GRUNDLAGEN EXPERIMENTELLEN DESIGNS IM BEREICH MACHINE LEARNING

Alexander Schindler
Scientist
Information Management
Center For Digital Safety & Security

AIT Austrian Institute Of Technology Gmbh
Alexander.schindler@ait.ac.at

OUTLINE

Vormittagsblock (inkl. Kaffeepause)

10:00 – 10:30 Einführung in Künstliche Intelligenz
10:30 – 11:30 Grundlagen Artificial Neural Networks,
Arten von Artificial Neural Networks
11:30 - 12:30 Anwendungsbereiche (Labor)

Nachmittagsblock (inkl. Kaffeepause)

13:30 - 14:00 Grundlagen experimentellen Designs
im Bereich Machine Learning Teil I
14:00 - 14:30 Grundlagen experimentellen Designs
im Bereich Machine Learning Teil II
14:30 - 15:30 Anwendungsbereiche (Labor)
15:30 - 16:00 Definition von Zielen / Ableitung von Zielfunktionen
16:00 - 16:30 Clustering von Datenpunkten / Zeitreihen
16:30 - 17:00 Anwendungsbereiche (Labor)

Data Science:

To gain insights into data through computation, statistics, and visualization

DATA SCIENCE

- **What makes it a science?**

- A definition of **science**:
  "knowledge or a system of knowledge covering general truths or the operation of general laws especially as obtained and tested through scientific method"

- Definition of **scientific method**:
  "principles and procedures for the systematic pursuit of knowledge involving the recognition and formulation of a problem, the collection of data through observation and experiment, and the formulation and testing of hypotheses"

Merriam-Webster

EXPERIMENT

- Definitions of **experiment**:

  “an operation or procedure carried out under controlled conditions in order to discover an unknown effect or law, to test or establish a hypothesis, or to illustrate a known law”

  Merriam-Webster

  “a procedure carried out to support, refute, or validate a hypothesis. Experiments provide insight into cause-and-effect by demonstrating what outcome occurs when a particular factor is manipulated. Experiments vary greatly in goal and scale, but always rely on repeatable procedure and logical analysis of the results.”

  WIKIPEDIA
  The Free Encyclopedia

TYPES OF EXPERIMENTS

- **Controlled experiments** (“lab conditions”)
  Based on manipulation of experimental (independent) variables and control (or measurement) of other factors of experiment; outcome: dependent variable
  → hypothesis: prediction of effect of independent variable on a dependent variable
- **Natural experiments** (“quasi experiments”)
  mere observation of variables, no controlled manipulation; collection of evidence to test hypotheses; cf. economics, meteorology
- **Field experiments**
  observations in natural settings → possibly more validity; experimental conditions difficult to control; cf. social sciences

WHICH EXPERIMENTS IN DATA SCIENCE?

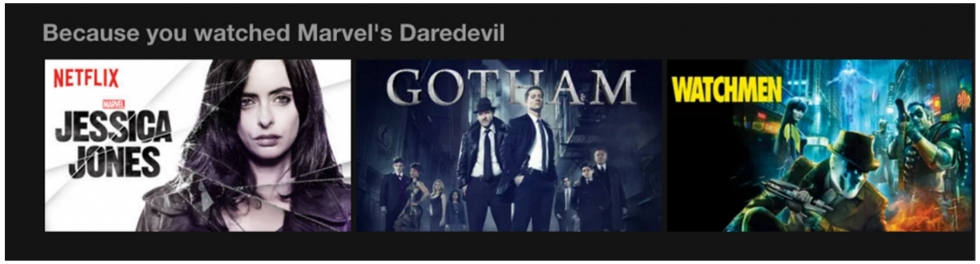
- Q: Which of these experiments are done in Data Science?
- A: **In the end controlled experiments**, but a bit of all…

Situation in practice:

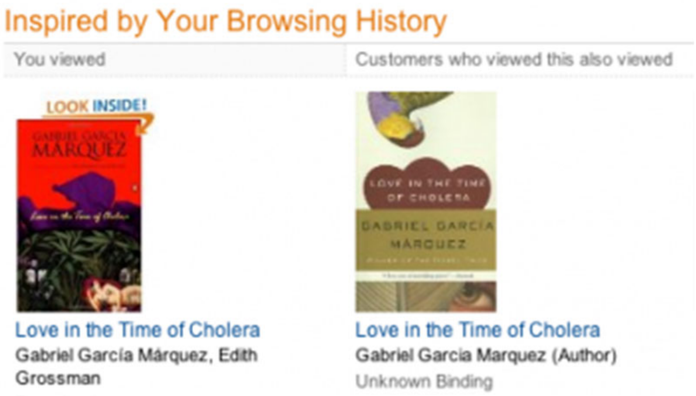
- Machine learning experiments → *controlled, repeatable*
- But also partly just controlled experiments on collected, observational data; collecting data to support a hypothesis
  → (not in a strict sense) “*quasi experiments”*
- We often do not collect specific data in order to test a hypothesis, but have to deal with data that happens to be available/automatically generated
  → (not in a strict sense) *“field experiments”*

EXAMPLE OF "FIELD EXPERIMENT" IN DATA SCIENCE

- Example: recommender system built from observed data
- E.g., which user watched which movie when / listened to which songs / bought which products / etc.



Because you watched Marvel's Daredevil



Inspired by Your Browsing History

You viewed

Love in the Time of Cholera
Gabriel García Márquez, Edith Grossman

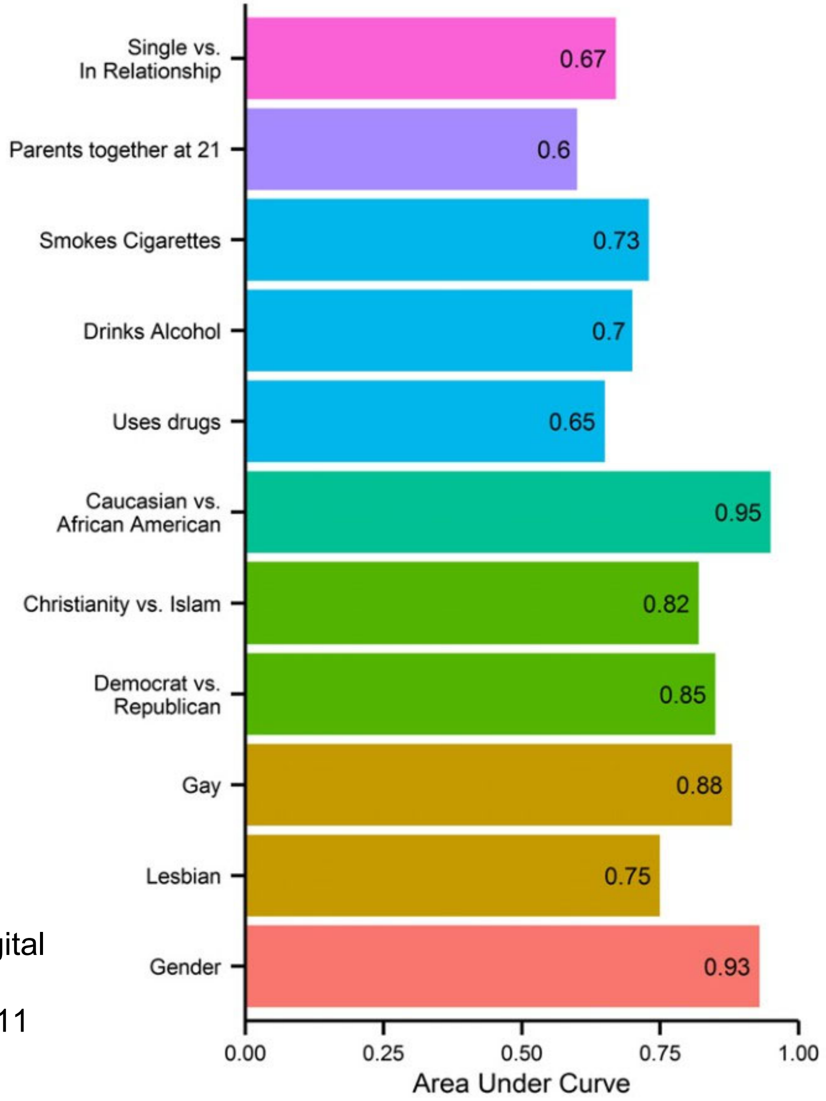
Customers who viewed this also viewed

Love in the Time of Cholera
Gabriel Garcia Marquez (Author)
Unknown Binding

- Record what people do "in the wild"
(time dependent, might vary when recorded again)
- "Snapshot" of measured data becomes "ground truth"
→ **repeatable and controlled experiments**
- In reality, these platforms are not just "observing" but experiment by controlling for factors and measure impact (A/B testing etc)

EXAMPLE OF “QUASI EXPERIMENT” IN DATA SCIENCE

- E.g., predicting personality from Facebook, Twitter, etc.
- Observed data: interactions, likes, network, content, …
- Hypothesis: personality and private traits predictable from this data
- To support hypothesis, needs additional data as ground truth
  → (personality) questionnaires



cf. Kosinksi et al., “Private traits and attributes are predictable from digital records of human behavior”, PNAS 110(15), 2013.
Golbeck et al, “Predicting Personality from Twitter”, SocialCom, 2011

EXAMPLE OF CONTROLLED EXPERIMENT

- E.g, Image classification
- Classic machine learning setup
  - Data points described by features (e.g. after feature extraction)
  - Target class (or value) for each data point
  - Machine learning algorithm to build a model that can predict target class or value from features
    (classification, regression, resp.)
- Possible hypotheses
  - Feature set X predicts targets better than feature set Y
  - Algorithm A predicts targets better than algorithm B

GT: horse cart
1: horse cart
2: minibus
3: oxcart
4: stretcher
5: half track

GT: birdhouse
1: birdhouse
2: sliding door
3: window screen
4: mailbox
5: pot

GT: forklift
1: forklift
2: garbage truck
3: tow truck
4: trailer truck
5: go-kart

GT: coucal
1: coucal
2: indigo bunting
3: lorikeet
4: walking stick
5: custard apple

GT: komondor
1: komondor
2: patio
3: llama
4: mobile home
5: Old English sheepdog

GT: yellow lady's slipper
1: yellow lady's slipper
2: slug
3: hen-of-the-woods
4: stinkhorn
5: coral fungus

GT: torch
1: stage
2: spotlight
3: torch
4: microphone
5: feather boa

GT: banjo
1: acoustic guitar
2: shoji
3: bow tie
4: cowboy hat
5: banjo

GT: go-kart
1: go-kart
2: crash helmet
3: racer
4: sports car
5: motor scooter

HYPOTHESIS AND CONTROL

- Hypothesis: testable (!) proposed explanation of a phenomenon not yet scientifically satisfactorily explained

→ *"how independent variable(s) affect dependent variable"*

- Needs to meet conditions of cause and effect:
  - presumed cause and presumed effect
  - cause must take place before the effect
  - rule out or take into account other (extraneous) variables
- Control: at least 2 different settings of independent variable to compare

HYPOTHESIS AND CONTROL

- E.g., testing the effects of a drug
- Hypothesis: Treatment with drug alleviates symptoms
  (=independent var) *(=dependent var)*
- Control *(= different settings of independent var)*
  - Group A (treatment group): patients receive drug
  - Group B (control group): patients receive placebo (no drug)

Control group

- accounts for extraneous variables:
  effects of procedure, suggestion, expectation, etc.
- allows to calculate effects of the extraneous variables
- allows to remove these effects from the treatment effect

- Testing: measuring (positive) effect on patients’ symptoms
  …comparing outcomes: health(A) > health(B)?

HYPOTHESIS AND CONTROL

- E.g., comparing two retrieval systems (search engines)
- Hypothesis: system X outperforms system Y
  (e.g. Google) (e.g. Bing)

- Independent var: system
- Dependent var: performance indicator

- Control: system X vs. system Y

- Testing:
  - System X retrieves more relevant documents than system Y
  - performance indicator(X) > performance indicator(Y)

CONTROLLED MACHINE LEARNING EXPERIMENTS

- Controlled variables in ML:
  - Model: k-NN, decision tree, SVM, neural network, …
  - Algorithm: optimization criteria, implementation, parallelization, …
  - Parameters: model parameters, learning rate, initialization, …
  - Selected features
  - Training data
  - Runtime environment: architecture, OS, number format, …
- Dependent variable(s):
  - System performance
  - Expressed as evaluation criteria: accuracy, precision, recall, F1, AUC, error, RMSE, etc.

- Various biases exist!

TELL ME MORE ABOUT PERFORMANCE MEASURES!

- Goal of experiments/success **needs to be quantifiable**
- **Operationalization:** the process of strictly defining variables into measurable factors
- The better the data is understood and the clearer the goal can be phrased, the more effective optimization will be
- **Goal and relevant performance measure need to be defined clearly before starting experimentation**
  (and defining the experimental setup…)

TELL ME MORE ABOUT PERFORMANCE MEASURES!

- Depending on task and goal (hypothesis), different performance criteria are relevant
  - Spam filtering: does not delete non-spam messages
  - Patent search: finds all relevant patents

- Not all criteria are directly expressible in a performance measure
  - Product recommendation: users are satisfied (?)
- Find a suitable approximation, ideally ruling out all other influences/variables (such as, e.g., user context)

REGRESSION AND NUMERICAL PREDICTION

- For regression tasks (prediction of numerical value), the residual between true value y_i and prediction $\hat{y}_i$ is a typical performance measure, e.g.,

- Mean absolute error: $MAE = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i|$
- Root mean squared error: $RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2}$

→ large errors are disproportionally penalized by squaring the difference

(n … number of test instances)

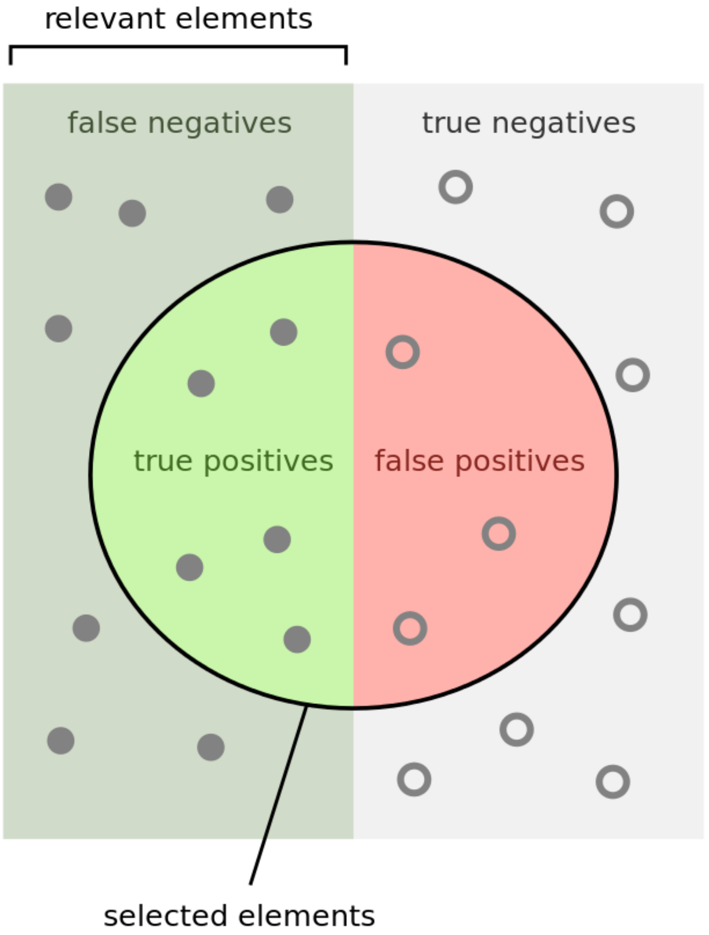
CLASSIFICATION PERFORMANCE

Quality of classifier: how well can the true labels be predicted?

Accuracy: percentage of correctly predicted instances

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

	Classified positive	Classified negative
Actual positive	$\text{TP}_{\#}$	$\text{FN}_{\#}$
Actual negative	$\text{FP}_{\#}$	$\text{TN}_{\#}$



Source: [Wikipedia]

CLASSIFICATION PERFORMANCE

- Example:

Classifier 1	**Classified positive**	**Classified negative**
Actual positive	$10_{\#}$	$15_{\#}$
Actual negative	$25_{\#}$	$100_{\#}$

$$Acc = \frac{10+100}{10+25+15+100} = 73.3\%$$

→ looks pretty ok!

- Now: a really stupid classifier (“computer says no”)

Naysayer	**Classified positive**	**Classified negative**
Actual positive	$0_{\#}$	$25_{\#}$
Actual negative	$0_{\#}$	$125_{\#}$

$$Acc = \frac{0+125}{10+25+15+100} = 83.3\%$$

→ looks even better!

CLASSIFICATION PERFORMANCE

- Class distribution can not be ignored

- Performance measures should always be reported together with a **baseline** reference score
  - e.g., “intelligent guessing” (predict always the most frequent class)
  - e.g., same algorithm without special optimizations (control!)
  - e.g., the currently best performing algorithm (state of the art)

- Accuracy alone is not often a good performance indicator

CLASSIFICATION PERFORMANCE

Gain more understanding of what classifier does

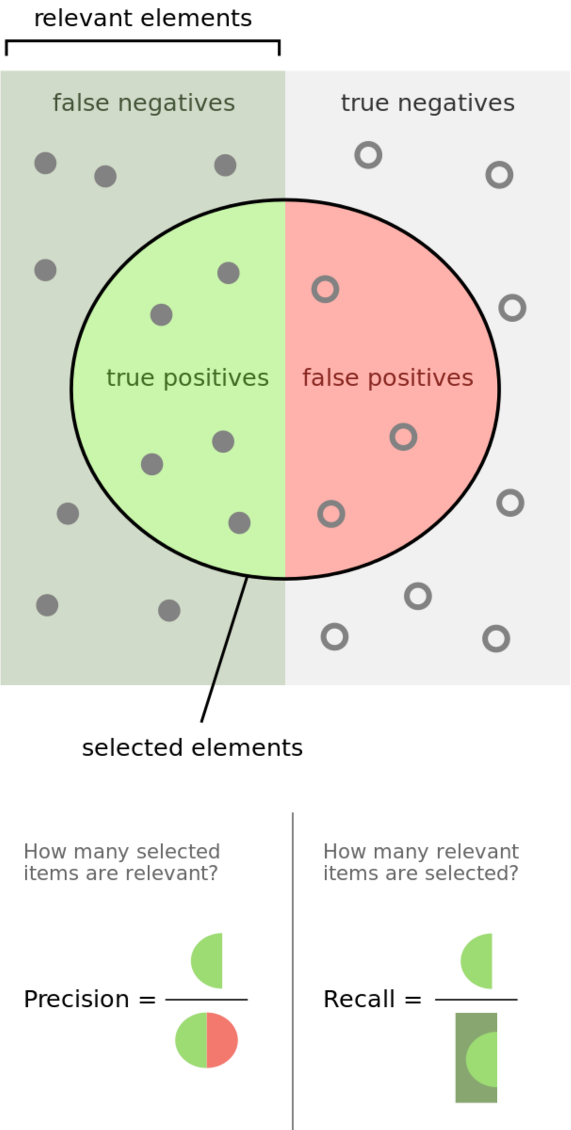
- **Precision**: how many of those predicted as class x are actually correct?

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

- **Recall**: how many of the instances of class x were actually predicted as such?

$$Rec = \frac{TP}{TP + FN}$$

- Precision and recall can be calculated for each class
- Based on parameter tuning, precision can be sacrificed in favor of recall and vice versa (cf. “always-no” example)



Source: [Wikipedia]

→ Which performance metrics would we look to optimize for *a)* a spam detector and *b)* a patent search system?

CLASSIFICATION PERFORMANCE

- If both precision and recall are important, what is the optimization objective?
- Combined measure to balance precision and recall
  (and punish low values of either)
- **F-measure**/F1-score: harmonic mean of precision and recall

$$F = 2 \cdot \frac{Prec \cdot Rec}{Prec + Rec}$$

- General F_β-measure can be tuned to favor Prec or Rec
- Precision, recall, F-measure and variants thereof often used as criteria in information retrieval

CHOOSING THE BEST MODEL

- The model (parameter combination) yielding the best performance according to the chosen criteria is used
- In practice, combinations of variations over several parameter ranges are explored automatically (**grid search**)
  - Brute force approach
  - More efficient heuristics exist, trying to find minima in the parameter space

- In order to compare two models, we should not look at just one number (e.g. a mean value) but compare range of outputs (e.g., variance over several runs, cf. later)
- Calls for **statistical significance testing** (t-test, U test, etc.)
  (not covered today due to time limit…)

ADDITIONAL CRITERIA

Classifier	Accuracy	Runtime
A	90%#	80ms#
B	92%#	95ms#
C	95%#	1,500ms#

- Optimize for accuracy as primary metric in this example
- However, other factors might be relevant, e.g. runtime (classifier C performs best but takes long)
- One strategy: value = accuracy – 0.5 x runtime
- Alternative: maximize accuracy subject to runtime <=100ms
  - Accuracy: **optimizing metric**
  - Runtime: **satisficing metric**

Example from Andrew Ng, Structuring Machine Learning Projects, Coursera.

EXPERIMENT DESIGNS

SIMULATING REAL WORLD BEHAVIOR

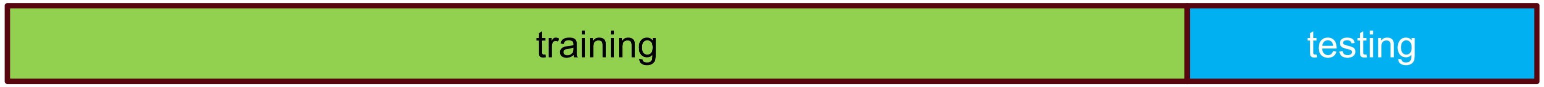
- Goal: learned model must **generalize** to also fit previously unseen data
- Performance on training data gives no information on this
- Need to simulate a more realistic scenario and find model that performs well on unseen data

- Idea: hold back some of the training data from training and use to test performance
- Data used for training and for testing should resemble each other (similar properties, same distribution)
- No data used for testing (or information extracted from it) must ever be used in the training of a model!

RANDOM SAMPLING

To test model on new data, draw a **random sample**

1. Random shuffle of the data!
2. Partition data into part for training and part for estimating performance (testing)



training

testing

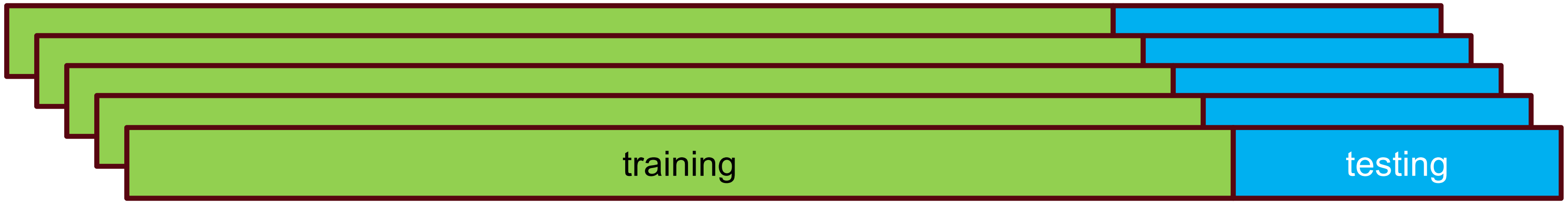
3. Calculate success criteria (performance metric) on testing set

Issues: small number of testing instances and by chance, we might not evaluate important instances (=bias)

(we need as much data as possible for training…)

REPEATED RANDOM SAMPLING

- Repeat that process n times (always shuffle anew!)



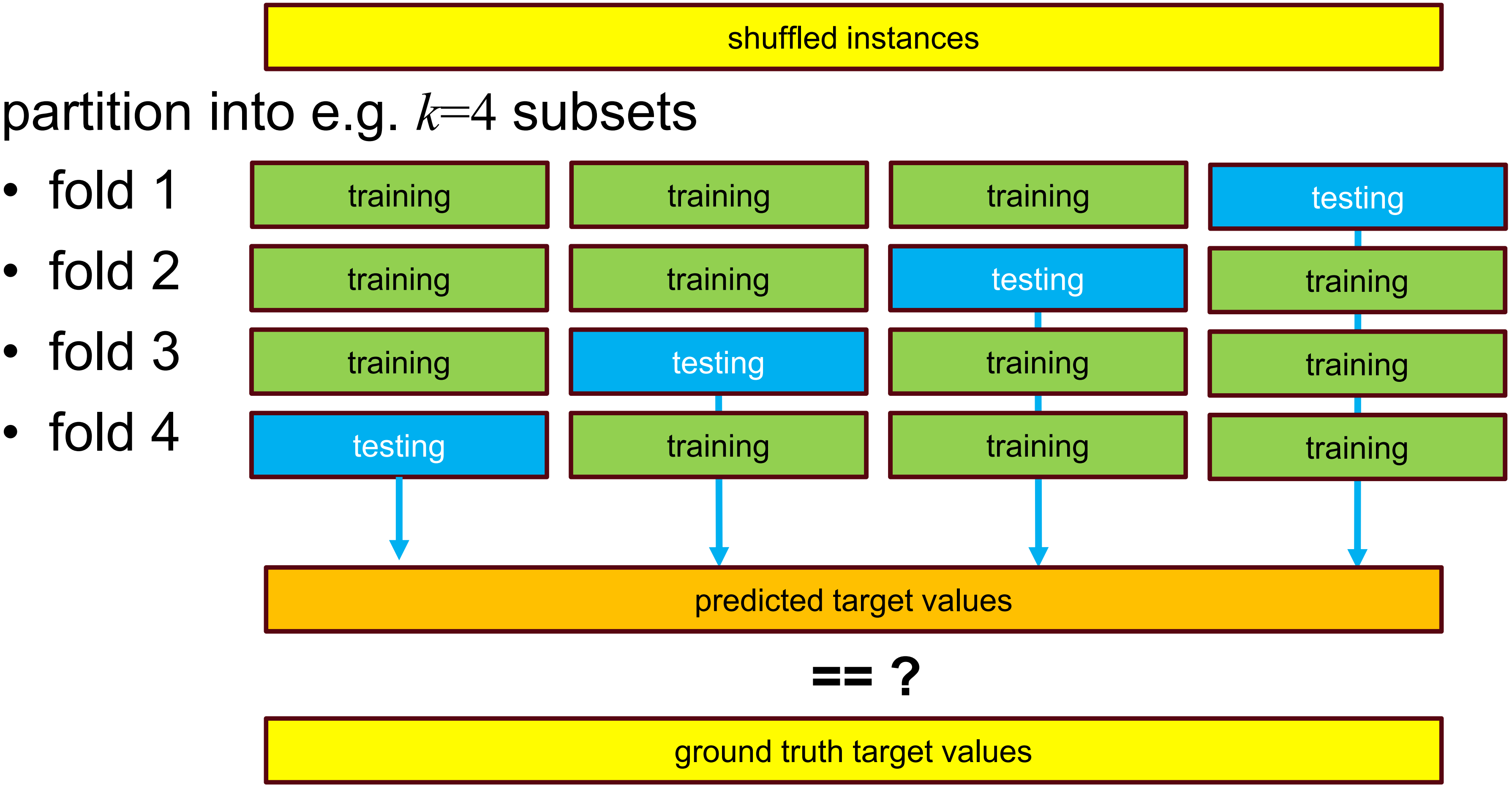
- Results in n models and n performance scores
- Aggregate scores (e.g. mean)
- Actually sample of scores from underlying distribution
- Compare models (parameters) via aggregated scores and chose the best one (a final model can be trained with the best settings using all the data)
- Issue: we might happen to favor some instances for testing performance (might appear in test sets more often)

SMALL NUMBER OF EXAMPLES

- To make use of all instances for testing and give equal impact of instances on performance measure
- ***k*-fold cross validation**

1. Shuffle data (!)
2. Partition data into k (near-)equally sized subsets
3. Train k models such that
   - k-1 subsets are used for training
   - The remaining subset is used for testing
   - No two models are tested on the same subset
4. Prediction available on all subsets (thus each instance used exactly once for testing)
5. Calculate performance over full predicted set
6. Optionally: repeat process multiple times

K-FOLD CROSS VALIDATION



shuffled instances

partition into e.g. k=4 subsets

• fold 1	training	training	training	testing
• fold 2	training	training	testing	training
• fold 3	training	testing	training	training
• fold 4	testing	training	training	training

predicted target values

== ?

ground truth target values

K-FOLD CROSS VALIDATION

- Typical range for k: $[2, 10]$

- Special variant: leave-one-out CV
  - $k = \#instances$
  - Closest simulation (models very similar to final model)
  - Typically too expensive, not seen very often anymore

- Variant: Stratified k-fold cross validation
  - each partition should resemble same distribution of target classes than overall class distribution
  - Only possible if $k <= \#\, instances\ of\ least\ frequent\ class$

K-FOLD CROSS VALIDATION

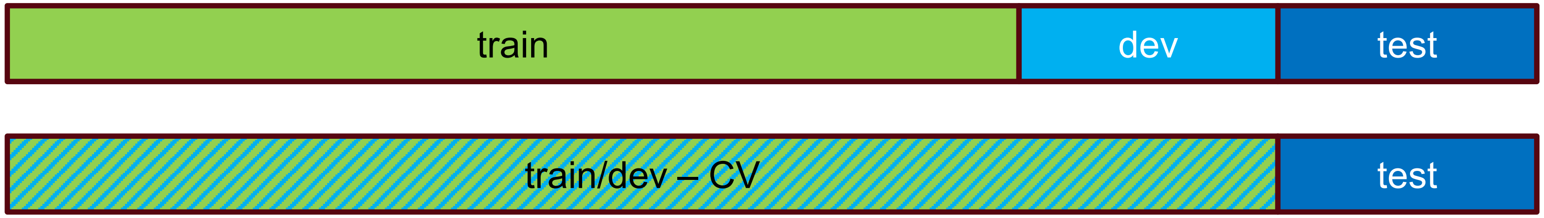
Common mistake made with CV:

- Estimation of parameters, feature selection, dimensionality reduction, normalization etc. performed prior to CV
- → information of data used for testing leaks into training!
- **All these steps need to be carried out for each fold individually!**
- Might make CV very expensive to calculate

- Also: same splits should be used when comparing models/settings!

TRAIN-, DEVELOPMENT-, TEST SETS

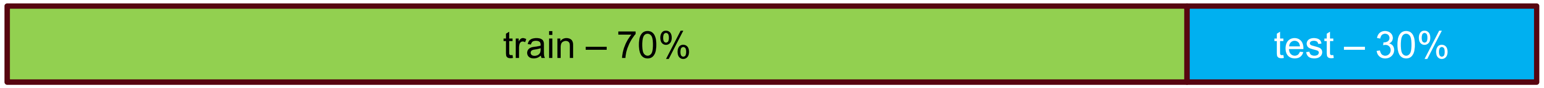
- Repeated splitting of same data for optimization might lead again to overfitting
- Final estimate of real-world performance should be made on another, independent test set
- Again, reserve part of the data with relevant properties for later testing, e.g.,



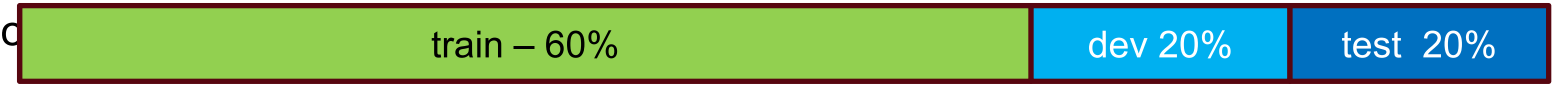
- NB: name of set for param. optimization (development set) not consistent in literature; e.g., often called **validation set**

THESE DAYS…

- Old-school way of splitting sets when data was scarce and precious (#instances … magnitude 100-100k)

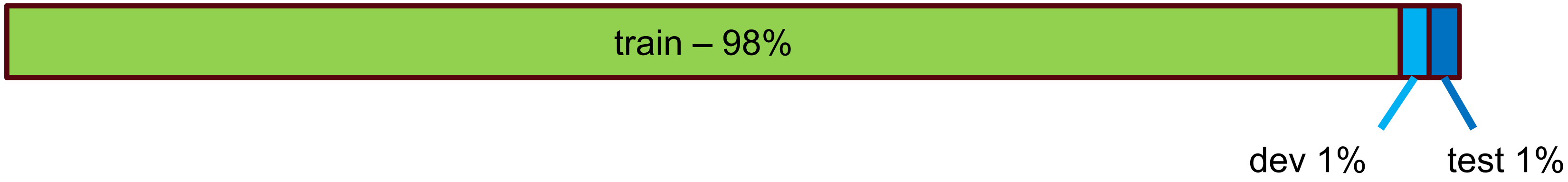


- To



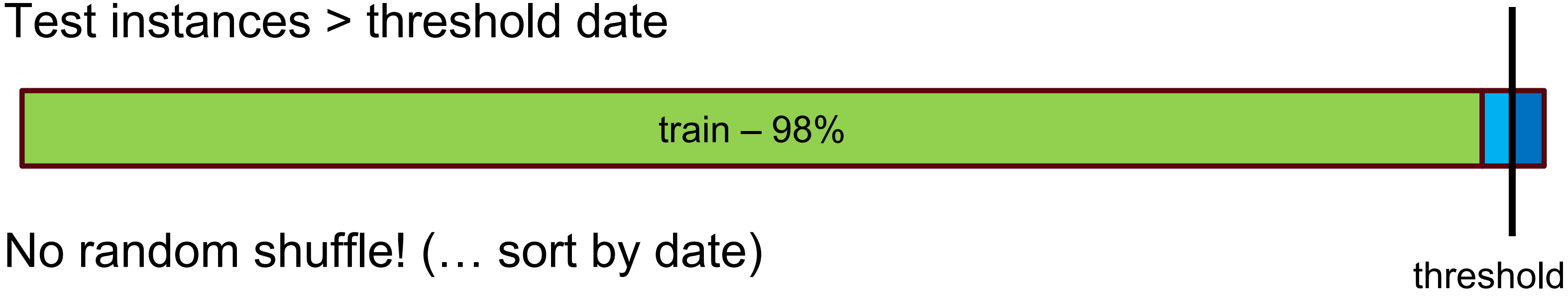
Still, dev and test set contain 10k instances…

Aim: test set should be big enough to give high confidence in overall system performance



TIME-BASED SPLIT

- Depending on the scenario to evaluate, other experiment setups might be more relevant (bias desired)
- Consider a recommender system:
- Goal: predict future behavior from collected data
- Simulation: predict interactions after a chosen point in time by learning from interactions before
- Potentially for each user individually
- **Time-based split**
- Training instances (+dev?) <= threshold date
  Test instances > threshold date



- No random shuffle! (… sort by date)

ONE LAST THING…

- **Pilot experiments** are important!

- Not necessarily designed to test the hypothesis but to test the experimental apparatus
- i.e., check whether pipeline actually can test the hypothesis

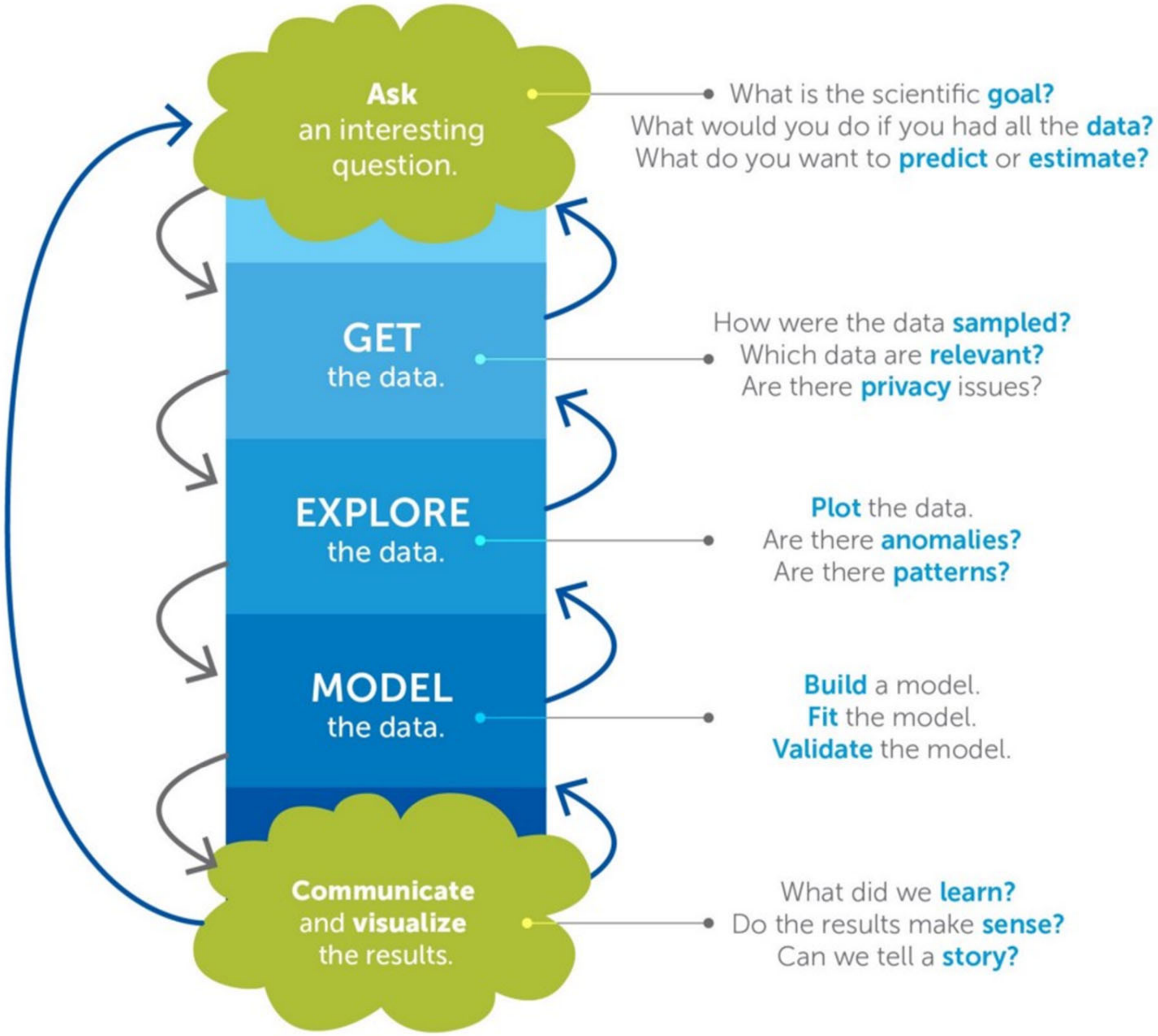
- In pilot experiments:
  - Provide preliminary data
  - Check if protocol works
  - Check for plausible results and outcomes
  - Try out statistical analysis
  - Find bugs!

AUSTRIAN INSTITUTE OF TECHNOLOGY

Data Science Process

EXPERIMENT DESIGNS

DATA SCIENCE PROCESS



Ask an interesting question.

What is the scientific **goal?**
What would you do if you had all the **data?**
What do you want to **predict** or **estimate?**

GET the data.

How were the data **sampled?**
Which data are **relevant?**
Are there **privacy** issues?

EXPLORE the data.

Plot the data.
Are there **anomalies?**
Are there **patterns?**

MODEL the data.

Build a model.
Fit the model.
Validate the model.

Communicate and visualize the results.

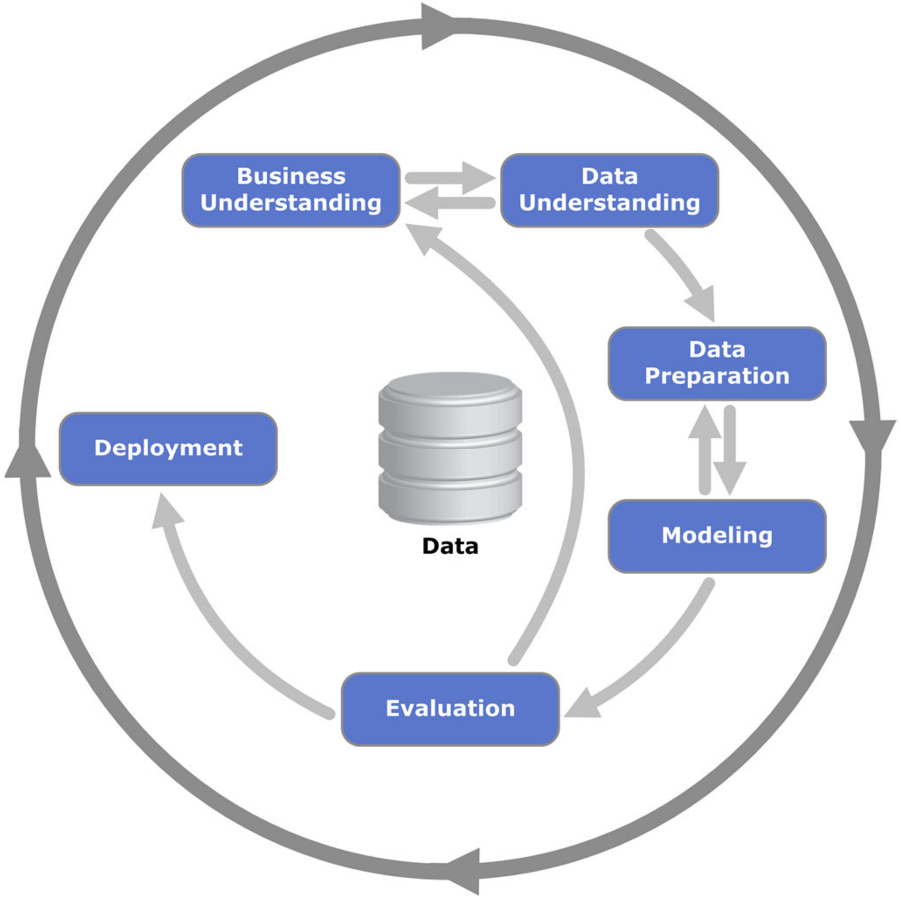
What did we **learn?**
Do the results make **sense?**
Can we tell a **story?**

Derived from the work of Joe Blitzstein and Hanspeter Pfister, originally created for the Harvard data science course http://cs109.org/.

- Build hypothesis, define target metric
- Check data, understand origin and distribution, prepare for experiments
- Learn model, optimize model
- Interpret results
- Try again…

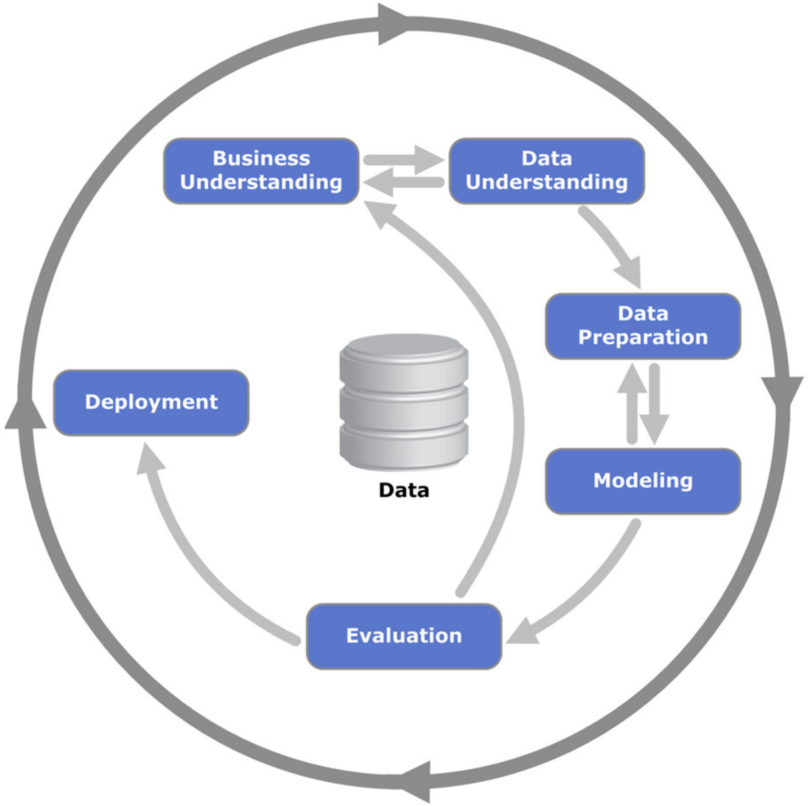
CF. CRISP-DM

- Cross-industry standard process for data mining (CRISP-DM) from 1996
- Business-oriented, iterative process developed to organize data mining
- 6 phases:
  1. Business understanding
  2. **Data understanding**
  3. **Data preparation**
  4. **Modeling**
  5. **Evaluation**
  6. Deployment



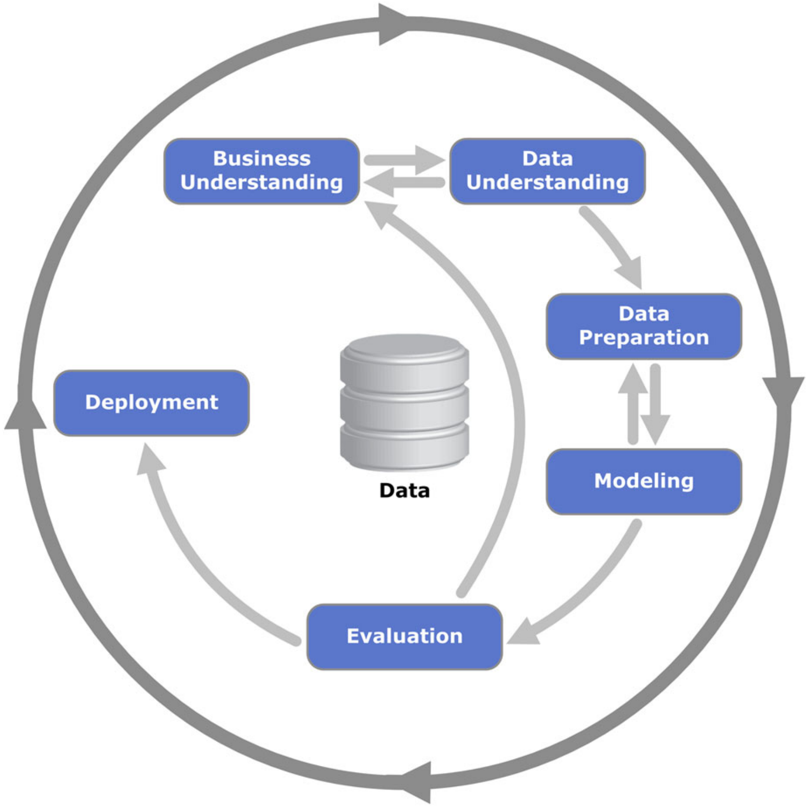
CF. CRISP-DM

1. **Business understanding**
   assessing the situation (business requirements, risks, cost, etc.), determining data mining goals, producing project plan
   (cf. “hypothesis building”)
2. **Data understanding**
   collecting, describing, exploring, verifying data
3. **Data preparation**
   selecting, cleaning, constructing data
4. **Modeling**
   selecting model, generating test design, building + assessing model



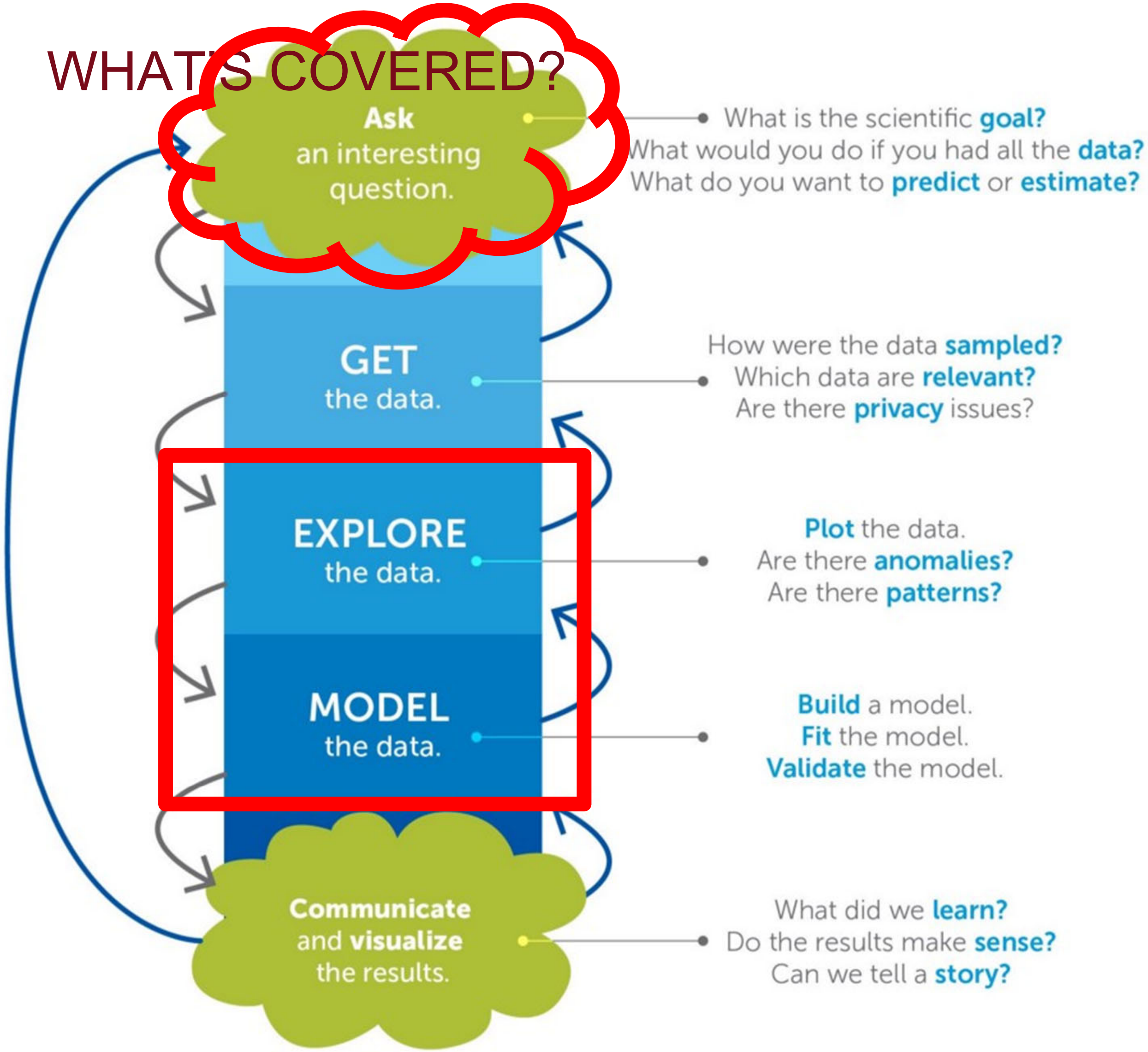
CF. CRISP-DM

5. **Evaluation**
   evaluating results, reviewing process, determining next steps
6. **Deployment**
   planning deployment, monitoring, maintenance, reporting



- References:
  - Colin Shearer, *The CRISP-DM model: the new blueprint for data mining*, Journal of Data Warehousing, 5(4), pp.13–22, 2000
  - IBM SPSS Modeler CRISP-DM Guide: ftp://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/18.0/en/ModelerCRISPDM.pdf

WHAT'S COVERED?



Ask an interesting question.

What is the scientific **goal?**
What would you do if you had all the **data?**
What do you want to **predict** or **estimate?**

GET the data.

How were the data **sampled?**
Which data are **relevant?**
Are there **privacy** issues?

EXPLORE the data.

Plot the data.
Are there **anomalies?**
Are there **patterns?**

MODEL the data.

Build a model.
Fit the model.
Validate the model.

Communicate and visualize the results.

What did we **learn?**
Do the results make **sense?**
Can we tell a **story?**

Derived from the work of Joe Blitzstein and Hanspeter Pfister, originally created for the Harvard data science course http://cs109.org/.

- ☞ **experimental design**

- ☞ **data preprocessing**
- ☞ **data-oriented programming**
- ☞ **basics in machine learning**

- ☞ **practical example of the whole process in Python**

WRAP-UP AND RECAP

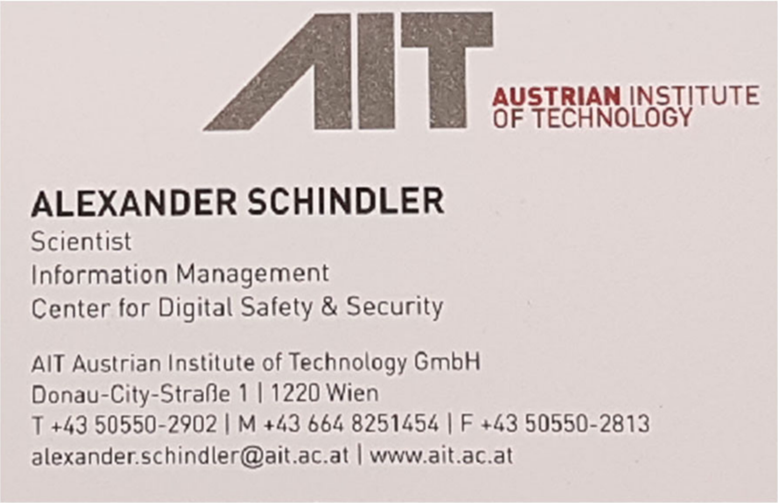
- Data Science is an empirical science
  - Investigate data transformation processes using scientific methods
  - Methods originally applied to natural objects (fundamental particles, chemicals, living organisms) or individuals and social groups
- Strategies from Data Mining and Machine Learning experimentation
  - Definition of target criteria measuring success
  - Preparation/selection of training, development, and test sets
- Iterative process
  1. Construct hypotheses/build (approximate) theories
  2. Test with empirical experiments
  3. Refine hypotheses and modelling assumptions

FURTHER READING

- Paul R. Cohen, *Empirical Methods for Artificial Intelligence*, MIT Press, 1995.
- Tom Mitchell, *Machine Learning*, McGraw Hill, 1997.
- Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern Classification (2nd Ed.)*, Wiley, 2000.
- Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag, Berlin, Heidelberg, 2006.
- Joel Grus, *Data Science from Scratch: First Principles with Python*, O’Reilly, 2015.

AUSTRIAN INSTITUTE OF TECHNOLOGY
TOMORROW TODAY

THANK YOU!



AUSTRIAN INSTITUTE OF TECHNOLOGY

ALEXANDER SCHINDLER
Scientist
Information Management
Center for Digital Safety & Security

AIT Austrian Institute of Technology GmbH
Donau-City-Straße 1 | 1220 Wien
T +43 50550-2902 | M +43 664 8251454 | F +43 50550-2813
alexander.schindler@ait.ac.at | www.ait.ac.at