



Wprowadzenie do klasyfikacji

ZeroR

- Odpowiada zawsze tak samo
- Decyzja to klasa większościowa ze zbioru uczącego

A	B	X
1	5	T
1	7	T
1	5	T
1	5	F
2	7	F

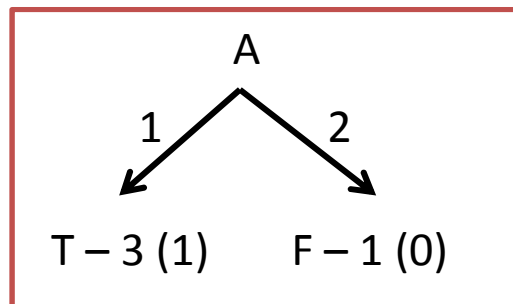
Tutaj jest więcej obiektów klasy T, więc klasyfikator ZeroR będzie zawsze odpowiadał T niezależnie od danych wejściowych

One rule model (OneR)

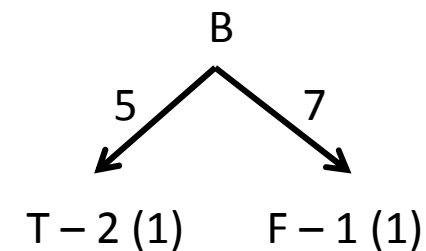
- Bardzo prosty mechanizm klasyfikacji
- Tworzy 1-dno poziomowe drzewo decyzyjne.

A	B	X
1	5	T
1	7	T
1	5	T
1	5	F
2	7	F

		X	
		T	F
A	1	3	1
	2	0	1

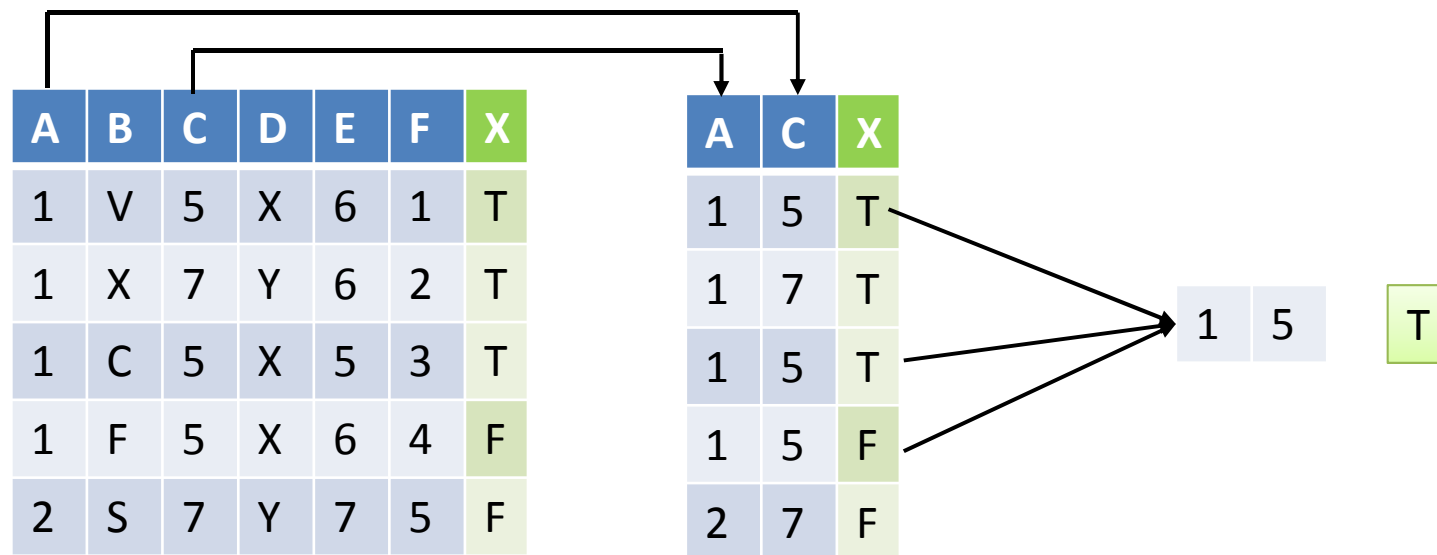


		X	
		T	F
B	5	2	1
	7	1	1



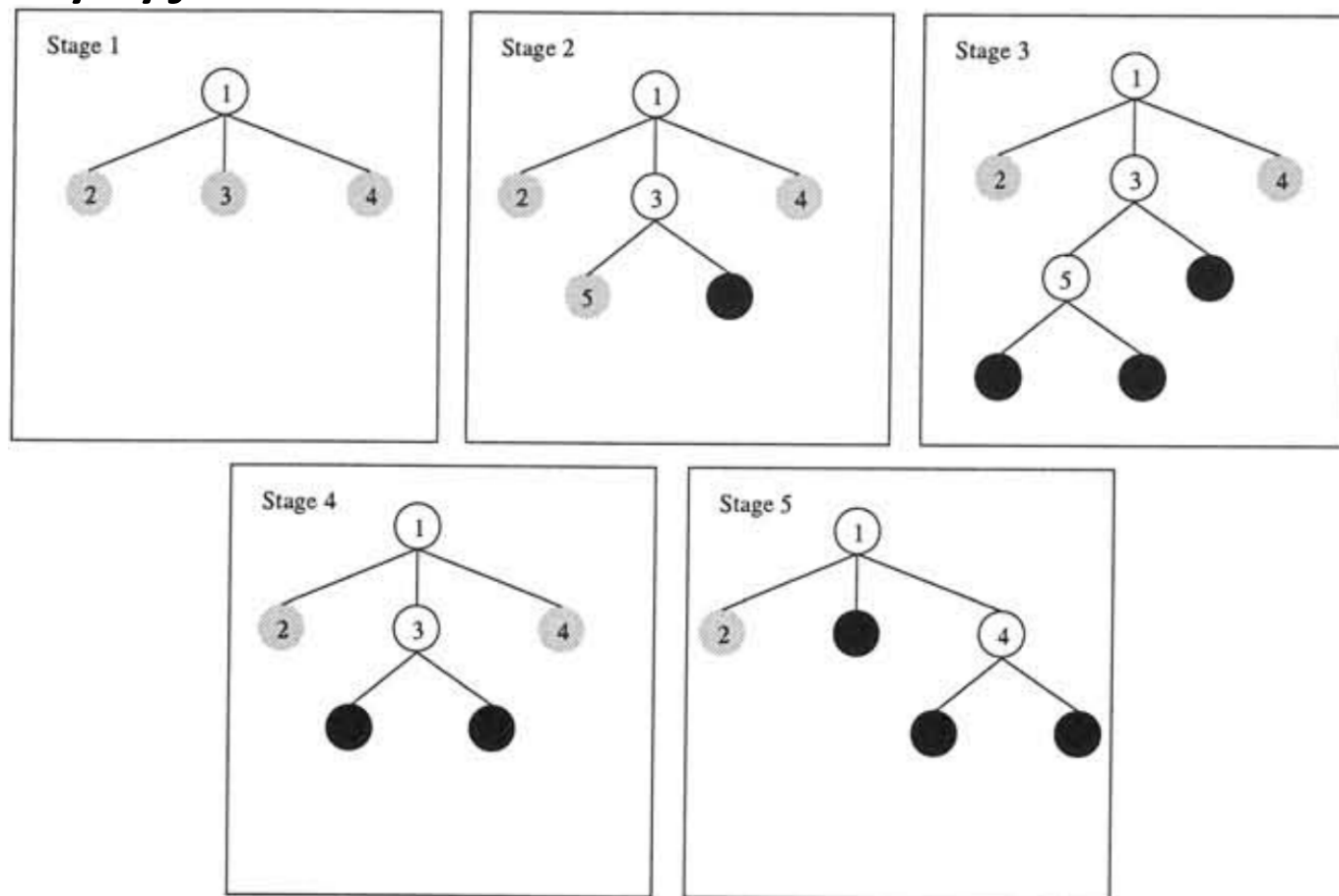
Decision table

- Mechanizm klasyfikacji trochę podobny do OneR i iBk
- Wykorzystuje „okrojone” przykłady uczące do klasyfikacji



PART (1)

- Skomplikowany algorytm generujący reguły decyzyjne.



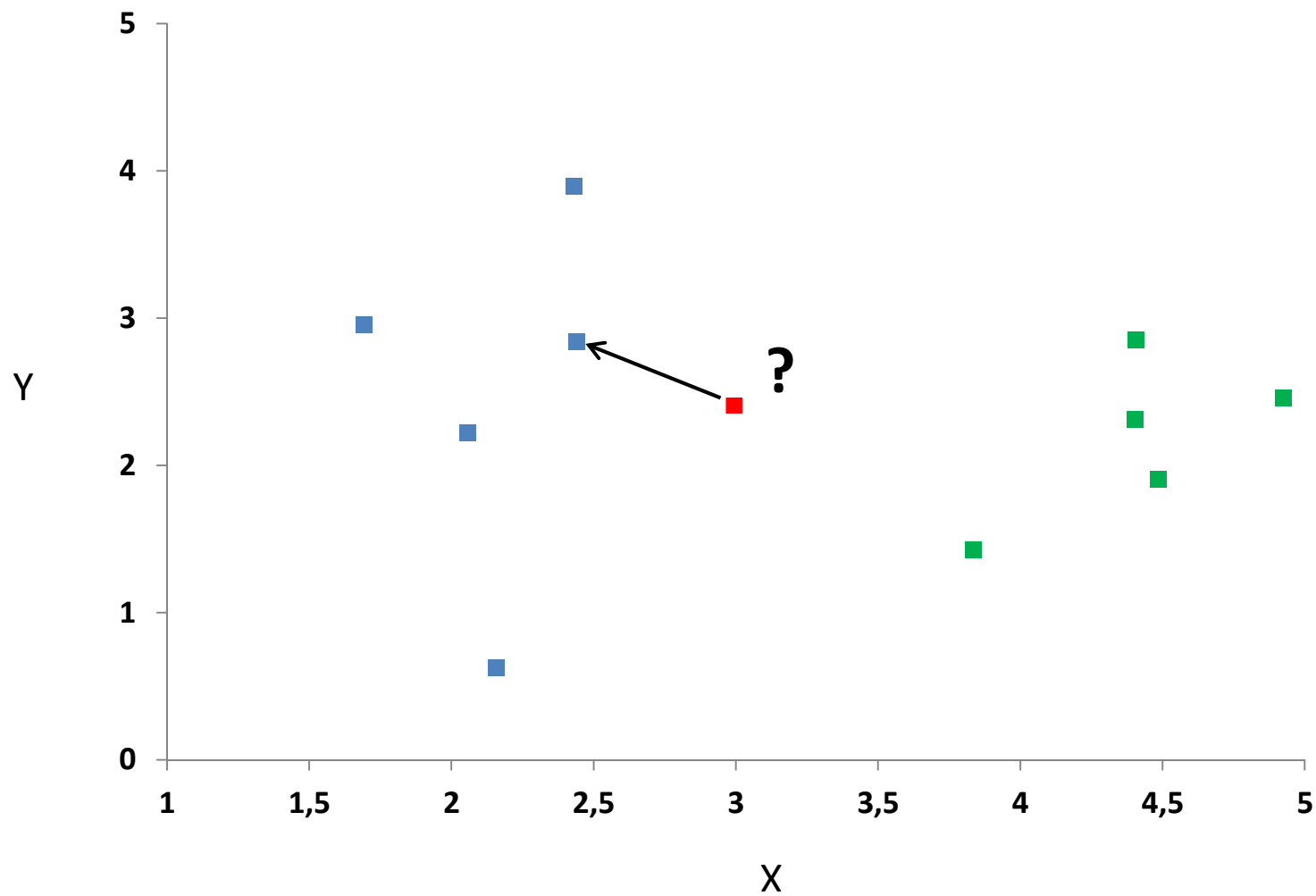
PART (2)

- Wygenerowanie jednej reguły działa następująco:
 1. Zaczynamy budowę drzewa tak jak normalnie się to robi, np. w C4.5
 2. Wybieramy liść w którym zbiór przykładów ma najmniejszą entropię
 3. Rekurencyjnie budujemy drzewo dalej.
 4. Kiedy dany węzeł w którym się znajdujemy posiada jedynie dzieci obcinamy go stosując standardowe algorytmy obcinania węzłów drzewa.
 5. Powtarzamy obcinanie aż jakieś obcięcie nie zostanie wykonane.
 6. Z powstałych liści wybieramy taki, który opisuje największą liczbę przykładów i tworzymy z niego regułę.
 7. Usuwamy przykłady pokryte przez utworzoną regułę ze zbioru i kontynuujemy aż pokryjemy wszystkie przykłady bądź utworzymy zadaną liczbę reguł.

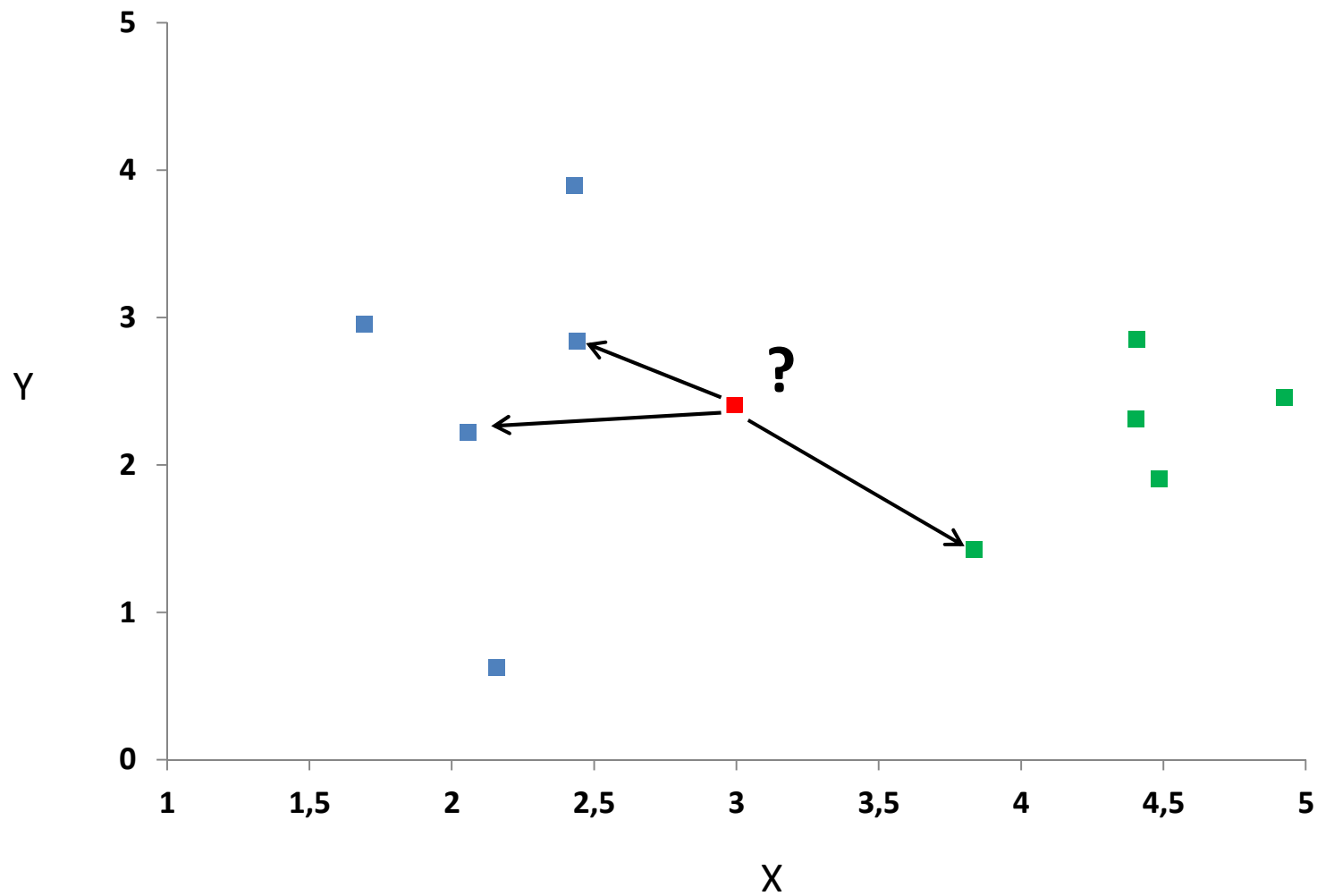
Conjunctive Rule

- Algorytm buduje jedną regułę.
 - W pierwszym etapie reguła jest budowana poprzez dodawanie warunków na atrybutach i łączenie ich za pomocą AND. Ocena testu odbywa się na podstawie miary entropii.
 - W drugim etapie reguła jest obcinana za pomocą metody Reduced Error Pruning (REP).
- Jeżeli przykład nie pasuje do reguły to odpowiedzią jest klasa większościowa wśród przykładów uczących nie pokrytych przez tą regułę.

IB1



IBk/K*



J48/C4.5

- Algorytm klasyfikacji opartej o drzewa decyzyjne.
- Rozszerzenie algorytmu ID3
- Obsługuje atrybuty nominalne i ciągłe
- „Przycina” za duże gałęzie
- Działa nawet dla zestawów danych z brakującymi wartościami
- J48 to open sourceowa implementacja C4.5 w Wece.

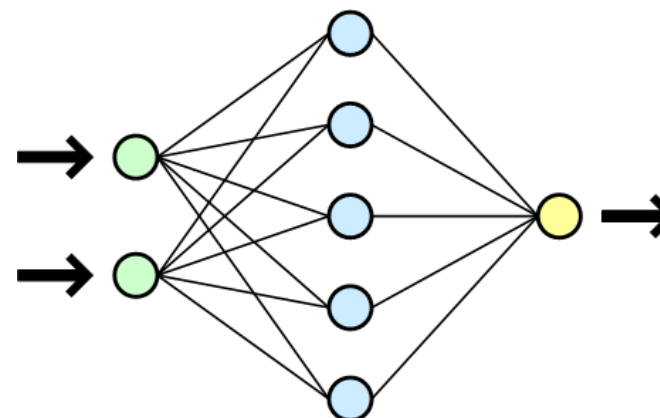


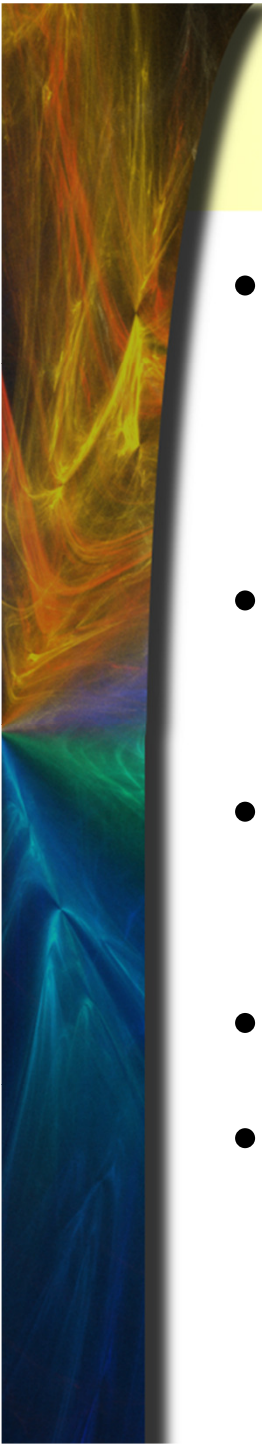
Multilayer Perceptron

- Klasyfikator oparty o sieć neuronową.
- Składa się z wielu warstw.
 - Wejściowej: tutaj na wejścia neuronów podaje się dane klasyfikowanego przypadku
 - Wyjściowej: wyjścia neuronów tej warstwy zwracają wynik klasyfikacji
 - Warstw pośrednich przekazujących wyniki pośrednie obliczeń pomiędzy sąsiednimi warstwami.

Multilayer Perceptron

- Każdy neuron jako wejście pobiera sumę wyników ze wszystkich neuronów z poprzedniej warstwy przemnożonych przez współczynniki zdefiniowane w synapsach.
- Jeżeli suma ta przekracza zadany próg, to wynikiem neuronu jest 1. W przeciwnym wypadku 0.





Ocena jakości klasyfikacji

- Zbiór danych dzielony jest na dwa podzbiory
 - Zbiór uczący
 - Zbiór testujący
- Klasyfikator budowany jest na podstawie zbioru uczącego.
- Zbudowany klasyfikator testowany jest na przykładach ze zbioru testującego.
- Wynikiem takiego testu jest macierz pomyłek.
- Na podstawie wartości z macierzy pomyłek oblicza się różne miary.

Macierz pomyłek

Zaklasyfikowano jako:

Faktyczna klasa:

	NIESTRAWNOŚĆ	ALERGIA	PRZECHŁODZENIE	CZARNA OSPA
NIESTRAWNOŚĆ	334	99	1	20
ALERGIA	13	234	23	100
PRZECHŁODZENIE	5	21	315	91
CZARNA OSPA	12	10	25	153

Miary jakości klasyfikacji

Alergia:		Zaklasyfikowano jako:	
Faktyczna klasa:		ALERGIA	NIE ALERGIA
	ALERGIA	TP 234	FN 136
	NIE ALERGIA	FP 130	TN 802

$$precision = \frac{TP}{TP+FP}$$

$$tp - rate = recall$$

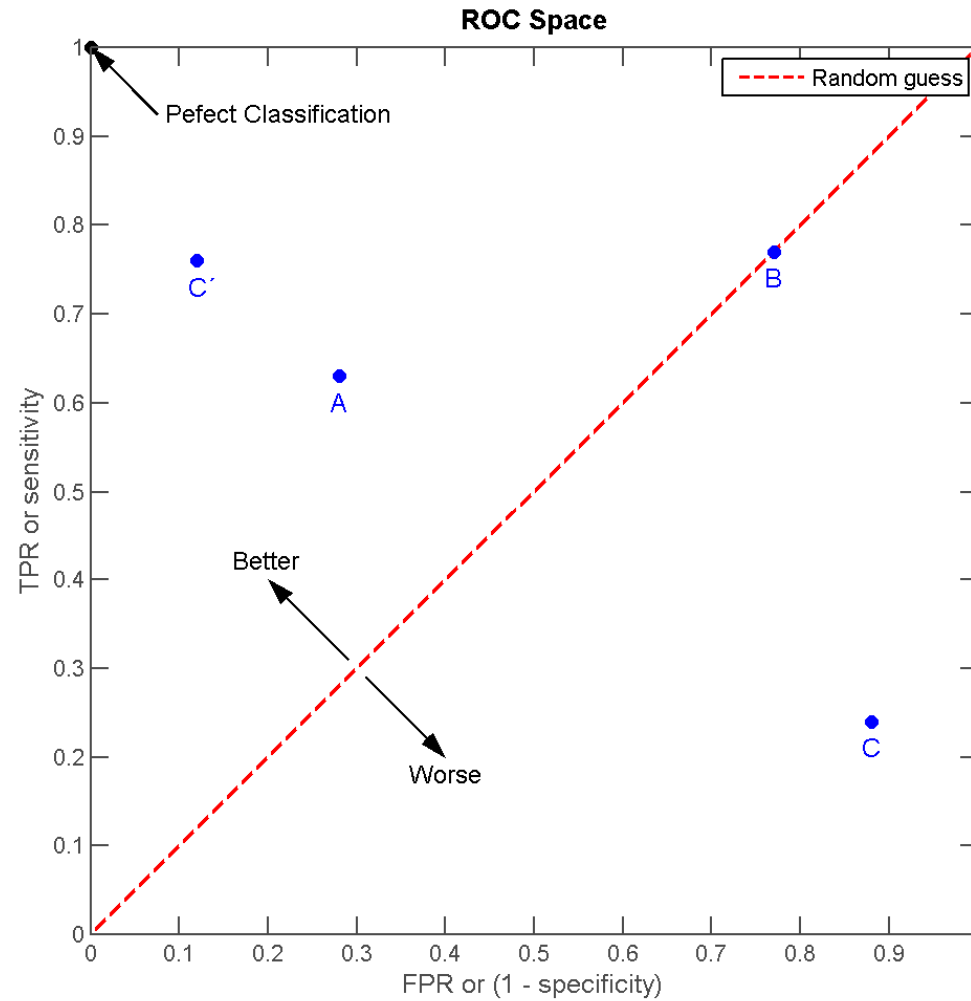
$$recall = \frac{TP}{TP+FN} \text{ (sensitivity)}$$

$$fp - rate = \frac{FP}{FP+TN}$$

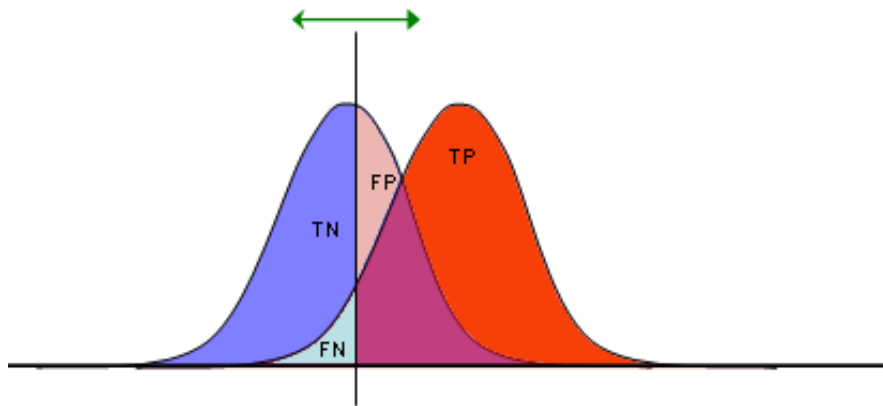
$$F - measure = \frac{2 * precision * recall}{precision + recall}$$

$$specificity = \frac{TN}{TN+FP}$$

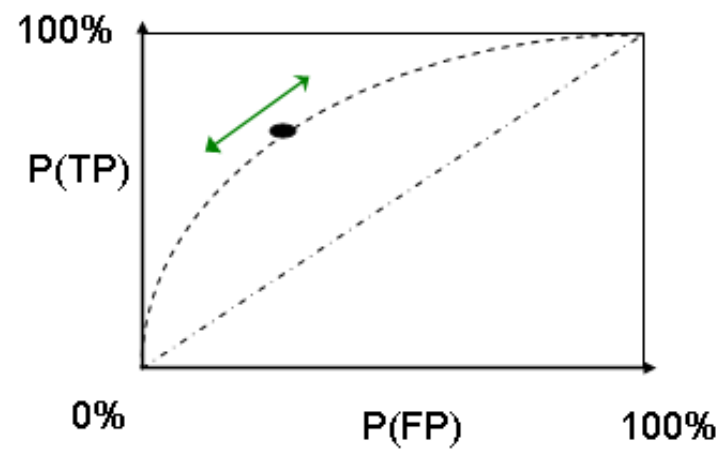
Przestrzeń ROC

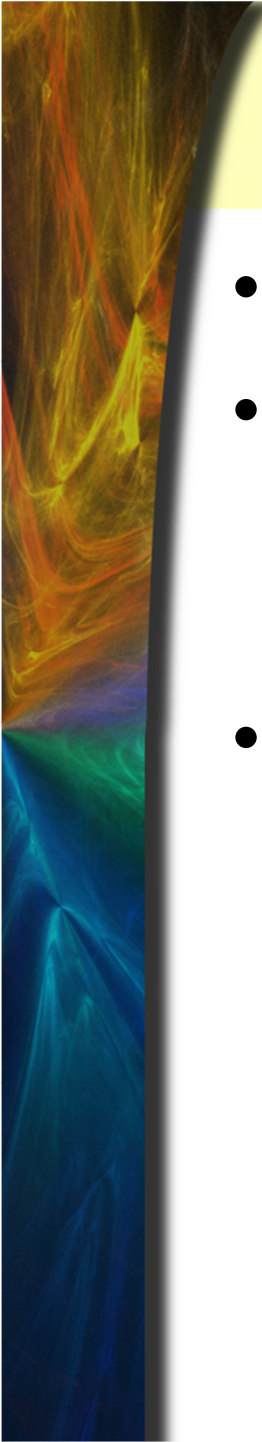


Krzywa ROC



TP	FP
FN	TN
1	1





Jak podzielić zbiór danych?

- Często $\frac{2}{3}$ zbiór uczący, $\frac{1}{3}$ zbiór testujący
- Uwaga! Istotne rekordy pozwalające na budowę dobrego klasyfikatora mogą zostać usunięte ze zbioru uczącego.
- Aby zatem uzyskać ocenę algorytmu klasyfikacji i jego parametrów, do danej dziedziny zastosowań stosujemy tzw. walidację skrośną albo n-fold crossvalidation.

Walidacja skrośna (1)

Dane										
	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}
1.	T_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}
2.	D_1	T_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}
3.	D_1	D_2	T_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}
4.	D_1	D_2	D_3	T_4	D_5	D_6	D_7	D_8	D_9	D_{10}
5.	D_1	D_2	D_3	D_4	T_5	D_6	D_7	D_8	D_9	D_{10}

.....

Walidacja skrośna (2)

- Cały zbiór danych, który posiadamy dzielony jest na n części (najczęściej 10). Następnie poniższa procedura wykonywana jest n razy:
 1. Weź k -tą część i uznaj ją za zbiór testujący
 2. Weź pozostałe części i uznaj je jako zbiór uczący
 3. Zbuduj klasyfikator na podstawie zbioru uczącego
 4. Sprawdź działanie klasyfikatora na zbiorze testującym
 5. Wyniki testowania umieść w macierzy pomyłek (zwiększ odpowiednie wartości z poprzedniej iteracji).
- Po przeprowadzeniu testów wyliczane są pozostałe miary jakości klasyfikacji na podstawie macierzy pomyłek.

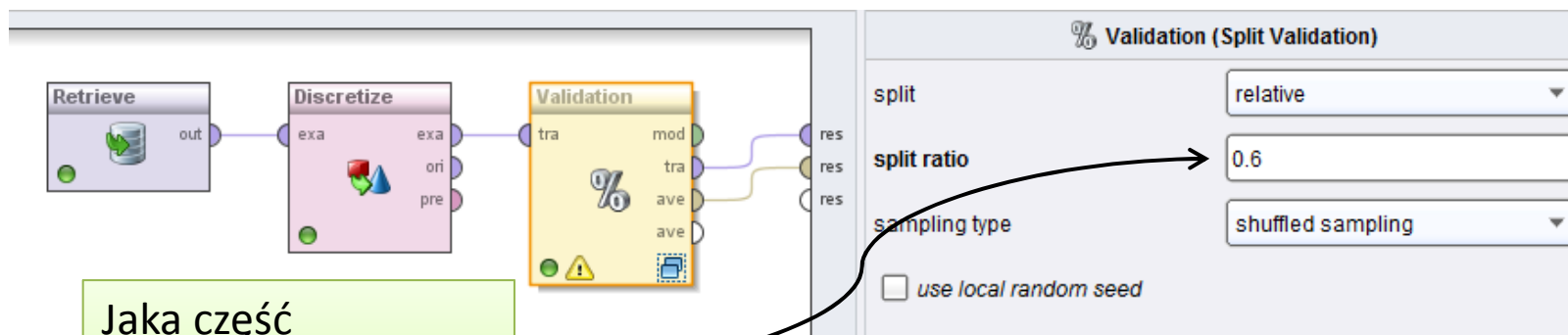
Leave one out

- Specjalny przypadek crosswalidacji
- Zbiór testowy składa się tylko z jednego przypadku.
- Odpowiada crosswalidacji dla $k = \text{liczba przykładów}$.
- Stosowany w sytuacjach, kiedy zbiór danych jest mały.

Ćwiczenie (weka)

- Pobierz pliki: weather.nominal.arff, car.arff i zoo.arff, labor.arff, vote.arff, iris.arff i spam.arff.
- Przeczytaj opisy tych plików w „Opisy zestawów danych”.
- Postępuj zgodnie z poleceniami pod linkiem „ocena jakosci.pdf”

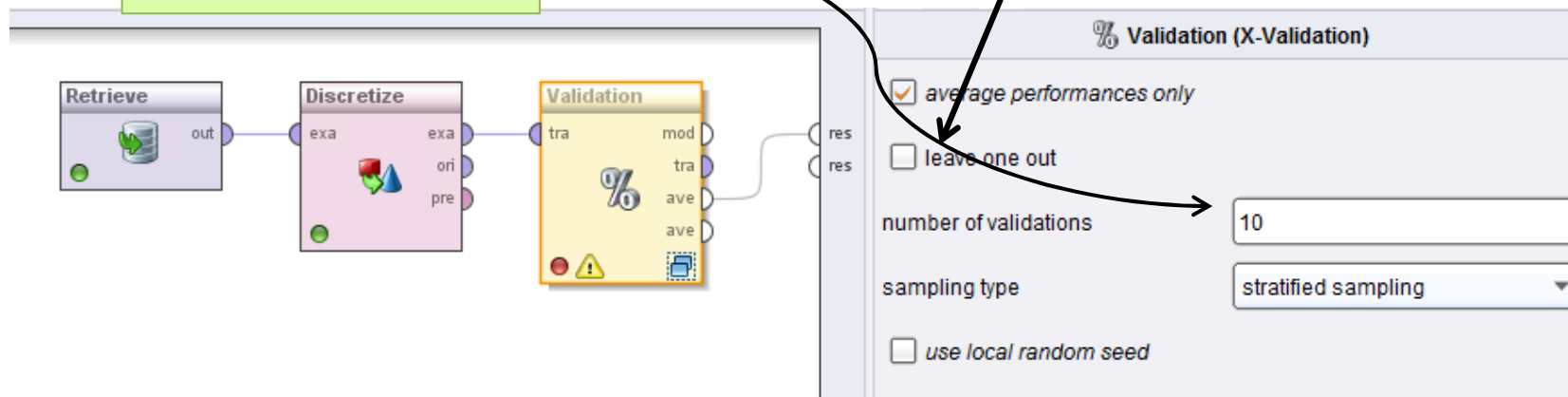
Ocena klasyfikatorów w RapidMiner



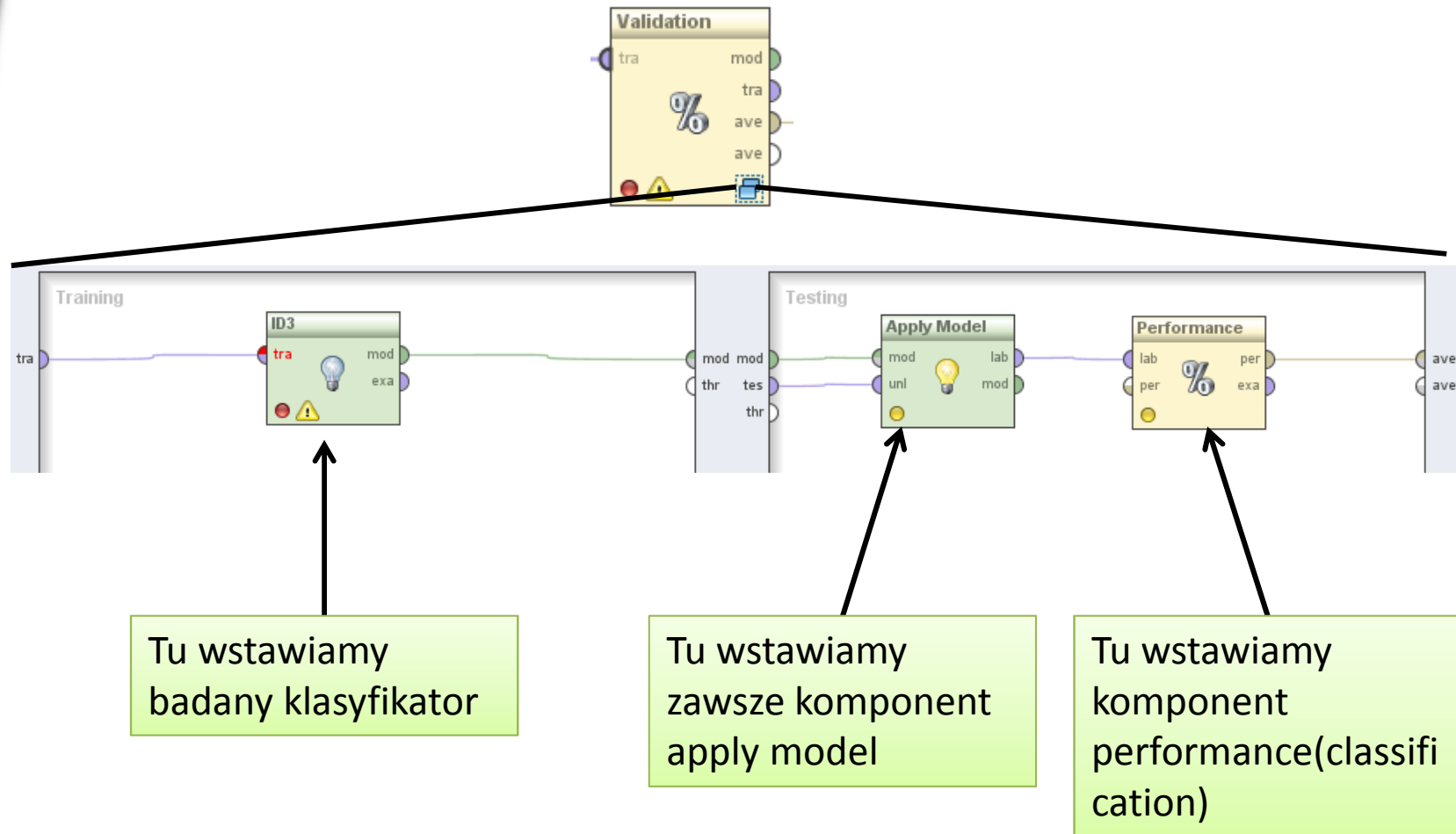
Jaka część przykładów ma być zbiorem uczącym

Na ile części dzielimy przykłady do crosswalidacji

Czy wykonujemy walidację leave one out?



Ocena klasyfikatorów w RapidMiner



Ćwiczenie 1 (rapid miner)

- Zbuduj przepływ testujący klasyfikator „Decision Tree” (podobny do C4.5).
- Do testowania wykorzystaj metodę crosswalidacji.
- Jako dane wejściowe wykorzystaj zbiór danych sonar, w którym każdy atrybut liczbowy został zdyskretyzowany przez podział na 5 binów.
- Porównaj jak sobie radzi ten klasyfikator do klasyfikatorów Single Rule, ID3, Naive Bayes i K-NN (dla $k=1$ i 10) (k-NN to to samo co IBx).

Ćwiczenie 2 (rapid miner)

- Jako dataset wybierz golf
- Atrybuty liczbowe zdyskretyzuj na 3 biny każdy
- Użyj metody crosswalidacji (10 foldów) do oceny jakości klasyfikatora Decision Tree.
- Dlaczego wyniki są takie złe?
- Użyj metody leave one out. Czy teraz lepiej? Dlaczego?