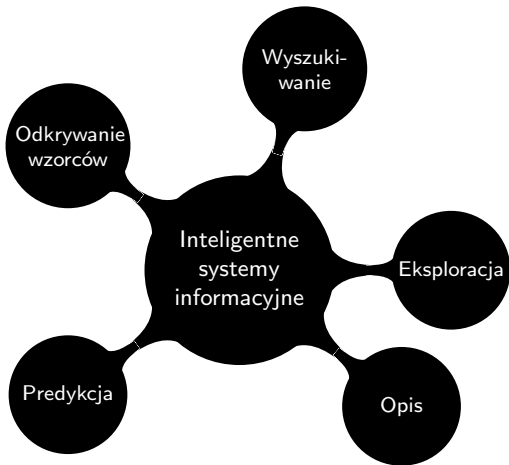


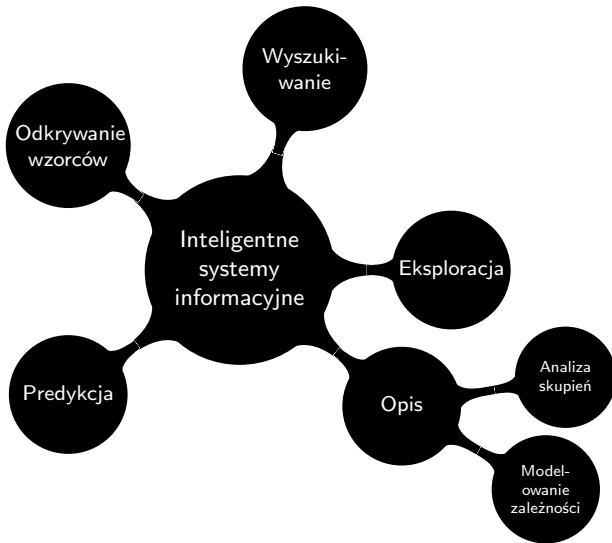
```
graph LR; A[Inteligentne systemy informacyjne] --- B[Filip Galiński]; A --- C[Grupowanie]
```

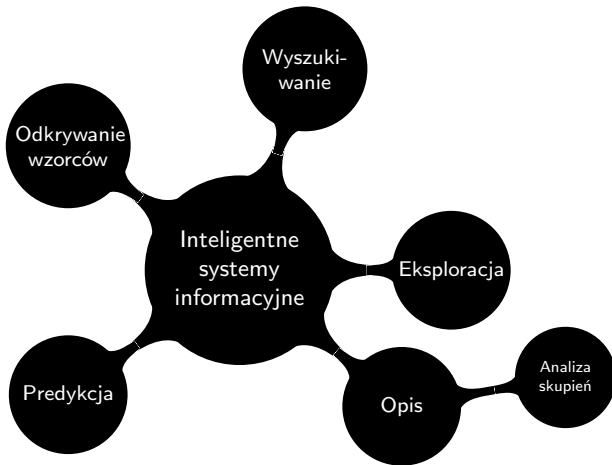
Inteligentne systemy informacyjne

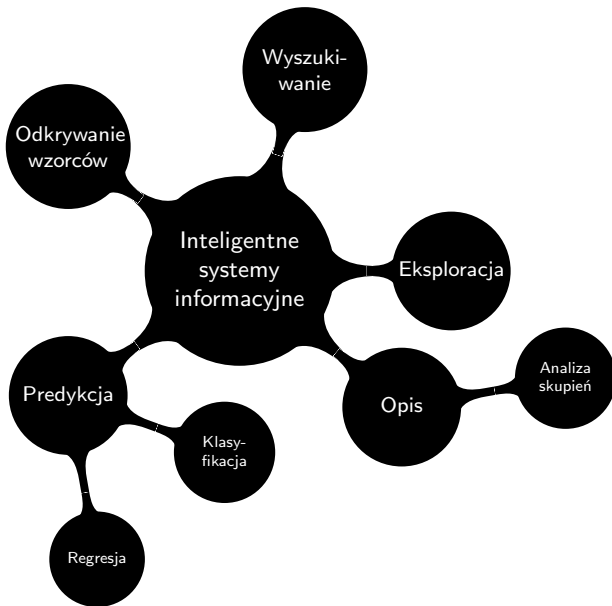
Filip
Galiński

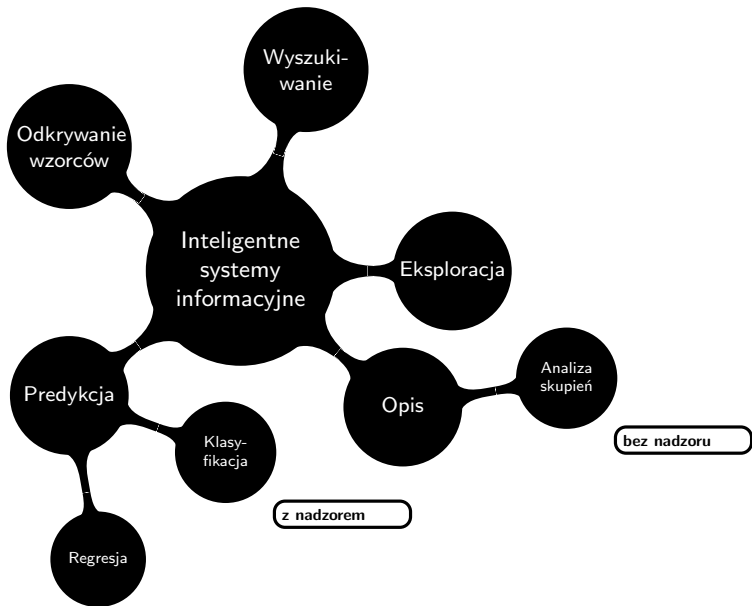
Grupowanie

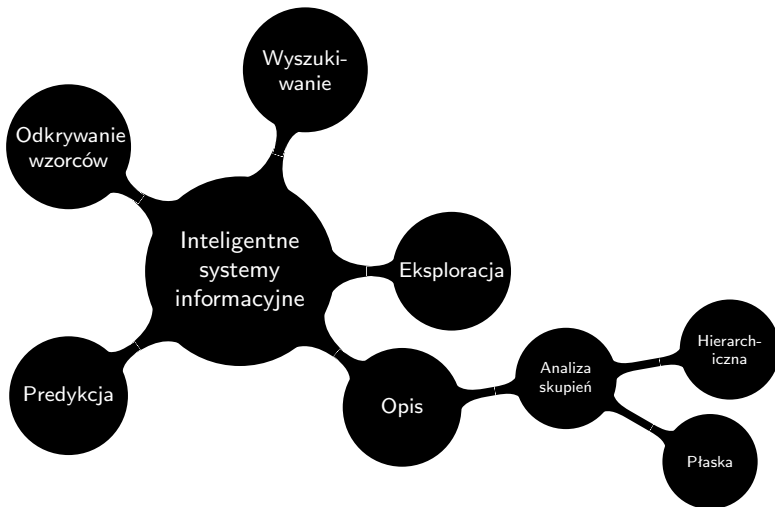


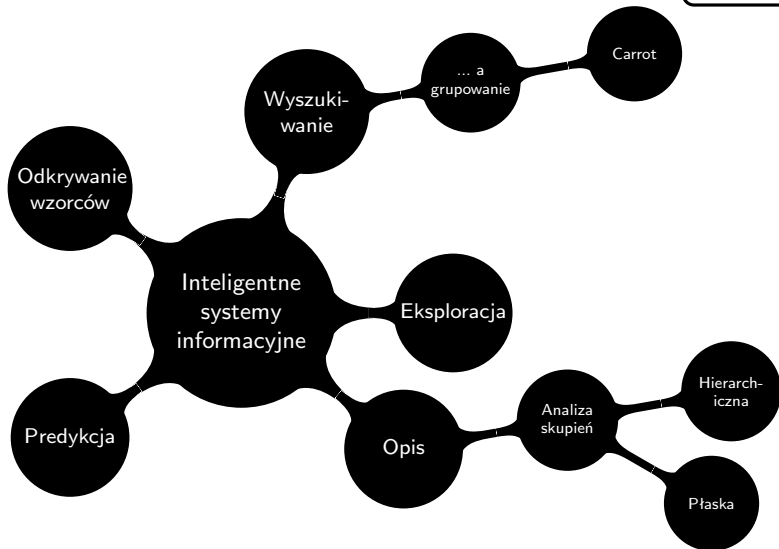








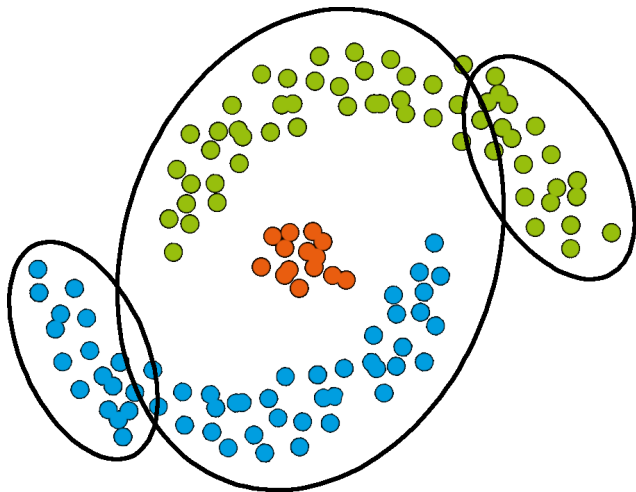




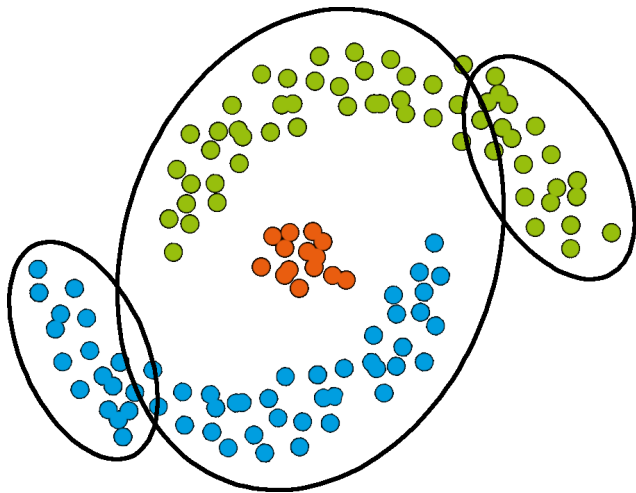
Cel:

- ▶ jak największe podobieństwo między elementami skupienia
- ▶ jak największa odległość między skupieniami

Co może pójść nie tak? (1)

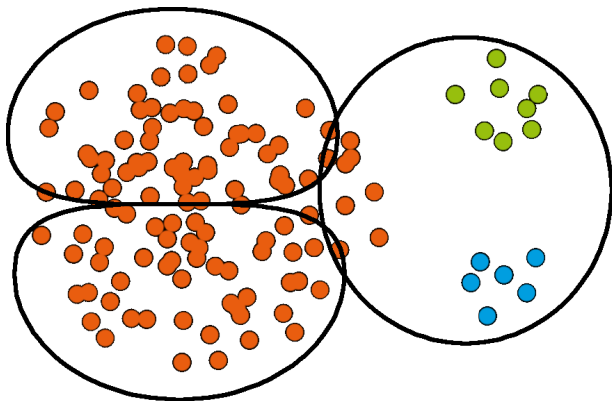


Co może pójść nie tak? (1)

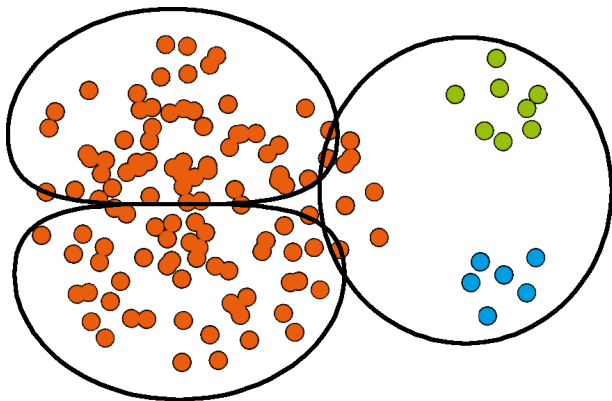


Niesferyczne grupy

Co może pójść nie tak? (2)

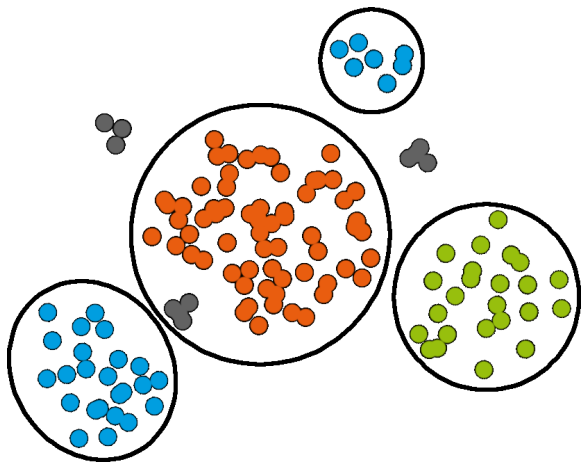


Co może pójść nie tak? (2)

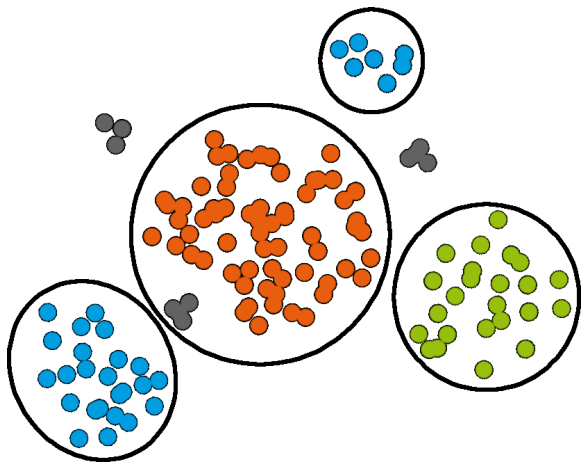


Grupy różnej wielkości

Co może pójść nie tak? (3)

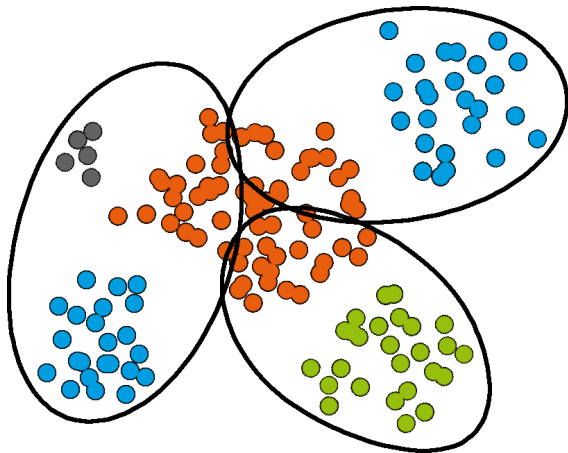


Co może pójść nie tak? (3)

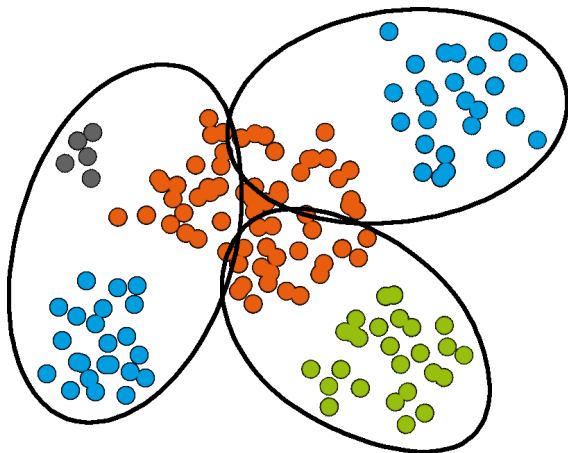


Nierozpoznane grupy małych rozmiarów

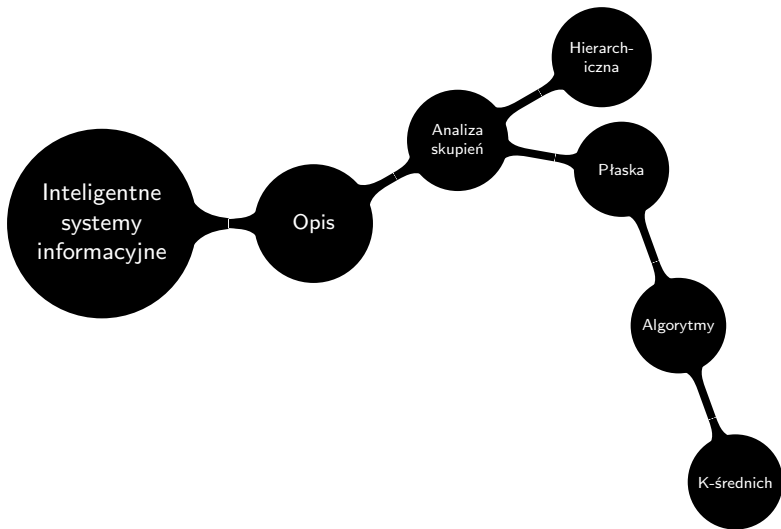
Co może pójść nie tak? (4)



Co może pójść nie tak? (4)



Zła liczba grup



Algorytm K-średnich

Require: liczba K grup, zbiór obiektów X

Ensure: podział na grupy C

- 1: Wylosuj K obiektów jako początkowe centroidy;
 - 2: **repeat**
 - 3: Dla każdego obiektu x_i znajdź najbliższy centroid c_j i przypisz go do grupy C_j ;
 - 4: Oblicz nowe centroidy dla każdej grupy;
 - 5: **until** przynajmniej jeden centroid się zmienił
-

Algorytm K-średnich

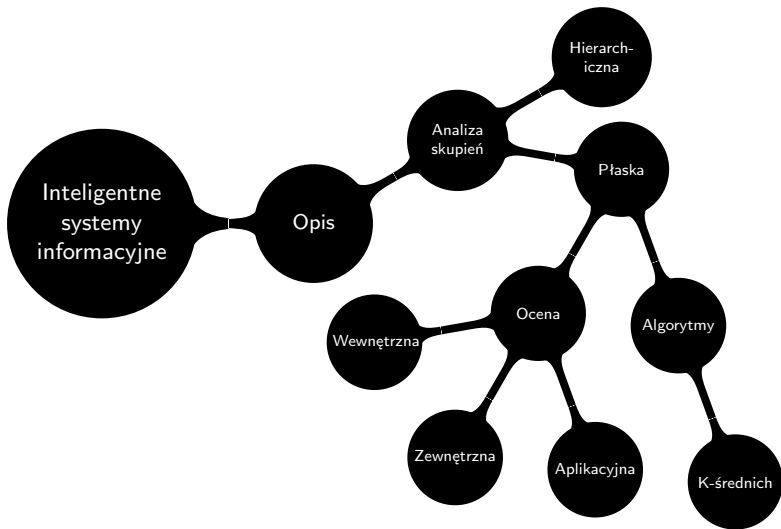
Require: liczba K grup, zbiór obiektów X

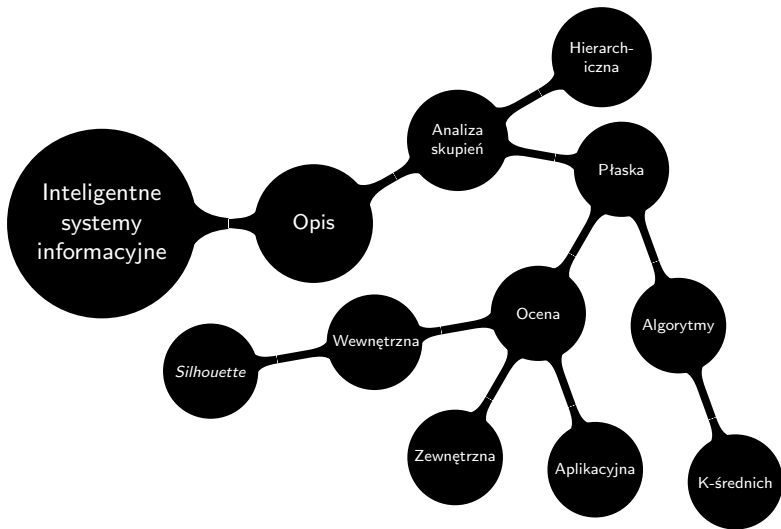
Ensure: podział na grupy C

- 1: Wylosuj K obiektów jako początkowe centroidy;
 - 2: **repeat**
 - 3: Dla każdego obiektu x_i znajdź najbliższy centroid c_j i przypisz go do grupy C_j ;
 - 4: Oblicz nowe centroidy dla każdej grupy;
 - 5: **until** przynajmniej jeden centroid się zmienił
-

Cel: minimalizacja $RSS(C)$

$$RSS(C) = \sum_{j=1}^k \sum_{x_i \in C_j} \|x_i - \bar{x}_j\|^2 \quad (1)$$





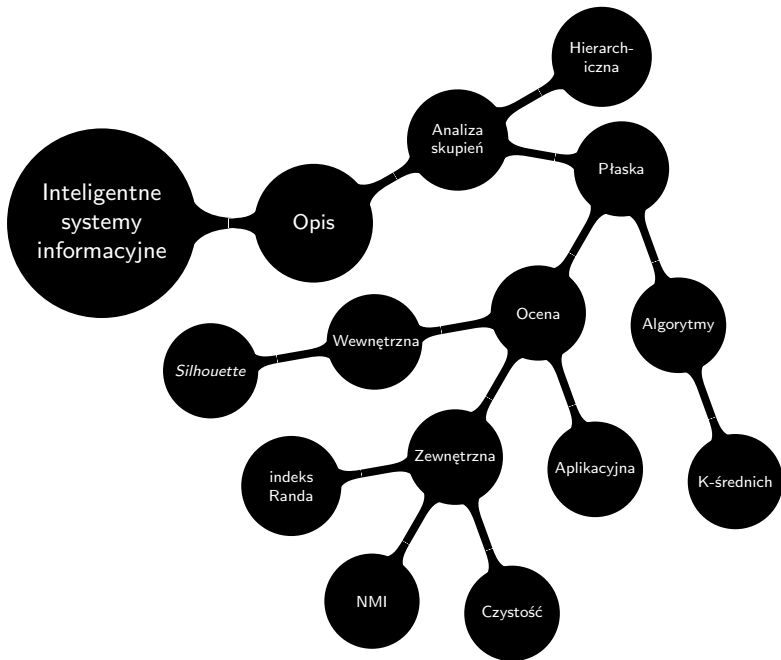
Silhouette

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (2)$$

Silhouette

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (2)$$

- ▶ $a(i)$ – średnia odległość x_i do obiektów z C_j ,
- ▶ $b(i)$ – najmniejsza ze średnich odległości x_i do obiektów z $C_{m \neq j}$.



Czystość

$$\text{purity}(C, L) = \frac{1}{n} \sum_k \max_j |c_k \cap l_j| \quad (3)$$

Znormalizowana wspólna informacja

$$NMI(C, L) = \frac{I(C, L)}{[H(C) + H(L)]/2} \quad (4)$$

Znormalizowana wspólna informacja

$$NMI(C, L) = \frac{I(C, L)}{[H(C) + H(L)]/2} \quad (4)$$

- ▶ $I(C, L)$ – wspólna informacja C i L ,
- ▶ $H(\cdot)$ – entropia.

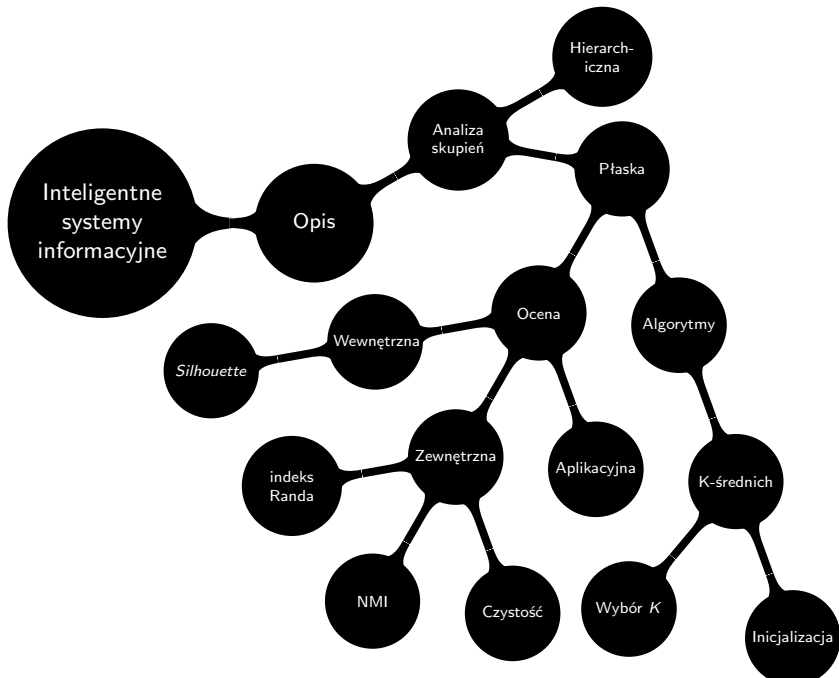
indeks Randa

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \quad (5)$$

indeks Randa

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \quad (5)$$

- ▶ TP – pary z tej samej klasy w tym samym skupieniu
- ▶ TN – pary z różnych klas w różnych skupieniach
- ▶ FP – pary z różnych klas w tym samym skupieniu
- ▶ FN – pary z tej samej klasy w różnych skupieniach

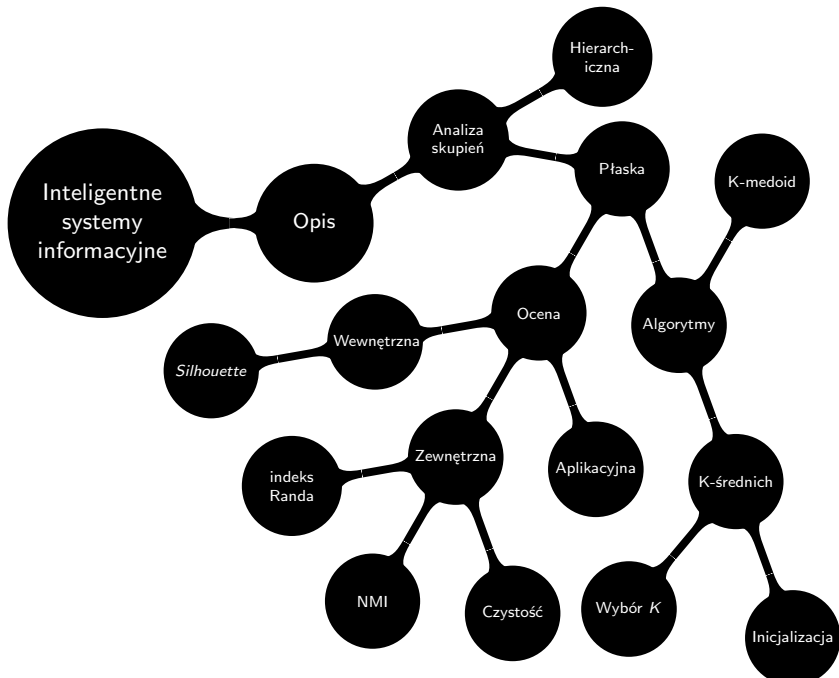


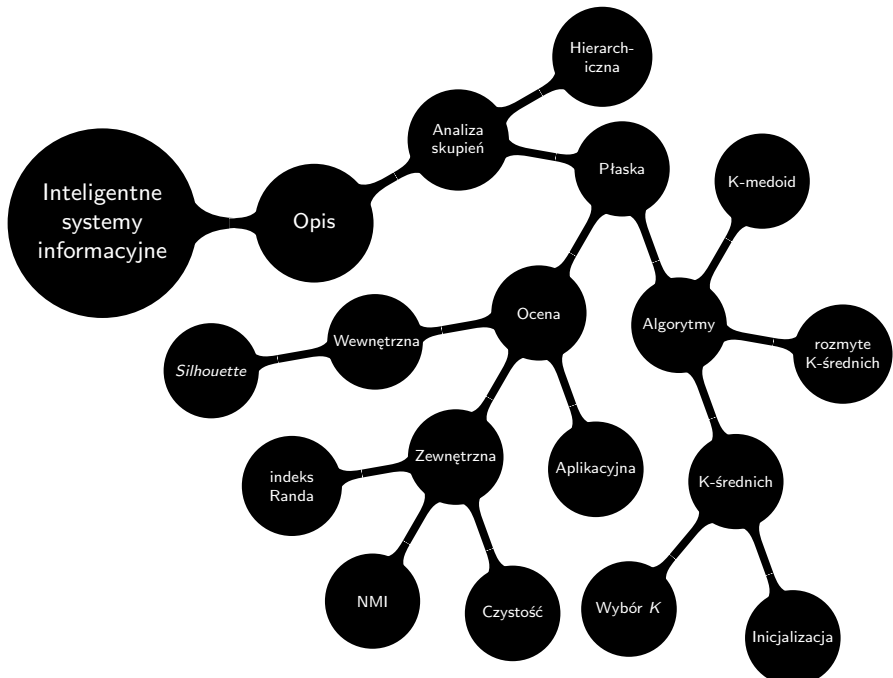
Algorytm K -średnich++

Require: liczba K grup, zbiór punktów X

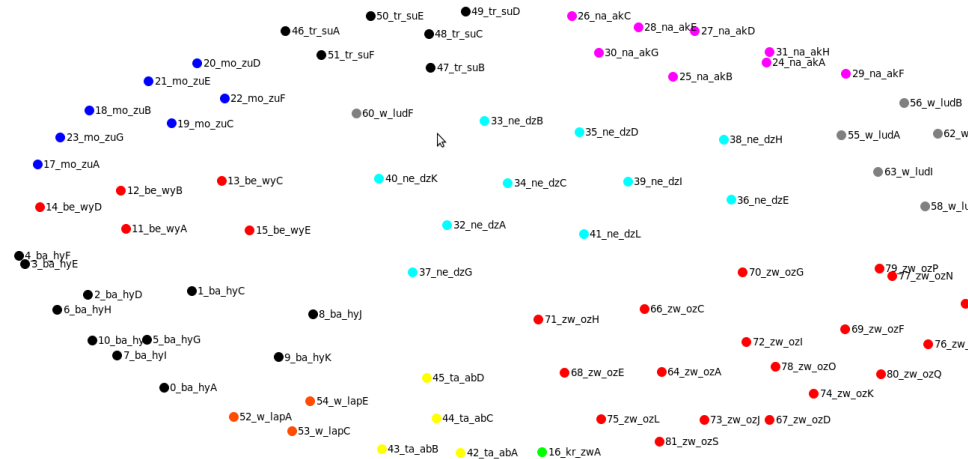
Ensure: K początkowych centroidów

- 1: Wybierz ze zbioru X pierwszy punkt c_1 w sposób losowy;
 - 2: **repeat**
 - 3: Dla każdego punktu x oblicz odległość do najbliższego centroidu $D(x)$;
 - 4: Wybierz jako kolejny centroid punkt $c_i = x'$ z najwyższą wartością współczynnika $\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$;
 - 5: **until** nie wybrano K początkowych centroidów
-

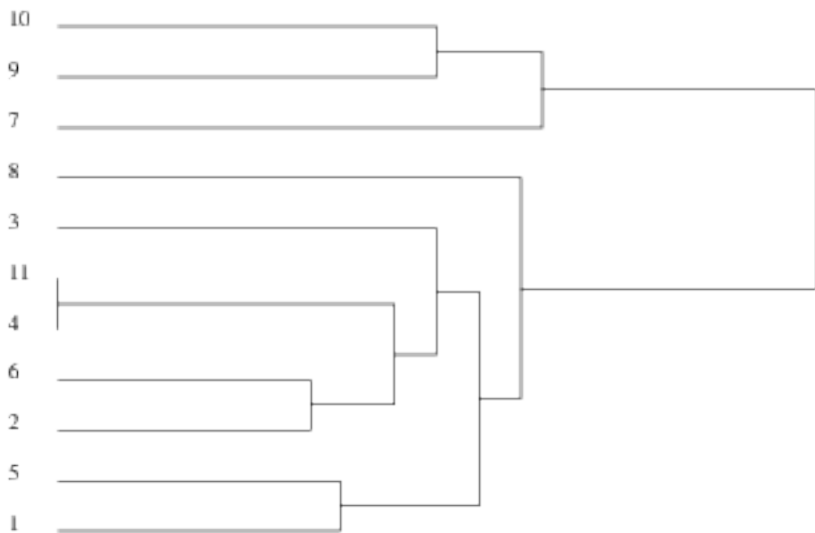


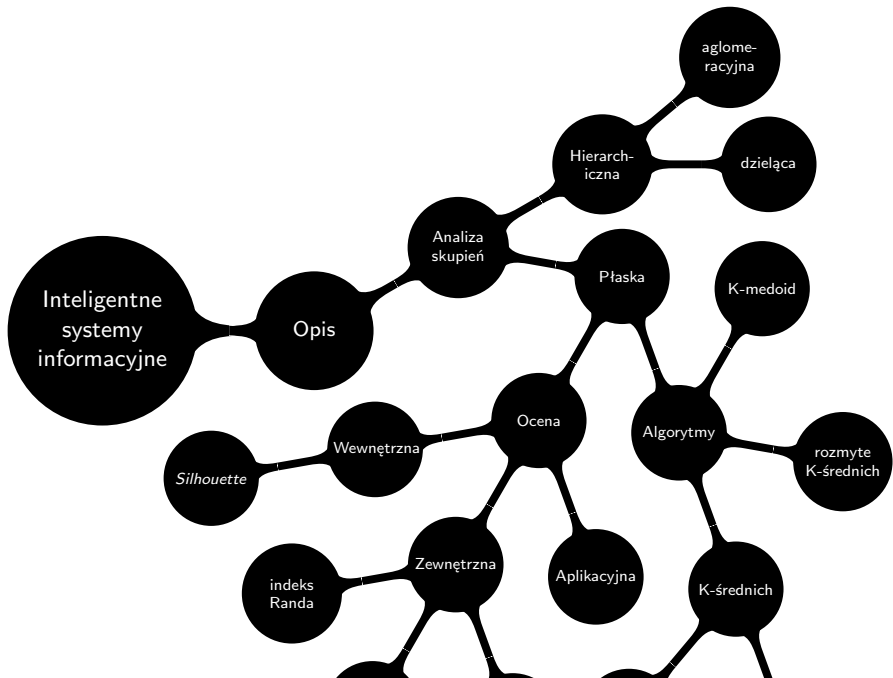


mapa legend miejskich



dendrogram



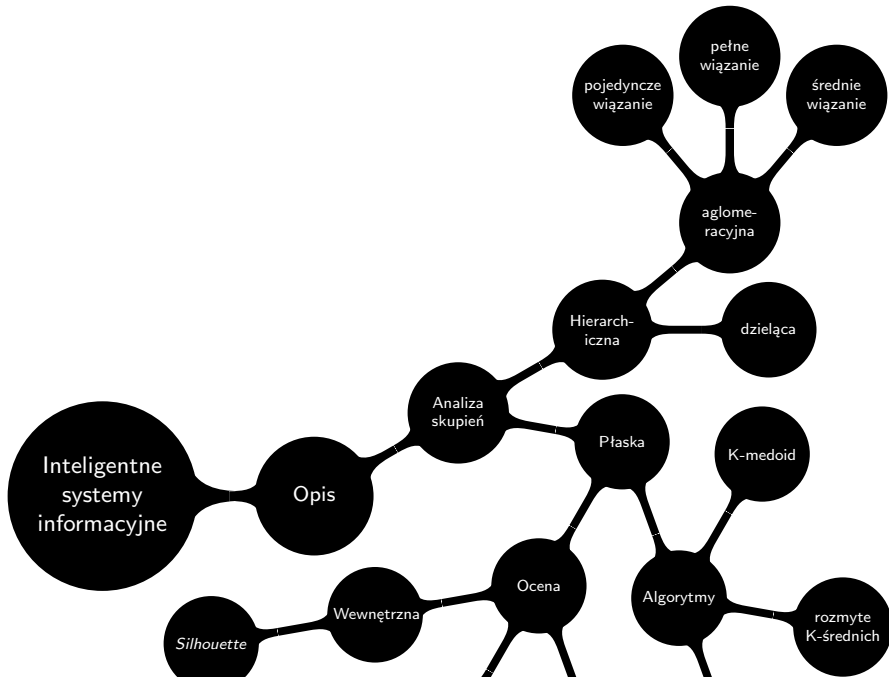


Aglomeracyjny algorytm hierarchiczny (HAC)

Require: zbiór dokumentów X

Ensure: podział na grupy

- 1: Utwórz grupę dla każdego dokumentu;
 - 2: **while** wszystkie dokumenty nie znajdują się w jednej grupie **do**
 - 3: Znajdź najbliższą parę grup (i, j) ;
 - 4: Połącz grupy i i j ;
 - 5: **end while**
-



Dzielący algorytm hierarchiczny (*HDC*)

Require: zbiór dokumentów X

Ensure: podział na grupy

- 1: Utwórz grupę zawierającą wszystkie dokumenty;
 - 2: **while** każdy dokument nie znajduje się w innej grupie **do**
 - 3: Zbadaj wszystkie możliwe podziały bieżących grup;
 - 4: Wybierz najlepszy podział zgodnie z ustalonym kryterium;
 - 5: **end while**
-

Algorytm dwudzielny K-średnich

Require: liczba K grup, liczba wewnętrznych iteracji J , zbiór dokumentów X

Ensure: podział na grupy C

- 1: Utwórz grupę zawierającą wszystkie dokumenty;
 - 2: **while** nie otrzymano K grup **do**
 - 3: Wybierz grupę do podziału C_i ;
 - 4: Znajdź metodą *K-średnich* podział grupy C_i na dwie grupy dla J różnych losowych par centroidów i wybierz najlepszy podział;
 - 5: **end while**
-

Intermezzo

Algorytm EM (oczekiwania-maksymalizacji)

Jaką klasę problemów możemy rozwiązać za pomocą EM?

- ▶ $D = \vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$ – obserwowane wektory danych
- ▶ $H = z_1, z_2, \dots, z_n$ – zmienne ukryte
- ▶ θ – parametry modelu probabilistycznego

Algorytm oczekiwania-maksymalizacji (EM)

Iteruj:

Krok E estymujemy rozkład Q_{k+1} zmiennych H biorąc pod uwagę konkretne θ_k

Krok M na podstawie Q_{k+1} wyznaczamy θ_{k+1}

Algorytm EM w grupowaniu

Zakładamy, że dane pochodzą z wielowymiarowego modelu mieszanego:

$$f(\vec{x}) = \sum_{k=1}^K \pi_k f_k(\vec{x}; \theta_k)$$

W niniejszym materiale użyto fragmentów pracy magisterskiej Romana Grundkiewicza *Automatyczne wyszukiwanie i grupowanie krótkich tekstów narracyjnych zamieszczanych w Internecie*