Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

2024 FS CAS PML
1 Feature Engineering
1.1 Einführung

Werner Dähler 2024

1 Feature Engineering - AGENDA

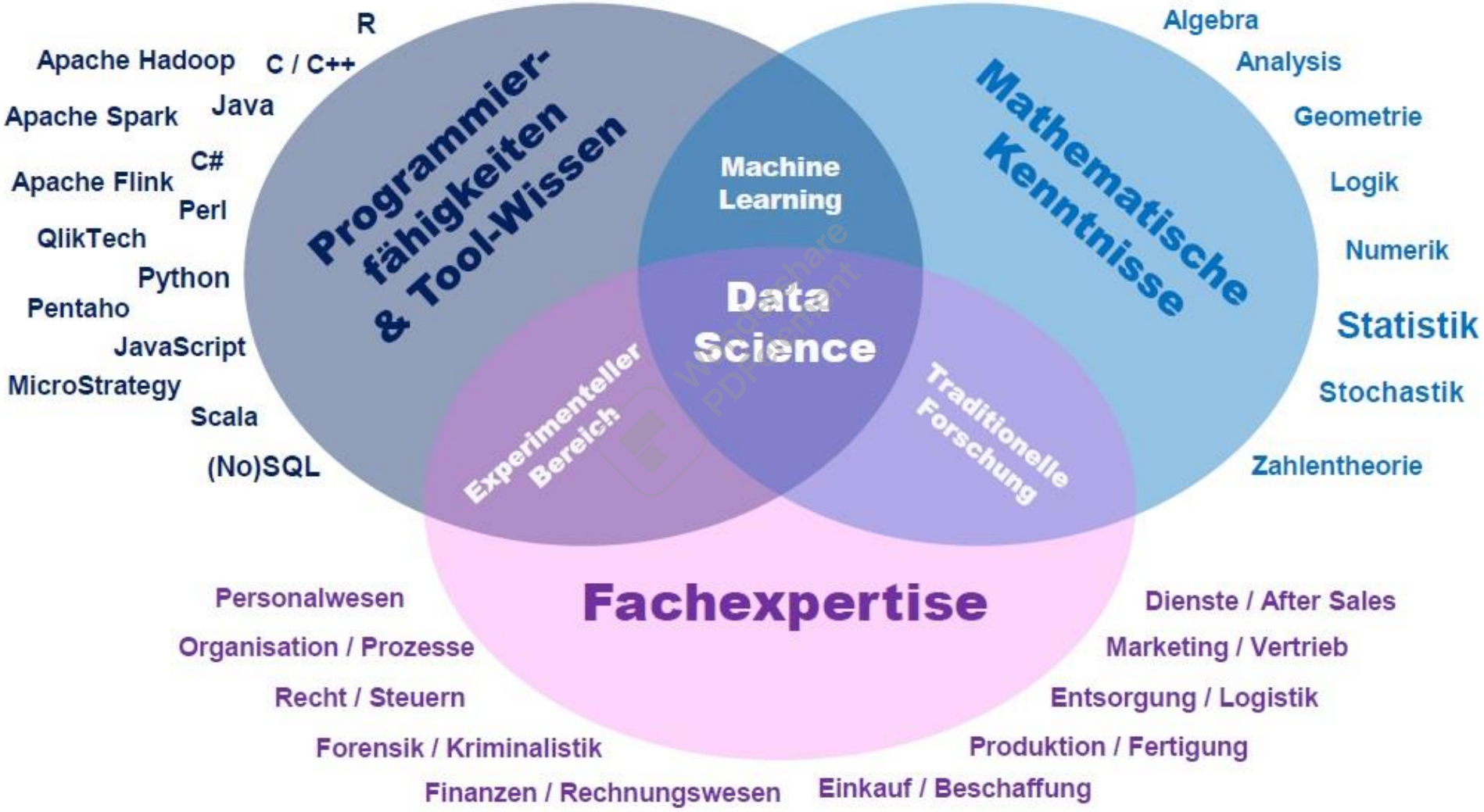
11. Einführung
12. Exploration
13. Transformation
14. Konstruktion
15. Selektion
16. Implementation
17. Nachträge

diese Einführung betrifft sowohl Feature Engineering wie Supervised Learning, da die beiden Tätigkeiten sehr eng ineinander verschränkt sind

die hinterlegten Links wurden am 25.05.2024 abgerufen

1.1 Feature Engineering - Einführung

1.1.1 Abgrenzungen



[Quelle]

1.1 Feature Engineering - Einführung

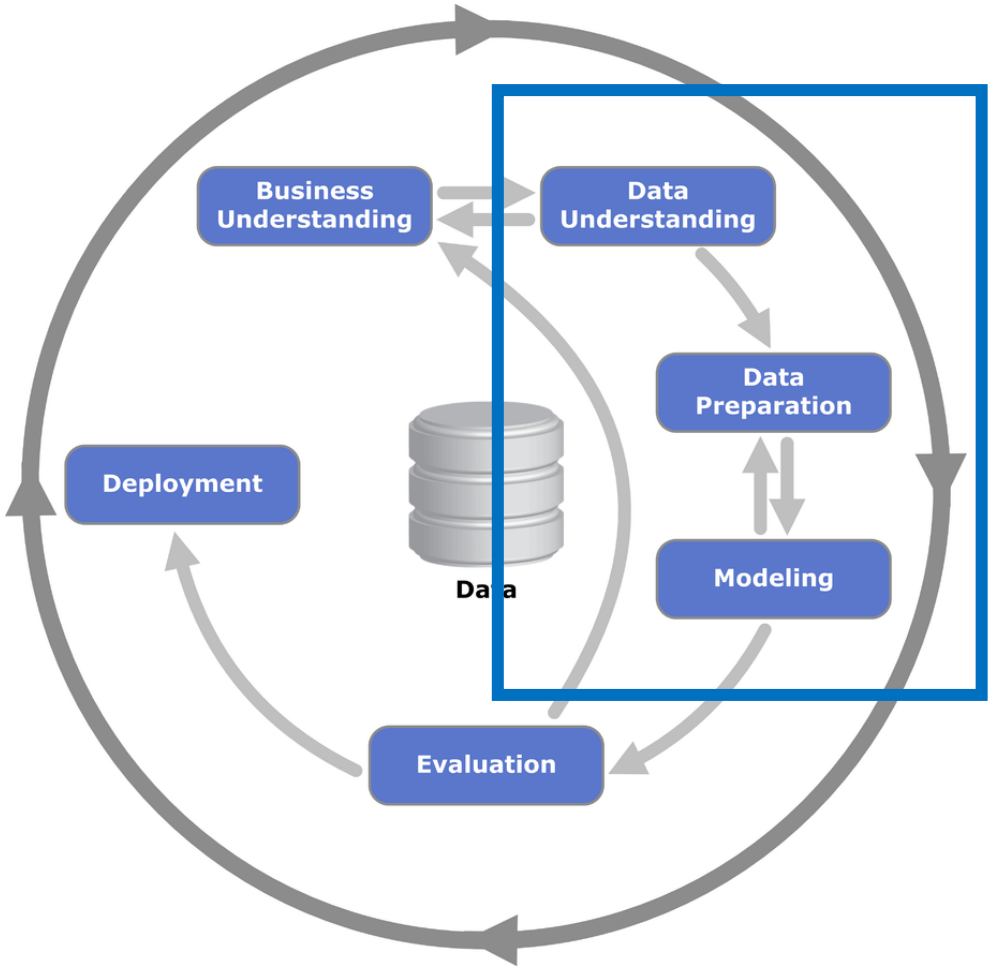
1.1.2 CRISP und die Gliederung des Kurses

Cross-industry standard process for data mining

hier hervorgehoben:

Phasen, die in diesem Kurs im Fokus stehen:

- Business Understanding
- Data Understanding
- Data Preparation
- Modeling
- Evaluation
- Deployment



- (der Prozess hiess ursprünglich CRISP-DM, inzwischen hat sich aber die Bezeichnung CRISP-ML ebenfalls etabliert)

1.1 Feature Engineering - Einführung

1.1.2 CRISP und die Gliederung des Kurses

die einzelnen Phasen:

- **Business Understanding**: (Project Understanding) Festlegung der Ziele und Anforderungen; Ableitung der konkreten Aufgabenstellung und der groben Vorgehensweise
- **Data Understanding**: Datensammlung bzw. erste Sichtung der zur Verfügung stehenden Daten; Ermittlung möglicher Probleme mit Datenqualität
- **Data Preparation**: Konstruktion des finalen Datensatzes für die Modellierung
- **Modeling**: Anwendung geeigneter Data Mining-Verfahren (!), Optimierung der Parameter; gewöhnlich Ermittlung mehrerer Modelle
- **Evaluation**: Auswahl des Modells, welches die Aufgabenstellung am besten erfüllt, sorgfältiger Abgleich mit der Aufgabenstellung.
- **Deployment**: Aufbereitung und Präsentation der Ergebnisse; evtl. Integration des Modells in einen Entscheidungsprozess des Auftraggebers (Operationalisieren)

  vgl. [Statistik Dresden]

1.1 Feature Engineering - Einführung

1.1.2 CRISP und die Gliederung des Kurses

- Machine Learning stellt recht restriktive Anforderungen an die Daten:
  - tabellarische Struktur (vgl. Kap. 1.1.3)
  - nur numerische Daten, keine Missing Values, etc.

  mehr dazu unter 1.1.6
- vor diesem Hintergrund können die Tätigkeiten wie folgt gegliedert werden
  - sichten der Daten in Hinblick auf Machine Learning
  - aufbereiten der Daten für Machine Learning
  - Modellieren

1.1 Feature Engineering - Einführung

1.1.2 CRISP und die Gliederung des Kurses

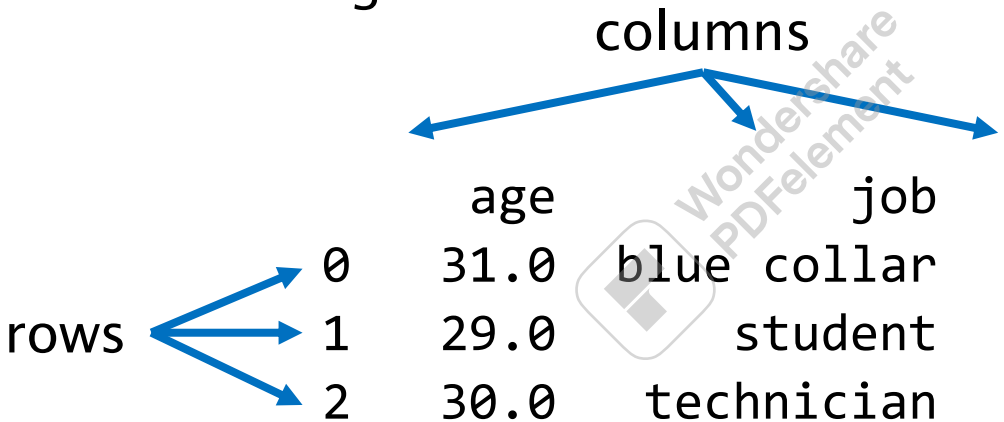
- daraus erfolgt für den gesamten ML-Prozess, und in Anlehnung an CRISP folgende Gliederung für diesen Kurs:

CRISP-Phase	Kapitel	Tätigkeiten / Ergebnisse
Data Understanding	Feature Exploration / EDA (Kap. 1.2)	sichten der Daten mit Fokus auf die Anforderungen für ML entdecken von Anomalien bedarfsweise in Rücksprache mit dem Fach Ergebnis: Empfehlungen für FE
Data Preparation	Feature Engineering FE (Kap. 1.3-1.7)	umsetzen der Empfehlungen aus EDA überprüfen von möglichen Alternativen Ergebnisse: - Umsetzungsscript - transformierte Daten
Modeling	Machine Learning ML (Kap. 2-5)	erstellen, testen und optimieren von ML Modellen Ergebnisse: - Erkenntnisse zu gefundenen Mustern (Unsup Learn) - trainierte Vorhersagemodelle (Sup Learn)

1.1.3 Strukturierte Daten

1.1.3.1 Strukturierte Daten - Aufbau und Organisation eines Data Frame

- Daten sind immer irgendwie strukturiert
- für den Einsatz von Methoden des Machine Learning wird aber ausschliesslich eine tabellarische Struktur vorausgesetzt:



	age	job	marital	duration	campaign	y
0	31.0	blue collar	single	70.0	2	no
1	29.0	student	single	670.0	1	yes
2	30.0	technician	single	229.0	3	yes
3	NaN	retired	married	288.0	3	yes
4	29.0	blue collar	married	658.0	1	yes
5	33.0	admin.	married	77.0	1	no
6	28.0	None	divorced	125.0	1	no
:						

1.1 Feature Engineering - Einführung

1.1.3.1 Strukturierte Daten - Aufbau und Organisation eines Data Frame

- Objekte / Beobachtungen sind in Zeilen (rows) angeordnet
- Merkmale / Attribute (Variablen, Features) sind in Spalten (columns) angeordnet
  - Spalten enthalten (idealerweise) sprechende Namen, über welche sie angesprochen werden können
  - pro Spalte ist ein Datentyp festgelegt, unterschiedliche Spalten können aber unterschiedliche Typen aufweisen
- analog einer Tabelle in einer Relationalen Datenbank
- pandas.DataFrame: der Objekttyp, mit welchem in Folgenden hauptsächlich gearbeitet wird

1.1 Feature Engineering - Einführung

1.1.3.1 Strukturierte Daten - Aufbau und Organisation eines Data Frame

- die für diesen Kurs vorbereiteten und abgegebenen Beispieldaten werden ausschliesslich im CSV-Format zur Verfügung gestellt
- die ersten Schritte sind daher
  - die Daten lokal hinterlegen (Filesystem)
  - den Datenpfad festlegen
  - Das Dataset (CSV) in ein (pandas.) DataFrame Objekt einlesen

- die hier vorgestellten Methoden beziehen die Daten danach ausschliesslich von geladenen DataFrame Objekten

1.1 Feature Engineering - Einführung

1.1.3.2 Strukturierte Daten - Charakterisierung der Spalten 1: Datentypen

- pandas, die bevorzugte Library zum Datenhandling, bietet Datentypen an, welche zumeist auf NumPy basieren, wie
  - `float`
  - `int`
  - `bool`
  - `string`
- für die Tätigkeiten in Feature Engineering und Machine Learning ist primär die folgende Unterscheidung relevant:
  - numerical (num: int und float)
  - not numerical (string, im Folgenden kategorial)
- bool wird intern als int dargestellt
  - 0 = `False`
  - 1 (und alle anderen) = `True`

1.1 Feature Engineering - Einführung

1.1.3.3 Strukturierte Daten - Charakterisierung der Spalten 2: Skalenniveaus (Skalierung)

der Begriff stammt aus der Statistik (Statistische Datenanalyse), wo mit folgenden Skalierungstypen (Skalenniveaus) gearbeitet wird:

Bezeichnung DA	Bezeichnung ML	Bedeutung	Dtype	Ops
nominal	kategorial (ev. nominal)	ungeordnete Gruppen	num/str	=,≠
ordinal	kategorial (ev. ordinal)	geordnete Gruppen	num/str	<,≤,≥,>
intervall	metrisch	Zahlen ohne def. Nullpunkt	num	+,-
ratio	metrisch	Zahlen mit def. Nullpunkt	num	*,/

- Beispiele
  - nominal: Geschlecht, Religion, Augenfarbe
  - ordinal: Grössenklasse, Einkommensklasse, Schulnoten (!)
  - intervall: Temperatur in Grad Celsius
  - ratio: Temperatur in Grad Kelvin

1.1 Feature Engineering - Einführung

1.1.3.3 Strukturierte Daten - Charakterisierung der Spalten 2: Skalenniveaus (Skalierung)

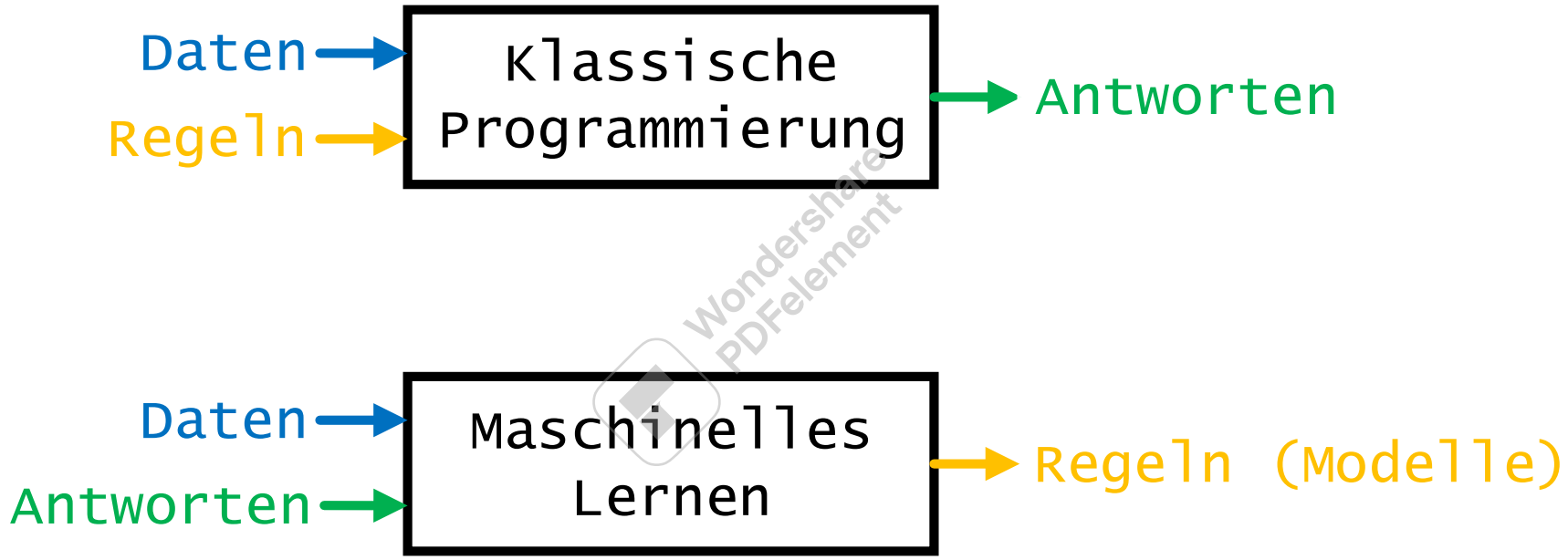
- die Unterscheidung zwischen intervall und ratio bei Metrischen Daten spielt im ML keine Rolle

- dagegen kann bei numerischen Variablen aufgrund der Verteilung zusätzlich folgende Differenzierung angebracht sein
  - stetig (meist Datentyp float)
  - diskret: (meist Datentyp int), verschiedene Interpretationsmöglichkeiten
    - Zählwerte
    - numerische Codierungen von Kategorialen Merkmalen (dies wäre im Zweifelsfall mit dem Fach abzuklären)

1.1 Feature Engineering - Einführung

1.1.4 Begriffe

1.1.4.1 Begriffe - Klassische Programmierung vs. Maschinelles Lernen



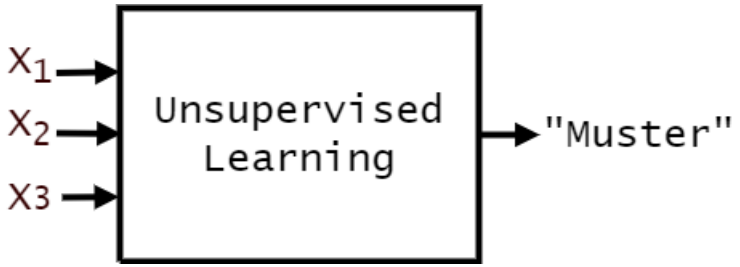
- "klassische" Programmierung und Maschinelles Lernen (hier Supervised Learning) arbeiten mit (mehr oder weniger) denselben Zutaten, aber mit anderer Anordnung derselben

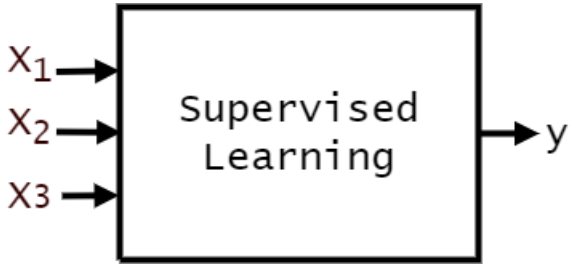
1.1 Feature Engineering - Einführung

1.1.4.2 Begriffe - Unüberwachtes vs. Überwachtes Lernen

folgende Unterscheidungen: (vgl. Einführung von Jürgen Vogel)

- Unüberwachtes Lernen (Unsupervised Learning):
  Lernen aus Daten
  - z.B. zum Erkennen von den Daten zugrundeliegenden Mustern
  - Thema des hier anschliessenden Kursteils (Unsup. Learn.)
- Überwachtes Lernen (Supervised Learning):
  Lernen aus Beispielen
  - Modellbildung zur Voraussage eines bestimmten Zielwertes (Label, Target, y) aus Daten für neue Daten ohne Label
  - **Prädiktive Modellierung**
  - Inhalt dieses Kursteils
- Bestärkendes Lernen (Reinforcement Learning): **Lernen aus Erfahrung**
  (dies hier nur der Vollständigkeit halber)





1.1 Feature Engineering - Einführung

1.1.4.2 Begriffe - Unüberwachtes vs. Überwachtes Lernen

oder: **Charakterisierung der Spalten 3: Rollen**

- Überwachtes Lernen setzt also im Gegensatz zu Unüberwachtem Lernen das Vorhandensein eines **Zielwertes** voraus, mit dessen Hilfe die Modelle trainiert werden
- die untenstehende Tabelle zeigt die wichtigsten Unterschiede der Bezeichnungen zwischen den Disziplinen Statistik und Machine Learning auf

Statistik / Datenanalyse	**Machine Learning**	**Kürzel**
Unabhängige Variable	**Feature**, Prädiktor, Attribut	X
Abhängige Variable	**Target**, Label, response, outcome	y

- die Bezeichnungen X (gross) und y (klein) sind angelehnt an Konventionen der Linearen Algebra, welche vielen der im folgenden vorgestellten Methoden zugrunde liegt
  - Matrizen werden mit Grossbuchstaben bezeichnet
  - Vektoren dagegen mit Kleinbuchstaben
- die meisten Methoden von scikit-learn erwarten die Features als Matrix, das Target dagegen als Vektor

1.1 Feature Engineering - Einführung

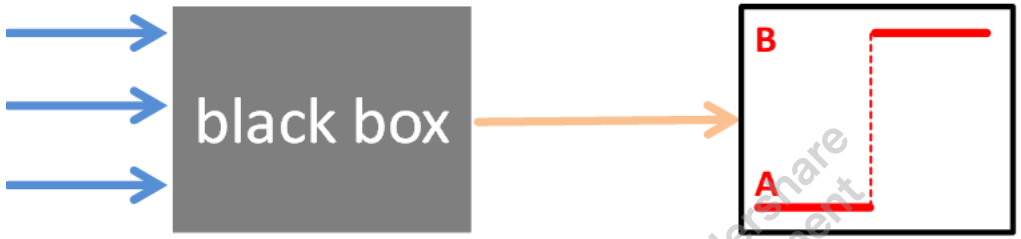
1.1.4.3 Begriffe - Colums vs. Variablen vs. Features

- die Unterscheidung insbesondere zwischen den Begriffen Columns, Variablen und Features wird in der Machine Learning Community unterschiedlich gehandhabt
- im Kontext dieses Kurses werden die Bezeichnungen wie folgt verwendet
  - **Column**: rein technische Bezeichnung einer Spalte in einem Data Frame
  - **Variable**: eine Spalte des Data Frames aus der Sicht von Data Analytics (EDA), resp. Feature Exploration
  - **Feature** (im weiteren Sinn, s.l.): eine Spalte des Data Frames aus Sicht von Machine Learning, das Ergebnis von Feature Engineering
    bei Überwachtem Lernen wird dann noch weiter unterschieden zwischen
    - Features (im engeren Sinn, s.str.)
    - Target (vgl. 1.1.4.2)

1.1.4.4 Begriffe - Klassifikation vs. Regression

unter Überwachtem Lernen werden die beiden folgenden Methodengruppen unterschieden:

- Klassifikation



Ausgangssignal kategoral

→ **Klassifikation**

- Regression

Ausgangssignal metrisch

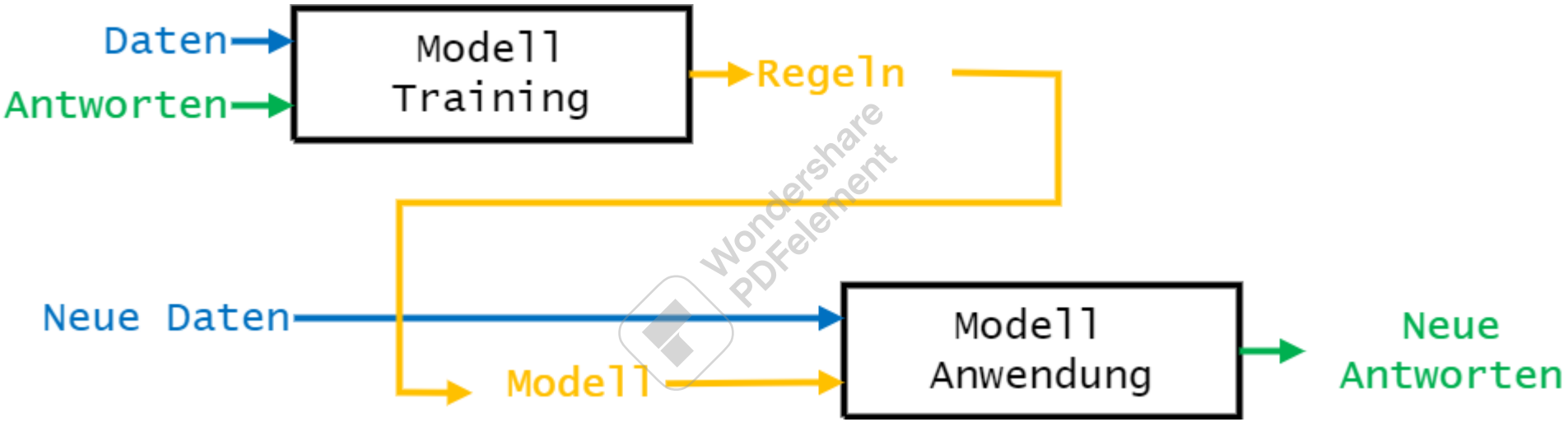
→ **Regression**

(kategorial / metrisch: vgl. Darstellung zu Skalenniveau, Kap. 1.1.3.3)

1.1 Feature Engineering - Einführung

1.1.4.5 Begriffe - Training vs. Anwendung

- und wozu das Ganze?



- das mittels der verschiedenen Tätigkeiten erstellte und optimierte Vorhersagemodell soll für neue Daten möglichst zuverlässige Vorhersagen zu den Werten des Targets

1.1 Feature Engineering - Einführung

1.1.4.5 Begriffe - Homonyme und Synonyme

- gilt nicht nur für Machine Learning:

> "*There are more things in Heaven and Earth, Horatio, than are dreamt of in your philosophy.*"
> [William Shakespeare, Hamlet]

- d.h. es existieren mehr Begriffe als uns dazu allgemein Bezeichnungen zur Verfügung stehen
- führt dazu, dass unterschiedliche Sachverhalte dieselbe Bezeichnung aufweisen können (was wir aushalten müssen)
- **Homonym**: dieselbe Bezeichnung für unterschiedliche Begriffe
  - z.B. "Klasse" (Überwachtes Lernen, Objektorientierte Programmierung)
    - Überwachtes Lernen: Kategorie des Targets
    - OOP: Objektdefinition
  - sollte aus dem jeweiligen Kontext erkennbar sein, andernfalls präzisieren ("im Sinne von...")

1.1.4.5 Begriffe - Homonyme und Synonyme

im Gegensatz dazu:

- **Synonym**: unterschiedliche Bezeichnungen für denselben Begriff
  - oft Domäne spezifisch
  - z.B. die Spalte in einem tabellarisch strukturierten Dataset
    - "column": technische Bezeichnung
    - "Variable": Verwendung in Statistik
    - "Feature": Verwendung in Machine Learning
    - "Attribute": z.T. verwendet in Data Mining

1.1 Feature Engineering - Einführung

1.1.5 Beispieldaten

Fallstudie 1: Bank Marketing Data Set
(für Klassifikation)

- stammt aus einer Marketing Campagne eines portugiesischen Bankinstituts
- das Ziel der Klassifikation ist es, vorherzusagen, ob sich ein Kunde für ein bestimmtes Anlageprodukt überzeugen lässt oder nicht
- im Original (im hinterlegten Link unter bank-additional-full.csv) besteht das Dataset aus 45211 rows und 21 columns
- aus praktischen Gründen wurde das Dataset für diesen Kurs etwas modifiziert
- dieses soll im Rahmen von Feature Engineering für das anschliessende Modellieren in Supervised Learning (Klassifikation) gemäss dessen Anforderungen aufbereitet werden
- Dateiname: bank_data.csv (auf Moodle)
- Target: "y"
- Delimiter: ","

1.1 Feature Engineering - Einführung

1.1.5 Beispieldaten

Fallstudie 2: Melbourne Housing Dataset

(für Regression)

- für das Kapitel 3 Überwaches Lernen - Regression wird eine andere Fallstudie eingesetzt: Immobiliendaten von Melbourne
- diese Daten wurden aus öffentlich zugänglichen Ergebnissen zusammengestellt, die jede Woche von Domain.com.au veröffentlicht werden
- der Datensatz enthält Adresse, Immobilientyp, Quartier, Verkaufsmethode, Anzahl Räume, Preis, Immobilienmakler, Verkaufsdatum und Entfernung von C.B.D. (Central Business District, Geschäftsviertel von Melbourne)
- Dateiname: melb_data.csv (auf Moodle)
- Target: "Price"
- Delimiter: ";"

1.1.5 Beispieldaten

Lehrbuchbeispiele

- viele Methoden werden mit idealisierten Datenbeispielen eingeführt
- man kann davon ausgehen, dass dort das Preprocessing (Feature Engineering) bereits erfolgt ist
- ausserdem sind sie derart optimiert, dass die entsprechenden Methoden verständlich visualisiert werden können
  - oft nur zwei Variablen
  - Target bei Klassifikationsbeispielen nur zwei Klassen
  - eine Modellvalidierung wird bei diesen Beispielen nicht vorgenommen, sie dienen primär dem Aufzeigen der jeweils zugrunde liegenden Lernalgorithmen

1.1 Feature Engineering - Einführung

1.1.5 Beispieldaten

Lehrbuchbeispiele

- die Daten werden entweder als CSV-Dateien abgegeben (vgl. Verteilung über Moodle)

```
import pandas as pd
demo_data_class = pd.read_csv('demo_data_class.csv')
demo_data_regr = pd.read_csv('demo_data_regr.csv')
```

- oder können direkt aus Packages geladen werden, z.B. aus seaborn:

```
import seaborn as sns
iris_data = sns.load_dataset('iris')
```

1.1 Feature Engineering - Einführung

1.1.6 Anforderungen an die Daten für Machine Learning

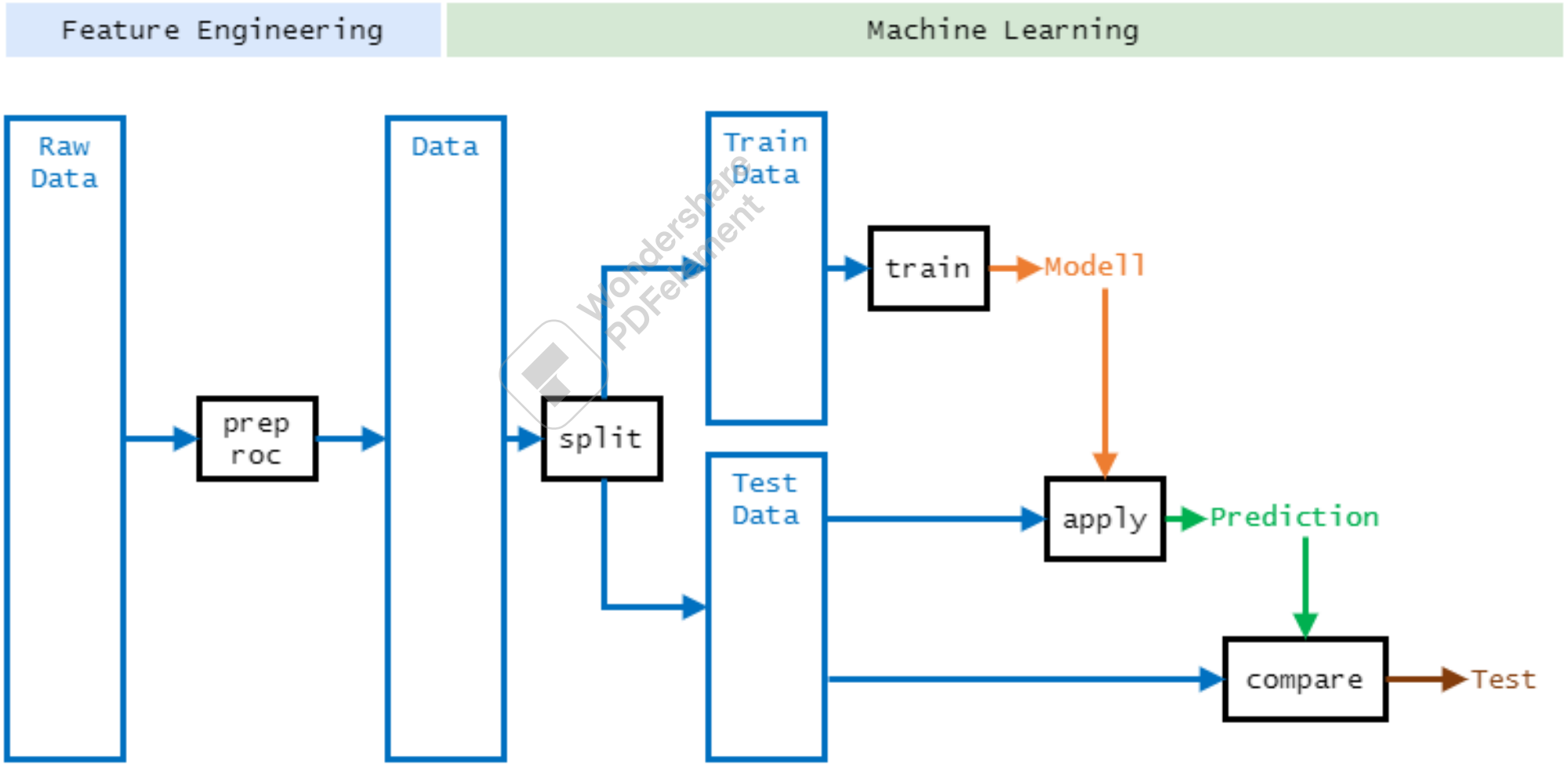
- das Trainieren von Vorhersagemodellen setzt bei den in diesem Kurs eingesetzten Methoden (scikit-learn) folgende Anforderungen an die Daten
- die wichtigsten:
  - alle Variablen (Features) müssen numerisch vorliegen
    - das Target darf dagegen auch nicht numerisch (kategorial) sein
  - Missing Values NAs werden nicht toleriert
  - die einzelnen Variablen müssen mindestens eine minimale Varianz aufweisen
  - vollständig lineare Zusammenhänge zwischen Variablen können problematisch sein und sollten vermieden werden

- Ziel des Feature Engineering wird daher sein, die genannten "Probleme" in den Rohdaten
  - zu entdecken
  - Alternativen zu deren Transformation zu erarbeiten ...
  - ... und entsprechend auszuführen

1.1 Feature Engineering - Einführung

1.1.7 Eine typische ML Sequenz

- ein kurzer Ausblick: der Ablauf gemäss einem häufig angewendeten Muster



1.1 Feature Engineering - Einführung

1.1.7 Eine typische ML Sequenz

dabei werden folgende Schritte durchlaufen:

- laden der Rohdaten
- **preprocess**: Aufbereiten der Rohdaten: vgl. Feature Engineering
- **split**: (Train - Test - Split) bereitstellen von Daten zum Trainieren und Testen, Details dazu in Kapitel 4
  - (hier für Holdout Validierung)
- **train / apply**: Modell trainieren mit Trainingsdaten und anwenden auf Testdaten, mehr dazu in Kapiteln 2 Klassifikation und 3 Regression
- **compare**: Modell bewerten, mehr dazu in Kapiteln 2, 3, 4

1.1 Feature Engineering - Einführung

1.1.7 Eine typische ML Sequenz

- dieses Vorgehen ist unabhängig vom Tool, gilt also auch für andere Machine Learning Umgebungen (R, pySpark, Azure Machine Learning, etc.)
- eine Spezialität von scikit-learn ist aber, dass auch Features und Target als eigenständige Objekte übergeben werden müssen
  - Features als Matrix (X)
  - Target als Vektor (y)
- dieser "Features - Target - Split" erfolgt gewöhnlich **vor** dem Train - Test - Split

1.1 Feature Engineering - Einführung

1.1.8.1 Python Libraries - Feature Engineering

- alle abgegebenen Code Beispiele basieren auf
  - Anaconda3-2024.02-1-Windows-x86_64 Distribution
  - Python 3.11.7
  - und den folgenden Libraries:

Library	Alias	Beschreibung	Vers.
numpy	np	Programmbibliothek für die Programmiersprache Python, die eine einfache Handhabung von Vektoren, Matrizen oder generell grossen mehrdimensionalen Arrays ermöglicht	1.26.4
pandas	pd	eine Programmbibliothek für die Programmiersprache Python, die Hilfsmittel für die Verwaltung von Daten und deren Analyse anbietet	2.1.4
matplotlib.pyplot	plt	mathematische Visualisierungen aller Art, die Synthax basiert auf Mathlab	3.7.5

1.1 Feature Engineering - Einführung

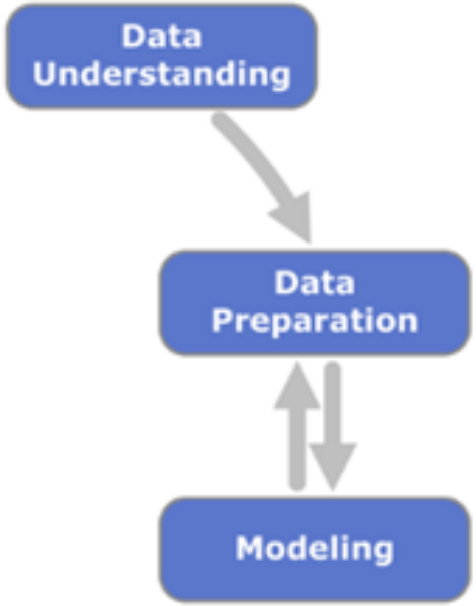
1.1.8.1 Python Libraries - Feature Engineering

Library	Alias	Beschreibung	Vers.
seaborn	sns	basierend auf matplotlib zum Erstellen von attraktiven und informativen statistischen Visualisierungen	0.12.2
pandas-profiling (ydata_profiling)	---	erstellt Profiling Reports, ausgehend von pandas DataFrames; erweitert pandas DataFrame mit der Methode ProfileReport () für schnelle Datenanalysen	4.8.3
scikit-learn (sklearn)	---	die wichtigste Programmbibliothek für den Kursteil Supervised Learning	1.4.2
imbalanced-learn (imblearn)	---	eine Library für den Umgang mit unbalancierten Daten	0.12.2
statsmodels	stat	ein Modul mit Funktionen und Klassen für klassisch-statistische Aufgabenstellungen (optional)	0.14.0

1.1.8.2 Python Libraries - Die Python Libraries und CRISP-DM

einige der genannten Libraries spielen in einzelnen CRISP Phasen jeweils eine zentrale Rolle

- Data Understanding:
  - pandas
  - matplotlib/seaborn
- Data Preparation:
  - numpy
  - pandas
- Modeling
  - scikit-learn (sklearn)



wobei punktuell auch andere Libraries zum Einsatz kommen können

1.1 Feature Engineering - Einführung

1.1.9 Begleitende Literatur

Nr	Titel	Autoren	Verlag	Jahr
2.	Introduction to Machine Learning with Python	A. Müller, S. Guido	O'Reilly	2016
4.	Feature Engineering for Machine Learning	Zheng Alice, Casari Amanda	O'Reilly	2018
5.	Feature Engineering and Selection	Kjell Johnson Max Kuhn	CRC Press [link]	2021
6.	Applied Predictive Modeling	Kjell Johnson Max Kuhn	Springe	2018
7.	Hands-On Machine Learing with Scikit-Learn & TensorFlow (Kap 1-7)	Aurélien Géron	O'Reilly	2022
8.	scikit-learn: , offizielle Webseite der im Kurs benutzten Library		https://scikit-learn.org/stable/	
ausserdem				
	https://towardsdatascience.com/machine-learning/home			

1.1 Feature Engineering - Einführung

Workshop 01 - Skalenniveaus

Gruppen zu 3 bis 4, Zeit: 30'

- untersuchen Sie die Variablen des Melbourne Housing Dataset (melb_data.csv) auf vorliegende Skalenniveaus:
  - kategorial (auch Unterscheidung nominal / ordinal)
  - metrisch
- erstellen Sie eine tabellarische Zusammenstellung mit einem Tool Ihrer Wahl derart, welche sie in folgenden Workshops mit weiteren Informationen ergänzen können (in der vorbereiteten Tabelle auf MS Teams können die Ergebnisse für die ganze Klasse konsolidiert hinterlegt werden)
- konsultieren Sie auch die Online-Dokumentation (ist sie zu den Daten konsistent?)
- falls Sie die Daten gleich mit Python sichten möchten, können diese mit untenstehendem Code geladen werden (der Pfad muss natürlich individuell angepasst werden)

```
import pandas as pd
pd.read_csv('../3_data/melb_data.csv')
```