Zaawansowane Metody Analizy Danych w Biologii Molekularnej

Semest Letni 2018

Prowadzący

Zespół Biologii Obliczeniowej IPI PAN

- dr Magdalena Mozolewska (5 zajęć)
- dr Michal J. Dąbrowski (5 zajęć)
- **dr inż. Michał Dramiński** (5 zajęć)

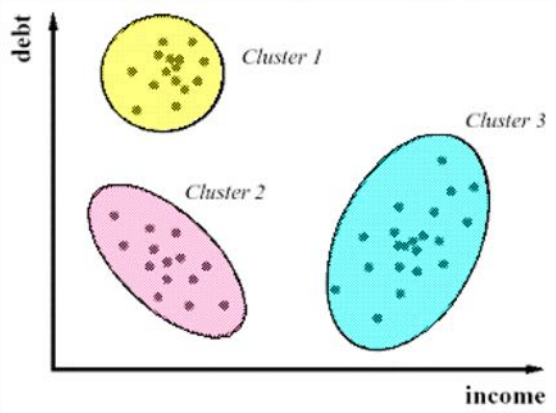
http://zmadbm.ipipan.waw.pl/

Agenda zajęć – R + Analiza danych

1. Wprowadzenie do przedmiotu. Wprowadzenie do R i RStudio.
2. Wstępna analiza danych. Wprowadzenie do statystyki.
3. Modelowanie danych oraz ocena jakości predykcji i klasyfikacji.
4. **Grupowanie obiektów oraz selekcja cech.**
5. Pułapki w analizie danych o dużym rozmiarze.

Grupowanie danych

Uczenie bez nadzoru



- **analiza skupień = grupowanie = clustering**
- **Polega na odkrywaniu jednorodnych grup w danych**. Zadanie optymalizacji gdzie maksymalizujemy **funkcję podobieństwa** dla obiektów wewnątrz grup i minimalizujemy dla obiektów będących w różnych grupach.
- similarity = 1-distance
- Odległość: Euklidesowa, Hamminga, Mahalanonisa itp.

Dane

zmienne/cechy (features)

zmienna decyzyjna

przykłady/obiekty

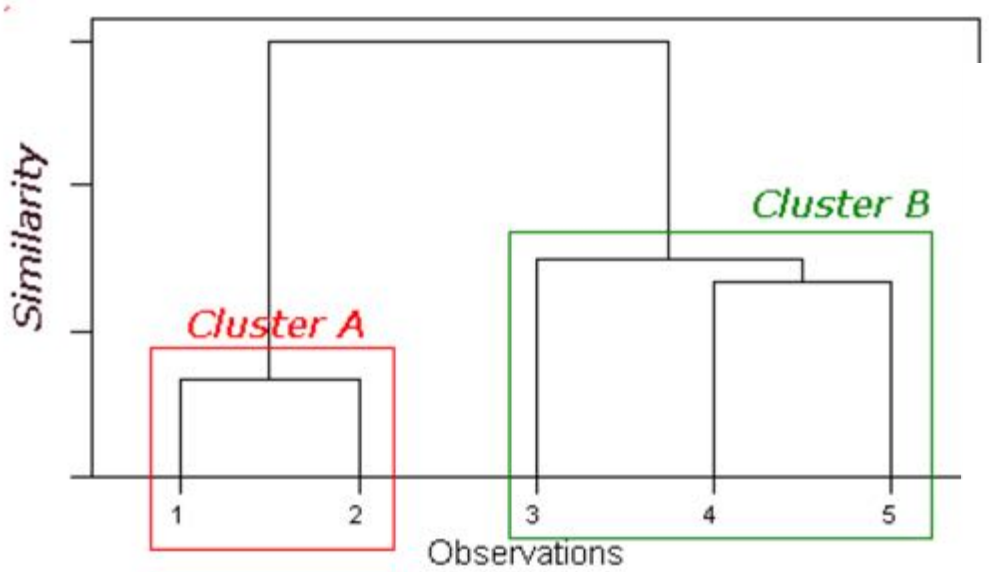
outlook	temperature	humidity	windy	play
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rainy	71	91	TRUE	no

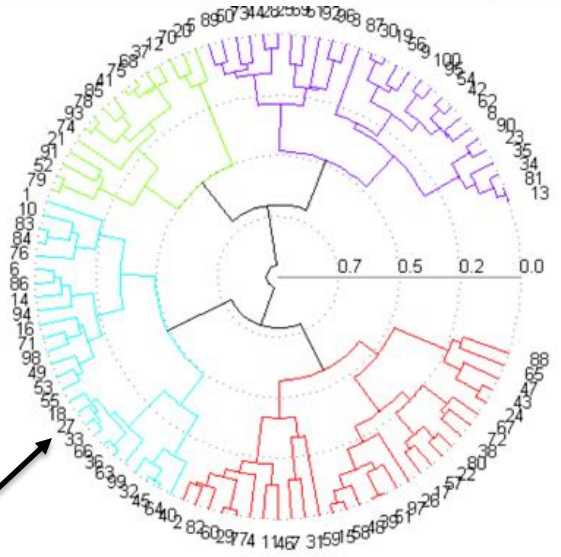
Grupowanie

- **Metody hierarchiczne** – budujemy drzewo podobieństwa tzw dendrogram.
- Grupa metod **k-średnich (k-means)** – na końcu każdy obiekt jest przydzielony do jednej kategorii.
- Metody **rozmytej analizy skupień** (*fuzzy clustering*) mogą przydzielać jeden obiekt do więcej niż jednej kategorii (np. *c-means*).

Grupowanie

- **metody hierarchiczne**
  - procedury aglomeracyjne (ang. *agglomerative*),
  - procedury deglomeracyjne (ang. *divisive*)



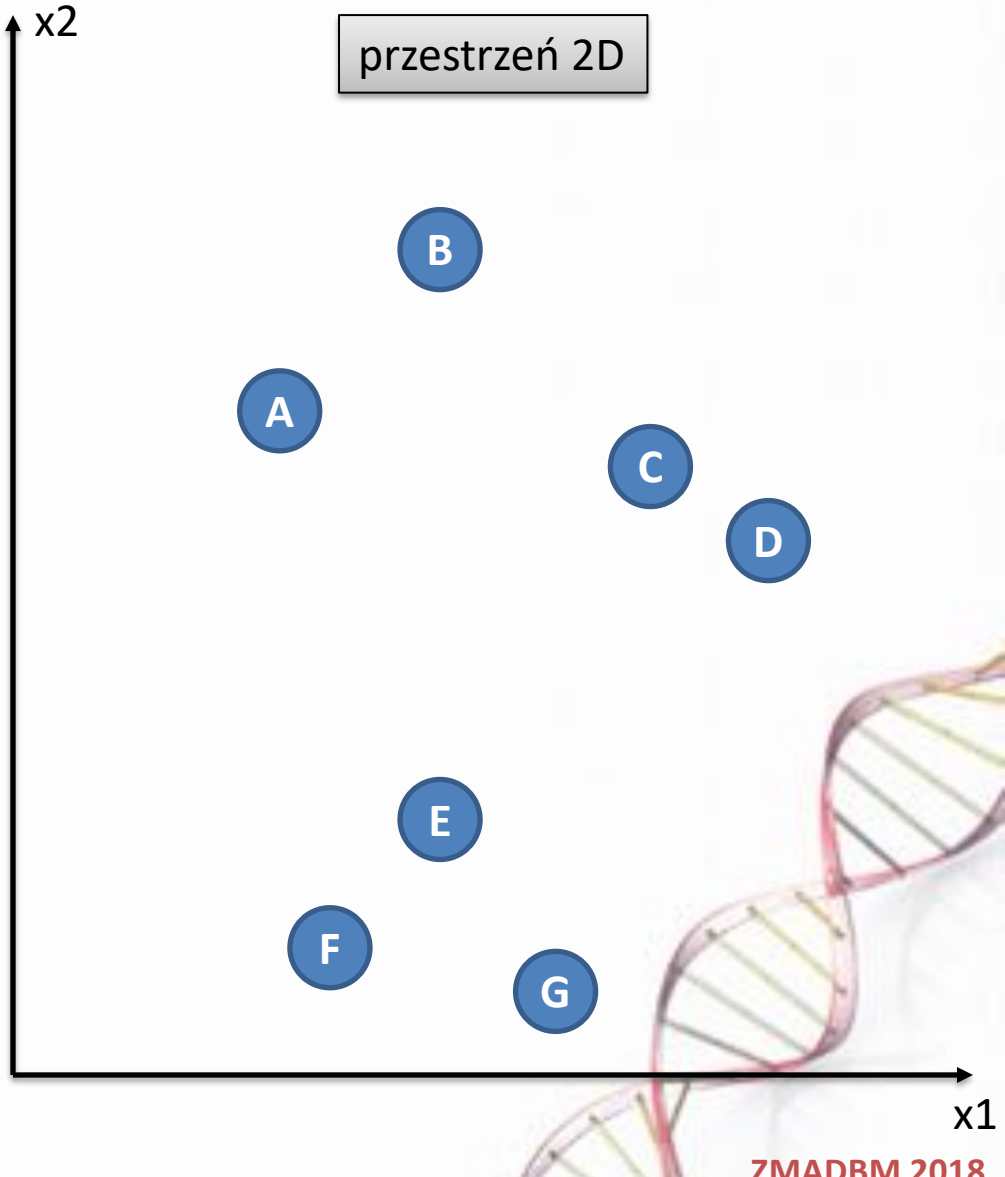


dendrogramy

Grupowanie hierarchiczne(aglomeracyjne)

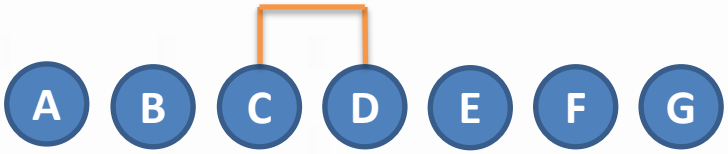
dendrogram

przestrzeń 2D

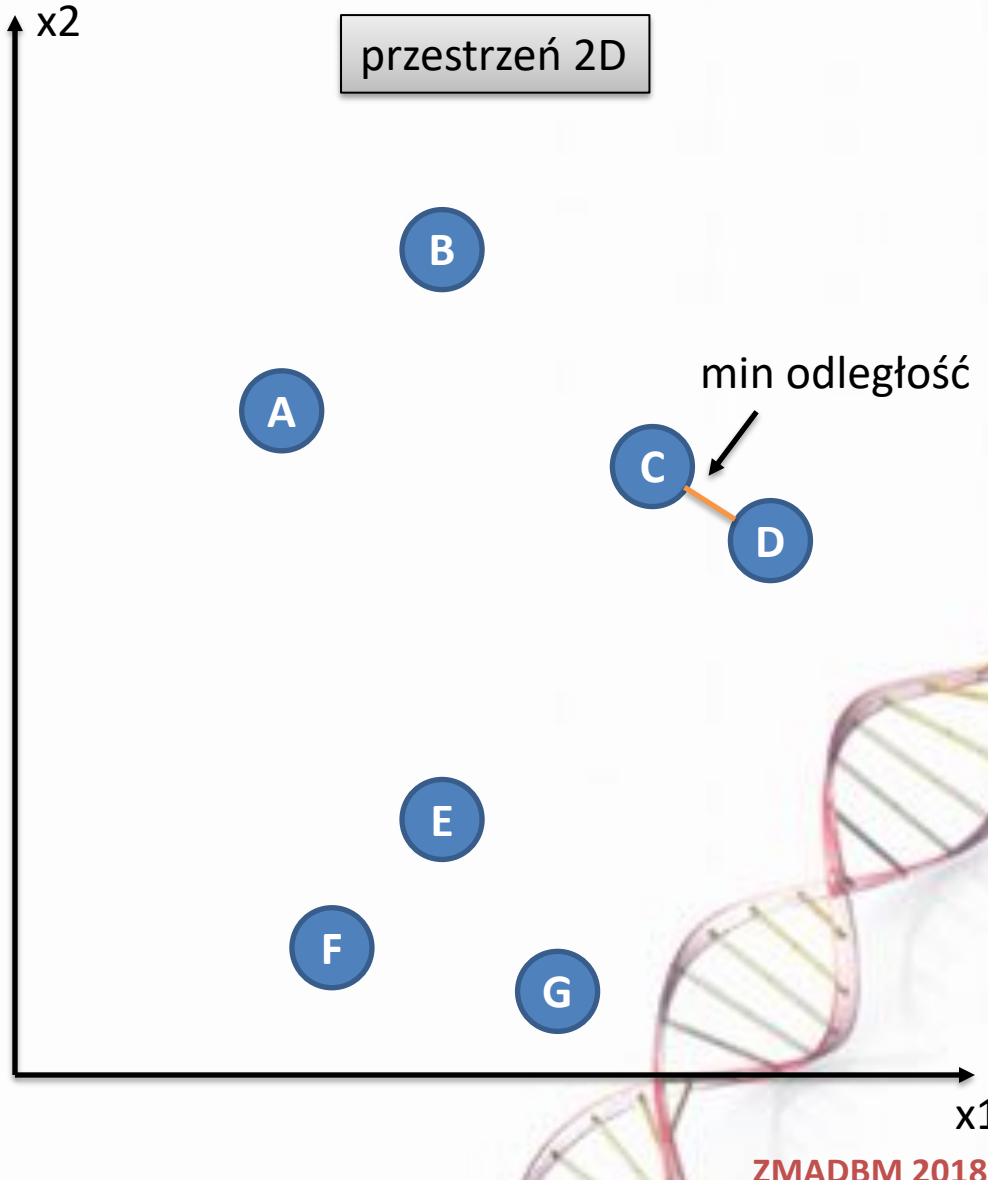


Grupowanie hierarchiczne(aglomeracyjne)

dendrogram

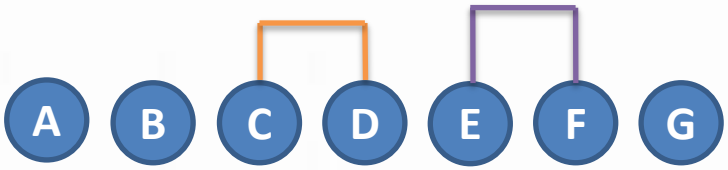


przestrzeń 2D

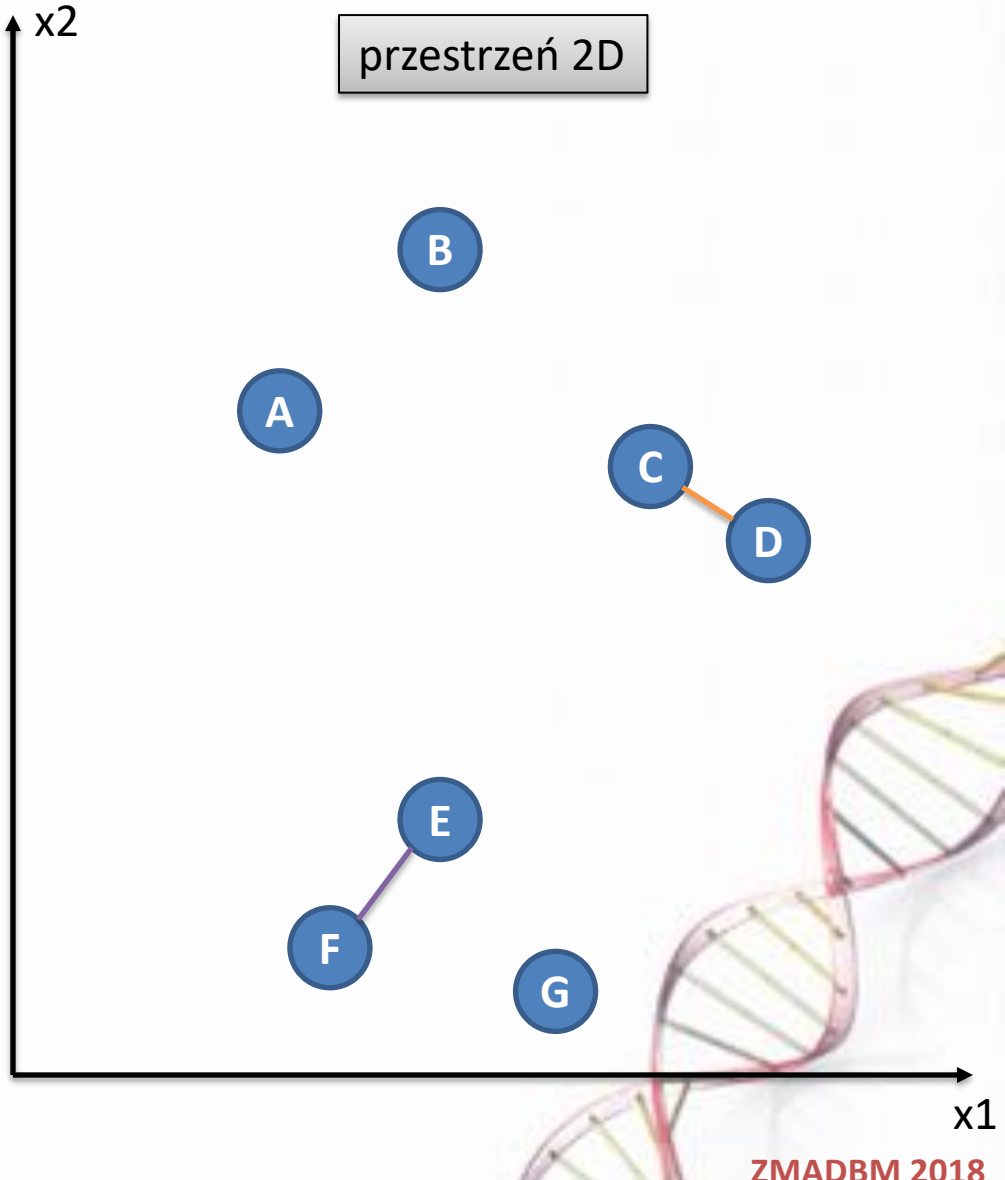


Grupowanie hierarchiczne(aglomeracyjne)

dendrogram

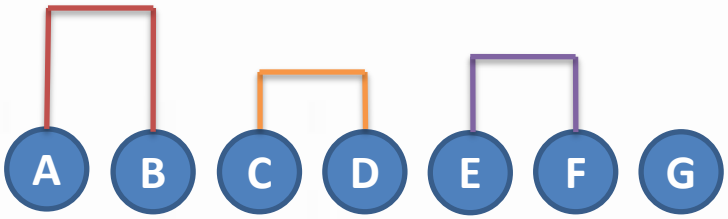


przestrzeń 2D

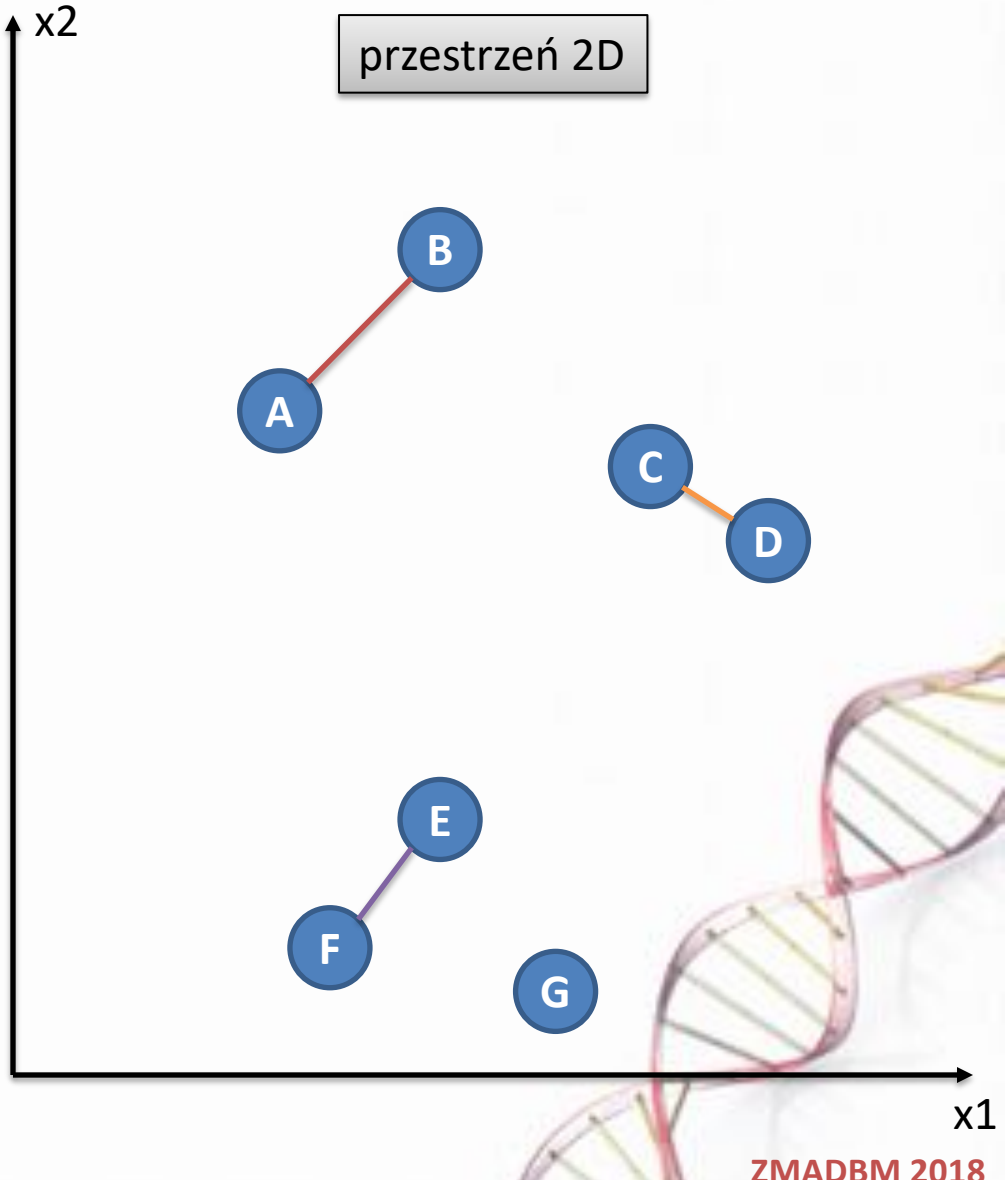


Grupowanie hierarchiczne(aglomeracyjne)

dendrogram

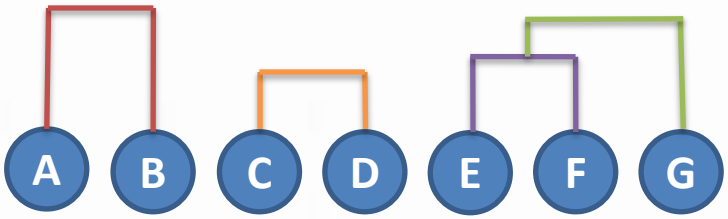


przestrzeń 2D

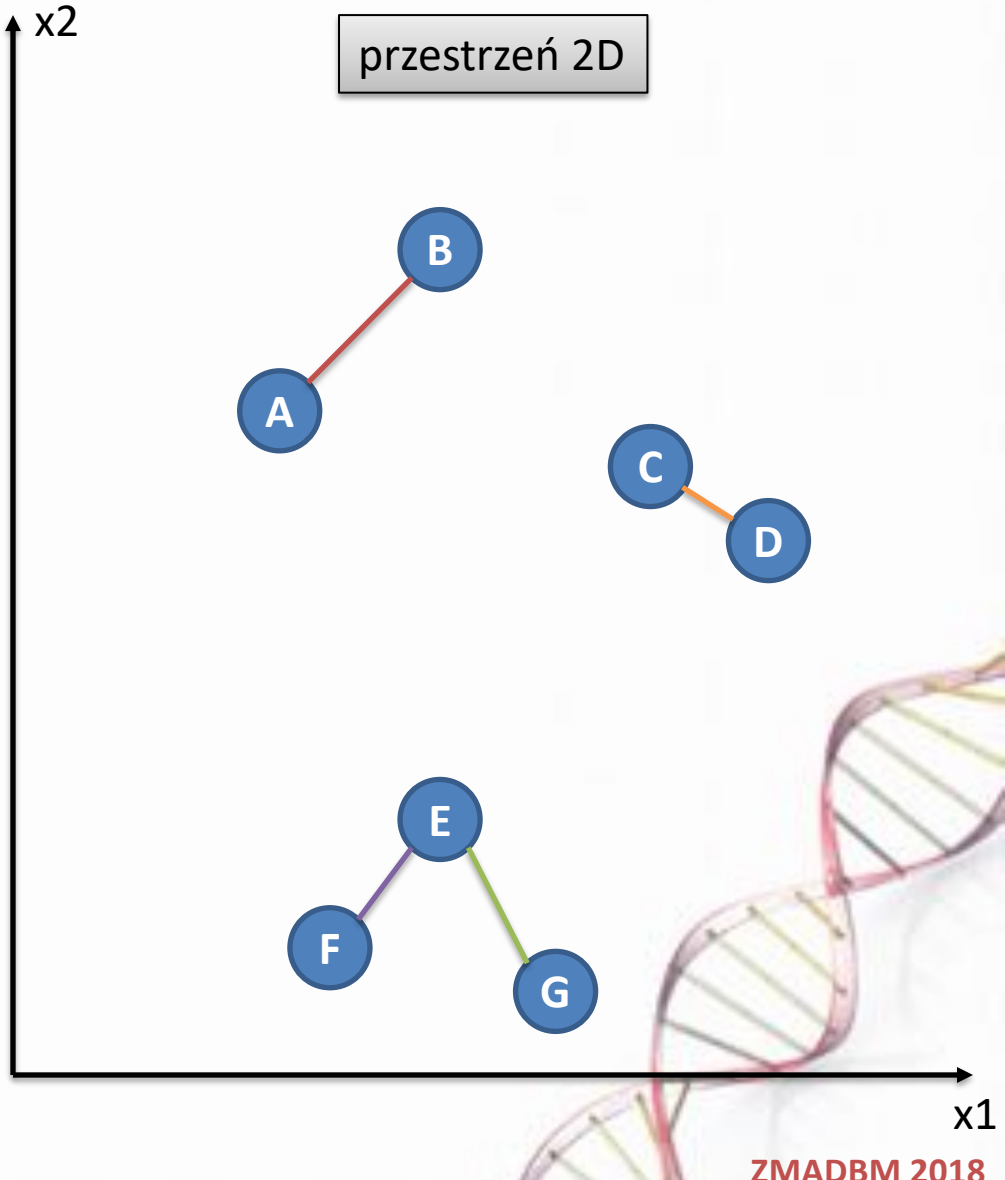


Grupowanie hierarchiczne(aglomeracyjne)

dendrogram

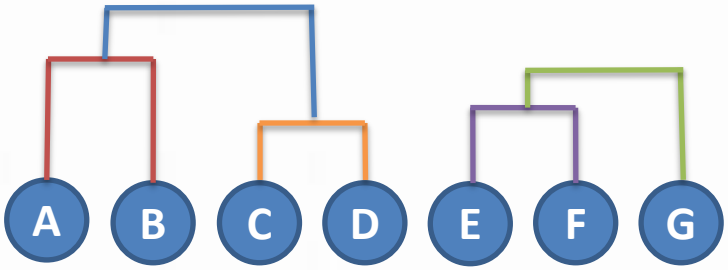


przestrzeń 2D

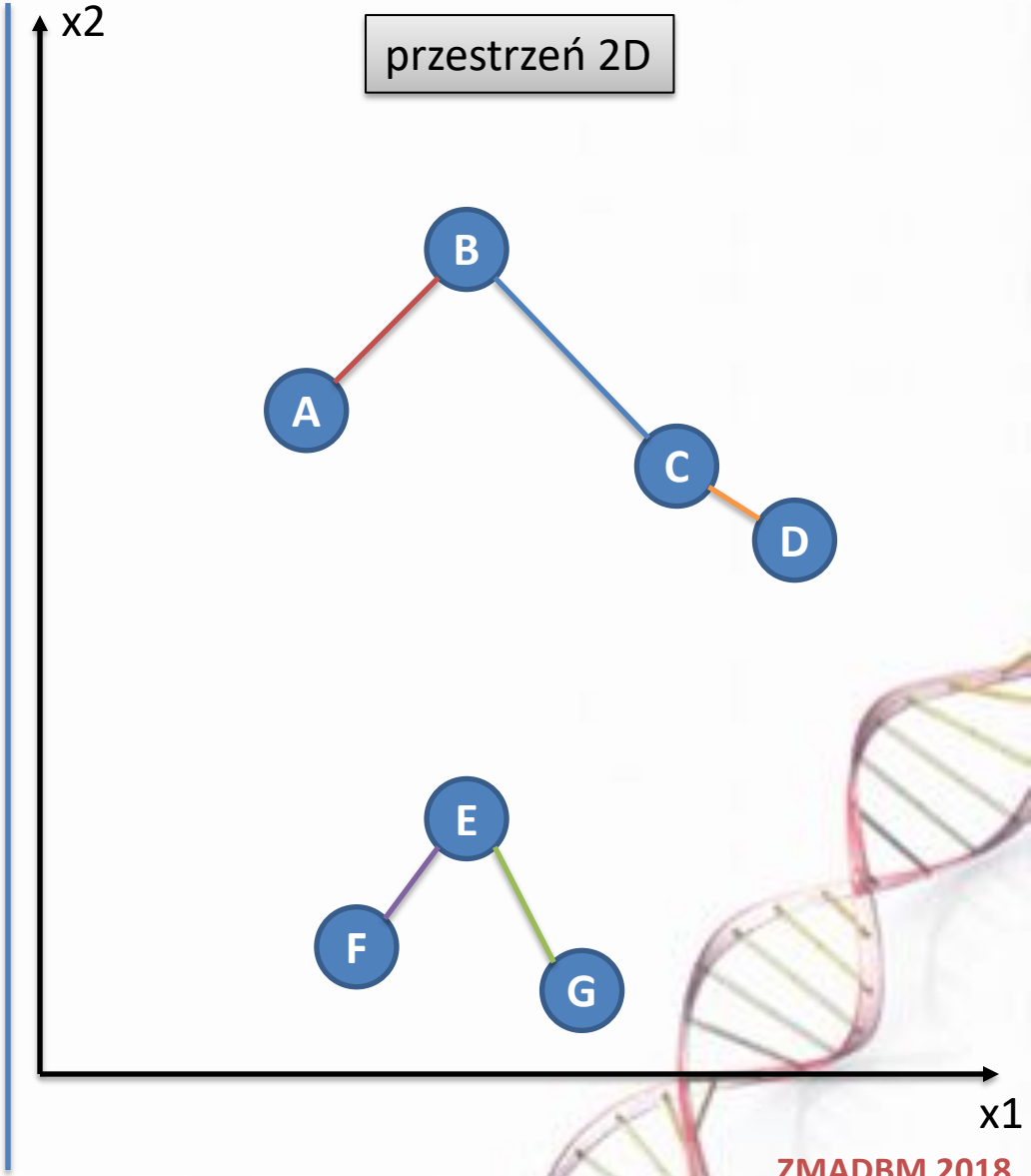


Grupowanie hierarchiczne(aglomeracyjne)

dendrogram

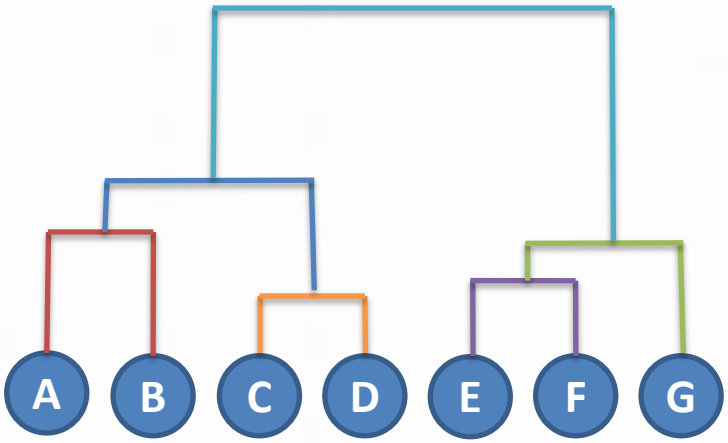


przestrzeń 2D

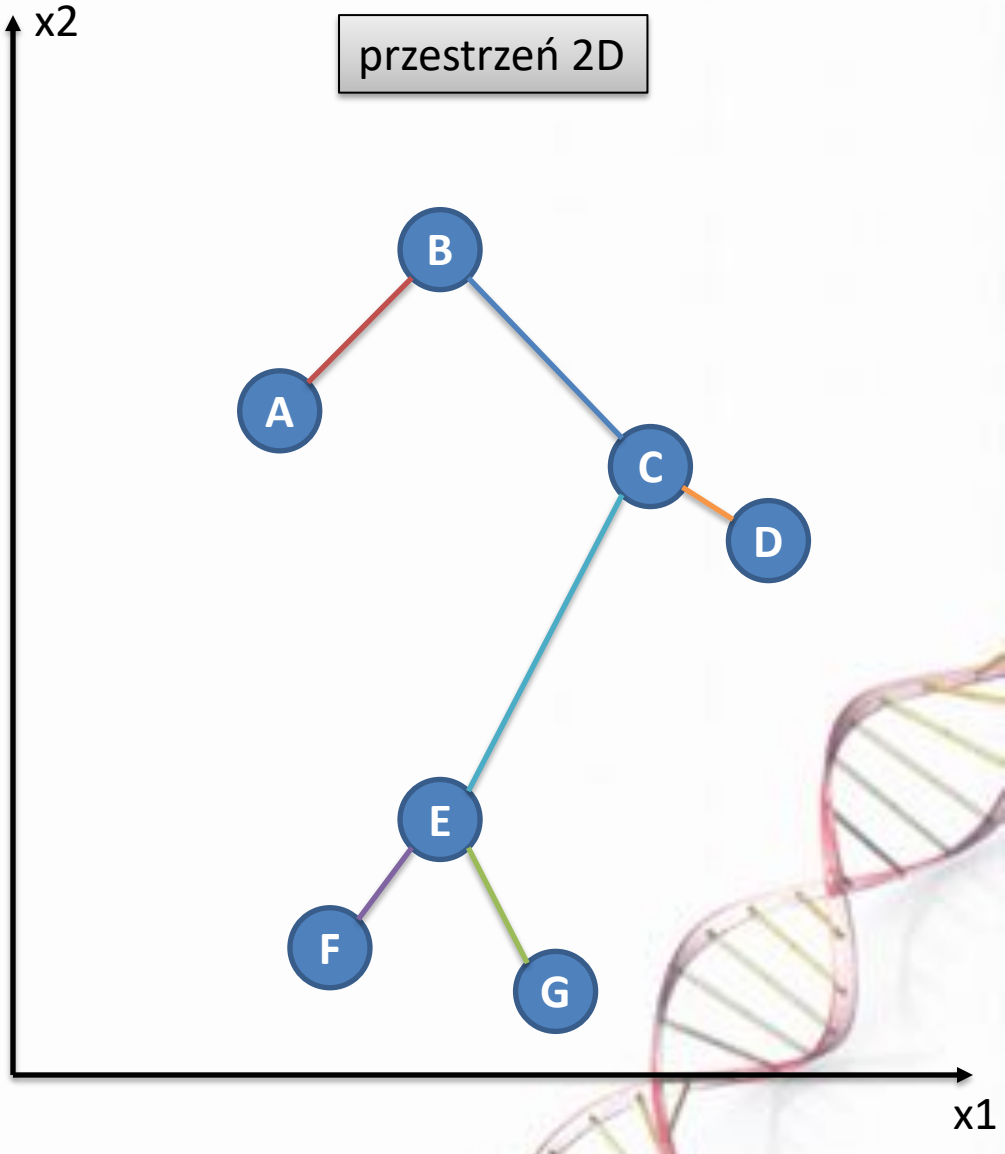


Grupowanie hierarchiczne(aglomeracyjne)

dendrogram



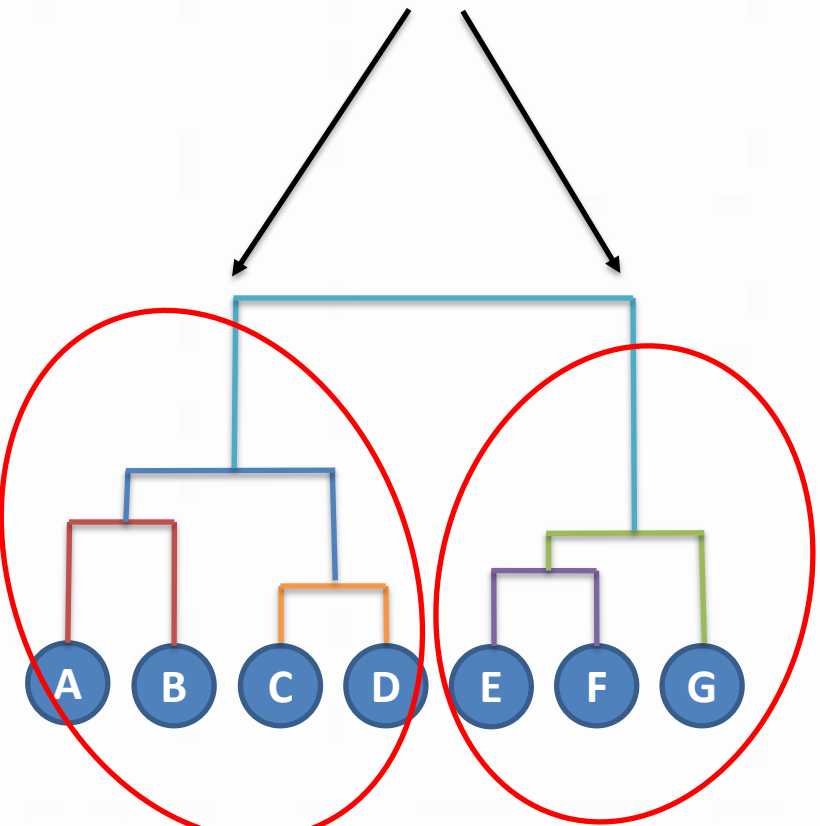
przestrzeń 2D



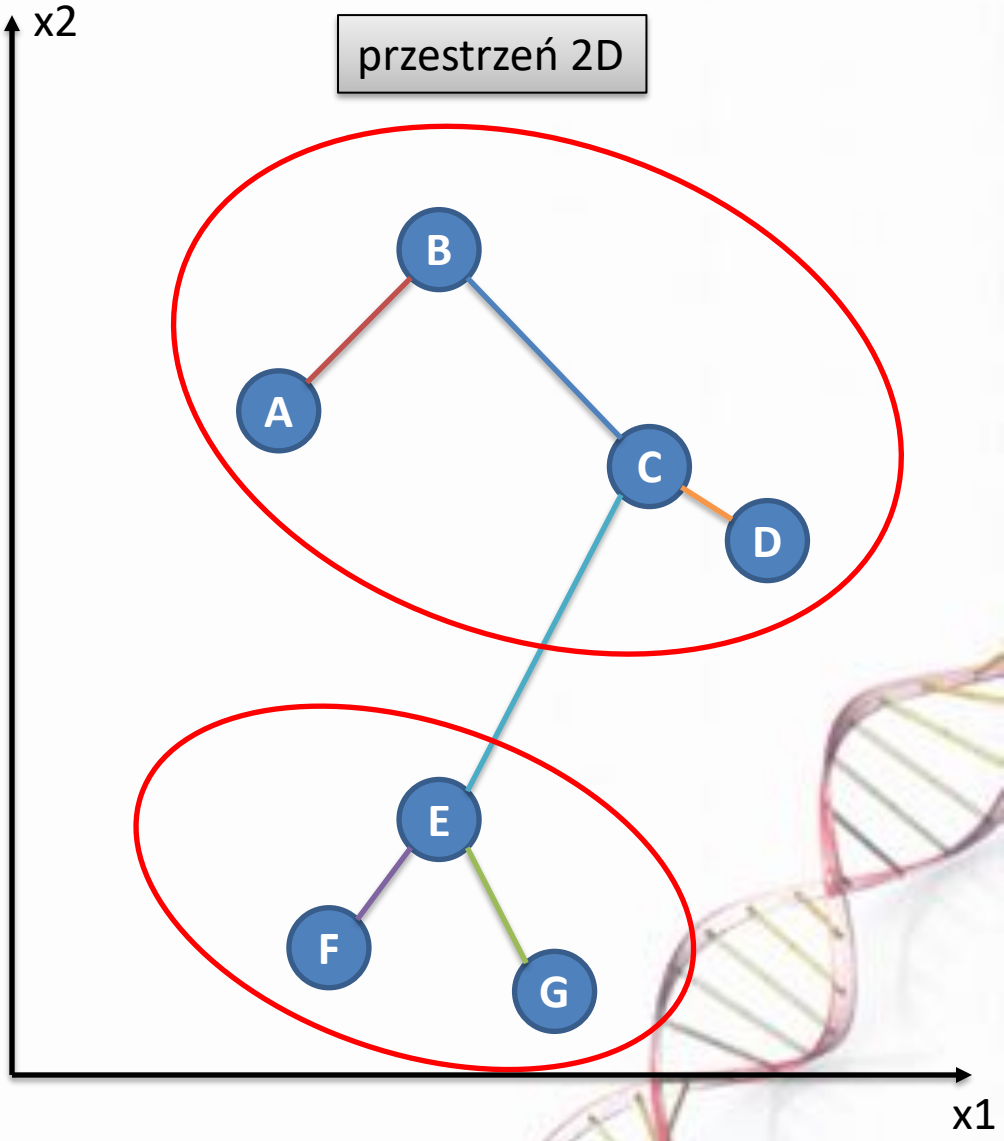
Grupowanie hierarchiczne(aglomeracyjne)

dendrogram

Dwie grupy na najwyższym poziomie



przestrzeń 2D



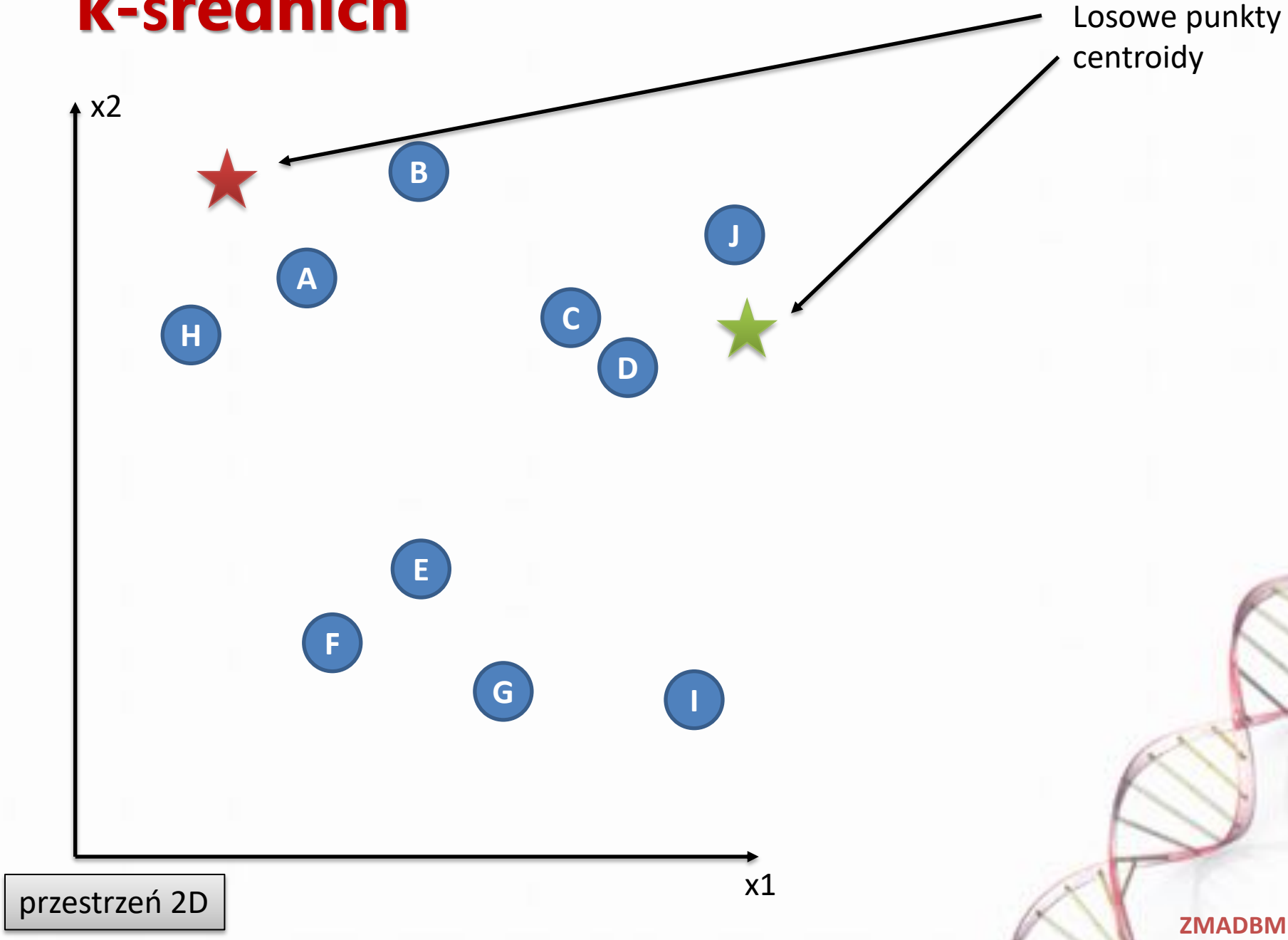
Grupowanie hierarchiczne(aglomeracyjne)

```
#grupowanie hierarchiczne
#działa tylko dla danych
numerycznych!!!
# macierz odległości
>d <- dist(as.matrix(mtcars))
# jest wejściem do procedury hclust
>hc <- hclust(d)
# narysuj dendrogram
>plot(hc)
# narysuj dendrogram
```

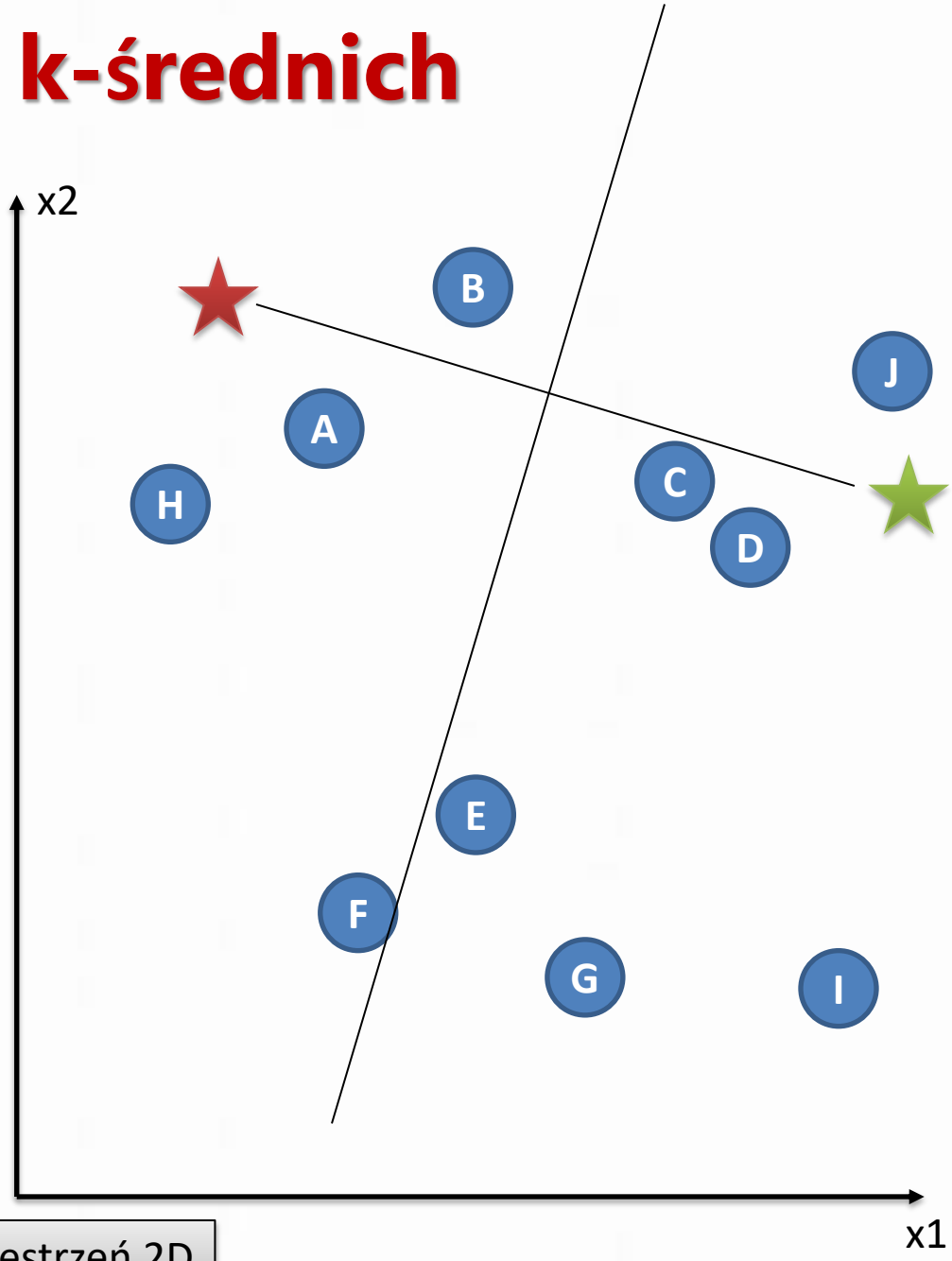
Algorytm k-średnich (k-means)

- Z góry zakładamy że w danych jest *k* grup.
- Losowo wybieramy *k* punktów startowych w przestrzeni *m* wymiarowej. To są nasze centroidy które wyznaczają punkty centralne grup.
- Iteracyjnie:
  - przypisujemy obiekty do najbliższych centroidów,
  - wyliczamy nowe środki skupień,
- Powtarzamy algorytm, aż do osiągnięcia kryterium zbieżności (najczęściej jest to krok, w którym nie zmieniła się przynależność punktów do klas lub zadana z góry liczba iteracji);

k-średnich



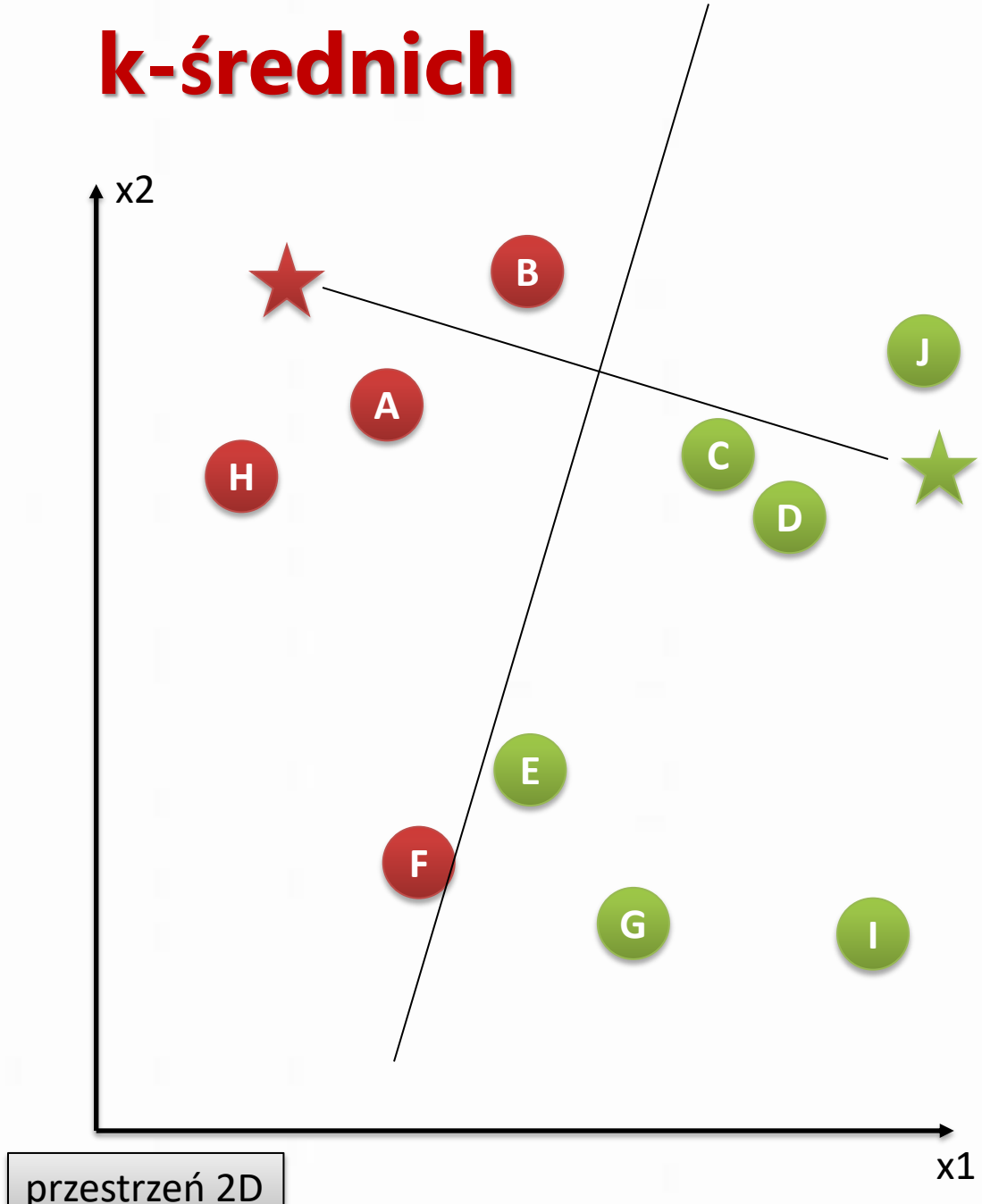
k-średnich



przestrzeń 2D

wyznacz odległości
od centroidów
do obiektów

k-średnich

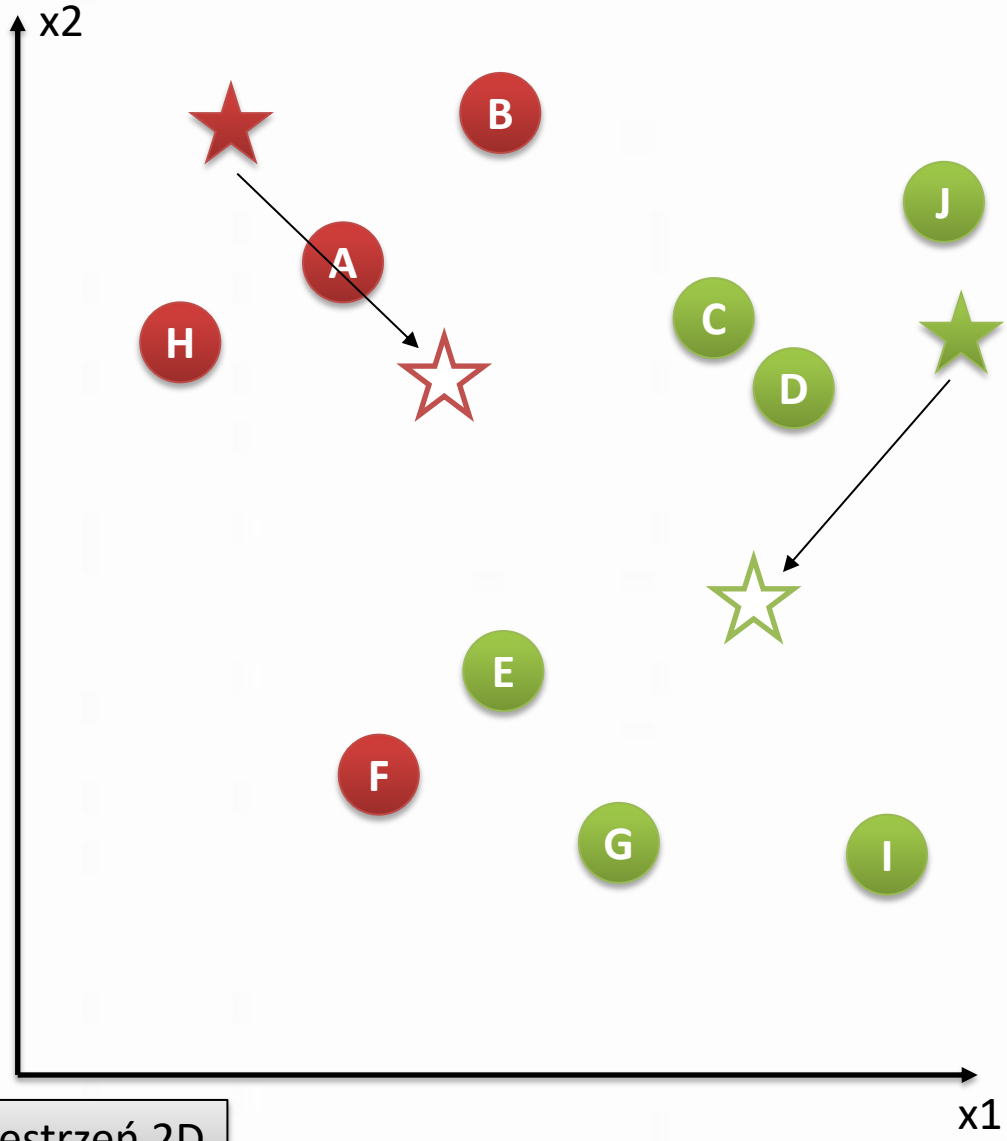


przestrzeń 2D

przydziel etykiety odpowiednich grup przykładom najbliższym centroidom

k-średnich

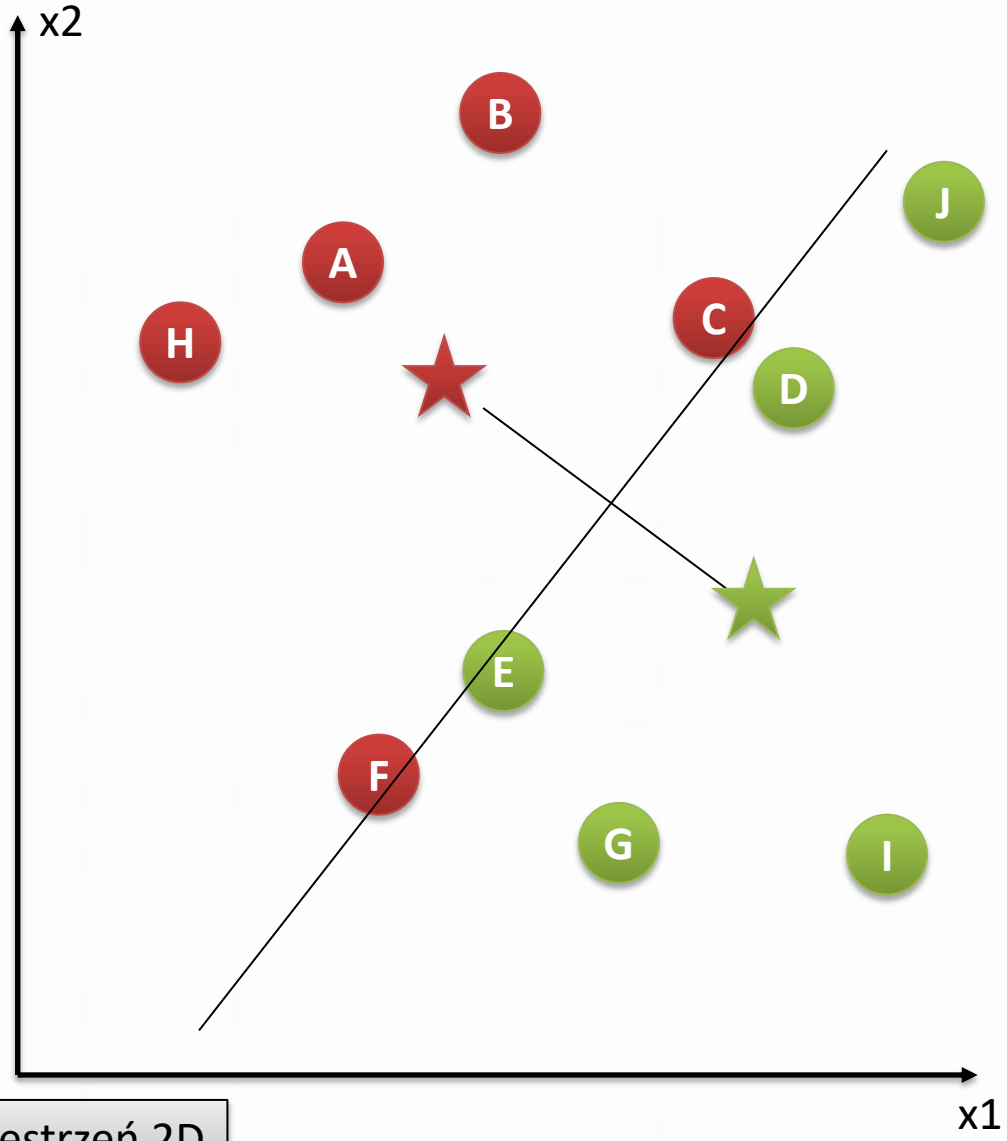
wyznacz nowe centroidy
(środki ciężkości grup)



przestrzeń 2D

k-średnich

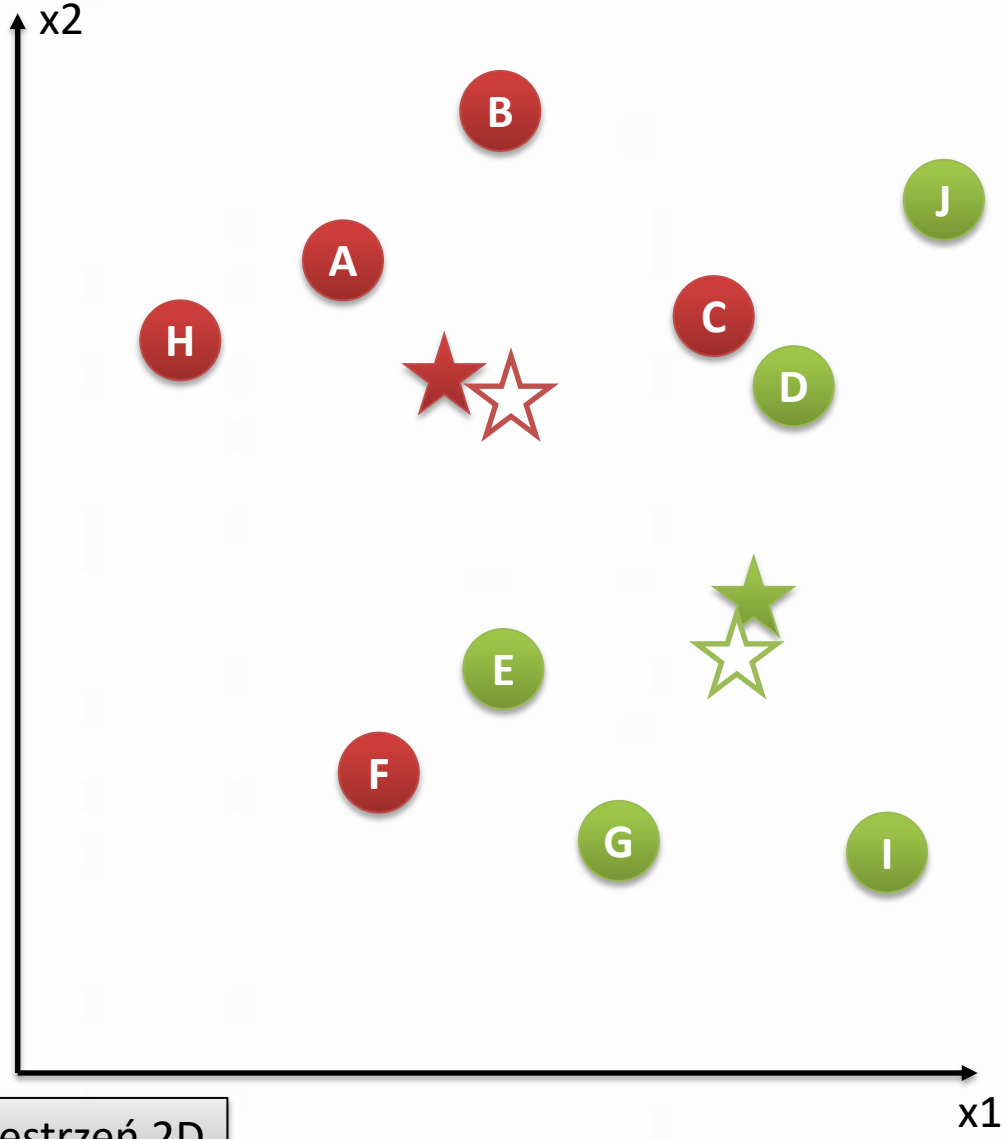
wyznacz odległości
od centroidów
do obiektów i przydziel
nowe etykiety grup



przestrzeń 2D

k-średnich

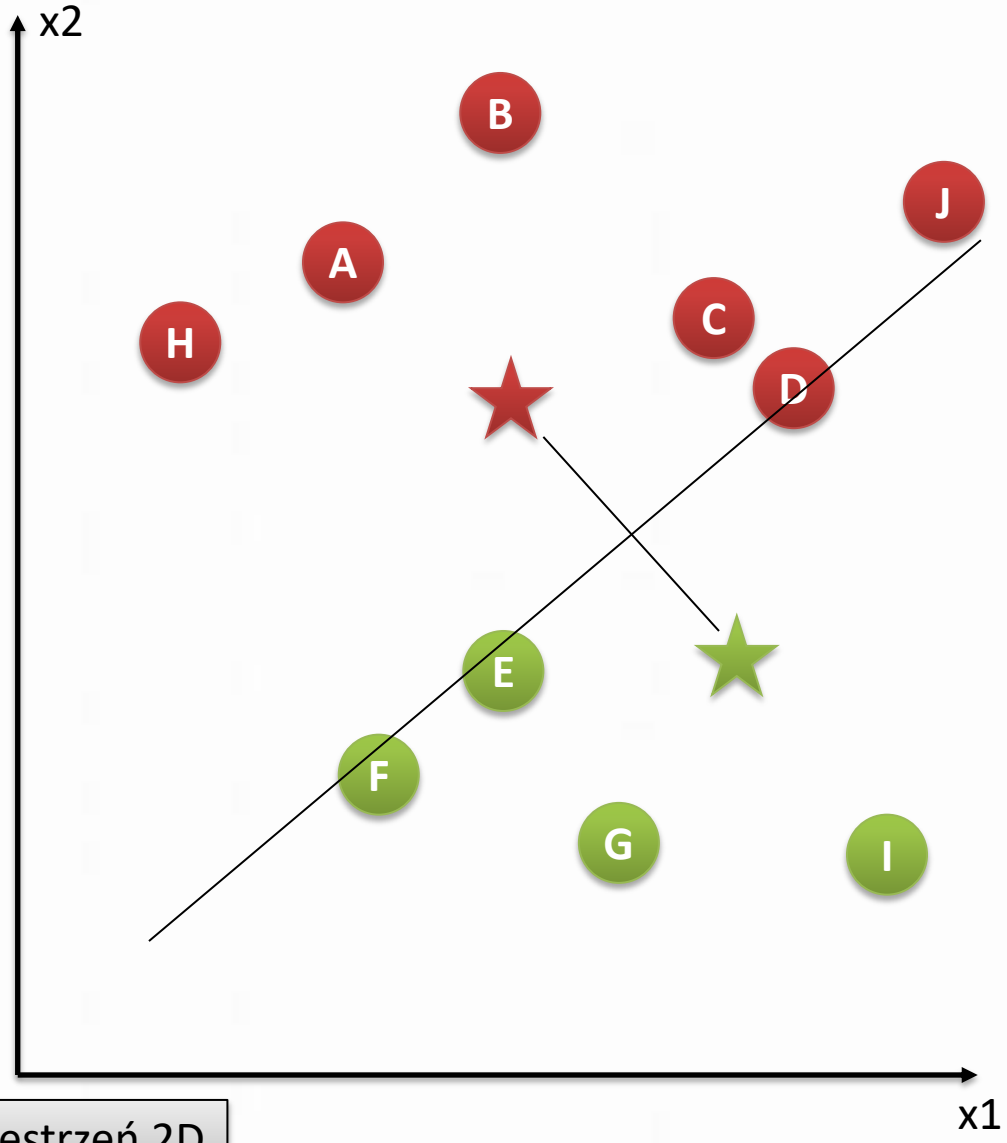
nowe centroidy



przestrzeń 2D

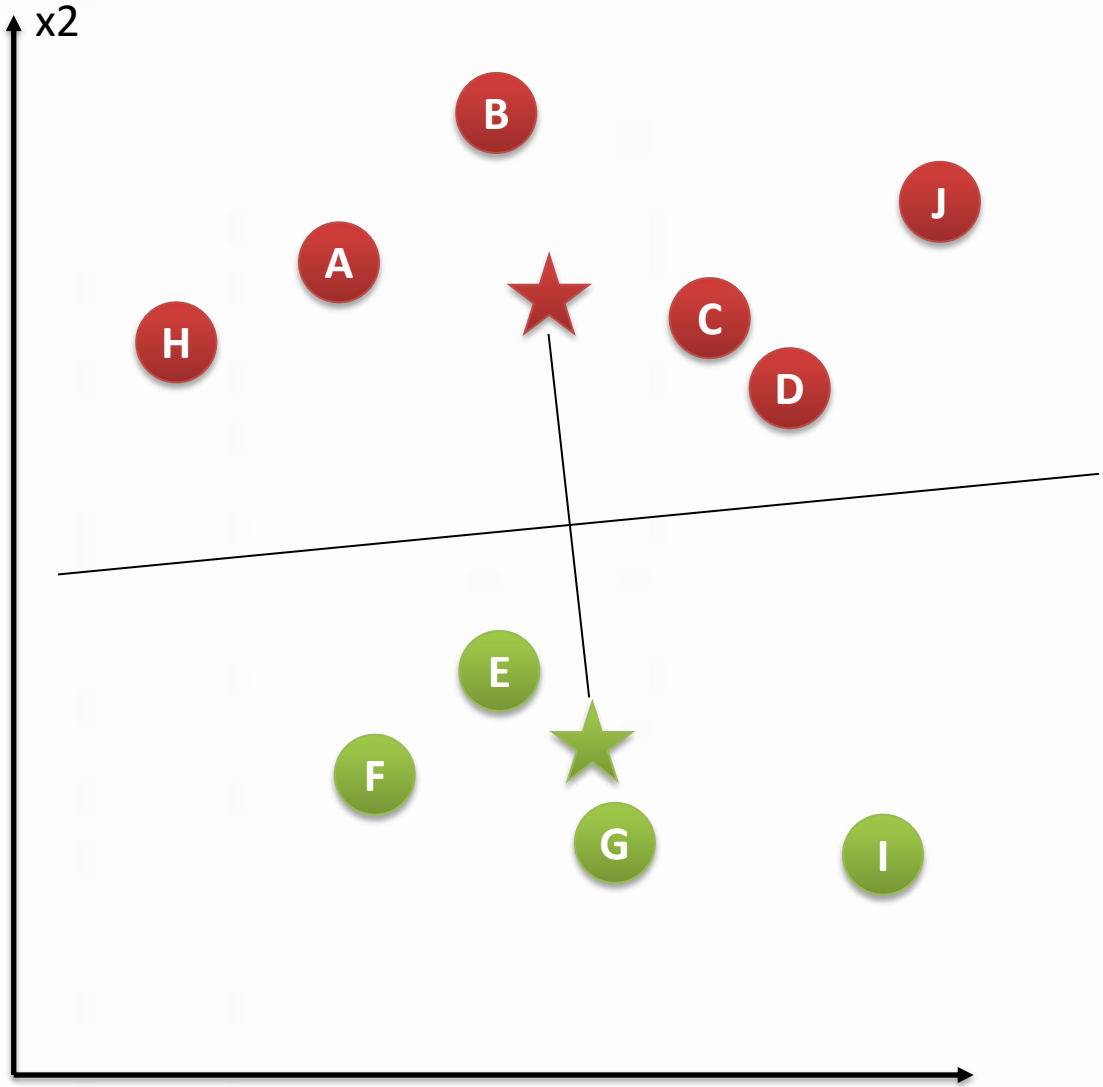
k-średnich

wyznacz odległości
od centroidów
do obiektów i przydziel
nowe etykiety grup



przestrzeń 2D

k-średnich



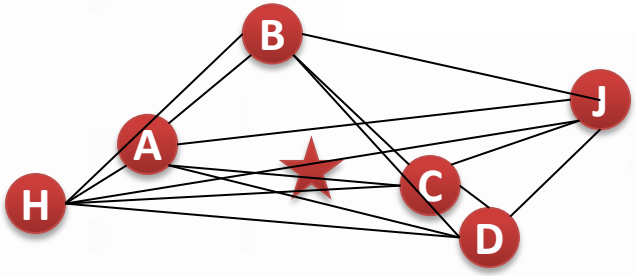
przestrzeń 2D

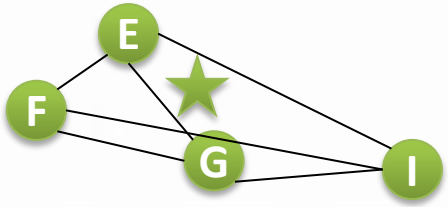
dla nowych centroidów wszystkie etykiety pozostają takie jak poprzednio - zbieżność została osiągnięta

k-średnich

Dla obiektów ($\mathbf{x}_1$, $\mathbf{x}_2$, $\cdots$, $\mathbf{x}_n$) z przestrzeni o d wymiarach k-means grupuje n obiektów do k ($\leq n$) grup $\mathbf{S} = \{S_1, S_2, \cdots, S_k\}$ by zminimalizować sumę kwaratów odległości wewnątrz grup (wariancję). Formalnie:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$





Algorytm k-średnich (k-means)

```
#grupowanie algorytmem k-means
>irisCluster <- kmeans(mtcars, centers
= 3, nstart = 20)
>irisCluster
```

Selekcja cech

Selekcja cech

Powody dla których dokonuje się selekcji cech:

1. Pozwala na szybsze uczenie się.
2. Podnosi jakość klasyfikacji/predykcji w przypadku odpowiedniego podzbioru cech.
3. Zmniejsza nadmierne dopasowanie (overfitting).
4. Redukuje złożoność modelu i pozwala na łatwiejszą interpretację.
5. W danych bioinformatycznych przeważnie mamy mało obiektów (pacjenci) i bardzo dużo zmiennych.

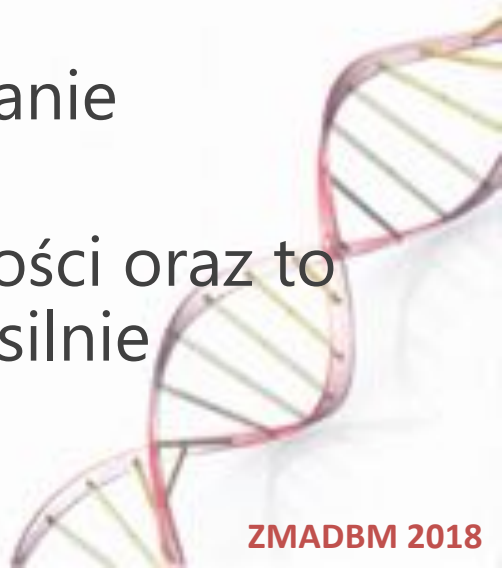
Selekcja cech

Grupy metod:

- Filtry
- Wrappery
- Metody wbudowane/specjalizowane

Filtry

- **Ocena pojedynczej** cechy vs zmienna decyzyjna.
- Oceniamy każdą cechę niezależnie.
- Popularne miary oceny:
  - Korelacja Pearsona/Spearmana
  - F-score
  - Information Gain
  - Gain Ratio
  - Chi^2
- Zaletami są szybkość działania, oraz uzyskanie rankingu cech.
- Dużą wadą jest brak uwzględnienia zależności oraz to iż w końcowym rankingu mogą być cechy silnie skorelowane ze sobą.



Filtry

- Oceniamy każdą cechę niezależnie.

outlook	temperature	humidity	windy	play
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rainy	71	91	TRUE	no

outlook vs **play**

$f(outlook,play) = I_{outlook}$

outlook	temperature	humidity	windy	play
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rainy	71	91	TRUE	no

temperature vs **play**

$f(temperature, play) = I_{temperature}$

Filtry

- Oceniamy każdą cechę niezależnie.

outlook	temperature	humidity	windy	play
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rainy	71	91	TRUE	no

humidity vs **play**

$$f(humidity,play) = I_{humidity}$$

outlook	temperature	humidity	windy	play
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rainy	71	91	TRUE	no

windy vs **play**

$$f(windy,play) = I_{windy}$$

Filtry - Ranking

- Na podstawie poszczególnych wartości I budujemy ranking cech. Od najistotniejszej do najmniej istotnej.

$I_{outlook} > I_{windy} > I_{humidity} > I_{temperature}$

1 outlook
2 windy
3 humidity
4 temperature

Wrappery

- **Ocena zestawu** cech vs zmienna decyzyjna.
- Oceniamy w oparciu o pewien klasyfikator i jego jakość predykcji na zbiorze testowym.
- Popularne strategie:
  - Forward selection (oryginalnie dla LM)
  - Backward elimination (oryginalnie dla LM)
  - Genetyczne przeszukiwanie
- Dużą zaletą jest uwzględnianie zależności między cechami.
- Wadami są: długi czas działania, silne dopasowanie do zbioru testowego, w zależności od alg. brak odporności na lokalne optima, raczej brak finalnego rankingu cech.

Forward selection

- 1 krok - oceniamy każdą cechę niezależnie.

outlook	temperature	humidity	windy	play
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rainy	71	91	TRUE	no

outlook vs **play**

Model(outlook, *play*) => $Acc_{outlook} = 0.71$

outlook	temperature	humidity	windy	play
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rainy	71	91	TRUE	no

temperature vs **play**

Model(temp ,*play*) => $Acc_{temperature} = 0.57$

Forward selection

- 1 krok - oceniamy każdą cechę niezależnie.

outlook	temperature	humidity	windy	play
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rainy	71	91	TRUE	no

humidity vs **play**

Model(humidity,*play*) => $Acc_{humidity} = 0.57$

outlook	temperature	humidity	windy	play
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rainy	71	91	TRUE	no

windy vs **play**

Model(windy,*play*) => $Acc_{windy} = 0.64$

Forward selection

- 2 krok – do najlepszego modelu (outlook) dodajemy kolejną cechę

outlook	temperature	humidity	windy	play
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rainy	71	91	TRUE	no

outlook + temp vs **play**

Model(outlook, temp, *play*) => $Acc_{outlook,temp}$

outlook	temperature	humidity	windy	play
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rainy	71	91	TRUE	no

outlook + humidity vs **play**

Model(outlook, humidity ,*play*) => $Acc_{outlook,\ humidity}$

Forward selection

- 2 krok – do najlepszego modelu (outlook) dodajemy kolejną cechę

outlook	temperature	humidity	windy	play
sunny	85	8	FALSE	no
sunny	80	9	TRUE	no
overcast	83	8	FALSE	yes
rainy	70	9	FALSE	yes
rainy	68	8	FALSE	yes
rainy	65	7	TRUE	no
overcast	64	6	TRUE	yes
sunny	72	9	FALSE	no
sunny	69	7	FALSE	yes
rainy	75	8	FALSE	yes
sunny	75	7	TRUE	yes
overcast	72	9	TRUE	yes
overcast	81	7	FALSE	yes
rainy	71	9	TRUE	no

outlook + windy vs **play**

Model(outlook, windy, *play*) => $Acc_{outlook,windy}$

- Całość kontynuujemy do momentu braku poprawy klasyfikacji

Forward selection

```
#Stepwise Regression - Forward Selection jest troche bardziej skomplikowane w uruchomieniu
#najpierw budujemy minimalny i makxymalny model
min.model <- lm(mpg~1, data=mtcars)
max.model <- lm(mpg~., data=mtcars)
#wyjscie jest za to bardzo jasne w interpretacji
step(min.model, scope=list(lower=min.model, upper=max.model), direction="forward")
```

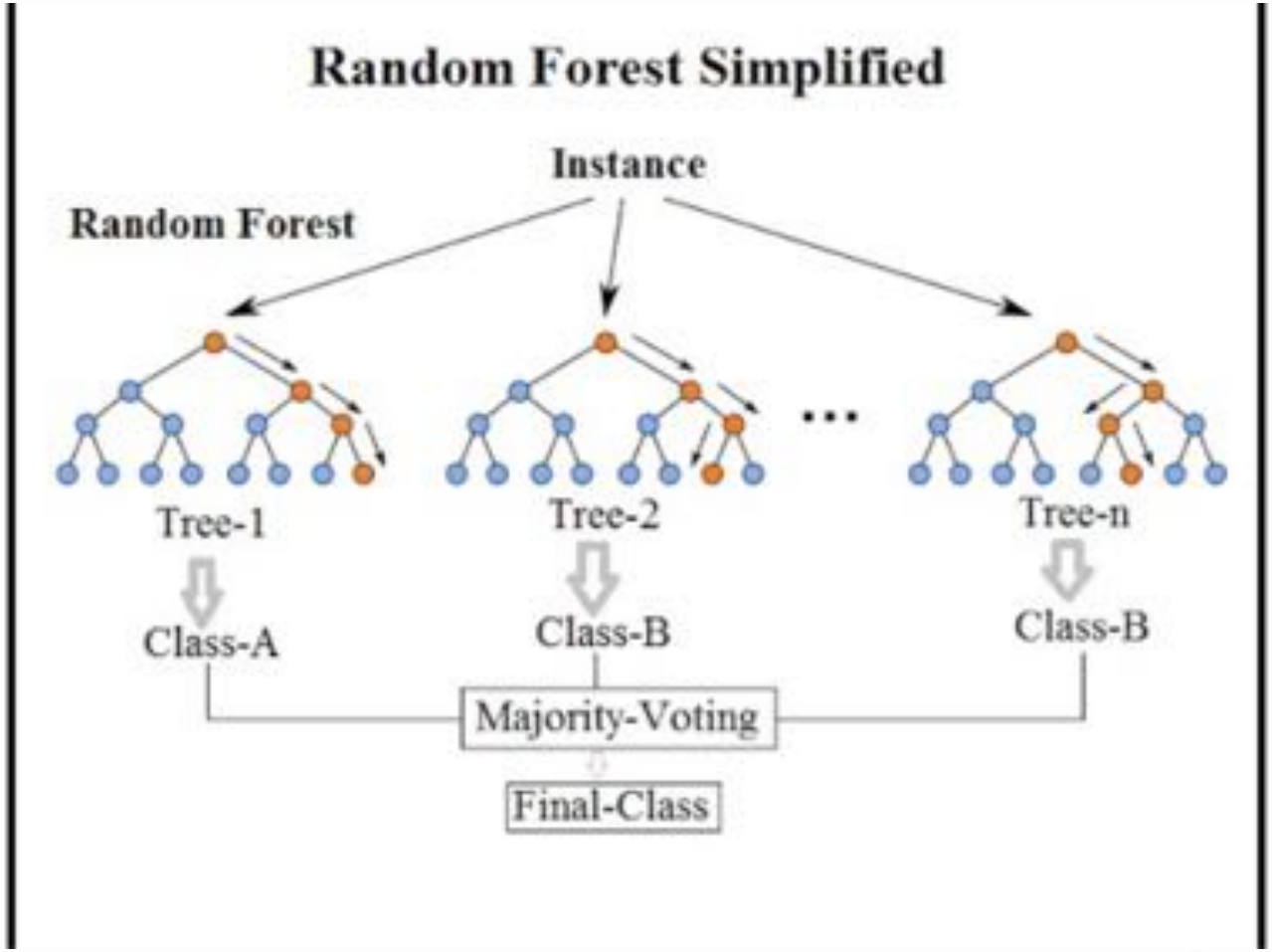
Metody wbudowane/specjalizowane

Popularne rekomendowane algorytmy:

- Las losowy
- Algorytm Boruta
- Algorytm MCFS-ID
- Lasso - Least Absolute Shrinkage and Selection Operator
- PCA – Principal Component Analysis

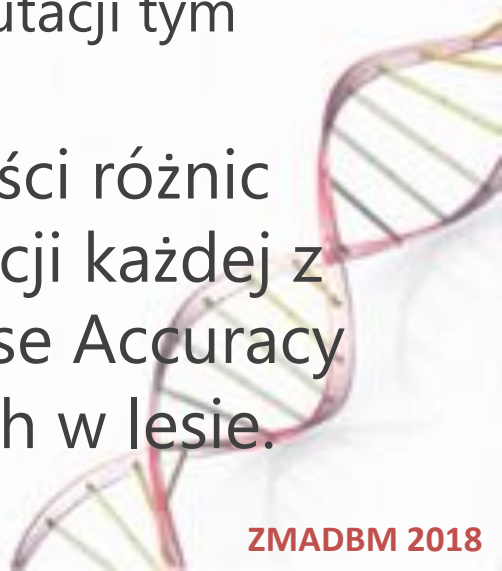
Największą zaletą jest specjalizacja powyższych algorytmów!

Las losowy (Random Forest)



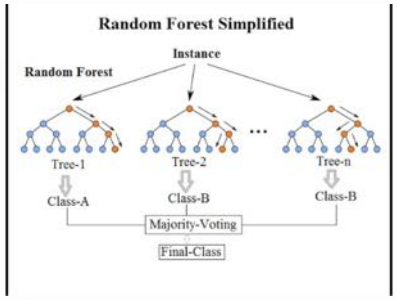
Las losowy – ranking cech

1. Zbuduj las losowy.
2. Policz błąd klasyfikacji dla danych wejściowych.
3. Dla każdej cechy:
   - dokonaj jej permutacji
   - policz błąd klasyfikacji danych z tak zmodyfikowaną zmienną
   - im większa zmiana błędu sprzed i po permutacji tym cecha jest istotniejsza!
4. Zbuduj ranking cech w oparciu o wartości różnic błędów klasyfikacji przed i po modyfikacji każdej z cech. W tym celu liczymy Mean Decrease Accuracy (MDA) – średnią po wszystkich drzewach w lesie.



Las losowy – ranking cech

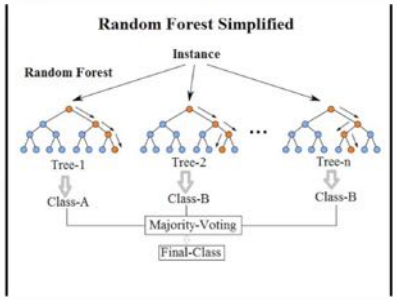
outlook	temperature	humidity	windy	play
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rainy	71	91	TRUE	no



MeanAcc

Permutacja zmiennej

outlook	temperature	humidity	windy	play
rainy	85	85	FALSE	no
overcast	80	90	TRUE	no
sunny	83	86	FALSE	yes
rainy	70	96	FALSE	yes
sunny	68	80	FALSE	yes
overcast	65	70	TRUE	no
rainy	64	65	TRUE	yes
sunny	72	95	FALSE	no
overcast	69	70	FALSE	yes
sunny	75	80	FALSE	yes
sunny	75	70	TRUE	yes
rainy	72	90	TRUE	yes
rainy	81	75	FALSE	yes
overcast	71	91	TRUE	no



$MeanAcc^*_{outloo}$

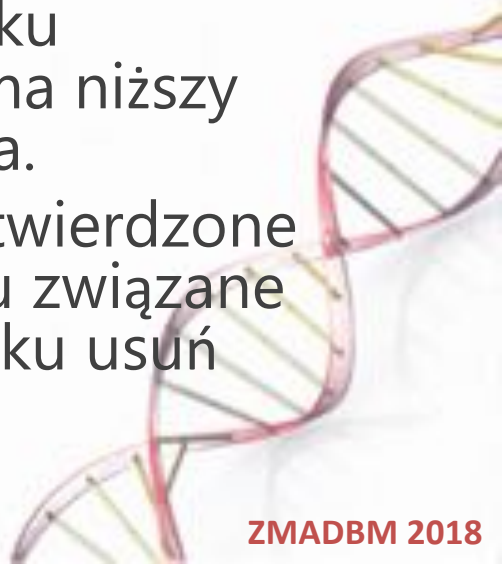
$MDA_{outlook} = MeanAcc - MeanAcc^*_{outlook}$

Las Losowy

```
### las losowy
library(randomForest)
# tu bedziemy szukac zmiennych pomagajacych w klasyfikacji cyl
my_mtcars <- mtcars
my_mtcars$cyl <- as.factor(my_mtcars$cyl)
classifier.rf <- randomForest(cyl~., data = my_mtcars, importance=T)
#zmienna importance lasu losowego przechowuje MeanDecreaseGini
#im wieksza wartość tym istotniejsza cecha
classifier.rf$importance
```

Algorytm Boruta

1. Dodaj do danych cechy kontrastowe (shadow features) które są permutacjami aktualnych cech wejściowych. Dla każdej cechy co najmniej 5 takich cech.
2. Zbuduj las losowy dla rozszerzonego zbioru wejściowego i użyj metody oceny cech z RF (domyślnie Mean Decrease Accuracy gdzie większa wartość oznacza wyższą istotność).
3. W każdej iteracjii sprawdź czy wyższą istotność ma cecha oryginalna od odpowiadających jej cech kontrastowych. Czy cecha ma wyższy Z-score (średnia dzielona przez odchylenie standardowe spadku accuracy) niż cechy kontrastowe. Jeśli cecha ma niższy Z-score to jest uznana za nieistotną i usuwana.
4. Zakończ jeśli wszystkie cechy zostały albo potwierdzone lub odrzucone lub osiągnięto kryterium stopu związane z zadaną liczbą iteracji. W przeciwnym wypadku usuń cechy kontrastowe i skocz do 1.



Algorytm Boruta

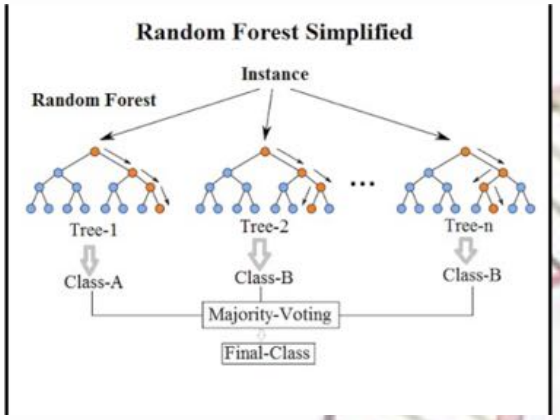
START

Dodaj cechy kontrastowe

outlook	temperature	humidity	windy	play
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rainy	71	91	TRUE	no

outlook	outlook1	outlook2	outlook3	outlook4	outlook5	temperature	temperature	temperature	temperature	temperature	humidity	humidity1	humidity2	humidity3	humidity4	humidity5	windy	windy1	windy2	windy3	windy4	windy5	play
sunny	overcast	rainy	sunny	rainy	sunny	85	65	64	72	72	85	90	90	86	90	85	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	no
sunny	rainy	overcast	rainy	rainy	rainy	80	75	83	75	64	90	70	86	90	96	70	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	no
overcast	overcast	rainy	rainy	overcast	rainy	83	85	85	75	65	86	85	80	95	95	90	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	yes
rainy	rainy	sunny	overcast	overcast	overcast	70	72	72	81	71	96	96	95	65	91	70	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	yes
rainy	sunny	overcast	sunny	overcast	overcast	68	83	69	83	70	80	75	65	70	86	91	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	yes
rainy	rainy	sunny	rainy	sunny	overcast	65	72	75	64	75	70	70	96	75	90	65	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	no
overcast	rainy	sunny	sunny	rainy	sunny	64	70	71	69	72	65	91	85	96	70	96	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	yes
sunny	sunny	overcast	rainy	rainy	rainy	72	64	68	70	80	95	95	80	80	85	75	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	no
sunny	sunny	rainy	sunny	overcast	rainy	69	68	65	80	75	70	70	70	91	80	86	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	yes
rainy	rainy	rainy	overcast	sunny	sunny	75	75	81	71	68	80	80	90	85	70	90	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	yes
sunny	sunny	sunny	overcast	sunny	sunny	75	81	72	65	83	70	65	75	80	65	80	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	yes
overcast	overcast	overcast	sunny	sunny	rainy	72	71	75	68	85	90	86	70	90	70	70	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	yes
overcast	overcast	rainy	rainy	sunny	overcast	81	80	70	85	81	75	90	70	70	80	80	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	yes
rainy	sunny	sunny	overcast	rainy	sunny	71	69	80	72	69	91	80	91	70	75	95	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	no

Buduj RF

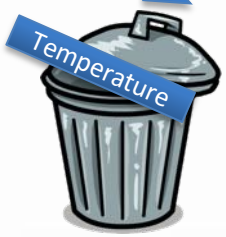


Z-score

1	outlook
2	windy
3	outlook1
4	humidity
5	temperature4
6	temperature1
7	temperature
8	windy5
9	windy4
10	outlook3
11	humidity2
12	windy2
13	temperature3
14	humidity5
15	humidity3
16	outlook2
17	temperature2
18	windy3
19	windy1
20	humidity4
21	humidity1
22	outlook4
23	outlook5

Pozostaw

Odrzuć



outlook	humidity	windy	play
sunny	85	FALSE	no
sunny	90	TRUE	no
overcast	86	FALSE	yes
rainy	96	FALSE	yes
rainy	80	FALSE	yes
rainy	70	TRUE	no
overcast	65	TRUE	yes
sunny	95	FALSE	no
sunny	70	FALSE	yes
rainy	80	FALSE	yes
sunny	70	TRUE	yes
overcast	90	TRUE	yes
overcast	75	FALSE	yes
rainy	91	TRUE	no

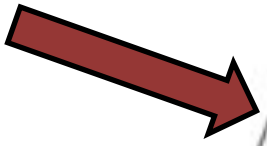
Nowy zbiór w
Kojejnej iteracji

Boruta

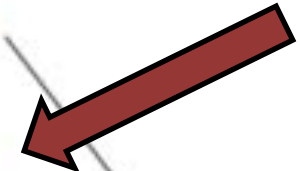
```
library(Boruta)
my_mtcars <- mtcars
my_mtcars$cyl <-
as.factor(my_mtcars$cyl)
boruta.result <- Boruta(cyl~., data =
my_mtcars)
#boruta potwierdza albo odrzuca
zmienne nie buduje rankingu
boruta.result$finalDecision
```

Algorytm MCFS

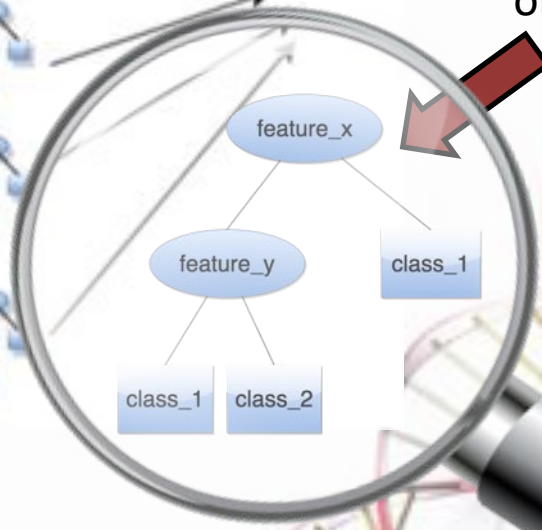
2. Random *st* splits into training and testing sets

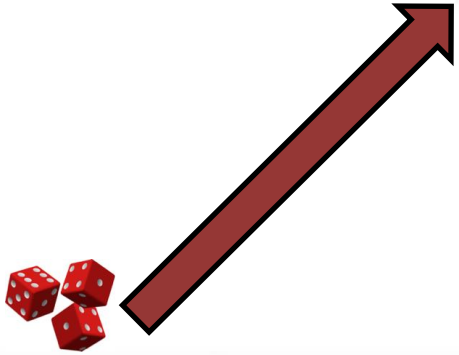


3. Creation and evaluation of decision trees (*dt*)



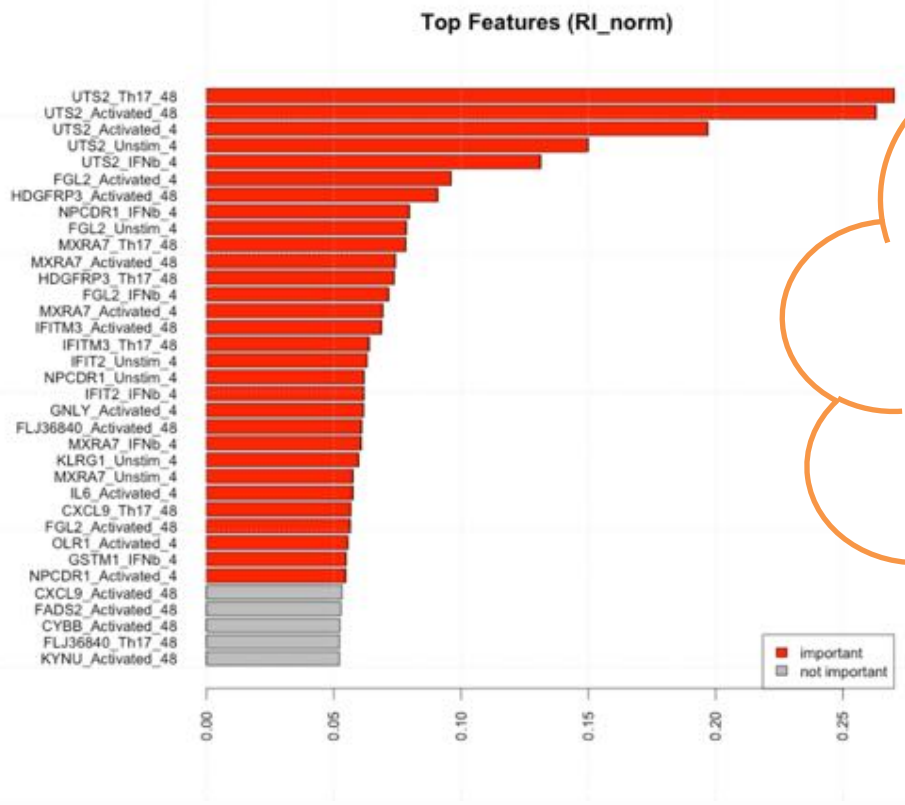
4. RI(f) is based on all *dt*





1. Random *s* selections of *m* features (*f*)

RI (Relative Importance) w MCFS



Istotne atrybuty:

1. Występują w wielu drzewach decyzyjnych (DT),
2. Występują blisko lub w korzeniu drzew (separują wiele obiektów),
3. Separują klasy wewnątrz węzłów z wysoką jakością,
4. DT oparte na nich dobrze klasyfikują nowe dane.

The relative importance of feature g_k, RI_{g_k}, is defined as

$$\mathrm{RI}_{g_k} = \sum_{\tau=1}^{s \cdot t} \mathrm{wAcc}_{\tau}^{u} \sum_{n_{g_k}(\tau)} \mathrm{IG}(n_{g_k}(\tau)) \left(\frac{\text{no. in } n_{g_k}(\tau)}{\text{no. in } \tau} \right)^{v},$$

1 3 4 2

MCFS-ID

```
### algorytm MCFS
library(rmcfs)
my_mtcars <- mtcars
my_mtcars$cyl <- as.factor(my_mtcars$cyl)
mcfs.result <- mcfs(cyl~., data = my_mtcars)
#zmienna RI wyniku zawiera df z rankingiem
#im wyższe RI_norm tym cecha bardziej
istotna
mcfs.result$RI
```

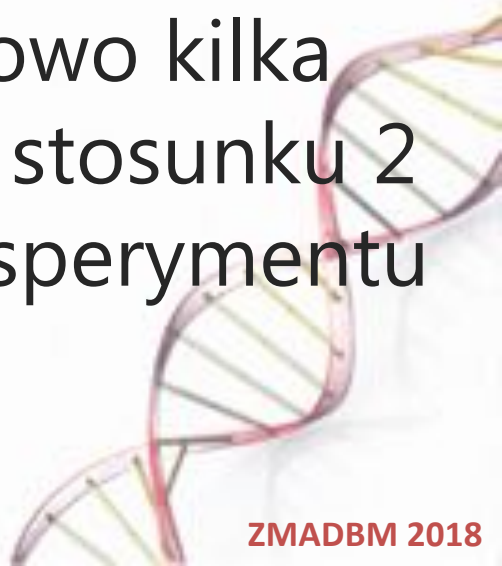
Ćwiczenia 8

michal.draminski@ipipan.waw.pl

Zdjęcia, schematy i rysunki zostały zaczerpnięte z internetu.

Ćwiczenia 8

Z1. Dla załączonych danych zbuduj ranking cech dowolną metodą. Następnie przedstaw na wykresie słupkowym zależność jakości klasyfikacji (accuracy) od wybranych cech top *n* vs losowo wybrane *n* dla dowolnie wybranego algorytmu klasyfikacji. Do oszacowania średniego accuracy (acc) podziel losowo kilka razy na zbiór trenujący i testujący (w stosunku 2 do 1). Opisz wnioski wynikające z eksperymentu i wykresu. **(5 pkt)**



Ćwiczenia 8

Z2. Dla załączonych danych usuń zmienną decyzyjną i następnie przeprowadź dwukrotnie grupowanie obiektów dowolną metodą raz dla losowo wybranych *n* zmiennych a za drugim razem dla najbardziej istotnych wybranych w Z1. Porównaj wyniki obu grupowań z oryginalną klasą decyzyjną i zapisz swoje wnioski. **(5 pkt)**