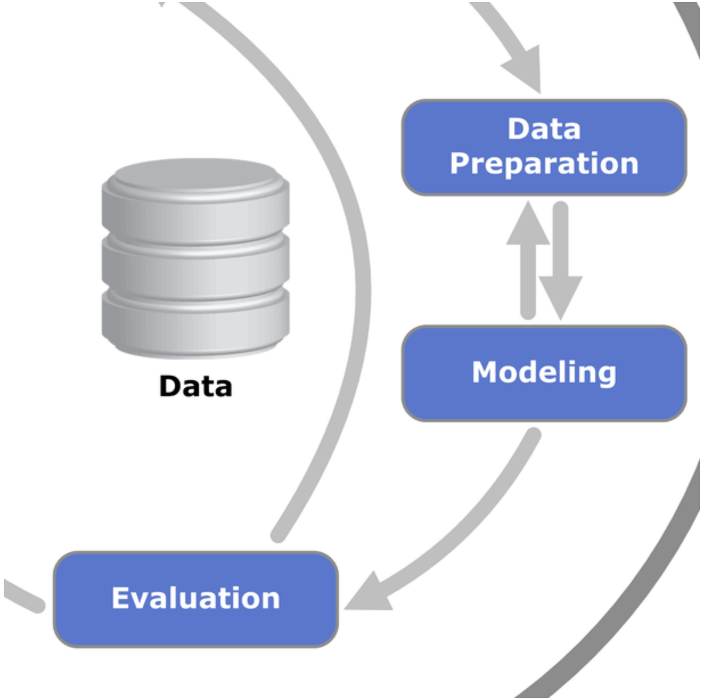
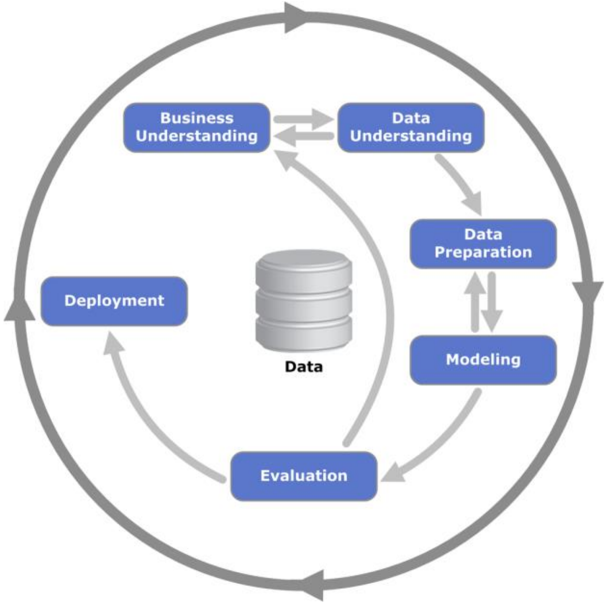
NDBI048 – Data Science
Modelling 1: Basics & Linear Models

Jan Hučín

18. 11. 2021

Where we are now





Outline

1. analytical and modelling approach
2. aim of modelling, basic terms
3. types of models
4. data for modelling: train, test, validation
5. basics of linear modelling
6. model evaluation
7. model regularization
8. model with interactions

Analytical (inferential) approach

PROFINIT

unit (human, animal, picture, action, proces, ...)

- I have **data** about it
- I have data about other features
  - day, time, salary, body height etc.

I want to: understand the world & make conclusions.

Dependent Variable · Population Y intercept · Population Slope Coefficient · Independent Variable · Random Error term

$$Y_i = \underbrace{\beta_0 + \beta_1 X_i}_{\text{Linear component}} + \underbrace{\varepsilon_i}_{\text{Random Error component}}$$

Modelling approach

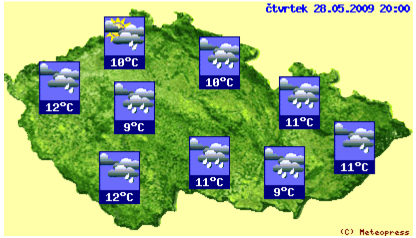
PROFINIT

unit (human, animal, picture, action, proces, ...)

- has (or will have) a **feature I don‘t know now**
  - Man, or woman? Fair, or deceit? Age? How many ºC? Which of kinds?

- but I know something else
  - living place; history; behaviour; body measures etc.

I want to: estimate / classify / predict the unit‘s unknown feature.



Get the difference

PROFINIT

Analytical approach

- I have data.
- Trying to describe the world.
- **Fitting relations in data.**
- outcome = explanation (inference).

Modelling approach

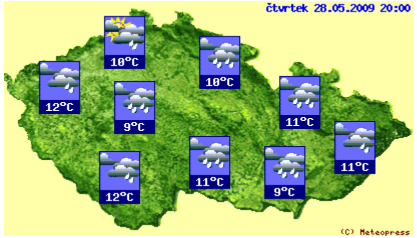
- I have a problem.
- Trying to get any info.
- **Fitting relations in data.**
- outcome = prediction, classification etc.

Same methods, same mathematics, different aims.

Basic terms

PROFINIT

- „unknown feature“ = **target, response**
- „what I know“ = **feature, predictor, explanatory var.**
- „result“ = **prediction, classification** etc. (see later)
- „how we get the result“ = **modelling method**
- „data for modelling“ = **dataset, model matrix**



Types of models

PROFINIT

- By info about target (data labeling)
  - yes, for enough & representative units → **supervised model**
  - yes, but for few or non-representative units → **semi-supervised model**
  - no → **unsupervised model**
- By target type
  - binary
  - categorical (ordinal, non-ordinal)
  - numerical
- Used method?
  - linear (regression etc.)
  - rule-based (decision trees etc.)
  - similarity (kNN etc.)
  - „blackbox“ (neural networks, gradient boosting etc.)

From now: **supervised** and mostly **binary** models.

Data and dataset

PROFINIT

- Data need to be **understood** and **prepared**.
- **Training dataset** = table (matrix):
  - columns = id, target(s), features
  - rows = units
- Dataset division:
  - **train** set – where we **fit** a model
  - **test** set – where we **evaluate** a model
- **Validation dataset**
  - where we **prove** the model is good
- see later

Image Id	No. of Exudates	Area of Largest Span	Largest Spot		yellowness
			Major Axis	Minor Axis	
Image001	12	1061	45.97	29.82	0.55
Image002	11	413	25.92	20.60	0.44
Image003	19	880	44.03	25.96	0.62
Image004	10	530	28.57	24.51	0.15
Image005	4	536	28.23	24.96	0.44
Image006	13	338	25.80	17.29	0.50
Image007	18	14809	169.96	113.87	0.59
Image008	8	5997	152.68	53.95	0.59
Image009	9	1967	64.20	40.08	0.16
Image010	10	4161	99.38	54.83	0.62
Image011	28	4023	86.23	61.83	0.64
Image012	21	630	53.66	15.68	0.62
Image013	8	1748	57.13	39.38	0.67
Image014	15	1079	62.78	22.55	0.53
Image015	12	383	27.64	18.43	0.54
Image016	20	1175	53.81	28.20	0.57
Image017	10	694	36.29	25.36	0.61
Image018	24	1601	71.12	29.01	0.64
Image019	4	535	28.47	24.81	0.61
Image020	11	626	35.54	23.97	0.57

Remind: linear regression

PROFINIT

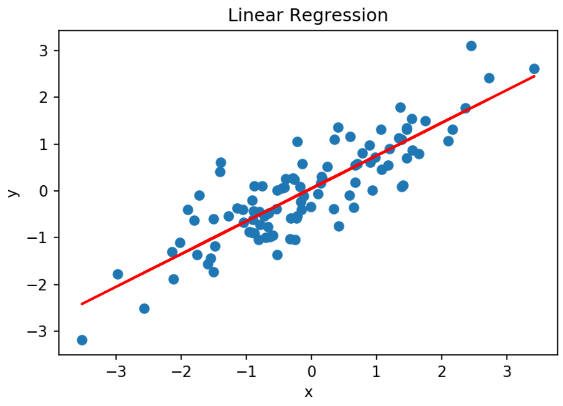
Given matrix $\boldsymbol{X}$ (n × k).

Random vector $\boldsymbol{Y}$ fits **linear regression** if vector $\boldsymbol{\beta}$ exists, so that:

$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} \sim N(0, \sigma^2\boldsymbol{I})$

Example:

$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ pairs of numbers



looking for best fit $y_i = \beta_1 x_i + \beta_0$

least-squares method:

minimize $\sum_{i=1}^{n}(y_i - \beta_1 x_i - \beta_0)^2$ wrt. β_0 and β_1

Linear regression and two approaches

PROFINIT

analytical approach:

› interested in trend (inferention)

› *„How does the world work?“*

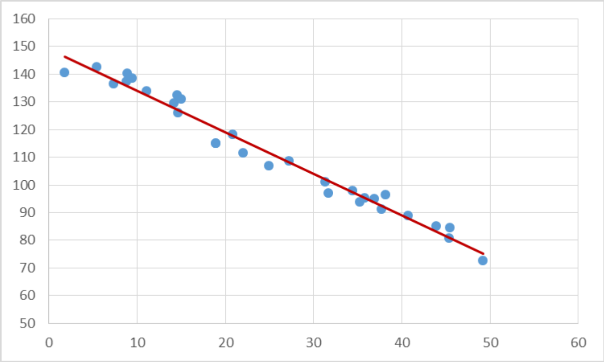
› getting β is a goal

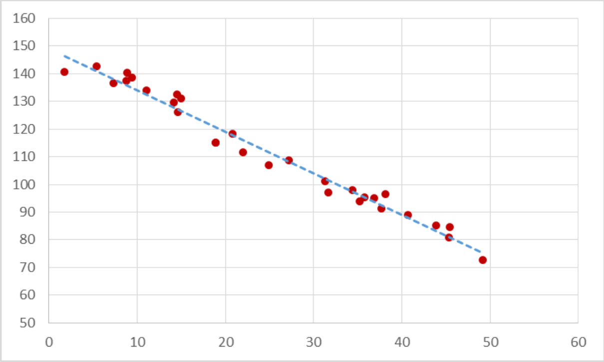
modelling approach:

› interested in points (estimate, prediction)

› *„If I have this value of x, how many will be y?“*

› getting β is a mean



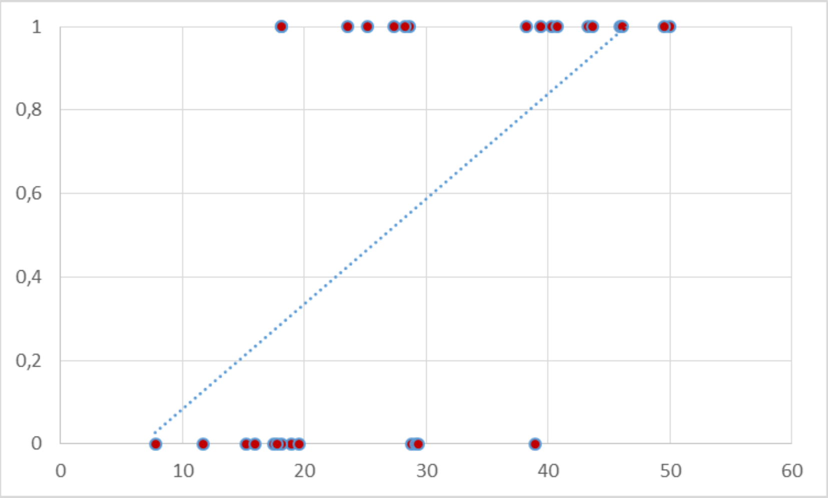


Linear regression with binary response

PROFINIT

- fitting a line has no sense
- but we feel: lower *x* has less positive responses than higher *x*
- how to express it?

→ generalization of linear regression



General linear model

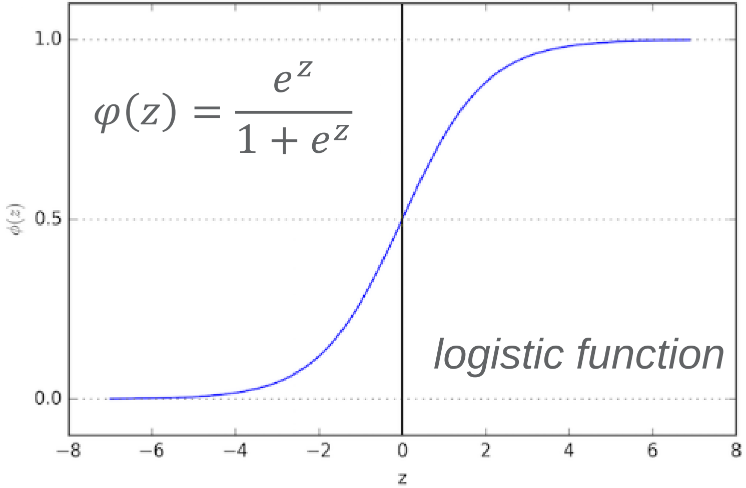
PROFINIT

$g(\mathrm{E}\boldsymbol{Y}) = \boldsymbol{X\beta}$ $\mathrm{E}\boldsymbol{Y} = g^{-1}(\boldsymbol{X\beta})$

- X – predictor matrix
- β – coefficients (parameters, effects)
- g – *link* function
  - identity: $g(t) = t$ $g^{-1}(z) = z$
  - logit: $g(t) = \ln\frac{t}{1-t}$ $g^{-1}(z) = \frac{e^z}{1+e^z}$
  - logarithm: $g(t) = \ln t$ $g^{-1}(z) = e^z$
  - ...
- error distribution: gaussian, binomial, poisson, ...
- „scoring model“: $\widehat{Y}_i = g^{-1}\left(\sum_{j=1}^{k}\beta_j X_{ij}\right)$ – **additive effects**

Logistic regression

PROFINIT



- for binary targets: score ⟶ probability
- E***Y*** **=** g^{-1}(***Xβ***); g^{-1}: ***R*** ⟶ (0; 1)
- g^{-1} is **logistic** function
- $g(t) = \ln\frac{t}{1-t}$ (**logit** link function)

$$\ln\frac{EY}{1-EY} = \boldsymbol{X}\beta$$

$$\frac{p}{1-p} = e^{X\beta}$$

Interpretation of β:

$$\frac{p}{1-p} = e^{\beta_0+\beta_1 x}$$

+1 increment in x ⟶ odd ratio increases exp(β_1) times

Logistic function and logits – summary

PROFINIT

p = $\frac{e^z}{1+e^z}$, $z \in R$ (logistic function)

z = $ln\left(\frac{p}{1-p}\right)$, $p \in (0; 1)$ (logit function)

logit = logarithm of odds ratio

p = 0,5 → odds 1 : 1 → logit = 0

+1 logit → odds change *e*-times

Probability axes

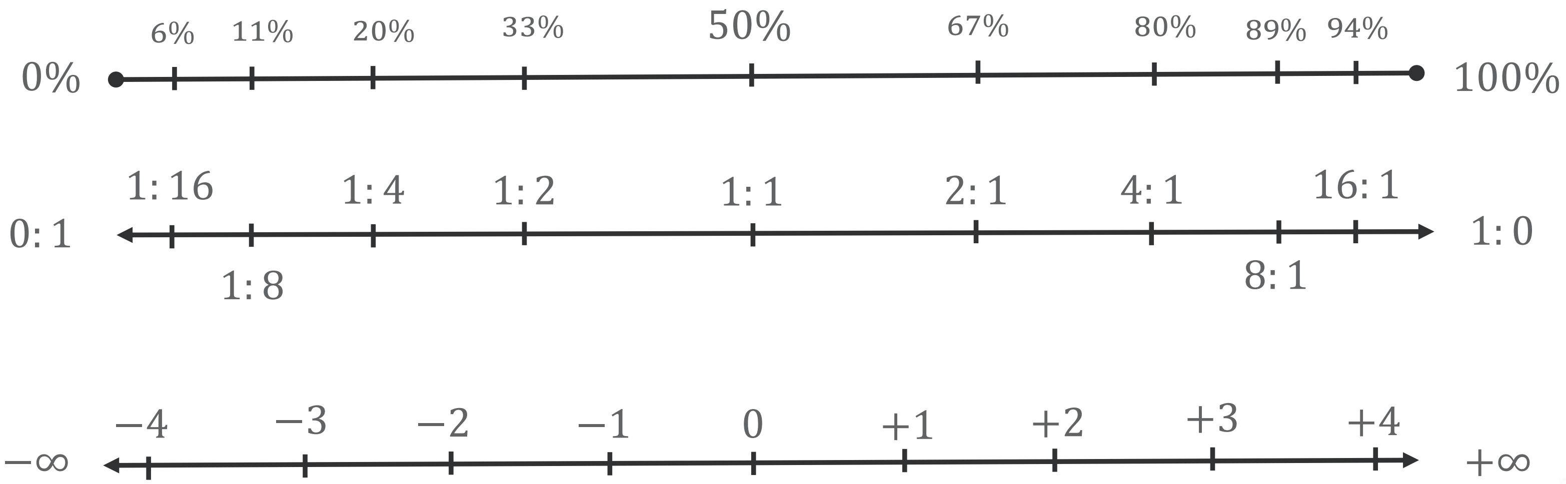
› For a binary event A

probability axis

$$P_A = \frac{N_A}{N_A + N_{\neg A}}$$

odds axis

$$O_A = N_A : N_{\neg A}$$



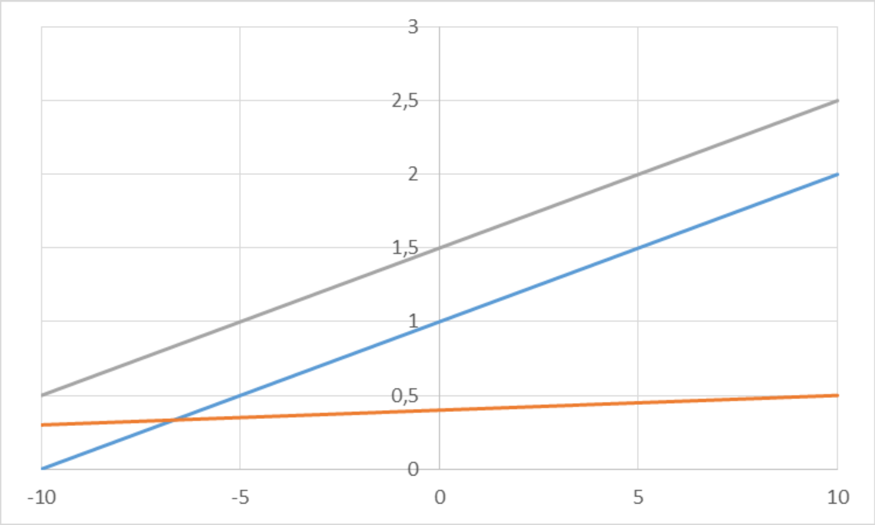
log odds axis

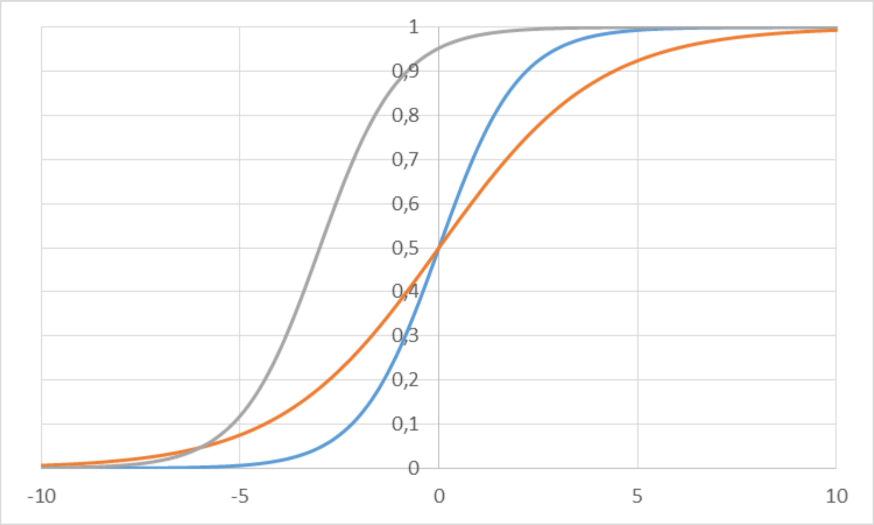
$$L_A = \log_2\left(\frac{N_A}{N_{\neg A}}\right)$$

33%	1:2	−1 bit				67%	2:1	+1 bit
20%	1:4	−2 bit	50%	1:1	0 bit	80%	4:1	+2 bit
11%	1:8	−3 bit				89%	8:1	+3 bit

Logistic and linear regression – parameters

PROFINIT





$$y = \beta_1 x + \beta_0$$

β got by least-squares

$$y = \frac{e^{(\beta_1 x + \beta_0)}}{1 + e^{(\beta_1 x + \beta_0)}}$$

β got by MLE

Remind: MLE (maximum likelihood estimate)

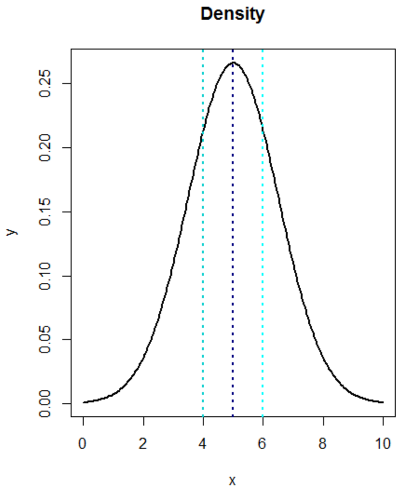
PROFINIT

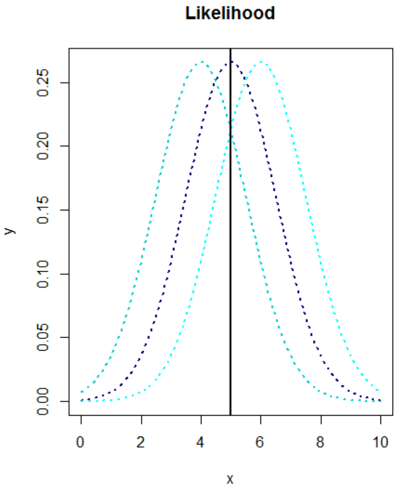
probability density: f(x, μ), μ fixed

- what *x* value do I expect most of all?

likelihood function: L(μ | x) = f(x, μ), x fixed (observed)

- what μ gives best fit?
- maximization L for μ

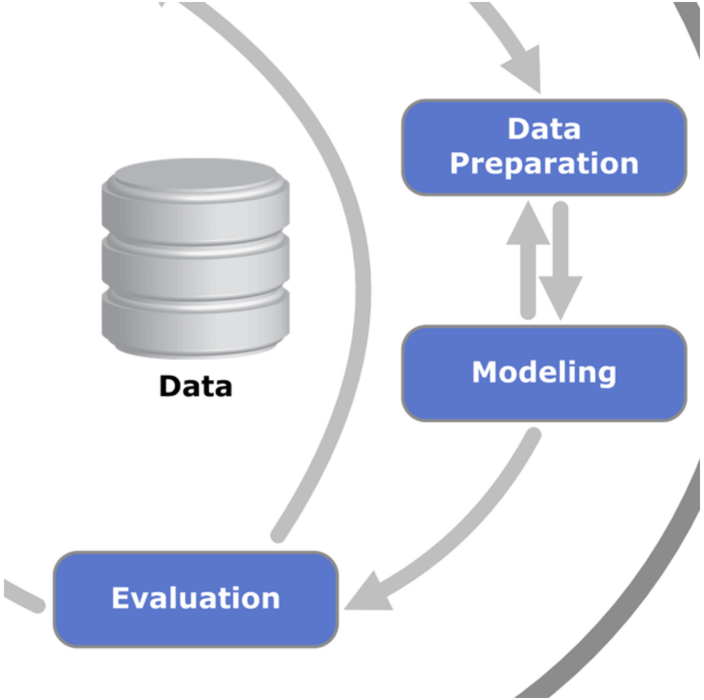


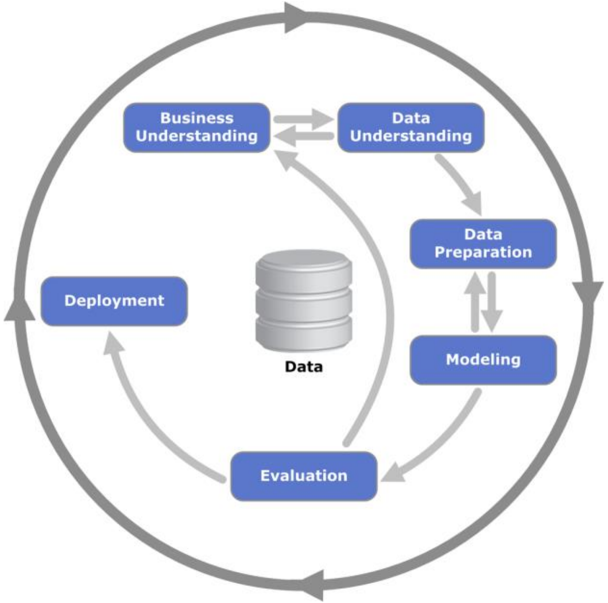


Example of fitting model

see Jupyter notebook

Model needs evaluation





Model evaluation

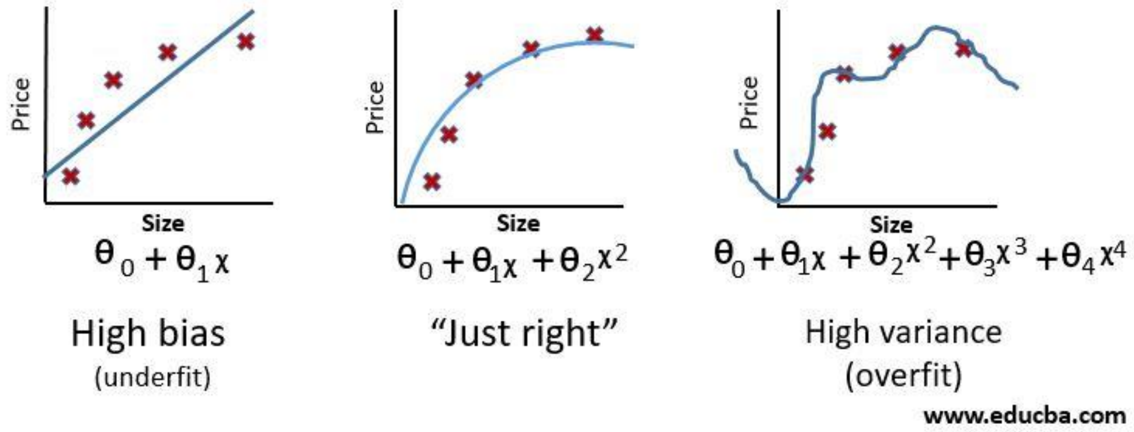
PROFINIT

~~How well does my model fit my past data?~~

How well would I predict (classify, estimate) in reality?

› Can‘t evaluate on the same data as for fit → **overfitting**

› I need to „simulate unknown reality“→ **cross-validation**



Cross-validation

PROFINIT

- Take 1/*k* of data as the **test** set, the rest as the **train** set.
- Fit on train set, predict on the test set.
- Repeat for each 1/*k* of data.
- Now we have predicted values for all units. Compare with actual target values.
- *Variation: compare for each 1/k separately and aggregate metrics.*

Iteration 1	Test	Train	Train	Train	Train
Iteration 2	Train	Test	Train	Train	Train
Iteration 3	Train	Train	Test	Train	Train
Iteration 4	Train	Train	Train	Test	Train
Iteration 5	Train	Train	Train	Train	Test

Model evaluation: metrics (binary target)

PROFINIT

- logLik
- Brier score
- metrics on confusion matrix
- ROC & AUC
- Lift

id	predicted	actual target
1	0.34	1
2	0.76	0
3	0.04	0
4	0.29	0
5	0.88	1
...	...	...

Model evaluation: metrics (binary target)

PROFINIT

logLik

- log(L)
- **AIC** = −2 log(L) + 2·(# of params)

only for math purpose, uninterpretable

Brier score

- $2 \cdot \sum(\text{actual} - \text{predicted})^2$
- similar to SSE
- good for comparison, bad for interpretation

id	predicted	actual target
1	0.34	1
2	0.76	0
3	0.04	0
4	0.29	0
5	0.88	1
...	...	...

Model evaluation: metrics (binary target)

PROFINIT

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** = $\frac{TP+TN}{TP+FN+FP+TN}$

	predicted true	**predicted false**
actual true	TP	FN
actual false	FP	TN

Model evaluation: metrics (binary target)

PROFINIT

confusion matrix in one number

- Phi coefficient
  (also Matthews corr. coef., *MCC*)
- $\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FN)(FP+TN)(TP+FP)(FN+TN)}}$

	predicted true	predicted false
actual true	TP	FN
actual false	FP	TN

Model evaluation: confusion matrix, example

PROFINIT

- **recall** (true positive rate), **sensitivity** =
  $= \frac{TP}{TP+FN} = 0.8$
- **precision** $= \frac{TP}{TP+FP} = 0.5$
- **specificity** (true negative rate) =
- $= \frac{TN}{TN+FP} = \frac{11}{15} \sim 0.73$
- **false positive rate** $= \frac{FP}{TN+FP} = \frac{4}{15} \sim 0.27$
- **false negative rate** $= \frac{FN}{TP+FN} = 0.2$
- **accuracy** $= \frac{TP+TN}{TP+FN+FP+TN} = 0.75$
- **Phi** = 0.48

	pred true	pred false	total
actual true	40	10	**50**
actual false	40	110	**150**
total	**80**	**120**	**200**

Model evaluation: metrics (binary target)

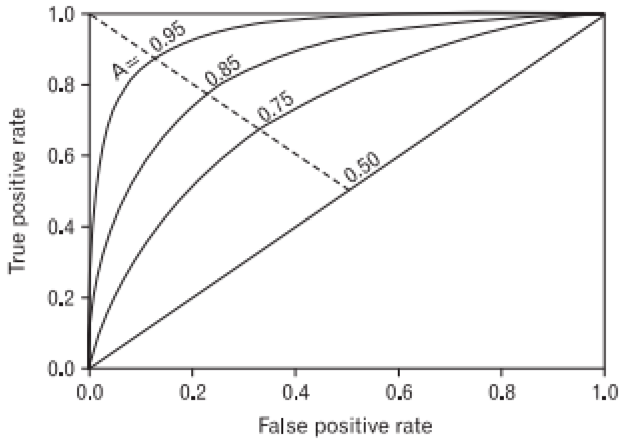
PROFINIT

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**

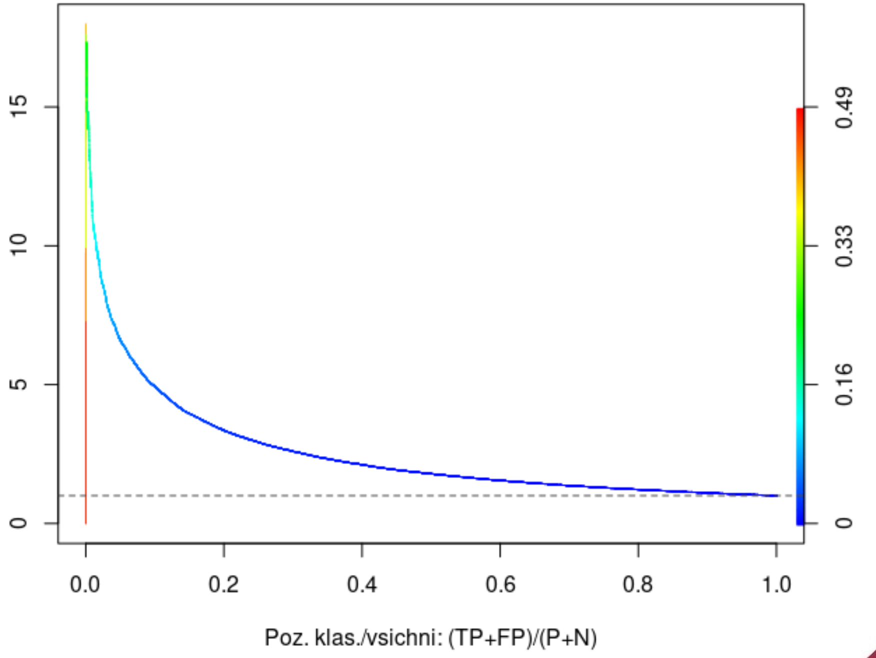


Model evaluation: metrics (binary target)

PROFINIT

Lift

- precision / overall target rate
- Take positive predicted:
  how many times more often
  we hit target
  than by random guessing?
- Lift chart: threshold runs 0→1



Model selection

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 gain

Backward

- start from full model (all predictors)
- omit a predictor & test (p-value, ANOVA; but ML metrics possible, too)
- choose the one with highest p-value or added performance, drop it
- repeat until the performance gets lower || p-values > 0.03

Model regularization

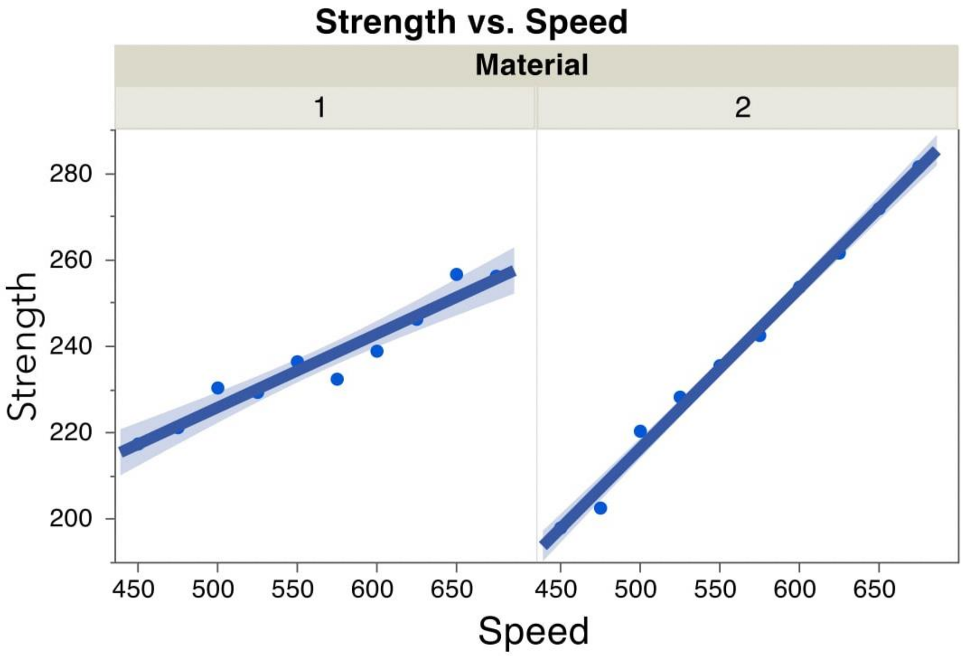
PROFINIT

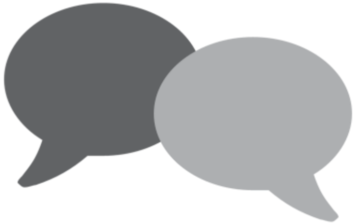
- When some predictors highly correlated – computation numerically unstable.
- Solution: prefer lower values of coefficients – penalization
- Methods: Lasso, L2 (ridge regression)

Model with interactions

PROFINIT

- effects values not independent
- → effects not additive
- model must be fitted with interactions





Questions?