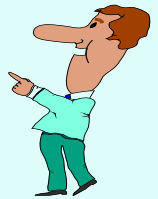
Metody oceny wiedzy klasyfikacyjnej odkrytej z danych – ocena zdolności predykcyjnej klasyfikatorów

Jerzy Stefanowski
Instytut Informatyki
Politechnika Poznańska

Wykład uczenie maszynowe
Poznań In PP – aktualizacja 2019

Ocena wiedzy klasyfikacyjnej - plan

1. **Metody oceny wiedzy o klasyfikacji obiektów – przegląd miar**
2. **Eksperymentalna ocena klasyfikatorów**
3. **Porównanie wielu klasyfikatorów w studiach przypadków − wykorzystanie testów statystycznych**



Uwagi do źródeł

Wykorzystano książki:

- S.Weiss, C.Kulikowski: Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems, Morgan Kaufmann 1991.
- N.Japkowicz, M. Shah: Evaluating Learning Algorithms: A Classification Perspective, Cambridge Presss 2011.
- I.Konennko, M.Kukar: Machine Learning and Data Mining, 2007.
- J.Han, M.Kember: Data mining. Morgan Kaufmann 2001.

oraz inspiracje ze slajdów wykładów:

- J.Han; G.Piatetsky-Shapiro; D.Page, A.Avati + materiały związane z WEKA i prezentacji W.Kotłowski nt. Statistical Analysis of Computational Experiments in Machine Learning

Wybrane artykuły

- Patrz następny slajd

Wybrane artykuły

1. Kohavi, R. (1995): A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proc. of the 14th Int. Joint Conference on Artificial Intelligence*, 1137—1143.
2. Salzberg, S. L. (1997): On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1, 317—328.
3. Dietterich, T. (1998): Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:7, 1895-–1924.
4. Bouckaert, R. R. (2003): Choosing between two learning algorithms based on calibrated tests. *ICML 2003*.
5. Bengio, Y., Grandvalet, Y. (2004): No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning Research*, 5, 1089—1105.
6. Demsar, J. (2006): Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7, 1–30.
7. S. Raschka (2018) Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning, arXiv 2018
8. Sesja spacjalna nt. oceny systemów uczących (N.Japkowicz) na ICML 2007 + tutorial

Oraz nowsze artykuły, np. o testach statystycznych Salvador Garcia Univ. Granada

Poszukiwanie i ocena wiedzy klasyfikacyjnej

- **Perspektywy odkrywania wiedzy**
  - **Predykcja klasy (-fikacji)–** przewidywanie przydziału nowych obiektów do klas / reprezentacja wiedzy wykorzystywana jako tzw. **klasyfikator** (ocena zdolności klasyfikacyjnej – na ogół jedno wybrane kryterium).
  - **Opis klasyfikacji obiektów –** wyszukiwanie wzorców charakteryzujących właściwości danych i prezentacja ich użytkownikowi w zrozumiałej formie (ocena trudniejsza i bardziej subiektywna)**.**

Spójrz też do książki : J.Stefanowski Algorytmy indukcji regul decyzyjnych w odkrywaniu wiedzy 2001. pdf dostępny na mojej stronie WWW

Kryteria oceny metod klasyfikacyjnych

- **Trafność predykcji** (klasyfikacja / regresja)
- Zdolności interpretacji modelu: np. drzewa decyzyjne vs. sieci neuronowe => patrz dalsze wykłady
- Złożoność struktury, np.
  - rozmiar drzew decyzyjnego,
  - miary oceny reguły
- Odporność (Robustness)
  - Szum (noise),
  - Inne trudności rozkładu danych,

------- oraz

- Szybkość i skalowalność:
  - czas uczenia się,
  - szybkość samego klasyfikowania

Dlaczego miary oceny klasyfikatorów?

Obserwacja wyłącznie procesu uczenia (training data) – „proxy” – funkcja zastępcza wobec rzeczywistego świata

Prowadzą do skupienia działania wokół precyzyjnego celu i wspierają decyzje co do zastosowania

Pozwalają na porównanie (obecne działanie vs. tzw. baseline; aktualne działania vs. oczekiwane – optymalizacja; porównywanie wielu alternatywnych rozwiązań, …)

Wspierają tzw. Monitoring lub debugging systemu

Oraz ….

Tworzenie i ocena klasyfikatorów

Jest procesem trzyetapowym:

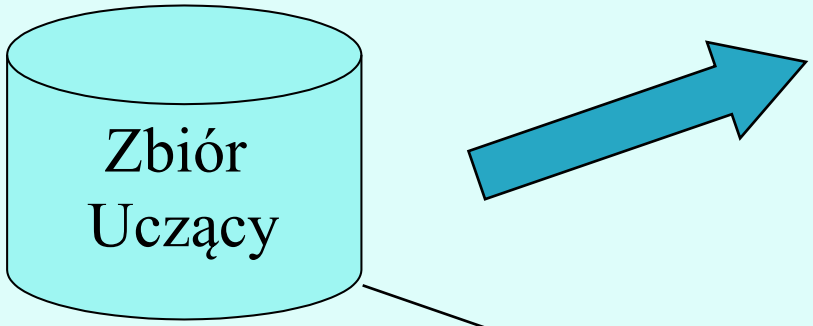
1. Konstrukcja modelu w oparciu o zbiór danych wejściowych (przykłady uczące).

   Przykładowe modele - klasyfikatory:

   - drzewa decyzyjne,
   - reguły (IF .. THEN ..),
   - sieci neuronowe, SVM, .

2. Ocena modelu (przykłady testujące)
3. Użycie modelu (klasyfikowanie nowych obiektów)

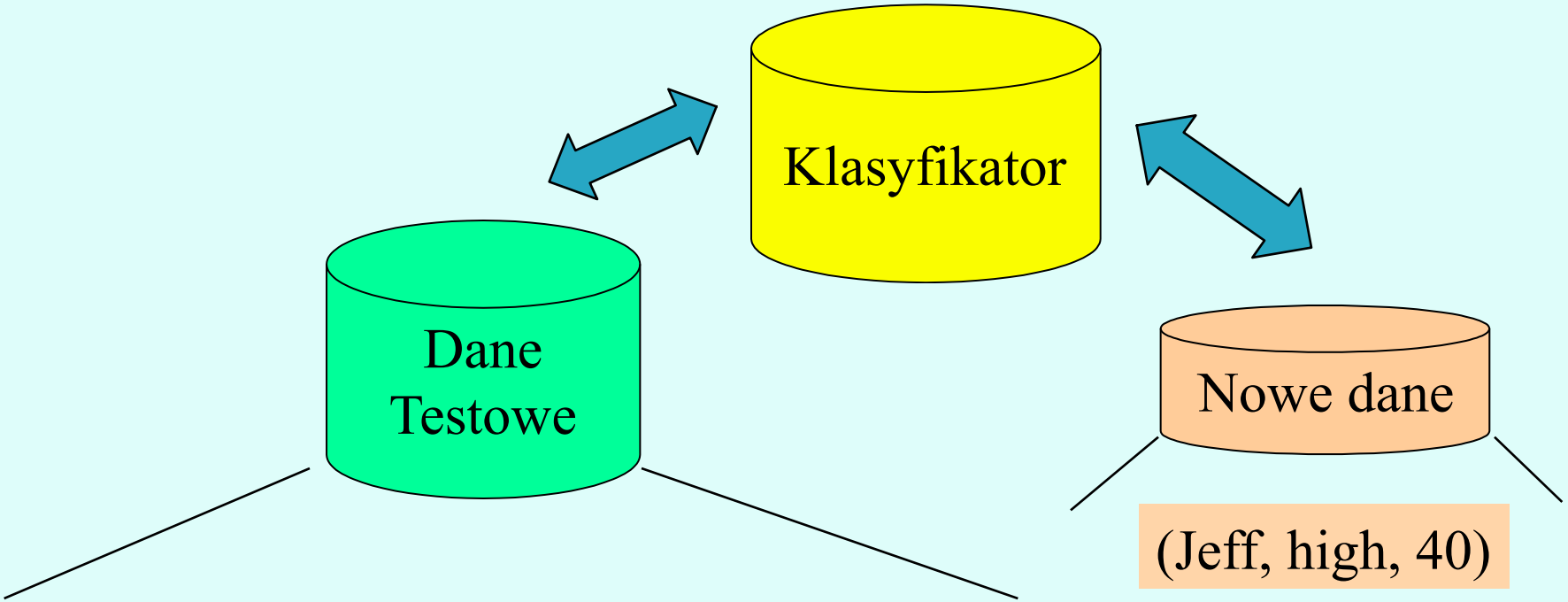
Uczenie klasyfikatora



NAME	INCOME	AGE	Rating
Mike	low	30	no
Mary	high	36	yes
Bill	high	45	yes
Jim	medium	50	yes
Dave	medium	27	no
Anne	low	28	no

IF income = ‘high’
OR age > 35
THEN rating = ‘yes’

Proces klasyfikowania obiektów



NAME	INCOME	AGE	RATING
Tom	low	24	no
Merlisa	medium	33	no
George	high	33	yes
Joseph	low	40	yes

Ocena trafności /błędu klasyfikowanie

- Typowo funkcja straty (statistical learning point of view):
- Ocena predykcji– prediction risk:

$$R(f) = E_{xy} L(y, f(x))$$

- gdzie $L(y, \hat{y})$ to " loss or cost function" dla predykcji wartości $\hat{y}$ gdy prawdziwa była y = a E is expected value over the joint distribution of all (x,y) for data to be predicted.
- Binarna klasyfikacja → zero-one loss function

$$L(y, \hat{y}) = \begin{cases} 0 & if\ y = f(y) \\ 1 & if\ y \neq f(y) \end{cases}$$

Trafność klasyfikowania

- Użyj przykładów testowych nie wykorzystanych w fazie uczenia klasyfikatora:
  - N_t – liczba przykładów testowych
  - N_c – liczba poprawnie sklasyfikowanych przykładów testowych
- Trafność klasyfikowania (ang. classification accuracy):

$$\eta = \frac{N_c}{N_t}$$

- Alternatywnie błąd klasyfikowania.

$$\varepsilon = \frac{N_t - N_c}{N_t}$$

Inne możliwości analizy:

- macierz pomyłek (ang. confusion matrix),
- koszty pomyłek i klasyfikacja binarna,
- miary Sensitivity i Specificity / krzywa ROC

Predykcja zmiennej y (liczbowej)

- Zmienna wyjściowa liczbowa : ocena jak odbiega predykcja y^ od właściwej wyjściowej y
- Loss function**: measures the error betw. y_i and the predicted value y_i^**
  - **Absolute error: $| y_i - y^_i|$**
  - **Squared error: $(y_i - y^_i)^2$**
- **Test error (generalization error): the average loss over the test set**
  - **Mean absolute error:** $\frac{\sum_{i=1}^{n} | y_i - \hat{y}_i |}{n}$ **Mean squared error:** $\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}$
  - **Relative absolute error:** $\frac{\sum_{i=1}^{n} | y_i - \hat{y}_i |}{\sum_{i=1}^{n} | y_i - \bar{y} |}$ **Relative squared error:** $\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$

  - Na ogół stosowane (square) root mean-square error, similarly, root relative squared error

Wiele innych miar oceny predykcji

Some of these are restricted to the binary classification case:

`precision_recall_curve` (y_true, probas_pred)	Compute precision-recall pairs for different probability thresholds
`roc_curve` (y_true, y_score[, pos_label, ...])	Compute Receiver operating characteristic (ROC)
`balanced_accuracy_score` (y_true, y_pred[, ...])	Compute the balanced accuracy

Others also work in the multiclass case:

`cohen_kappa_score` (y1, y2[, labels, weights, ...])	Cohen's kappa: a statistic that measures inter-annotator agreement.
`confusion_matrix` (y_true, y_pred[, labels, ...])	Compute confusion matrix to evaluate the accuracy of a classification
`hinge_loss` (y_true, pred_decision[, labels, ...])	Average hinge loss (non-regularized)
`matthews_corrcoef` (y_true, y_pred[, ...])	Compute the Matthews correlation coefficient (MCC)

Some also work in the multilabel case:

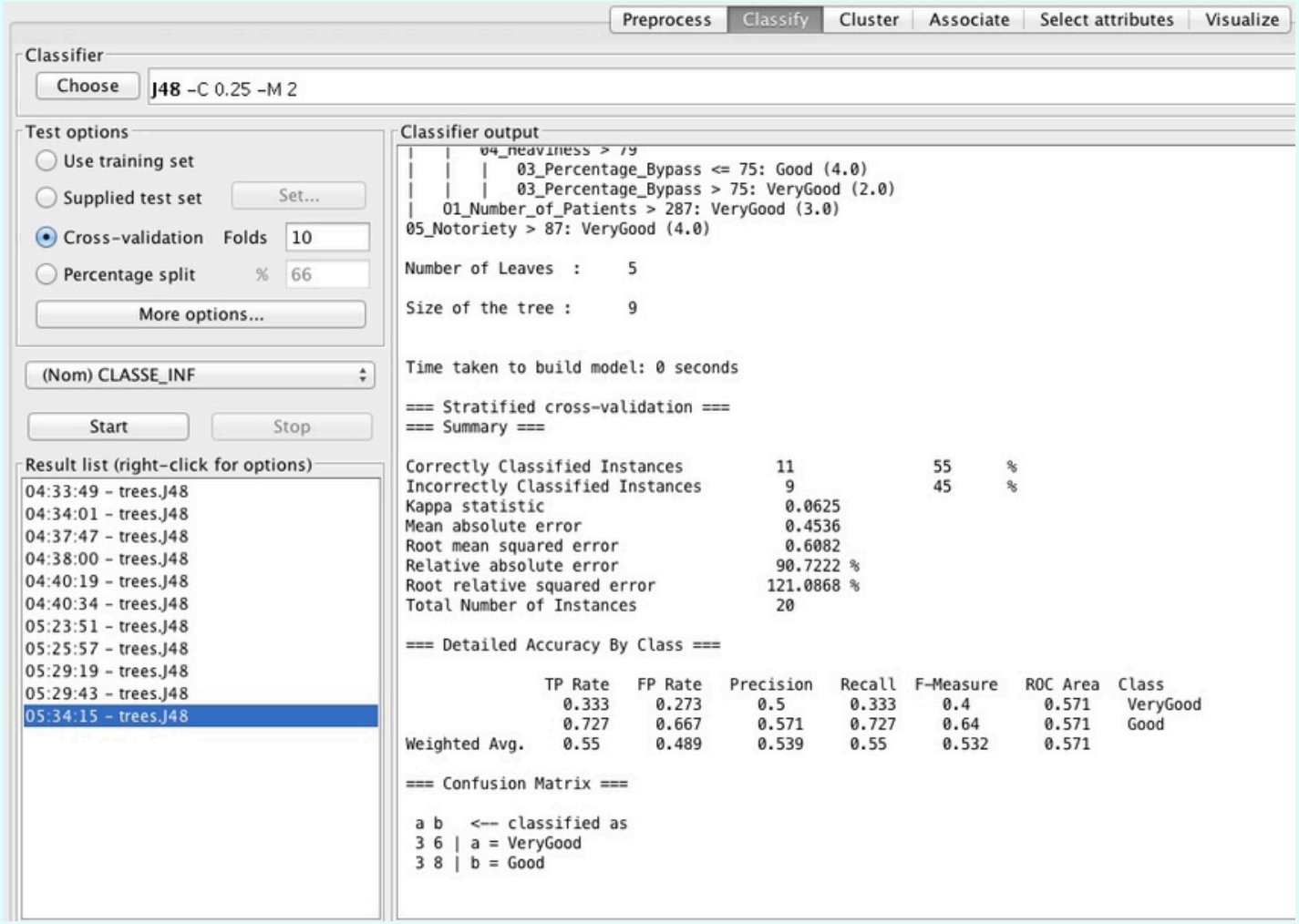
`accuracy_score` (y_true, y_pred[, normalize, ...])	Accuracy classification score.
`classification_report` (y_true, y_pred[, ...])	Build a text report showing the main classification metrics
`f1_score` (y_true, y_pred[, labels, ...])	Compute the F1 score, also known as balanced F-score or F-measure
`fbeta_score` (y_true, y_pred, beta[, labels, ...])	Compute the F-beta score
`hamming_loss` (y_true, y_pred[, labels, ...])	Compute the average Hamming loss.
`jaccard_similarity_score` (y_true, y_pred[, ...])	Jaccard similarity coefficient score
`log_loss` (y_true, y_pred[, eps, normalize, ...])	Log loss, aka logistic loss or cross-entropy loss.
`precision_recall_fscore_support` (y_true, y_pred)	Compute precision, recall, F-measure and support for each class
`precision_score` (y_true, y_pred[, labels, ...])	Compute the precision
`recall_score` (y_true, y_pred[, labels, ...])	Compute the recall
`zero_one_loss` (y_true, y_pred[, normalize, ...])	Zero-one classification loss.

And some work with binary and multilabel (but not multiclass) problems:

`average_precision_score` (y_true, y_score[, ...])	Compute average precision (AP) from prediction scores
`roc_auc_score` (y_true, y_score[, average, ...])	Compute Area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores.

Za dokumentacją scikit learn

WEKA evaluation



Miary – zależność od zadania

Klasyfikacja binarna

- Wskazanie etykiety vs. scoring predictions
  - Miary punktowe np. Accuracy, Precision, Recall / Sensitivity, Specificity, F-score, G-mean
  - Miary przeglądowe-graficzne: ROC, PR

Wieloklasowość / Wielo-etykietowość

- Nie wszystkie miary binarne można uogólnić

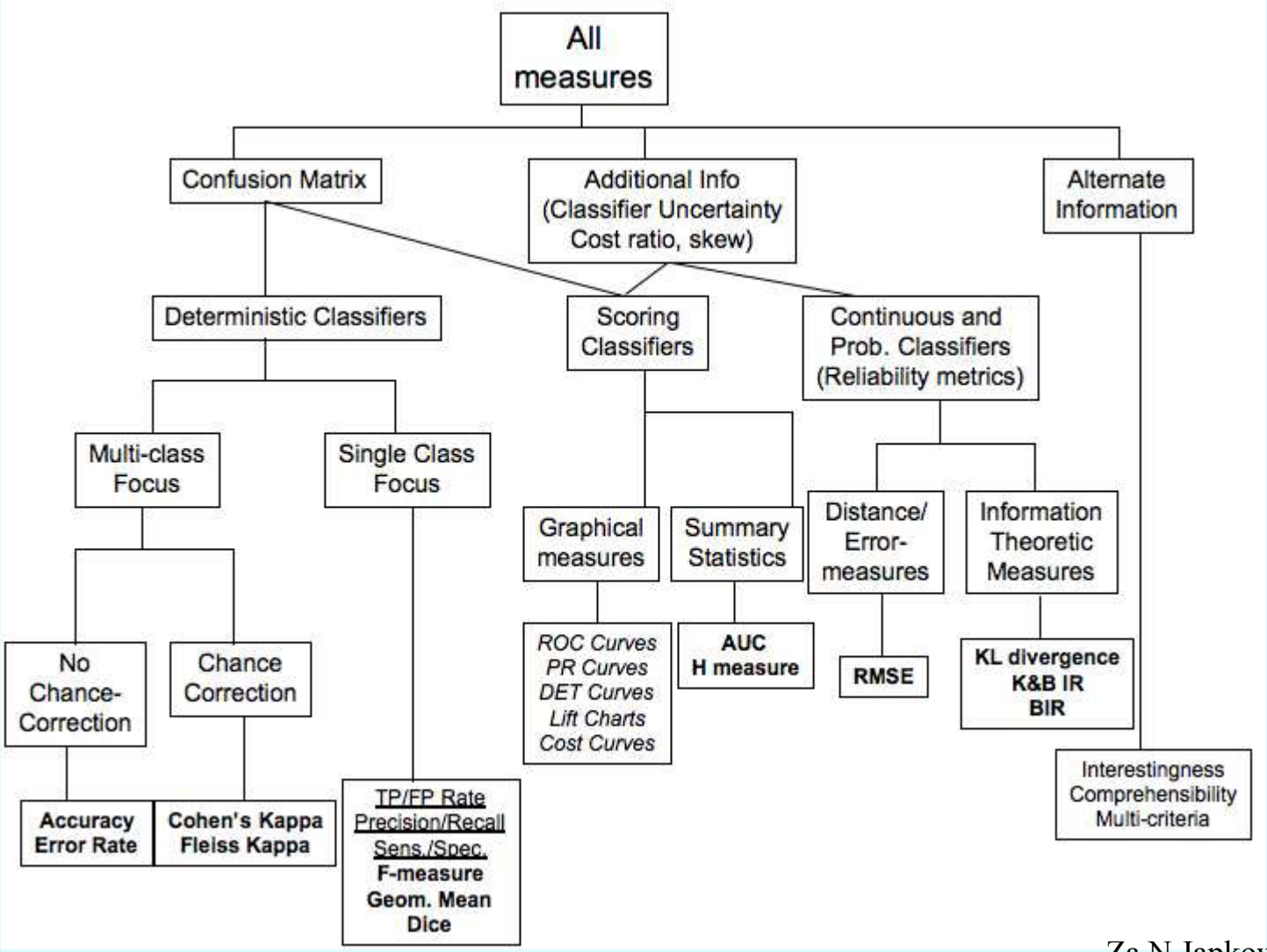
Specyfika danych

- Imbalanced data oraz cost sensitive learning

Predykcja ciągła

- Błędy RSME

Kategoryzacja różnym miar oceny



Macierz pomyłek

- Analiza pomyłek w przydziale do różnych klas przy pomocy tzw. macierz pomyłek (ang. *confusion matrix*)
- Macierz $r \times r$, gdzie wiersze odpowiadają poprawnym klasom decyzyjnym, a kolumny decyzjom przewidywanym przez klasyfikator; na przecięciu wiersza i oraz kolumny j - liczba przykładów $n\text{-}ij$ należących oryginalnie do klasy i-tej, a zaliczonej do klasy j-tej

Przykład:

	Przewidywane klasy decyzyjne		
Oryginalne klasy	K_1	K_2	K_3
K_1	50	0	0
K_2	0	48	2
K_3	0	4	46

Cost sensitive matrix

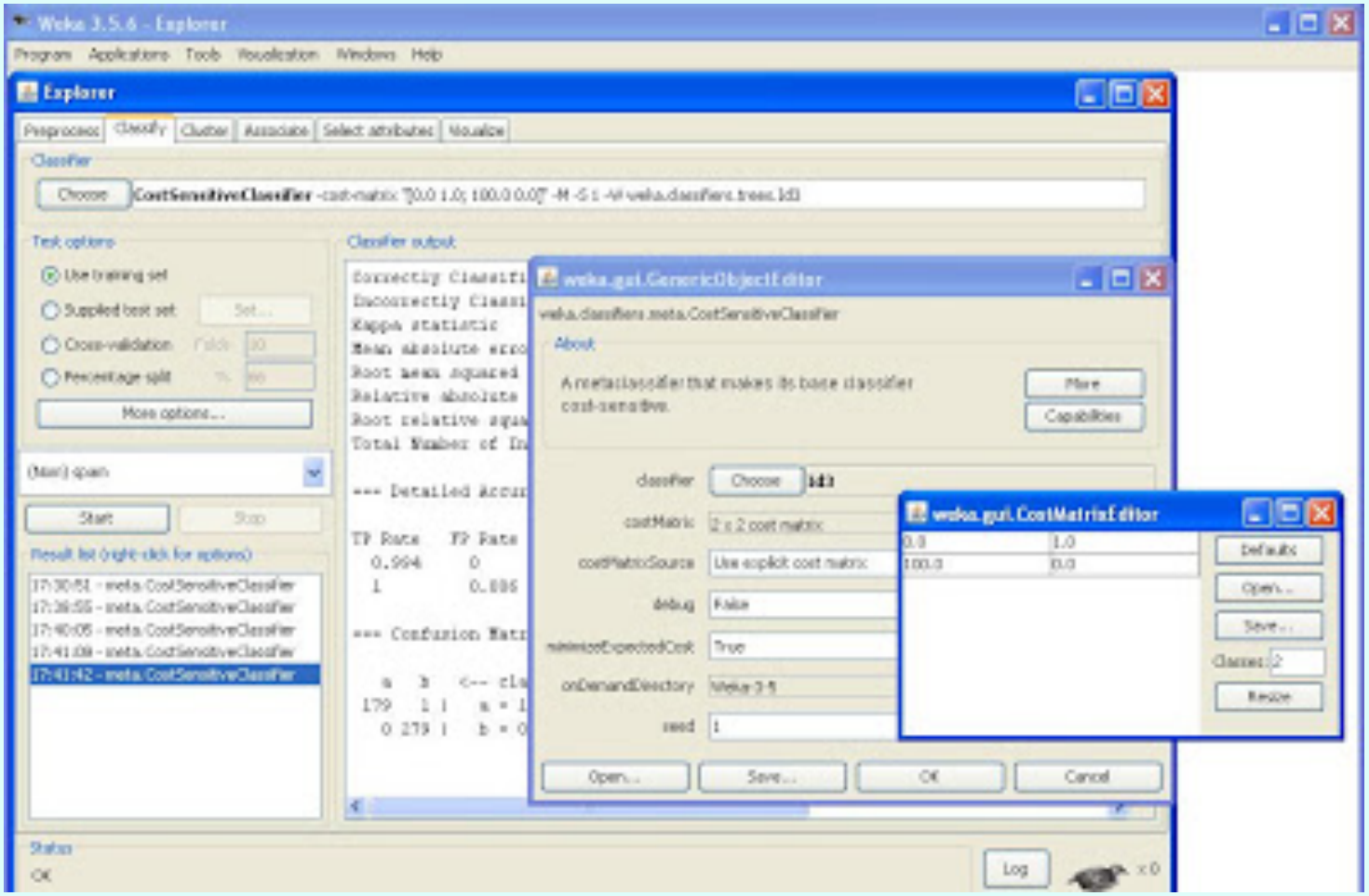
	Actual Positive $y_i = 1$	Actual Negative $y_i = 0$
Predicted Positive $c_i = 1$	C_{TP_i}	C_{FP_i}
Predicted Negative $c_i = 0$	C_{FN_i}	C_{TN_i}

Oryginalne klasy	Przewidywane klasy decyzyjne	
	Pozytywna	Negatywna
Pozytywna	*0*	*15*
Negatywna	*5*	*0*

Pomyłki mają różną interpretacje i praktyczne znaczenie

Implementacje: np. WEKA costsensitiveclassifiers

WEKA tutorial



Klasyfikacja binarna

- Niektóre zastosowania → jedna z klas posiada szczególne znaczenie, np. diagnozowanie poważnej choroby. Zadanie → klasyfikacja binarna.

Oryginalne klasy	Przewidywane klasy decyzyjne	
	Pozytywna	Negatywna
Pozytywna	*TP*	*FN*
Negatywna	*FP*	*TN*

- Nazewnictwo (inspirowane medycznie):
  - *TP* (ang. *true positive*) – liczba poprawnie sklasyfikowanych przykładów z wybranej klasy (ang. *hit*),
  - *FN* (ang. *false negative*) – liczba błędnie sklasyfikowanych przykładów z tej klasy, tj. decyzja negatywna podczas gdy przykład w rzeczywistości jest pozytywny (błąd pominięcia - z ang. *miss*),
  - *TN* (ang. *true negative*) – liczba przykładów poprawnie nie przydzielonych do wybranej klasy (poprawnie odrzuconych z ang. *correct rejection*),
  - *FP* (ang. *false positive*) – liczba przykładów błędnie przydzielonych do wybranej klasy, podczas gdy w rzeczywistości do niej nie należą (ang. *false alarm*).

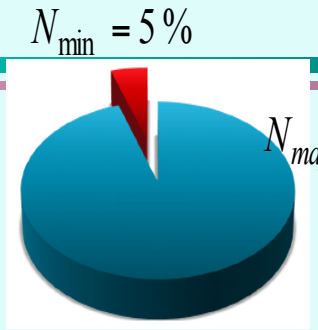
Trudności oceny trafności

True class →	Pos	Neg
Yes	200	100
No	300	400
	P=500	N=500

True class →	Pos	Neg
Yes	400	300
No	100	200
	P=500	N=500

- Oba klasyfikatory = 60% accuracy
- Lecz różnie w predykcji poszczególnych klas:
  - Lewa tabela: weak positive recognition rate/strong negative recognition rate
  - Prawa tabela: strong positive recognition rate/weak negative recognition rate

Niezbalansowane klasy – pamiętaj o innych podejściach i miarach oceny



- Czasami klasy mają mocno niezrównoważoną liczebność
  - Attrition prediction: 97% stay, 3% attrite (in a month)
  - medical diagnosis: 90% healthy, 10% disease
  - eCommerce: 99% don't buy, 1% buy
  - Security: >99.99% of Americans are not terrorists
- Podobna sytuacja dla problemów wieloklasowych.
- Skuteczność rozpoznawania klasy większościowej 97%, ale bezużyteczne dla klasy mniejszościowej o specjalnym znaczeniu (tylko kilka procent).

Miary punktowe dla niezbalansowania klas

Rozpoznawanie klasy mniejszościowej z …

Wiele miar definiowany na podstawie macierzy pomyłek

$$Sensitivity = Recall = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP} \qquad Precision = \frac{TP}{TP + FP}$$

Original	Predicted	
	+	-
+	*TP*	*FN*
-	*FP*	*TN*

Inne miary:

False-positive rate = *FP* / (*FP*+*TN*), czyli 1 – specyficzność

Agregacje: $$G\text{-}mean = \sqrt{Sensitivity * Specificity}$$

$$F\text{-}measure = \frac{(1 + \beta)^2 * Precision * Recall}{\beta^2 * Recall + Precision}$$

Analiza macierzy… spróbuj rozwiązać…

$$Sensitivity = \frac{TP}{TP+FN} = ?$$

$$Specificity = \frac{TN}{TN+FP} = ?$$

Co przewidywano

Rzeczywista Klasa	1	0
1	60	30
0	80	20

60+30 = 90 przykładów w danych należało do Klasy 1

80+20 = 100 przykładów było w Klasy 0

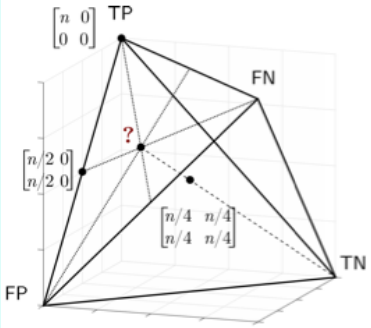
90+100 = 190 łączna liczba przykładów

Który klasyfikator jest najlepszy – Miary mogą oceniać inne aspekty! (np., UCI Breast Cancer)

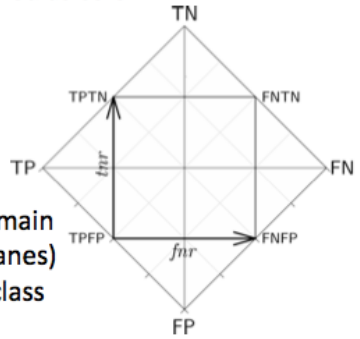
Algo	Acc	RMSE	TPR	FPR	Prec	Rec	F	AUC	Info S
NB	71.7	.4534	.44	.16	.53	.44	.48	.7	48.11
C4.5	75.5	.4324	.27	.04	.74	.27	.4	.59	34.28
3NN	72.4	.5101	.32	.1	.56	.32	.41	.63	43.37
Ripper	71	.4494	.37	.14	.52	.37	.43	.6	22.34
SVM	69.6	.5515	.33	.15	.48	.33	.39	.59	54.89
Bagg	67.8	.4518	.17	.1	.4	.17	.23	.63	11.30
Boost	70.3	.4329	.42	.18	.5	.42	.46	.7	34.48
RanFR	69.23	.47	.33	.15	.48	.33	.39	.63	20.78

Tetrahedron – analiza miar oceny klasyfikatorów

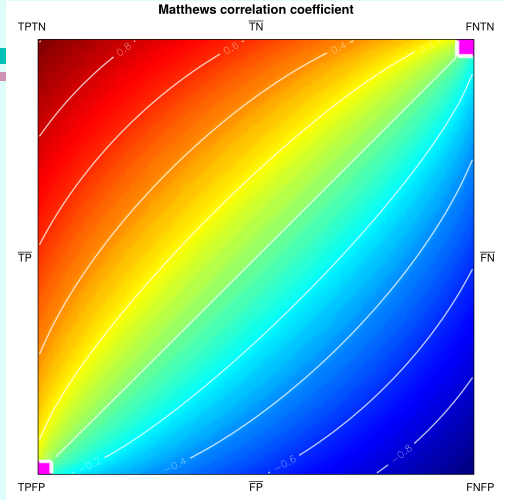
Barycentric Visualization

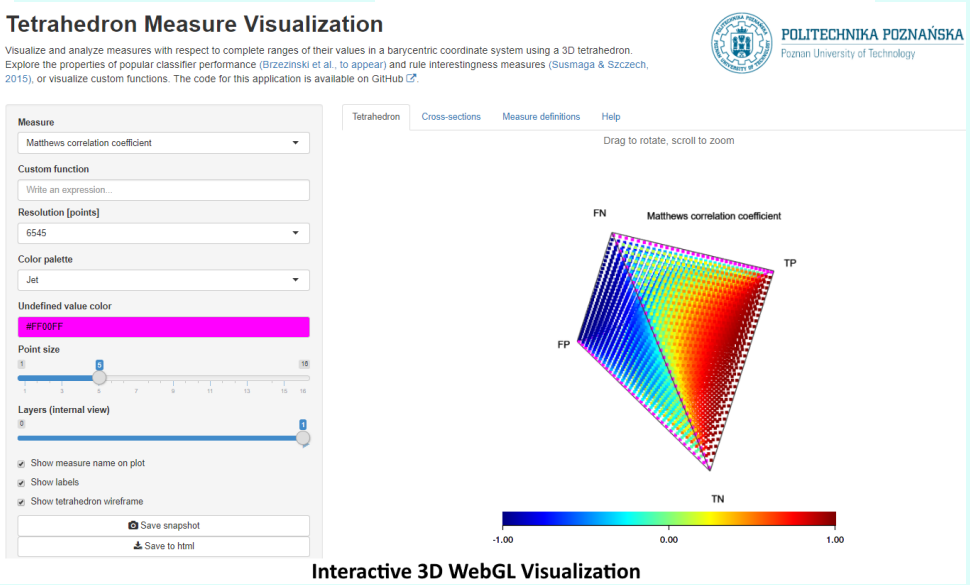


- In the barycentric coordinate system each confusion matrix is represented as a point of a 3D tetrahedron
- The value of a measure based on the depicted four values may be rendered as color



- Visual **comparison** of measures
- Visual detection of **properties**
- Insight into **full range** of measure's domain
- Consecutive **cross-sections** (cutting planes) depict measure behavior in changing class or prediction proportions





Interactive 3D WebGL Visualization

D.Brzeziński, J.Stefanowski, R.Susmaga, I.Szczech ECMLPKDD 2017, oraz artykuł w Information Science 2018

Score classifier – odpowiedź także liczbowa

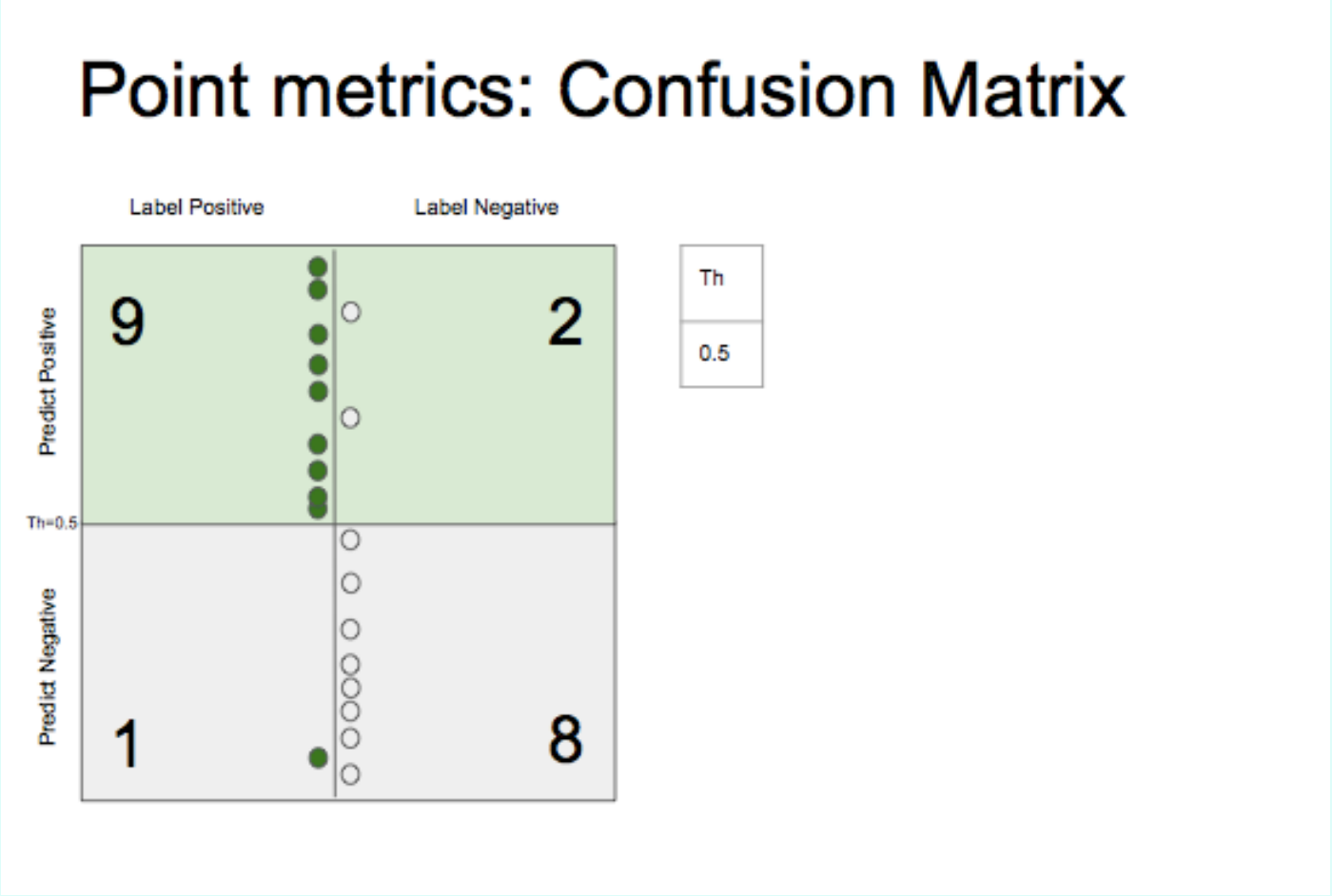
Score based models

●	Positive labelled example
○	Negative labelled example

$$\text{Prevalence} = \frac{\#\text{positives}}{\#\text{positives} + \#\text{negatives}}$$

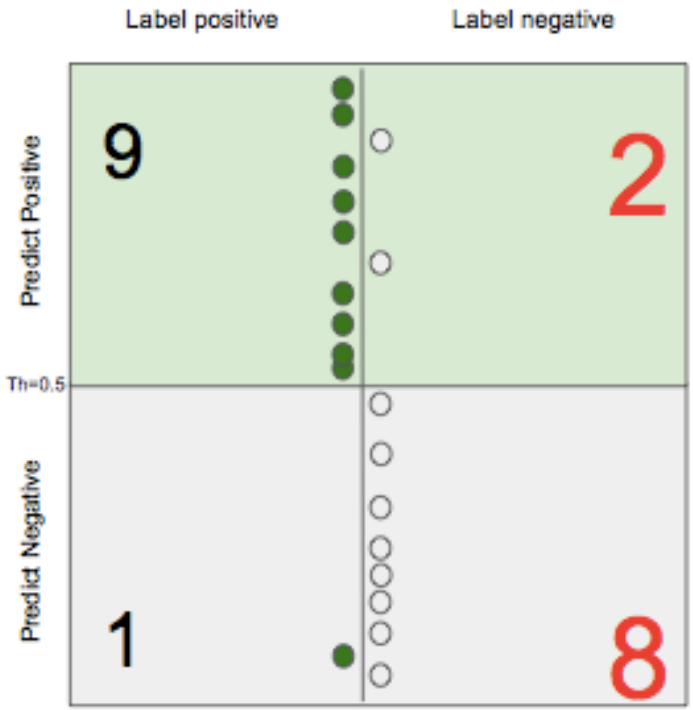
Klasyfikator - możliwość progowania wyjścia predyktora

Scores – tworzenie macierzy pomyłek



Macierz pomyłek zależy od definicji progu

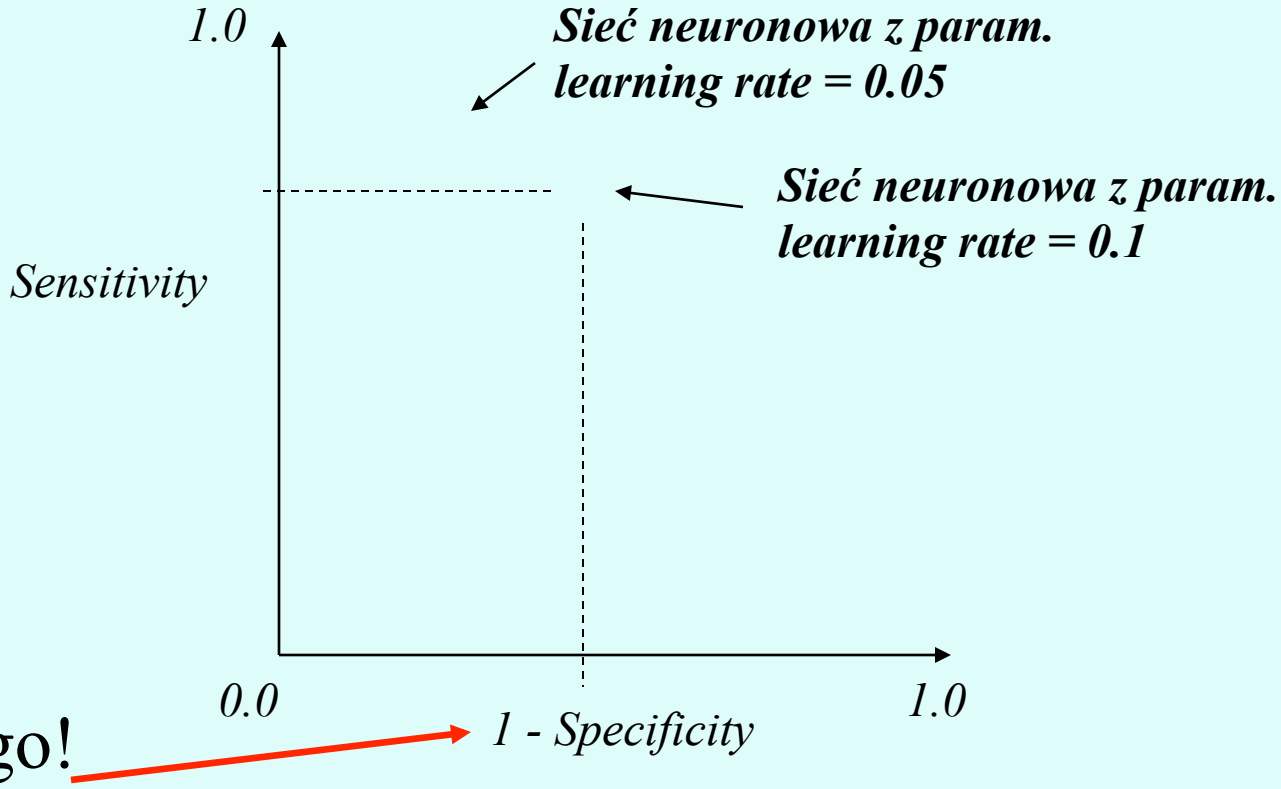
Możliwości def. miar punktowych



Th	TP	TN	FP	FN	Acc	Pr	Recall	Spec
0.5	9	8	2	1	.85	.81	.9	0.8

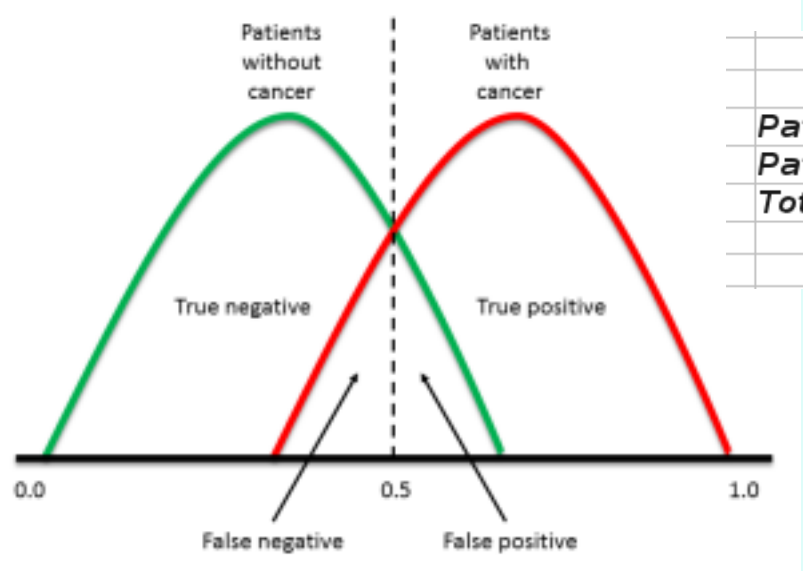
Analiza krzywej ROC

Każda technika budowy klasyfikatora może być scharakteryzowana poprzez pewne wartości miar ‘sensitivity’ i ‘specificity’. Graficznie można je przedstawić na wykresie ‘sensitivity’ vs. 1 – ‘specificity’.

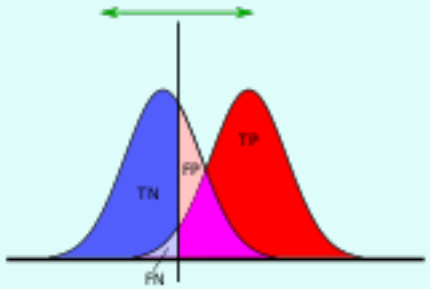


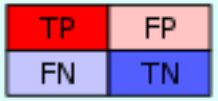
Dlaczego!

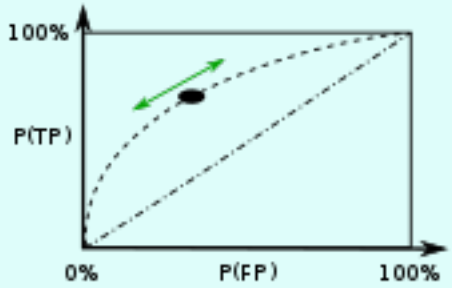
Klasyfikator paramteryzowany



	Test Positive	Test Negative	Total
Patient Diseased	160	40	200
Patient Healthy	29940	69860	99800
Total	30100	69900	100000

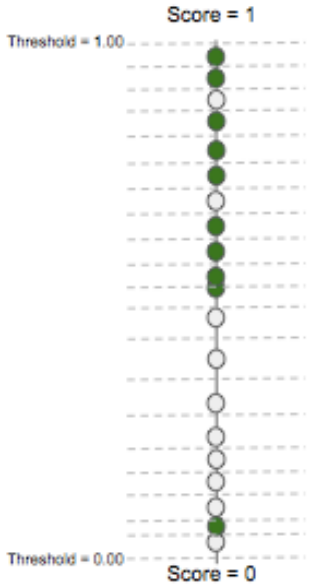






Przykład doboru progu w predykcji

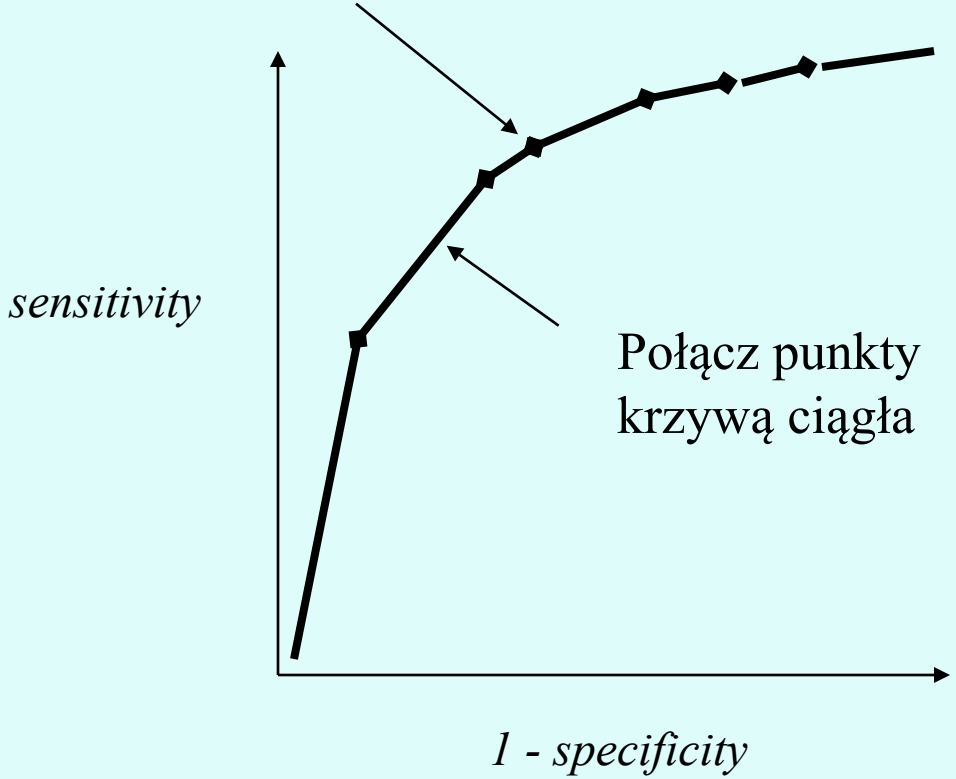
Powrót do przykładu – całościowa ocena



Threshold	TP	TN	FP	FN	Accuracy	Precision	Recall	Specificity	F1
1.00	0	10	0	10	0.50	1	0	1	0
0.95	1	10	0	9	0.55	1	0.1	1	0.182
0.90	2	10	0	8	0.60	1	0.2	1	0.333
0.85	2	9	1	8	0.55	0.667	0.2	0.9	0.308
0.80	3	9	1	7	0.60	0.750	0.3	0.9	0.429
0.75	4	9	1	6	0.65	0.800	0.4	0.9	0.533
0.70	5	9	1	5	0.70	0.833	0.5	0.9	0.625
0.65	5	8	2	5	0.65	0.714	0.5	0.8	0.588
0.60	6	8	2	4	0.70	0.750	0.6	0.8	0.667
0.55	7	8	2	3	0.75	0.778	0.7	0.8	0.737
0.50	8	8	2	2	0.80	0.800	0.8	0.8	0.800
0.45	9	8	2	1	0.85	0.818	0.9	0.8	0.857
0.40	9	7	3	1	0.80	0.750	0.9	0.7	0.818
0.35	9	6	4	1	0.75	0.692	0.9	0.6	0.783
0.30	9	5	5	1	0.70	0.643	0.9	0.5	0.750
0.25	9	4	6	1	0.65	0.600	0.9	0.4	0.720
0.20	9	3	7	1	0.60	0.562	0.9	0.3	0.692
0.15	9	2	8	1	0.55	0.529	0.9	0.2	0.667
0.10	9	1	9	1	0.50	0.500	0.9	0.1	0.643
0.05	10	1	9	0	0.55	0.526	1	0.1	0.690
0.00	10	0	10	0	0.50	0.500	1	0	0.667

ROC - analiza

Algorytm może być parametryzowany, i w rezultacie otrzymuje się serie punktów odpowiadających doborowi parametrów



Wykres nazywany ‘krzywą’ ROC.

Probabilistyczne podstawy ROC

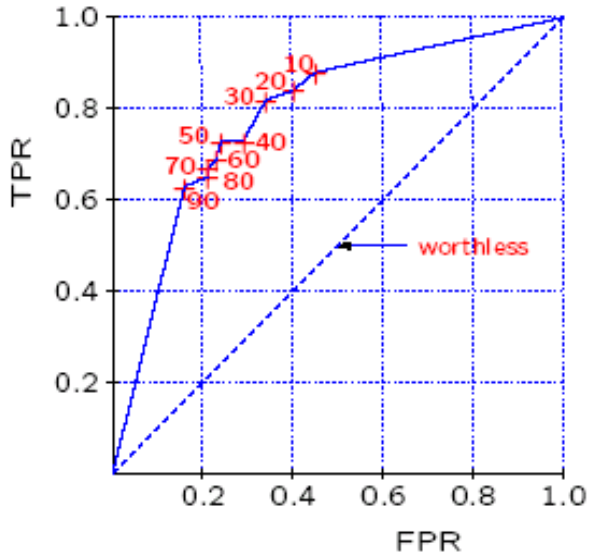
- Output of probabilistic classifier:

$$c_{max} = \arg\max_{C} P(C \mid \mathcal{E})$$

may not yield the best performance

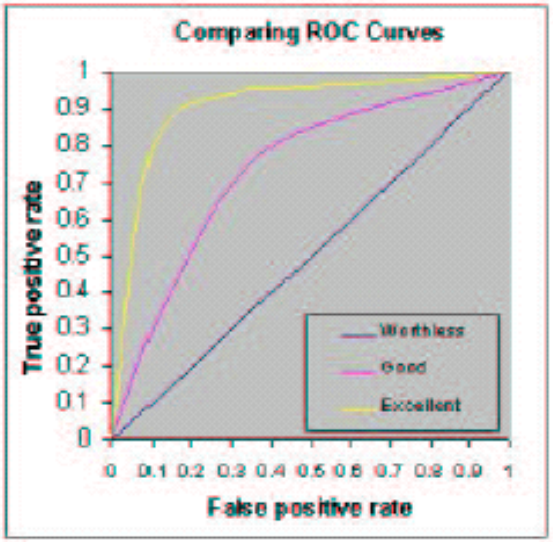
- Alternative: Receiver Operating Characteristic (ROC): determine threshold d, such that

$$C = \begin{cases} c & \text{if } P(c \mid \mathcal{E}) \geq d \\ \neg c & \text{otherwise} \end{cases}$$

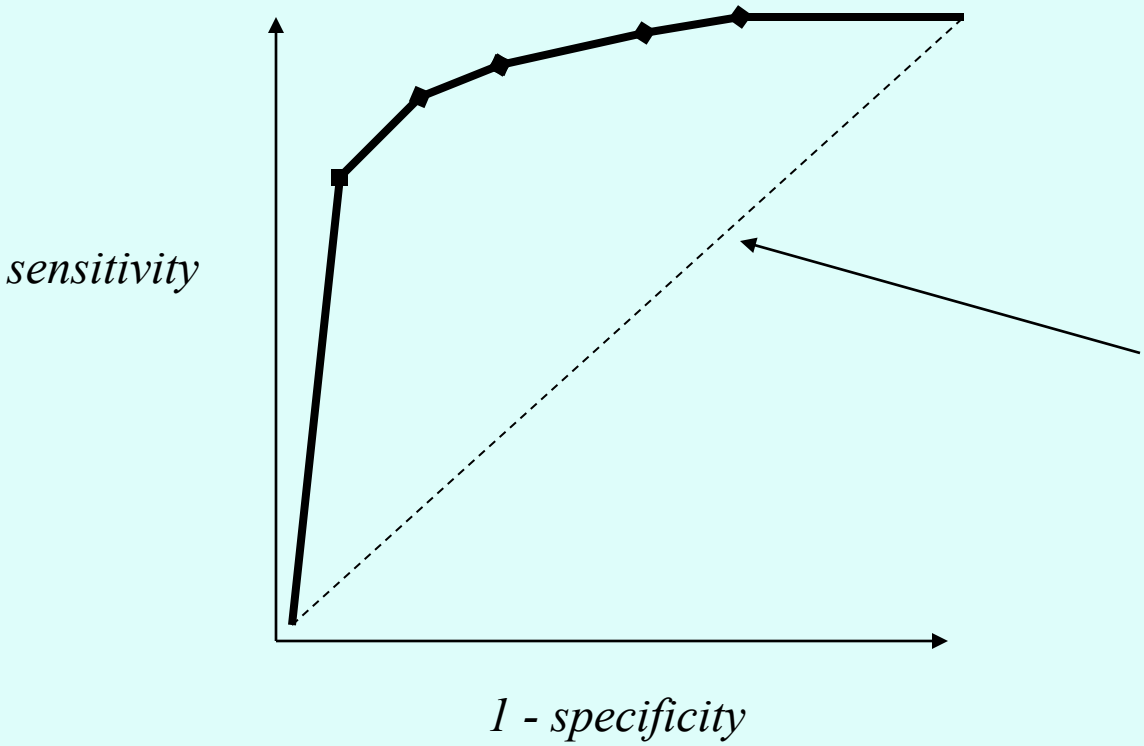


When comparing various techniques:

- actual performance for particular thresholds (cut-off points) may vary
- area under the ROC curve $A_f = \int_0^1 f(x)\mathrm{d}x$ offers good measure for comparison, with f relationship between FPR and TPR for classifier



Krzywa ROC



Im krzywa bardziej wygięta ku górnemu lewemu narożnikowi, tym lepszy klasyfikator .

Przekątna odpowiada losowemu „zgadywaniu”. Im bliżej niej, tym gorszy klasyfikator

Można porównywać działanie kilku klasyfikatorów.
Miary oceny np. AUC – pole pod krzywą,..

Porównywanie działania klasyfikatorów na ROC

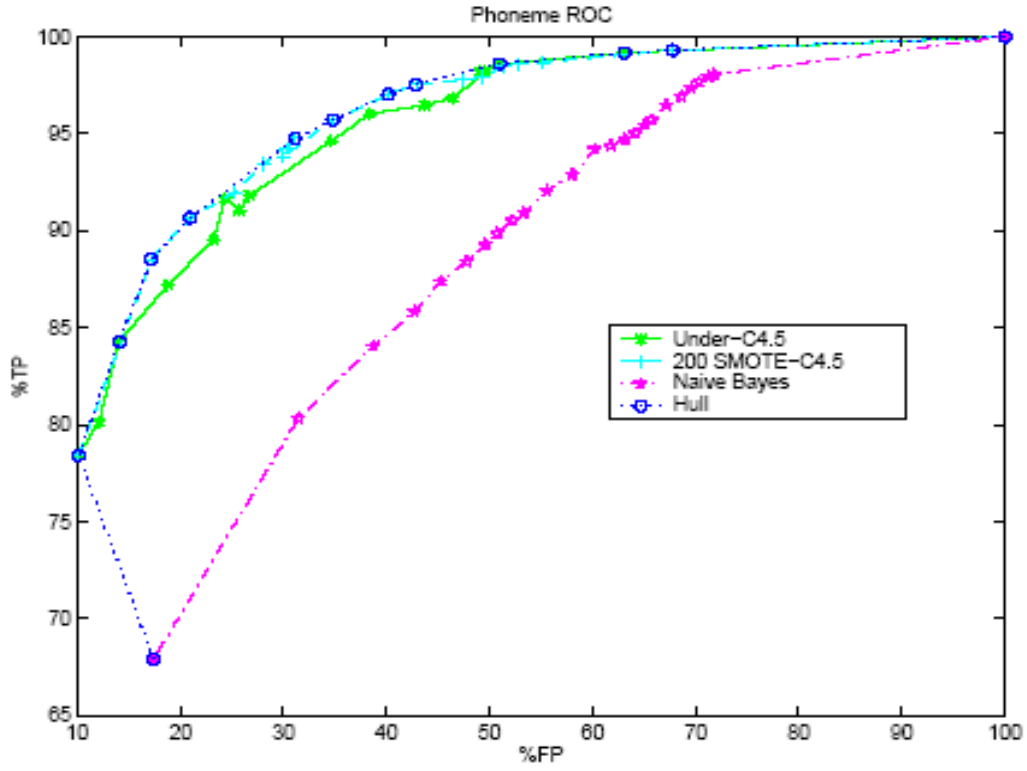


Figure 7: Phoneme. Comparison of SMOTE-C4.5, Under-C4.5, and Naive Bayes. SMOTE-C4.5 dominates over Naive Bayes and Under-C4.5 in the ROC space. SMOTE-C4.5 classifiers are potentially optimal classifiers.

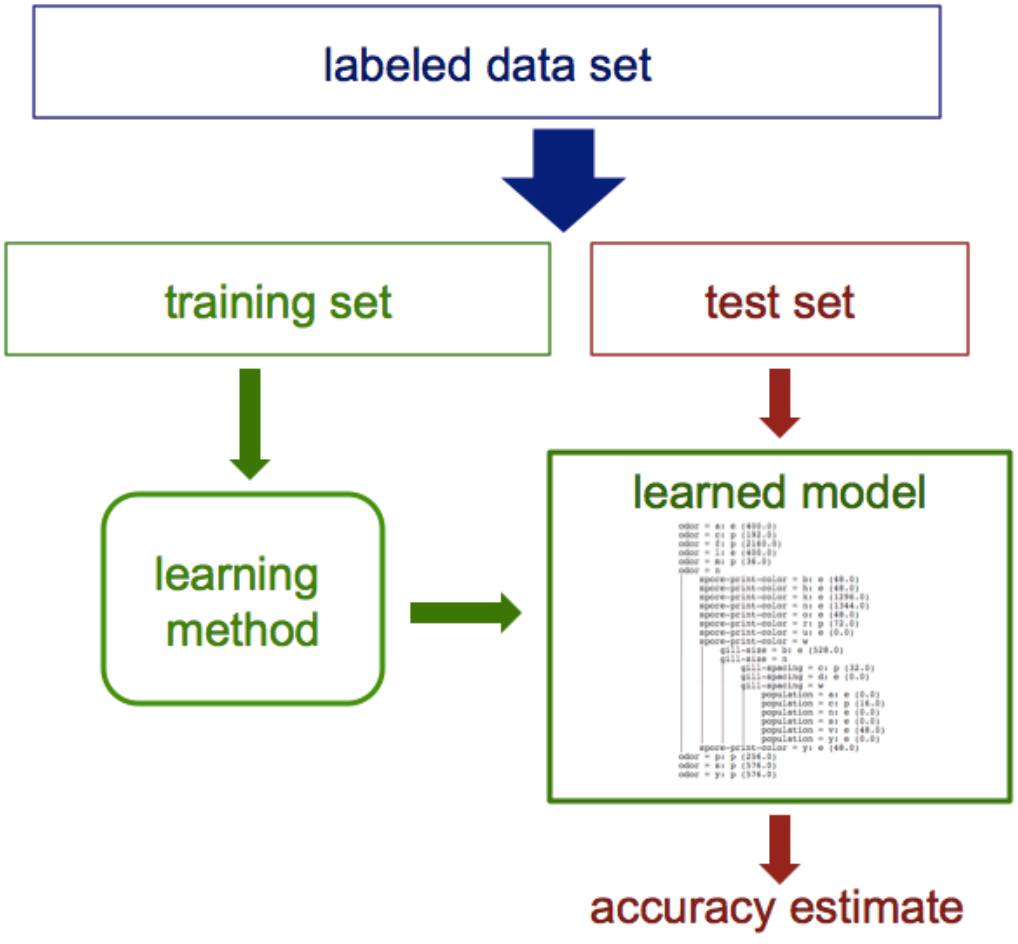
Za artykułem N.Chawla et al. nt. SMOTE

Jak szacować wiarygodnie ?

- **Zależy od perspektywy użycia wiedzy:**
  - **Predykcja klasyfikacji albo opisowa**
- **Ocena na zbiorze uczącym nie jest wiarygodna jeśli rozważamy predykcję nowych faktów!**
  - Nowe obserwacje najprawdopodobniej nie będą takie same jak dane uczące!
  - Choć zasada reprezentatywności próbki uczącej …
- Problem przeuczenia (ang. overfiting)
  - Nadmierne dopasowanie do specyfiki danych uczących powiązane jest najczęściej z utratą zdolności uogólniania (ang. generalization) i predykcji nowych faktów!

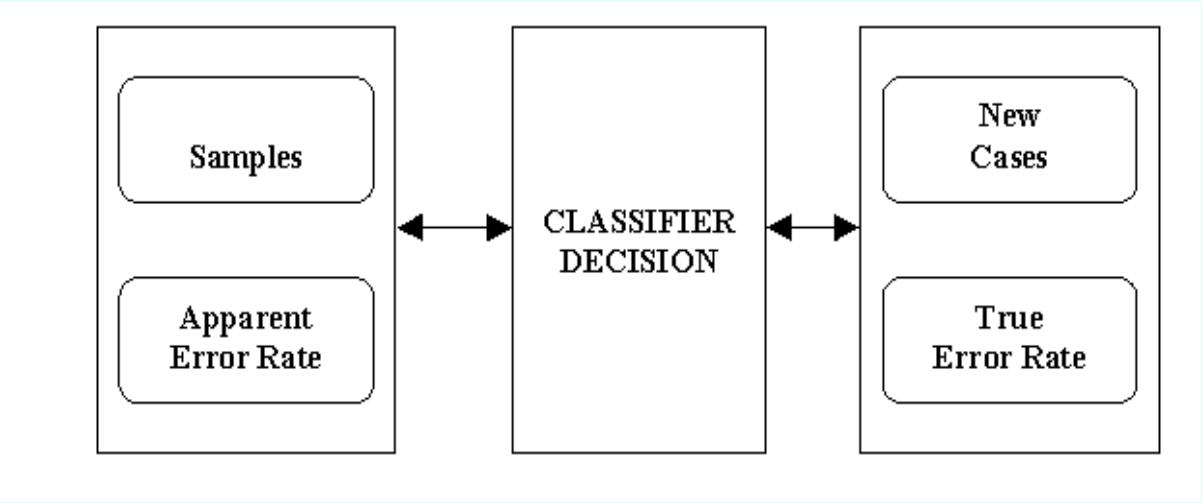
Zasada eksperymentalnej oceny

Niezależny zbiór przykładów testowych - nie wykorzystuj w fazie uczenia klasyfikatora!



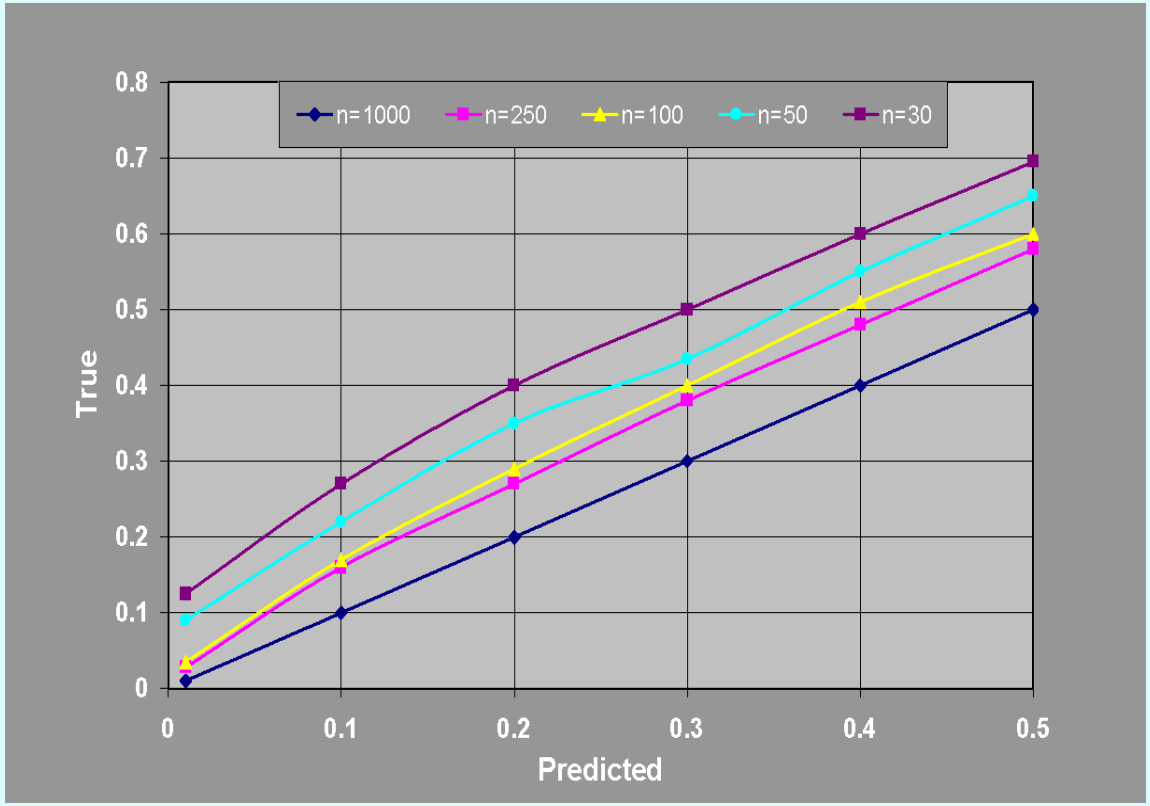
Podejście empiryczne

- Zasada „Train and test” (ucz i testuj)
- Gdy nie ma podziału zadanego przez nauczyciela, to co wykorzystasz - losowe podziały.
  - **Podziały – próba losowa LECZ ile i jakie przykłady!**
- Nadal pytanie jak szacować wiarygodnie?



True vs. aparent accuracy i liczba przykładów testowych

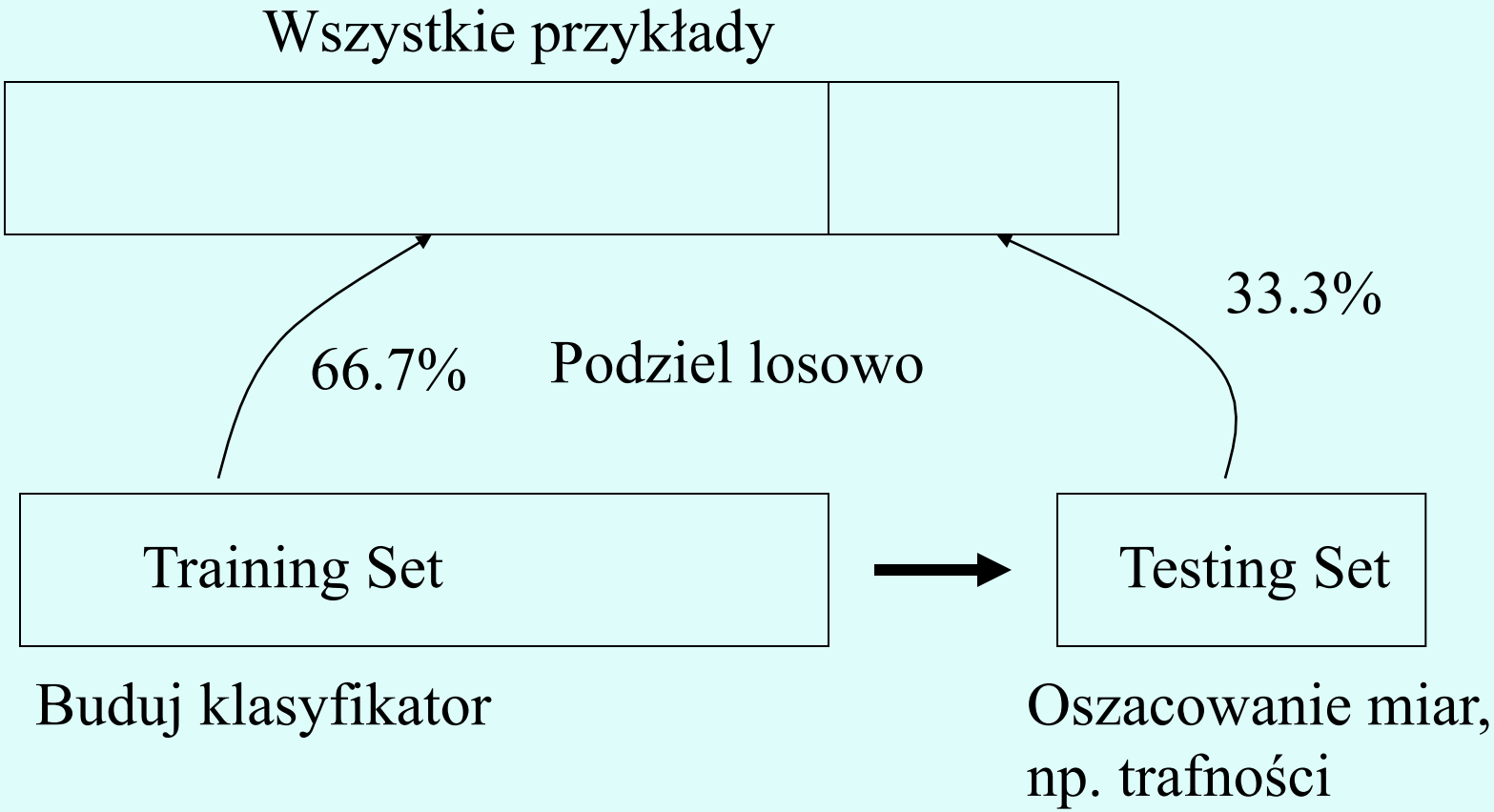
- Za S.Weiss, C.Kulikowski – podsumowanie eksperymentów ze sztucznymi danymi – ile testowych przykładów z zadanego rozkładu jest potrzebne w odpowiednim podziale losowym.



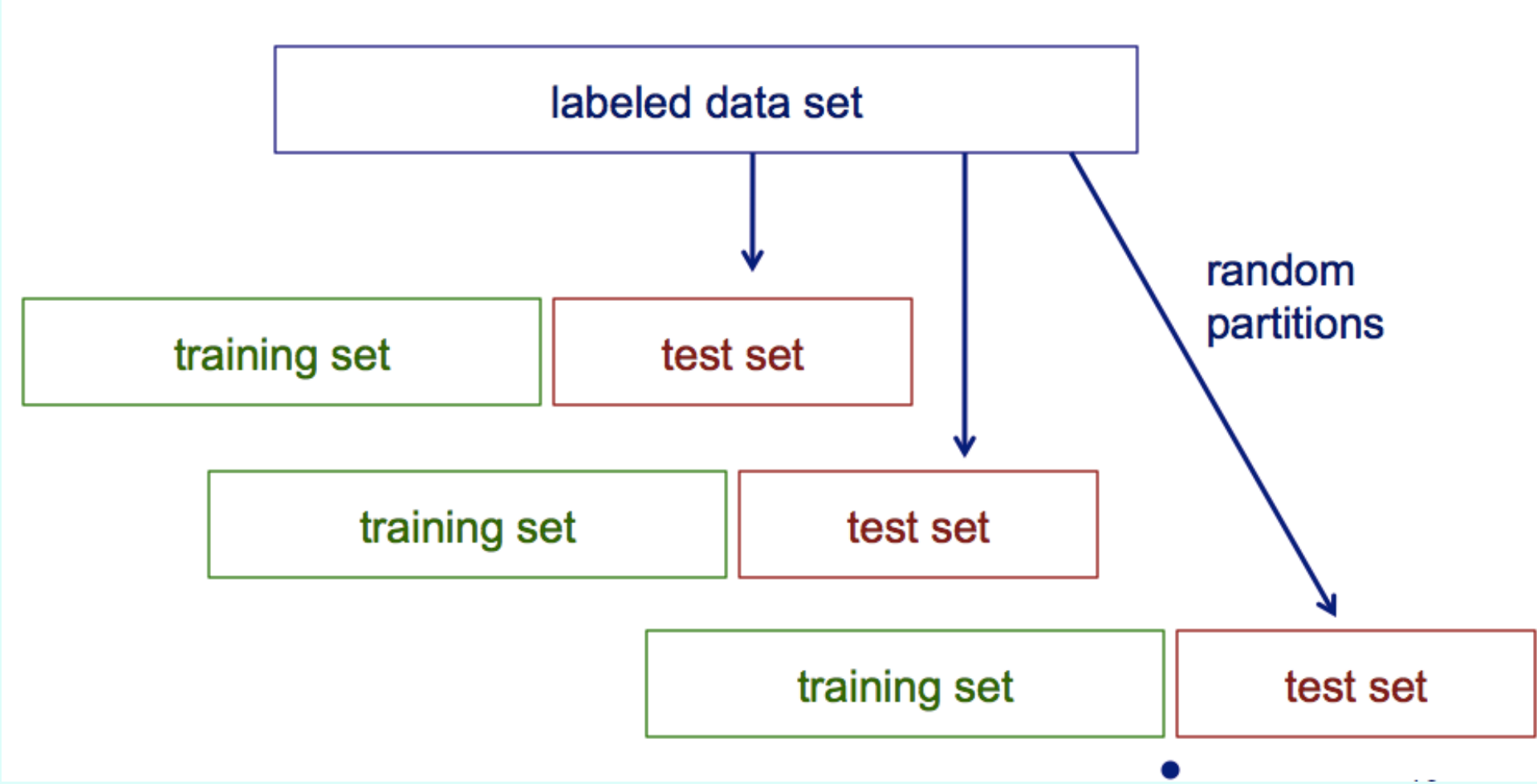
Empiryczne metody estymacji

- **Techniki podziału: „hold-out”**
  - Użyj dwóch niezależnych zbiorów: uczącego (2/3), testowego (1/3)
  - Jednokrotny podział losowy stosuje się dla dużych zbiorów (hold-out)
- **„Cross-validation” - Ocena krzyżowa**
  - Podziel losowo dane w *k* podzbiorów (równomierne lub warstwowe)
  - Użyj *k-1* podzbiorów jako części uczącej i pozostałej jako testującej (*k*-fold cross-validation).
  - Oblicz wynik średni.
  - Stosowane dla danych o średnich rozmiarach (najczęściej $k = 10$) Uwaga opcja losowania warstwowego (ang. stratified sampling).
- **Bootstrapping i leaving-one-out**
  - Dla małych rozmiarów danych.
  - „Leaving-one-out” jest szczególnym przypadkiem, dla którego liczba iteracji jest równa liczbie przykładów

Jednokrotny podział (hold-out) – duża liczba przykładów (> tysięcy)

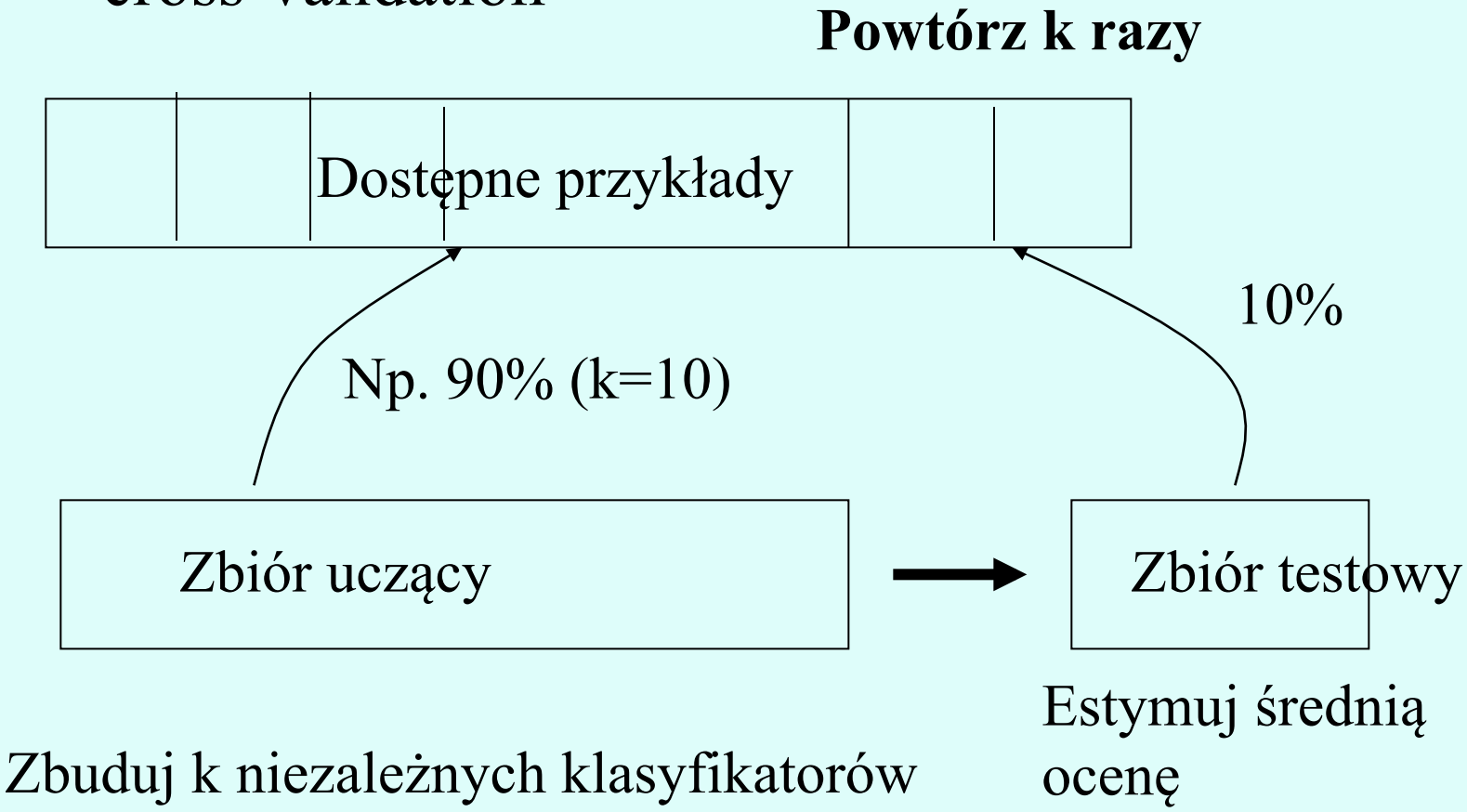


Wielokrotne podziały losowe



Mniejsza liczba przykładów (od 100 do kilku tysięcy)

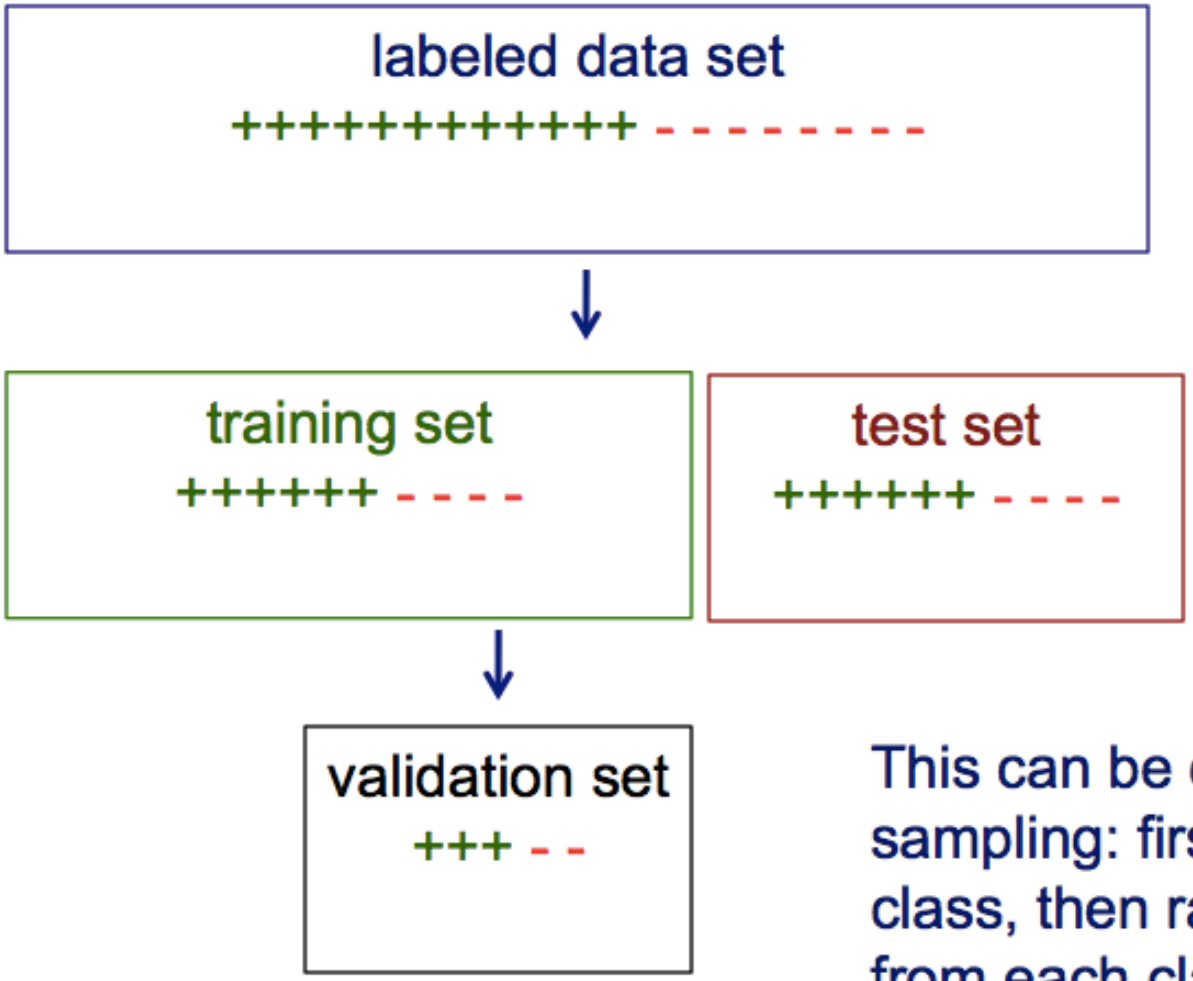
* cross-validation



K –fold cross-validation:

- Podziel losowo w k części (folds) w przybliżeniu tej samej wielkości
- Użyj jedenego podziału do testowania a reszty do budowy klasyfikatora
- Test
- Repeat k times

Losowanie warstwowe (stratified)



This can be done via stratified sampling: first stratify instances by class, then randomly select instances from each class proportionally.

Uwagi o 10 fold cross-validation

- Stosuj wersję: stratified ten-fold cross-validation
- Dlaczego 10? Doświadczenie badaczy głównie eksperymentalne (zwłaszcza związane CART)
- Stratification – warstwowość ogranicza wariancje estymaty błędy!
- Lepsza wersja: repeated stratified cross-validation”
  - np. 10-fold cross-validation is powtórzone kilka razy (z innym ziarnem rozkładu prawdopodobieństwa) i wynik średni z wielu powtórzeń.

Przykład – C4.5 cross validation

C4.5 CRX (15 attributes, 490 training cases, 200 test cases)

Data Tree Rules Cross-validation Special Help

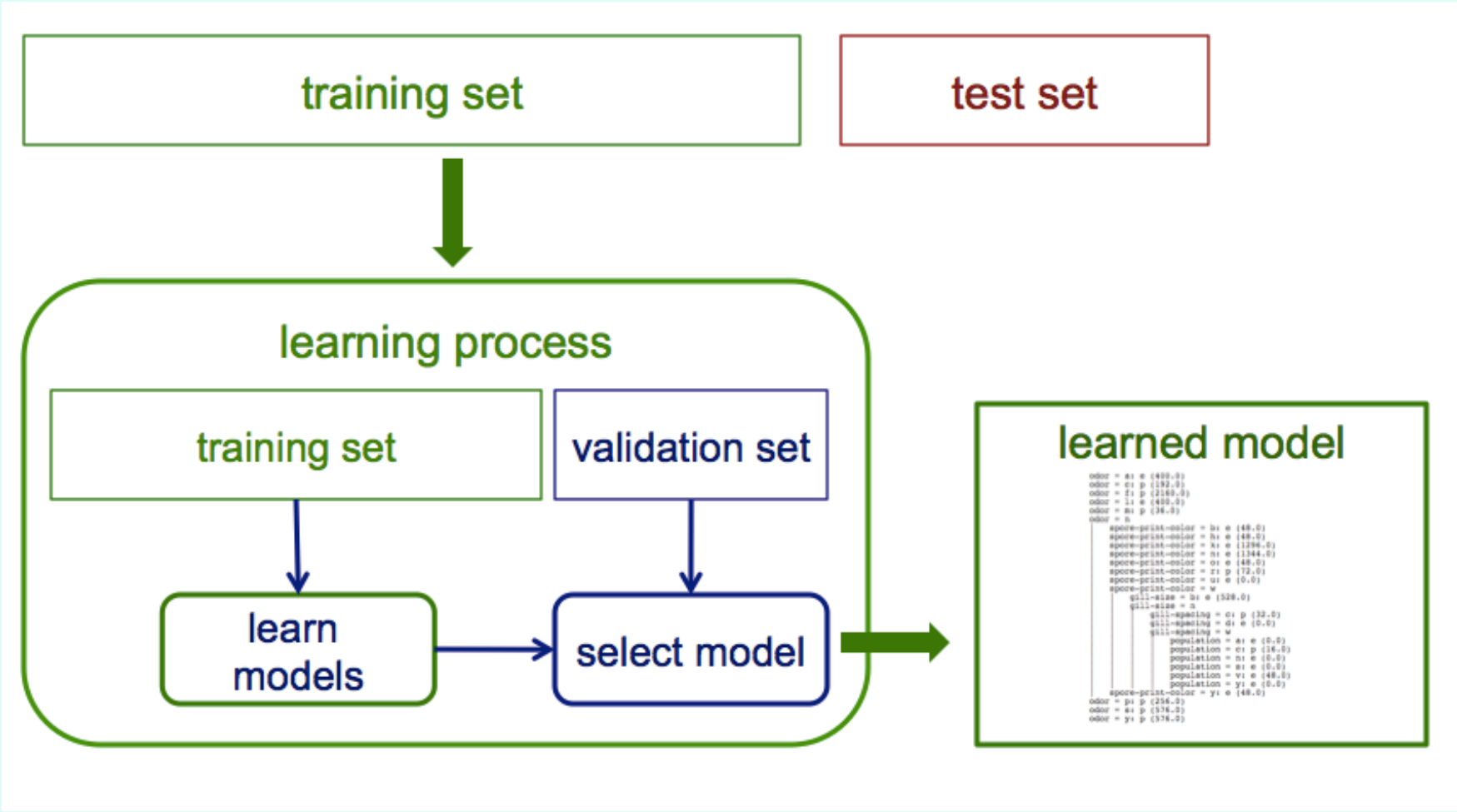
	Before pruning			After pruning			
Tree	Size	Errors	Errors (test)	Size	Errors	Errors (test)	Esti
1	101	18 (4.1%)	5 (10.2%)	50	28 (6.3%)	4 (8.2%)	15.
2	91	16 (3.6%)	9 (18.4%)	44	26 (5.9%)	9 (18.4%)	13.
3	95	16 (3.6%)	8 (16.3%)	48	23 (5.2%)	8 (16.3%)	13.
4	94	20 (4.5%)	8 (16.3%)	46	27 (6.1%)	7 (14.3%)	14.
5	102	17 (3.9%)	6 (12.2%)	51	26 (5.9%)	6 (12.2%)	14.
6	98	23 (5.2%)	11 (22.4%)	9	54 (12.2%)	5 (10.2%)	15.
7	112	21 (4.8%)	4 (8.2%)	41	30 (6.8%)	5 (10.2%)	14.
8	107	19 (4.3%)	13 (26.5%)	3	58 (13.2%)	8 (16.3%)	15.
9	88	25 (5.7%)	7 (14.3%)	40	29 (6.6%)	7 (14.3%)	14.
10	121	24 (5.4%)	7 (14.3%)	46	30 (6.8%)	7 (14.3%)	14.
Avg.	100.9	19.9 (4.5%)	7.8 (15.9%)	37.8	33.1 (7.5%)	6.6 (13.5%)	14.

Cross-validation (rules)

Ruleset	Size	Errors	Errors (test)
1	5	60 (13.6%)	1 (2.0%)
2	15	32 (7.3%)	10 (20.4%)
3	10	38 (8.6%)	9 (18.4%)
4	7	42 (9.5%)	7 (14.3%)
5	6	47 (10.7%)	5 (10.2%)
6	4	51 (11.6%)	6 (12.2%)
7	8	43 (9.8%)	6 (12.2%)
8	2	58 (13.2%)	8 (16.3%)
9	10	40 (9.1%)	6 (12.2%)
10	5	49 (11.1%)	7 (14.3%)
Avg.	7.2	46.0 (10.4%)	6.5 (13.2%)

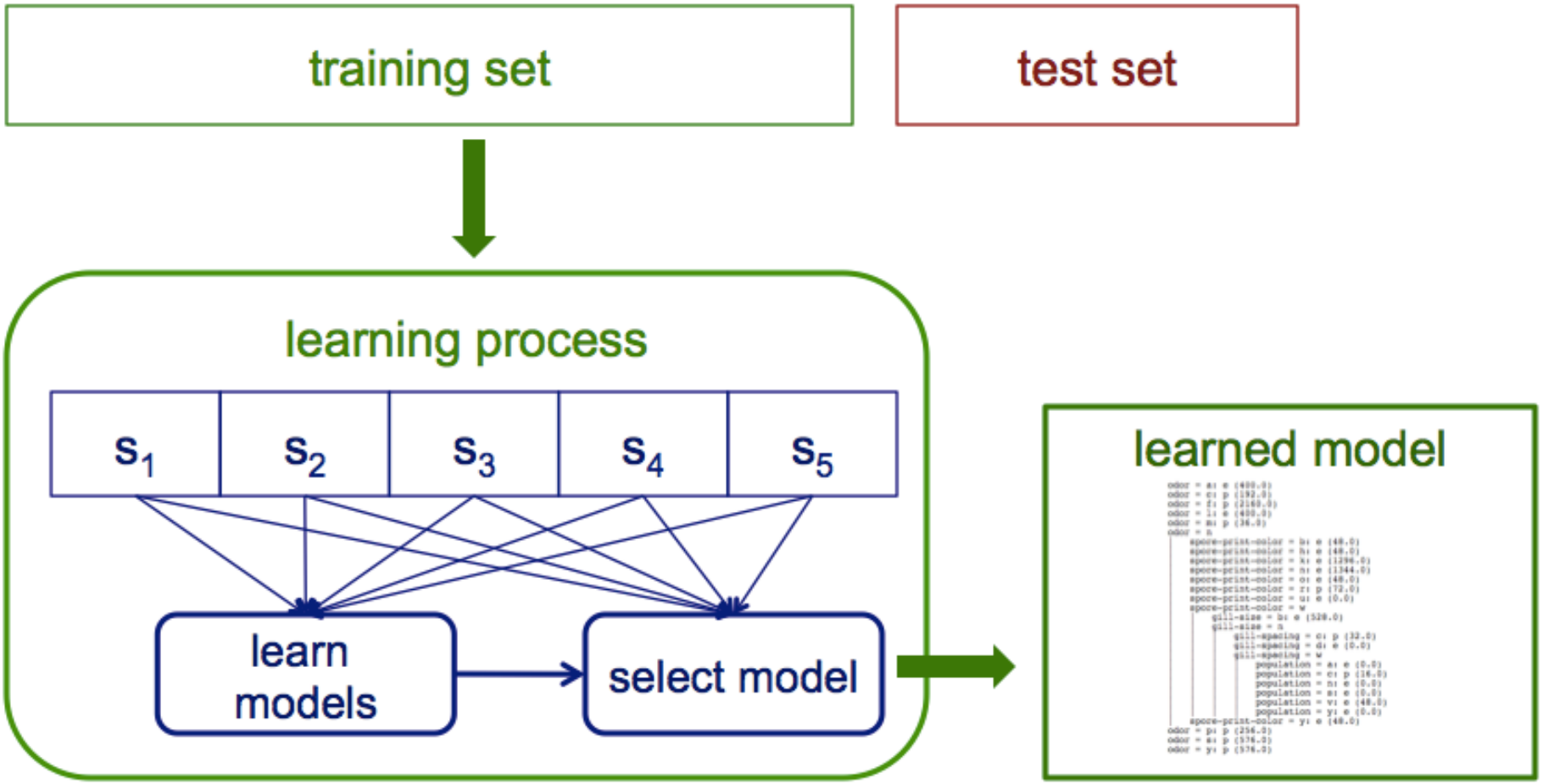
Oryginalna wersja JR Quinlana – możliwość analizy każdego z podziałów = patrz laboratorium

Strojenie klasyfikatora – validation examples



Wydzielenie zbioru walidującego z części uczącej

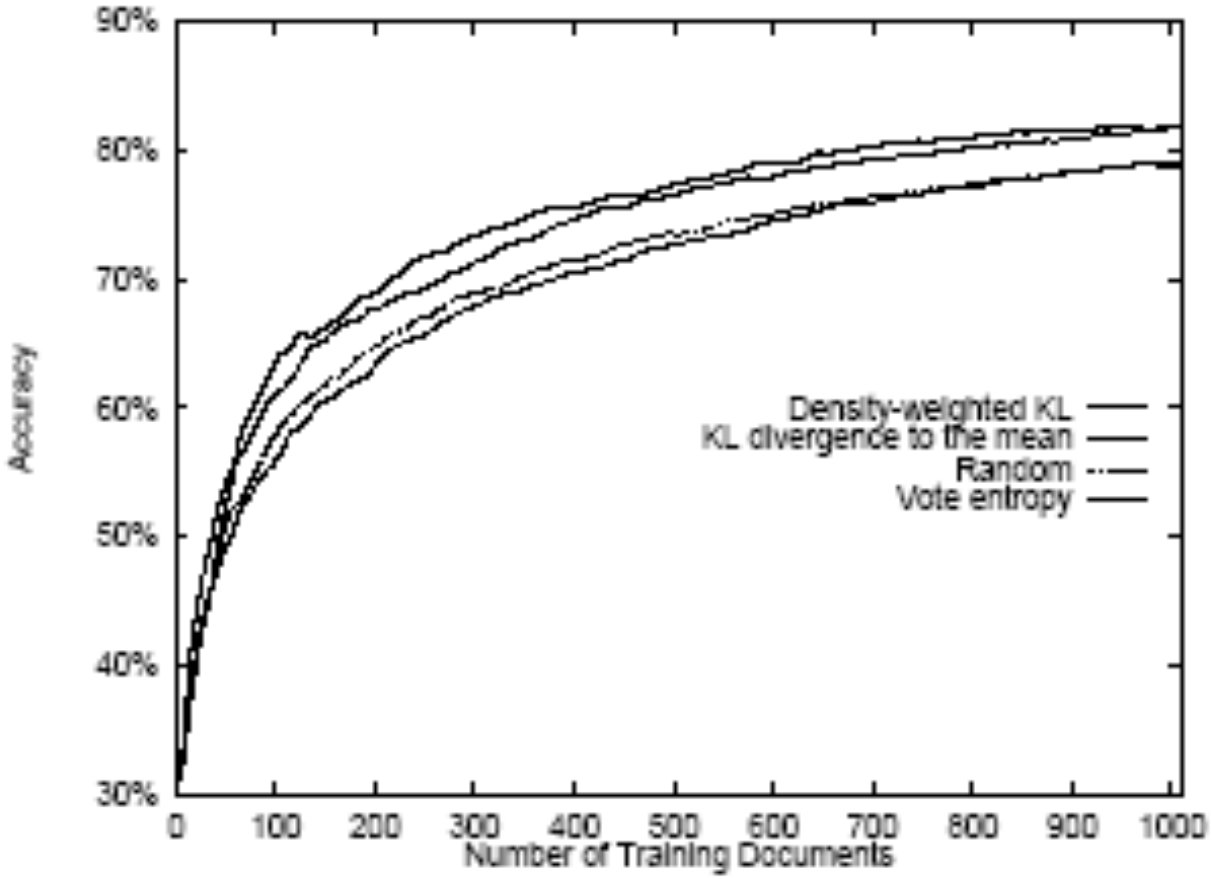
Strojenie klasyfikatora – validation examples



Wykorzystaj wewnętrzną ocenę krzyżową

Krzywe uczenia się - learning curve

Klasyfikacja tekstów przez Naive Bayes 20 Newsgroups dataset -



[McCallum & Nigam, 1998]

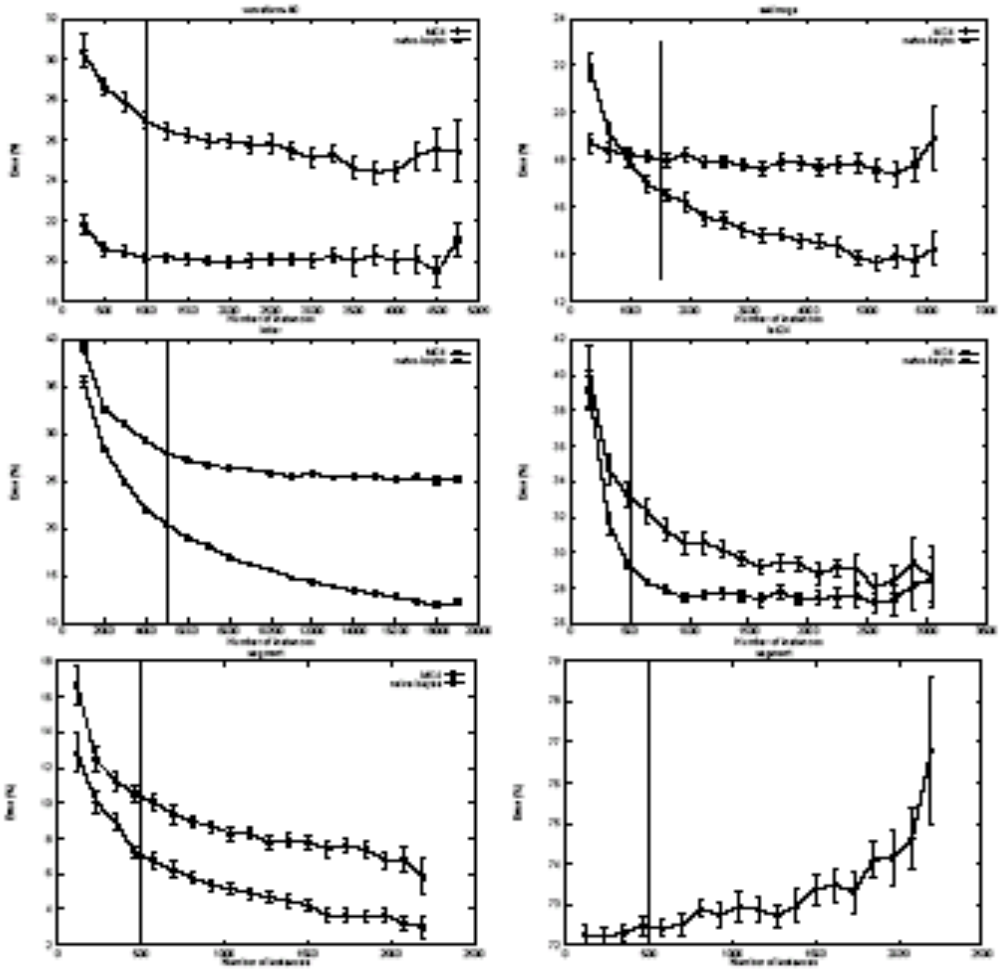
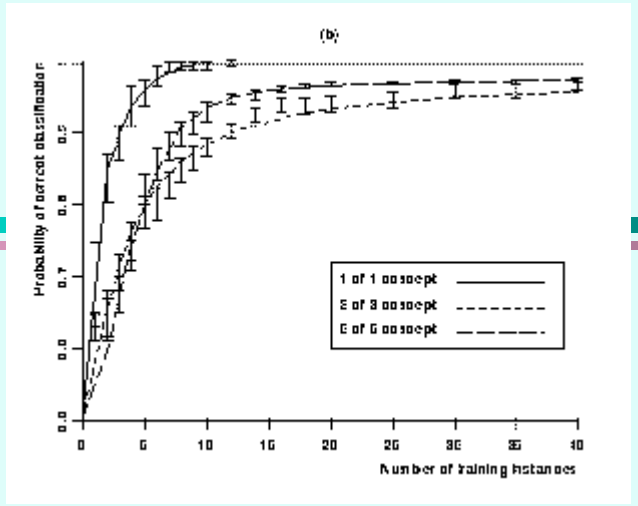
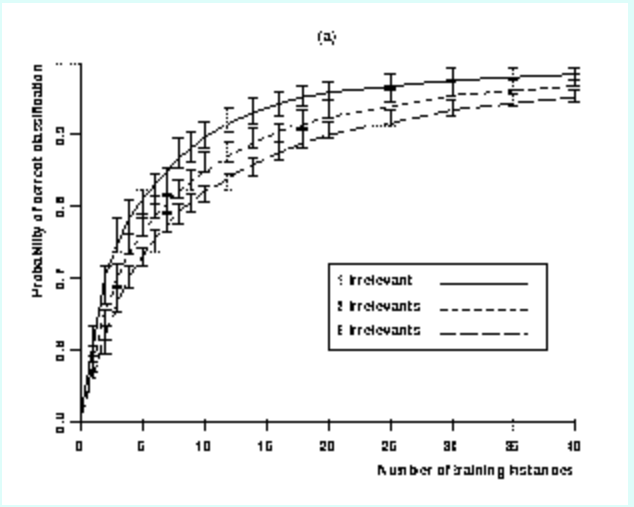


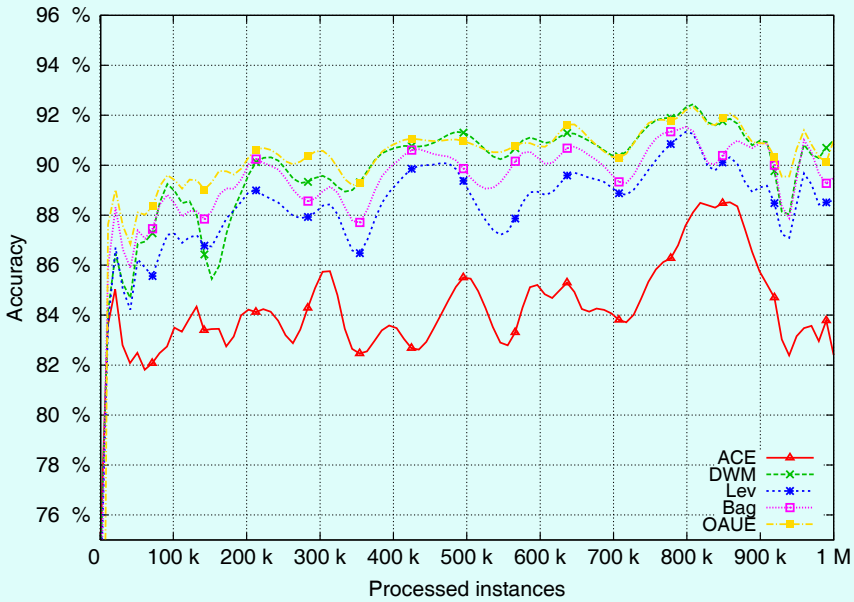
Figure 4. Learning curves for selected datasets showing different behaviors of MC4 and Naive-Bayes. Waveform represents stabilization at about 3,000 instances; satimage represents a cross-over as MC4 improves while Naive-Bayes does not; letter and segment (left) represent continuous improvements, but at different rates in letter and similar rates in segment (left); LED24 represents a case where both algorithms achieve the same error rate with large training sets; segment (right) shows MC4(1), which exhibited the surprising behavior of degrading as the training set size grew (see text). Each point represents the mean error rate for 20 runs for the given training set size as tested on the holdout sample. The error bars show one standard deviation of the estimated error. Each vertical bar shows the training set size we chose for the rest of the paper following our desiderata. Note (e.g., in waveform) how small training set sizes have high standard deviations for the estimates because the training set is small and how large training set sizes have high standard deviations because the test set is small.



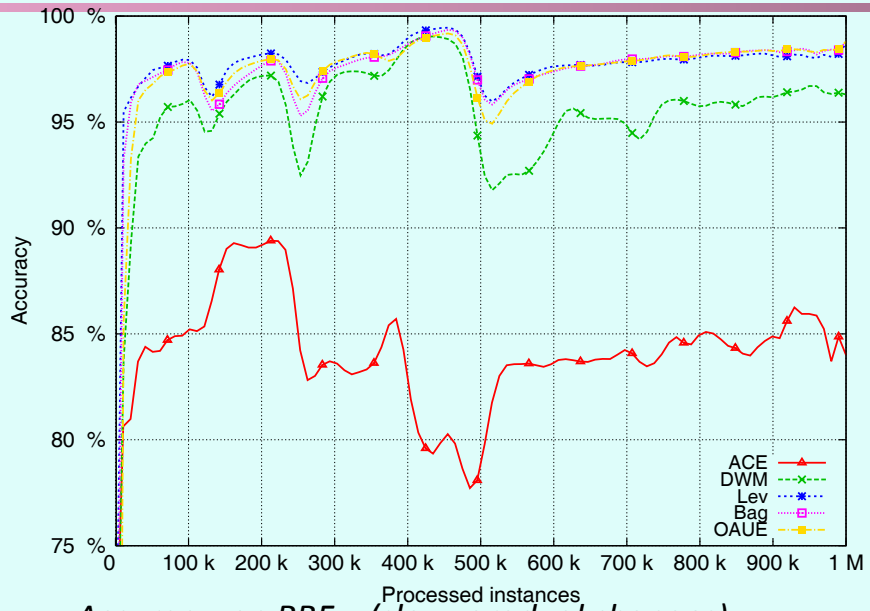


- Przykład prezentacji z artykułu Bauer, Kohavi nt. porównania różnych rozwiązań w klasyfikatorach złożonych.

Reacting to concept drifts

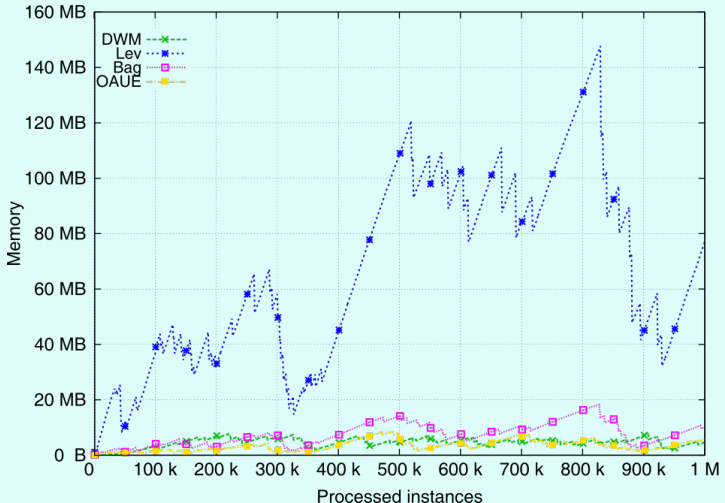


Prequential classification accuracy on Hyper$_F$ (fast, sudden changes)



Accuracy on RBF$_{GR}$ (slow, gradual changes)

Memory Hyper$_F$

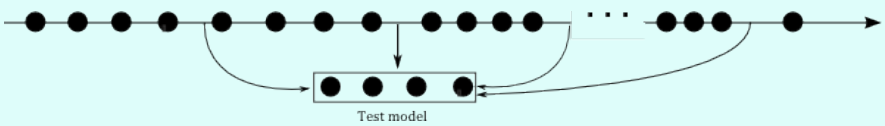


More → D. Brzezinski, J. Stefanowski, Combining Block-based and Online Methods in Learning Ensembles from Concept Drifting Data Streams. Information Sciences, 265, (2014) 50-67.

Ocena klasyfikatorów strumieniowych

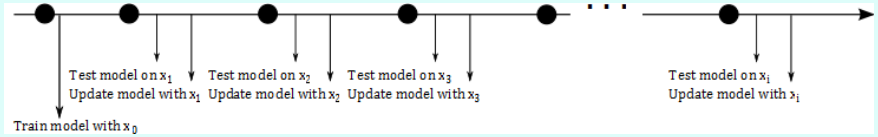
- **Holdout**

  [np., Kirkby 2007]



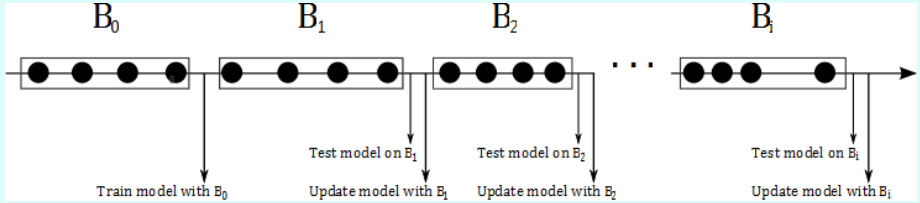
- **Test-then-train**

  [np., Kirkby 2007]



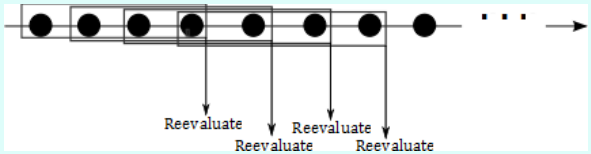
- **Block-based evaluation**

  [np., Brzezinski & Stefanowski 2010]



- **Prequential accuracy**

  [Gama et al. 2013]



- **Inne miary**

  [Bifet & Frank 2010, Zliobaite et al. 2014]

Porównywanie wielu klasyfikatorów

- Często należy porównać dwa klasyfikatory
- Uwaga: porównanie z niezależnością od danych?
  - Generatory losowe
  - Rzeczywiste dane (problem dependent)
- Oszacuj 10-fold CV estimates.
- Trudność: wariancja oszacowania.
- Możesz oczywiście zastosować „repeated CV”.
- Ale jak wiarygodnie ustalić konkluzję – który jest lepszy?

Porównywanie klasyfikatorów

- Jak oceniać skuteczność klasyfikacyjną dwóch różnych klasyfikatorów na tych samych danych?
- Ograniczamy zainteresowanie wyłącznie do trafności klasyfikacyjnej – oszacowanie techniką 10-krotnej oceny krzyżowej (ang. *k-fold cross validation*).
- Zastosowano dwa różne algorytmy uczące *AL*1 i *AL*2 do tego samego zbioru przykładów, otrzymując dwa różne klasyfikatory *KL*1 i *KL*2. Oszacowanie ich trafności klasyfikacyjnej (10-fcv):
  - klasyfikator *KL*1 → 86,98%
  - klasyfikator *KL*2 → 87,43%.
- Czy uzasadnione jest stwierdzenie, że klasyfikator *KL*2 jest skuteczniejszy niż klasyfikator *KL*1?

Analiza wyniku oszacowania trafności klasyfikowania

Podział	Kl_1	Kl_2
1	87,45	88,4
2	86,5	88,1
3	86,4	87,2
4	86,8	86
5	87,8	87,6
6	86,6	86,4
7	87,3	87
8	87,2	87,4
9	88	89
10	85,8	87,2
Srednia	**86,98**	**87,43**
Odchylenie	**0,65**	**0,85**

- Test statystyczny (t-Studenta dla par zmiennych/zależnych)
- H0 : ?
- *t*emp = 1,733 ($p = 0,117$) ???
- ALE !!! W art. naukowych zastosuj odpowiednie poprawki przy wykonaniu testu (kwestia naruszenia założeń co do rozkładu *t*).
- Nadeau i Bengio 2003 proponują modyfikacje testu.

Corrected t-test [Nadeau & Bengio, 2003]

Correction

Change the denominator of the t statistics:

$$\frac{\bar{\ell}}{\sqrt{\frac{1}{K}S_\ell^2}} \Longrightarrow \frac{\bar{\ell}}{\sqrt{(\frac{1}{K} + \frac{n_T}{n-n_T})S_\ell^2}}$$

Experimentally proved to be reliable, without enlarged type I error, with very good power curve.

Przykład zastosowania „paired t-test” $\alpha = 0{,}05$

Table 1. Comparison of classification accuracies [%] obtained by the single MODLEM based classifier and the bagging approach

Name of dataset	Single MODLEM	Bagging - with different T 3	5	7	10
bank	93.81 ± 0.94	95.05 ± 0.91	94.95 ± 0.84	95.22 ± 1.02	93.95* ± 0.94
buses	97.20 ± 0.94	98.05* ± 0.97	99.54 ± 1.09	97.02* ± 1.15	97.45* ± 1.13
zoo	94.64 ± 0.67	93.82* ± 0.68	93.89* ± 0.71	93.47 ± 0.73	93.68 ± 0.70
hepatitis	78.62 ± 0.93	82.00 ± 1.14	84.05 ± 1.1	81.05 ± 0.97	84.0 ± 0.49
iris	94.93 ± 0.5	95.13* ± 0.46	94.86* ± 0.54	95.06* ± 0.53	94.33* ± 0.59
automobile	85.23 ± 1.1	82. 98 ± 0.86	83.0 ±0.99	82.74 ±0.9	81.39 ± 0.84
segmentation	85.71 ± 0.71	86.19* ± 0.82	87.62 ± 0.55	87.61 ± 0.46	87.14 ± 0.9
glass	72.41 ± 1.23	68.5 ± 1.15	74.81 ± 0.94	74.25 ± 0.89	76.09 ± 0.68
bricks	90.32* ± 0.82	90.3 * ± 0.54	89.84* ± 0.65	91.21* ± 0.48	90.77* ± 0.72
vote	92.67 ± 0.38	93.33* ± 0.5	94.34 ± 0.34	95.01 ± 0.44	96.01 ± 0.29
bupa	65.77 ± 0.6	64.98* ± 0.76	76.28 ± 0.44	70.74 ± 0.96	75.69 ± 0.7
election	88.96 ± 0.54	90.3 ± 0.36	91.2 ± 0.47	91.66 ± 0.34	90.75 ± 0.55
urology	63.80 ± 0.73	64.8 ± 0.83	65.0 ± 0.43	67.40 ± 0.46	67.0 ± 0.67
german	72.16 ± 0.27	73.07* ± 0.39	76.2 ± 0.34	75.62 ± 0.34	75.75 ± 0.35
crx	84.64 ± 0.35	84.74* ± 0.38	86.24 ± 0.39	87.1 ± 0.46	89.42 ± 0.44
pima	73.57 ± 0.67	75.78* ± 0.6	74.35* ± 0.64	74.88 ± 0.44	77.87 ± 0.39

Jeden z klasyfikatorów złożonych vs. pojedynczy klasyfikator – z pracy J.Stefanowski

- z pracy T.Diettricha o ensembles

Table 2: Results on Five Domains (best error rate in **boldface**)

Task	Test set size	C4.5	200-fold bootstrap C4.5	200-fold random C4.5
Vowel	462	0.5758	0.5152	**0.4870***
Soybean	376	0.1090	**0.0984**	0.1090
Part-of-Speech	3060	0.0827	**0.0765**	0.0788[a]
NETtalk	7242	0.3000	0.2670***	**0.2500*****
Letter Recognition	4000	0.2010	0.0038***	**0.0000*****

Difference from C4.5 significant at $p < 0.05$*, 0.001***. [a]256-fold random.

Porównanie działania dwóch klasyfikatorów DT oraz n2 na wielu zbiorach danych (wyniki średnie z 10-oceny krzyżowej wraz z przedziałem ufności α=0,95)

Data set	Classification accuracy *DT* (%)	Classification accuracy n^2 (%)	Improvement n^2 vs. *DT* (%)
Automobile	85.5 ± 1.9	87.0 ± 1.9	1.5*
Cooc	54.0 ± 2.0	59.0 ± 1.7	5.0
Ecoli	79.7 ± 0.8	81.0 ± 1.7	1.3
Glass	70.7 ± 2.1	74.0 ± 1.1	3.3
Hist	71.3 ± 2.3	73.0 ± 1.8	1.7
Meta-data	47.2 ± 1.4	49.8 ± 1.4	2.6
Primary Tumor	40.2 ± 1.5	45.1 ± 1.2	4.9
Soybean-large	91.9 ± 0.7	92.4 ± 0.5	0.5*
Vowel	81.1 ± 1.1	83.7 ± 0.5	2.6
Yeast	49.1 ± 2.1	52.8 ± 1.8	3.7

Problem Statement

Comparison of two algorithms on m datasets:

datasets	classifier 1	classifier 2	difference (ℓ_j)
dataset 1	0.09	0.06	$+0.03$
dataset 2	0.25	0.36	-0.11
dataset 3	0.019	0.012	$+0.007$
...	...	...	...

We treat the difference in errors on each dataset ℓ_j, $j = 1, \ldots, m$, as a separate observation.

The differences cannot be simply compared, because the errors on each dataset may have different scales (see example above).

Globalna ocena (2 alg. wiele zb. danych)

Wilcoxon test (sparowany test rangowy)

H0: nie ma różnicy oceny klasyfikatorów

1. Różnice oceny klasyfikatorów uporządkuj wg. wartości bezwzględnych i przypisz im rangi.
2. R+ suma rang dla sytuacji gdy klasyfikator 1 jest lepszy niż klasyfikator 2 // R- sytuacja odwrotna
3. Oblicz statystykę T = min{R+;R-}

   Rozkład T jest stabelaryzowany / prosta reguła decyzyjna
4. Dla odpowiednio dużej liczby *m* zbiorów danych można stosować przybliżenie z

$$z = \frac{\min\{R+;R-\} - \frac{1}{4}m(m-1)}{\sqrt{\frac{1}{24}m(m+1)(2m+1)}}$$

Example of AUC comparison [Demšar, 2006]

	C4.5	C4.5+m	difference	rank
adult (sample)	0.763	0.768	+0.005	3.5
breast cancer	0.599	0.591	−0.008	7
breast cancer wisconsin	0.954	0.971	+0.017	9
cmc	0.628	0.661	+0.033	12
ionosphere	0.882	0.888	+0.006	5
iris	0.936	0.931	−0.005	3.5
liver disorders	0.661	0.668	+0.007	6
lung cancer	0.583	0.583	0.000	1.5
lymphography	0.775	0.838	+0.063	14
mushroom	1.000	1.000	0.000	1.5
primary tumor	0.940	0.962	+0.022	11
rheum	0.619	0.666	+0.047	13
voting	0.972	0.981	+0.009	8
wine	0.957	0.978	+0.021	10

$$R_+ = 3.5 + 9 + 12 + 5 + 6 + 14 + 11 + 13 + 8 + 10 + 1.5 = 83$$

$$R_- = 7 + 3.5 + 1.5 = 12$$

$$z = -2.54 < -1.96 \quad \text{(Null hypothesis rejected.)}$$

Recommended in [Demšar, 2006].

Porównywanie wielu klasyfikatorów na wielu zbiorach danych

Most Popular Approach: "win-loss-tie" Matrix

Example [Dietterich, 1999]

	C4.5	AdaBoost C4.5	Bagged C4.5
Randomized C4.5	14 – 0 – 19	1 – 7 – 25	6 – 3 – 24
AdaBoost C4.5	11 – 0 – 22	1 – 8 – 24	
Bagged C4.5	17 – 0 – 6		

Often only *significant* win/losses counted. Significance usually checked with t-test.

Not wrong but does not lead to a statistical hypothesis testing.

Friedman Test [Demšar, 2006]

Procedure (N classifiers, m files)

- H_0 : All classifiers perform equally well.
  H_1 : Some of the classifiers perform better then the others.
- For each dataset, rank all classifiers according to their results; calculate average ranks $\bar{r}_j$, $j = 1, \ldots, m$:

datasets	classifier 1	classifier 2	classifier 3
dataset 1	1	3	2
dataset 2	1.5	1.5	3
dataset 3	1	2	3
dataset 4	2	3	1
dataset 5	2.5	2.5	1
average	1.6	2.4	2.0

Friedman Test [Demšar, 2006]

Procedure (N classifiers, m files) – continued

- Using average ranks calculate Friedman's statistics:

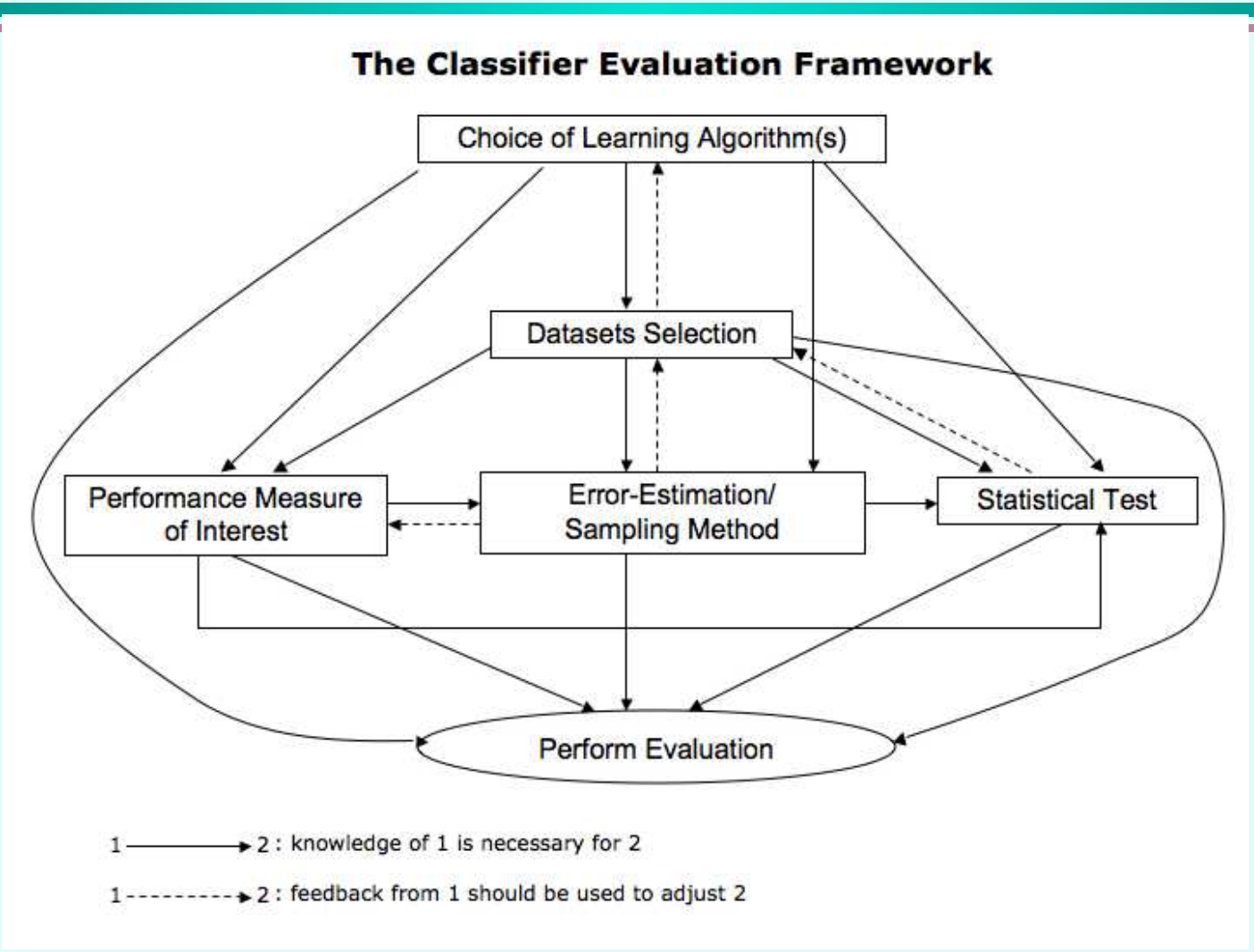
$$\chi_F^2 = \frac{12m}{N(N+1)} \left(\sum_{j=1}^{m} \bar{r}_j^2 - \frac{N(N+1)^2}{4} \right)$$

  which is distributed approx. as χ^2 with $N-1$ df.
- Reject null hypothesis if $\chi_F^2 > \chi_{crit}^2$ and proceed to a post-hoc analysis.
- Post-hoc analysis (Nemenyi test): calculate critical difference:

$$CD = q_\alpha \sqrt{\frac{N(N+1)}{6m}}$$

  where q_α is the critical value (based on α and N).
- All algorithms with the differences in average ranks greater than CD are significantly different.

Podsumowanie oceny

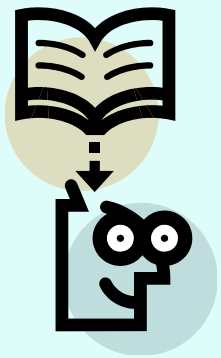


Za N.Japkowicz, Mshah: ICMLA Tutorial

Podejścia teoretyczne

- **Obliczeniowa teoria uczenia się (COLT)**
  - PAC model (Valiant)
  - Wymiar Vapnik Chervonenkis → VC Dimension
- Pytania o ogólne prawa dotyczące procesu uczenia się klas pewnych funkcji z przykładów - rozkładów prawdopodobieństwa.
- Silne założenia i ograniczone odniesienia do problemów praktycznych (choć SVM?).

I to by było na tyle …



Nie zadawalaj się tym
co usłyszałeś – poszukuj więcej!
Czytaj książki oraz samodzielnie
badaj problemy ML!