Next

- DM und KDD, Phasen
- Aufgabenstellungen des DM
- Wissensrepräsentation
- Entscheidungsbäume I – Repräsentation
- Entscheidungsbäume II – Lernen

- **Entscheidungsbäume III – Praktisch**
  - Ausblick ID3 zu C4.5
  - Ein Problem
  - Tools
  - RapidMiner

- Performance von Klassifikatoren

ID3-Algorithmus

- Wie beschrieben mit Gain als Auswahlkriterium des Attributes
- Entwickler Ross Quinlan (Sydney), exakt das Beispiel im Paper
- J. R. Quinlan: **Induction of decision trees,** Machine Learning, vol. 1, 1986.

- Kein Backtracking => nicht optimal, lokale Minima mgl.
- ID3 sehr erfolgreich, Zerlegung anhand des Informationsgewinns einleuchtend

- Kleine Probleme in der Praxis, deshalb von Quinlan weiterentwickelt zu
  - **C4.5** (J48 in Weka, DecisionTree in RapidMiner) [Quinlan1993]
  - C5.0 (kommerziell, unveröff. Verfahren, ähnlich C4.5 aber schneller, C-Code single-threaded version under GPL -> R-Package)

Probleme des ID3 – Ein Beispiel

Hinter einem Vorhang verbirgt sich ein Mitarbeiter oder Professor des Fachbereiches. Sie sollen herausbekommen, wer es ist.

Welches Attribut fragen Sie als erstes ab?

Problem des Information Gain (möglichst reine Knoten)

- **Attribute mit vielen Attributwerten** werden **stark bevorzugt**, denn sie enthalten sehr viel Information – bis zur Identifikation

- Andere Maße um das beste Attribut zu bestimmen
  - Gain ratio (möglichst reine Knoten, aber nicht zu viele)
  - Gini Index …

Probleme des ID3

- Attribute mit vielen Werten (Primärschlüssel, ID)
- Numerische Attribute nicht erlaubt
- Fehlende Werte
  - Bei der Baumkonstruktion
  - Beim Klassifizieren
- Overfitting: Trainingsdaten werden korrekt klassifiziert, neue Daten hingegen nicht
  - Prepruning (TDIDT rechtzeitig stoppen) vermeidet Overfitting

Diese Probleme sind gelöst im C4.5

- Operator ‚Decisiontree' im RapidMiner
- Details in Kap. 6.1 in [WF01]

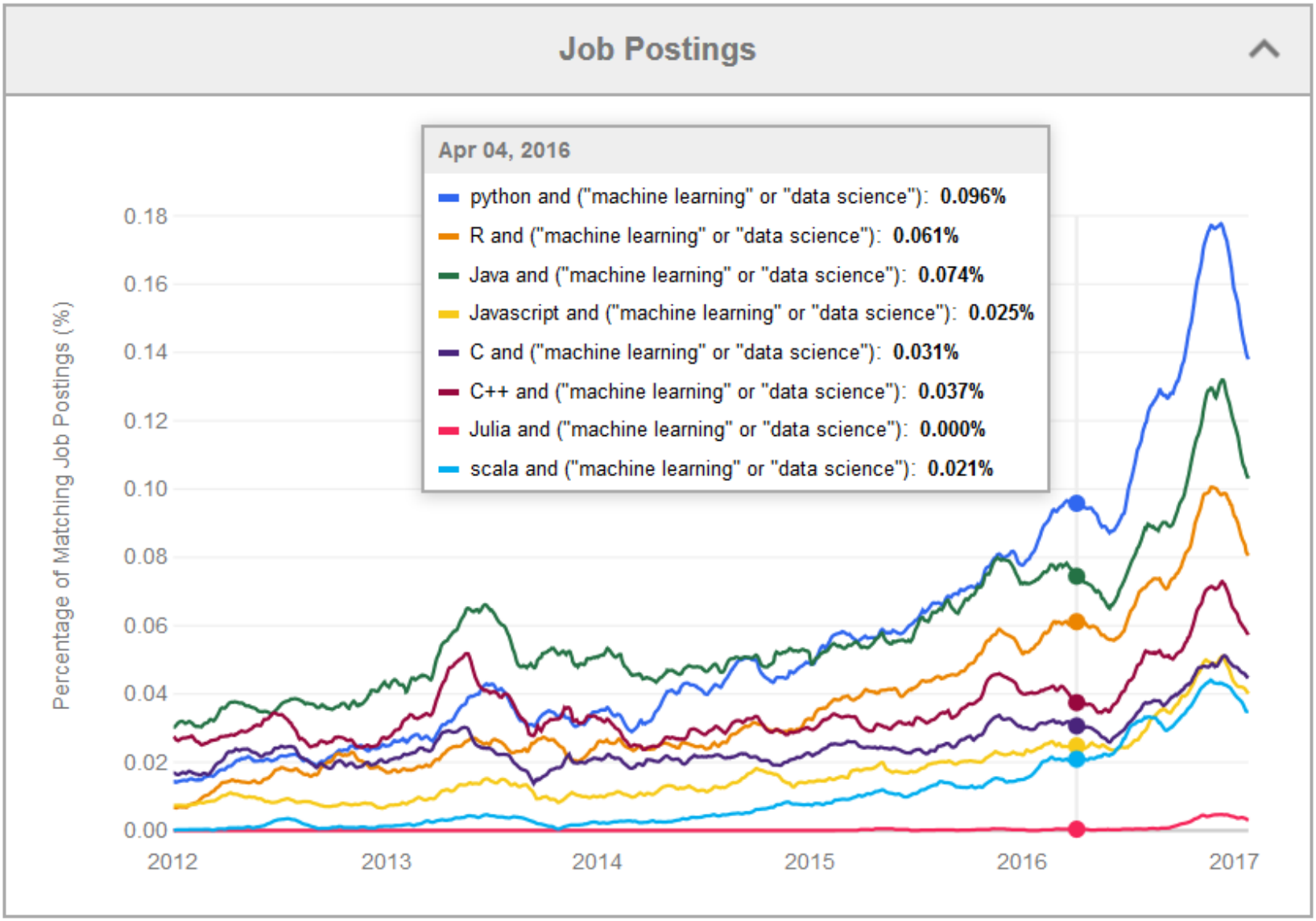
Tools und Sprachen

Kostenfreie Nutzung:

- oft Duale Lizenz: GPL + kommerzielle mit Service und Erweiterungen

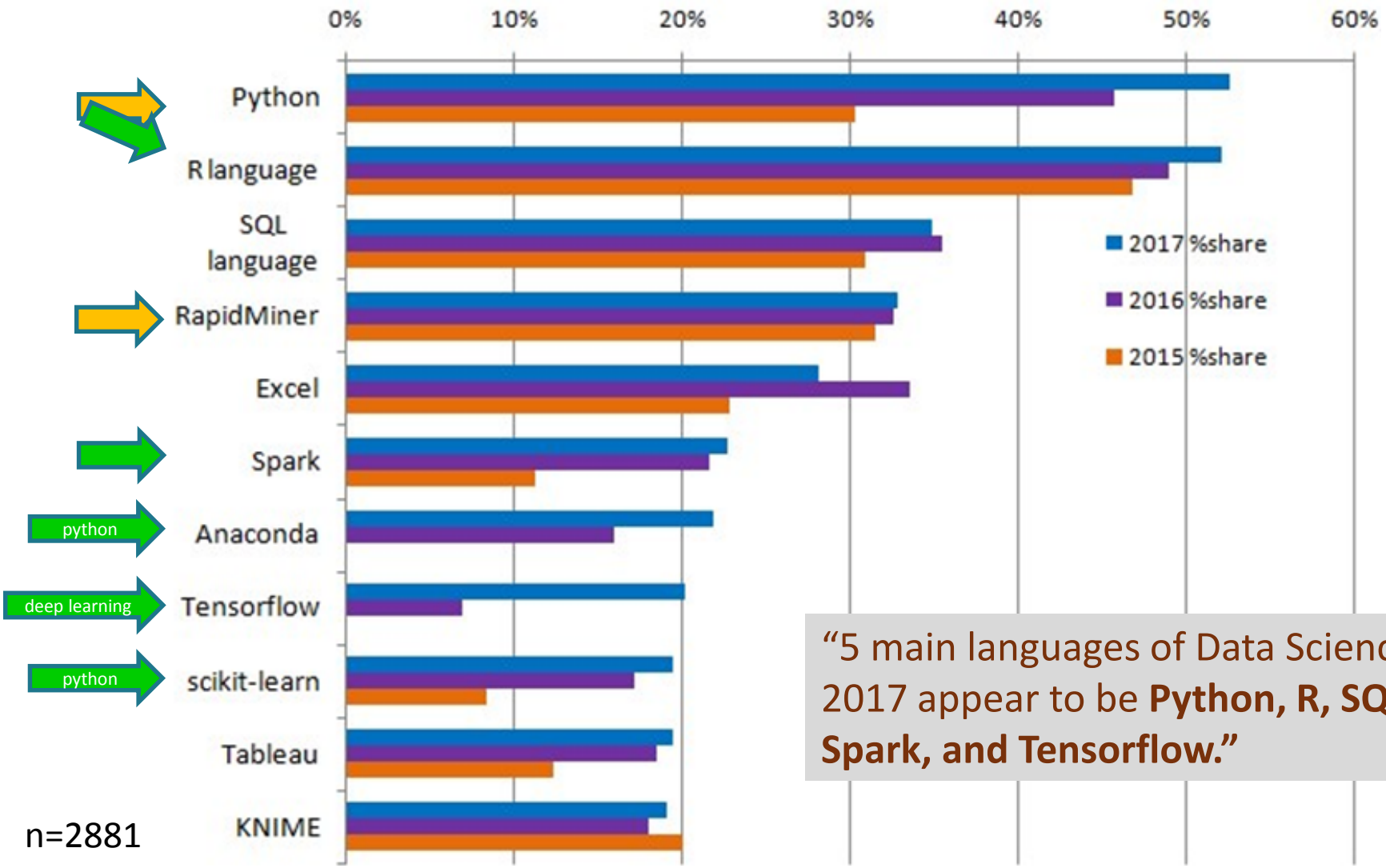
RapidMiner	Open + Closed Source, Uni. Dortmund + RapidMiner GmbH
R	GPL, Programmiersprache
Python	GPL, Programmiersprache
WEKA	GPL, Universität von Waikato (Neuseeland)
KNIME	GPL + dual, Uni. Konstanz + KNIME.com GmbH (Zürich)
Orange	GPL + Qt, University of Ljubljana (Slowenien)
Rattle	in R, Open Source, Fa. Togaware (Australien)

Kommerziell: bspw. SPSS Modeller



Jean-Francois Puget, http://www.kdnuggets.com/2017/01/most-popular-language-machine-learning-data-science.html,
Abfrage auf https://www.indeed.com/jobtrends

KDnuggets Analytics, Data Science, Machine Learning Software Poll, top tools share, 2015-2017

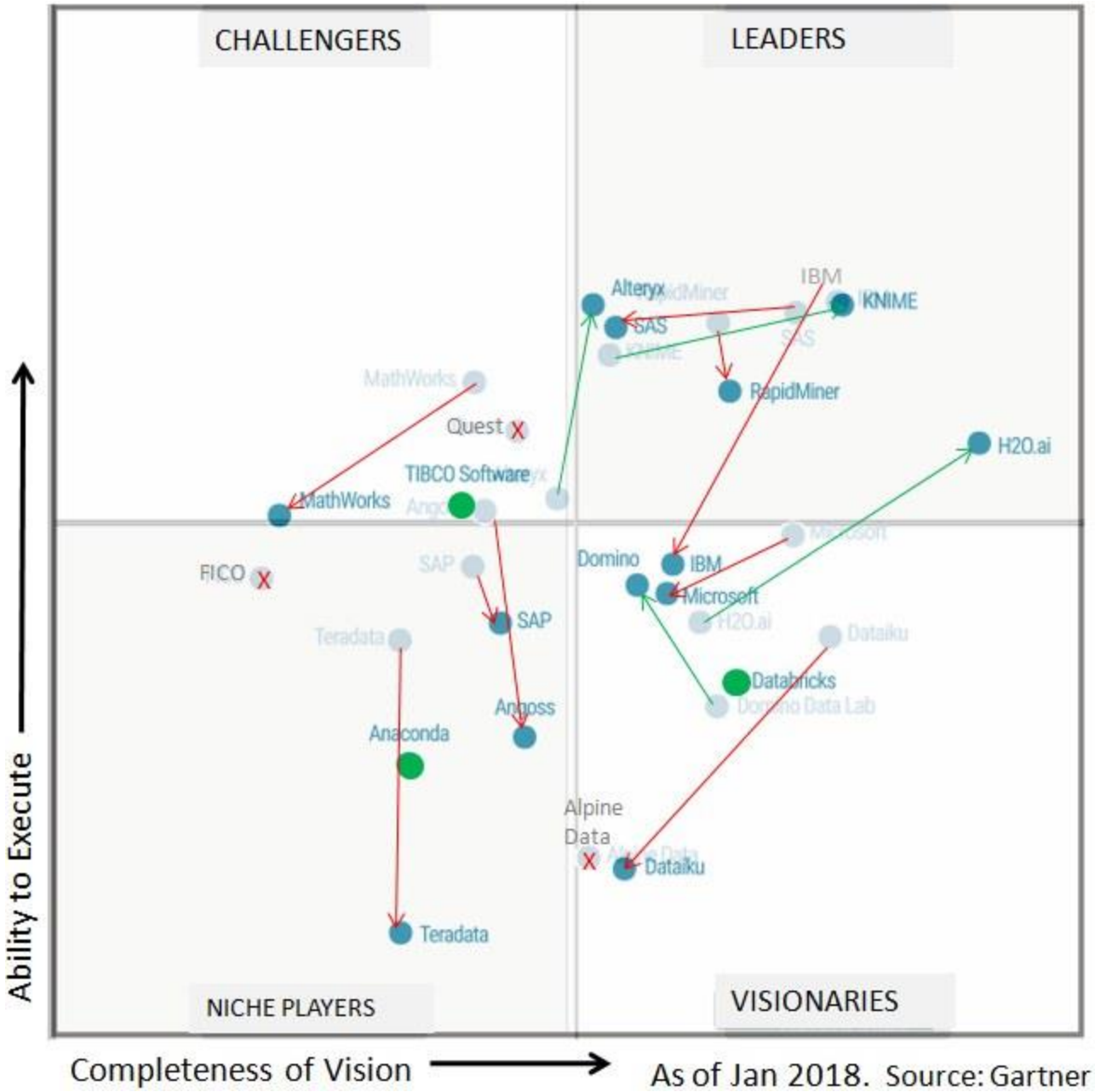


n=2881

"5 main languages of Data Science in 2017 appear to be **Python, R, SQL, Spark, and Tensorflow.**"

https://www.kdnuggets.com/2017/05/poll-analytics-data-science-machine-learning-software-leaders.html

https://www.kdnuggets.com/2018/02/gartner-2018-mq-data-science-machine-learning-changes.html

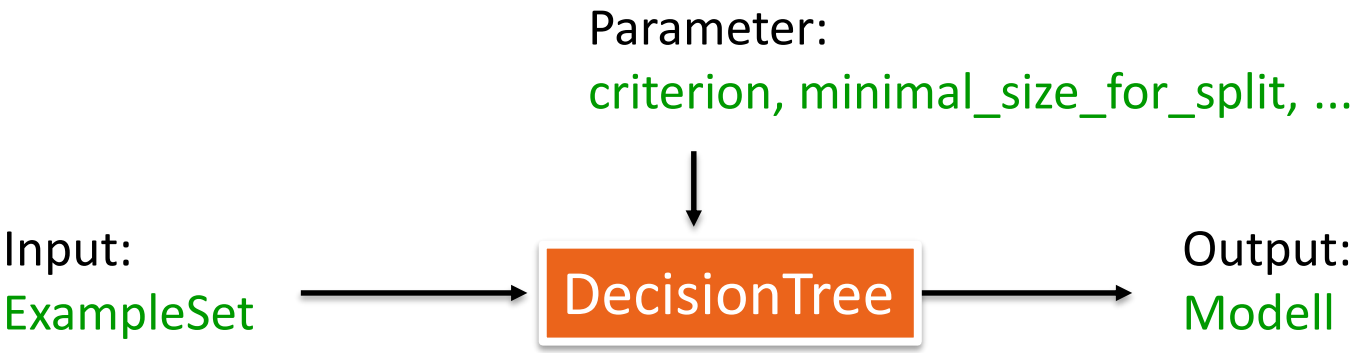


https://www.kdnuggets.com/2018/02/gartner-2018-mq-data-science-machine-learning-changes.html

RapidMiner Studio

- Data Mining Tool
- ehemals YALE (Yet Another Learning Environment)
- Open Source (Community Edition = Vorgängerversion), JAVA
- freie akademische Lizenz
- GUI oder offline
- DM-Prozess als Operatorgraph (XML-File)

  ==> **Visuelles Programmieren**
- Operator: erwartet Input und Parameter, liefert Output



Seminar-Angebote zu RapidMiner, Stand 2018

PRODUCTS & SOLUTIONS | RESOURCES | PRICING | PARTNERS | COMPANY

We are offering Virtual Instructor Led Training (VILT) for o...
RapidMiner & DataScience: Foundations, RapidMiner & DataScience: Advanced, Text and Web Mining with RapidMiner and RapidMiner Server: Deployment and Web Apps courses.

2 Day Classes

This version consists of 4 hours live online training on two consecutive days (4 hrs. each). The live portion is amended by an additional two times 4 hours of self-study and solving exercises. Depending on your time zone you will receive the live portion in the morning or the after noon of Monday and Tuesday or Wednesday and Thursday, respectively, so that you are left with the afternoons or the mornings of the next day for completion of the exercises. On the following Friday you can optionally attend a live Q&A session.

4 hr live sessions start at:

10am EST/4pm CET/7:30pm IST

Delivery Datasheet

Course fee is $1400 or 1250€ per course and student**.

RapidMiner Analyst Bootcamp

Are you looking for the fast-track option which offers you a great deal? Well then the 'RapidMiner Analyst Bootcamp' offering is what you are looking for. In this bundle you attend the 2-day classes of 'RapidMiner & DataScience' in one week and then you are granted access to complete the 'RapidMiner Analyst Certification' exam during the following 60 days.

2hr live sessions start at:

10am EST/4pm CET/7:30pm IST

Analyst Bootcamp Datasheet

The bundle which includes both 'RapidMiner & DataScience: Foundations', 'RapidMiner & DataScience: Advanced' and the 'RapidMiner Analyst Certification' exam is $2800 or 2500€ per student**.

Something urgent came up and you can't attend the full class? Well that will be no problem. You will be provided access to online materials covering all the course content and a Q&A online forum for 60 days starting from your first training day.

VILT Classes Schedule

Delivery Style	Course	Dates	Register
2 Day Class	Text & Web Mining with RapidMiner	Apr 16 & 17	in **USD / EUR**
2 Day Class	RapidMiner & DataScience: Foundations	May 14 & 15	in **USD / EUR**
2 Day Class	RapidMiner & DataScience: Advanced	May 16 & 17	in **USD / EUR**
Analyst Bootcamp Bundle	RapidMiner & DataScience + Certification	May 14 – 18	in **USD / EUR**
2 Day Class	Text & Web Mining with RapidMiner	on request	**Waiting List**
2 Day Class	RapidMiner Server: Deployment and Web Apps	on request	**Waiting List**
2 Day Class	RapidMiner & DataScience: Foundations	Jun 18 & 19	**Register**
2 Day Class	RapidMiner & DataScience: Advanced	Jun 20 & 21	**Register**
2 Day Class	RapidMiner & DataScience: Foundations	Jul 23 & 24	**Register**
2 Day Class	RapidMiner & DataScience: Advanced	Jul 25 & 26	**Register**
2 Day Class	RapidMiner & DataScience: Foundations	Aug 20 & 21	**Register**
2 Day Class	RapidMiner & DataScience: Advanced	Aug 22 & 23	**Register**
2 Day Class	RapidMiner & DataScience: Foundations	Sep 17 & 18	**Register**
2 Day Class	RapidMiner & DataScience: Advanced	Sep 19 & 20	**Register**

** If you are working for an official RapidMiner partner, in the academic field or if you are based in an emerging markets country (GDP (PPP) per capita < 25k USD as defined by IMF) you might be entitled for a discount. Please reach out to apply: Contact Us

None of the dates fits you calendar? Leave us your contact details and we will keep you update on next available dates!

Vorgängerversion als Open Source

https://github.com/rapidminer

```
19  package com.rapidminer.operator.learner.tree;
20
21  import com.rapidminer.example.ExampleSet;
22  import com.rapidminer.operator.OperatorCapability;
23  import com.rapidminer.operator.OperatorDescription;
24  import com.rapidminer.operator.OperatorException;
25  import com.rapidminer.parameter.ParameterType;
26
27  import java.util.LinkedList;
28  import java.util.List;
29
30
31  /**
32   * This operator learns decision trees without pruning using nominal
     attributes only. Decision trees
33   * are powerful classification methods which often can also easily be
     understood. This decision tree
34   * learner works similar to Quinlan's ID3.
35   *
36   * @author Ingo Mierswa
37   */
38  public class ID3Learner extends AbstractTreeLearner {
39
40      public ID3Learner(OperatorDescription description) {
41          super(description);
42      }
```

Splitkriterien in Entscheidungsbäumen:

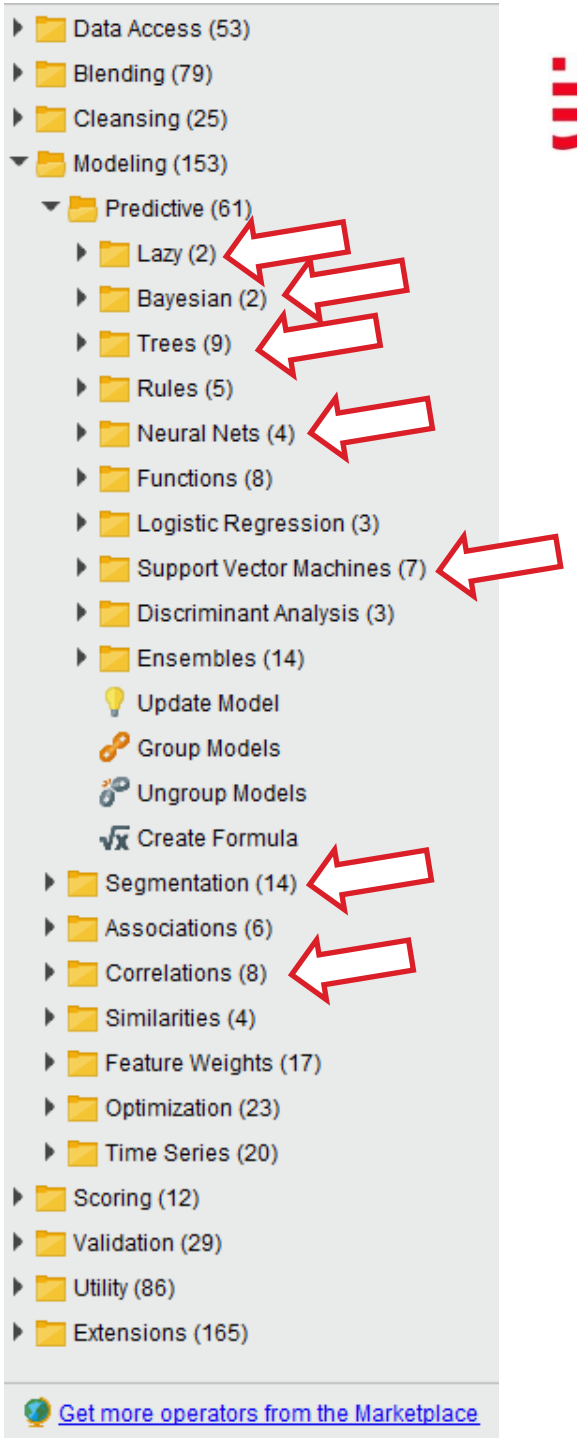
AbstractTreeLearner.java

```
23  import com.rapidminer.example.Attribute;
24  import com.rapidminer.example.ExampleSet;
25  import com.rapidminer.example.Statistics;
26  import com.rapidminer.operator.Model;
27  import com.rapidminer.operator.OperatorDescription;
28  import com.rapidminer.operator.OperatorException;
29  import com.rapidminer.operator.UserError;
30  import com.rapidminer.operator.learner.AbstractLearner;
31  import com.rapidminer.operator.learner.PredictionModel;
32  import com.rapidminer.operator.learner.tree.criterions.AbstractCriterion;
33  import com.rapidminer.operator.learner.tree.criterions.AccuracyCriterion;
34  import com.rapidminer.operator.learner.tree.criterions.Criterion;
35  import com.rapidminer.operator.learner.tree.criterions.GainRatioCriterion;
36  import com.rapidminer.operator.learner.tree.criterions.GiniIndexCriterion;
37  import com.rapidminer.operator.learner.tree.criterions.InfoGainCriterion;
38  import com.rapidminer.parameter.ParameterType;
39  import com.rapidminer.parameter.ParameterTypeDouble;
40  import com.rapidminer.parameter.ParameterTypeInt;
41  import com.rapidminer.parameter.ParameterTypeStringCategory;
42
```

RapidMiner Operatoren

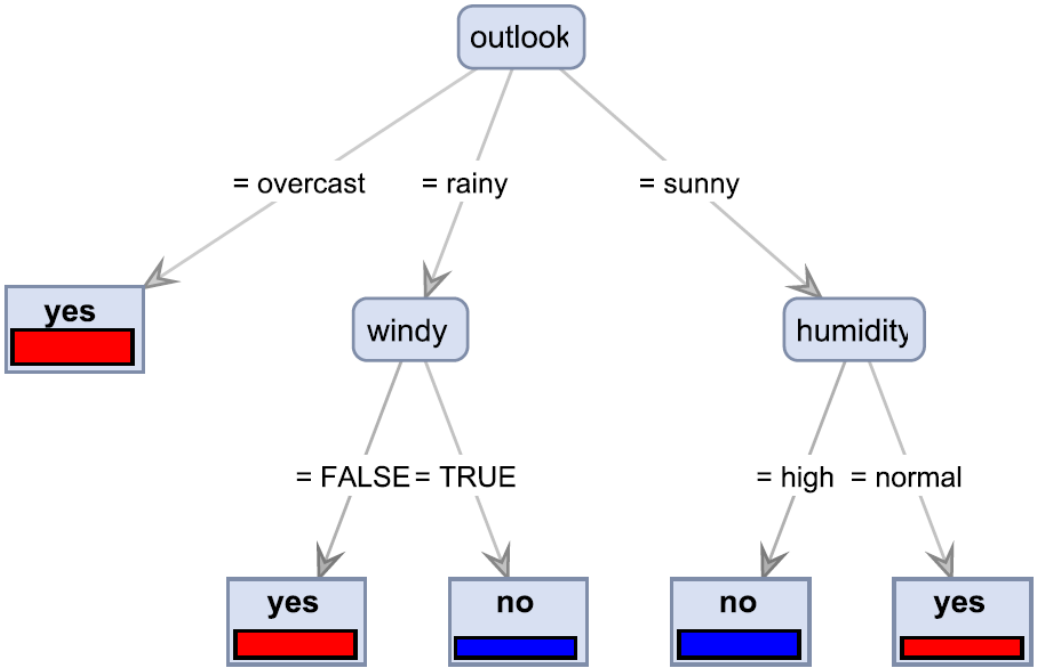
- Data preprocessing (Blending, Cleansing)
- Feature operators
- **Machine learning algorithms für numerische Prognose und Klassifikation: SVM, Decisiontrees, Lazy, Bayes, Rules, KNN, k-NN, ..., Metalernen**
- **Clustern**
- **Performance evaluation**
- **Visualization**
- Extensions: Process Mining, Text Processing ...
- In- and output

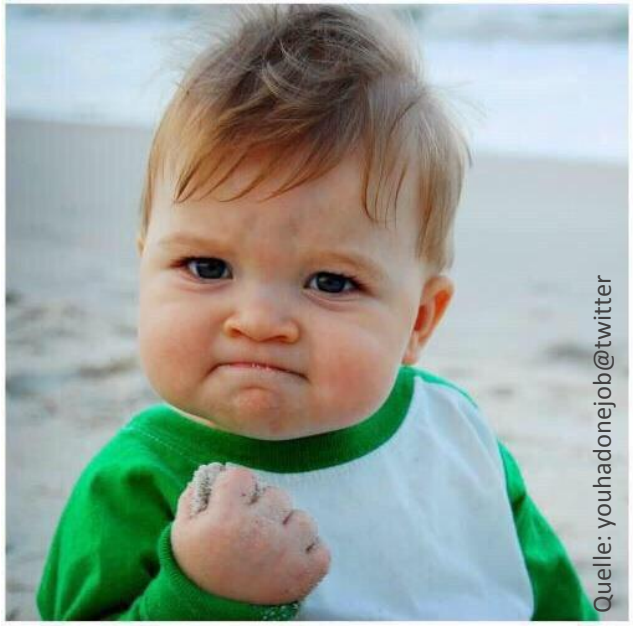
übersichtlich im PDF
„RapidMiner 8 Operator Reference Manual“



ID3 in RapidMiner ...

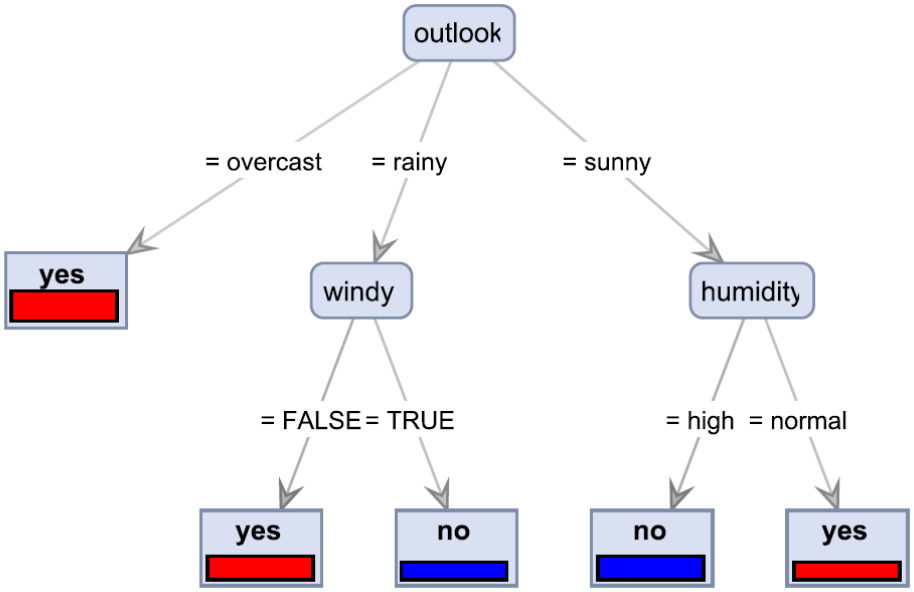
... erzeugt genau den uns schon bekannten Baum.





Wir kennen nun: DT, DT anwenden, DT erstellen, Tool
-> Applikation Spongebob: magic

Baum als Text




```
Tree

outlook = overcast: yes
outlook = rainy
| windy = FALSE: yes
| windy = TRUE: no
outlook = sunny
| humidity = high: no
| humidity = normal: yes
```

Next

- DM und KDD, Phasen
- Aufgabenstellungen des DM
- Wissensrepräsentation
- Entscheidungsbäume I – Repräsentation
- Entscheidungsbäume II – Lernen
- Entscheidungsbäume III – Praktisch
  - **Erinnerung an die Übung: RapidMiner, Scatterplot, ID3, Decisiontree**
- **Performance von Klassifikatoren**
- Ethik

In der Übung:

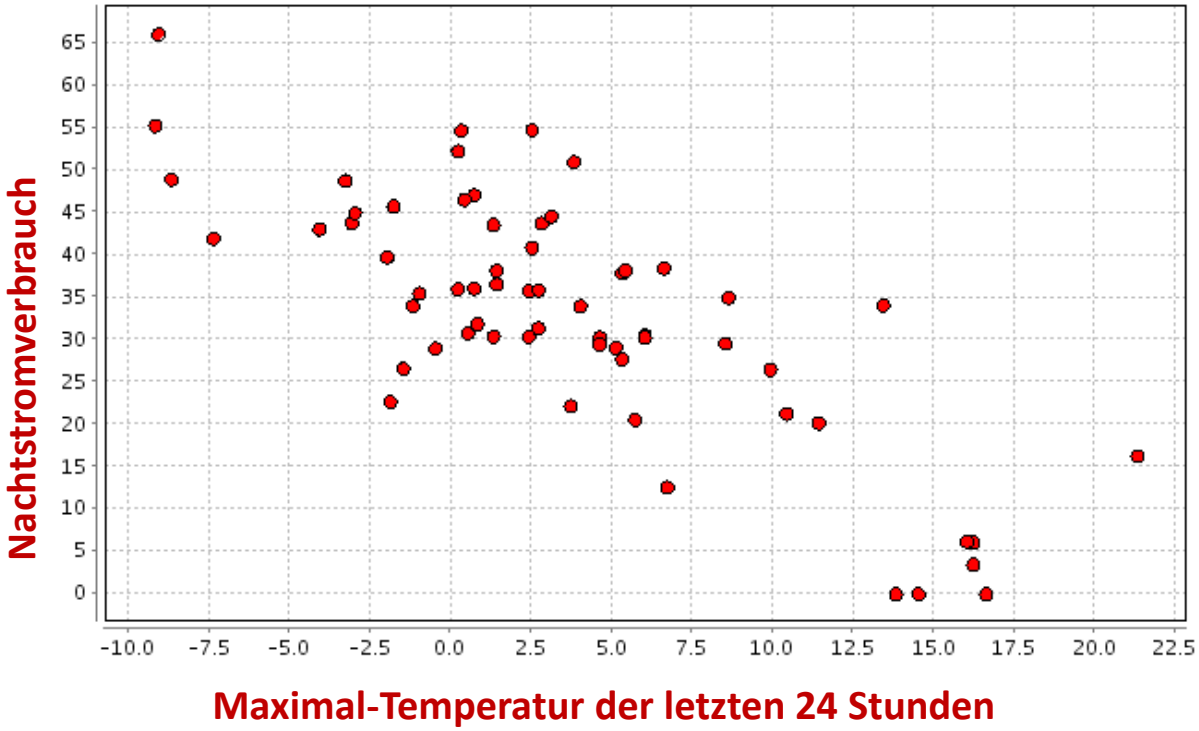
- Protokoll
- Datenmenge öffnen, visualisieren, Streudiagramm
- Baum mit ID3 aus Datenmenge erzeugen
- Modell anwenden auf unbekannten Datensatz
- Entscheidungsbaum der Tiere – Tiefe 20, Tiefe 4

Streudiagramm (Scatterplot)

- 2 Variablen als Achsen, ein Punkt pro Datensatz
- Überdeckung von Punkten durch **Jitter** vermeidbar
- Explorative Analyse: bivariate **Korrelationen**, Cluster, Kompaktheit, Streuung

Beispiel:
Wovon hängt der Nachtstrom-Verbrauch ab?

Ursache oder nur Korrelation?



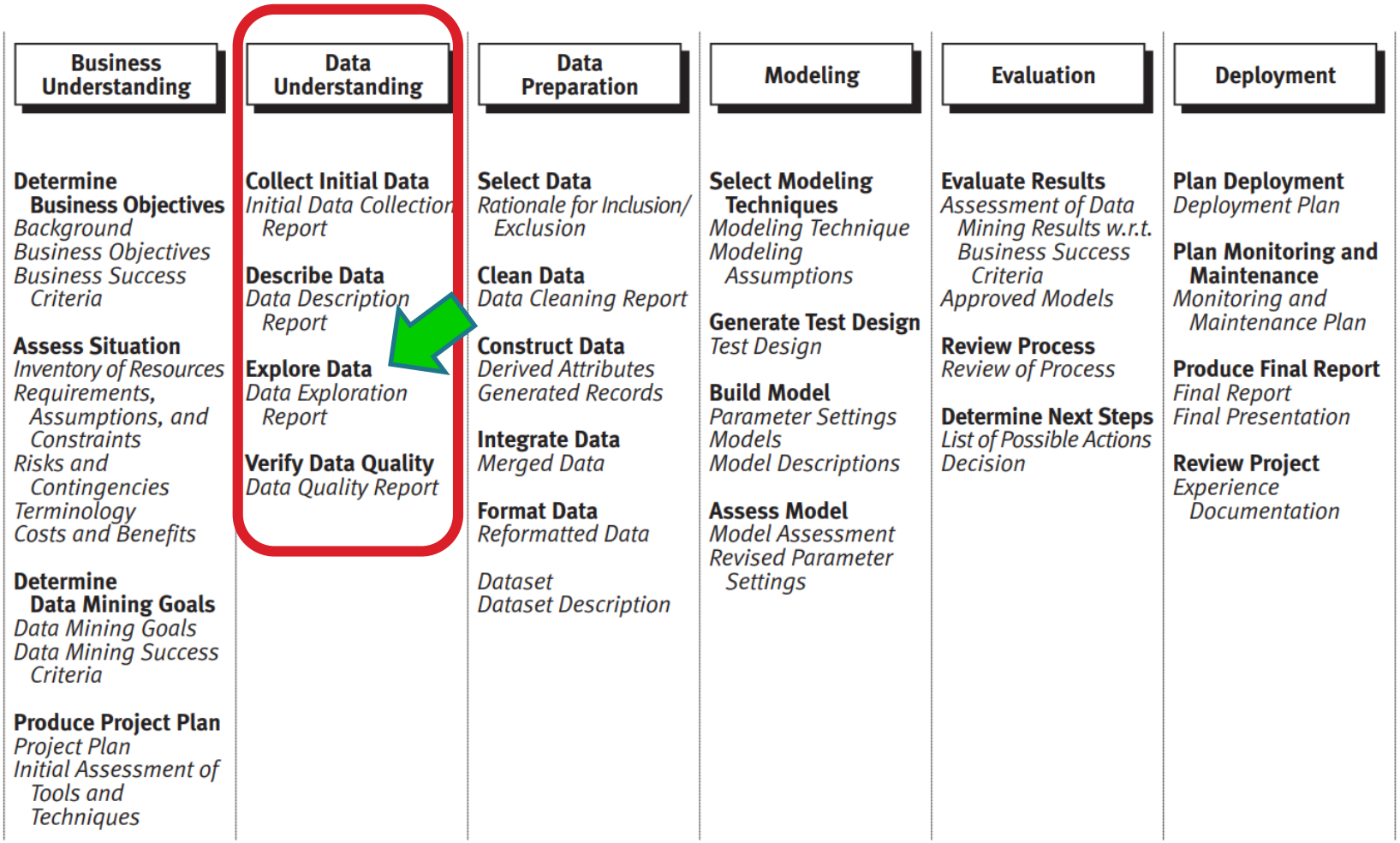
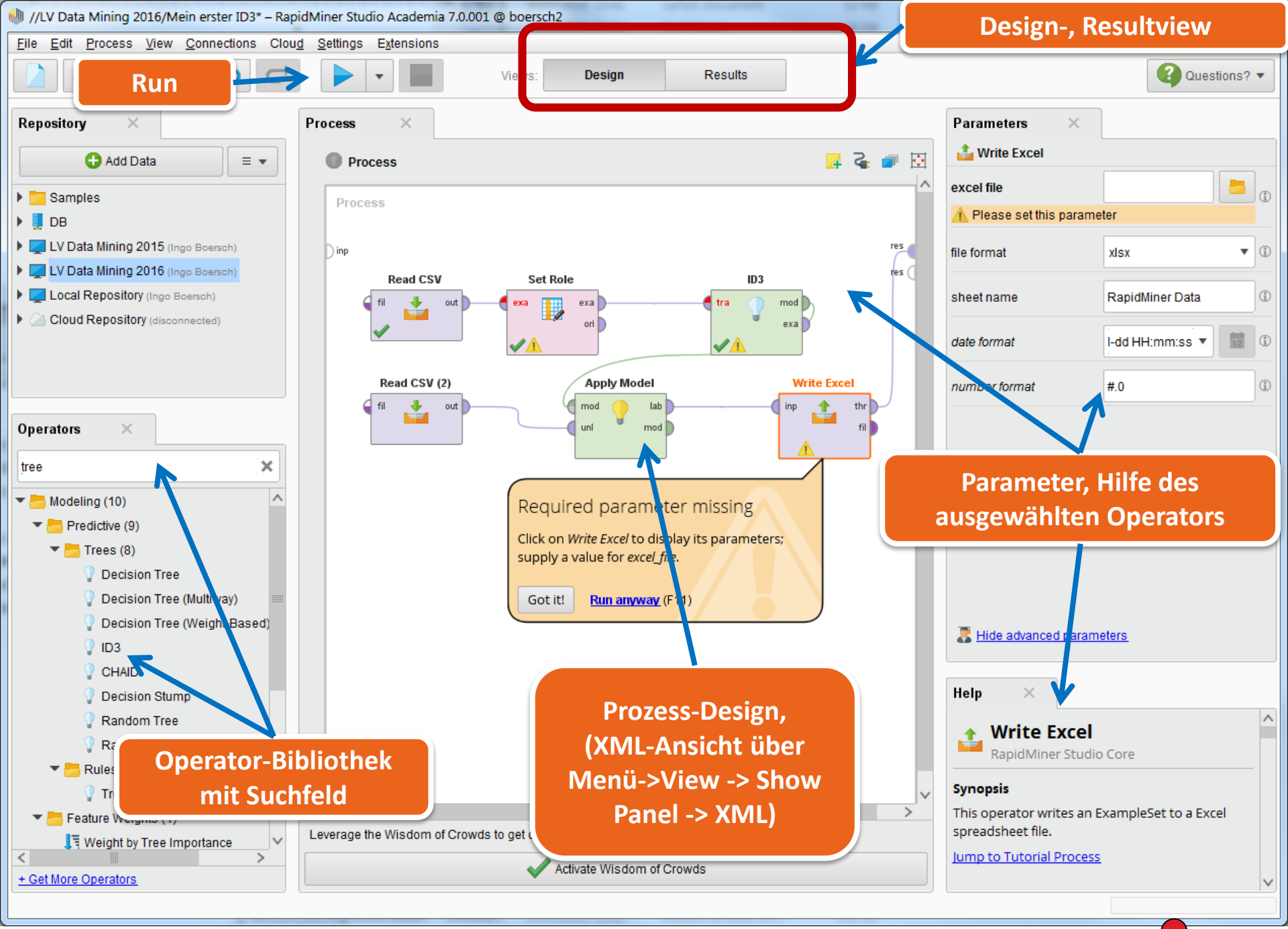
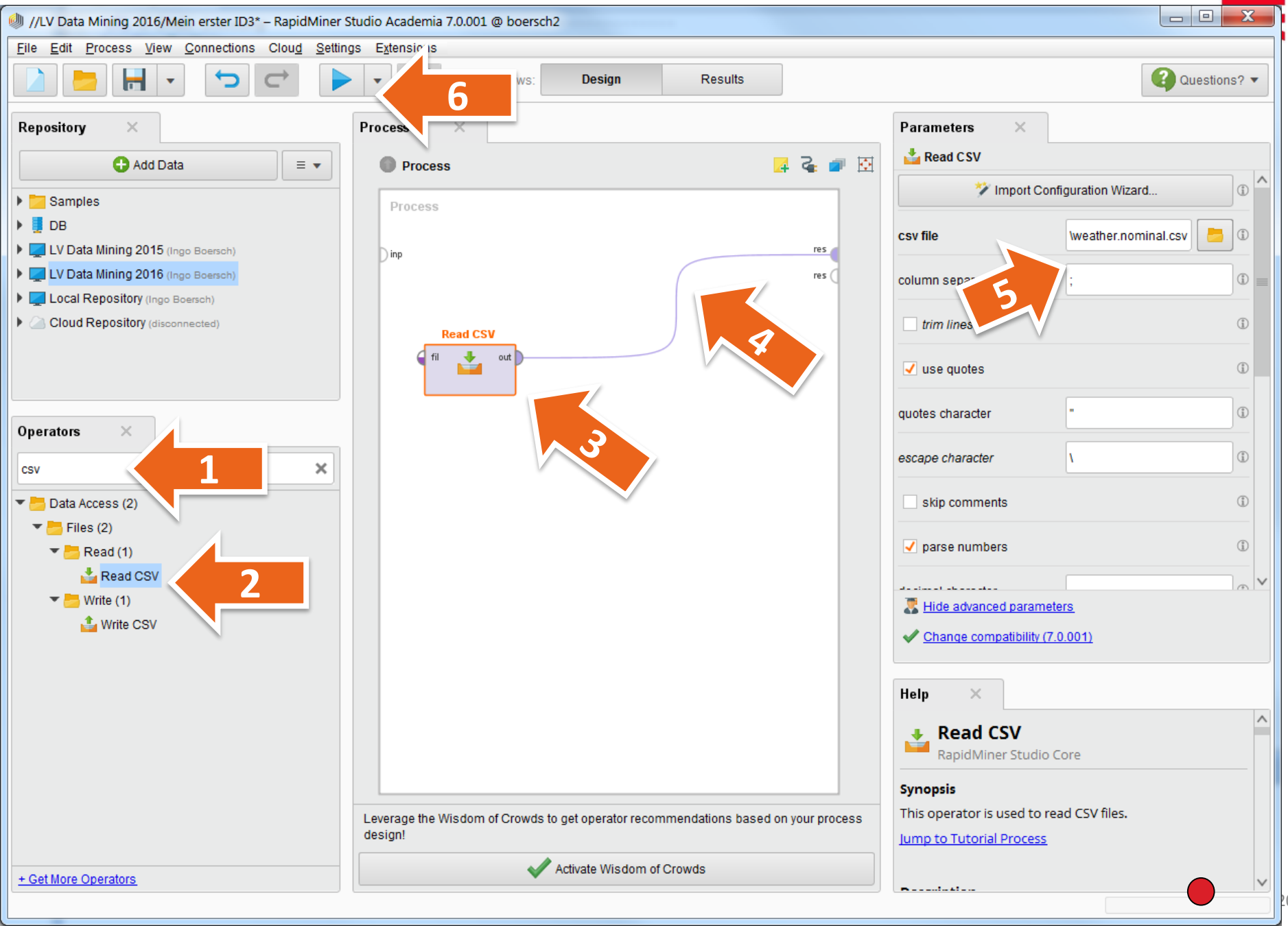
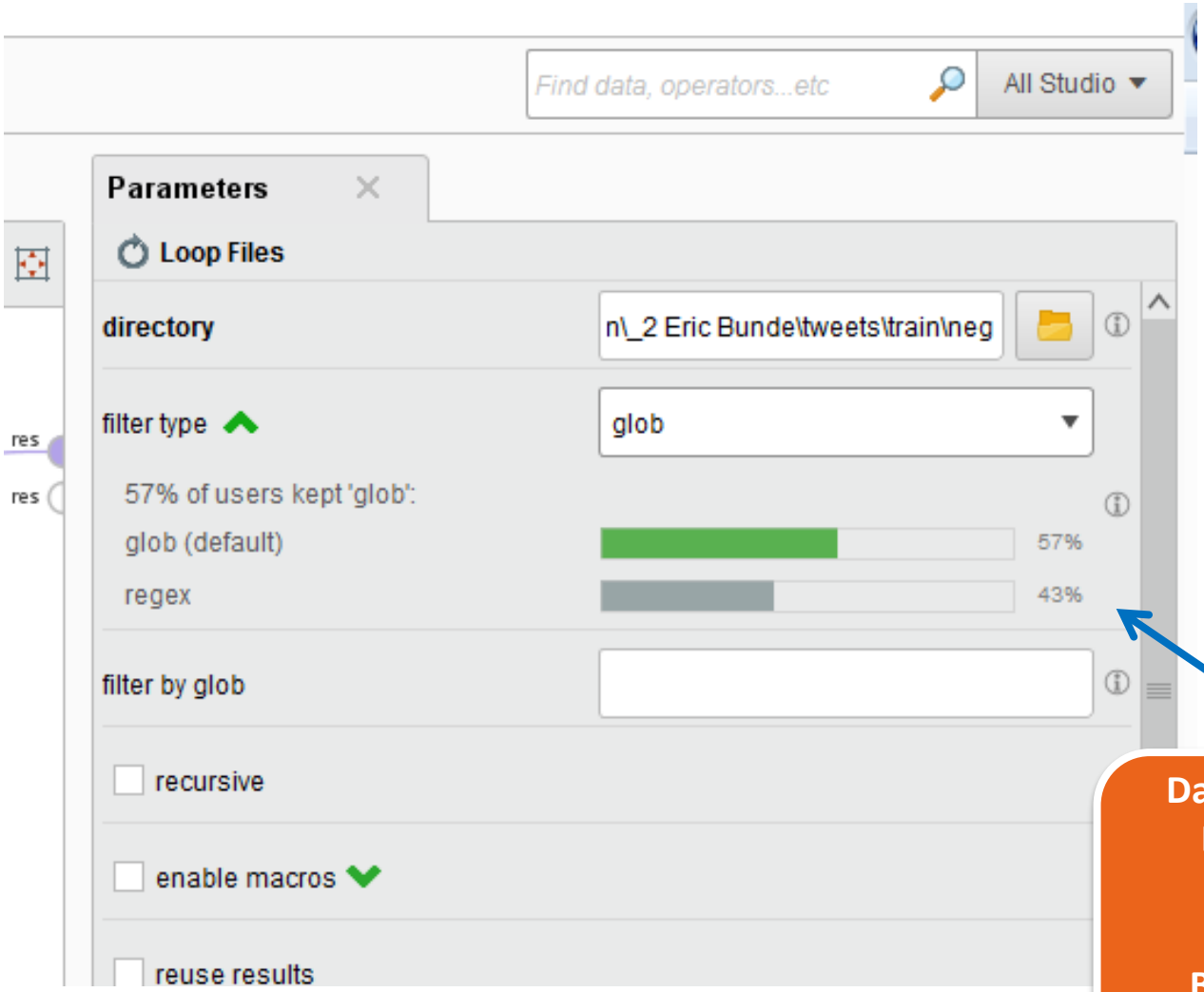


Figure 3: Generic tasks (bold) and outputs (italic) of the CRISP-DM reference model


//LV Data Mining 2016/Mein erster ID3* – RapidMiner Studio Academia 7.0.001 @ boersch2
File Edit Process View Connections Cloud Settings Extensions
Run
Design
Results
Design-, Resultview
Questions?
Repository
Add Data
Samples
DB
LV Data Mining 2015 (Ingo Boersch)
LV Data Mining 2016 (Ingo Boersch)
Local Repository (Ingo Boersch)
Cloud Repository (disconnected)
Operators
tree
Modeling (10)
Predictive (9)
Trees (8)
Decision Tree
Decision Tree (Multiway)
Decision Tree (Weight Based)
ID3
CHAID
Decision Stump
Random Tree
Rules
Feature Weights (1)
Weight by Tree Importance
+ Get More Operators
Operator-Bibliothek mit Suchfeld
Process
Process
Read CSV
Set Role
ID3
Read CSV (2)
Apply Model
Write Excel
Required parameter missing
Click on Write Excel to display its parameters; supply a value for excel_file.
Got it!
Run anyway (F11)
Leverage the Wisdom of Crowds to get
Activate Wisdom of Crowds
Prozess-Design, (XML-Ansicht über Menü->View -> Show Panel -> XML)
Parameters
Write Excel
excel file
Please set this parameter
file format
xlsx
sheet name
RapidMiner Data
date format
number format
#.0
Hide advanced parameters
Parameter, Hilfe des ausgewählten Operators
Help
Write Excel
RapidMiner Studio Core
Synopsis
This operator writes an ExampleSet to a Excel spreadsheet file.
Jump to Tutorial Process

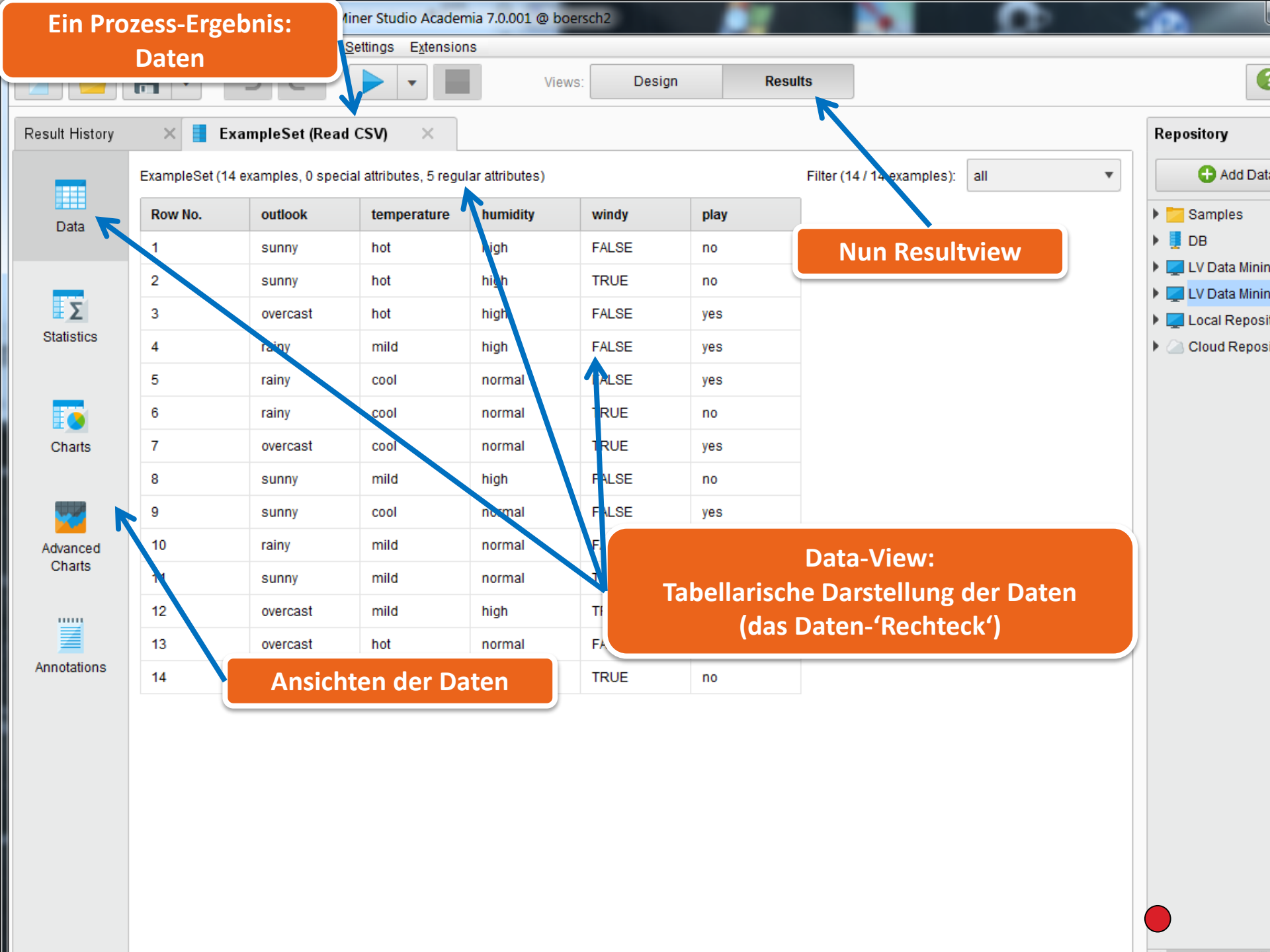
Einlesen des CSV-Files



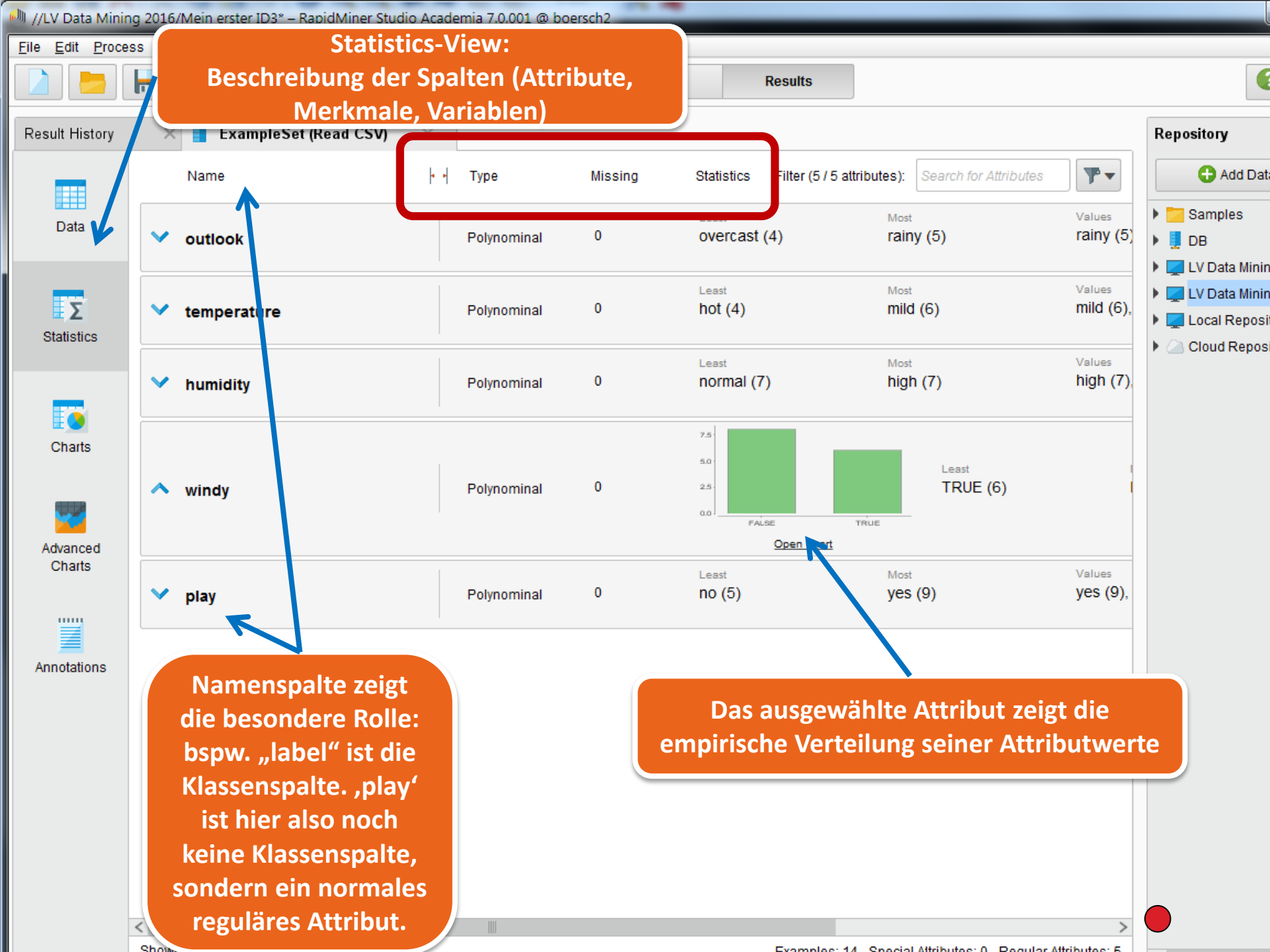


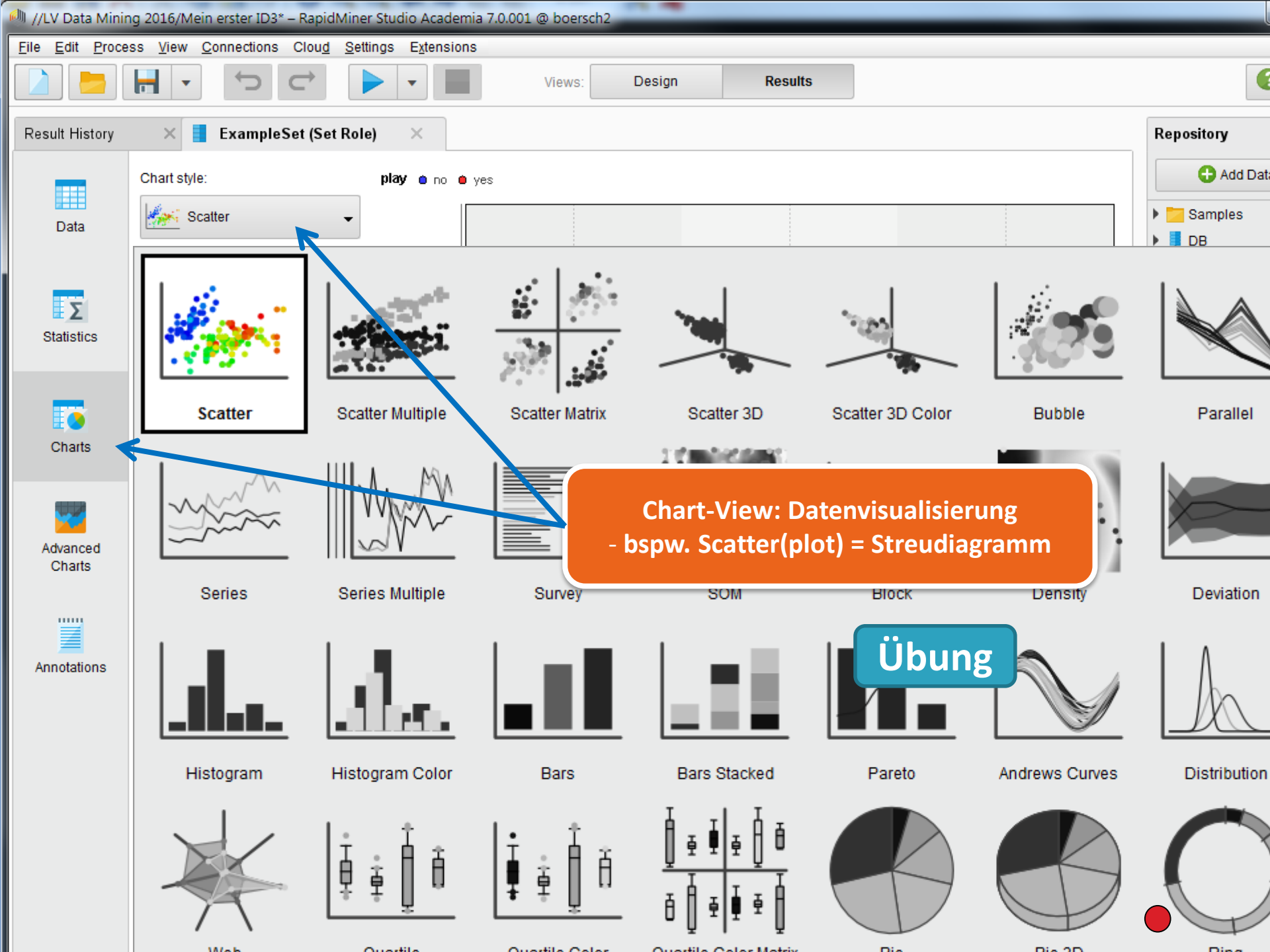
Daten sammeln für Entscheidungs-unterstützung

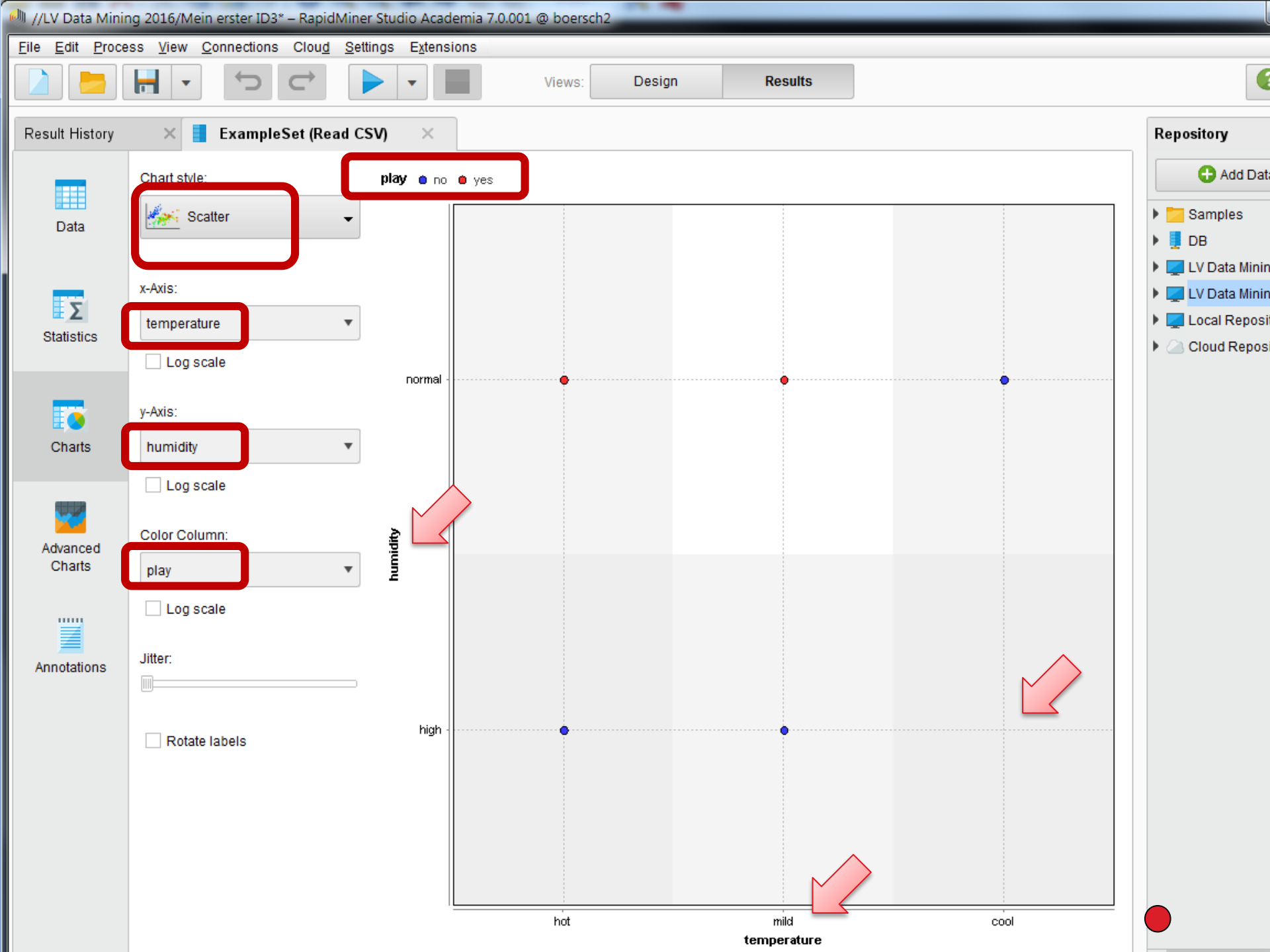
Bspw. auch in ID DIACOS: ICD->OPS

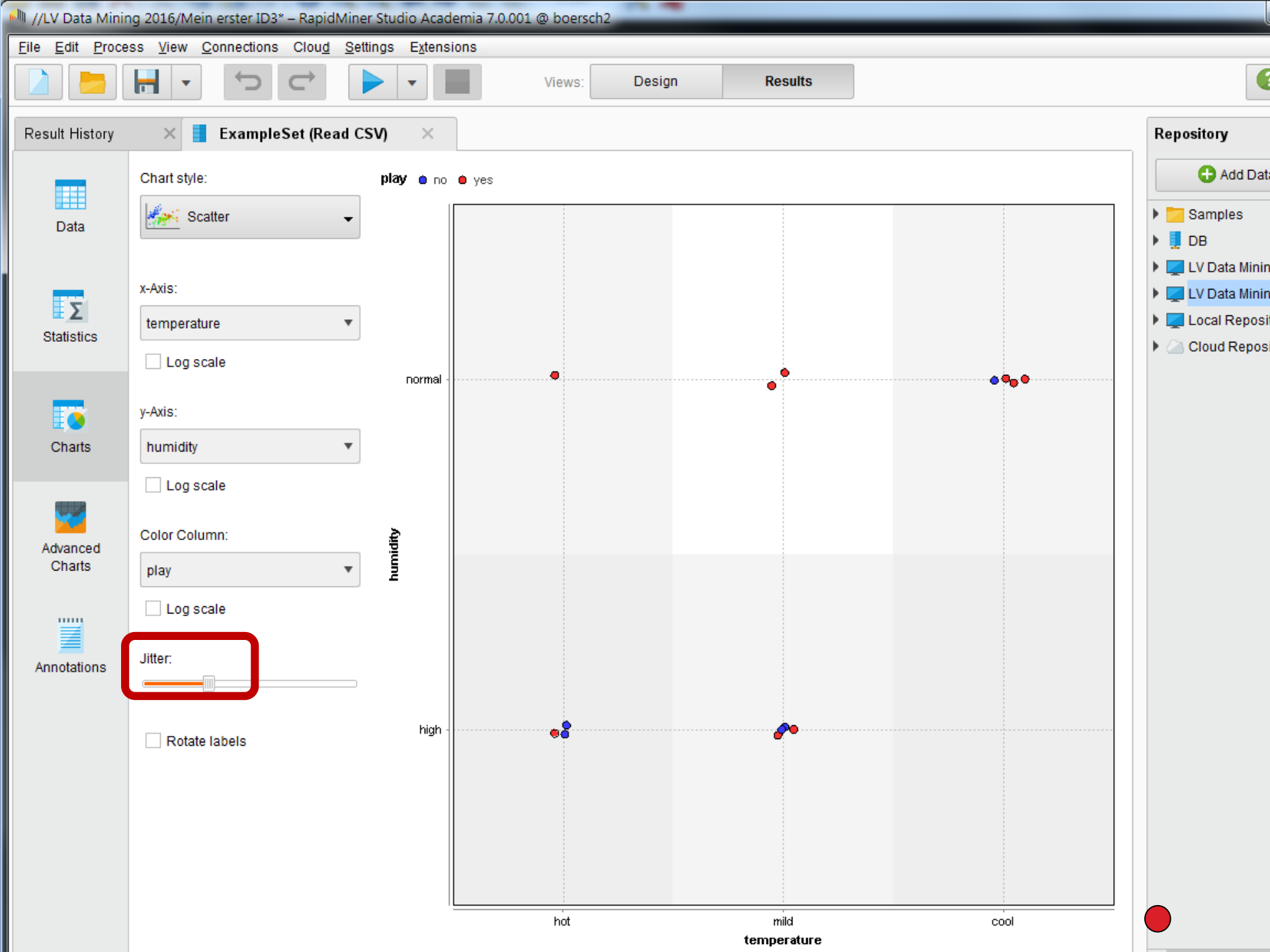


Row No.	outlook	temperature	humidity	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	F	
11	sunny	mild	normal	T	
12	overcast	mild	high	TI	
13	overcast	hot	normal	FA	
14				TRUE	no


//LV Data Mining 2016/Mein erster ID3* – RapidMiner Studio Academia 7.0.001 @ boersch2
File Edit Process
Results
Statistics-View:
Beschreibung der Spalten (Attribute, Merkmale, Variablen)
Result History
ExampleSet (Read CSV)
Data
Statistics
Charts
Advanced Charts
Annotations
Name
Type
Missing
Statistics
Filter (5 / 5 attributes):
Search for Attributes
outlook
Polynominal
0
Most
rainy (5)
Values
rainy (5)
temperature
Polynominal
0
Least
hot (4)
Most
mild (6)
Values
mild (6),
humidity
Polynominal
0
Least
normal (7)
Most
high (7)
Values
high (7),
windy
Polynominal
0
Least
TRUE (6)
FALSE
TRUE
Open Chart
play
Polynominal
0
Least
no (5)
Most
yes (9)
Values
yes (9),
Repository
Add Data
Samples
DB
LV Data Minin
LV Data Minin
Local Reposit
Cloud Reposi
Namenspalte zeigt die besondere Rolle: bspw. „label“ ist die Klassenspalte. ‚play‘ ist hier also noch keine Klassenspalte, sondern ein normales reguläres Attribut.
Das ausgewählte Attribut zeigt die empirische Verteilung seiner Attributwerte


//LV Data Mining 2016/Mein erster ID3* – RapidMiner Studio Academia 7.0.001 @ boersch2
File Edit Process View Connections Cloud Settings Extensions
Views: Design Results
Result History
ExampleSet (Set Role)
Repository
Add Data
Samples
DB
Data
Statistics
Charts
Advanced Charts
Annotations
Chart style:
Scatter
play no yes
Scatter
Scatter Multiple
Scatter Matrix
Scatter 3D
Scatter 3D Color
Bubble
Parallel
Series
Series Multiple
Survey
SOM
Block
Density
Deviation
Chart-View: Datenvisualisierung
- bspw. Scatter(plot) = Streudiagramm
Übung
Histogram
Histogram Color
Bars
Bars Stacked
Pareto
Andrews Curves
Distribution
Web
Quartile
Quartile Color
Quartile Color Matrix
Pie
Pie 3D
Ring


//LV Data Mining 2016/Mein erster ID3* – RapidMiner Studio Academia 7.0.001 @ boersch2
File Edit Process View Connections Cloud Settings Extensions
Views: Design Results
Result History
ExampleSet (Read CSV)
Data
Statistics
Charts
Advanced Charts
Annotations
Chart style:
Scatter
x-Axis:
temperature
Log scale
y-Axis:
humidity
Log scale
Color Column:
play
Log scale
Jitter:
Rotate labels
play no yes
normal
high
humidity
hot
mild
cool
temperature
Repository
Add Data
Samples
DB
LV Data Minin
LV Data Minin
Local Reposit
Cloud Reposi


//LV Data Mining 2016/Mein erster ID3* – RapidMiner Studio Academia 7.0.001 @ boersch2
File Edit Process View Connections Cloud Settings Extensions
Views: Design Results
Result History
ExampleSet (Read CSV)
Repository
Add Data
Samples
DB
LV Data Minin
LV Data Minin
Local Reposi
Cloud Reposi
Data
Statistics
Charts
Advanced Charts
Annotations
Chart style:
Scatter
x-Axis:
temperature
Log scale
y-Axis:
humidity
Log scale
Color Column:
play
Log scale
Jitter:
Rotate labels
play no yes
normal
high
humidity
hot
mild
cool
temperature

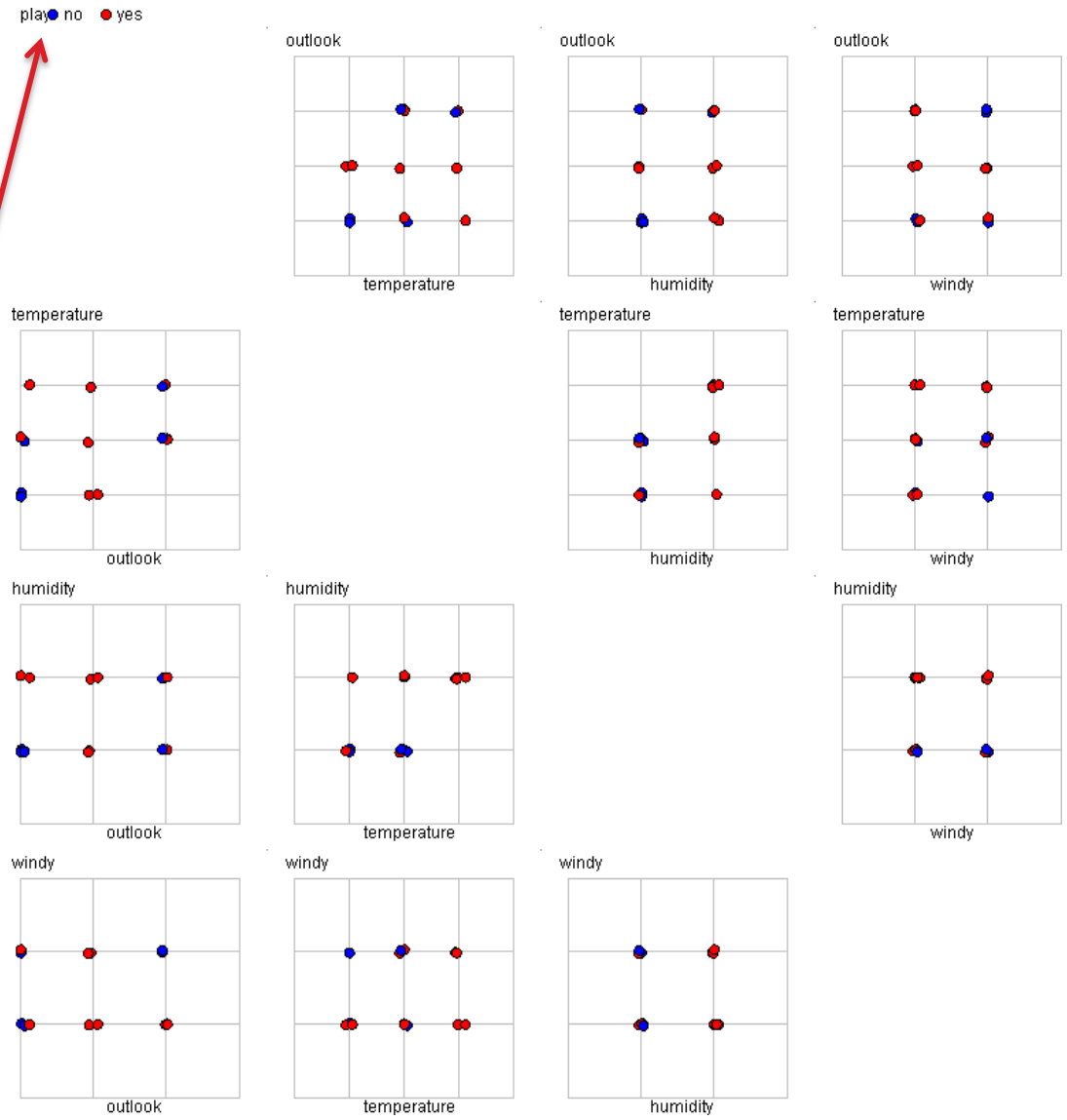
Matrix aller Streudiagramme

- Auch Scatterplotmatrix (**SPLOM**)

k Variablen ->
k x k - Matrix

Matrixelemente:
Streudiagramme

Einfärbung (blau, rot)
durch Zielklasse



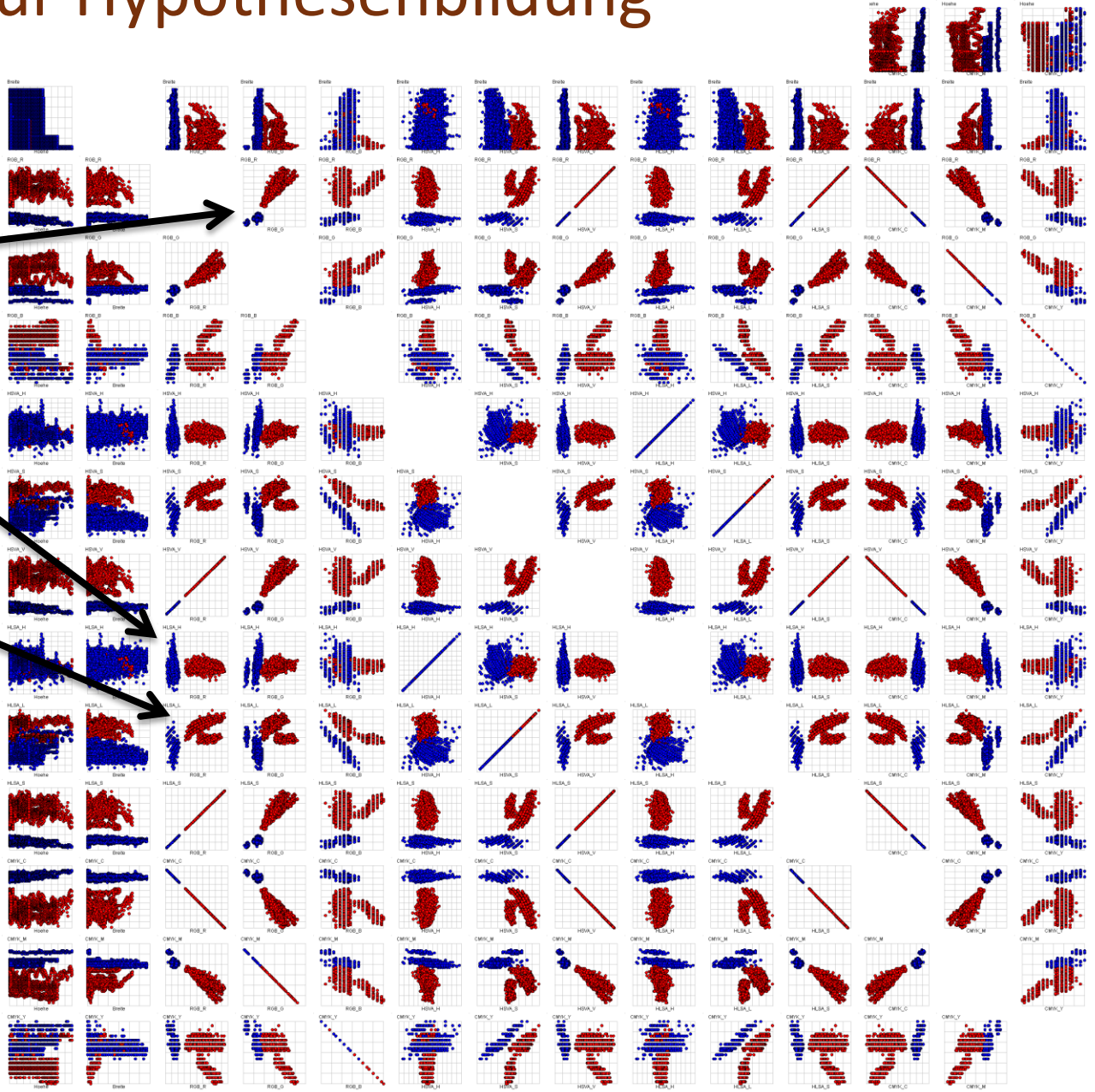
Reale SPLOM für Hypothesenbildung

Welche Merkmale sind abhängig?

Welche Merkmale eignen sich für die Klassifizierung?

Cluster

Visual Analytics (Sonderheft Informatikspektrum , Dezember 2010)



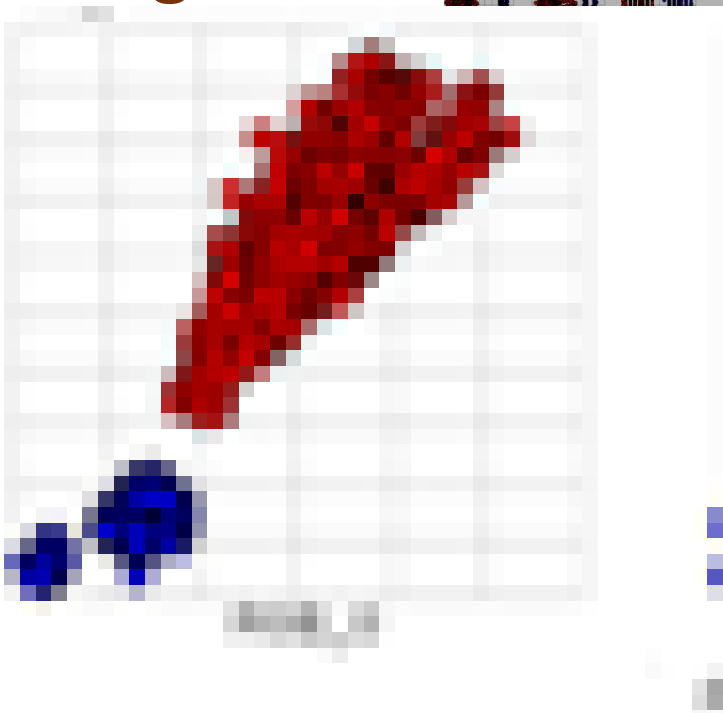
Reale SPLOM für Hypothesenbildung

Welche Merkmale sind abhängig?

Welche Merkmale eignen sich für die Klassifizierung?

Cluster

Visual Analytics (Sonderheft Informatikspektrum , Dezember 2010)



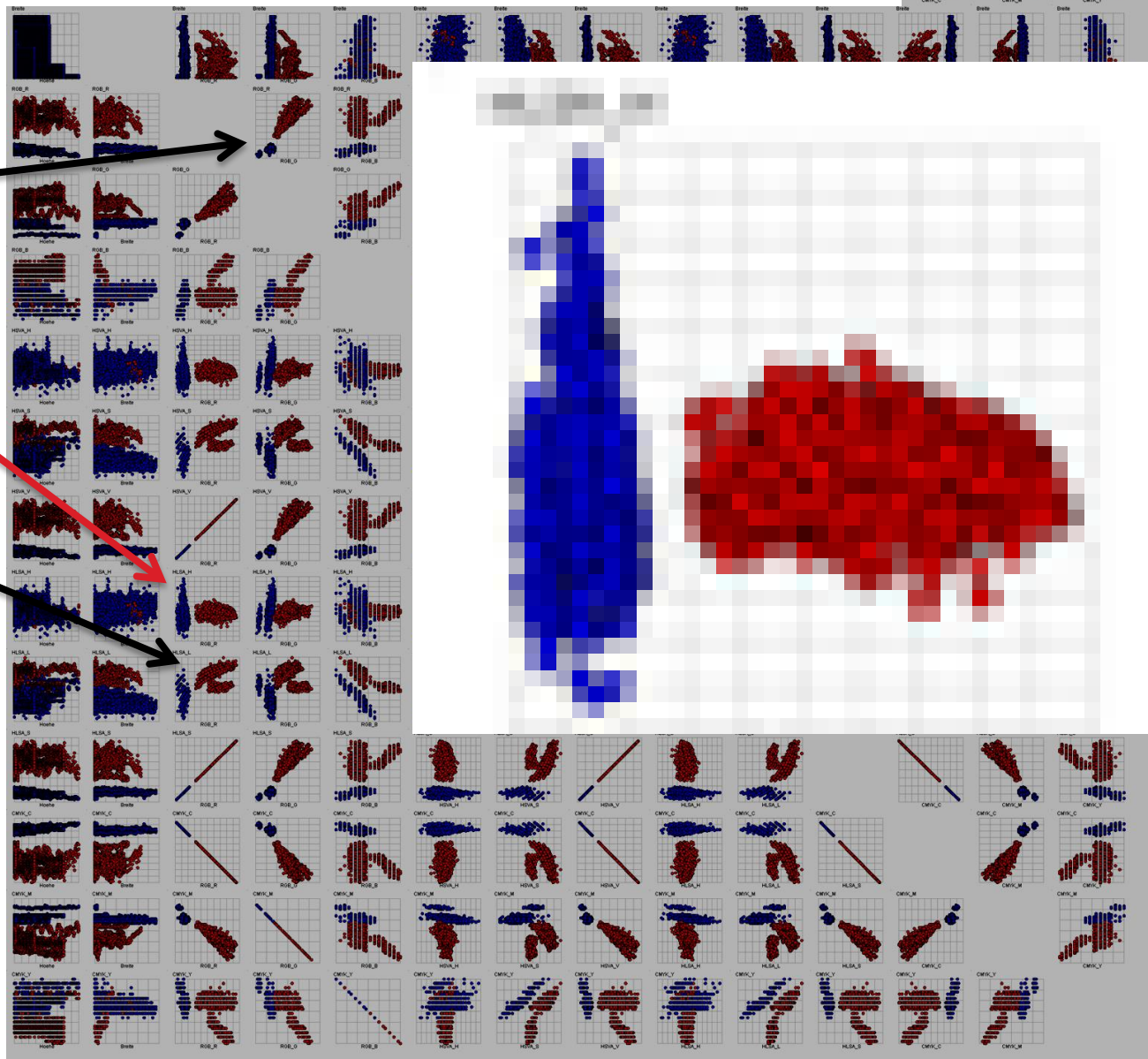
Reale SPLOM für Hypothesenbildung

Welche Merkmale sind abhängig?

Welche Merkmale eignen sich für die Klassifizierung?

Cluster

Visual Analytics (Sonderheft Informatikspektrum , Dezember 2010)



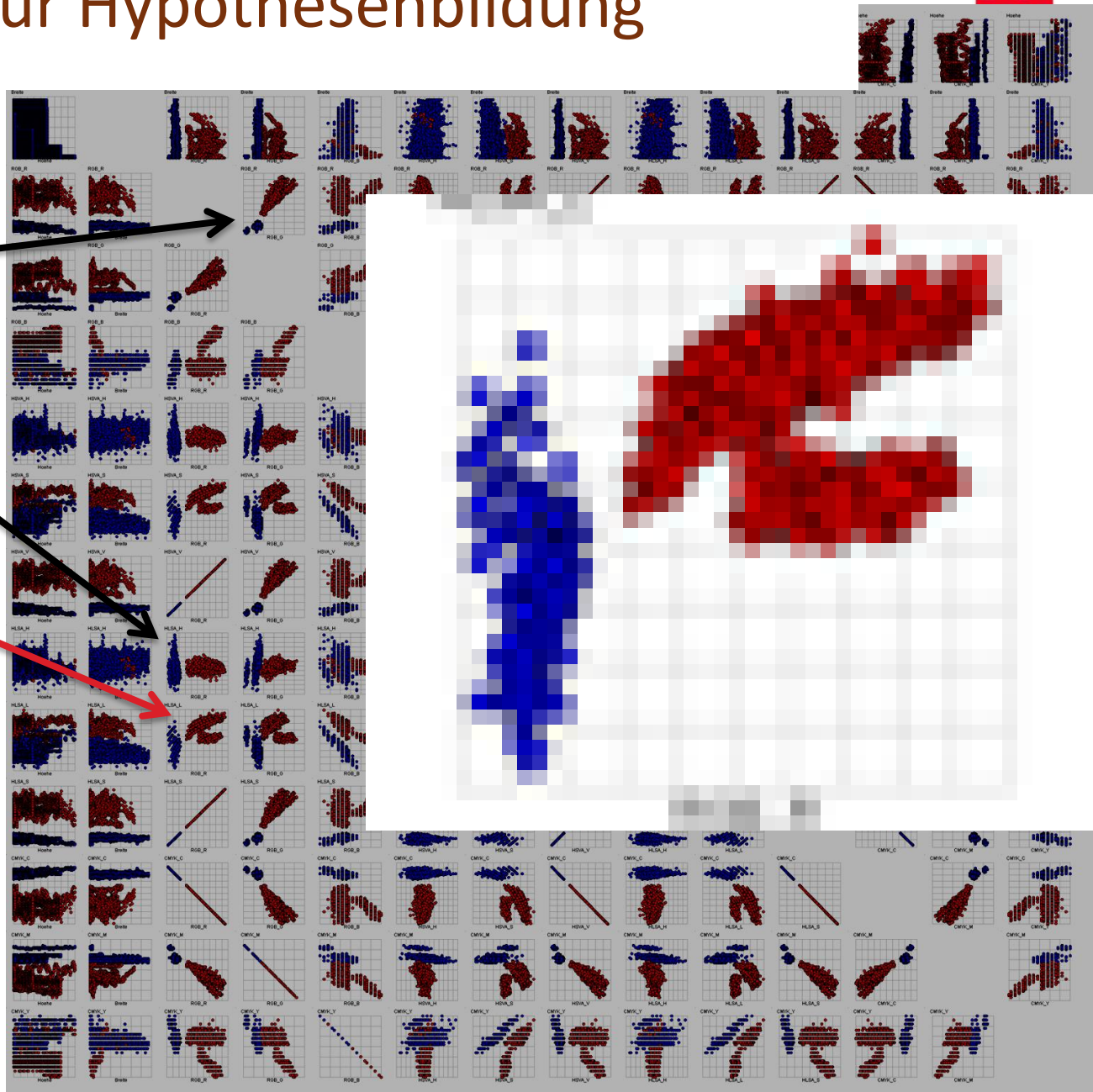
Reale SPLOM für Hypothesenbildung

Welche Merkmale sind abhängig?

Welche Merkmale eignen sich für die Klassifizierung?

Cluster

Visual Analytics (Sonderheft Informatikspektrum , Dezember 2010)



Spongebob und Crabs

- Lernen von Objektbeschreibungen durch Zeigen von Beispielen

- Diplomarbeit Benjamin Kieper:
  Entwurf und Implementierung einer Anwendung zum dialogbasierten, überwachten Lernen von Objektmodellen aus Bildern

Einführung

WAS ist Data Mining?

- Data Mining (DM) und Knowledge Discovery in Databases (KDD)
- Aufgabenstellungen des DM, viele Beispiele
- Wissensrepräsentation

WIE funktioniert es?

- Entscheidungsbäume:
  - Repräsentation,
  - Lernen / Konstruieren
  - Praktisch

WIE gut sind die Modelle?

- Performance von Klassifikatoren

Einschätzung des Gelernten

Fragen in der Praxis:

1. Welche Lernmethoden sollen auf welches Problem angewendet werden?
2. Welcher trainierte Klassifikator (Modell) soll letztendlich verwendet werden?

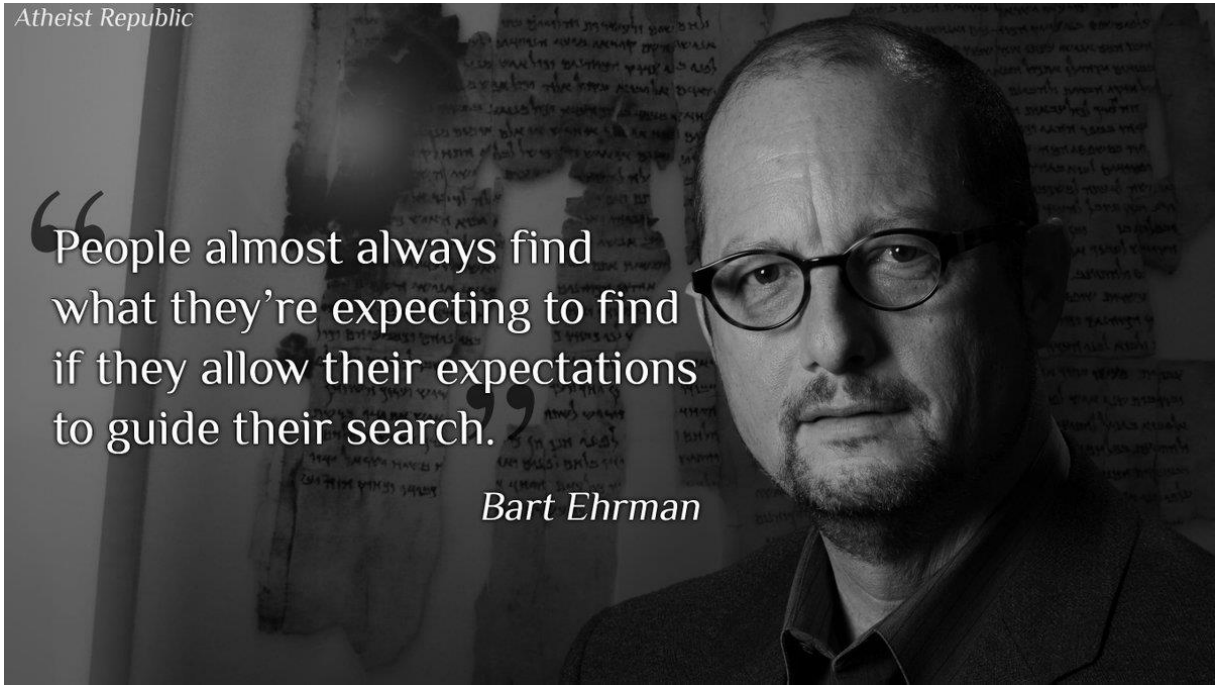
$\Rightarrow$ Lernmethoden und Modelle müssen **verglichen** werden.

$\Rightarrow$ Performance von Klassifikatoren

Next

- Performance von Klassifikatoren, Einschätzung des Gelernten,
  - Konfusionsmatrix
  - Fehlermaße: Fehlerrate, Erfolgsrate
  - **Trainingsfehler**
  - Bsp kNearest Neighbor -> Der Trainingsfehler hat ein Problem
  - [Overfitting]

- Fehlerschätzung – Wie sieht die Praxis aus:
  - Holdout-Methode + Stratifikation + Wiederholung -> **Kreuzvalidierung**
  - [leave-one-out, bootstrap]
  - **Praktisches Vorgehen**
  - Automated ML kann das auch



Expectation Bias

The tendency for experimenters to believe, certify, and publish data that agree with their expectations for the outcome of an experiment, and to disbelieve, discard, or downgrade the corresponding weightings for data that appear to conflict with those expectations.

https://en.wikipedia.org/wiki/List_of_cognitive_biases

Güte eines Modells

Wie kann die Prognosegüte eines Modells beurteilt werden:

Wir schätzen die Performance (bspw. Fehlerrate, Erfolgsrate), die das Modell bei seiner Anwendung in der Praxis erreichen wird.

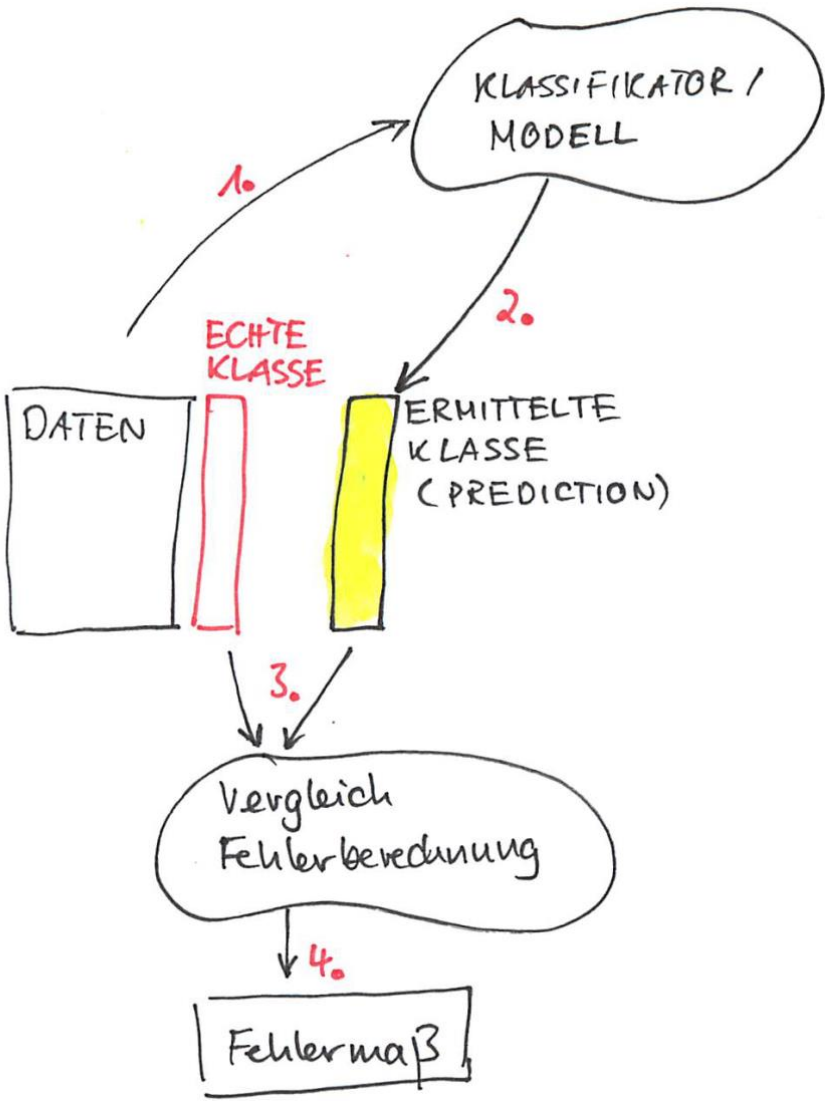
Wie? Zwei Wege (mindestens)

- Der einfache Weg
- Der bessere Weg

Modell auf Daten mit bekannter Klassifikation anwenden

1. Modell erstellen
2. Modell auf die Daten anwenden und Prognose der Klasse ermitteln (*prediction*)
3. Performance berechnen aus Vergleich der ermittelten mit der tatsächlichen Klasse

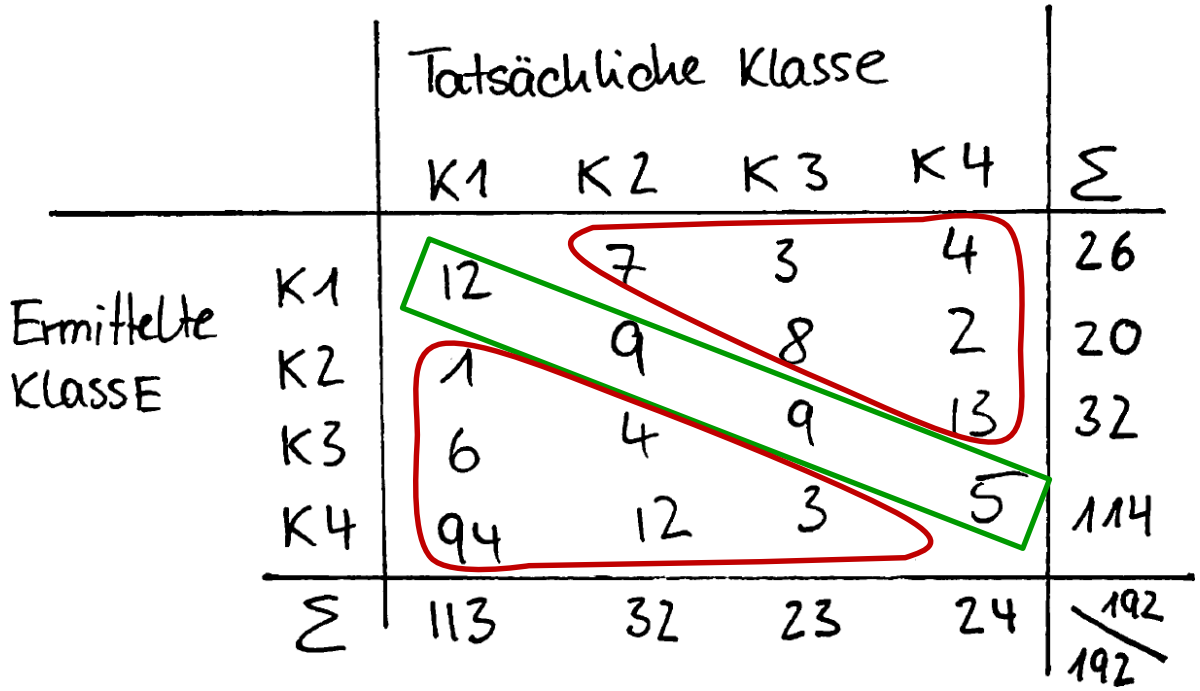
- Vielzahl von Performancemaßen



Konfusionsmatrix

Gegenüberstellung der Häufigkeiten der ermittelten Klassen (***predicted***) und der tatsächlichen Klassen (***real***) eines Testfalles.

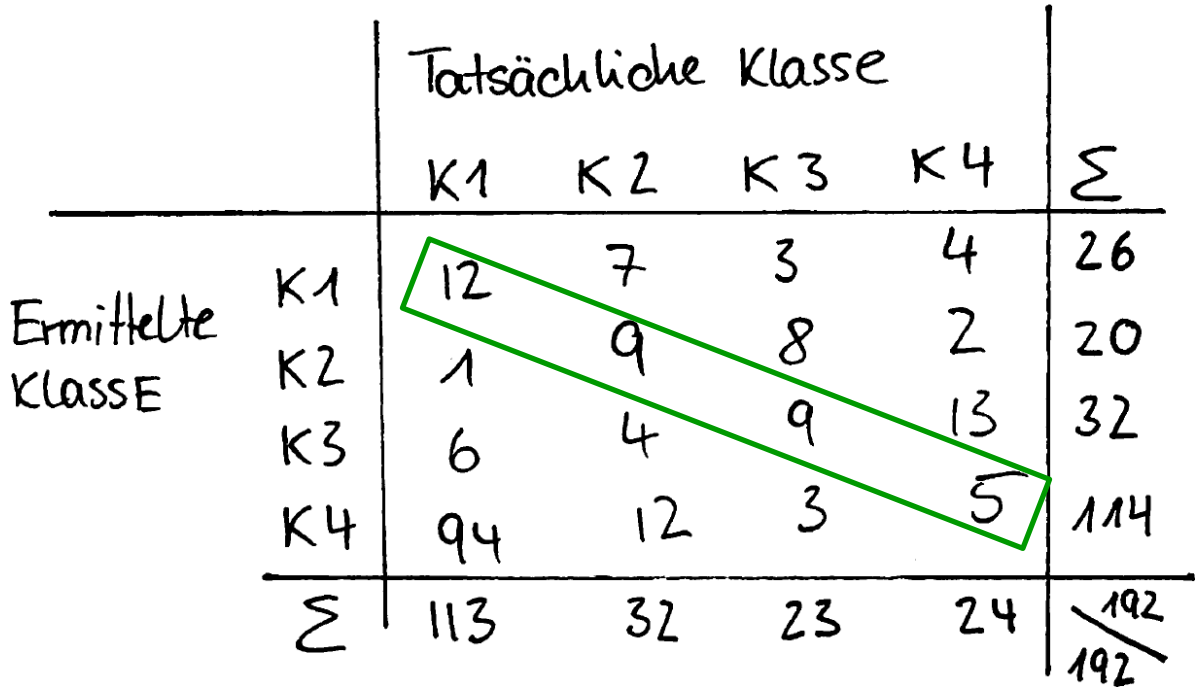
Dimension K x K mit K als Anzahl der Klassen



Hauptdiagonale =
richtig
Vorhergesagte

Aus der Konfusionsmatrix: Erfolgsrate, Fehlerrate

- **Erfolgsrate** = #richtig Vorhergesagte / #Alle
- **Fehlerrate** = #falsch Vorhergesagte / #Alle
- Engl. *accuracy, classification error*



Beispiel:

Erfolgsrate =
(12+9+9+5) / 192
= 18.2%

Fehlerrate =
(7+3+…+12+3) / 192
= 81.8%

18.2 + 81.8 = 100

Spezialfall: Konfusionsmatrix bei binärer Klassifikation

Genau 2 Klassen bspw: {ja, nein}, {positiv, negativ}, {0,1}, {wahr, falsch}, {krank, gesund} …

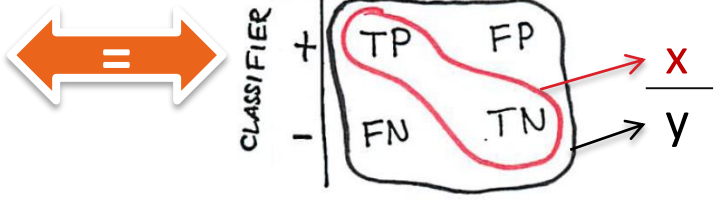
→ Konfusionsmatrix 2 x 2:

		Tatsächliche Klasse	
		positiv	**negativ**
Ermittelte Klasse	**positiv**	TP = True positive	FP = False positiv
	negativ	FN = False negative	TN = True negative

Fehler 1. Art

Fehler 2. Art

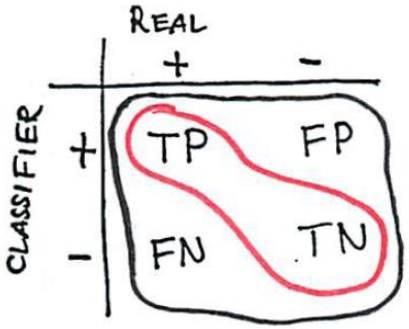
- Erfolgsrate = (TP+TN) / (TP + FP + FN +TN)
- Fehlerrate = (FP+FN) / (TP + FP + FN +TN)



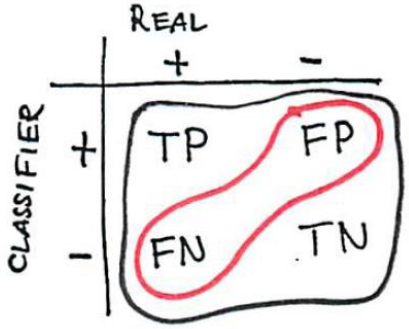
Diagnose mit 95% <Gütemaß>

krank (+, positiver Befund), gesund (-, negativer Befund)

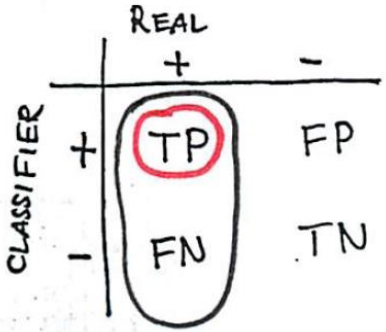
grüne Performancewerte möglichst hoch, rote niedrig



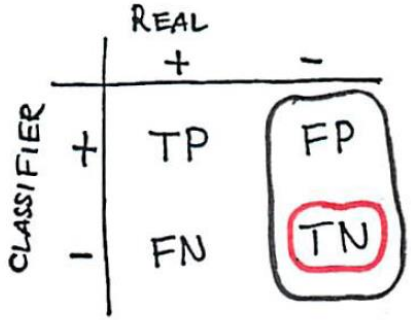
accuracy



classification error



sensitivity, recall,
true positive rate (TPR)
= TP / (TP + FN)



specificity,
true negative rate (TNR)
= TN / (TN + FP)

95% Erfolgsrate:

Die Diagnose stimmt in 95% der Fälle.

95% Fehlerrate:

95% der Diagnosen sind falsch.

95%: Sensitivität:

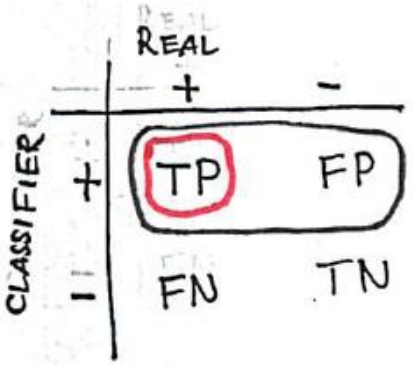
95% der Kranken werden als krank erkannt.

95% Spezifität:

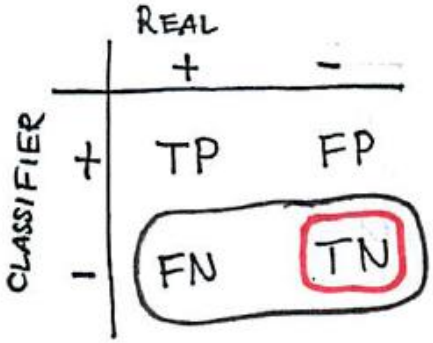
95% der Gesunden werden als gesund erkannt.

Lies: 95% der schwarz-markierten sind rot-markiert.
Beispiel vom Anfang: Herzinfarkt: Sensitivity = 81.4% Specificity = 92.1% PPV = 72.9% Accuracy = 89.9%

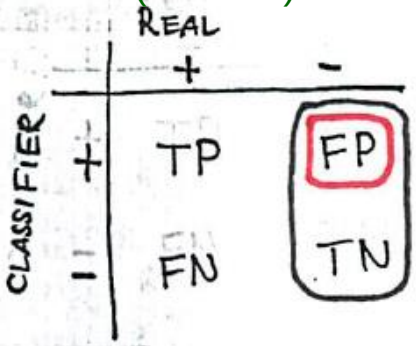
Weitere Maße



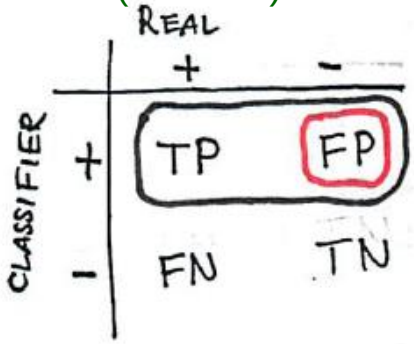
precision,
positiv predictive value (PPV)
= TP / (TP + FP)



negative predictive
value (NPV)
= TN / (FN + TN)



false positive rate (FPR)
= FP / (FP + TN)



false discovery rate (FDR)
= FP / (TP + FP)

95% Positiver Vorhersagewert:

Von krank Erklärten sind 95% krank (und 5% gesund)

95% Negativer Vorhersagewert:

95% der als gesund Erklärten sind gesund.

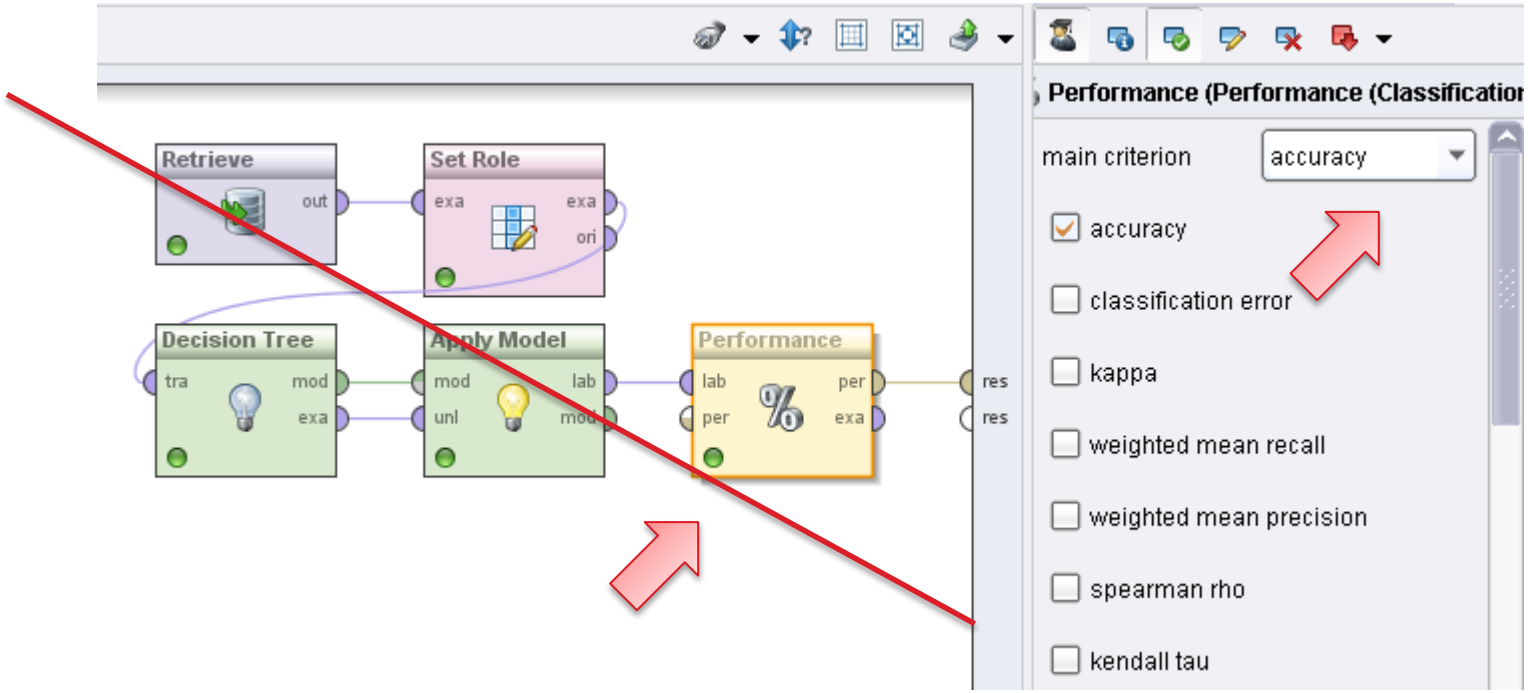
95% Ausfallrate:

95% der Gesunden werden als krank erklärt.

95% false discovery rate:

95% der **krank Erklärten sind gesund.**

So nicht!



Wo ist das Problem?

Die Performance eines Klassifizierers auf seinen Trainingsdaten ist kein guter Indikator für seine Leistung auf anderen Daten (Testmenge.)

Die Evaluierung ist viel zu optimistisch!

Trainingsfehler

auch Resubstitutionsfehler, Resubstitution = „nochmal verwenden“

- **Trainingsfehler :=** Fehlerrate eines Klassifizierers auf seinen Trainingsdaten
- Die Trainingsdaten werden also *nochmal* verwendet (Training und Test)
- Entsprechend **Trainingserfolgsrate** etc.

Warum ist der Trainingsfehler keine gute Schätzung für die spätere Leistung des Modells?

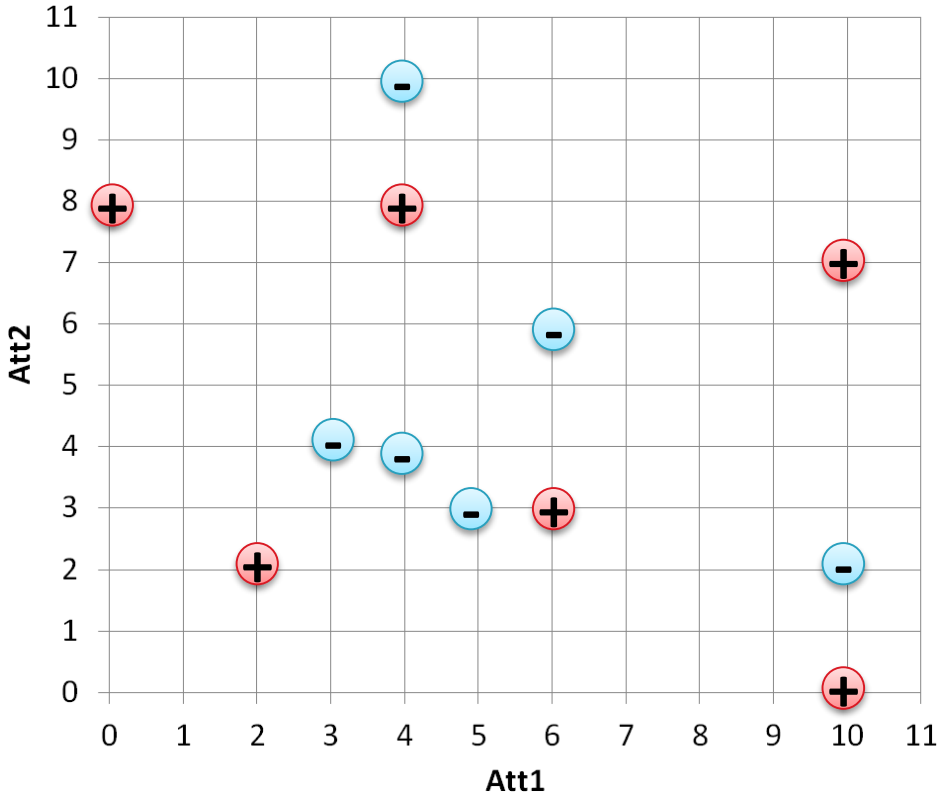
Gegenbeispiel: k-Nearest Neighbor (k-NN) - ein sog. Lazy learner („faul“)

- Lernen: Speichern aller Examples (Modell == Trainingsmenge)
- Anwenden auf unbekanntes x: Finden der k ähnlichsten Beispiele in der Trainingsmenge -> deren häufigste Klasse ausgeben
- Quiz: Wie groß ist der Trainingsfehler eines k-NN mit k = 1 ?

Trainingsfehler eines k-Nearest Neighbor mit k = 1

Att1	Att2	Klasse
6	3	+
10	0	+
5	3	-
6	6	-
4	8	+
2	2	+
0	8	+
3	4	-
5	3	-
10	7	+
4	4	-
10	2	-
4	10	-

=



- Modell == Trainingsmenge
- Wie wird ein Datensatz X aus der Trainingsmenge, bspw. X = (6,6), durch den k-NN mit k=1 klassifiziert?

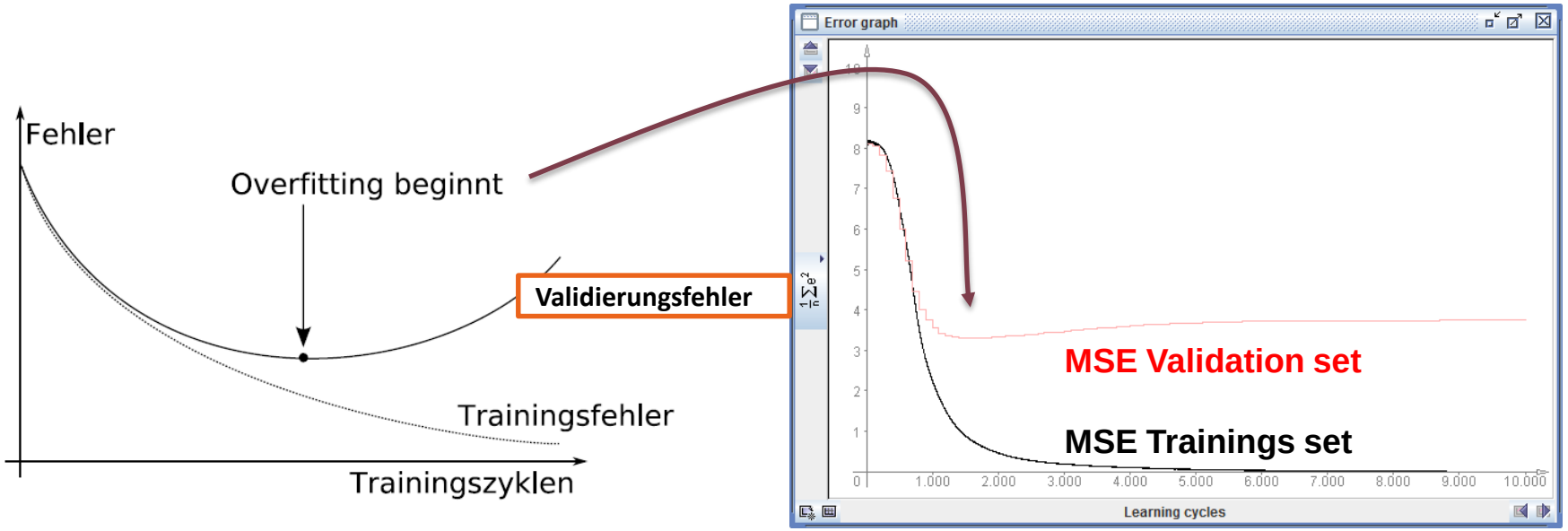
Die Trainings-Erfolgsrate eines k-NN mit k = 1 ist stets 100%.

Das ist eine **sehr** optimistische Schätzung für den zukünftigen Fehler, oder?

Overfitting

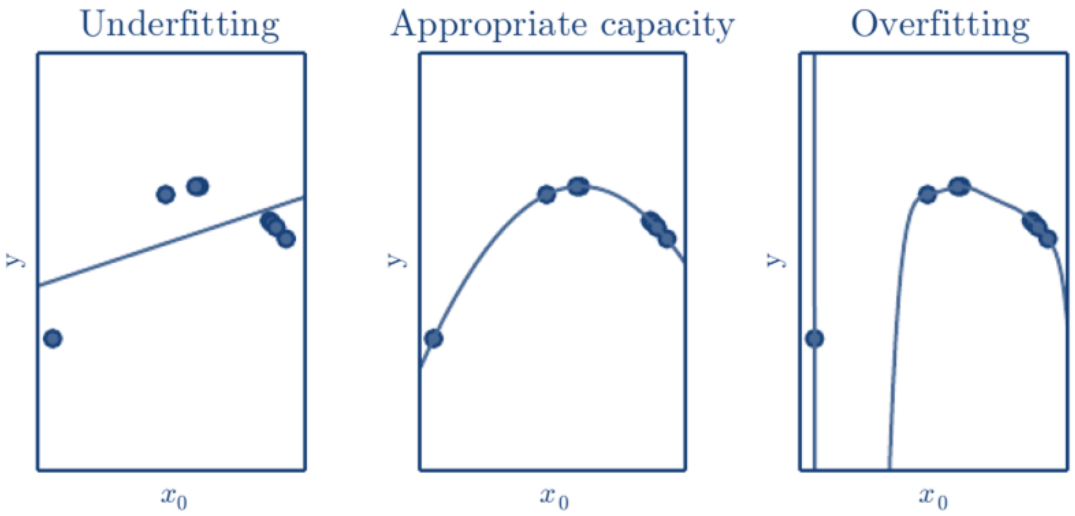
Def: Leistung auf Trainingsdaten zwar sehr gut, **aber nicht im späteren Einsatz**

- überangepasste Wissensrepräsentation, **Auswendiglernen** der Trainingsdaten,
- Generalisierungsfähigkeit fehlt
- nicht nur bei neuronalen Netzen



Kapazität, Ausdruckskraft, Parameteranzahl

(Geheime) quadratische Funktion $y = f(x_0)$ erzeugt Datenpunkte



Fit und Predict:

Links: lineare Funktion, hoher Trainingsfehler, hoher Testfehler

Mitte: quadratische Funktion, geringer Trainingsfehler, geringer Testfehler

Rechts: Polynom 9-ten Grades, geringer Trainingsfehler, hoher Testfehler

[GBC16] Goodfellow, Ian ; Bengio, Yoshua ; Courville, Aaron: Deep Learning. MIT Press, 2016. – http://www.deeplearningbook.org

Maßnahmen gegen Overfitting

- Mehr Daten,
  - gerne auch künstlich erzeugte (*real-world variation*)
  - *data set augmentation* (Bilder verzerren, Rauschen einbringen (Daten, Gewichte))
- *early stopping*: Training bis zur Verschlechterung beim *validation set*
- Größe (Freiheitsgrade) der Modelle beschränken,
  - Pruning bei DT
  - wenige hidden neurons (aber gerade deep learning: riesige Neuronanzahl)
  - *parameter sharing*, bspw. *convolution* bei deep learning
  - *regularization* bei NN (Strafe für große Gewichte)

Quiz: Was stimmt?

falsch

1. Ein Lernalgorithmus, der auf den Trainingsdaten einen hohe Erfolgsrate erreicht, liefert Modelle, die beim späteren Einsatz eine hohe Erfolgsrate zeigen.

richtig

2. Ein Lernalgorithmus, der auf den Trainingsdaten einen niedrige Erfolgsrate erreicht, liefert Modelle, die beim späteren Einsatz eine niedrige Erfolgsrate zeigen.

Wie können wir den zukünftigen Fehler besser schätzen?

Die Trainings-Erfolgsrate eine k-NN mit k = 1 ist stets 100%. Ein hochangepasster Klassifikator.

Der Trainingsfehler ist nicht geeignet, den Fehler während der späteren Anwendung des Klassifikators zu schätzen.

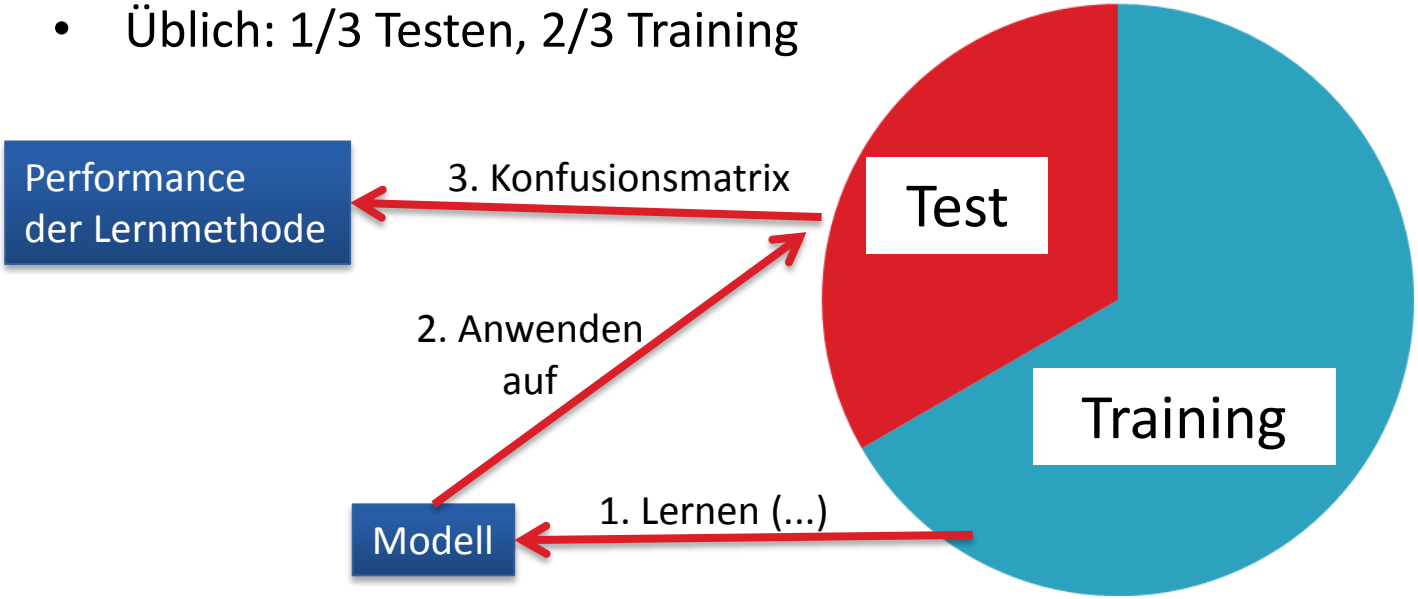
- Trotzdem wird diese Methode oft verwendet.
  - Als Heuristik: ein großer Trainingsfehler zeigt, dass die Modellierung mit diesem Lernverfahren nicht gelingt.

- Einfacher Weg - Trainingsfehler
- Besserer Weg - kommt jetzt

Holdout-Methode

- Holdout-Methode reserviert Daten für das **Testen** und verwendet nur den Rest fürs **Training**
- 2 disjunkte Mengen
  - Trainingsdaten: Erstellung des Klassifikators, Lernen
  - Testdaten: Schätzung der Fehlerrate des Klassifikators

- Üblich: 1/3 Testen, 2/3 Training



Stratifikation

Problem: Möglicherweise sind die für die Testmenge gewählten Daten nicht repräsentativ

Stratifikation:

- Datenmenge so teilen, dass die relativen Klassenhäufigkeiten in Test- und Trainingsdaten übereinstimmen

-> **Stratifizierte Holdout Methode**

Beispiel:

Die Daten enthalten in der Klassenspalte 33% Damen und 67% Herren. Dann sollte die Teilung in Trainings- und Testmenge so erfolgen, dass beide ebenfalls 33% Damen und 67% Herren enthalten (und nicht alle Damen zufällig in der Testmenge landen).

Wiederholtes Holdout

- Idee: Holdout-Methode **mehrmals mit unterschiedlichen Datenteilungen** wiederholen und
- den Mittelwert alle Fehlerraten bilden

- damit erfolgt eine Kompensierung der Fehler, die durch die Auswahl der Stichprobe (Teilung) entstanden sind

- Das wichtigste Verfahren des wiederholten Holdout:

  Kreuzvalidierung (engl. cross validation, CV)

n-fache Kreuzvalidierung (CV)

Datensätze in n Mengen (**Partitionen, *folds***) aufteilen, z.B. n=3 oder n=10.

Für i=1 bis n:

- i-te Menge ist Testmenge
- restliche n-1 Mengen sind Trainingsmenge -> Lerner erzeugt Modell i
- bestimme die Leistung des Modells auf der Testmenge -> Fehler i

-> n Fehlerwerte

Bestimme Mittelwert und ggf. Streuung des Fehlers -> Ergebnis der CV

- Das Ergebnis der CV ist eine Schätzung des zu erwartenden Generalisierungsfehlers auf unbekannten Daten.
- Das Modell für den späteren Einsatz wird mit allen verfügbaren Daten erstellt (sog. 100%-Modell)
- Ergebnis das Data-Mining-Prozesses = Modell + Fehlerschätzung des Modells

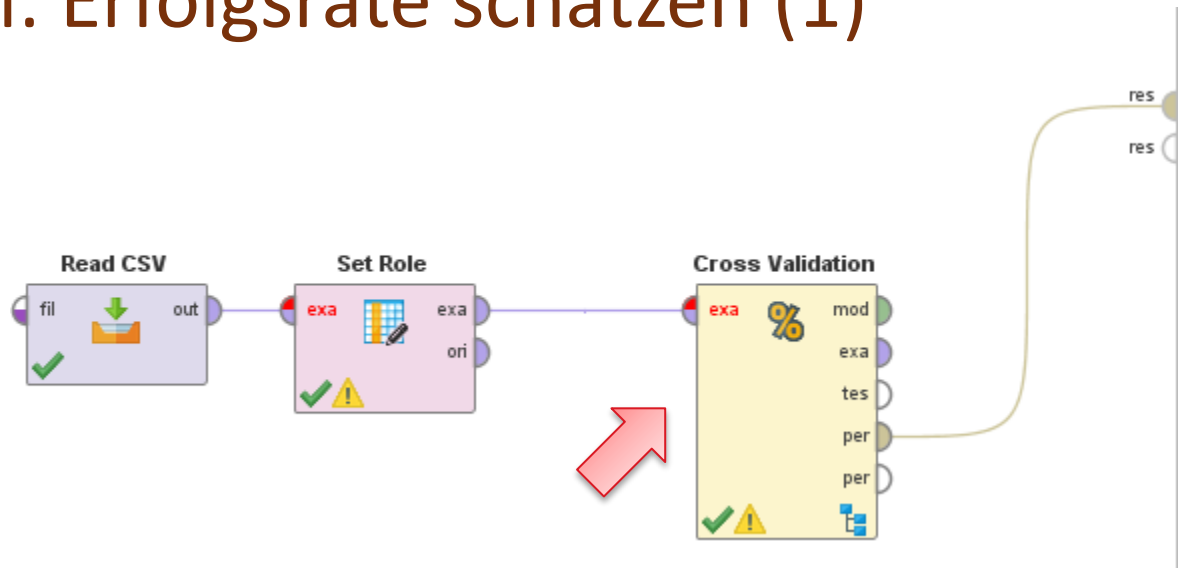
Stratifizierte Kreuzvalidierung

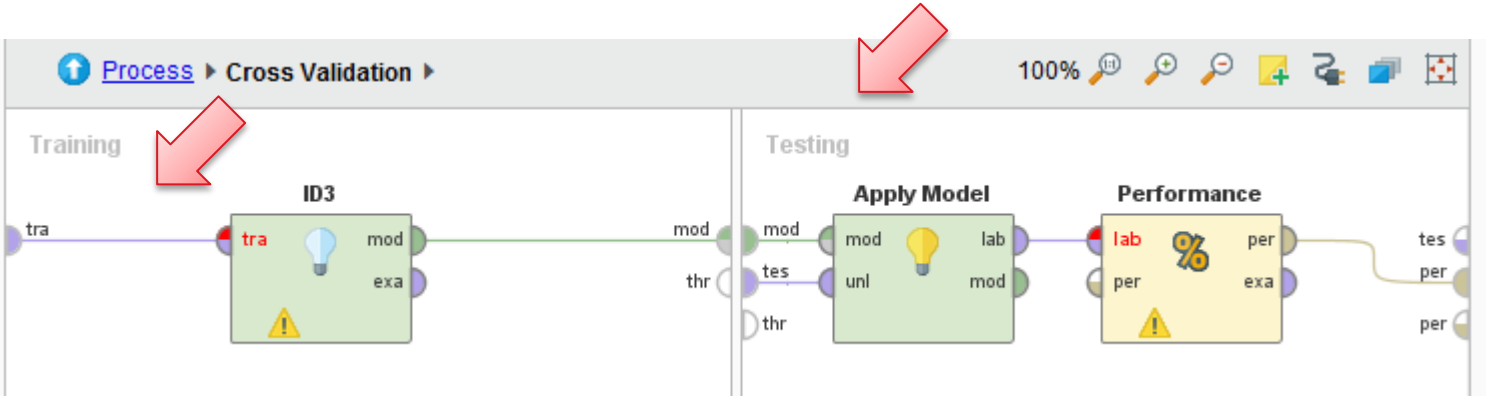
- Besser: Stratifikation
  - Relative Klassenhäufigkeiten in allen Partitionen annähernd gleich zur Verteilung in der Gesamtmenge
  - -> **stratifizierte Kreuzvalidierung**

- Noch besser: **wiederholte Kreuzvalidierung**
  - Bsp: 10mal wiederholte 10fache Kreuzvalidierung
  - -> 100 Modelle zu lernen und zu testen

- in RapidMiner: Cross Validation
- Beispiel: Erfolgsrate eines ID3 und NN auf der Datenmenge weather vergleichen

Beispiel: Erfolgsrate schätzen (1)

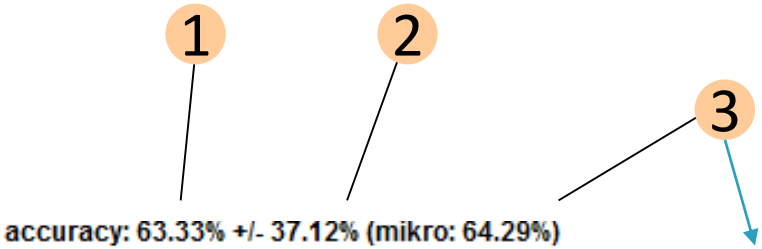




weather.nominal.csv, n=5

Erfolgsrate schätzen (2)

- Erfolgsrate (ID3) = 63.33%



	true no	true yes	class precision
pred. no	3	3	50.00%
pred. yes	2	6	75.00%
class recall	60.00%	66.67%	

Erfolgsrate schätzen (3)

5 Konfusions-matrizen:

	real	
0 2 / 1 0	1 1 / 0 1	0 0 / 1 1	1 0 / 0 2	1 0 / 0 2	predicted

						Mittelwert
Classification_error (FP+FN)/All	3/3	1/3	1/2	0/3	0/3	36.67%
Accuracy (TP+TN)/All	0/3	2/3	1/2	3/3	3/3	**63.33%** 1

Standardabweichung der Erfolgsraten e_i: s= $\sqrt{\frac{1}{n}(e_i - \bar{e})}$ = **0,371184291** 2

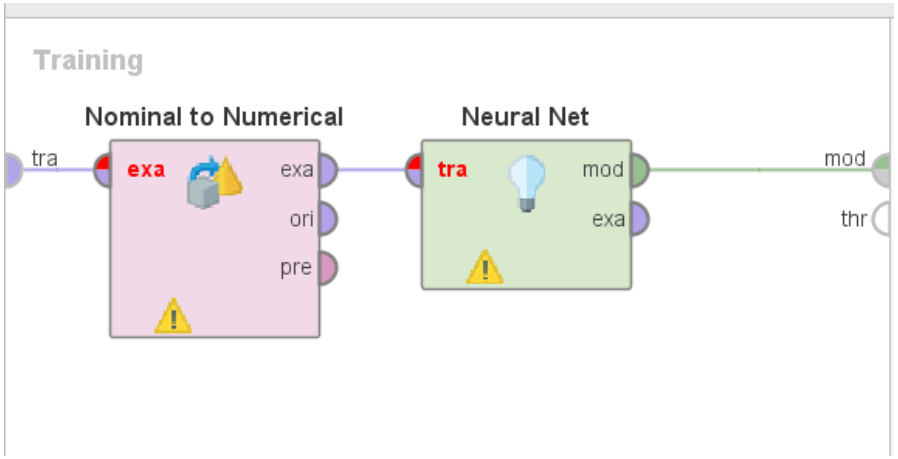
Kumulierte Konfusionsmatrix:

3	3
2	6

Erfolgsrate der kumulierten Konfusionsmatrix: 9/14 = **0,642857143** 3

Erfolgsrate schätzen (4)

- Neuronales Netz
- Erfolgsrate (KNN) = 63.33 %



Sie wissen noch, wie ein Backpropagation-Netzwerk funktioniert?

Die gesamte Theorie ist hier nur ein kleiner Operator. Unter vielen.

Andere Schätzverfahren – Leave one out

- Leave one out
  - n-fache Kreuzvalidierung mit n=Instanzenanzahl
  - Testmenge = eine Instanz, Rest ist Trainingsmenge
  - Keine zufällige Stichprobenwahl -> deterministisch
  - Hoher Rechenaufwand: n-mal lernen
  - Realistische Schätzung
  - keine praktische Bedeutung

Andere Schätzverfahren - Bootstrap

- Bootstrap
  - aus einer Datenmenge mit n Instanzen wird n-mal zufällig eine Instanz gewählt und in die Trainingsdaten kopiert
  - Nicht gewählte Instanzen -> Testdaten
  - p=1/n für jede Instanz, dass sie ausgewählt wird
  - p=1-1/n, dass sie nicht ausgewählt wird
  - Potenzierung über n Auswahlvorgänge führt zur Wahrscheinlichkeit, dass eine bestimmte Instanz in der Trainingsmenge vorkommt von:

$$1-(1-1/n)^n \xrightarrow{n\to\infty} 1-e^{-1} \approx 0.632$$

  - Pessimistische Schätzung, weil Trainingsmenge nur 63% der ursprünglichen Instanzen enthält
  - geeignet für kleine Datenmengen, beliebig oft wiederholbar

Praktisches Vorgehen,
bspw. beim Data Mining Cup, Data Challenge

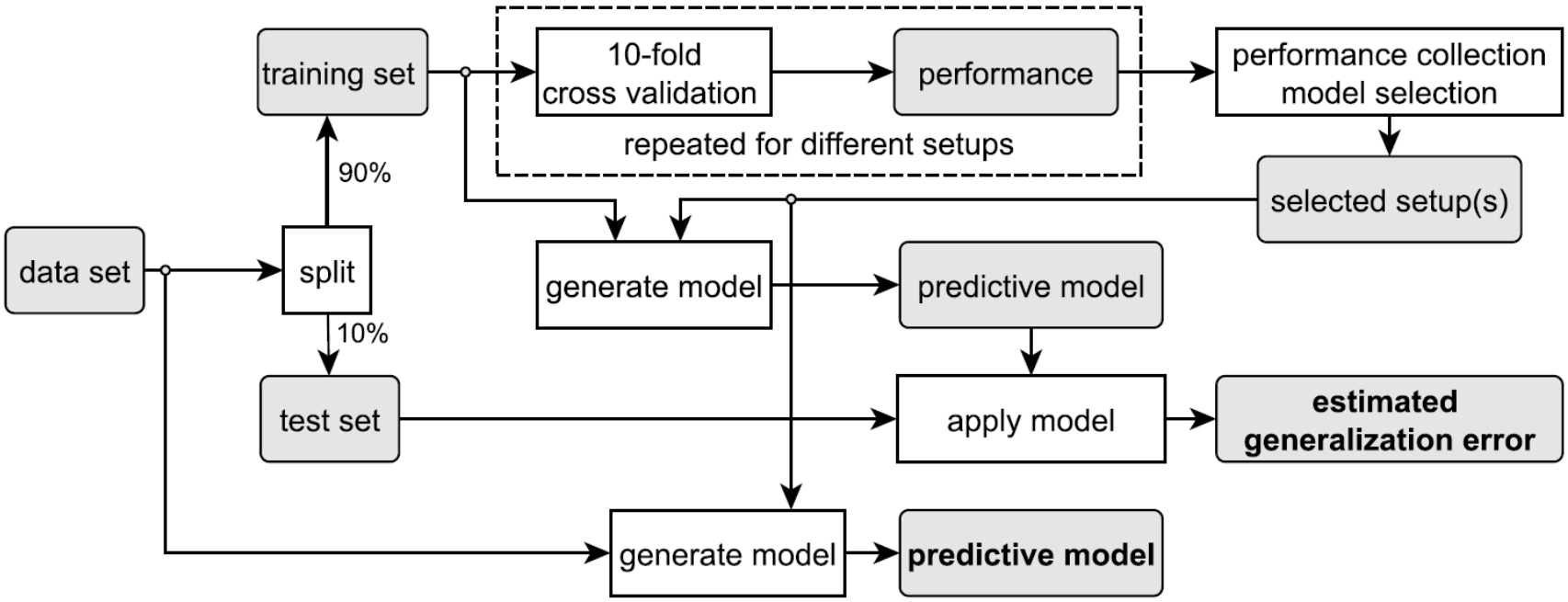


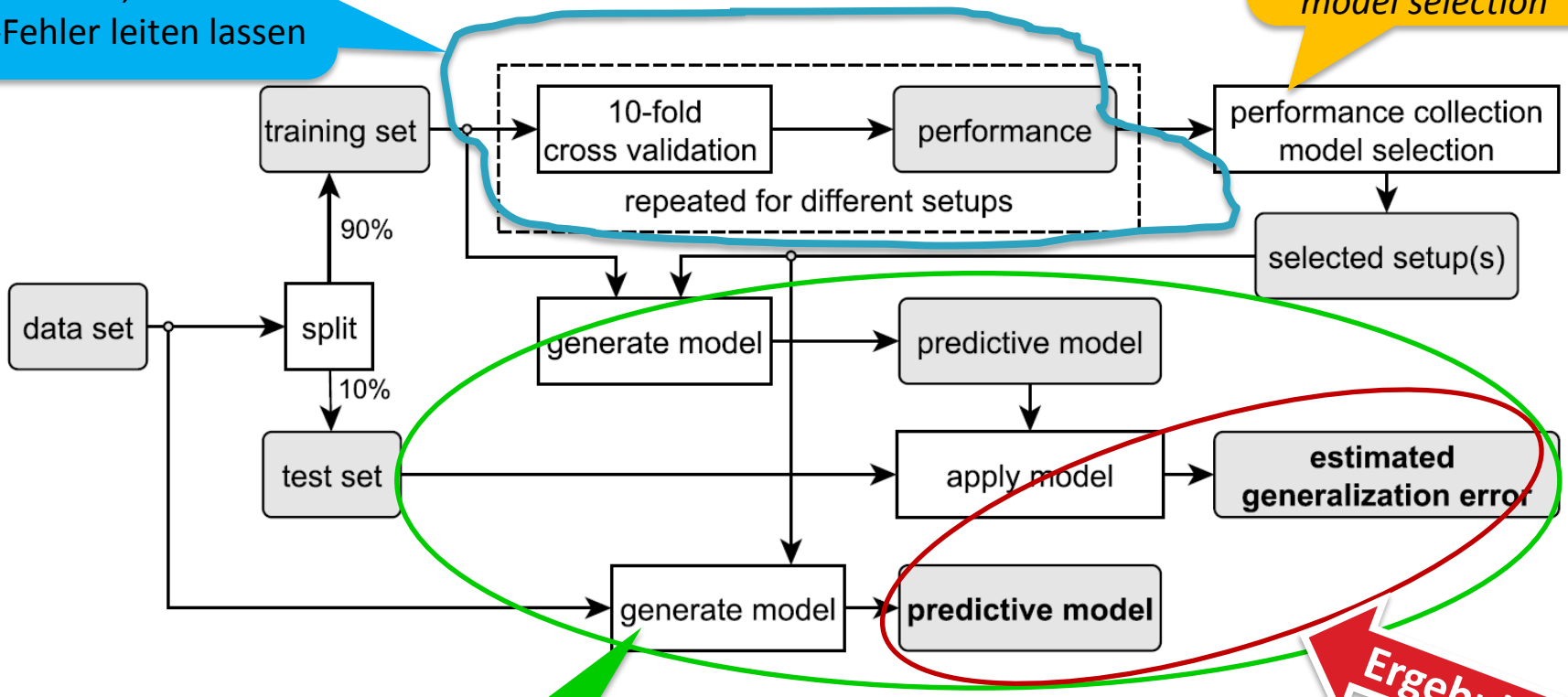
Fig. 7 The evaluation process used to estimate the performance of each setup.

Boersch, Ingo; Füssel, Uwe; Gresch, Christoph; Großmann, Christoph; Hoffmann, Benjamin:
Data mining in resistance spot welding: A non-destructive method to predict the welding spot diameter by monitoring process parameters.
In: The International Journal of Advanced Manufacturing Technology (2016), 1--15. http://dx.doi.org/10.1007/s00170-016-9847-y, ISSN 1433-3015

Praktisches Vorgehen,
bspw. beim Data Mining Cup, Data Challenge

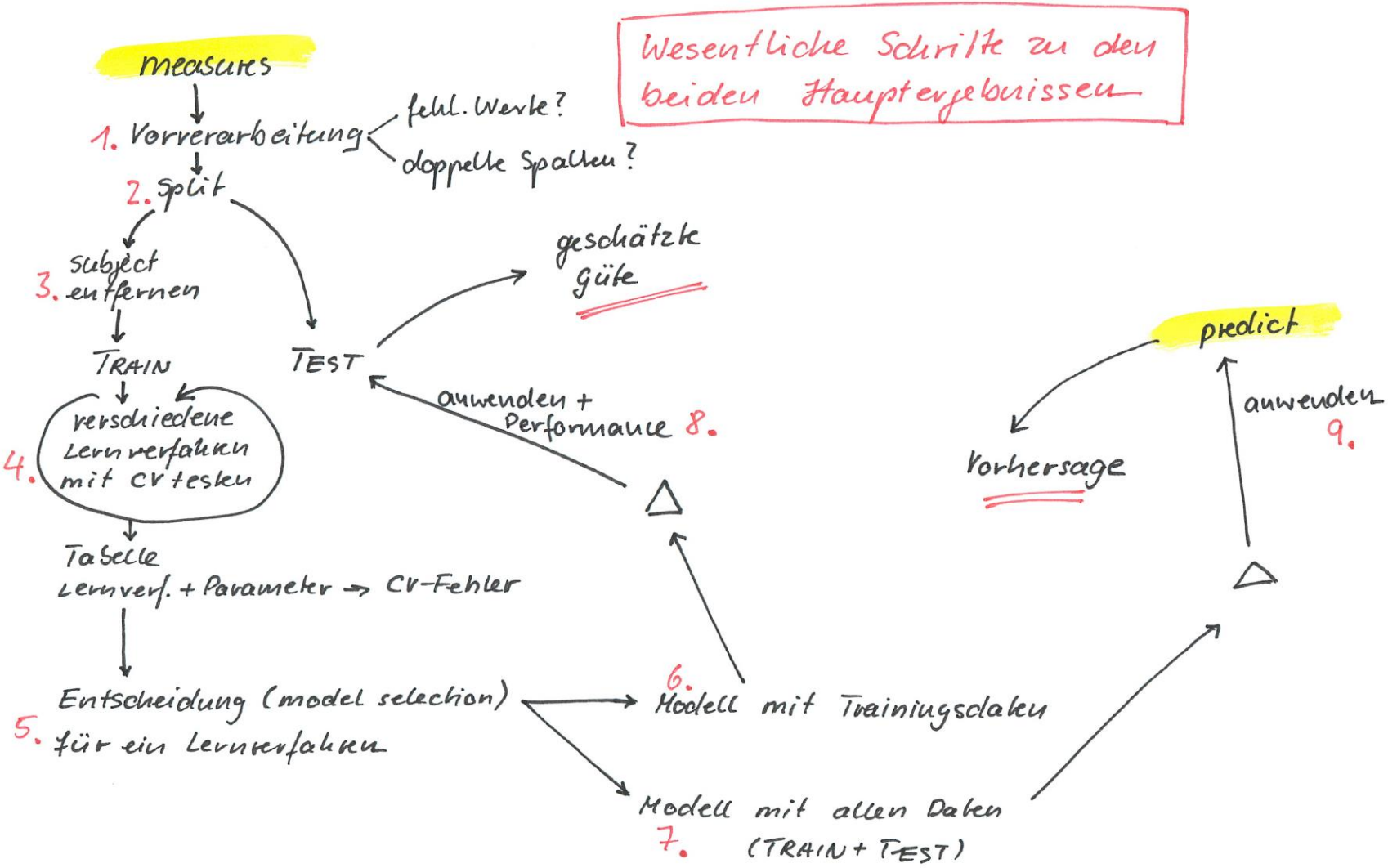
Modelle probieren, optimieren, dabei vom CV-Fehler leiten lassen

CV-Fehler in einer Tabelle sammeln – Basis für *model selection*

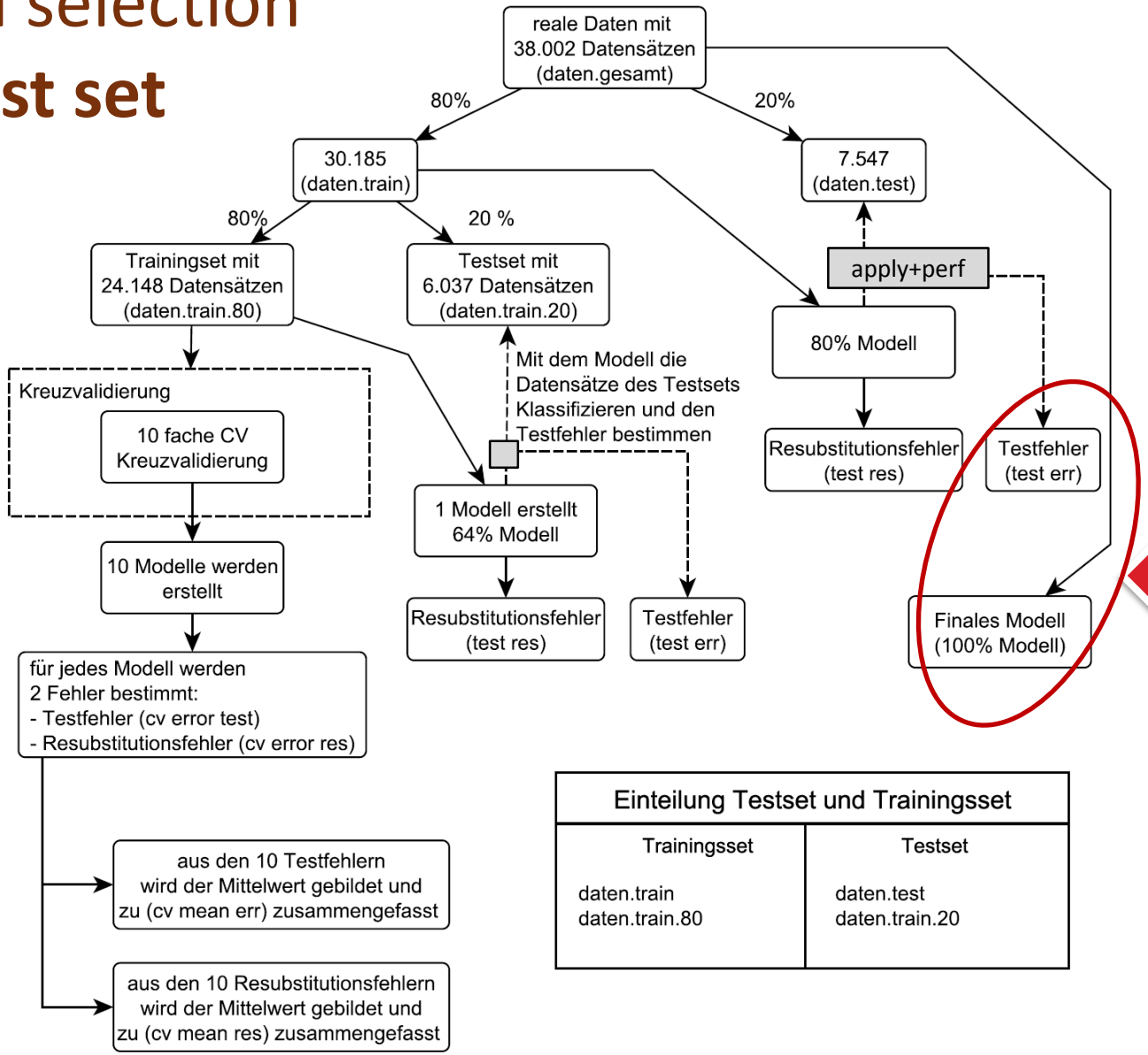


nur für das ausgewählte Lernverfahren

Praktisches Vorgehen, konkreter für Data Challenge



model selection **mit test set**



Warum?
- **Suche**
- **Modell**
- **Güte**

Ergebnis

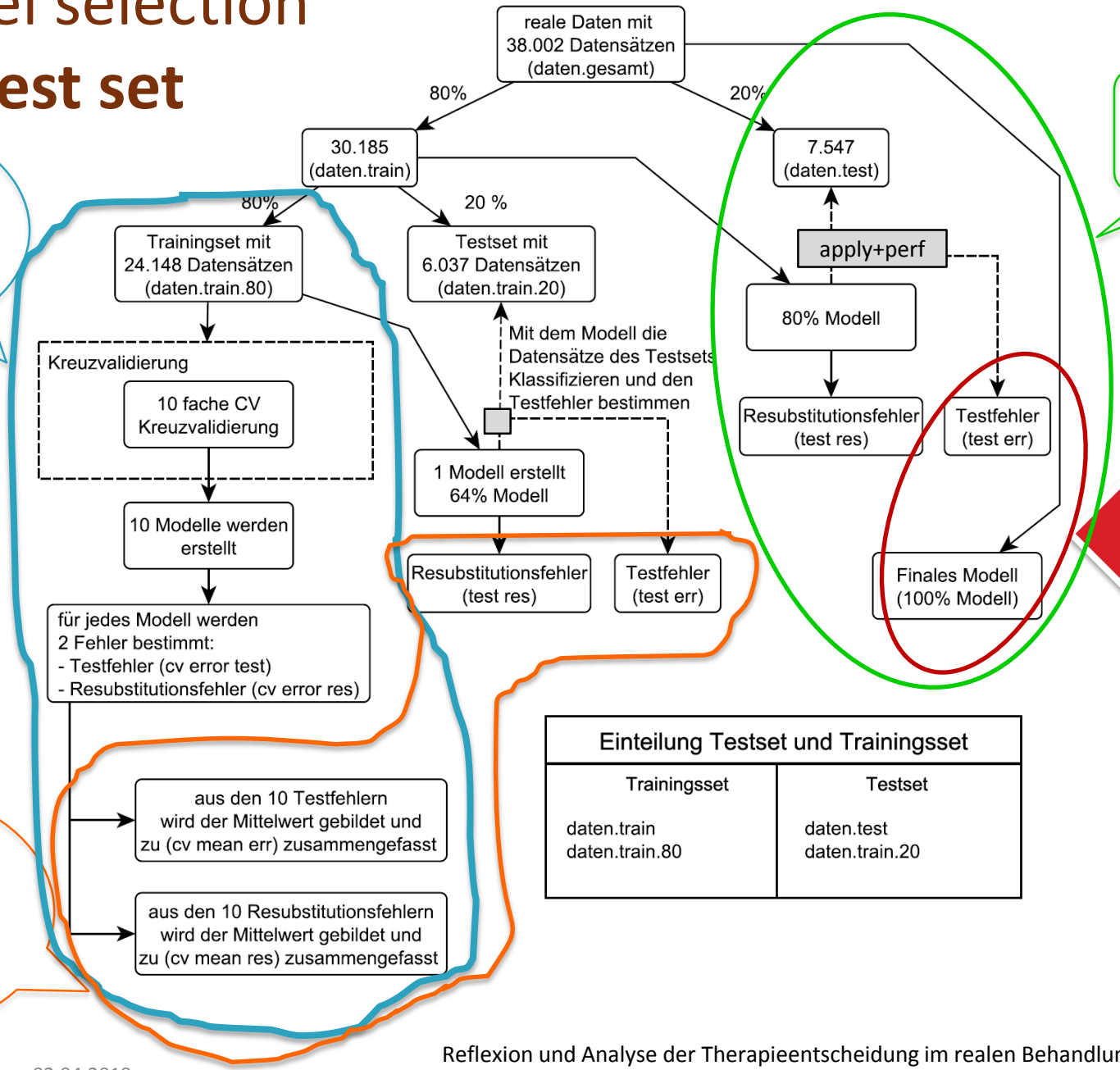
Einteilung Testset und Trainingsset	
Trainingsset	Testset
daten.train daten.train.80	daten.test daten.train.20

Reflexion und Analyse der Therapieentscheidung im realen Behandlungsprozess des Mammakarzinoms, Masterarbeit von Maik-Peter Jacob, THB, 28. Februar 2016

model selection
mit test set

Modelle probieren, optimieren, dabei vom CV-Fehler leiten lassen

Warum?
- **Suche**
- **Modell**
- **Güte**



nur für das ausgewählte Lernverfahren

Ergebnis

Einteilung Testset und Trainingsset	
Trainingsset	Testset
daten.train daten.train.80	daten.test daten.train.20

in einer Tabelle sammeln – Basis für *model selection*

Reflexion und Analyse der Therapieentscheidung im realen Behandlungsprozess des Mammakarzinoms, Masterarbeit von Maik-Peter Jacob, THB, 28. Februar 2016

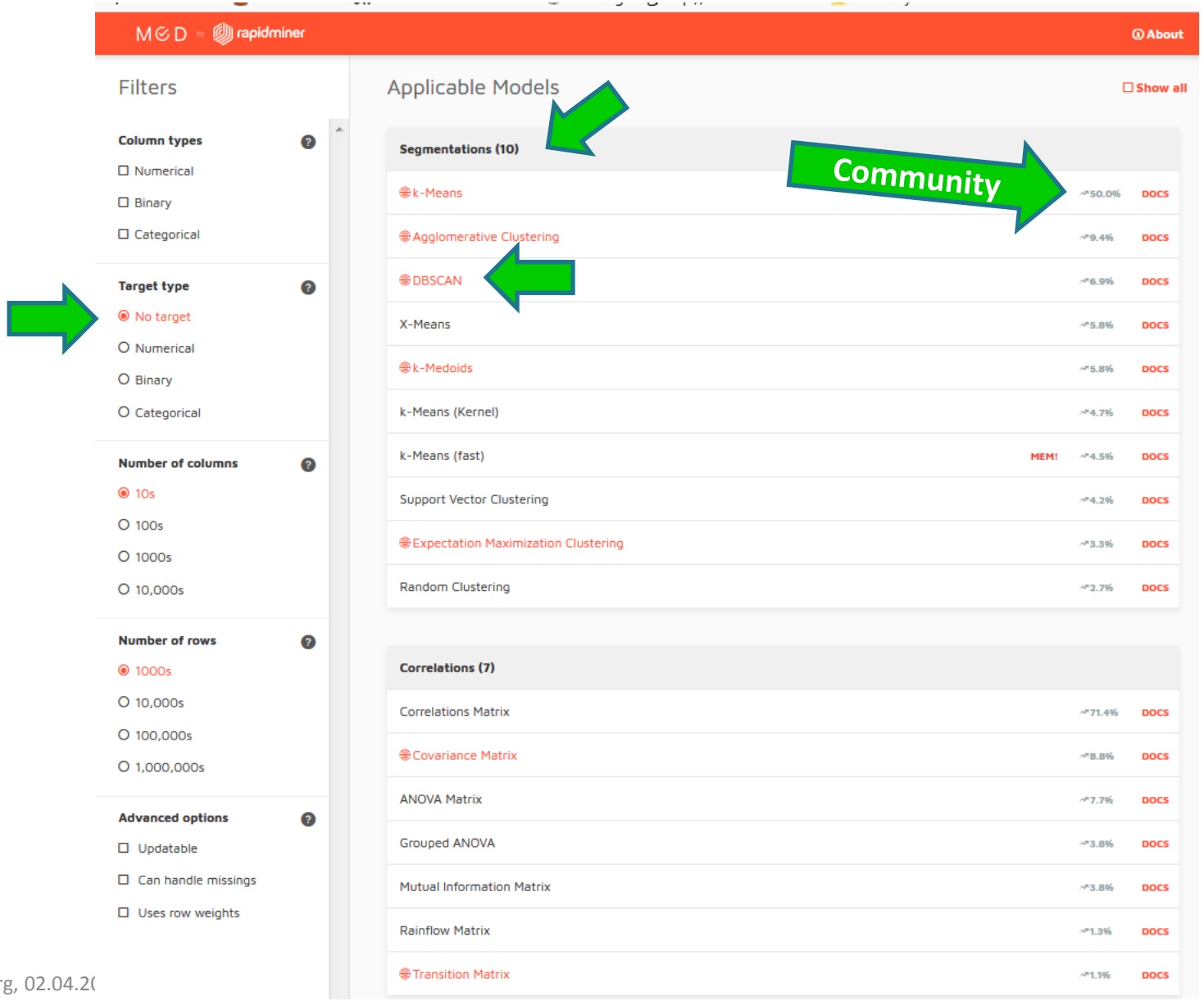
Trend: Self-Service-Analytics

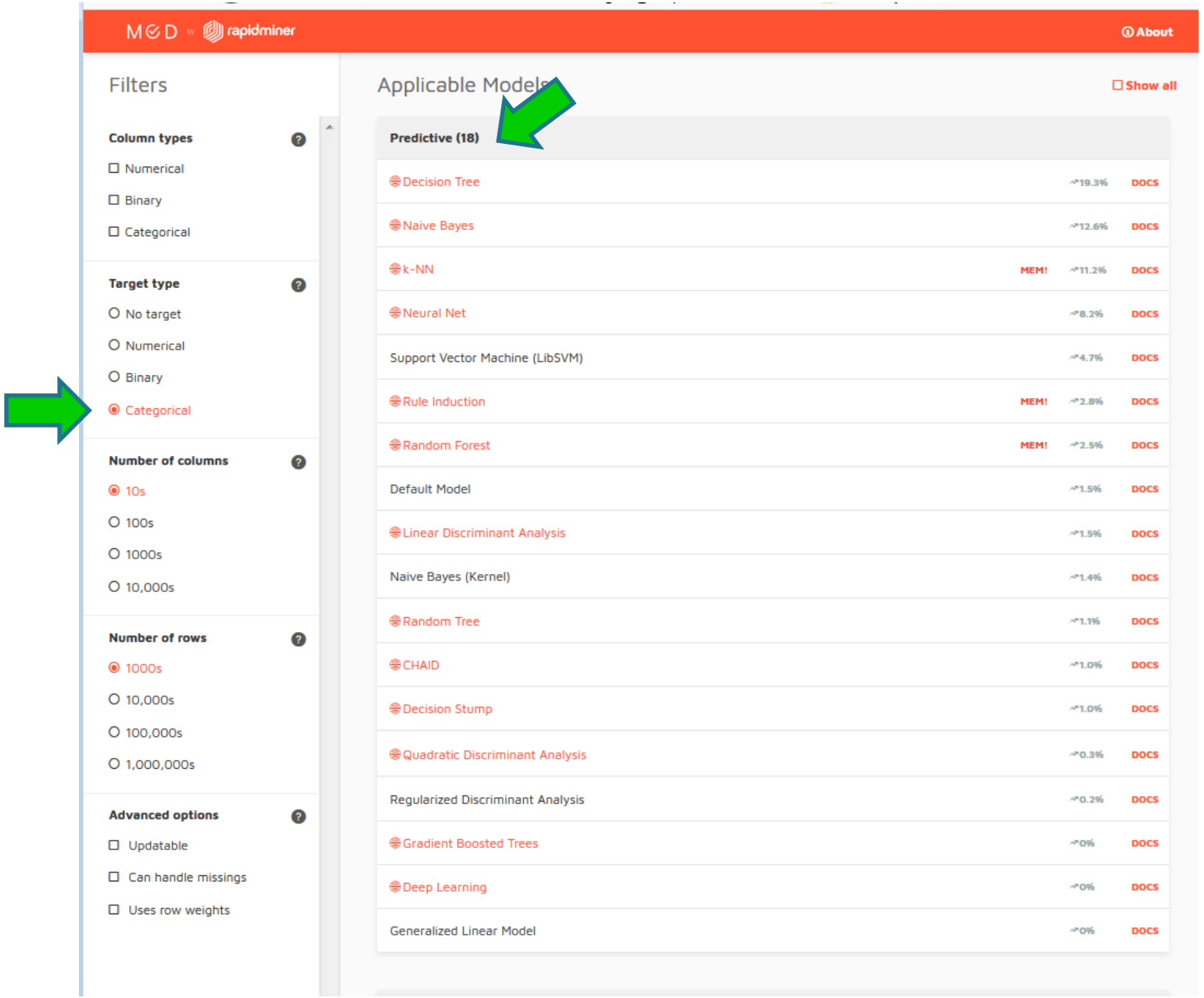
- einfach zu bedienende Analysewerkzeuge
- ohne Vorkenntnisse, ohne Programmierung
- datenbasierte Meetings

Beispiele RapidMiner

1. Welches Modell ist bei meinen Daten anwendbar?
   http://mod.RapidMiner.com/#app
2. Welches von denen ist das beste mit den besten Parametern?
   Auto Model
   - nur in Educational und Professional Licence
   - Prediction, Clustering, Outlier
   - https://docs.RapidMiner.com/latest/studio/auto-model/

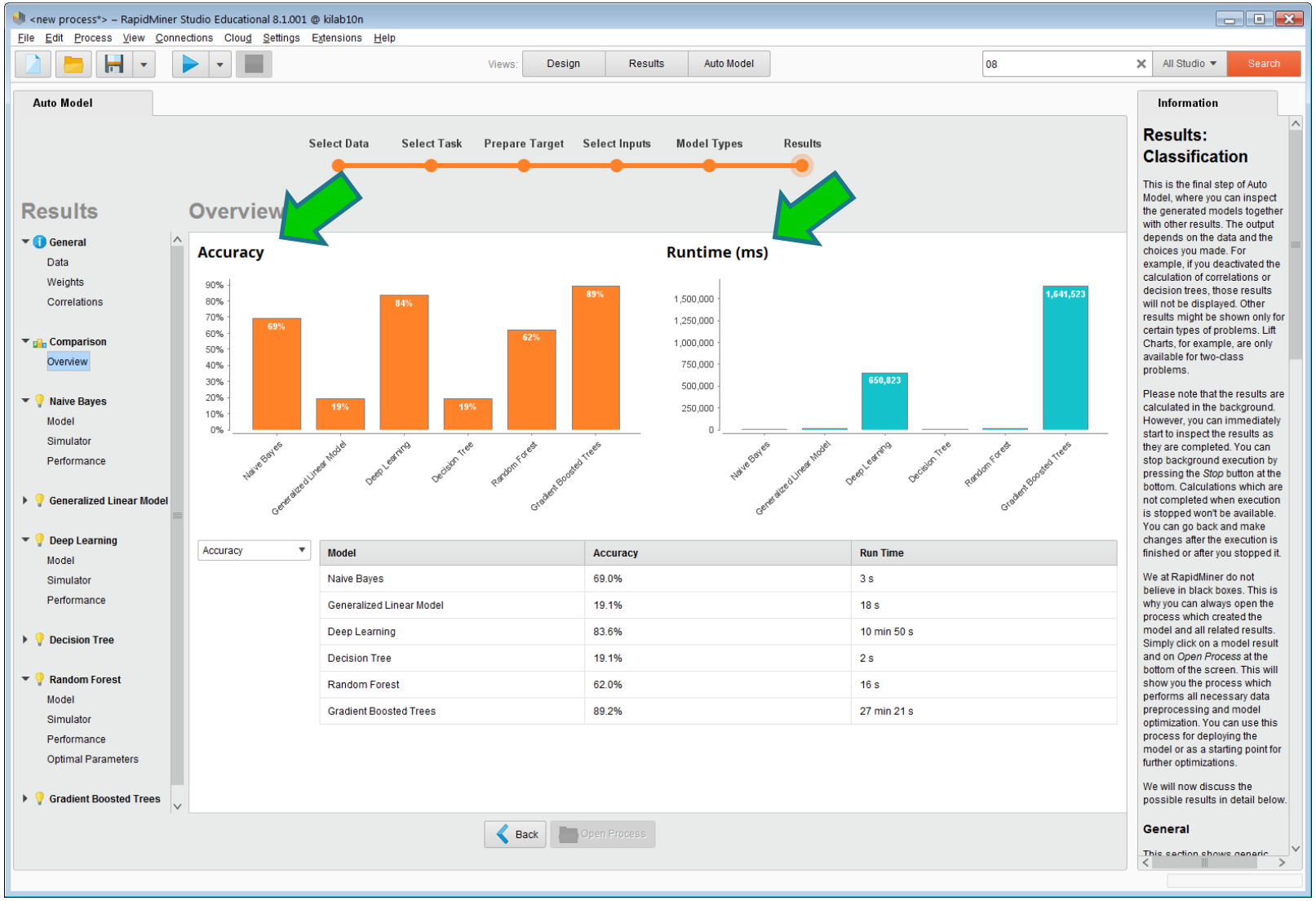
http://mod.RapidMiner.com/#app

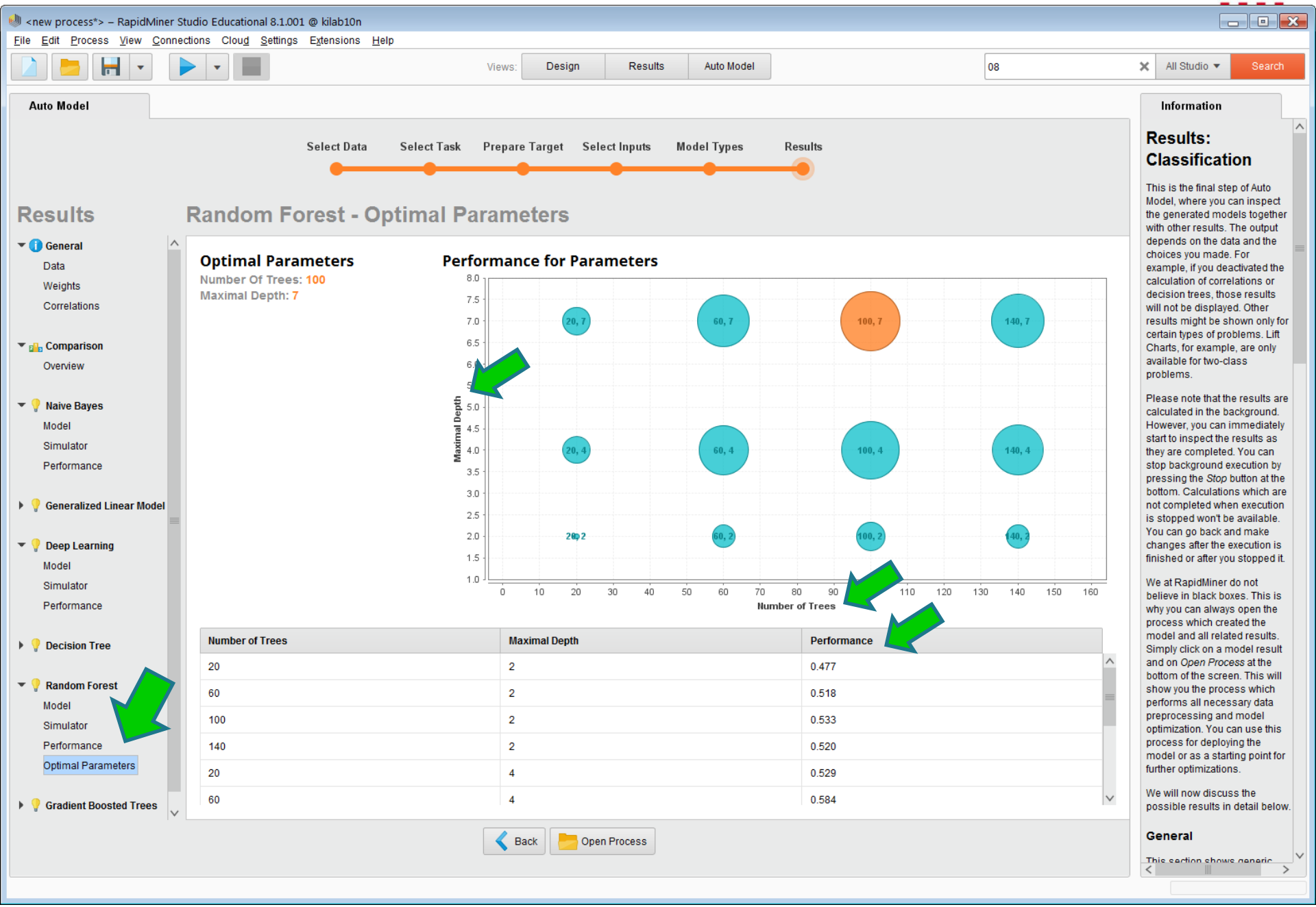


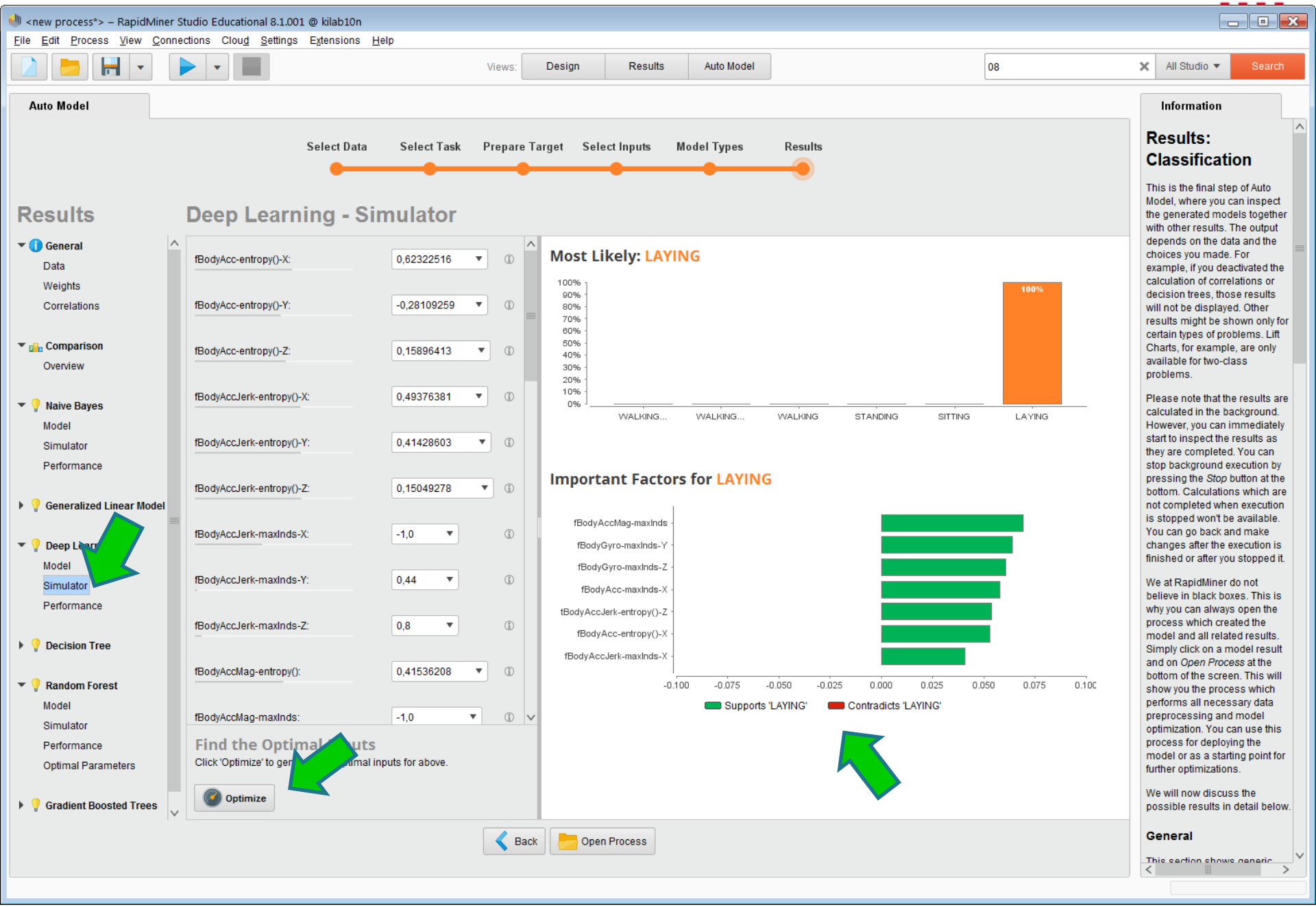

M⊂D by rapidminer
About
Filters
Column types
Numerical
Binary
Categorical
Target type
No target
Numerical
Binary
Categorical
Number of columns
10s
100s
1000s
10,000s
Number of rows
1000s
10,000s
100,000s
1,000,000s
Advanced options
Updatable
Can handle missings
Uses row weights
Applicable Models
Show all
Predictive (18)
Decision Tree ^19.3% DOCS
Naive Bayes ^12.6% DOCS
k-NN MEM! ^11.2% DOCS
Neural Net ^8.2% DOCS
Support Vector Machine (LibSVM) ^4.7% DOCS
Rule Induction MEM! ^2.8% DOCS
Random Forest MEM! ^2.5% DOCS
Default Model ^1.5% DOCS
Linear Discriminant Analysis ^1.5% DOCS
Naive Bayes (Kernel) ^1.4% DOCS
Random Tree ^1.1% DOCS
CHAID ^1.0% DOCS
Decision Stump ^1.0% DOCS
Quadratic Discriminant Analysis ^0.3% DOCS
Regularized Discriminant Analysis ^0.2% DOCS
Gradient Boosted Trees ^0% DOCS
Deep Learning ^0% DOCS
Generalized Linear Model ^0% DOCS

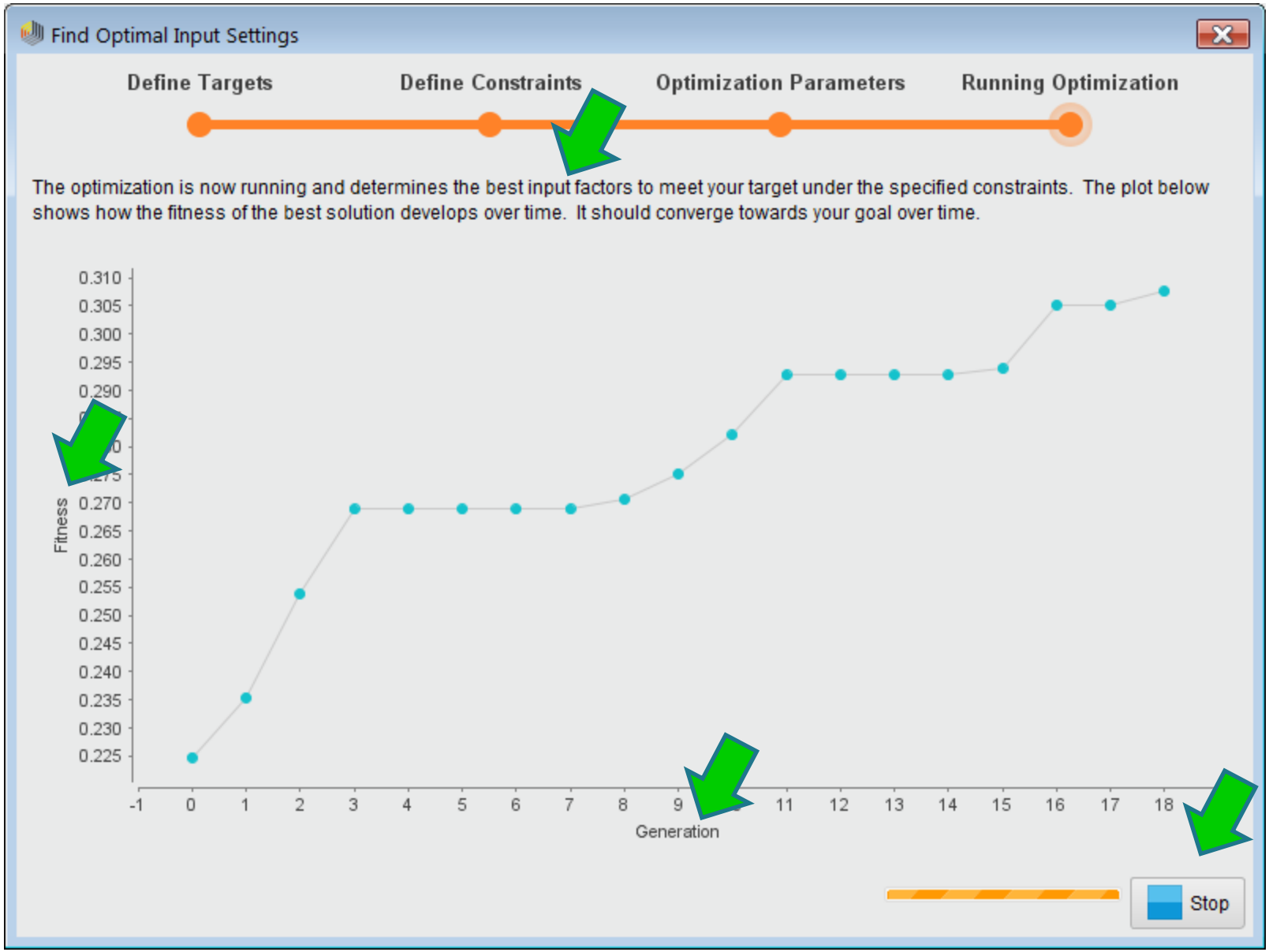
Auto Model: Prediction Task

Training geeigneter Modelle, teilweise mit Parameteroptimierung

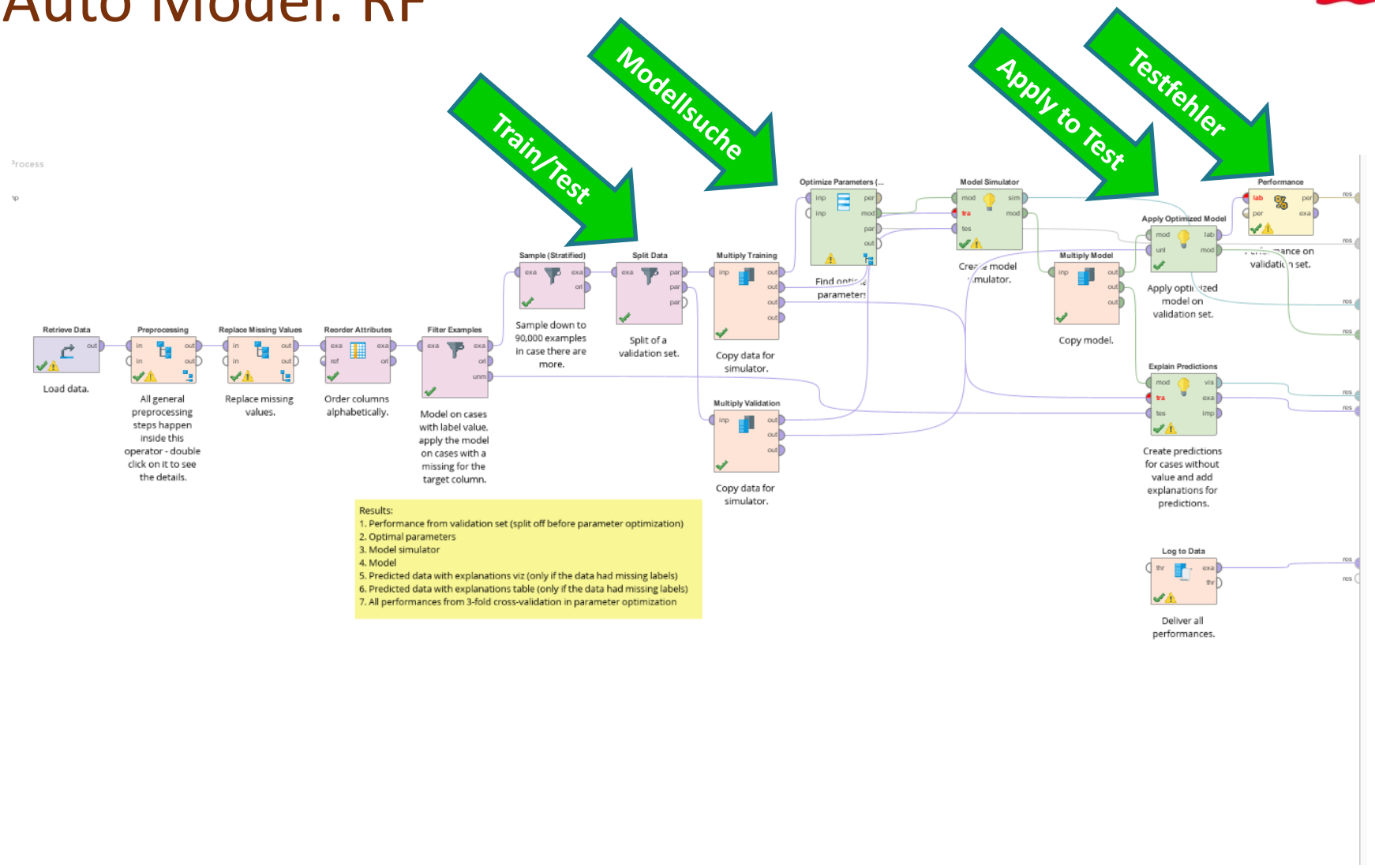


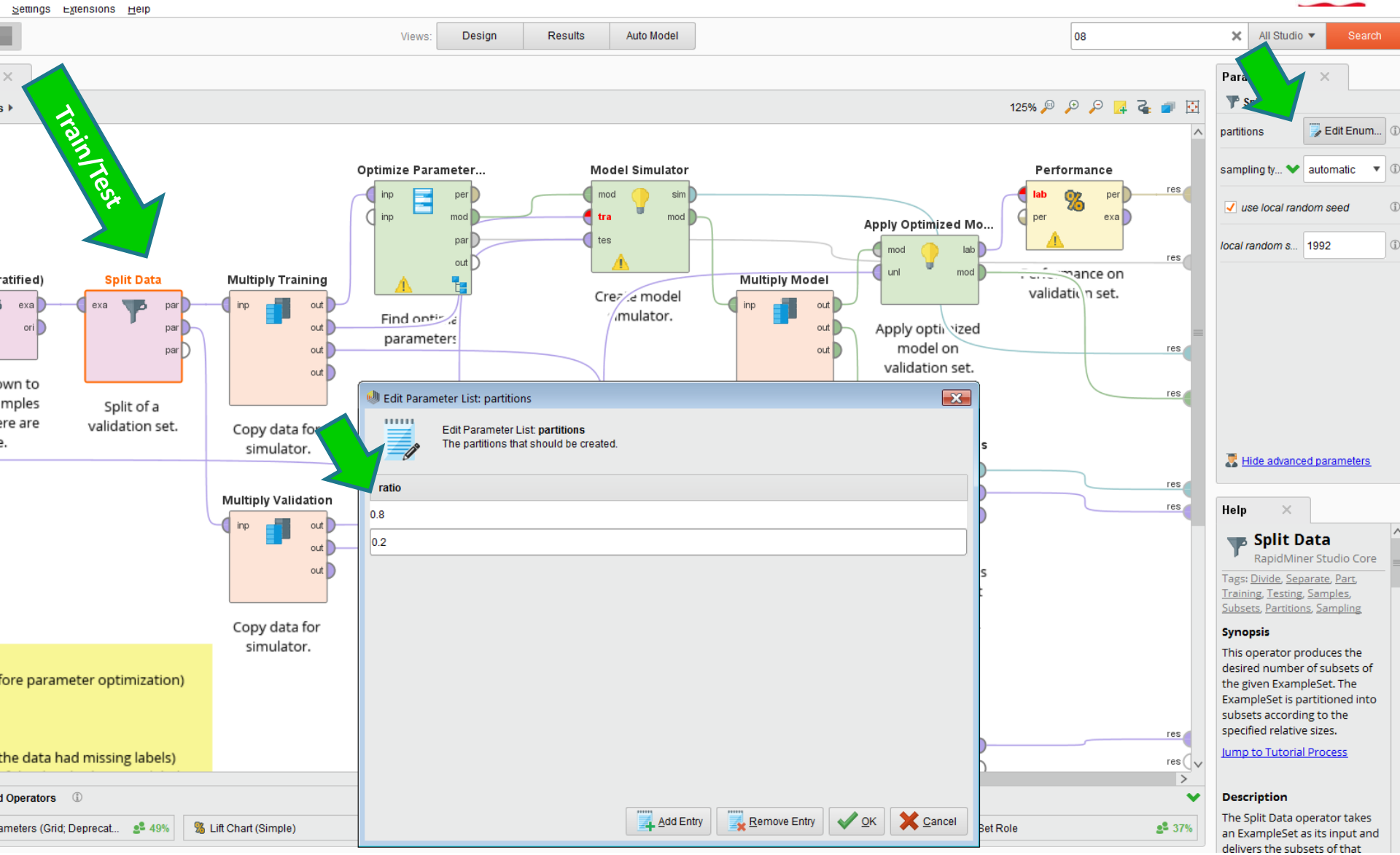

<new process*> – RapidMiner Studio Educational 8.1.001 @ kilab10n
File Edit Process View Connections Cloud Settings Extensions Help
Views: Design Results Auto Model
08
All Studio
Search
Auto Model
Select Data
Select Task
Prepare Target
Select Inputs
Model Types
Results
Results
Random Forest - Optimal Parameters
General
Data
Weights
Correlations
Comparison
Overview
Naive Bayes
Model
Simulator
Performance
Generalized Linear Model
Deep Learning
Model
Simulator
Performance
Decision Tree
Random Forest
Model
Simulator
Performance
Optimal Parameters
Gradient Boosted Trees
Optimal Parameters
Number Of Trees: 100
Maximal Depth: 7
Performance for Parameters
Maximal Depth
Number of Trees
20, 7
60, 7
100, 7
140, 7
20, 4
60, 4
100, 4
140, 4
60, 2
100, 2
Number of Trees | Maximal Depth | Performance
20 | 2 | 0.477
60 | 2 | 0.518
100 | 2 | 0.533
140 | 2 | 0.520
20 | 4 | 0.529
60 | 4 | 0.584
Back
Open Process
Information
Results: Classification
This is the final step of Auto Model, where you can inspect the generated models together with other results. The output depends on the data and the choices you made. For example, if you deactivated the calculation of correlations or decision trees, those results will not be displayed. Other results might be shown only for certain types of problems. Lift Charts, for example, are only available for two-class problems.
Please note that the results are calculated in the background. However, you can immediately start to inspect the results as they are completed. You can stop background execution by pressing the Stop button at the bottom. Calculations which are not completed when execution is stopped won't be available. You can go back and make changes after the execution is finished or after you stopped it.
We at RapidMiner do not believe in black boxes. This is why you can always open the process which created the model and all related results. Simply click on a model result and on Open Process at the bottom of the screen. This will show you the process which performs all necessary data preprocessing and model optimization. You can use this process for deploying the model or as a starting point for further optimizations.
We will now discuss the possible results in detail below.
General

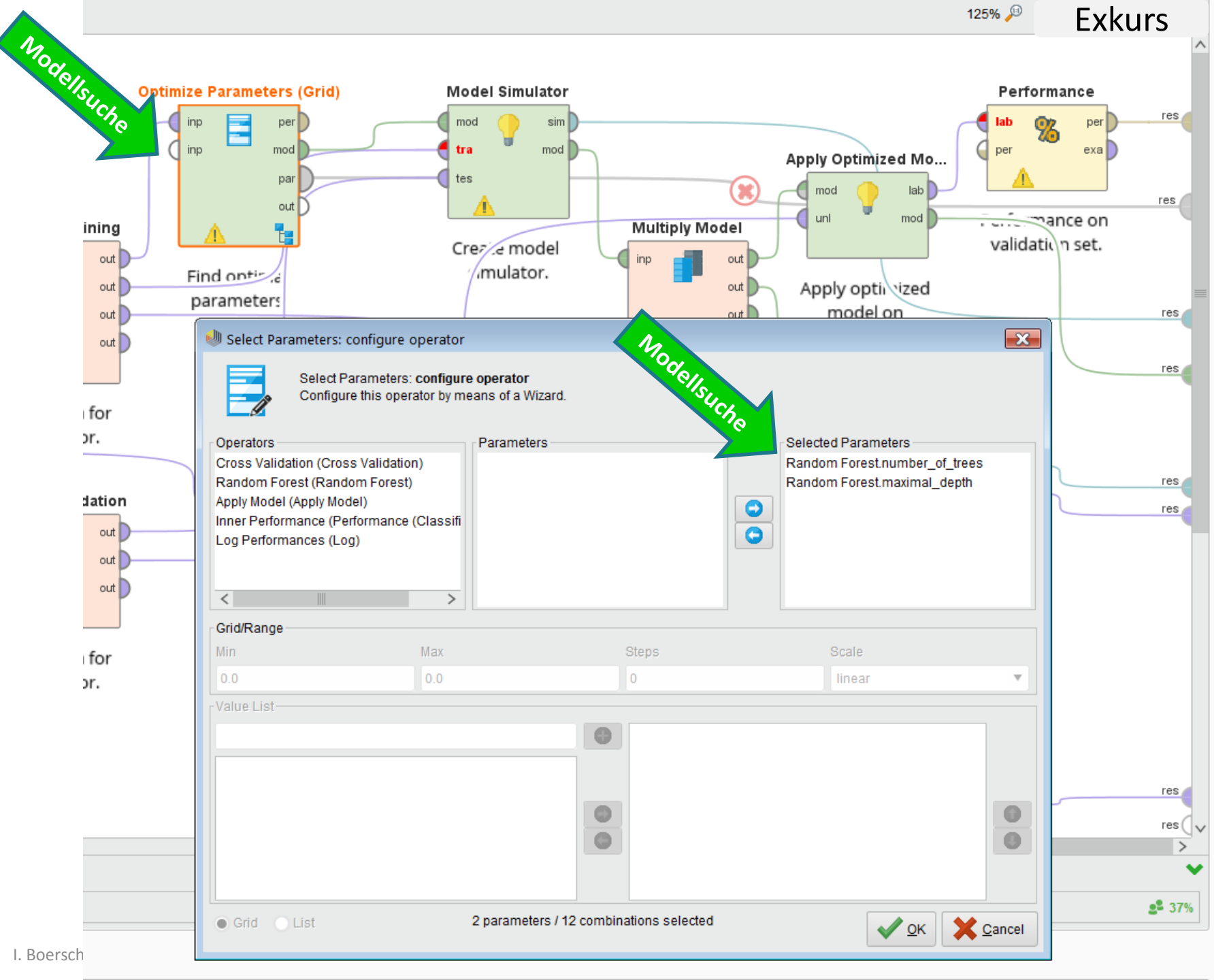

<new process*> – RapidMiner Studio Educational 8.1.001 @ kilab10n
File Edit Process View Connections Cloud Settings Extensions Help
Views: Design Results Auto Model
08
All Studio
Search
Auto Model
Select Data
Select Task
Prepare Target
Select Inputs
Model Types
Results
Results
Deep Learning - Simulator
General
Data
Weights
Correlations
Comparison
Overview
Naive Bayes
Model
Simulator
Performance
Generalized Linear Model
Deep Learn
Model
Simulator
Performance
Decision Tree
Random Forest
Model
Simulator
Performance
Optimal Parameters
Gradient Boosted Trees
fBodyAcc-entropy()-X: 0,62322516
fBodyAcc-entropy()-Y: -0,28109259
fBodyAcc-entropy()-Z: 0,15896413
fBodyAccJerk-entropy()-X: 0,49376381
fBodyAccJerk-entropy()-Y: 0,41428603
fBodyAccJerk-entropy()-Z: 0,15049278
fBodyAccJerk-maxInds-X: -1,0
fBodyAccJerk-maxInds-Y: 0,44
fBodyAccJerk-maxInds-Z: 0,8
fBodyAccMag-entropy(): 0,41536208
fBodyAccMag-maxInds: -1,0
Find the Optimal
Click 'Optimize' to ge
umal inputs for above.
Optimize
Most Likely: LAYING
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
100%
WALKING...
WALKING...
WALKING
STANDING
SITTING
LAYING
Important Factors for LAYING
fBodyAccMag-maxInds
fBodyGyro-maxInds-Y
fBodyGyro-maxInds-Z
fBodyAcc-maxInds-X
tBodyAccJerk-entropy()-Z
fBodyAcc-entropy()-X
fBodyAccJerk-maxInds-X
-0.100
-0.075
-0.050
-0.025
0.000
0.025
0.050
0.075
0.100
Supports 'LAYING'
Contradicts 'LAYING'
Back
Open Process
Information
Results: Classification
This is the final step of Auto Model, where you can inspect the generated models together with other results. The output depends on the data and the choices you made. For example, if you deactivated the calculation of correlations or decision trees, those results will not be displayed. Other results might be shown only for certain types of problems. Lift Charts, for example, are only available for two-class problems.
Please note that the results are calculated in the background. However, you can immediately start to inspect the results as they are completed. You can stop background execution by pressing the Stop button at the bottom. Calculations which are not completed when execution is stopped won't be available. You can go back and make changes after the execution is finished or after you stopped it.
We at RapidMiner do not believe in black boxes. This is why you can always open the process which created the model and all related results. Simply click on a model result and on Open Process at the bottom of the screen. This will show you the process which performs all necessary data preprocessing and model optimization. You can use this process for deploying the model or as a starting point for further optimizations.
We will now discuss the possible results in detail below.
General

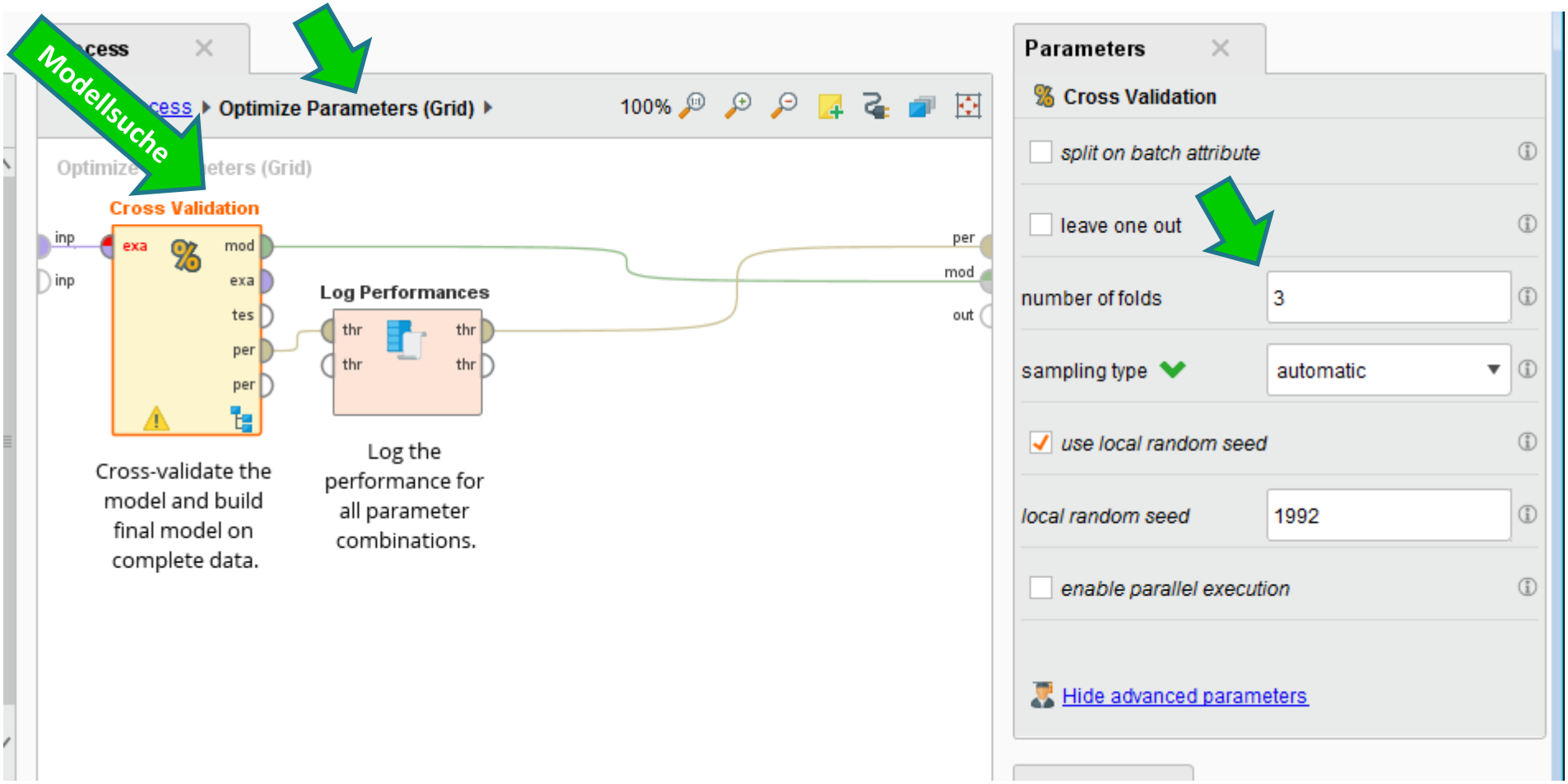

Find Optimal Input Settings
Define Targets
Define Constraints
Optimization Parameters
Running Optimization
The optimization is now running and determines the best input factors to meet your target under the specified constraints. The plot below shows how the fitness of the best solution develops over time. It should converge towards your goal over time.
Fitness
0.310
0.305
0.300
0.295
0.290
0.270
0.265
0.260
0.255
0.250
0.245
0.240
0.235
0.230
0.225
-1
0
1
2
3
4
5
6
7
8
9
11
12
13
14
15
16
17
18
Generation
Stop

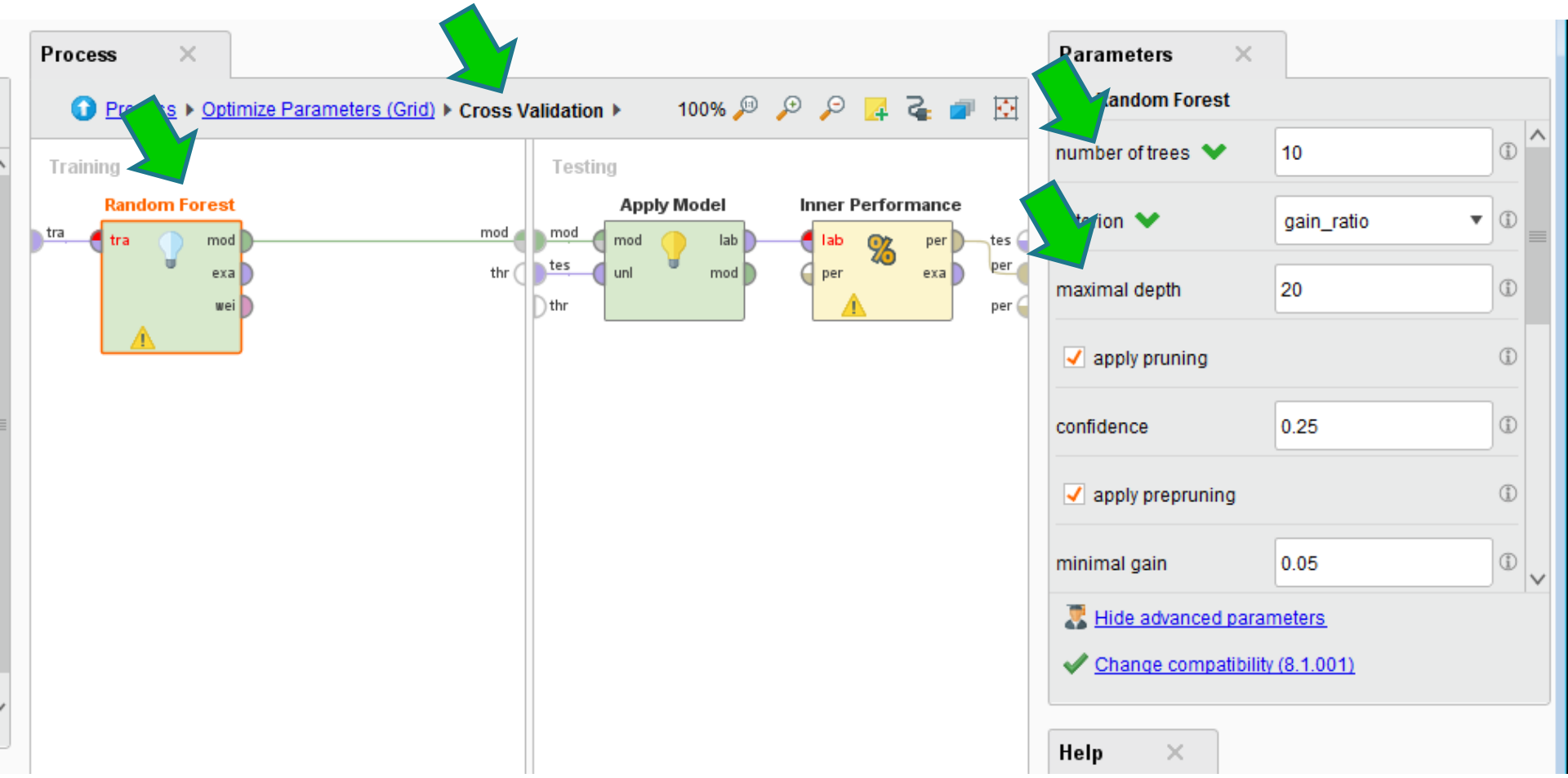
Auto Model: RF

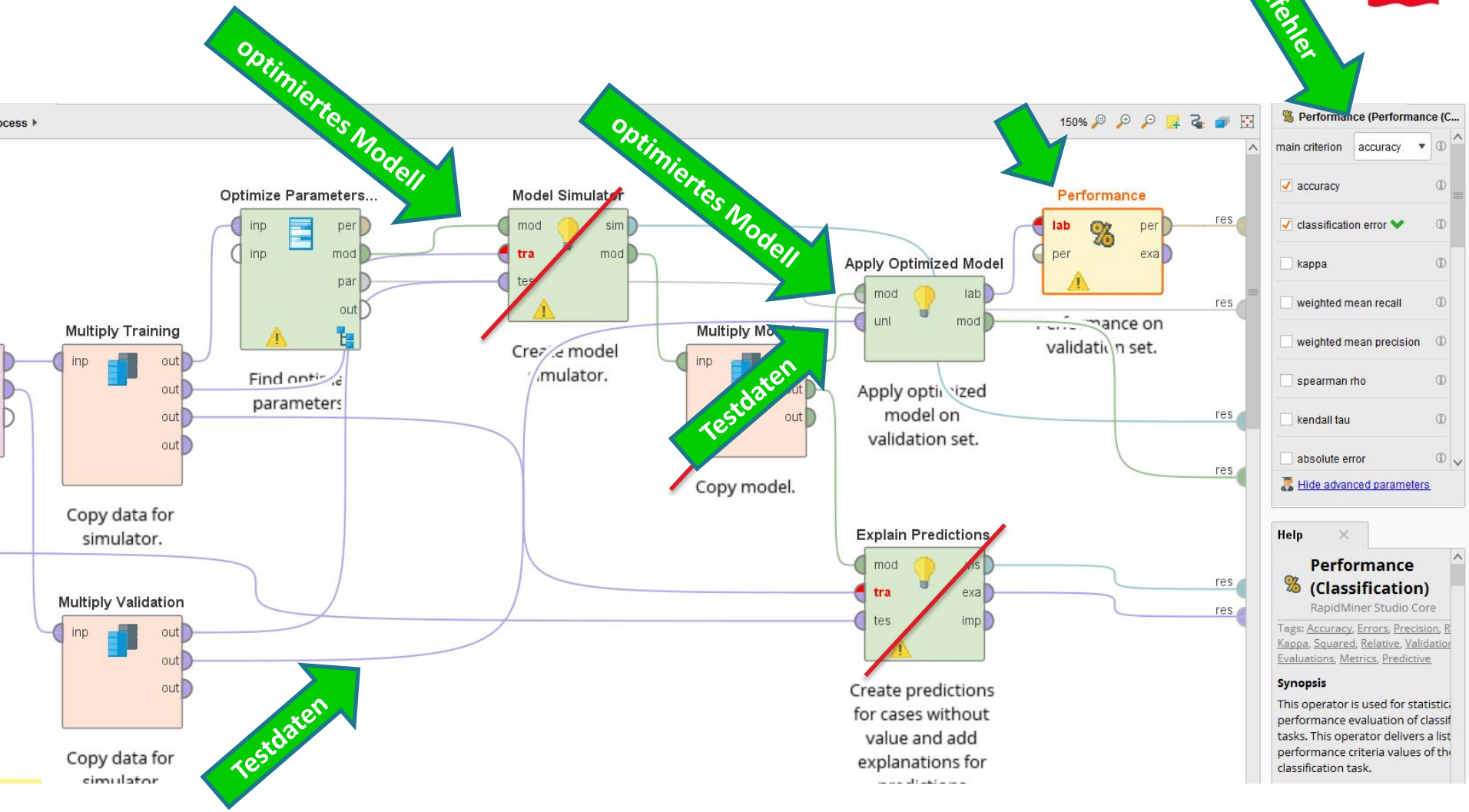



Settings Extensions Help
Views: Design Results Auto Model
08
All Studio
Search
Train/Test
125%
Optimize Parameter...
Model Simulator
Performance
Apply Optimized Mo...
Multiply Model
Split Data
Multiply Training
Split of a validation set.
Copy data for simulator.
Multiply Validation
Copy data for simulator.
Find optimal parameters
Create model simulator.
Apply optimized model on validation set.
Performance on validation set.
partitions
Edit Enum...
sampling ty...
automatic
use local random seed
local random s...
1992
Hide advanced parameters
Edit Parameter List: partitions
Edit Parameter List: partitions
The partitions that should be created.
ratio
0.8
0.2
Add Entry
Remove Entry
OK
Cancel
Help
Split Data
RapidMiner Studio Core
Tags: Divide, Separate, Part, Training, Testing, Samples, Subsets, Partitions, Sampling
Synopsis
This operator produces the desired number of subsets of the given ExampleSet. The ExampleSet is partitioned into subsets according to the specified relative sizes.
Jump to Tutorial Process
Description
The Split Data operator takes an ExampleSet as its input and delivers the subsets of that
fore parameter optimization)
the data had missing labels)
Operators
Lift Chart (Simple)


Modellsuche
Optimize Parameters (Grid)
Model Simulator
Performance
Apply Optimized Mo...
Multiply Model
Select Parameters: configure operator
Select Parameters: configure operator
Configure this operator by means of a Wizard.
Operators
Cross Validation (Cross Validation)
Random Forest (Random Forest)
Apply Model (Apply Model)
Inner Performance (Performance (Classifi
Log Performances (Log)
Parameters
Selected Parameters
Random Forest.number_of_trees
Random Forest.maximal_depth
Grid/Range
Min
Max
Steps
Scale
Value List
Grid
List
2 parameters / 12 combinations selected
OK
Cancel
Modellsuche

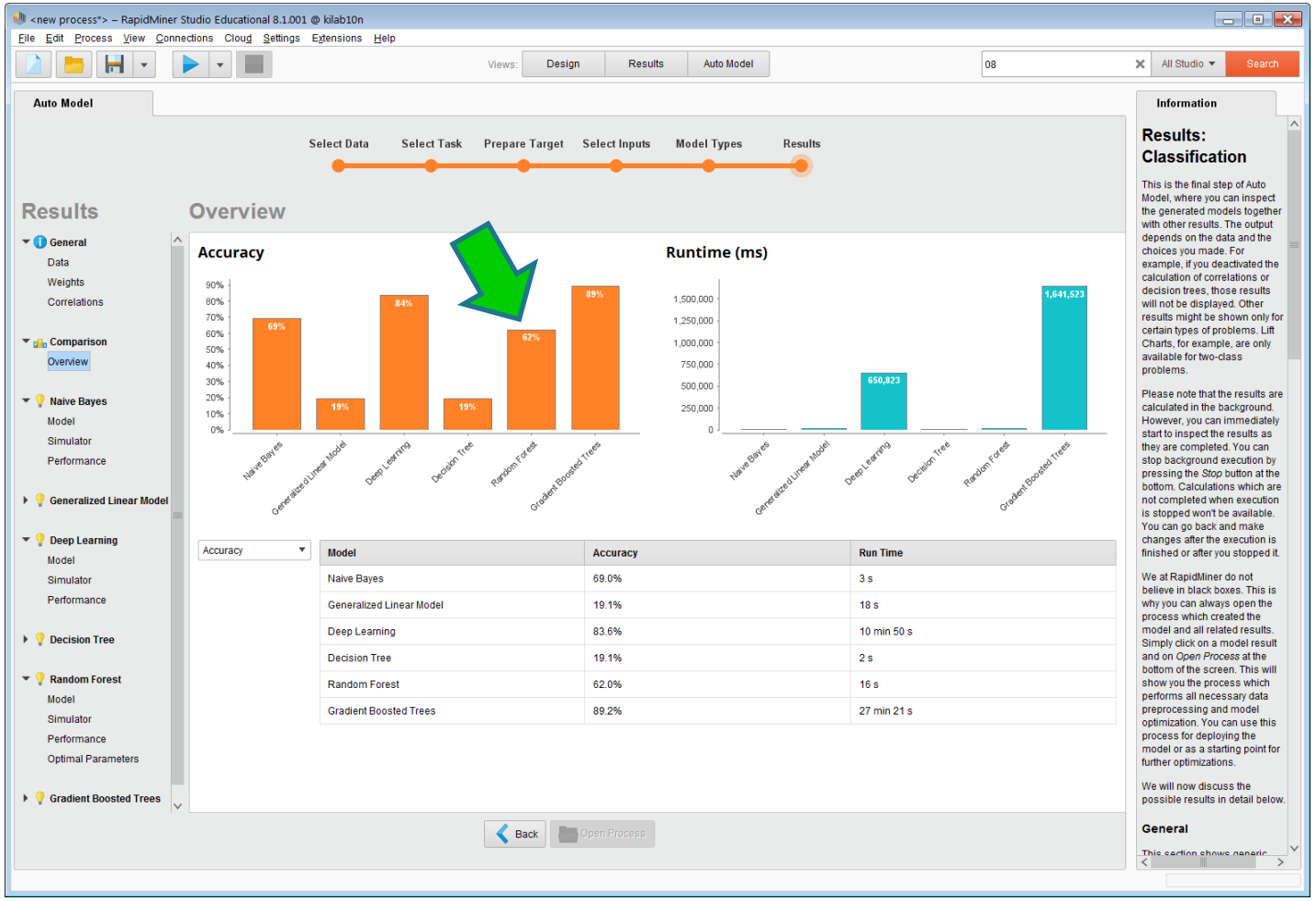

Modellsuche
Optimize Parameters (Grid)
100%
Cross Validation
inp
inp
exa
mod
exa
tes
per
per
Log Performances
thr
thr
thr
thr
per
mod
out
Cross-validate the model and build final model on complete data.
Log the performance for all parameter combinations.
Parameters
Cross Validation
split on batch attribute
leave one out
number of folds
3
sampling type
automatic
use local random seed
local random seed
1992
enable parallel execution
Hide advanced parameters


Process
Optimize Parameters (Grid)
Cross Validation
100%
Training
Testing
Random Forest
Apply Model
Inner Performance
tra
mod
exa
wei
thr
unl
lab
per
tes
Parameters
Random Forest
number of trees
10
gain_ratio
maximal depth
20
apply pruning
confidence
0.25
apply prepruning
minimal gain
0.05
Hide advanced parameters
Change compatibility (8.1.001)
Help


optimiertes Modell
optimiertes Modell
Testfehler
Testdaten
Testdaten
Optimize Parameters...
Model Simulator
Performance
Apply Optimized Model
Multiply Training
Multiply Validation
Explain Predictions
Copy data for simulator.
Copy model.
Apply optimized model on validation set.
Create predictions for cases without value and add explanations for
Performance (Performance (C...
main criterion
accuracy
classification error
kappa
weighted mean recall
weighted mean precision
spearman rho
kendall tau
absolute error
Hide advanced parameters
Help
Performance (Classification)
RapidMiner Studio Core
Synopsis

Auto Model: Prediction Task

Training geeigneter Modelle, teilweise mit Parameteroptimierung



Rückblick

Performance von Klassifikatoren

- Konfusionsmatrix
- Fehlermaße
- Trainingsfehler nicht als Schätzung geeignet
- Bsp. k-Nearest Neighbor
- (Overfitting)
- Fehlerschätzung: Kreuzvalidierung
- Praktisches Vorgehen, 100%-Modell