najbliżej siebie i zastępowania ich nowym większym klastrem, stanowiącym połączenie dwóch poprzednich. Proces ten stopniowo postępuje aż do chwili, w której zostanie osiągnięta właściwa liczba klastrów (określona przez użytkownika) lub do momentu gdy wszystkie wektory znajdują się w jednym klastrze.

Charakterystyczną cechą tego algorytmu jest momzliwość reprezentacji struktury klasteryzacji w postaci drzewa dendrogramu. Dendrogram na osi x posiada etykiety punktów, natomiast na osi y podobieństwo między grupami. Taka reprezentacja wyników klasteryzacji daje szerego możliwości jak np. ocenę liczby klastrów (jeśli wcześniej liczba ta jest nie znana), możliwość analizy pojawienia się wektorów odstających itp.

Przykład procesu grupowania hierarchicznego:





Proces klasteryzacji w tym algorytmie można podzielić na trzy etapy:

1. Liczenie odległości
2. Budowa drzewa na podstawie odległości
3. Przycięcie drzewa na określonym poziomie

Gdzie w drugim etapie jednym z parametrów jest określenie sposobu liczenia podobieństwa pomiędzy poszczególnymi grupami.

Typowymi parametrami są tutaj:

- Uśredniona wartość odległości pomiędzy wektorami w grupach

$$d(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} dist(x_{ri}, x_{sj})$$

- Wyszukiwanie dwóch najodleglejszych obiektów w klastrach

$$d(r,s) = max(dist(x_{ri}, x_{sj})), i \in (1, \ldots, n_r), j \in (1, \ldots, n_s)$$

- Wyszukiwanie dwóch najbardziej podobnych obiektów w klastrach

$$d(r,s) = min(dist(x_{ri}, x_{sj})), i \in (i, \ldots, n_r), j \in (1, \ldots, n_s)$$

- Wyznaczenie centroidy dla każdej grupy I liczenie odległości pomiędzy centroidami

$$d(r,s) = \left\| \bar{x}_r - \bar{x}_s \right\|_2 \text{ gdzie } \bar{x}_r = \frac{1}{n_r} \sum_{i=1}^{n} x_{ri}$$

- Wyznaczenie centroidy dla każdej grupy I liczenie odległości pomiędzy centroidami, jednak centoida wyznaczana jest jako średnia centroida z już istniejących centroid (tylko odległośc Euklidesa, duża szybkość)

  $d(r,s) = \|\tilde{x}_r - \tilde{x}_s\|_2$ gdzie $\tilde{x}_r$ i $\tilde{x}_s$ są centroidami klasterów *r* i *s* powstałymi z dwóch centroid *p* i *q* na podstawie których powstał dany klaster $\tilde{x}_r = \frac{1}{2}(\tilde{x}_p + \tilde{x}_q)$

Cechą algorytmu grupowania hierarchicznego jest możliwość wpływu na kształt powstałych grup poprzez wybór odpowiedniego typu łączenia grup. Cechy tej nie posiadają inne algorytmy grupowania, w szczególności bazujące na centroidach jak algorytm k-średnich czy VQ .

Zadania:

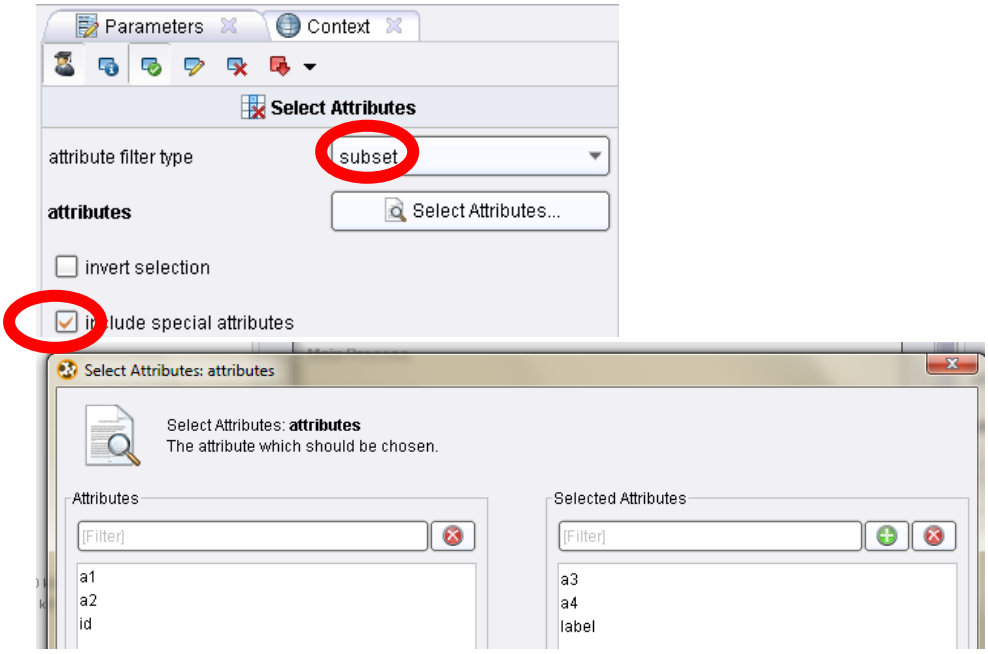
1. Zbuduj dendrogram dla danych opisanych za pomocą jednej zmiennej:

Id	1	2	3	4	5	6	7	8	9	10
Wartości	0	0	1	3	7	7	12	12	14	14

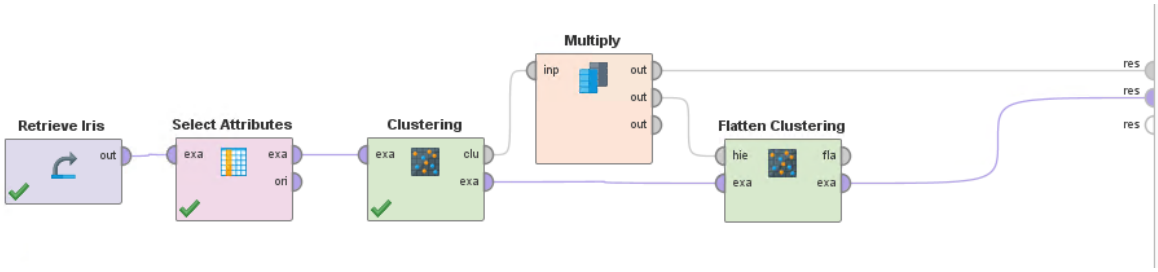
   Do budowy dendrogramu wykorzystaj łączenie typu najbliższego linku (single link)
2. Zbuduj dendrogram dla danych z poprzedniego zadania.
   Do budowy dendrogramu wykorzystaj łączenie typu kompletnego linku (complete link)
3. Czy zbudowane dendrogramy różnią się?

Przejdź do RapidMiner i …..

4. Wczytaj zbiór *Iris*, a następnie korzystając z operatora: ***Select Attributes*** dokonaj selekcji jedynie atrybutów z numerem 3 i 4 oraz label. Konfiguracja jak na rys.



   Następnie podłącz operator grupowania: **Agglomerative Clustering,** Jego wyjście **clu** podłącz na wyjście procesu. Na jego wyjście podłącz również operator **Flatten Clustering.** Operator ten odpowiada za przycięcie dendrogramu – określa ile klastrów chcemy uzyskać**.** Wynikowy zbiór danych podepnij na wyjścię procesu – tak jak pokazano na rys.

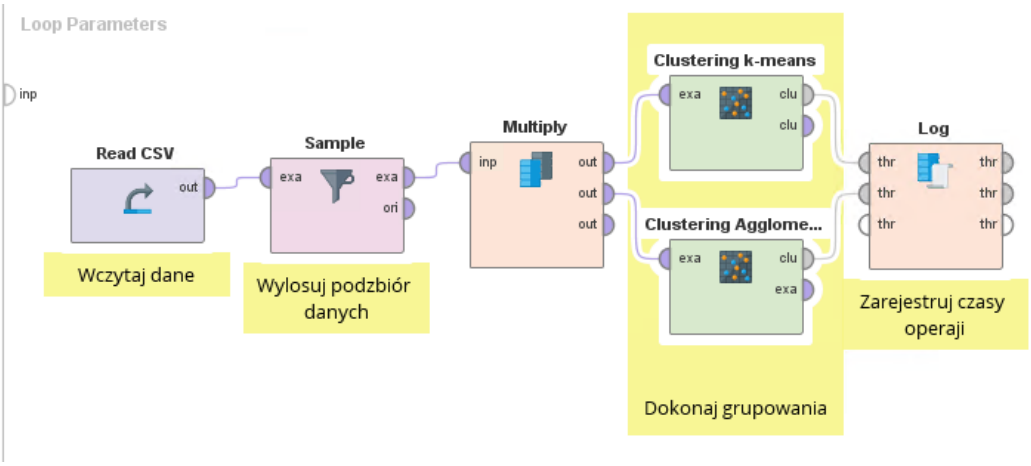


Ustaw liczbę klastrów na 3 (Operator Flatten CLustering) i zmieniaj wartości *mode* w operatorze **Agglomerative Clustering.** Parametr **mode** pozwala ustalić typ linkowania**.** Porównaj uzyskane wyniki klasteryzacji. Każdorazowo (dla każdej wartości **mode**)zwróć uwagę i zapamiętaj stworzony dendrogram.

Czym różnią się uzyskane dendrogramy? Sprawdź podział na dwa klastry dla różnych wartości parametru **mode.** Czy uzyskane wyniki przy podziale na dwa klastry różnią się między różnymi wartościami parametru **mode?** Każdorazowo obserwuj Scatter-plot z wynikami.

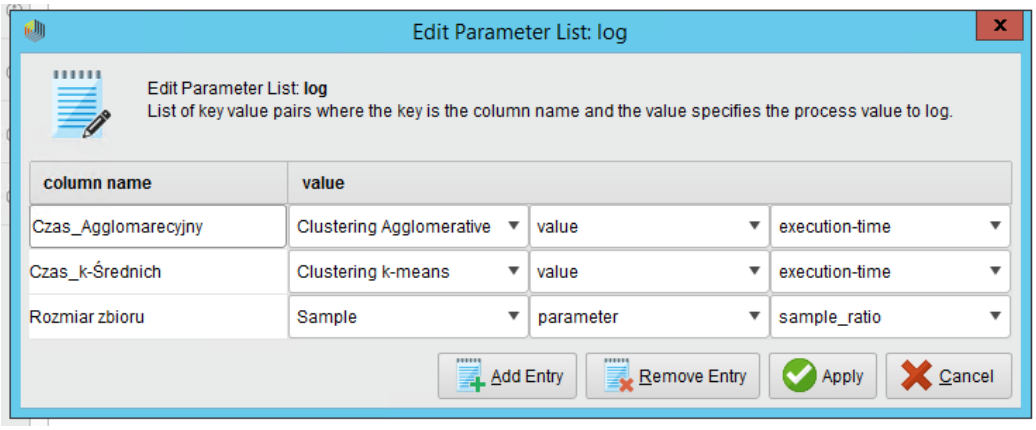
5. Przed operatorem grupowania danych dodaj operator Normalize. Czy i jak wpłynął on na uzyskane wyniki. Które wyniki uległy zmianie (dokonaj kontroli przy podziale na 3 klastry).
6. Ustal liczbę klastrów na 5 i powtórz kilkukrotnie obliczenia rejestrując wyniki graficzne. Czy uzyskane wyniki są powtarzalne (zawsze takie same) lub innymi słowy czy algorytm ten jest stabilny? Uwaga, pamiętaj aby przy kolejnych uruchomieniach zmieniać parametr Radnom_Seed głównego procesu.
7. Wczytaj zbiór *spiral* i powtórz obliczenia dla powyższych metod grupowania. Porównaj różne metody linkowania (parametr **mode**) Czy dla którejś z metod linkowania algorytm był w stanie prawidłowo podzielić dane na dwa klastry
8. Dokonaj porównania wydajności obydwu poznanych metod grupowania danych. W tym celu pobierz zbiór danych Pima Indian Diabetes. Idea badania polega na stopniowym wzroście rozmiaru danych poddanych procesowi grupowania i rejestracji czasu wykonania tej operacji przez obydwa algorytmy grupowania.
   W celu realizacji tego zadania: Stwórz nowy proces i dodaj do niego operator **Loop Parameters** (pozwala on na iterowanie po parametrach konfiguracyjnych operatorów znajdujących się w podprocesie tego operatora.

Wewnątrz **Loop Parameters** stwórz podproces jak na rys:

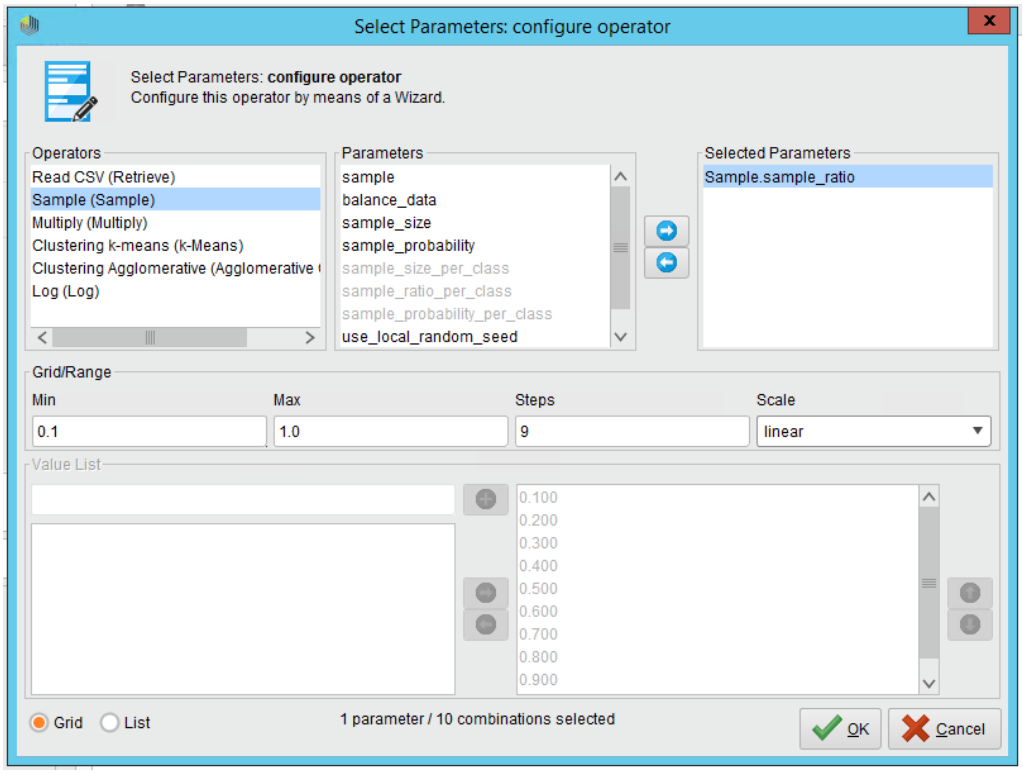


Operator **Sample** służy do wylosowania podzbioru danych podając procent danych który chcemy aby uzyskać na wyjściu. Uwaga ustaw w nim parametr **sample** na *relavtive*

Dokonaj konfiguracji operatora **Log** jak na rys:



Dokonaj konfiguracji operatora Loop Parameters tak aby stopniowo zwiększał się rozmiar zbioru od 10% aż po 100%. W tym celu dokonaj jego konfiguracji jak na rys:



Uruchom proces i zarejestruj wyniki. Uwaga w tym procesie nie musimy niczego podłączać na wyjście ponieważ każdy operator **Log** jest automatycznie dostarczany na wyjście procesu. Narysuj wykres przedstawiający czasy działania obydwu algorytmów. W tym celu wykorzystaj wykres typu Series, tak jak pokazano na rys: