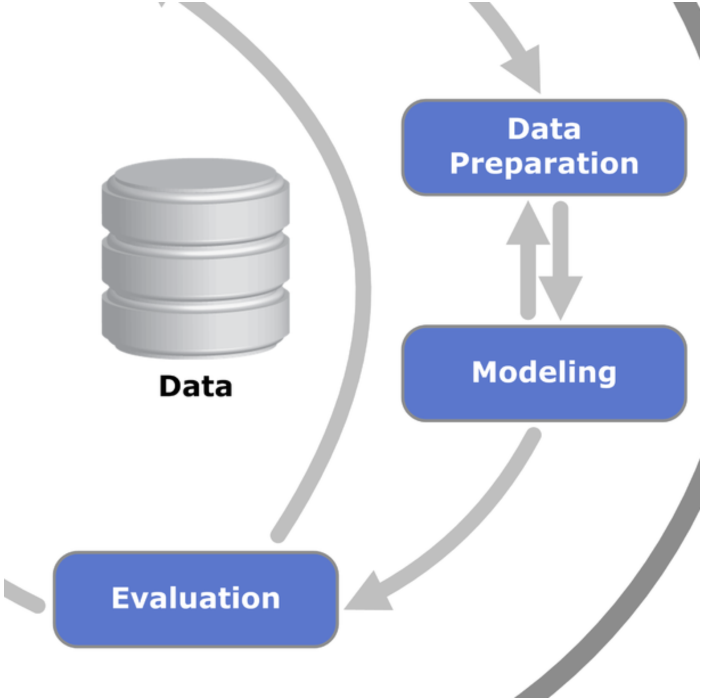
PROFINIT

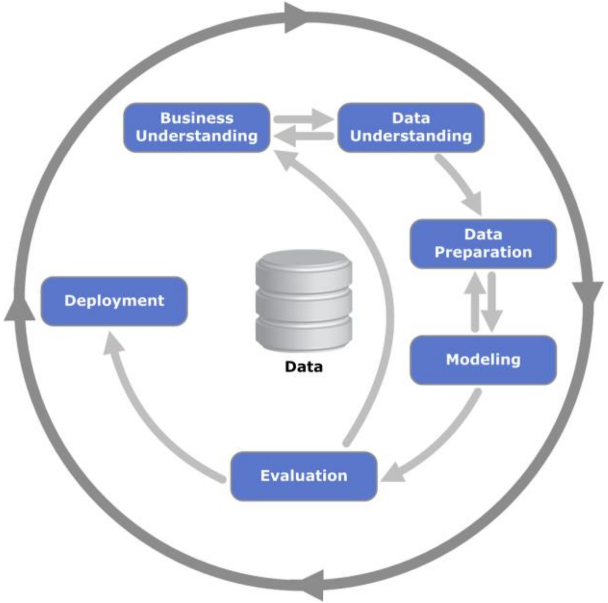
NDBI048 – Data Science Modelling 2: Model selection

Jan Hučín

22. 11. 2023

Where we are now





Outline

1. complex evaluation metrics
2. criteria for model
3. feature selection
4. model methods

Model evaluation: metrics (binary target)

PROFINIT

id	predicted probabililty
1	0.34
2	0.76
3	0.04
4	0.29
5	0.48
...	...

Model evaluation: metrics (binary target)

PROFINIT

id	predicted probabililty	predicted (thresh 0.5)	predicted (thresh 0.3)	actual target
1	0.34	0	1	1
2	0.76	1	1	1
3	0.04	0	0	0
4	0.29	0	0	0
5	0.48	0	1	0
...	...			...

different threshold → different recall, FPR etc.

Model evaluation: metrics (binary target)

PROFINIT

	predicted true	predicted false	
actual true	TP	FN	P
actual false	FP	TN	N
	$\hat{P}$	$\hat{N}$	S

confusion matrix

- give a threshold for pos/neg prediction
- similar to hypothesis testing (error type I, II)
- **recall** (true positive rate) = $\frac{TP}{TP+FN}$
- **sensitivity** = recall
- **precision** = $\frac{TP}{TP+FP}$
- **specificity** (true negative rate) = $\frac{TN}{TN+FP}$
- **false positive rate** = $\frac{FP}{TN+FP}$, **false negative rate** = $\frac{FN}{TP+FN}$
- **accuracy** = $\frac{TP+TN}{TP+FN+FP+TN}$

Model evaluation: metrics (binary target)

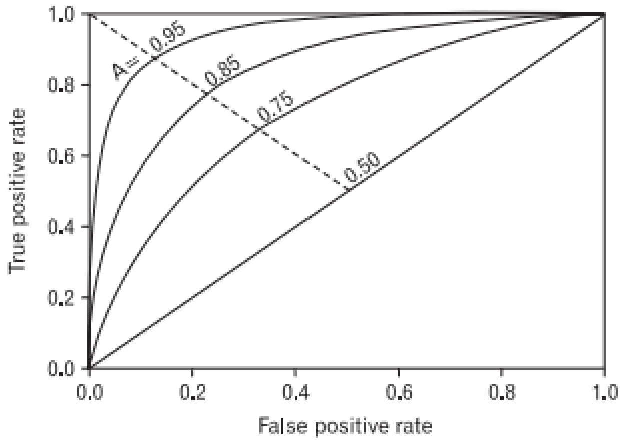
PROFINIT

Confusion matrix depends on the threshold value:

- small threshold → high recall, but high FPR too
- and vice versa

→ receiver operation curve (**ROC**)

- threshold runs 0→1
- for various thresholds, we count TPR & FPR
- we make curve of points [FPR; TPR]
- random guessing – diagonal
- perfect model – through top left
- performance: area under curve – **AUC**



Model evaluation: metrics (binary target)

PROFINIT

	predicted true	predicted false	
actual true	TP	FN	P
actual false	FP	TN	N
	$\hat{P}$	$\hat{N}$	S

Gain

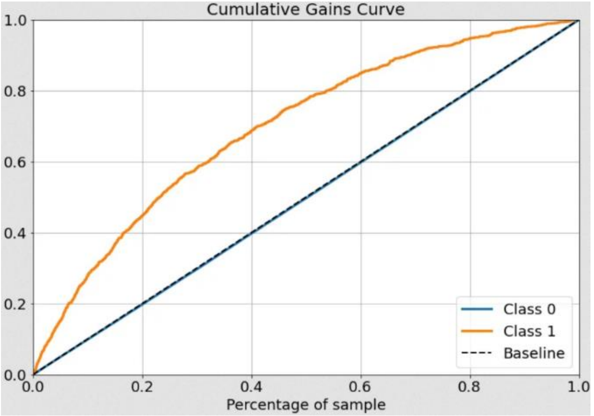
› recall in sample by model vs. recall in random sample

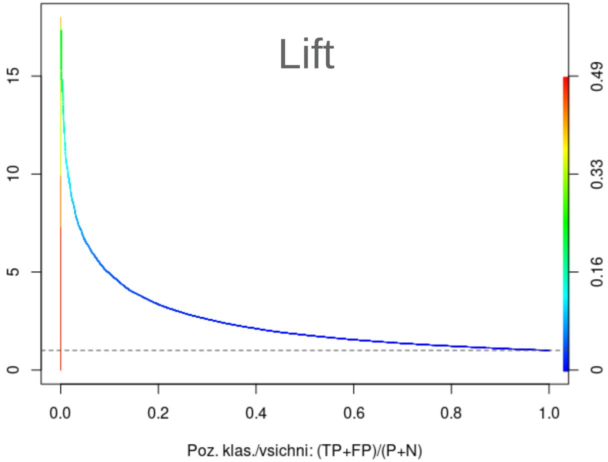
› (TP / P) vs. ($\hat{P}$ / S)

Lift

› precision in sample by model over precision in random sample

› (TP / $\hat{P}$) : (P / S) = (TP / $\hat{P}$) : (P / S)

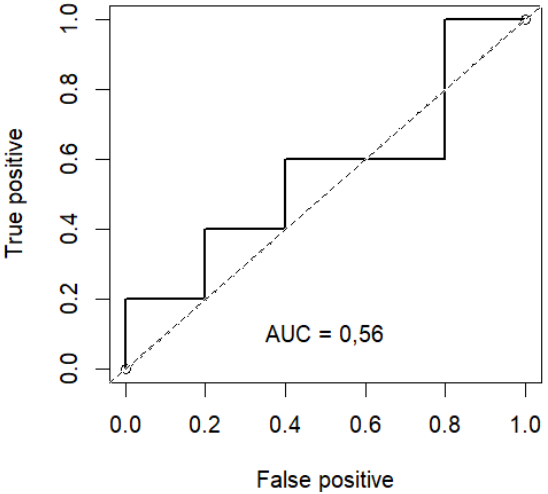




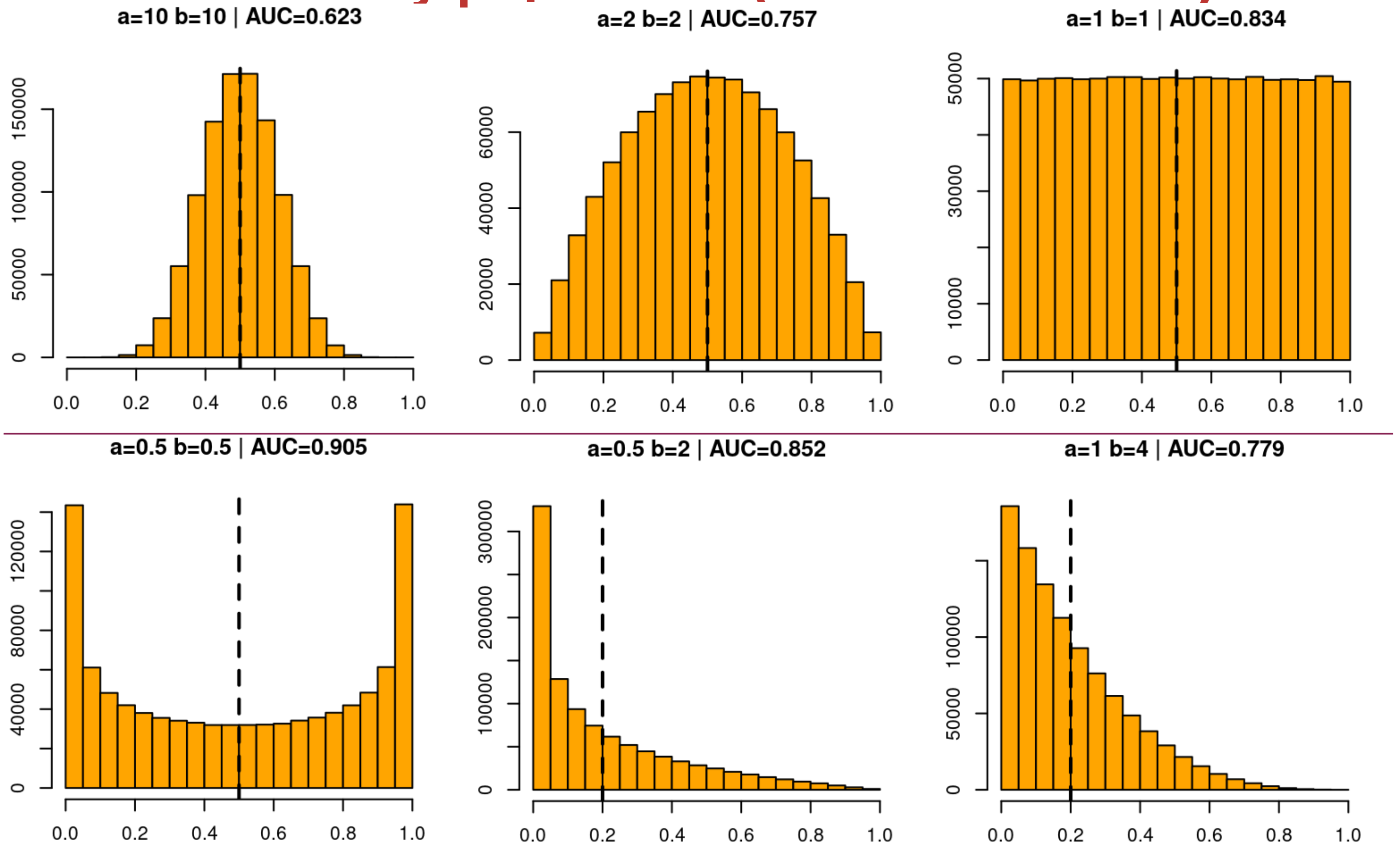
Metric limits

0,41	0,43	0,45	0,47	0,49	0,51	0,53	0,55	0,57	0,59	P (exact)
0	0	0	0	0	1	1	1	1	1	prediction
0	1	1	0	0	1	0	1	0	1	result

- exact probabilities but low performance
- why?
- **classification**: exact classification possible
- **prediction**: exact prediction impossible – due to randomness

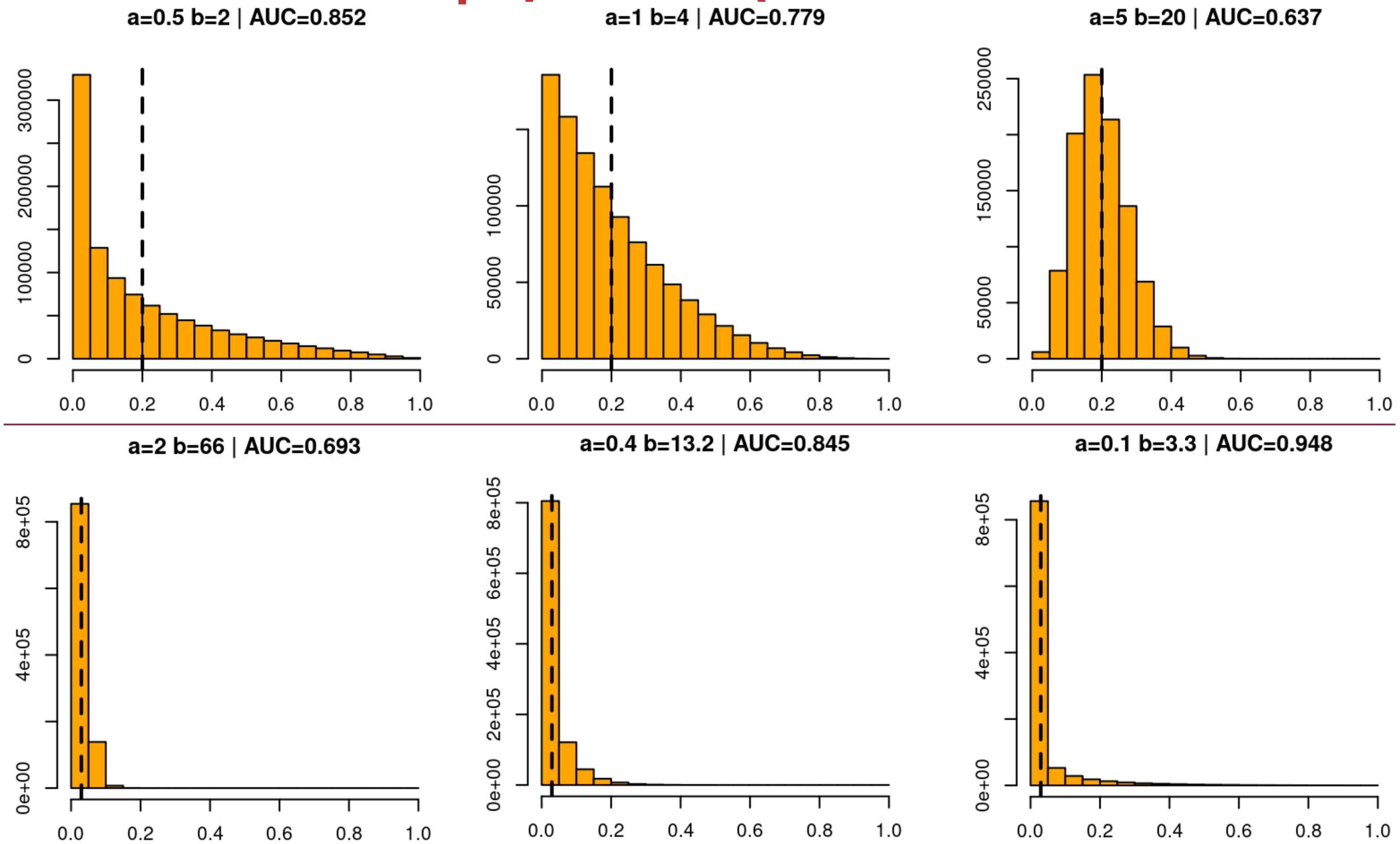


Metric limits by population (beta distribution)



Metric limits by population (beta distribution)

PROFINIT



Model requirements

- meeting customer requirements
- high performance
- fast
- cheap
- interpretable
- easy to implement and maintain

Requirements for the model

PROFINIT

- requested mode (real-time, near real-time, batch) → **SLA**
- how much data to process for a result?
- can I / need I have something precomputed?
- is partial or approximate result allowed?
- technologies (SQL, Big Data, R/Python/C/Java)

Implementation requirements

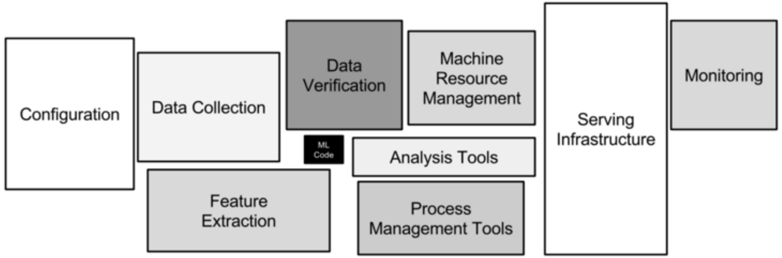
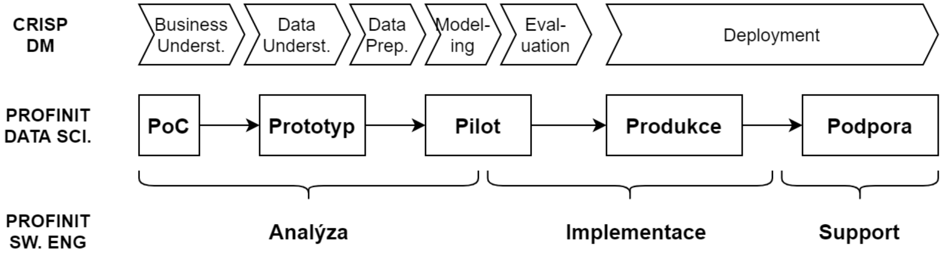


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.



- technologies
- knowledge
- connection to the world
- maintenance
- → price

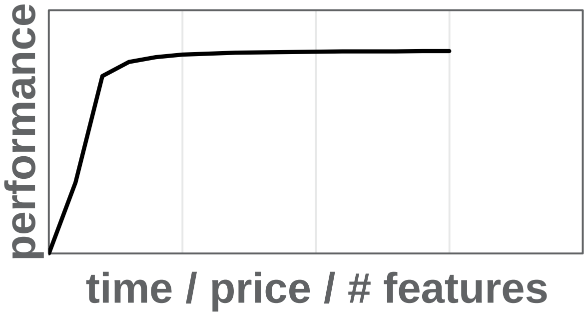
Model building

PROFINIT

- no or random model
- simple model
- basic / referential model
- final model

Model building

PROFINIT



simple model

- domain knowledge
- DIY

basic / referential model

- strong and easily available features
- simple method (regression, small tree)
- sometimes sufficient

final model

- long journey, good to automate (MLops)

Model selection – features

PROFINIT

Forward

- start from null model (intercept only)
- try a predictor & evaluate performance
- choose the one with the highest added performance, add it
- repeat until there is no performance increase

Backward

- start from full model (all predictors)
- omit a predictor & test (p-value, ML metrics)
- choose the one with highest p-value or added performance, drop it
- repeat until the performance gets worse

Model selection – forward or backward?

PROFINIT

forward

- in early steps, for referential model building
- good for simple and interpretable methods

backward

- exploration of a new feature family
- estimation of performance limit
- requires huge sources, regularization, automatized process
- good for a complex methods

Model selection – method

PROFINIT

referential model – preferably simple, interpretable

final model:

- by customer constraints (e. g. interpretable methods only)
- by technical limits
- by performance : price ratio
- by maintenance requirements **(bus factor)**

Modeling methods – linear model

PROFINIT

X = predictor matrix, Y = target, β – coefficients (parameters, effects)

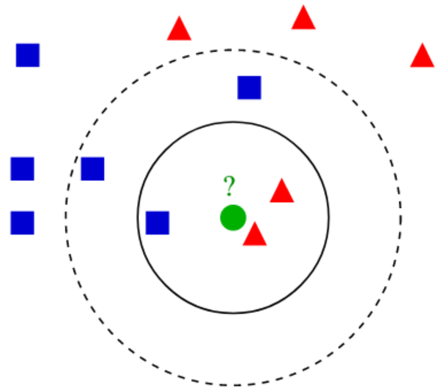
- E$\boldsymbol{Y} = \boldsymbol{X\beta}$ **linear regression**
- P($\boldsymbol{Y}$ = 1) = $\frac{e^{X\beta}}{1+e^{X\beta}}$ **logistic regression**
- „scoring model“: $\widehat{Y}_i = f\left(\sum_{j=1}^{k} \beta_j X_{ij}\right)$ – additive effects

Modeling methods – nearest neighbors

PROFINIT

Similar units will have **similar target**.

1. Train set: units with known target (labeled).
2. New unit (unknown target) arrives.
3. By some distance metric, we found *k* nearest units from the train set (nearest neighbors).
4. Estimated target = aggregation of neighbors‘ targets.



Distance metric: e. g. euclidean, cosine, Levenshtein…

Aggregation: voting, weighted mean, median

Modeling methods – Bayes classifier

PROFINIT

Conditional probability $P(A|B) = P(A \cap B)/P(B)$

- probability of event A in case we know B is true
- *probability of raining given the fact, we are on Sahara*

Bayes theorem

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)}$$

› $P(H|E)$ – probability of hypothesis H given observation / evidence E

› $P(E|H)$ – probability of observing E given H aka likelihood of H given E

› $P(H)$ – prior probability of hypothesis H

› $P(E)$ – overall probability of observing evidence E

Modeling methods – Bayes classifier

PROFINIT

Bayes classifier

$$P(Y = C_i | \boldsymbol{X} = \boldsymbol{x}) = \frac{P(\boldsymbol{X} = \boldsymbol{x} | Y = C_i) \cdot P(Y = C_i)}{P(\boldsymbol{X} = \boldsymbol{x})}$$

- Y = target, C_i = category, $\boldsymbol{X}$ = predictors, $\boldsymbol{x}$ = observed values
- find i where $P(\boldsymbol{X} = \boldsymbol{x} | Y = C_i) \cdot P(Y = C_i)$ biggest → classification
- for binary target:

$$\frac{P(Y = 1|E)}{P(Y = 0|E)} = \frac{\frac{P(E|Y = 1) \cdot P(Y = 1)}{P(E)}}{\frac{P(E|Y = 0) \cdot P(Y = 0)}{P(E)}} = \frac{P(Y = 1)}{P(Y = 0)} \cdot \frac{P(E|Y = 1)}{P(E|Y = 0)}$$

Suppose you live in Scotland (rainy 80% of days). What are the odds of being sunny tomorrow if weather forecast (accurate 2/3 of time) say so?

Modeling methods – naive Bayes classifier

PROFINIT

„naive“ assumption: all predictors are independent

i. e. $P(\boldsymbol{X} = \boldsymbol{x}) = P(X_1 = x_1) \cdot P(X_2 = x_2) \cdot \ldots \cdot P(X_k = x_k)$

$$P(Y = C_i | \boldsymbol{X} = \boldsymbol{x}) = \frac{\prod_j P\left(X_j = x_j | Y = C_i\right) \cdot P(Y = C_i)}{P(\boldsymbol{X} = \boldsymbol{x})}$$

1. From the train set, compute $P\left(X_j = x | Y = C_i\right)$ for all *i*, *j* and *x*.
2. Give prior probabilities for categories $P(Y = C_i)$.
3. For new unit, compute numerator for each *i* and take maximazing.

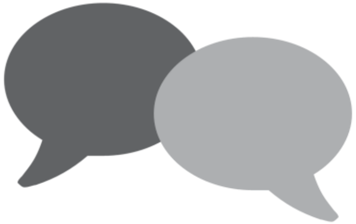
Model selection – business view

PROFINIT

- quantitative change: beware of complexity ($O(N^2)$, $O(N^3)$, …)
- qualitative change: usually risky
  - technology / version change
  - workflow change
  - data format change
  - new result requirements
  - → **should be robust**
- stable (champion) vs. candidate (challenger) model
- automatic monitoring

Don‘t change a winning team.
English proverb

PROFINIT



Questions?