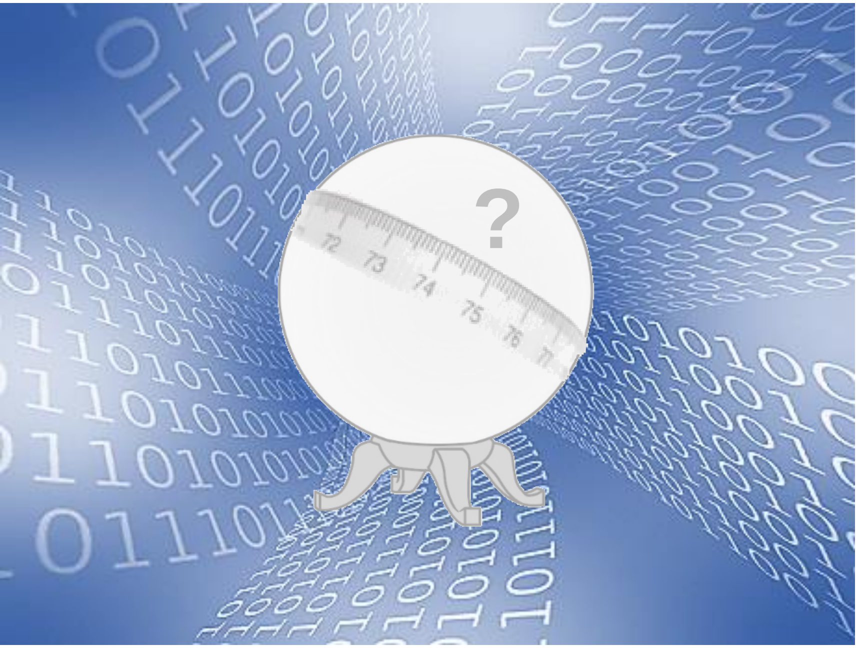
Prof. Dr. Bastian Amberg
School of Business & Economics
Department of Information Systems



Business Intelligence

13 What is a good model?

Prof. Dr. Bastian Amberg
(summer term 2024)
28.6.2024

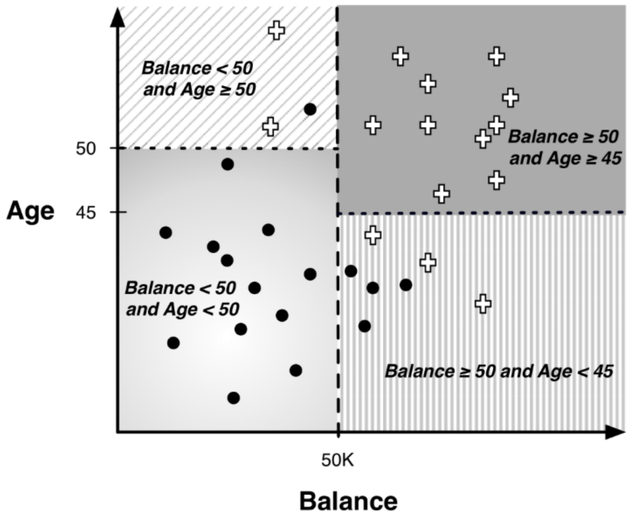
Schedule

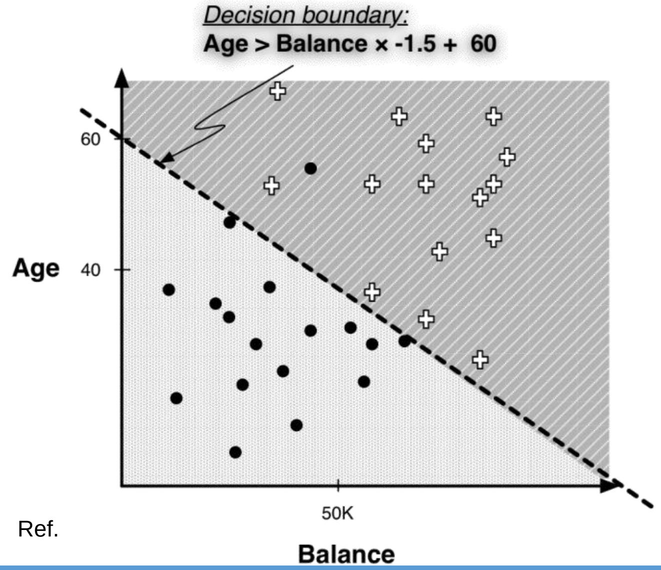
		Wed., 10:00-12:00		Fr., 14:00-16:00 (Start at 14:30)		Self-study	
Basics	W1	17.4.	(Meta-)Introduction	19.4.		Python-Basics	Chap. 1
	W2	24.4.	Data Warehouse – Overview & OLAP	26.4.	*[Blockveranstaltung SE Prof. Gersch]*		Chap. 2
	W3	1.5.		3.5.			Chap. 3
	W4	8.5.	Data Warehouse Modeling I & II	10.5.	Data Mining Introduction		
Main Part	W5	15.5.	CRISP-DM, Project understanding	17.5.	Python-Basics-Online Exercise	Python-Analytics	Chap. 1
	W6	22.5.	Data Understanding, Data Visualization I	24.5.	*No lectures, but bonus tasks 1.) Co-Create your exam 2.) Earn bonus points for the exam*		Chap. 2
	W7	29.5.	Data Visualization II	31.5.			
	W8	5.6.	Data Preparation	7.6.	Predictive Modeling I (10:00 -12:00)	BI-Project	Start
	W9	12.6.	Predictive Modeling II	14.6.	Python-Analytics-Online Exercise		|
	W10	19.6.	*Guest Lecture Dr. Ionescu*	21.6.	Fitting a Model		|
	W11	26.6.	How to avoid overfitting	28.6.	What is a good Model?		|
Deep-ening	W12	3.7.	Project status update Evidence and Probabilities	5.7.	Similarity (and Clusters) From Machine to Deep Learning I		|
	W13	10.7.		12.7.	From Machine to Deep Learning II		|
	W14	17.7.	Project presentation	19.7.	Project presentation		End
Ref.					*Klausur 1.Termin, 31.7.'24 Klausur 2.Termin, 2.10.'24*	Projektbericht	

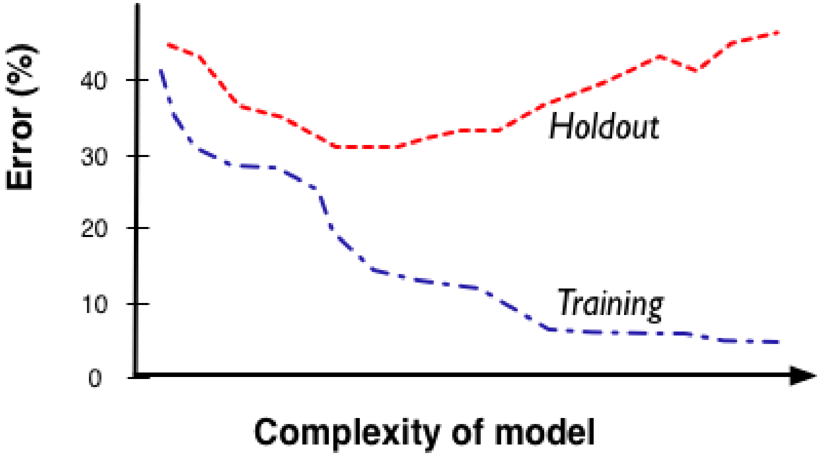
Case Study

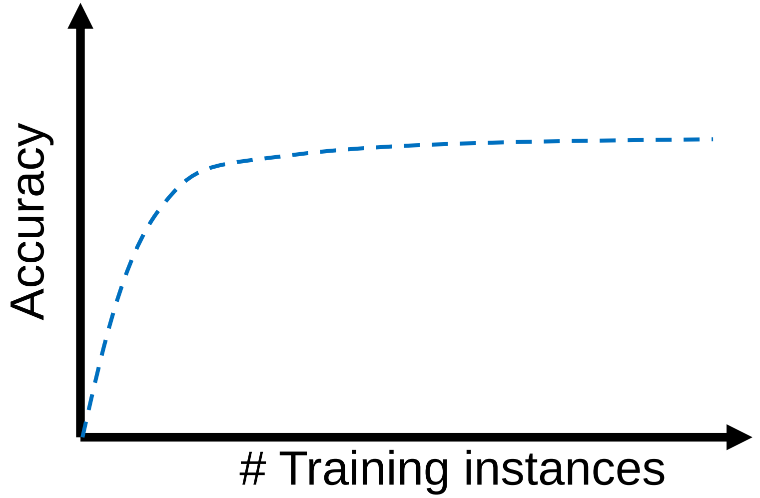
Last Lesson(s) – Exercise

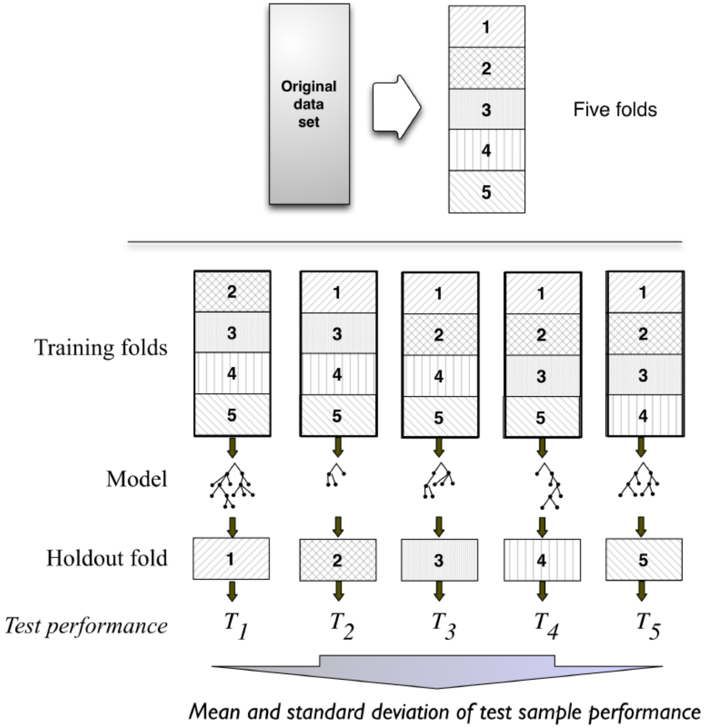
Explain what you see.











Kahoot-Fragen
www.kahoot.it
(über Smartphone oder Laptop)
PIN folgt

10 Min.

Ref.

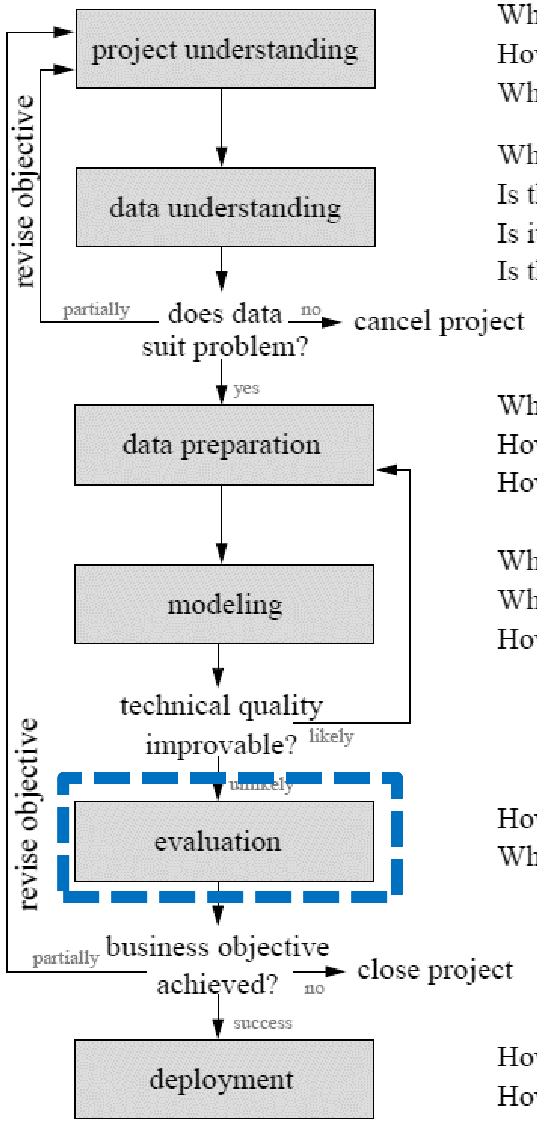
CRISP-DM

Cross
Industry
Standard
Process for
Data
Mining

Iteration as
a rule

Process of data
exploration

Implementation of the
KDD Process



What exactly is the problem, the expected benefit?
How would a solution look like?
What is known about the domain?

What data do we have available?
Is the data relevant to the problem?
Is it valid? Does it reflect our expectations?
Is the data quality, quantity, recency sufficient?

Which data should we concentrate on?
How is the data best transformed for modeling?
How may we increase the data quality?

What kind of model architecture suits the problem best?
What is the best technique/method to get the model?
How good does the model perform technically?

How good is the model in terms of project requirements?
What have we learned from the project?

How is the model best deployed?
How do we know that the model is still valid?

Ref. Wirth / Hipp (2000), Azevedo (2008)

Modeling: Open issue from last lesson:

Avoiding Overfitting and Complexity Control in Parametric Learning/Optimization

$\arg\max_{\mathbf{w}} \ \mathrm{fit}(\mathbf{x}, \mathbf{w})$ → ***penalize complexity***

$\arg\max_{\mathbf{w}} \ [\mathrm{fit}(\mathbf{x}, \mathbf{w}) - \lambda \cdot \mathrm{penalty}(\mathbf{w})]$

we will come back to this topic later

Agenda

(1) Measuring accuracy

- Confusion matrix
- Unbalanced classes

(2) Expected Value

- Evaluate classifier use
- Frame classifier evaluation

(3) Evaluation and baseline performance

Ref.

Introduction

What is desired from data mining results?

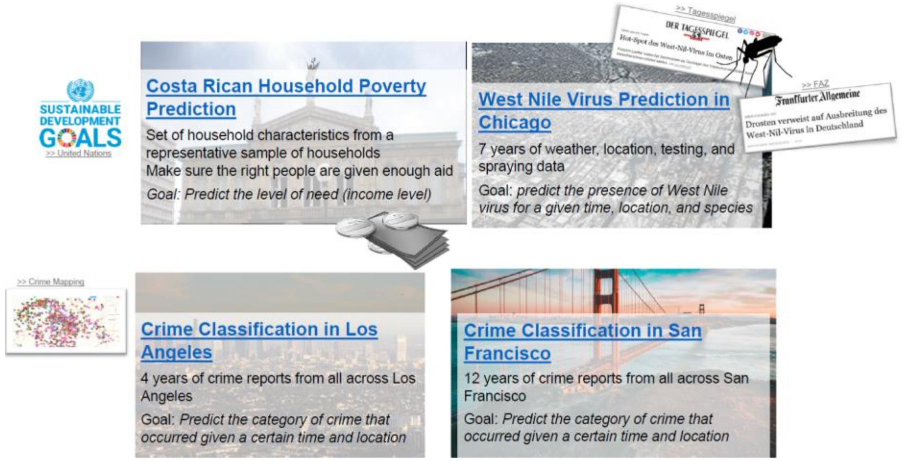
How would you **measure** that your model is any good?
How to measure performance in a meaningful way?

Model evaluation is **application-specific**

- We look at common issues and themes in evaluation

Frameworks and metrics for classification and instance scoring

Think about the specific BI project you are working on....



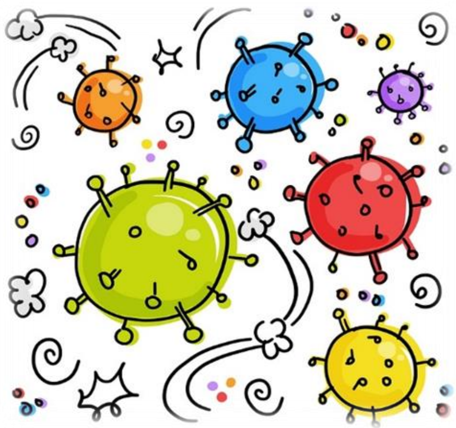
Ref.

Bad positives and harmless negatives

Classification terminology

a **bad** outcome → a “positive” example [alarm!]

a **good** outcome → a “negative” example [uninteresting]



Further examples

medical test: positive test → disease is present

fraud detector: positive test → unusual activity on account

A classifier tries to distinguish the majority of cases (**negatives**, the uninteresting)
from the small number of alarming cases (**positives**, alarming)

number of mistakes made on **negative** examples (false positive errors) will be relatively high / may dominate

cost of each mistake made on a **positive** example (false negative error) will be relatively high / will be higher

Ref.

Measuring accuracy and its problems

Up to now: measure a model‘s performance by some simple metric
classifier error rate, accuracy

Simple example: accuracy

$$accuracy = \frac{\text{Number of correct decisions made}}{\text{Total number of decisions made}}$$

Classification accuracy is popular, but usually **too simplistic** for applications of data mining to real business problems

Decompose and count the different types of correct and incorrect decisions made by a classifier

Ref.

The confusion matrix

A **confusion matrix** for a problem involving n classes is an $n \times n$ matrix with the columns labeled with actual classes and the rows labels with predicted classes.

For a binary variable it is as follows:

$$\begin{array}{ccc} & & \begin{array}{cc} \textbf{p} \qquad\qquad & \qquad\qquad \textbf{n} \end{array} \\ \text{Predicted} & \begin{array}{c} \textbf{Y} \\ \textbf{N} \end{array} & \begin{pmatrix} \text{True positives} & \text{False positives} \\ \text{False negatives} & \text{True negatives} \end{pmatrix} \end{array}$$

Each example in a test set has an **actual class label** and the **class predicted** by the classifier

Interpretation:

The confusion matrix separates the decisions made by the classifier

- actual/true classes: **p**(ositive), **n**(egative)
- predicted classes: **Y**(es), **N**(o)
- The main diagonal contains the count of correct decisions

```
From sklearn import metrics

metrics.confusion_matrix(expected_list, predicted_list)
```

(https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html)

Ref.

Mini-Exercise - The confusion matrix

$$\begin{array}{ccc} & & \textbf{p} \quad\quad\quad\quad \textbf{n} \\ \text{Predicted} & \begin{array}{c}\textbf{Y}\\ \textbf{N}\end{array} & \begin{pmatrix} \text{True positives} & \text{False positives} \\ \text{False negatives} & \text{True negatives} \end{pmatrix} \end{array}$$

Diese Folie ist nach der Vorlesung mit Lösungen verfügbar

Kahoot-Fragen
www.kahoot.it
(über Smartphone oder Laptop)
PIN folgt

10 Min.

Ref.

The confusion matrix

Unbalanced classes

In most real world classification problems, one class is often **rare**

- Classification is used to find a relatively small number of **unusual ones** (defrauded customers, defective parts, targeting consumers who actually would respond, …)
- The class distribution is unbalanced ("skewed")

Evaluation based on **accuracy** does not work

Example: 999:1 ratio
always choose the most prevalent class – 99.9% accuracy!

Fraud detection: skews of 1:99
Is a model with 80% accuracy always better than a model with 37% accuracy?

We need to have more information about the population

Ref.

Image: Elliott Brown (2016) | Flickr

The confusion matrix

Unbalanced classes

Consider two models A and B for the churn example (~every tenth customer churns)

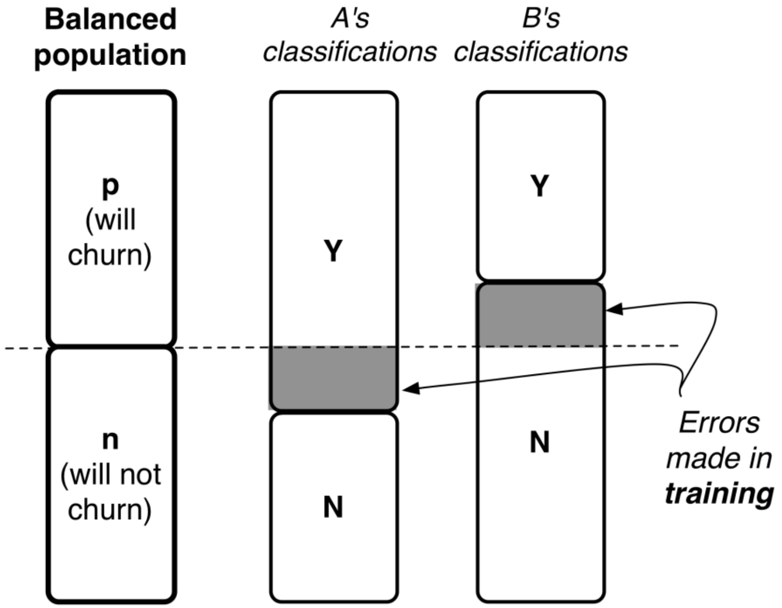
We train with a balanced population (1000 customer)

1. Both models correctly classify 80% of the balanced pop.
2. Classifier A often falsely predicts that customers will churn
3. Classifier B makes many opposite errors

Unbalanced population:

4. A‘s accuracy is 40%, B‘s accuracy is 97%

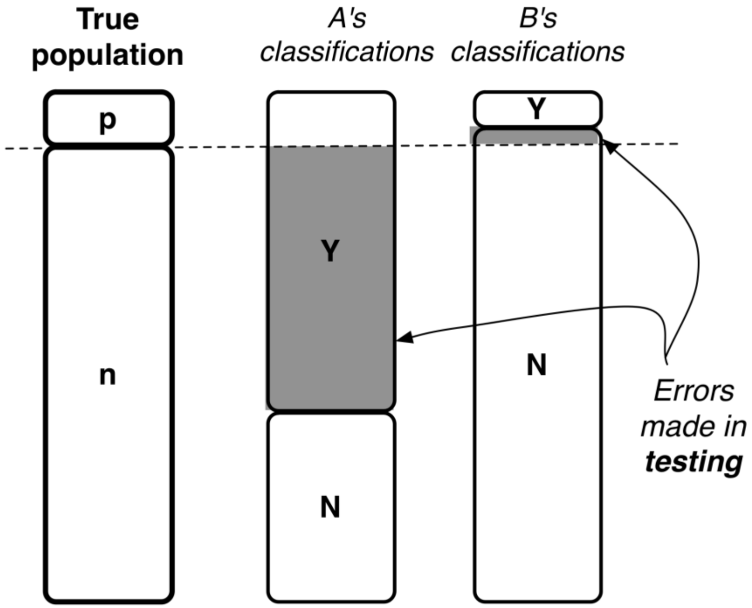
Note the **different performances** of the models in form of a confusion matrix:



$$CM_A = \begin{array}{c} \\ Y \\ N \end{array} \begin{array}{cc} \text{churn} & \text{not churn} \\ \end{array}$$

$$CM_A = \begin{array}{c|cc} & \text{churn} & \text{not churn} \\ \hline Y & 500 & 200 \\ N & 0 & 300 \end{array}$$

$$CM_B = \begin{array}{c|cc} & \text{churn} & \text{not churn} \\ \hline Y & 300 & 0 \\ N & 200 & 500 \end{array}$$



$$CM_A = \begin{array}{c|cc} & \text{churn} & \text{not churn} \\ \hline Y & 100 & 600 \\ N & 0 & 300 \end{array}$$

$$CM_B = \begin{array}{c|cc} & \text{churn} & \text{not churn} \\ \hline Y & 70 & 0 \\ N & 30 & 900 \end{array}$$

Ref.

Which model is better?

Unequal costs and benefits

How much do we care about the different **errors** and correct decisions?

- Classification accuracy makes no distinction between **false positive** and **false negative** errors
- In real-world applications, different kinds of errors lead to different consequences.

Examples for medical diagnosis:

a patient has cancer (although she/he does not)
*→ **false positive error**, expensive, but not life threatening*

a patient has cancer, but she/he is told that she/he has not
*→ **false negative error**, more serious*

Errors should be counted separately:
Estimate cost or benefit of each decision

Ref.

Agenda

(1) Measuring accuracy

- Confusion matrix
- Unbalanced classes

(2) Expected Value

- Evaluate classifier use
- Frame classifier evaluation

(3) Evaluation and baseline performance

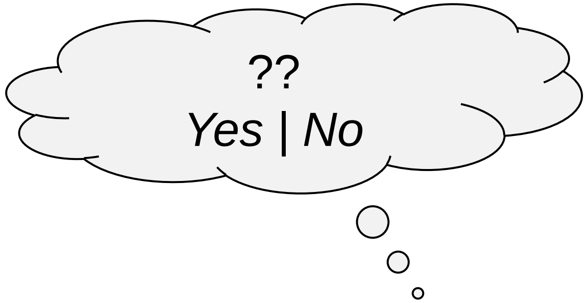
Ref.

The expected value framework

Expected value calculation includes **enumeration of the possible outcomes** of a situation

Expected value = weighted average of the values of different possible outcomes, where the weight given to each value is the probability of its occurrence

- Example: different levels of profit
- We focus on the maximization of expected profit



General form of expected value computation:

$$EV = p(o_1) \cdot v(o_1) + p(o_2) \cdot v(o_2) + \cdots +$$

with o_i as possible decision outcome,
$p(o_i)$ as its probability, and $v(o_i)$ as its value.

Probabilities can be **estimated** from available data

We consider two cases

I. Evaluate classifier use
II. Frame classifier evaluation

Ref.

(I) Expected value for use of a classifier

Use of a classifier: predict a class and take some action

Example target marketing: assign each consumer to either a class „likely responder“ or „not likely responder“

Response is usually relatively low – so no consumer may seem like a likely responder

Computation of the expected value

A model gives an estimated probability of response $\hat{p}_R(\boldsymbol{x})$ for any consumer with a feature vector $\boldsymbol{x}$

Calculate expected benefit (or costs) of targeting consumer $\boldsymbol{x}$: $\hat{p}_R(\boldsymbol{x}) \cdot v_R + (1 - \hat{p}_R(\boldsymbol{x})) \cdot v_{NR}$ with v_R being the value of a response and v_{NR} the value from no response

Example:

Price of product: \$200, costs of product: \$100

Targeting a consumer: \$1, profit $v_R = \$99$, $v_{NR} = -\$1$

Do we make a profit? Is the expected value (profit) of targeting greater than zero?

$$\hat{p}_R(\boldsymbol{x}) \cdot \$99 + (1 - \hat{p}_R(\boldsymbol{x})) \cdot (-\$1) > 0$$

$$\leftrightarrow \qquad \hat{p}_R(\boldsymbol{x}) \cdot \$99 > (1 - \hat{p}_R(\boldsymbol{x})) \cdot \$1$$

$$\leftrightarrow \qquad \hat{p}_R(\boldsymbol{x}) > 0.01$$

We should target the consumer as long as the estimated probability of responding is greater than 1%

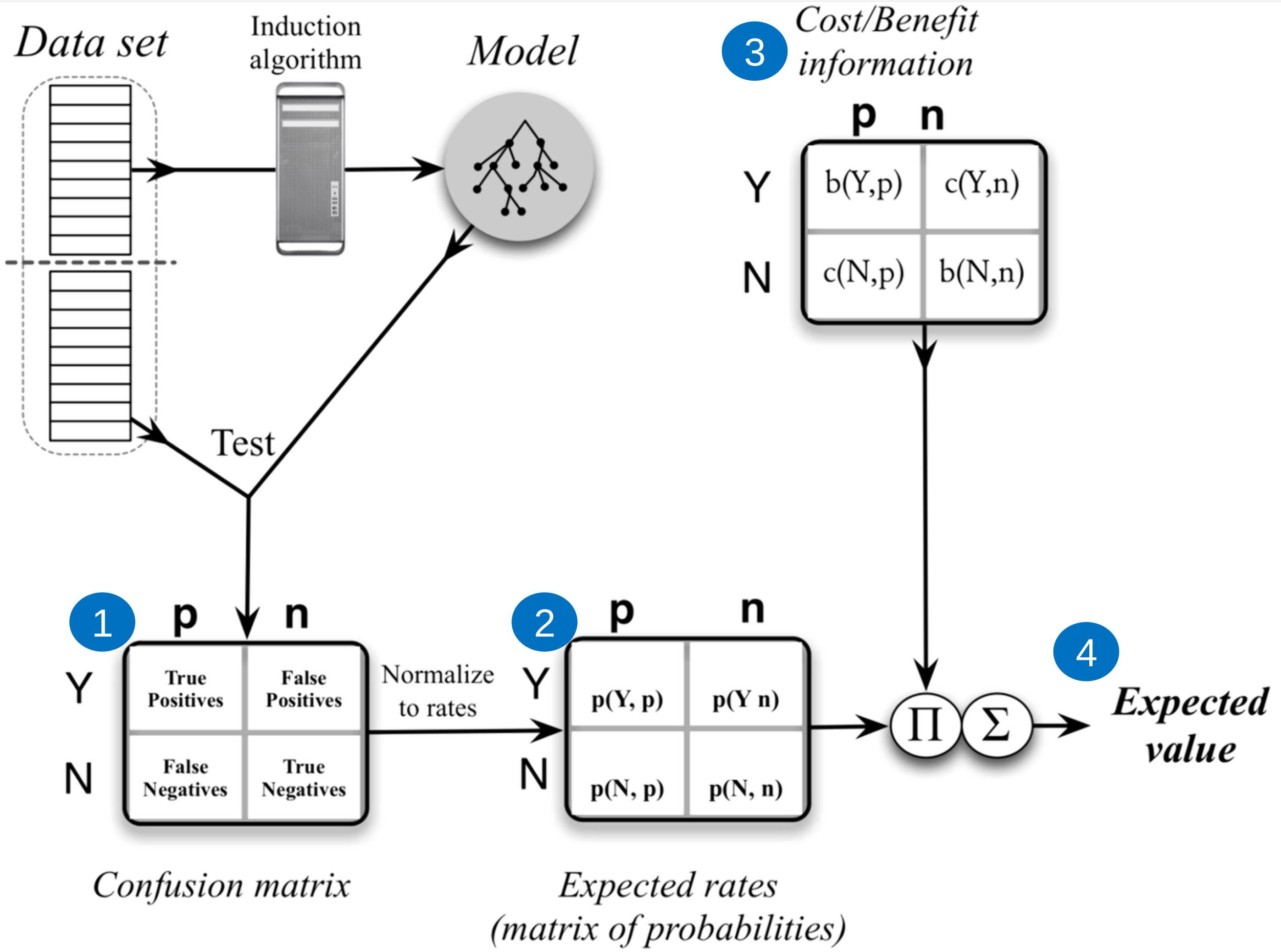
Ref.

(II) Expected value classification

Goal: **compare the quality of different models** with each other

- Does the data-driven model perform better than a hand-crafted model?
- Does a classification tree work better than a linear discriminant model?
- Do any of the models perform substantially better than a baseline model?

In short:
How well does each model perform with regards to its expected value?



Ref.

Expected rates for evaluation of a classifier

2

Aggregate together all the different cases:

- When we target consumers, what is the probability that they (do not) respond?
- What about when we do not target consumers, would they have responded?

This information is available in the confusion matrix

- Each o_i corresponds to one of the possible combinations of the class we predict/the actual class

Where do the probabilities of errors and correct decisions actually come from?

- Each cell of the confusion matrix contains a count of the number of decisions corresponding to the combination of (predicted, actual)
- $count(h, a)$

Compute estimated probabilities as

- $p(h, a) = count(h, a)/Total$

Example confusion matrix/estimates of probability

Predicted	Actual p	Actual n
Y	56	7
N	5	42

$T = 110, P = 61, N = 49$ (**P**ositive, **N**egative)

$$p(Y, \mathrm{p}) = \frac{56}{110} = 0.51, \quad p(Y, \mathrm{n}) = \frac{7}{110} = 0.06$$

$$p(N, \mathrm{p}) = \frac{5}{110} = 0.05, \quad p(N, \mathrm{n}) = \frac{42}{110} = 0.38$$

Ref.

Costs and benefits

Freie Universität

Please read on your own

3

Compute **cost-benefit values** for each decision pair

A cost-benefit matrix specifies for each (predicted,actual) pair the cost or benefit for making such a decision

		Actual	
		p	**n**
Predicted	**Y**	b(**Y,p**)	c(**Y,n**)
	N	c(**N,p**)	b(**N,n**)

Correct classifications (true positives and negatives) correspond to $b(Y,p)$ and $b(N,n)$, respectively

Incorrect classifications (false positives and negatives) correspond to c(Y,n) and c(N,p), respectively [often negative benefits or costs]

Costs and benefits cannot be estimated from data

How much is the true value for retaining a customer? (i.e., CLV)

Ref. Often use of average estimated costs and benefits

Targeted marketing example

- **False positive** occurs when we classify a consumer as a likely responder and therefore target her, but she does not respond
→ cost $c(Y,\mathrm{n}) = -1$ //or negative benefit $b(Y,\mathrm{n})$
- **False negative** is a consumer who was predicted not to be a likely responder, but would have bought if offered. No money spent, nothing gained
→ cost $c(N,\mathrm{p}) = 0$ //or negative benefit $b(N,\mathrm{p})$
- **True positive** is a consumer who is offered the product and buys it
→ benefit $b(Y,\mathrm{p}) = 200 - 100 - 1 = 99$
- **True negative** is a consumer who was not offered a deal but who would not have bought it
→ benefit $b(N,\mathrm{n}) = 0$

Sum up in cost-benefit matrix

		Actual	
		p	n
Predicted	Y	99	-1
	N	0	0

Expected profit computation

4

	p	n
Y	p(Y, p)	p(Y n)
N	p(N, p)	p(N, n)

	p	n
Y	b(**Y,p**)	c(**Y,n**)
N	c(**N,p**)	b(**N,n**)

Sufficient for comparison of various models

Compute **expected profit** by cell-wise multiplication of the matrix of costs and benefits against the matrix of probabilities:

$$EP = p(Y,\mathrm{p}) \cdot b(Y,\mathrm{p}) + p(N,\mathrm{p}) \cdot c(N,\mathrm{p}) + p(N,\mathrm{n}) \cdot b(N,\mathrm{n}) + p(Y,\mathrm{n}) \cdot c(Y,\mathrm{n})$$

Alternative calculation: factor out the probabilities of seeing each class (class priors) [Use $p(x,y) = p(y) \cdot p(\mathrm{x} \mid y)$]

- Class priors $p(\mathrm{p})$ and $p(\mathrm{n})$ specify the likelihood of seeing positive versus negative instances
- Factoring out allows us to separate the influence of class imbalance from the predictive power of the model

Factoring out priors yields the following **alternative expression** for expected profit:

$$EP = p(Y|\mathrm{p}) \cdot p(\mathrm{p}) \cdot b(Y,\mathrm{p}) + p(N|\mathrm{p}) \cdot p(\mathrm{p}) \cdot c(N,\mathrm{p}) + p(N|\mathrm{n}) \cdot p(\mathrm{n}) \cdot b(N,\mathrm{n}) + p(Y|\mathrm{n}) \cdot p(\mathrm{n}) \cdot c(Y,\mathrm{n})$$

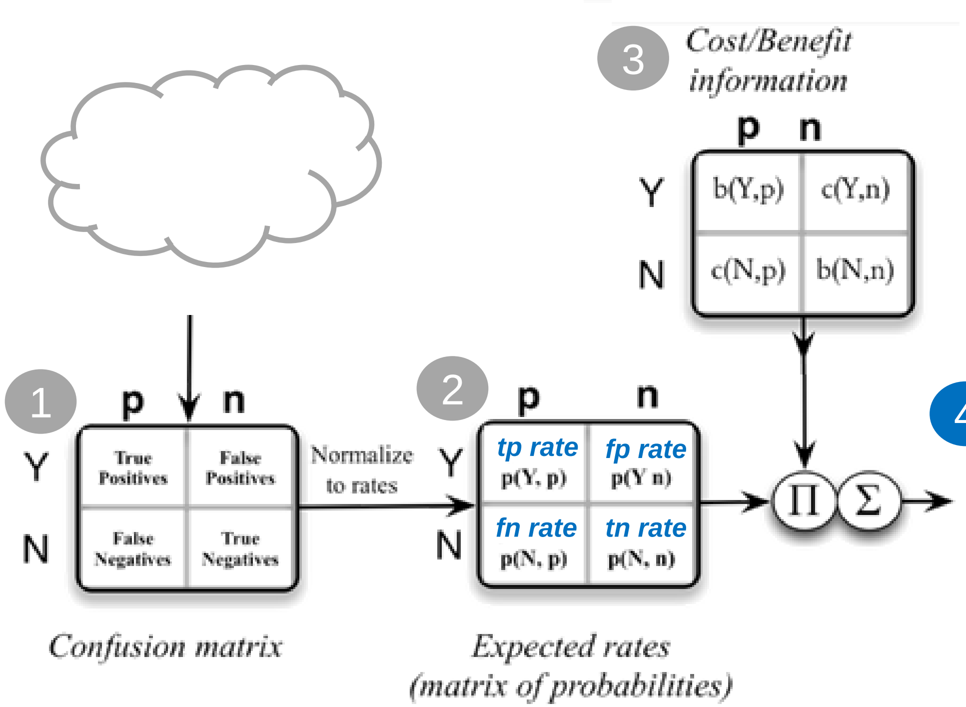
$$EP = p(p) \cdot [p(Y|p) \cdot b(Y,p) + p(N|p) \cdot c(N,p)] + p(n) \cdot [p(N|n) \cdot b(N,n) + p(Y|n) \cdot c(Y,n)]$$

The first component corresponds to the expected profit from the **positive examples**, whereas the second corresponds to the expected profit from the **negative examples**

We call:
p(Y|p) : true positive rate
p(N|p) : false negative rate
p(Y|n) : false positive rate
p(N|n) : true negative rate

Exercise – Expected value computation

Example alternative expression



3

Predicted \ Actual	p	n
Y	99	-1
N	0	0

1

Predicted \ Actual	p	n
Y	56	7
N	5	42

2

$$P = 61, \quad N = 49,$$
$$p(\mathbf{p}) = 0.55, \quad p(\mathbf{n}) = 0.45,$$
$$tp\ rate = 56/61 = 0.92, \quad fp\ rate = 7/49 = 0.14,$$
$$fn\ rate = 5/61 = 0.08 \quad tn\ rate = 42/49 = 0.86$$

4

$$EP = p(p) \cdot [p(Y|p) \cdot b(Y,p) + p(N|p) \cdot c(N,p)] + p(n) \cdot [p(N|n) \cdot b(N,n) + p(Y|n) \cdot c(Y,n)]$$

This expected value means that …

Todo for Wednesday

5 Min.

Ref.

Other evaluation metrics

Based on the entries of the confusion matrix, we can describe various evaluation metrics

Accuracy (count of correct decisions): $\frac{TP+TN}{P+N}$

		p	n
Predicted	Y	True positives	False positives
	N	False negatives	True negatives

True positive rate / Recall / Specificity : $\frac{TP}{TP+FN}$

		p	n
Predicted	Y	True positives	False positives
	N	False negatives	True negatives

False negative rate: $\frac{FN}{TP+FN}$

		p	n
Predicted	Y	True positives	False positives
	N	False negatives	True negatives

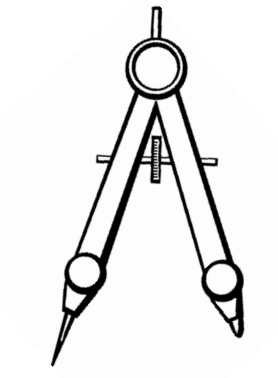
Sensitivity: $\frac{TN}{TN+FP}$

		p	n
Predicted	Y	True positives	False positives
	N	False negatives	True negatives

Precision (accuracy over the cases predicted to be positive): $\frac{TP}{TP+FP}$

		p	n
Predicted	Y	True positives	False positives
	N	False negatives	True negatives

F-measure (harmonic mean): $2 \cdot \frac{precision \cdot recall}{precision + recall}$



Ref.

Agenda

(1) Measuring accuracy

- Confusion matrix
- Unbalanced classes

(2) Expected Value

- Evaluate classifier use
- Frame classifier evaluation

(3) Evaluation and baseline performance

Ref.

Baseline performance (1/3)

Consider what would be a **reasonable baseline** against which to compare model performance

Demonstrate stakeholder that data mining has added value (or not)

What is the appropriate baseline for comparison?

Depends on the actual application

There are two basic tests that any weather forecast must pass to demonstrate its merit:
(1) It must do ***better than*** *what meteorologists call* ***persistence****: the assumption that the weather will be the same tomorrow (and the next day) as it was today.*
(2) It must also ***beat*** *climatology, the* ***long-term historical average*** *of conditions on a particular date in a particular area (not only dependent to time/seasonal effects).*

Ref.

Baseline performance (2/3)

Baseline performance for classification

- Compare to a completely random model (very easy)
- Implement a simple (but not simplistic) alternative model

Majority classifier = a naive classifier that always chooses the majority class of the training data set

- May be challenging to outperform: classification accuracy of 94%, but only 6% of the instances are positive
  → majority classifier also would have an accuracy of 94%!

Pitfall: don't be surprised that many models simply predict everything to be of the **majority class**

Maximizing simple prediction accuracy is usually not an appropriate goal

> Hint:
>
> DummyClassifier is a classifier that makes predictions using simple rules.This classifier is useful as a simple baseline to compare with other (real) classifiers.
>
> Strategies, e.g.,
> "most_frequent":
> always predicts the most frequent label in the training set.
> "uniform":
> generates predictions uniformly at random.
>
> (https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html)

Ref.

Baseline performance (3/3)

Further alternative:
how well does a simple “conditional” model perform?

- Conditional → prediction different based on the value of the features
- Just **use the most informative variable** for prediction
- Decision tree: build a tree with only one internal node (decision stump) → tree induction selects the single most informative feature to make a decision

Compare quality of models based on data sources

- Quantify the value of each source

Implement models that are based on **domain knowledge**

Ref.

Fragen?

- ✓ Measuring accuracy
  - ✓ Confusion matrix
  - ✓ Unbalanced classes
- ✓ A key analytical framework: Expected value
  - ✓ Evaluate classifier use
  - ✓ Frame classifier evaluation
- ✓ Evaluation and baseline performance

Todos for next Week

- Please remember:
  Gemäß Vorlesungsplanung ist für nächsten Mittwoch Project status update vorgesehen.
  Was heißt das nun eigentlich?
  - *Bitte jede Projektgruppe* ***bis Mittwoch 3.7****. einen kurzen „Zwischenbericht“ liefern*
  - *Formlos per E-Mail an bastian.amberg@fu-berlin.de , Betreff „BI Projektgruppe – Zwischenstand“*

    - *Aktueller Stand (z.B. durchgeführte Bearbeitungsschritte nach CRISP-DM)*
    - *Offene Punkte bzw. grobes weiteres Vorgehen*
    - ***kurz (d.h. ca. 4-5 Sätze)***
    - *Optionaler weiterer Punkt: Besteht Sprechstundenbedarf?*
      Sprechstunde dann in der Woche vom 8.7. bis 12.7.

- Please do the “Exercise – Expected value computation“
  Slide 23

Ref.

Recommended reading

What is a good model:

Provost, F., Fawcett, T.	Data Science for Business Chapter 7
Berthold et al.	Guide to Intelligent Data Analysis, Chapter 5

Further reading (beyond this class):

Provost, F., Fawcett, T.	Data Science for Business Chapter 8, Visualizing Model Performance (discusses graphical views of model behaviour)

Ref.

Business Intelligence – Interim summary

We have three goals. After this course:

- You know how to solve business problems by **data-analytic thinking**
- You know **tools** and ways of how to practically **implement** solution methods
- You have an overview about principles of **how to model and solve** upcoming **business problems.**

Main focus:

Data Warehousing / Data Engineering ► How to **store and access** huge amounts of data?

- DW Overview (incl. OLAP) — L02
- DW Modelling — L03, L04

Data Mining / Data Science ► How to **derive knowledge and profitable business action** out of (large) databases?

- DM Overview — L04, L05
- CRISP-DM
  - ► project understanding — L05
  - ► data understanding — L06, L07
  - ► data preparation — L08
  - ► modeling — L09, L10, L11, L12, ...
  - ► evaluation — L13 ...

Exkurs: Weiterführend siehe Schulz et al. 2020 DASC-PM