

# architektur SPICKER

Übersichten für die konzeptionelle Seite der Softwareentwicklung

**MEHR WISSEN IN KOMPAKTER FORM:**

Weitere Architektur-Spicker gibt es als kostenfreies PDF unter [www.architektur-spicker.de](http://www.architektur-spicker.de)

NR. 10

## IN DIESER AUSGABE

- Machine Learning eingeordnet
- Supervised, Unsupervised und Reinforcement Learning
- Typische Qualitätsziele und Kompromisse
- Entscheidungshilfen für ML-Projekte



# Machine Learning Lösungen entwerfen

Machine Learning (kurz ML) wird häufig mystifiziert. Tatsächlich eröffnet es ganz neue Möglichkeiten. Dabei unterscheiden sich Herangehensweise und Werkzeuge deutlich von klassischer Softwareentwicklung. Dieser Spicker führt unaufgeregt in das Thema ML ein und weist den Weg in eigene Experimente.



## Worum geht's?

- ➔ ML kennt verschiedene Spielarten: Wann wenden Sie welche an?
- ➔ ML-Vorhaben haben oft Experiment-Charakter, doch gibt es typische Beteiligte, Phasen und Werkzeuge. Wie sieht die Anatomie eines ML-Projekts aus?
- ➔ Sie haben Daten für Ihr Vorhaben identifiziert. Wie finden Sie Anwendungen für ML?
- ➔ Die Explorationsphase ist entscheidend für den Erfolg jedes ML-Vorhabens. Wie gehen Sie dort genau vor?

## Software 2.0: Traditionelle Geschäftslogik vs. Machine Learning



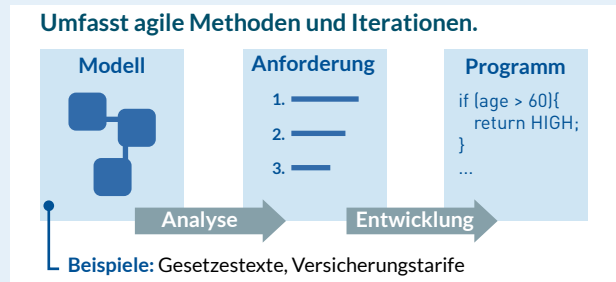
### ML als alternativer Ansatz der Softwareentwicklung

Die zwei Herangehensweise gegenübergestellt:

#### Traditionelle Software-Entwicklung Analyse-getrieben

Wann anwenden?

- Die Domäne ist zumindest so gut verstanden, dass man Anforderungen sinnvoll als Regeln aufschreiben kann.
- Das Problem ist analytisch lösbar und für interessante Größen berechenbar



#### Machine Learning Daten-getrieben

Wann anwenden?

- Sie haben Daten oder können sie erzeugen.
- Ihnen steht kein analytischer Zugang zur Problemstellung zur Verfügung.
- Sie können mit Unsicherheiten und Fehlern leben.

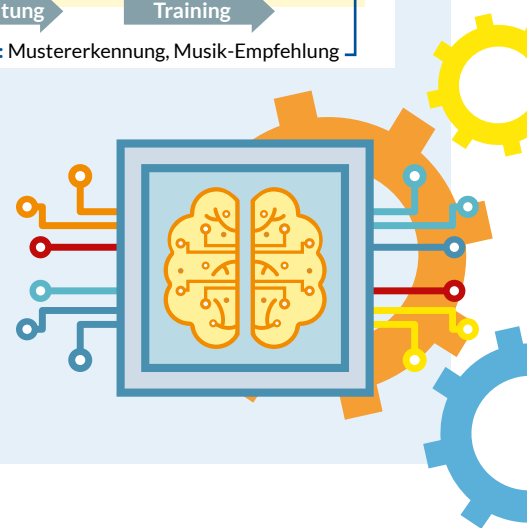
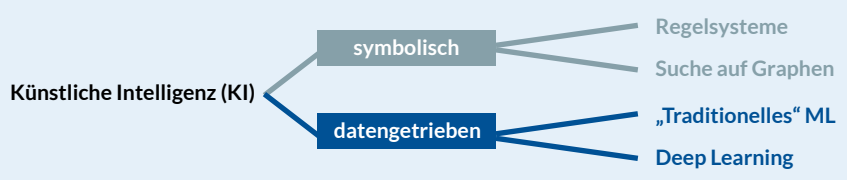


➔ Oft sind beide Ansätze mit ihren jeweiligen Vor- und Nachteilen möglich.



### ML eingeordnet

Maschinelles Lernen ist ein Teilgebiet der Künstlichen Intelligenz.



# Machine Learning Wegweiser



## Anatomie eines Projekts

Ein typisches ML Projekt zerfällt in drei Phasen. Die erste unterscheidet sich signifikant von traditioneller Softwareentwicklung. Daher setzen wir in diesem Spicker den Schwerpunkt auf diese Phase.



Phase I Exploration			Phase II Umsetzung			Phase III Betrieb
<b>Problem strukturieren</b>	<b>Daten vorbereiten</b>	<b>Modelle evaluieren</b>	<b>ML-Pipeline aufbauen</b>	<b>Integration in System</b>	<b>Monitoring aufsetzen</b>	<b>ML-Lösung überwachen</b>
Klären von Einsatzidee, Zielsetzung, Nutzen, Projektparametern (nötige Genauigkeit, Erklärbarkeit) etc. Projekt aufsetzen.	Beschaffung der Daten. Sichtung, Visualisierung, Prüfung und Säuberung der Daten. Auftrennung in Train-, Test- und Validierungsset	Baseline etablieren und verschiedene Modelle prüfen. Features wählen und Lernstrategie inkl. Hyperparameter festlegen.	Von Jupyter-Notebooks zu produktivem Code, inkl. wiederholbarer, automatisierter Pipeline, Datenprüfungen und (Unit-)Tests.	Zusammenspiel mit der bestehenden Lösung implementieren, API schaffen, Fallbacks für den Fehlerfall etablieren	Logging und Monitoring für die ML-Lösung aufsetzen, inkl. Laufzeitinformationen, Performance des Modells, Trendmetriken etc.	Metrikdaten sammeln und aggregieren, „Drops“ monitoren, Datenbatches sammeln und Qualitätsprüfungen durchführen.
PO, RE, ML-Experte	Data Scientist (ML-Experte)	ML-Experte (Data Scientist)	ML-Experte SW-Entwickler	SW-Entwickler	Entwickler/Ops, ML-Experte	Ops ML-Experte
ppt, xls, Mindmaps	Jupyter Notebooks	Jupyter Notebooks	Code, Tests, Config, Spec.	API, Code, Tests, Doku	Config, Metriken, Dashboards	Dashboards, Alerts, Logs

Legende: Rollen Artefakte



## Elementare Lern-Strategien

Maschinelles Lernen wird häufig in drei unterschiedliche Arten von Strategien unterteilt. Die Auswahl treffen Sie anhand von Voraussetzungen an die Daten oder die Lernsituation.

Überwachtes Lernen ist in der Praxis bei weitem am häufigsten vertreten.

	<b>Überwachtes Lernen</b> (Supervised Learning)	<b>Unüberwachtes Lernen</b> (Unsupervised Learning)	<b>Verstärkendes Lernen</b> (Reinforcement Learning)
<b>Was ist das?</b>	Das Modell imitiert Verhalten, das es durch viele Beispiele von Ein-/Ausgabe-Paaren gelernt hat.	Das Modell lernt ohne weitere Hinweise automatisch Zusammenhänge, Muster und Ausreißer aus Daten.	Anstatt bestehende Daten zu nehmen, werden diese durch Experimente erzeugt. Dabei muss durch eine Belohnung angegeben werden, was ein guter Ausgang eines Experiments ist.
<b>Voraussetzung für die Anwendung</b>	Sie haben Paare von klar definierten Ein- und Ausgaben, d.h. Sie müssen die richtigen Antworten kennen.	Sie haben Fälle von ähnlicher Natur, aber keine richtigen Antworten. Was das trainierte Modell als Ausgabe hat, ist flexibler als beim überwachten Lernen.	Sie können gefahrlos Experimente in großer Anzahl in der echten Welt machen oder diese angemessen simulieren.

Ein Problem kann auf unterschiedliche Arten angegangen werden, je nachdem wie man es dreht und wendet.

# Eine Anwendung auswählen



## Typische Anwendungsgebiete

Lassen Sie sich zur Auswahl einer Anwendung aus Ihrem Arbeitsbereich inspirieren:

### Zeithorizont

- Bereits im praktischen industriellen Einsatz
- Forscher können das schon
- Nicht absehbar



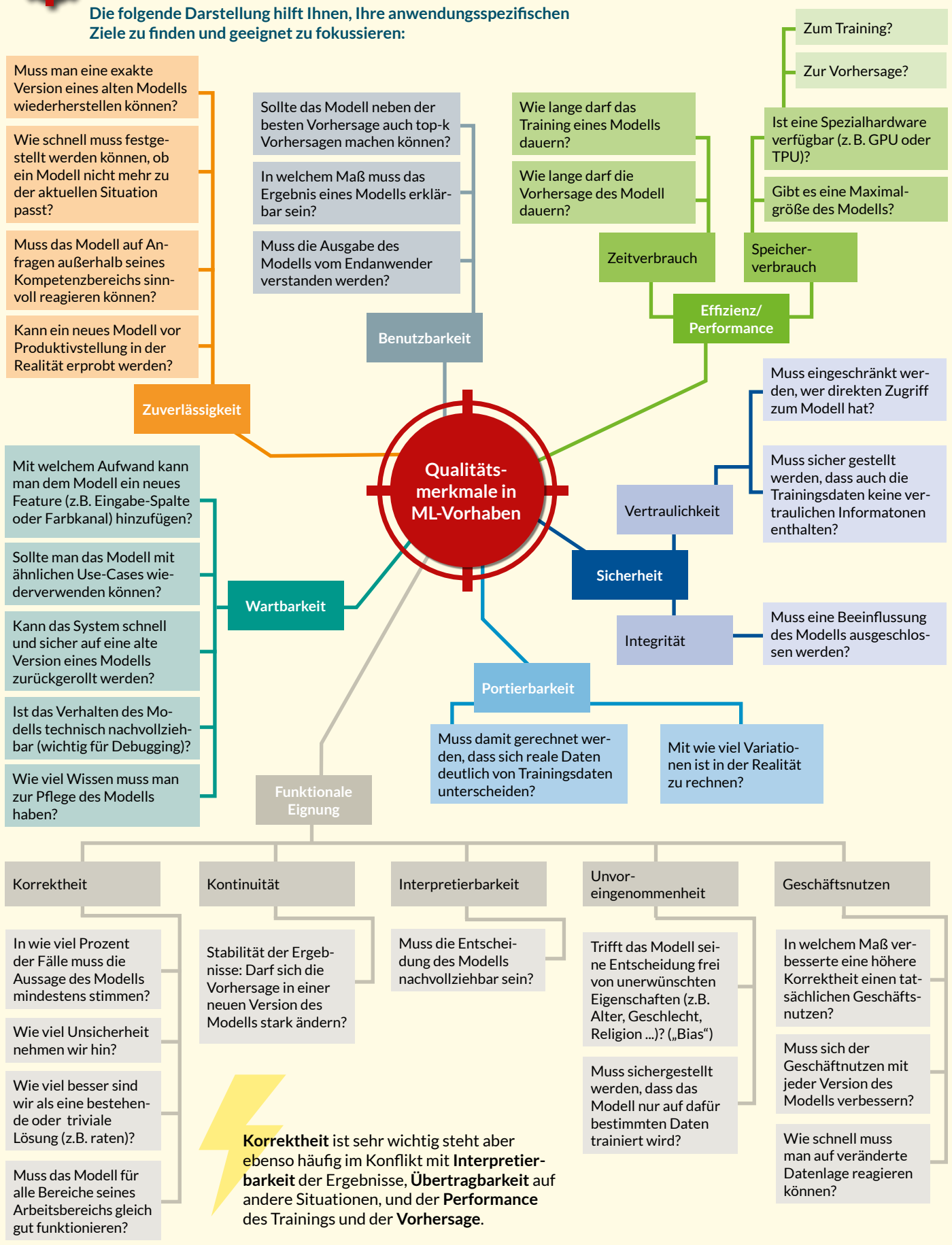
Zentrale Frage sollte immer sein, welches Ziel erreicht oder - konkreter - welche Geschäfts-Metrik verbessert soll. Z.B. „Anteil junger Kunden erhöhen“, „Bearbeitungszeit pro Vorgang verkürzen“, „Welches Produkt würden meine Kunden noch haben wollen“?

# Anforderungen schärfen



## Qualitätsziele für Ihre ML-Anwendung

Die folgende Darstellung hilft Ihnen, Ihre anwendungsspezifischen Ziele zu finden und geeignet zu fokussieren:



# Den richtigen Ansatz finden



## Checkliste für ML Experimente

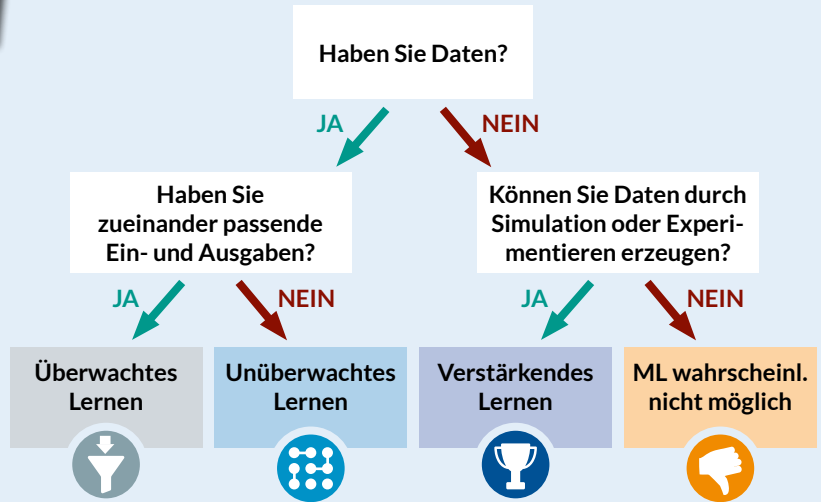
- ✓ Maßgebliche Stakeholder kennen die realistischen Möglichkeiten von ML.
- ✓ Die Anwendung für ML und der gewünschte Geschäftsnutzen sind realistisch definiert.
- ✓ Wichtigste Qualitätsmerkmale sind festgelegt (top-k).
- ✓ Konflikte in Qualitätsmerkmalen sind aufgedeckt und in Kompromissen aufgelöst.
- ✓ Daten sind verfügbar oder können generiert werden.
- ✓ Die Lernstrategie ist ausgewählt.
- ✓ Der Explorations-Prozess für ein ML-Projekt ist im Unternehmenskontext verstanden.
- ✓ Eine sogenannte Baseline\* ist definiert. Sie bietet eine einfache Lösung des Problems als Vergleichspunkt.

\*Die Baseline kann im Extremfall „Raten“ sein oder auch eine bereits bestehende Lösung, die Sie verbessern wollen



## Lernstrategie auswählen

Anhand der Frage ob und welche Art von Daten Ihnen zur Verfügung stehen, leitet Sie dann der Entscheidungsbaum zur Auswahl der Lern-Strategie.



## Explorationsprozess

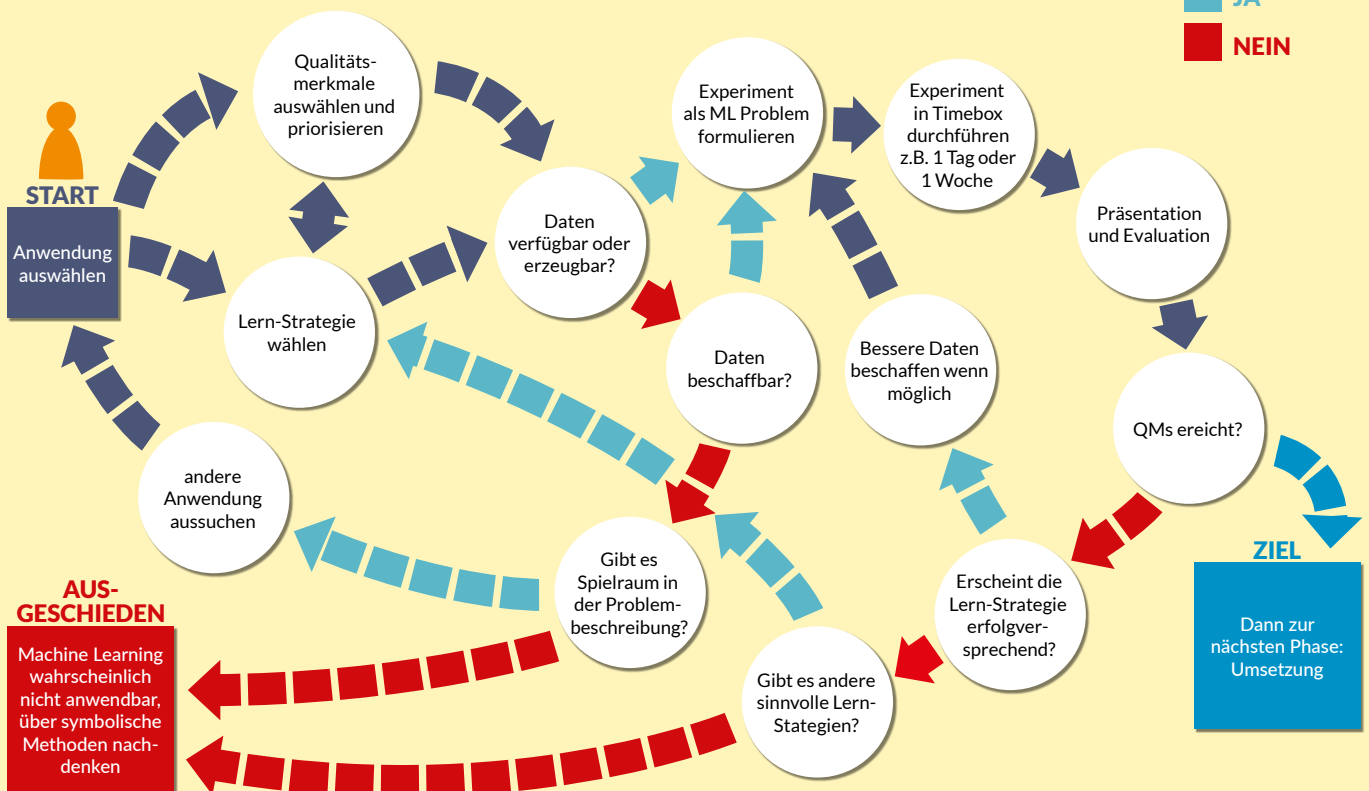
Das Vorgehen in der ersten Phase eines Machine Learning Projekts ähnelt weniger einem Software-Entwicklungsprozess als einem Forschungsprojekt. Dementsprechend muss als ein möglicher Ausgang einkalkuliert werden, dass der gewählte Ansatz nicht zielführend oder Machine Learning gar nicht anwendbar ist. Am ehesten ähnelt dieses Vorgehen einem agilen Prozess, bei dem möglichst schnell in kurzen Iterationen festgestellt werden soll, ob ein weiteres Vorgehen sinnvoll ist und wenn ja wie es aussehen soll.

**Aufgabe:** Bewegen Sie sich unter Zuhilfenahme der vorherigen Erklärungen durch diesen Prozess. Treffen Sie Annahmen für den Ausgang von Experimenten.

Legende:

■ JA

■ NEIN



# Machine Learning in Ihrem Projekt



## Typische Herausforderungen in ML

### Organisatorische Herausforderung

Das Vorgehensmodell für Machine Learning beißt sich mit etablierten Prozessen.



### Umgang damit ...

Der Explorationsprozess beschreibt nur die Phase I der Anatomie auf → Seite 2. Dieser ist einer produktiven Implementierung vorgelagert. Ab Phase II kann wieder klassisch vorgegangen werden.

Daten sind vorhanden, man bekommt aber keinen Zugriff, z.B. auf sensible Informationen.



Schaffen Sie ein Bewusstsein für die Notwendigkeit von Daten für Machine Learning durch Illustrationen wie auf → Seite 1. Arbeiten Sie mit den Daten die Sie bekommen.

Die Unternehmenskultur erlaubt keine Fehlschläge.



Machen Sie von vorn herein transparent, dass scheiternde Experimente etwas positives sind, weil man früh einen unpassenden Ansatz ausschließen kann. Dies illustriert der Prozess auf → Seite 5.

Selbst Erfolg kann problematisch sein und vorherige Arbeitsergebnisse diskreditieren.



ML ist für die meisten Unternehmen neu und kann häufig die Ergebnisse klassischer Programmierung deutlich übertreffen. Das ist aber keine Schande.

Die Erwartungen an ML sind unrealistisch. Wirklich nützliche Lösungen sind oft unspektakulär.



Machen Sie alle Stakeholder mit möglichen Anwendungen wie auf → Seite 3 vertraut. Binden Sie diese bei der Anwendungsfindung ein.

### Methodische Herausforderung

Daten sind vorhanden, passen aber nicht zur Realität, entweder von Anfang an oder schleichend.



### Umgang damit ...

Etablieren Sie Prozesse und Monitoring, die dies offensichtlich machen (mehr dazu in „Anatomie eines ML Projekts“ auf → Seite 2).

Man hat keinen Maßstab um abzuschätzen wie gut das Modell funktioniert oder der Maßstab ist zu schwach?



Finden Sie eine Baseline wie in der Checkliste auf → Seite 5 beschrieben.

Es wird aufwändig mit einem Ansatz weiter gearbeitet, obwohl er nicht per Experiment bestätigt wurde.



Halten Sie sich an den Explorationsprozess wie er auf → Seite 5 beschrieben ist.

Qualitätsmerkmale wurden nicht definiert, sodass nicht festgestellt werden kann, ob ein Experiment erfolgreich war.



Definieren Sie Ihre top-5 Qualitätsmerkmale und achten Sie auf Wechselwirkungen wie auf → Seite 4 gezeigt.

## Weitere Informationen



### Online-Ressourcen

- **Online-Kurs** „Elements of AI“ (Englisch, Deutsch, ...), University of Helsinki.  
<https://course.elementsofai.com/de/>
- **Video:** Oliver Zeigermann: „Deep Learning Crash Course“  
<https://www.manning.com/livevideo/deep-learning-crash-course>



### Die Autoren dieses Spickers

- Oliver Zeigermann  
Kontakt: [oliver.zeigermann@embarc.de](mailto:oliver.zeigermann@embarc.de)  
Twitter: @DJCordhose
- Stefan Zörner  
Kontakt: [stefan.zoerner@embarc.de](mailto:stefan.zoerner@embarc.de)  
Twitter: @StefanZoerner



### Print

- Chi Nhan Nguyen, Oliver Zeigermann  
„Machine Learning – kurz & gut: Eine Einführung mit Python, Pandas und Scikit-Learn“, O'Reilly Verlag 2018

Wir freuen uns auf Ihr Feedback: [spicker@embarc.de](mailto:spicker@embarc.de)

<https://architektur-spicker.de>



<https://www.embarc.de>  
[info@embarc.de](mailto:info@embarc.de)



<https://www.sigs-datacom.de>  
[info@sigs-datacom.de](mailto:info@sigs-datacom.de)