

UCZENIE MASZYNOWE

DRZEWA DECYZYJNE - KLASYFIKACJA

Dr hab. inż. Grzegorz Dudek
Wydział Elektryczny
Politechnika Częstochowska

Projekt finansowany w ramach programu Ministra Nauki i Szkolnictwa Wyższego pod nazwą „Regionalna Inicjatywa Doskonałości” w latach 2019 - 2022 nr projektu 020/RID/2018/19 kwota finansowania 12 000 000 PLN

Plan prezentacji

1. Informacje wstępne
2. Idea
3. Konstrukcja drzewa klasyfikacyjnego
4. Proces klasyfikacji za pomocą drzева
5. Testy w węzłach drzева
6. Przycinanie drzева
7. Drzewo klasyfikacyjne w Matlabie
8. Przykład zastosowania drzewa klasyfikacyjnego
9. Podsumowanie

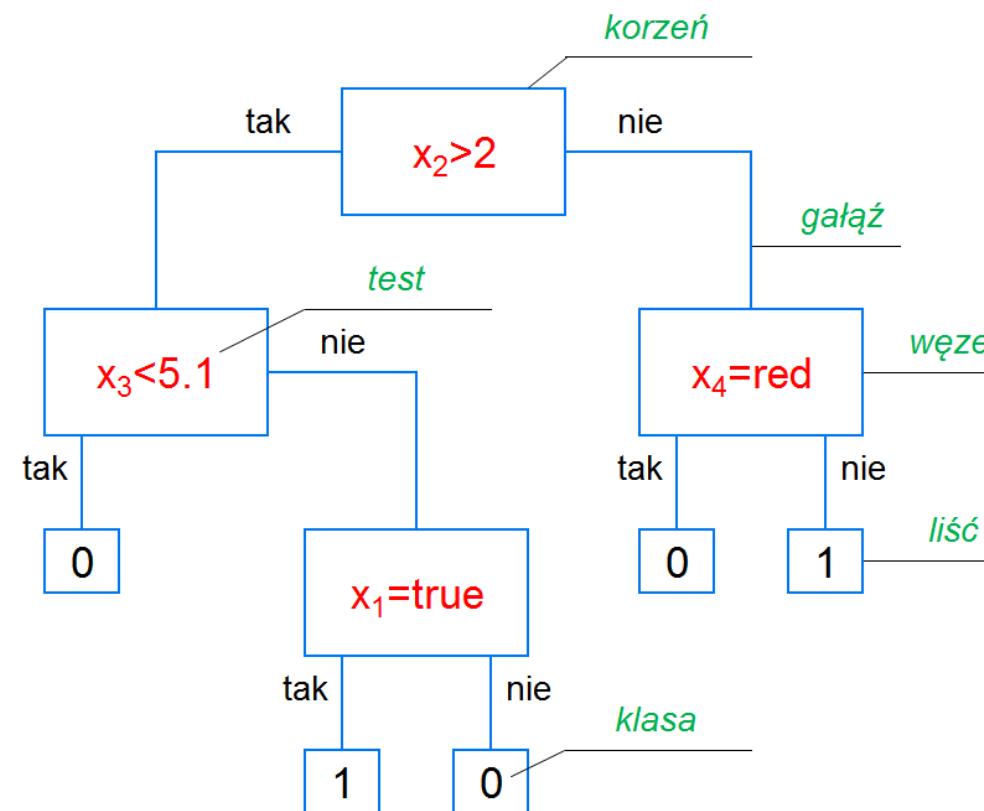
Informacje wstępne

Drzewa decyzyjne są metodą indukcyjnego uczenia się pojęć i reprezentacji hipotez.

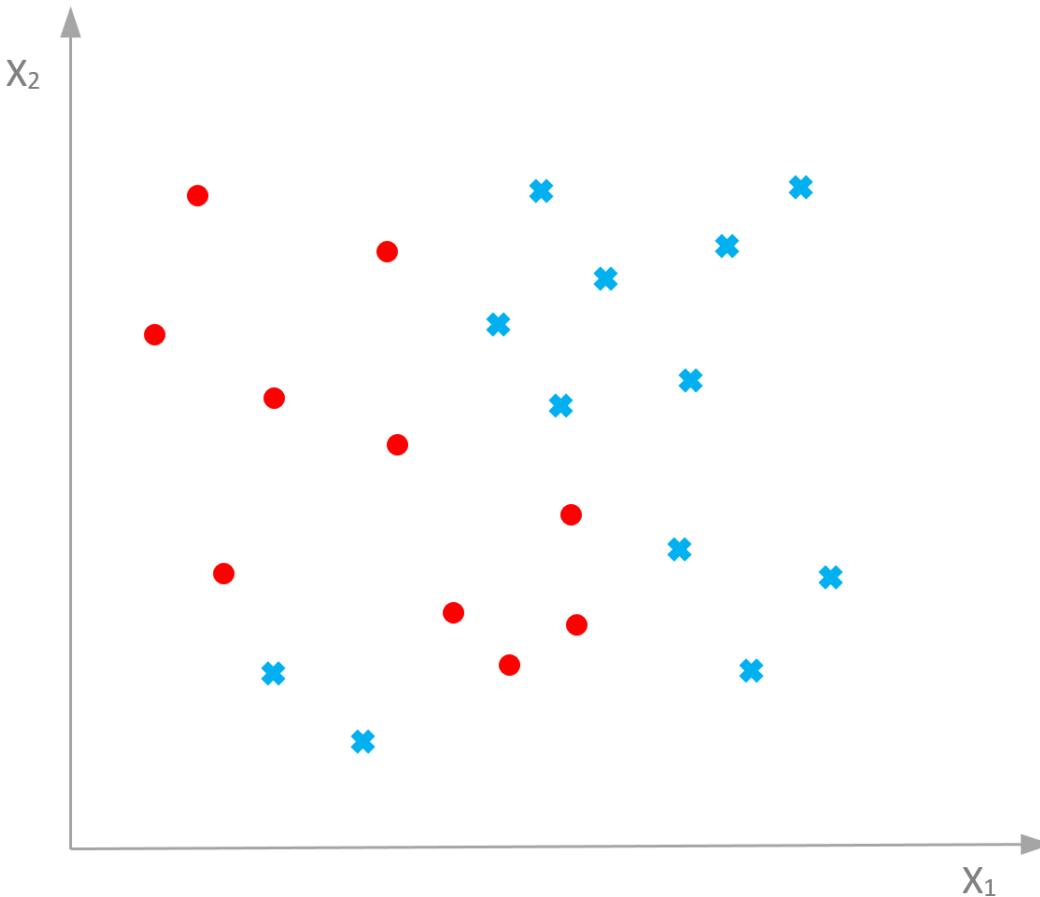
Drzewo decyzyjne jest strukturą złożoną z **węzłów, gałęzi i liści** (węzłów terminalnych).

Węzły odpowiadają **testom** przeprowadzanym na wartościach atrybutów, gałęzie odpowiadają wynikom tych testów, a liście – etykietom kategorii (klas).

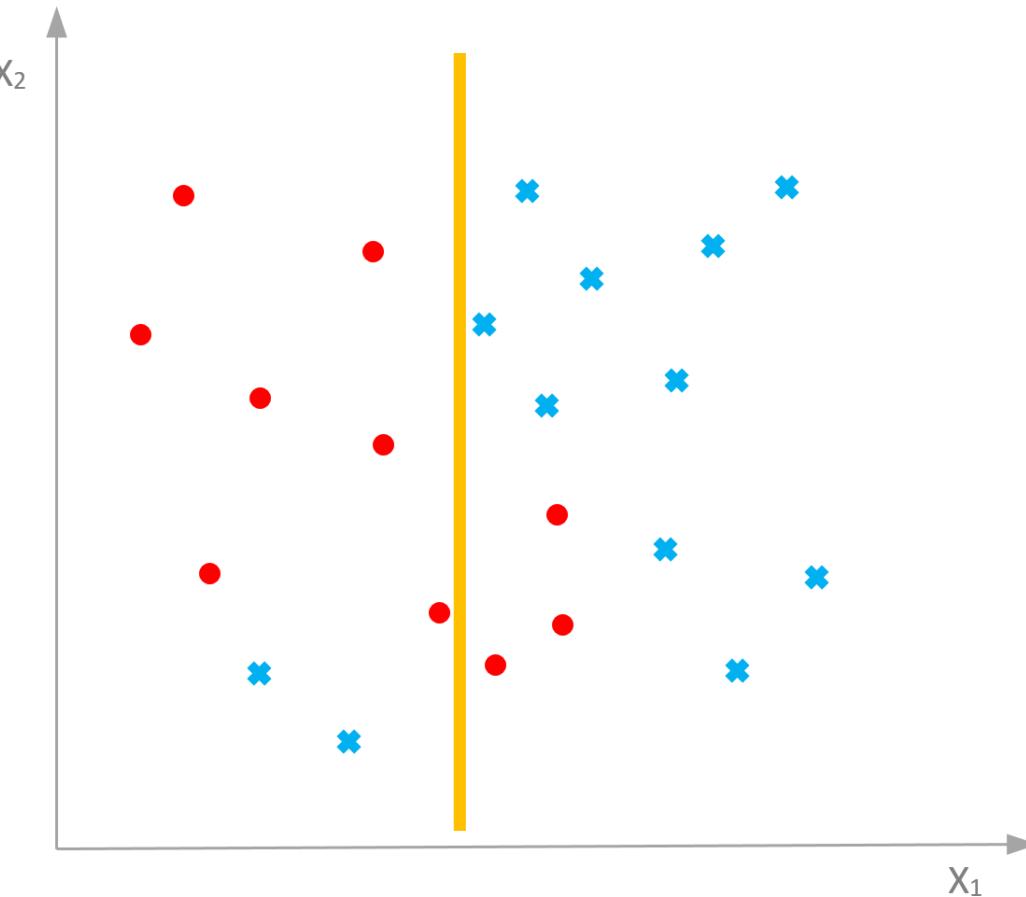
Przykładowe drzewo decyzyjne z testami binarnymi, dla dwóch kategorii i czterech atrybutów:



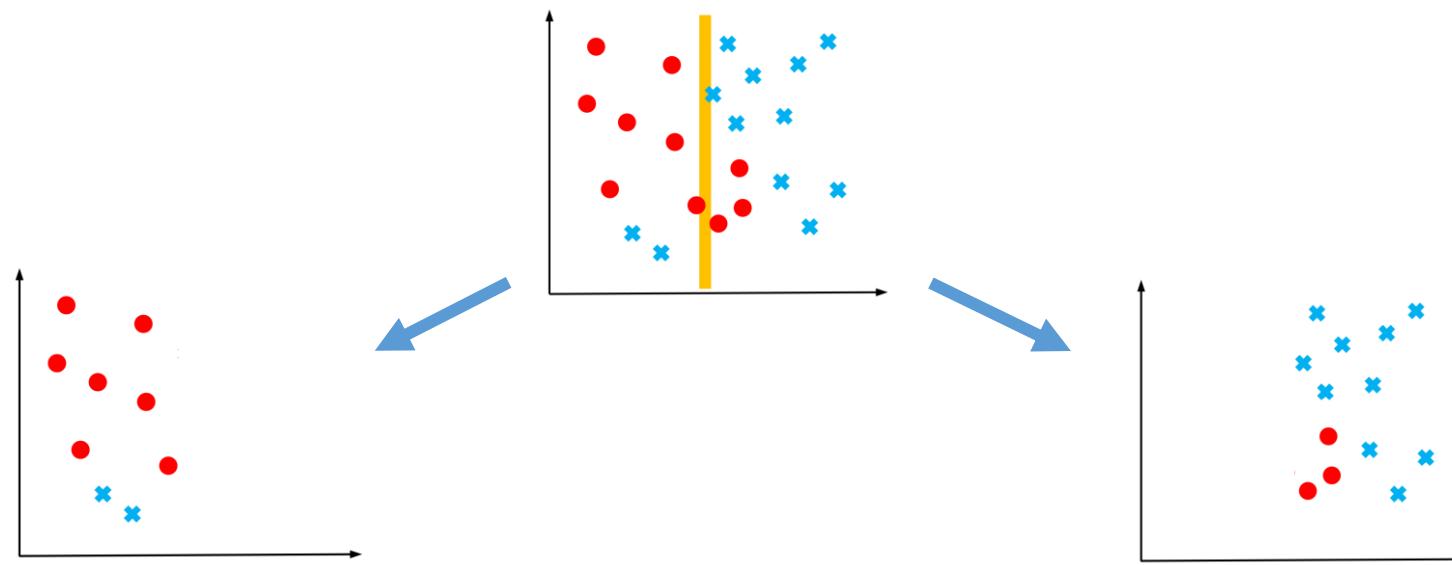
Idea



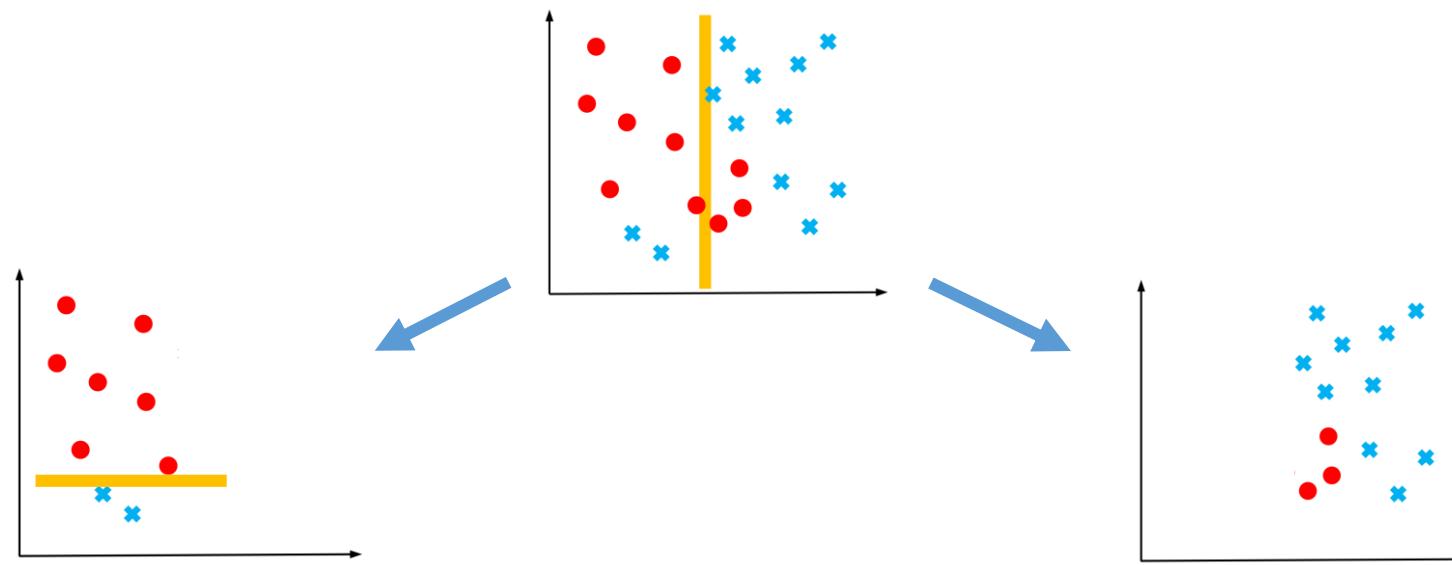
Idea



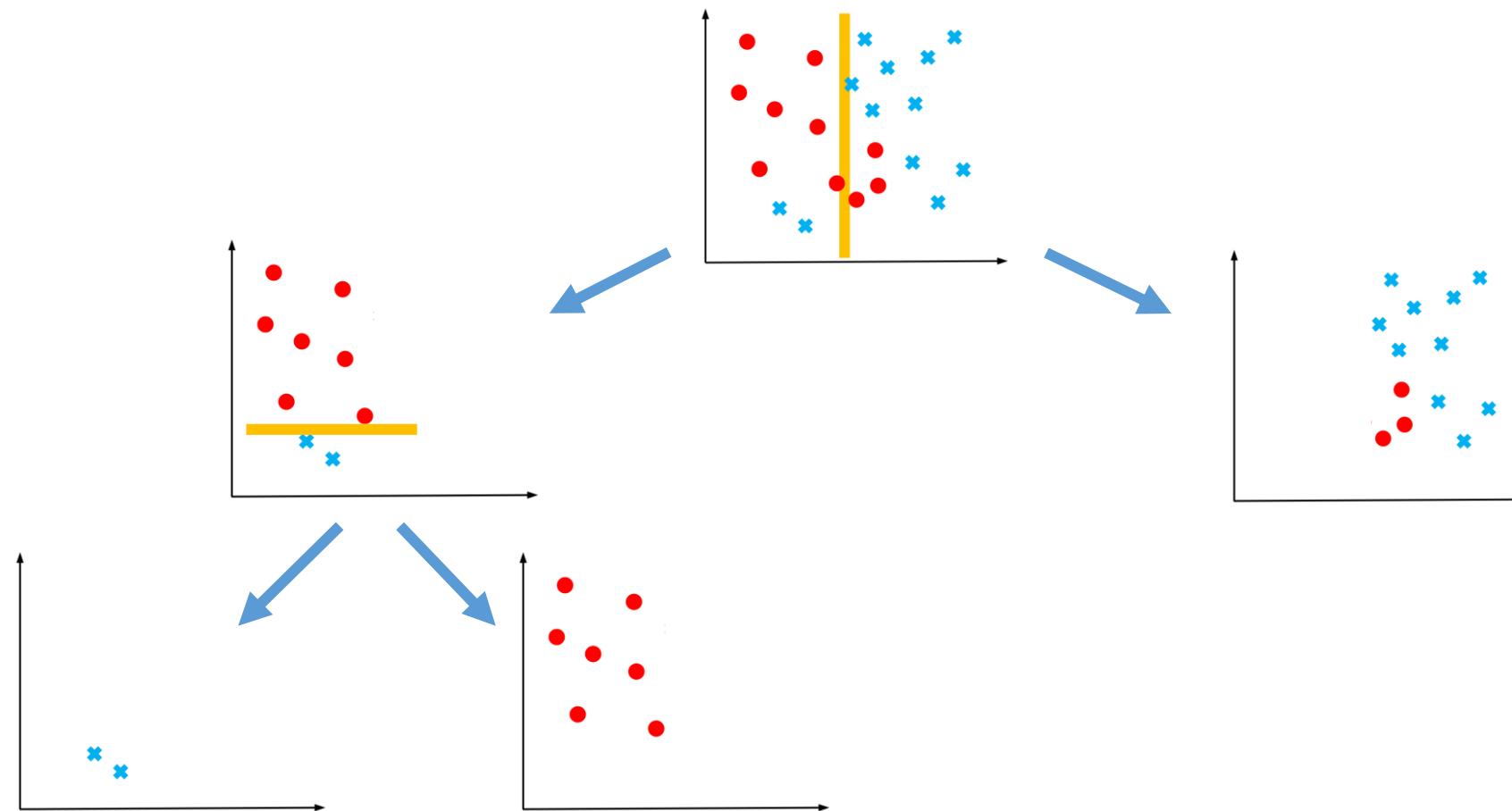
Idea



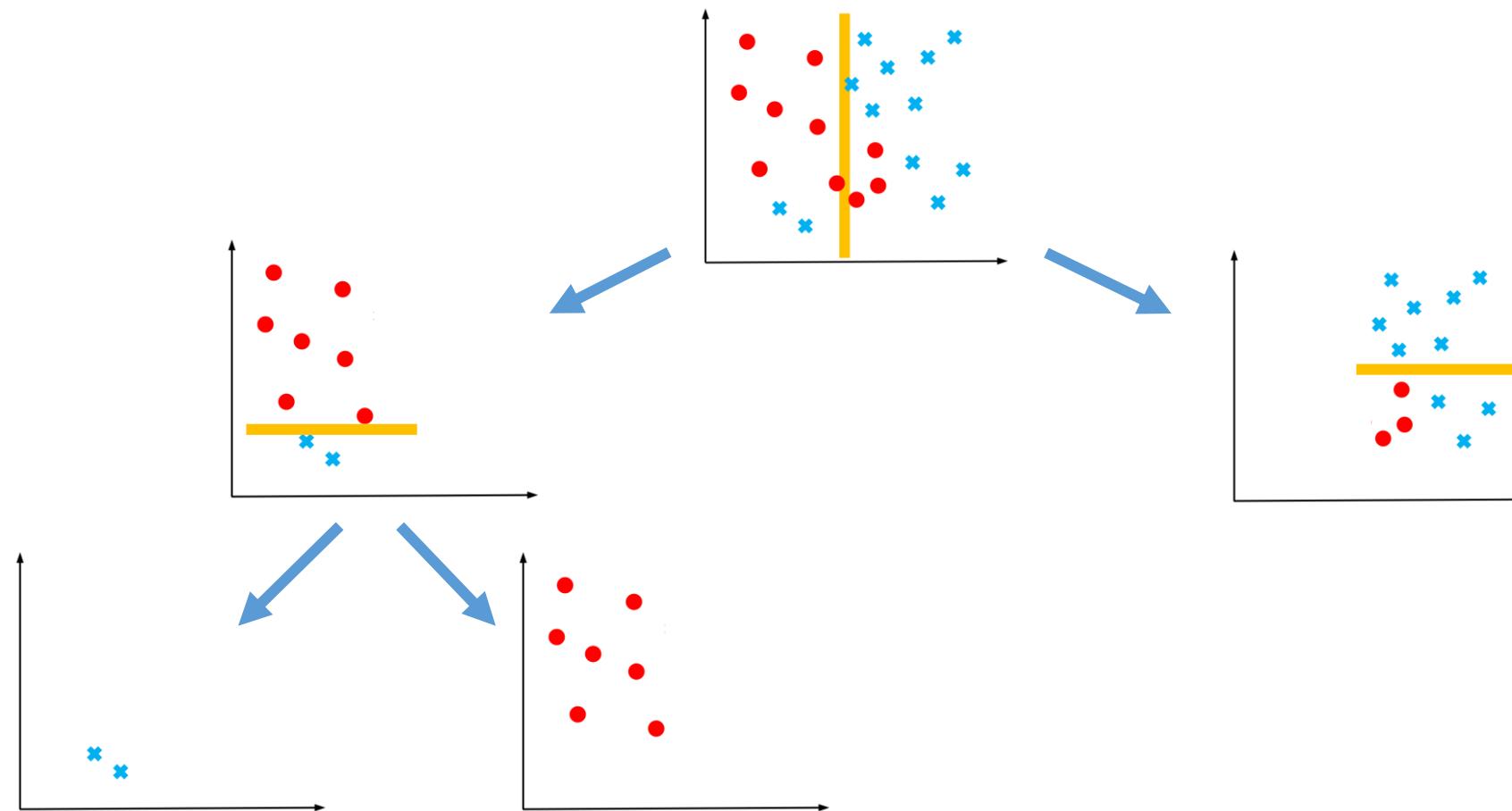
Idea



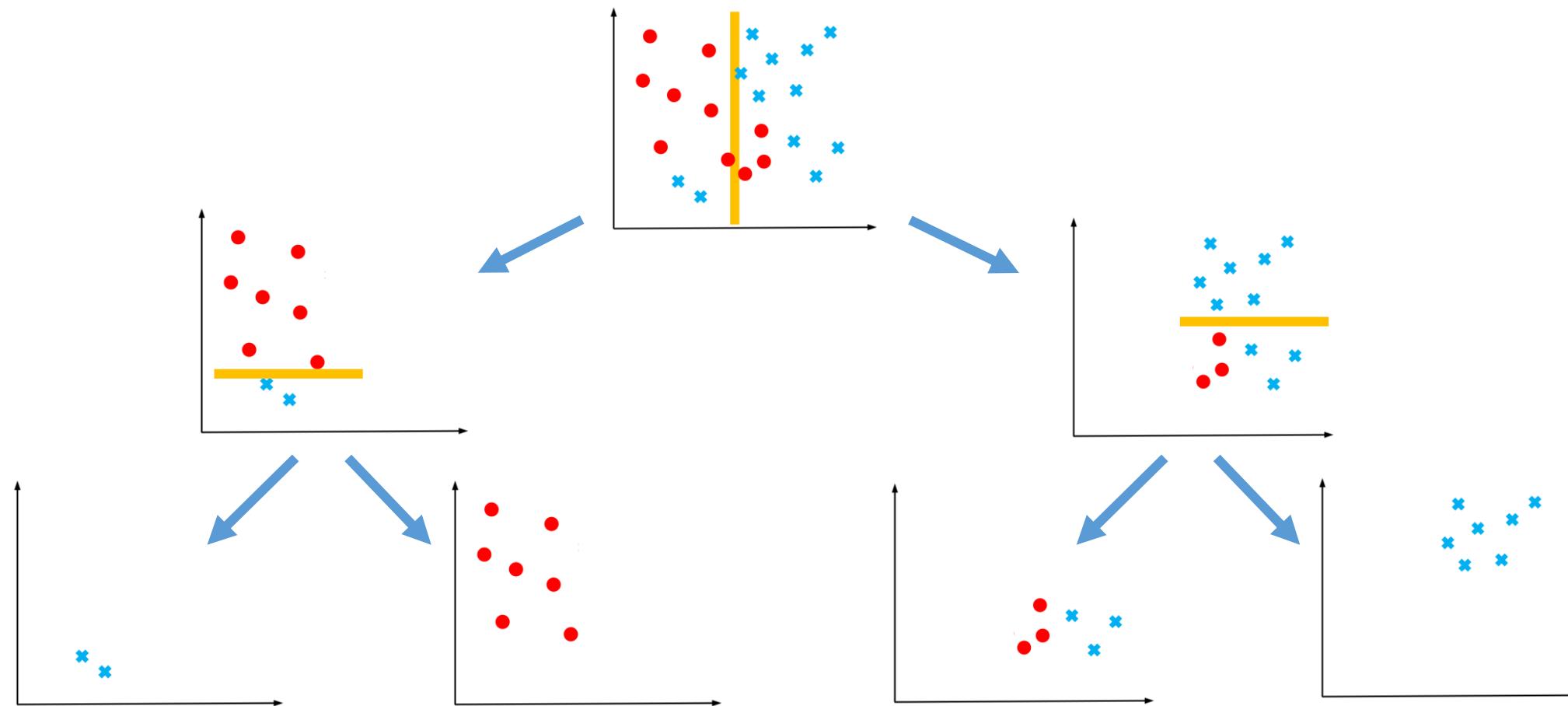
Idea



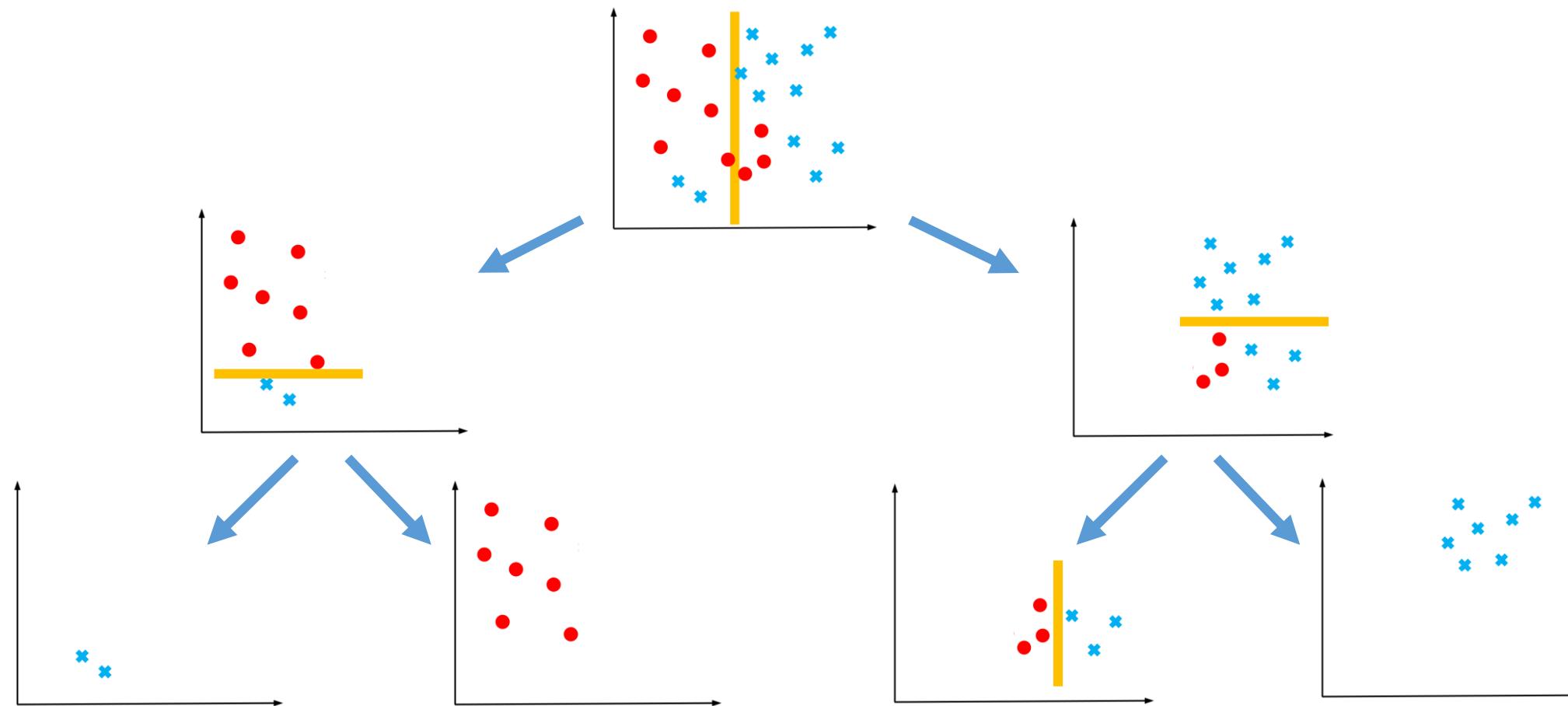
Idea



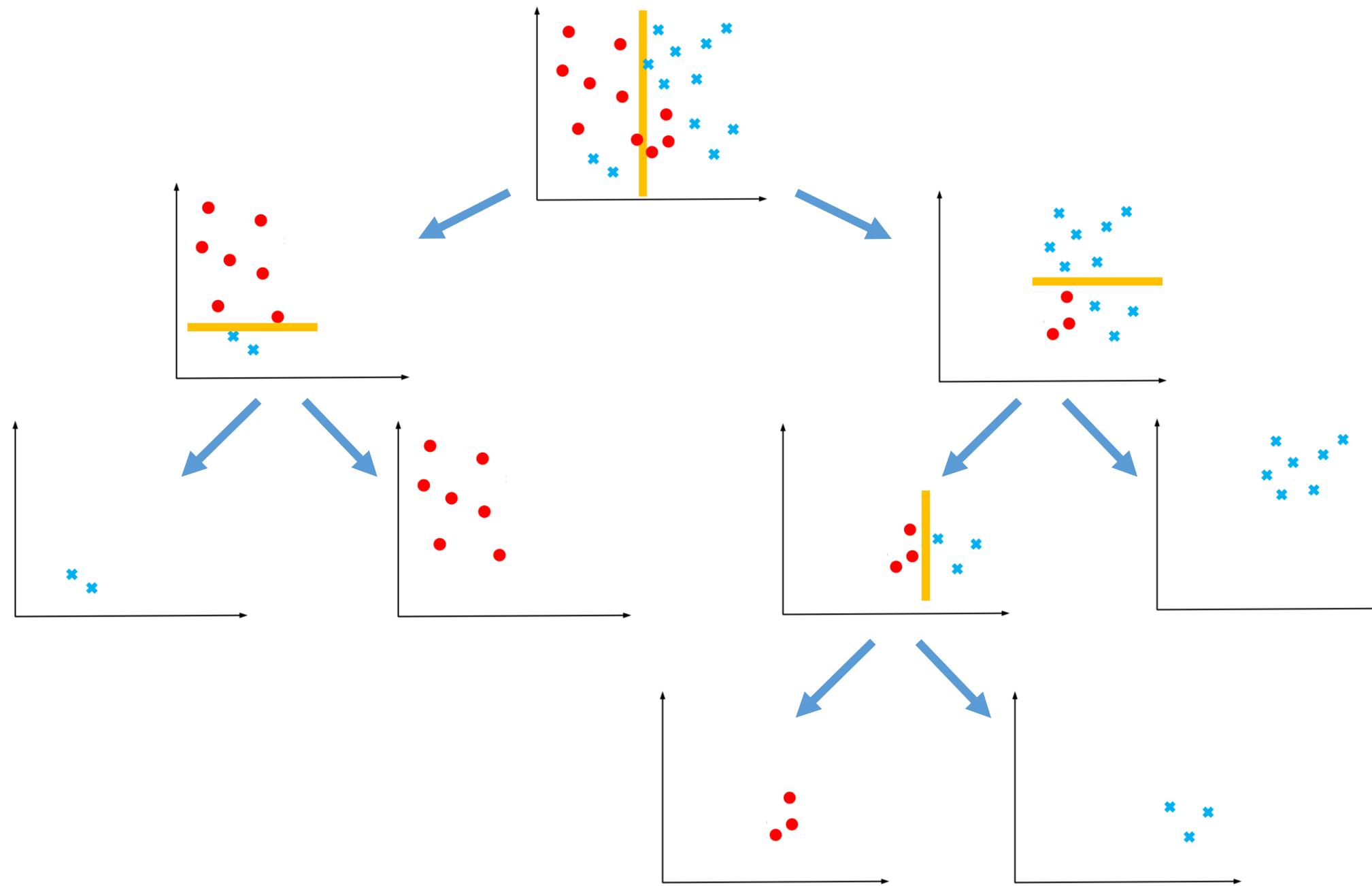
Idea



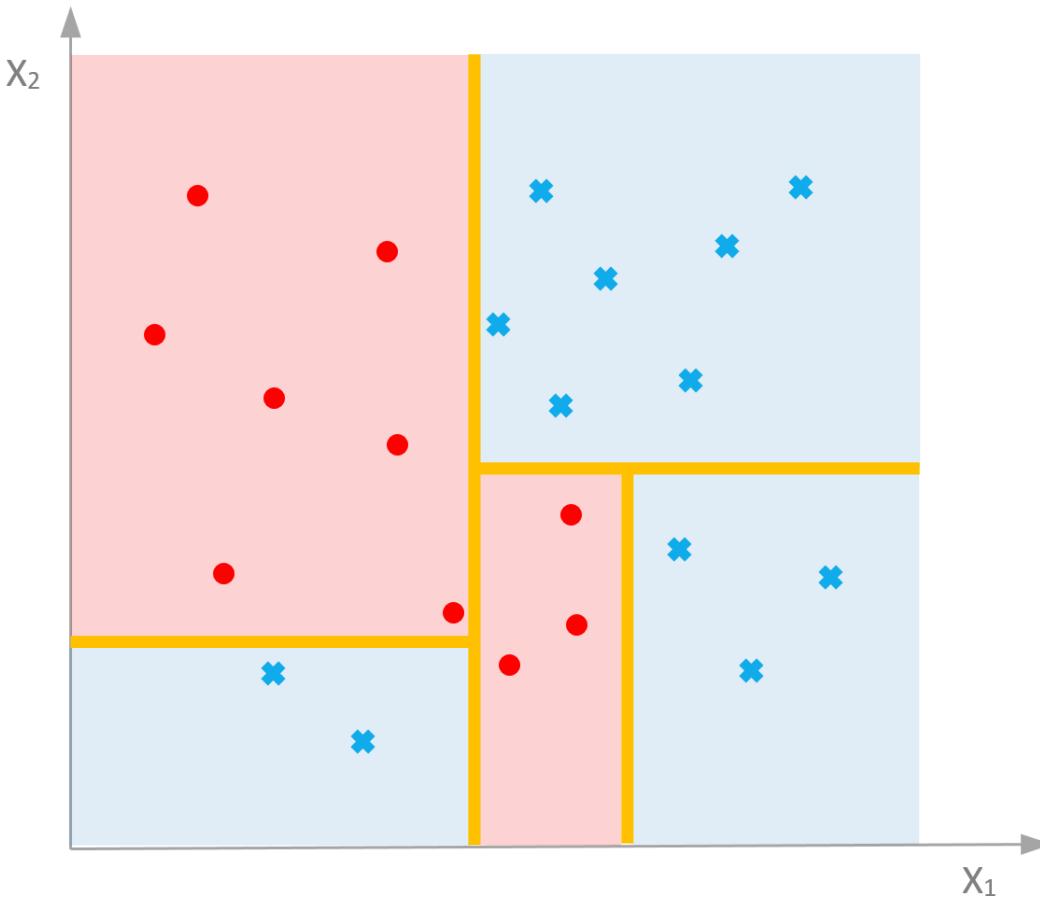
Idea



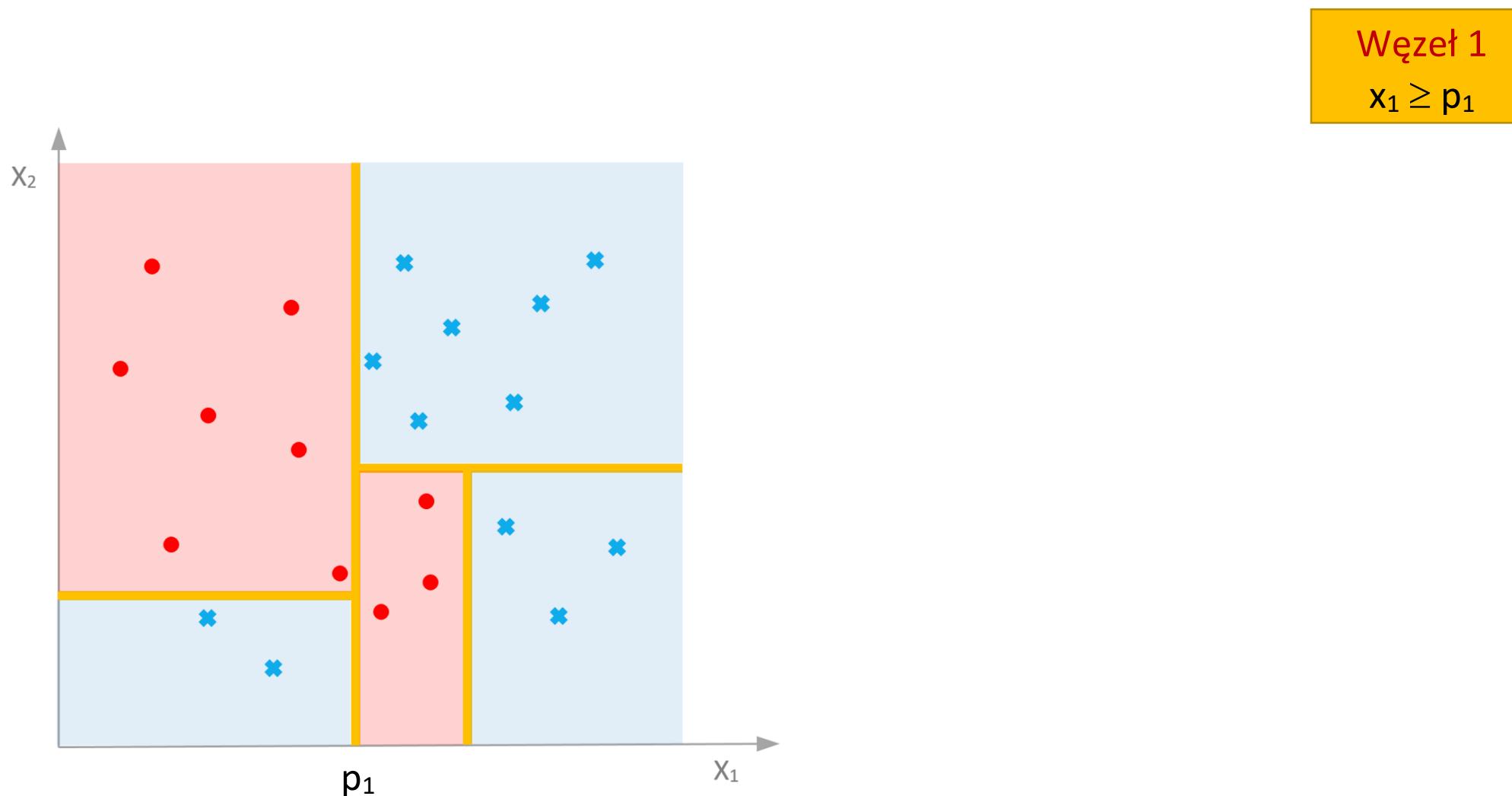
Idea



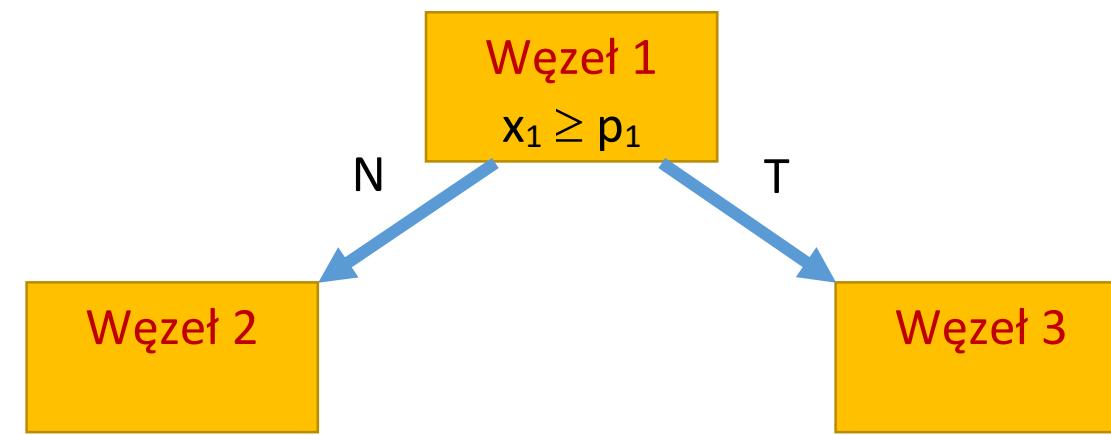
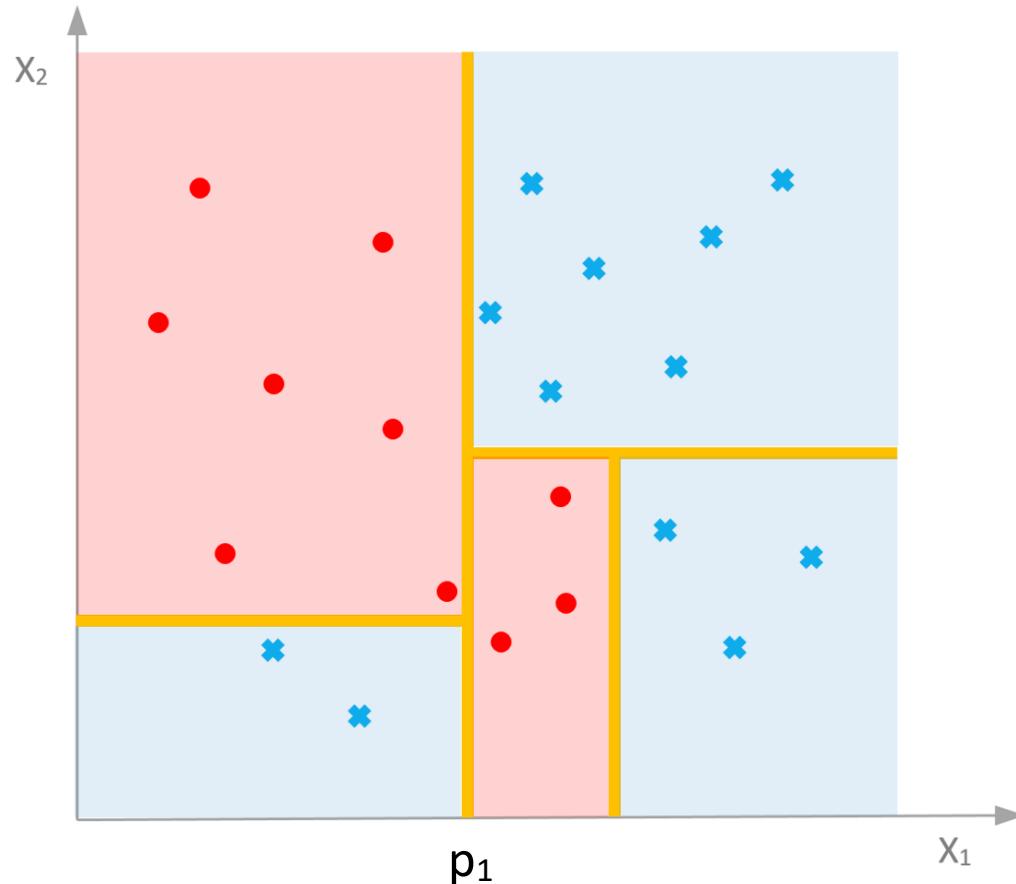
Idea



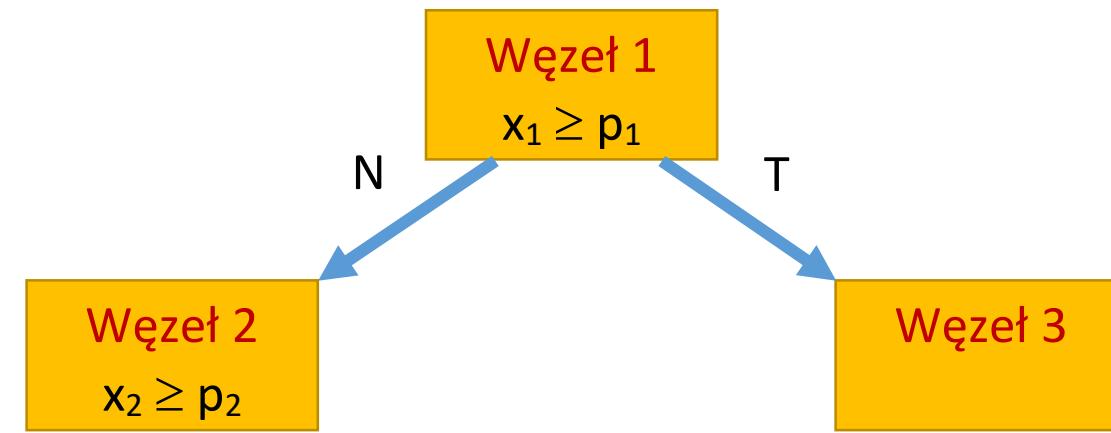
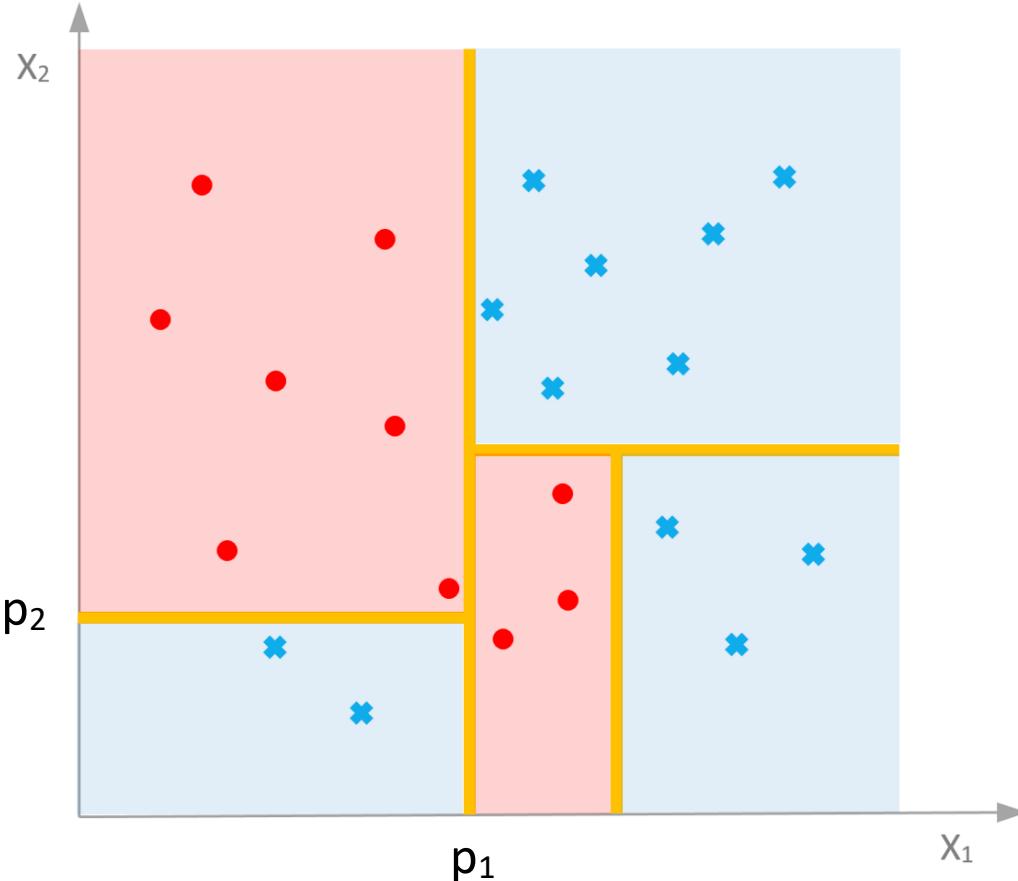
Konstrukcja drzewa



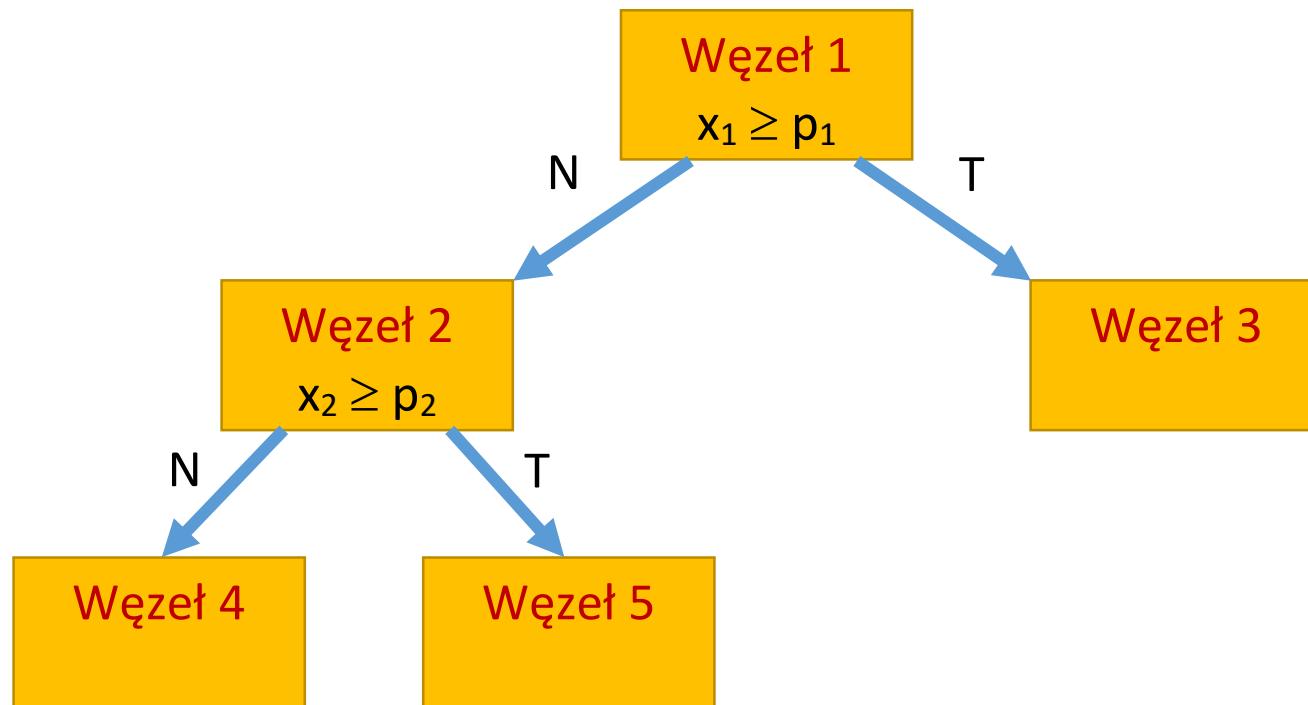
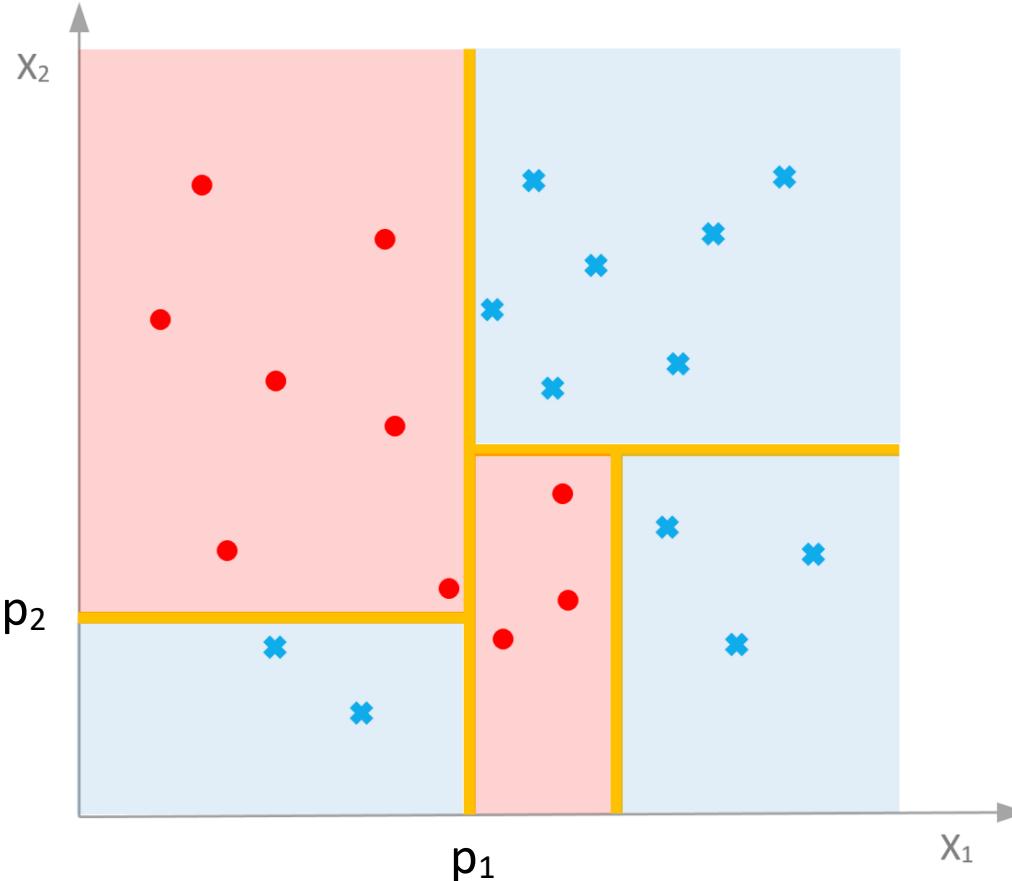
Konstrukcja drzewa



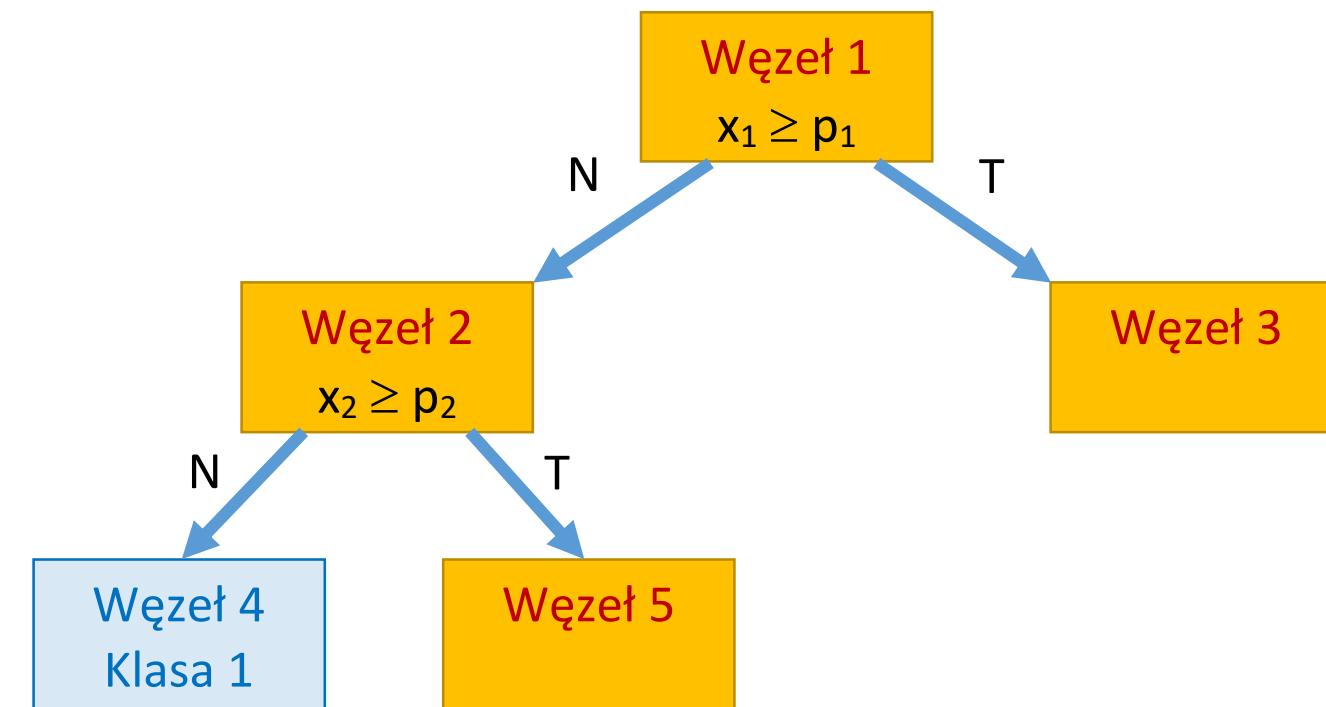
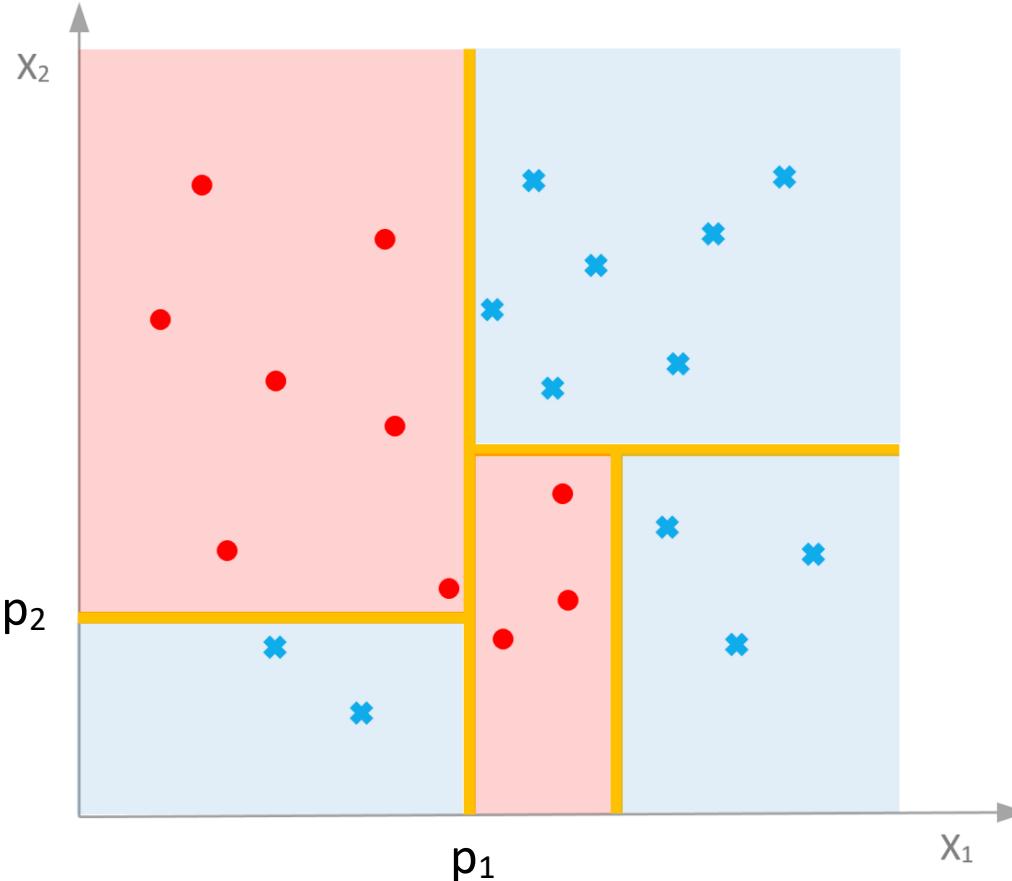
Konstrukcja drzewa



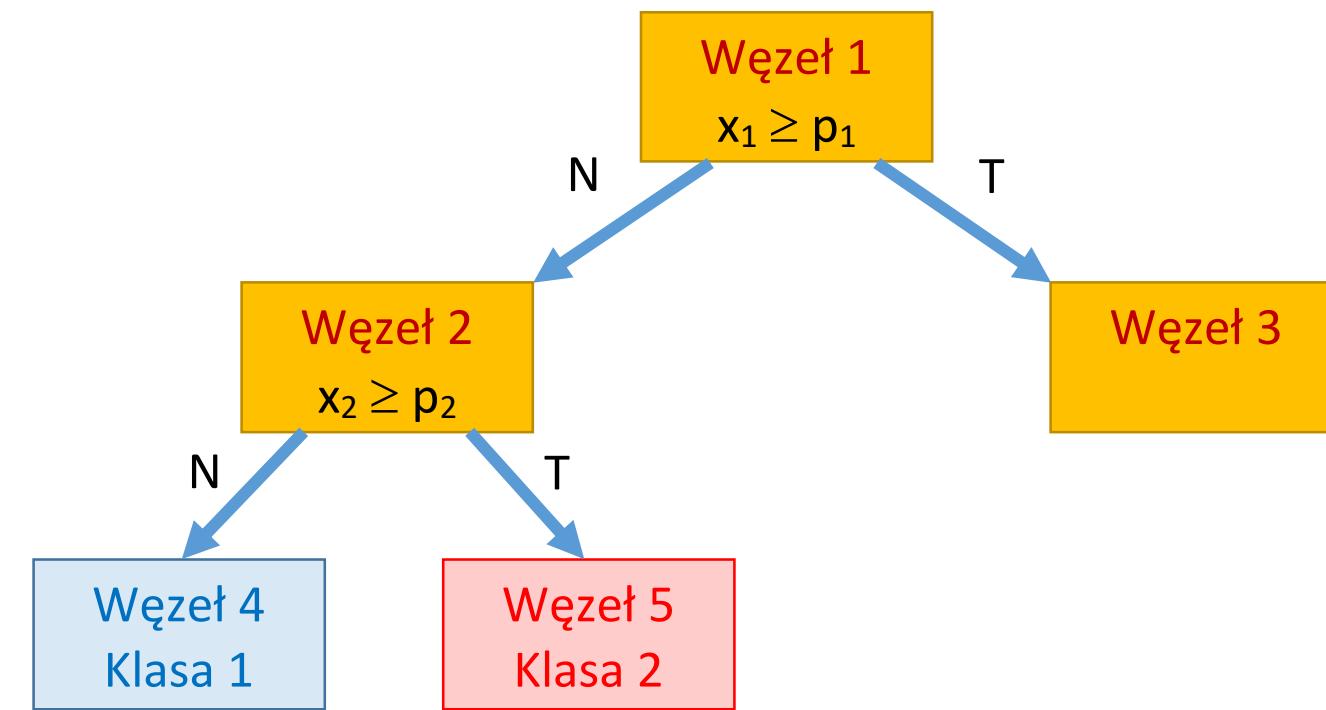
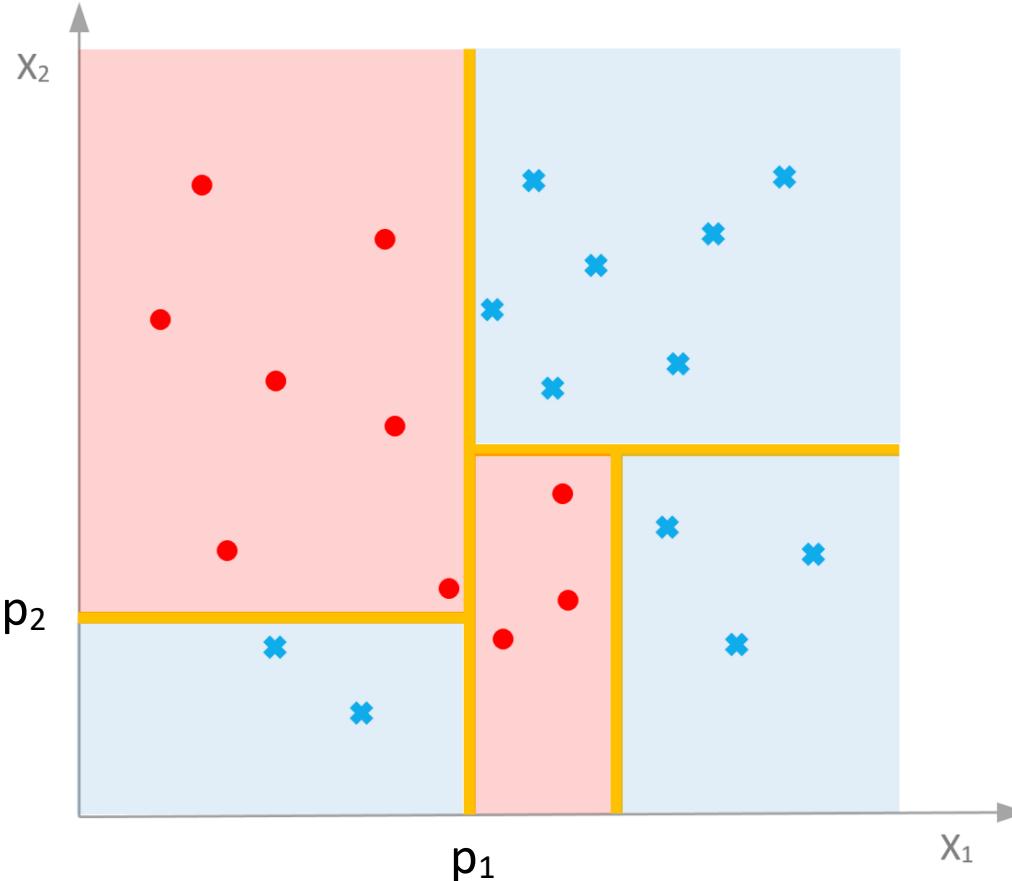
Konstrukcja drzewa



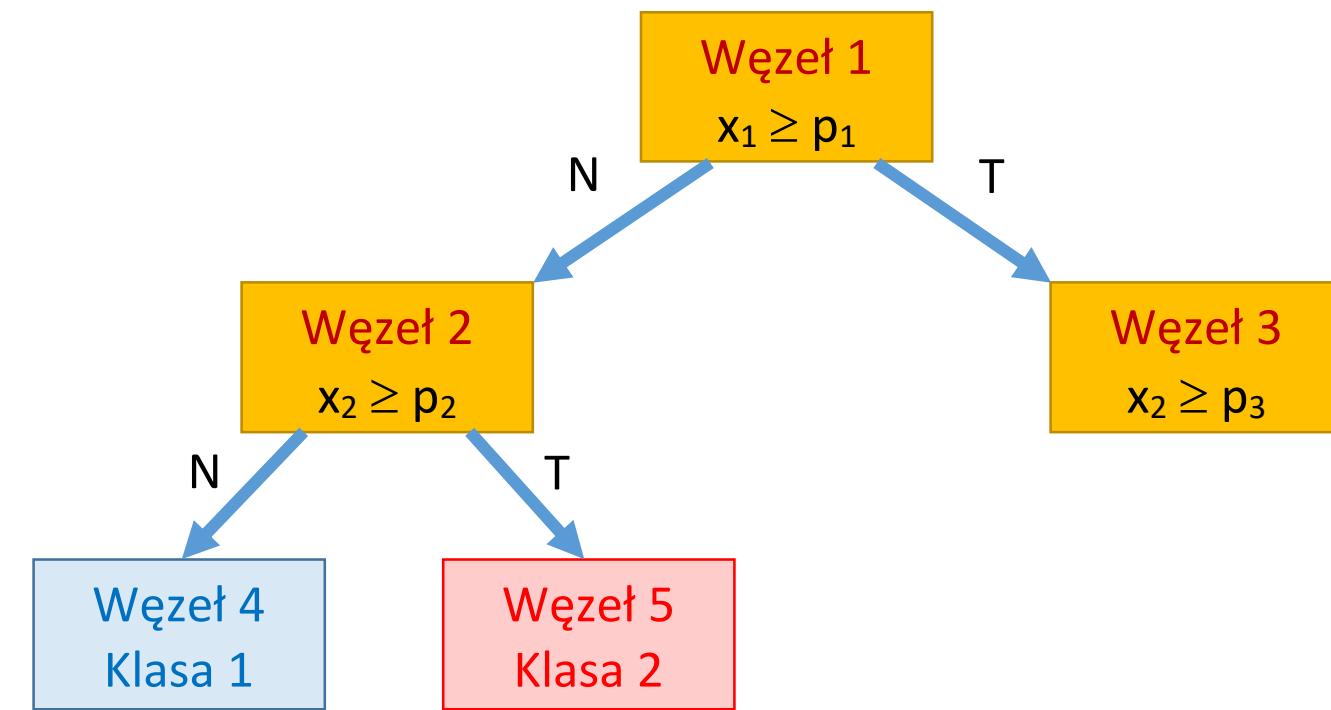
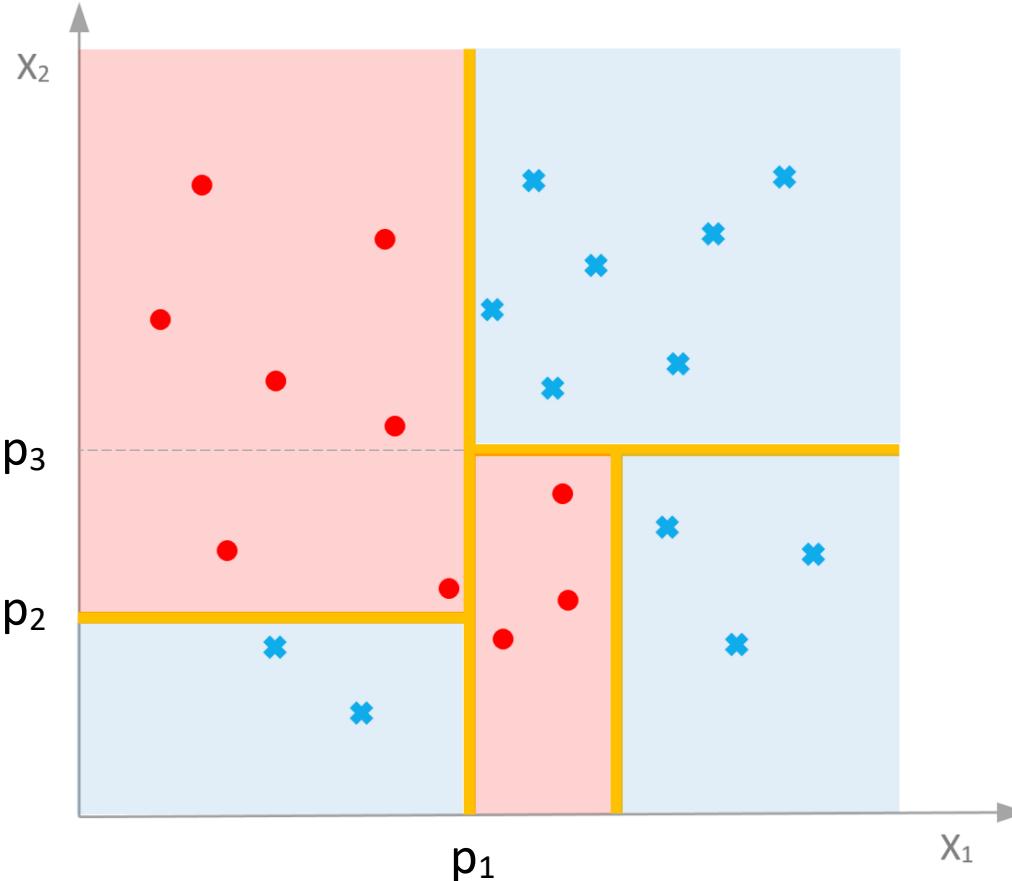
Konstrukcja drzewa



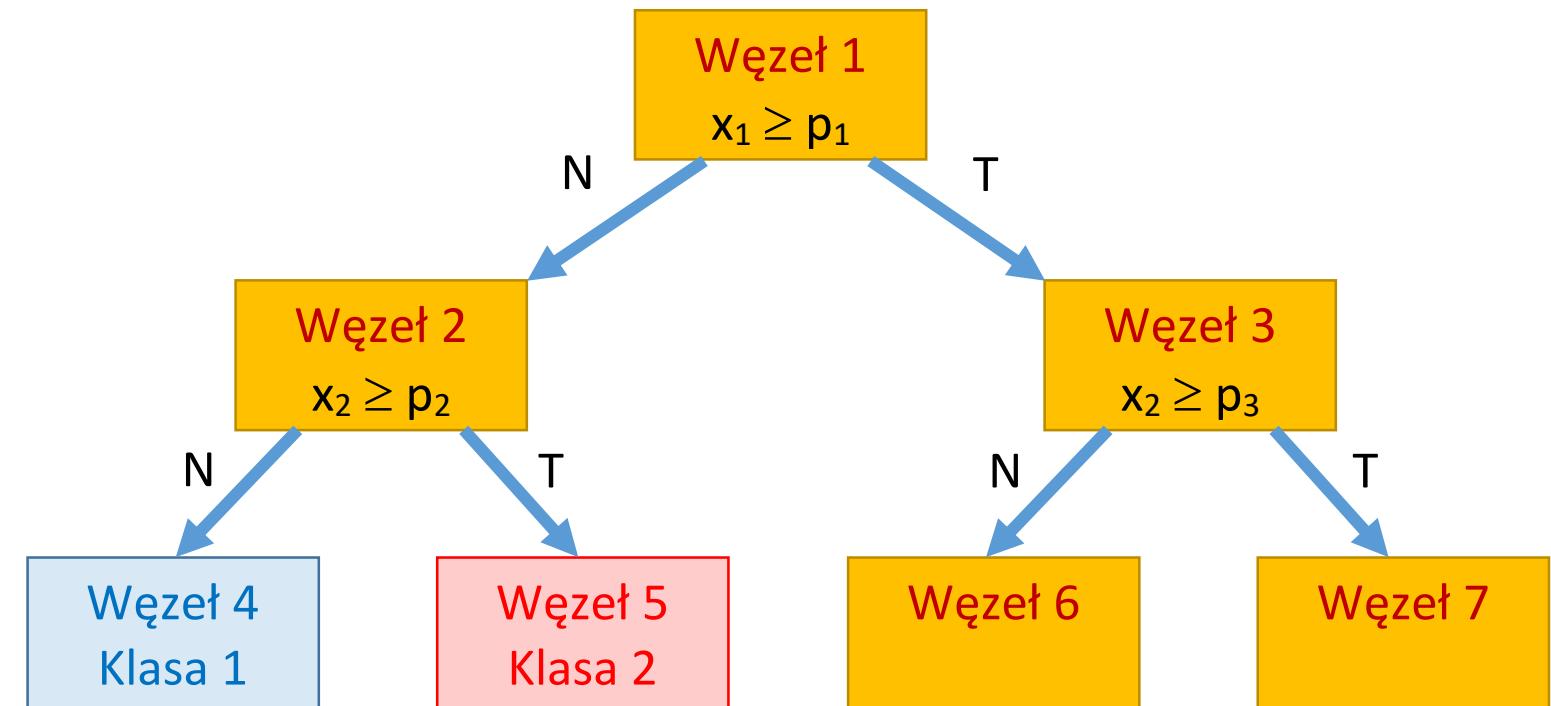
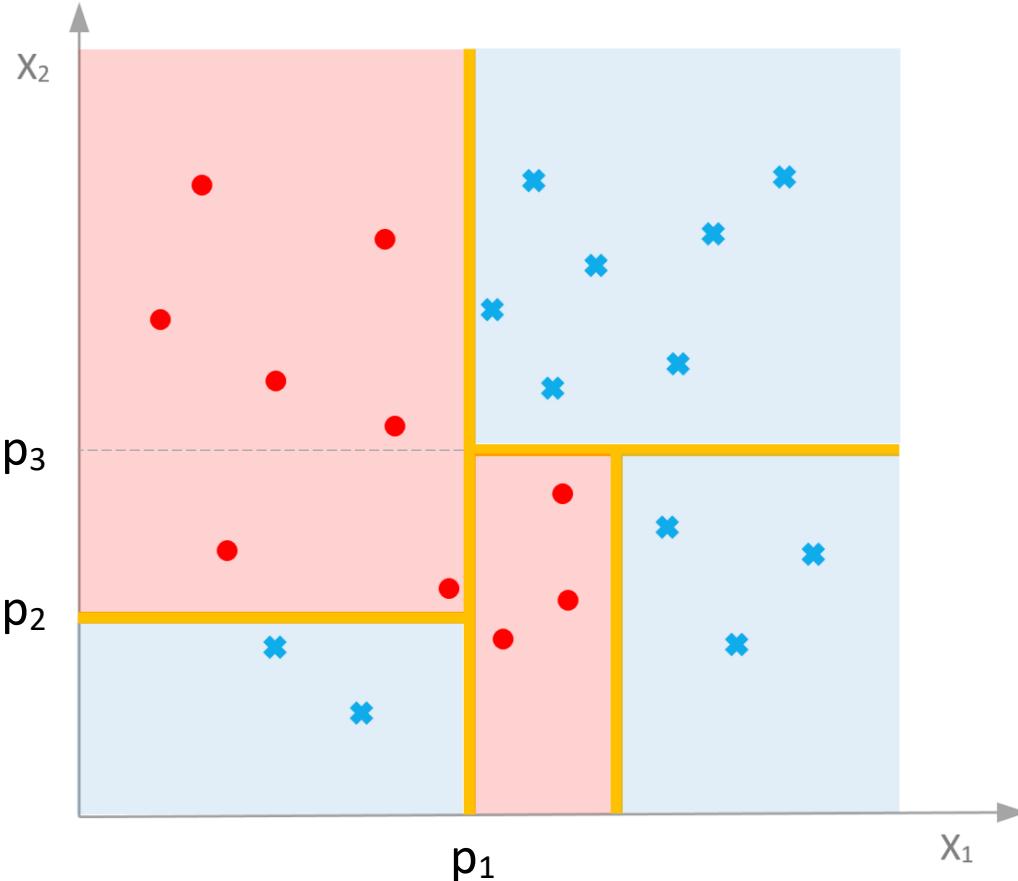
Konstrukcja drzewa



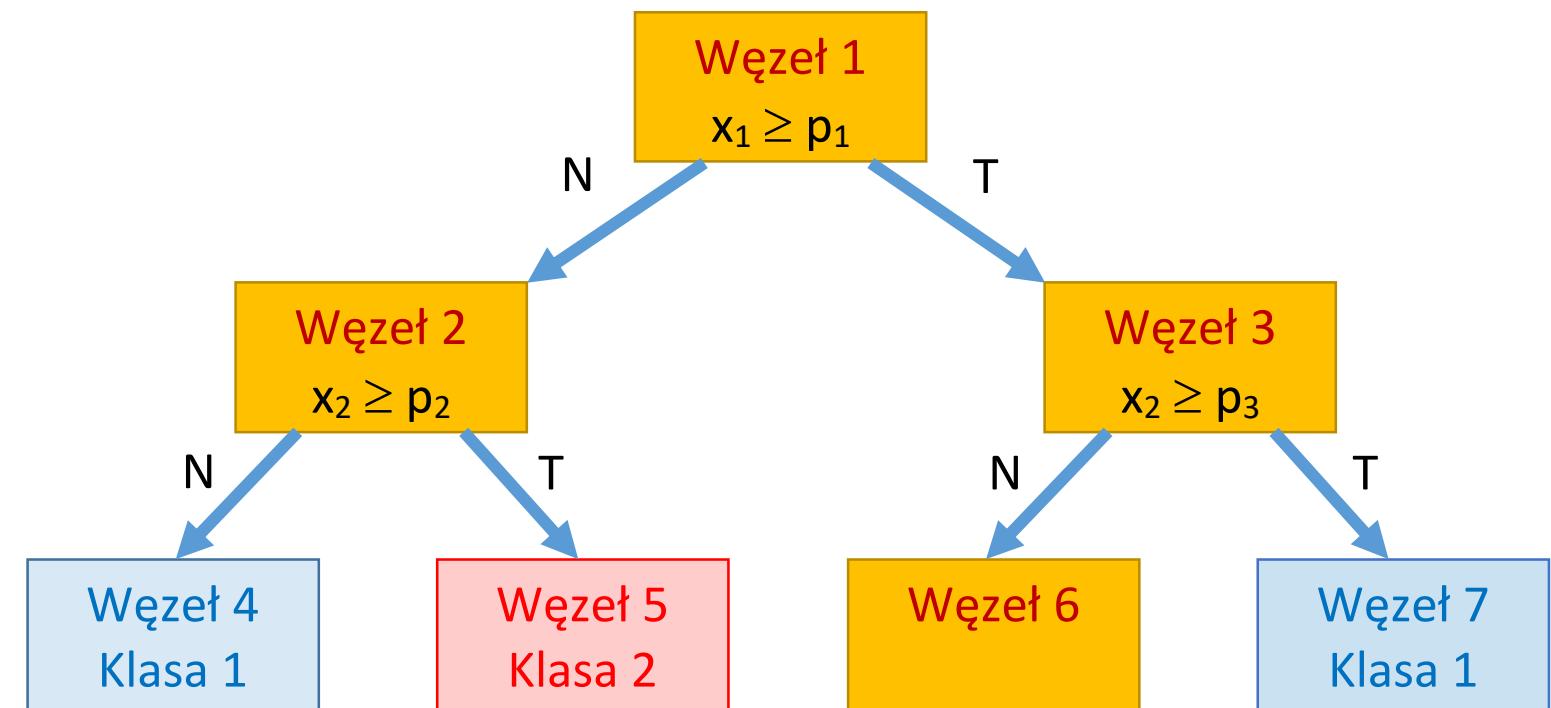
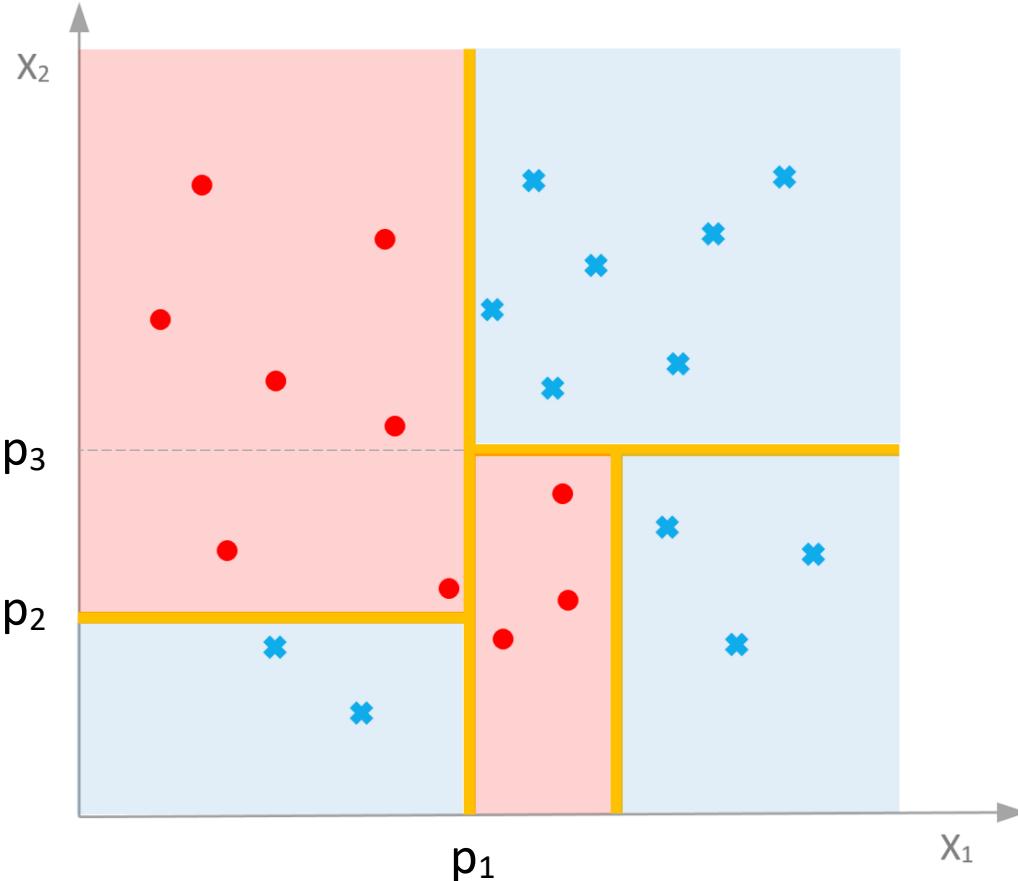
Konstrukcja drzewa



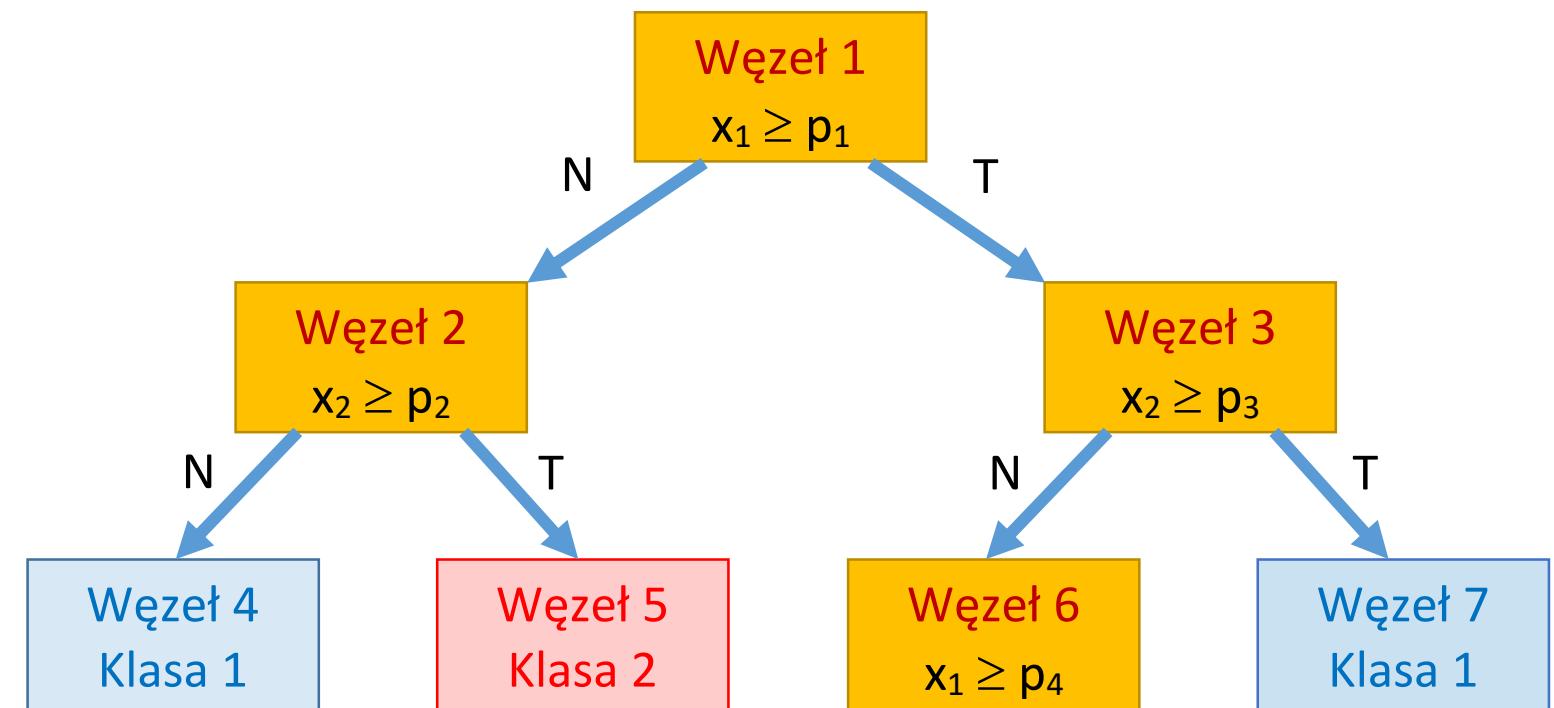
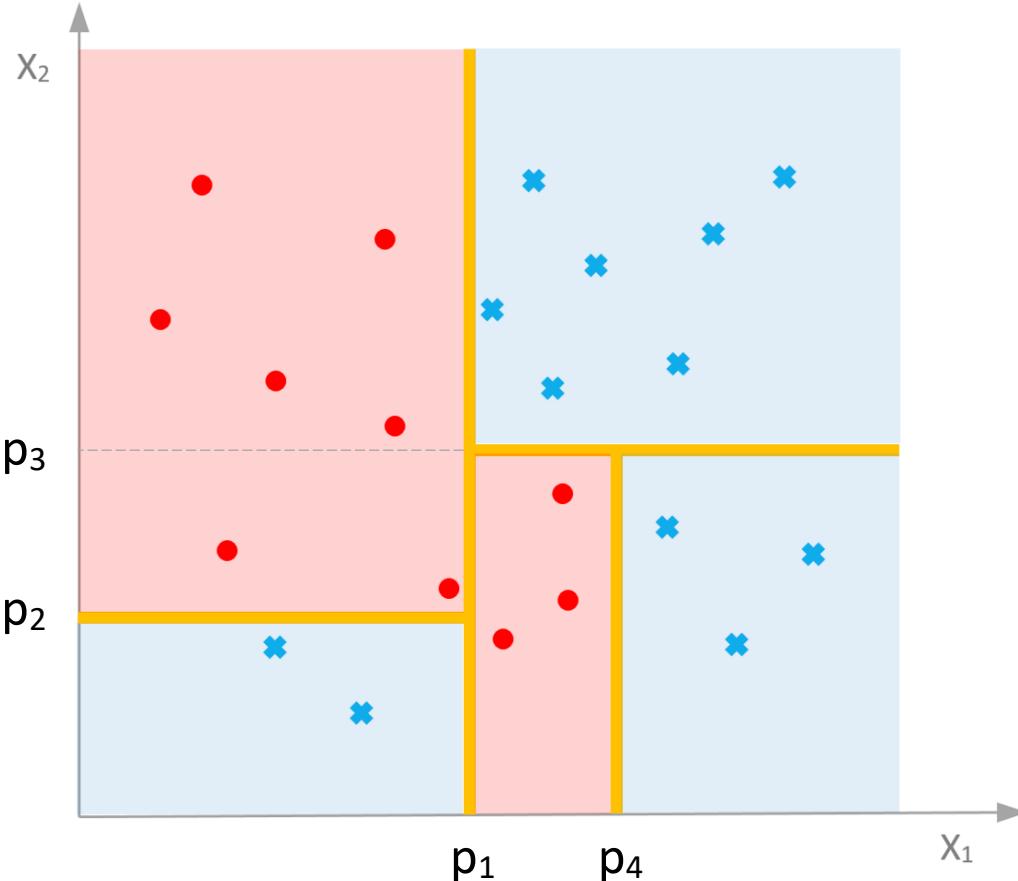
Konstrukcja drzewa



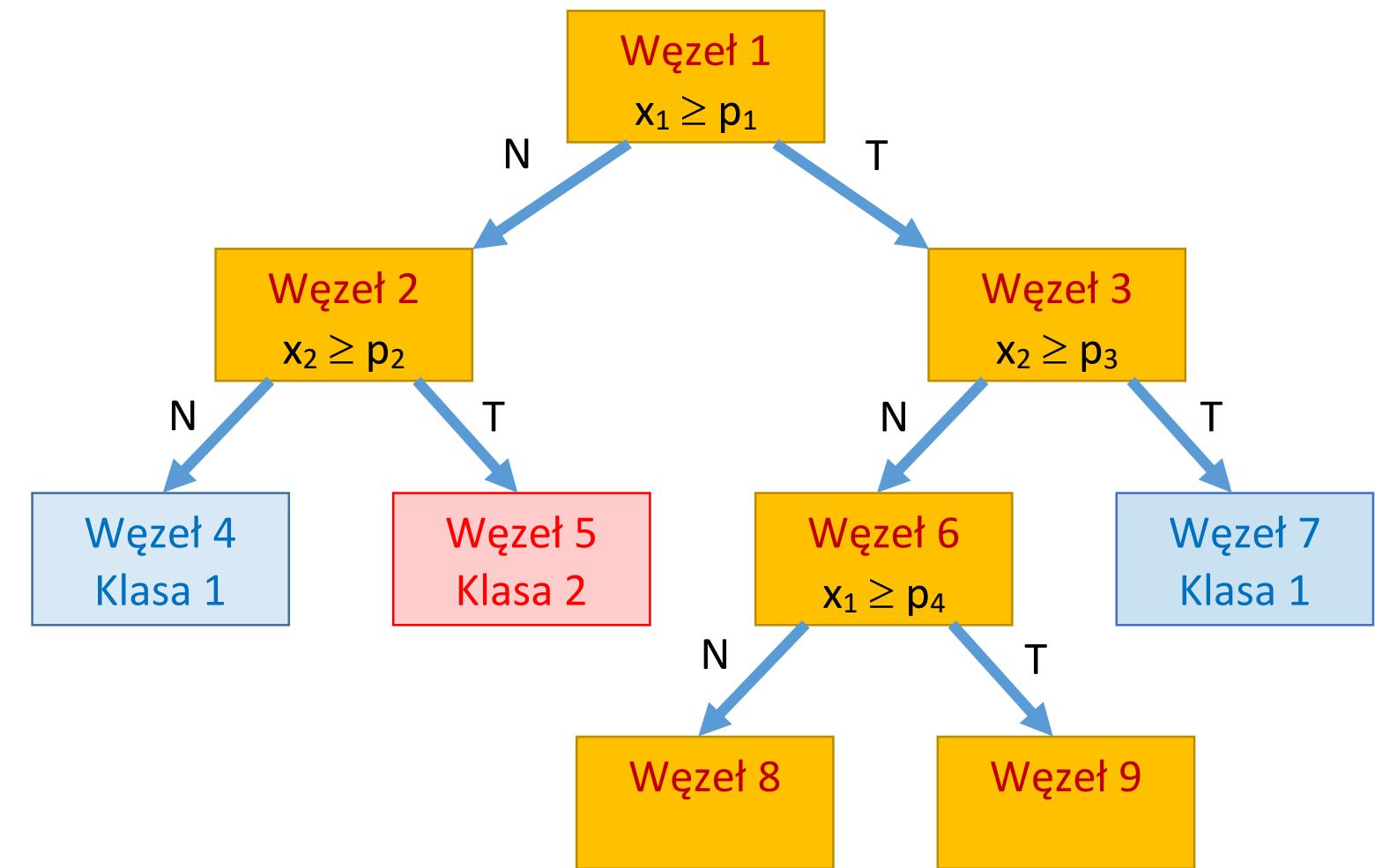
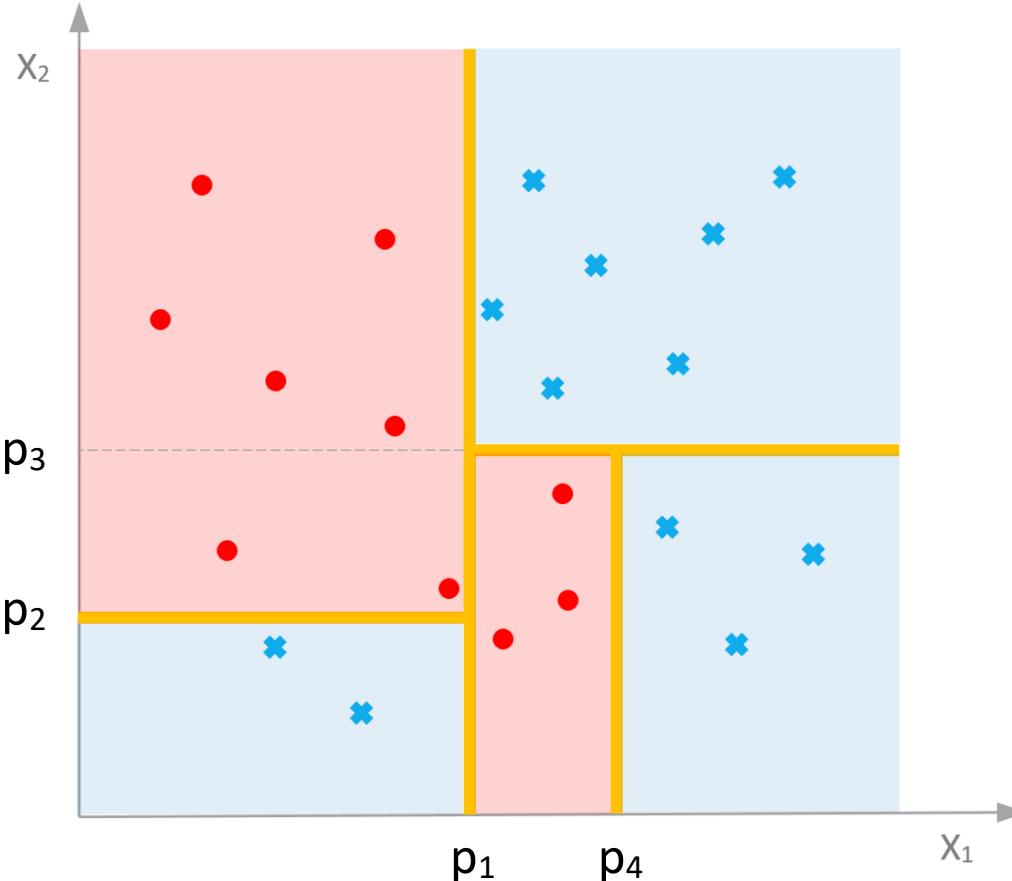
Konstrukcja drzewa



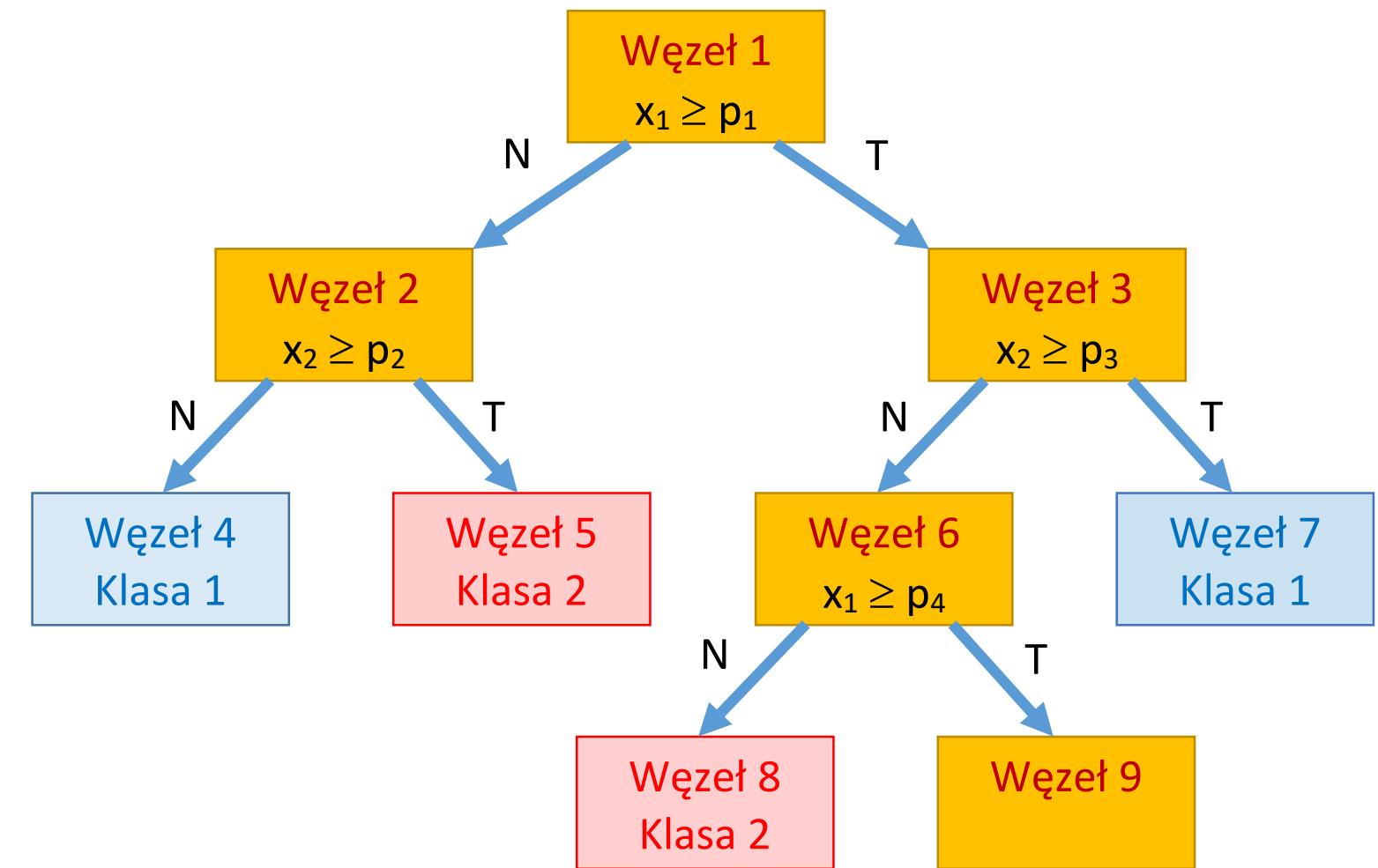
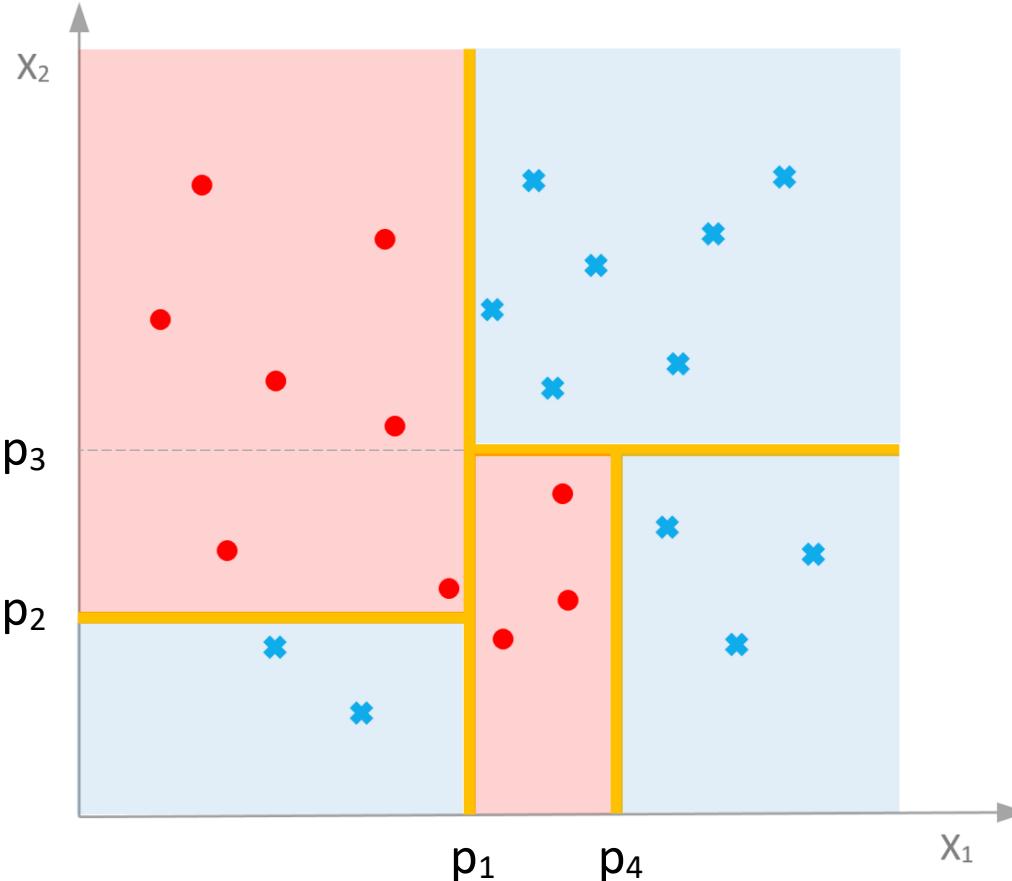
Konstrukcja drzewa



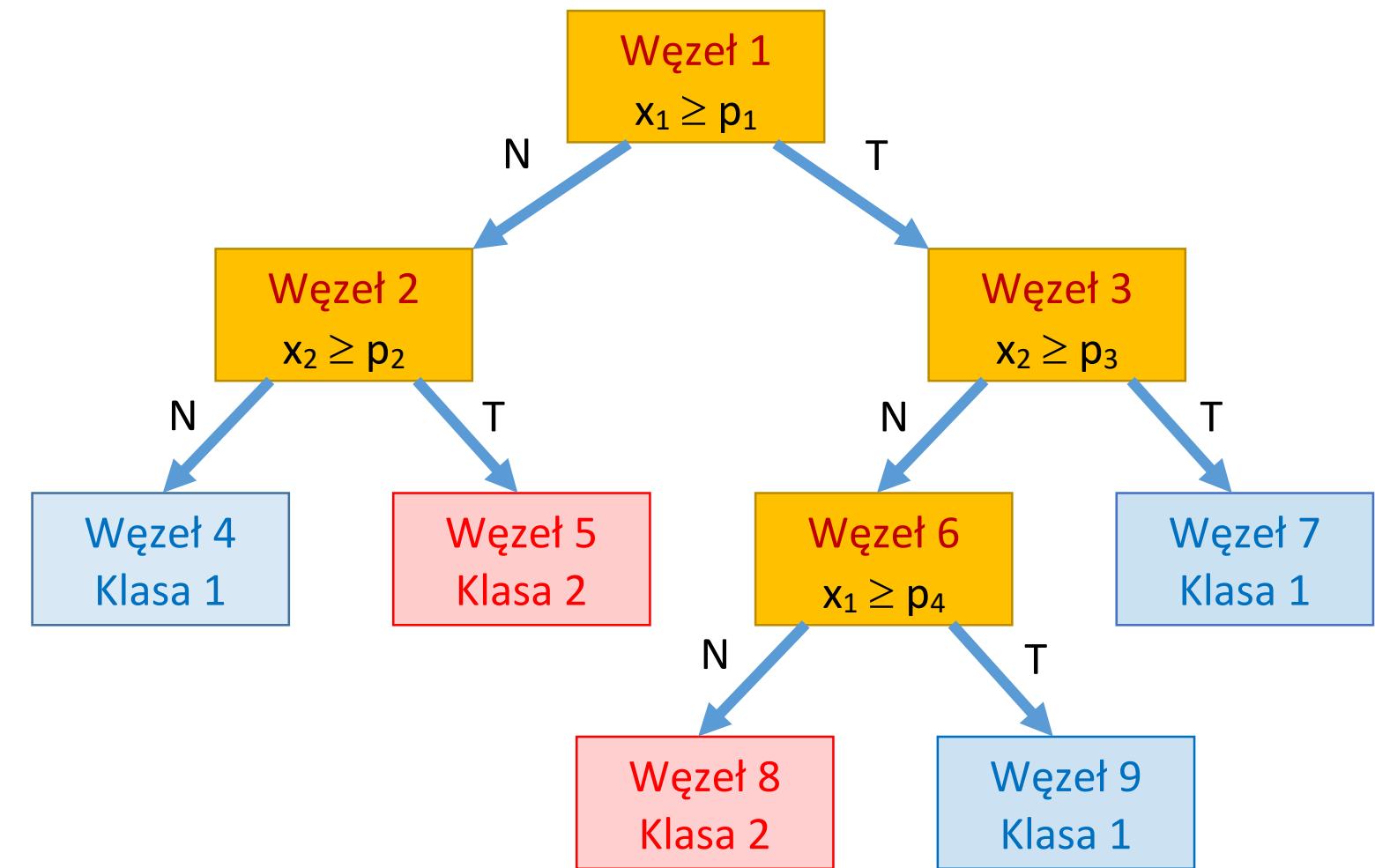
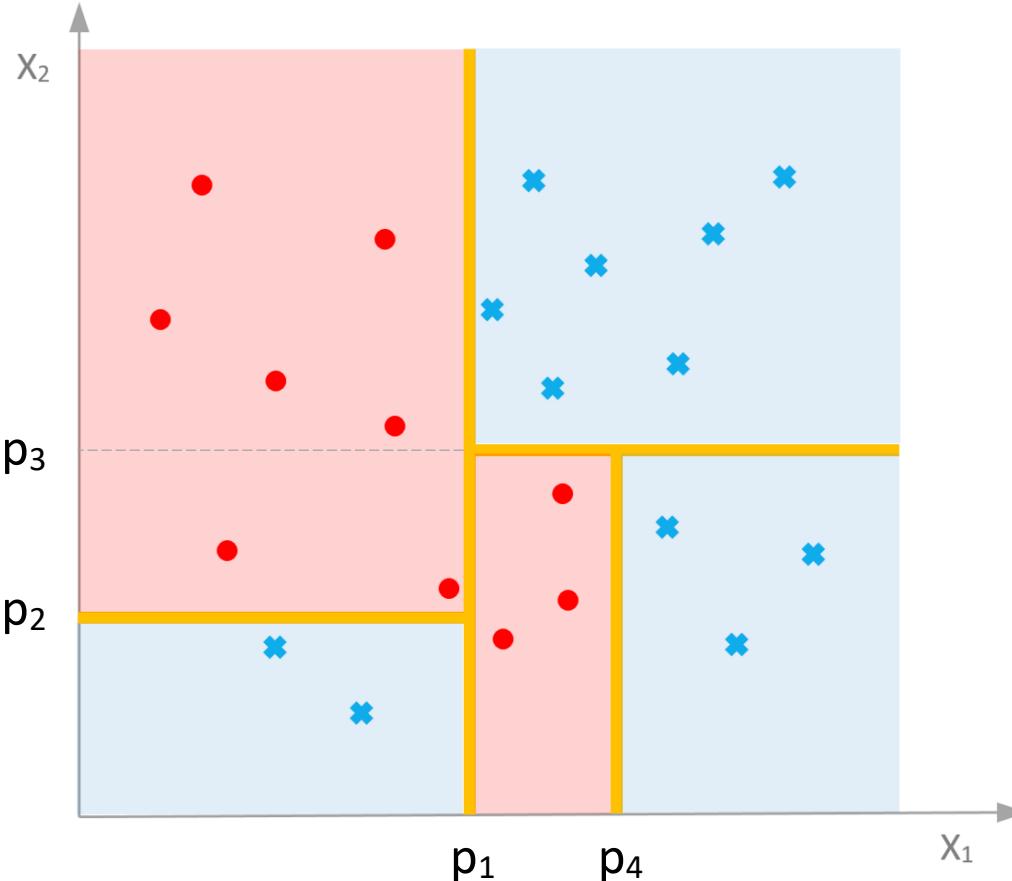
Konstrukcja drzewa



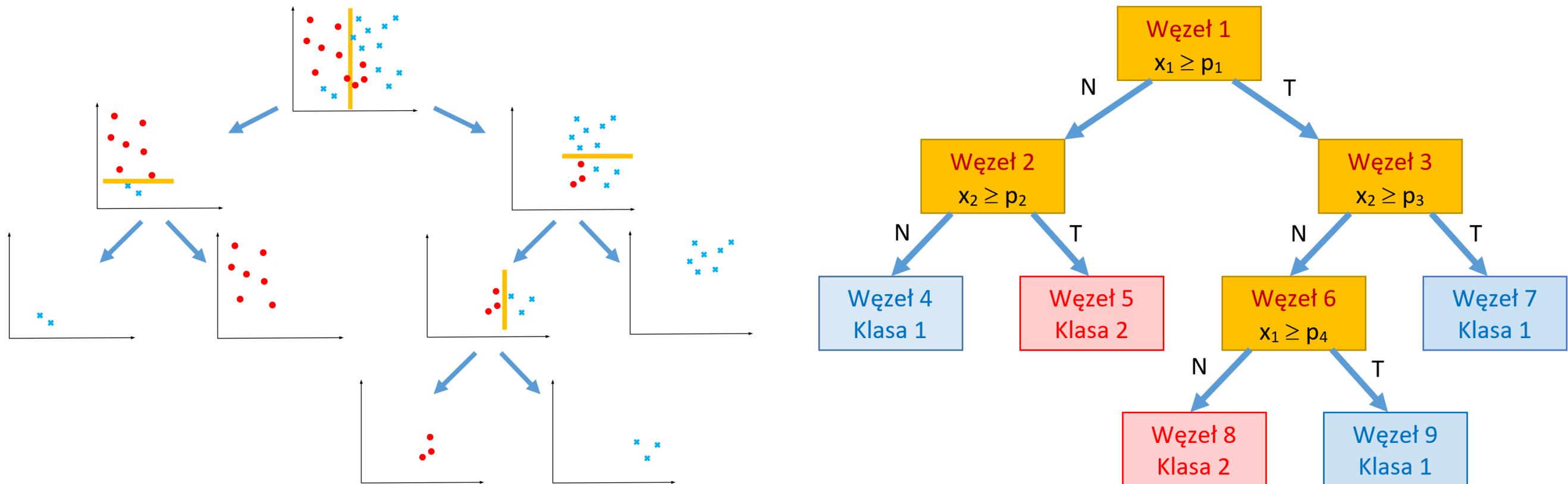
Konstrukcja drzewa



Konstrukcja drzewa

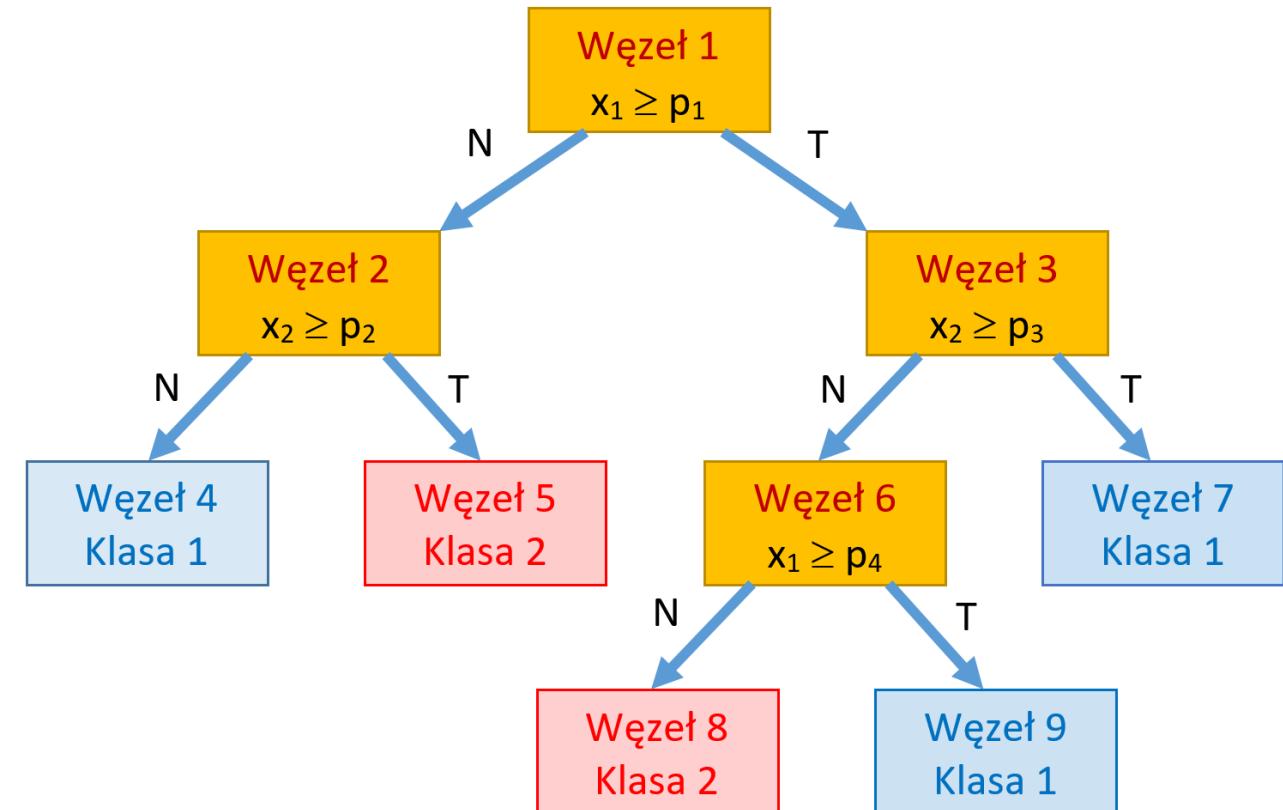


Konstrukcja drzewa



Drzewo decyzyjne

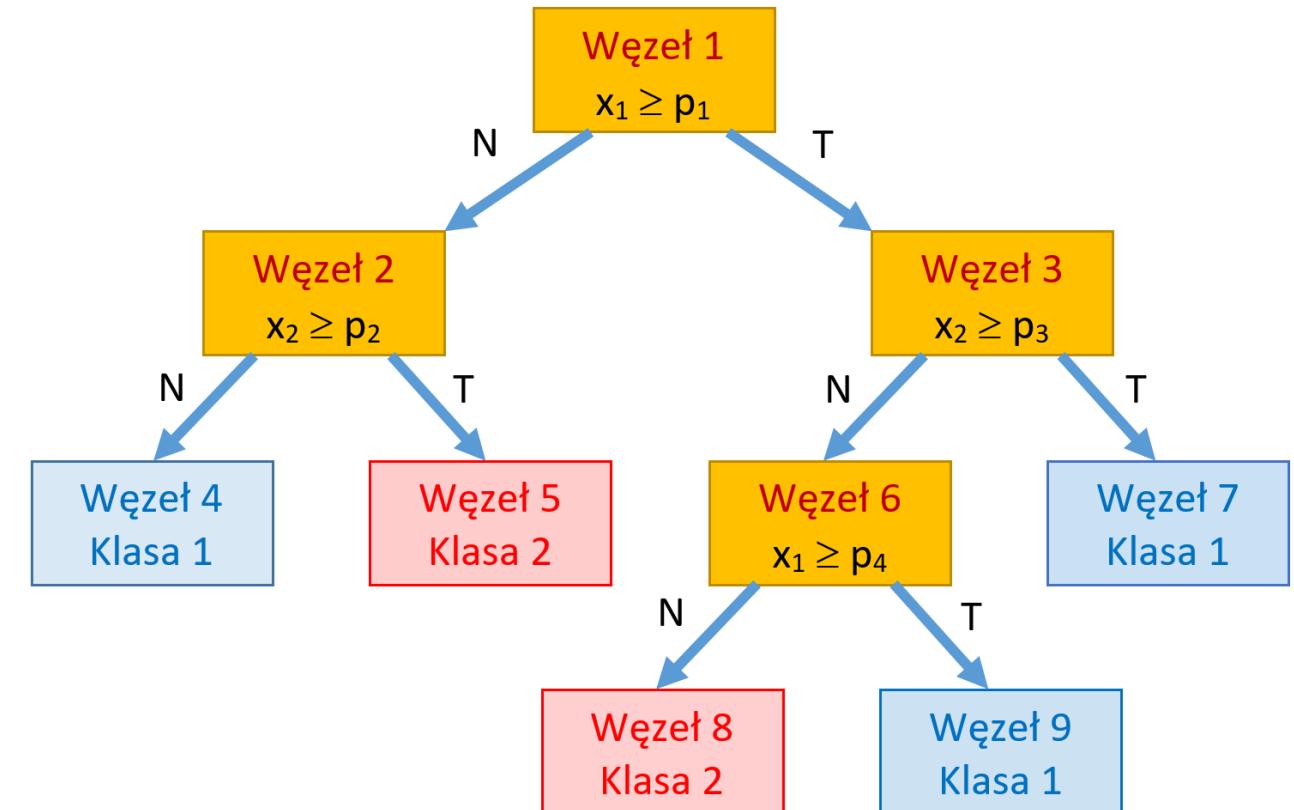
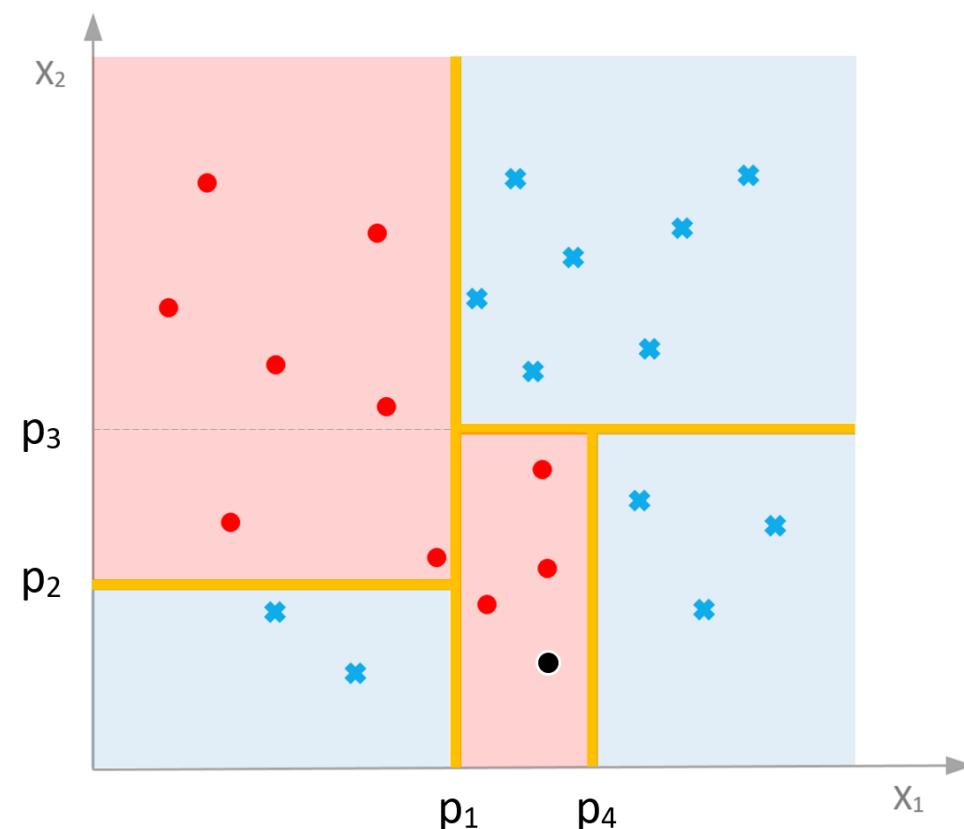
- Drzewo decyzyjne jest strukturą złożoną z **węzłów**, **gałęzi** i **liści** (węzłów terminalnych)
- Węzły odpowiadają **testom** przeprowadzanym na wartościach atrybutów, gałęzie odpowiadają wynikom tych testów, a liście – etykietom klas
- Drzewo konstruowane jest iteracyjnie. W każdej iteracji zachodzi podział zbioru danych na dwa podzbiory
- Drzewo binarne – z każdego węzła wychodzą tylko dwie gałęzie (test daje dwa wyniki)



Klasyfikacja nowego przykładu

Określenie klasy przykładu za pomocą drzewa decyzyjnego polega na przejściu od korzenia do jednego z liści.

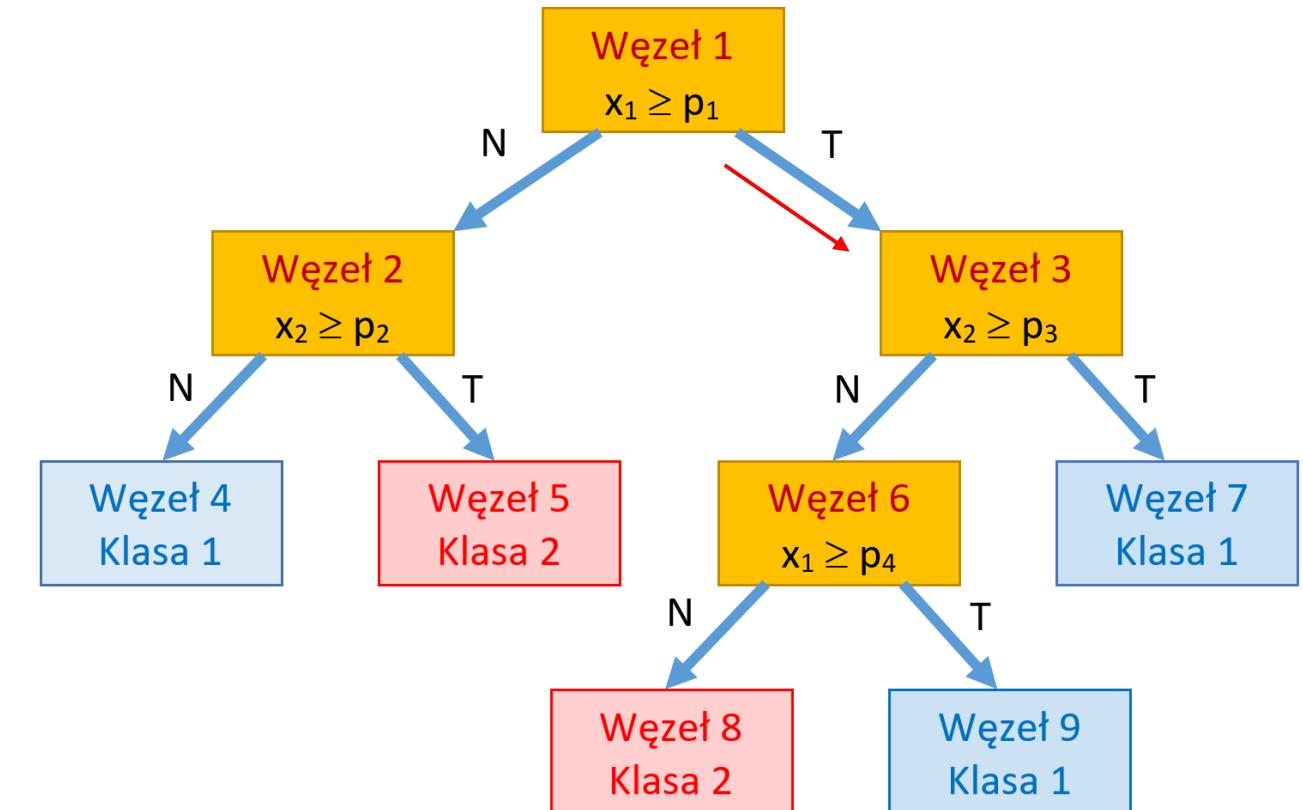
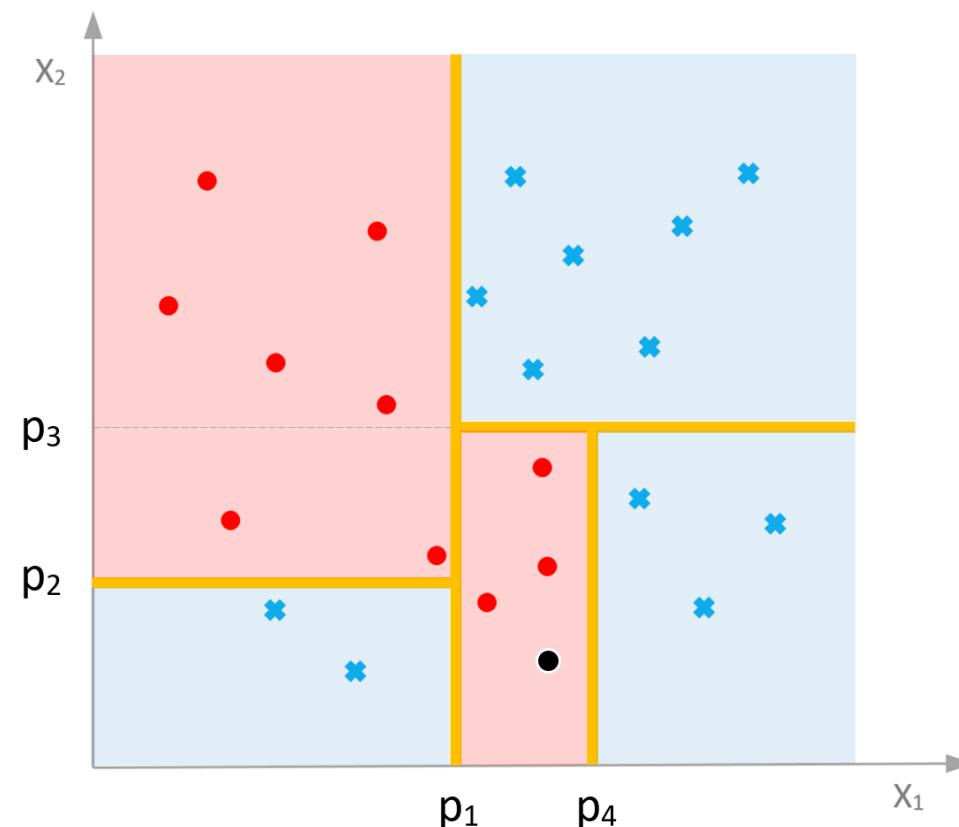
W odwiedzanych węzłach wykonujemy związane z nimi testy i poruszamy się po gałęziach odpowiadających uzyskiwanym wynikom.



Klasyfikacja nowego przykładu

Określenie klasy przykładu za pomocą drzewa decyzyjnego polega na przejściu od korzenia do jednego z liści.

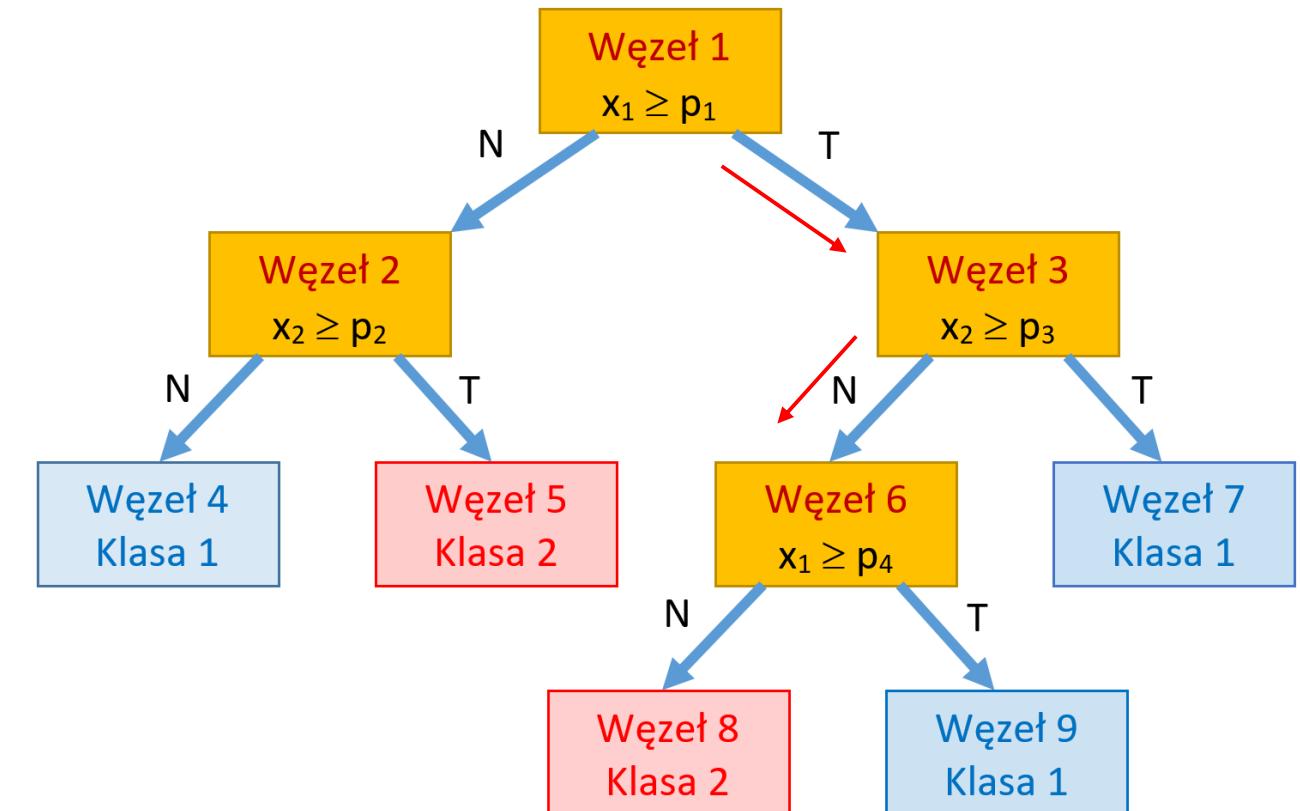
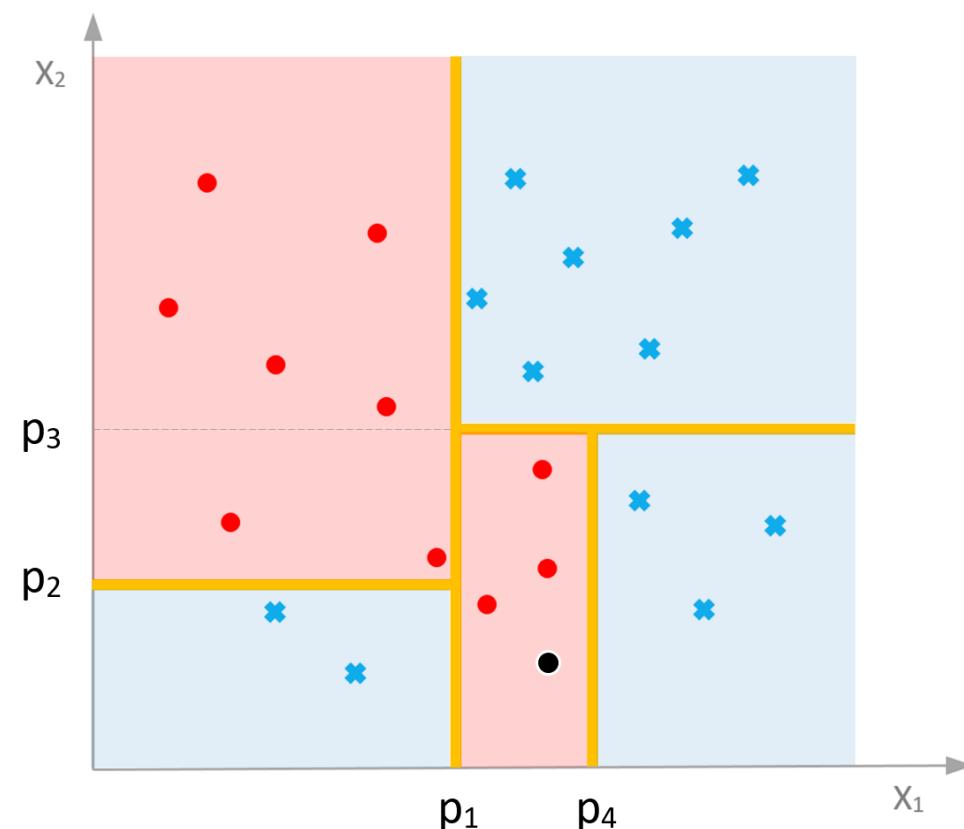
W odwiedzanych węzłach wykonujemy związane z nimi testy i poruszamy się po gałęziach odpowiadających uzyskiwanym wynikom.



Klasyfikacja nowego przykładu

Określenie klasy przykładu za pomocą drzewa decyzyjnego polega na przejściu od korzenia do jednego z liści.

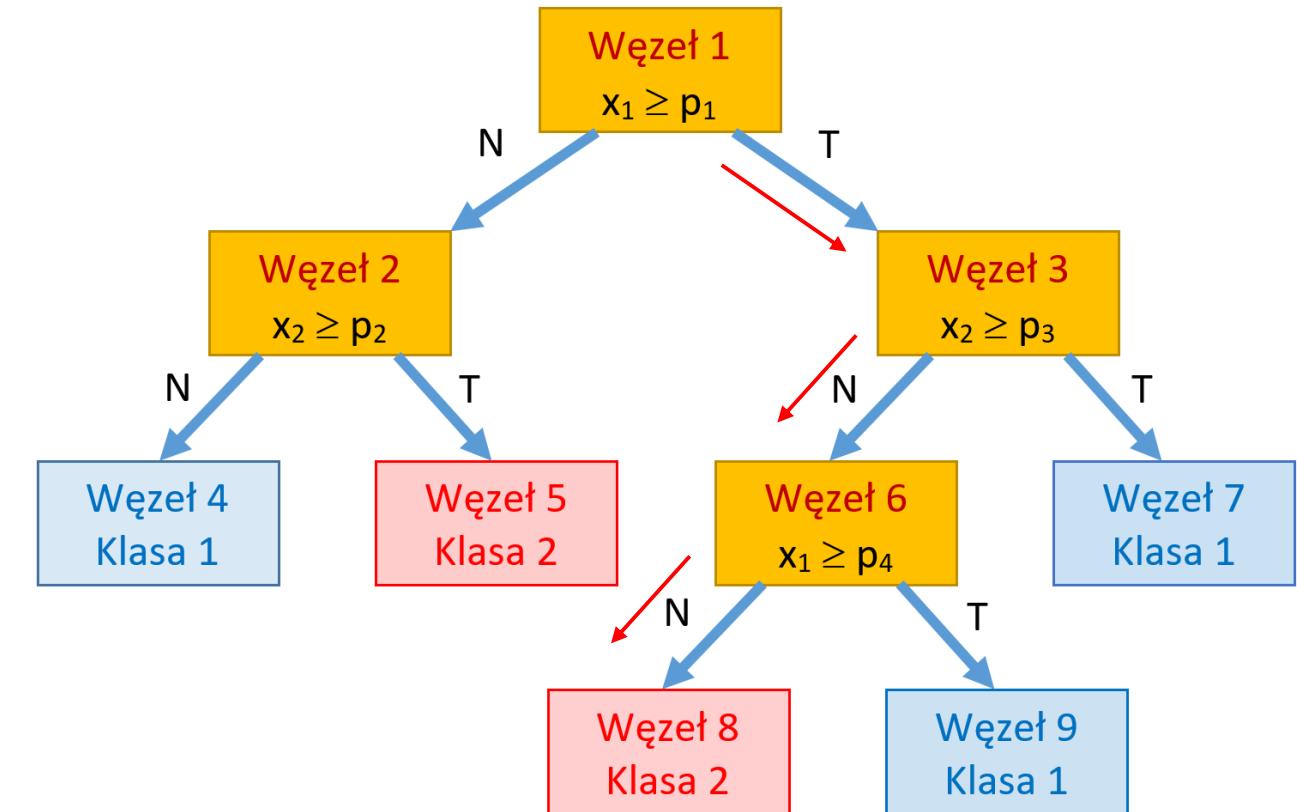
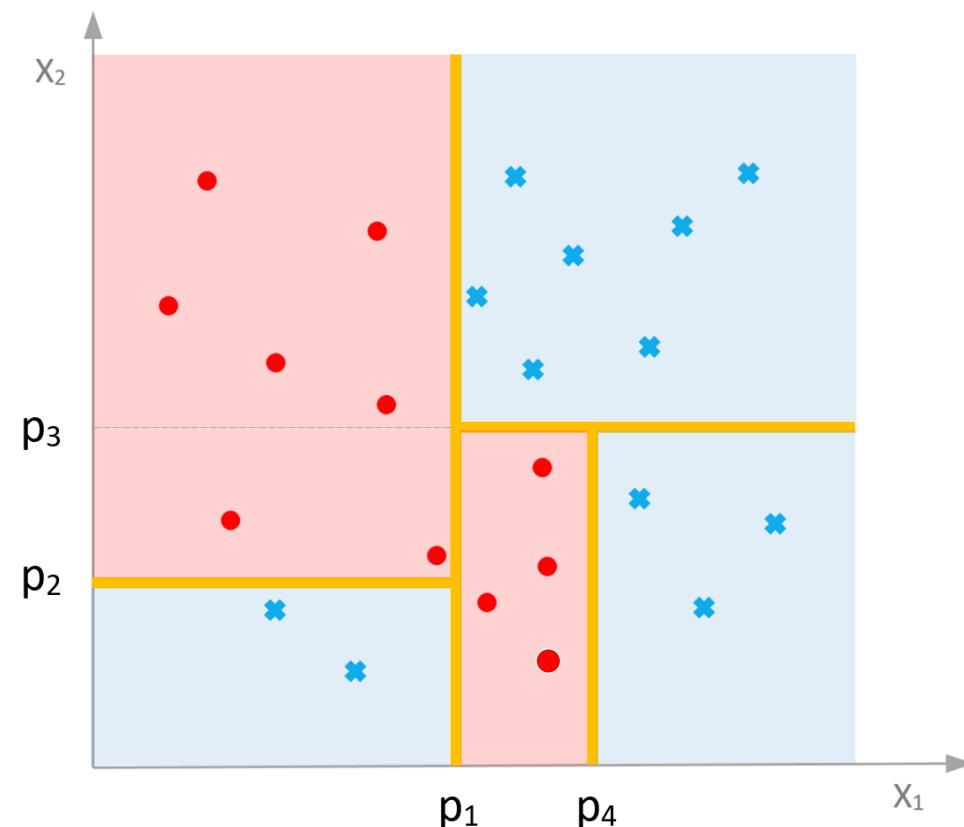
W odwiedzanych węzłach wykonujemy związane z nimi testy i poruszamy się po gałęziach odpowiadających uzyskiwanym wynikom.



Klasyfikacja nowego przykładu

Określenie klasy przykładu za pomocą drzewa decyzyjnego polega na przejściu od korzenia do jednego z liści.

W odwiedzanych węzłach wykonujemy związane z nimi testy i poruszamy się po gałęziach odpowiadających uzyskiwanym wynikom.



Zstępujący schemat konstrukcji drzew decyzyjnych

- Zaczynając od korzenia (poziom 0) przemieszczamy się do kolejnych poziomów podejmując decyzje o utworzeniu liścia bądź węzła.
- Jeśli **kryterium stopu** jest spełnione tworzony jest liść i na podstawie podzbioru przykładów, które do niego dotarły ustalana jest jego etykieta.
- W przeciwnym przypadku tworzony jest węzeł i wybierany jest dla niego test.

Kryterium stopu i ustalenie etykiety

Liść tworzony jest w przypadkach gdy:

- aktualny zbiór przykładów zawiera wyłącznie przykłady jednej klasy; wtedy do etykiety liścia wpisuje się symbol tej klasy,
- wyraźna większość przykładów z aktualnego zbioru przykładów ma tę samą klasę; wtedy do etykiety liścia wpisuje się symbol tej klasy,
- wyczerpał się zbiór testów; wtedy do etykiety liścia wpisuje się symbol klasy domyślnej.

Rodzaje i liczba testów

- testy tożsamościowe – sprawdzamy wartość atrybutu; wynikiem testu jest wartość atrybutu, np. dla atrybutu *wiatr* wynikiem testu może być *słaby* lub *silny* (N, P)

Liczba możliwych testów – tyle ile jest atrybutów

Test dla atrybutów:
N – nominalnych,
P – porządkowych,
C – ciągłych

Rodzaje i liczba testów

- testy tożsamościowe – sprawdzamy wartość atrybutu; wynikiem testu jest wartość atrybutu, np. dla atrybutu *wiatr* wynikiem testu może być *słaby* lub *silny* (**N, P**)

Liczba możliwych testów – tyle ile jest atrybutów

Test dla atrybutów:
N – nominalnych,
P – porządkowych,
C – ciągłych

- testy równościowe (**N, P**)

$$test(x) = \begin{cases} 1 & \text{jeśli } a(x) = v \\ 0 & \text{jeśli } a(x) \neq v \end{cases}, \quad \text{gdzie } a \text{ jest atrybutem przykładowa } x, v \text{ jest jedną z możliwych wartości atrybutu}$$

Liczba możliwych testów – możliwych testów dla jednego atrybutu jest tyle ile ten atrybut ma wartości

Rodzaje i liczba testów

- testy tożsamościowe – sprawdzamy wartość atrybutu; wynikiem testu jest wartość atrybutu, np. dla atrybutu *wiatr* wynikiem testu może być *słaby* lub *silny* (**N, P**)

Liczba możliwych testów – tyle ile jest atrybutów

Test dla atrybutów:
N – nominalnych,
P – porządkowych,
C – ciągłych

- testy równościowe (**N, P**)

$$test(x) = \begin{cases} 1 & \text{jeśli } a(x) = v \\ 0 & \text{jeśli } a(x) \neq v \end{cases}, \quad \text{gdzie } a \text{ jest atrybutem przykładowa } x, v \text{ jest jedną z możliwych wartości atrybutu}$$

Liczba możliwych testów – możliwych testów dla jednego atrybutu jest tyle ile ten atrybut posiada wartości

- testy przynależnościowe (**N, P, C**)

$$test(x) = \begin{cases} 1 & \text{jeśli } a(x) \in V \\ 0 & \text{jeśli } a(x) \notin V \end{cases}, \quad \text{gdzie } V \text{ jest podzbiorem wartości atrybutu}$$

Liczba możliwych testów – możliwych testów dla jednego atrybutu jest tyle ile podzbiorów V zdefiniujemy

Rodzaje i liczba testów

- testy podziałowe (N, P, C)

$$test(x) = \begin{cases} 1 & \text{jeśli } a(x) \in V_1 \\ 2 & \text{jeśli } a(x) \in V_2 \\ \dots \\ m & \text{jeśli } a(x) \in V_m \end{cases}, \quad \text{gdzie parami rozłączne zbiory } V_1, V_2, \dots, V_m \text{ stanowią wyczerpujący podział przeciwdziedziny atrybutu}$$

Test dla atrybutów:
N – nominalnych,
P – porządkowych,
C – ciągłych

Liczba możliwych testów – możliwych testów dla jednego atrybutu jest tyle ile podzbiorów V zdefiniujemy

Rodzaje i liczba testów

- testy podziałowe (N, P, C)

$$test(x) = \begin{cases} 1 & \text{jeśli } a(x) \in V_1 \\ 2 & \text{jeśli } a(x) \in V_2 \\ \dots \\ m & \text{jeśli } a(x) \in V_m \end{cases}, \quad \text{gdzie parami rozłączne zbiory } V_1, V_2, \dots, V_m \text{ stanowią wyczerpujący podział przeciwdziedziny atrybutu}$$

Test dla atrybutów:
N – nominalnych,
P – porządkowych,
C – ciągłych

Liczba możliwych testów – możliwych testów dla jednego atrybutu jest tyle ile podzbiorów V zdefiniujemy

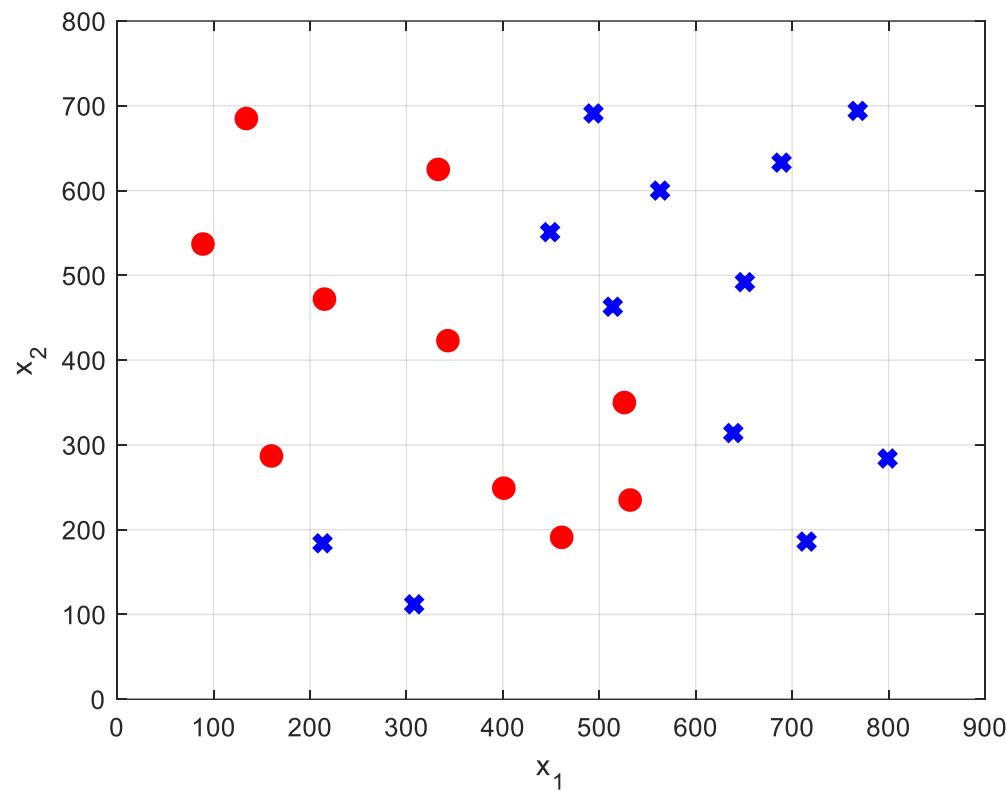
- testy nierównościowe (P, C)

$$test(x) = \begin{cases} 1 & \text{jeśli } a(x) \geq p \\ 0 & \text{jeśli } a(x) < p \end{cases}, \quad \text{gdzie } p \text{ jest wartością progową z przeciwdziedziny atrybutu}$$

Liczba możliwych testów – możliwych testów dla jednego atrybutu jest tyle ile przyjmuje on różnych wartości w zbiorze uczącym minus 1

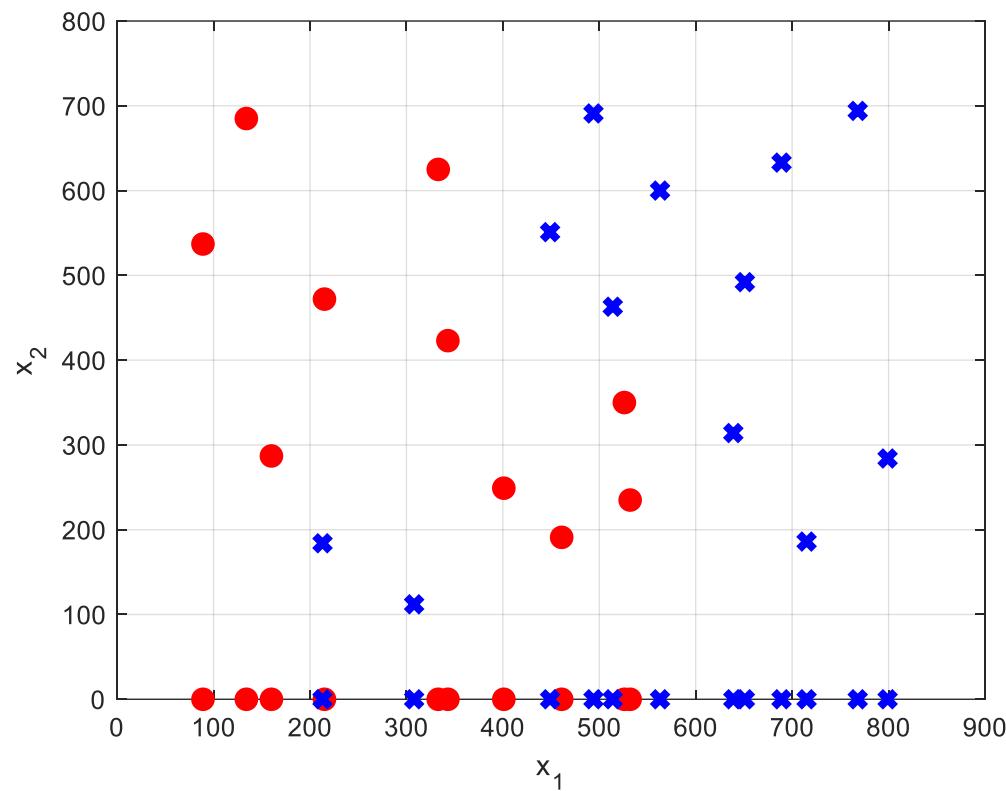
Rodzaje i liczba testów

Liczba możliwych testów nierównościowych



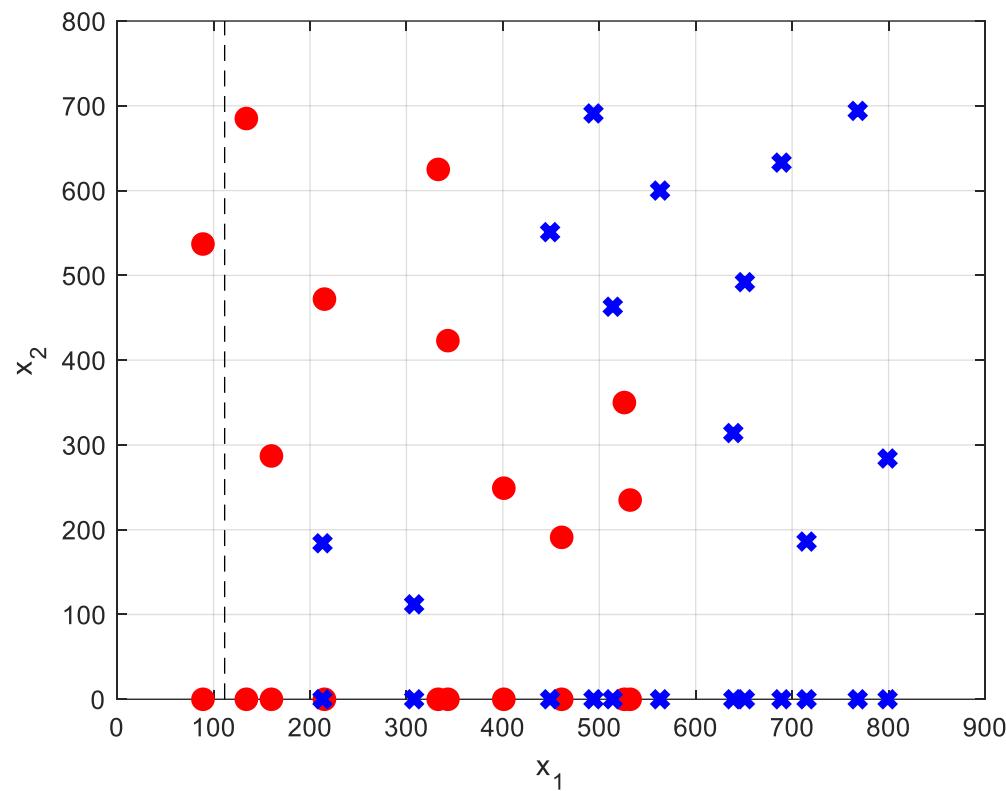
Rodzaje i liczba testów

Liczba możliwych testów nierównościowych



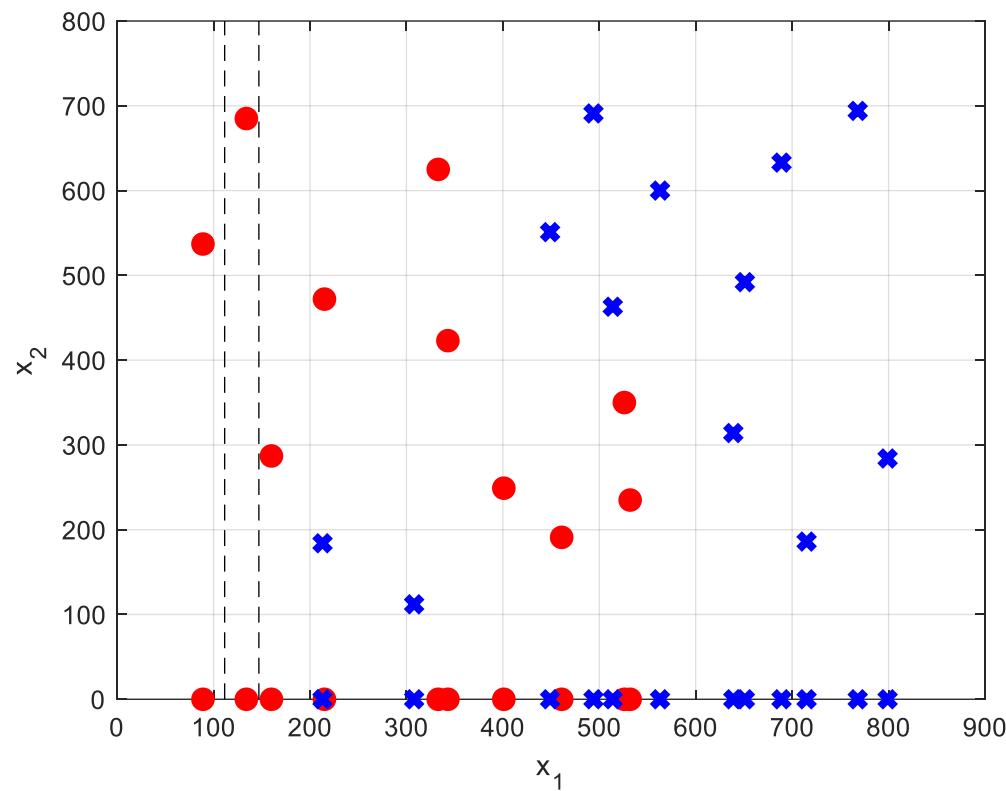
Rodzaje i liczba testów

Liczba możliwych testów nierównościowych



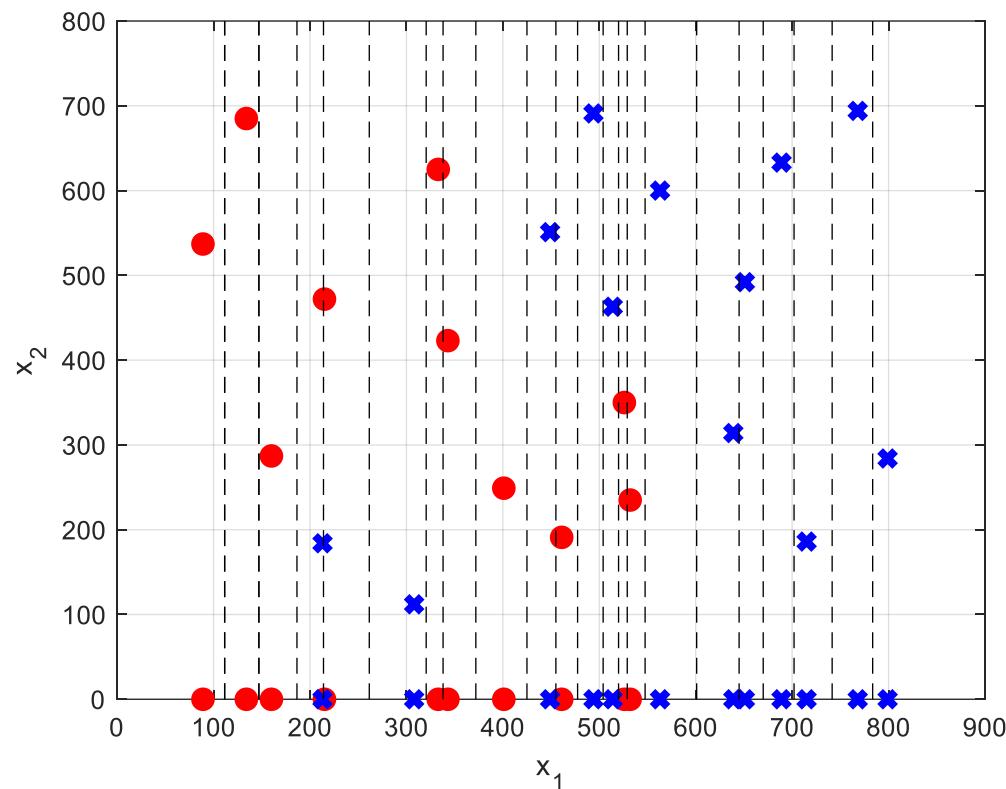
Rodzaje i liczba testów

Liczba możliwych testów nierównościowych



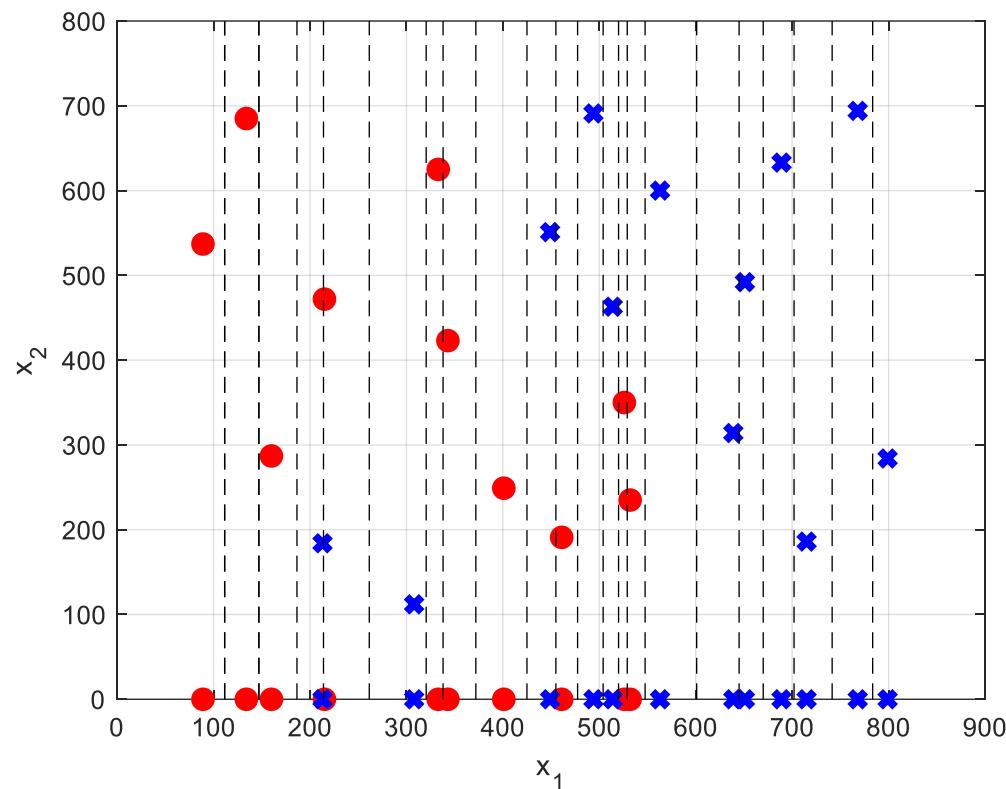
Rodzaje i liczba testów

Liczba możliwych testów nierównościowych



Rodzaje i liczba testów

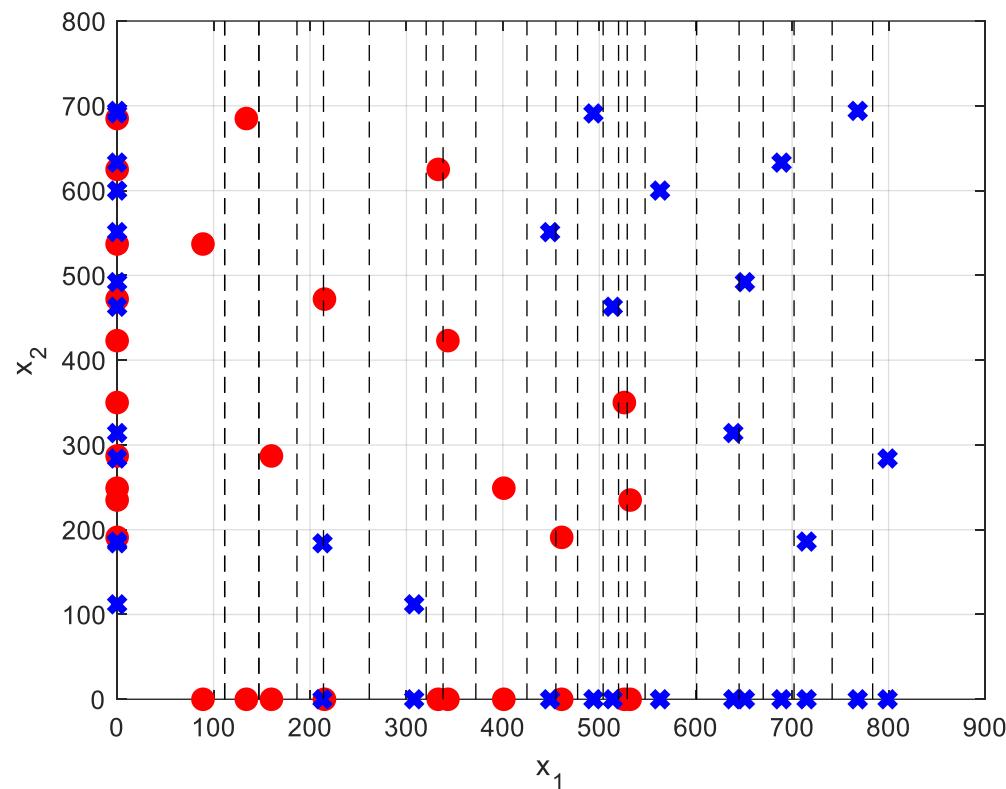
Liczba możliwych testów nierównościowych



Liczba testów dla $x_1: N - 1$

Rodzaje i liczba testów

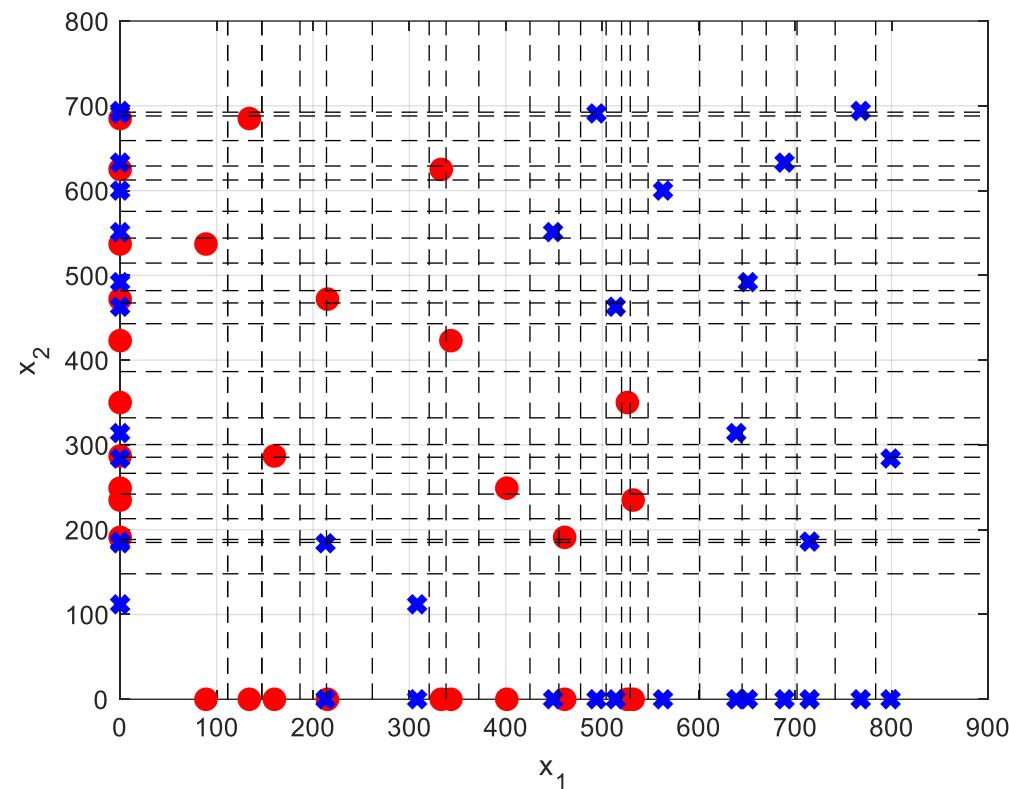
Liczba możliwych testów nierównościowych



Liczba testów dla $x_1: N - 1$

Rodzaje i liczba testów

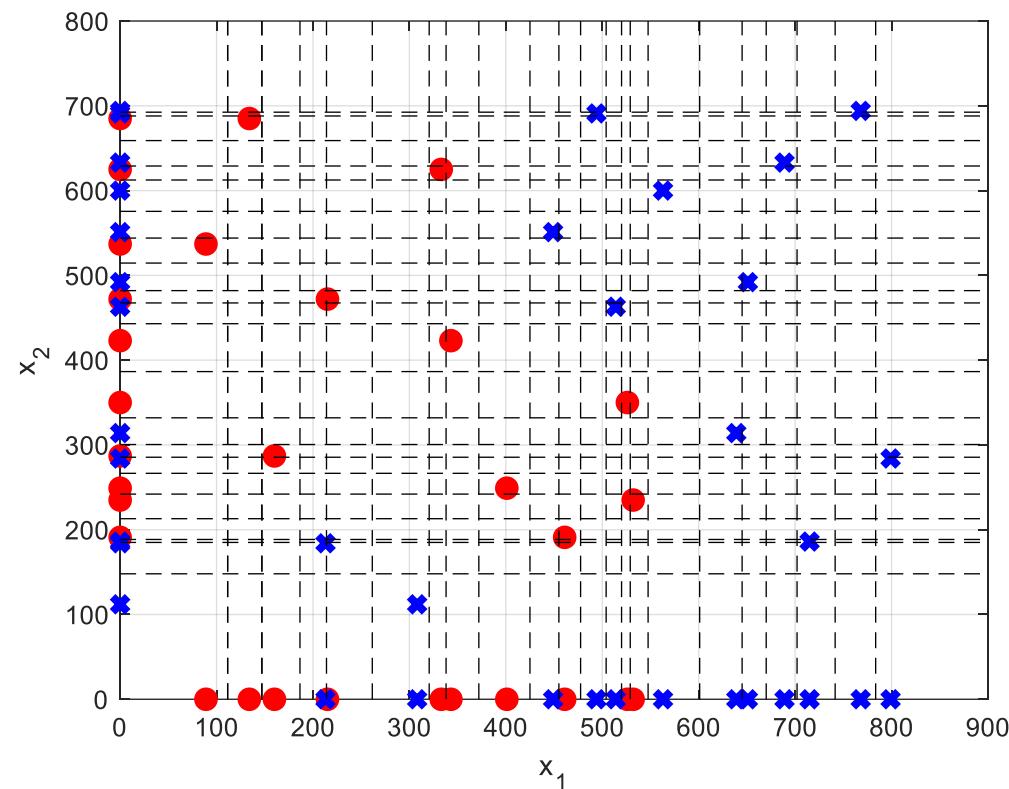
Liczba możliwych testów nierównościowych



Liczba testów dla $x_1: N - 1$

Rodzaje i liczba testów

Liczba możliwych testów nierównościowych

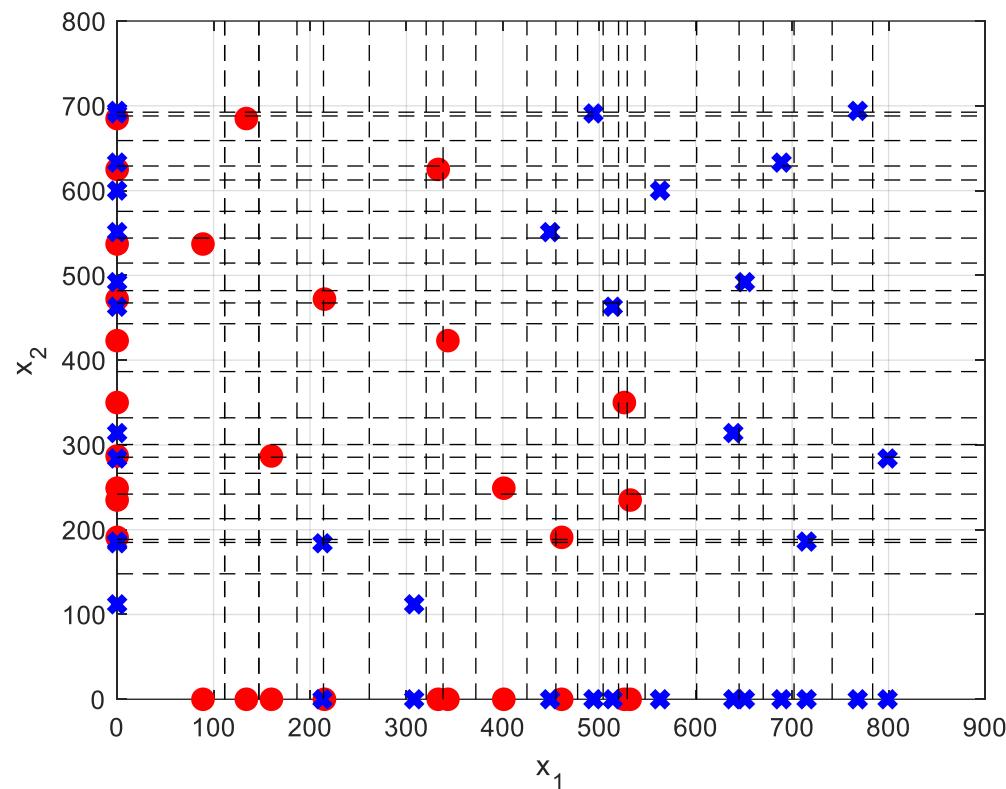


Liczba testów dla $x_1: N - 1$

Liczba testów dla $x_2: N - 1$

Rodzaje i liczba testów

Liczba możliwych testów nierównościowych



Liczba testów dla $x_1: N - 1$

Liczba testów dla $x_2: N - 1$

Liczba wszystkich testów:

$$L_t \leq (N - 1) n$$

W naszym przypadku:

$$L_t = 21 \cdot 2 = 42$$

Kryterium wyboru testu

Test powinien dzielić zbiór przykładów tak, aby po podziale zróżnicowanie klas w podzbiorach było maksymalne.

Jako miary rozkładu klas w zbiorze przykładów (miary **zanieczyszczenia węzła**, miary informacji w zbiorze) używa się:

- entropi:

$$I_E(A) = -\sum_{k=1}^K p_k \log_2 p_k$$

- współczynnika Giniego:

$$I_G(A) = 1 - \sum_{k=1}^K p_k^2$$

- błędu klasyfikacji:

$$I_B(A) = 1 - \max p_k$$

gdzie p_k jest proporcją liczby przykładów z klasy k ($N_{A,k}$) do liczby wszystkich przykładów w zbiorze A (N_A): $p_k = \frac{N_{A,k}}{N_A}$

Kryterium wyboru testu

Test powinien dzielić zbiór przykładów tak, aby po podziale zróżnicowanie klas w podzbiorach było maksymalne.

Jako miary rozkładu klas w zbiorze przykładów (miary **zanieczyszczenia węzła**, miary informacji w zbiorze) używa się:

- entropi:

$$I_E(A) = -\sum_{k=1}^K p_k \log_2 p_k$$

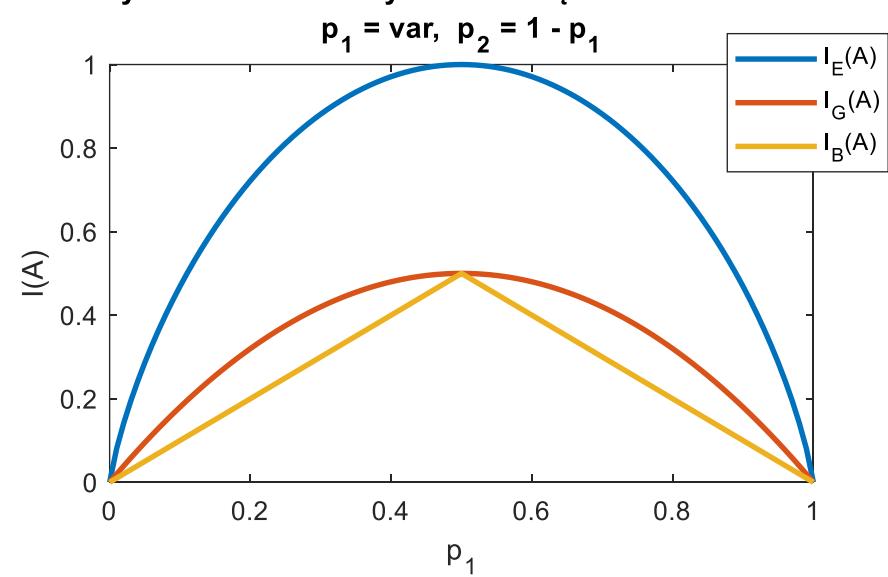
- współczynnika Giniego:

$$I_G(A) = 1 - \sum_{k=1}^K p_k^2$$

- błędu klasyfikacji:

$$I_B(A) = 1 - \max p_k$$

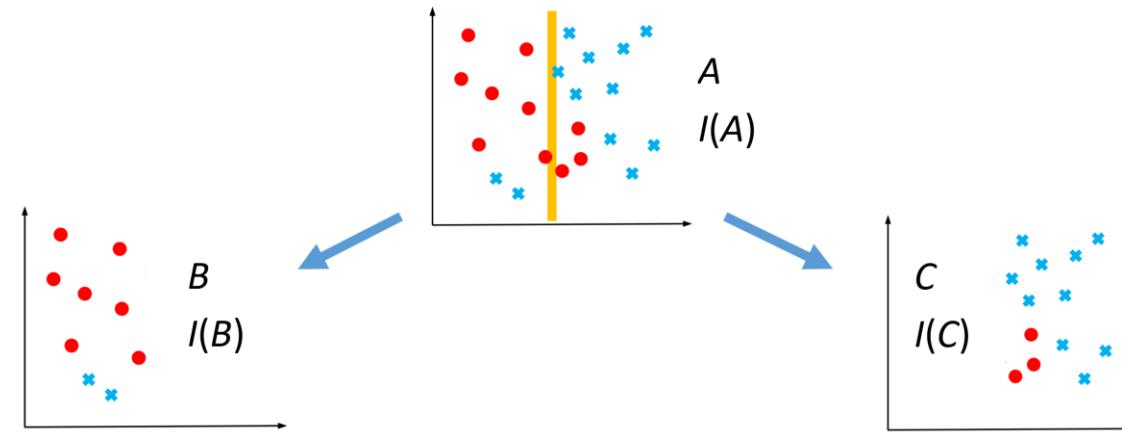
Wykres miar zanieczyszczenia węzła dla dwóch klas



gdzie p_k jest proporcją liczby przykładów z klasy k ($N_{A,k}$) do liczby wszystkich przykładów w zbiorze A (N_A): $p_k = \frac{N_{A,k}}{N_A}$

Kryterium wyboru testu

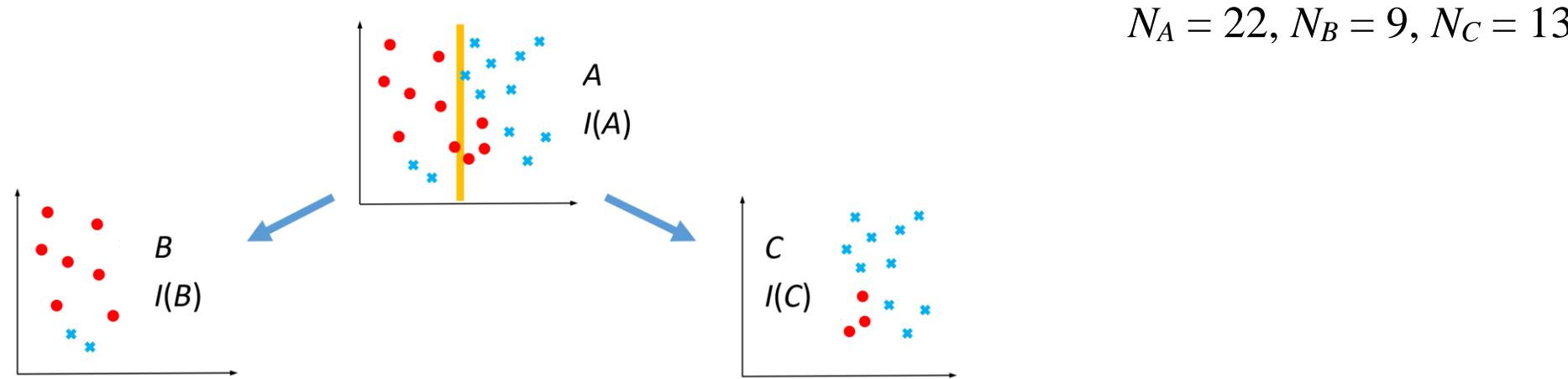
Przyrost informacji wynikający z zastosowania testu t do podziału zbioru przykładów A na podzbiory B i C :



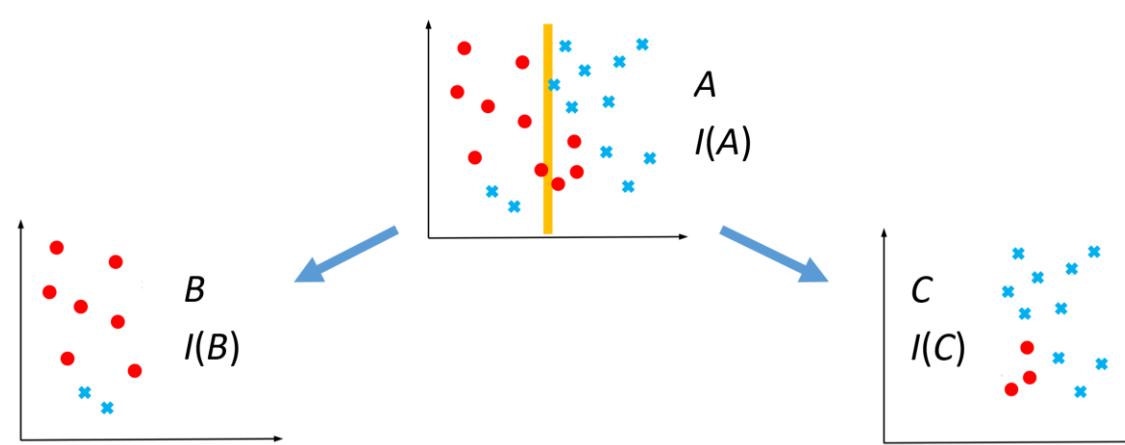
$$\Delta I(A) = I(A) - \frac{N_B}{N_A} I(B) - \frac{N_C}{N_A} I(C)$$

gdzie $I(\cdot)$ jest zanieczyszczeniem węzła, a N oznacza liczbę przykładów w odpowiednim zbiorze.

Kryterium wyboru testu



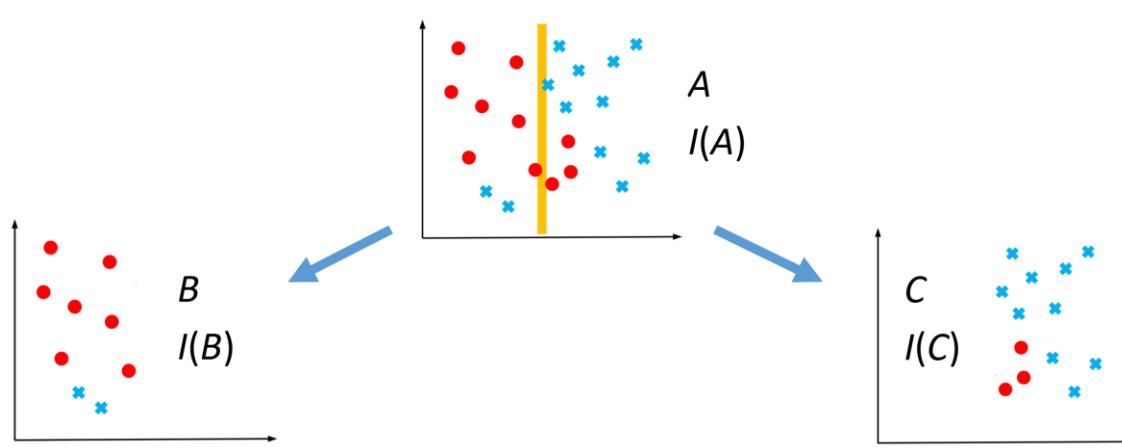
Kryterium wyboru testu



$$N_A = 22, N_B = 9, N_C = 13$$

$$p_{A,1} = \frac{N_{A,1}}{N_A} = \frac{12}{22} = 0.55$$

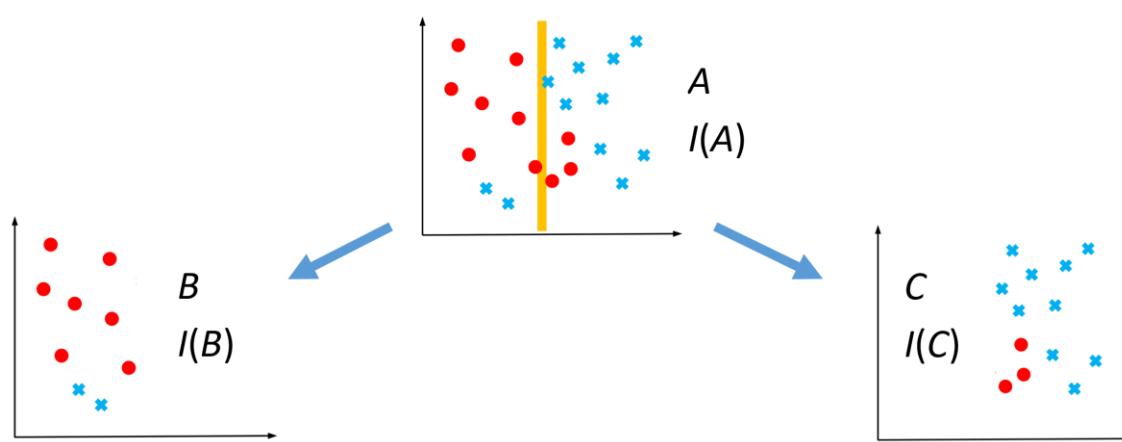
Kryterium wyboru testu



$$N_A = 22, N_B = 9, N_C = 13$$

$$p_{A,1} = \frac{N_{A,1}}{N_A} = \frac{12}{22} = 0.55, \quad p_{A,1} = \frac{N_{A,1}}{N_A} = \frac{12}{22} = 0.55$$

Kryterium wyboru testu

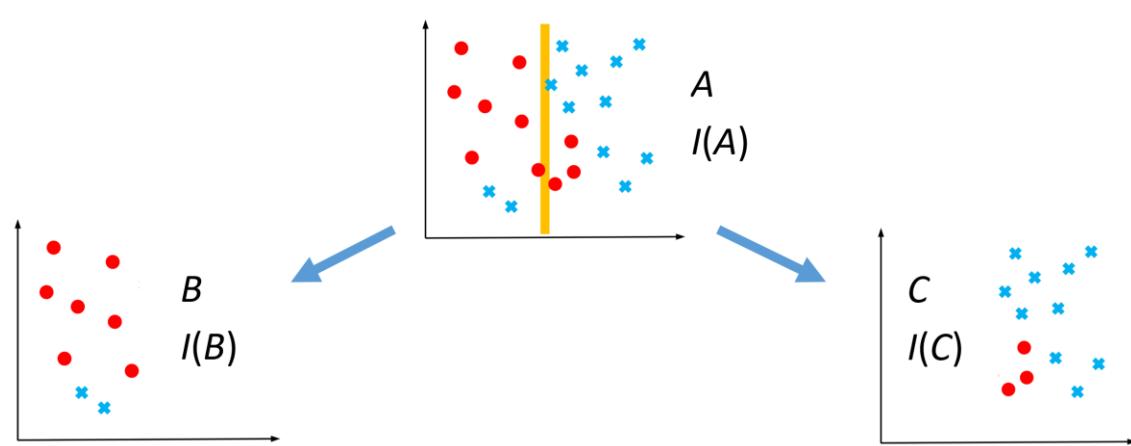


$$N_A = 22, N_B = 9, N_C = 13$$

$$p_{A,1} = \frac{N_{A,1}}{N_A} = \frac{12}{22} = 0.55, \quad p_{A,1} = \frac{N_{A,1}}{N_A} = \frac{12}{22} = 0.55$$

$$p_{B,1} = \frac{N_{B,1}}{N_B} = \frac{2}{9} = 0.21, \quad p_{B,1} = \frac{N_{B,1}}{N_B} = \frac{2}{9} = 0.21$$

Kryterium wyboru testu



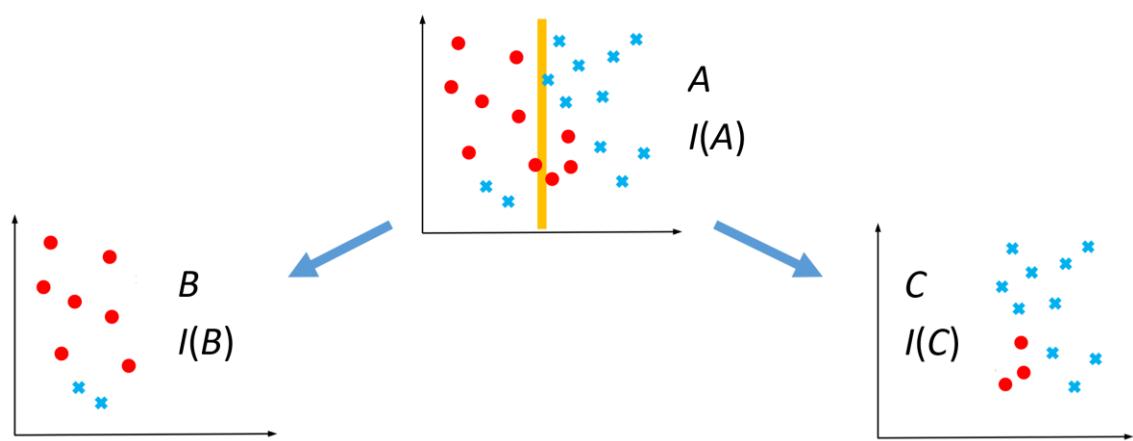
$$N_A = 22, N_B = 9, N_C = 13$$

$$p_{A,1} = \frac{N_{A,1}}{N_A} = \frac{12}{22} = 0.55, \quad p_{A,1} = \frac{N_{A,1}}{N_A} = \frac{12}{22} = 0.55$$

$$p_{A,2} = \frac{N_{A,2}}{N_A} = \frac{10}{22} = 0.45, \quad p_{B,2} = \frac{N_{B,2}}{N_B} = \frac{7}{9} = 0.89$$

$$p_{B,2} = \frac{N_{B,2}}{N_B} = \frac{7}{9} = 0.89, \quad p_{C,2} = \frac{N_{C,2}}{N_C} = \frac{3}{13} = 0.23$$

Kryterium wyboru testu



$$N_A = 22, N_B = 9, N_C = 13$$

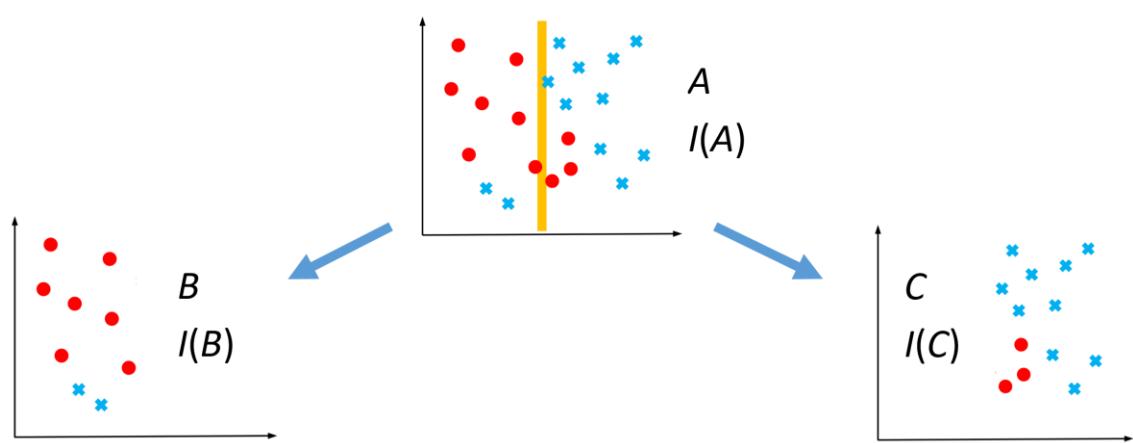
$$p_{A,1} = \frac{N_{A,1}}{N_A} = \frac{12}{22} = 0.55, \quad p_{A,1} = \frac{N_{A,1}}{N_A} = \frac{12}{22} = 0.55$$

$$p_{A,2} = \frac{N_{A,2}}{N_A} = \frac{10}{22} = 0.45, \quad p_{B,1} = \frac{N_{B,1}}{N_B} = \frac{2}{9} = 0.21$$

$$p_{B,2} = \frac{N_{B,2}}{N_B} = \frac{7}{9} = 0.89, \quad p_{C,2} = \frac{N_{C,2}}{N_C} = \frac{3}{13} = 0.23$$

$$I_E(A) = -p_{A,1} \log_2 p_{A,1} - p_{A,2} \log_2 p_{A,2} = -0.55 \log_2 0.55 - 0.45 \log_2 0.45 = 0.99$$

Kryterium wyboru testu



$$N_A = 22, N_B = 9, N_C = 13$$

$$p_{A,1} = \frac{N_{A,1}}{N_A} = \frac{12}{22} = 0.55, \quad p_{A,1} = \frac{N_{A,1}}{N_A} = \frac{12}{22} = 0.55$$

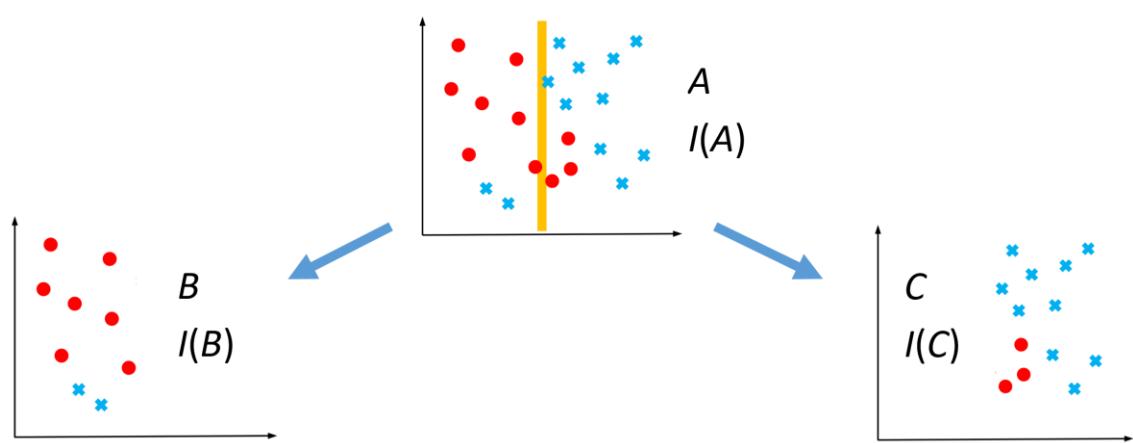
$$p_{A,2} = \frac{N_{A,2}}{N_A} = \frac{10}{22} = 0.45, \quad p_{B,1} = \frac{N_{B,1}}{N_B} = \frac{2}{9} = 0.21$$

$$p_{B,2} = \frac{N_{B,2}}{N_B} = \frac{7}{9} = 0.89, \quad p_{C,2} = \frac{N_{C,2}}{N_C} = \frac{3}{13} = 0.23$$

$$I_E(A) = -p_{A,1} \log_2 p_{A,1} - p_{A,2} \log_2 p_{A,2} = -0.55 \log_2 0.55 - 0.45 \log_2 0.45 = 0.99$$

$$I_E(B) = -p_{B,1} \log_2 p_{B,1} - p_{B,2} \log_2 p_{B,2} = -0.21 \log_2 0.21 - 0.89 \log_2 0.89 = 0.62$$

Kryterium wyboru testu



$$N_A = 22, N_B = 9, N_C = 13$$

$$p_{A,1} = \frac{N_{A,1}}{N_A} = \frac{12}{22} = 0.55, \quad p_{A,1} = \frac{N_{A,1}}{N_A} = \frac{12}{22} = 0.55$$

$$p_{A,2} = \frac{N_{A,2}}{N_A} = \frac{10}{22} = 0.45, \quad p_{B,1} = \frac{N_{B,1}}{N_B} = \frac{2}{9} = 0.21$$

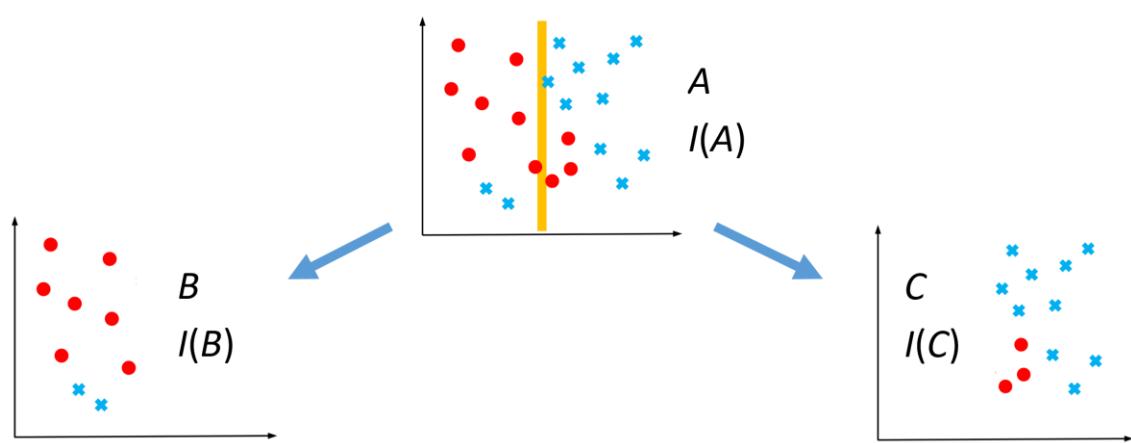
$$p_{B,2} = \frac{N_{B,2}}{N_B} = \frac{7}{9} = 0.89, \quad p_{C,2} = \frac{N_{C,2}}{N_C} = \frac{3}{13} = 0.23$$

$$I_E(A) = -p_{A,1} \log_2 p_{A,1} - p_{A,2} \log_2 p_{A,2} = -0.55 \log_2 0.55 - 0.45 \log_2 0.45 = 0.99$$

$$I_E(B) = -p_{B,1} \log_2 p_{B,1} - p_{B,2} \log_2 p_{B,2} = -0.21 \log_2 0.21 - 0.89 \log_2 0.89 = 0.62$$

$$I_E(C) = -p_{C,1} \log_2 p_{C,1} - p_{C,2} \log_2 p_{C,2} = -0.77 \log_2 0.77 - 0.23 \log_2 0.23 = 0.79$$

Kryterium wyboru testu



$$N_A = 22, N_B = 9, N_C = 13$$

$$p_{A,1} = \frac{N_{A,1}}{N_A} = \frac{12}{22} = 0.55, \quad p_{A,1} = \frac{N_{A,1}}{N_A} = \frac{12}{22} = 0.55$$

$$p_{B,1} = \frac{N_{B,1}}{N_B} = \frac{2}{9} = 0.21, \quad p_{B,1} = \frac{N_{B,1}}{N_B} = \frac{2}{9} = 0.21$$

$$p_{B,2} = \frac{N_{B,2}}{N_B} = \frac{7}{9} = 0.89, \quad p_{C,2} = \frac{N_{C,2}}{N_C} = \frac{3}{13} = 0.23$$

$$I_E(A) = -p_{A,1} \log_2 p_{A,1} - p_{A,2} \log_2 p_{A,2} = -0.55 \log_2 0.55 - 0.45 \log_2 0.45 = 0.99$$

$$I_E(B) = -p_{B,1} \log_2 p_{B,1} - p_{B,2} \log_2 p_{B,2} = -0.21 \log_2 0.21 - 0.89 \log_2 0.89 = 0.62$$

$$I_E(C) = -p_{C,1} \log_2 p_{C,1} - p_{C,2} \log_2 p_{C,2} = -0.77 \log_2 0.77 - 0.23 \log_2 0.23 = 0.79$$

$$\Delta I(A) = I(A) - \frac{N_B}{N_A} I(B) - \frac{N_C}{N_A} I(C) = 0.99 - \frac{9}{22} 0.62 - \frac{13}{22} 0.79 = 0.27$$

Kryterium wyboru testu

Wybierz test maksymalizujący przyrost informacji

$$\arg \max_t \Delta I_t(A)$$

Kryterium wyboru testu

Przykład.

Dany jest zbiór przykładów opisanych atrybutem ciągłym *masa* i nominalnym *kolor*.

Nr przykładu	1	2	3	4	5	6	7	8	9	10
Masa	2	6	4	8	7	6	1	6	8	8
Kolor	R	B	R	G	G	R	B	B	B	G
Klasa	0	0	1	1	1	0	0	1	1	0

Ustal zbiór testów kandydujących S i wybierz najlepszy test.

Kryterium wyboru testu

Przykład.

Dany jest zbiór przykładów opisanych atrybutem ciągłym *masa* i nominalnym *kolor*.

Nr przykładu	1	2	3	4	5	6	7	8	9	10
Masa	2	6	4	8	7	6	1	6	8	8
Kolor	R	B	R	G	G	R	B	B	B	G
Klasa	0	0	1	1	1	0	0	1	1	0

Ustal zbiór testów kandydujących *S* i wybierz najlepszy test.

Testy dla *kolor*: *kolor = R*, *kolor = G*, *kolor = B*.

Kryterium wyboru testu

Przykład.

Dany jest zbiór przykładów opisanych atrybutem ciągłym *masa* i nominalnym *kolor*.

Nr przykładu	1	2	3	4	5	6	7	8	9	10
Masa	2	6	4	8	7	6	1	6	8	8
Kolor	R	B	R	G	G	R	B	B	B	G
Klasa	0	0	1	1	1	0	0	1	1	0

Ustal zbiór testów kandydujących *S* i wybierz najlepszy test.

Testy dla *kolor*: *kolor = R*, *kolor = G*, *kolor = B*.

Testy dla *masa*:

1. Sortujemy wartości *masy*: 1 | 2 | 4 | 6 6 6 | 7 | 8 8 8
2. Wyznaczamy progi *p*: 1.5, 3, 5, 6.5, 7.5
3. Testy kandydujące: *masa ≤ 1.5*, *masa ≤ 3*, *masa ≤ 5*, *masa ≤ 6.5*, *masa ≤ 7.5*

Kryterium wyboru testu

Przykład.

Dany jest zbiór przykładów opisanych atrybutem ciągłym *masa* i nominalnym *kolor*.

Nr przykładu	1	2	3	4	5	6	7	8	9	10
Masa	2	6	4	8	7	6	1	6	8	8
Kolor	R	B	R	G	G	R	B	B	B	G
Klasa	0	0	1	1	1	0	0	1	1	0

Ustal zbiór testów kandydujących S i wybierz najlepszy test.

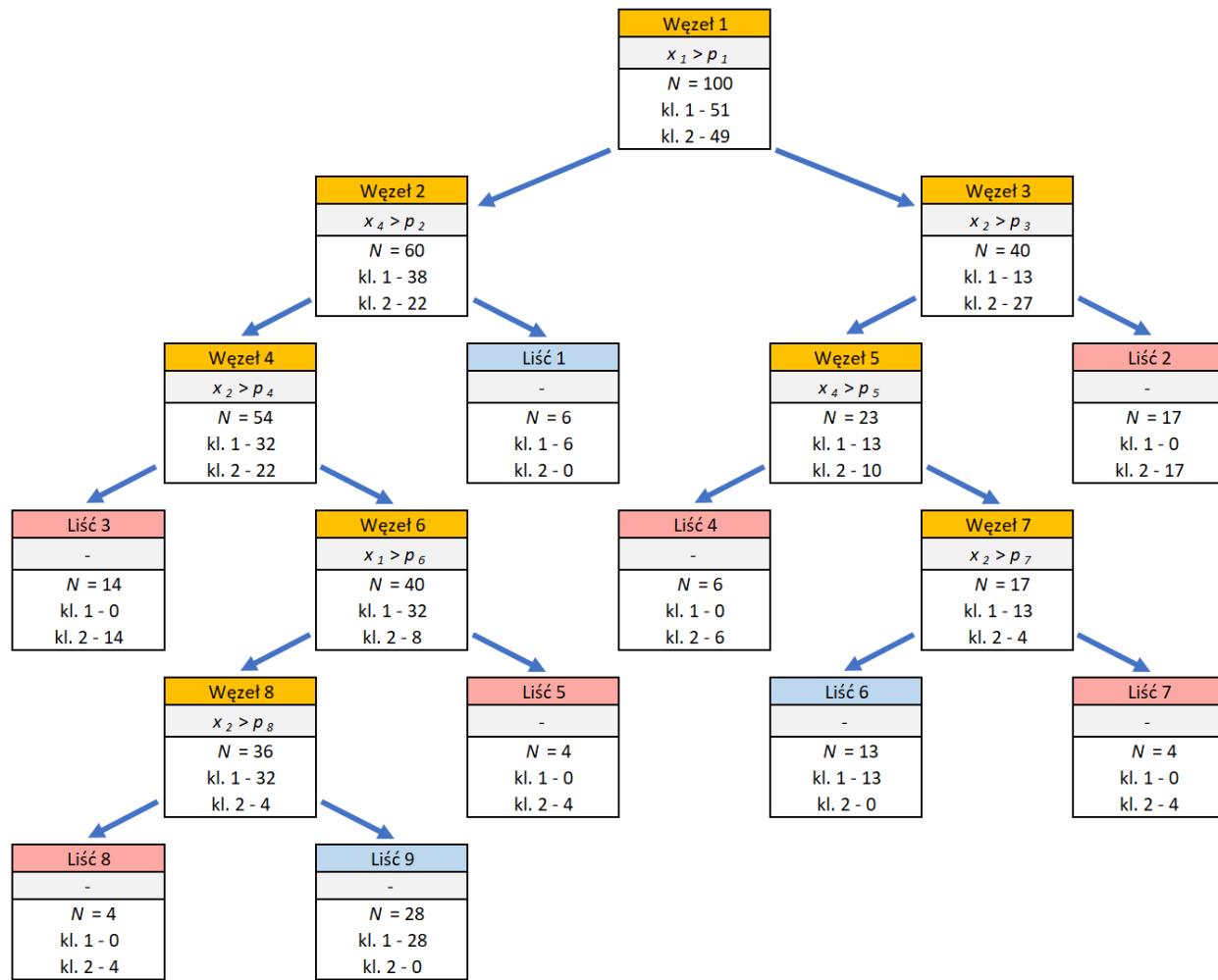
Testy dla kolor: *kolor = R*, *kolor = G*, *kolor = B*.

Testy dla *masa*:

1. Sortujemy wartości $masy$: 1 | 2 | 4 | 6 6 6 | 7 | 8 8 8
 2. Wyznaczamy progi p : 1.5, 3, 5, 6.5, 7.5
 3. Testy kandydujące: $masa \leq 1.5$, $masa \leq 3$, $masa \leq 5$, $masa \leq 6.5$, $masa \leq 7.5$

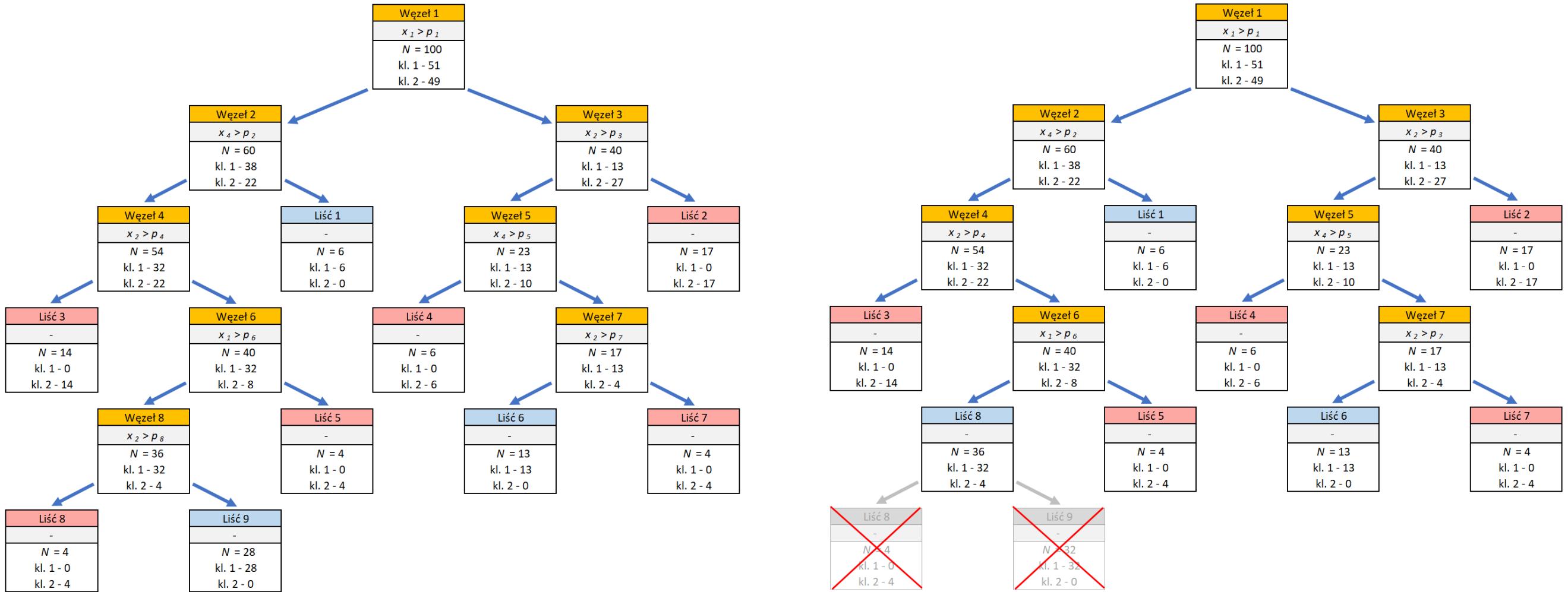
Przycinanie drzewa

Przycinanie drzewa – zastępowanie wybranych węzłów (a tym samym całych poddrzew) liśćmi, którym przypisuje się kategorię większości przykładów uczących, związanych z eliminowanym węzłem.



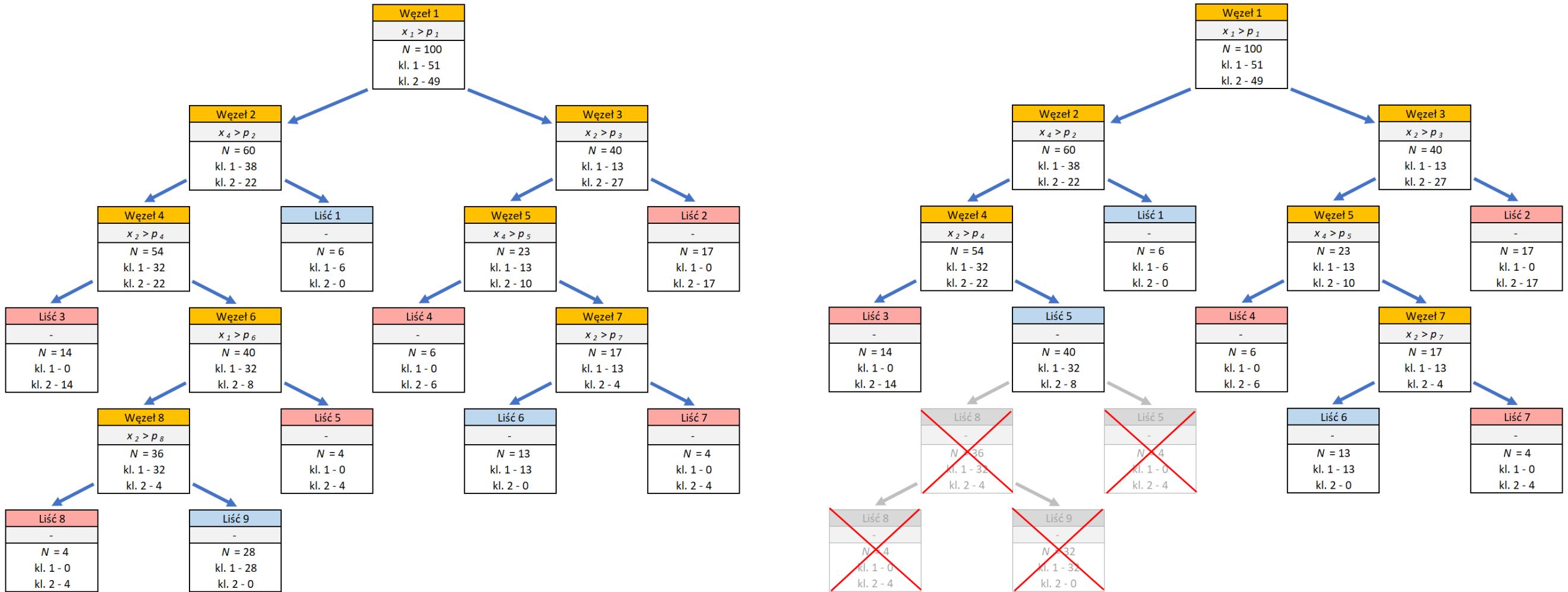
Przycinanie drzewa

Przycinanie drzewa – zastępowanie wybranych węzłów (a tym samym całych poddrzew) liśćmi, którym przypisuje się kategorię większości przykładów uczących, związanych z eliminowanym węzłem.



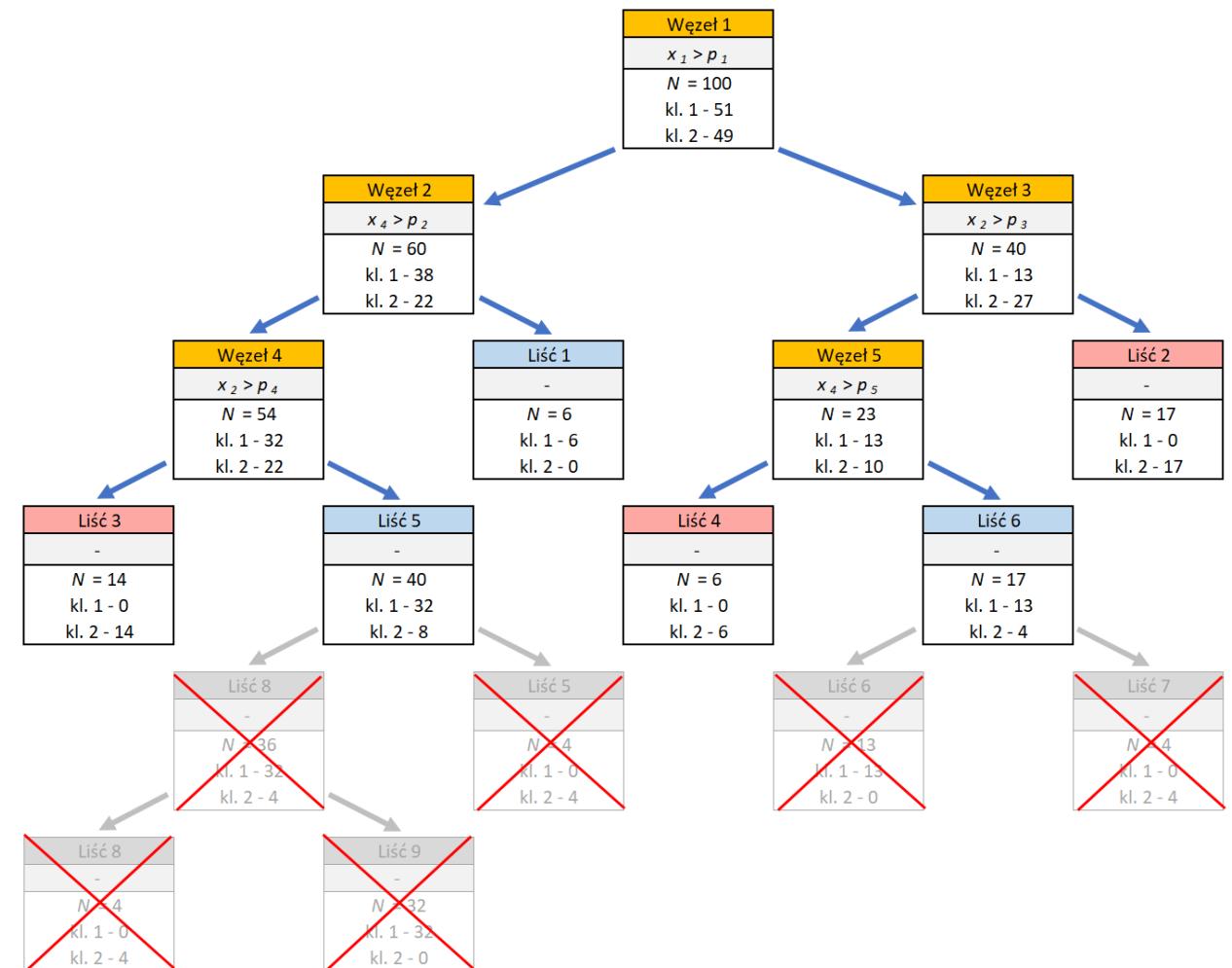
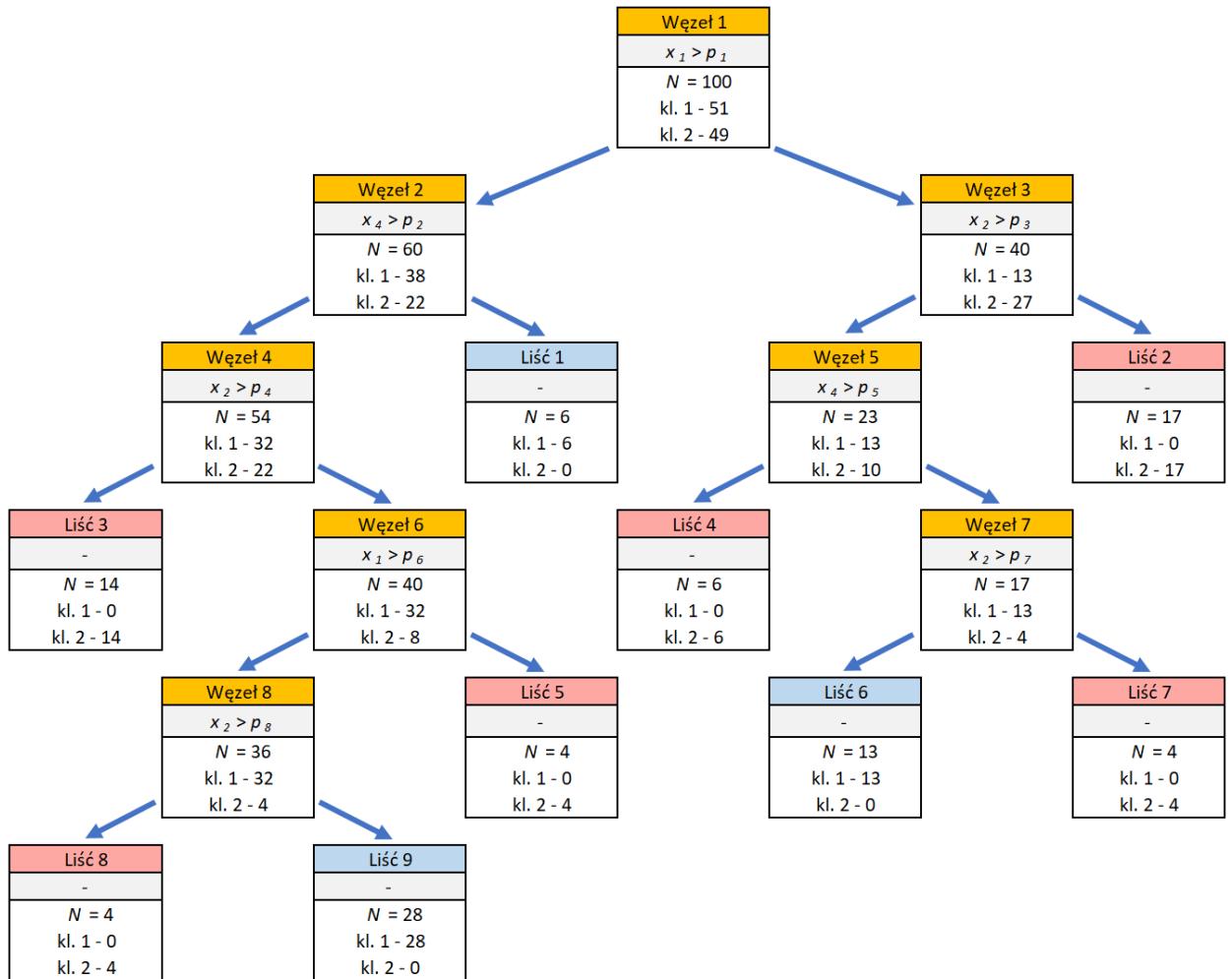
Przycinanie drzewa

Przycinanie drzewa – zastępowanie wybranych węzłów (a tym samym całych poddrzew) liśćmi, którym przypisuje się kategorię większości przykładów uczących, związanych z eliminowanym węzłem.



Przycinanie drzewa

Przycinanie drzewa – zastępowanie wybranych węzłów (a tym samym całych poddrzew) liśćmi, którym przypisuje się kategorię większości przykładów uczących, związanych z eliminowanym węzłem.

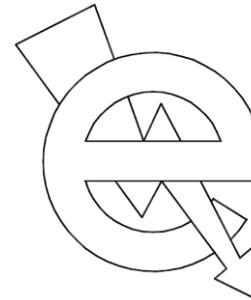


Przycinanie drzewa

Przycinanie drzewa – zastępowanie wybranych węzłów (a tym samym całych poddrzew) liśćmi, którym przypisuje się kategorię większości przykładów uczących, związanych z eliminowanym węzłem.

- Przycinanie drzewa redukuje efekt przeuczenia (mały błąd uczący, duży testowy).
- Alternatywą przycinania jest zapobieganie nadmiernemu wzrostowi w trakcie konstrukcji drzewa poprzez tworzenie liści w miejsce węzłów (z etykietą kategorii większościowej).
- Zazwyczaj przycinanie realizowane jest w sposób wstępujący, tzn. w pierwszej kolejności rozważa się przycięcie węzłów położonych najniżej w drzewie. Podstawowe znaczenie ma kryterium przycinania, które decyduje, czy węzeł będzie zastąpiony liściem.
- Najczęściej kryterium przycinania jest błąd na oddzielnym zbiorze przykładów (niezależnym od zbioru trenującego). Przycięcie następuje, jeśli na tym oddzielnym zbiorze liść uzyskuje błąd nie większy, niż błąd poddrzewa.

Drzewa klasyfikacyjne. Część 2 – implementacja w Matlabie i aplikacje



Grzegorz Dudek

Wydział Elektryczny

Politechnika Częstochowska

dudek@el.pcz.czest.pl

<https://gdudek.el.pcz.pl>

Praca powstała w ramach projektu finansowanego z programu Ministra Nauki i Szkolnictwa Wyższego pod nazwą „Regionalna Inicjatywa Doskonałości” w latach 2019 – 2022. Nr projektu 020/RID/2018/19

Plan prezentacji

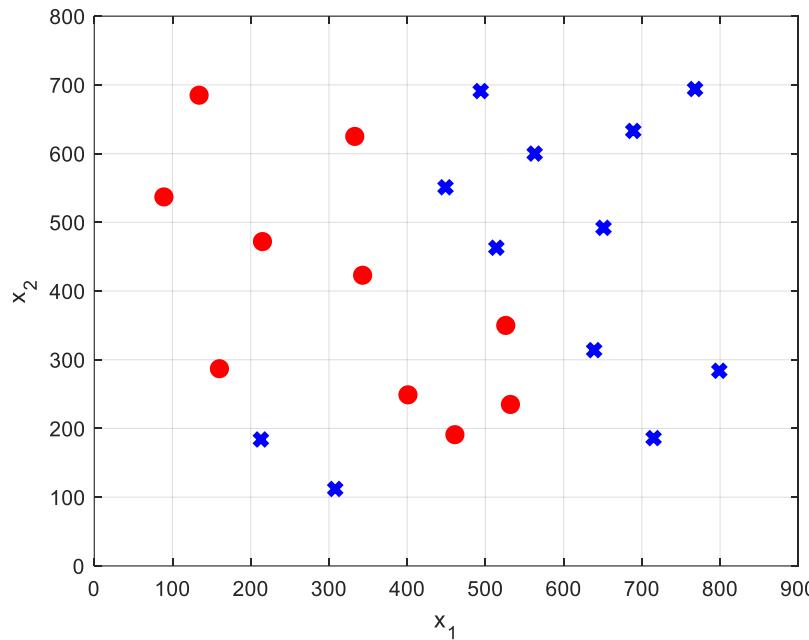
Część 1

1. Informacje wstępne
2. Idea
3. Konstrukcja drzewa klasyfikacyjnego
4. Proces klasyfikacji za pomocą drzewa
5. Testy w węzłach drzewa
6. Przycinanie drzewa

Część 2

7. Drzewo klasyfikacyjne w Matlabie
8. Przykład zastosowania drzewa klasyfikacyjnego
9. Podsumowanie

Drzewo klasyfikacyjne w Matlabie



- Przykłady uczące (współrzędne punktów i ich klasy)

```
x=[134 685;89 537;215 472;333 625;160 287;343 423;401 249;461 191;526 350;532 235;213 184; ...  
308 112;449 551;494 691;514 463;563 600;651 492;689 633;768 694;639 314;715 186;799 284];
```

```
y=[2; 2; 2; 2; 2; 2; 2; 2; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1];
```

Drzewo klasyfikacyjne w Matlabie

- Przykłady uczące (współrzędne punktów i ich klasy)

```
x=[134 685;89 537;215 472;333 625;160 287;343 423;401 249;461 191;526 350;532 235;213 184; ...  
308 112;449 551;494 691;514 463;563 600;651 492;689 633;768 694;639 314;715 186;799 284];
```

```
y=[2; 2; 2; 2; 2; 2; 2; 2; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1];
```

- Utworzenie drzewa klasyfikacyjnego

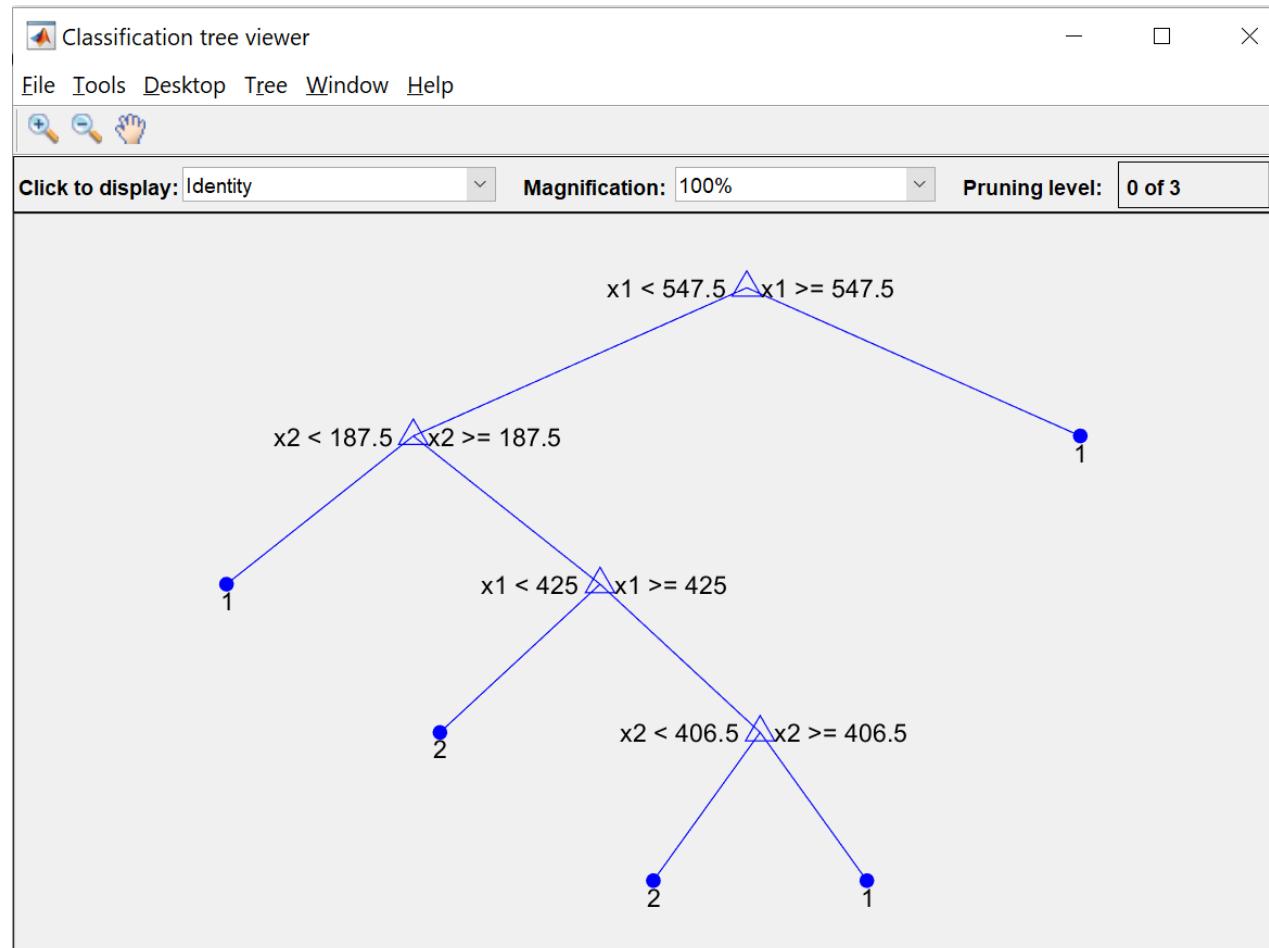
```
tc = fitctree(x,y,'MinParentSize',1);
```

- 'MinParentSize' – minimalna liczba przykładów w węźle nieterminalnym
- 'MinLeafSize' – minimalna liczba przykładów w węźle terminalnym (liściu)
- 'MaxNumSplits' – maksymalna liczba podziałów zbioru danych (węzłów nieterminalnych)

Drzewo klasoryfikacyjne w Matlabie

- Wizualizacja drzewa

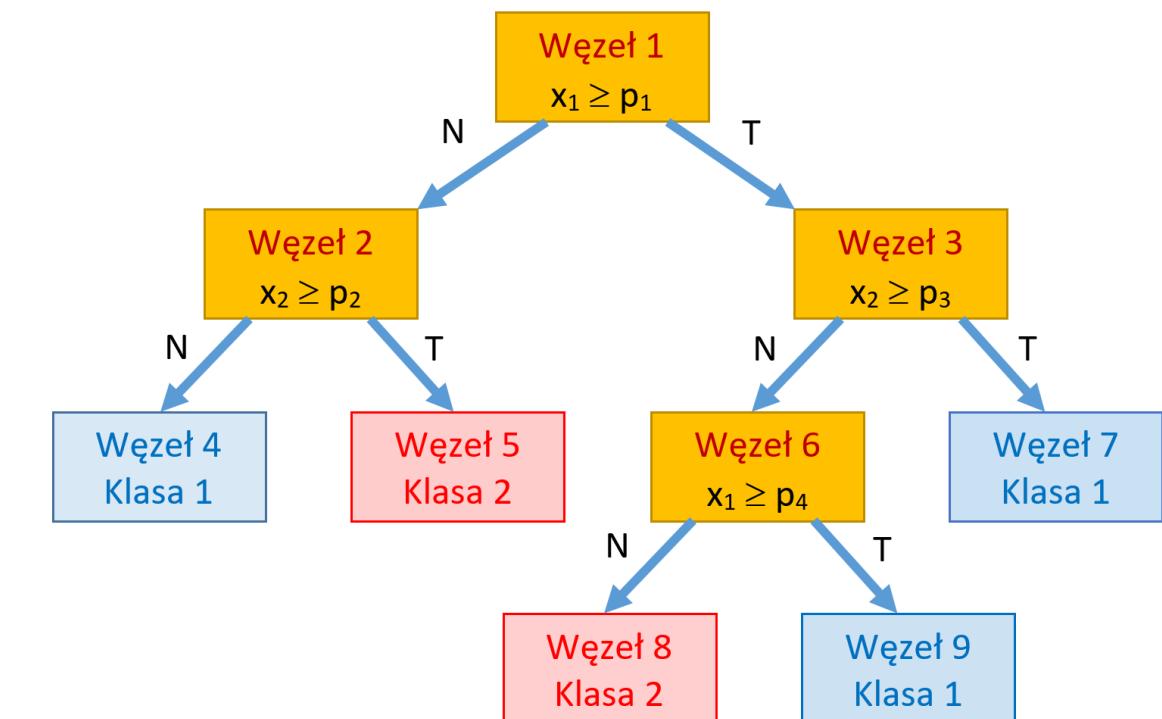
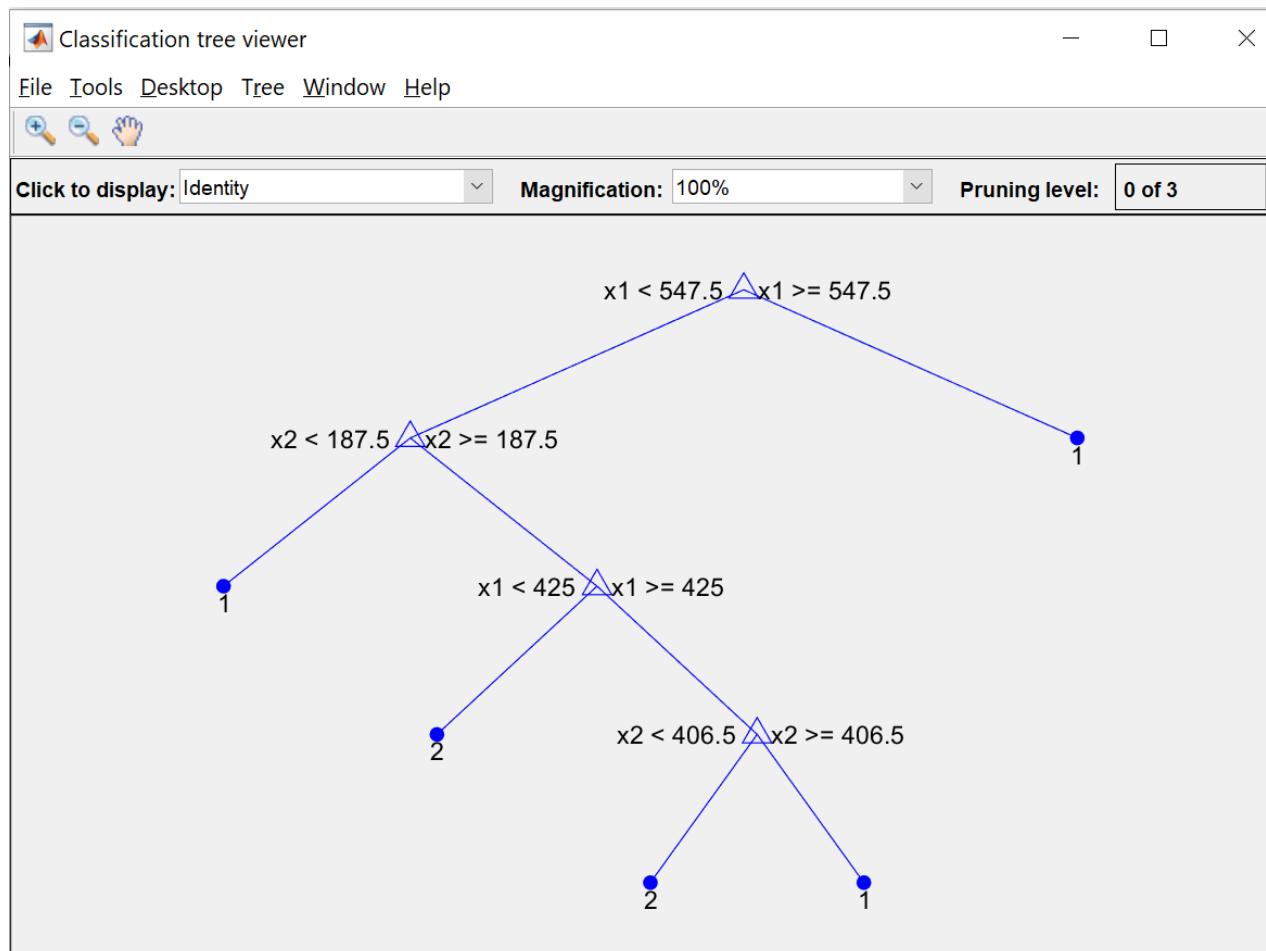
```
view(tc, 'Mode', 'graph')
```



Drzewo klasycfikacyjne w Matlabie

- Wizualizacja drzewa

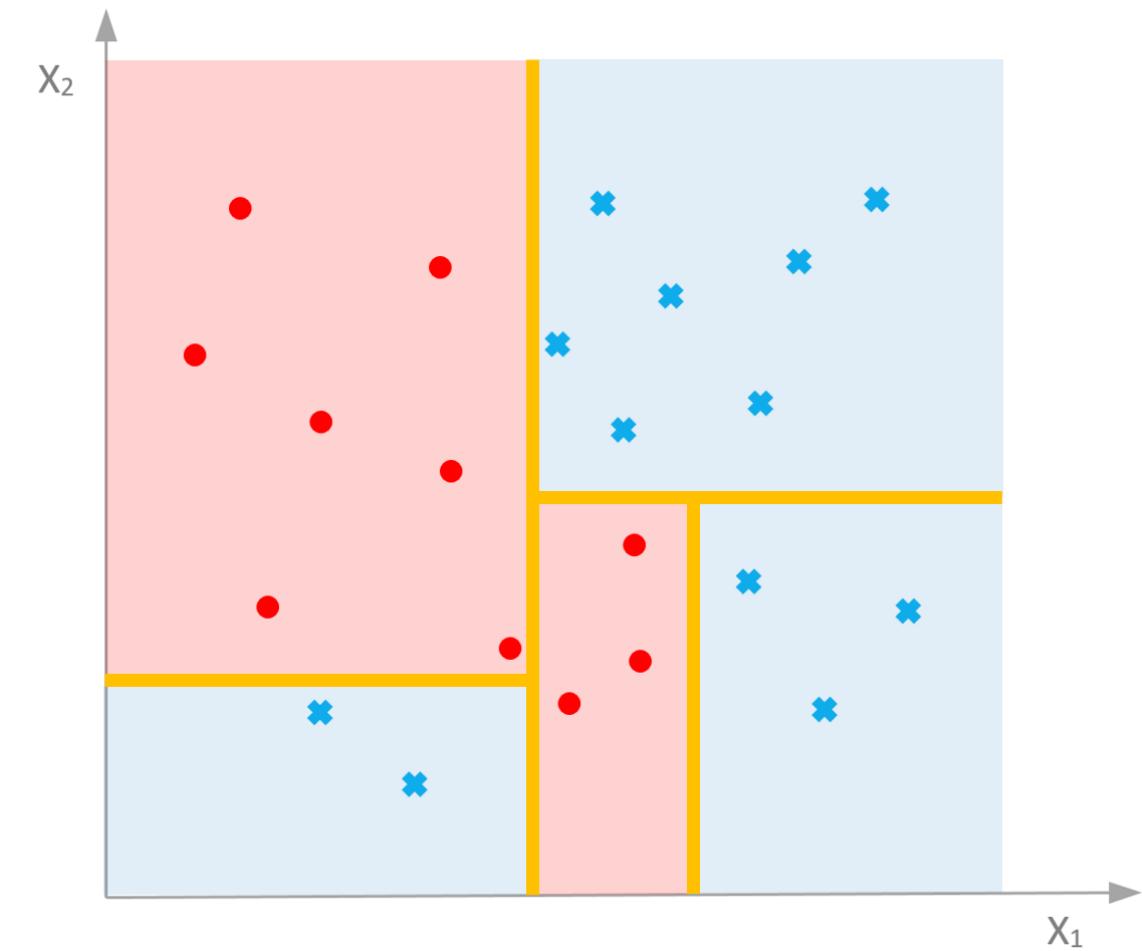
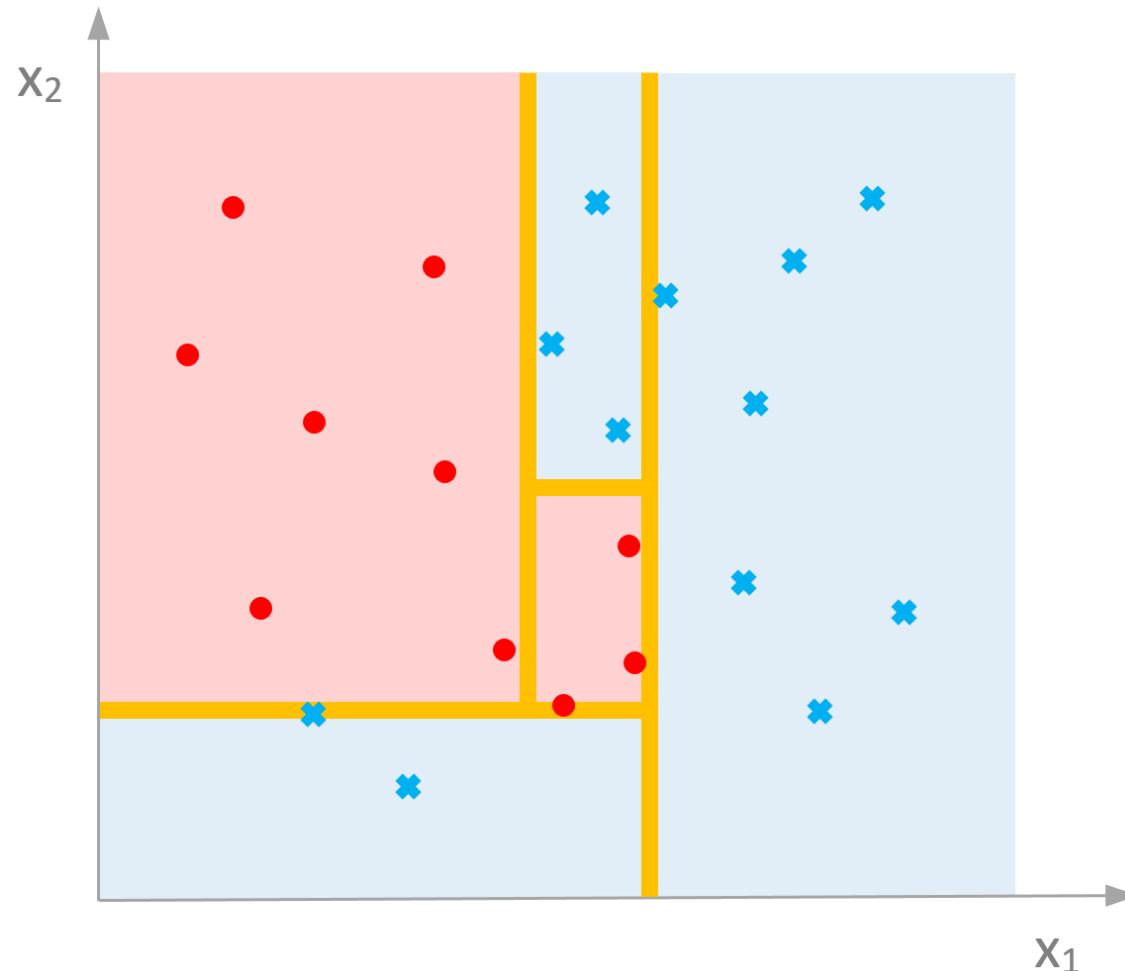
```
view(tc, 'Mode', 'graph')
```



Drzewo klasyfikacyjne w Matlabie

- Wizualizacja drzewa

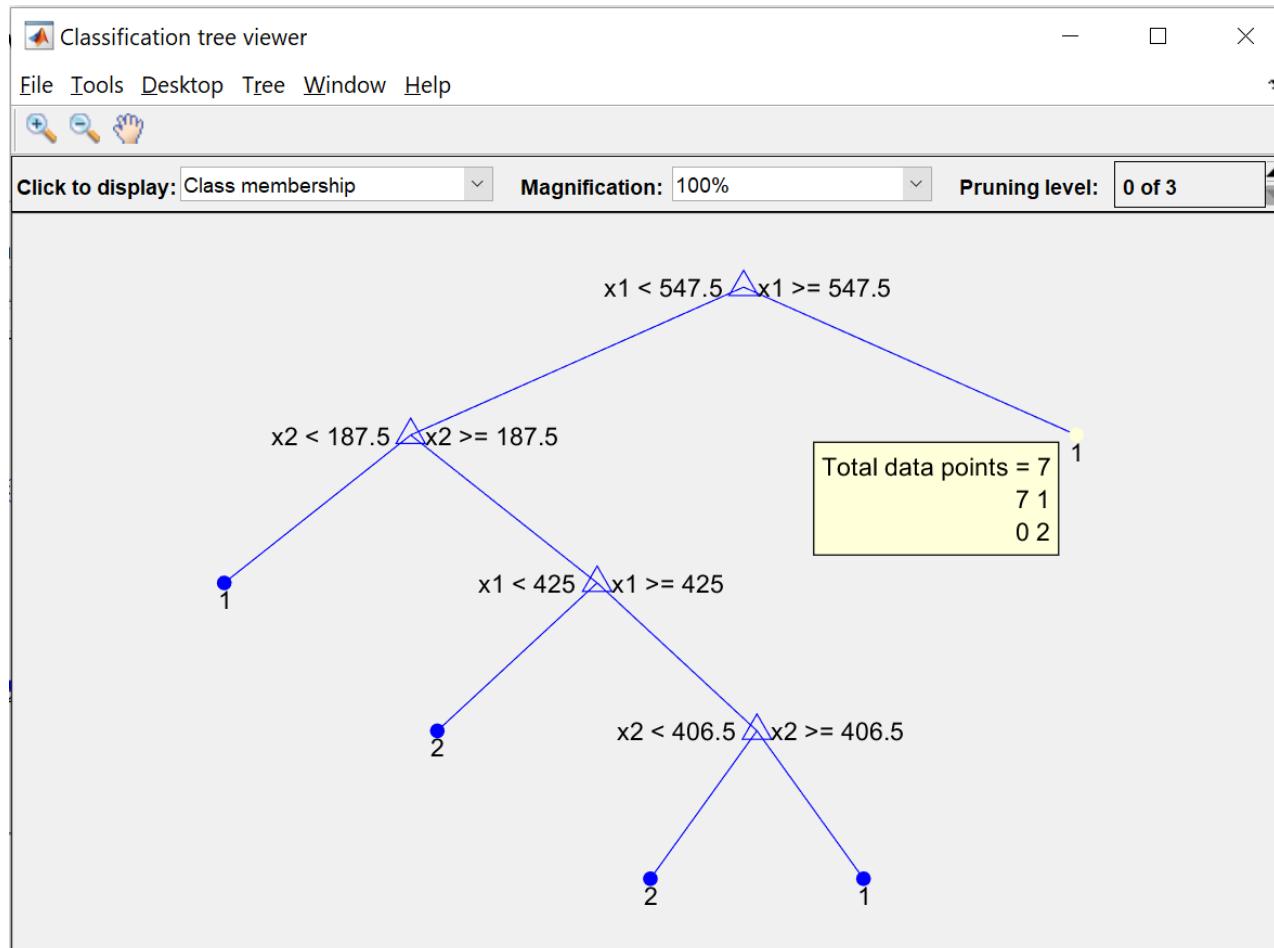
```
view(tc, 'Mode', 'graph')
```



Drzewo klasoryfikacyjne w Matlabie

- Wizualizacja drzewa

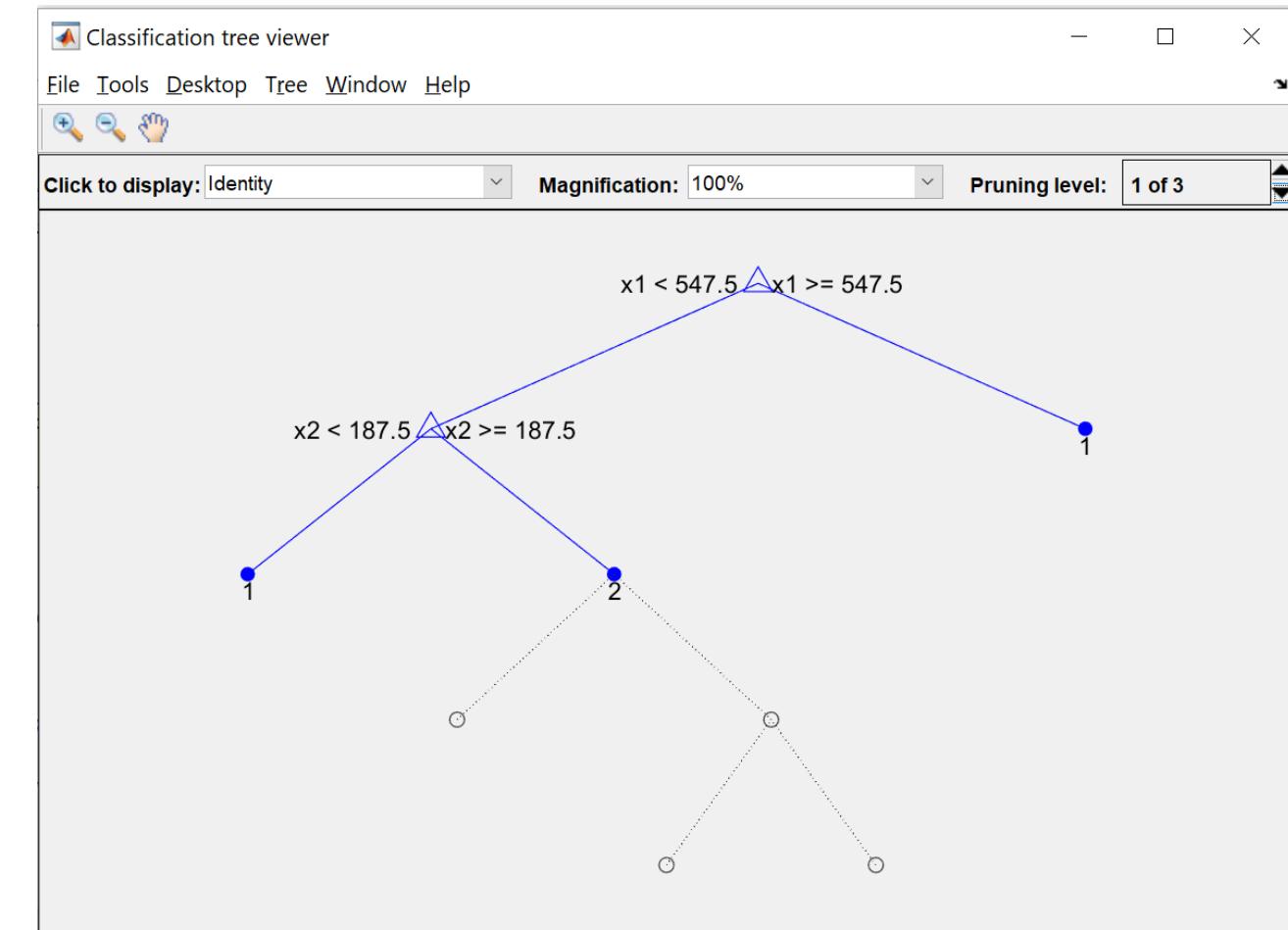
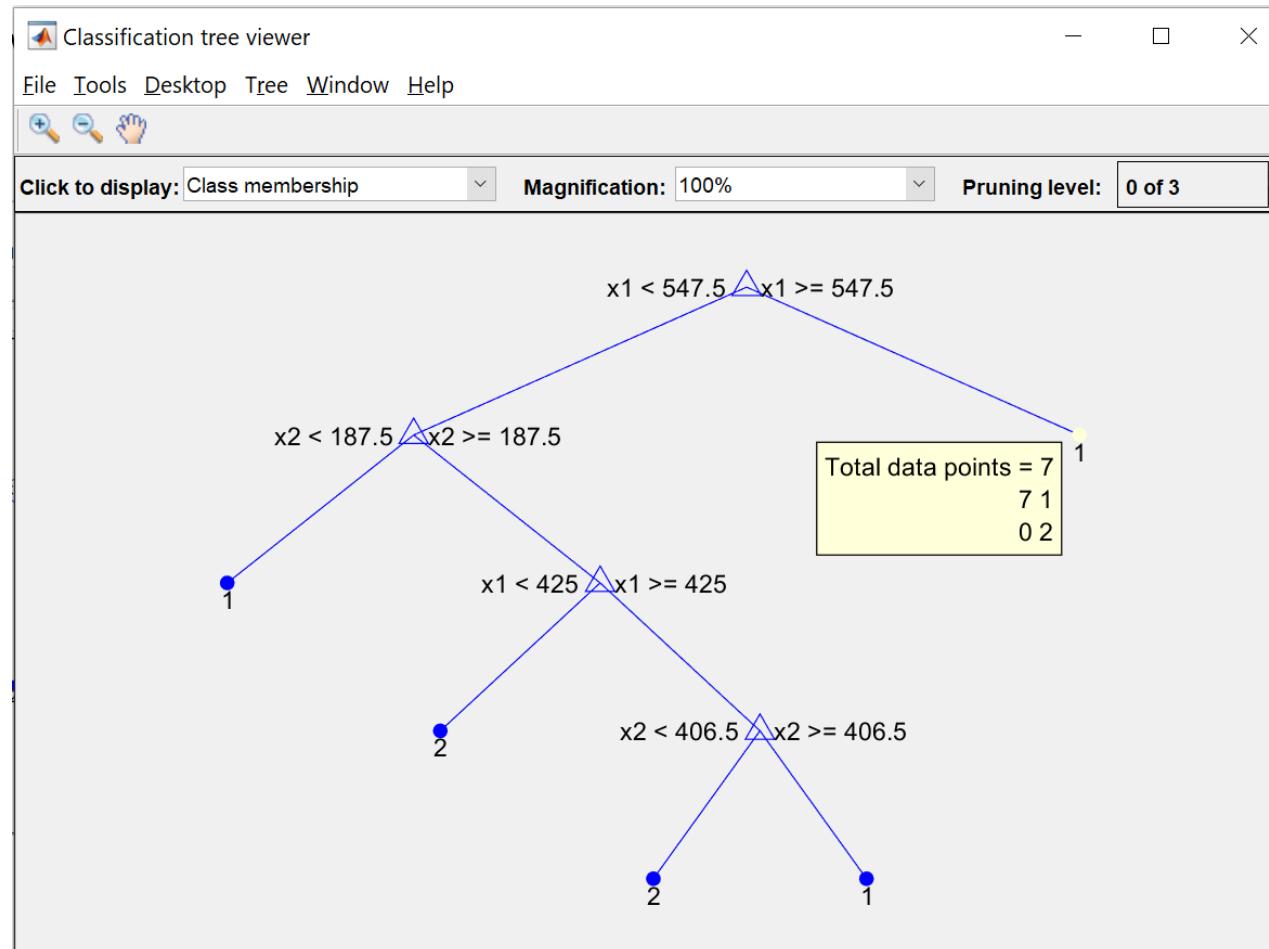
```
view(tc, 'Mode', 'graph')
```



Drzewo klasoryfikacyjne w Matlabie

- Wizualizacja drzewa

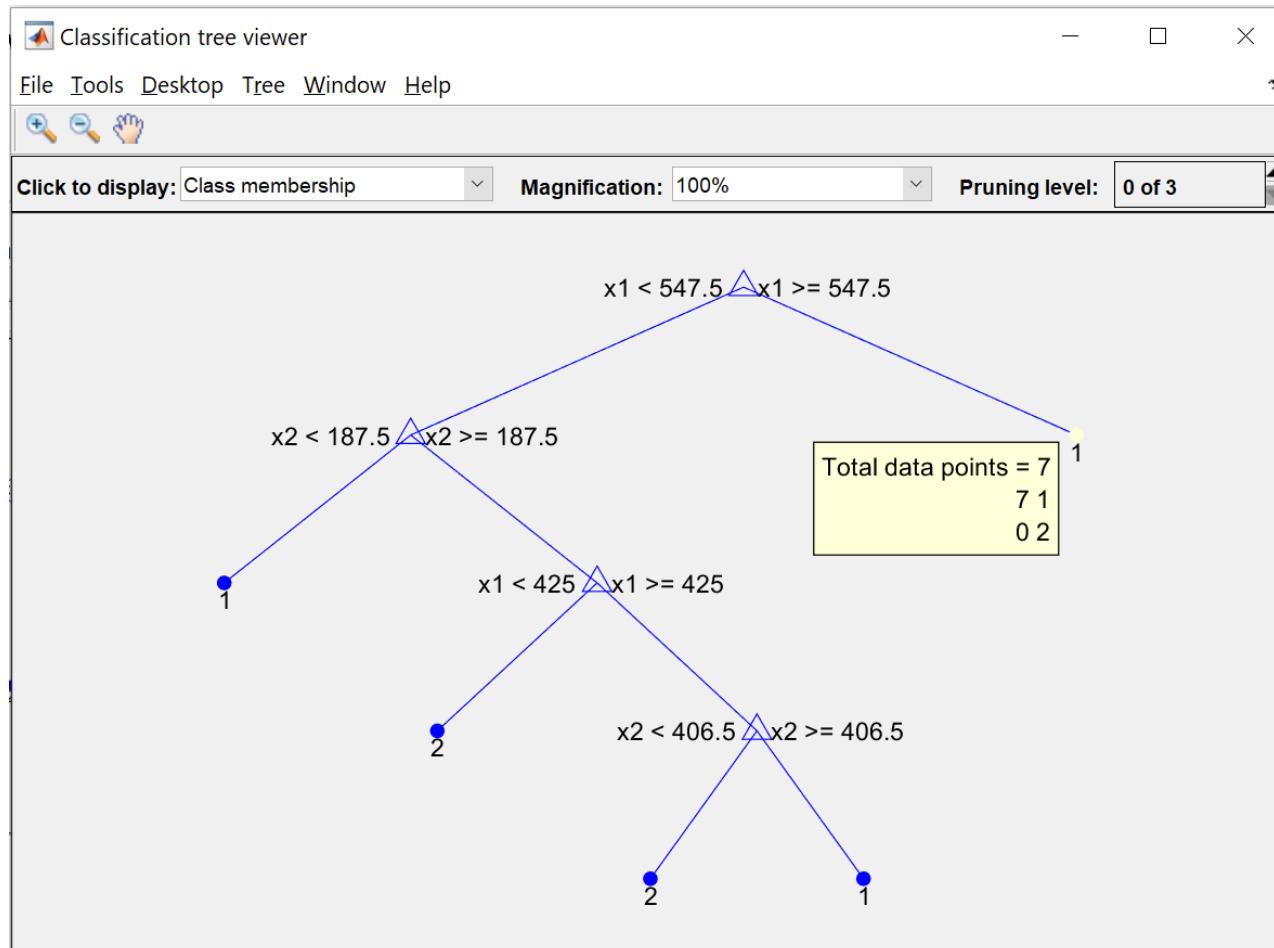
```
view(tc, 'Mode', 'graph')
```



Drzewo klasoryfikacyjne w Matlabie

- Wizualizacja drzewa

```
view(tc, 'Mode', 'graph')
```



```
view(tc)
```

```
1 if x1<547.5 then node 2 elseif x1>=547.5 then node 3 else 1
2 if x2<187.5 then node 4 elseif x2>=187.5 then node 5 else 2
3 class = 1
4 class = 1
5 if x1<425 then node 6 elseif x1>=425 then node 7 else 2
6 class = 2
7 if x2<406.5 then node 8 elseif x2>=406.5 then node 9 else 1
8 class = 2
9 class = 1
```

Drzewo klasyfikacyjne w Matlabie

- Klasyfikacja nowych przykładów

```
Xnew = [530 120; 580 300; 180 600];
```

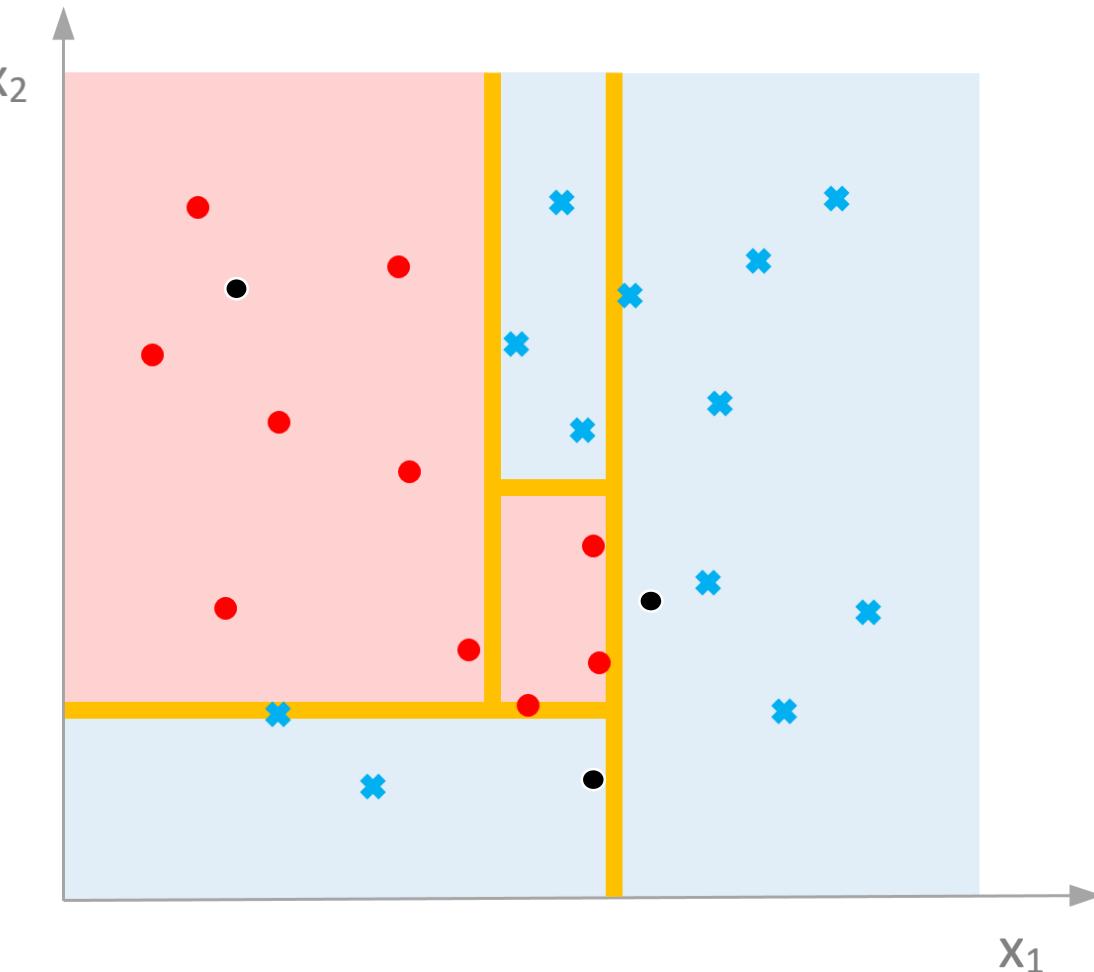
```
Ynew = predict(tc,Xnew)
```

```
Ynew =
```

```
1
```

```
1
```

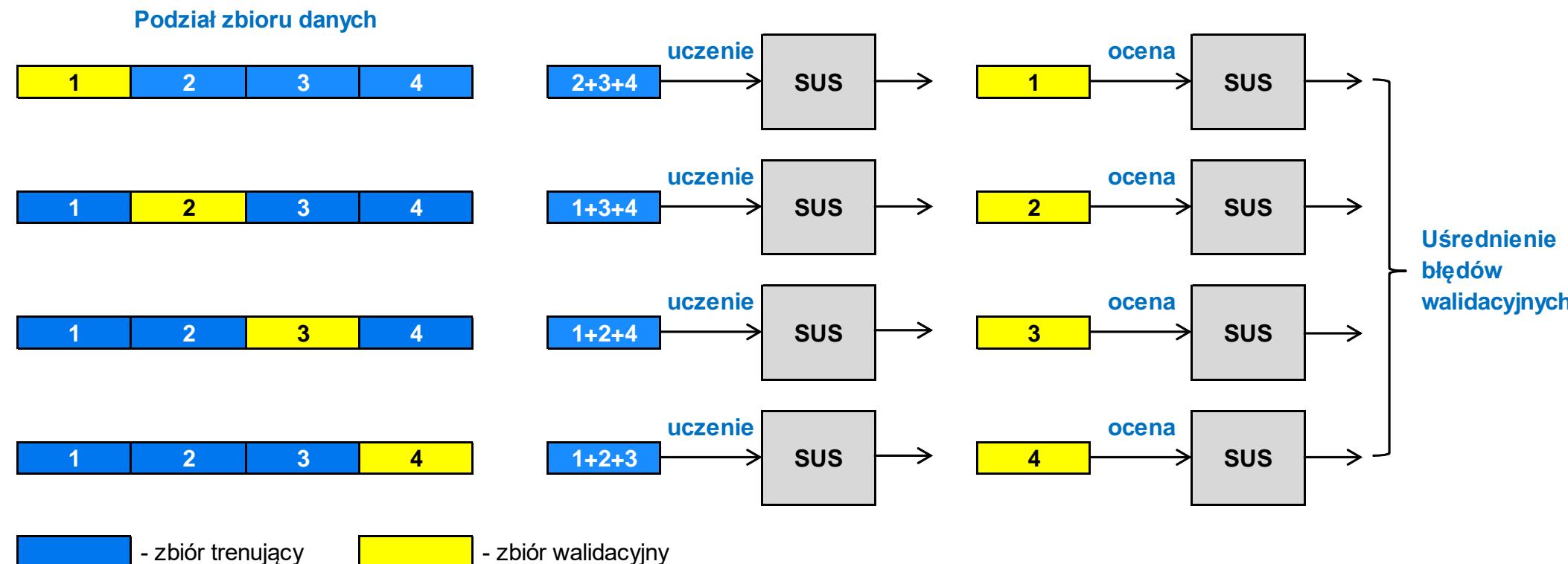
```
2
```



Drzewo klasyfikacyjne w Matlabie

- Błąd klasyfikacji szacowany w procedurze kroswalidacji

Kroswalidacja – procedura uczenia i oceny modelu, w której zbiór przykładów dzieli się losowo na m równolicznych, rozłącznych podzbiorów. Następnie kolejno każdy z tych podzbiorów bierze się jako zbiór walidacyjny, a pozostałe razem jako zbiór trenujący, którym uczy się model. Błąd generalizacji szacuje się uśredniając błędy walidacyjne obliczane po każdej z m sesji uczących.



Drzewo klasyfikacyjne w Matlabie

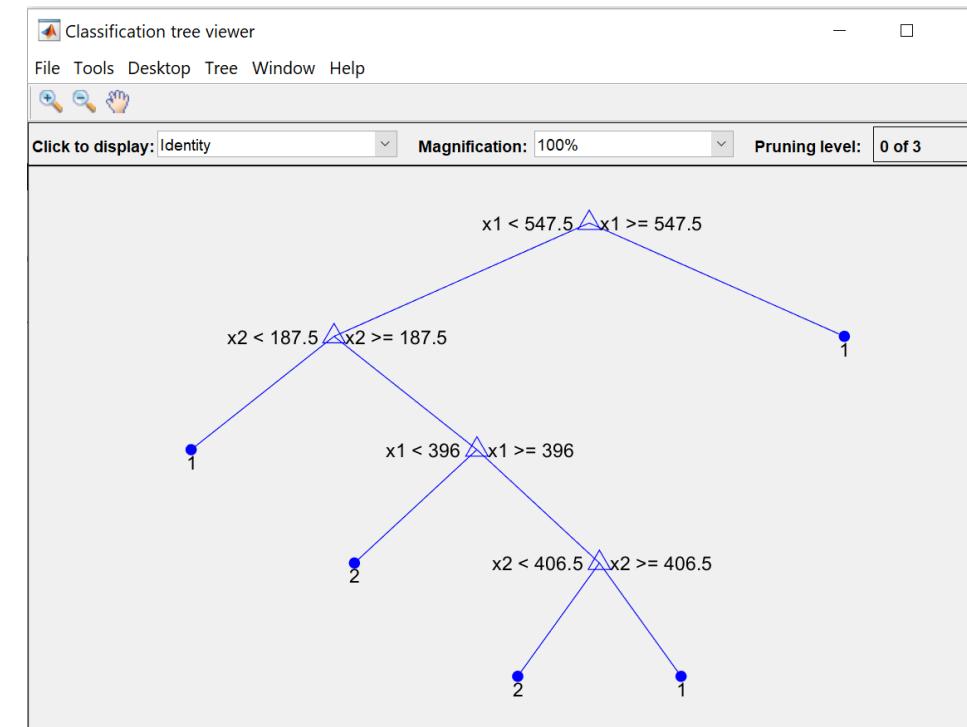
- Błąd klasyfikacji szacowany w procedurze kroswalidacji

```
tccv = fitctree(x,y,'MinParentSize',1,'CrossVal','on');
```

Drzewo klasyfikacyjne w Matlabie

- Błąd klasyfikacji szacowany w procedurze kroswalidacji

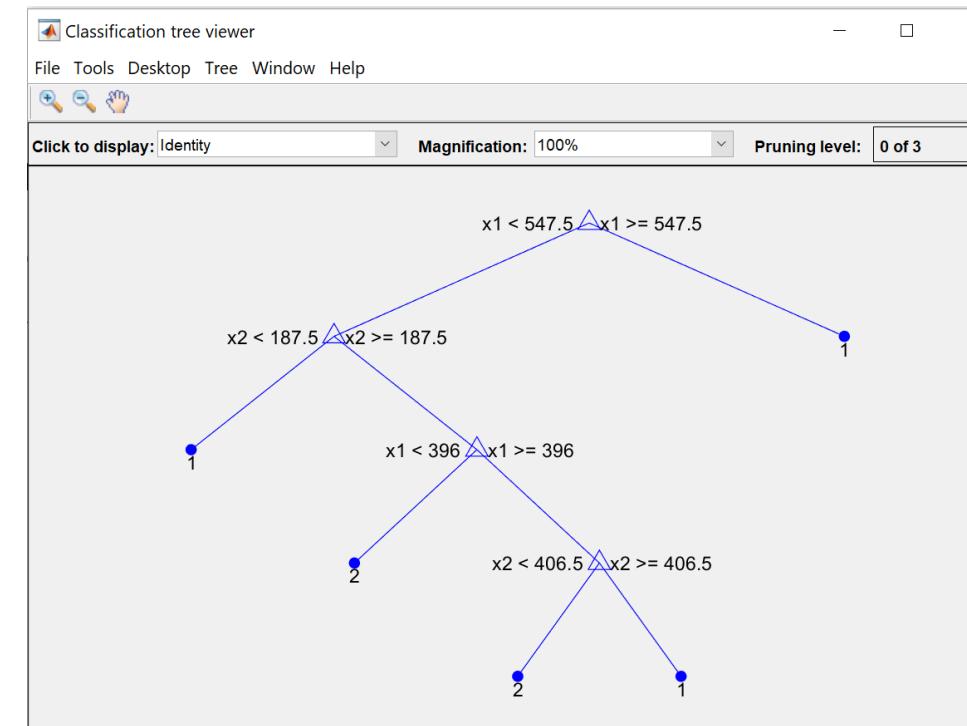
```
tccv = fitctree(x,y,'MinParentSize',1,'CrossVal','on');  
view(tccv.Trained{1}, 'Mode', 'graph')
```



Drzewo klasyfikacyjne w Matlabie

- Błąd klasyfikacji szacowany w procedurze kroswalidacji

```
tccv = fitctree(x,y,'MinParentSize',1,'CrossVal','on');  
view(tccv.Trained{1}, 'Mode', 'graph')
```



```
E = kfoldLoss(tccv)  
E =  
0.2727
```

Drzewo klasyfikacyjne w Matlabie

Przykład. Zaprojektuj drzewo klasyfikacyjne do klasyfikacji zbioru danych Ionosphere.

Drzewo klasyfikacyjne w Matlabie

Przykład. Zaprojektuj drzewo klasyfikacyjne do klasyfikacji zbioru danych Ionosphere.

Załadowanie zbioru danych

```
load ionosphere
```

The screenshot shows the Matlab environment with three main windows:

- Workspace:** Shows variables `Description`, `X`, and `Y`. `Description` is a 5x79 `char` array, `X` is a 351x34 `double` matrix, and `Y` is a 351x1 `cell`.
- Command Window:** Displays the command `>> Description` and its output, which describes the Ionosphere dataset as a binary classification problem with 351 samples, 34 predictors, and two classes ('g' for good, 'b' for bad).
- Data Preview:** Shows two tables. The left table, titled "X", displays the first 17 rows of the 351x34 matrix. The right table, titled "Y", displays the first 17 rows of the 351x1 cell array, where values are converted to text ("g" or "b").

Drzewo klasyfikacyjne w Matlabie

Dopasowanie drzewa

```
ctree = fitctree(X,Y)
```

Property	Value
Y	351x1 cell
X	351x34 double
RowsUsed	[]
W	351x1 double
ModelParameters	1x1 TreeParams
NumObservations	351
HyperparameterOptimizationRe...	[]
PredictorNames	1x34 cell
CategoricalPredictors	[]
ResponseName	'Y'
ExpandedPredictorNames	1x34 cell
ClassNames	2x1 cell
Prior	[0.3590 0.6410]
Cost	[0 1;1 0]
ScoreTransform	'none'
CategoricalSplit	0x0 cell
Children	37x2 double
ClassCount	37x2 double
ClassProbability	37x2 double
CutCategories	37x2 cell
CutPoint	37x1 double
CutType	37x1 cell
CutPredictor	37x1 cell
IsBranchNode	37x1 logical
NodeClass	37x1 cell
NodeError	37x1 double
NodeProbability	37x1 double
NodeRisk	37x1 double
NodeSize	37x1 double
NumNodes	37
Parent	37x1 double
PruneAlpha	[0;0.0014;0.0028;0.0043;0.005...
PruneList	37x1 double
SurrogateCutCategories	0x0 cell
SurrogateCutFlip	0x0 cell
SurrogateCutPoint	0x0 cell
SurrogateCutType	0x0 cell
SurrogateCutPredictor	0x1 cell
SurrogatePredictorAssociation	0x0 cell

Drzewo klasoryfikacyjne w Matlabie

Dopasowanie drzeca

```
ctree = fitctree(X,Y)
```

Property	Value
Y	351x1 cell
X	351x34 double
RowsUsed	[]
W	351x1 double
ModelParameters	1x1 TreeParams
NumObservations	351
HyperparameterOptimizationRe...	[]
PredictorNames	1x34 cell
CategoricalPredictors	[]
ResponseName	'Y'
ExpandedPredictorNames	1x34 cell
ClassNames	2x1 cell
Prior	[0.3590 0.6410]
Cost	[0 1;1 0]
ScoreTransform	'none'
CategoricalSplit	0x0 cell
Children	37x2 double
ClassCount	37x2 double
ClassProbability	37x2 double
CutCategories	37x2 cell
CutPoint	37x1 double
CutType	37x1 cell
CutPredictor	37x1 cell
IsBranchNode	37x1 logical
NodeClass	37x1 cell
NodeError	37x1 double
NodeProbability	37x1 double
NodeRisk	37x1 double
NodeSize	37x1 double
NumNodes	37
Parent	37x1 double
PruneAlpha	[0;0.0014;0.0028;0.0043;0.005...
PruneList	37x1 double
SurrogateCutCategories	0x0 cell
SurrogateCutFlip	0x0 cell
SurrogateCutPoint	0x0 cell
SurrogateCutType	0x0 cell
SurrogateCutPredictor	0x1 cell
SurrogatePredictorAssociation	0x0 cell

Property	Value
SplitCriterion	'gdi'
MinParent	10
MinLeaf	1
MaxSplits	350
NVarToSample	'all'
MergeLeaves	'on'
Prune	'on'
PruneCriterion	'error'
QE Toler	[]
NSurrogate	0
MaxCat	10
AlgCat	'auto'
PredictorSelection	'allsplits'
UseChisqTest	1
Stream	[]
Version	2
Method	'Tree'
Type	'classification'

Drzewo klasyfikacyjne w Matlabie

Dopasowanie drzewa

```
ctree = fitctree(X,Y)
```

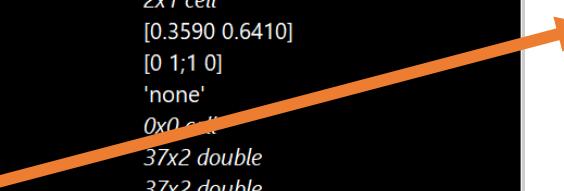
Property	Value
Y	351x1 cell
X	351x34 double
RowsUsed	[]
W	351x1 double
ModelParameters	1x1 TreeParams
NumObservations	351
HyperparameterOptimizationRe...	[]
PredictorNames	1x34 cell
CategoricalPredictors	[]
ResponseName	'Y'
ExpandedPredictorNames	1x34 cell
ClassNames	2x1 cell
Prior	[0.3590 0.6410]
Cost	[0 1;1 0]
ScoreTransform	'none'
CategoricalSplit	0x0 cell
Children	37x2 double
ClassCount	37x2 double
ClassProbability	37x2 double
CutCategories	37x2 cell
CutPoint	37x1 double
CutType	37x1 cell
CutPredictor	37x1 cell
IsBranchNode	37x1 logical
NodeClass	37x1 cell
NodeError	37x1 double
NodeProbability	37x1 double
NodeRisk	37x1 double
NodeSize	37x1 double
NumNodes	37
Parent	37x1 double
PruneAlpha	[0;0.0014;0.0028;0.0043;0.005...
PruneList	37x1 double
SurrogateCutCategories	0x0 cell
SurrogateCutFlip	0x0 cell
SurrogateCutPoint	0x0 cell
SurrogateCutType	0x0 cell
SurrogateCutPredictor	0x1 cell
SurrogatePredictorAssociation	0x0 cell

ctree.Children		
	1	2
1	2	3
2	4	5
3	6	7
4	0	0
5	8	9
6	10	11
7	12	13
8	0	0
9	0	0
10	0	0
11	14	15
12	0	0
13	16	17
14	18	19
15	20	21
16	0	0
17	22	23
18	24	25
19	26	27

Drzewo klasyfikacyjne w Matlabie

Dopasowanie drzewa

```
ctree = fitctree(X,Y)
```



The image shows two MATLAB workspace windows. The left window displays the properties of the 'ctree' variable, which is a 'ClassificationTree' object. The right window displays the 'ClassCount' table, which contains the number of observations for each class (1 and 2) across 20 categories. An orange arrow points from the 'ctree' variable in the left workspace to the 'ClassCount' table in the right workspace.

ctree

1x1 ClassificationTree

Property	Value
Y	351x1 cell
X	351x34 double
RowsUsed	[]
W	351x1 double
ModelParameters	1x1 TreeParams
NumObservations	351
HyperparameterOptimizationRe...	[]
PredictorNames	1x34 cell
CategoricalPredictors	[]
ResponseName	'Y'
ExpandedPredictorNames	1x34 cell
ClassNames	2x1 cell
Prior	[0.3590 0.6410]
Cost	[0 1;1 0]
ScoreTransform	'none'
CategoricalSplit	0x0 cell
Children	37x2 double
ClassCount	37x2 double
ClassProbability	37x2 double
CutCategories	37x2 cell
CutPoint	37x1 double
CutType	37x1 cell
CutPredictor	37x1 cell
IsBranchNode	37x1 logical
NodeClass	37x1 cell
NodeError	37x1 double
NodeProbability	37x1 double
NodeRisk	37x1 double
NodeSize	37x1 double
NumNodes	37
Parent	37x1 double
PruneAlpha	[0;0.0014;0.0028;0.0043;0.005...
PruneList	37x1 double
SurrogateCutCategories	0x0 cell
SurrogateCutFlip	0x0 cell
SurrogateCutPoint	0x0 cell
SurrogateCutType	0x0 cell
SurrogateCutPredictor	0x1 cell
SurrogatePredictorAssociation	0x0 cell

ctree

ctree.ClassCount

	1	2
1	126	225
2	73	4
3	53	221
4	67	0
5	6	4
6	14	208
7	39	13
8	1	4
9	5	0
10	3	1
11	11	207
12	19	0
13	20	13
14	9	46
15	2	161
16	8	0
17	12	13
18	3	42
19	6	4
20	1	0

Drzewo klasyfikacyjne w Matlabie

Dopasowanie drzewa

```
ctree = fitctree(X,Y)
```

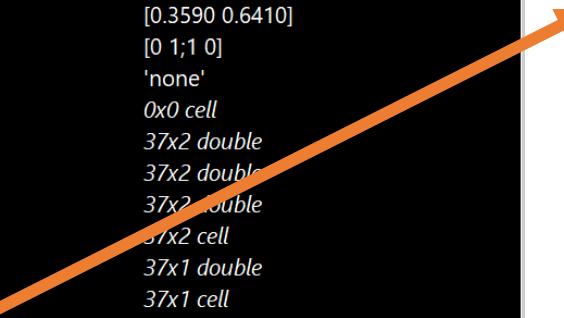
Property	Value
Y	351x1 cell
X	351x34 double
RowsUsed	[]
W	351x1 double
ModelParameters	1x1 TreeParams
NumObservations	351
HyperparameterOptimizationRe...	[]
PredictorNames	1x34 cell
CategoricalPredictors	[]
ResponseName	'Y'
ExpandedPredictorNames	1x34 cell
ClassNames	2x1 cell
Prior	[0.3590 0.6410]
Cost	[0 1;1 0]
ScoreTransform	'none'
CategoricalSplit	0x0 cell
Children	37x2 double
ClassCount	37x2 double
ClassProbability	37x2 double
CutCategories	37x2 cell
CutPoint	37x1 double
CutType	37x1 cell
CutPredictor	37x1 cell
IsBranchNode	37x1 logical
NodeClass	37x1 cell
NodeError	37x1 double
NodeProbability	37x1 double
NodeRisk	37x1 double
NodeSize	37x1 double
NumNodes	37
Parent	37x1 double
PruneAlpha	[0;0.0014;0.0028;0.0043;0.005...
PruneList	37x1 double
SurrogateCutCategories	0x0 cell
SurrogateCutFlip	0x0 cell
SurrogateCutPoint	0x0 cell
SurrogateCutType	0x0 cell
SurrogateCutPredictor	0x1 cell
SurrogatePredictorAssociation	0x0 cell

ctree.CutPoint
1
0.2315
0.0414
0.9999
NaN
-0.0560
-0.8967
0.5000
NaN
NaN
NaN
0.7313
NaN
0.7300
0.1710
-0.7559
NaN
0.4771
0.9256
0.2320
NaN

Drzewo klasyfikacyjne w Matlabie

Dopasowanie drzewa

```
ctree = fitctree(X,Y)
```



The figure shows two MATLAB workspace windows. The left window displays the properties of a 'ClassificationTree' object named 'ctree'. The right window shows the variable 'ctree.CutPredictor' which contains a vector of 20 elements, each representing a cut predictor for a specific node. An orange arrow points from the 'CutPredictor' entry in the left table to the 'ctree.CutPredictor' variable in the right window.

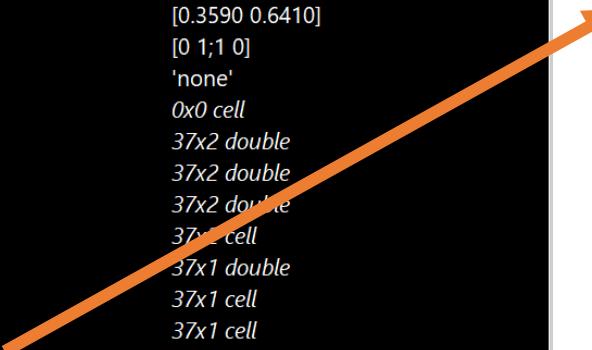
Property	Value
Y	351x1 cell
X	351x34 double
RowsUsed	[]
W	351x1 double
ModelParameters	1x1 TreeParams
NumObservations	351
HyperparameterOptimizationRe...	[]
PredictorNames	1x34 cell
CategoricalPredictors	[]
ResponseName	'Y'
ExpandedPredictorNames	1x34 cell
ClassNames	2x1 cell
Prior	[0.3590 0.6410]
Cost	[0 1;1 0]
ScoreTransform	'none'
CategoricalSplit	0x0 cell
Children	37x2 double
ClassCount	37x2 double
ClassProbability	37x2 double
CutCategories	37x2 cell
CutPoint	37x1 double
CutType	37x1 cell
CutPredictor	37x1 cell
IsBranchNode	37x1 logical
NodeClass	37x1 cell
NodeError	37x1 double
NodeProbability	37x1 double
NodeRisk	37x1 double
NodeSize	37x1 double
NumNodes	37
Parent	37x1 double
PruneAlpha	[0;0.0014;0.0028;0.0043;0.005...
PruneList	37x1 double
SurrogateCutCategories	0x0 cell
SurrogateCutFlip	0x0 cell
SurrogateCutPoint	0x0 cell
SurrogateCutType	0x0 cell
SurrogateCutPredictor	0x1 cell
SurrogatePredictorAssociation	0x0 cell

Index	Value
1	x5
2	x5
3	x27
4	
5	x24
6	x8
7	x1
8	
9	
10	
11	x3
12	
13	x3
14	x14
15	x10
16	
17	x22
18	x7
19	x6
20	

Drzewo klasyfikacyjne w Matlabie

Dopasowanie drzewa

```
ctree = fitctree(X,Y)
```



The figure shows two tables related to a classification tree model named 'ctree'. The left table, titled 'ctree', displays various properties of the tree, such as Y (351x1 cell), X (351x34 double), and ModelParameters (1x1 TreeParams). The right table, titled 'ctree.IsBranchNode', contains two columns: '1' and '2'. The '1' column lists node indices from 1 to 200, where values 1, 3, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, and 19 are marked with a yellow background. The '2' column is entirely black.

ctree	
1x1 ClassificationTree	
Property	Value
Y	351x1 cell
X	351x34 double
RowsUsed	[]
W	351x1 double
ModelParameters	1x1 TreeParams
NumObservations	351
HyperparameterOptimizationRe...	[]
PredictorNames	1x34 cell
CategoricalPredictors	[]
ResponseName	'Y'
ExpandedPredictorNames	1x34 cell
ClassNames	2x1 cell
Prior	[0.3590 0.6410]
Cost	[0 1;1 0]
ScoreTransform	'none'
CategoricalSplit	0x0 cell
Children	37x2 double
ClassCount	37x2 double
ClassProbability	37x2 double
CutCategories	37x2 cell
CutPoint	37x1 double
CutType	37x1 cell
CutPredictor	37x1 cell
IsBranchNode	37x1 logical
NodeClass	37x1 cell
NodeError	37x1 double
NodeProbability	37x1 double
NodeRisk	37x1 double
NodeSize	37x1 double
NumNodes	37
Parent	37x1 double
PruneAlpha	[0;0.0014;0.0028;0.0043;0.005...
PruneList	37x1 double
SurrogateCutCategories	0x0 cell
SurrogateCutFlip	0x0 cell
SurrogateCutPoint	0x0 cell
SurrogateCutType	0x0 cell
SurrogateCutPredictor	0x1 cell
SurrogatePredictorAssociation	0x0 cell

ctree	
ctree.IsBranchNode	
	1
1	1
2	1
3	1
4	0
5	1
6	1
7	1
8	0
9	0
10	0
11	1
12	0
13	1
14	1
15	1
16	0
17	1
18	1
19	1
20	0

Drzewo klasyfikacyjne w Matlabie

Dopasowanie drzewa

```
ctree = fitctree(X,Y)
```

Property	Value
Y	351x1 cell
X	351x34 double
RowsUsed	[]
W	351x1 double
ModelParameters	1x1 TreeParams
NumObservations	351
HyperparameterOptimizationRe...	[]
PredictorNames	1x34 cell
CategoricalPredictors	[]
ResponseName	'Y'
ExpandedPredictorNames	1x34 cell
ClassNames	2x1 cell
Prior	[0.3590 0.6410]
Cost	[0 1;1 0]
ScoreTransform	'none'
CategoricalSplit	0x0 cell
Children	37x2 double
ClassCount	37x2 double
ClassProbability	37x2 double
CutCategories	37x2 cell
CutPoint	37x1 double
CutType	37x1 cell
CutPredictor	37x1 cell
IsBranchNode	37x1 logical
NodeClass	37x1 cell
NodeError	37x1 double
NodeProbability	37x1 double
NodeRisk	37x1 double
NodeSize	37x1 double
NumNodes	37
Parent	37x1 double
PruneAlpha	[0;0.0014;0.0028;0.0043;0.005...
PruneList	37x1 double
SurrogateCutCategories	0x0 cell
SurrogateCutFlip	0x0 cell
SurrogateCutPoint	0x0 cell
SurrogateCutType	0x0 cell
SurrogateCutPredictor	0x1 cell
SurrogatePredictorAssociation	0x0 cell

ctree	ctree
1	
1	g
2	b
3	g
4	b
5	b
6	g
7	b
8	g
9	b
10	b
11	g
12	b
13	b
14	g
15	g
16	b
17	g
18	g
19	b
20	b

Drzewo klasyfikacyjne w Matlabie

Dopasowanie drzewa

```
ctree = fitctree(X,Y)
```

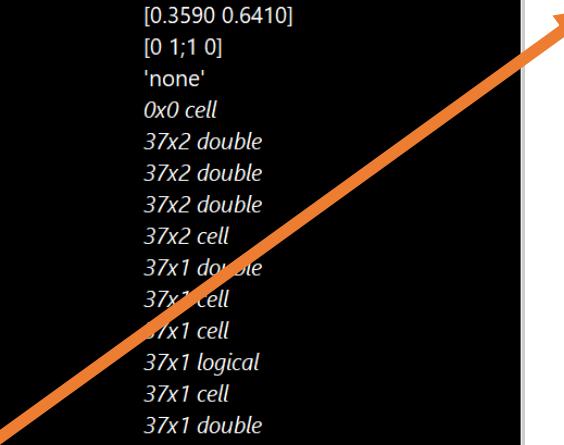
Property	Value
Y	351x1 cell
X	351x34 double
RowsUsed	[]
W	351x1 double
ModelParameters	1x1 TreeParams
NumObservations	351
HyperparameterOptimizationRe...	[]
PredictorNames	1x34 cell
CategoricalPredictors	[]
ResponseName	'Y'
ExpandedPredictorNames	1x34 cell
ClassNames	2x1 cell
Prior	[0.3590 0.6410]
Cost	[0 1;1 0]
ScoreTransform	'none'
CategoricalSplit	0x0 cell
Children	37x2 double
ClassCount	37x2 double
ClassProbability	37x2 double
CutCategories	37x2 cell
CutPoint	37x1 double
CutType	37x1 cell
CutPredictor	37x1 cell
IsBranchNode	37x1 logical
NodeClass	37x1 cell
NodeError	37x1 double
NodeProbability	37x1 double
NodeRisk	37x1 double
NodeSize	37x1 double
NumNodes	37
Parent	37x1 double
PruneAlpha	[0;0.0014;0.0028;0.0043;0.005...
PruneList	37x1 double
SurrogateCutCategories	0x0 cell
SurrogateCutFlip	0x0 cell
SurrogateCutPoint	0x0 cell
SurrogateCutType	0x0 cell
SurrogateCutPredictor	0x1 cell
SurrogatePredictorAssociation	0x0 cell

ctree	ctree
1	
1	0.3590
2	0.0519
3	0.1934
4	0
5	0.4000
6	0.0631
7	0.2500
8	0.2000
9	0
10	0.2500
11	0.0505
12	0
13	0.3939
14	0.1636
15	0.0123
16	0
17	0.4800
18	0.0667
19	0.4000
20	0

Drzewo klasyfikacyjne w Matlabie

Dopasowanie drzewa

```
ctree = fitctree(X,Y)
```



The figure shows two MATLAB workspace windows. The left window, titled 'ctree', displays a table of properties for a classification tree model. The right window, also titled 'ctree', displays a table of node probabilities for the tree.

ctree Properties:

Property	Value
Y	351x1 cell
X	351x34 double
RowsUsed	[]
W	351x1 double
ModelParameters	1x1 TreeParams
NumObservations	351
HyperparameterOptimizationRe...	[]
PredictorNames	1x34 cell
CategoricalPredictors	[]
ResponseName	'Y'
ExpandedPredictorNames	1x34 cell
ClassNames	2x1 cell
Prior	[0.3590 0.6410]
Cost	[0 1;1 0]
ScoreTransform	'none'
CategoricalSplit	0x0 cell
Children	37x2 double
ClassCount	37x2 double
ClassProbability	37x2 double
CutCategories	37x2 cell
CutPoint	37x1 double
CutType	37x1 cell
CutPredictor	37x1 cell
IsBranchNode	37x1 logical
NodeClass	37x1 cell
NodeError	37x1 double
NodeProbability	37x1 double
NodeRisk	37x1 double
NodeSize	37x1 double
NumNodes	37
Parent	37x1 double
PruneAlpha	[0;0.0014;0.0028;0.0043;0.005...
PruneList	37x1 double
SurrogateCutCategories	0x0 cell
SurrogateCutFlip	0x0 cell
SurrogateCutPoint	0x0 cell
SurrogateCutType	0x0 cell
SurrogateCutPredictor	0x1 cell
SurrogatePredictorAssociation	0x0 cell

ctree.NodeProbability:

	1	2
1	1.0000	
2	0.2194	
3	0.7806	
4	0.1909	
5	0.0285	
6	0.6325	
7	0.1481	
8	0.0142	
9	0.0142	
10	0.0114	
11	0.6211	
12	0.0541	
13	0.0940	
14	0.1567	
15	0.4644	
16	0.0228	
17	0.0712	
18	0.1282	
19	0.0285	
20	0.0028	

Drzewo klasyfikacyjne w Matlabie

Dopasowanie drzewa

```
ctree = fitctree(X,Y)
```

Property	Value
Y	351x1 cell
X	351x34 double
RowsUsed	[]
W	351x1 double
ModelParameters	1x1 TreeParams
NumObservations	351
HyperparameterOptimizationRe...	[]
PredictorNames	1x34 cell
CategoricalPredictors	[]
ResponseName	'Y'
ExpandedPredictorNames	1x34 cell
ClassNames	2x1 cell
Prior	[0.3590 0.6410]
Cost	[0 1;1 0]
ScoreTransform	'none'
CategoricalSplit	0x0 cell
Children	37x2 double
ClassCount	37x2 double
ClassProbability	37x2 double
CutCategories	37x2 cell
CutPoint	37x1 double
CutType	37x1 cell
CutPredictor	37x1 cell
IsBranchNode	37x1 logical
NodeClass	37x1 cell
NodeError	37x1 double
NodeProbability	37x1 double
NodeRisk	37x1 double
NodeSize	37x1 double
NumNodes	37
Parent	37x1 double
PruneAlpha	[0;0.0014;0.0028;0.0043;0.005...
PruneList	37x1 double
SurrogateCutCategories	0x0 cell
SurrogateCutFlip	0x0 cell
SurrogateCutPoint	0x0 cell
SurrogateCutType	0x0 cell
SurrogateCutPredictor	0x1 cell
SurrogatePredictorAssociation	0x0 cell

ctree	ctree.
1	
1	351
2	77
3	274
4	67
5	10
6	222
7	52
8	5
9	5
10	4
11	218
12	19
13	33
14	55
15	163
16	8
17	25
18	45
19	10
20	1

Drzewo klasyfikacyjne w Matlabie

Dopasowanie drzewa

```
ctree = fitctree(X,Y)
```

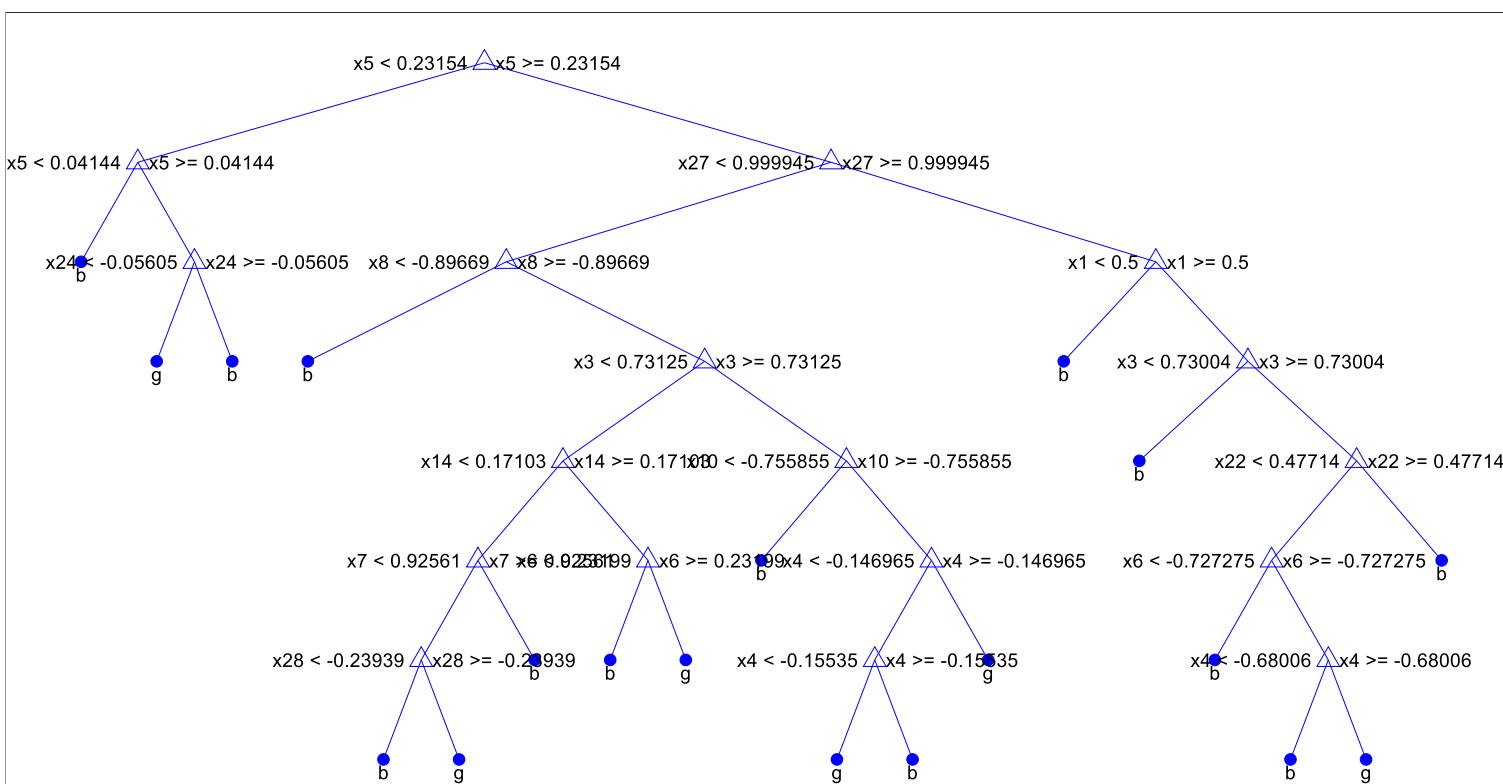
Property	Value
Y	351x1 cell
X	351x34 double
RowsUsed	[]
W	351x1 double
ModelParameters	1x1 TreeParams
NumObservations	351
HyperparameterOptimizationRe...	[]
PredictorNames	1x34 cell
CategoricalPredictors	[]
ResponseName	'Y'
ExpandedPredictorNames	1x34 cell
ClassNames	2x1 cell
Prior	[0.3590 0.6410]
Cost	[0 1;1 0]
ScoreTransform	'none'
CategoricalSplit	0x0 cell
Children	37x2 double
ClassCount	37x2 double
ClassProbability	37x2 double
CutCategories	37x2 cell
CutPoint	37x1 double
CutType	37x1 cell
CutPredictor	37x1 cell
IsBranchNode	37x1 logical
NodeClass	37x1 cell
NodeError	37x1 double
NodeProbability	37x1 double
NodeRisk	37x1 double
NodeSize	37x1 double
NumNodes	37
Parent	37x1 double
PruneAlpha	[0;0.0014;0.0028;0.0043;0.005...
PruneList	37x1 double
SurrogateCutCategories	0x0 cell
SurrogateCutFlip	0x0 cell
SurrogateCutPoint	0x0 cell
SurrogateCutType	0x0 cell
SurrogateCutPredictor	0x1 cell
SurrogatePredictorAssociation	0x0 cell

ctree.Parent
1
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

Drzewo klasycfikacyjne w Matlabie

Dopasowanie drzewa

```
ctree = fitctree(X,Y)
view(ctree,'mode','graph');
view(ctree);
```



```
1 if x5<0.23154 then node 2 elseif x5>=0.23154 then node 3 else g
2 if x5<0.04144 then node 4 elseif x5>=0.04144 then node 5 else b
3 if x27<0.999945 then node 6 elseif x27>=0.999945 then node 7 else g
4 class = b
5 if x24<-0.05605 then node 8 elseif x24>=-0.05605 then node 9 else b
6 if x8<-0.89669 then node 10 elseif x8>=-0.89669 then node 11 else g
7 if x1<0.5 then node 12 elseif x1>=0.5 then node 13 else b
8 class = g
9 class = b
10 class = b
11 if x3<0.73125 then node 14 elseif x3>=0.73125 then node 15 else g
12 class = b
13 if x3<0.73004 then node 16 elseif x3>=0.73004 then node 17 else b
14 if x14<0.17103 then node 18 elseif x14>=0.17103 then node 19 else g
15 if x10<-0.755855 then node 20 elseif x10>=-0.755855 then node 21 else g
16 class = b
17 if x22<0.47714 then node 22 elseif x22>=0.47714 then node 23 else g
18 if x7<0.92561 then node 24 elseif x7>=0.92561 then node 25 else g
19 if x6<0.23199 then node 26 elseif x6>=0.23199 then node 27 else b
20 class = b
21 if x4<-0.146965 then node 28 elseif x4>=-0.146965 then node 29 else g
22 if x6<-0.727275 then node 30 elseif x6>=-0.727275 then node 31 else g
23 class = b
24 if x28<-0.23939 then node 32 elseif x28>=-0.23939 then node 33 else g
25 class = b
26 class = b
27 class = g
28 if x4<-0.15535 then node 34 elseif x4>=-0.15535 then node 35 else g
29 class = g
30 class = b
31 if x4<-0.68006 then node 36 elseif x4>=-0.68006 then node 37 else g
32 class = b
33 class = g
34 class = g
35 class = b
36 class = b
37 class = g
```

Drzewo klasyfikacyjne w Matlabie

Błędy klasyfikacji dla zbioru uczącego

```
Ynew = predict(ctree,X)
```

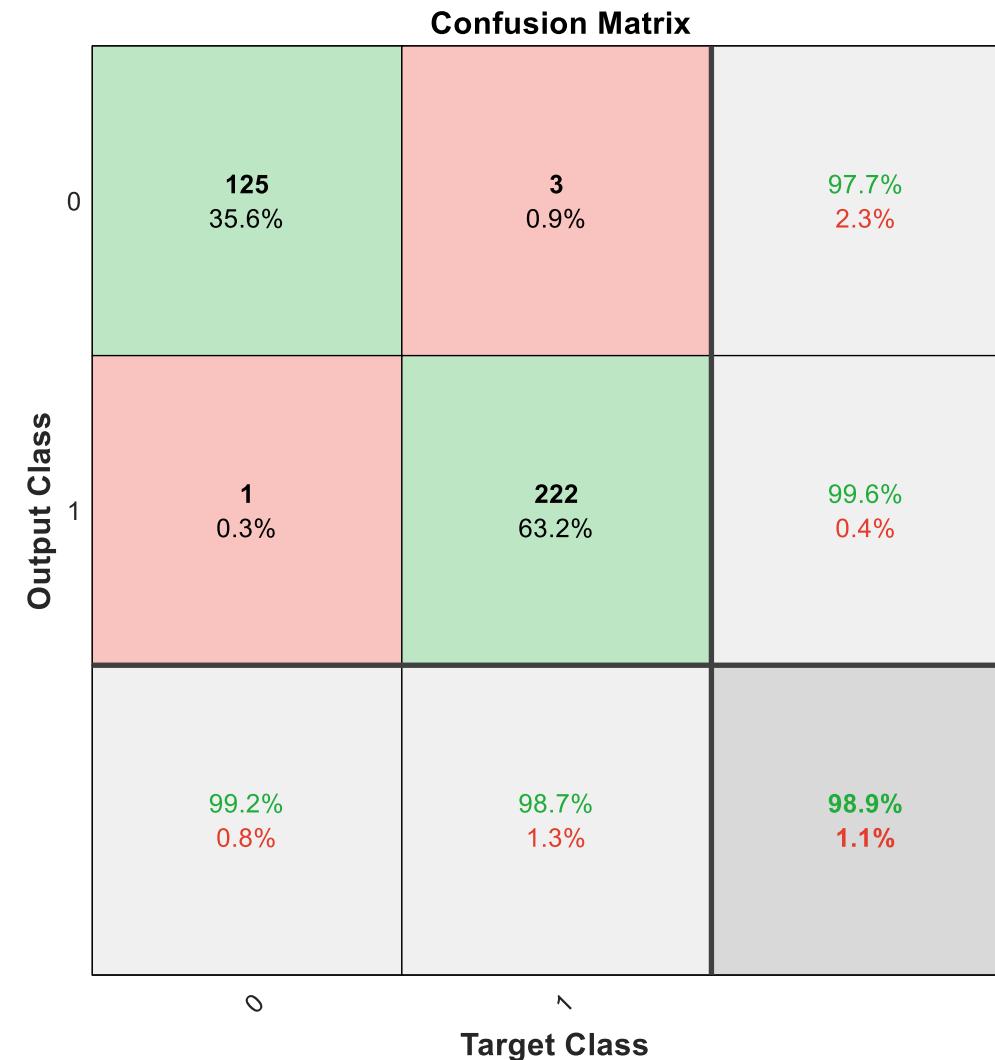
```
Y1(cell2mat(Y)=='b')=0;
```

```
Y1(cell2mat(Y)=='g')=1;
```

```
Ynew1(cell2mat(Ynew)=='b')=0;
```

```
Ynew1(cell2mat(Ynew)=='g')=1;
```

```
plotconfusion(Y1,Ynew1)
```



Drzewo klasyfikacyjne w Matlabie

Błędy klasyfikacji estymowane w kroswalidacji

```
cvctree = crossval(ctree);
```

cvctree	
1x1 ClassificationPartitionedModel	
Property	Value
ClassNames	2x1 cell
Cost	[0 1;1 0]
Prior	[0.3590 0.6410]
ScoreTransform	'none'
CrossValidatedModel	'Tree'
PredictorNames	1x34 cell
CategoricalPredictors	[]
ResponseName	'Y'
NumObservations	351
X	351x34 double
Y	351x1 cell
W	351x1 double
ModelParameters	1x1 EnsembleParams
Trained	10x1 cell
KFold	10
Partition	1x1 cvpartition

cvctree.Trained	
1	
1	1x1 CompactClassificationTree
2	1x1 CompactClassificationTree
3	1x1 CompactClassificationTree
4	1x1 CompactClassificationTree
5	1x1 CompactClassificationTree
6	1x1 CompactClassificationTree
7	1x1 CompactClassificationTree
8	1x1 CompactClassificationTree
9	1x1 CompactClassificationTree
10	1x1 CompactClassificationTree

cvctree.Partition	
cvctree.Partition	
Property	Value
Type	'kfold'
NumTestSets	10
TrainSize	[316 315 316 316 316 316 316 31...
TestSize	[35 36 35 35 35 35 35 35 35 35 ...
NumObservations	351

cvctree.Trained{1, 1}	
Property	Value
CategoricalSplit	0x0 cell
Children	41x2 double
ClassCount	41x2 double
ClassProbability	41x2 double
CutCategories	41x2 cell
CutPoint	41x1 double
CutType	41x1 cell
CutPredictor	41x1 cell
IsBranchNode	41x1 logical
NodeClass	41x1 cell
NodeError	41x1 double
NodeProbability	41x1 double
NodeRisk	41x1 double
NodeSize	41x1 double
NumNodes	41
Parent	41x1 double
PruneAlpha	[0;0.0016;0.003
PruneList	41x1 double
SurrogateCutCategories	0x0 cell
SurrogateCutFlip	0x0 cell
SurrogateCutPoint	0x0 cell
SurrogateCutType	0x0 cell
SurrogateCutPredictor	0x1 cell
SurrogatePredictorAsso...	0x0 cell
ClassNames	2x1 cell
Prior	[0.3590 0.6410]
Cost	[0 1;1 0]
ScoreTransform	'none'
PredictorNames	1x34 cell
CategoricalPredictors	[]
ResponseName	'Y'
ExpandedPredictorNames	1x34 cell

Drzewo klasyfikacyjne w Matlabie

Błędy klasyfikacji estymowane w kroswalidacji

```
cvctree = crossval(ctree);
```

```
cvloss = kfoldLoss(cvctree)
```

```
cvloss =
```

```
0.1140
```

Drzewo klasyfikacyjne w Matlabie

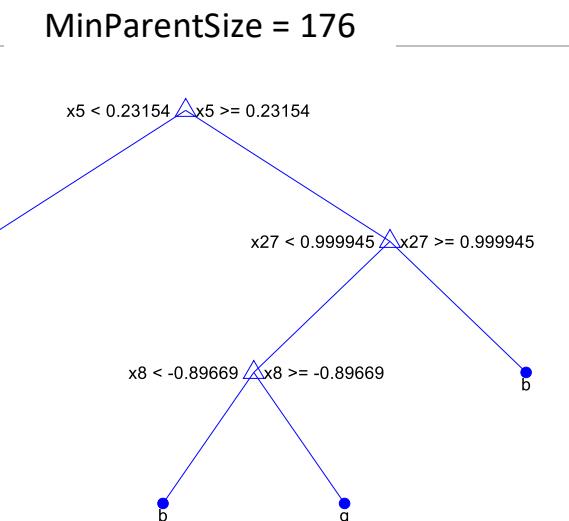
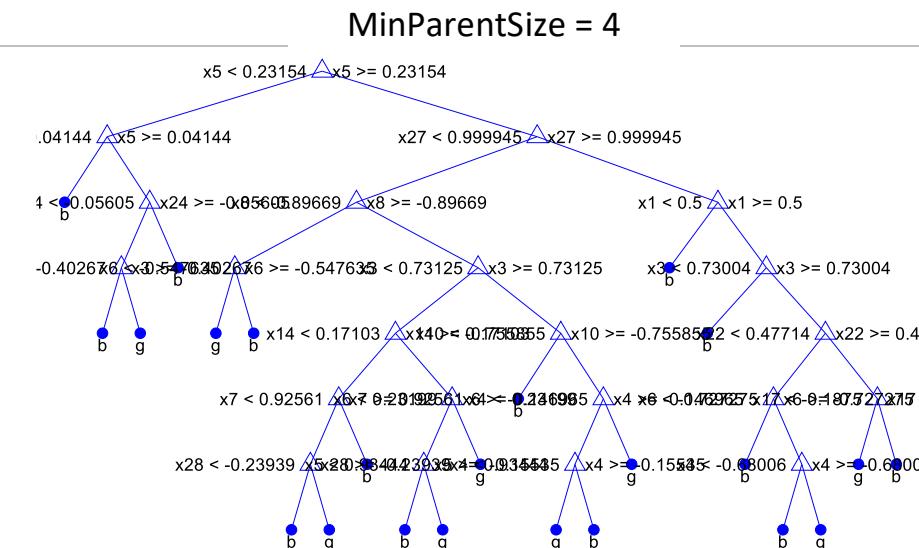
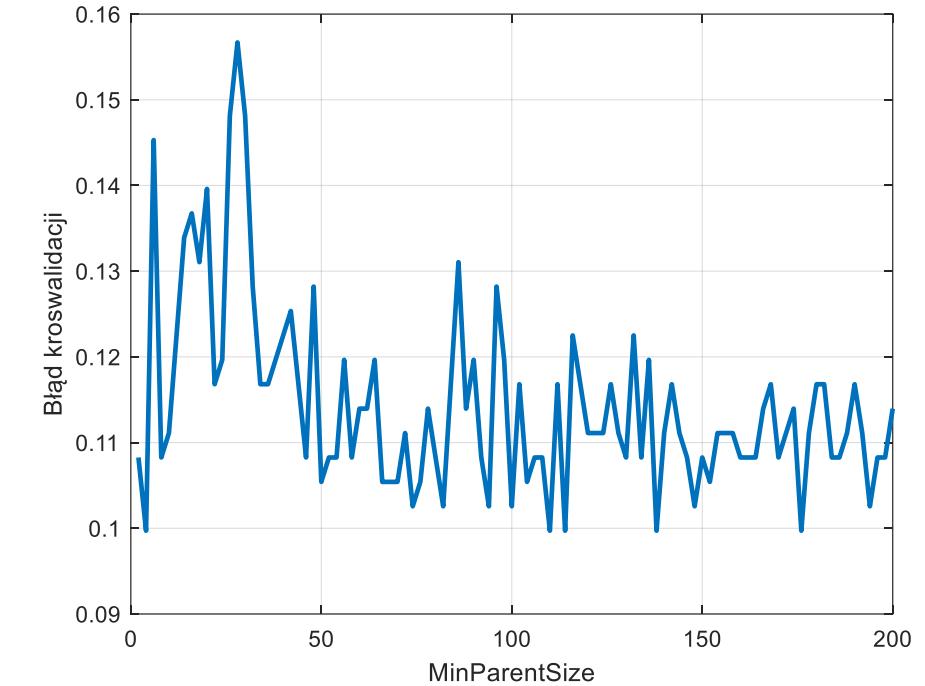
Dobór parametru *MinParentSize*

```
err = nan(100,1);
m=2:2:200;
for i=1:100
    tree = fitctree(X,Y,'MinParentSize',m(i),'CrossVal','on');
    err(i) = kfoldLoss(tree);
end
plot(m,err);
xlabel('MinParentSize');
ylabel('Błąd kroswalidacji');
```

Drzewo klasoryfikacyjne w Matlabie

Dobór parametru *MinParentSize*

```
err = nan(100,1);
m=2:2:200;
for i=1:100
    tree = fitctree(X,Y, 'MinParentSize', m(i), 'CrossVal', 'on');
    err(i) = kfoldLoss(tree);
end
plot(m,err);
xlabel('MinParentSize');
ylabel('Błąd kroswalidacji');
```



Drzewo klasyfikacyjne w Matlabie

Dobór parametru MinParentSize

```
m_opt = 176;  
OptimalTree = fitctree(X,Y,'MinParentSize',m_opt);
```

Drzewo klasifykacyjne w Matlabie

Dobór parametru MinParentSize

```
m_opt = 176;
```

```
OptimalTree = fitctree(X,Y,'MinParentSize',m_opt);
```

```
errOpt1 = resubLoss(OptimalTree) %błąd na zbiorze uczącym dla OptimalTree
```

0.0826

```
err1 = resubLoss(ctree) %błąd na zbiorze uczącym dla ctree
```

0.0114

Drzewo klasifykacyjne w Matlabie

Dobór parametru *MinParentSize*

```
m_opt = 176;
```

```
OptimalTree = fitctree(X,Y,'MinParentSize',m_opt);
```

```
errOpt1 = resubLoss(OptimalTree) %błąd na zbiorze uczącym dla OptimalTree
```

0.0826

```
err1 = resubLoss(ctree) %błąd na zbiorze uczącym dla ctree
```

0.0114

```
errOpt2 = kfoldLoss(crossval(OptimalTree)) %błąd kroswalidacji dla OptimalTree
```

0.0997

```
err2 = kfoldLoss(crossval(ctree)) %błąd kroswalidacji dla ctree
```

0.1140

Drzewo klasyfikacyjne w Matlabie

Przycinanie drzewa

```
PruneTree = prune(ctree,'Level',1);
```

Drzewo klasyfikacyjne w Matlabie

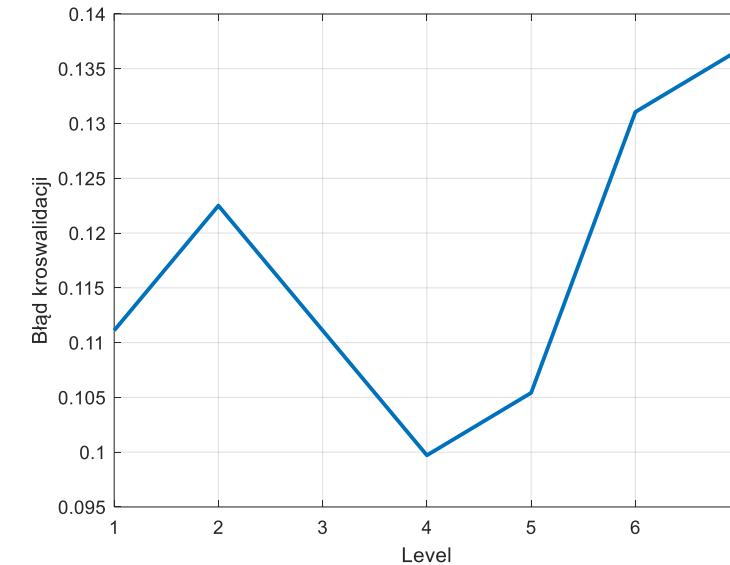
Przycinanie drzewa

```
err = nan(7,1);
m=1:7;
for i=1:7
    tree = prune(ctree, 'Level', m(i));
    err(i) = kfoldLoss(crossval(tree));
end
plot(m,err);
xlabel('Level');
ylabel('Błąd kroswalidacji');
```

Drzewo klasyfikacyjne w Matlabie

Przycinanie drzewa

```
err = nan(7,1);
m=1:7;
for i=1:7
    tree = prune(ctree, 'Level', m(i));
    err(i) = kfoldLoss(crossval(tree));
end
plot(m,err);
xlabel('Level');
ylabel('Błąd kroswalidacji');
```



Drzewo klasyfikacyjne w Matlabie

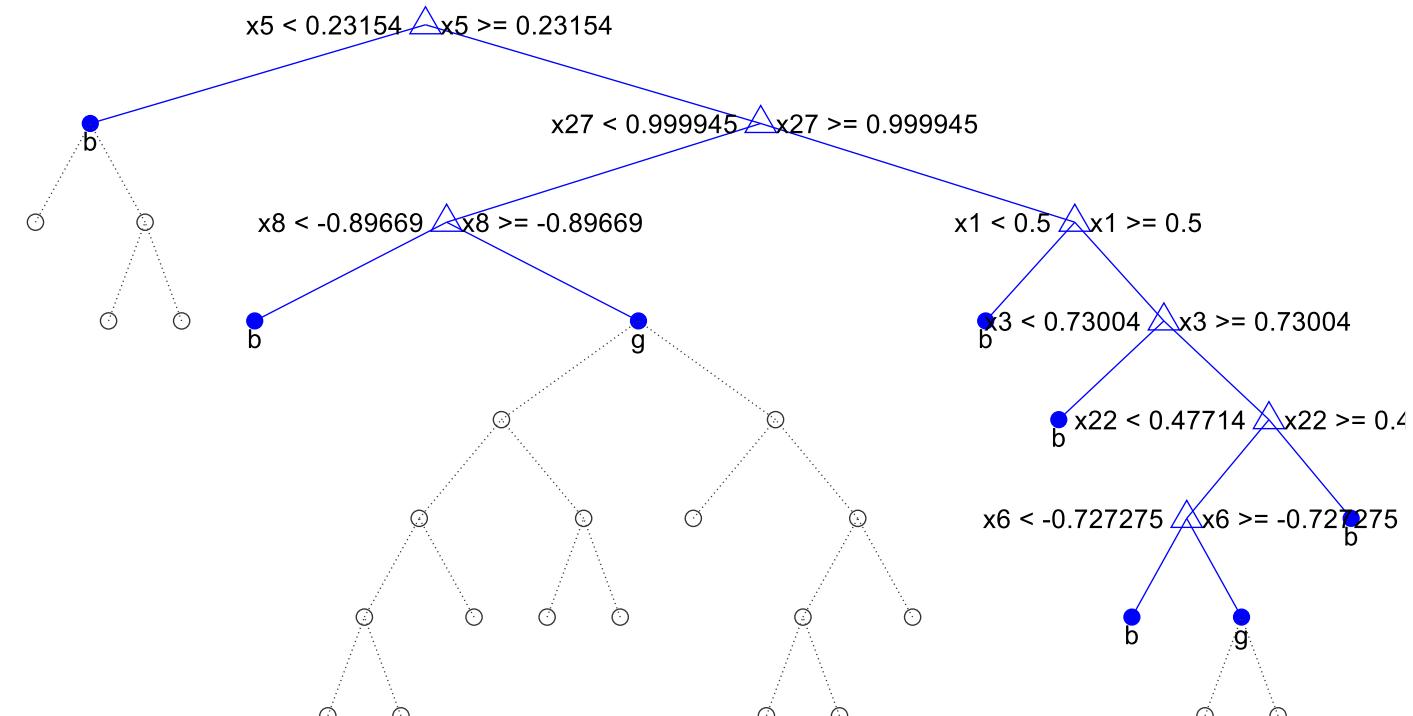
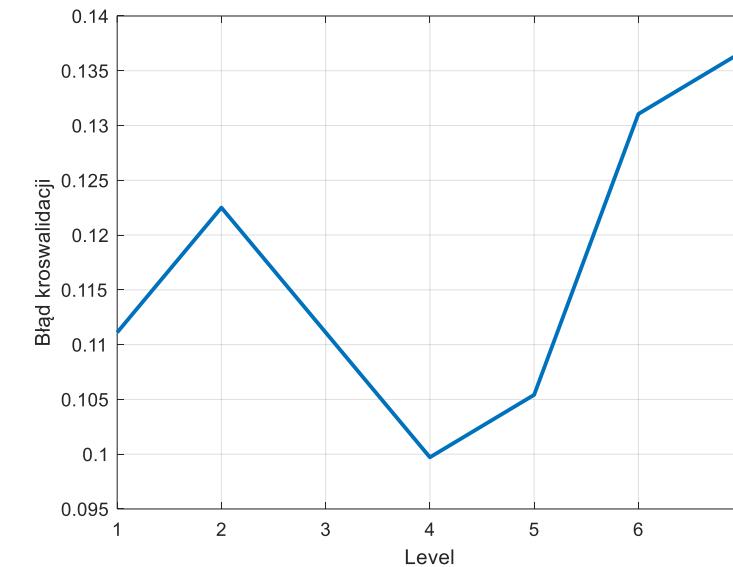
Przycinanie drzewa

```

err = nan(7,1);
m=1:7;
for i=1:7
    tree = prune(ctree, 'Level', m(i));
    err(i) = kfoldLoss(crossval(tree));
end
plot(m,err);
xlabel('Level');
ylabel('Błąd kroswalidacji');

```

```
PruneTree = prune(ctree, 'Level',4);  
view(PruneTree, 'Mode', 'graph');
```



Drzewo klasyfikacyjne w Matlabie

Przycinanie drzewa

```

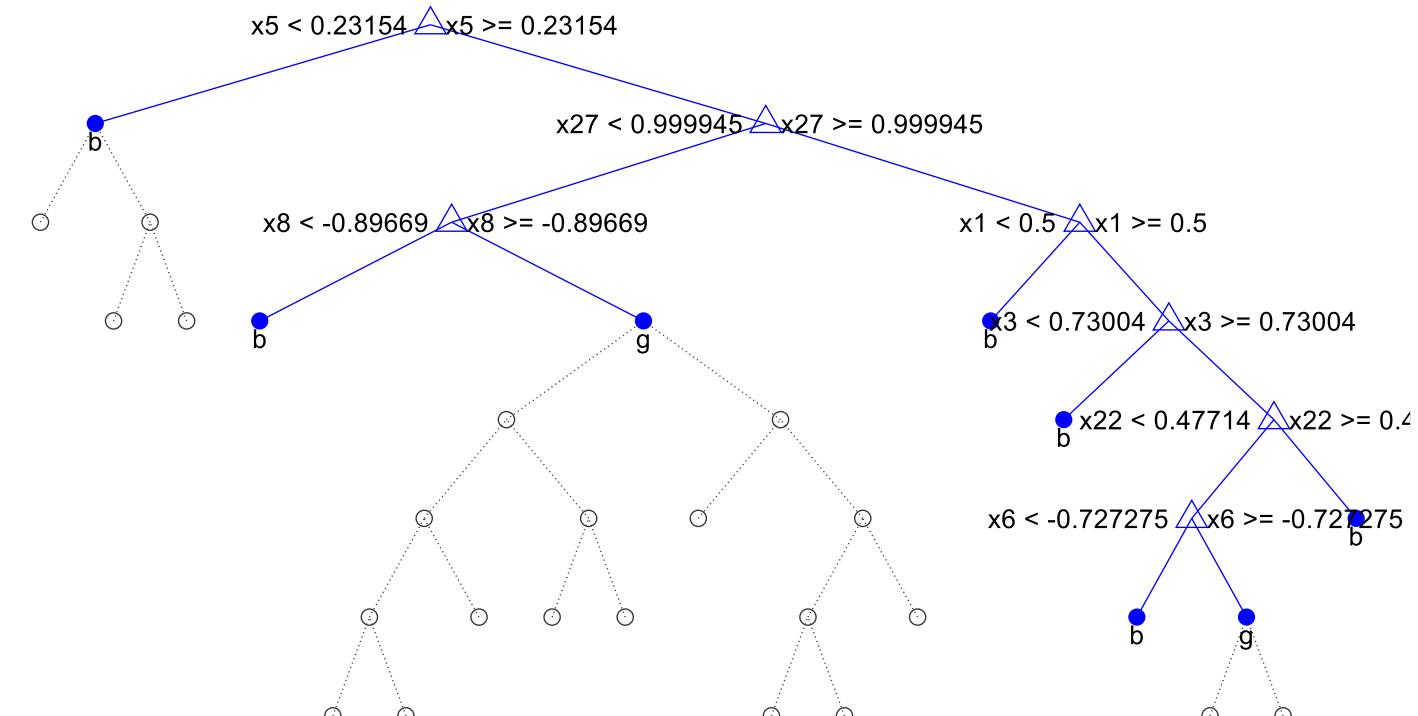
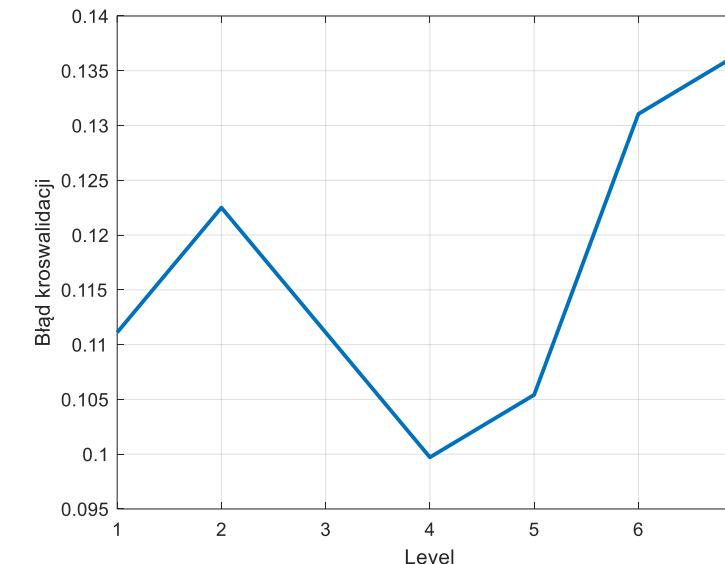
err = nan(7,1);
m=1:7;
for i=1:7
    tree = prune(ctree, 'Level', m(i));
    err(i) = kfoldLoss(crossval(tree));
end
plot(m,err);
xlabel('Level');
ylabel('Błąd kroswalidacji');

```

```
PruneTree = prune(ctree, 'Level',4)  
view(PruneTree, 'Mode', 'graph');
```

```
%błąd na zbiorze uczącym dla PruneTree  
errPrun1 = resubLoss(PruneTree)  
0.0513
```

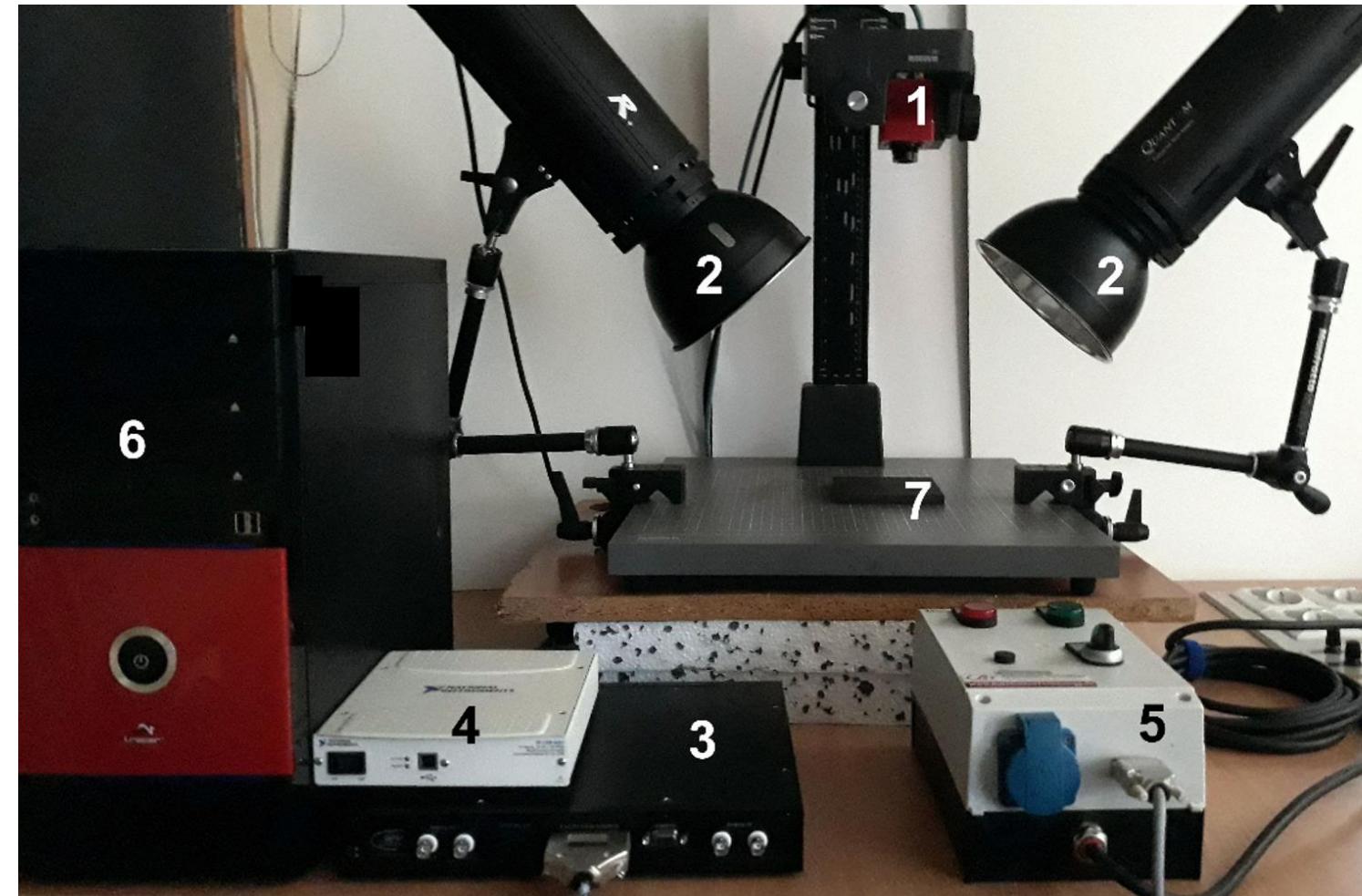
```
%błąd kroswalidacji dla PruneTree  
errPrun2 = kfoldLoss(crossval(PruneTree))  
0.1083
```



Przykład zastosowania drzewa klasyfikacyjnego

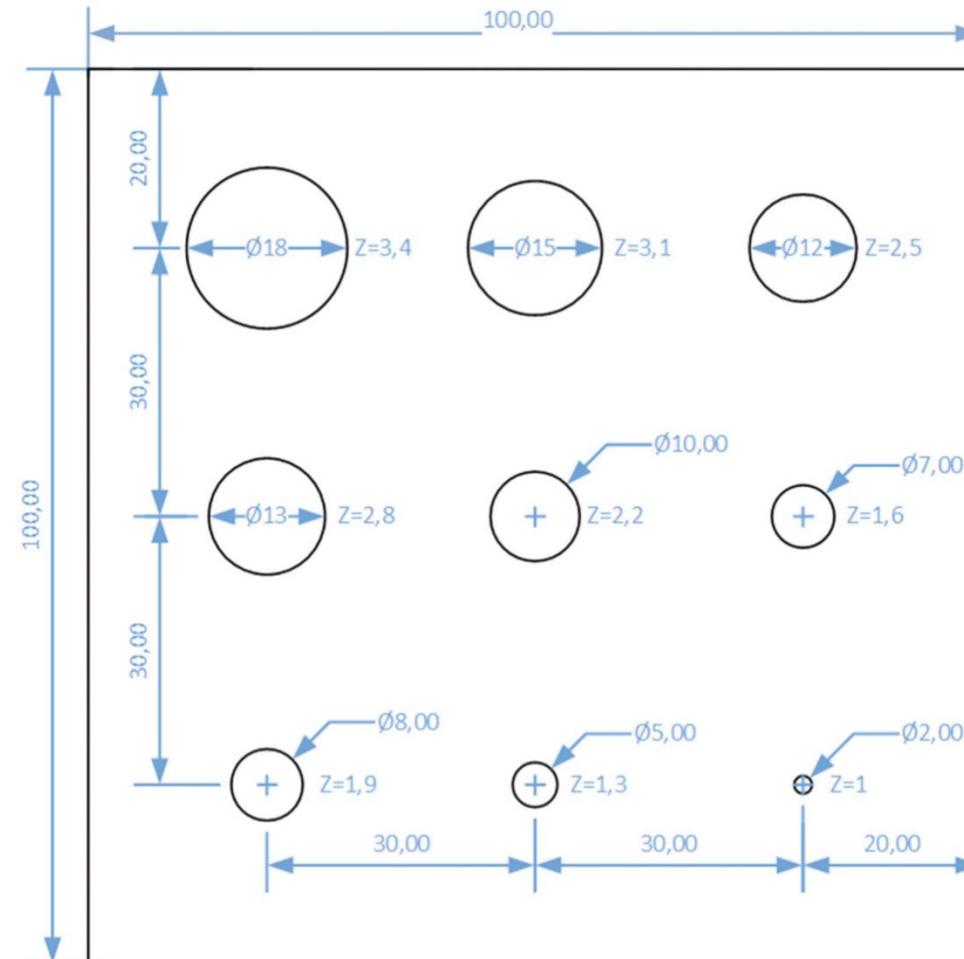
Drzewo klasyfikacyjne do wykrywania defektów materiałowych przy użyciu termografii aktywnej

Układ pomiarowy



Przykład zastosowania drzewa klasyfikacyjnego

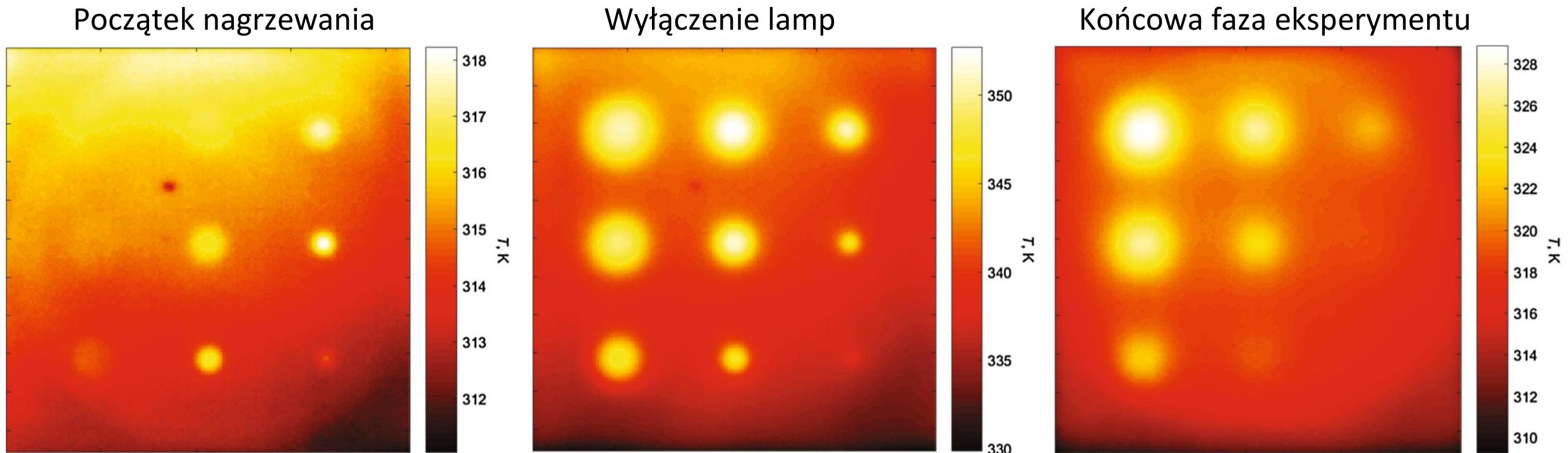
Próbka



Przykład zastosowania drzewa klasyfikacyjnego

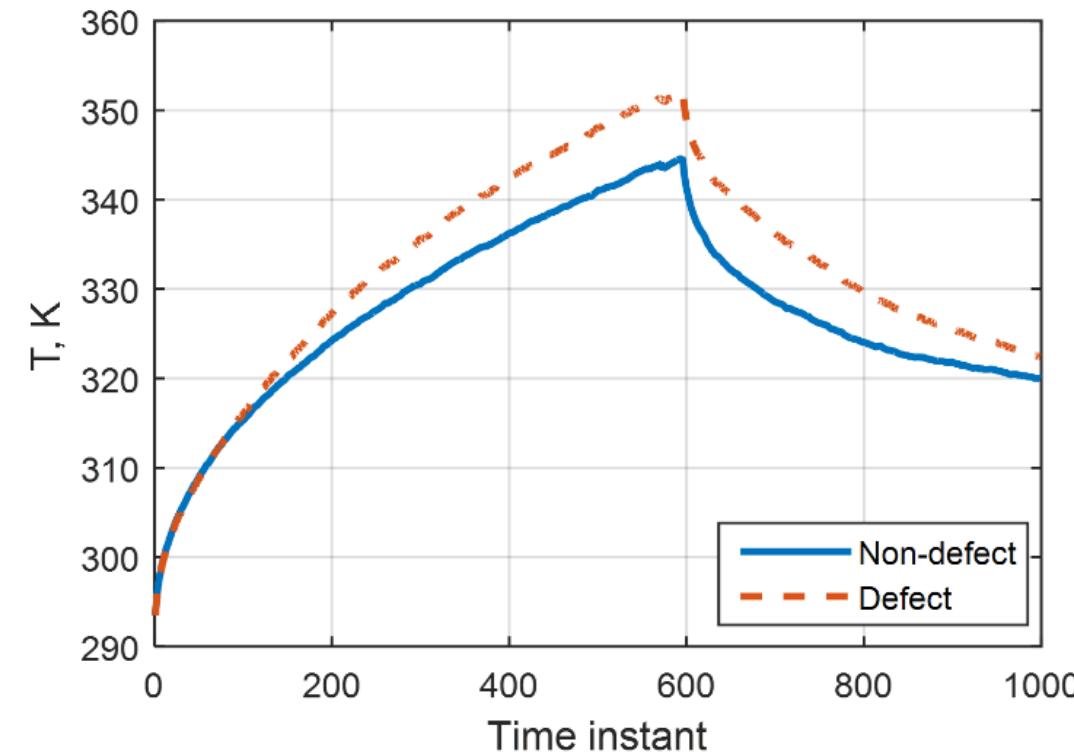
Próbka jest nagrzewana przez czas t_1 . Później lampy są wyłączane. Rejestrujemy termogramy w fazie nagrzewania (t_1) i stygnięcia (t_2) z częstotliwością f Hz.

Każdy termogram ma rozmiar 189×190 pikseli.



Przykład zastosowania drzewa klasyfikacyjnego

Krzywe nagrzewania-stygnięcia



Przykład zastosowania drzewa klasyfikacyjnego

Drzewo klasyfikacyjne rozwiązuje problem klasyfikacji n wymiarowych wektorów \mathbf{x} reprezentujących krzywe nagrzewania-stygnięcia zarejestrowane w poszczególnych punktach termogramu do jednej z dziesięciu klas odpowiadających rozmiarom defektów.

Przykłady:

$$\mathbf{x}_i = [T_{i,1} \ T_{i,2} \ \dots \ T_{i,n}], \ i = 1, 2, \dots, 189 \times 190$$

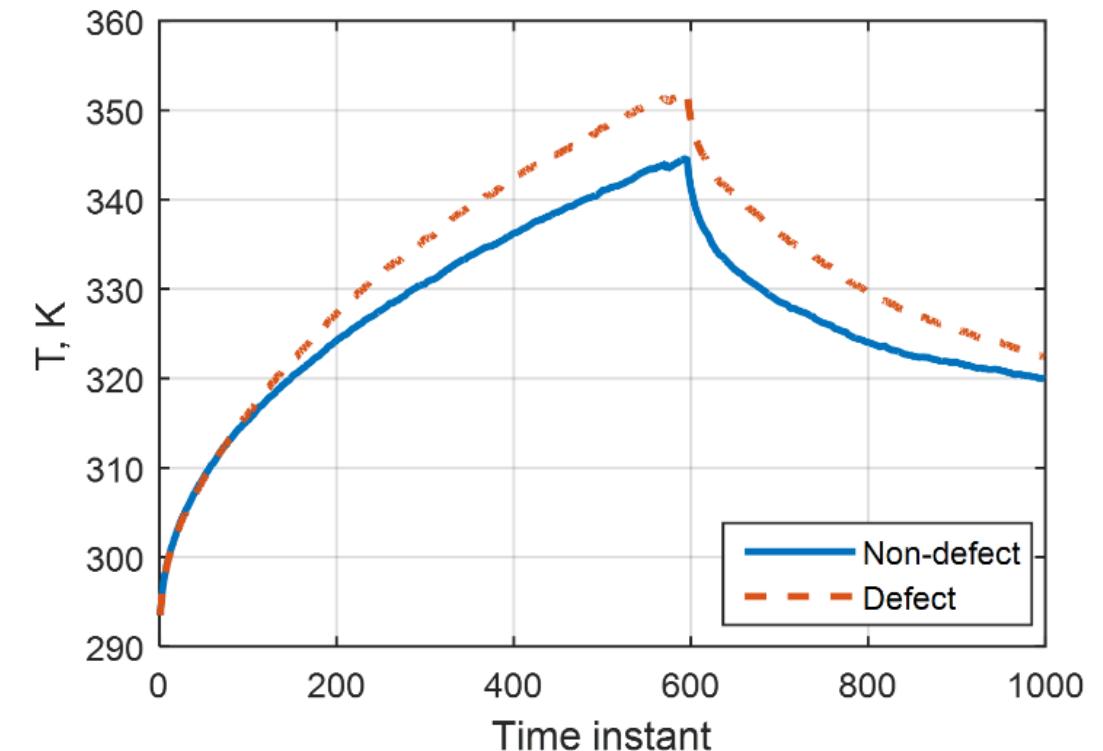
Klasy:

$$y = 0 - \text{brak defektu}, 1 - \text{defekt 1}, \dots, 9 - \text{defekt 9}$$

Zbiór uczący:

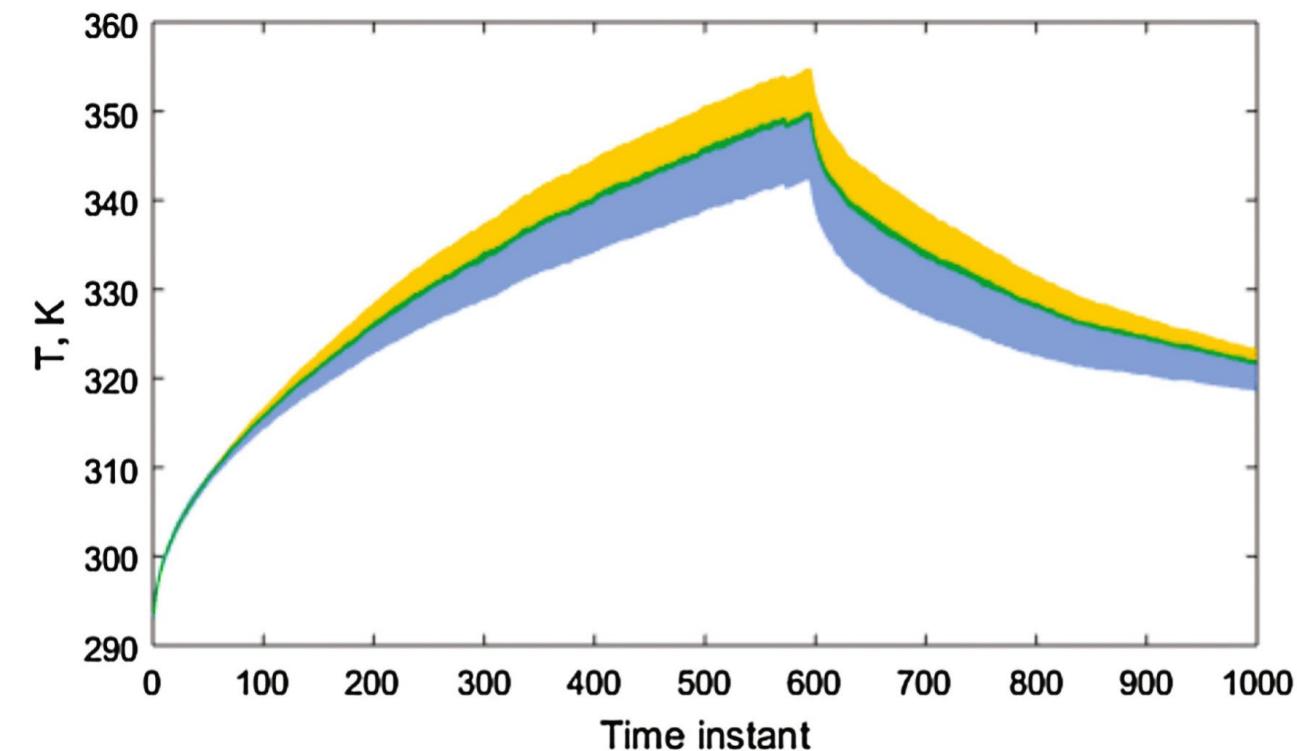
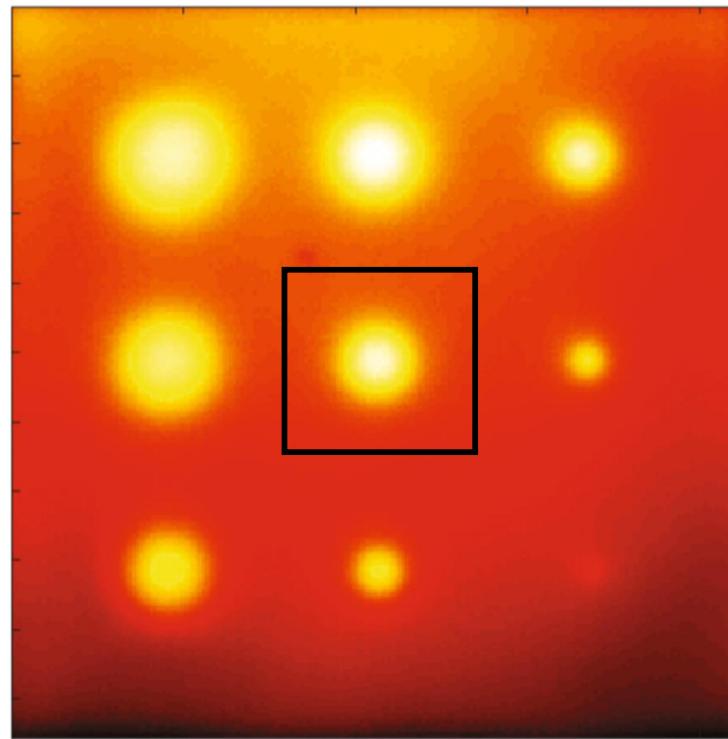
$$\Omega = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\},$$

zawiera $N = 189 \times 190$ sklasyfikowanych przykładów



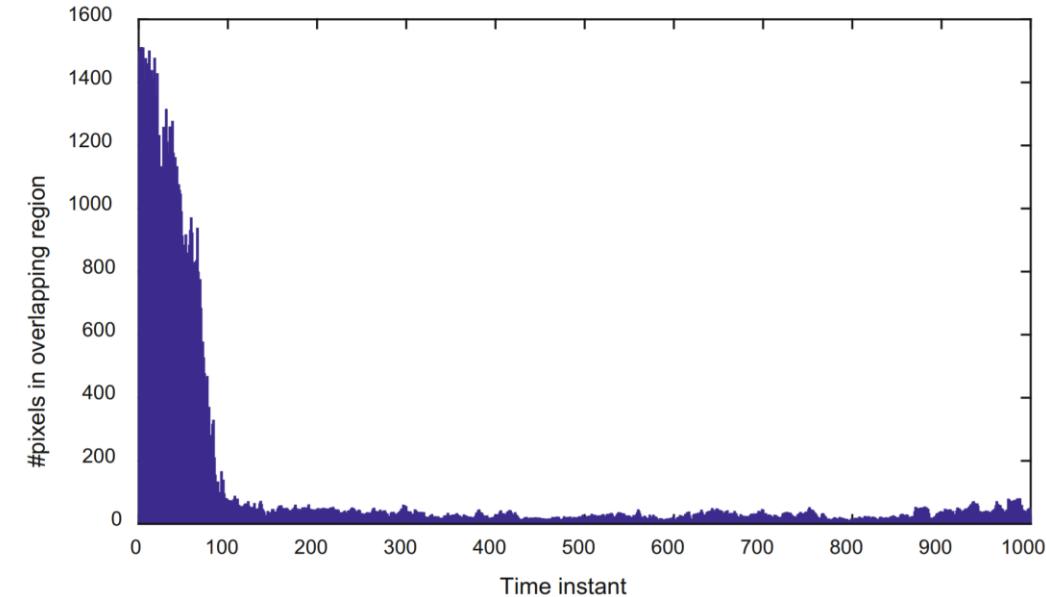
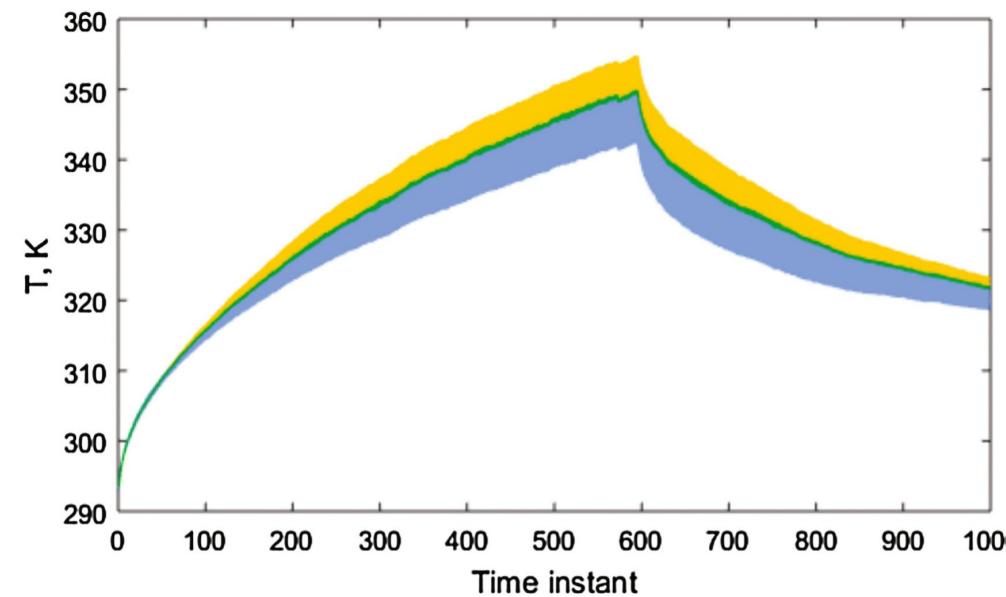
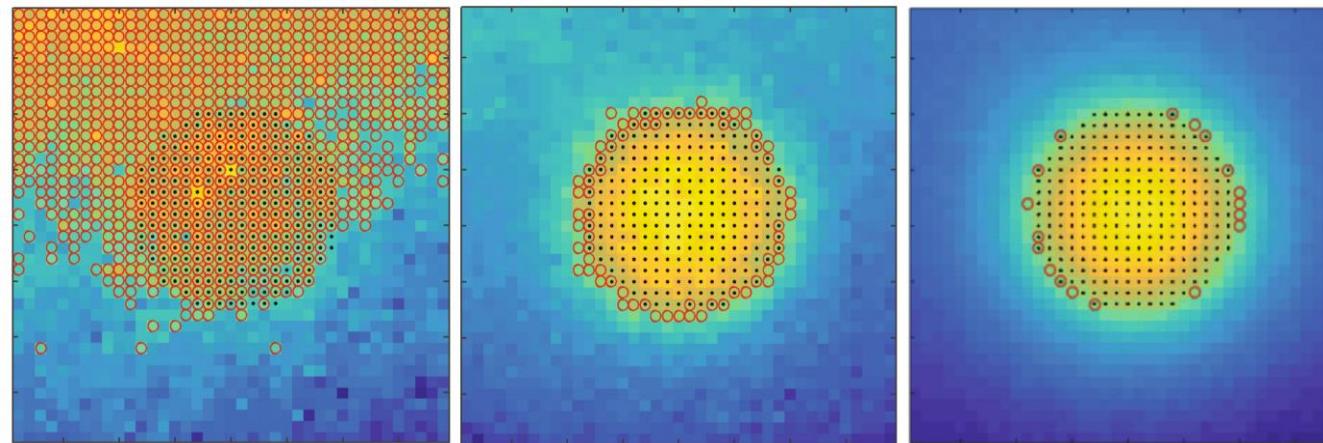
Przykład zastosowania drzewa klasyfikacyjnego

Wiązka krzywych N-S dla środkowego fragmentu termogramu



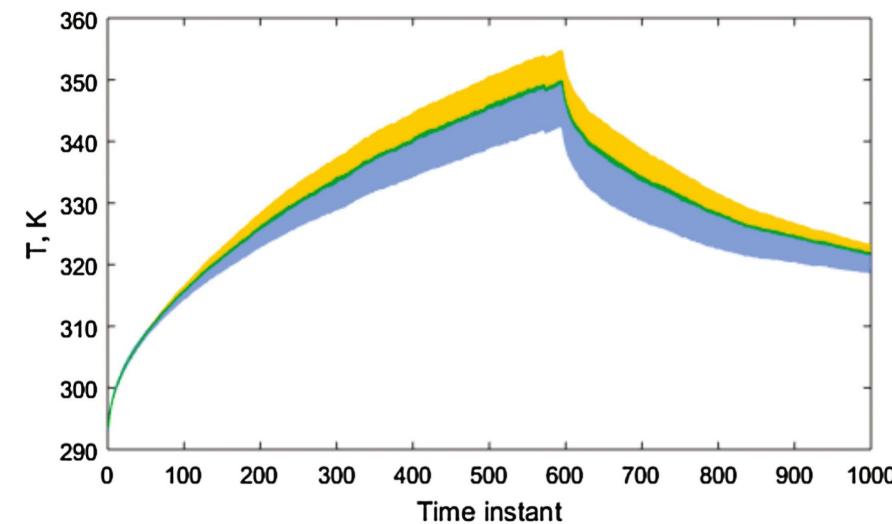
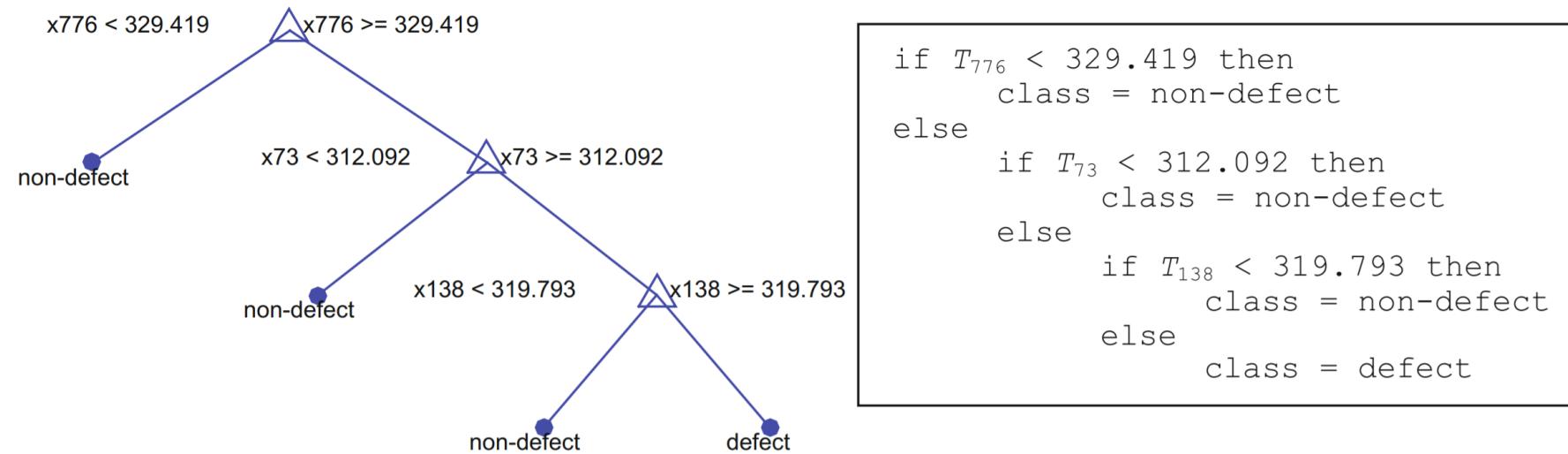
Przykład zastosowania drzewa klasyfikacyjnego

Rozmiar pasma wspólnego dla klas „bez defektu” i „defekt”



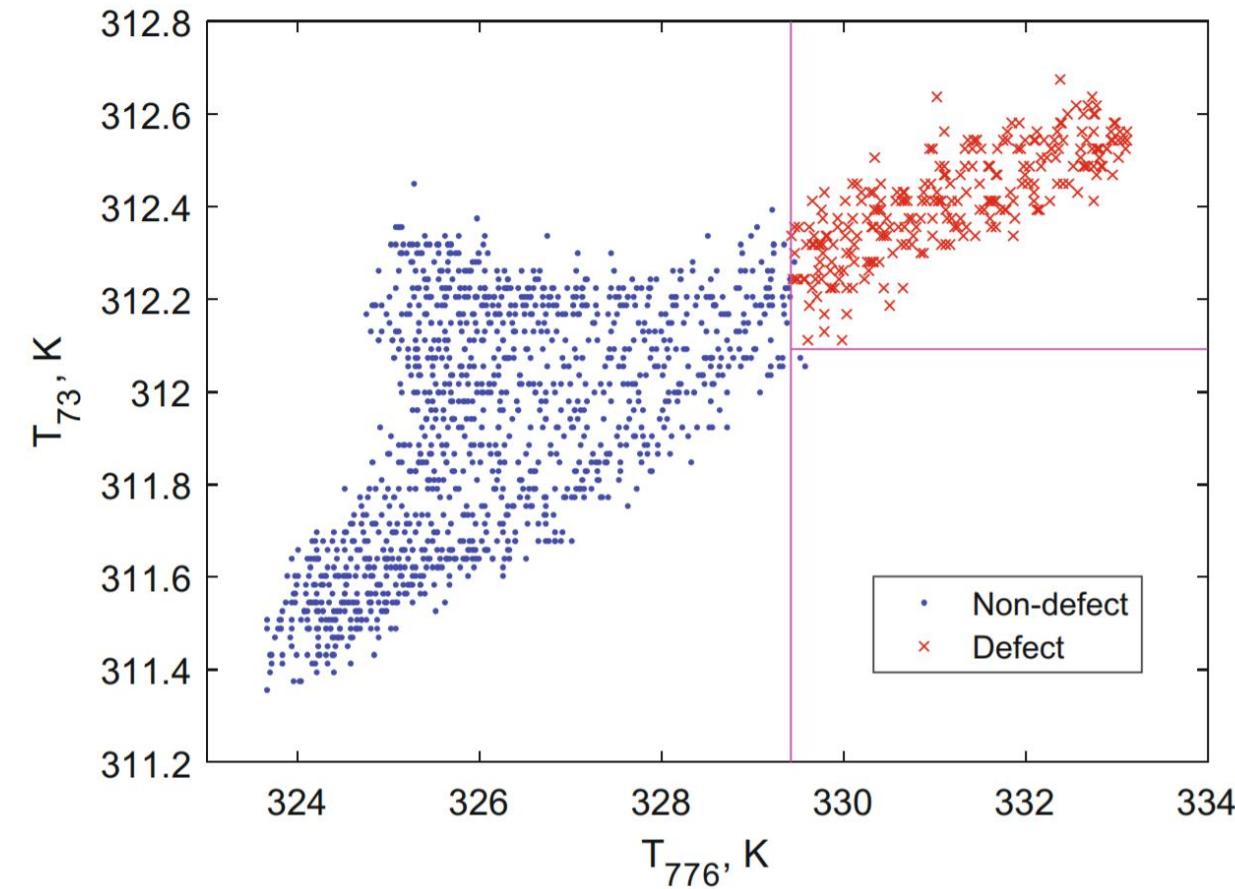
Przykład zastosowania drzewa klasyfikacyjnego

Drzewo decyzyjne dla środkowego fragmentu termogramu



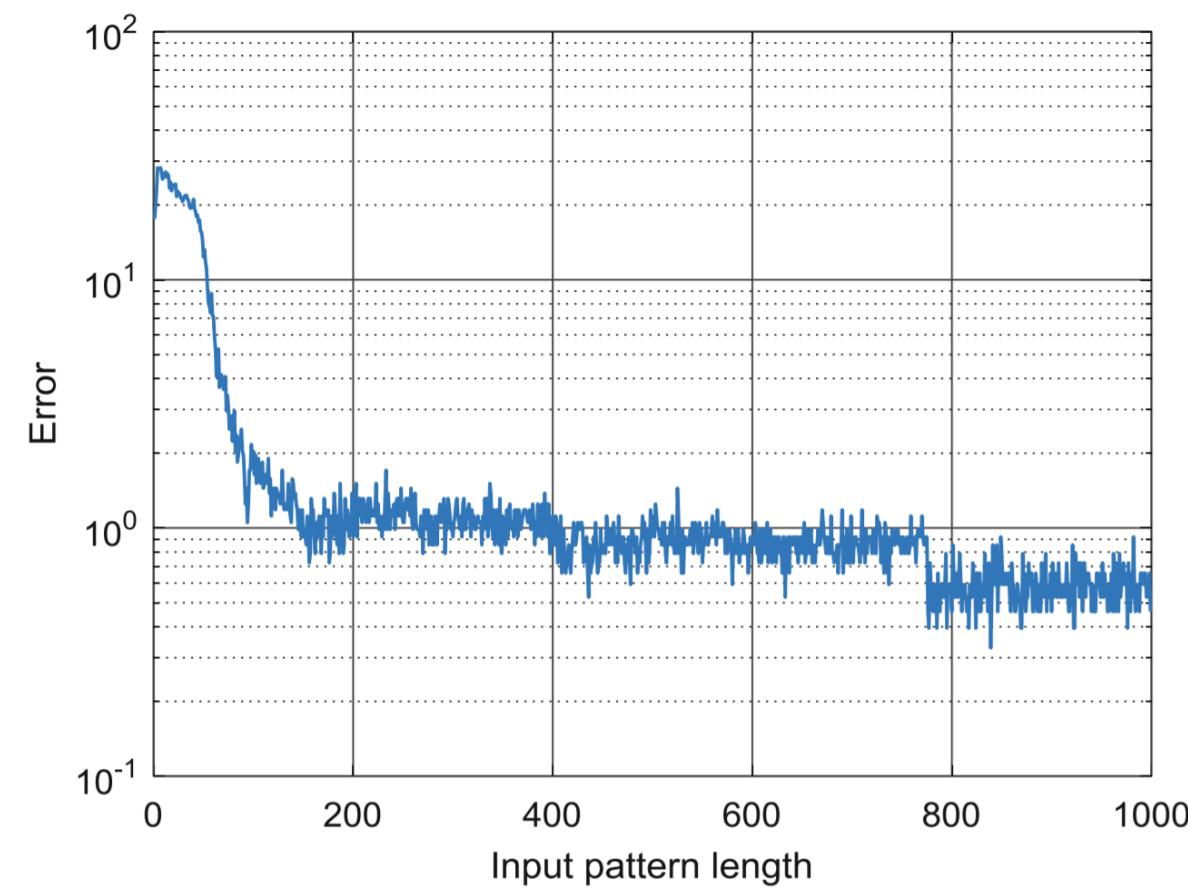
Przykład zastosowania drzewa klasyfikacyjnego

Podział przestrzeni wejściowej dla środkowego fragmentu termogramu



Przykład zastosowania drzewa klasyfikacyjnego

Jak długość przykładu x wpływa na błąd?



Przykład zastosowania drzewa klasyfikacyjnego

Zadanie rozpoznawania dziewięciu defektów

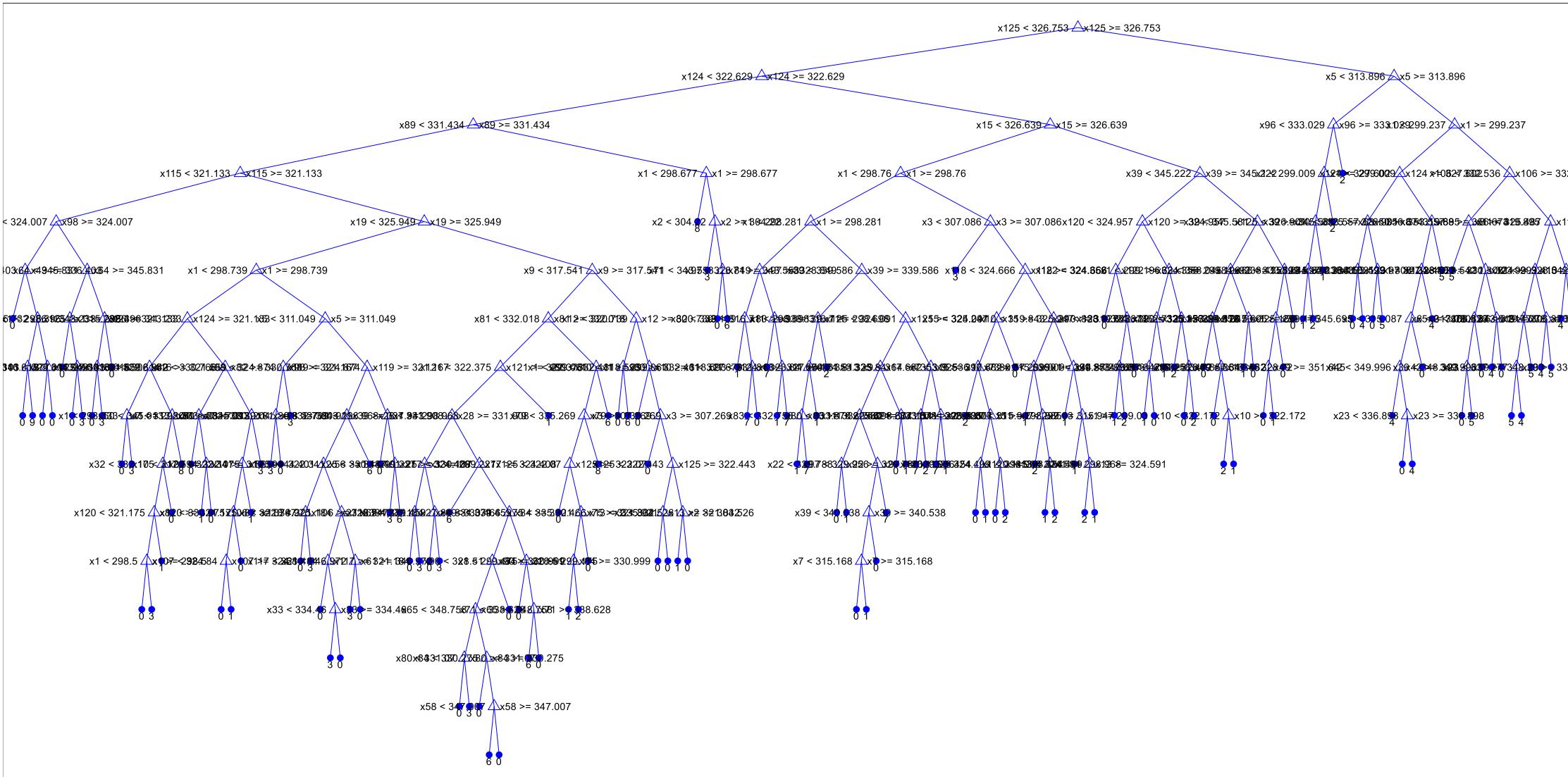
Problem niezrównoważenia klas:

- do klasy 0 należy ponad 90% przykładów
- do klasy 1 należy ok. 3% przykładów
- do klasy 9 należy ok. 0,06% przykładów

Zagrożenie – klasy o niewielkich licznościach będą rozpoznawane z niską dokładnością

Przykład zastosowania drzewa klasycyjnego

Drzewo decyzyjne dla dziesięciu klas



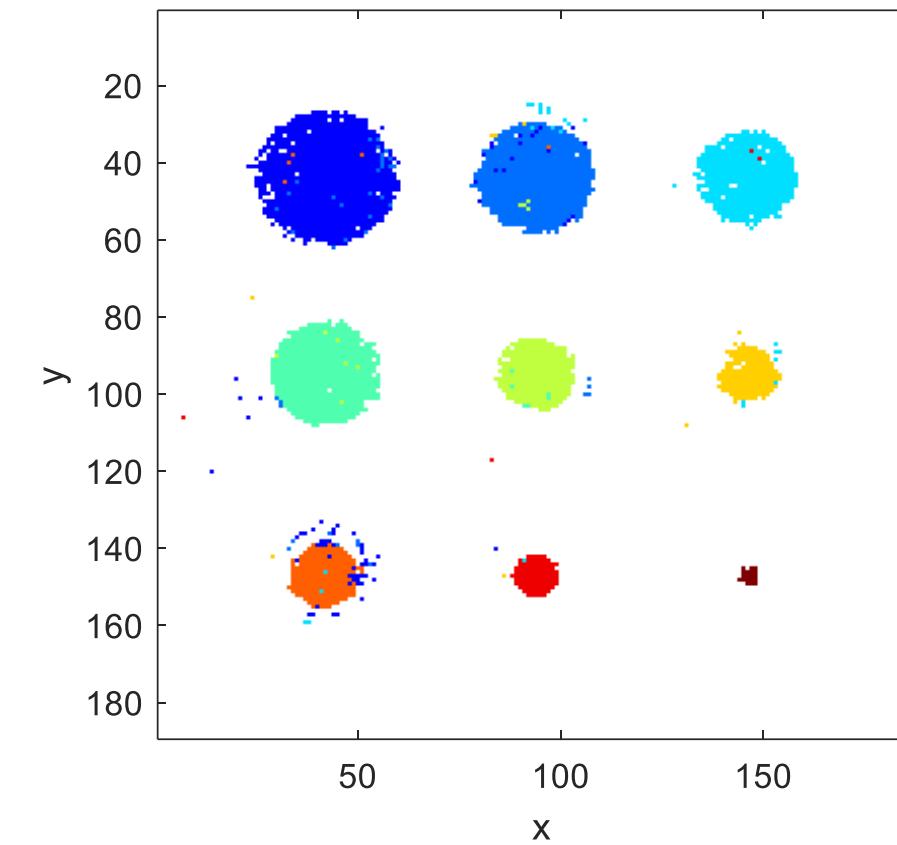
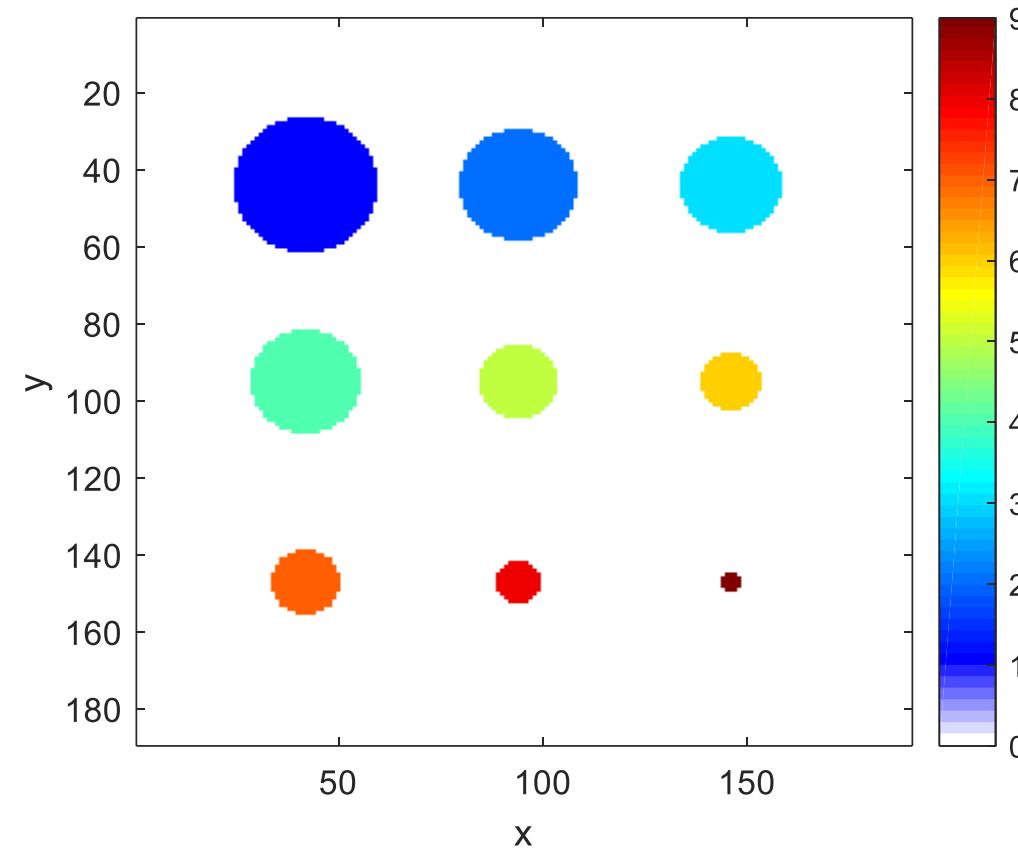
Przykład zastosowania drzewa klasyfikacyjnego

Tablica pomyłek dla dziesięciu klas

Confusion Matrix										
Output Class	0	1	2	3	4	5	6	7	8	9
	322223 89.7%	70 0.2%	30 0.1%	42 0.1%	14 0.0%	9 0.0%	23 0.1%	5 0.0%	2 0.0%	5 0.0%
	45 0.1%	873 2.4%	21 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	8 0.0%	0 0.0%	0 0.0%
	30 0.1%	22 0.1%	610 1.7%	1 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%
	38 0.1%	0 0.0%	1 0.0%	442 1.2%	0 0.0%	0 0.0%	1 0.0%	1 0.0%	1 0.0%	0 0.0%
	12 0.0%	0 0.0%	1 0.0%	0 0.0%	549 1.5%	6 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%
	11 0.0%	0 0.0%	1 0.0%	0 0.0%	12 0.0%	278 0.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%
	13 0.0%	0 0.0%	1 0.0%	2 0.0%	0 0.0%	0 0.0%	153 0.4%	0 0.0%	0 0.0%	0 0.0%
	12 0.0%	8 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	211 0.6%	0 0.0%	0 0.0%
	5 0.0%	0 0.0%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	94 0.3%	0 0.0%
99.5% 0.5%	89.7% 10.3%	91.7% 8.3%	90.4% 9.6%	95.1% 4.9%	94.9% 5.1%	86.4% 13.6%	93.8% 6.2%	96.9% 3.1%	76.2% 23.8%	98.7% 1.3%

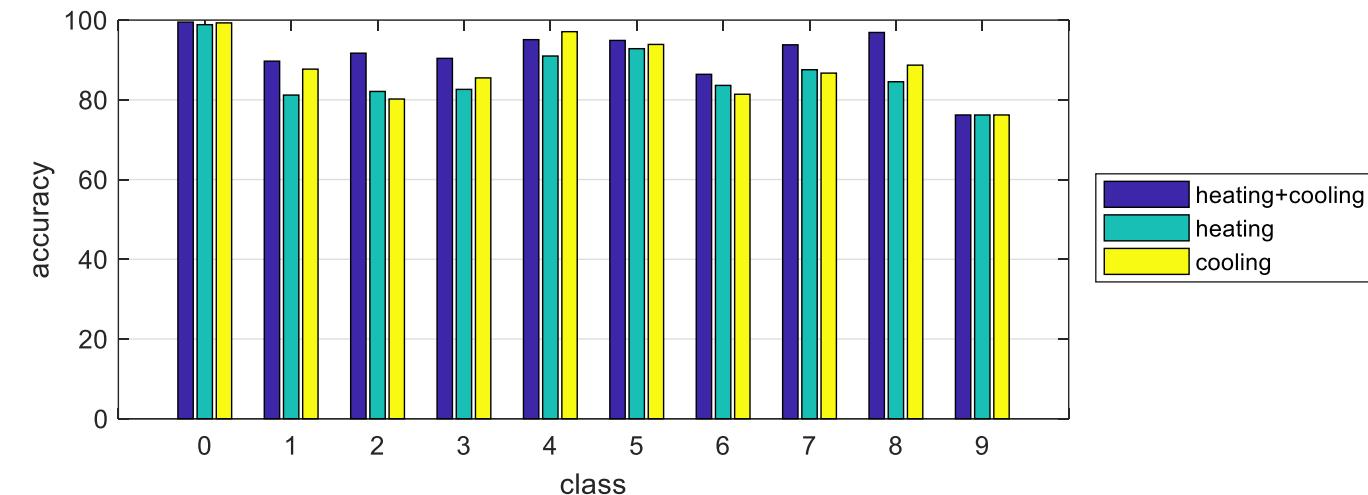
Przykład zastosowania drzewa klasyfikacyjnego

Poprawne i przewidywane rozkłady klas

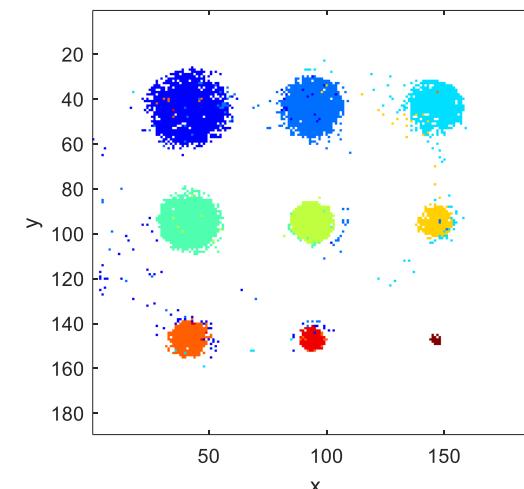


Przykład zastosowania drzewa klasyfikacyjnego

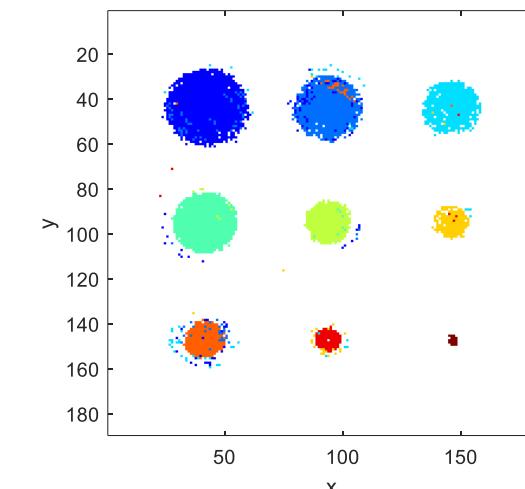
Dokładność klasyfikacji dla fazy nagrzewania i schładzania



Nagrzewanie

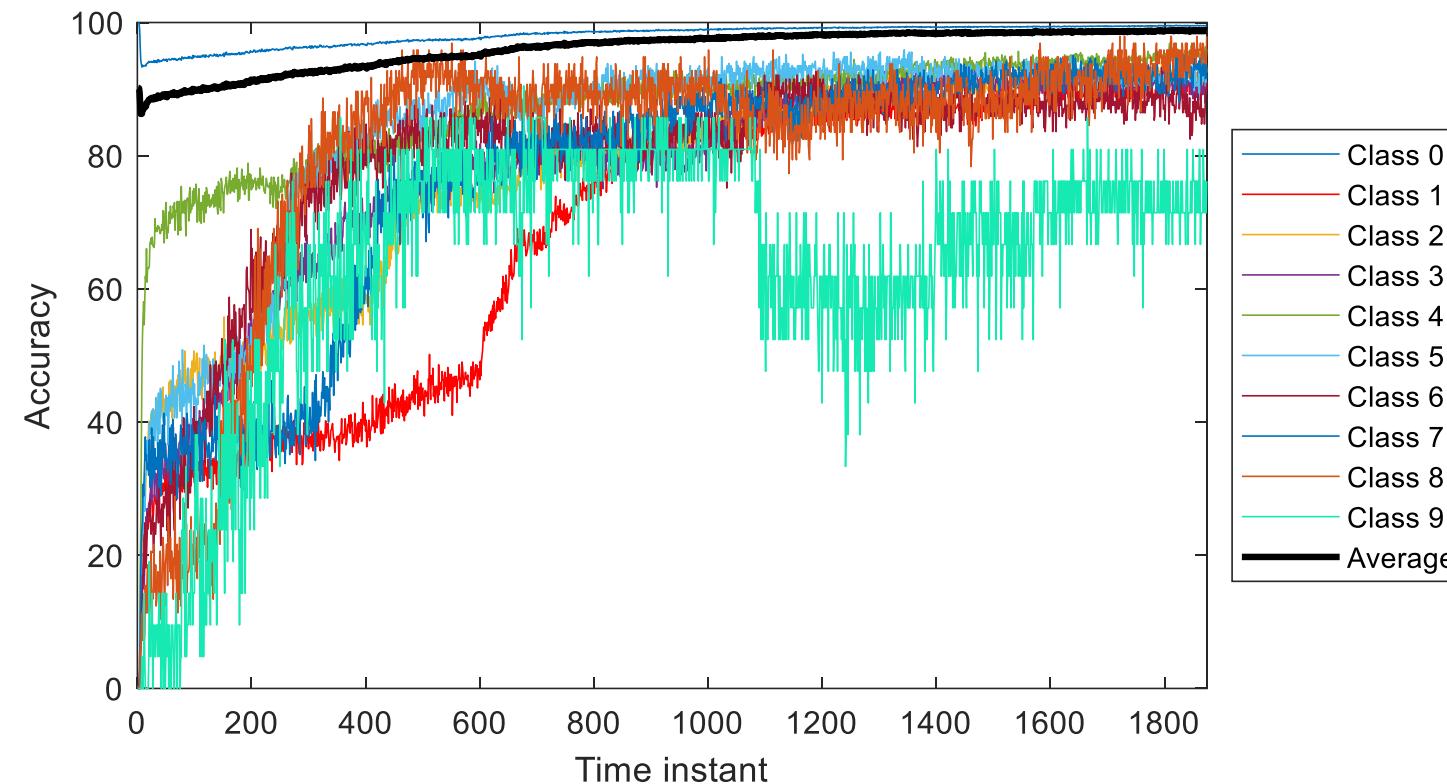


Schładzanie



Przykład zastosowania drzewa klasyfikacyjnego

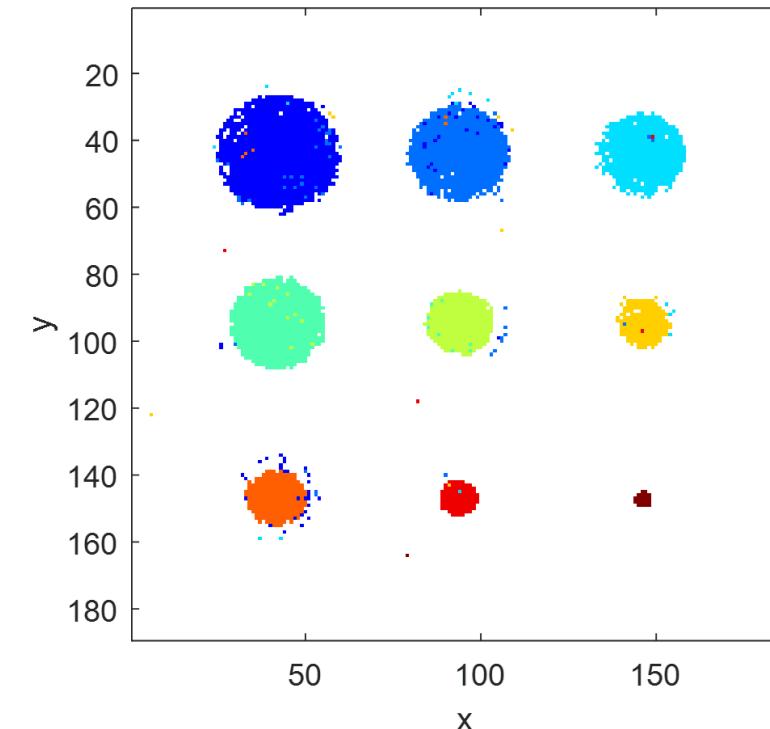
Dokładność w zależności od długości przykładu x



Przykład zastosowania drzewa klasyfikacyjnego

Dokładność w zależności od częstotliwości próbkowania – 1 Hz zamiast 15 Hz

Confusion Matrix											
Output Class	1	2	3	4	5	6	7	8	9	10	
	32229 89.7%	56 0.2%	36 0.1%	38 0.1%	10 0.0%	11 0.0%	17 0.0%	5 0.0%	4 0.0%	2 0.0%	99.4% 0.6%
	46 0.1%	889 2.5%	22 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 0.0%	0 0.0%	0 0.0%	92.4% 7.6%
	27 0.1%	23 0.1%	605 1.7%	2 0.0%	1 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	91.8% 8.2%
	28 0.1%	1 0.0%	0 0.0%	448 1.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	93.7% 6.3%
	20 0.1%	0 0.0%	0 0.0%	0 0.0%	555 1.5%	7 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	95.4% 4.6%
	10 0.0%	0 0.0%	0 0.0%	0 0.0%	11 0.0%	275 0.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	92.9% 7.1%
	17 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	158 0.4%	0 0.0%	1 0.0%	0 0.0%	89.8% 10.2%
	9 0.0%	4 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	215 0.6%	0 0.0%	0 0.0%	93.5% 6.5%
	5 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	91 0.3%	0 0.0%	92.9% 7.1%
	2 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	19 0.1%	90.5% 9.5%
	99.5% 0.5%	91.4% 8.6%	91.0% 9.0%	91.6% 8.4%	96.2% 3.8%	93.9% 6.1%	89.3% 10.7%	95.6% 4.4%	93.8% 6.2%	90.5% 9.5%	98.8% 1.2%



Podsumowanie

- Model interpretowalny – reprezentacja drzewiasta lub zbiór reguł decyzyjnych "jeśli – to"
- Czytelny algorytm konstrukcji modelu
- Działanie na atrybutach nominalnych, porządkowych i ciągłych
- Nie wymaga specjalnego przygotowania danych
- Wbudowany mechanizm selekcji atrybutów
- Podział przestrzeni cech na hiperprostopadłościany
- Lokalna aproksymacja funkcji stałą wewnątrz hiperprostopadłościanu (aproksymacja dyskretna)
- Zależnie od funkcji docelowej drzewo decyzyjne może pełnić rolę klasyfikatora lub modelu regresyjnego
- Zachłanny algorytm optymalizacyjny nie zawsze prowadzi do optymalnych rozwiązań
- Model niestabilny – niewielkie zmiany w danych uczących mogą powodować duże zmiany modelu
- Na bazie drzew decyzyjnych budowane są komitety: lasy losowe, *boosted decision trees* (XGBoost, Light GMB, CatBoost)