Vorlesung
Wissensentdeckung in Datenbanken
Einführung

Kristian Kersting, (Katharina Morik), Claus Weihs

LS 8 Informatik
Computergestützte Statistik
Technische Universität Dortmund

10.4.2014

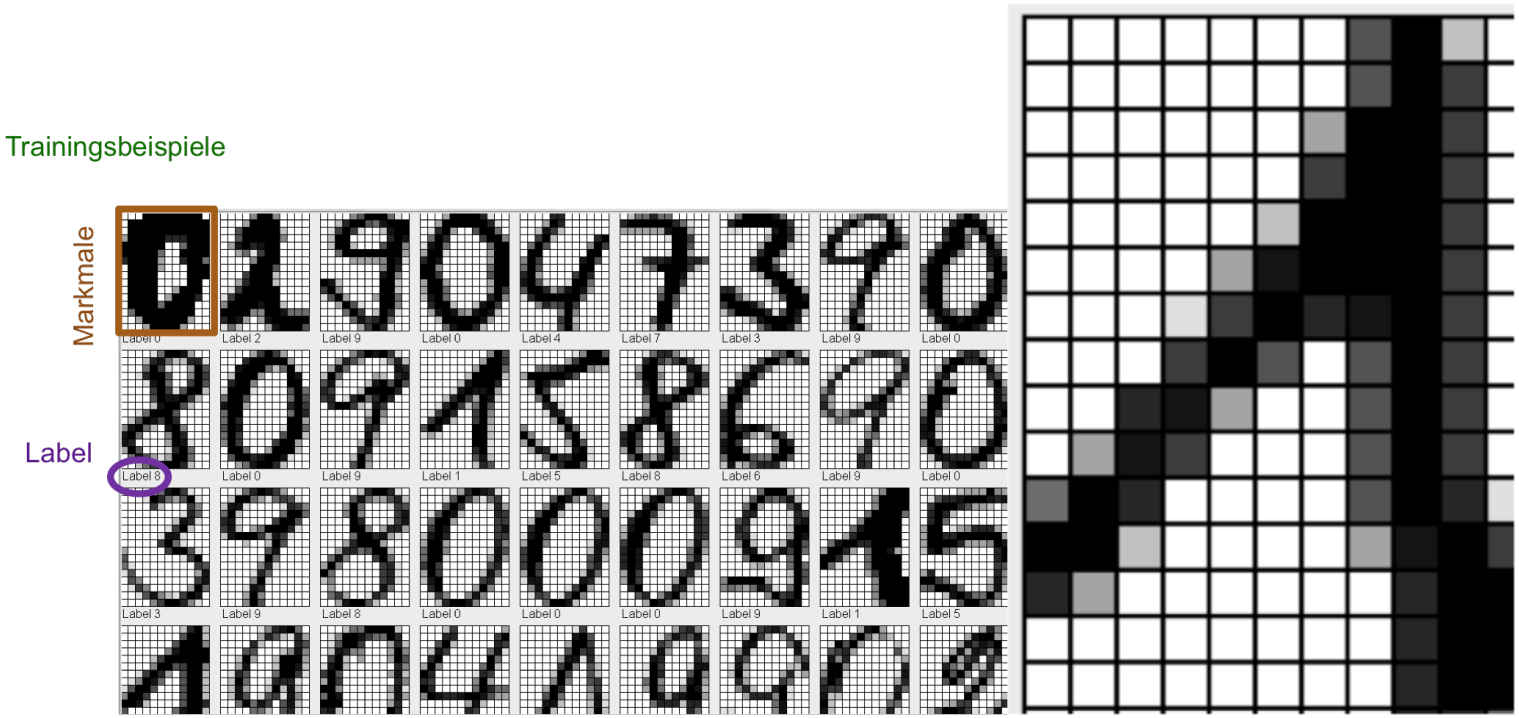
Was wissen Sie bisher?

- Daten sind das neue Öl!
- Data Scientists sind in Industrie und Forschung gefragt!
- Sie haben Anwendungsbeispiele gesehen.
- Als wesentliche Aufgaben der Modellbildung haben Sie Clustering, Klassifikation, Regression gesehen.

Heute

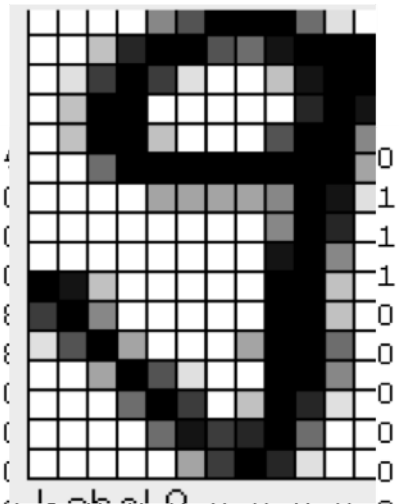
1. Modellbildung und Evaluation
2. Verlaufsmodell der Wissensentdeckung
3. Einführung in das Werkzeug RapidMiner

Welche Zahl sehen wir hier?



Bilder als Vektoren

- Zwei Bilder repräsentieren die gleiche Ziffer, wenn die Bilder sich ähnlich sind
- Ähnlich = ähnliche Grauwertverteilung
- Wir stellen Bilder als eine Matrix von Grauwerten dar.
- Ziffer $=12 \times 16$ Matrix von Grauwerten in $[0,1]$.
- Ziffer $=$Vektor von Grauwerten der Länge 192.



0.0	0.0	0.0	0.2	0.3	0.4					0.3	0.1
0.0	0.5	0.8	0.9	1.0	1.0					1.0	0.5
0.3	1.0	1.0	1.0	1.0	1.0					1.0	0.9
0.3	1.0	1.0	1.0	1.0	1.0					1.0	0.8
0.3	1.0	1.0	1.0	1.0	0.8					0.8	0.2
0.0	0.7	1.0	1.0	1.0	0.8					0.4	0.0
0.0	0.2	1.0	1.0	1.0	1.0					0.0	0.0
0.0	0.1	0.7	1.0	1.0	1.0					0.0	0.0
0.0	0.6	1.0	1.0	1.0	1.0					0.0	0.0
0.6	1.0	1.0	1.0	1.0	1.0	1.0	0.3	0.0	0.0	0.0	0.0
0.8	1.0	1.0	0.5	0.1	0.7	1.0	0.8	0.2	0.0	0.0	0.0
0.5	1.0	1.0	0.3	0.0	0.0	0.9	1.0	0.9	0.1	0.0	0.0
0.4	1.0	1.0	0.3	0.0	0.0	0.5	1.0	1.0	0.5	0.0	0.0
0.0	0.4	1.0	1.0	0.5	0.3	0.5	1.0	1.0	1.0	0.2	0.0
0.0	0.0	0.5	1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.1	0.0
0.0	0.0	0.0	0.2	0.5	0.7	1.0	1.0	0.9	0.3	0.0	0.0

Ein erster Data Mining Algorithmus: *k*-Nearest-Neighbors

- Sie werden in der Vorlesung einige Verfahren kennenlernen. Diese können in globale und lokale Ansätze unterteilt werden
- Lineare Modelle und und die Stützvektormethode (später in der Vorlesung) finden eine trennende Hyperebene.
- Die Ebene trennt alle Beispiele auf einmal. Lineare Modelle sind daher global.
- Klassifiziert man ein Beispiel nur anhand der Beispiele seiner Umgebung, spricht man von einem lokalen Modell.
- Nächste Nachbarn sind ein lokales Modell.

Nächste Nachbarn

- Das kNN-Modell betrachtet nur die k nächsten Nachbarn eines Beispiel $\vec{x}$:

$$\hat{f}(\vec{x}) = \frac{1}{k} \sum_{\vec{x}_i \in N_k(\vec{x})} y_i \tag{1}$$

- Die Nachbarschaft $N_k(\vec{x})$ wird durch ein Abstandsmaß, z.B. den Euklidschen Abstand bestimmt.
- Wenn sich die Nachbarschaften nicht überlappen, gibt es maximal $\frac{N}{k}$ Nachbarschaften und in jeder bestimmen wir den Durchschnitt (1).

Regression und Klassifikation

Regression $f : X \to Y$ mit $y \in \mathcal{R}$

Gleichung (1) gibt als Regressionsfunktion den Mittelwert der y_i zurück.

$$\hat{f}(\vec{x}) = \frac{1}{k} \sum_{\vec{x}_i \in N_k(\vec{x})} y_i$$

Klassifikation $f : X \to Y$ mit $y \in \{+1, -1\}$

Durch einen Schwellwert können wir aus der Regression eine Klassifikation machen:

$$\hat{y} = \begin{cases} 1, & \textit{falls } \hat{f}(\vec{x}) \geq 0,5 \\ 0, & \textit{sonst} \end{cases}$$

Die grünen und roten Datenpunkte werden in Nachbarschaften gruppiert

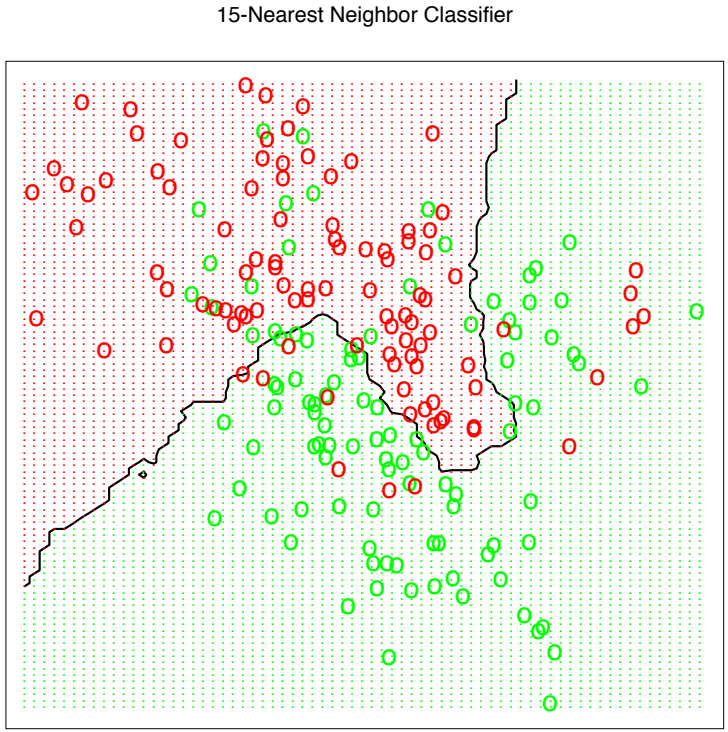
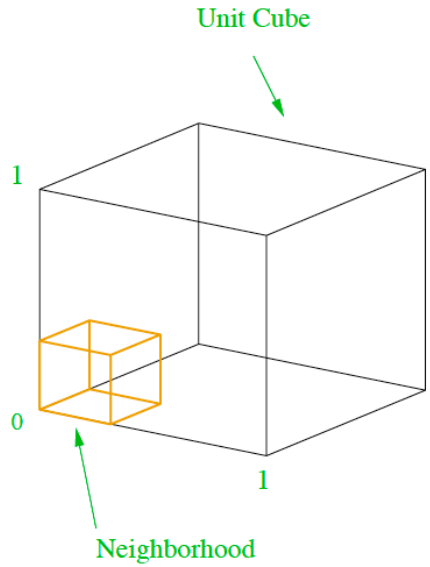


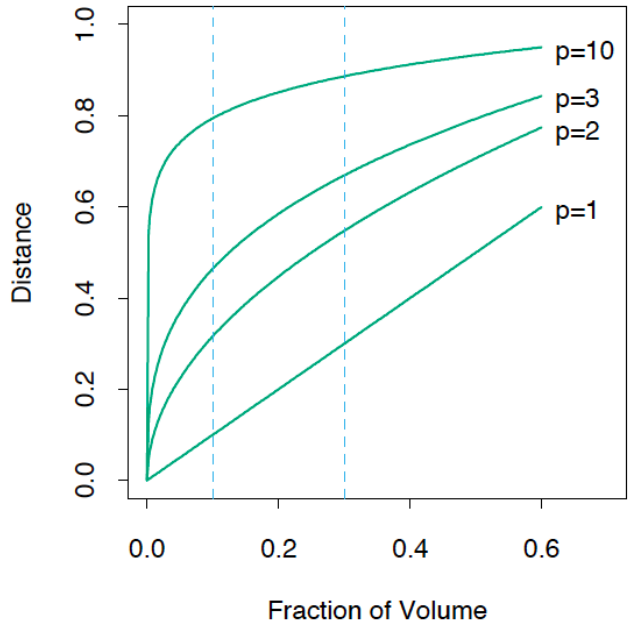
Figure 2.2: *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (GREEN $= 0$, RED $= 1$) *and then fit by* 15*-nearest-neighbor averaging as in (2.8). The predicted class is hence chosen by majority vote amongst the* 15*-nearest neighbors.*

Und ganz allgemein: Der kNN is asymptotisch korrekt!

- Wenn N, k gegen ∞ gehen, so dass das Verhältnis k/N gegen 0 geht, dann konvergiert die vorhergesagte Klasse auch zur optimalen Vorhersage $E(Y|X = x)$ unter schwachen Annahmen über die Verbundwahrscheinlichkeit $P(X, Y)$. (Hastie/etal/2001, S. 19)
- Haben wir also den perfekten Lernalgorithmus gefunden? Mission completed?

Fluch der hohen Dimension bei *k*NN





- Seitenlänge des Subquaders um *r*-Prozent des Volumens abzudecken für verschiedene Dimensionen *p*.
- Wenn 100 Beispiele bei $p = 1$ einen dichten Raum ergeben, muss man für die selbe Dichte $N^{1/p}$ bei $p = 10$ schon 100^{10} Beispiele sammeln: $100^{1/1} = 100, 100^{10 \cdot \frac{1}{10}} = 100$

Wir brauchen zu viele Beispiele!

Mission not completed !

Problem

- Wir haben nur eine endliche Menge von Beispielen. Alle Funktionen, deren Werte durch die Beispiele verlaufen, haben einen kleinen Fehler.
- Wir wollen aber für alle Beobachtungen das richtige y voraussagen. Dann sind nicht mehr alle Funktionen, die auf die Beispiele gepasst haben, gut.
- Wir kennen nicht die wahre Verteilung der Beispiele.
- Wie beurteilen wir da die Qualität unseres Lernergebnisses?

Reicht Auswendiglernen?

- Falls $\vec{x} = \vec{x}' \rightarrow y = y'$, gibt es bei $k = 1$ keinen Trainingsfehler.
- Wenn allein der Trainingsfehler das Optimierungskriterium ist, würden wir stets $k = 1$ nehmen und nur auswendig lernen.
- Vermutlich ergibt das aber auf noch nicht gesehenen Daten einen großen Fehler!

Overfitting: Gut auf den gegebenen Daten, aber schlechte Generalisierung

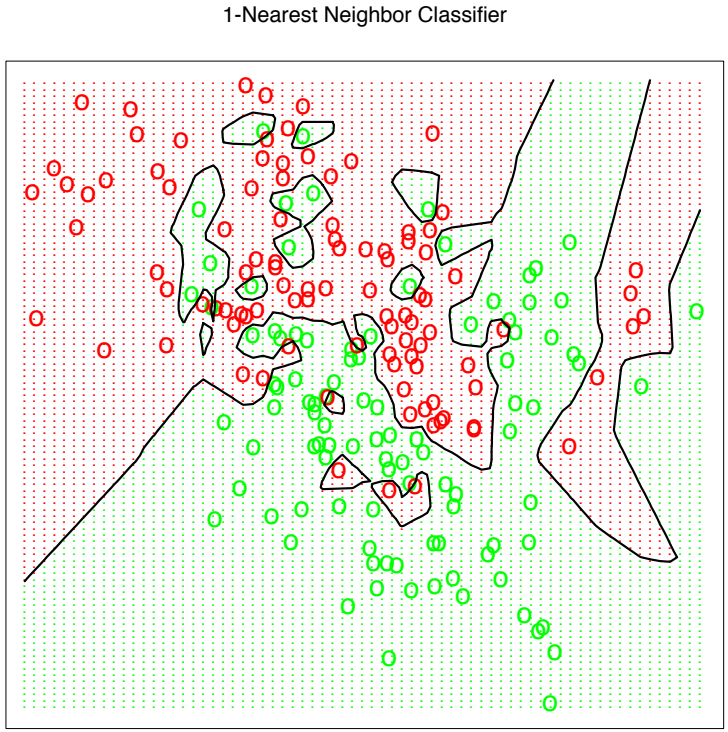


Figure 2.3: *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (GREEN $= 0$, RED $= 1$), *and then predicted by* 1*-nearest-neighbor classification.*

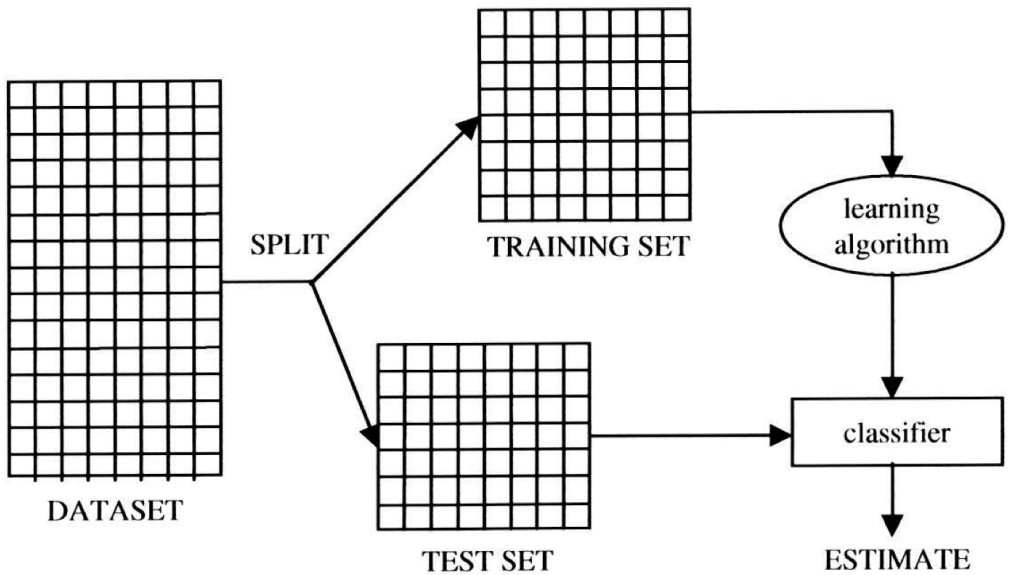
Besser: Lern- und Testmenge

Wir teilen die Daten, die wir haben, auf:

Lernmenge: Einen Teil der Daten übergeben wir unserem Lernalgorithmus. Daraus lernt er seine Funktion $f(x) = \hat{y}$.

Testmenge: Bei den restlichen Daten vergleichen wir $\hat{y}$ mit y.

Besser: Lern- und Testmenge



Aufteilung in Lern- und Testmenge

- Vielleicht haben wir zufällig aus lauter Ausnahmen gelernt und testen dann an den normalen Fällen. Um das zu vermeiden, verändern wir die Aufteilung mehrfach.

  leave-one-out: Der Algorithmus lernt aus $N - 1$ Beispielen und testet auf dem ausgelassenen. Dies wird N mal gemacht, die Fehler addiert.

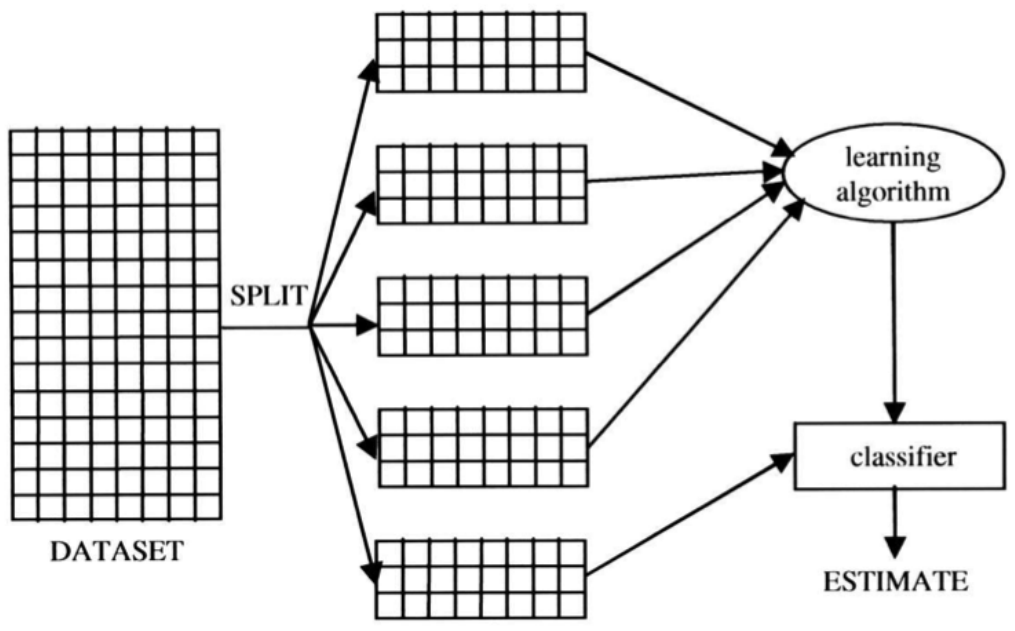
- Aus Zeitgründen wollen wir den Algorithmus nicht zu oft anwenden.

  Kreuzvalidierung: Die Lernmenge wird zufällig in n Mengen aufgeteilt. Der Algorithmus lernt aus $n - 1$ Mengen und testet auf der ausgelassenen Menge. Dies wird n mal gemacht.

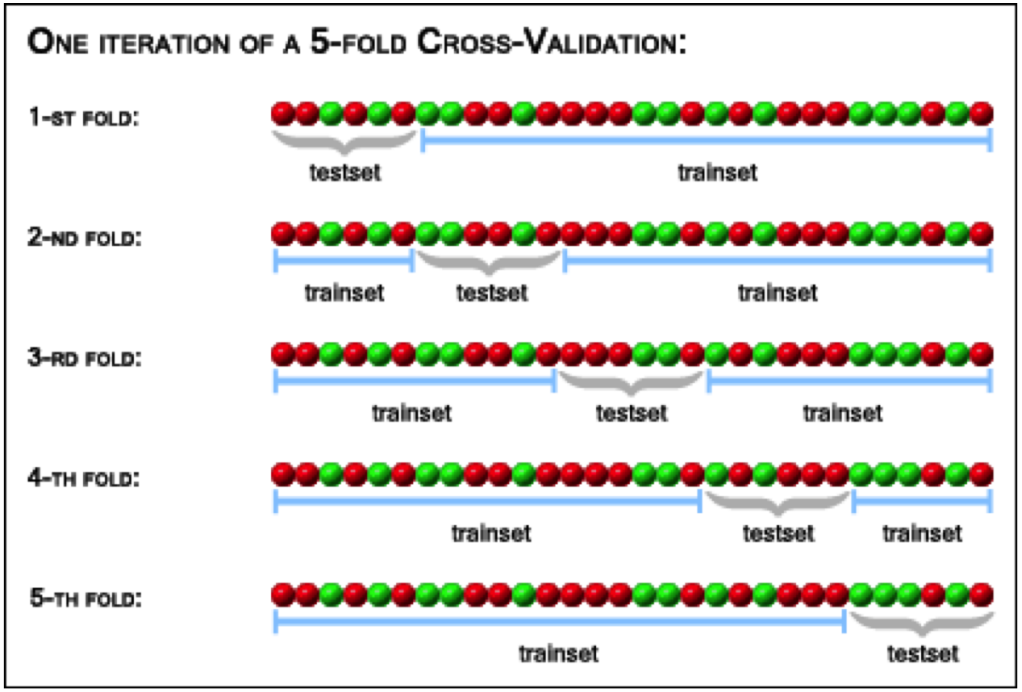
Kreuzvalidierung

- Man teile alle verfügbaren Beispiele in n Mengen auf. z.B. $n = 10$.
- Für i=1 bis i=n:
  - Wähle die i-te Menge als Testmenge,
  - die restlichen $n - 1$ Mengen als Lernmenge.
  - Messe die Qualität auf der Testmenge.
- Bilde das Mittel der gemessenen Qualität über allen n Lernläufen. Das Ergebnis gibt die Qualität des Lernergebnisses an.

5-fache Kreuzvalidierung



5-fache Kreuzvalidierung



Was wissen Sie jetzt?

- Sie haben Anwendungsbeispiele gesehen.
- Als Aufgaben der Modellbildung haben Sie Clustering, Klassifikation, Regression gesehen.
- Sie haben kNN kennengelernt.
- Sie haben den Fluch der hohen Dimension kennengelernt.
- Sie wissen, was die Kreuzvalidierung ist.

Was wissen Sie noch nicht?

- Es gibt viele verschiedene Modellklassen. Damit werden die Lernaufgaben spezialisiert.
- Es gibt unterschiedliche Qualitätsfunktionen. Damit werden die Lernaufgaben als Optimierungsaufgaben definiert.
- Es gibt auch noch mehr Aufgaben: Finden häufiger Mengen!
- Der Gesamtablauf des Data Mining hat eine feste Struktur, die mehr enthält als nur den Lernschritt.
- Das Ziehen von Stichproben und wie diese verwendet werden können, lernen Sie kennen.
- Die Dimensionsreduktion ist ein wichtiger Vorverarbeitungsschritt.
- ... und vieles, vieles, vieles mehr

Verlaufsmodell der Wissensentdeckung

Übersicht

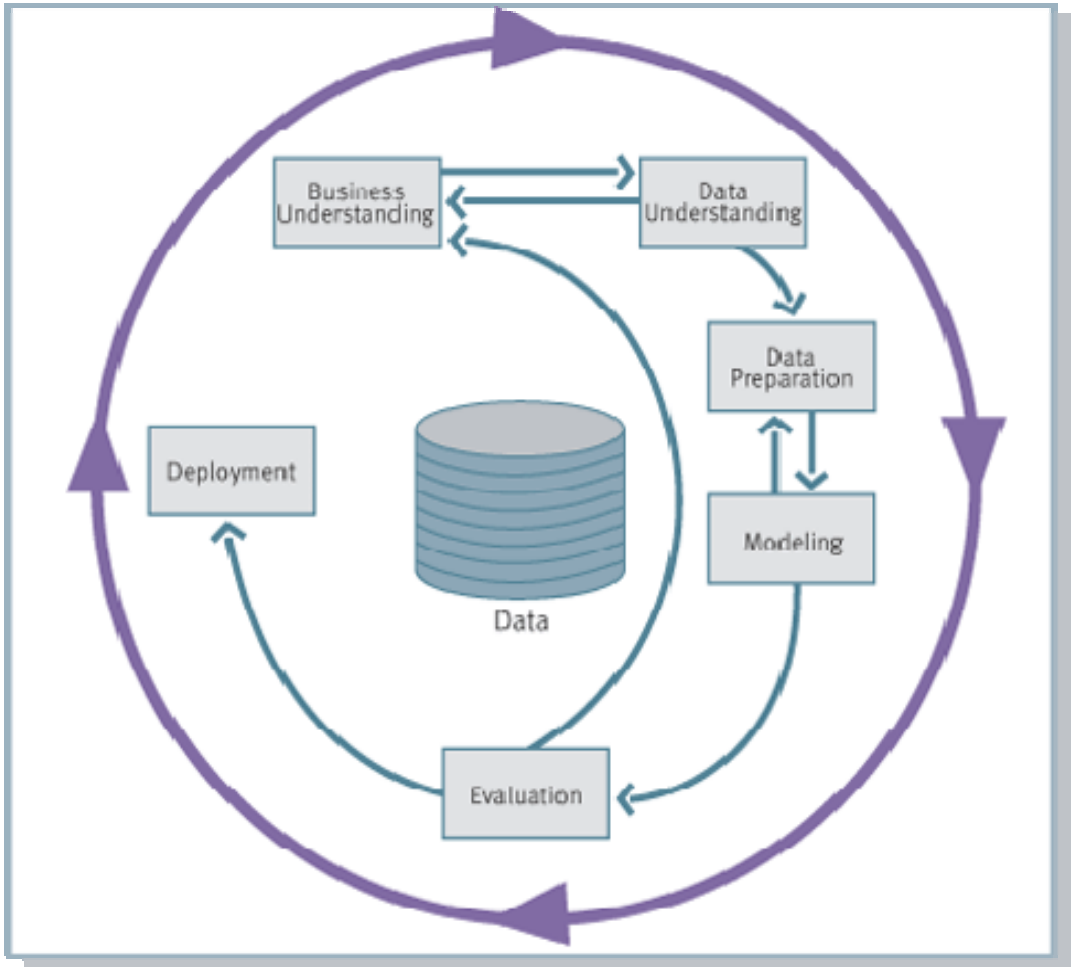


Figure : Übersicht über CRISP-DM

CRISP: Schritte

- **Problem verstehen**: Analyseziele, Situationsbewertung, Datenanalyseziele, Projektplan
- **Daten verstehen**: Sammeln, beschreiben, untersuchen, Qualität von Rohdaten
- **Daten aufbereiten**: Ein- und Ausschluss, Bereinigung, Transformation von Variablen
- **Modellierung**: Methoden- und Testdesignwahl, Schätzung, Modellqualität
- **Evaluierung**: Modell akzeptieren, Prozess überprüfen, nächste Schritte
- **Nachbereitung**: Anwendungs- und Wartungsplan, Präsentation, Bericht

Vorverarbeitung

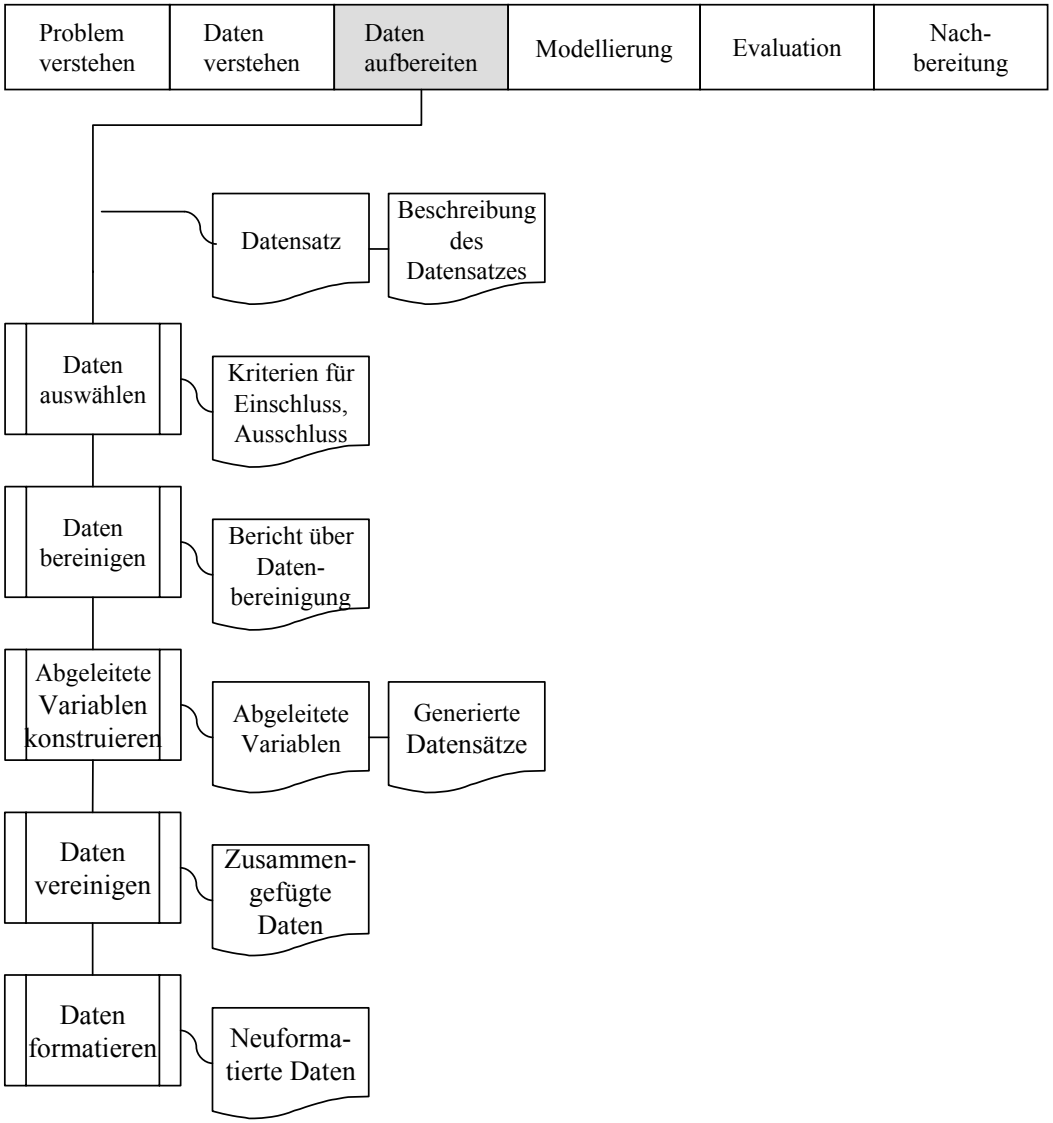


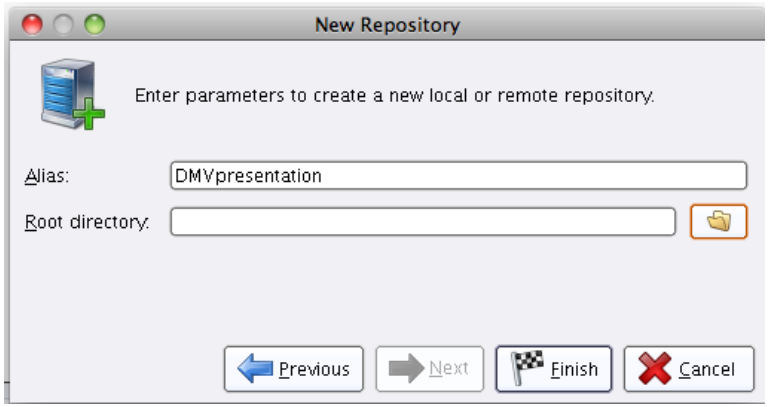
Figure : Datenvorverarbeitung bei CRISP-DM

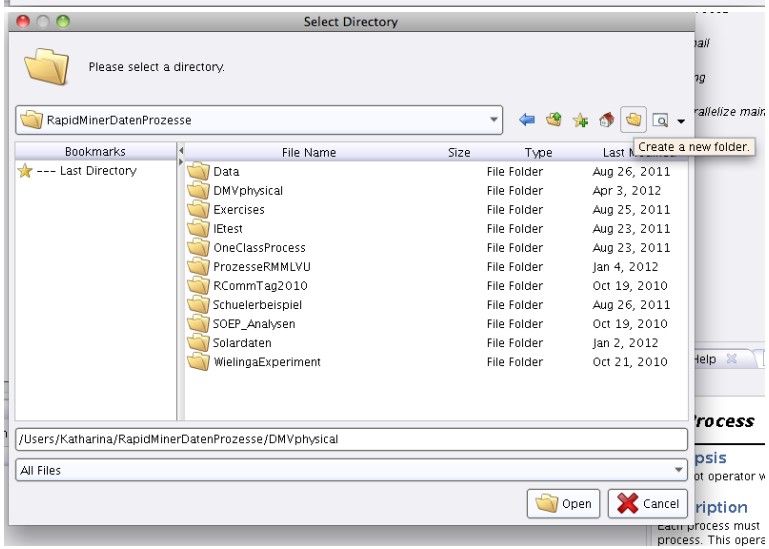
RapidMiner ist eine Umgebung, die den gesamten Prozess unterstützt.

- Operatoren für alle Verarbeitungsschritte,
- Prozess wird mit allen Parametern etc. dokumentiert –> Reproduzierbarkeit!
- Leicht zu erweitern durch eigene Operatoren, plug-ins.

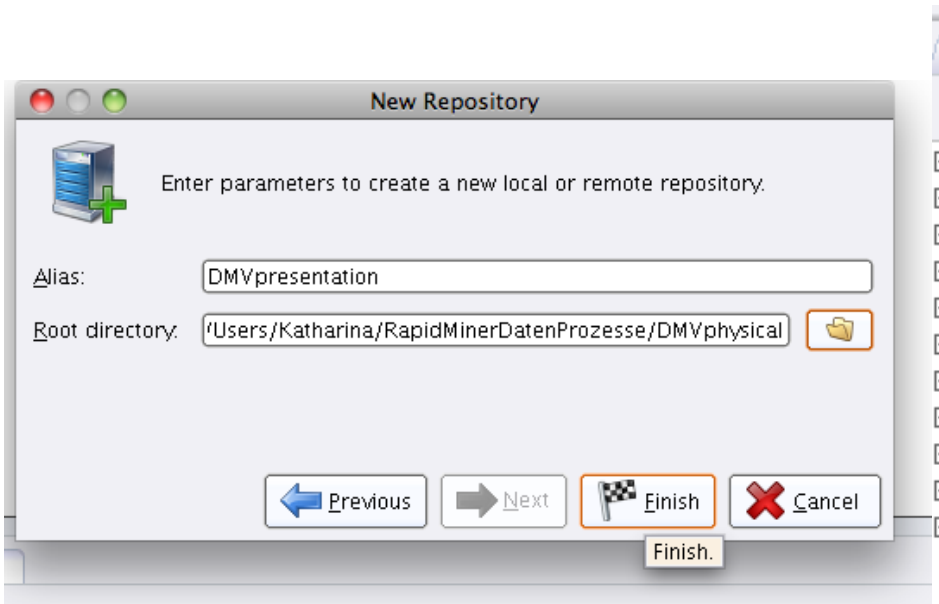
Prozesse und Daten speichern

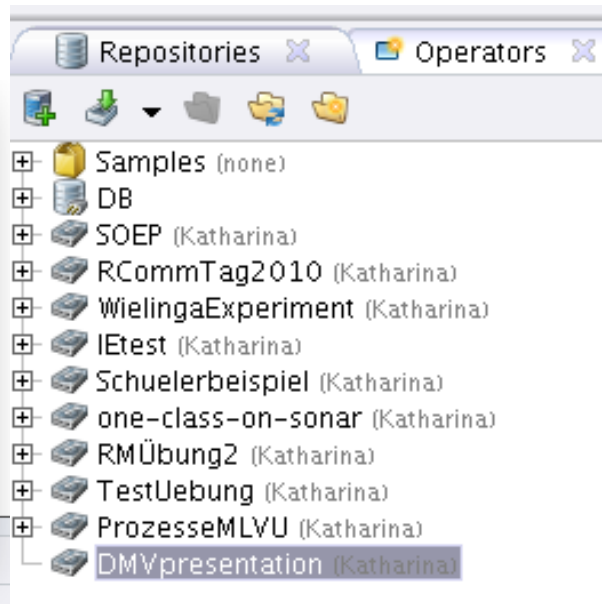
Für Ihre Daten und Prozesse müssen Sie einen Ort auf Ihrem Rechner vorsehen. Dieser Ort erhält einen symbolischen Namen als *Alias*und wird physikalisch bezeichnet durch ein *Root directory*.



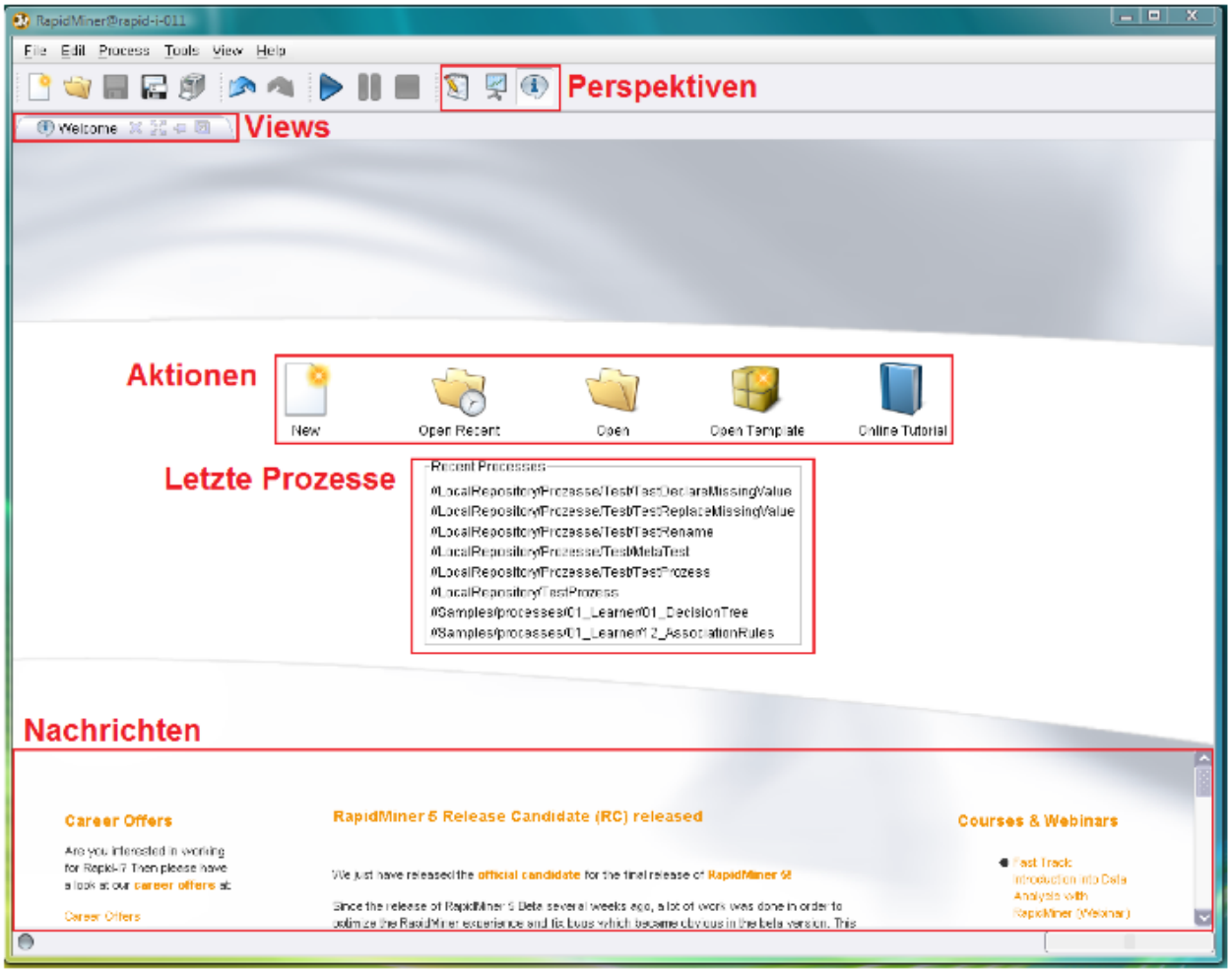


Repository

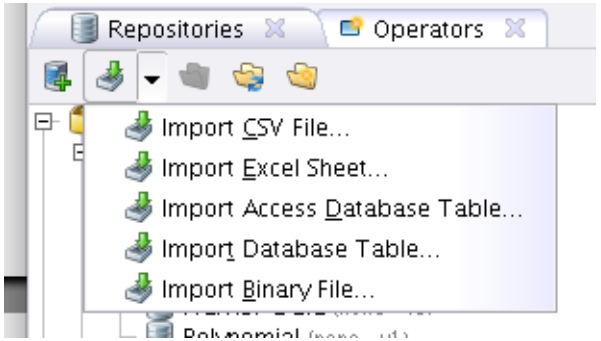

New Repository
Enter parameters to create a new local or remote repository.
Alias:
DMVpresentation
Root directory:
/Users/Katharina/RapidMinerDatenProzesse/DMVphysical
Previous
Next
Finish
Cancel
Finish.


Repositories
Operators
Samples (none)
DB
SOEP (Katharina)
RCommTag2010 (Katharina)
WielingaExperiment (Katharina)
IEtest (Katharina)
Schuelerbeispiel (Katharina)
one-class-on-sonar (Katharina)
RMÜbung2 (Katharina)
TestUebung (Katharina)
ProzesseMLVU (Katharina)
DMVpresentation (Katharina)

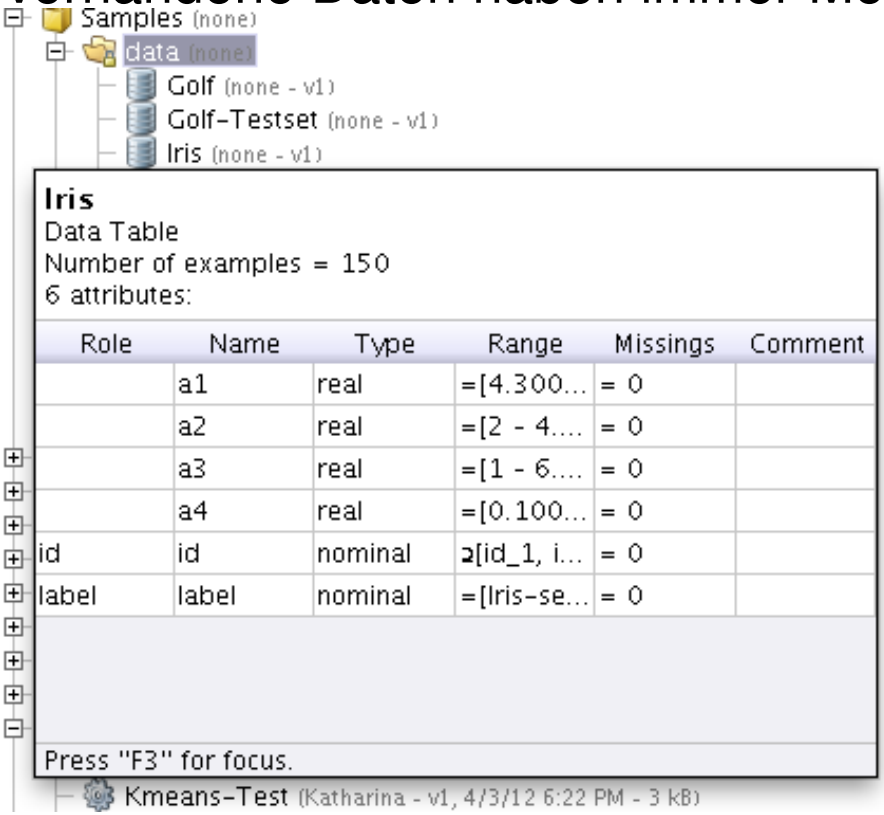
RapidMiner Ansicht



Daten einlesen

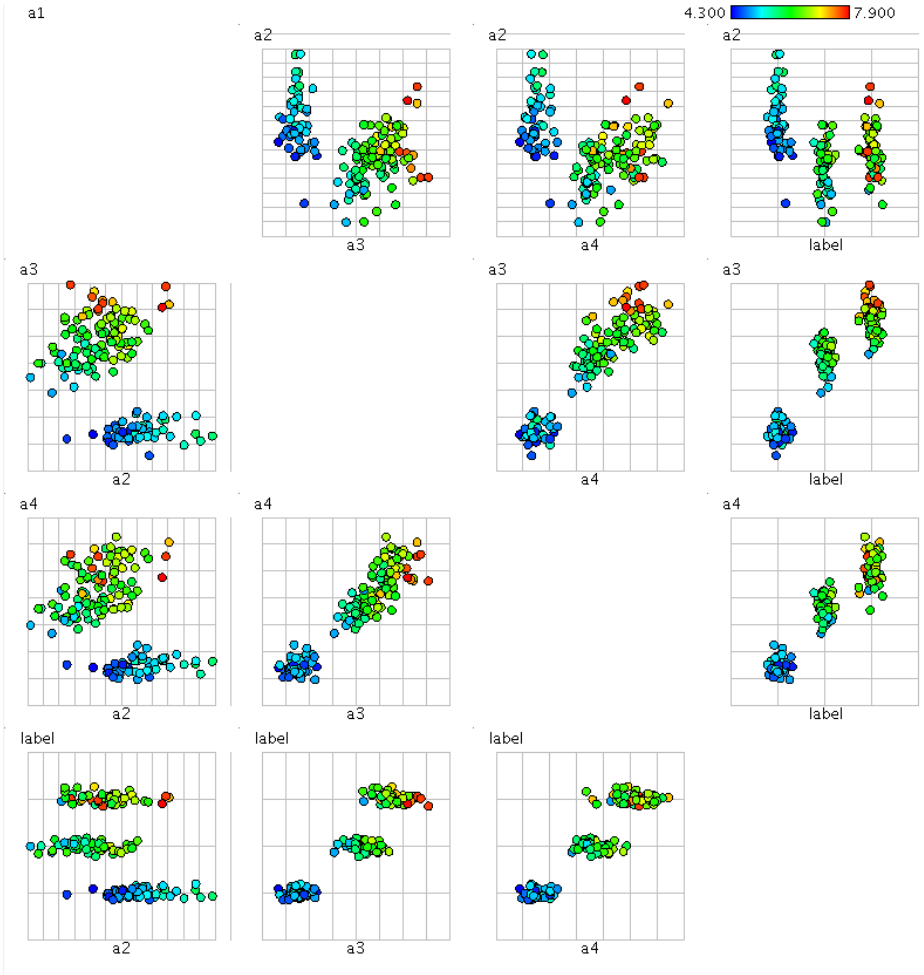


Einlesen ist für viele Formate vorbereitet.
Vorhandene Daten haben immer Metadaten dabei!



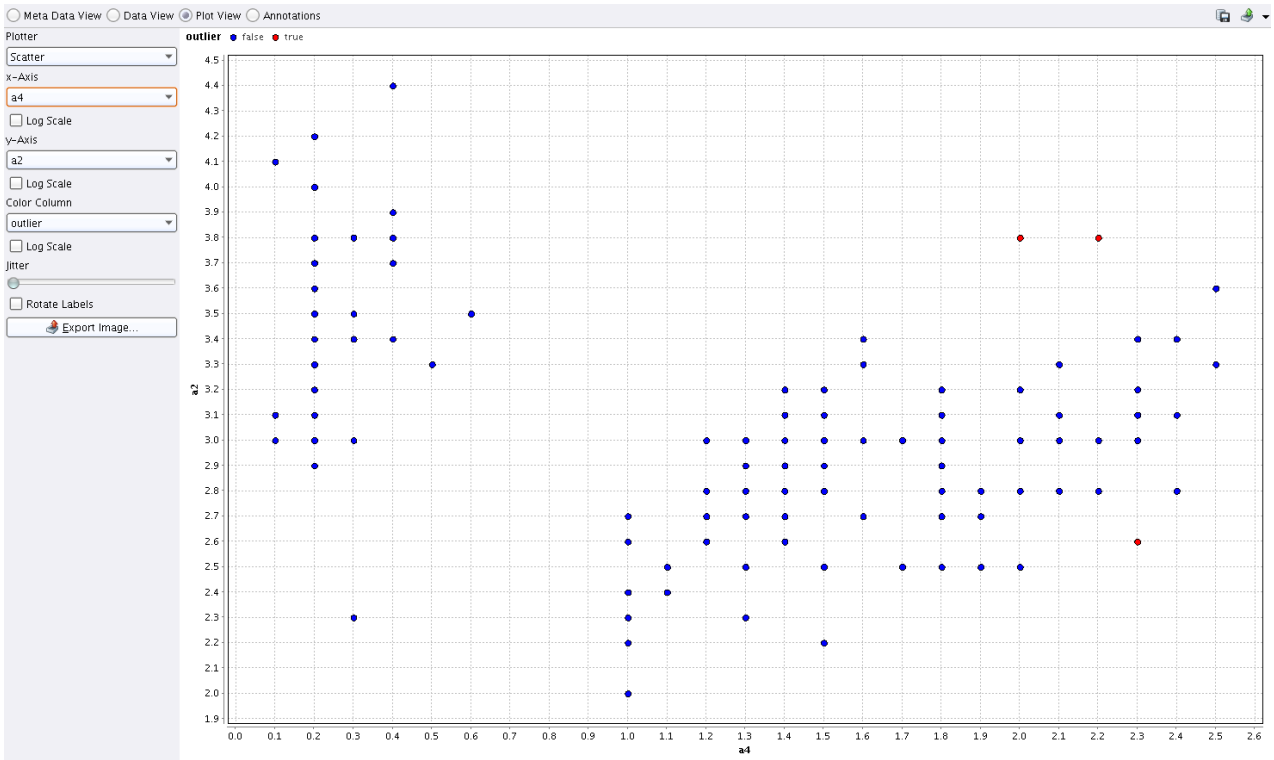
Daten ansehen

Retrieve, Prozess ablaufen, Ergebnisperspektive, Plot, z.B. Scatter Matrix (a1 als Farbe, Korrelation der anderen).



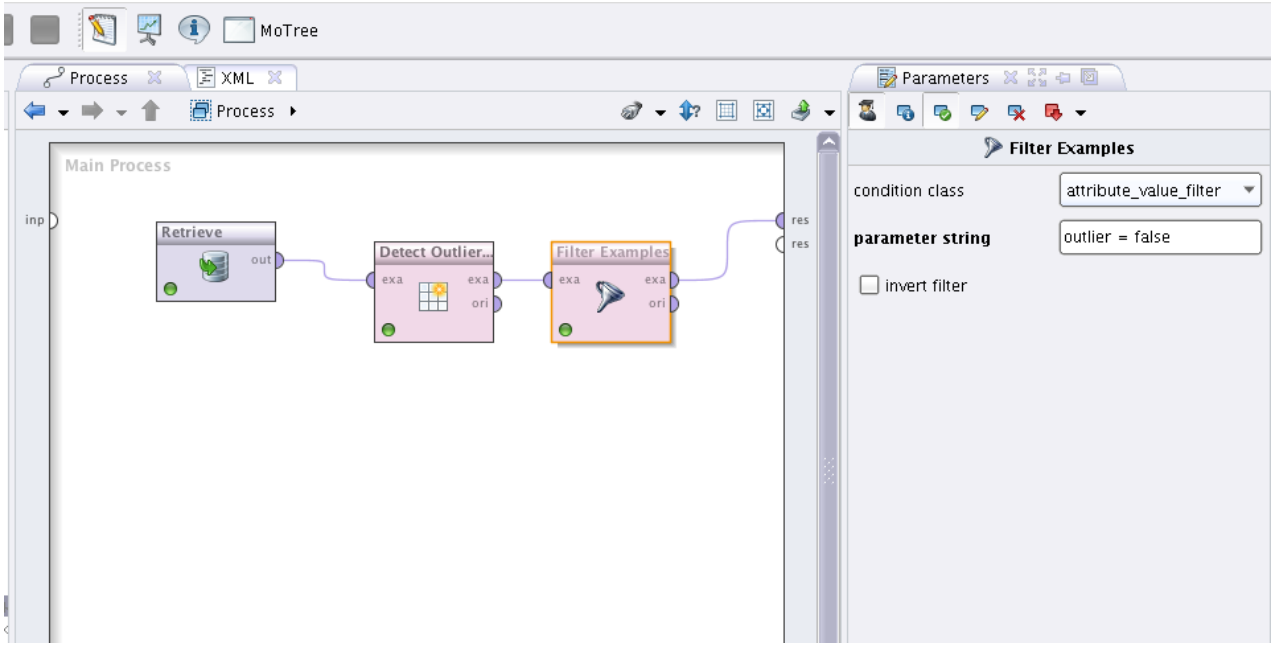
Ausreisser entdecken

RapidMiner hat unter Data Transformation/Data Cleansing Methoden zur Ausreissererkennung, hier Distanz-basiert.



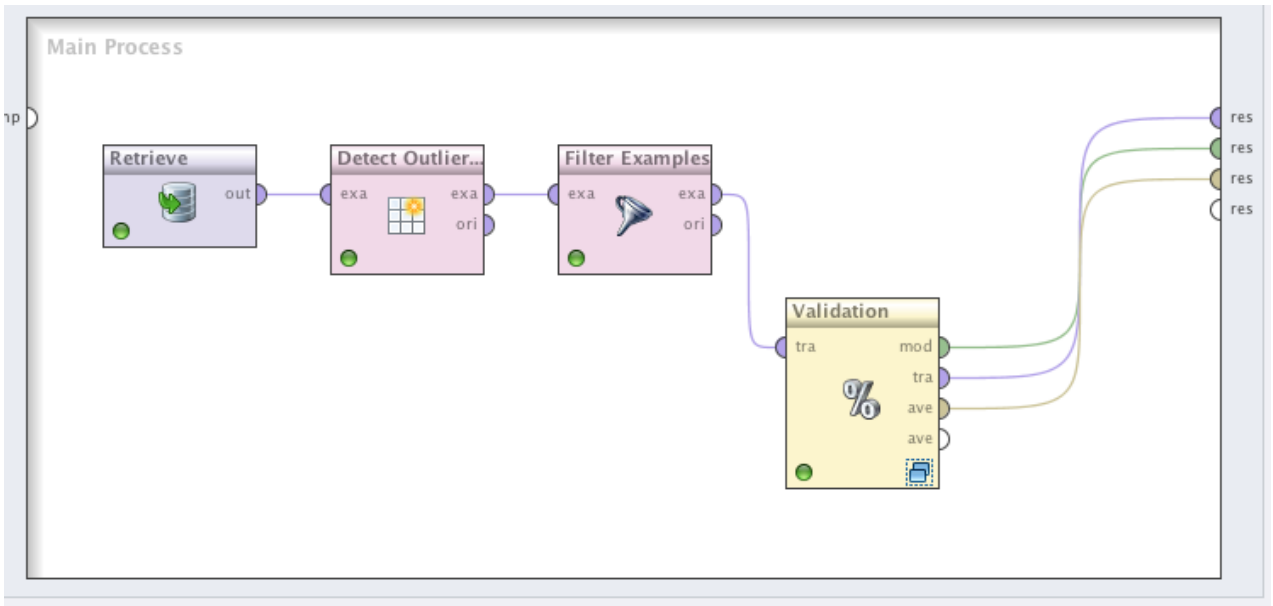
Ausreisser entfernen

Ausreisser sind durch *true* in der neuen Spalte *Outlier* gekennzeichnet.



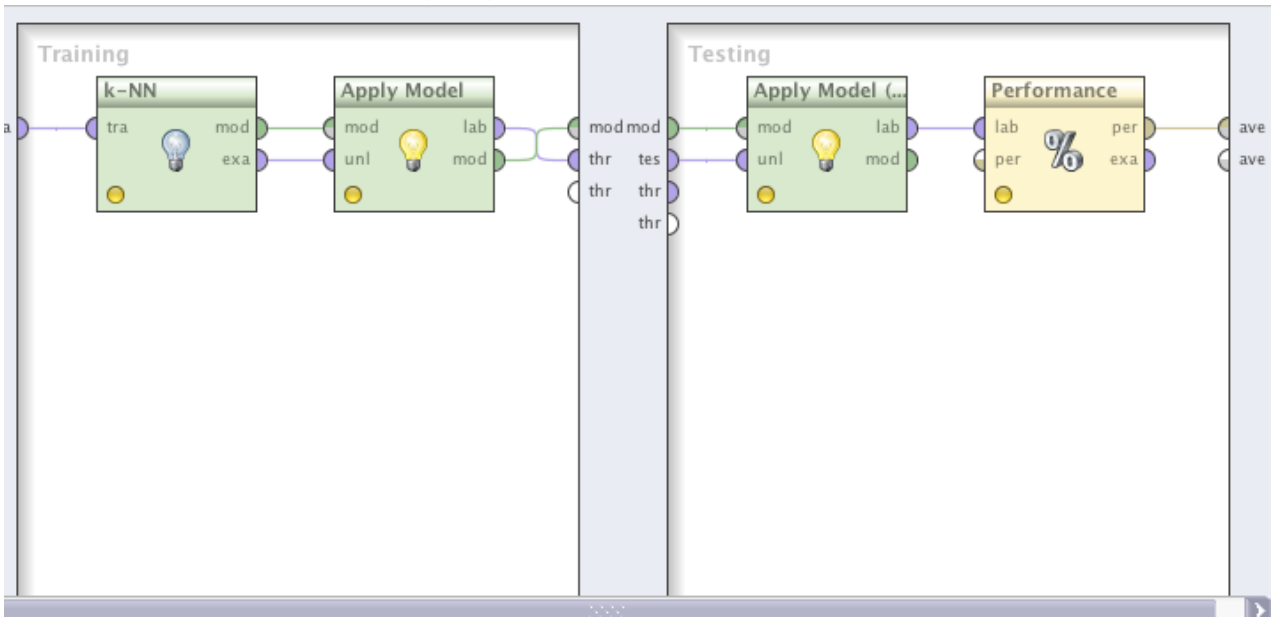
Filtern, so dass nur Beispiele mit *Outlier = false* übrig bleiben.

Kreuzvalidierung



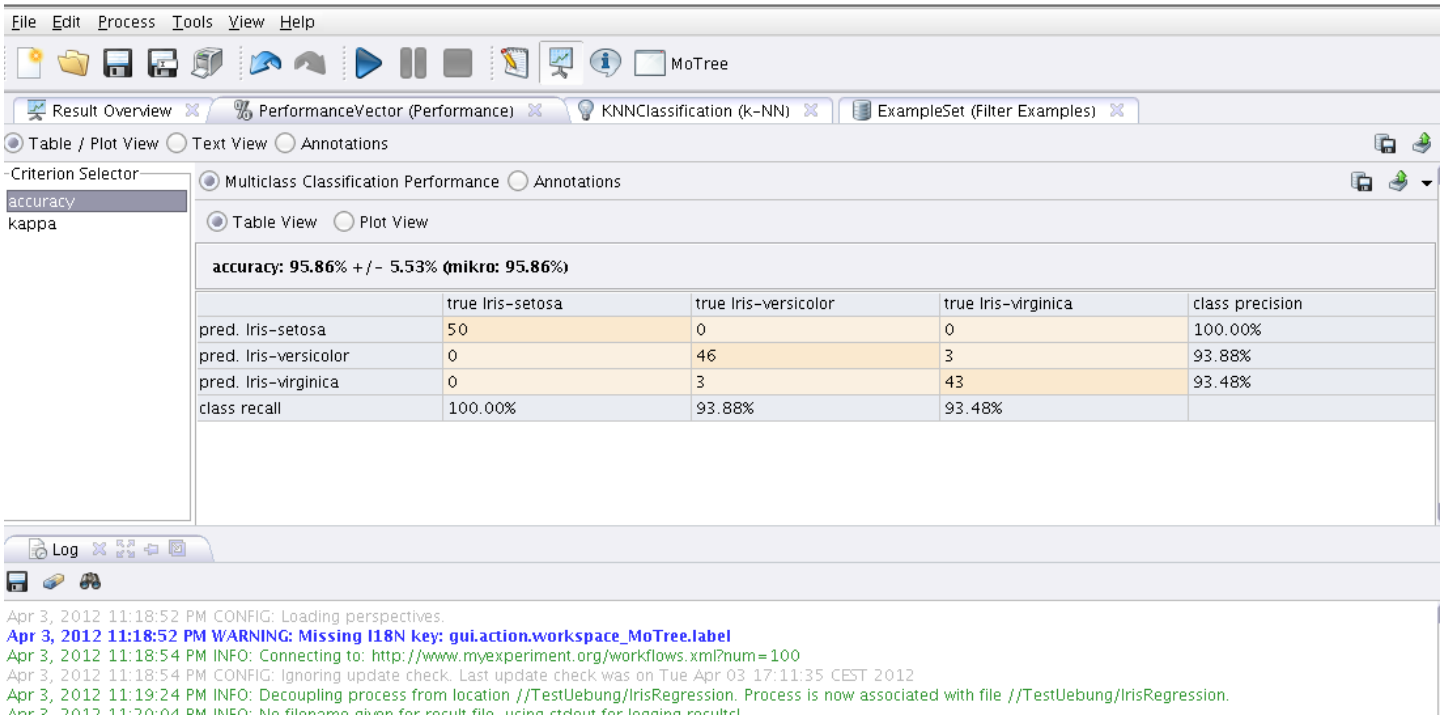
Modellierung

Innerhalb der Kreuzvalidierung wird das Modell gelernt, hier durch k-NN. Dann wird das gelernte Modell getestet und die Performanz als Durchschnitt ausgegeben.



Ergebnis

Resultatsperspektive



Was wissen Sie jetzt?

- Sie haben das CRISP kennengelernt, das den gesamten Ablauf der Wissensentdeckung beschreibt.
- Als Aufgaben der Modellbildung haben Sie Clustering, Klassifikation, Regression gesehen.
- Sie wissen, was die Kreuzvalidierung ist.
- Sie haben RapidMiner kennen gelernt:
  - Repository anlegen
  - Daten einlesen
  - Daten ansehen
  - Daten bereinigen
  - Daten analysieren
  - Modell evaluieren.