

A still from the movie Inception showing Leonardo DiCaprio in a dark pinstripe suit, standing in the center of a large, crowded room. He has his arms outstretched to the sides. The room is filled with people, some in formal wear and others in more casual or festive attire. Golden confetti is falling all around him, creating a celebratory atmosphere. The background is slightly blurred, focusing attention on DiCaprio.

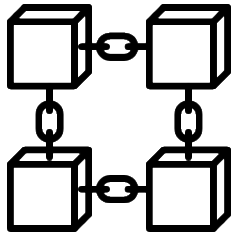
# **Künstliche Intelligenz in der Banking Branche**

**IKT**

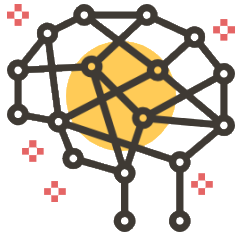
**Marcel Dutt, Christian Zull**

**19 Juni 2018**

# Wir haben jetzt die Theorie zu ...



Blockchain



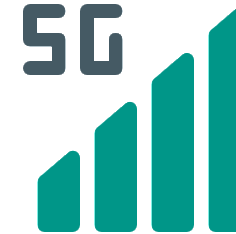
KI



Cloud  
Computing



AR / VR

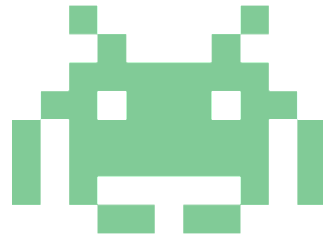


5G



API

# Wir können Programmieren ...



IT 1



IT 2

# Was machen wir daraus?

# Was haben wir eigentlich daraus gemacht?

Entwicklung einer Anwendung  
zur Berechnung des intrinsischen Werts eines Unternehmens.

Anwendung erhält Unternehmenskennzahlen als Input  
und gibt eine Unternehmensbewertung aus.

Programmierung der eines neuronalen Netzes in Python.

Anwendung kann dazu beitragen, Arbeitszeit von Investmentbankern effizienter zu gestalten.





**Welche Erwartungen hast Du an  
den Vortrag?**

# Ziele des Vortrags

## Was sollst Du aus dem Vortrag mitgenommen haben?

**“Wow-Effekt” vor künstlicher Intelligenz verlieren  
Ein bisschen Fleiß und unsere IT Vorlesungen reichen!**

**Überblick über verfügbare Tools am Markt bekommen.**

**Einfache Methoden für die Umsetzung von Machine Learning Modellen.**

**Einblick in den Expert Automation & Augmentation Software Markt bekommen.**

# Gliederung

1

**Einleitung**

2

**Expert Automation & Augmentation Software Markt**

3

**Projektmanagement**

4

**Datensammlung**

5

**Data Cleaning**

6

**Modell / Hyperparameter Optimierung / Training / Validierung**

7

**Anwendung**

# Gliederung

1

Einleitung

2

**Expert Automation & Augmentation Software Markt**

3

Projektmanagement

4

Datensammlung

5

Data Cleaning

6

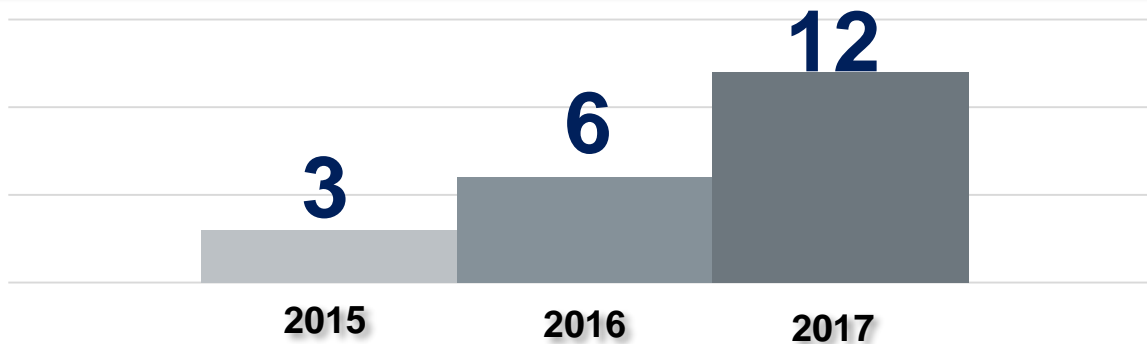
Modell / Hyperparameter Optimierung / Training / Validierung

7

Anwendung

# Künstliche Intelligenz ist momentan besonders “anziehend” für Investoren aus allen Bereichen – bis dato 1031 Start-ups mit Funding!

## Investment in AI & ML



## Wertschöpfungstreiber

- Zunehmender Einsatz von **Machine Learning**
- Erhöhte Performance **Natural Language Processing** und anderen Technologien zur Sprachverarbeitung
- Solide Business Cases zur **Kosteneinsparung** und zusätzlichen **Kundennutzen** und **Zusatzgeschäft**

## Market Size

**5,05-Milliarden-Dollar Markt bis 2020**

## Branchen

- Medien
- Werbung
- Einzelhandel
- Finanzsektor
- Gesundheitssektor

## Major Player



Quelle: Google



Quelle: Nuance

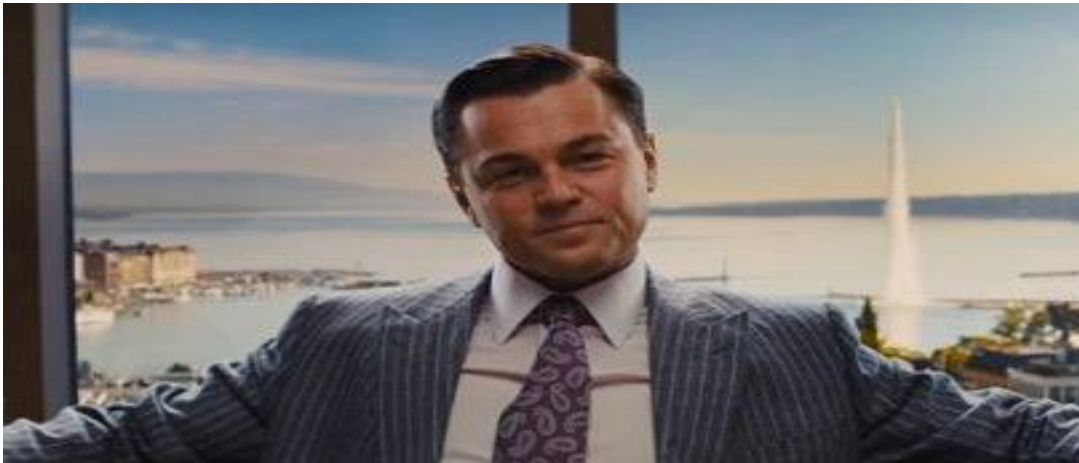


Quelle: Microsoft



# Es werden keine MBAs gesucht, sondern Talente in „Expert Automation & Augmentation Software“ (EAAS)

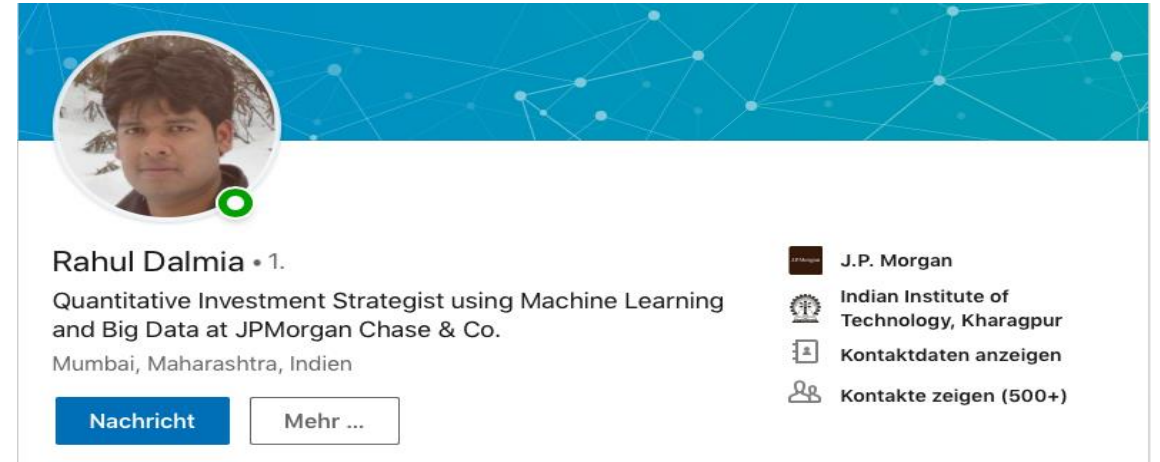
## Banker früher



Quelle: GlobalNews

- Grundstudium Finance / Rechnungswesen
- MBA an einer Top-Tier Uni
- Spaß am Spiel mit Geld und Macht (so sagt man zumindest)

## Banker heute



Quelle: LinkedIn

- Grundstudium Elektrotechnik / IT
- Master / Promotion in Computer Wissenschaft / Data Scientist
- Fortbildung durch anerkannte Verbände

**Mittlerweile gibt es Master in Applied Data Science an der Frankfurt School of Finance und einen Certified Financial Engineer (EIFD)**

# Die Branche wird nicht von großen namhaften Playern dominiert sondern von kleinen Start-ups mit starken Partnern

## Robo Advisory



Quelle: Gründerszene

- Automatisiertes aktives Portfolio Management
- Verwaltetes Vermögen 1 Mrd. €
- Investment ab 10 T€
- Risikomanagement Fokus

## Quant Strategy



Quelle: Algoriz

- Entwerfen automatisierten Handelsstrategien
- Backtesting mit historischen Daten
- Testen Live-Marktdaten
- Risikomanagement



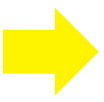



## EAAS



Quelle: CBInsights

- Datenpunkte von hunderten Quellen
- Marktanalysen und Berichte
- Target Lists
- Schwerpunkt VC, Start-ups, Patente, Partnerschaften, News

# Regulationsdruck und sinkende Gewinnmargen machen Investmentbanken schwer zu schaffen

MA Anzahl	Gehalt	Automatisierungspotential	Auswirkung
Directors	850 K		
Associates	325 K		
Analysten	100 K		

## Gewinnoptimierung

Goldman Sachs Analysten prognostizieren, dass AI **ab 2025** einen Marktwert von **39,5 Mrd. € p.a.** schaffen wird.

# Ein Analyst hat viele Aufgabenfelder – Präzision, Ausdauer und hohe Datenmengen sind gefordert

Aktienbewertung  
Investment Management



M&A Bewertung  
Investment Banking



Risikobeurteilung  
Riskomanagement



Unternehmensvergleich  
IB & IM



Pitchbook  
Investment Banking



Unternehmensanalyse  
IB & IM



## Fokus des Projekts

## Auto- matisierung der Aktien- bewertung

## Gedankenspiel:

**Falls unser Projekt tatsächlich Kundenwert schafft, welche Hypothesen sollten wir testen?**

Der Kunde ist bereit auf das Ergebnis zu vertrauen,  
ohne den Rechenweg vorgelegt zu bekommen.

Der Kunde spart tatsächlich Personalkosten ein und hat nicht on-top seinen üblichen  
Personalaufwendungen SaaS Aufwendungen.

Der Aktienmarkt ist in allen Zyklen mit unserem System bewertbar.

Banken möchten Analysten automatisieren.



# Gliederung

1

Einleitung

2

Expert Automation & Augmentation Software Markt

3

**Projektmanagement**

4

Datensammlung

5

Data Cleaning

6

Modell / Hyperparameter Optimierung / Training / Validierung

7

Anwendung

Mit Trello lässt sich ein Projekt gut strukturieren,  
es muss nur leider trotzdem gemacht werden ...

Scrum



Product Backlog

Sprint Backlog

Storypoints

LOC

642

Zeitaufwand

210 h

Lernergebnis

lohnenswert

Projekt  
vs.  
Vortrag

Zeitverteilung:

90 %

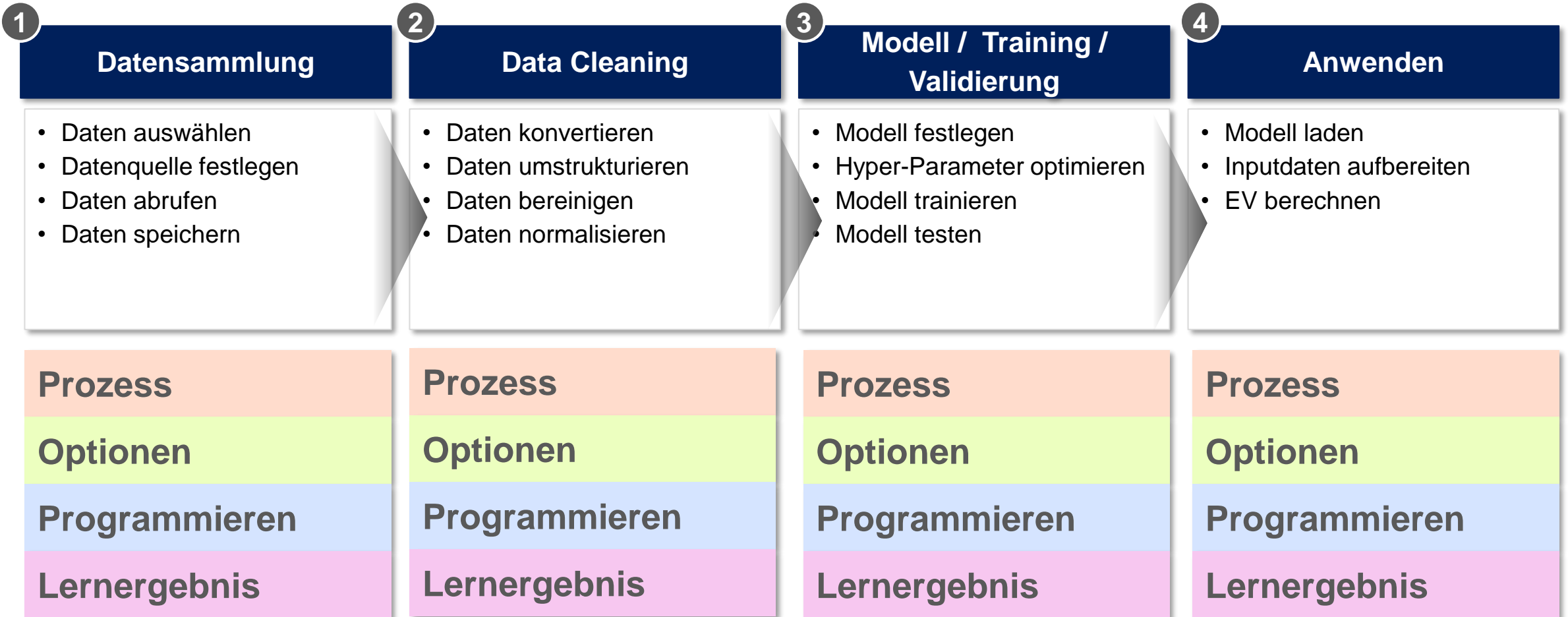
Projekt

10 %

Vortrag

# Entwicklungsmodell

## Zur künstlichen Intelligenz in 4 Schritten



# Der Vortrag / Skript ist in vier Kategorien aufgeteilt. Jede Kategorie verfolgt einen bestimmten Zweck.

<i>Folientyp</i>	<i>Beschreibung</i>	<i>Zielsetzung</i>
Prozess	<ul style="list-style-type: none"><li>• Übersicht im Entwicklungsprozess</li></ul>	<ul style="list-style-type: none"><li>• Entwicklungsprozess verstehen</li></ul>
Optionen	<ul style="list-style-type: none"><li>• Darstellung der gängigen Handlungsoptionen</li></ul>	<ul style="list-style-type: none"><li>• Wissen auf andere Problemstellungen anwenden</li></ul>
Programmieren	<ul style="list-style-type: none"><li>• Oberflächliche Beschreibung der technischen Umsetzung</li></ul>	<ul style="list-style-type: none"><li>• Erhöhung der Methodenkompetenz und tiefenverständnis</li></ul>
Lernergebnis	<ul style="list-style-type: none"><li>• Beschreibung unserer Fehler und Lernergebniss</li></ul>	<ul style="list-style-type: none"><li>• Besser aus Fehlern von anderen, als an den eigenen wachsen</li></ul>

# Gliederung

1

Einleitung

2

Expert Automation & Augmentation Software Markt

3

Projektmanagement

4

**Datensammlung**

5

Data Cleaning

6

Modell / Hyperparameter Optimierung / Training / Validierung

7

Anwendung



## Die Qualität der Daten ist kritisch. Data Science unterscheidet zwischen 3 ver. Verwendungszwecken.

### Shit in – Shit out – Einfaches Prinzip

Das Modell ist nur so gut, wie die Datengrundlage, mit welcher das Modell trainiert wurde.

### Features & Label

Features

Input als Wert

Labels

Ergebnis

### Daten

Training

Training Daten

Training Daten

Test

Test Daten

Test Daten

Anwendungsdaten

Entwicklung

Betrieb

## Welche Vor- und Nachteile haben interne bzw. externe Datenquellen?

### Intern



PostgreSQL

Datenbanken (Zweck)



Textfiles

Menge ●○○○

Individulität ●●●●

Kosten ●●●○



Excel Daten



Datenbanken Intern

### Extern



Kostenlos (i.d.R.)

**API**

Kostenpflichtig

Menge ●●●●

Individulität ●○○○

Kosten ●●○○

**Der Zugang zu Daten ist groß. Die eigentliche Arbeit liegt bei der Selektion der richtigen Quelle.**

## Data Science Allgemein



Google Cloud Platform

Quelle: Dataquest

Große Datensätze z.B.  
Kriminalität in Chicago



Quelle: Dataquest

E-Commerce Plattform Daten

# Quandl

Quelle: Quandl

Ökonomische & Finanzdaten  
z.B. Chinesisches  
Gesundheitssystem



THE WORLD BANK

Quelle: Dataquest

Finanzdaten aus  
Entwicklungsländern

# FiveThirtyEight

Quelle: Dataquest

News und Sport Daten



Quelle: Dataquest

Daten der US Regierung

# INTRINIO

Quelle: Intrinio

Bilanzen, GuV, Cashflow  
Statements, Indexe, etc.

# YAHOO!

FINANCE

Quelle: Yahoo Finance

Bilanzen, GuV, Cashflow  
Statements, Indexe, etc.

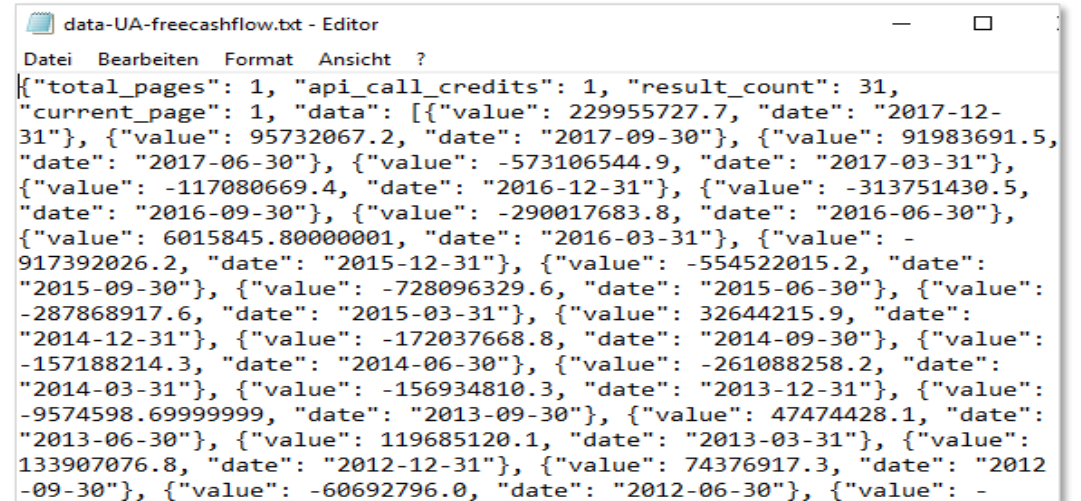
## Datenstream von APIs ist sehr einfach, vor allem mit der Verwendung von passenden Datenformaten.

### Daten Download von Intrinio

```
USER = '349b60cec630e203f91192e0e0eca807'  
PW = '2c4c43249913c328f56a25d8ac49d994'  
Values = ["dividendyield", "earningsyield", "enterprisevalue", "evtoebit",  
          "evtoebitda", "evtofcff", "evtoinvestedcapital", "evtonopat",  
          "evtoof", "pricetoearnings", "ebitdagrowth", "freecashflow",  
          "revenuegrowth"]  
def download(company, value):  
    ...  
    for item1 in company:  
        for item2 in value:
```

1. Mit Intrinio verbinden durch Kennung
2. Features und Unternehmen auswählen
3. Anfordern der Daten durch "get-request"

### Format JSON



The screenshot shows a text editor window titled "data-UA-freecashflow.txt - Editor". The editor contains a JSON array of objects, each representing a data point with a "value" and a "date". The data points are for various financial metrics like "freecashflow" and "ebitdagrowth" for different companies, spanning from 2012 to 2017. The JSON is compact and machine-readable.

Quelle: Eigene Darstellung

- Kompakt
- Vom Menschen lesbar
- Gibt Datentyp an
- Große Datenmengen leicht übertragbar

## Datenbeschaffung ist schwieriger als man denkt ...

Lernergebnis	Beispiel
<ul style="list-style-type: none"><li>• Das Sammeln von Daten wird absolut unterschätzt.</li></ul>	<ul style="list-style-type: none"><li>• Das Sammeln von Daten wird absolut unterschätzt.</li></ul>
<ul style="list-style-type: none"><li>• Es entstehen einige Unternehmen, die sich um Datenbeschaffung für Data Science kümmern.</li></ul>	<ul style="list-style-type: none"><li>• Ohne Intrinio wäre die Umsetzung unseres Projekts nicht möglich gewesen.</li></ul>
<ul style="list-style-type: none"><li>• Nicht jeder Datensatz ist geeignet, sei es Datenqualität oder Verfügbarkeit. Am besten vorher einen Plan B haben!</li></ul>	<ul style="list-style-type: none"><li>• Wir haben ursprünglich mit Europe Small 200 gerechnet, mussten allerdings auf S&amp;P 500 umschwenken.</li></ul>



# Gliederung

1

Einleitung

2

Expert Automation & Augmentation Software Markt

3

Projektmanagement

4

Datensammlung

5

**Data Cleaning**

6

Modell / Hyperparameter Optimierung / Training / Validierung

7

Anwendung

## Das Data Cleaning bringt Daten in eine für das Modell verwertbare Form.

### Input

```
data-UA-freecashflow.txt - Editor
Datei Bearbeiten Format Ansicht ?
{"total_pages": 1, "api_call_credits": 1, "result_count": 31,
"current_page": 1, "data": [{"value": 229955727.7, "date": "2017-12-31"}, {"value": 95732067.2, "date": "2017-09-30"}, {"value": 91983691.5,
"date": "2017-06-30"}, {"value": -573106544.9, "date": "2017-03-31"},
{"value": -117080669.4, "date": "2016-12-31"}, {"value": -313751430.5,
"date": "2016-09-30"}, {"value": -290017683.8, "date": "2016-06-30"},
{"value": 6015845.80000001, "date": "2016-03-31"}, {"value": -
917392026.2, "date": "2015-12-31"}, {"value": -554522015.2, "date":
"2015-09-30"}, {"value": -728096329.6, "date": "2015-06-30"}, {"value":
-287868917.6, "date": "2015-03-31"}, {"value": 32644215.9, "date":
"2014-12-31"}, {"value": -172037668.8, "date": "2014-09-30"}, {"value":
-157188214.3, "date": "2014-06-30"}, {"value": -261088258.2, "date":
"2014-03-31"}, {"value": -156934810.3, "date": "2013-12-31"}, {"value":
-9574598.699999999, "date": "2013-09-30"}, {"value": 47474428.1, "date":
"2013-06-30"}, {"value": 119685120.1, "date": "2013-03-31"}, {"value":
133907076.8, "date": "2012-12-31"}, {"value": 74376917.3, "date": "2012
-09-30"}, {"value": -60692796.0, "date": "2012-06-30"}, {"value": -
```

Quelle: Eigene Darstellung

**JSON-Format**

**Verschachtelte  
Datenstruktur**

**Verschiedenste  
Größenordnungen**

**6.565 Dateien**

data\_cleaning(folder,  
file\_features, file\_labels)

### Output

	A	B	C	D	E	F	G
1	new_index	dividendyield	earningsyield	ebitdagrowth	evtoebit	evtoebitda	evtofcff
2	ALV2017-12-31	-1.0503792739179543	0.40205723006889293	-1.142814998006331	-0.83827131803		
3	ALV2018-03-31	-1.3534177246878225	0.01032059440202213	-1.2515114109675147	0.11998192315		
4	DTE2017-12-31	0.5061422392062255	1.6478549061895087	1.0306636229998636	-1.407382436638		
5	MRK2017-12-31	0.8846515112605836	-1.0465210138395984	0.9422516954304887	1.209362099215		
6	MRK2018-03-31	1.013003248138969	-1.013711716820825	0.42141109054349296	0.9163097322991		
7							
8							
9							
10							

Quelle: Eigene Darstellung

**CSV-Files (Features und  
Labels)**

**Normalisiert (Z-Wert)**

**330 Datensätze**

**Keine NaN-Wert**

## Python bietet als Allzweck-Programmiersprache viele Funktionen zum Data Cleaning.

Quelle: flaticon.com



### Vorteil

- Übersichtlich
  - Gewohnte Umgebung
  - Intuitive Bedienung
- 
- Optimiert für Data Science
  - Umfangreiche Funktionen
- 
- Allzweck-Programmiersprache
  - Umfangreiche Funktionen

### Nachteil

- Begrenzte Anzahl an Datenpunkten verarbeitbar
  - Komplexe Operationen sind aufwändig
  - Kein TensorFlow
- 
- Funktionen außerhalb des Data Science-Felds oft nicht vorhanden
  - Weniger intuitiv als Excel
  - Kein TensorFlow
- 
- Data Science-Funktionen oft nicht optimal
  - Weniger intuitiv als Excel

## 3 wichtige Methoden zum Data Cleaning

### Python

```
glob.glob(  
    pathname  
)
```

```
pandas.DataFrame.apply(  
    func, axis=0,  
    broadcast=False,  
    raw=False,  
    reduce=None,  
    args=(), **kwargs)
```

```
pandas.DataFrame.pivot(  
    index=None,  
    columns=False,  
    values=None  
)
```

### Funktion

Holt alle Dateinamen in einem angegebenen Pfad, die gegebenem Pattern entsprechen.

Wendet eine Funktion auf ein DataFrame entlang der angegebenen Achse an.

Erstellt einen „Pivot“ Table aus gegebenen Werten

### Beispiel

```
glob.glob(*.txt)
```

```
all_stmt.apply(lambda x: x["date"])
```

```
all_stmt.pivot(  
    index = „new_index“  
    columns=„item“,  
    values=„value“  
)
```

## Data Cleaning bedeutet aus Scheiße Gold zu machen.

Lernergebnis	Beispiel
<ul style="list-style-type: none"><li>• Rechne mit allem!</li></ul>	<ul style="list-style-type: none"><li>• Es gab Duplikate (mehrere Datensätze zum selben Unternehmen, zur selben Eigenschaft, zum selben Zeitpunkt)</li></ul>
<ul style="list-style-type: none"><li>• Überlege bei jeder Zeile Code, ob sie notwendig ist. Ansonsten kann es sehr lange dauern, den Code auszuführen.</li></ul>	<ul style="list-style-type: none"><li>• Data Cleaning von 6.565 Dateien benötigte zwischenzeitlich bis zu 15min.</li></ul>
<ul style="list-style-type: none"><li>• Versuche, so viele Daten wie möglich zu erhalten, solange diese gut genug bleiben.</li></ul>	<ul style="list-style-type: none"><li>• Aus 6.565 Dateien (500 Unternehmen) haben wir 330 Datensätze gewonnen.</li></ul>



# Gliederung

1

Einleitung

2

Expert Automation & Augmentation Software Markt

3

Projektmanagement

4

Datensammlung

5

Data Cleaning

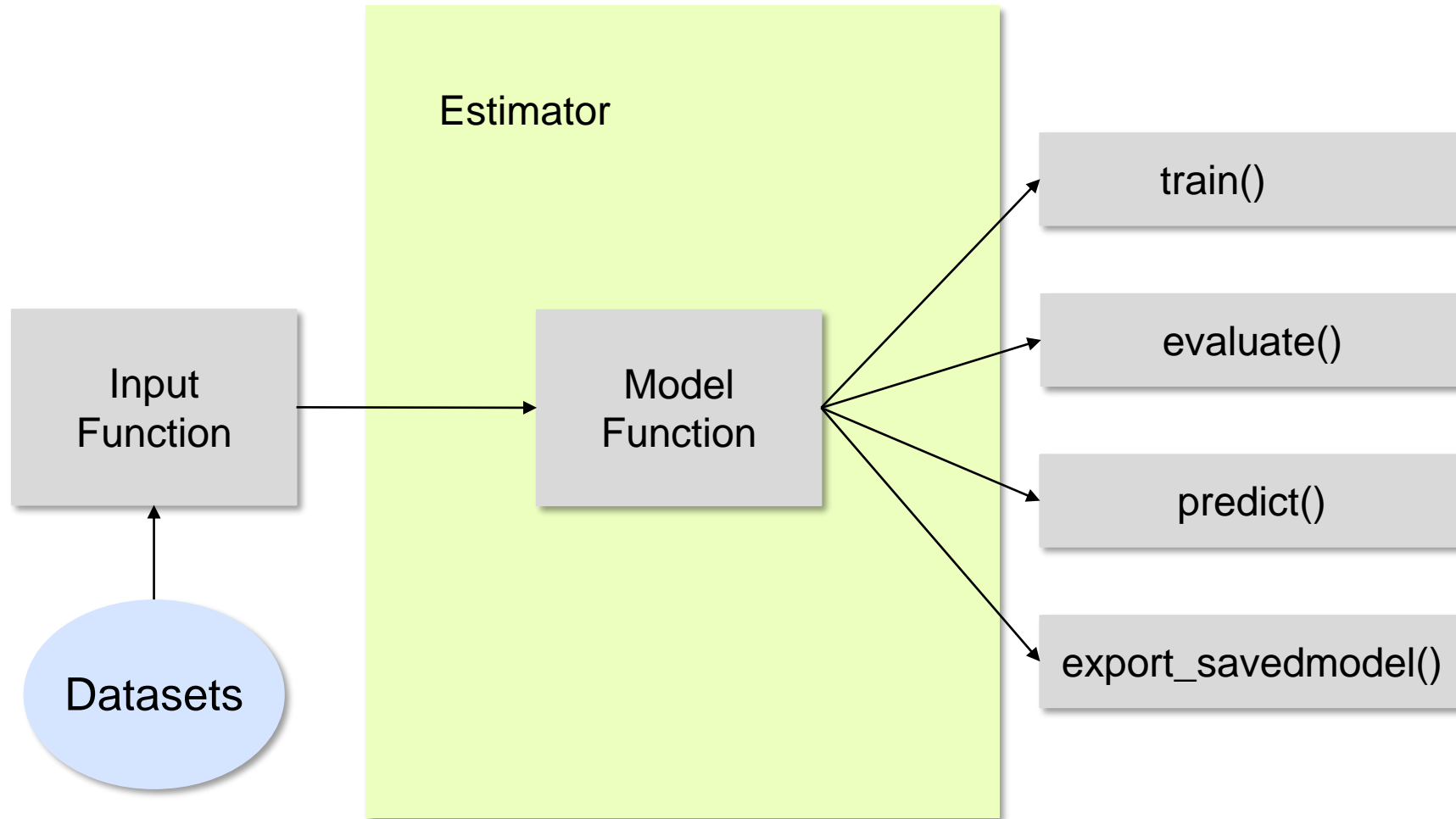
6

**Modell / Hyperparameter Optimierung / Training / Validierung**

7

Anwendung

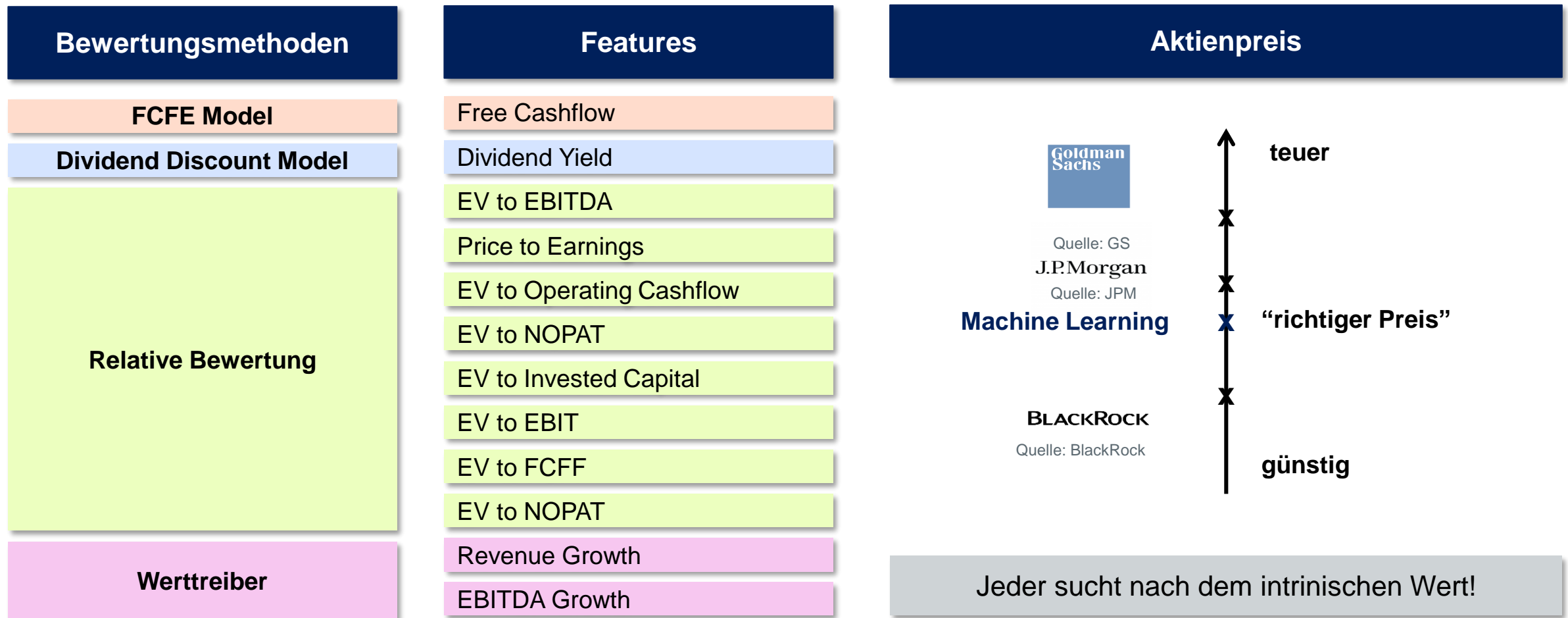
## Modelle in TensorFlow



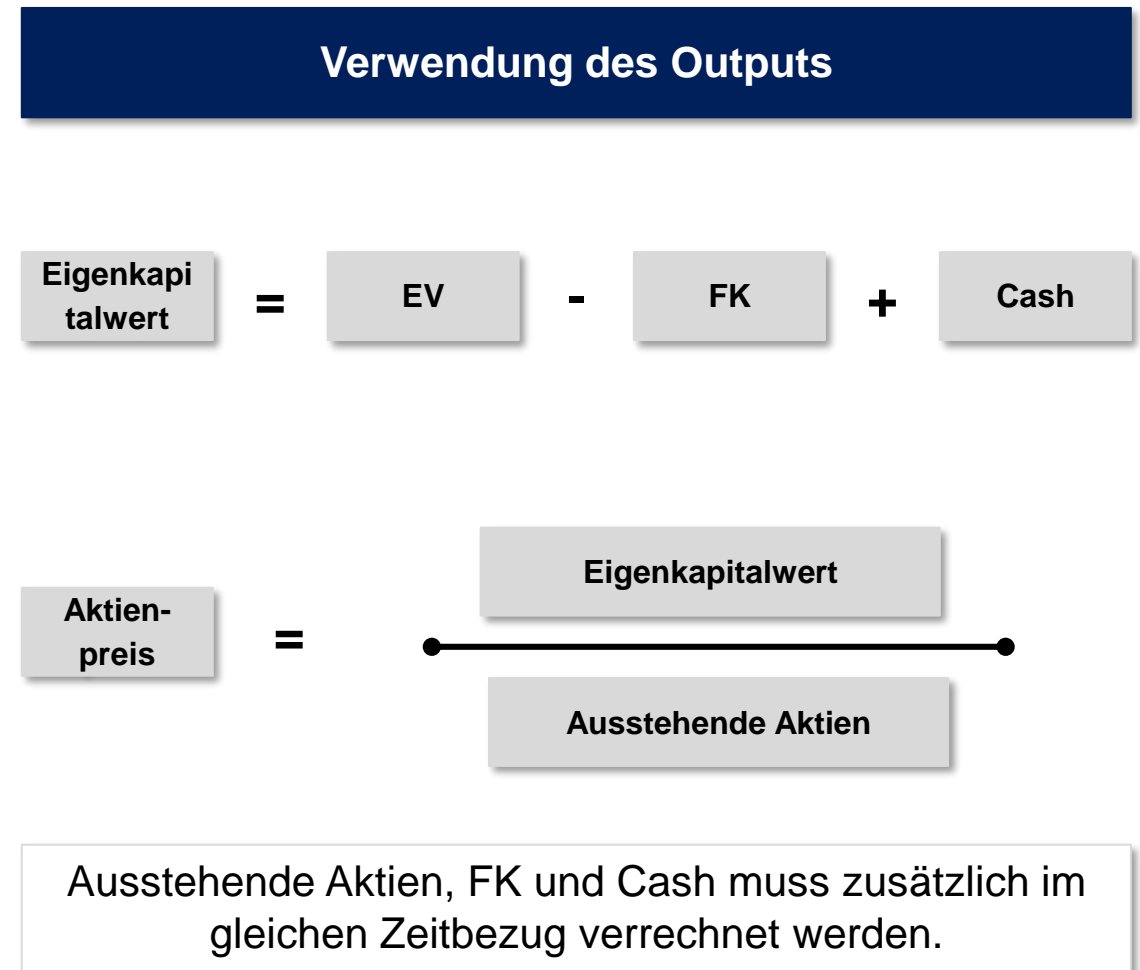
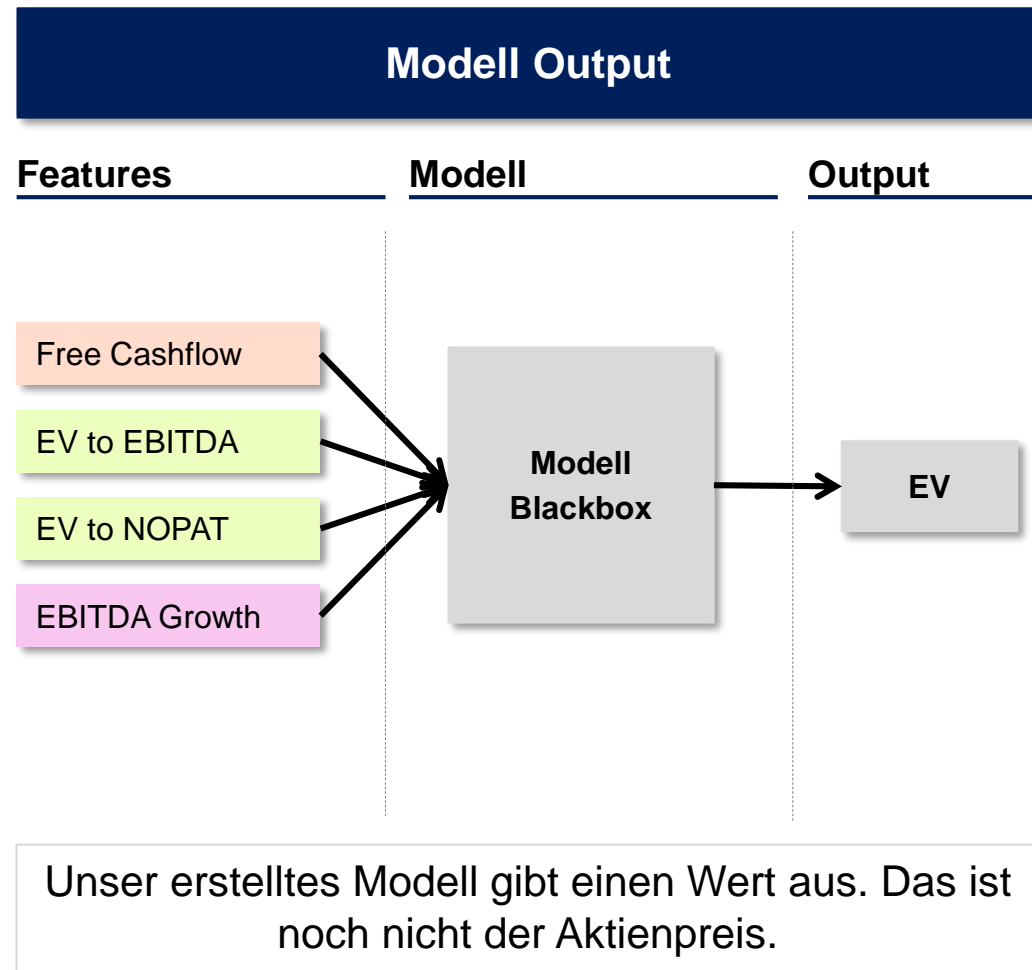
**Modelle heißen in TensorFlow Estimator.**

Ein Estimator bekommt Input von einer Input Function und kann mit diesem die Model Function trainieren, testen, anwenden oder speichern.

## Der Preis einer Aktie hängt von Informationen, Bewertungsmethoden und der Qualität der Daten ab. Wo liegt die Wahrheit?



**Der Output unseres Modell (EV) wird durch einfache finanzmathematische Zusammenhänge zu unserem Zielwert, dem Aktienpreis.**



## Gibt es einfache Regressionsmodelle in der Praxis?



Quelle: Tensorflow



AWS Machine Learning  
(Amazon)

Quelle: AWS



Quelle: Pytorch



Azure ML  
(Microsoft)

Quelle: Azure

## Tools für Daten- getriebenes Modellieren

Viele Unternehmen bieten "Vorlagen" an, in denen Machine Learning-Modelle und Funktionen implementiert sind.

Diese können nach dem Baukasten-Prinzip zu eigenen Modellen zusammengesetzt werden.

## tf.estimator.DNNRegressor initialisiert ein Regressormodell.

### Python

```
tf.estimator.DNNRegressor(  
    hidden_units,  
    feature_columns,  
    model_dir=None,  
    label_dimension=1,  
    weight_column=None,  
    optimizer='Adagrad',  
    activation_fn=tf.nn.relu,  
    dropout=None,  
    input_layer_partitioner=None,  
    config=None,  
    warm_start_from=None,  
    loss_reduction=losses.Reduction.SUM  
)
```

### Funktion

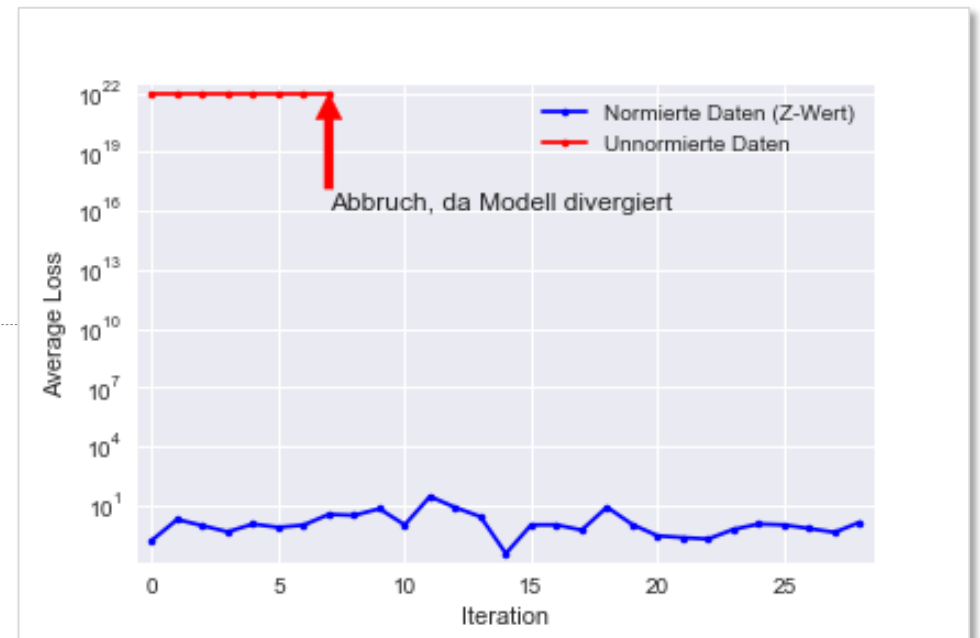
Initialisiert einen Regressor  
(=Modell).

### Beispiel

```
tf.estimator.DNNRegressor(  
    feature_columns = feature_cols(),  
    hidden_units = [5, 5],  
    activation_fn = tf.nn.tanh,  
    dropout = 0.7  
)
```

## Neuronale Netze funktionieren nur in geeigneter Umgebung.

Lernergebnis	Beispiel
<ul style="list-style-type: none"><li>Selbst mäßig komplexe Modelle benötigen viel Rechenkapazität</li></ul>	<ul style="list-style-type: none"><li>Programm stürzte ab, weil adressierbare Speicherzellen ausgehen.</li><li>Insgesamt haben unsere Modelle mehrere Tage trainiert.</li></ul>
<ul style="list-style-type: none"><li>TensorFlow ist für Linux konzipiert. Linux ist nicht Windows.</li></ul>	<ul style="list-style-type: none"><li>Modellverzeichnisse können in Windows nicht angegeben werden, da TensorFlow Pfade im Stil von Linux angibt.</li></ul>
<ul style="list-style-type: none"><li>Normalisiere deinen Input.</li></ul>	<ul style="list-style-type: none"><li>Bei großen Werten lernt das neuronale Netz nicht.</li><li>Deshalb alle Daten in den Bereich von ca. <math>[-1, 1]</math> normalisieren.</li></ul>



Quelle: Eigene Darstellung

# Gliederung

1

Einleitung

2

Expert Automation & Augmentation Software Markt

3

Projektmanagement

4

Datensammlung

5

Data Cleaning

6

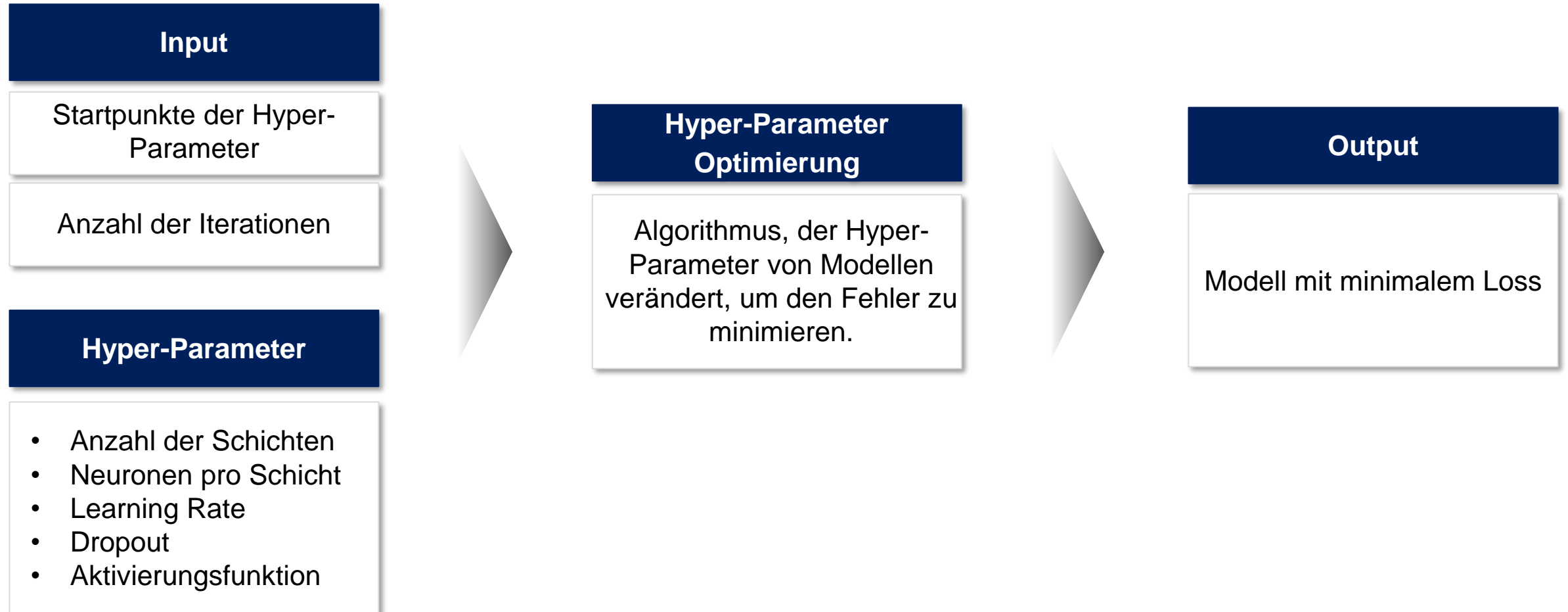
**Modell / Hyperparameter Optimierung / Training / Validierung**

7

Anwendung



# Generischer Ablauf der Hyper-Parameter Optimierung.



## Die Bayes'sche Optimierung bietet eine zielorientierte Optimierung neuronaler Netze.

	<i>Erläuterung</i>	<i>Vorteil</i>	<i>Nachteil</i>
<b>Random Search</b>	<ul style="list-style-type: none"><li>Hyper-Paramter werden zufällig variiert.</li></ul>	<ul style="list-style-type: none"><li>Zeiteffizient bei komplexen Modellen.</li></ul>	<ul style="list-style-type: none"><li>Ergebnis ist zufällig.</li><li>Optimale Lösung nicht garantiert.</li></ul>
<b>Grid Search</b>	<ul style="list-style-type: none"><li>Hyper-Paramter werden systematisch durchprobiert.</li></ul>	<ul style="list-style-type: none"><li>Systematisches Vorgehen und einfache Berechnung der nächsten Iteration.</li></ul>	<ul style="list-style-type: none"><li>Bei komplexen Modellen sehr zeitaufwändig.</li></ul>
<b>Bayes'sche Optimierung</b>	<ul style="list-style-type: none"><li>Hyper-Paramter werden sukzessive optimiert.</li></ul>	<ul style="list-style-type: none"><li>Verbessert Modell zielgerichtet.</li></ul>	<ul style="list-style-type: none"><li>Berechnung der Hyper-Paramter erzeugt Mehraufwand.</li></ul>

**gp\_minimize** aus der Library **scikit-optimize** wird verwendet, um den Fehler einer Menge von Modellen zu minimieren.

## Python

```
skopt.gp_minimize(  
    func, dimensions,  
    base_estimator=None,  
    n_calls=100,  
    n_random_starts=10,  
    acq_func='gp_hedge',  
    acq_optimizer='auto',  
    x0=None, y0=None,  
    random_state=None,  
    verbose=False,  
    callback=None,  
    n_points=10000,  
    n_restarts_optimizer=5,  
    xi=0.01, kappa=1.96,  
    noise='gaussian',  
    n_jobs=1  
)
```

## Funktion

Minimiert die Funktion `func`  
durch Bayes'sche Optimierung.

## Beispiel

```
skopt.gp_minimize(func = hparams_fitness,  
    dimensions =  
    hparams_dimensions(),  
    acq_func = 'EI',  
    n_calls = 40,  
    x0 = Default_parameters)
```

## “Größer und komplexer” ergibt nicht immer bessere Ergebnisse.

Lernergebnis	Beispiel
<ul style="list-style-type: none"><li>Unsere Modelle sind oft nicht sehr komplex. Ein Grid Search wäre daher besser gewesen.</li></ul>	<ul style="list-style-type: none"><li>Unsere besten Modelle haben oft nur 1-3 Layer.</li></ul>
<ul style="list-style-type: none"><li>Tiefer ist nicht immer besser.</li></ul>	<ul style="list-style-type: none"><li>Unser bestes Modell hat 5 Neuronen und 1 Layer.</li></ul>

The heatmap visualizes the average loss across different model configurations. The vertical axis represents the number of layers (1 to 10), and the horizontal axis represents the number of nodes per layer (100 to 500). The color intensity indicates the average loss, with a scale from 0.0 (dark) to 28.8 (light). A blue arrow points to the 'optimal model' at 1 layer and 100 nodes per layer, which shows the lowest loss.

# Gliederung

1

Einleitung

2

Expert Automation & Augmentation Software Markt

3

Projektmanagement

4

Datensammlung

5

Data Cleaning

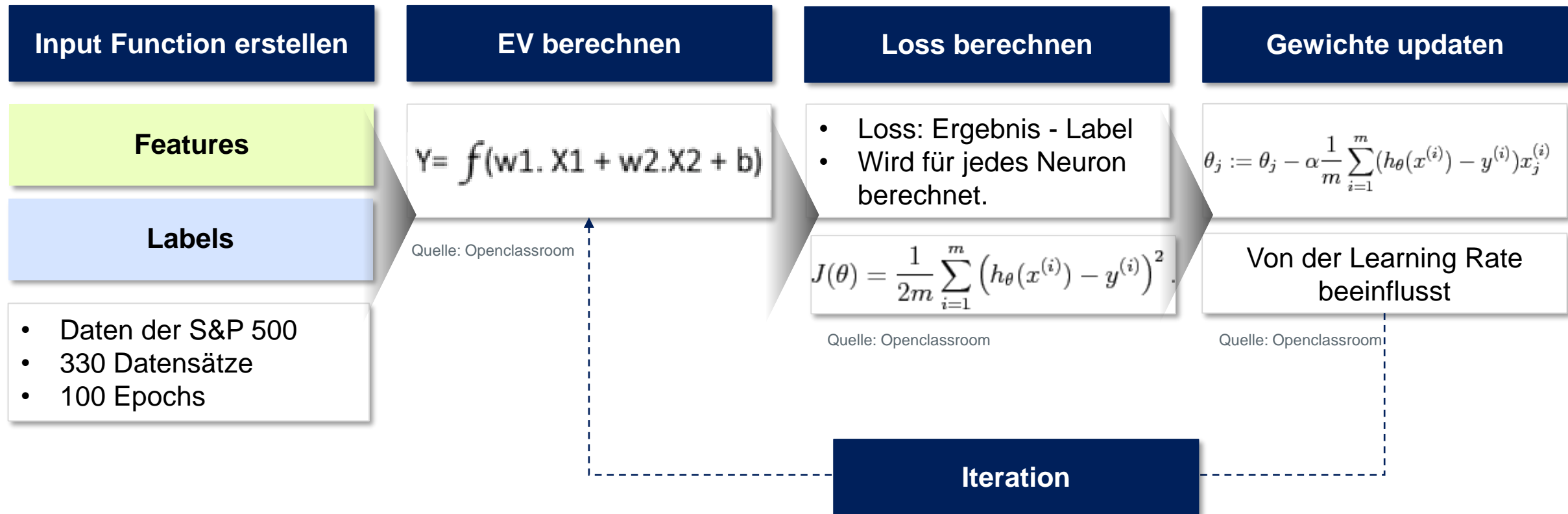
6

**Modell / Hyperparameter Optimierung / Training / Validierung**

7

Anwendung

## Wie lernt eine Maschine?



## Wie oft soll mit denselben Datensätzen trainiert werden?

### Mehr Epochs

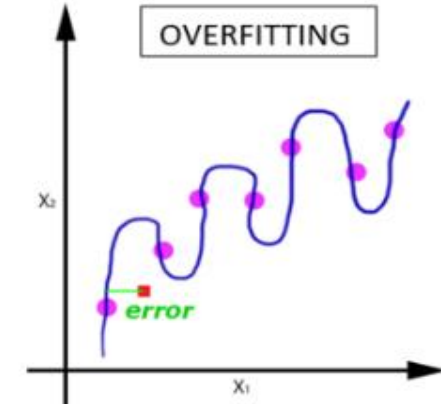
- Weniger Datensätze nötig
- Verringert Underfitting

- Sorgt für Overfitting

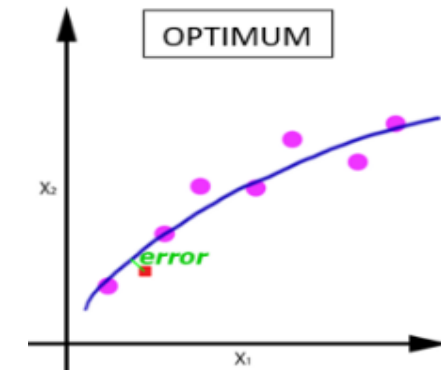
### Weniger Epochs

- Stellt u.U. bessere Generalisierung sicher

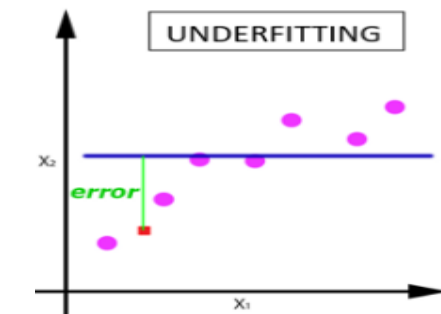
- Sorgt u.U. für Underfitting
- Mehr Datensätze notw.



Quelle: Openclassroom



Quelle: Openclassroom



Quelle: Openclassroom

## 3 wichtige Methoden zum Training

### Python

```
tf.constant(
    value,
    dtype=None,
    shape=None,
    name='Const',
    verify_shape=False
)
```

```
tf.data.Dataset
.from_tensor_slices(tensors)
```

```
tf.estimator.DNNRegressor.train(
    input_fn,
    hooks=None,
    steps=None,
    max_steps=None,
    saving_listeners=None
)
```

### Funktion

Erstellt ein Tensor-Objekt aus  
gegebenen Daten.

Erstellt ein Dataset-Objekt aus  
Tensor-Objekten.

Trainiert ein gegebenes Modell  
mit einer Input Function

### Beispiel

```
{k:
tf.constant(features[k].values)
for k in features}
```

```
tf.data.Dataset.
from_tensor_slices(
(features, labels))
```

```
model.train(input_fn = lambda: value,
            max_steps = training_steps)
```



## ...jetzt klar, warum Facebook und Google führend in AI sind??

Lernergebnis	Beispiel
<ul style="list-style-type: none"><li>Dictionary != Tensor != Dataset != Iterator</li></ul>	<ul style="list-style-type: none"><li>Daten müssen zunächst in Tensor-Objekte konvertiert werden, um zu Dataset-Objekten konvertiert zu werden, ..., um zum Training verwendet zu werden.</li></ul>
<ul style="list-style-type: none"><li>Wenig Daten bleiben wenig Daten</li></ul>	<ul style="list-style-type: none"><li>Mit mehr Daten und weniger Epochs könnten wir vermutlich bessere Ergebnisse erzielen. Viele Modelle waren overfitted. Das Beste Modell war sehr klein.</li></ul>

# Gliederung

1

Einleitung

2

Expert Automation & Augmentation Software Markt

3

Projektmanagement

4

Datensammlung

5

Data Cleaning

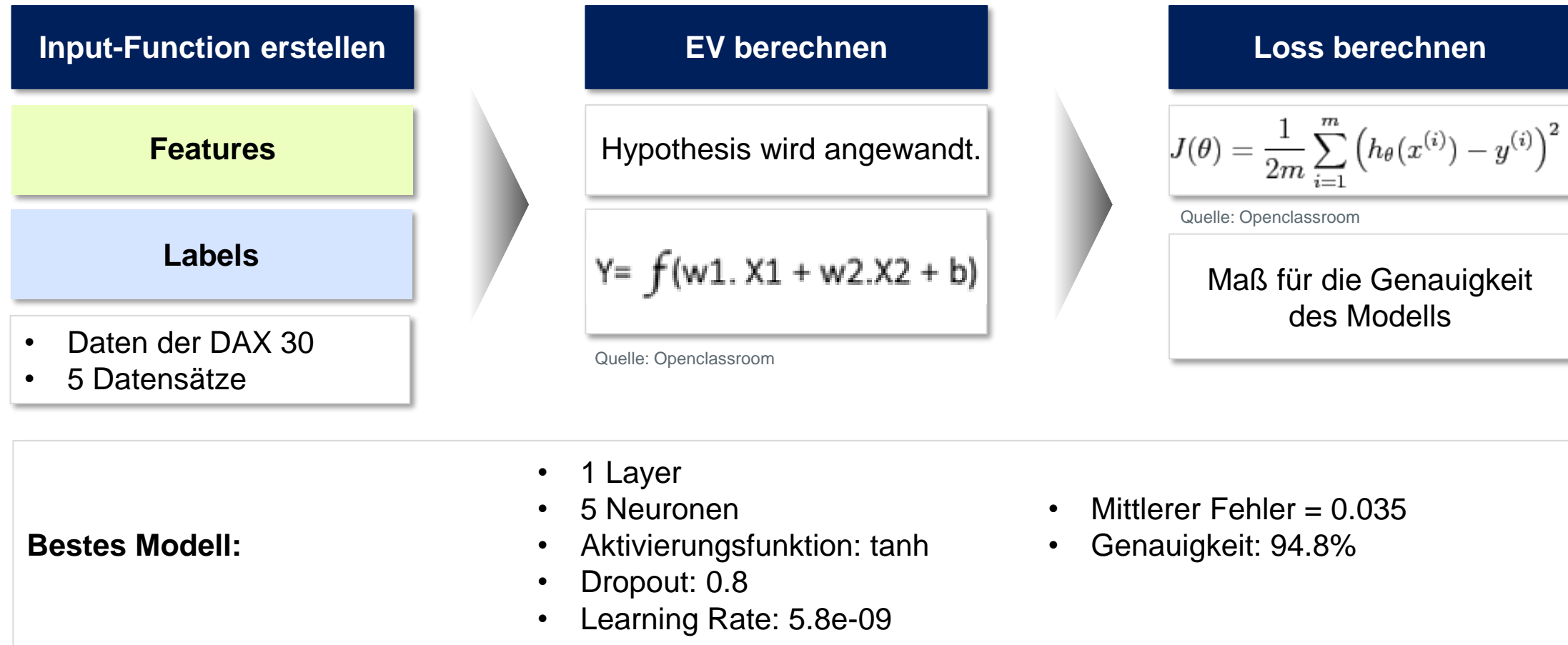
6

**Modell / Hyperparameter Optimierung / Training / Validierung**

7

Anwendung

**Modelle werden mit Daten getestet, die nicht für das Training verwendet wurden. So kann Overfitting erkannt werden.**



## Wie viele Daten sollen zu Testzwecken vorbehalten werden?

	<i>Vorteil</i>	<i>Nachteil</i>
Testdaten aus Trainings-Daten nehmen	<ul style="list-style-type: none"><li>• Alle Datensätze können zum Training genutzt werden</li></ul>	<ul style="list-style-type: none"><li>• Modell kann nicht auf Overfitting geprüft werden.</li></ul>
Zusätzliche Daten verwenden	<ul style="list-style-type: none"><li>• Modell wird auf Overfitting geprüft</li></ul>	<ul style="list-style-type: none"><li>• Weniger Datensätze stehen zum Training zur Verfügung.</li></ul>

## Mit `tf.estimator.DNNRegressor.evaluate()` werden Modelle getestet.

### Python

```
tf.estimator.DNNRegressor.evaluate(  
    input_fn,  
    steps=None,  
    hooks=None,  
    checkpoint_path=None,  
    name=None  
)
```

### Funktion

Testet ein gegebenes Modell  
mit einer gegebenen Input-  
Funktion

### Beispiel

```
model.evaluate(  
    input_fn = lambda: value,  
    steps = 5)
```

## Der Teufel steckt im Detail.

### Lernergebnis

- Unsere Modelle overfitten schnell.
- Validierungs-Datensatz muss mit Mean und STD der Trainings-Daten normalisiert werden.

### Beispiel

- Loss bei Testdaten war oft das Zehnfache der Daten im Training.
- ```
print(features_eval.mean(0) / features.mean(0))
```

|                     |           |
|---------------------|-----------|
| dividendyield       | 1.330753  |
| earningsyield       | 0.631827  |
| ebitdagrowth        | -0.077676 |
| evtoebit            | 0.762084  |
| evtoebitda          | 0.831240  |
| evtofcff            | 2.149520  |
| evtoinvestedcapital | 0.315222  |
| evtonopat           | 0.920268  |
| freecashflow        | 0.590800  |
| pricetoearnings     | 0.821262  |
| revenuegrowth       | 0.566221  |
| dtype:              | float64   |

# Gliederung

1

Einleitung

2

Expert Automation & Augmentation Software Markt

3

Projektmanagement

4

Datensammlung

5

Data Cleaning

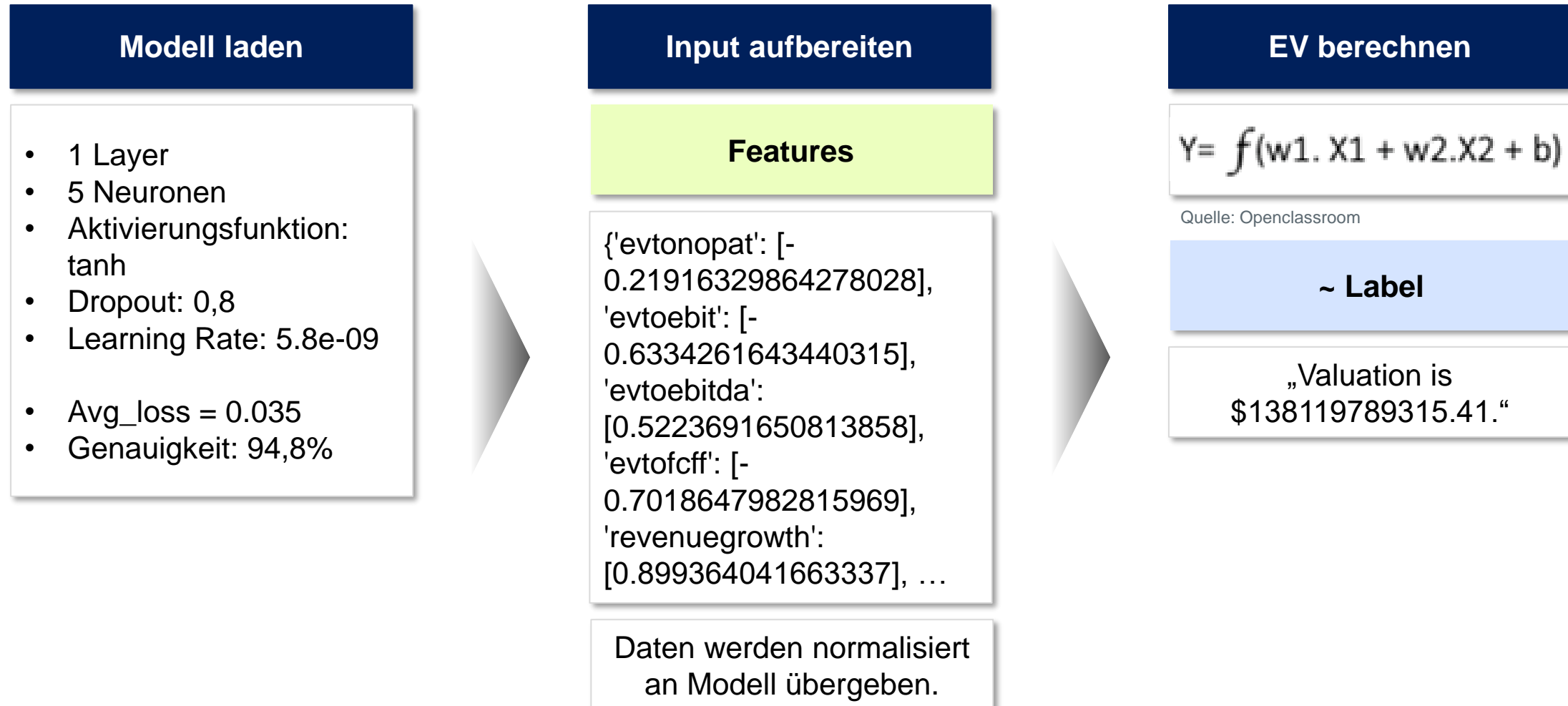
6

Modell / Hyperparameter Optimierung / Training / Validierung

7

**Anwendung**

## Beim Anwenden des Modells wird ein unbekannter Enterprise Value berechnet.

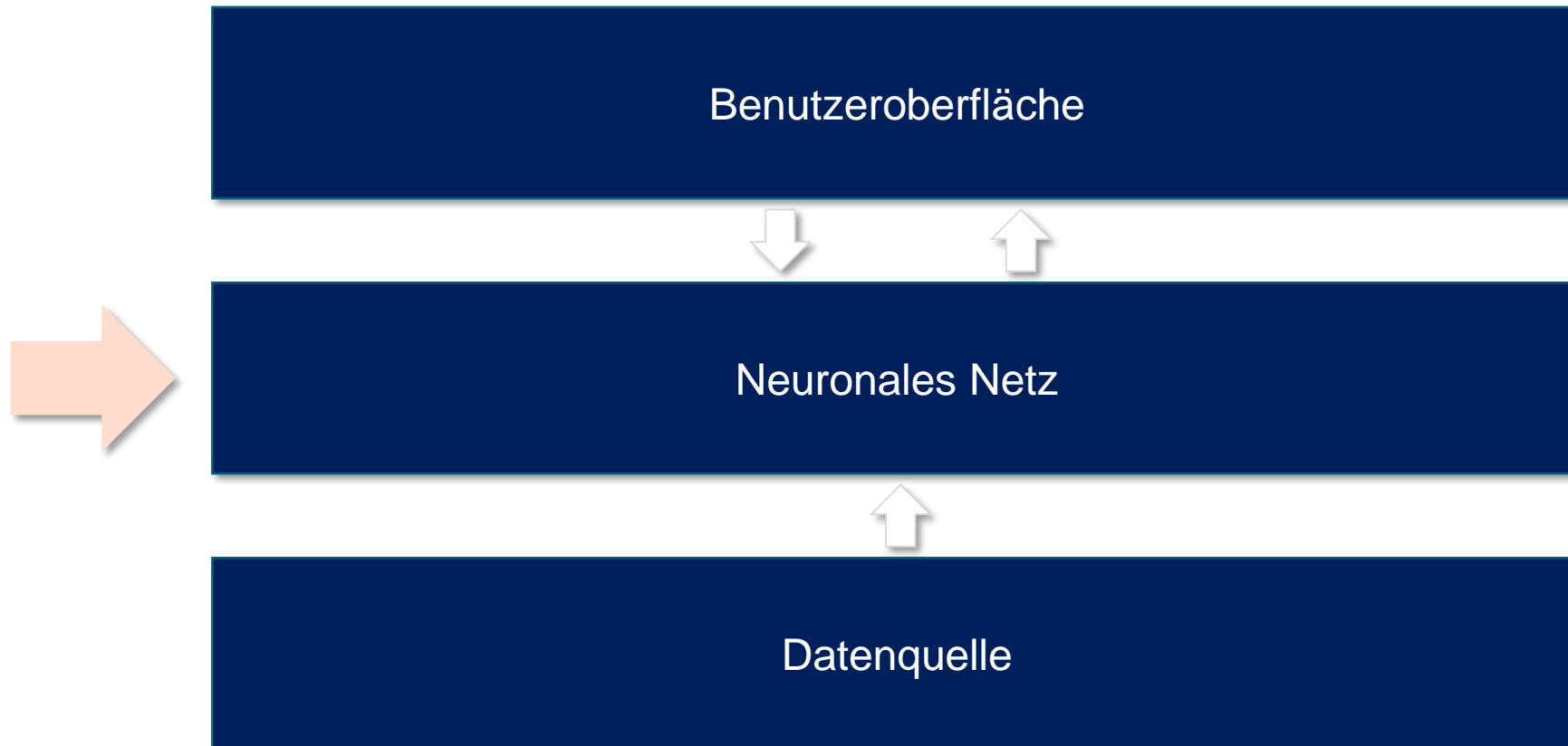






# Projektergebnis

**Das Modell kann nun in Anwendungen eingebunden werden.**



Das trainierte Modell kann nun in eine komplexere Anwendung integriert werden und Inputs von Datenquellen oder aus einer Benutzeroberfläche verarbeiten.

## Mit `tf.estimator.DNNRegressor.predict()` werden Modelle angewandt.

### Python

```
tf.estimator.DNNRegressor
.predict(
    input_fn,
    predict_keys=None,
    hooks=None,
    checkpoint_path=None,
    yield_single_examples=True
)
```

### Funktion

Macht Vorhersagen für  
gegebene Features anhand  
eines trainierten Modells

### Beispiel

```
model.predict(input_fn = lambda:
    tf.data.Dataset
        .from_tensor_slices(
            predict_x)
        .batch(len(predict_x)))
```

## Hat alles geklappt?

### Lernergebnis

- Keine Fehler gemacht?
- 
- Normalisiere auch deine Inputdaten bei der Anwendung

### Beispiel

- Ein Modell funktioniert auch, wenn es falsch erstellt wurde. Weißt du, dass du nichts falsch gemacht hast?
- 
- Unser Modell wurde mit normalisierten Daten erstellt. Es funktioniert auch deshalb nur mit genauso (gleicher Mittelwert, gleiche Standardabweichung) normalisierten Daten.

# Quellenangaben

- Ang, C., 2018. *DataCamp*. [Online]  
Available at: <https://www.datacamp.com/instructors/csa>  
[Zugriff am 18 Juni 2018].
- Butcher, D., 2017. *Goldman Sachs has created an elite tech team to tackle AI, big projects*. [Online]  
Available at: <https://news.efinancialcareers.com/us-en/301350/goldman-building-new-rd-engineering-group-hiring-ai-team>  
[Zugriff am 18 Juni 2018].
- Dalmia, R., 2018. *LinkedIn*. [Online]  
Available at: <https://www.linkedin.com/in/rahul-dalmia-a73a1712/>  
[Zugriff am 18 Juni 2018].
- Dataquest, 2018. *Free Datasets for Projects*. [Online]  
Available at: <https://www.dataquest.io/blog/free-datasets-for-projects/>  
[Zugriff am 18 Juni 2018].
- Developers, G., 2018. *Youtube: Effective TensorFlow for Non-Experts (Google I/O '17)*. [Online]  
Available at: <https://www.youtube.com/watch?v=5DknTFbcGVM>  
[Zugriff am 18 Juni 2018].
- developers, s.-l., 2017. 3.2. *Tuning the hyper-parameters of an estimator*. [Online]  
Available at: [http://scikit-learn.org/stable/modules/grid\\_search.html](http://scikit-learn.org/stable/modules/grid_search.html)  
[Zugriff am 18 Juni 2018].
- Europäisches Institut für Financial Engineering und Derivateforschung, 2018. *CERTIFIED FINANCIAL ENGINEER*. [Online]  
Available at: <http://www.eifd.de/inhalte/eifd-startseite/>  
[Zugriff am 18 Juni 2018].
- Fool, M., 2016. *10 Statistiken über künstliche Intelligenz, die dich vom Hocker hauen werden*. [Online]  
Available at: <https://www.fool.de/2016/06/27/10-statistiken-ueber-kuenstliche-intelligenz-die-dich-vom-hocker-hauen-werden/>  
[Zugriff am 16 Juni 2018].
- Frankfurt School of Finance, 2017. *Master in Applied Data Science*. [Online]  
Available at: <https://www.frankfurt-school.de/home/programmes/master/data-science.html>  
[Zugriff am 18 Juni 2018].
- Golda, A., 2004. *Katedra Elektroniki AGH*. [Online]  
Available at: [http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html)  
[Zugriff am 18 Juni 2018].
- Huijskens, T., 2017. *Youtube: Thomas Huijskens - Bayesian optimisation with scikit-learn*. [Online]  
Available at: <https://www.youtube.com/watch?v=jtRPxRnOXnk>  
[Zugriff am 18 Juni 2018].
- Ivanov, S., 2017. *37 Reasons why your Neural Network is not working*. [Online]  
Available at: <https://blog.slavv.com/37-reasons-why-your-neural-network-is-not-working-4020854bd607>  
[Zugriff am 18 Juni 2018].
- Keskar, N. S., 2017. *arxiv.org*. [Online]  
Available at: <https://arxiv.org/pdf/1609.04836.pdf>  
[Zugriff am 18 Juni 2018].
- Laboratories, H., 2018. *Youtube: TensorFlow Tutorial #19 Hyper-Parameter Optimization*. [Online]  
Available at: <https://www.youtube.com/watch?v=oaxf3rk0KGM&feature=youtu.b&pp=desktop>  
[Zugriff am 18 Juni 2018].
- Maney, K., 2017. *Goldman Sacked: How Artificial Intelligence Will Transform Wall Street*. [Online]  
Available at: <http://www.newsweek.com/2017/03/10/how-artificial-intelligence-transform-wall-street-560637.html>  
[Zugriff am 18 Juni 2018].
- Pedersen, M. E. H., 2018. *ensorFlow Tutorial #19 Hyper-Parameter Optimization*. [Online]  
Available at: [https://github.com/Hvass-Labs/TensorFlow-Tutorials/blob/master/19\\_Hyper-Parameters.ipynb](https://github.com/Hvass-Labs/TensorFlow-Tutorials/blob/master/19_Hyper-Parameters.ipynb)  
[Zugriff am 18 Juni 2018].
- Reed, S. E., 2015. *Arxiv.org*. [Online]  
Available at: <https://arxiv.org/pdf/1412.6596.pdf>  
[Zugriff am 18 Juni 2018].
- Ruder, S., 2017. *An overview of gradient descent optimization algorithms*. [Online]  
Available at: <http://ruder.io/optimizing-gradient-descent/>  
[Zugriff am 18 Juni 2018].
- Schlenk, C. T., 2018. *Blackrock investiert in Münchner Scalable Capital*. [Online]  
Available at: <https://www.gruenderszene.de/allgemein/blackrock-scalable-capital>  
[Zugriff am 18 Juni 2018].
- Srivastava, N., 2014. *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. [Online]  
Available at: <http://jmlr.org/papers/volume15/srivastava14a.old/srivastava14a.pdf>  
[Zugriff am 18 Juni 2018].
- Tensorflow, 2018. *Tinker With a Neural Network Right Here in Your Browser..* [Online]  
Available at: <http://playground.tensorflow.com/>  
[Zugriff am 18 Juni 2018].
- ujjwalkarn, 2016. *A Quick Introduction to Neural Networks*. [Online]  
Available at: <https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/>  
[Zugriff am 18 Juni 2018].



**Vielen Dank für die  
Aufmerksamkeit!**