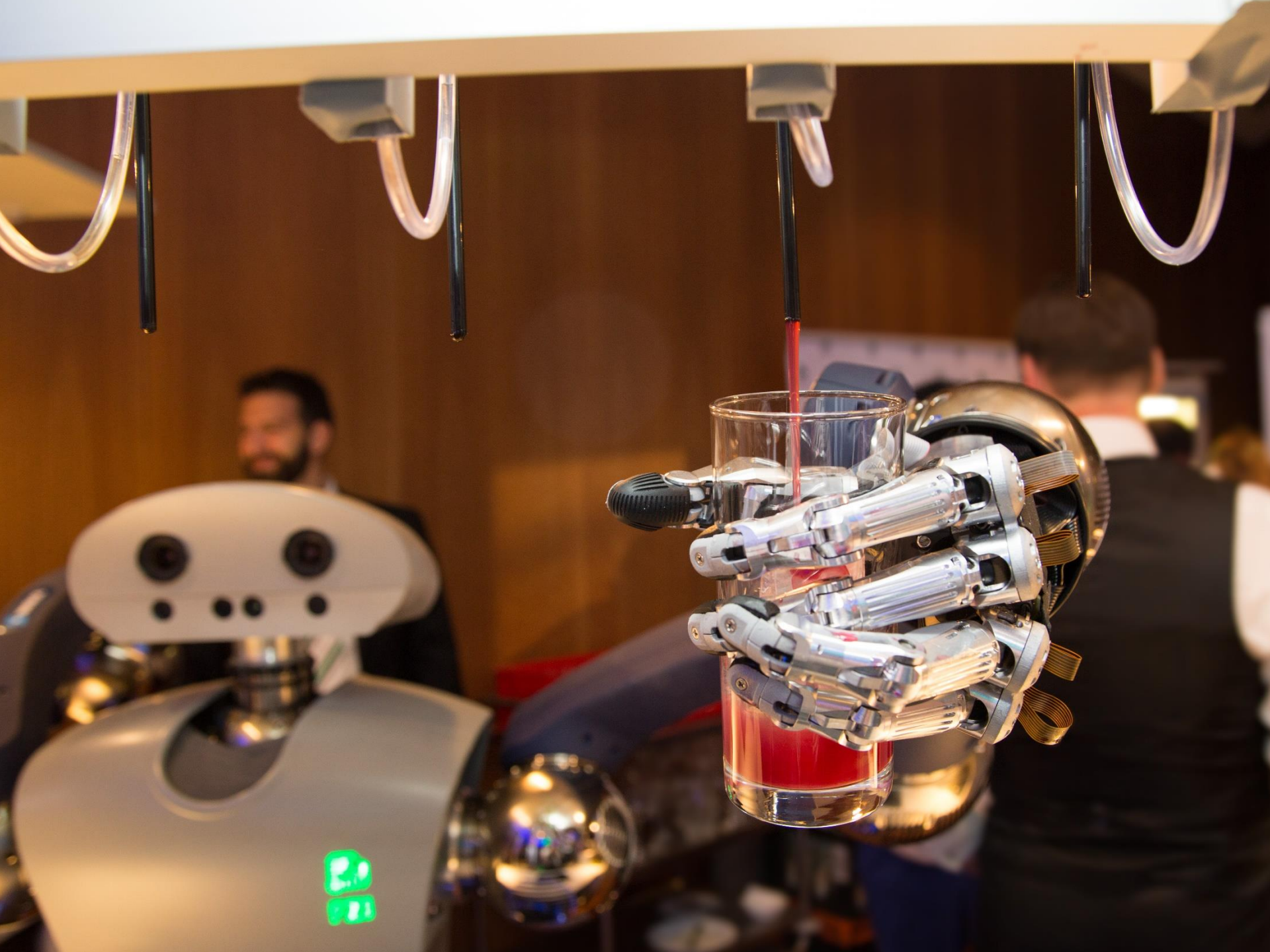
DIZ-Konfigurator:
Eine Fallstudie zur Empfehlung von Methoden und Modellen zur automatischen Stammdatenklassifikation mittels Semantik

Forschungsbereich IPE, Abteilung Wissensmanagement
Ignacio Traverso-Ribón, Dr. Benedikt Kämpgen

FZI FORSCHUNGSZENTRUM INFORMATIK



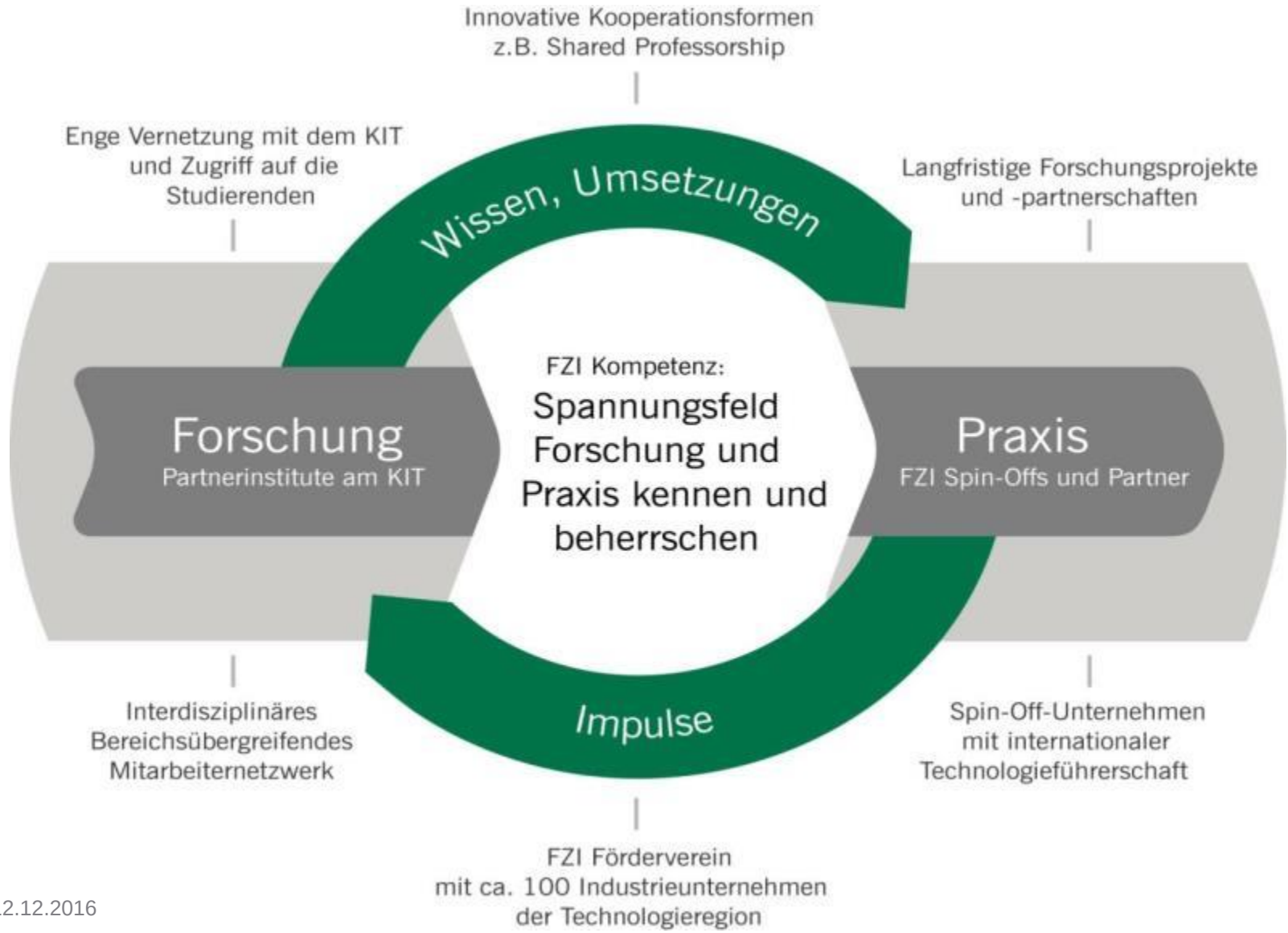
Agenda

- Überblick FZI
- DIZ-Projekt "Konfigurator"
- Fallstudie: Automatische Stammdatenklassifikation mit IFCC GmbH
- Zusammenfassung

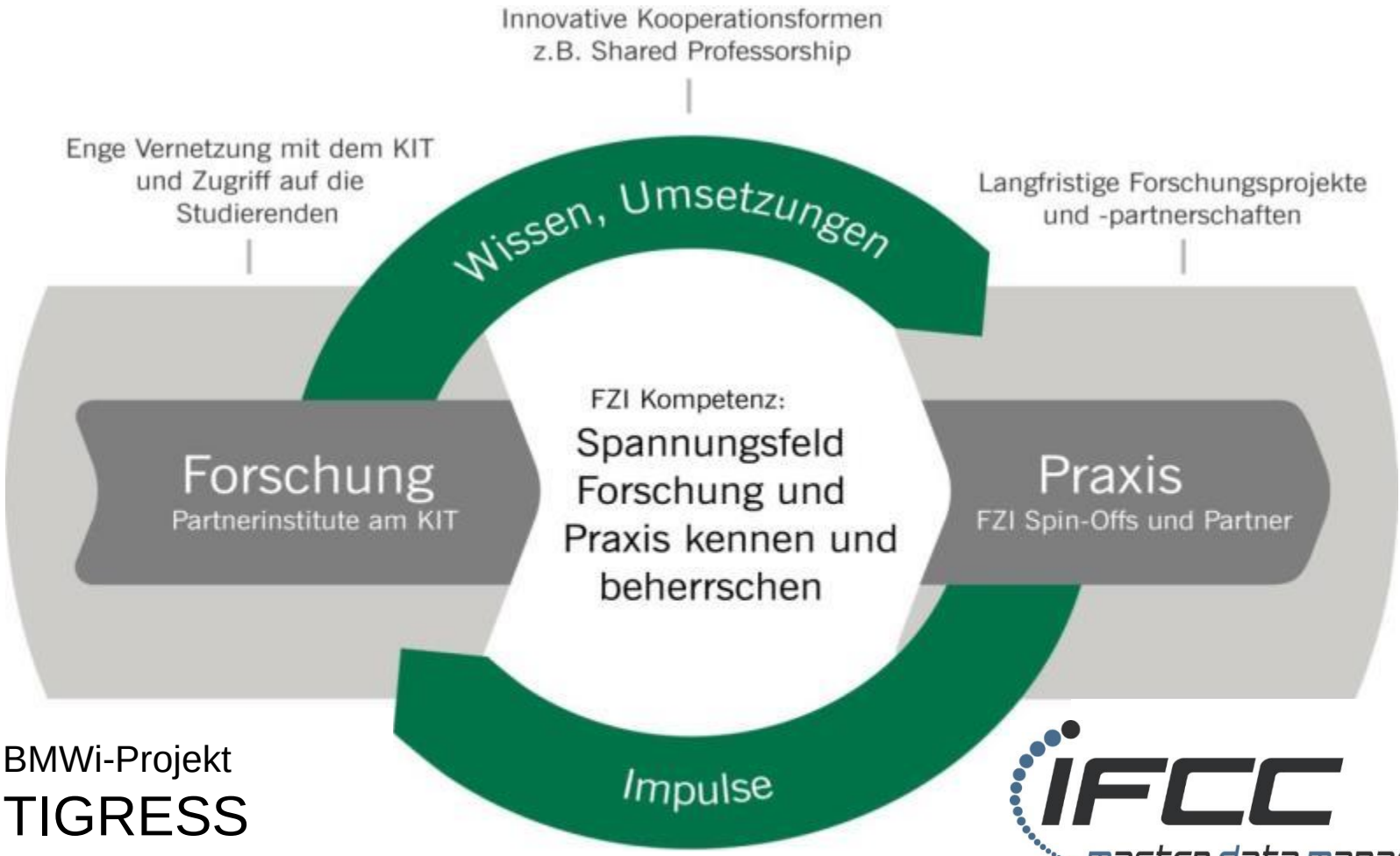
Die Einrichtung FZI

- Unabhängige gemeinnützige Stiftung des bürgerlichen Rechts für Anwendungsforschung in der Informatik
- Mitglied der Innovationsallianz Baden-Württemberg und Technologieregion Karlsruhe
- Innovationsdrehscheibe in Baden-Württemberg im Bereich Informationstechnologie
- Innovationspartner des KIT im Bereich IT

FZI Erfolgsrezept:
Starke Vernetzung und Interdisziplinarität



FZI Erfolgsrezept:
Starke Vernetzung und Interdisziplinarität



BMWi-Projekt
TIGRESS

IFCC master.data.management

DIZ DIGITALES INNOVATIONS ZENTRUM

CyberForum
HIGHTECH. UNTERNEHMER. NETZWERK.

FZI Forschungsbereich
Information Process Engineering

Direktoren





Prof. Dr.-Ing. Kai Furmans

Prof. Dr. Stefan Nickel

Prof. Dr. Rudi Studer

Prof. Dr. York Sure-Vetter

Prof. Dr. Christof Weinhardt

Leiter Shared Research Group

„Corporate Services and Systems“

Prof. Dr. Thomas Setzer

Bereichsleiter

Dr.-Ing. Iris Heckmann

Forschungsbereich IPE

Intelligente Informationslogistik und adaptive Infrastrukturen für die vernetzte Welt

Abteilungsleiter

Dr. Boris Amberg

Dr.-Ing. Benedikt Kämpgen

Dr.-Ing. Anne Meyer

Dr. Alexander Schuller

Dr. Stefan Zander





Agenda

- Überblick FZI
- **DIZ-Projekt "Konfigurator"**
- Fallstudie: Automatische Stammdatenklassifikation mit IFCC GmbH
- Zusammenfassung

DIZ-Teilprojekt "Konfigurator"

Motivation

We believe it's possible to automate certain aspects of data science, and specifically to have machines learn from prior example how to construct new models.

DARPA Goes "Meta" with Machine Learning for Machine Learning

http://www.darpa.mil/news-events/2016-06-17

Defense Advanced Research Projects Agency

Motivation

Typische **Entscheidungsprobleme** von Unternehmen:

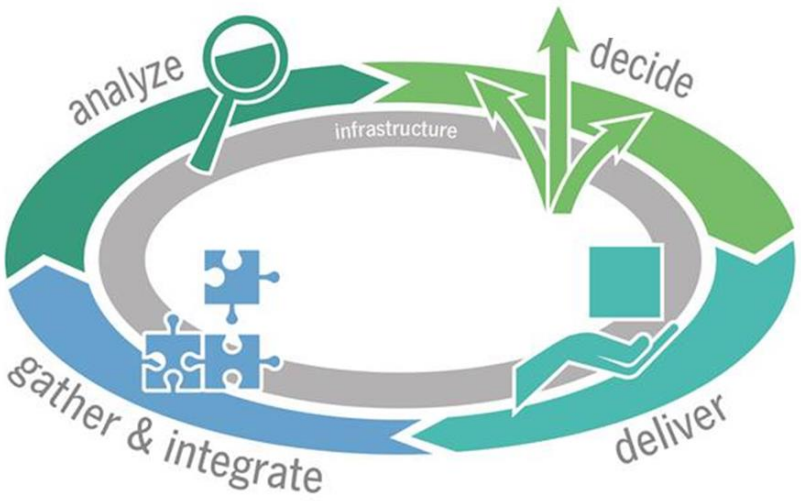
- Optimierung
  - Z.B. Logistik-Firma: Welches ist die schnellste Route? Wie ist ein LKW zu beladen, um das Ausliefern zu vereinfachen?
- Simulation
  - Welche Konsequenzen hat eine Entscheidung? Wie verlässlich ist die Simulation?
- Klassifikation
  - Z.B. Kunden-Meinungen: Welche Produktaspekte werden als positiv oder als negativ wahrgenommen?
- Clustering
  - Wer sind Kunden? In welche Gruppen können Kunden unterteilt werden? Welche Merkmale hat jede Gruppe?
- Vorhersage
  - Wie können Kunden antizipiert werden? Wie kann eine optimale Lagerhaltung vorhergesagt werden?

Motivation (2)

Wir lösen diese Probleme in einem Zyklus der **datengetriebenen Entscheidungsunterstützung**:

How to gather and integrate all relevant data about the problem at hand?

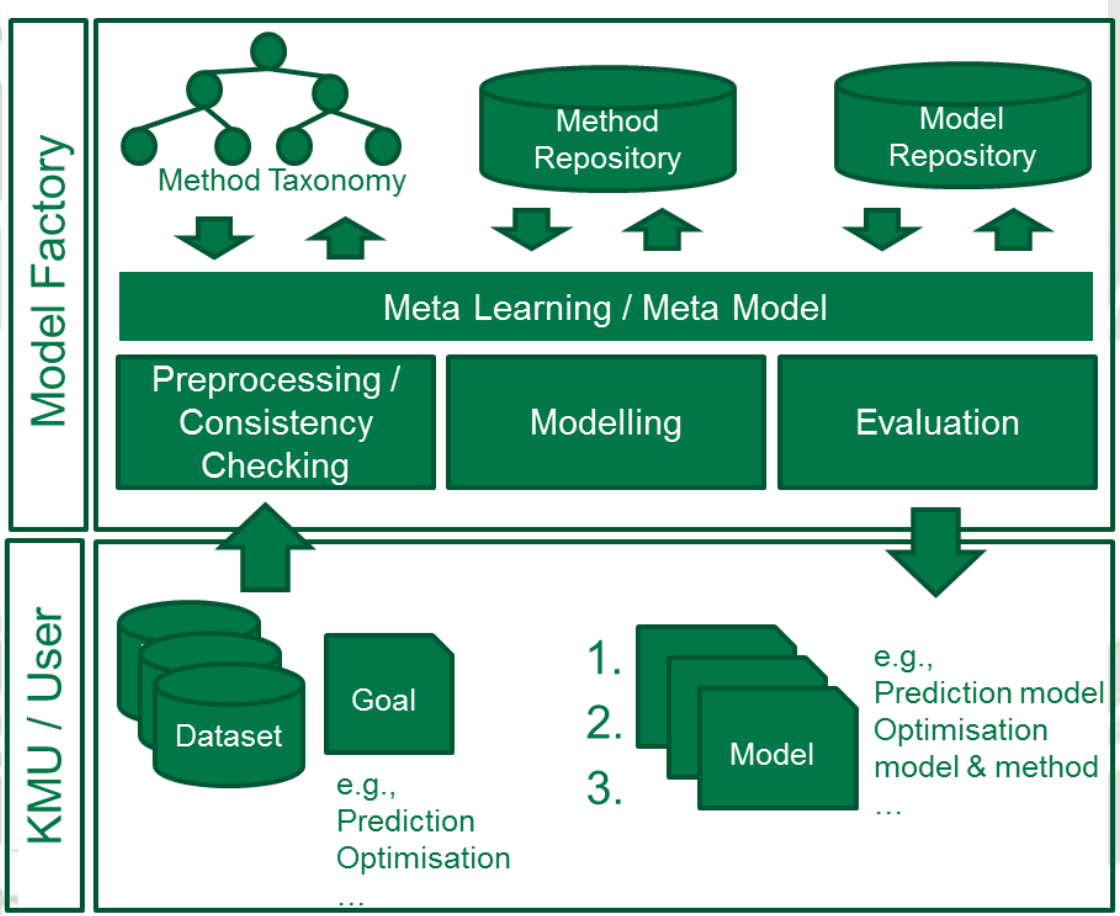
Welche Modelle sind geeignet zum Lösen des Problems?



Wie Daten vorverarbeiten und bewerten? Welche Daten berücksichtigen?

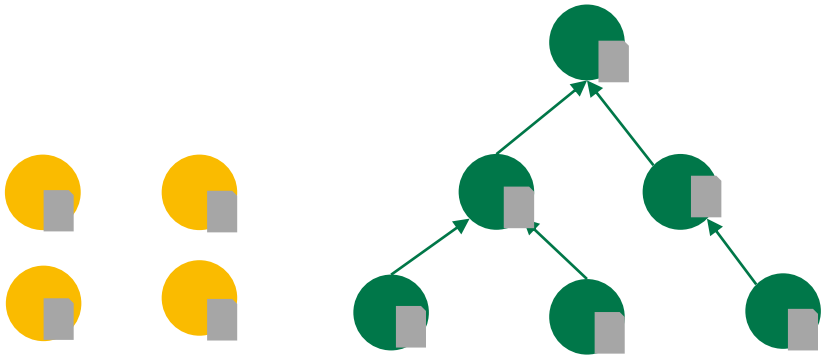
Wie kann die Lösung genutzt und daraus gelernt werden?

DIZ-Konfigurator: Architektur

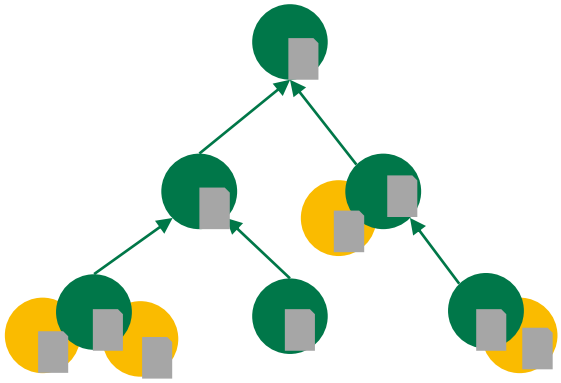


DIZ-Konfigurator für Klassifikation

Problem

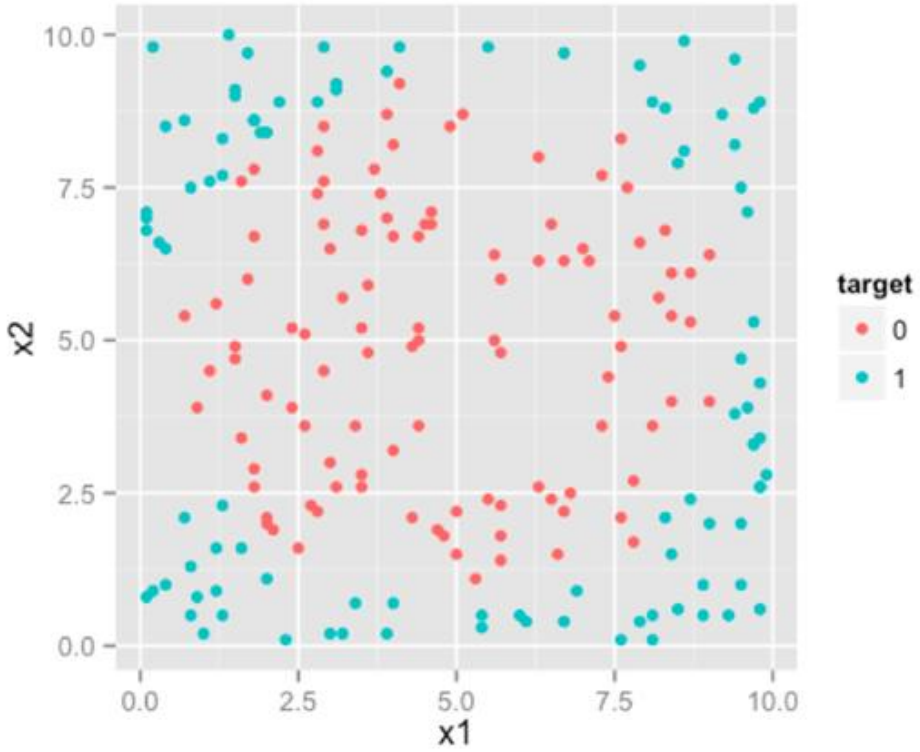


Instanzen Klassifikationssystem



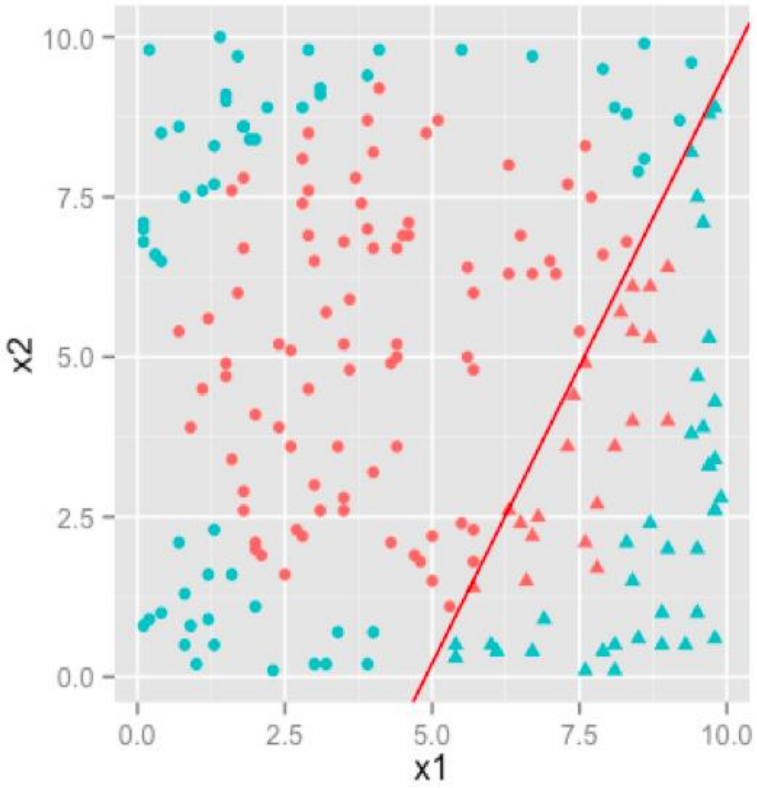
Klassifikation

Problembeispiel Binäre Klassifikation



Fiktiver Feature-Raum (http://www.edvancer.in)

Klassifikation durch "Logistic Regression"



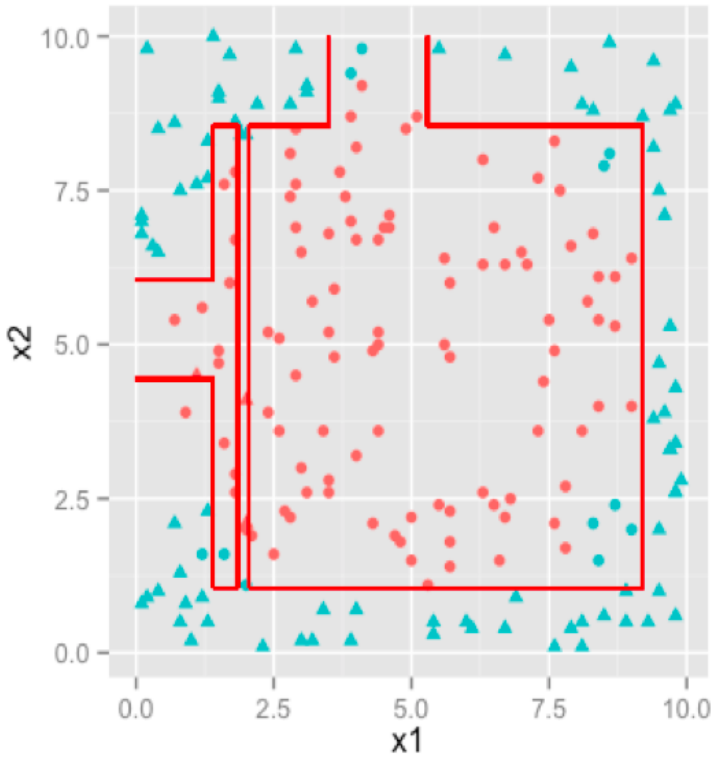
Logistic Regression:

Klassifikatoren trennen den Feature-Raum in zwei Teile (rote Linie).

Problem: Oft sind Feature-Räume nicht linear trennbar.

Fiktiver Feature-Raum (http://www.edvancer.in)

Klassifikation durch "Entscheidungsbaum"



Fiktiver Feature-Raum

http://www.edvancer.in

Entscheidungsbäume:

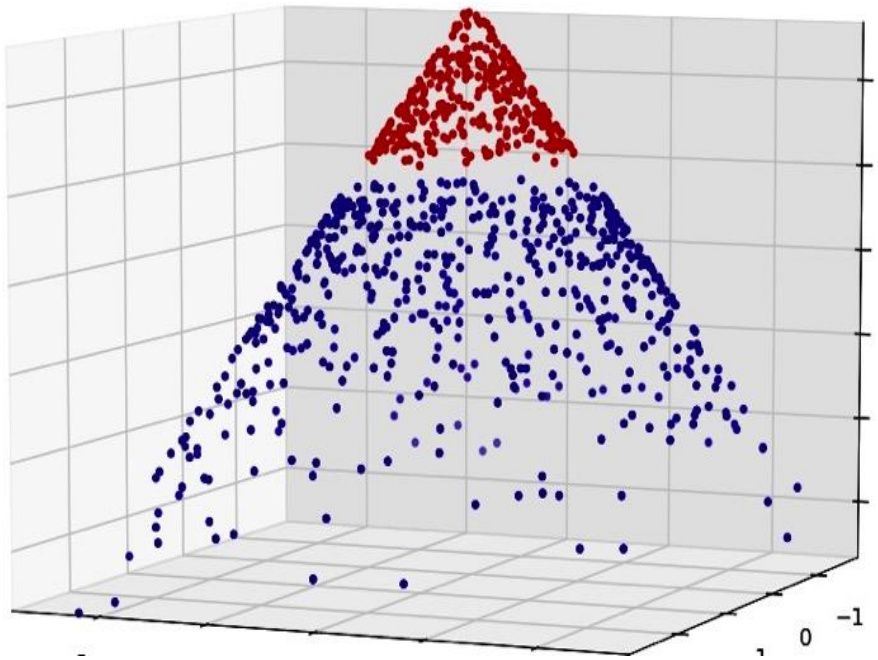
Klassifikatoren trennen den Feature-Raum in zwei Teile durch mehrere Linien (schematisch).

Auch anwendbar bei nicht-linear-trennbaren Feature-Räumen.

Problem:

- langsamer in der Ausführung als Logistic Regression
- höheres Risiko des Overfittings (lässt sich durch Lernen mehrerer Entscheidungsbäume – Random Forest – reduzieren)

Klassifikation durch "Kernel Function"



Fiktiver Feature-Raum
(http://www.edvancer.in)

Kernel Funktionen:

Überführen Instanzen in einen höherdimensionalen Raum, in dem die Instanzen linear trennbar sind.

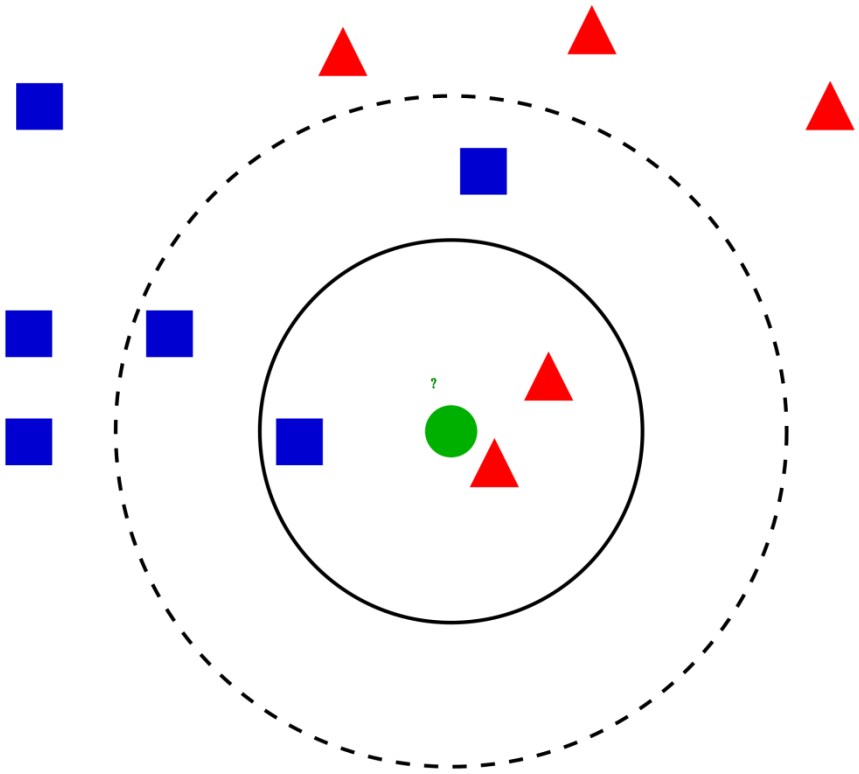
Auch anwendbar bei nicht-linear-trennbaren Feature-Räumen.

Geringeres Risiko von Overfitting.

Problem:

- langsamer in der Ausführung als Logistic Regression

Klassifikation durch "K-Nearest-Neighbor"



Wikimedia Commons

K-Nearest-Neighbor-Algorithmus:

Kein Training notwendig.

Problem:

- die Klassifikation ist vergleichsweise langsam (je nach verwendetem Ähnlichkeitsmaß)

"Konfigurator" für Klassifikation

- Kriterien zur Auswahl von Klassifikationsmethoden
  - Anzahl an Klassen im Klassifikationssystem
  - Häufigkeit von Trainingsinstanzen pro Klasse
  - Typen der Merkmale (Features) der Instanzen (numerisch, textuell)
  - Trennbarkeit der Klassen im Feature-Raum
  - Qualität der Instanz-/Klassenbeschreibungen

- Geeignete Methode je nach Ausprägung der Kriterien

Kriteria	Linear Regression	Entscheidungs-bäume	SVM	KNN
Anzahl an Klassen	wenig	wenig	mittel	mittel bis viel
Trainingsdaten pro Klasse	mittel bis viel	Viel	mittel bis viel	wenig bis mittel
Features	numerisch	beide	numerisch	beide
Trennbarkeit	lineare	lineare	nicht lineare	nicht lineare
Qualität	mittel-hoch	hoch	niedrig bis mittel	hoch

Aber: Weitere Kriterien für Klassifikationsprobleme

- Häufigkeitsverteilung der Instanzen pro Klasse. Je niedriger die Varianz, desto besser für das Lernen eines Modells.
- Häufigkeitsverteilung der Wörter der Instanzbeschreibungen pro Klasse. Je niedriger die Varianz, desto besser für das Lernen eines Modells.
- Häufigkeitsverteilung der Ähnlichkeitswerte der Instanzbeschreibungen pro Klasse. Je höher die Ähnlichkeit, desto besser für das Lernen eines Modells.
- Häufigkeit von Stopp-Wörtern in den Beschreibungen. Je niedriger, desto besser für das Lernen eines Modells; dabei lassen sich Stopp-Wörter zu Verbesserung des Lernens entfernen.
- Häufigkeitsverteilung der erkannten Entitäten pro Instanz(-beschreibung) und Klasse(-nbeschreibung). Je mehr erkannte Entitäten, desto besser für das Lernen einen Modells (insbesondere für Sprachunabhängigkeit).

Verbesserte Klassifikation mittels Semantik

- Möglichkeit 1: "Embeddings durch Neuronale Netze"
  - Die Repräsentation von Instanzen durch Embeddings/Vektorrepräsentationen, die die Bedeutung der enthaltenen Wörter anhand eines großen Textkorpusses beschreiben.
  - Tool: Word2Vec / Doc2Vec [1]

- Möglichkeit 2: "Ähnlichkeiten auf Basis semantischer Annotationen"
  - Die Ähnlichkeitsmessung anhand von semantischen Beziehungen zwischen Instanzen und Klassen.
  - Tool: GADES [2]

[1] https://deeplearning4j.org/word2vec

[2] Ignacio Traverso Ribón, Maria-Esther Vidal, Benedikt Kämpgen, York Sure-Vetter: GADES: A Graph-based Semantic Similarity Measure. SEMANTICS 2016: 101-104

Agenda

- Überblick FZI
- DIZ-Projekt "Konfigurator"
- **Fallstudie: Automatische Stammdatenklassifikation mit IFCC GmbH**
- Zusammenfassung

Fallstudie zur automatischen Stammdatenklassifikation mit IFCC GmbH

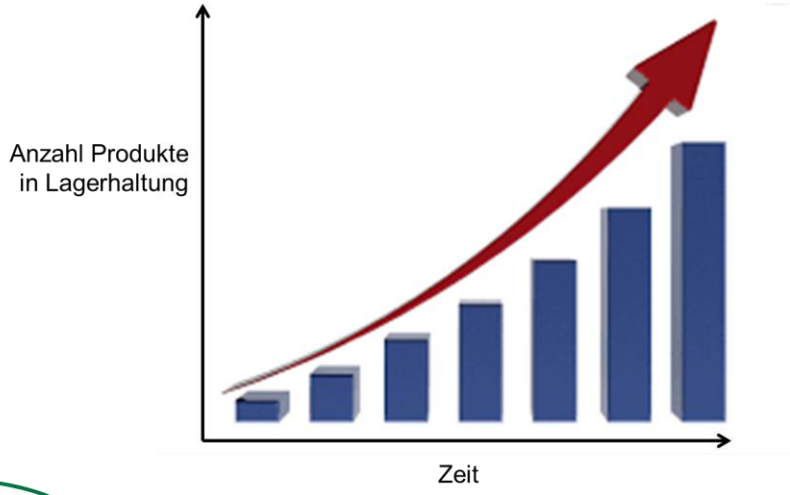
Definition "Master Data Management"

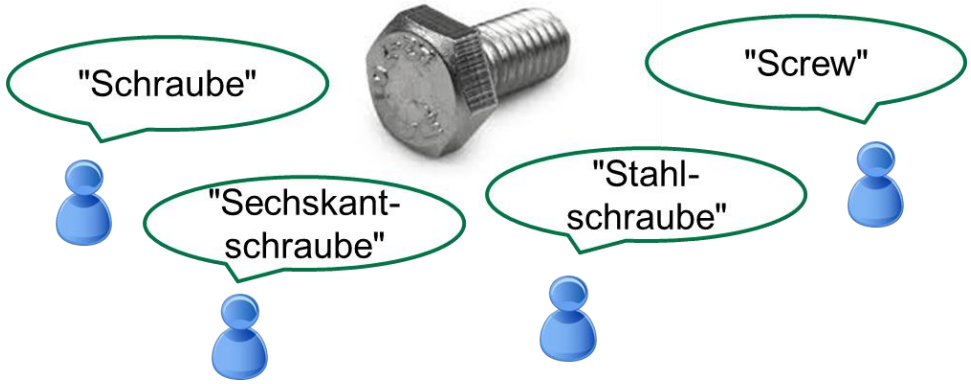
- Master Data = Stammdaten
  - Daten, die über einen längeren Zeitraum unverändert bleiben
  - Enthalten Informationen, die in gleicher Weise immer wieder benötigt werden (Referenzdaten)

- Master Data Management = Pflege von Stammdaten
  - Wichtig für IT-Infrastruktur und Transaktionen eines Unternehmens
  - Bspw. Warenwirtschaftssystem, Produktionsplanungs- und Steuerungssystems

Motivation der Pflege von Stammdaten

Anstelle vorhandene Produkte zu verkaufen oder für Ersatzteile herzunehmen, steigt ihre Anzahl mit der Zeit und damit die Lagerungskosten an.





Probleme

- Je nach Kontext wird unterschiedliche Detailstufe verlangt (Schraube vs. Sechskantschraube)
- Unterschiedliche Sprachen
- Synonyme
- …

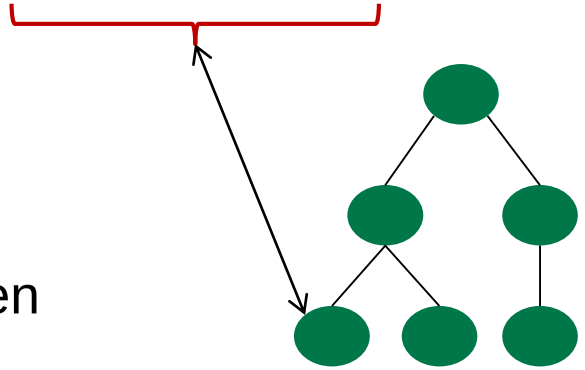
Häufiges Problem: Stammdaten effizient digital zu dokumentieren, auszutauschen und zu analysieren (z.B. für Suche, Übersicht)

Klassifikation von Stammdaten

- Klassifikationssysteme wie eCl@ss erleichtern solche Problematiken durch die Einordnung von Stammdaten in Klassen

SD	LD	Code	CodeDesc
VORLEGIERUNGSMETALL-ZN1 99,99%	ZINK 99,99% HÜTTENZINK IN FORM VON STÜCKEN IN CA. 1 KG GESCHNITTEN, GEM. EINKAUFSSPEZIFIKATION EKS-ASW-000030; REVISION 3, GÜLTIG AB DEM 06.01.2012	38150401	Zink

Beispiel für Stammdatum mit Beschreibung und Klassenzugehörigkeit



- Aber: Hoher manueller Aufwand, um die Einordnung durchzuführen und aktuell zu halten

- Firmen wie IFCC GmbH unterstützen diesen Prozess

Automatische Stammdatenklassifikation bei IFCC GmbH

eCl@ss

"verlinkt"

Masterklas-
sifikation

"unterstützt durch Trainingsdaten"

Kunden-
stammdaten

Input

Automatische
Klassifikation

Output

Klassifizierte
Kunden-
stammdaten

p1 p2 p3

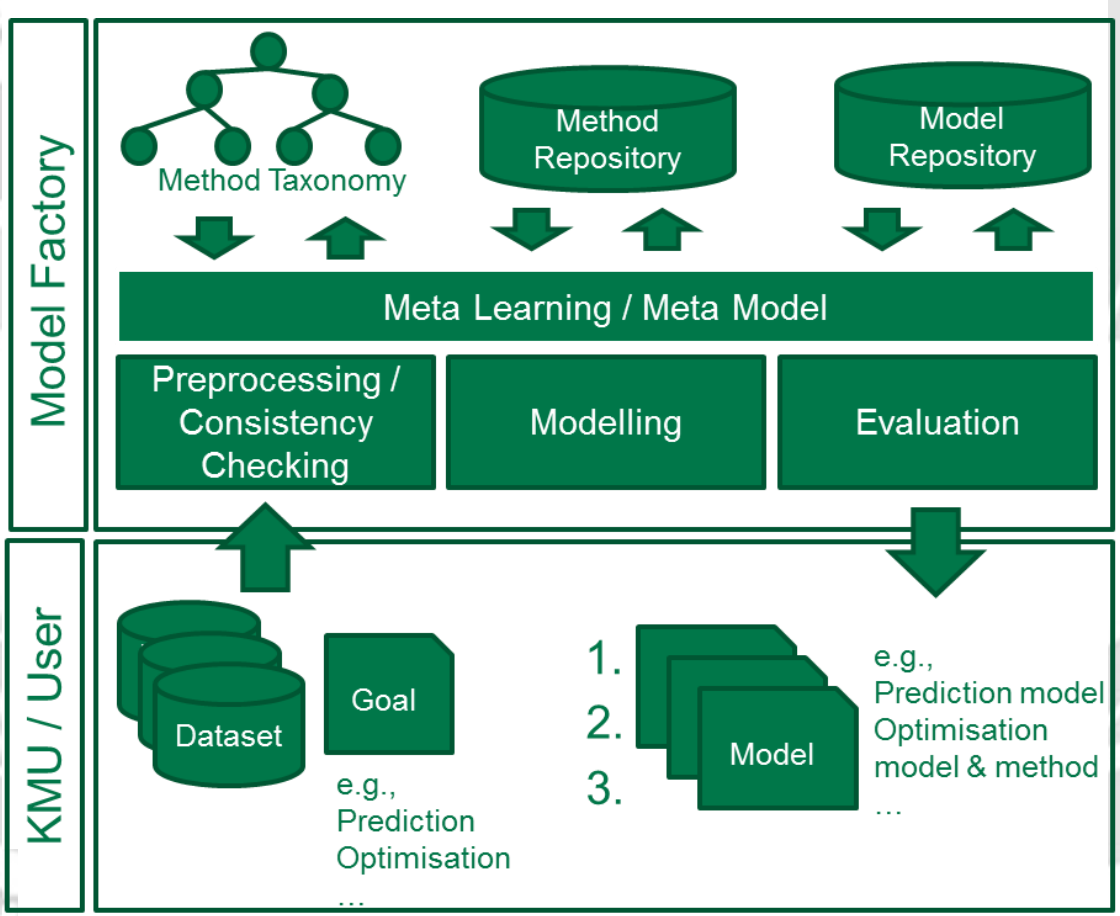
Kunden-
Klassifikations-
system

"unterstützt manuell"

IFCC-Klassifikationsentwickler

Iterative/Agile Verfeinerung

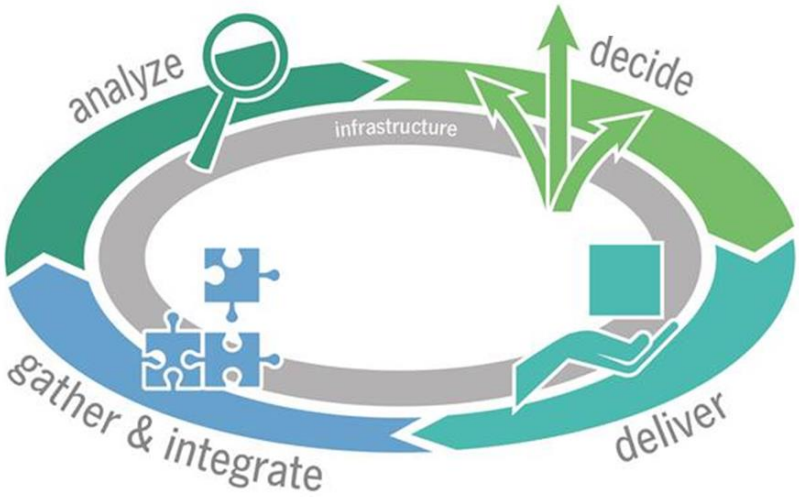
"Konfigurator" für die automatische Stammdatenklassifikation bei IFCC GmbH



"Konfigurator" für die automatische Stammdatenklassifikation bei IFCC GmbH – Zyklus

Welche Methoden sind geeignet?
Welche Modelle erstellen/erweitern?

Welche Modelle sind geeignet zum Lösen des Problems?



Wie Daten vorverarbeiten und bewerten?
Welche Daten berücksichtigen?

Wie kann die Lösung genutzt und daraus gelernt werden?

Wie Daten vorverarbeiten und bewerten? Welche Daten berücksichtigen?

- Konkatenieren der Kurz- und Langbeschreibung (SD + LD)

SD	LD
Zink	Zink Metal
Monitor	Bildschirm
AB	CD

SD + LD
Zink Metal Metal
Monitor Bildschirm
AB CD

- N-Gram anstatt Wörterbuch-basierte Methode (5-gram ausgewählt)

Wie Daten vorverarbeiten und bewerten? Welche Daten berücksichtigen? (2)

- TF-IDF (term frequency–inverse document frequency) anstelle "Auftreten Frequenz"- oder Word2Vec-Methode ausgewählt

SD + LD
Zink Metal Metal
Monitor Bildschirm
AB CD

w1	w2	w3	w4	w5	w6	w7
1,5	3,5	2,7	1,5	3,5	2,7	1,7
-0,8	-3,6	5,4	-0,8	-3,6	5,4	2,8
2,3	1,45	-1,6	2,3	1,45	-1,6	-0,3

- Feature Selection (Reduktion auf 100 Features mit Singular Vector Decomposition SVD)

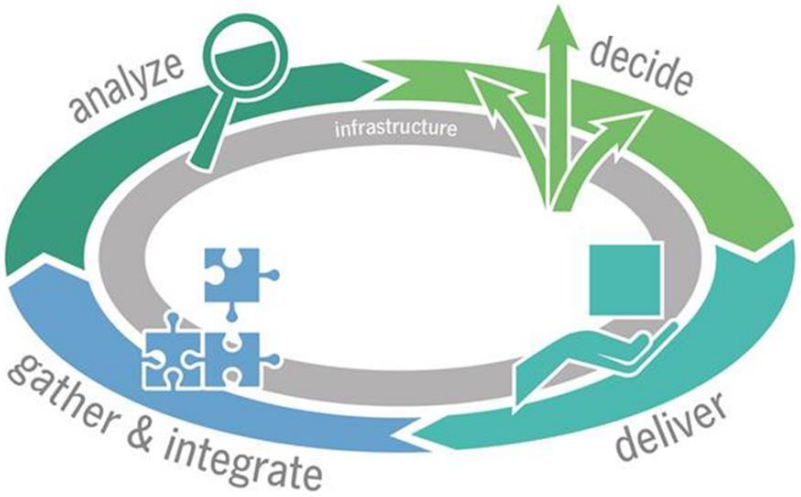
w1	w2	w3	w4	w5	w6	w7
1,5	3,5	2,7	1,5	3,5	2,7	1,7
-0,8	-3,6	5,4	-0,8	-3,6	5,4	2,8
2,3	1,45	-1,6	2,3	1,45	-1,6	-0,3

w1	w2	w3	w4
1,5	3,5	2,7	1,7
-0,8	-3,6	5,4	2,8
2,3	1,45	-1,6	-0,3

"Konfigurator" für die automatische Stammdatenklassifikation bei IFCC GmbH – Zyklus

Welche Methoden sind geeignet?
Welche Modelle erstellen/erweitern?

Welche Modelle sind geeignet zum Lösen des Problems?



Wie Daten vorverarbeiten und bewerten?
Welche Daten berücksichtigen?

Wie kann die Lösung genutzt und daraus gelernt werden?

Welche Methoden sind geeignet?
Welche Modelle erstellen/erweitern?

- Potentiell **tausende Klassen** (40k bei eCl@ss) und mono-hierarchische Klassifikation
- Masterklassifikation (zum Training) weist viele Klassen mit **wenigen Instanzen** auf
- Lediglich kurze und lange **textuelle Beschreibung** als Features; nicht-lineare Trennbarkeit des Feature-Raums
- Stammdaten weisen im Durchschnitt nur **4,7 relevante Wörter** auf; Produkte ohne Beschreibung, Tippfehler etc.
- Sehr **technische Beschreibungen**. Wenig automatische Wikipedia-Annotationen möglich. Spezielle Keine Hintergrundinformationen wie Produktkataloge, Bilder, Videos.
- Manuelle Erstellung/Pflege von Regeln (Regular Expressions) anhand der Stammdaten aufwändig und erreicht Präzision von nur 40%
- Ggf. Mehrsprachigkeit der Stammdaten, **Große Anzahl** an Stammdaten.

Kriteria	Linear Regression	Entscheidungs-bäume	SVM	KNN
Anzahl an Klassen	wenig	wenig	mittel	mittel bis viel
Trainingsdaten pro Klasse	mittel bis viel	Viel	mittel bis viel	wenig bis mittel
Features	numerisch	beide	numerisch	beide
Trennbarkeit	lineare	lineare	nicht lineare	nicht lineare
Qualität	mittel-hoch	hoch	niedrig bis mittel	hoch

Welche Methoden sind geeignet?
Welche Modelle erstellen/erweitern?

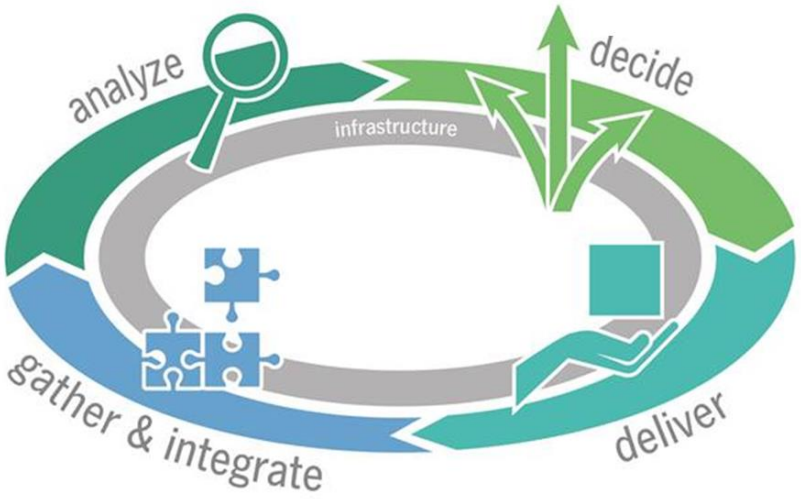
- Potentiell **tausende Klassen** (40k bei eCl@ss) und mono-hierarchische Klassifikation
- Masterklassifikation (zum Training) weist viele Klassen mit **wenigen Instanzen** auf
- Lediglich kurze und lange **textuelle Beschreibung** als Features; nicht-lineare Trennbarkeit des Feature-Raums
- Stammdaten weisen im Durchschnitt nur **4,7 relevante Wörter** auf; Produkte ohne Beschreibung, Tippfehler etc.
- Sehr **technische Beschreibungen**. Wenig automatische Wikipedia-Annotationen möglich. Spezielle Keine Hintergrundinformationen wie Produktkataloge, Bilder, Videos.
- Manuelle Erstellung/Pflege von Regeln (Regular Expressions) anhand der Stammdaten aufwändig und erreicht Präzision von nur 40%
- Ggf. Mehrsprachigkeit der Stammdaten, **Große Anzahl** an Stammdaten.

Kriteria	Linear Regression	Entscheidungs-bäume	SVM	KNN
Anzahl an Klassen	wenig	wenig	mittel	mittel bis viel
Trainingsdaten pro Klasse	mittel bis viel	Viel	mittel bis viel	wenig bis mittel
Features	numerisch	beide	numerisch	beide
Trennbarkeit	lineare	lineare	nicht lineare	nicht lineare
Qualität	mittel-hoch	hoch	niedrig bis mittel	hoch

"Konfigurator" für die automatische Stammdatenklassifikation bei IFCC GmbH – Zyklus

Welche Methoden sind geeignet?
Welche Modelle erstellen/erweitern?

Welche Modelle sind geeignet zum Lösen des Problems?



Wie Daten vorverarbeiten und bewerten?
Welche Daten berücksichtigen?

Wie kann die Lösung genutzt und daraus gelernt werden?

Welche Modelle sind geeignet zum Lösen des Problems?

- Auswahl der Parameter anhand Vorevaluation von Modellen
  - KNN (K = 3), Random Forest (10 Entscheidungsbäume)
  - Reduzierter Datensatz mit 108.636 Produkten + 10-Fold Cross Validation

Hierarchy level	KNN	RandomForest
1	0,85	0,84
2	0,77	0,77
All	0,693	0,687

Präzision der Klassifikation in verschiedenen Ebenen der Hierarchie

Aufgabe	KNN	Random Forest
Training	150s	40 Min
Klassifikation	125s	7s

Trainings- und Klassifikationszeit für KNN und Random Forest

Welche Modelle sind geeignet zum Lösen des Problems?

- Auswahl der Parameter anhand Vorevaluation von Modellen
  - KNN (K = 3), Random Forest (10 Entscheidungsbäume)
  - Reduzierter Datensatz mit 108.636 Produkten + 10-Fold Cross Validation

Hierarchy level	KNN	RandomForest
1	0,85	0,84
2	0,77	0,77
All	0,693	0,687

Präzision der Klassifikation in verschiedenen Ebenen der Hierarchie

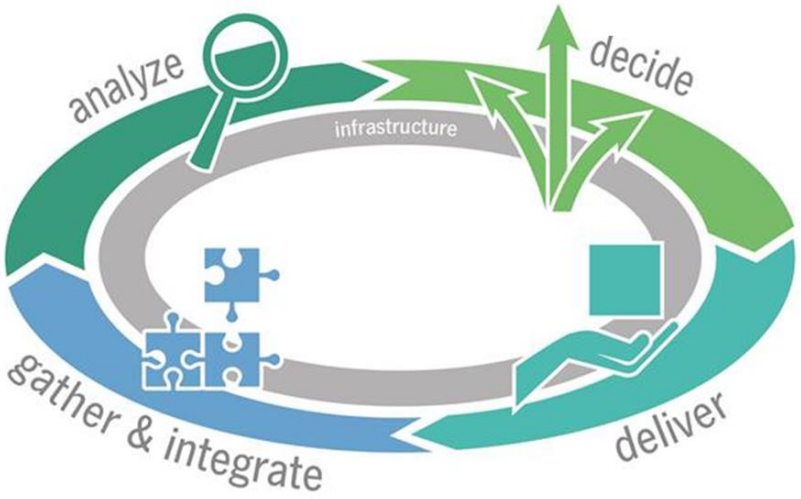
Aufgabe	KNN	Random Forest
Training	150s	40 Min
Klassifikation	125s	7s

Trainings- und Klassifikationszeit für KNN und Random Forest

"Konfigurator" für die automatische Stammdatenklassifikation bei IFCC GmbH – Zyklus

Welche Methoden sind geeignet?
Welche Modelle erstellen/erweitern?

Welche Modelle sind geeignet zum Lösen des Problems?



Wie Daten vorverarbeiten und bewerten?
Welche Daten berücksichtigen?

Wie kann die Lösung genutzt und daraus gelernt werden?

Wie kann die Lösung genutzt und daraus gelernt werden?

- Gesamter Datensatz mit 785.883 Produkten; 10-Fold Cross Validation

Hierarchy level	KNN
1	0,928
2	0,90
All	0,827

Präzision der Klassifikation in verschiedenen Ebenen der Hierarchie

Aufgabe	KNN
Training	8 Min
Klassifikation	7 Min

Trainings- und Klassifikationszeit für KNN

Wie kann die Lösung genutzt und daraus gelernt werden? (2)

- Implementierung
  - Python (Jupyter Notebook)
  - Bibliotheken: SciKit, Pandas
  - Workstation-Rechner mit 32 GB RAM und 4 Prozessorkernen

File Edit View Insert Cell Kernel Help Python 2

Code CellToolbar

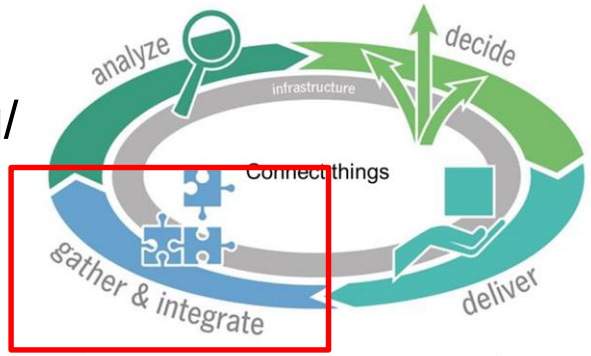
In [1]:
```
import pandas as pd
import numpy as np
import re
from sklearn.cross_validation import KFold
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn import svm
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import SGDClassifier, LogisticRegression
from sklearn.pipeline import Pipeline, FeatureUnion
from stop_words import get_stop_words
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.decomposition import TruncatedSVD, PCA, LatentDirichletAllocation
from sklearn.metrics import precision_score, recall_score, precision_recall_fscore_support
import sklearn.base
import math
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
#jFile is the name of the dataset file
#columns is a vector containing the labels of the columns containing the textual information (Short und Long Description
#class column corresponds to the label of the column containing the class information
def readProducts(jFile, columns, classColumn):
    if "json" in jFile:
        jsonFile = pd.read_json(jFile)
    else:
```

Lessons Learned

- KNN-/Random-Forest Methoden erreichen zufriedenstellende Präzision, dauern jedoch zu lange oder benötigen zu viel Speicher für iterative/agile Stammdatenklassifikation
  - Heuristiken oder Parallelisierung über Big-Data-Stack notwendig

- Stammdatenklassifikation sehr komplex zum vollständigen Automatisieren im "Konfigurator"
  - Insbesondere bei Datenvorverarbeitung
  - Weitere Fallstudien und Generalisierung/ Formalisierung notwendig

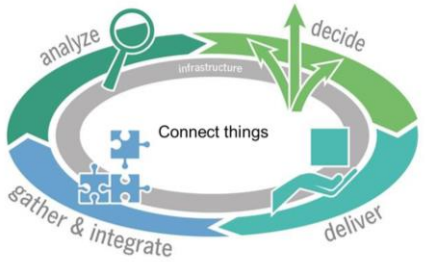


Agenda

- Überblick FZI
- DIZ-Projekt "Konfigurator"
- Fallstudie: Automatische Stammdatenklassifikation mit IFCC GmbH
- **Zusammenfassung**

Zusammenfassung

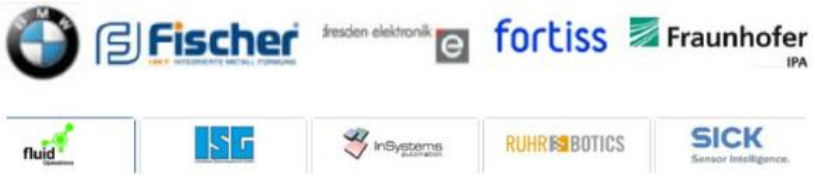
- Datengetriebene Entscheidungsunterstützung ist in vielen Domänen (Robotik uvm.) ein Thema.

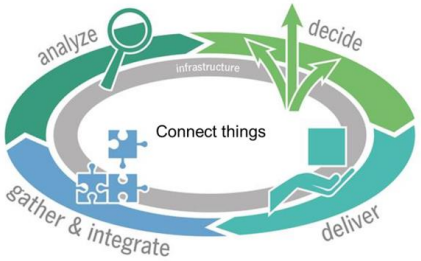


- Im Digitalen Innovationszentrum (DIZ) entsteht aktuell ein "Konfigurator" für die teilautomatische Entscheidungsunterstützung.

- Eine Fallstudie zur automatischen Stammdatenklassifikation mit der IFCC GmbH zeigt vielversprechende Ergebnisse und weitere spannende Forschungsthemen

Auswahl unserer Projektpartner und Themen

Lösung	Partner
Kontext-abhängiges Monitoring von Logistikprozessen mittels Smart Phone im Projekt ProveIT	
Benutzerfreundliche Erstellung von Big-Data-Applikationen über Datenströme mittels StreamPipes-Werkzeug aus Projekt ProaSense	
Automatisches Ableiten von Fähigkeiten und Sicherheitsrisiken in Roboter-zentrierten Arbeitsstätten durch Ontologien im Projekt ReApp	
Verbessertes Fehlermanagement in Produktionsumgebungen mittels Maschinellem Lernen über partiell-erfüllten Datenstrommustern im Projekt BigPro	



DIZ DIGITALES INNOVATIONS ZENTRUM

IFCC master.data.management

Danke!

Kontakt:

Ignacio Traverso-Ribon: traverso@fzi.de

Benedikt Kämpgen: kaempgen@fzi.de