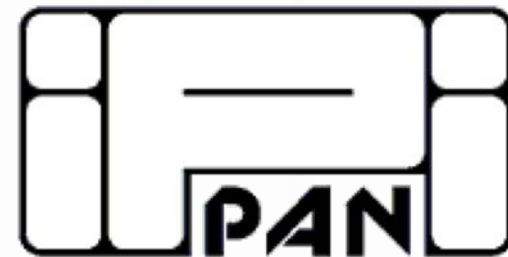




Zaawansowane Metody Analizy Danych w Biologii Molekularnej

Semest Letni 2018

Prowadzący



Zespół Biologii Obliczeniowej IPI PAN

- dr Magdalena Mozolewska (5 zajęć)
- dr Michał 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. Selekcja oraz budowanie nowych cech.
5. Pułapki w analizie danych o dużym rozmiarze.



AI



Sztuczna inteligencja (AI) ???



Inteligencja

- **Inteligencja** to zdolność uczenia się.
- **Inteligencja** to zdolność do aktywnego przetwarzania informacji w celu lepszego przystosowywania się do zmiennego środowiska.
- **Inteligencja** to zdolność rozwiązywania problemów.
- **Inteligencja** to zespół zdolności umysłowych umożliwiających jednostce sprawne korzystanie z nabytej wiedzy oraz skuteczne zachowanie się wobec nowych zadań i sytuacji.
- **Inteligencja** to zdolność do przetwarzania informacji na poziomie abstrakcyjnych idei.

[źródło WIKIPEDIA]

Sztuczna inteligencja (AI)

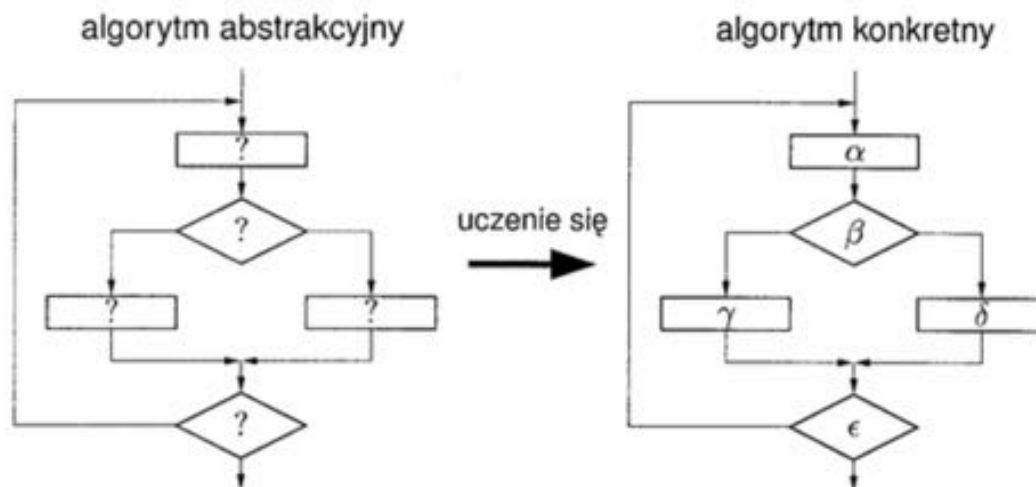
- Test Alana Turinga (1950)



- John McCarthy (1955) „AI to konstruowanie maszyn, o których działaniu dałoby się powiedzieć, że są podobne do ludzkich przejawów inteligencji”.
- “Jeśli maszyna zachowuje równie inteligentnie jak człowiek wtedy możemy uznać, że jest równie inteligentna jak człowiek.” (Alan Turing)
- dane wejściowe -> wiedza -> aplikacja/adaptacja

Maszynowe uczenie się

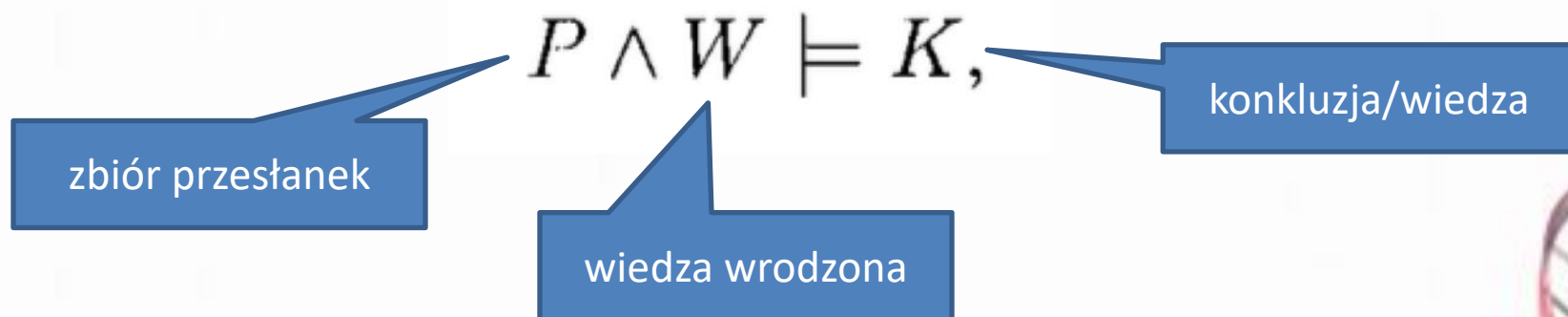
- Učeniem się systemu jest każda autonomiczna zmiana w systemie zachodząca na podstawie doświadczeń, która prowadzi do poprawy jakości jego działania [Cichosz, 2000].
- Celem uczenia maszynowego się jest tworzenie automatycznego systemu potrafiącego doskonalić się przy pomocy zgromadzonego doświadczenia (czyli danych) i nabywania na tej podstawie nowej wiedzy bez uprzedniego wyraźnego zaprogramowania.



Rys. 1.1. Uczenie się jako konkretyzacja algorytmu

Wnioskowanie – indukcja/dedukcja

- **Dedukcja** to rozumowanie od ogółu do szczegółu. Wyprowadzamy z teorii uznanej za prawdziwą jej logiczne następstwa, które też uznajemy za prawdziwe. Dedukcja zakłada wiarygodność zarówno przesłanek, jak i wniosków.



z P wynika K

Wnioskowanie – indukcja/dedukcja

- **Indukcja** sprowadza się do formułowania ogólnych teorii na podstawie jednostkowych eksperymentów i obserwacji faktów; czyli przechodzenie od szczegółu do ogółu.

$$P \wedge W \models K,$$

Konkluzja/wiedza

wiedza wrodzona

zbiór trenujący

z fałszywego P wynika fałsz K

Co z tego wynika?

- (Jeżeli) **pies** => (to) **4 nogi**



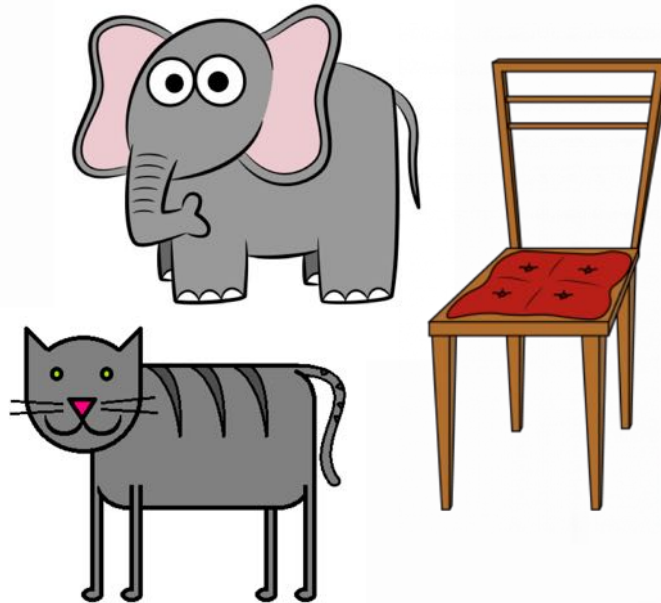
Co z tego wynika?

- (Jeżeli) **pies** \Rightarrow (to) **4 nogi**
- (Jeżeli) **4 nogi** \Rightarrow (to) **pies???**



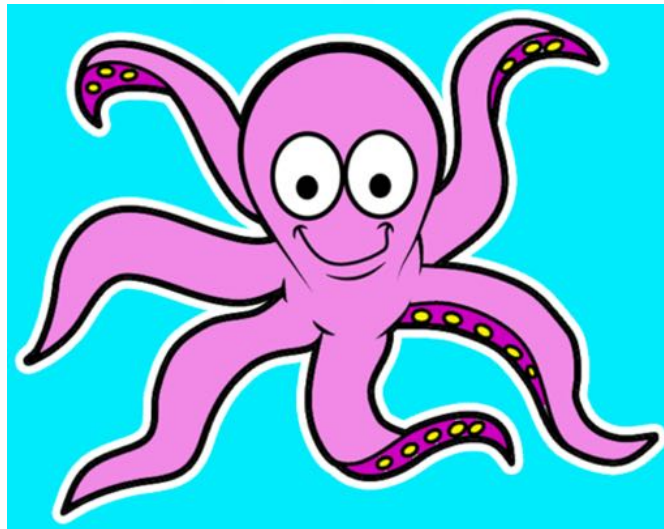
Co z tego wynika?

- (Jeżeli) **pies** => (to) **4 nogi**
- ~~(Jeżeli) **4 nogi** => (to) **pies**???~~



Co z tego wynika?

- (Jeżeli) **pies** => (to) **4 nogi**
- ~~(Jeżeli) **4 nogi** => (to) **pies**???~~
- (Jeżeli nie) **~4 nogi** => (to nie) **~pies**



Dane zmienne/cechy/atributy (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

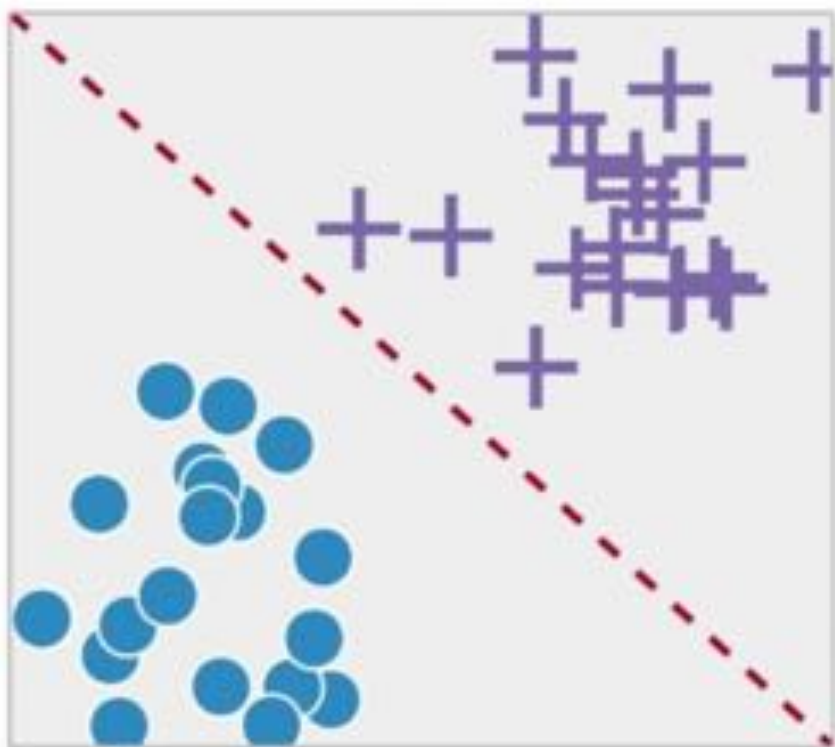
Zmienne/Cechy/Atrybuty

- **Nominalne/Dyskretne np.**
 - Kolory
 - Płeć
 - Typ choroby (raka)
 - itp.
- **Numeryczne/Ciągłe np.**
 - Wiek
 - Waga
 - Wzrost
 - itp.
- **Porządkowe np.**
 - Dni tygodnia
 - Miesiące
 - Stopień zaawansowania choroby
 - itp.

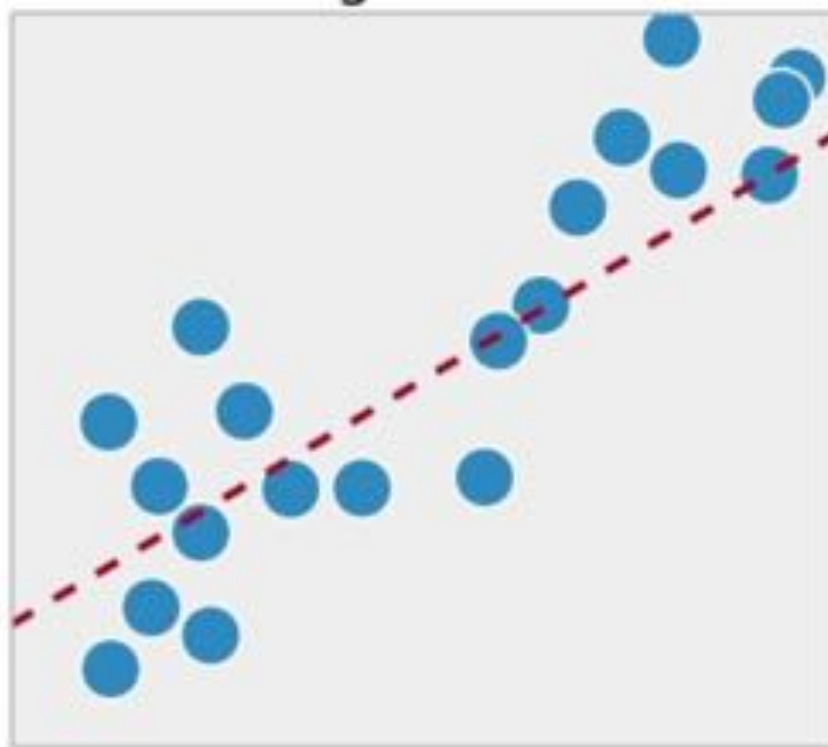


Klasyfikacja vs Regresja/Predykcja

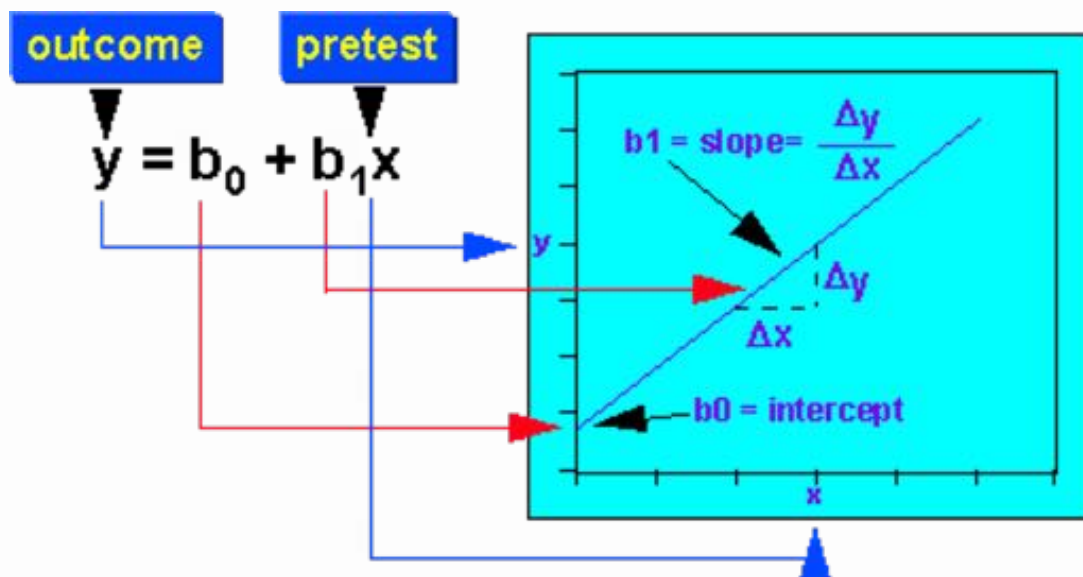
Classification



Regression



Model liniowy (regresja liniowa)



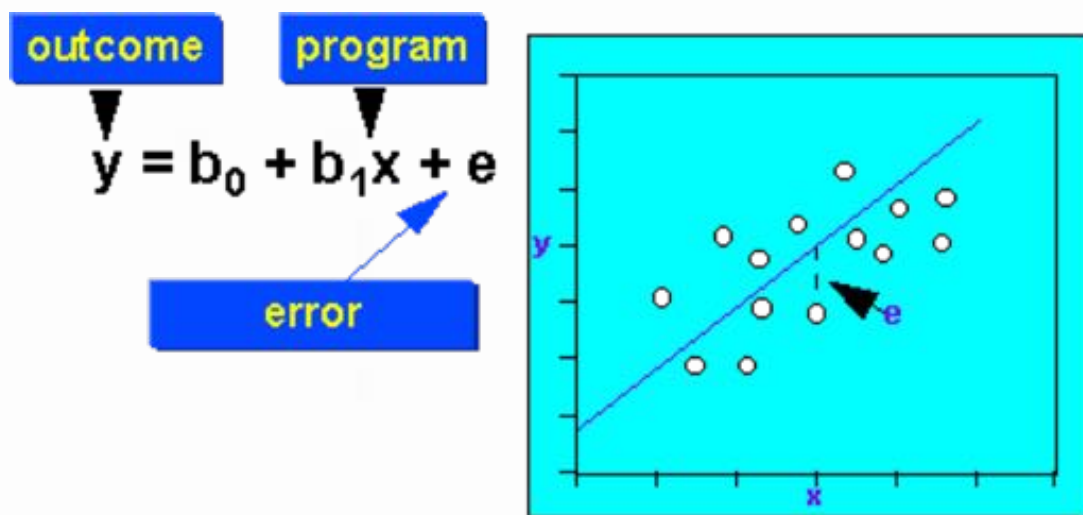
Gdy jedna zmienna wejściowa x

- $y = b_0 + b_1x + e$

Wielowymiarowa przestrzeń (n wymiarowa)

- $y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \dots + b_nx_n + e$

Model liniowy (regresja liniowa)



residuum = obserwacja – predykcja

- $e_i = y_i - \hat{y}_i$

metoda najmniejszych kwadratów

- $\min(\sum e_i^2)$

Formuła w R

```
#zmienna predykcyjna (objaśniana) znak  
tyldy
```

```
#i kropka oznaczająca użyj wszystkich  
zmiennych wejściowych (objaśniających)
```

```
>fit <- lm(hp~., mtcars)
```

```
# tu przykład jak uzyc formuly ze stringa
```

```
>f <- as.formula(paste0("Species","~."))
```

```
>class(f)
```

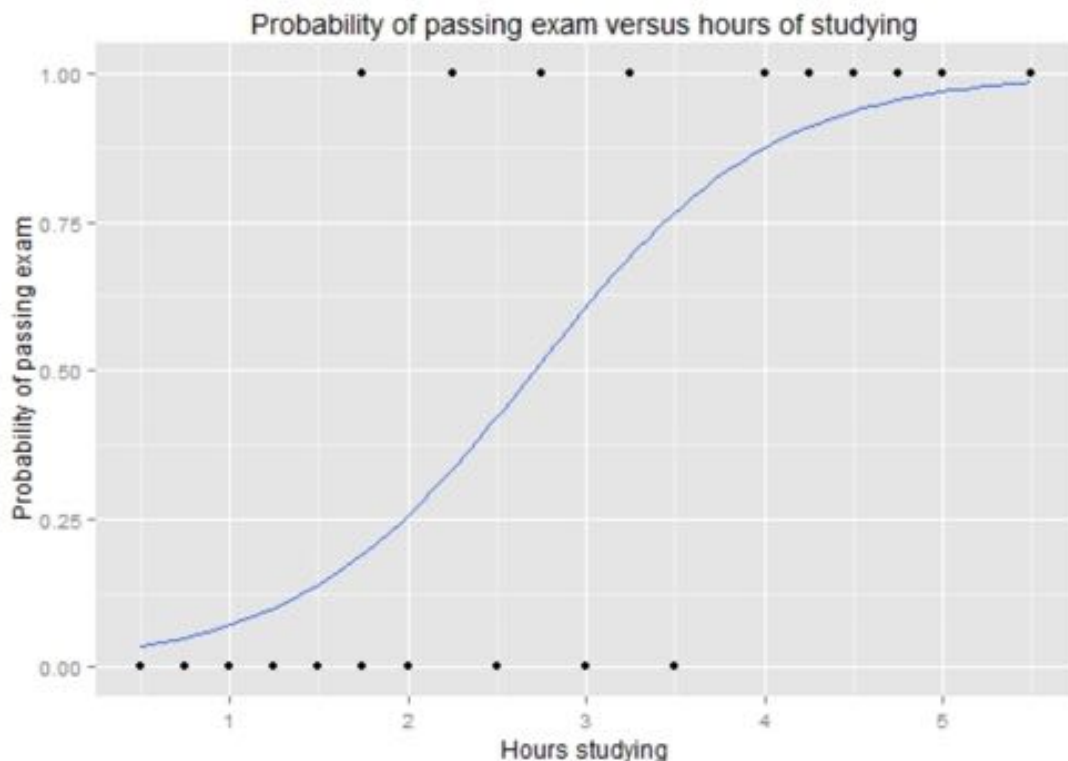


Model liniowy (regresja liniowa)

```
#model liniowy  
#pierwszy parametr to formula  
>fit <- lm(hp~., mtcars)  
  
>fit  
  
>plot(fit)  
  
>coefficients(fit) # model coefficients  
>fitted(fit) # predicted values  
>residuals(fit) # residuals  
#wartości wynikające z modelu vs  
oryginalne  
plot(mtcars$hp,fitted(fit))
```

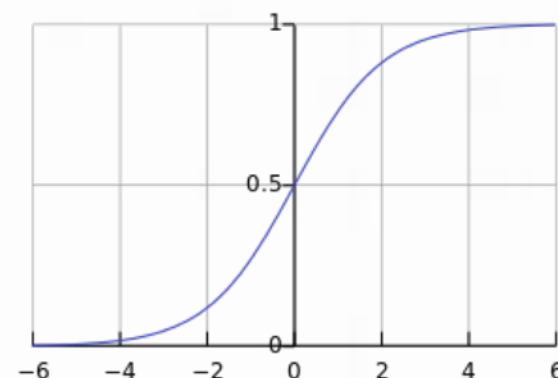


Regresja logistyczna – binarna decyzja



Funkcja

$$\sigma(a) = 1/(1+e^{-a})$$



$$b_0 + b_1x_1 + b_2x_2 + b_3x_3 = b^T x$$

$$P(\text{dec}|x) = \sigma(b^T x)$$

$$P(\text{dec}|x) = p = 1/(1+e^{-(b_0+b_1x_1+b_2x_2+b_3x_3)})$$

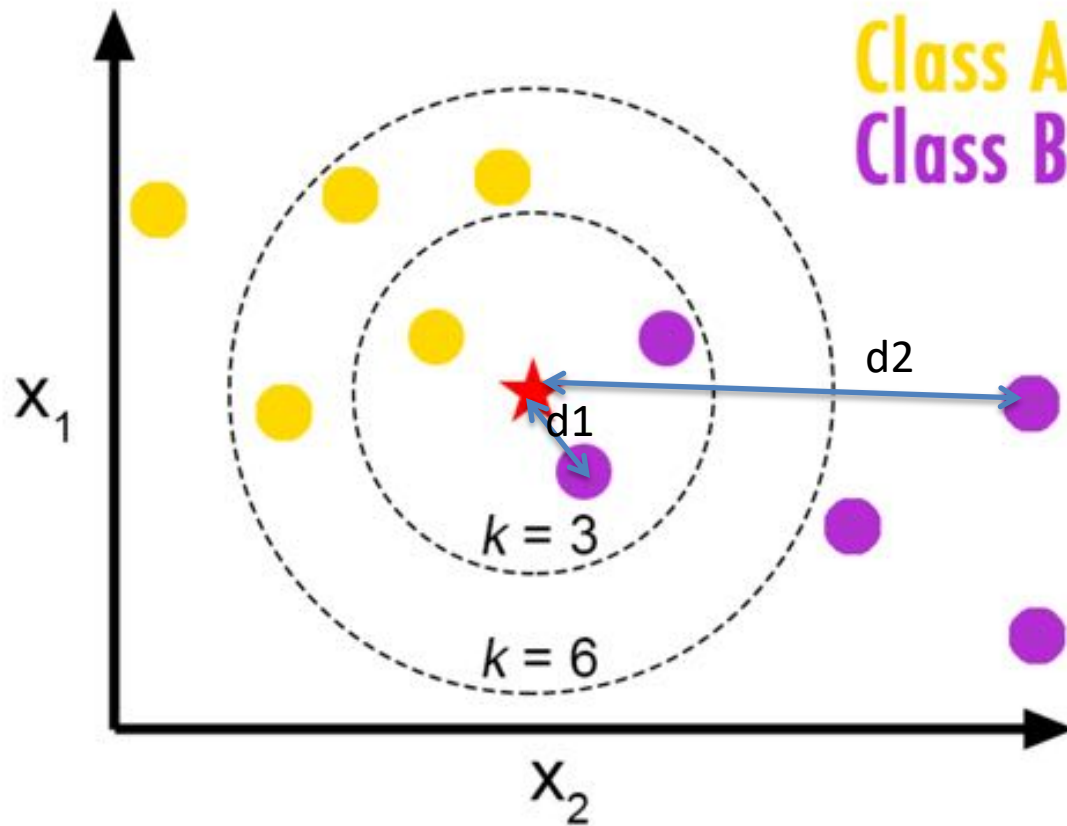
$$\text{Logit}(p) = \ln(p/(1-p)) = y^* = b_0 + b_1x_1 + b_2x_2 + b_3x_3$$

Regresja logistyczna – binarna decyzja

```
#regresja logistyczna w pakiecie nnet  
>library(nnet)  
classifier.nnet <-  
nnet::multinom(Species~., data = iris)  
>predictions <-  
as.character(predict(classifier.nnet,  
iris), type="response")  
>table(iris$Species, predictions)
```



kNN (k Nearest Neighbours)



- Funkcja odległości:
- $d(x_1, x_2)$

kNN (k Nearest Neighbours)

```
#kNN (działa tylko na danych  
numerycznych)
```

```
>library(class)
```

```
>classifier.knn <-
```

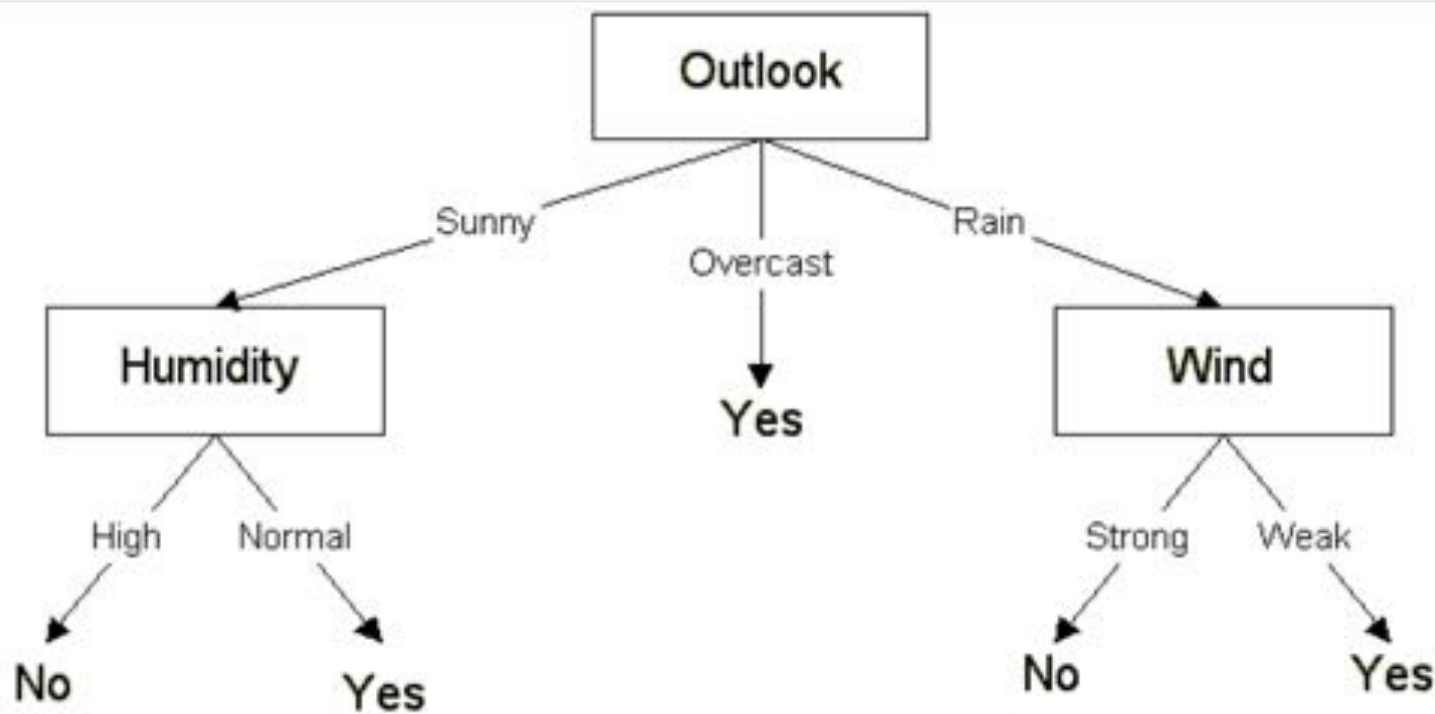
```
class::knn(train=iris[,-ncol(iris)],  
test=iris[,-ncol(iris)], cl=iris$Species,  
k = 3, prob=TRUE)
```

```
>predictions <-
```

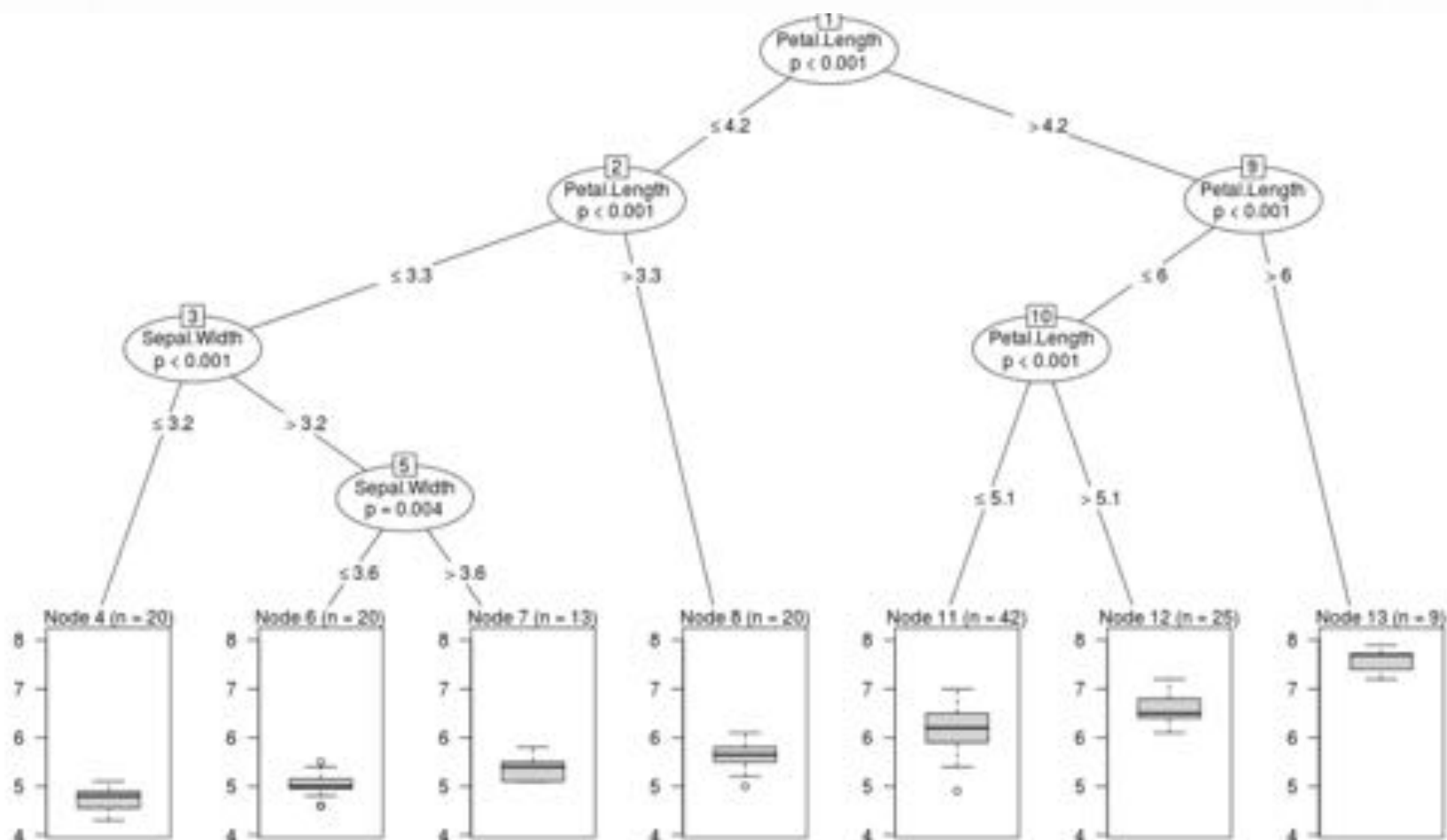
```
as.character(as.vector(classifier.knn))  
table(iris$Species, predictions)
```



Drzewo decyzyjne ID3, CART, C4.5



Drzewo regresyjne

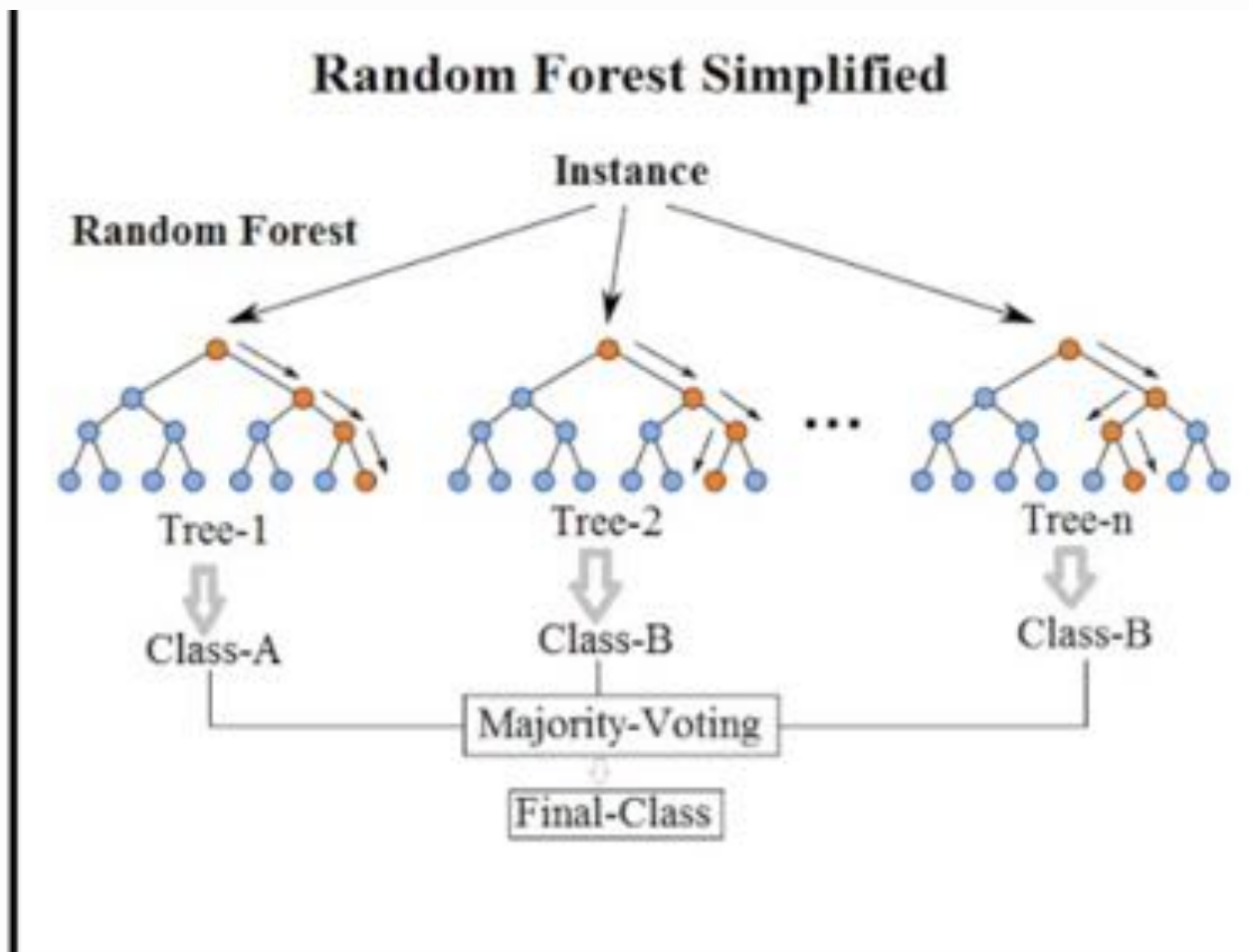


Drzewo decyzyjne ID3, CART, C4.5

```
### drzewo decyzyjne rpart  
>library(rpart)  
>classifier.rpart <-  
rpart::rpart(Species~., data = iris)  
>predictions <-  
as.character(predict(classifier.rpart, iris,  
type="class"))  
>table(iris$Species, predictions)
```



Las losowy (Random Forest)



Las losowy (Random Forest)

```
### las losowy  
>library(randomForest)  
>classifier.rf <-  
randomForest::randomForest(Species~.,  
data = iris)  
>predictions <-  
as.character(predict(classifier.rf,iris,  
type="class"))  
>table(iris$Species, predictions)
```



Naiwny Bayes (Naive Bayes)

- Wzór Bayesa

$$P(A | B) = \frac{P(B | A) P(A)}{P(B)}$$

- Pstwo klasy decyzyjnej

$$P(d_i | v_1, \dots, v_n) = \frac{P(v_1, \dots, v_n | d_i) P(d_i)}{P(v_1, \dots, v_n)}$$

- 'Naiwnie' zakładamy niezależność zmiennych

$$P(v_1, \dots, v_n | d_i) = P(v_1 | d_i) * \dots * P(v_n | d_i)$$

- I dochodzimy do finalnego wzoru

$$P(d_i | v_1, \dots, v_n) = C * P(v_1 | d_i) * \dots * P(v_n | d_i) * P(d_i)$$

Naiwny Bayes (Naive Bayes)

$$P(d_i | v_1, \dots, v_n) = C * P(v_1 | d_i) * \dots * P(v_n | d_i) * P(d_i)$$

- $P(d=no) = 5/14$
- $P(d=yes) = 9/14$
- $P(\text{outlook}=sunny|no)=3/5$
- $P(\text{outlook}=sunny|yes)=2/9$
- $P(\text{outlook}=rainy|no)=2/5$
- $P(\text{outlook}=rainy|yes)=3/9$
- $P(\text{windy}=true|no)=3/5$
- $P(\text{windy}=true|yes)=3/9$
- itd.

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

Naiwny Bayes (Naive Bayes)

#działa tylko na danych nominalnych

```
>library(e1071)
```

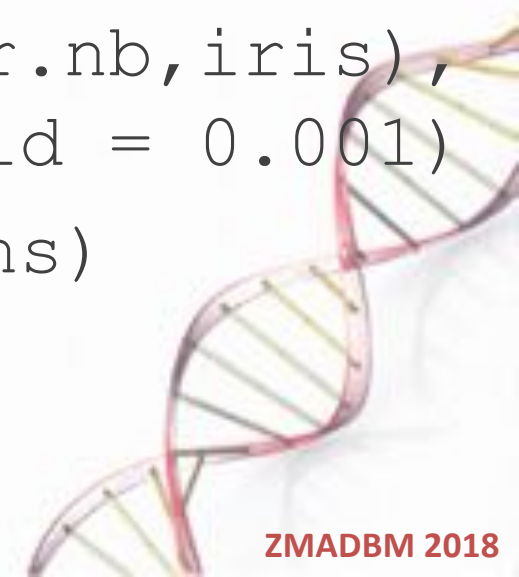
```
>classifier.nb <-
```

```
e1071::naiveBayes(Species~., data = iris,  
laplace = 0, na.action = na.pass)
```

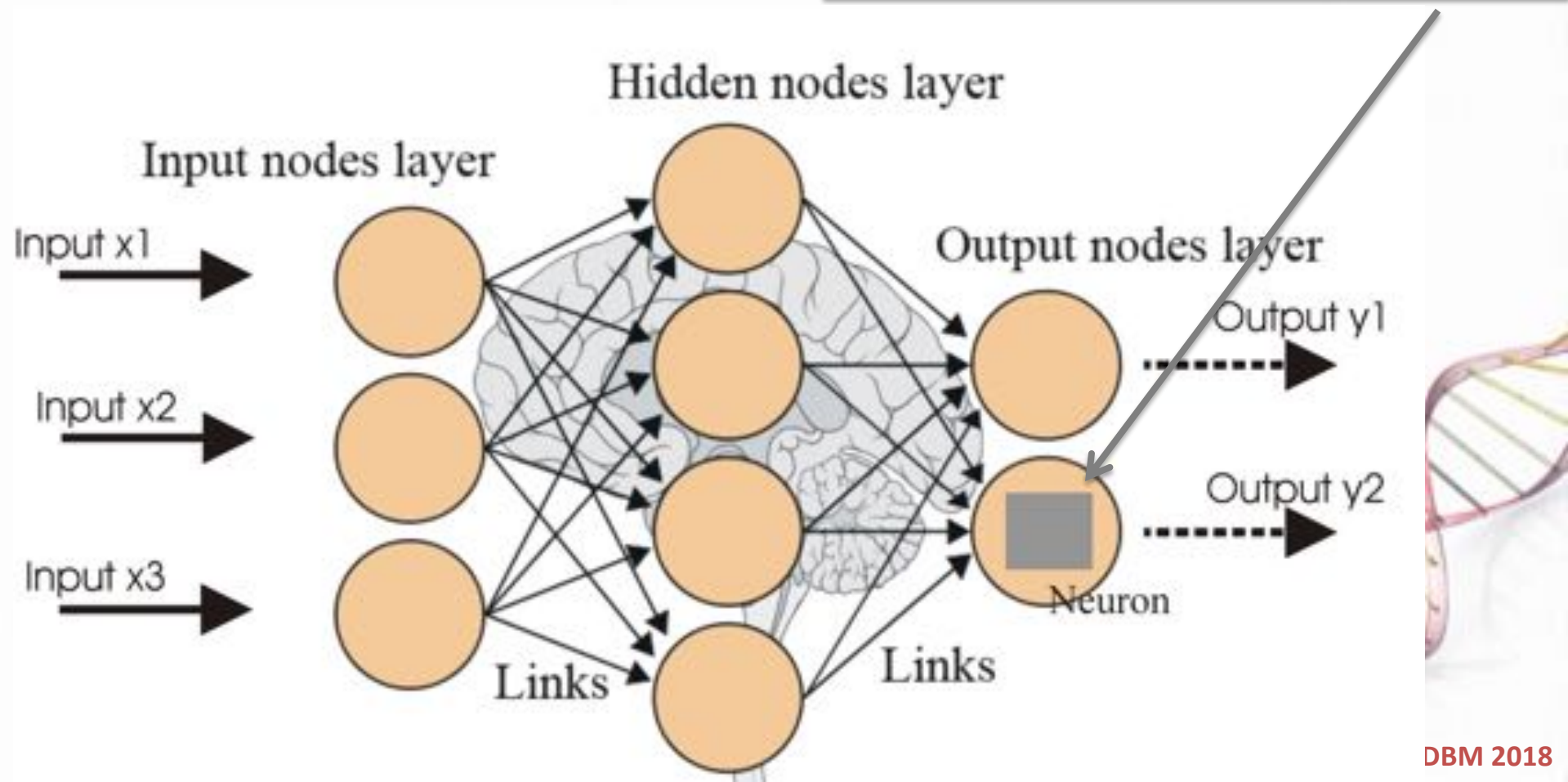
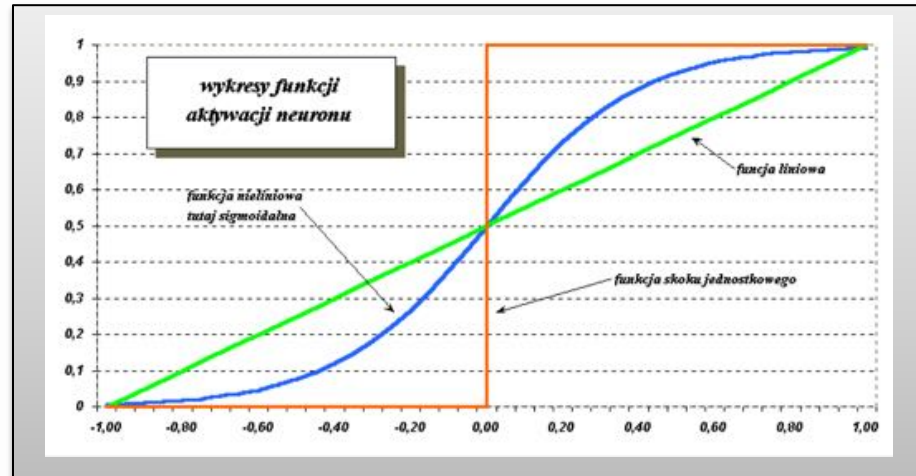
```
>predictions <-
```

```
as.character(predict(classifier.nb, iris),  
type=c("class", "raw"), threshold = 0.001)
```

```
>table(iris$Species, predictions)
```



Sieć neuronowa



Sieć neuronowa

```
#neural net
>library("nnet")
>n <- names(iris)
>f <- as.formula(paste("Species ~",
  paste(n[!n %in% "Species"], collapse = " + "))
>classifier.nn <- nnet(f,data=iris,size=10,linout=T)
predictions <-
as.character(predict(classifier.nn,iris,type="class"))
>table(iris$Species, predictions)
#show nnet
>library(devtools)
>source_url('https://gist.githubusercontent.com/fawda123/
7471137/raw/466c1474d0a505ff044412703516c34f1a4684a5/nnet
_plot_update.r')
>plot.nnet(classifier.nn)
```

Indukcja reguł decyzyjnych (AQ, CN2, LEM, Ripper)

Warunek -> **Teza** (gdzie warunek to koniunkcja testów na cechach)

Przykład:

IF Outlook=overcast THEN play=YES

IF Outlook=rain AND Wind=strong THEN play=NO

IF Outlook=rain AND Wind=weak THEN play=YES

IF Outlook=sunny AND Humidity=high THEN play=NO

IF Outlook=sunny AND Humidity=normal THEN play=YES

Pokrycie: $\text{support}(w \rightarrow t) = n_{wt} / n_t$ (jeśli = 1 tzn pokrywa **wszystkie** przykłady pozytywne)

Trafność: $\text{accuracy}(w \rightarrow t) = n_{wt} / n_w$ (jeśli = 1 tzn pokrywa **tylko** przykłady pozytywne)

Indukcja reguł decyzyjnych (AQ, CN2, LEM, Ripper)

	Outlook	Temp.	Humid.	Wind	Sport?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cold	Normal	Weak	Yes
6	Rain	Cold	Normal	Strong	No
7	Overcast	Cold	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cold	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

$\text{Temp}=\text{Mild} \wedge \text{Humid}=\text{Normal} \Rightarrow \text{Sport}=\text{Yes}$

Wsparcie reguły: 0,22

(2 obiekty pasujące, 9 obiektów w klasie Sport=Yes)

Dokładność reguły: 1

$\text{Wind}=\text{Weak} \Rightarrow \text{Sport}=\text{Yes}$

Wsparcie reguły: 0,66

(6 obiektów pasujących do obu stron, 9 obiektów w klasie Sport=Yes)

Dokładność reguły: 0,75

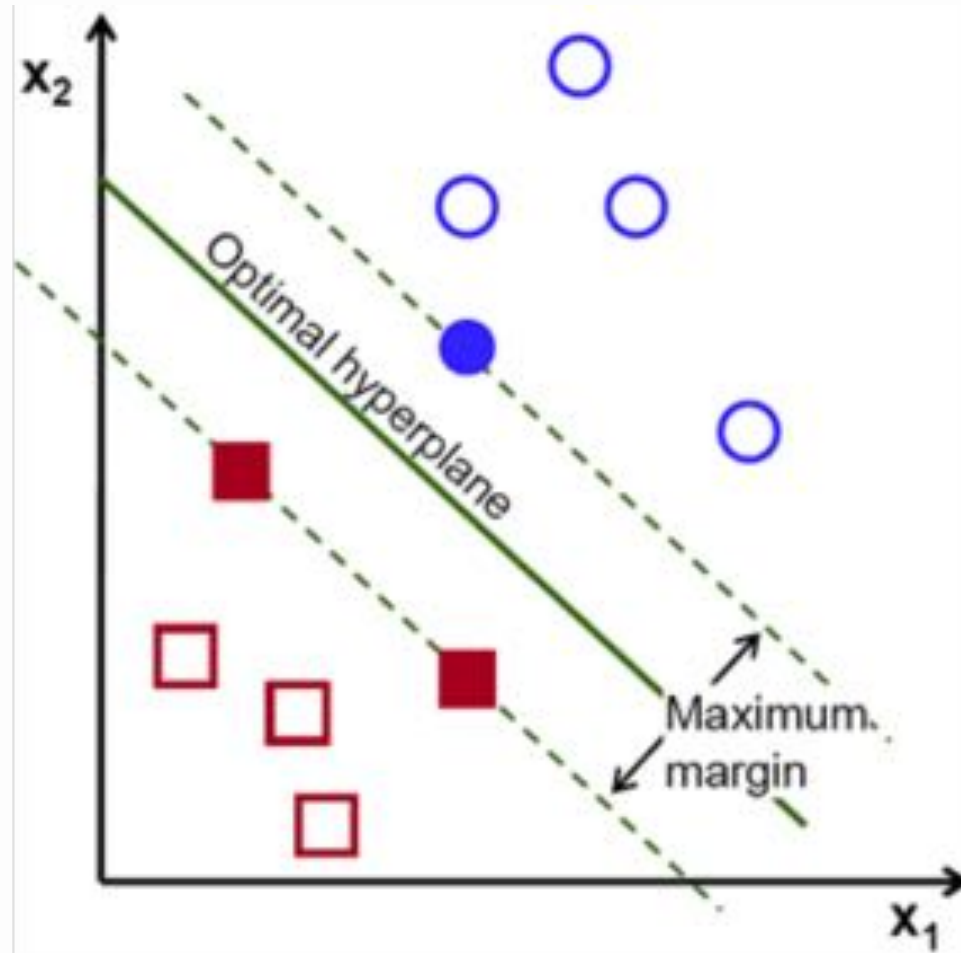
(6 obiektów pasujących do obu stron, 8 obiektów mających Wind=Weak)

Indukcja reguł decyzyjnych (AQ, CN2, LEM, Ripper)

```
# rule induction
library("RoughSets")
>rule.data <- SF.asDecisionTable(iris, decision.attr = ncol(iris))
>cut.values <- D.discretization.RST(rule.data, type.method =
"local.discernibility")
rule.data <- SF.applyDecTable(rule.data, cut.values)
#Algorytm AQ
>rules <- RI.AQRules.RST(rule.data, confidence = 0.9, timesCovered
= 3)
>predictions <- predict(rules,rule.data)
>table(iris$Species, predictions$predictions)
#Algorytm LEM2
>rules <- RI.LEM2Rules.RST(rule.data)
>predictions <- predict(rules,rule.data)
>table(iris$Species, predictions$predictions)
#Algorytm CN2
>rules <- RI.CN2Rules.RST(rule.data, K = 5)
>predictions <- predict(rules,rule.data)
>table(iris$Species, predictions$predictions)
```

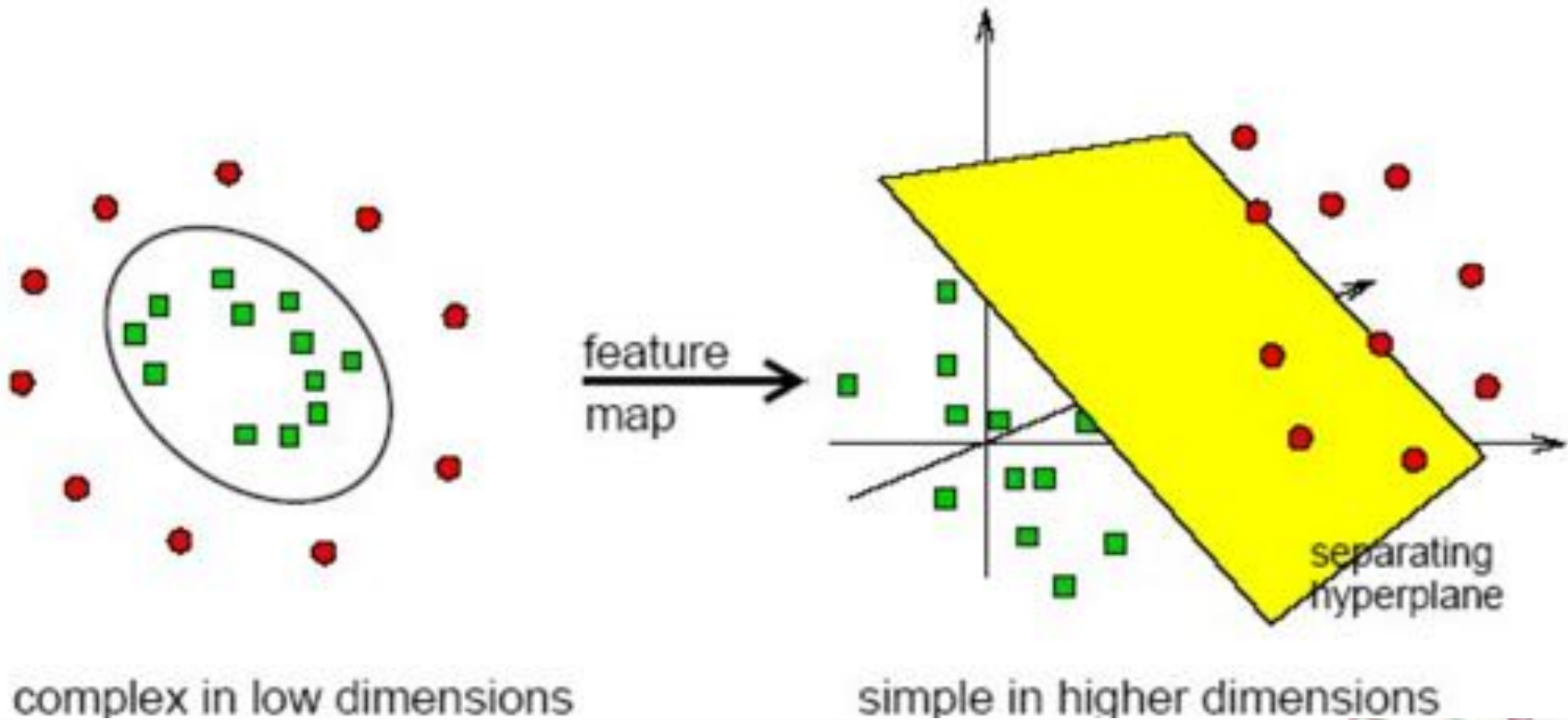


Support Vector Machines (SVM)



Support Vector Machines (SVM)

Separation may be easier in higher dimensions



Support Vector Machines (SVM)

```
### Support Vector Machines (SVM)
```

```
>library(e1071)
```

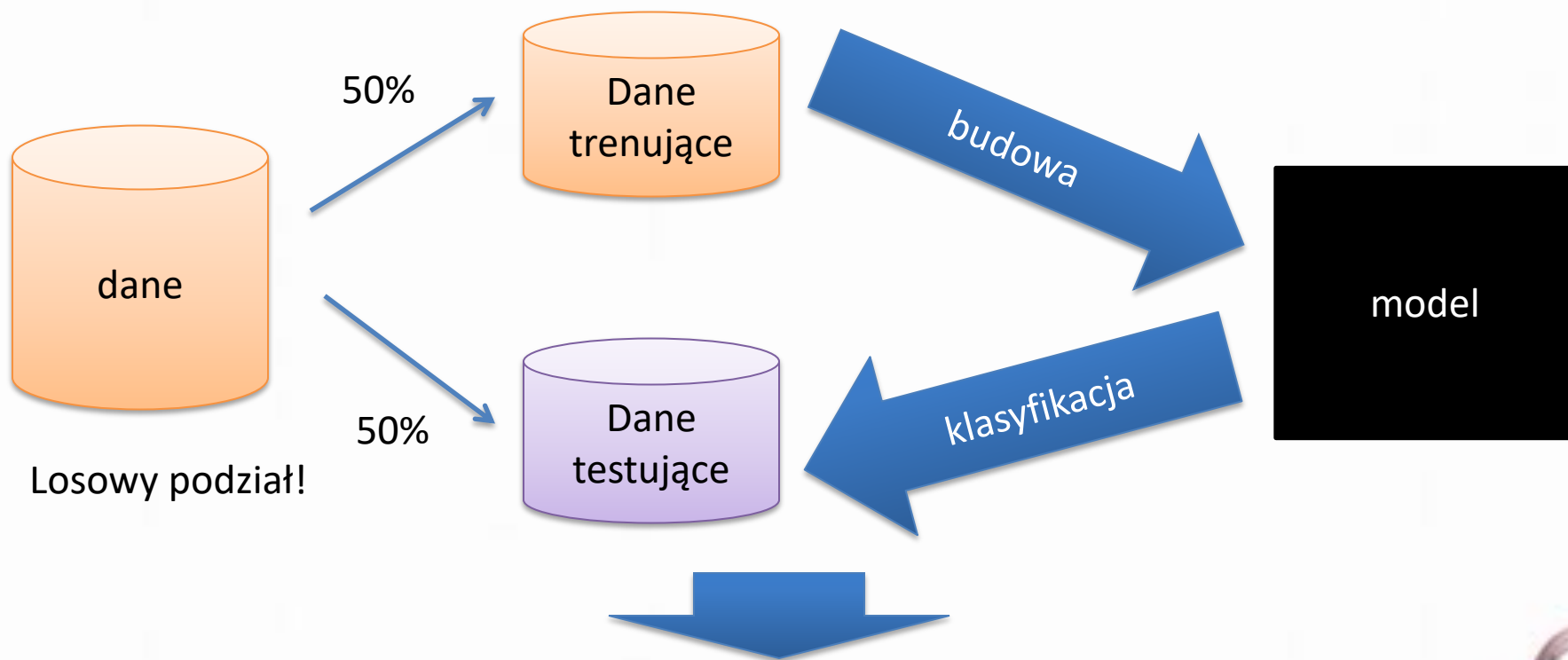
```
>classifier.svm <- e1071::svm(Species~., data = iris,  
cost = 1000, gamma = 0.0001)
```

```
>predictions <-  
as.character(predict(classifier.svm,iris),type=c("class",  
"raw"),threshold = 0.001)
```

```
>table(iris$Species, predictions)
```



Schemat trenuj testuj



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

Macierz błędów (confusion matrix)

Predictive Model: Evaluation

Accuracy = $\frac{tp + tn}{tp + tn + fp + fn}$

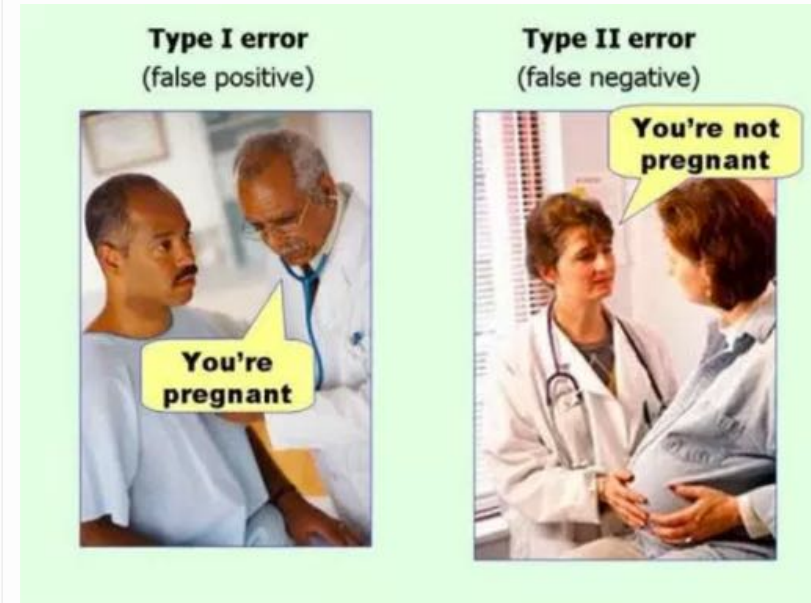
		actual result / classification	
		yes	no
predictive result / classification	yes	tp (true positive)	fp (false positive) ← Type 1 error
	no	fn (false negative)	tn (true negative)

Precision = $\frac{tp}{tp + fp}$

Recall = $\frac{tp}{tp + fn}$

$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$

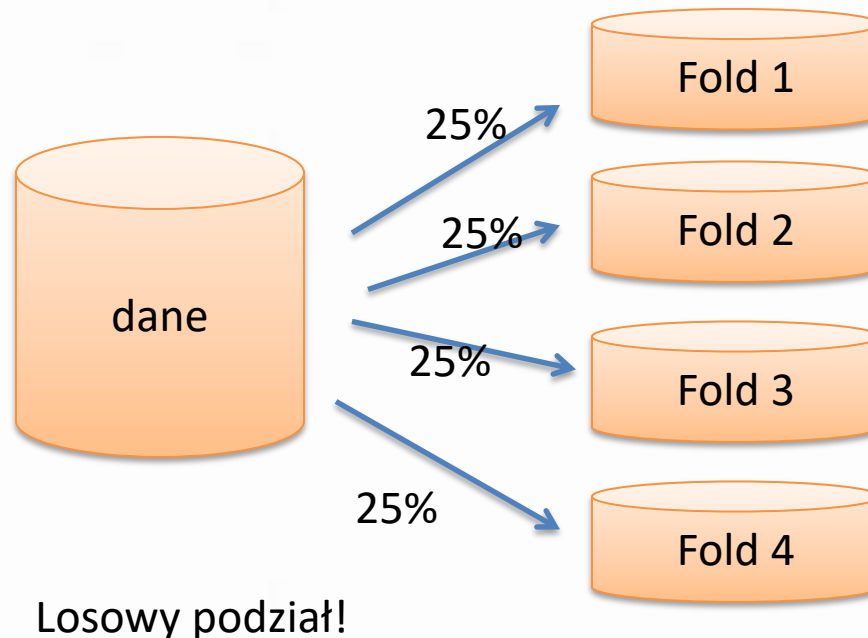
True Negative Rate = $\frac{tn}{tn + fp}$



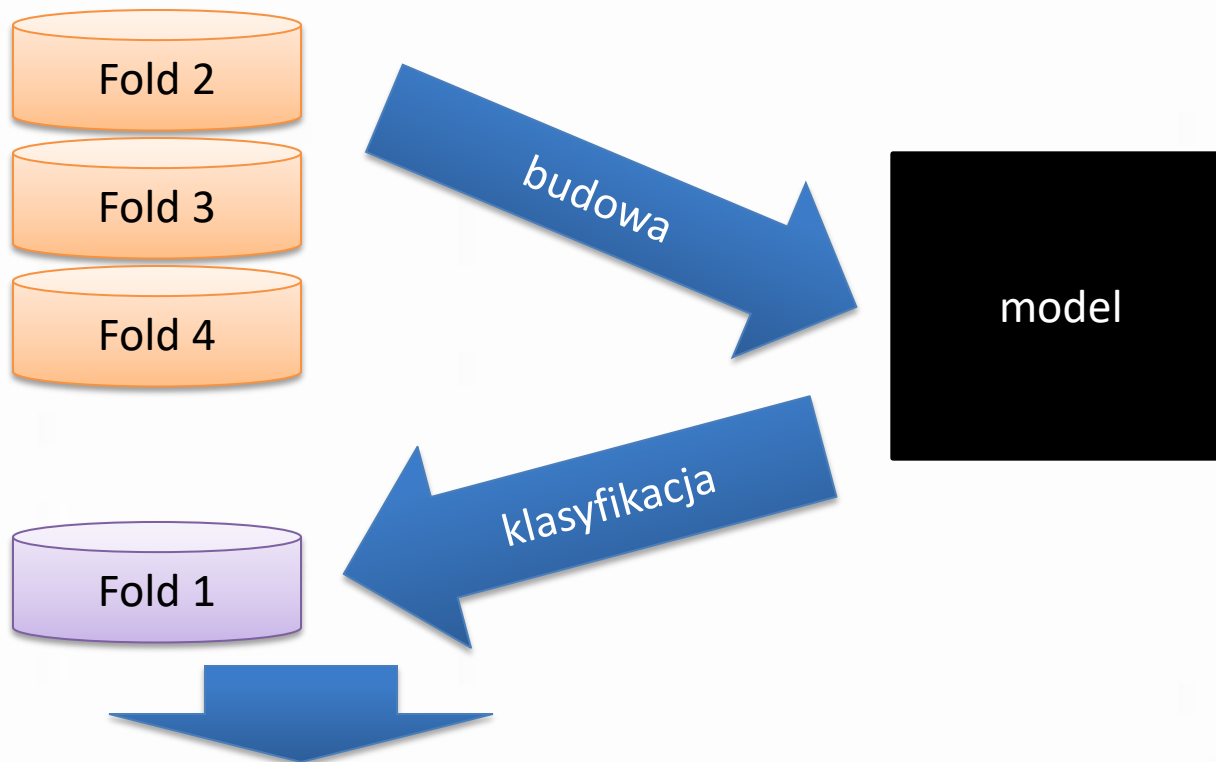
- **Precision** (positive predictive value PPV) = $tp / (tp + fp)$
- Sensitivity, **Recall**, hit rate, lub true positive rate (TPR) = $tp / (tp + fn)$
- Specificity lub **True Negative Rate** (TNR) = $tn / (tn + fp)$

Walidacja krzyżowa (cross validation)

- 4 fold -> dzielimy na równe 4 podzbiory
- Tyle iteracji ile podzbiorów
- W każdej kolejny zbiór jest traktowany jako testujący a pozostałe jako trenujące
- Dzięki temu finalnie wszystkie przykłady są poddane klasyfikacji



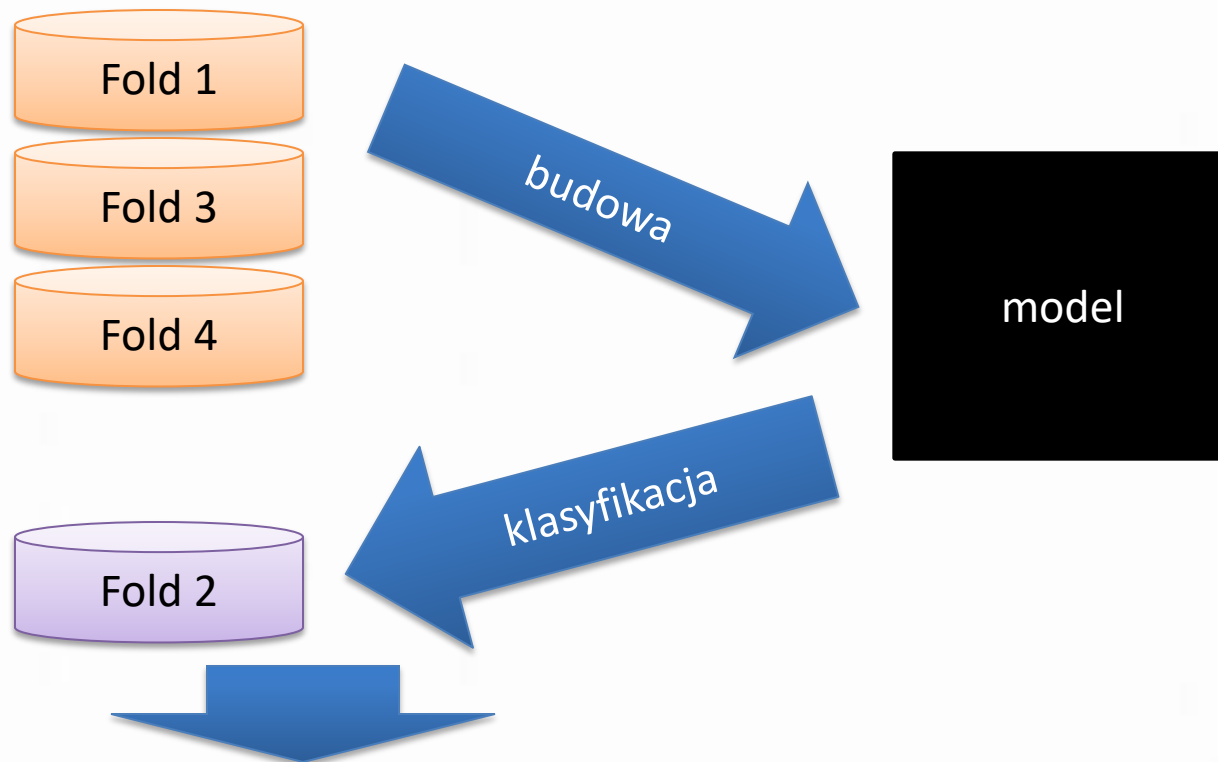
Walidacja krzyżowa (cross validation)



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

Podzbiór 1 jako testujący!

Walidacja krzyżowa (cross validation)



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

**Podzbiór 2 jako testujący!
itd.**



Ćwiczenia 5

michal.draminski@ipipan.waw.pl

Zdjęcia, schematy i rysunki zostały zaczerpnięte z internetu.

Ćwiczenia 5

Z1. Napisz funkcję która implementuje schemat train/test dla dwóch dowolnie wybranych klasyfikatorów. Funkcja na wejściu przyjmuje parametry:

- rep - liczba powtórzeń Train/test (domyślnie 3),
- zbiór danych (data.frame),
- nazwę kolumny decyzyjnej.
- Typ/nazwę klasyfikatora

Funkcja zwraca listę której elementami są:

- macierz błędów,
- jakość klasyfikacji (accuracy),
- Wartość F

Uruchom funkcję na zbiorze students dla dyskretnej wartości G1 (0,10] i (10,20]. Usuń skorelowane z nią G2 i G3. Porównaj wyniki dla obu algorytmów i zapisz swoje wnioski/przemyślenia **(10 pkt)**.