

Online-Zertifikatslehrgang

Data Analyst IHK

Die neue Generation digitaler
IHK-Weiterbildungen

#GemeinsamOnlineWeiterbilden

IHK ■ Die Weiterbildung



Modul 3 - DATA ANALYTICS FÜR FORTGESCHRITTENE – DATENBANKEN, MACHINE LEARNING, WORKFLOW CONTROL

➤ Workflowkontrolle

- Flow Variablen
- Benutzerinterfaces
- Schalter
- Schleifen

➤ Datenbanken

➤ Fehlermanagement und Streaming

➤ ML und KI Teil 1

- Übersicht
- Überwachtes Lernen
- Unüberwachtes Lernen

➤ ML und KI Teil 2

- Neuronale Netze
- Modelle bewerten
- Modelle optimieren

Workflow Kontrollstrukturen

Wie lassen sich Daten flexibel und situationsabhängig in Workflows bearbeiten?

Workflow-Variablen erstellen

1. Datenbasiert

Die Workflow-Variablen ergeben sich aus Attributen von Datensätzen, die für die Einstellung der Parameter berechnet oder extrahiert wurden.

Diese Form von Erstellung der Workflow-Variablen ist besonders für vollautomatische Workflows relevant, da hier in der Regel kein manuelles Eingreifen mehr notwendig ist.

Beispiel:

Liste der 5 beliebtesten Pflanzen des letzten Verkaufsmonats (Berechnet und extrahiert aus den Verkaufszahlen).

Workflow-Variablen erstellen

2. Benutzerdefiniert

In bestimmten Fällen ist ein manuelles Eingreifen in Workflows erforderlich oder erwünscht.

Beispielsweise wenn Nutzer von Datenauswertungen Parameter wie Zeitraum, Thema, oder ähnliches auswählen können.

In diesem Fall werden die Workflow-Variablen über eine [Benutzerschnittstelle](#) ermittelt und in den Workflow eingespielt.

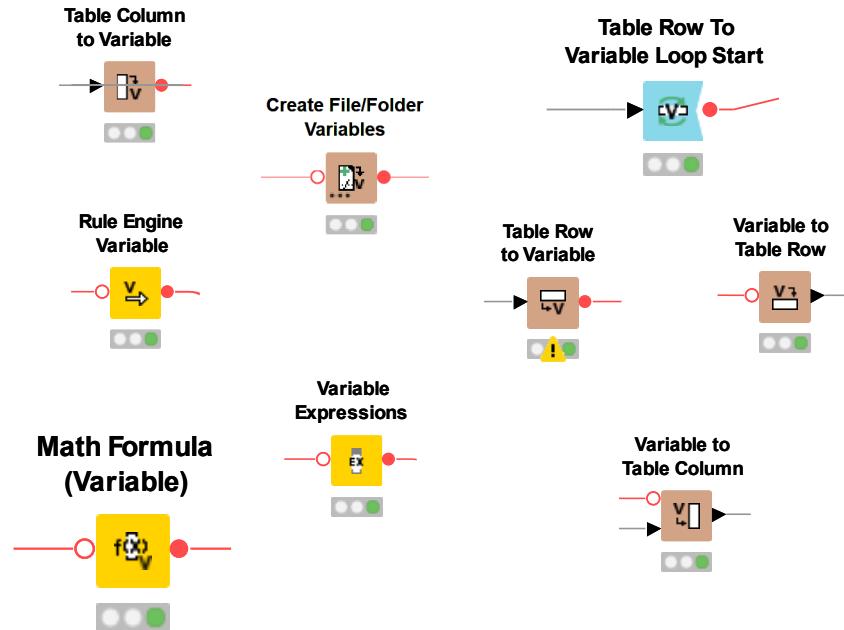
Beispiel:

Für die Verkaufszahlen der Pflanzen möchte der Online-Blumenhändler flexibel den Zeitraum der Auswertung bestimmen. Über eine Datumseingabe verändert er den Parameter für die Filtereinstellung.

Workflow-Variablen: Datentypen

Als **Parameter** für datenabhängige Arbeitsschritte

- Workflow Variablen sind unabhängig von den Daten
- Integer, Double und String
- Lokal oder global
- Lokale Flow Variablen werden stromabwärts automatisch weitergegeben
(Ausnahme: Components)



In KNIME: Workflow Variablen Knoten

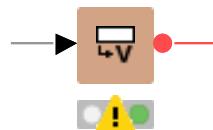
Table Row to Variable

Erstellt Variablen für jede Spalte der Eingangstabelle

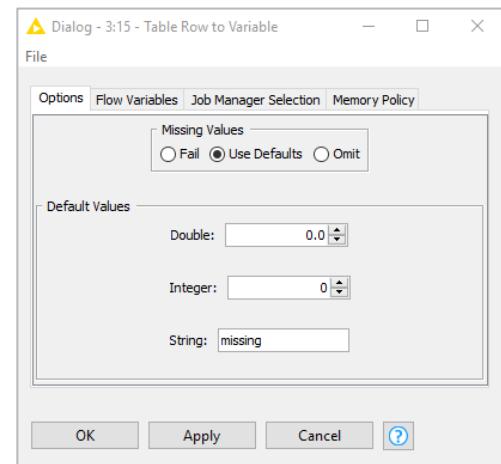
Variablen erhalten die Werte der **ersten** Zeile

Option für datentypspezifische Standardwerte bei fehlenden Werten

**Table Row
to Variable**



Flow Variables			
Index	Owner ID	Name	Value
0	3:15	s BR	421
0	3:15	i Count(FIN)	7806
0	3:15	s RowID	Row10
0		s knime.workspace	D:\Data Science\KNIME



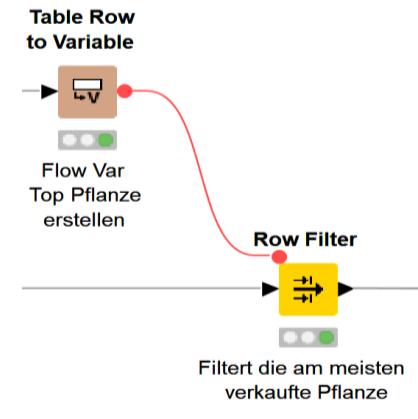
Wie können Workflow Variablen in KNIME eingesetzt werden?

Workflow-Variablen wirken in der Konfiguration

Workflow-Variablen werden wirksam, wenn in der Konfiguration der Operatoren die Einstellungen für Filter, Berechnungen, etc. nicht durch Fixwerte sondern durch die Workflow-Variablen definiert werden.

Die Übergabe der Workflow-Variablen wird in visuellen Analytics-Anwendungen durch einen separaten Verbindungstyp repräsentiert. In KNIME wird dieser durch eine rote Linie und roten punktförmigen Data Ports dargestellt.

Um Daten aus dem Workflow als Parameter nutzen zu können, müssen diese zunächst in Workflow-Variablen umgewandelt werden:

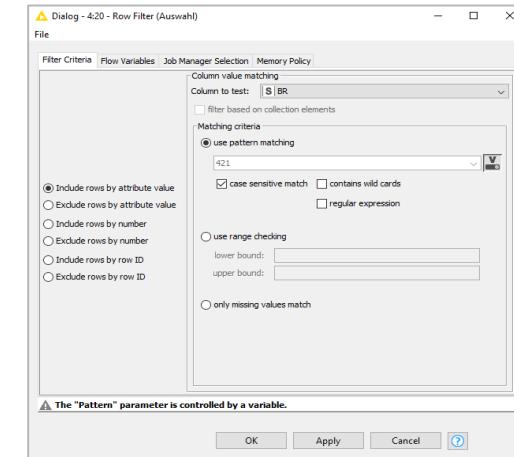
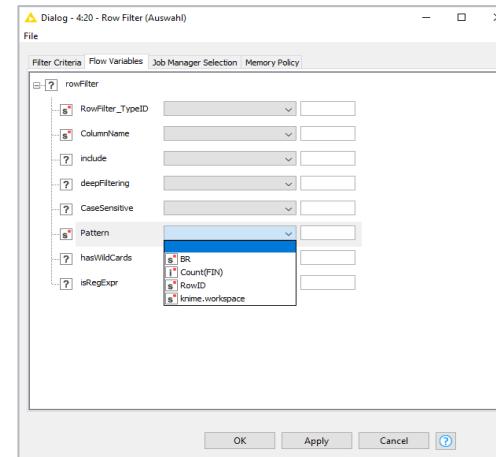
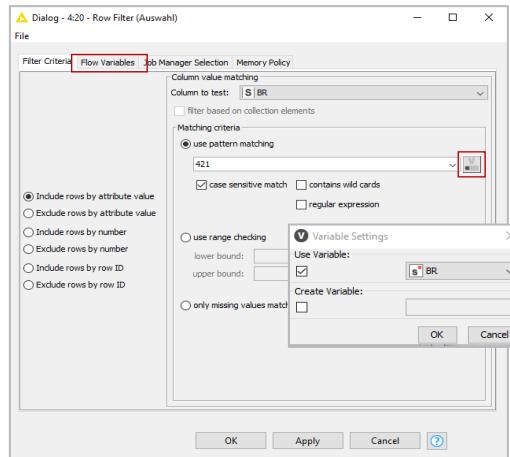


Anwenden von Workflow Variablen

Allgemeine Methode:

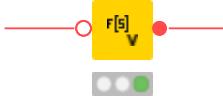
Im Reiter Flow Variables im Konfigurationsmenü: Alle Optionen eines Knotens können durch Workflow Variablen festgelegt werden

→ Einfügen der Variable bei gewünschter Option



Weitere Flow Variable Nodes

**String Manipulation
(Variable)**



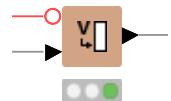
- Wie der Knoten „String Manipulation“, aber für Flow Variablen

**Create File/Folder
Variables**



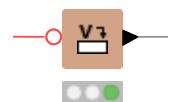
- Erstellt einen Pfad aus 3 Komponenten:
Verzeichnis, Namensstamm, Dateiendung

**Variable to
Table Column**



- Erstellt neue Tabellenspalten mit dem Namen der Variablen
- Jeweiliger Variablenwert wird in alle Zeilen der korrespondierenden Spalten geschrieben
- Auswahl der Variablen möglich

**Variable to
Table Row**



- Erstellt eine neue Tabelle mit einer Zeile und eine Spalte pro Variable
- Auswahl der Variablen möglich

Wie können Interaktionen mit dem Nutzer in den Workflow integriert werden?

Konfigurationseingaben

Bei Benutzerschnittstellen für Konfigurationseingaben geht es darum, dass dem Benutzer die Möglichkeit gegeben wird, Einstellungen für den Ablauf des Workflows anzupassen.

Hier können beispielsweise der Zeitraum für die Analyse gewählt werden, Bestimmte Kategorien oder Attribute, die analysiert werden sollen, oder auch welche Datenquellen verwendet werden sollen.

Die Auswahl der Konfiguration wird mittels Workflow-Variablen in den Workflow übertragen. Damit ist im Prinzip jede Einstellung, die über eine Workflow-Variable getätigt werden kann, auch über die Benutzerschnittstelle wählbar.

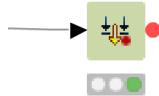
Konfigurationseingaben sind in der Regel stark vordefiniert. So können Datenquellen oder Kategorien aus Listen oder Datumsangaben aus Kalendern ausgewählt werden.

Freitexteingaben sind hier eher selten, da die Möglichkeiten begrenzt und eine Freitexteingabe unnötige Fehlerquellen hinzufügen würde.

In KNIME

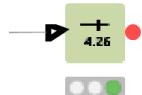
Konfigurationsknoten

Column Selection
Configuration



Ermöglicht die Auswahl einer Spalte für die nachfolgende Bearbeitung

Integer Slider
Configuration



Erzeugt einen Schieber für eine Integer-Spalte zur Auswahl eines Wertes

Date&Time
Configuration



Ermöglicht die Auswahl von Datum und Zeit oder Anfangs- und Enddatum bzw. Zeit
Kann auch als Zeitstempel genutzt werden (use execution time)

Dateneingabe

Neben dem Zugriff auf Datenquellen (Datenbanken und Dateien) können Daten auch direkt vom Benutzer in den Workflow eingegeben werden. Dies ist insbesondere hilfreich, wenn einzelne Datenerfassungen in einem abgesicherten Prozess erhoben und weiterverarbeitet werden sollen.

Beispiel:

Sollen neue Benutzer in einem Workflow erfasst und weiterverarbeitet werden, können die jeweiligen Eingaben in der Schnittstelle so eingerichtet werden, dass Müssfelder berücksichtigt werden, Datentypen überprüft werden (keine Buchstaben in Zahlenfelder), oder bestimmte Zeichenkombinationen, wie sie z.B. bei IDs verwendet werden, beachtet werden. Auch hier kann unterstützend mit Auswahlfeldern die Fehleingabe reduziert werden.

Die Dateneingaben werden üblicher Weise auch zunächst als Workflow-Variablen erfasst und nach erfolgter Prüfung als Datensatz in die Zieltabelle integriert.

In KNIME

Eingabeknoten

Integer Widget



Ermöglicht die Eingabe eines Integers.
Der Wertebereich kann definiert werden.
Ausgabe als Workflow-Variablen

String Widget



Ermöglicht die Eingabe eines oder mehrere Zeichensätze.
Die Gültigkeit kann definiert werden (nach Regex).
Ausgabe als Workflow-Variablen

Refresh Button Widget



Ermöglicht die erneute Ausführung des Workflows, während man sich
in der interaktiven Ansicht befindet. Alle „downriver“ Knoten werden
mit den letzten Einstellungen neu ausgeführt.



Übung Variablen

Wie kann der Ablauf eines Workflows situationsbedingt gesteuert werden?

Switches

(konditionelle Prozesse, Verzweigungen)

Switches geben dem Entwickler von Workflows die Möglichkeit, die Wahl der durchzuführenden Bearbeitungsschritte in Abhängigkeit einer Bedingung zu stellen.

Diese Abfragen gehören zu den wichtigsten Elementen der Programmierung, da hier ein Workflow oder Programm auf unterschiedliche Zustände oder Eingaben reagieren kann.

Welche weiteren Arbeitsschritte durchgeführt werden, hängt von der Art des Switches ab.

Es gibt 2 Typen von Switches:

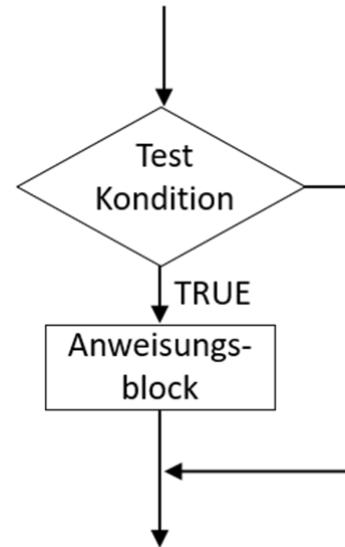
Bedingte Anweisung

Eine **bedingte Anweisung** besteht aus einer Bedingung und einem Workflowabschnitt, der durchgeführt werden soll, wenn die Bedingung erfüllt ist.

Eine solche bedingte Anweisung wird in der Programmierung auch „**If-Verzweigung**“ genannt.

Wird die Bedingung nicht erfüllt, wird der Anweisungsblock umgangen und der Workflow fortgeführt.

If-Verzweigung

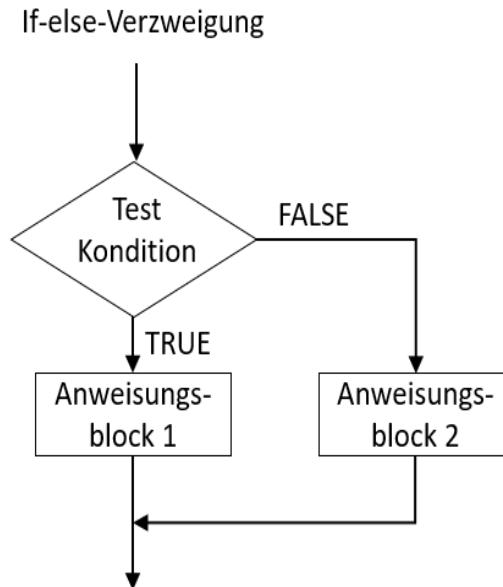


Verzweigungen

Eine **Verzweigung** ist ähnlich wie eine bedingte Anweisung aufgebaut. Wird die Bedingung jedoch nicht erfüllt, wird ein zweiter (alternativer) Anweisungsblock durchgeführt, statt den Workflow einfach fortzuführen.

Eine solche Verzweigung wird in der Programmierung auch „**If-else-Verzweigung**“ genannt. Durch Staffelung der Bedingungsabfragen können auch mehr als zwei Verzweigungen entstehen.

In diesem Fall wäre dann der Anweisungsblock 2 eine weitere „If- Verzweigung“ oder „If-else-Verzweigung“.



In KNIME

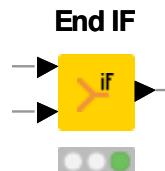
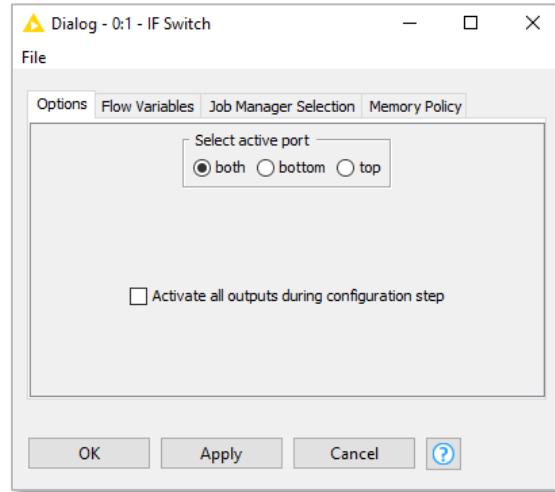
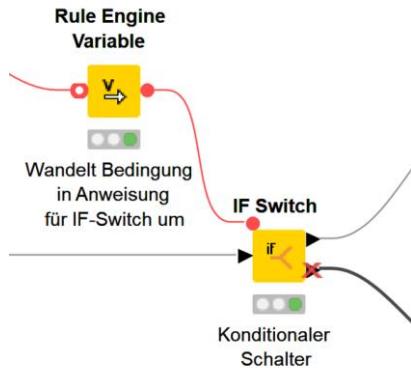
IF Switch & End IF

IF Switch:

Erstellt eine Verzweigung durch eine Abfrage.

Die Information für die Abfrage stammt von einer Workflow-Variable oder aus der Konfiguration

Erlaubte Variablenwerte sind „top“, „bottom“ oder „both“



Mit „End IF“ werden die Verzweigung beendet und die beiden Workflow-Pfade wieder zusammengeführt

In KNIME

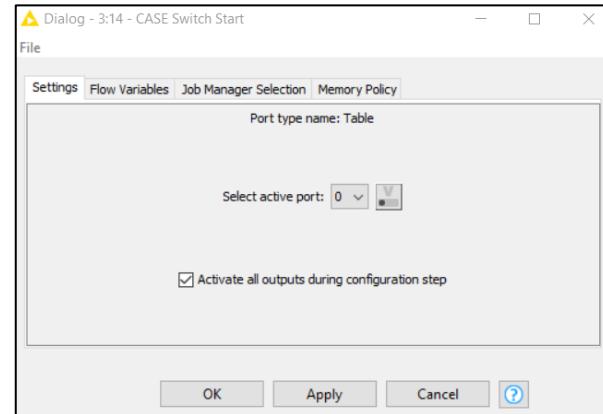
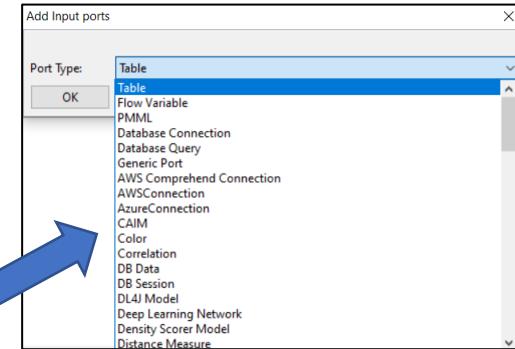
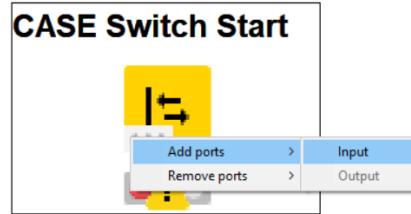
Case Switch Start & ... End

Ähnlich wie IF Switch

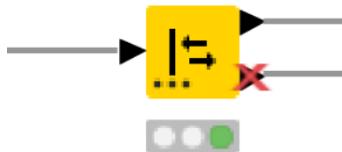
Mit einstellbaren Ports und **nur** einzeln zu schalten

Erlaubte Variablenwerte:

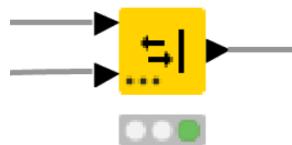
0, 1, 2,... (Format Integer!)



CASE Switch Start



CASE Switch End





Übung

Workflow-Kontroll-Strukturen - Teil 1

Wie lassen sich repetitive Aufgaben in Workflows kompakt bearbeiten?

Schleifen

Beispiel:

Der Blumenhändler hat für das angelaufenen Jahr für jeden Monat einen Bericht erstellt und möchte diese jetzt zu einem einzigen Bericht zusammenfassen. Allerdings will er nicht 12 Workflows (für jeden Monat einen) erstellen, sondern fragt die einzelnen Berichte mir einer Schleife in einem Workflow ab.

Schleifen dienen zum Abarbeiten von **repetitiven Operationen**. Statt jeden Workflow einzeln auszuschreiben, werden Bereiche im Workflow so lange wiederholt, bis das definierte Ende der Schleife erfüllt ist.

Dabei werden je nach Schleifentyp verschiedene Parameter eingesetzt, um die unterschiedlich Eingaben zu verarbeiten (Wie hier im Beispiel, für jeden Monat, der hier als Parameter dient).

Aufbau von Schleifen

Schleifen sind nach einem Grundschema aufgebaut:

1. Schleifenkopf:

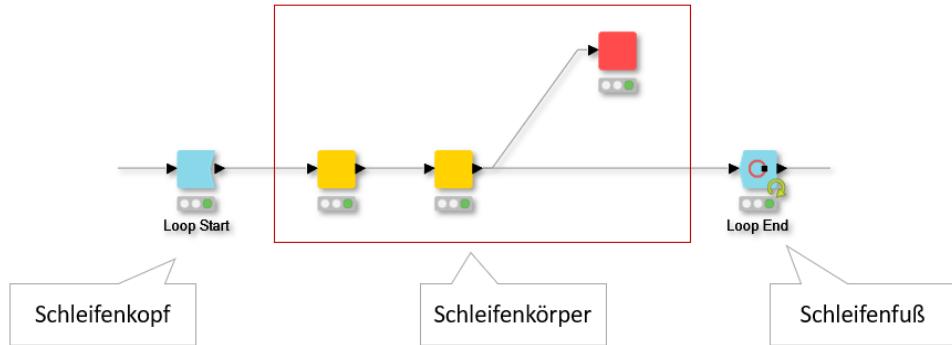
Hier beginnt der Zyklus, Iterationsnummer und weitere Parameter werden als Variablen weitergegeben

2. Schleifenfuß:

Hier endet der Zyklus und der Workflow springt nach vorne. Die Ergebnisse der einzelnen Iterationen werden gesammelt

3. Schleifenkörper:

Workflow-Abschnitt der wiederholt bearbeitet werden soll.



Welche Schleifentypen gibt es und wie werden sie eingesetzt?

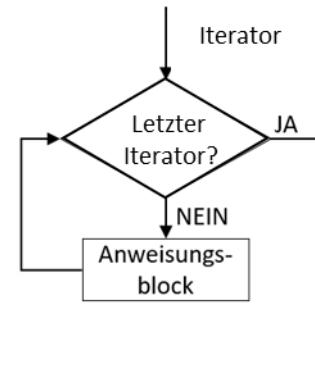
Schleifentypen

1. Zählgesteuerte Schleifen:

Bei einer **zählgesteuerten Schleife** wird im Workflow von einer Anfangszahl bis zu einer Endzahl gezählt und wiederholt dabei jedes Mal einen definierten Operatorenblock („Schleifenkörper“).

Die aktuelle Zahl wird in eine Variable („Iterator“) gesetzt, und mit jedem Schleifendurchlauf („Iteration“) um eins erhöht oder reduziert. Wenn die Endzahl erreicht ist , wird der Workflow fortgesetzt.

Zählgesteuerte Schleife
For/next



Schleifentypen

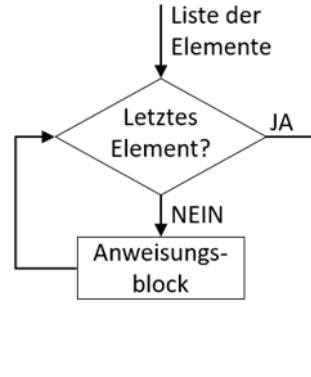
1. Zählgesteuerte Schleifen:

Eine spezielle Form der zählgesteuerten Schleifen sind **Mengenschleifen**.

Eine Mengenschleife führt den Schleifenkörper für jedes Element einer Menge (z. B. eine Produktliste) aus.

Die Reihenfolge der Abarbeitung der Elemente ist beliebig und die Schleife wird beendet, sobald das letzte Element der Liste verarbeitet wurde.

Mengengesteuerte Schleife
For/next



Schleifentypen

2. Kopfgesteuerte Schleifen

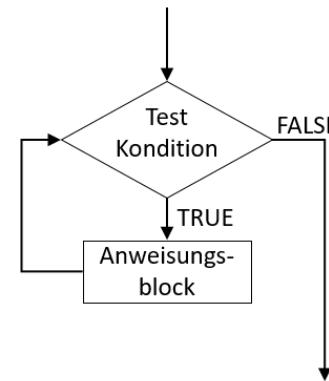
Bei einer kopfgesteuerten Schleife erfolgt eine Abfrage einer Bedingung, bevor der Schleifenkörper ausgeführt wird, also am Kopf der Schleife.

Eine Bedingungsabfrage könnte beispielsweise lauten:
„Anzahl(Bestellung) > 0“.

Solange noch Bestellungen vorliegen und die Bedingung wahr ist, werden die Anweisungen innerhalb der Schleife ausgeführt. Ist die Anzahl der Bestellungen gleich 0 wird der Workflow fortgesetzt.

Wichtig: Wird der Gegenstand der Abfrage in der Verarbeitung des Schleifenkörpers nicht verändert, kann dies zu einer Endlosschleife führen!

Kopfgesteuerte Schleife
While



Schleifentypen

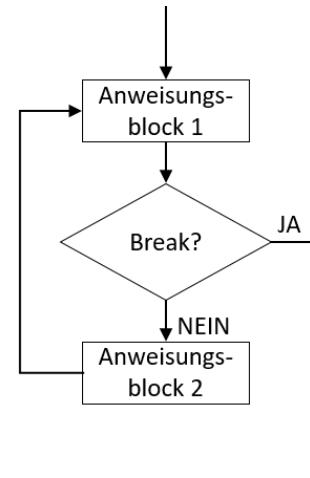
3. Fußgesteuerte Schleifen

Bei einer fußgesteuerten Schleife erfolgt die Abfrage der Bedingung, nachdem der Schleifenkörper ausgeführt wurde, also am Fuß des Konstruktes.

Im Gegensatz zur kopfgesteuerten Schleife führt die Erfüllung der Bedingung zum Abbruch der Schleife und der Workflow wird fortgesetzt.

Auch hier kann bei falscher Bearbeitung der Bedingung eine Endlosschleife entstehen.

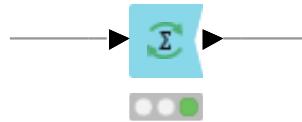
Fußgesteuerte Schleife
Repeat/Break



In KNIME

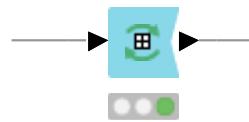
Schleifenköpfe

Counting Loop Start



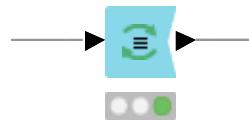
- Schleife wird N Mal wiederholt

Group Loop Start



- Mengenschleife: Die Tabelle wird nach den ausgewählten Attributten nacheinander abgearbeitet
- Erleichtert die Aggregation von Daten über umfangreichere Algorithmen

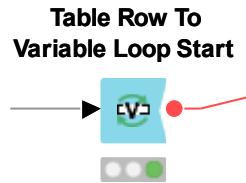
Chunk Loop Start



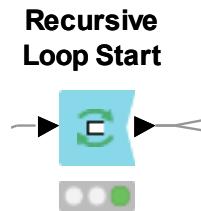
- Feste Anzahl von Zeilen wird in jeder Iteration zur Abarbeitung in den Schleifenkörper geschoben.

In KNIME

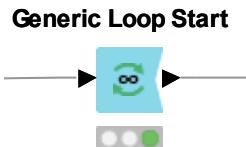
Schleifenköpfe



- Die Schleife arbeitet die Input Tabelle zeilenweise ab und leitet die Werte der Zeile als Variable weiter.



- Das Ergebnis einer Iteration wird in der nächsten Iteration als Input verwendet
- Zusammen mit recursive loop end Knoten

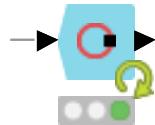


- Läuft bis zum Erreichen eines vordefinierten Wertes einer Loopvariable
- Benötigt den Knoten „Variable Condition Loop End“ um ein Ende festzulegen

In KNIME

Schleifenfüße

Loop End



Standard Schleifenfuß, sammelt Ergebnisse und verbindet sie vertikal
(concatenate)

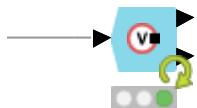
Optional: Spalte mit Iteration im Ergebnis

Variable Loop End



Schleifenfuß für Schleifen, die kein Datenausgang haben (z.B. bei Writer)

Variable Condition Loop End



Schleifenfuß für fußgesteuerte Schleife: Prüft eine Kondition um den Schleife zu beenden (mit Generic Loop Start)



Übung

Workflow-Kontroll-Strukturen – Teil 2

Ergänzungen

Interaktivität im WebPortal

1 Workflow Ausführung gestartet

The screenshot shows the KNIME WebPortal interface. On the left, there's a sidebar titled "Workflow Repository" containing links like "Anlaufmanagement - Use case", "Example Workflows", "Examples", "IParts Prototype", and "KNIME Blogpost". The main area displays a message: "Iparts_Protoype 2020-01-28 13.12.16" followed by "Workflow is currently executing...". Below this is a "Cancel" button. At the top right, there are "Settings" and "Logout" buttons.

2 Consumer wird aufgefordert eine Datei hoch zu laden

This screenshot shows the same KNIME WebPortal interface as above. The "IParts Prototype" item in the sidebar has been expanded, showing its sub-workflows. In the main area, there's a dialog box prompting the user to "Bitte IParts Export csv Datei angeben" (Please specify IParts Export csv file). It includes a "Select File" button and a dropdown showing "<no file selected>". Below the dialog are "Back", "Discard", and "Next >" buttons. The top right corner still has "Settings" and "Logout" buttons.

Interaktivität im WebPortal

3 Filtermöglichkeiten nach Upload, bspw. Bestellnummer oder Monat

The image displays two screenshots of the KNIME WebPortal interface, illustrating filter selection dialog boxes.

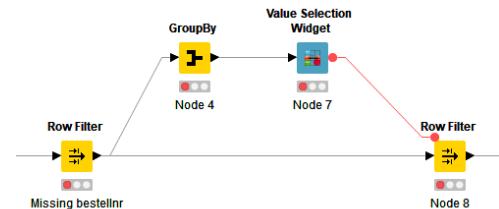
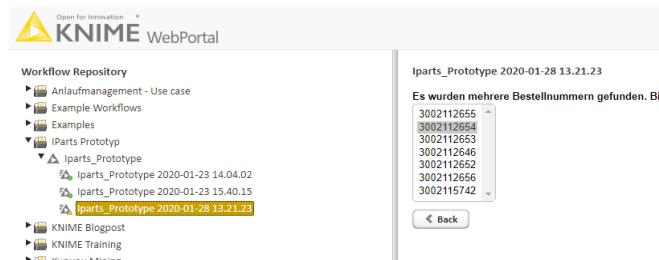
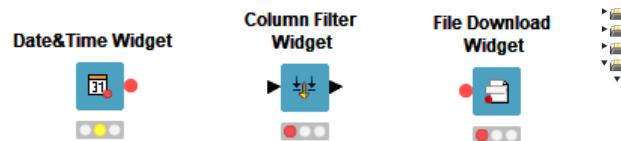
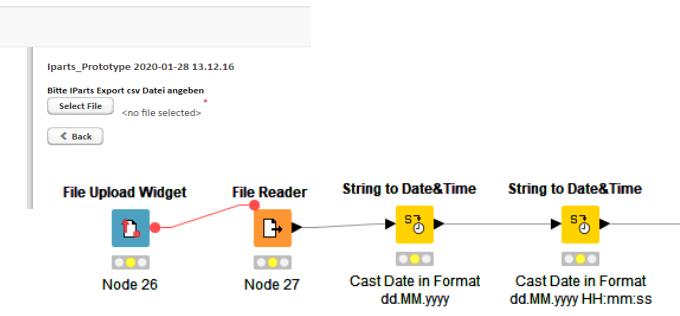
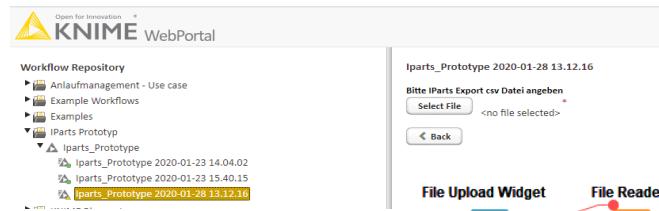
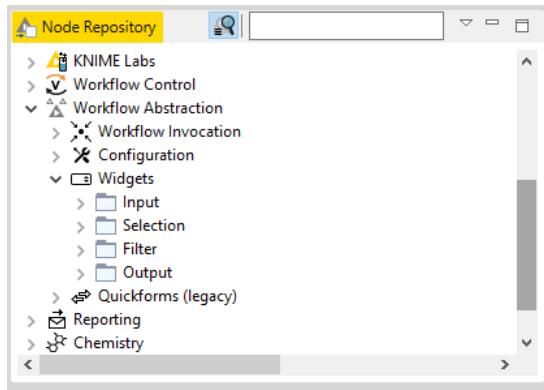
Screenshot 1: Filter by Order Number

The left sidebar shows the "Workflow Repository" with items like "Anlaufmanagement - Use case", "Example Workflows", "Examples", and "Iparts Prototype". The "Iparts Prototype" item is expanded, showing sub-items: "Iparts_Protoype 2020-01-23 14.04.02", "Iparts_Protoype 2020-01-23 15.40.15", and "Iparts_Protoype 2020-01-28 13.21.23". The third item is highlighted with a yellow background. The main content area shows a dialog box titled "Iparts_Protoype 2020-01-28 13.21.23" with the message: "Es wurden mehrere Bestellnummern gefunden. Bitte eine Auswahl:". A dropdown menu lists several order numbers: 3002112655, 3002112654, 3002112653, 3002112646, 3002112652, 3002112656, and 3002115742. Below the dropdown are "Back", "Discard", and "Next >" buttons.

Screenshot 2: Filter by Month

The left sidebar is identical to the first screenshot. The main content area shows a dialog box titled "Iparts_Protoype 2020-01-28 13.21.23" with the message: "Bitte Monat auswählen:". A dropdown menu shows "Oktober" and "November", with "November" currently selected. Below the dropdown are "Back", "Discard", and "Next >" buttons.

Interaktion über Widget Nodes im WF



Arbeiten mit Datenbanken

Nennen Sie häufig verwendete Datenbanktypen

- Worin unterscheiden sie sich?
- Welche sind ihre Vor- und Nachteile?

Datenbanktypen

Es gibt verschiedene Arten von Datenbanken, die ihren Benutzern jeweils unterschiedliche Funktionen bieten. Die gängigen heute verwendeten Typen sind:

1. Relationale Datenbanken

Der häufigste verwendete Datenbanktyp sind relationale Datenbanken. Sie sind gut strukturiert und lassen wenig bis gar keinen Raum für Fehler. Relationale Datenbanken werden in der Regel über die Skript-Sprache SQL gesteuert.

2. NoSQL-Datenbanken

NoSQL sind Datenbanken, die nicht oder nicht nur über SQL gesteuert werden.

Sie sind deutlich freier strukturiert und damit flexibler auf bestimmte Anforderungen (z.B. Big Data) anwendbar.

Kommunikation mit Datenbanken

Der Datenaustausch mit einer Datenbank wird über eine **Datenbankschnittstelle** geregelt.

Sie ermöglicht die Kommunikation zwischen einer Softwareapplikation und der Datenbank. Durch eine definierte Datenbankschnittstelle können Datensätze ausgelesen oder verändert werden, ohne die Verwaltungs- und Speicherungsstruktur der Datenbank zu kennen.

Nicht immer ist ein direkter Zugriff auf Datenbanken möglich oder erwünscht.

Insbesondere bei Anwendungsgebundenen Datenbanken wie beispielsweise bei CRM- oder ERP-System habe häufig eine abschirmte Datenbank ohne Schnittstell nach außen. Im- und Export von Daten ist hier nur über Dateien wie CSV oder Exceltabellen möglich.

Datenbankschnittstellen

Datenbankschnittstellen sind häufig spezifisch an die Skript- und Programmiersprache der Datenbanken angepasst. Es gibt aber auch sprachunabhängige Standards, die von vielen Datenbanken akzeptiert werden:

- [JDBC](#) (Java DataBase Connectivity) ist eine von Sun Microsystems entwickelte Treiberfamilie, die hauptsächlich mit der Programmiersprache Java eingesetzt wird.
- [OpenDBX](#) ist eine in der Programmiersprache C geschriebene und auf Geschwindigkeit und Flexibilität optimierte Datenbankschnittstelle.
- [ADO.NET](#) ist eine von Microsoft entwickelte objektorientierte Zugriffsschicht für die .NET Klassenbibliothek.

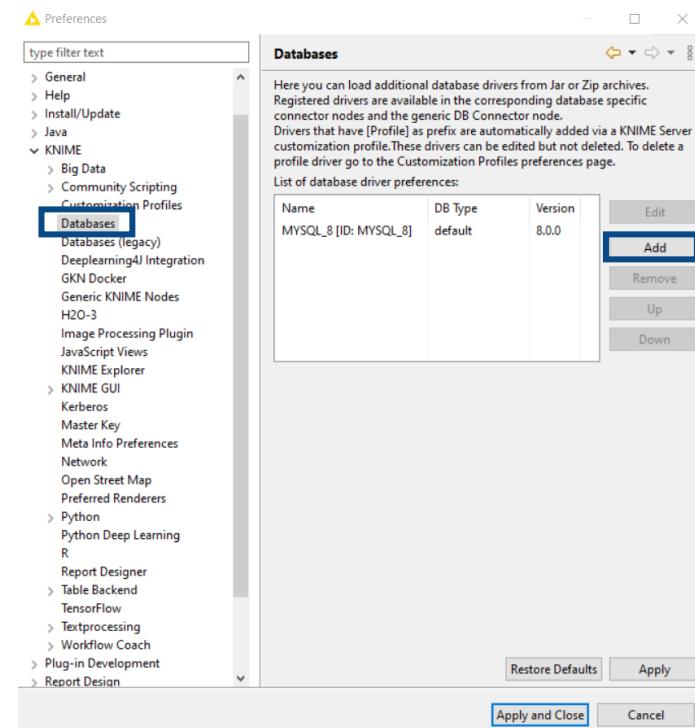
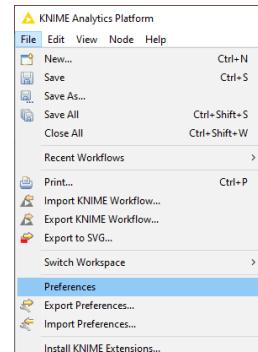
Datenbankschnittstellen in KNIME

KNIME verwendet JDBC als Datenbankschnittstelle.

Gängige Treiber sind bereits installiert.

Sollte einer fehlen, kann man ihn beim Datenbankentwickler herunterladen und in den Preferences installieren:

Unter „Preferences“ kann man bei KNIME > Databases den entsprechenden Treiber einrichten.



Relationale Datenbanken

Relationale Datenbanken

Der häufigste verwendete Datenbanktyp ist die **relationale Datenbank**.

In einer relationalen Datenbank sind den Tabellen **Schlüssel** (auch ID) zugeordnet. Sie helfen bei der Bereitstellung einer schnellen Datenbankzusammenfassung oder beim Zugriff auf bestimmte Zeilen oder Spalten.

Die Tabellen, die auch als Entitäten bezeichnet werden, stehen alle in Beziehung zueinander.

Schlüssel

Kunden - ID	Nachname	Vorname
001	Anders	Achim
002	Brücke	Beate
003	Chor	Christian
004	Denker	Dagmar
005	Eisen	Ernst
006	Funkel	Frauke

Vor- und Nachteile von relationalen Datenbanken

Vorteile:

- Relationale Datenbanken folgen einem strikten Schema. Das bedeutet, dass jeder neue Eintrag unterschiedliche Komponenten haben muss, damit er in die vorgeformte Vorlage passt. Auf diese Weise sind die Daten vorhersehbar und können leicht bewertet werden.
- Sie sind gut strukturiert und lassen wenig bis gar keinen Raum für Fehler.

Nachteile:

- Unhandlich bei großen Datenmengen,
- Die Daten sind schwer oder gar nicht segmentierbar
- Objekteigenschaften und Objektverhalten schlecht abbildbar

1. Wie können Daten effektiv in einer Datenbank bearbeitet werden?
2. Welche Regeln und Prinzipien sind dabei zu beachten?

In-Database-Processing

Viele Datenbankanbieter, die Datenbankprogramme für große Unternehmen herstellen, bieten die Funktion an, Verarbeitungen und Berechnungen innerhalb des Datenbank-Warehouses durchzuführen.

Zu den Vorteilen gehört:

- **Ein großer Geschwindigkeitsschub**

Die Zeit für den Export entfällt und die Datenbank wird deutlich schneller (bis hin zur Echtzeitauswertung)

- **Nahezu keine Ungenauigkeiten.**

Das Verschieben von Daten kann zu Ungenauigkeiten führen, wenn die Datenbankstruktur nicht sehr exakt für diese Aufgaben konfiguriert wurde.

Anwendungsbereiche

Viele große Datenbanken, z. B. Betrugserkennungs- und Börsendatenbanken, verwenden **In-Database-Processing**.

Gerade im **Big Data** Bereich findet dies Technologie Anwendung. Abfragen von großen Datensätzen werden innerhalb der Datenbank transformiert, bevor sie an die Zielanwendung exportiert werden.

Insbesondere **Machine Learning Prozesse** mit großen Datenmengen, wie Bild- oder Spracherkennung profitieren von dem enormen Leistungsvorteil der integrierten Lösung.

Zugriff auf relationale Datenbanken

Auf nahezu alle relationalen Datenbanksysteme, so auch MySQL oder MS-SQL, kann mit Standard SQL zugegriffen werden.

SQL ist eine Standard-Abfragesprache für relationale Datenbanken.

Mit der Abfragesprache werden Tabellen erstellt oder abgefragt z.B.: „Wie viele Kunden heißen Meier?“.

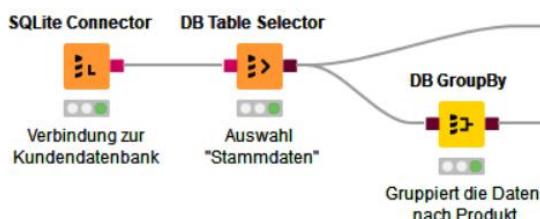
SQL steht für [Structured Query Language](#), übersetzt „strukturierte Abfrage-Sprache“.

In-Database-Processing in KNIME

Die Knoten in KNIME „übersetzen“ die Datenoperationen in SQL

Das SQL-Skript kann überprüft und exportiert werden

Die Datenansicht enthält eine Vorschau, um die ersten Zeilen des Resultats der Bearbeitung zu prüfen.



File

Table Preview | DB Spec - Columns: 2 | DB Query | DB Session | Flow Variables

```
SELECT "Familienstand", COUNT("table_1196414327"."Kunden-ID") AS "COUNT(Kunden-ID)" FROM (SELECT * FROM "Stammdaten") AS "table_1196414327" GROUP BY "Familienstand"
```

File

Table Preview | DB Spec - Columns: 2 | DB Query | DB Session | Flow Variables

Cache no. of rows: 100

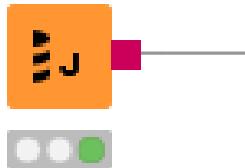
Row ID	Familienstand	COUNT(Kunden-ID)
Row0	ledig	8473
Row1	verheiratet	10011

Einfaches Extrahieren von Daten mit KNIME

Verbindung zur Datenbank herstellen

- Adäquaten Datenbanktreiber auswählen
- URL der DB angeben
- Anmelddaten angeben (User & Passwort)

DB Connector

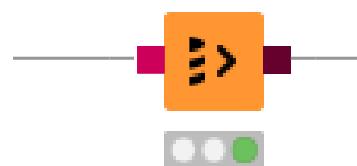


Datenbankverbindung als Ausgangspunkt

Elemente in der Datenbank auswählen

- Tabellen in der Datenbank auswählen über *select a Table*
→ Metadaten abfragen
→ Tabelle auswählen

DB Table Selector

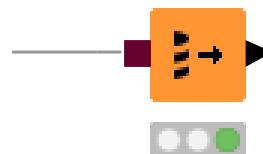


Tabellenselektion

Daten extrahieren

- Tabellen aus der Datenbank nach KNIME exportieren

DB Reader

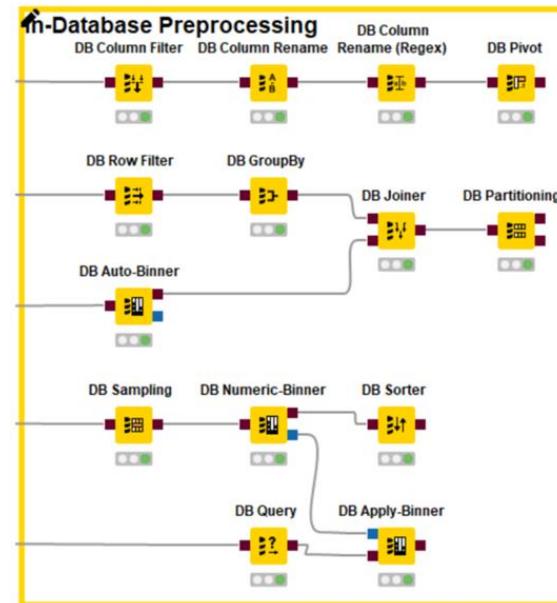


KNIME Table

Query-Knoten in KNIME

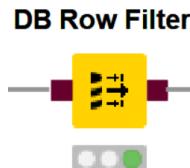
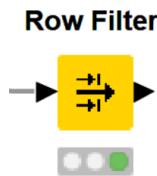
Viele Transformationen funktionieren
analog wie mit den herkömmlichen Knoten:

- Zeilen und Spalten filtern
- Join Tabellen / Abfragen
- Stichproben extrahieren
- Binning von numerischen Spalten
- Sortieren
- Aggregieren
- Partitionieren



Beispiel Row Filter

Die Konfiguration der Knoten unterscheiden sich leicht von den herkömmlichen Knoten, da sie der Logik und der Skriptsprache der Datenbank folgen:



The screenshot shows two windows for configuring a Row Filter. The top window is titled 'Filter Criteria' and shows settings for filtering by column value matching ('S | Familienstand'). It includes options for case sensitivity, wildcards, regular expressions, range checking, and missing values. The bottom window is titled 'Conditions' and shows a query view with a condition being edited: 'S | Familienstand ='. A dropdown menu lists operators: '=', '!=', 'LIKE', 'IS NOT NULL', and 'IS NULL'. A note at the bottom says '- Value is empty'.

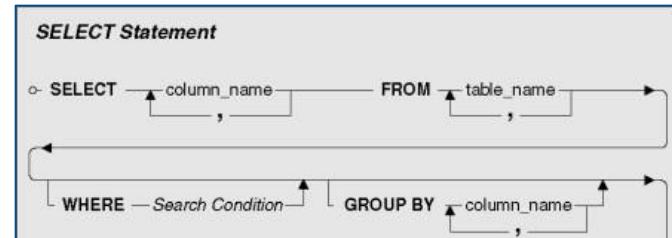
SQL Statements

SELECT: Wählt Spalte(n) aus

FROM: Gibt die Tabelle an, aus der gelesen werden soll

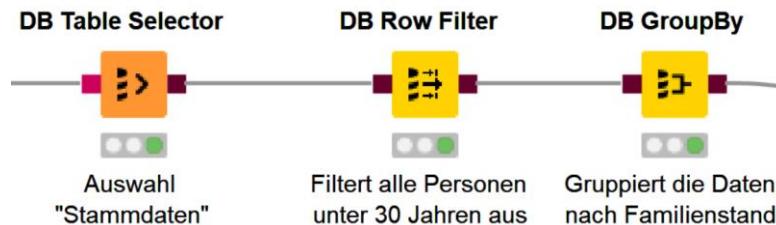
WHERE: Erlaubt die Auswahl von Reihen über Konditionen

GROUP BY: Aggregation



Beispiel:

`SELECT * FROM Stammdaten WHERE Alter > 29 GROUP BY Familienstand`



Weiter wichtige SQL Statements

JOIN: Verknüpft Tabellen reihenweise über ein oder mehrere Felder

Existiert als **INNER JOIN, RIGHT JOIN, LEFT JOIN, FULL JOIN, SELF JOIN**

UPDATE: ändert Zeilen

SET: benennt Spalten und Werte die geschrieben werden sollen

CREATE: erstellt DBs, Tabellen, Abfragen...

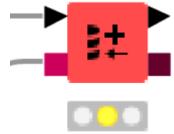
DROP: löscht Tabellen und DBs! **Achtung: was gelöscht ist, ist gelöscht!**



Weitere Informationen und Tutorials zu SQL Datenbanken: z.B.
<https://www.w3schools.com/sql/default.asp>
<http://www.datenbanken-verstehen.de/sql-tutorial/>

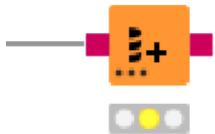
Weitere Knoten

DB Writer



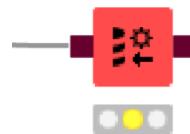
- Schreibt Tabelle in verbundene Datenbank (via Connector)
- Ermöglicht Aktualisieren oder Überschreiben von bestehenden Daten

DB Table Creator



- Erstellt eine neue Tabelle in einer Datenbank
- Spaltennamen und Datentypen können frei gewählt werden

DB Connection Table Writer



- Schreibt oder aktualisiert die Tabelle in der Datenbank.

Das ACID-Prinzip

Das ACID-Prinzip stellt Regeln auf, wie mit Transaktionen in relationalen Datenbanken zu verfahren ist:

Atomicity oder Atomarität:

Eine Transaktion besteht aus einer Sequenz einzelner Aktionen. Diese müssen komplett oder gar nicht ausgeführt werden. Bei Fehlern müssen sämtliche erfolgten Änderungen zurückgenommen werden. (Protokollierung)

Consistency oder Konsistenz:

Eine abgeschlossene Transaktion muss einen konsistenten Zustand hinterlassen. Führt eine Transaktion zur Verletzung von definierten Konsistenzbedingungen, wird sie zurückgewiesen und der Datenbankzustand zurückgesetzt. Die Konsistenz ist vor und nach einer Transaktion sicherzustellen. Während der Transaktion dürfen durchaus inkonsistente Zustände auftreten.

Isolation oder Abgrenzung:

Mit einer Datenbank arbeiten mehrere Benutzer oder Prozesse gleichzeitig. Die Isolation (Abgrenzung) stellt sicher, dass die Nutzung der Datenbank durch mehrere Anwender keine negativen Auswirkungen nach sich zieht. Für jeden Benutzer erscheint das Datenbankmanagementsystem wie ein exklusiv genutztes System. Datenbanksysteme realisieren die Isolation mithilfe von Sperrverfahren.

Durability oder Dauerhaftigkeit:

Ist eine Transaktion ausgeführt und konsistent, sind ihre Informationen dauerhaft in der Datenbank gespeichert. Die Dauerhaftigkeit lässt sich ähnlich wie die Atomarität durch Logging-Maßnahmen realisieren. Mit einem Transaktionslog sind die Informationen nach einem Systemausfall reproduzierbar.

Normalisierung relationaler Daten

Das Ziel der Normalisierung ist die Beseitigung von redundanten Informationen. Durch die Beseitigung der Redundanzen erhöht sich die Konsistenz der Datenbank. Weiteres zentrales Ziel der Normalisierung ist das Entfernen von Anomalien die durch unvollständige Änderungen zustande kommen.

- Schaffung eines Datenbankmodells mit klarer Struktur
- Beseitigung aller vermeidbarer Redundanzen
- Vermeidung jeglicher Anomalien

Die erste Normalform (1NF)

In der ersten Normalform darf jedes Attribut nur einen atomaren (nicht zusammengesetzten) Wertebereich haben. Relationen müssen frei von Wiederholungsgruppen sein.

Beispiel: Das Feld "Name" ist in einer Tabelle in die beiden Datenbankfelder "Vorname" und "Nachname" aufzuspalten.

Die zweite Normalform (2NF)

Der Datensatz darf nur einen einzigen Sachverhalt abbilden. Beinhaltet eine Tabelle mehrere Sachverhalte, wird sie in mehrere Tabellen aufgespalten.

Beispiel: Aufteilung in thematische Tabellen wie "Produktdaten", "Bestelldaten" und "Kundendaten".

Die dritte Normalform (3NF)

Die dritte Normalform gilt als eingehalten, wenn die zweite Normalform erfüllt ist und keine transitiven (indirekten) Abhängigkeiten mehr bestehen.

Beispiel: Aufspalten einer Tabelle mit "Kunde", "Postleitzahl" und "Ort" in zwei Tabellen mit "Kunde" und "Ort" sowie "Ort" und "Postleitzahl".

NoSQL Datenbanken

NoSQL Datenbanken

Der Begriff **NoSQL** im Sinne von **Not only SQL** wurde Anfang 2009 von Johan Oskarsson für ein Treffen über verteilte strukturierte Datenspeicher neu eingeführt. Der Name war ein Versuch einer gemeinsamen Begriffsfindung für die wachsende Zahl an nicht relationalen, verteilten Datenspeichersystemen.

Die gebräuchlichsten Vertreter sind:

- Dokumentenorientierte Datenbanken
- Spaltenorientierte Datenbanken
- Key-Value-Datenbanken
- Objektorientierte Datenbanken

Dokumentenorientierte Datenbanken, spaltenorientierte Datenbanken und Key-Value-Datenbanken sind sich vom Modell sehr ähnlich, daher wird exemplarisch im Folgenden die dokumentenorientierte Datenbank beschrieben.

Dokumentenorientierte Datenbanken

- Elementar für Dokumentendatenbanken ist die Idee, dass zusammenhängende Daten stets gemeinsam an einem Ort (im Dokument) gespeichert werden.
- Daten werden in Dokumenten abgespeichert, beispielsweise in Form von **JSON**, **XML** aber auch in beliebig anderer Form.
- Die Daten werden in sogenannten Key/Value-Paaren gespeichert und bestehen somit aus einem „Schlüssel“ und einem „Wert“.
- Dabei sind diese Dokumente **schemafrei**, das heißt es existieren keine Regeln, nach denen der Inhalt dieser Dokumente aufgebaut werden muss.
- Jedes einzelne Dokument ist also in der Lage, einen völlig anderen Inhalt zu speichern.
- Häufig werden diese Datenbanken im Bereich der Web-Applikationen verwendet und bei unstrukturierten Daten.

Vor- und Nachteile von NoSQL-Datenbanken

Vorteile:

- Zentrale Speicherung von zusammengehörigen Daten in einzelnen Dokumenten,
- freie Struktur,
- multimediale Ausrichtung

Nachteile:

- Relativ hoher Organisationsaufwand,
- oft sind Programmierkenntnisse erforderlich

Zugriff auf NoSQL Datenbanken

Hier am Beispiel einer dokumentorientierten Datenbank:

Dokumentorientierte Datenbanken bieten in der Regel nur Funktionen zu Datenabfrage und -speicherung. Datentransformation und Machine Learning wird in separierten Anwendungen ausgeführt.

Eine Abfrage der Daten erfolgt in der Regel über web-basierte Anwendungen, sogenannte [API \(Application Programming Interface\)](#).

Ein häufig verwendetes Protokoll für die Datenabfrage ist „[REST](#)“ (Representational State Transfer).

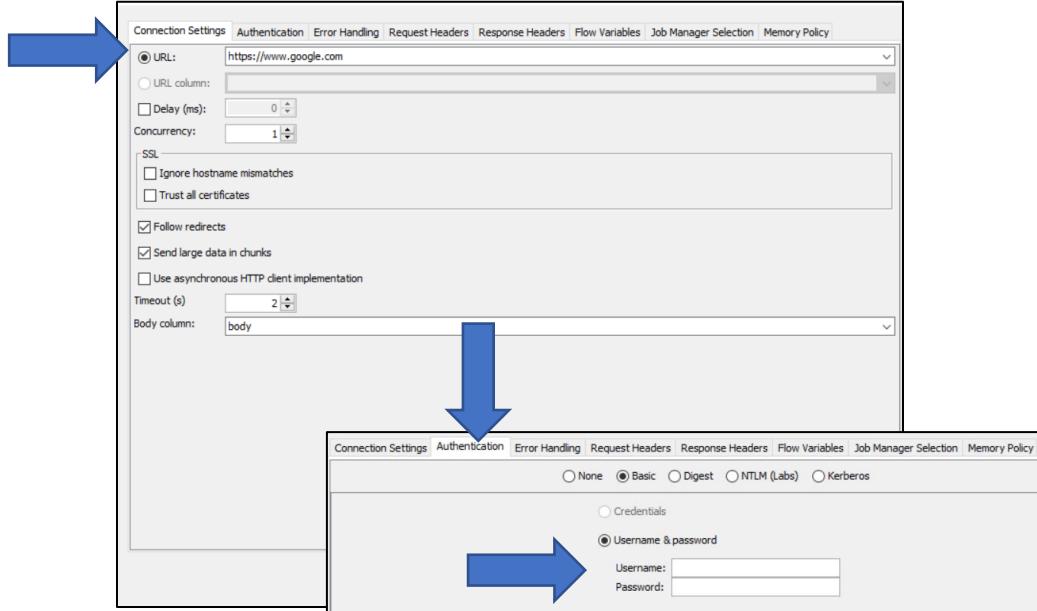
Anwendungen, die diese Protokolle nutzen, heißen auch „RESTful Services“.

RESTful Web Services in KNMIE

GET Request baut eine Verbindung zur No SQL Datenbank auf:

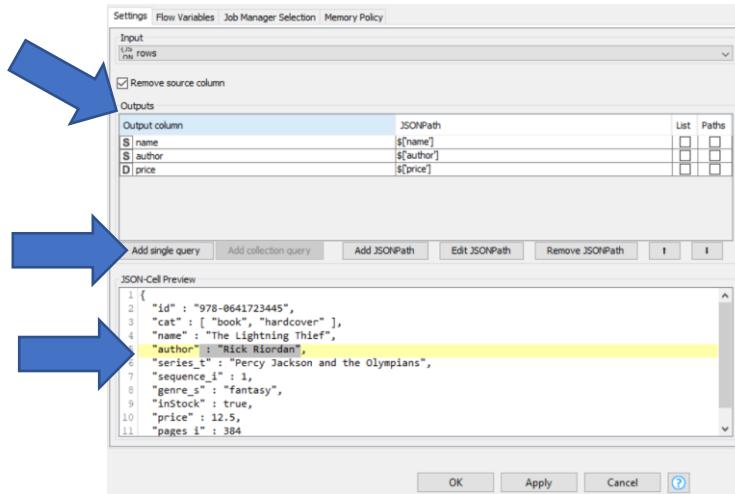
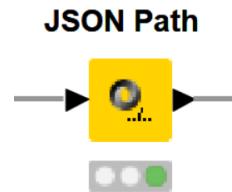
- Den Link setzt man unter URL
- Username und Passwort können im Register „Authentication“ hinterlegt werden

GET Request



Key-Value-Paare extrahieren

Anhand der Schlüssel werden mit **JSON Path** gezielt die relevanten Attribute extrahiert:



The screenshot shows a table with two rows of extracted data:

Row ID	name	author	price
Row0_1	The Lightning Thief	Rick Riordan	12.5
Row0_2	The Sea of Monsters	Rick Riordan	6.49

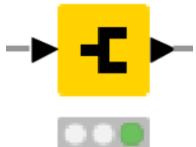
The screenshot shows a table with four rows of extracted data:

Row ID	name	author	price
Row0_1	The Lightning Thief	Rick Riordan	12.5
Row0_2	The Sea of Monsters	Rick Riordan	6.49
Row0_3	Sophie's World : The Greek Philosophers	Jostein Gaarder	3.07
Row0_4	Lucene in Action, Second Edition	Michael McCandless	30.5

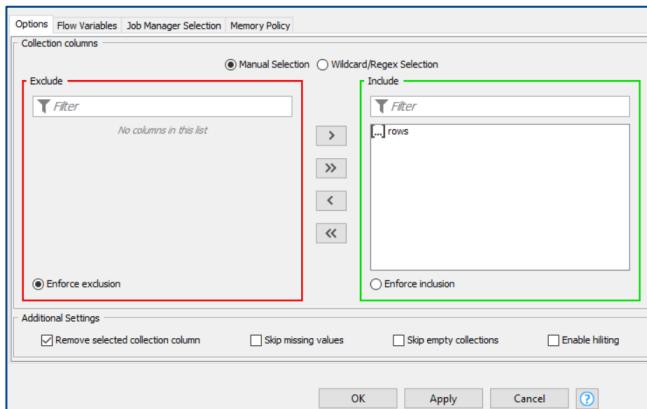
Zeilenstruktur aufbauen

Mit **Ungroup** werden die gruppierten Kategorien auf Zeilen verteilt

Ungroup



ROWS
[{"id": "978-0641723445", "cat": ["book", "hardcover"], "name": "The Lightning Thief", "author": "Rick Riordan", "series_i": "Percy Jackson and the Olympians", "sequence_i": 1, "genre_s": "fantasy"}]



Row ID	IS ON ROWS
Row0_1	{ "id": "978-0641723445", "cat": ["book", "hardcover"], "name": "The Lightning Thief", "author": "Rick Riordan", "series_i": "Percy Jackson and the Olympians", "sequence_i": 1, "genre_s": "fantasy", "instock": true, "price": 12.9, "pages_i": 384 }
Row0_2	{ "id": "978-1423103345", "cat": ["book", "paperback"], "name": "The Sea of Monsters", "author": "Rick Riordan", "series_i": "Percy Jackson and the Olympians", "sequence_i": 2, "genre_s": "fantasy", "instock": true, "price": 6.49, "pages_i": 304 }



Übung Datenbanken

Ergänzungen

Automatisierung

Datenbankadministratoren und Data Analysts sind häufig mit zeitaufwändigen manuellen Aufgaben wie der Verwaltung und Pflege von Daten und Datenprozessen beschäftigt.

Manuelle Eingriffe in Datenprozesse sind eine der Hauptquellen für Fehler und mangelnde Datenqualität. Tritt dies gehäuft auf, so kann es sich erheblich negativ auf die Verfügbarkeit, Performance und die Sicherheit der Daten auswirken.

Die Lösung für dieses Problem ist:

- So viel Automatisierung wie möglich
- So wenig manuelle Schritte wie nötig

Dies erreicht man beispielsweise durch Zeit- und Eventsteuerung:

Zeit- und Eventsteuerung

Alle Prozesse (sowohl für das Datawarehousing als auch für die Transformations- und Analyseprozesse) müssen als automatisierbare Module vorliegen.

Sie werden dann entweder durch eine **Zeitsteuerung** (beispielsweise jeden Monatsanfang, alle 24h, etc.) ausgelöst und durchgeführt.

Eine andere Möglichkeit ist die **Eventsteuerung**. Hier werden die Prozesse durch definierte Ereignisse initiiert, z.B. ein Datenupdate ist verfügbar, ein Nutzer fragt eine standardisierte Datenauswertung ab, etc.

Beim Erstellen von Workflows muss darauf geachtet werden, dass regelmäßig an wichtigen Stellen **automatisierte Kontrollen sowie Backups** durchgeführt werden. Kommt es zu einem Fehler, kann so das System auf die letzte validierte Version zurückgestellt werden (nach Möglichkeit geschieht das auch automatisiert). Workflow-Protokolle helfen dem Data Analyst, die Fehler schnell zu identifizieren und zu beseitigen.

Big Data Datenbanken

Big Data Datenbaken

Big-Data-Datenbanken sollen die unterschiedlichsten Datentypen schnell und effizient verarbeiten.

Abhängig von den Anforderungen, die an die Datenverarbeitung gestellt werden, werden verschiedene Datenbanktypen verwendet:

Relationale Datenbanken:

Sie unterstützen die traditionelle Abfragesprache SQL, Streaming-Daten und Machine Learning. Besonders Apache Spark erfreut sich durch seine spezifische Architektur, hohe Performance und vielseitige Programmierbarkeit starker Verbreitung.

NoSQL:

Mit NoSQL-Systemen können große Datenmengen mit hoher Performance gespeichert und abgefragt werden. Sie sind daher überall dort geeignet, wo SQL-Datenbanken an ihre Grenzen stoßen. Besonders bei komplexen und flexiblen Abfragen von unstrukturierten Daten spielen die NoSQL-Lösungen ihre Vorteile voll aus. Die Daten müssen nicht in ein SQL-Korsett gepresst werden, sondern sind flexibel prozessierbar.

Beispiele für Big Data Datenbanken

Relationale Datenbanken

- Hadoop und Spark
- SQL basierte Datenbanken, die für paralleles Dataprocessing optimiert sind. Enthalten Machine Learning Funktionen

Key Value Stores

- Amazon Dynamo, Voldemort, Membase, Redis ...
- nur einfache key-basierte Lookup/Änderungs-Zugriffe (get, put)

Spaltenorientiert (erweiterte Record-Stores / Wide Column Store)

- Google BigData / Hbase, Hypertable, Cassandra ...
- Tabellen-basierte Speicherung mit flexibler Erweiterung um neue Attribute

Dokument-Datenbanken

- CouchDB, MongoDB ...
- Speicherung semistrukturierter Daten als Dokument (z.B. JSON)

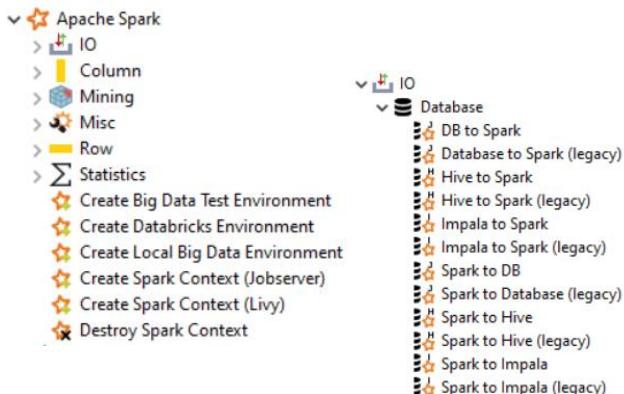
Graph-Datenbanken

- Neo4J, OrientDB ...
- Speicherung / Auswertung großer Graph-Strukturen

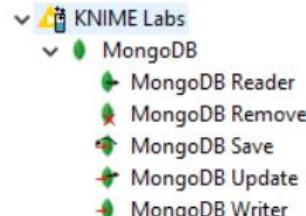
Big Data in KNIME

KNIME besitzt eine sehr umfangreich **Spark**-Integration

Mit ihr können nahezu alle Funktionen in einer Hadoop-Datenbank mit Spark ausgeführt werden.



Für die **Mongo Datenbank** gibt es Schnittstellen zum Datenabruf und Speichern, jedoch kein In-Database-Processing



Fehlermanagement

Welche Hilfsmittel stehen zur Überwachung
und zum Fehlermanagement zur Verfügung?

Checkpoints

So wie Eingangsdaten auf Korrektheit geprüft werden können, können auch die Daten nach den jeweiligen Verarbeitungsschritten auf ihre Integrität und Korrektheit untersucht werden.

Dafür fügt man in den Workflow Checkpoints ein, die anhand definierter Kriterien die Daten aber auch Workflow-Variablen überprüfen:

1. Prüfsummen:

Manche Berechnungen (insbesondere bei Finanzberichten) lassen sich gegenseitig aufsummieren und sollten dann ein bestimmtes Ergebnis liefern (bei Kontobuchungen z.B. = 0). Diese Summen können zur Kontrolle gebildet werden und mit einem Eintrag im Workflow-Protokoll dokumentiert werden.

Checkpoints

2. Grenzwerte

Regelmäßig bearbeitete Daten bewegen sich normaler Weise in einem bestimmten Wertebereich. Wird dieser Bereich verlassen, kann dies ein Hinweis für einen Fehler in den Daten oder der Datenverarbeitung sein. Auch hier können die Daten kontinuierlich überprüft werden, und bei Abweichung einen Eintrag im Workflow-Protokoll tätigen.

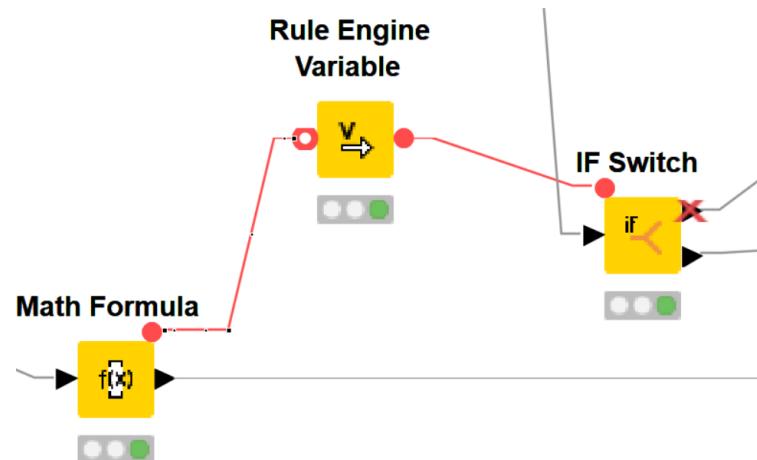
3. Zustandswerte

Wie die Daten selbst, können auch die Vorgänge des Workflows Fehler enthalten. Dies macht sich insbesondere an den Workflow-Variablen bemerkbar. Werden hier falsche oder gar keine Werte übergeben, hat das einen gravierenden Einfluss auf die Bearbeitung der Daten. Auch hier können die Variablen kontinuierlich überprüft werden, und bei Abweichung einen Eintrag im Workflow-Protokoll tätigen.

In KNIME Prüfstrukturen

Einfaches Prüfungskonstrukt:

Der Knoten Math-Formula überprüft, ob die Prüfsumme = 0 ist. Weicht der Wert davon ab, aktiviert der Knoten Rule Engine Variable beim Knoten IF Switch den oberen Workflow-Pfad frei, durch den dann ein Eintrag im Workflow-Protokoll ausgeführt wird (nicht dargestellt)



Breakpoints

Sind die Fehler jedoch so gravierend, dass eine weitere Verarbeitung keinen Sinn macht bzw. sogar zu Schaden führen könnte (überlastete oder blockierte Server), sollte der Workflow umgehend abgebrochen werden.

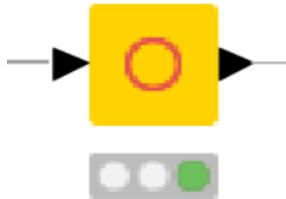
Auch hierfür gibt es automatische Überwachungselemente, sogenannte Breakpoints. Diese funktionieren ähnlich wie die Checkpoints und überprüfen Daten und Zustände. Wird eine kritisches Prüfkriterium ermittelt, verursacht der Breakpoint den Abbruch des Workflows.

Breakpoints sind ein wichtiges Hilfsmittel, um Schleifen unter Kontrolle zu halten. Gerade Kopf- oder Fußgesteuerte Schleifen können bei falschen Iterations-Variablen zu Endlosschleifen werden. Hier kann ein Breakpoint gesetzt werden, der beispielsweise den Workflow bei Erreichen des 100. Schleifendurchgangs abbricht.

In KNIME Breakpoints

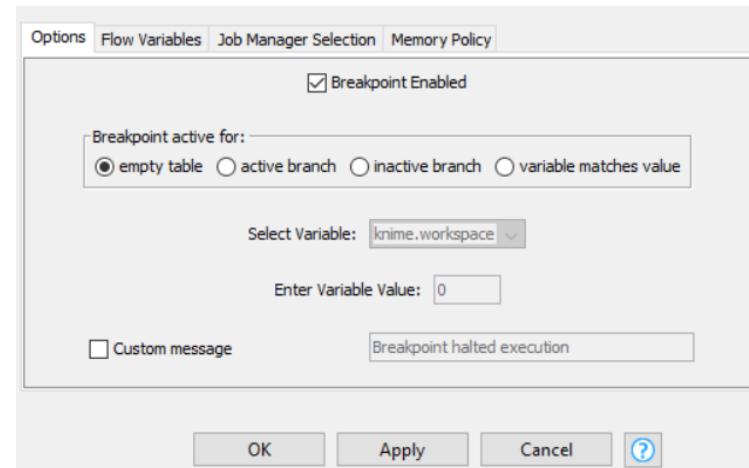
Der Knoten Breakpoint lässt den Workflow so lange laufen, bis ein Abbruchkriterium erfüllt wird.

Breakpoint



Breakpoint Konfiguration:

Ein Abbruch kann erfolgen, wenn beispielsweise keine Daten im Workflow enthalten sind oder eine Variable einen definierten Wert annimmt. Unter „custom message“ kann eine vorbereitete Fehlermeldung als Workflow-Variable ausgegeben werden.



Monitoring

Sind alle Fehler im Workflow und in den Daten gefunden, der Workflow selbst läuft stabil und die Auswertungen und Analysen laufen stabil und liefern die erwarteten Ergebnisse gilt es nun, den Datenprozess stabil zu halten und dabei zu überwachen.

Dabei geht es einerseits darum, die ausgeführten Datenverarbeitungen zu dokumentieren, um eventuelle Fehler nachzuverfolgen zu können.

Andererseits soll auch die Leistungsfähigkeit der Analyse und Ergebnisproduktion überwacht werden, um beispielsweise eine eventuelle Verschlechterung des Maschine Learning Modells zu entdecken und zu beheben.

Das Workflow-Protokoll (Log)

In der Regel protokolliert jedes Analytics Tool und jedes Datenbank Management System alle Vorgänge, die auf ihnen durchgeführt werden, sehr detailliert.

Allerdings sind diese Protokolle aus der Sicht der Programmierer der Anwendungen geschrieben und daher nicht unbedingt eine geeignete Dokumentationsform für den Ablauf der Workflows.

Es empfiehlt sich also eine eigene Protokollstruktur aufzubauen und dort alle relevanten Aktionen, die im Prozess des Workflows durchgeführt werden zu dokumentieren.

Inhalt eines Workflow-Protokolls

Folgende Informationen sollten in einem Workflow-Protokoll zu finden sein:

- Zeitstempel: bsp. 2021-05-14 13:36:44
- Nutzer oder Trigger: bsp. Nachtdurchlauf
- Workflow-Bezeichnung: bsp. Datenaktualisierung
- Datenquellen, die abgerufen oder geändert wurden: bsp Kundendatenbank, Kundenstammdaten
- Welche Aktion durchgeführt wurde: bsp. Aktualisierung der Kundenstammdaten
- Fehlermeldungen: bsp. Verbindung mit Datenbank konnte nicht aufgebaut werden.
- Leistungsdaten: bsp Modellgenauigkeit, Berechnungszeit, ...
- Zusätzliche Informationen, wie Adressen, Pfade, Backups etc.

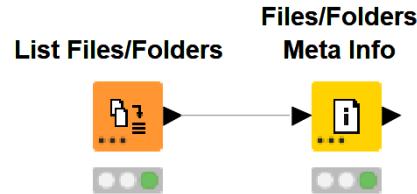
Einbau von Kontroll- und Monitoring-Strukturen im Workflow

Automatische Prüfmechanismen helfen die Integrität der Workflows zu überwachen!

Beispiel:

a) Überwachen der Metadaten (Pfad und Zeitstempel).

- Wurden die richtigen Daten verwendet?
- Entsprechen sie der beabsichtigten Auswertungsperiode?



Knoten „Files/Folder Meta Info“ in Verbindung mit „List Files/Folders“.

Es werden zu jeder Datei in dem untersuchten Ordner die Metadaten ausgelesen und in eine Tabelle geschrieben. Diese kann dann mit den Sollwerten verglichen werden.

Output Table - 0:15:12 - Files/Folder Meta Info								
Table "default" - Rows: 4 Spec - Columns: 7 Properties Flow Variables								
Row ID	P Path	B Directory	L Size	S Size (h...)	Last modified date	Creation date	B Exists	
Row0	.../01_Daten/Input/titanic_1class_data.xlsx	false	36203	35 KB	2020-03-09T14:33:28...	2021-02-22T11:40:01.385...	true	
Row1	.../01_Daten/Input/titanic_1class_survived.csv	false	4440	4 kB	2020-03-09T14:33:28...	2021-02-22T11:40:01.389...	true	
Row2	.../01_Daten/Input/titanic_rest_data.xlsx	false	72767	71 KB	2020-03-09T14:33:28...	2021-02-22T11:40:01.391...	true	
Row3	.../01_Daten/Input/titanic_rest_survived.csv	false	7102	5 kB	2020-03-09T14:33:28...	2021-02-22T11:40:01.393...	true	

Einbau von Kontroll- und Monitoring-Strukturen im Workflow

b) Arbeiten mit Grenzwerten

Häufig bewegen sich Daten in bestimmten Bereichen. Überwacht man diese Bereiche und kommt es zu Abweichungen, so kann das ein Hinweis für eine Veränderung der Datenqualität sein.

Zur Überwachung eignen sich:

- Minum, Maximum und Durchschnitt
- Aber auch Anzahl und Häufigkeit von Fehlenden Werten oder Fehleinträgen
- „Checksums“ – Abgleichungen von Rechenwerten

Fast man diese Überprüfungen zu kurzen Berichten zusammen, ist ein durchgängiger Nachweis der Datenintegrität sehr einfach.

Analyse der Analyse

Auch ein Workflow selbst produziert interessante Daten, die erfasst und analysiert werden können. Die einfachste Form diese zu nutzen, ist ein regelmäßiger Bericht, der die wichtigen Leistungsmerkmale und Fehler des Workflows enthält.

Darüber hinaus können auch tiefere Analysen über Leistungszusammenhänge oder aber auch Vorhersagen getroffen werden.

Inwieweit sich die Mühe dafür lohnt, hängt von der Komplexität und den Optimierungsanforderungen des Workflows ab. Eine einfache Datentransformation wird dies vielleicht noch nicht rechtfertigen. Kann man aber dadurch bei Datendurchläufen von mehreren Stunden signifikant Zeit einsparen, hat sich der Aufwand gelohnt.



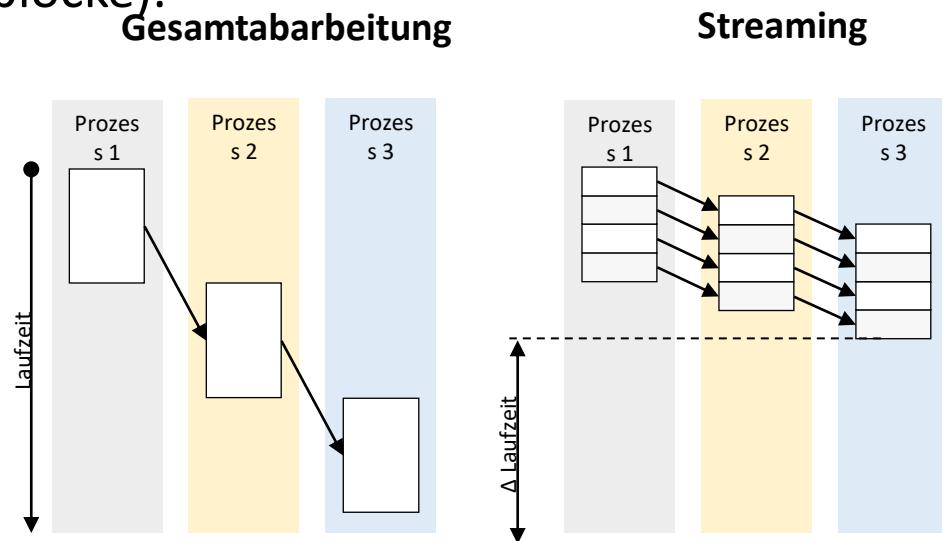
Übung Fehlermanagement

Laufzeitoptimierung

Streaming

Streaming erlaubt gleichzeitige Bearbeitung oder Nutzung eines Datensatzes in Chunks (Datenblöcke).

- Vorteil: Effizienzsteigerung durch verkürzte Laufzeiten der Datenbearbeitung
- Nachteil: Unabhängigkeit der Chunks muss gewährleistet sein...
- Beispiel: Video Streaming aus dem Netz



Streaming-fähige Knoten

Streaming läuft zeilenweise

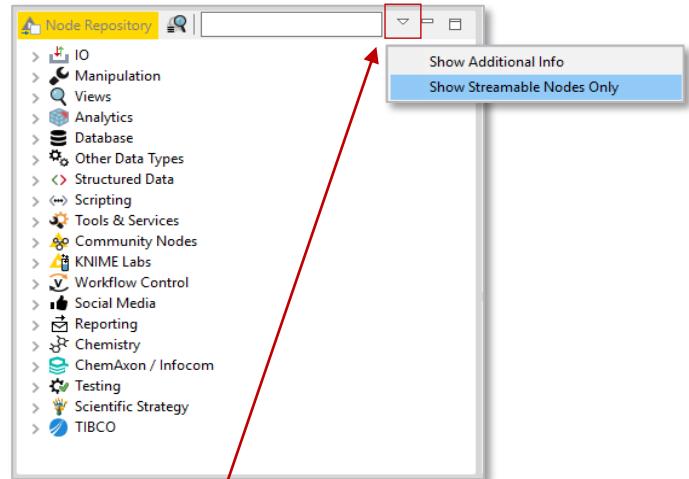
Vorteile bei großen Datensätzen

- Schnellere Abarbeitung der Workflows
- Effizienzsteigerung

Jeder Knoten sendet Teilergebnisse an den nächsten Knoten, und arbeitet weiter am nächsten Teilergebnis. Der nächste Knoten startet dann mit Bearbeitung der Teilergebnisse usw.

Streaming ist nur möglich, wenn die einzelnen Reihen unabhängig voneinander bearbeitbar sind, d.h. Teilergebnisse sinnvoll sind

Auswahl/Anzeige der „streamable“ Knoten im Node Repository

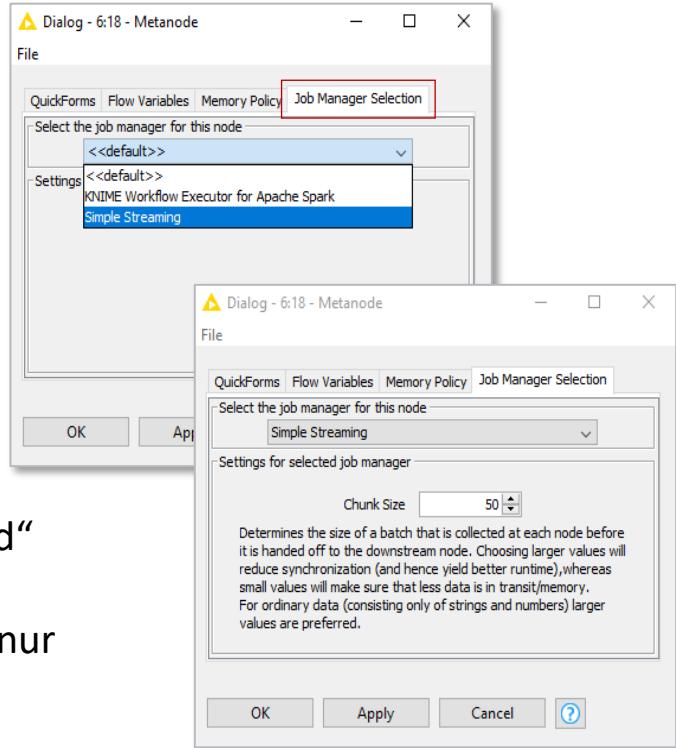
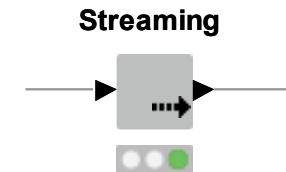


Streaming-fähige Knoten

Streaming-Funktion bei Knoten aktivieren:

1. Knoten in eine Komponente packen
2. Im Konfigurationsmenü der Komponente im Reiter „Job Manager Selection“ **Simple Streaming** auswählen
3. Gegebenenfalls Stapelgröße verändern

- Alle Knoten in der Komponente werden ge-„streamed“
- Streaming wird graphisch angezeigt >>>
- Knoten können nicht mehr einzeln gestartet werden, nur noch die gesamte Komponente



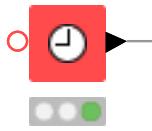
Timer Info & Global Timer Info

Geben Auskunft zu Laufzeiten der Nodes im Workflow bzw. insgesamt.

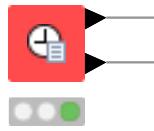
Helfen bei der Identifizierung von eventuellen Problemstellen in Workflows und Laufzeitoptimierung.

Mit einer Verbindung über Flow Variable kann der Ausführungszeitpunkt gesteuert werden.

Timer Info



Global Timer Info



Output table - 0:55 - Timer Info							
Row ID	S	Name	L	Execution Time	L	Execution Time once last Run	S
Node 33		Scatter Plot (Devic...	39664	39664	123414	1	2
Node 34		Scatter Plot (Devic...	123	123	123414	1	2
Node 49		Bar Chart (Devic...	1583	1583	1583	1	1
Node 9		Bar Chart (Devic...	14681	4981	28847	1	2
Node 20		Scatter Plot (Devic...	1205	1205	28847	1	2
Node 6		String Manipulation	9755	9763	9765	1	1
Node 8		Scatter Plot (Devic...	9676	9676	18454	1	2
Node 48		Excel Writer (OLD)	8646	8646	8646	1	1
Node 30		Text To Number	7942	7942	7942	1	1
Node 32		Text To Number	7733	7733	7733	1	1
Node 47		Excel Writer (OLD)	6395	6395	30415	1	2
Node 1		Bar Chart (Devic...	6096	6096	19979	1	2
Node 10		Scatter Plot (Devic...	5664	5664	5664	1	2
Node 7		Line Plot (Devic...	3989	3989	18328	1	2
Node 32		Table Writer	3187	3187	3187	1	1
Node 19		Text To Number	1485	1485	1485	1	2
Node 13		Scatter Plot	1248	1248	2317	1	2
Node 8		GroupBy	1211	1211	1211	1	1
Node 34		Text To Table	1055	1055	2320	1	2
Node 47		Table Writer	601	601	1440	1	2
Node 40		Image Port Reader	513	513	514	1	2
Node 29		Text To Table	500	500	508	1	2
Node 14		Image To Table	247	247	508	1	2
Node 38		Global Timer Info	183	183	191	1	2
Node 39		Global Timer Info	172	172	406	1	2
Used Nodes - 0:54 - Global Timer Info							
Row ID	S	Name	L	Overall	I	Overall	S
Row 99		org:base-node:min:mine:cluster:hierarchical:Cluster:Assigner	1	1	1	1	1
Row 100		org:base-node:min:mine:cluster:hierarchical:Cluster:Writer	1	1	1	1	1
Row 97		org:base-node:min:mine:cluster:hierarchical:HierarchicalCluster:Factory	2	2	2	2	2
Row 98		org:base-node:min:mine:cluster:hierarchical:HierarchicalCluster:View	63	1	2	2	1
Row 96		org:base-node:min:mine:cluster:hierarchical:HierarchicalCluster:Factory	123	1	2	2	1
Row 95		org:base-node:min:mine:cluster:hierarchical:HierarchicalCluster:Factory	3744	7	2	2	1
Row 94		org:base-node:min:mine:cluster:hierarchical:HierarchicalCluster:Factory	79	1	2	2	1
Row 93		org:base-node:min:mine:cluster:hierarchical:HierarchicalCluster:Factory	9	0	2	2	1
Row 92		org:base-node:min:mine:cluster:hierarchical:Nodes:Filter	219	1	1	1	1
Row 91		org:base-node:min:mine:cluster:hierarchical:Nodes:Preprocessing:Document:Filter	305	1	1	1	1
Row 90		org:base-node:min:mine:cluster:hierarchical:Nodes:Remove:Filter	305	1	1	1	1
Row 89		org:base-node:min:mine:cluster:hierarchical:Nodes:Reported:Filter	13242	60	3	3	1
Row 88		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Selector	18087	56	3	3	1
Row 87		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Table:Selector	12886	5	1	1	1
Row 86		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Table:Table:Selector	45	5	1	1	1
Row 85		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Table:Table:Table:Selector	0	0	1	1	1
Row 84		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Table:Table:Table:Table:Selector	0	0	1	1	1
Row 83		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Table:Table:Table:Table:Table:Selector	2290	7	2	2	1
Row 82		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Table:Table:Table:Table:Table:Table:Selector	80375	306	12	12	1
Row 81		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Table:Table:Table:Table:Table:Table:Table:Selector	208359	285	18	18	1
Row 80		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Table:Table:Table:Table:Table:Table:Table:Table:Selector	29	1	1	1	1
Row 79		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Selector	184	1	1	1	1
Row 78		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Selector	105	2	1	1	1
Row 77		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Selector	517	7	1	1	1
Row 76		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Selector	854	10	2	2	1
Row 75		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Selector	9	2	1	1	1
Row 74		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Selector	984	10	2	2	1
Row 73		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Table:Selector	9335	6	3	3	1
Row 72		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Selector	11382	6	3	3	1
Row 71		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Selector	84358	29	2	2	1
Row 70		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Selector	2142	8	1	1	1
Row 69		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Selector	1132	0	1	1	1
Row 68		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Selector	287	7	1	1	1
Row 67		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Selector	10244	1	0	0	1
Row 66		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Selector	3574	1	0	0	1
Row 65		org:base-node:min:mine:cluster:hierarchical:Nodes:Table:Selector	14144	1	0	0	1



Übung Streaming

Künstliche Intelligenz und Maschinelles Lernen

Was ist Künstliche Intelligenz und wo
begegnet man ihr?

Was ist Künstliche Intelligenz?

Künstliche Intelligenz (KI) ist ein Teilbereich der Informatik, der sich mit automatisierten Lösungen von Aufgaben und Problemstellungen befasst.

Durch die Übertragung von Erkenntnissen zu menschlichem Denken und Lernen auf den Computer kann eine KI selbstständig Probleme lösen. Statt für alle Aufgabenstellung programmiert zu werden, kann eine KI Fragen beantworten, deren Lösungswege sie vorher selbstständig erlernt hat.

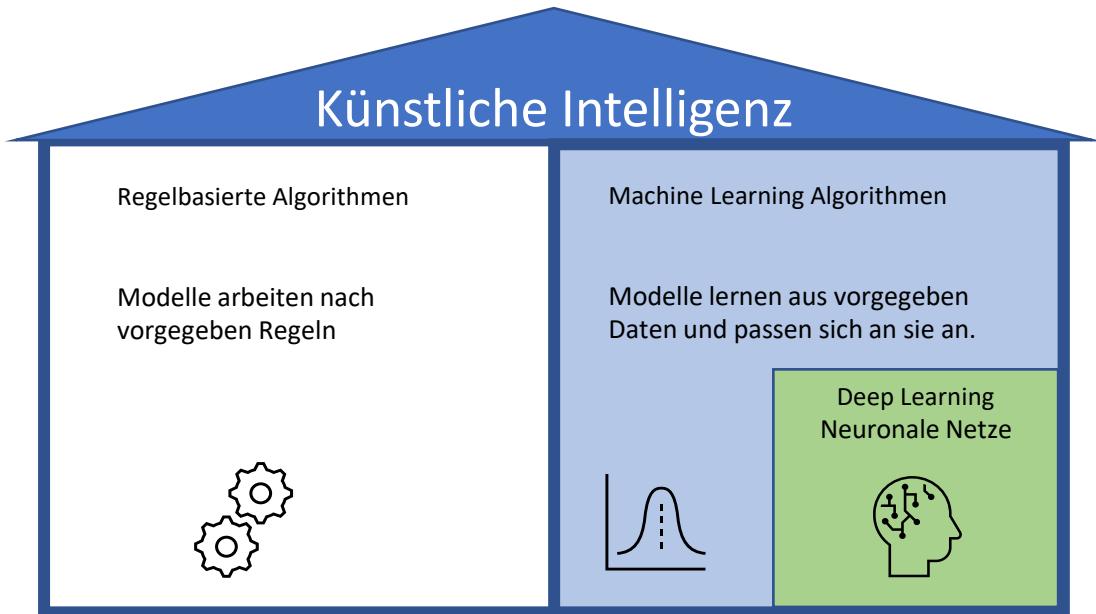
Eine „lernende“ KI wird z.B. für Prognosen über Wetter oder Energieverbrauch eingesetzt, aber auch über Entwicklung von Aktienkurse oder beim autonomen Fahren.

Die Bedeutungszunahme von KI lässt sich auf die enorme Zunahme von Daten (Big Data) zurückführen. Künstliche Intelligenz wird eingesetzt, um den hohen Automatisierungsgrad und die enorme Leistungsfähigkeit bei der Verarbeitung und der Analysen großer Datenmengen zu nutzen.

Künstliche Intelligenz in der Data Analytics

Grundlage für KI sind leistungsfähige Algorithmen:

Ein Algorithmus ist eine Art Spielregel (Handlungsanweisung), um Probleme zu lösen. Es handelt sich nicht um Software, sondern um eindeutige und ausführbare Vorschriften, die der Computer befolgen kann.



Der Algorithmus

Regelbasierter Algorithmus

Ein regelbasierter Algorithmus verarbeitet die eingehenden Daten durch Anwendung von expliziten Regeln oder lexikalischem Wissen.

Regelbasierter Algorithmus: Wenn A → dann B

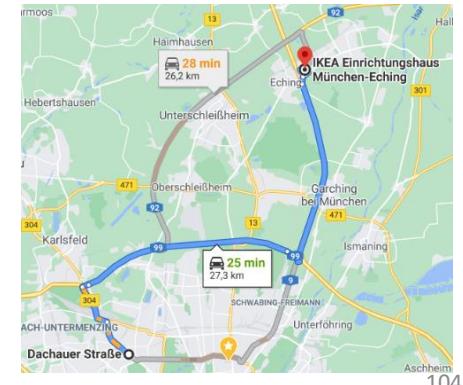
In der Waschmaschine messen Sensoren das Gewicht der Wäsche und ihren Verschmutzungsgrad. Abhängig von diesen Daten werden Wassermenge und Waschzeit automatisch eingestellt.

Machine-Learning-Algorithmus

Ein Machine-Learning-Algorithmus versucht, in den eingehenden Daten ein Muster zu erkennen und damit Informationen zu kategorisieren oder Vorhersagen zu treffen.

Was ist der schnellste Weg von A nach B?

Diese Algorithmen wenden statistische Modelle an. Sie suchen nach Mustern und Regelmäßigkeiten in den Daten, um das Problem (schnellster Weg) zu lösen. Als Ergebnis des Lernens entsteht ein Modell, das angewendet wird, um für unbekannte Daten eine Lösung zu berechnen (Umplanen bei Stau).



Künstliche Intelligenz

Starke KI

„Menschen als Maschine“

mit gleichen oder höheren intellektuellen Fähigkeiten wie Menschen

- Aktives Handeln
- Logisches Denkvermögen
- Entscheidungsfähigkeit auch bei Unsicherheit
- Planungs- und Lernfähigkeit
- Fähigkeit zur Kommunikation in natürlicher Sprache
- Kombinieren aller Fähigkeiten zur Erreichung eines übergeordneten Ziels

Schwache KI

Simulation intelligenten Verhaltens mit Mitteln der Mathematik und der Informatik bei konkreten Anwendungsproblemen.

- Reaktiv
- System ist in der Lage sich selbst zu optimieren (Lernfähigkeit) und übermenschliche Ergebnisse zu erreichen
- Umgang mit Unsicherheiten und probabilistischen Informationen
- Nur sehr begrenzt in der Lage auf andere Anwendungen zu abstrahieren

Alle heutigen Anwendung gehören noch zur schwachen KI.



Betrugs- und Risikomanagement

Beschreibung

- Versicherungen, Gutachter und Schadensprüfer müssen sich täglich mit einer Vielzahl von Versicherungsfällen sowie Neuanträgen mit einfachen bis komplexen Zusammenhängen auseinandersetzen.
- Routineprüfungen und Verwaltungsarbeiten sind zeitaufwendig und personalintensiv.

Aufgabe der KI

- Die KI kann zur Automatisierung des Versicherungsprozesses beitragen, etwa bei der Abwicklung von Schadenfällen (Mustererkennung) und beim Abschluss von Policen (Texterkennung und -verarbeitung).
- First Level Kontakt mit Kunden, beantworten einfacher Fragen und Vorbearbeitung zu Neuanträgen



Privatkunden ▾

Auto, Haus & Recht

Gesundheit & Freizeit

Vorsorge & Vermögen

Beratung

Meine Allianz & Services

Auto

Kfz-Versicherung

Schutzbrieft

Oldtimerversicherung

Motorradversicherung

IM ÜBERBLICK

Haus und Wohnen

Hausratversicherung

Wohngebäudeversicherung

Haus-Haftpflicht

Baufinanzierung

IM ÜBERBLICK

Haftpflicht

Privat-Haftpflicht

Hausrat-Haftpflicht Kombi

Tierhalter-Haftpflicht

Haus-Haftpflicht

IM ÜBERBLICK



Kfz-Versicherung
Jetzt wechseln und sparen!
→ MEHR ERFAHREN



Neue Produkte:

Telematik-Tarife in der Kfz-Versicherung anbieten, bei denen das Fahrverhalten mittels einer Blackbox überwacht wird und von einer KI ausgewertet werden kann.



Predictive Maintenance

Beschreibung

- Unternehmen stehen vor großen Herausforderungen trotz ungeplanter Produktionsausfälle, plötzlichen Maschinenstopps sowie alternde Infrastruktur eine effiziente Produktion aufrecht zu erhalten.

Aufgabe der KI

- Auswerten der Leistungsdaten der laufenden Produktion.
- Erkennen von Mustern die mit Performance-Einbußen oder Ausfällen korrelieren, die sonst unerkannt bleiben.
- Durch Abgleich mit Wartungsprotokollen und Simulation für die folgende Produktion in einem digitalen Zwilling ist eine genauere Vorhersage für notwendige Wartungsplanungen möglich.

The screenshot shows the COPADATA website's navigation bar with links for Produkte, Branchen, Support, News, Downloads, and Unternehmen. Below the navigation is a breadcrumb trail: Home > Produkte > Predictive Maintenance. A sidebar on the left has links for Zurück and Predictive Maintenance. The main content area displays a photograph of two men in a factory setting: one man is working on a piece of industrial equipment, while the other man, wearing a grey cardigan, holds a laptop and looks towards the camera. The text "Predictive Maintenance" is overlaid on the image.

Die Firma COPADATA in Verbindung mit Azure Machine Learning von Microsoft wertet Produktionsdaten aus und errechnet Wartungspläne zur Vermeidung von ungeplanten Produktionsausfällen.





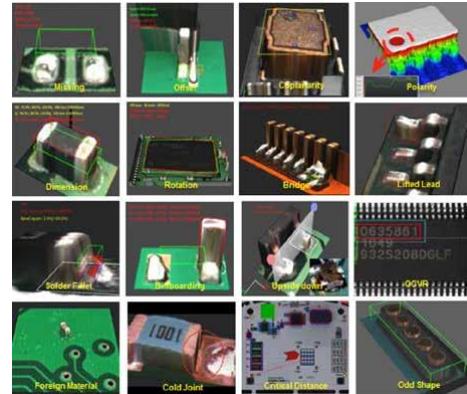
Qualitätskontrolle und -analyse

Beschreibung

- Qualität spielt zunehmend eine größere Rolle zur Vermeidung von Verschwendungen und Produktausfällen. Während des Produktionsprozesses können anhand von zahlreichen Sensoren und Meswerten rechtzeitig Mängel festgestellt und Gegenmaßnahmen eingeleitet werden.

Aufgabe der KI

- Durch automatisierte Datenanalyse (z.B. Bildanalyse) können in sehr kurzer Zeit Abweichungen im Produkt und Produktionsprozess erkannt werden.
- Korrelationen zu anderen Betriebsdaten (z.B. Produktionseinstellungen, Temperatur, Komponentenherkunft) ermöglichen eine frühzeitige Erkennung von systematischen Abweichungen bzw. eine Optimierung des Produktionsprozesse.



Innovative Solutions

We supply 3D total solutions for inline production that play a major role in SMT process optimization and yield improvements based on the world leading technology. The solutions monitor process quality in real-time and optimize the process by diagnosing defects accurately based on 3D measurement in printing, mounting, and reflow stages.



3D-Qualitätskontrolle in der Elektronikproduktion



Wie funktioniert Maschinelles Lernen und welche Arten des ML gibt es?

Machine Learning (ML)

*Algorithmen sind bei der KI der Punkt
bei der Wissen aus Erfahrung generiert wird.*

Lernphase

- Aufbau eines statistischen Modells (Mustererkennung) anhand von Eingabedaten

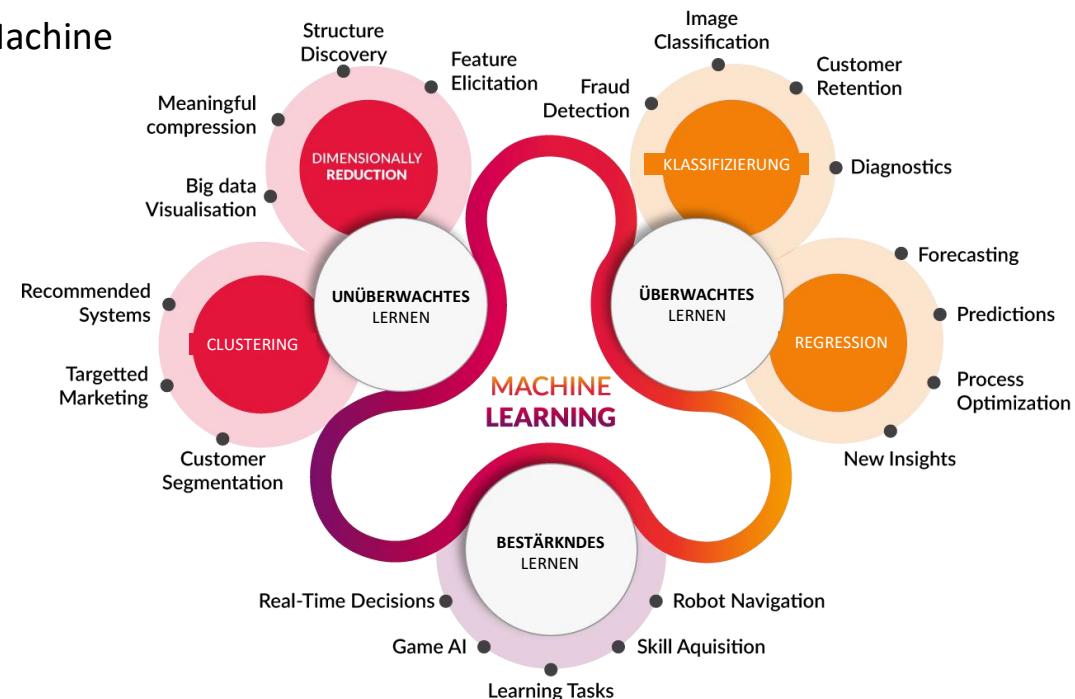
Testphase

- Abstraktion und Anwendung auf unbekannte Daten vom gleichen Typ

Typen des Machine Learnings

Es gibt 3 Hauptkategorien im Machine Learning:

- Überwachtes
- Unüberwachtes
- Bestärkendes



Bestärkendes Lernen

Beim bestärkenden Lernen (englisch: Reinforcement Learning) lässt man das Modell frei in einer Simulation interagieren, häufig ohne Vorwissen (ohne Daten) oder Beschränkungen. Der Lernprozess geschieht anhand von „Belohnungen“, indem vorteilhafte Versuche positiv bewertet werden (trial and error).

Das Modell entwickelt so eine Strategie zur Lösung des Problems, indem es versucht, seine Belohnungen zu maximieren. Dadurch entstehen häufig sehr effiziente Lösungen.

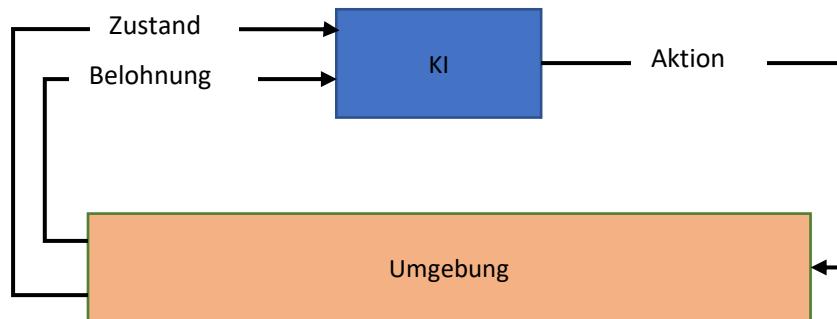
Anwendungen:

- Robotics (Optimierung von Bewegungsabläufen von Produktionsrobotern)
- Verkehrssteuerung (Züge, Straßenverkehr, Transport, ...)
- Spiele-KI
- Autonomes Fahren, Assistenzsysteme beim Einparken
- Prozessoptimierung (Optimierung von Logistikprozessen)



Reinforcement Learning

Lernen nach dem Prinzip Try and Error



- Feedbacksysteme: Richtige Handlungen werden belohnt und falsche bestraft.
- Der Simulation wird kein Domainwissen vorgegeben.
- Es wird menschliches Lernen simuliert:
→ Mit der steigender Zahl der Versuche findet das Modell eine optimalere Lösung

Überwachtes Lernen

Erklären Sie den Prozess der überwachten Lernens.

- Welche Voraussetzungen sind wichtig?
- Wie funktionieren die unterschiedlichen Algorithmen?

Überwachtes Lernen

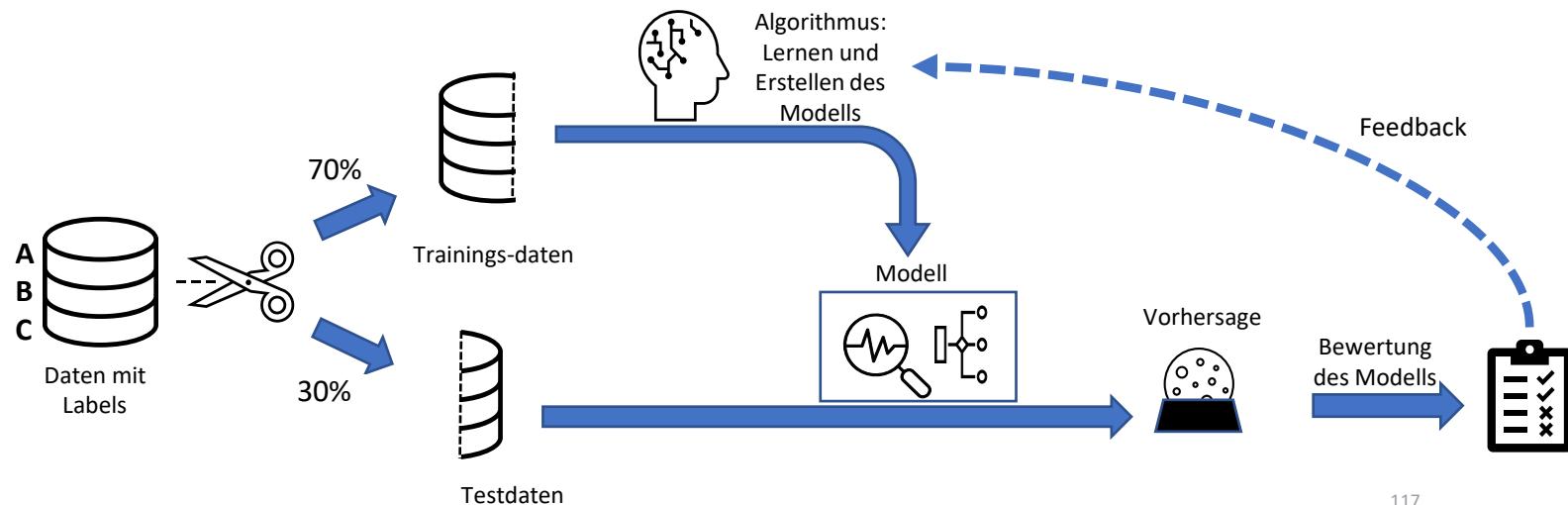
Beim Überwachten Lernen (engl. Supervised Machine Learning) werden bekannte Daten genutzt, um Muster und Zusammenhänge abzuleiten.

Das Modell wird trainiert mit Eingangsdaten, für die man das Ergebnis kennt. Dadurch lernt es einen Berechnungsprozess, sodass es unbekannte Daten bearbeiten kann.

Beim Überwachten Lernen soll entweder eine **Klassifikation** (Zuordnung in eine Kategorie) oder eine **Regression** (Zuordnung eines berechneten Wertes) erlernt und auf unbekannte Daten angewendet werden.

Generieren eines Modells im Überwachten Lernen: Trainieren und Testen

- Strukturierte Daten werden zur Beurteilung der Qualität des Modells in Trainings- und Testdaten aufgeteilt.
- Die Beurteilung geschieht anhand des Vergleichs zwischen den bekannten Kategorien der Daten und der Vorhersage.



Strukturierte und unstrukturierte Daten



Data analysis is a process of inspecting, cleansing, transforming and modeling data with the goal of discovering useful information, informing conclusions and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, and is used in different business, science, and social science domains. In today's business world, data analysis plays a role in making decisions more scientific and helping businesses operate more effectively.

Xia, B. S., & Gong, P. (2015). Review of business intelligence through data analysis. *Benchmarking*, 21(2), 300-311. doi:10.1108/BM-08-2012-0050

Unstrukturierte Daten



#Essen, #Fastfood

- Burger
- Pommes frites

	A	D	E	F	G	H
1	row ID	PassagierID	Klasse	Alter	Geschwister	Preis
2	Row0	1	1	60	0	76,2917
3	Row1	4	1	37	1	52,5542
4	Row2	7	1	30	1	57,75
5	Row3	9	1	71	0	49,5042
6	Row4	10	1	48	1	76,7292
7	Row5	14	1	49	0	26
8	Row6	23	1	41	0	134,5
9	Row7	27	1	47	1	227,525
10	Row8	29	1	61	0	32,3208
11	Row9	45	1	58	0	153,4625
12	Row10	47	1	42	0	26,55
13	Row11	49	1	61	1	262,375

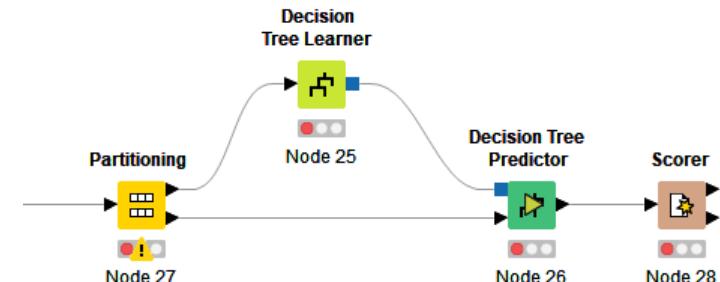
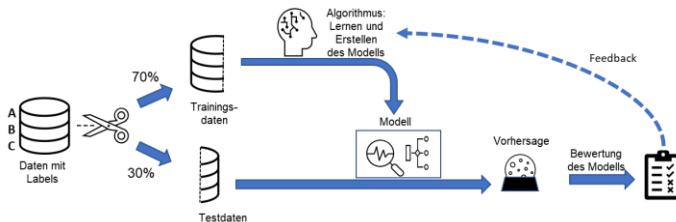
Strukturierte Daten:

- Speicherung von Text in einer zeilen- und spaltenorientierten Datenbank
- Labels und Identifikatoren zu Bildern

Das Vorgehen: Learner/Predictor

Diese Logik wird in KNIME genauso umgesetzt:

- Aufteilung der Daten in Test und Training
- Lernen des Modells mit Training Daten (Learner)
- Anwendung des Modells auf Test Daten (Predictor)
- Bewertung des Modells (Scorer)

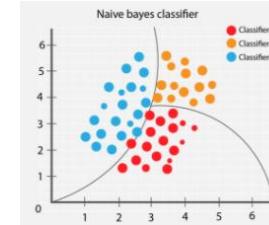


Wichtige Algorithmen für überwachtes Lernen

Klassifikation

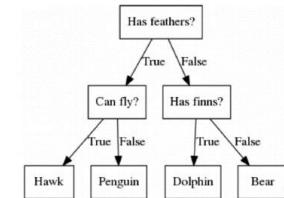
Naives Bayes

Erkenntnisgewinnung bei der Anwendung von großen und unbekannten Datensätzen; Schnelligkeit vor Genauigkeit (z. B. Spamfilter, Sentiment Analysen)



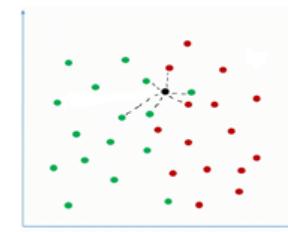
Decision Tree und Tree Ensembles (Entscheidungsbäume)

Verwendung bei bekannten Daten mit Abhängigkeiten und Kausalitäten; Genauigkeit vor Schnelligkeit (z. B. Kreditwürdigkeit von Bankkunden klassifizieren)



K-nearest Neighbor (kNN)

Verfahren, mit dem neue benachbarte Daten auf Basis vorhandener Daten klassifiziert werden können (z. B. Vorhersage Fertigungsprozesse)

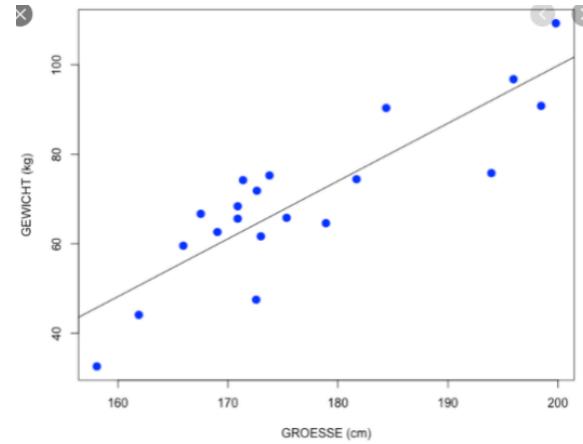


Wichtige Algorithmen für überwachtes Lernen

Regression

Lineare Regression

Aus einem Datensatz mit linearer Abhängigkeit werden unbekannte Daten quantitativ vorhergesagt (z. B. Vorhersage des durchschnittlichen Gewichts bei bestimmten Körpergrößen).



Naive Bayes

Der Naives Bayes Algorithmus

Der Naive Bayes Algorithmus basiert auf dem Satz von Bayes. Hierbei geht es um bedingte Wahrscheinlichkeiten, also die Wahrscheinlichkeit (P) eines Ereignisses A, wenn ein Ereignis B bereits eingetreten ist:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Dabei müssen die Ereignisse bekannt oder zumindest berechenbar sein. Für die Klassifikation nimmt man die Häufigkeiten, mit denen die Ereignisse auftreten.

Dabei wird vorausgesetzt, dass die beiden Ereignisse voneinander unabhängig eintreten und gleichwertig sind.

Beispiel: Naive Bayes

Daten

Allgemein	Temperatur	Feuchtigkeit	Windig	Spielen
Sonnig	Mild	Normal	Ja	Ja
Bewölkt	Mild	Normal	Nein	Ja
Regen	Mild	Hoch	Ja	Nein
...

Umwandlung

Allgemein		Temperatur		Luftfeuchtigkeit		Windig		Spielen					
	Yes	No		Yes	No		Yes	No	Yes	No			
Sonnig	2	3	Heiß	2	2	Hoch	3	4	Ja	6	2	9	5
Bewölkt	4	0	Mild	4	2	Normal	6	1	Nein	3	3		
Regen	3	2	Kalt	3	1								
Sonnig	2/9	3/5	Heiß	2/9	2/5	Hoch	3/9	4/5	Ja	6/9	2/5	9/14	5/14
Bewölkt	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	Nein	3/9	3/5		
Regen	3/9	2/5	Kalt	3/9	1/5								



Neue Daten

Allgemein	Temperatur	Feuchtigkeit	Windig	Spielen
Sonnig	Kalt	Hoch	Ja	??



Wahrscheinlichkeit je Klasse:

$$\text{Spielen „Ja“} = 2/9 \times 3/9 \times 3/9 \times 6/9 \times 9/14 = 0,0053$$

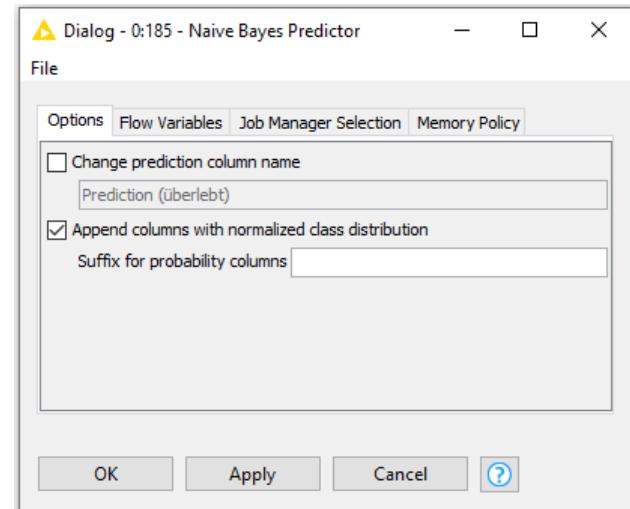
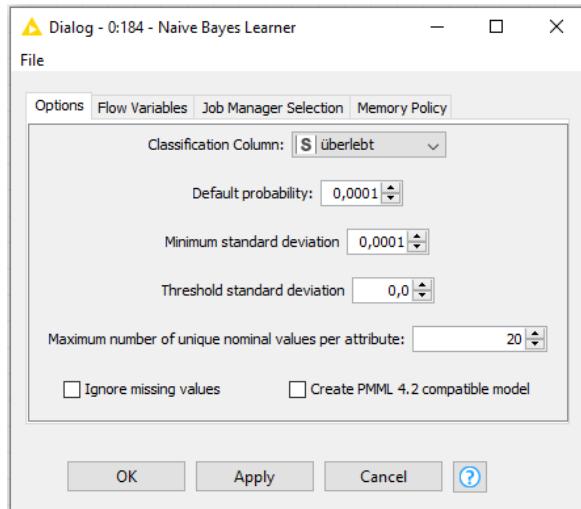
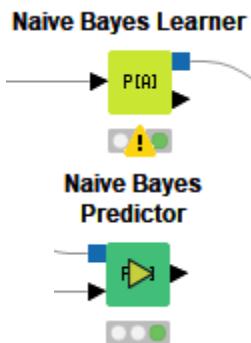
$$\text{Spielen „Nein“} = 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0,0206$$

$$\Pr(\text{Ja}) = 0,0053 / (0,0053 + 0,0206) = 0,205 \rightarrow 20,5\%$$

$$\Pr(\text{Nein}) = 0,0206 / (0,0053 + 0,0206) = 0,795 \rightarrow 79,5\%$$

In KNIME: Naive Bayes Knoten

Nach Partitionierung der Daten folgt Naive Bayes dem Learner-Predictor-Schema



Console

KNIME Console

WARN Naive Bayes Learner 0:184

The following attributes are skipped: Name/Too many values, Vorname/Too many values, Nachname/Too many values, Kabine/Too many values

Output der Naive Bayes Knoten

Statistics table - 0:184 - Naive Bayes Learner

File Hilitc Navigation View

Table "default" - Rows: 32 Spec - Columns: 6 Properties Flow Variables

Row ID	Attribute	Value	Class	Count	Mean	Standa...
Row0	Alter	?	0	98	42.473	12.872
Row1	Alter	?	1	160	36.261	13.002
Row2	Alter_bekannt	0	0	15	?	?
Row3	Alter_bekannt	0	1	18	?	?
Row4	Alter_bekannt	1	0	83	?	?
Row5	Alter_bekannt	1	1	142	?	?
Row6	Ausgangsha...	C	0	36	?	?
Row7	Ausgangsha...	C	1	83	?	?
Row8	Ausgangsha...	Q	0	0	?	?
Row9	Ausgangsha...	Q	1	2	?	?
Row10	Ausgangsha...	S	0	62	?	?
Row11	Ausgangsha...	S	1	75	?	?
Row12	Eltern_Kinder	?	0	98	0.286	
Row13	Eltern_Kinder	?	1	160	0.413	
Row14	Familiengröße	?	0	98	1.561	
Row15	Familiengröße	?	1	160	1.938	
Row16	Geschlecht	female	0	4	?	
Row17	Geschlecht	female	1	114	?	
Row18	Geschlecht	male	0	94	?	
Row19	Geschlecht	male	1	46	?	
Row20	Geschwister	?	0	98	0.276	
Row21	Geschwister	?	1	160	0.525	
Row22	Klasse	?	0	98	1	
Row23	Klasse	?	1	160	1	
Row24	PassagierID	?	0	98	665.306	375.8
Row25	PassagierID	?	1	160	614.681	355.157
Row26	Preis	?	0	98	71.499	64.322
Row27	Preis	?	1	160	101.462	93.852
Row28	Preis pro Pe...	?	0	98	45.813	39.168
Row29	Preis pro Pe...	?	1	160	62.853	74.24
Row30	überlebt	?	0	98	?	?
Row31	überlebt	?	1	160	?	?

The classified data - 0:185 - Naive Bayes Predictor

File Hilitc Navigation View

Table "default" - Rows: 65 Spec - Columns: 19 Properties Flow Variables

Row ID	S Name	S Geschlech...	D Passagier...	D Klasse	D Alter	D Familie...	D Preis pr...	S Nachnam...	S Vorname	D P (überle...)	D P (überle...)	S Predict...
Row0_Row1...	Bucknell, Mrs. William Robert (Emma Eliza Ward)	female	1	1	60	1	76.292	Bucknell	William Robert (Emma Eliza Ward)	0.208	0.792	1
Row5_Row4...	Burns, Miss. Elizabeth Margaret	female	23	1	41	1	134.5	Burns	Elizabeth Margaret	0.014	0.986	1
Row7_Row10...	Astor, Col. John Jacob	male	27	1	47	2	113.763	Astor	John Jacob	0.096	0.904	1
Row13_Row1...	Crosby, Miss. Harriet R	female	53	1	36	3	23.667	Crosby	Harriet R	0.043	0.957	1
Row17_Row1...	Beattie, Mr. Thomson	male	55	1	36	1	75.242	Beattie	Thomson	0.867	0.133	0
Row19_Row1...	Bedwith, Mrs. Richard Leonard (Sallie Monyp...	female	60	1	47	3	17.518	Bedwith	Richard Leonard (Sallie Monyp...	0.037	0.963	1
Row21_Row1...	Hippach, Mrs. Louis Albert (Ida Sophia Fischer)	female	76	1	44	2	28.99	Hippach	Louis Albert (Ida Sophia Fischer)	0.106	0.894	1
Row22_Row1...	Crosby, Capt. Edward Gifford	male	82	1	70	3	23.667	Crosby	Edward Gifford	0.833	0.167	0
Row24_Row1...	Fortune, Mr. Charles Alexander	male	97	1	19	6	43.833	Fortune	Charles Alexander	0.003	0.997	1
Row27_Row1...	Duff Gordon, Sir. Cosmo Edmund ("Mr Morgan")	male	98	1	49	2	28.465	Duff Gordon	Cosmo Edmund ("Mr Morgan")	0.774	0.226	0
Row28_Row1...	Brown, Mrs. James Joseph (Margaret Tobin)	female	142	1	44	1	27.721	Brown	James Joseph (Margaret Tobin)	0.17	0.83	1
Row38_Row2...	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	146	1	56	2	41.579	Potter	Thomas Jr (Lily Alexenia Wilson)	0.146	0.854	1
Row43_Row3...	Widener, Mrs. George Dunton (Eleanor Elkins)	female	163	1	50	3	70.5	Widener	George Dunton (Eleanor Elkins)	0.003	0.997	1
Row52_Row2...	Bird, Miss. Ellen	female	193	1	29	1	221.779	Bird	Ellen	0.003	0.997	1
Row54_Row2...	Newson, Miss. Helen Monypeny	female	202	1	19	3	8.761	Newson	Helen Monypeny	0.02	0.98	1
Row57_Row2...	Carter, Mr. William Ernest	male	218	1	36	4	30	Carter	William Ernest	0.183	0.817	1
Row59_Row2...	Millet, Mr. Francis Davis	male	229	1	65	1	26.55	Millet	Francis Davis	0.979	0.021	0
Row60_Row1...	Kimball, Mr. Edwin Nelson Jr	male	230	1	42	2	26.277	Kimball	Edwin Nelson Jr	0.834	0.166	0

Wie lassen sich Ergebnisse von
Klassifikationen bewerten?

Confusion Matrix (Wahrheitsmatrix)

Zur Bewertung von Klassifikationsmodellen vergleicht man die Vorhersage mit den „wahren“ Werten.

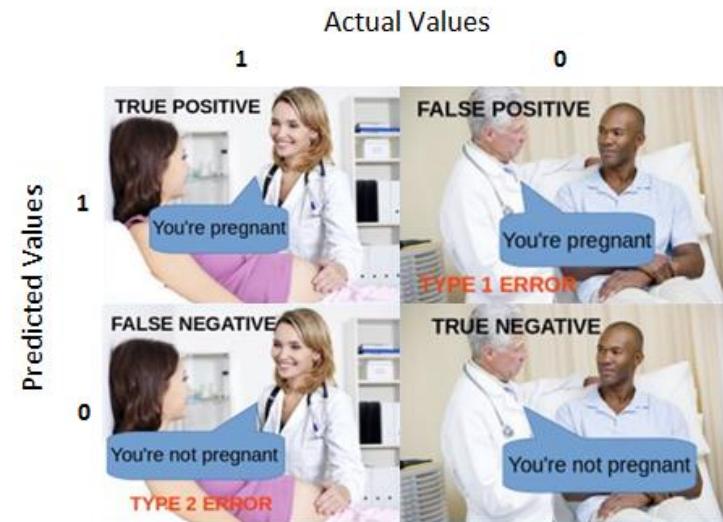
		Actual = Yes	Actual = No	
Predicted = Yes	TP	FP		
	FN	TN		

(TP) Richtig positiv: Der Patient ist krank, und der Test hat dies richtig angezeigt.

(FN) Falsch negativ: Der Patient ist krank, aber der Test hat ihn fälschlicherweise als gesund eingestuft.

(FP) Falsch positiv: Der Patient ist gesund, aber der Test hat ihn fälschlicherweise als krank eingestuft.

(TN) Richtig negativ: Der Patient ist gesund, und der Test hat dies richtig angezeigt





Übung Naive Bayes

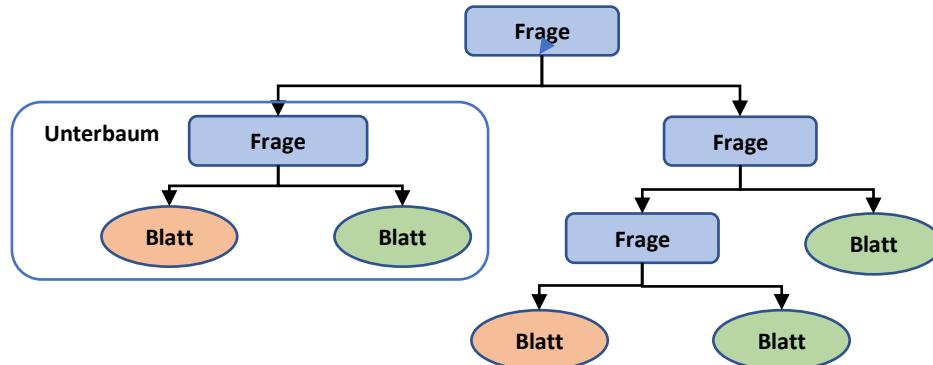
Entscheidungsbäume

Entscheidungsbäume (Decision Trees)

Entscheidungsbäume (Decision Trees) sind Hilfsmittel bei der systematischen Auswahl mehrerer Möglichkeiten. Dabei beschreibt ein Entscheidungsbaum einen mehrstufigen Entscheidungsprozess, indem er alle möglichen Entscheidungsoptionen visualisiert.

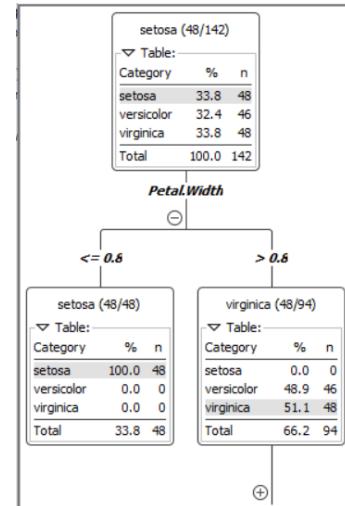
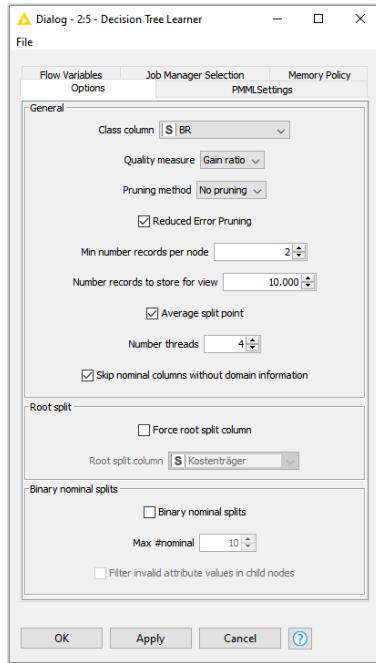
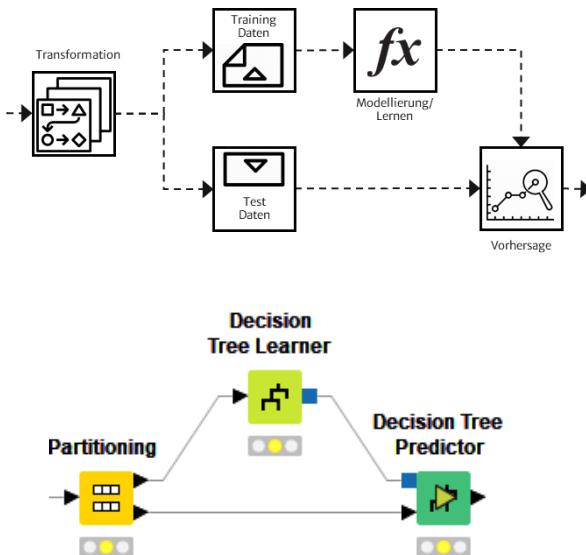
Die Entscheidungswege verzweigen sich abhängig von den verschiedenen Optionen. Diese Struktur hat große Ähnlichkeit von Ästen an einem Baum, weshalb man auch von einem Baumdiagramm spricht.

Der Sinn eines Entscheidungsbäums liegt darin, anhand von verschiedenen, visualisierten Antwortoptionen auf konkrete Fragen zu einer finalen Entscheidung zu gelangen.



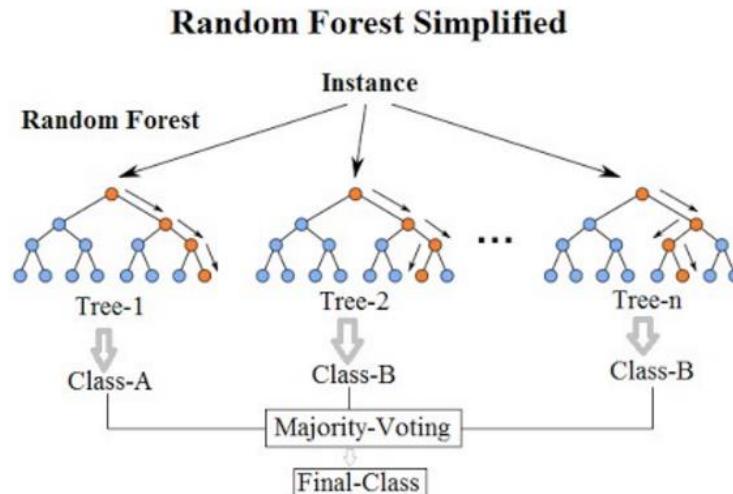
Überwachtes Lernen

- Decision-Tree



Tree Ensemble Models

Kombination der Ergebnisse einer Vielzahl von Decision Trees um ein besseres Gesamtergebnis zu bekommen – „Wisdom of the crowd“.



Zur Klassifikation: häufigste Klasse der individuellen Bäume

Zur Regression: Durchschnitt der individuellen Bäume

Random Forest

Eine Variante des Tree Ensemble Models.

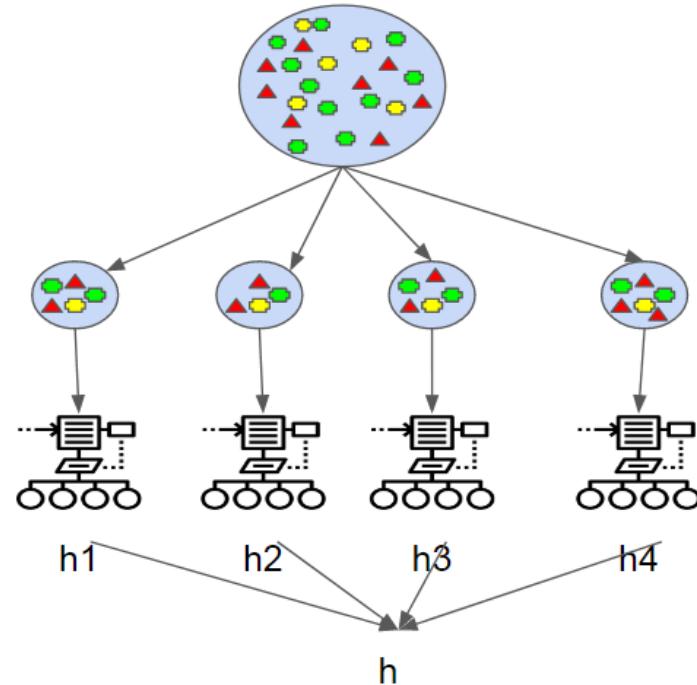
Die vielen Bäume kommen zu unterschiedlichen Ergebnissen aufgrund von zwei Methoden:

- Bagging
- Feature Randomness

→ Bereits geringe Änderungen an den Trainingsdaten beeinflussen die Baumstruktur stark!

Bagging

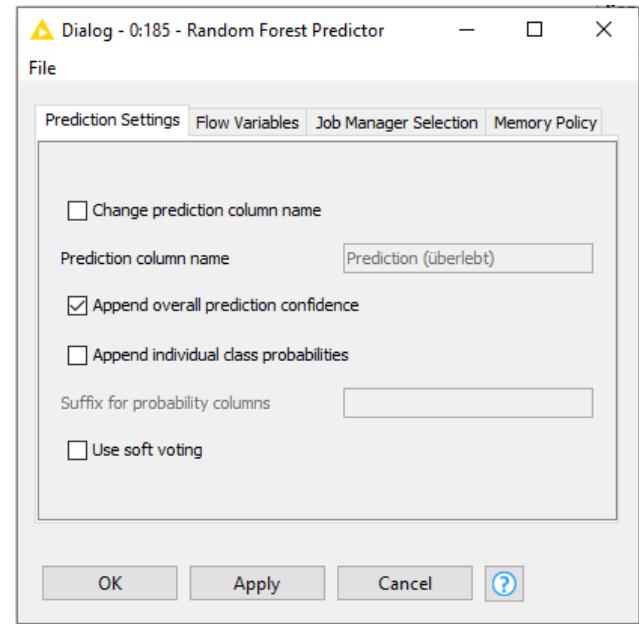
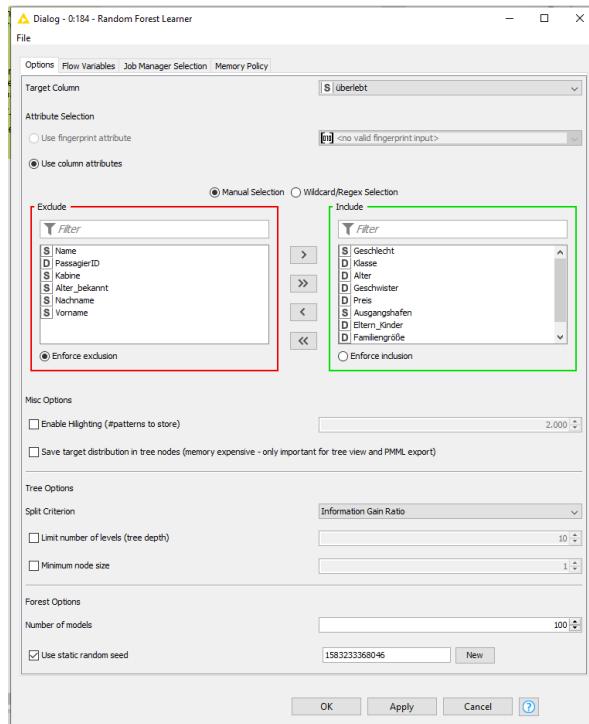
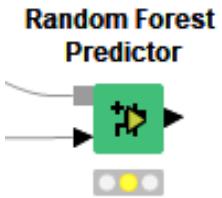
- Jeder Baum enthält nur eine Teilmenge der Zeilen des Trainingsdatensatzes.
- Die Zeilen werden zufällig ausgewählt und mehrere Bäume können dieselben Zeilen enthalten („zufälliges Ziehen mit Zurücklegen“)
- Vorteil: **mehr Daten zum Testen** der Ergebnisse
- Einzelne Bäume tragen nur zum Ergebnis bei, wenn die entsprechenden Zeilen nicht zum Training genutzt



Feature Randomness

- Jedem Baum steht nur ein Teil der Variablen (Spalten) des Trainingsdatensatzes zur Verfügung.
- In der Regel entspricht die **Anzahl der Variablen**: \sqrt{N} mit N = Anzahl Spalten. Dadurch verkürzt sich die Trainingsdauer enorm.
- Random Forest Modelle sind gegenüber **Overfitting**, im Gegenteil zu Decision Tree Modellen, die dazu tendieren die Daten auswendig zu lernen, sehr robust.

Random Forest Nodes – Learner und Predictor

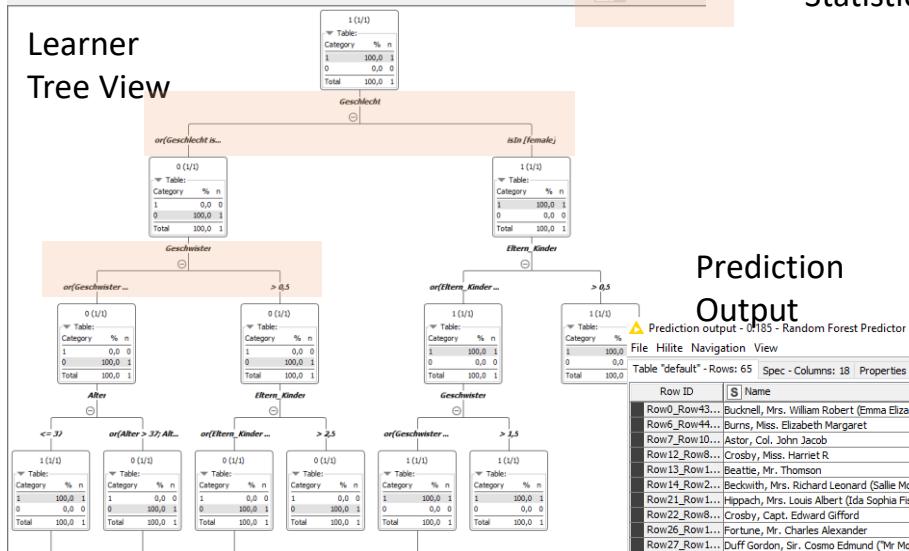


Output Random Forest Nodes

Tree View - 0:184 - Random Forest Learner

File Tree Hilite

Learner Tree View



Attribute Statistics

Attribute Statistics - 0:184 - Random Forest Learner

File Hilite Navigation View

Table "Tree Ensemble Column Statistic" - Rows: 10 Spec - Columns: 6 Properties Flow Variables

Row ID	#splits ...	#splits ...	#splits ...	#candi...	#candi...	#candi...
Geschlecht	37	22	25	37	61	105
Klasse	0	0	0	36	63	98
Alter	9	27	45	34	60	93
Geschwister	10	27	10	31	77	78
Preis	8	30	43	29	57	102
Ausgangshafen	15	7	22	27	55	99
Eltern_Kinder	4	11	22	27	52	101
Alter_bekannt	0	2	10	25	65	100
FamiliengröÙe	5	15	29	25	57	110
Preis pro Person	12	22	35	29	53	92

Prediction Output

Prediction output - 0:185 - Random Forest Predictor

File Hilite Navigation View

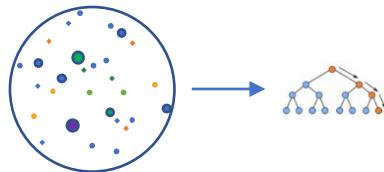
Table "default" - Rows: 65 Spec - Columns: 18 Properties Flow Variables

Row ID	S Name	S Geschle...	D Passagi...	D	milie...	D Preis p...	S Nachnam...	S Vorname	S Predicti...	D Predicti...
Row0_Row43...	Bucknell, Mrs. William Robert (Emma Eliza Ward)	female	1	1	76.292	Bucknell	William Robert (Emma Eliza Ward)	1	0.96	
Row5_Row44...	Burns, Miss. Elizabeth Margaret	female	23	1	134.5	Burns	Elizabeth Margaret	1	0.98	
Row7_Row10...	Astor, Col. John Jacob	male	27	1	113.763	Astor	John Jacob	0	0.85	
Row12_Row16...	Crosby, Miss. Harriet R	female	53	1	23.657	Crosby	Harriet R	1	0.97	
Row13_Row1...	Beattie, Mr. Thomson	male	55	1	75.242	Beattie	Thomson	0	0.7	
Row14_Row2...	Beddoe, Mrs. Richard Leonard (Sallie Monypeny)	female	60	1	17.518	Beddoe	Richard Leonard (Sallie Monypeny)	1	0.97	
Row21_Row1...	Hippach, Mrs. Louis Albert (Ida Sophia Fischer)	female	76	1	26.99	Hippach	Louis Albert (Ida Sophia Fischer)	1	0.98	
Row22_Row8...	Crosby, Capt. Edward Gifford	male	82	1	23.657	Crosby	Edward Gifford	0	0.78	
Row26_Row1...	Fortune, Mr. Charles Alexander	male	97	1	43.833	Fortune	Charles Alexander	1	0.75	
Row27_Row1...	Duff Gordon, Mrs. Cosmo Edmund ("Mr Morgan")	male	98	1	28.465	Duff Gordon	Cosmo Edmund ("Mr Morgan")	1	0.55	
Row37_Row4...	Brown, Mrs. James Joseph (Margaret Tobin)	female	142	1	27.721	Brown	James Joseph (Margaret Tobin)	1	0.88	
Row38_Row2...	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	146	1	41.579	Potter	Thomas Jr (Lily Alexenia Wilson)	1	0.96	
Row43_Row3...	Widener, Mrs. George Dunton (Eleanor Elkins)	female	163	1	70.5	Widener	George Dunton (Eleanor Elkins)	1	0.87	
Row52_Row2...	Bird, Miss. Ellen	female	193	1	221.779	Bird	Ellen	1	0.92	
Row54_Row5...	Newson, Miss. Helen Monypeny	female	202	1	8.761	Newson	Helen Monypeny	1	0.94	
Row57_Row5...	Carter, Mr. William Ernest	male	218	1	30	Carter	William Ernest	0	0.52	
Row59_Row2...	Millet, Mr. Francis Davis	male	229	1	26.55	Millet	Francis Davis	0	0.95	
Row60_Row1...	Kimball, Mr. Edwin Nelson Jr	male	230	1	26.277	Kimball	Edwin Nelson Jr	0	0.71	
					76.55	Kimball	George (George Arthur Brautman)	0	0.83	

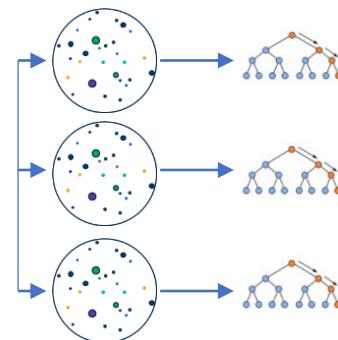
Boosting

- Gradient Boosted Trees:
- Bäume werden sequentiell gebaut
- Der nachfolgende Baum fokussiert sich auf die Fehler seines Vorgängers

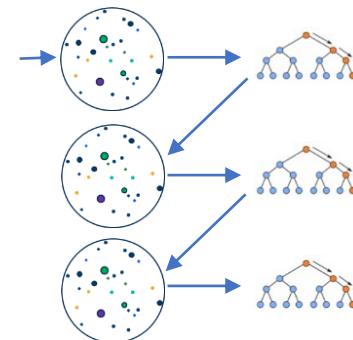
Decision Tree



Bagging (parallel)



Boosting (sequentiell)



Gradient Boosted Trees Nodes

The image shows the KNIME interface with three main components:

- Gradient Boosted Trees Learner**: A node icon with a green gradient background and a yellow 'L' symbol.
- Gradient Boosted Trees Predictor**: A node icon with a green gradient background and a yellow 'P' symbol.
- Dialog - 6:2 - Gradient Boosted Trees Learner**: A configuration dialog window.

The Dialog window contains the following sections:

- Target Column**: Set to "S überlebt".
- Attribute Selection**:
 - Use fingerprint attribute (disabled)
 - Use column attributes (selected)
- Exclude**: A list of attributes:
 - S Name
 - D PassagierID
 - S Kabine
 - S Nachname
 - S Vorname
- Include**: A list of attributes:
 - S Name
 - D PassagierID
 - D Alter
 - D Geschwister
 - D Preis
 - S Ausgangshafen
 - S Eltern_Kinder
 - S Alter_bekannt
 - D FamiliengröÙe
 - D Durchnummer
- Tree Options**:
 - Use mid point splits (only for numeric attributes)
 - Use binary splits for nominal columns
- Missing value handling**: Set to "XGBoost".
- Bagging Options**:
 - Data Sampling (Rows):
 - Fraction of data to learn single model (disabled)
 - With replacement (selected)
 - Without replacement
 - Attribute Sampling (Columns):
 - All columns (no sampling) (selected)
 - Sample (square root)
 - Sample (linear fraction)
 - Sample (absolute value)
- Attribute Selection**:
 - Use same set of attributes for entire tree (selected)
 - Use different set of attributes for each tree node
- Use static random seed**: Set to "1583751012828".

At the bottom of the dialog are OK, Apply, Cancel, and Help buttons.



Übung Entscheidungsbäume

k-Nearest Neighbor

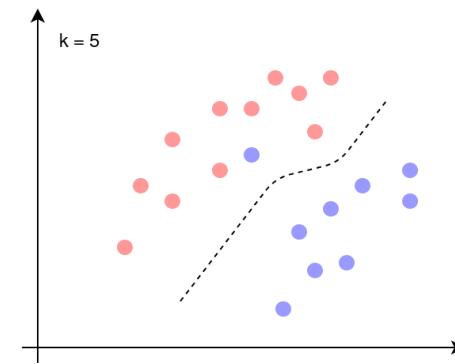
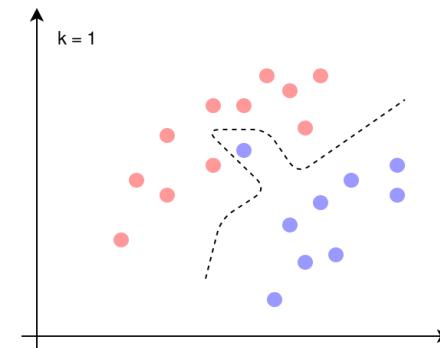
Wie werden die Nachbarn ermittelt?

1. Anzahl der Nachbarn= k

Der optimale Wert für k kann vom Analyst gewählt oder auch algorithmisch bestimmt werden, d.h. es wird mit Trainings- und Testdaten der optimalen Wert für k in Hinblick auf die Genauigkeit der Klassifizierung bestimmt.

Generell gilt:

Niedrige Werte für k machen die Klassifikation anfällig für Varianz, also einzelne falsch klassifizierte Punkte. Für zu hohe k wird die Klassifikation zu ungenau, da zu viele Punkte mit einbezogen werden, die durch ihre Distanz nicht wirklich ähnlich zu dem zu klassifizierenden Punkt sind.



Wie werden die Nachbarn ermittelt?

2. Welche Nachbarn werden berücksichtigt

Häufig werden in Analytics Anwendungen die nächsten Nachbarn nach dem Euklidischen Abstand zwischen zwei Datenpunkten berechnet:

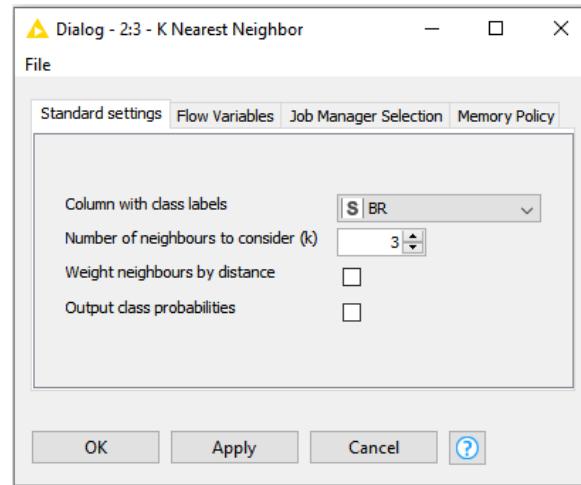
$$d(x, y) = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2}$$

Diese Abstandswerte werden für die Datenpunkte in der Umgebung berechnet und die k-kleinsten Abstandswerte berücksichtigt.

Es gilt jedoch zu beachten, dass mit dieser Methode die **Größenordnungen der Variablen x und y eine wichtige Rolle spielen**. Sind die Zahlenwerte einer dieser Variablen im Verhältnis zur anderen deutlich größer, so bekommt sie auch eine größere Gewichtung in der Auswahl der nächsten Nachbarn. Um dies auszugleichen, sollten die Wertebereiche der beiden variablen skaliert oder normalisiert werden, um sie aneinander anzugeichen.

k-Nearest-Neighbour Node

K Nearest Neighbor





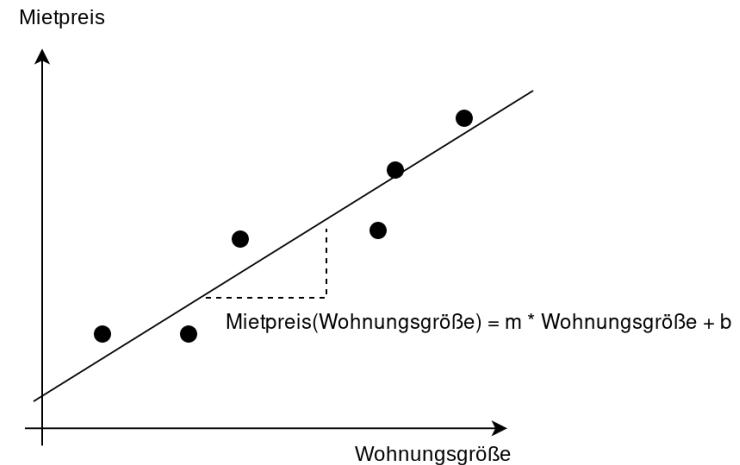
Übung k-NN

Lineare Regression

Bestimmung von Steigung und Geradenposition

Die Bestimmung der Steigung der Geraden wird anhand von Gradienten abgeleitet. Diese geben an, wie sich eine Funktion entlang bestimmter Achsen ändert.

Die Ableitung beim sogenannten Gradientenabstieg wird für jede betrachtete Dimension bzw. Attribut durchgeführt und mit ihrer Hilfe die Parameter für Steigung (m) und Geradenposition (b) angepasst.



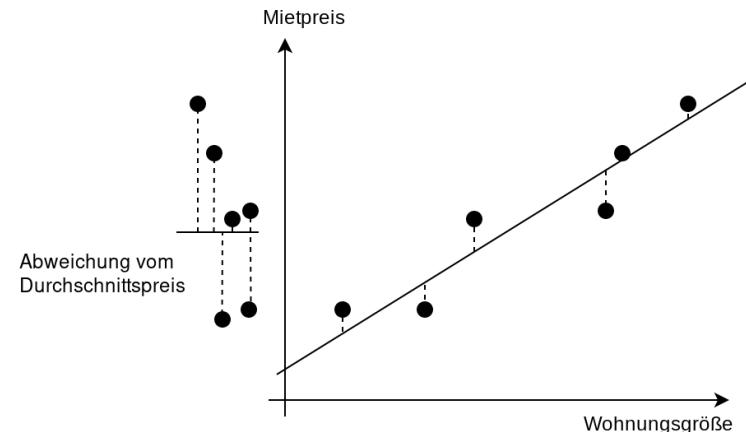
Modellgenauigkeit: R²

Um festzustellen, wie gut das Modell den Zusammenhang zwischen den Variablen abbildet hat, wird der Wert R² errechnet, mit dem sich Lineare Regressionsmodelle vergleichen lassen.

Dafür werden die Varianzen der Regressionsgeraden mit den Varianzen der Mittelwerte der Daten anhand folgender Formel verglichen und stellt einen dimensionslosen zwischen 0 und 1 dar, der aussagt, wie gut das Modell die abhängige Variable vorhersagen kann:

$$R^2 = \frac{\text{Varianz (Mittelwert)} - \text{Varianz (Regressionsgerade)}}{\text{Varianz (Mittelwert)}}$$

Ist R² = 1 liegen alle Punkte auf einer Linie und unser Modell kann zu 100% genau die Datenpunkte vorhersagen. Ist R² = 0, dann hat das Modell keine Aussagekraft und die beiden Variablen hängen wahrscheinlich nicht zusammen.





Übung Lineare Regression

Unüberwachtes Lernen

Unüberwachtes Lernen

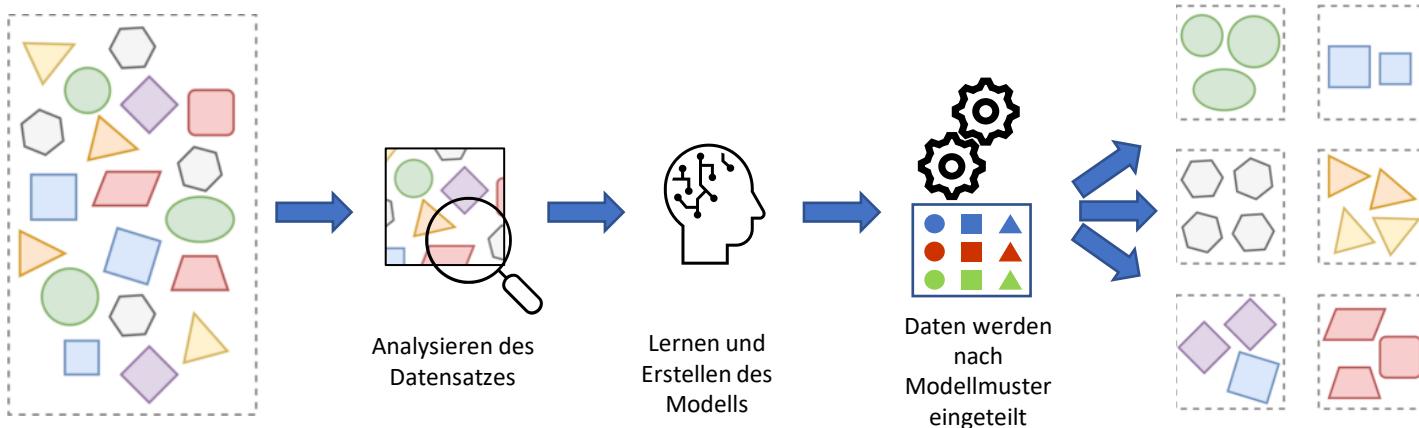
Beim unüberwachten Lernen (engl.: Unsupervised Machine Learning) wird ein Modell mit unbekannten und unstrukturierten Daten aufgebaut. **Das Modell soll eigenständig Gruppen und Muster erkennen.**

Der Einsatz von unüberwachten Lernen hilft beim Kategorisieren von unstrukturierten Daten (**Clusteranalysen**) und beim Ordnen sowie Veranschaulichen von unübersichtlichen Datenmengen (**Dimensionsreduktion**).

Durch die Reduzierung von vielen Eigenschaften auf wenige, aussagekräftige neue Mischeigenschaften verringert man die Datenmenge und benötigt weniger Rechenleistung.

Unüberwachtes Lernen

- Die Daten liegen unstrukturiert und ohne Kennzeichnungen vor.
- Ziel und Ergebnis sind unbekannt.
- Das Modell lernt Muster und sortiert die Daten danach.
- Nachdem das Modellmuster erlernt wurde, kann es auf unbekannte Daten angewandt werden.

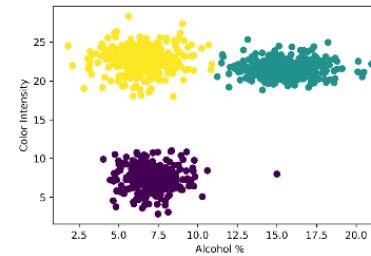


Wichtige Algorithmen für unüberwachtes Lernen

Clusteranalyse

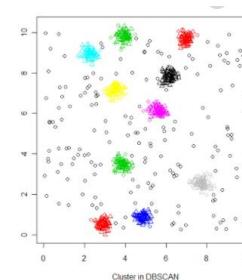
K-means:

Einteilung des Datensatzes in eine Anzahl von k Cluster, deren Datenpunkte möglichst nahe beieinanderliegen; funktioniert gut bei klaren Cluster-Strukturen und geringem Rauschen (z. B. Produktempfehlungen, Kundensegmentierung, Marktsegmentierung)



DBSCAN:

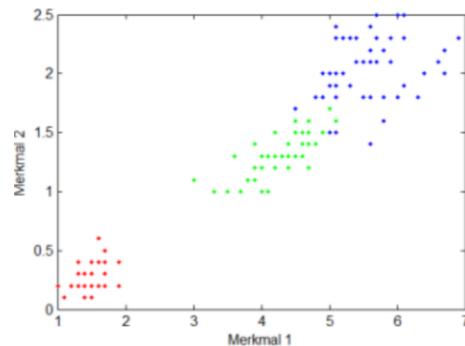
Clustert Punkte, die nah beieinanderliegen und genügend Nachbarpunkte aufweisen; kann mehrere benachbarte Cluster erkennen und ist unempfindlicher gegenüber Rauschen (z. B. räumliche Cluster unterschiedlicher Dichte, Größe und Form, komplexe dreidimensionale Datensätze)



Wichtige Algorithmen für unüberwachtes Lernen

Dimensionsreduktion

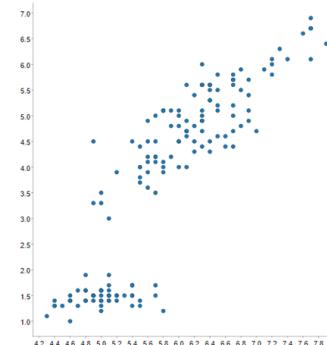
Principal Component Analysis (PCA): ersetzt die Ursprungsvariablen durch komprimierte neue Variablen unter Beibehaltung der Information; die Reduktion der Variablen soll das Modell verbessern (z. B. Satellitenbildanalyse)



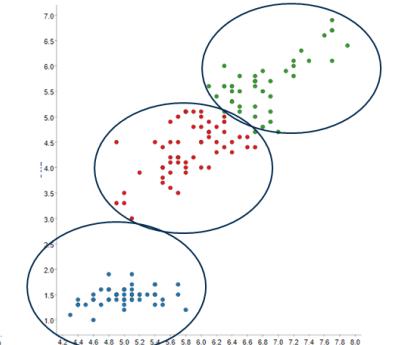
Cluster Analysen: k-means und DBSCAN.

Unüberwachtes Lernen: Clustering

- Clusteranalyse: Gruppieren von ähnlichen Datenpunkten zu Clustern
- Verschiedene Herangehensweisen an die Lösung.
- Daten Clustern (es gibt keine Label)
- Evaluation ist komplizierter (man kann Ergebnis nicht mit Labels vergleichen)
- Clustering Modell kann auf neue Daten angewandt werden



Ungruppierte Daten



Daten gruppiert in 3 Cluster

Beispiele:

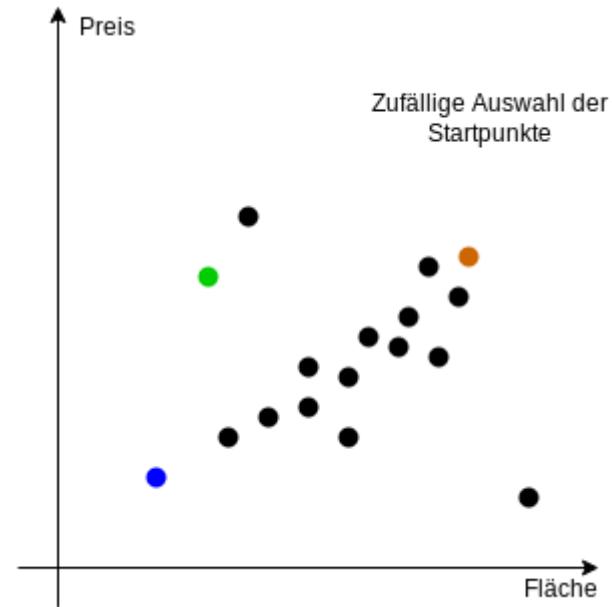
- Kundensegmentierung
- Topic Extraktion

Funktionsweise von k-means

Der Algorithmus startet mit der Festlegung der Startpunkte. Diese repräsentieren den Mittelpunkt eines Clusters. Der Data Analyst legt zu Beginn der Berechnung diese Startpunkte vor, indem er dem Algorithmus die Anzahl der Cluster mit der Zahl k vorgibt.

Jeder Punkt im Diagramm wird nun dem Cluster zugeordnet, dessen Mittelpunkt ihm am nächsten liegt. Sind alle Punkte zugeordnet, wird ein neuer Clustermittelpunkt berechnet und das Verfahren wiederholt.

Der Mittelpunkt verschiebt sich dadurch in seine optimale Position. Wenn sich der Mittelpunkt nicht mehr verschiebt, dann ist das Clustering beendet.



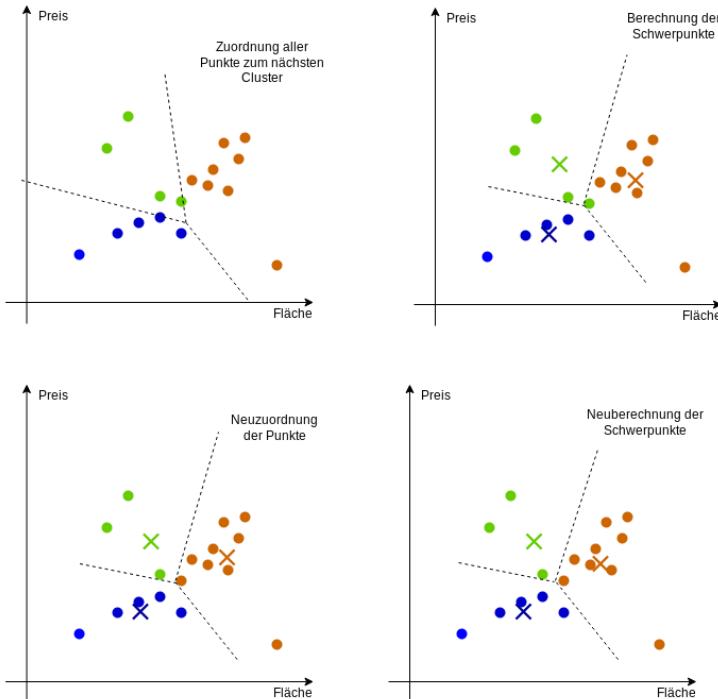
Optimierung der Cluster

Die Lage der Startpunkte bestimmt Geschwindigkeit und teilweise auch den Erfolg des K-means-Algorithmus. Werden Sie nicht vom Data Analyst vorgegeben, definieren viele Data Analytics-Anwendungen sie zufällig.

Für jeden Durchlauf werden die Varianzen der Cluster addiert. Dadurch lässt sich vergleichen, wie die Qualität eines Clusters ist.

Ist die Gesamtvarianz eines Durchlaufs hoch, heißt das, dass alle Cluster weit verstreut sind. Innerhalb der Cluster sollten die Punkte jedoch so nah wie möglich beieinander liegen.

Der optimale Durchlauf ist also der, bei dem die Gesamtvarianz am niedrigsten ist.



Clustering in KNIME

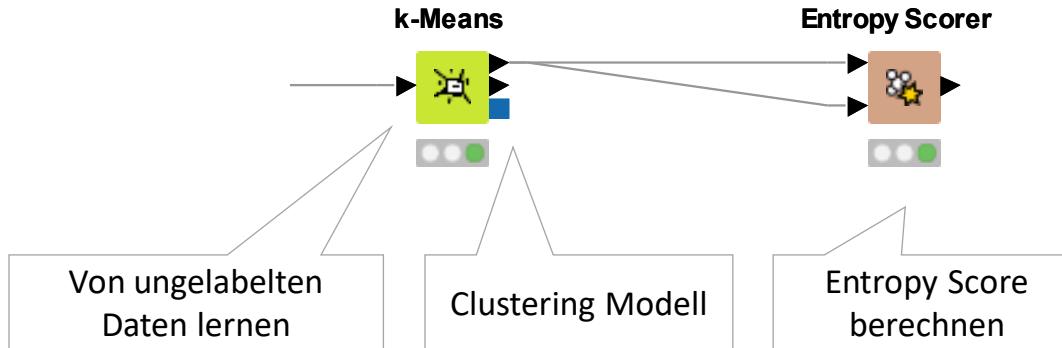


Table "default" - Rows: 288 Spec - Columns: 12 Properties Flow Variables														
Row ID	City	Country	Country	IP Addr...	Lat	Long	region_...	region	areacode	metro_...	zipcode	Cluster		
Row150	San Jose	US	United States	198.4.83.150	37.339	-121.895	CA	California	408	807	?	cluster_1		
Row162	San Jose	US	United States	99.190.99.162	37.339	-121.895	CA	California	408	807	?	cluster_1		
Row163	San Francisco	US	United States	67.188.37.163	37.761	-122.484	CA	California	415	807	94122	cluster_2		
Row183	San Diego	US	United States	66.27.120.183	32.957	-117.198	CA	California	858	825	92130	cluster_0		
Row205	San Francisco	US	United States	216.239.12...	37.775	-122.419	CA	California	415	807	?	cluster_2		
Row206	Baldwin Park	US	United States	71.160.182...	34.096	-117.967	CA	California	626	803	91706	cluster_0		
Row234	San Diego	US	United States	63.196.132...	32.715	-117.157	CA	California	619	825	?	cluster_0		
Row236	Los Angeles	US	United States	98.154.124...	34.052	-118.244	CA	California	323	803	?	cluster_0		
Row239	Oakland	US	United States	50.201.47.220	37.800	-122.227	CA	California	510	807	94122	cluster_2		



Übung k-Means

DBSCAN

Der DBSCAN Algorithmus

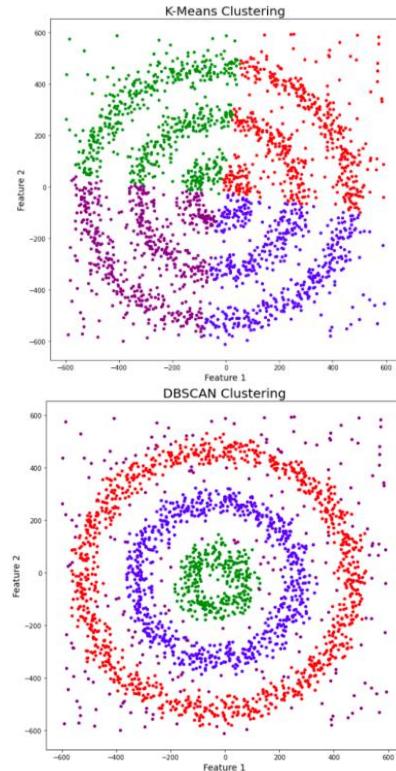
DBSCAN steht für “Density-Based Spatial Clustering of Applications with Noise”. Es ist in der Lage, beliebig geformte Cluster und Cluster mit Rauschen (also Ausreißern) zu finden.

DBSCAN ist da effektiv, wo k-means keine guten Ergebnisse erzielt, da kompliziert geformte Cluster bzw. verrauschte Daten die Abgrenzungen schwierig machen.

DBSCAN kann Cluster in großen räumlichen Datensätzen identifizieren, indem es die lokale Dichte der Datenpunkte betrachtet.

Ein wichtiger Vorteil gegenüber k-means besteht in der Fähigkeit Rausche in Daten zu erkennen und die entsprechenden Datenpunkte als Ausreißer zu kennzeichnen.

Hinzu kommt, dass die Anzahl der Cluster anhand der Daten bestimmt wird und nicht im Vorhinein vorgegeben werden muss.



Funktionsweise von DBSCAN

DBSCAN funktioniert mit zwei Parametern: Epsilon (ε) und minPoints (minPts). Epsilon ist der Radius des Kreises, der um jeden Datenpunkt zu erstellen ist, um die Dichte zu überprüfen, und minPoints ist die minimale Anzahl von Datenpunkten, die innerhalb dieses Kreises erforderlich sind, damit dieser Datenpunkt als Kernpunkt klassifiziert wird.

Ein Datenpunkt liegt entweder im Inneren einer dichten Region (Kernpunkt), auf dem Rand einer solchen (Randpunkt) oder in einem spärlich besetzten Gebiet (Rauschpunkt). Etwas präziser:

Kernpunkt (rot):

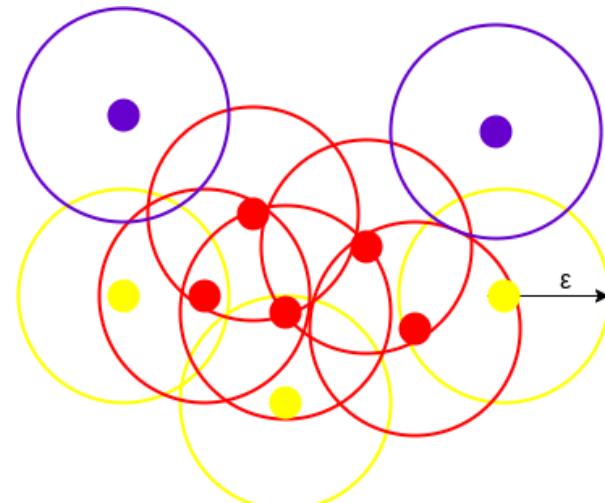
Die Anzahl der Datenpunkte in der ε -Umgebung des Kernpunkts beträgt mindestens MinPts.

Randpunkt (gelb):

Ein Randpunkt ist kein Kernpunkt, liegt aber in der ε -Umgebung eines Kernpunktes.

Rauschpunkt (violett):

Ein Rauschpunkt ist weder Kern- noch Randpunkt.



Der Analyseprozess

1. Start an einem beliebigen Punkt, an dem mit dem Parameters Epsilon die Anzahl der Nachbarn bestimmt wird.
2. Hat der Punkt genug weitere Datenpunkte in der Umgebung gefunden, dann ist das ein Kernpunkt und startet hier ein Cluster sonst wird er als Rauschen eingestuft.
3. Kernpunkte formen zusammen ein Cluster bis auf Randpunkte stoßen.
4. Der Prozess setzt sich fort, bis ein Cluster komplett erfasst ist.
5. Im Anschluss startet der Algorithmus wieder an einem weiteren beliebigen Punkt und wiederholt die Schritte 1-4 bis alle Punkte untersucht wurden.

Einfluss von Epsilon und minPts:

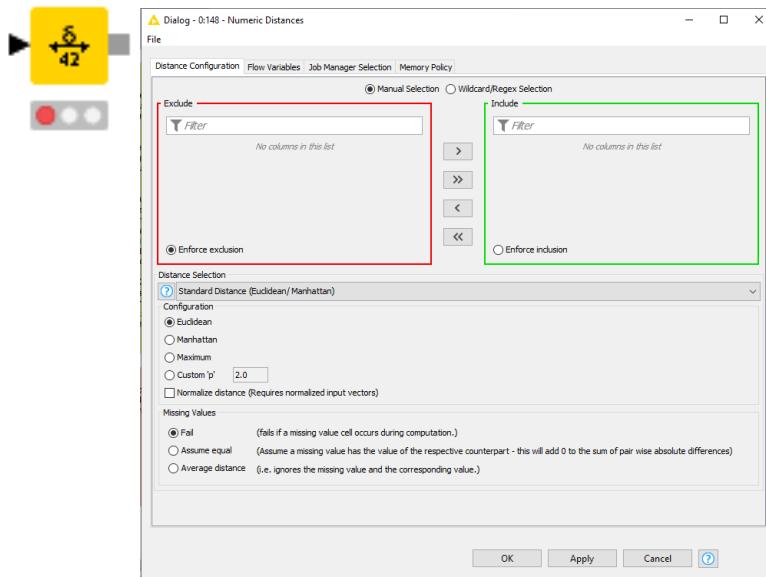
Niedrige Epsilon-Werte verhindern ggf. Erkennen von Clustern bei hohen Werten wachsen viele Cluster zusammen

Faustregel zu minPts: (Anzahl der Attribute) +1. Je mehr Rauschen vorhanden ist, desto höher sollte der Wert gewählt werden.

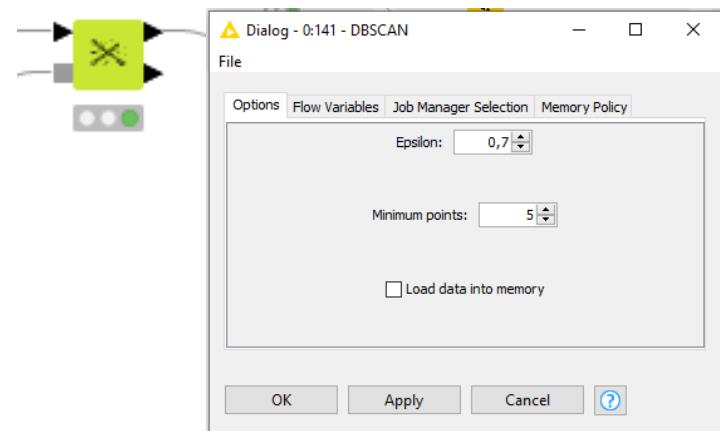
In KNIME

Numeric Distances und DBSCAN

Numeric Distances



DBSCAN





Übung DBSCAN

Wozu dient die Dimensionsreduktion und wie funktioniert sie am Beispiel der PCA?

Dimensionsreduktion

Bestimmte Fragestellungen und Datensammlungen enthalten sehr viele Variablen. Bsp: Vorhersage des Bruttoinlandsprodukts

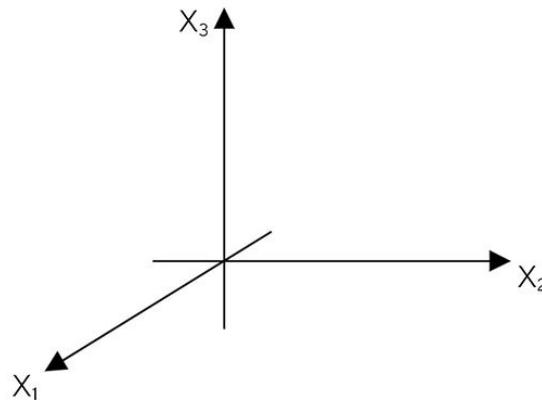
Problem:

- Man müsste die Beziehungen zwischen all diesen Variablen durchdenken
- Overfitting und Rauschen durch unnötige Variablen soll vermieden werden
- Werden die Voraussetzungen der Algorithmen berücksichtigt?
- → Feature Elimination
- → Feature Extraction

Funktionsweise von PCA

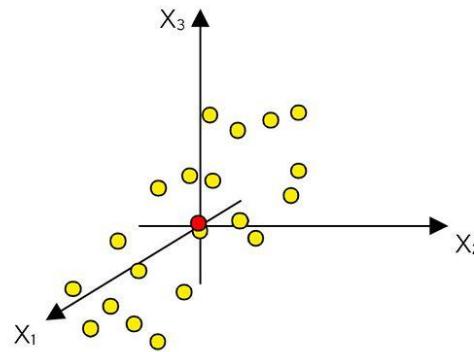
Beim Ausführen der PCA werden die numerischen Attribute durch eine orthogonale Repräsentation ihrer Werte ersetzt. Die Werte werden dabei linear kombiniert, so dass die wichtigsten Informationen (z.B. Korrelationen) erhalten bleiben. Durch den Vorgang entstehen neue Merkmale, die Principal components oder auch Hauptmerkmale.

1. Jedes numerische Attribut wird zunächst standardisiert, um Unterschiede in der Skalierung auszugleichen.
2. Jedes der k (Anzahl) numerischen Attribute (x_1 - x_k) wird dann als Dimension(Achse) in einem k -dimensionalen Raum dargestellt

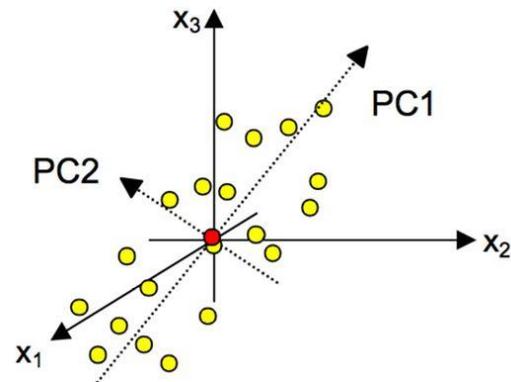


Funktionsweise von PCA

3. Die Datenpunkte werden in diesem Raum eingetragen, der Mittelwert gebildet (roter Punkt) und der Datensatz so verschoben, dass der Mittelwert den 0-Punkt bildet:

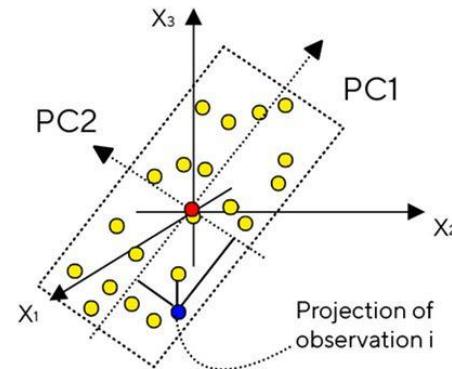


4. Die erste Principal Component (PC1) wird als Gerade, die durch den Mittelwert läuft, über eine lineare Regression für die Datenpunkte berechnet.
5. Eine zweite Principal Component (PC2) wird senkrecht durch den Mittelwert zur ersten berechnet, so dass auch sie die Daten bestmöglich wiedergibt.



Funktionsweise von PCA

6. Die zwei Geraden formen eine Ebene, für die jeder Datenwerte eine relative Position besitzt. Diese Positionen in der Ebenen sind die extrahierten Merkmale aus der PCA.





Übung PCA

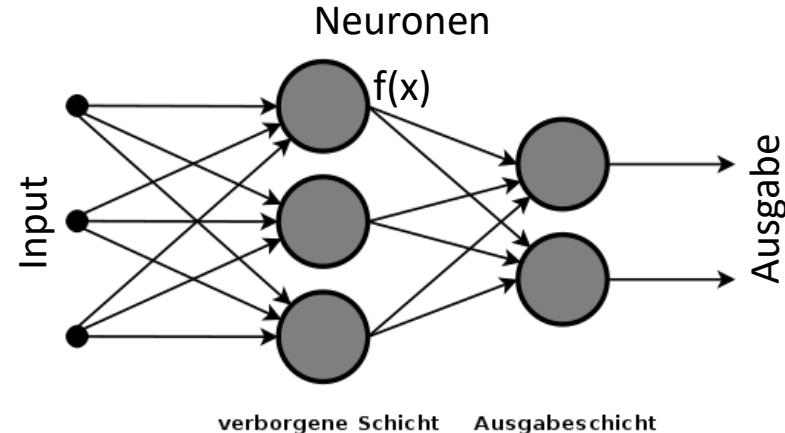
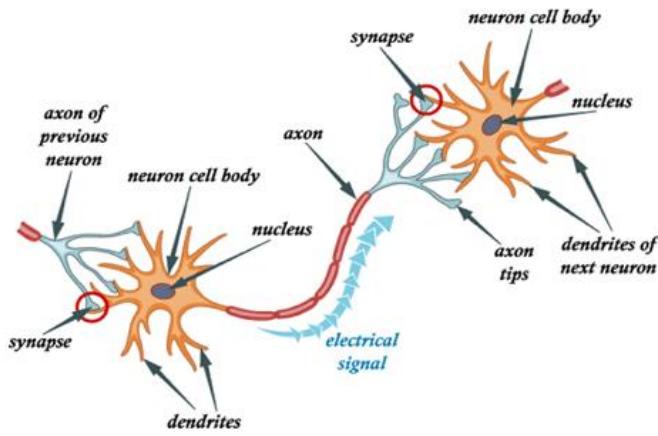
Neuronale Netze

Wie sind Neuronale Netze aufgebaut und wie werden Daten in ihnen verarbeitet?

Funktionsprinzip neuronaler Netze

Adaption der Strukturen des Menschlichen Gehirns (Neuronen)

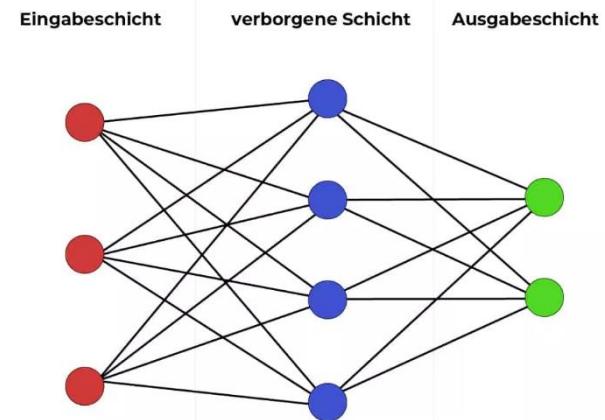
- Ein Neuron hat Eingänge (Inputs), die gewichtet miteinander verarbeitet werden.
- Aus der Summe der Eingänge wird mit Hilfe einer sogenannten Aktivierungsfunktion das Ergebnis (Output) des Neurons errechnet.
- Dieses Ergebnis wird an andere Neuronen weitergegeben, die auf die gleiche Weise funktionieren.



Aufbau neuronaler Netze

Ein neuronales Netz besteht in der Regel aus drei verschiedene Schichten, denen jeweils eine Art Neuronen zugeordnet werden kann:

- Input-Neuronen (Eingabeschicht),
- Output-Neuronen(Ausgabeschicht)
- Hidden-Neuronen (verborgene Schichten)



Eingabeschicht:

Die Eingangsschicht versorgt das neuronale Netz mit den notwendigen Informationen, indem sie diese gewichtet an die nächste Schicht weiterleitet.

Verbogene Schicht:

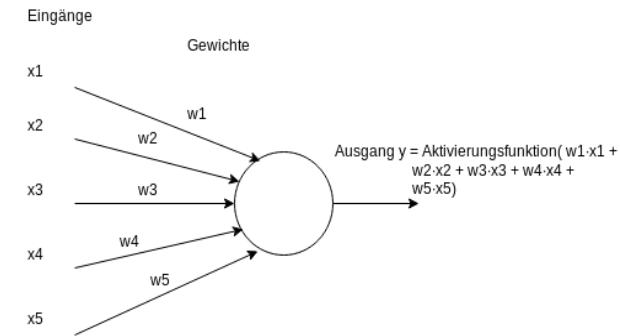
Hier werden die empfangenen Informationen erneut gewichtet und weitergereicht. Während in der Ein- und Ausgabeschicht die eingehenden und ausgehenden Daten sichtbar sind, ist der innere Bereich des Neuronalen Netzes im Prinzip eine Black Box.

Ausgabeschicht:

Die Ausgabeschicht ist die letzte Schicht und schließt unmittelbar an die letzte Ebene der verborgenen Schicht an. Die Output-Neuronen beinhalten die resultierende Entscheidung, die als Informationsfluss hervorgeht.

Wie funktioniert die Datenverarbeitung in neuronalen Netzen?

- Die Daten werden über die Eingangsschicht erfasst und von jedem Neuron mit einem individuellen Gewicht bewertet.
- Das Ergebnis dieser Berechnung wird an die nächsten Neuronen der nächsten Schicht oder des nächsten Layers weitergegeben.
- Je nach Ergebnis der berechneten Inputwerte wird ein Neuron aktiviert oder deaktiviert.
- Eine Berechnung des Gesamtergebnis geschieht am Outputlayer.



Beim folgenden Durchlauf der Daten, werden die Gewichtungen anhand der Fehler angepasst. Auf diese Weise „lernt“ das neuronale Netz mit jedem Mal besser, von den Inputdaten auf bekannte Outputdaten zu schließen.

Backpropagation und Lern-Rate

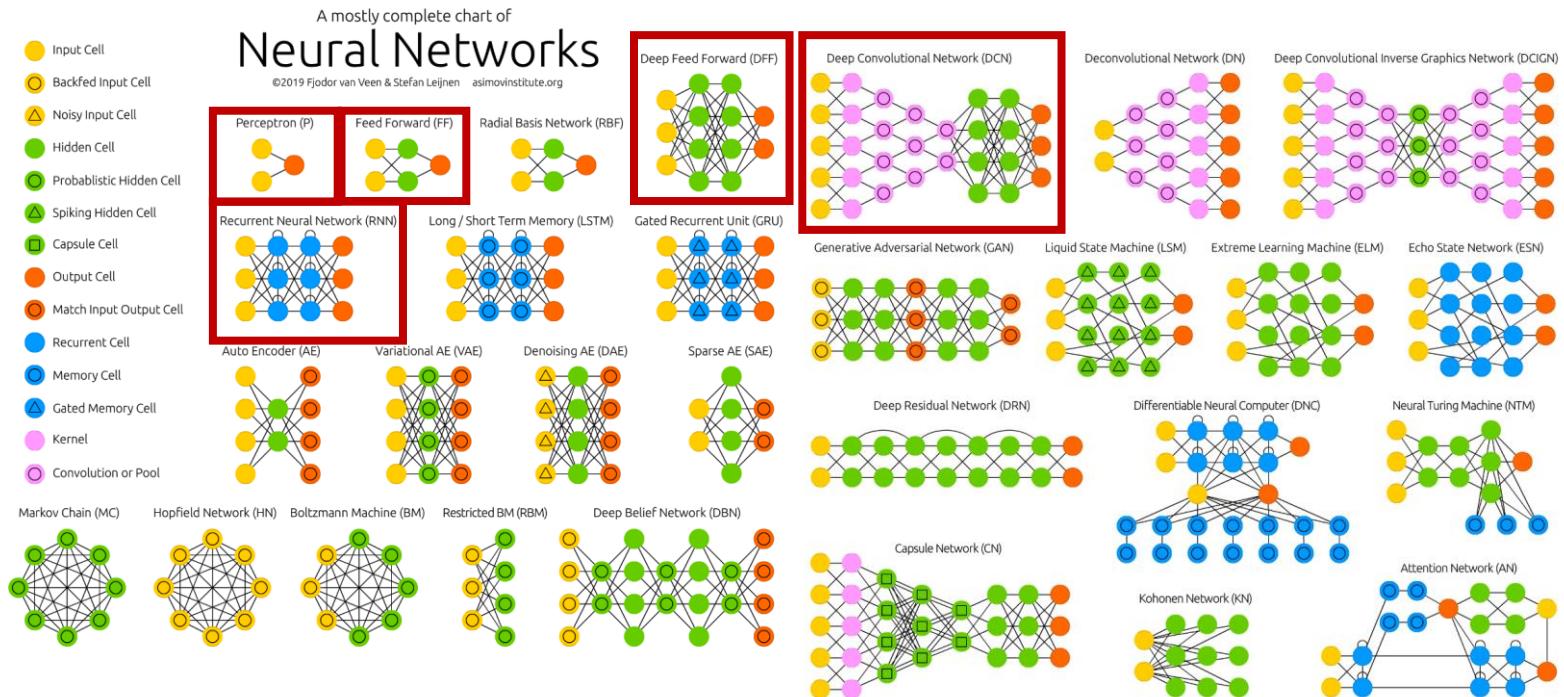
Backpropagation

- Damit ein NN lernen kann benötigt es Feedback. Dazu vergleicht das NN den produzierten Output mit dem Soll-Output.
- Mit Hilfe der Abweichung vom Soll-Output durchläuft das NN den Prozess nun rückwärts. Dabei wird die Gewichtung der Verbindungen zwischen den Units angepasst.
- Dabei wird jede Verbindung dahingehend inspiziert wie sich das Ergebnis verändern würde mit einer anderen Gewichtung.

Lern-Rate

- Legt fest wie stark ein NN die Gewichtung je Durchlauf verändert.

Neuronale Netzwerktypen



<https://www.jaai.de/post/kuenstliche-neuronale-netze-aufbau-funktion>

Perceptron

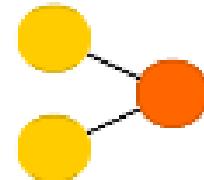
Das einfachste und älteste neuronale Netz. Es nimmt die Eingabeparameter, addiert diese, wendet die Aktivierungsfunktion an und schickt das Ergebnis an die Ausgabeschicht.

Das Ergebnis ist binär, also entweder 0 oder 1 und damit vergleichbar mit einer Ja- oder Nein-Entscheidung. Die Entscheidung erfolgt, indem man den Wert der Aktivierungsfunktion mit einem Schwellwert vergleicht.

Bei Überschreitung des Schwellwertes, wird dem Ergebnis eine 1 zugeordnet, hingegen 0 wenn der Schwellwert unterschritten wird.

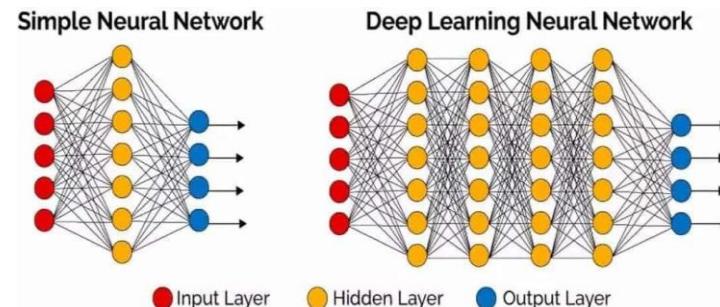
Typische Anwendungen sind die Analyse von Messdaten oder die Bilderkennung.

Perceptron (P)



Deep Learning

Neuronale Netze mit mehr als zwei Hidden Layers werden als Deep Neural Networks bezeichnet.



<https://datasolut.com/was-ist-deep-learning/>

Deep Learning kommt dann zum Einsatz, wenn andere maschinelle Lernverfahren an Grenzen stoßen und wenn auf Feature Engineering verzichtet werden muss. NN können über mehrere Schichten viele Eingabe-Dimensionen von selbst auf die Features reduzieren, die für den korrekten Output notwendig sind.

Um Neuronale Netze trainieren zu können, braucht man extrem viele Daten und eine hohe Rechenleistung. Dramatischer Preisverfall der Hardware & extremer Anstieg der Rechenleistung unterstützen daher den verbreiteten Einsatz.

Feed forward neural networks

Sie zeichnen sich dadurch aus, dass die Schichten lediglich mit der nächst höheren Schicht verbunden sind.

Es gibt keine zurückgerichteten Berechnungen.

Der Trainingsprozess eines Feed Forward Neural Network (FF) läuft dann in der Regel so ab:

- alle Knoten sind verbunden
- Aktivierung läuft von Eingangs- zu Ausgangsschicht
- mindestens eine Schicht (Layer) zwischen Eingangs- und Ausgangsschicht
- Wenn besonders viele Schichten zwischen Eingangs- und Ausgangsschicht sind, spricht man von „Deep Feed Forward Neural Networks“

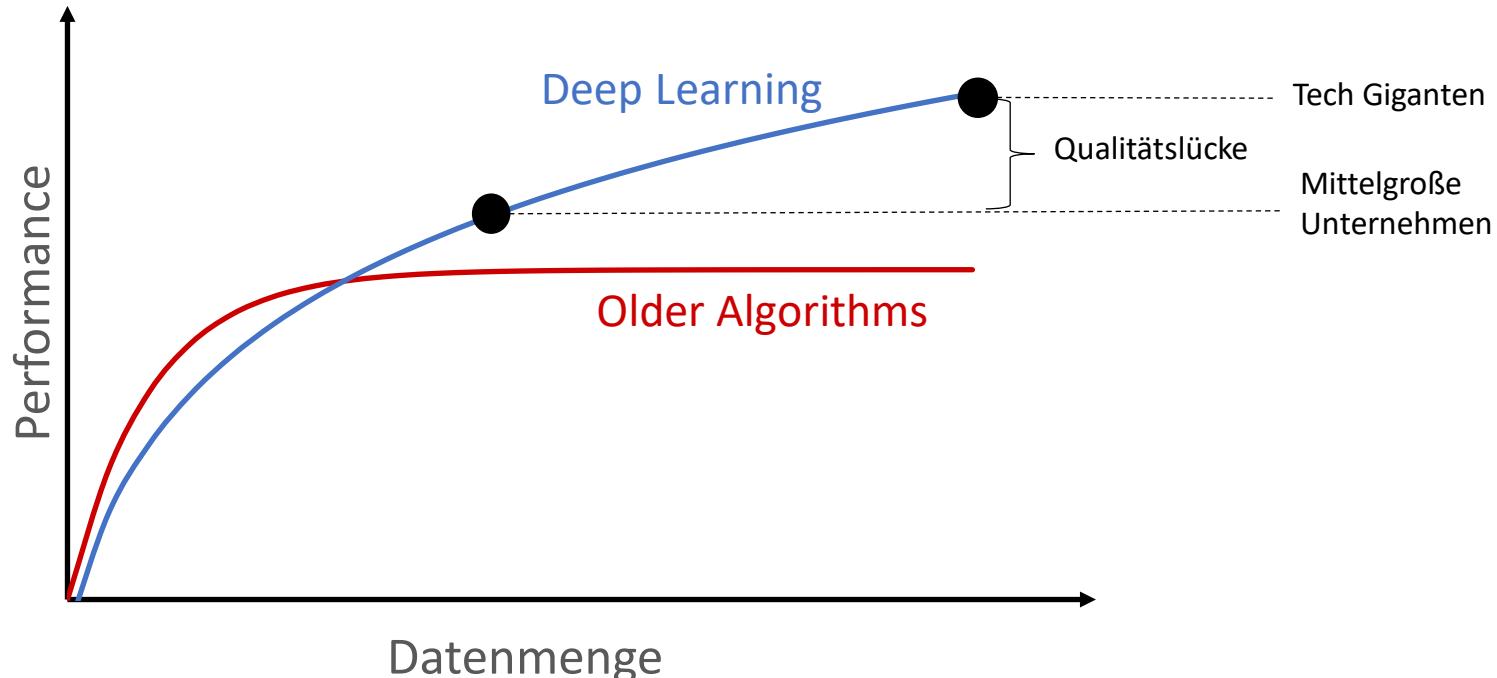
Feed Forward (FF)



Deep Feed Forward (DFF)



Deep Learning



In Analogie <https://www.slideshare.net/ExtractConf>

Wie funktioniert Muster- und Bilderkennung?

Convolutional Neural Network

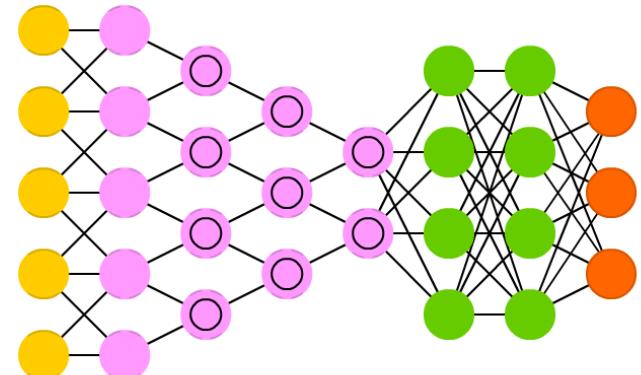
Das Convolutional Neural Network (zu Deutsch in etwa faltendes neuronales Netzwerk) wird insbesondere im Bereich der Bild- und Audioverarbeitung häufig eingesetzt.

Üblicherweise besteht ein solches Convolutional Neural Network aus mindestens 5 Schichten.

Innerhalb jeder dieser Schichten wird eine Mustererkennung durchgeführt.

Jede Schicht präzisiert dabei die Mustererkennung auf Basis des Outputs der vorherigen Schicht.

Deep Convolutional Network (DCN)

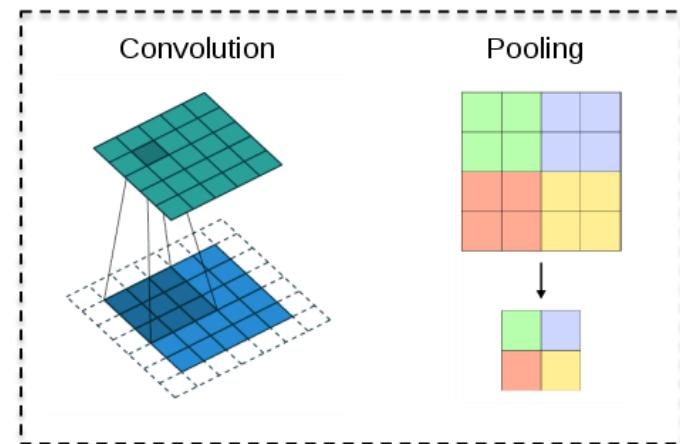


- Input Cell
- Kernel
- Convolution or Pool
- Output Cell

Convolutional Neural Network (CNN)

Convolution:

- Unterteilt das Bild in kleine Ausschnitte
 - Lernt die Beziehung der Pixel zu einander
- detektiert Merkmale wie bspw. Kanten
- Hidden Units werden nur mit einem Bruchteil der Input Units verbunden



Pooling:

- Reduziert die Anzahl der Parameter bei Großen Bildern wie beim „Subsampling“
- Reduktion der Dimensionen aber beibehalten der Informationen

Convolution

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1

FILTER / KERNEL 3x3

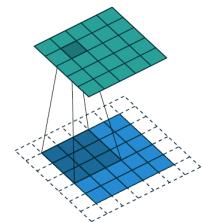
BILD 5x5x1



1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0



1		
2		
3		
4		
5		
6		
7		
8		
9		



1							
1	0	1	1	1	0	0	0
0	0	1	1	0	1	0	1
0	0	0	1	1	1	0	1
0	0	1	0	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	0	0	0	0	0
0	1	1	0	0	0	0	0

⋮

9							
1	1	1	0	0			
0	1	1	1	0	0		
0	0	1	1	0	1	1	
0	0	1	0	1	1	0	0
0	1	1	1	0	0	0	1
0	1	1	0	0	0	0	0
0	1	1	0	0	0	0	0

4	3	4
2	4	3
2	3	4

FEATURE MAP

4	3	4
2	4	3
2	3	4

Filter oder „Kernel“

Im Rahmen der Convolution können mit Hilfe unterschiedlicher Filter oder Kernel

- Kanten erkannt werden
- das Bild geschärft
- oder verschwommen

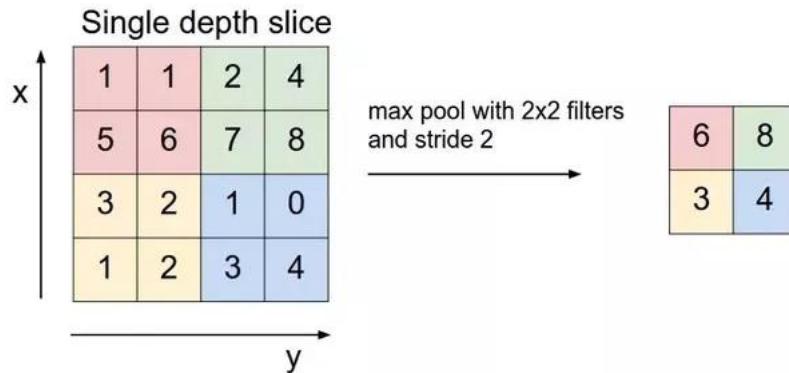
werden.

Es können größere Schritte bei der Convolution angewandt werden. Stride =2 überspringt 2 Pixel beim unterteilen des Bilds.

	Operation	Filter	Convolved Image
Identity		$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection		$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
		$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen		$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)		$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

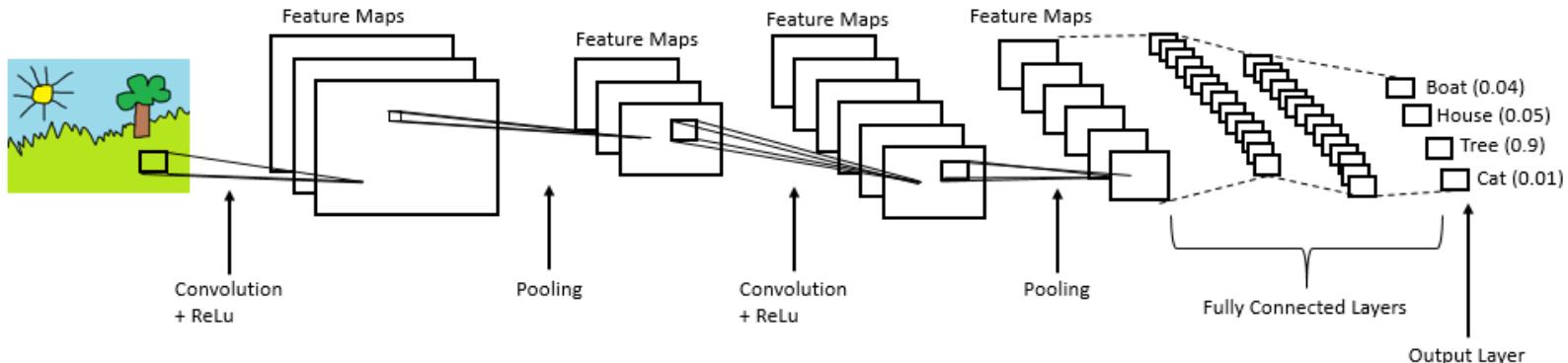
Pooling

- Verschiedene Vorgehensweisen:
- Max Pooling: beibehalten des größten Elements
- Average Pooling:
- Sum Pooling: Summe aller Elemente der Feature Map



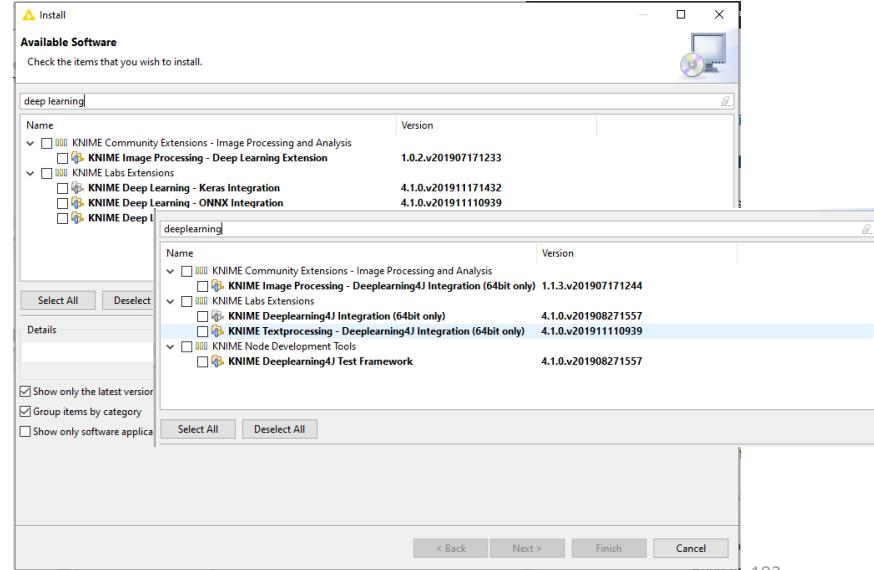
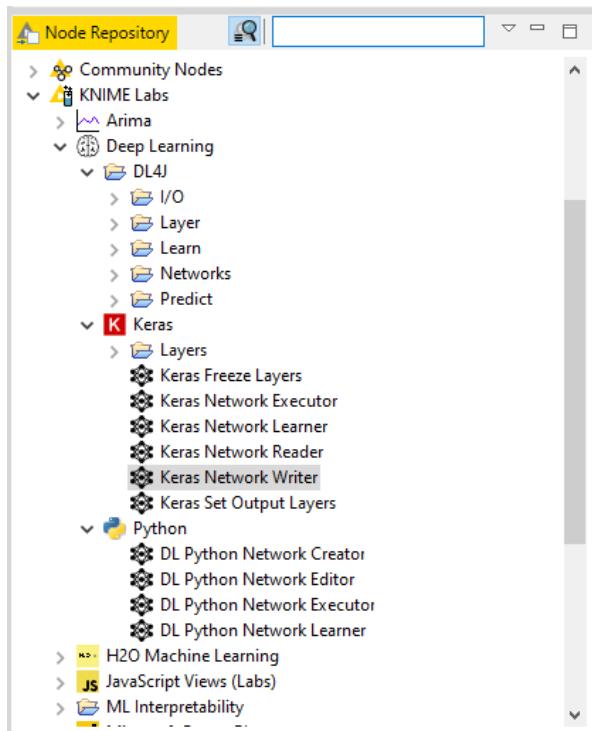
CNN - Gesamtablauf

- Bild als Input
- Parameter: Filter, Stride, Padding
- Ausführung Convolution
- Ausführung Pooling
- Weitere Layer: Convolution & Pooling
- Übergabe an „Fully Connected Layer“
- Ausgabe der Klasse

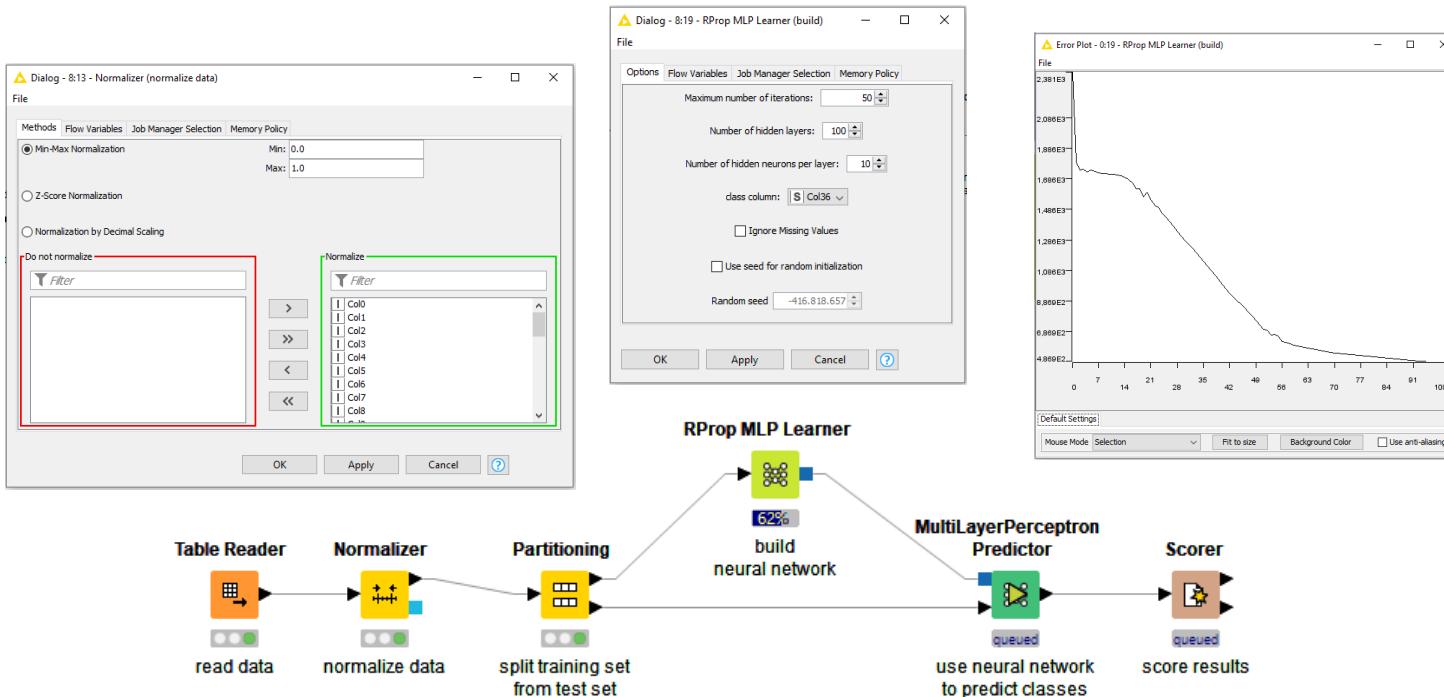


Neuronales Netz Nodes

Die KNIME Neural Network Nodes erfordern die Installation von Erweiterungspaketen:



Beispiel Workflow: Neuronale Netze





Übung Neuronale Netze

Modelle bewerten

Was sind Baselines und wozu werden sie eingesetzt?

Baselines

- Für eine Baseline wird in der Regel nur 1/10 der Bearbeitungszeit benötigt und es können bereits bis zu 90 % der Ergebnisse gezeigt werden
- Baselines helfen komplexe Zusammenhänge auf wenige wichtige Elemente zu reduzieren und so verständlicher zu machen.
- Baseline dienen dazu, um die Leistungsfähigkeit von Modellen zu bewerten.

Baselinetypen:

1. Minimale Genauigkeit, die ein Modell erreichen muss, um beispielsweise besser als eine zufällige Bestimmung zu sein (Beispiel Klassifikation mit 2 Werten; Zufall wäre 1:1)
2. Was ein Mensch ohne Computerunterstützung berechnen könnte. Sicherlich gibt es da Unterschiede, aber es dient zumindest als Richtwert.
3. Welche minimale Leistung ist erforderlich, um beispielsweise bei Budgetvorhersagen einen Mehrwert zu schaffen?

Erstellen von Baselines

Klassifizierung:

Bei Klassifizierungen können die Klassen, die am häufigsten vorkommen als Ergebnis für alle Vorhersagen verwendet werden. Berechnet man den Anteil der Klasse an der Gesamtzahl aller Klassen, erhält man die Genauigkeit, die das Modell überbieten muss.

Beispiel: in 136 von 195 Staaten herrscht Rechtsverkehr auf der Straße. Nimmt man dies als Vorhersage für alle Staaten, so hat man eine Genauigkeit von 70%.

Regression:

Bei Regressionen ist häufig eine gute erste Annäherung der Mittelwert oder der Median. Die Genauigkeit kann dann mit dem Bestimmtheitsmaß (R^2) überprüft werden.

Beispiel: Vorgestern lagen die Höchstwerte bei 18°C , gestern bei 20°C und heute bei 19°C . Als erste Annäherung wäre die Vorhersage für morgen der Mittelwert von 19°C . (Mal sehen wie viel besser die Wettervorhersage liegt 😊)

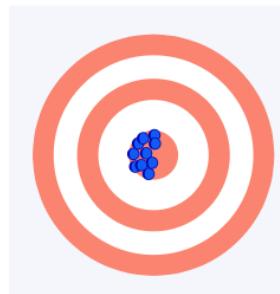
Was ist der Unterschied zwischen Genauigkeit und Präzision/Zuverlässigkeit?

Genauigkeit und Präzision/Zuverlässigkeit

Die Genauigkeit ist ein Maß für die Nähe von Messungen zu einem bestimmten Wert (dem Zielwert).

Präzision/Zuverlässigkeit ist ein Maß für die Nähe der Messungen zueinander aber nicht notwendiger Weise zum Zielwert.

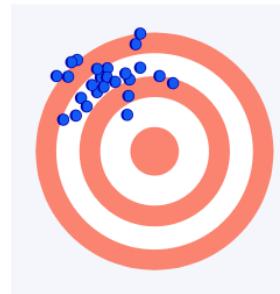
Die beiden Konzepte sind bestehen voneinander unabhängig, was bedeutet, dass ein Datensatz sowohl genau, präzise, präzise und genau oder keins von beides sein kann.



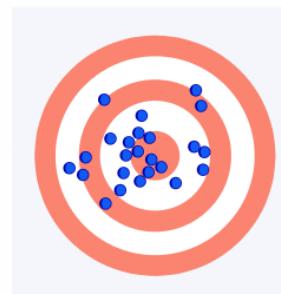
A: genau und präzise



B: präzise, aber nicht genau



C: weder genau noch präzise

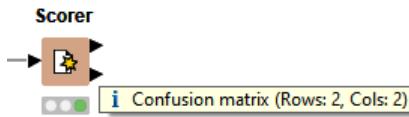


D: genau, aber nicht präzise

Wie lassen sich Vorhersagen von Klassifikationen bewerten?

Confusion Matrix (Wahrheitsmatrix)

Der Knoten Scorer bewertet die Ergebnisse und erstellt eine Confusion Matrix

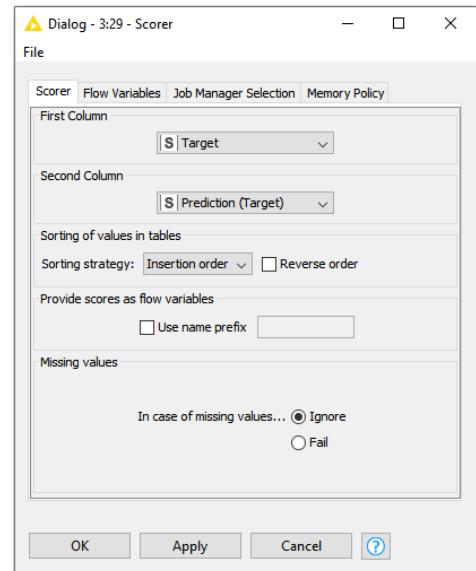


A dialog box titled 'Confusion matrix - 3:29 - Scorer' showing the same confusion matrix data as the node configuration.

Row ID	1	0
1	3250	1316
0	1721	2845

A dialog box titled 'Accuracy statistics - 3:29 - Scorer' showing various performance metrics. The table includes columns for True Positive, False Positive, True Negative, False Negative, Recall, Precision, Sensitivity, Specificity, F-measure, Accuracy, and Cohen's Kappa. The 'Overall' row shows values in red, indicating lower performance.

Row ID	TruePos...	FalsePo...	TrueNeg...	FalseNeg...	Recall	Precision	Sensitivity	Specificity	F-meas...	Accuracy	Cohen's...
1	3250	1721	2845	1316	0.712	0.654	0.712	0.623	0.682	?	?
0	2845	1316	3250	1721	0.623	0.684	0.623	0.712	0.652	?	?
Overall	?	?	?	?	?	?	?	?	?	0.667	0.335



Metriken für Klassifikationen

Genauigkeit

Die Genauigkeit ist wahrscheinlich die am ehesten nachvollziehbare Metrik und stellt die Anzahl der richtigen Klassifizierungen allen zu tätigen Klassifizierungen gegenüber.

$$\text{Genauigkeit} = \frac{\# \text{ richtig positiv} + \# \text{ richtig negativ}}{\# \text{ aller Klassifizierungen}}$$

		Actual = Yes	Actual = No
Predicted = Yes	TP	FP	
	FN	TN	
Actual = Yes			

Accuracy statistics - 3:29 - Scorer

File Hilit Navigation View

Table "default" - Rows: 3 Spec - Columns: 11 Properties Flow Variables

Row ID	TruePositives	FalsePositives	TrueNegatives	FalseNegatives	Recall	Precision	Sensitivity	Specificity	F-measure	Accuracy	Cohen's kappa
1	3250	1721	2845	1316	0.712	0.654	0.712	0.623	0.682	?	?
0	2845	1316	3250	1721	0.623	0.684	0.623	0.712	0.652	?	?
Overall	?	?	?	?	?	?	?	?	?	0.667	0.335

Warum ist die Betrachtung der Genauigkeit manchmal nicht ausreichend?

Metriken für Klassifikationen

Recall (Sensitivität)

Der Recall-Wert beschreibt das Verhältnis der korrekt positiv klassifizierten Werte zu allen Werten, bei denen das Ereignis eingetreten ist:

$$\text{Recall} = \frac{\text{\# richtig positiv}}{\text{\# richtig positiv} + \text{\# falsch negativ}}$$

Spezifität

Ähnlich wie der Recall, bezieht sich jedoch auf die negativen Klassifizierungen, also der richtig negativ klassifizierten Werte zu allen Werten, bei denen das Ereignis nicht eingetreten ist:

$$\text{Spezifität} = \frac{\text{\# richtig negativ}}{\text{\# richtig negativ} + \text{\# falsch positiv}}$$

		Actual = Yes	Actual = No
Predicted = Yes	TP	FP	
	FN	TN	
Predicted = No			

Accuracy statistics - 3:29 - Scorer

File Hilite Navigation View

Table "default" - Rows: 3 Spec - Columns: 11 Properties Flow Variables

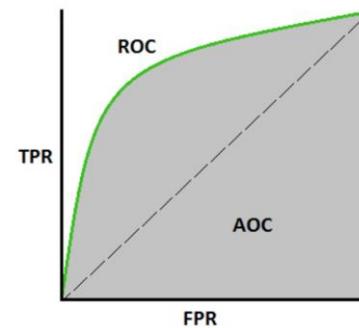
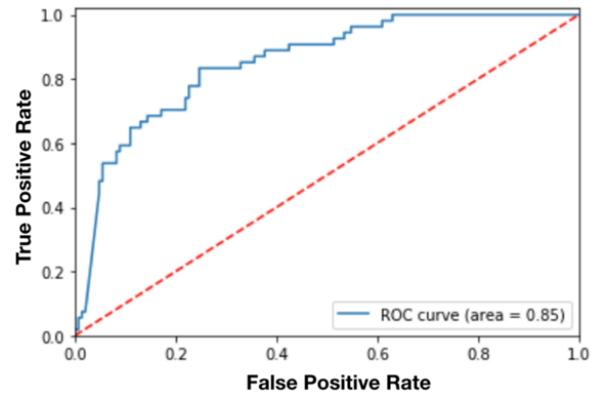
Row ID	TruePo...	FalsePo...	TrueNe...	FalseNe...	D Recall	D Precision	D Sensitivity	D Specificity	D F-meas...	D Accuracy	D Cohen'...
1	3250	1721	2845	1316	0.712	.654	0.712	.623	.682	?	?
0	2845	1316	3250	1721	0.623	.684	0.623	.712	.652	?	?
Overall	?	?	?	?	?		?			0.667	0.335

Klassifizierungen grafisch bewerten: ROC-Kurve

Mit der ROC-Kurve (Receiver Operating Characteristic) werden die richtig und falsch positiven Werte in absteigender Reihenfolge ihrer Vorhersagewahrscheinlichkeit aufgetragen und stellen somit die Güte eines Modells dar. Richtig positive Werte verschieben die Kurve in Richtung Y- Achse, falsch positive in Richtung x-Achse.

Die Fläche unter der Kurve (AUC) ist ein aggregiertes Maß für die Leistung eines binären Klassifikators für alle möglichen Schwellenwerte.

Die AUC berechnet die Fläche unter der ROC-Kurve und liegt daher zwischen 0 und 1. Ist der Wert gleich 1 handelt es sich um ein perfektes Modell, bei 0 um ein komplett falsches Modell.



Wie lassen sich Vorhersagen von Regressionen bewerten?

Metriken für Regressionen

MSE

Der Mittlere quadratischer Fehler (MSE) findet im Wesentlichen den durchschnittlichen quadratischen Fehler zwischen den vorhergesagten und den tatsächlichen Werten.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

MAE

Der mittlere absolute Fehler ist eine weitere Metrik, die den durchschnittlichen absoluten Abstand zwischen den vorhergesagten und den Zielwerten ermittelt.

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

Der MAE ist gegenüber Ausreißern in der Regel robuster als der MSE. Der Hauptgrund ist, dass beim MSE durch Quadrieren die Fehler der Ausreißer (die normalerweise höhere Fehler aufweisen als andere Stichproben) mehr Gewichtung im resultierenden berechneten Fehler erhalten und so die Modellparameter beeinflussen.

Metriken für Regressionen

Modellgenauigkeit: R²

Um festzustellen, wie gut das Modell den Zusammenhang zwischen den Variablen abbildet hat, wird der Wert R² errechnet, mit dem sich Lineare Regressionsmodelle vergleichen lassen.

Dafür werden die Varianzen der Regressionsgeraden mit den Varianzen der Mittelwerte der Daten anhand folgender Formel verglichen und stellt einen dimensionslosen zwischen 0 und 1 dar, der aussagt, wie gut das Modell die abhängige Variable vorhersagen kann:

$$R^2 = \frac{\text{Varianz (Mittelwert)} - \text{Varianz (Regressionsgerade)}}{\text{Varianz (Mittelwert)}}$$

Ist R² = 1 liegen alle Punkte auf einer Linie und unser Modell kann zu 100% genau die Datenpunkte vorhersagen. Ist R² = 0, dann hat das Modell keine Aussagekraft und die beiden Variablen hängen wahrscheinlich nicht zusammen.

Wie lassen sich Vorhersagen von Regressionen bewerten?

Metriken für Cluster-Analysen

Rand-Index

Der Rand-Index (benannt nach William M. Rand) ist in der Cluster-Analyse ein Maß für die Ähnlichkeit zwischen zwei Cluster-Einteilungen. Der Rand-Index ist mathematisch ähnlich der Berechnung der Genauigkeit für Klassifizierungen, kann aber auch dann angewendet werden, wenn keine Klassenbezeichnungen vorhanden sind.

$$R = \frac{a + b}{a + b + c + d}$$

a = Anzahl von Datenpunkten, die sich im **gleichen** Cluster in Modell 1 und im **gleichen** Cluster in Modell 2 befinden.

b = Anzahl von Datenpunkten, die sich in **unterschiedlichen** Clustern in Modell 1 und in **unterschiedlichen** Clustern in Modell 2 befinden.

c = Anzahl von Datenpunkten, die sich im **gleichen** Cluster in Modell 1 und in **unterschiedlichen** Clustern in Modell 2 befinden.

d = Anzahl von Datenpunkten, die sich in **unterschiedlichen** Clustern in Modell 1 und im **gleichen** Cluster in Modell 2 befinden.

Der Rand-Index gibt Werte zwischen 0 und 1 aus. Dabei sind Cluster-Einteilungen mit 1 identisch und 0 absolut verschieden.



Übung Modelle bewerten

Modelle optimieren

Wie hängen Verzerrung und Varianz mit
Genauigkeit und Präzision zusammen?

Verzerrung-Varianz-Dilemma

Verzerrung = Fehler ausgehend von falschen Annahmen im Lernalgorithmus

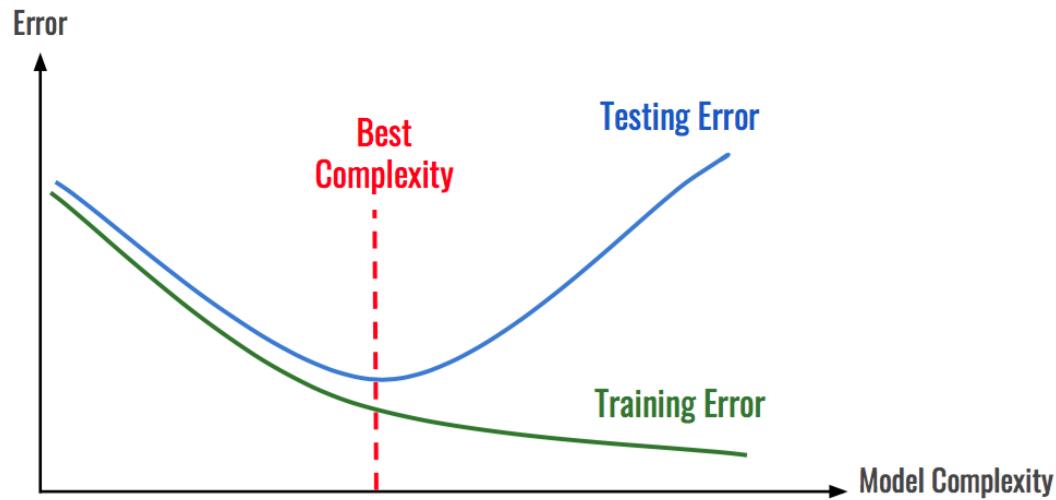
Algorithmus modelliert nicht die entsprechenden Beziehungen zwischen Eingabe und Ausgabe → **Genauigkeit**

Varianz = Fehler ausgehend von der Empfindlichkeit auf Änderungen in den Trainingsdaten

Trainingsdaten werden auswendig gelernt anstatt der Modellierung der vorgesehenen Ausgabe → **Präzision**

Verzerrung-Varianz-Dilemma

- Testdaten geben Auskunft über Generalisierbarkeit des Modells
- Test Error soll im Bereich des Training Errors liegen!



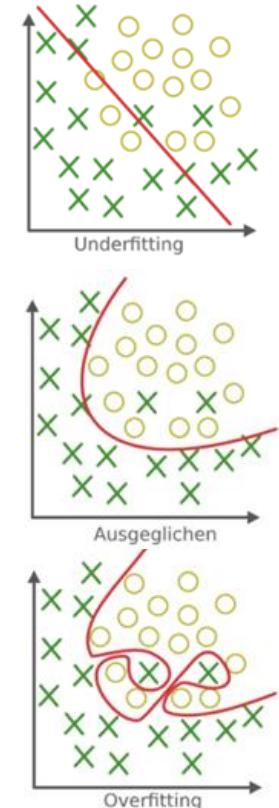
Over – und Underfitting

Bei der Optimierung von Modellen liegt häufig der Fokus nur auf der Minimierung der Fehler bzw. Abweichungen (z.B. Genauigkeit oder MSE) . Dabei wird das Modell sehr präzise an die Trainingsdaten angepasst. Wird es jedoch auf unbekannte Daten angewendet, ist die Leistung dann aber schlecht.

Dieses Phänomen ist als **Overfitting** (Überanpassung) bekannt. Es tritt auf, wenn ein Modell zu eng an die Trainingsdaten angepasst wird.

Underfitting (Unteranpassung) könnte man dagegen als Lernschwäche oder Lernstörung bezeichnen. Der Algorithmus kann aus den vorhandenen Daten keine relevanten Schlüsse ziehen um später Aussagen über neue Daten treffen zu können. Die anfänglichen Trainingsdaten sind also zu schwach, um daraus gültiges Wissen ableiten zu können.

Underfitting liegt in der Regel vor, wenn der Trainingsfehler hoch ist, das Modell also nicht in der Lage ist einen ausreichend niedrigen Fehler aus dem Trainingssatz zu erhalten.



Wie kann Overfitting eines Modells entdeckt werden?

Kreuzvalidierung

Beim Training eines Modells ist es wichtig, es nicht mit komplexen Algorithmen zu sehr (Overfitting) oder mit einfachen Algorithmen zu wenig (Underfitting) anzupassen.

Die Wahl des Trainings- und Testsatzes ist entscheidend für die Reduzierung dieses Risikos.

Es ist jedoch schwierig, den Datensatz so aufzuteilen, dass das Lernen und die Gültigkeit der Testergebnisse maximiert werden.

Hier kommt die Kreuzvalidierung zum Einsatz.

Einer der häufigsten verwendeten Techniken zur Kreuzvalidierung von Algorithmen ist die N-fache Kreuzvalidierung

N-fache Kreuzvalidierung

Bei der N-fachen Kreuzvalidierung werden Daten zufällig in N Blöcke (= Folds) aufgeteilt. Ein Block wird zum Testen der Daten verwendet und die restlichen Blöcke zum Trainieren.

Der Unterschied zur herkömmlich Aufteilung ist, dass danach der ganze Prozess mit einem anderen Block als Testdaten und den übrigen Blöcken als Trainingsdaten wiederholt wird. Das ganze Prozedere wiederholt sich N mal.

Am Ende war jeder Block einmal für die Testdaten zuständig und daraus wird ein Durchschnitt berechnet.

Damit wird das Risiko reduziert, zufällig nicht-repräsentative Testdaten auszuwählen. Durch das Wiederholen wird das Ergebnis robuster, da die Varianz sinkt.

4-fold validation (k=4)



Wie kann die Verzerrung eines Modells vermieden werden?

Unausgeglichenene Datensätze

Eine weitere große Herausforderung im Bereich Machine-Learning stellt der Umgang mit unausgewogenen (engl. unbalanced) Datensätzen bei der Klassifizierung dar.

Man bezeichnet einen Datensatz als unausgewogen, wenn eine Klasse des Datensatzes gegenüber den anderen deutlich über-/unterrepräsentiert ist.

Beispiel Betrugsentdeckung

Ein großer Teil (meist mehr als 90%) des Datensatzes sind als „normales Verhalten“ und sehr wenig „Betrug“ klassifiziert werden.

Problem: Das Modell klassifiziert alles als „normales Verhalten“ ist dabei sehr genau.

Jedoch hilft dies nicht bei der Betrugsentdeckung

Besser funktioniert hier der Recall-Wert, der sich auf die positiven Klassifizierungen konzentriert.

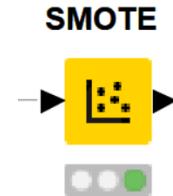
Aber auch die Trainingsdaten müssen für den Lernprozess so angepasst werden, damit sie beide Zustände ausreichend berücksichtigen.

Methoden zur Ausbalancierung eines Datensatzes

1. Over-Sampling der Minderheitsklasse

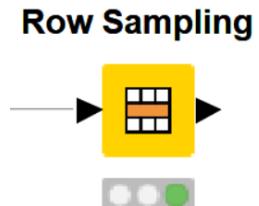
Eine häufig gebrauchte Methode ist SMOTE (Synthetic Minority Over-sampling Technique).

Hierbei werden neue Zeilen berechnet, indem zu den bestehenden Werten einer gegebenen Klasse mithilfe des nearest Neighbor Algorithmus weitere Werte der gleichen Klasse extrapoliert und hinzugefügt werden. Die Gefahr bei dieser Methode ist, dass durch das künstliche einfügen von Daten Artefakte geschaffen werden, die von den realen Daten abweichen.



2. UnderSampling der Mehrheitsklasse

Beim Undersampling werden die Daten aus der Mehrheitsklasse reduziert, d.h. Teile aus dem Datensatz entfernt. Dadurch wird das Verhältnis zwischen beiden Klassen ausgeglichen, jedoch gehen viele Informationen verloren.



3. Eine Kombination aus beiden Methoden

Was sind Parameter im ML und wie lassen Sie sich zum Optimieren eines Modells einsetzen?

Parameter im maschinellen Lernen

Modelle für maschinelles Lernen setzen sich aus zwei verschiedenen Arten von Parametern zusammen:

Hyperparameter sind alle Parameter, die vom Benutzer vor Trainingsbeginn beliebig eingestellt werden können (z.B. Anzahl der „Bäume“ im Random Forest oder Anzahl der Nachbarn in k-NN).

Modellparameter werden während des Modelltrainings gelernt (z.B. Gewichte in Neuronalen Netzen, Lineare Regression).

Die Modellparameter definieren, wie Eingabedaten verwendet werden, um die gewünschte Ausgabe zu erzielen, und werden beim Trainieren des Modells erlernt. Der äußere Einfluss ist hier nur begrenzt, die Zusammensetzung der Trainingsdaten ist hier entscheidend. Über die Hyperparameter lässt es sich dagegen steuern wie das Modell aufgebaut ist.

Methoden zur Hyperparameter Optimierung

Beim Optimieren von Machine Learning-Modellen stellt sich das Problem, dass häufig mehrere Parameter gleichzeitig verändert werden können, die sich auch wechselwirkend auf das Modell auswirken.

Um die richtige Kombination von Parametern finden gibt es verschiedene Ansätze zur Hyperparameter Optimierung:

- Manuelle Suche
- Zufällige Suche
- Rastersuche
- Automatisiertes Hyperparameter-Tuning

Methoden zur Hyperparameter Optimierung

Manuelle Suche

Bei der manuellen Suche werden einige Modell-Hyperparameter basierend auf Einschätzung und Erfahrung ausgewählt. Damit das Modell trainiert und auf Genauigkeit bewertet. Diese Schleife wird so lange wiederholt, bis ein zufriedenstellendes Ergebnis erzielt wird.

Diese Methode ist sinnvoll bei unkomplizierten Algorithmen und bei entsprechender Erfahrung des Data Analysts. Andernfalls kann es zu einer langwierigen Suche führen.

Zufällige Suche

Bei der Zufallssuche erstellen wird eine Matrix von zufällig zusammengestellten Hyperparametern zum Trainieren und Testen des Modells verwendet. Unterstützend kann die Modellstabilität durch Kreuzvalidierung getestet werden.

Diese Methode ist sehr zufallsabhängig, kann aber erste Zusammenhänge der Hyperparameter aufzeigen. Anschließend kann das Modell noch manuell verfeinert werden.

Methoden zur Hyperparameter Optimierung

Rastersuche

Bei der Rastersuche werden alle Hyperparameter in allen sinnvollen Kombinationen einer Matrix zusammengebracht und so für das Modell genutzt.

Diese Methode ist die gründlichste aber auch die zeitaufwendigste. Je nach Anzahl und Wertebereich der Hyperparameter können durchaus 100.000 und mehr Kombinationen getestet werden. Es ist wichtig, vorab nur die entscheidenden Hyperparameter auszuwählen und ihren Wertebereich und vor allem die Anzahl der Intervalle deutlich zu begrenzen

Automatisiertes Hyperparameter-Tuning

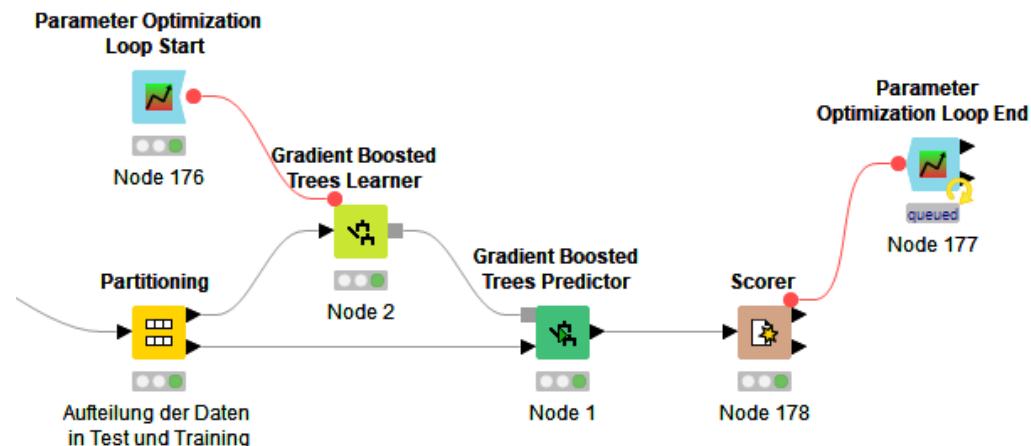
Beim automatisierten Hyperparameter Tuning werden Algorithmen eingesetzt, die die Suche nach der besten Kombination von Hyperparameter unterstützen. Dafür werden zum Beispiel Wahrscheinlichkeitsberechnungen (Bayessche Optimierung) genutzt oder es werden lokale Maxima durch Steigungsberechnung gesucht (hill climbing).

Diese Methode kann die Zeit zur Berechnung der besten Parameter deutlich reduzieren und liefert trotzdem sehr gute Ergebnisse.

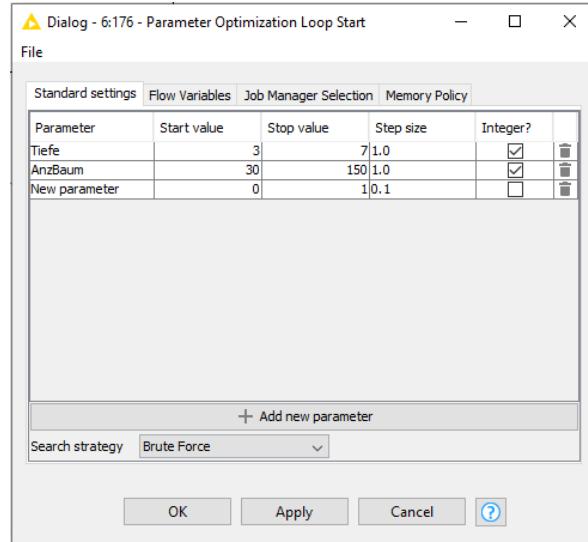
Hyperparameter Tuning / Parameter Optimization

Flow Variable!

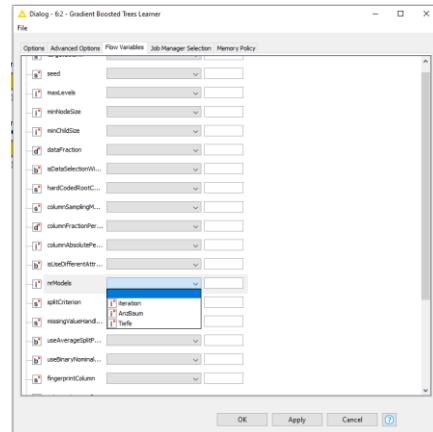
- Parameter definieren
- FV in der Model Konfiguration hinterlegen
- Ziel im Loop End definieren



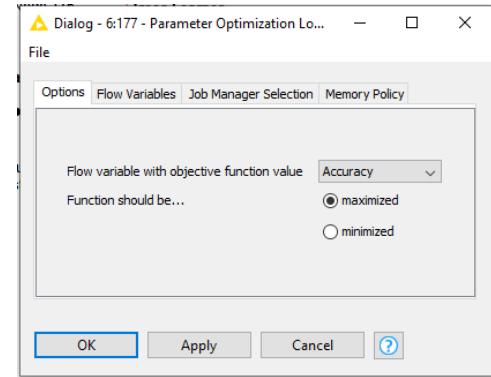
Hyperparameter Tuning / Parameter Optimization



Parameter definieren



Flow Variable in der
Model Konfiguration
hinterlegen



- Ziel im Loop End definieren



Übung Modelle optimieren