

1. Übersicht über die Veranstaltung

Zusammenlegung zweier Vorlesungen:

| Veranstaltung | Zielgruppe | Dozent |
|----------------------------|--|----------------------------|
| Healthcare Data Analytics | Bachelor Medizininformatik (Pflichtfach) | Prof. Dr. Frank Schirmeier |
| Einführung in Data Science | Bachelor Informatik (Wahlpflichtfach) | Prof. Dr. Frank Schirmeier |

Organisatorisches:

| | Raum | Uhrzeit |
|----------------------|--------|--------------------------|
| Vorlesung: | S0.11 | Di, 14:00-15:30 |
| Praktikum (ab 1.4.): | SU.103 | Di, 10:00-11:30 (WK+MED) |
| | SU.103 | Di, 11:45-13:15 (IF) |

Es sind Übungsaufgaben zu bearbeiten und in Moodle hochzuladen.

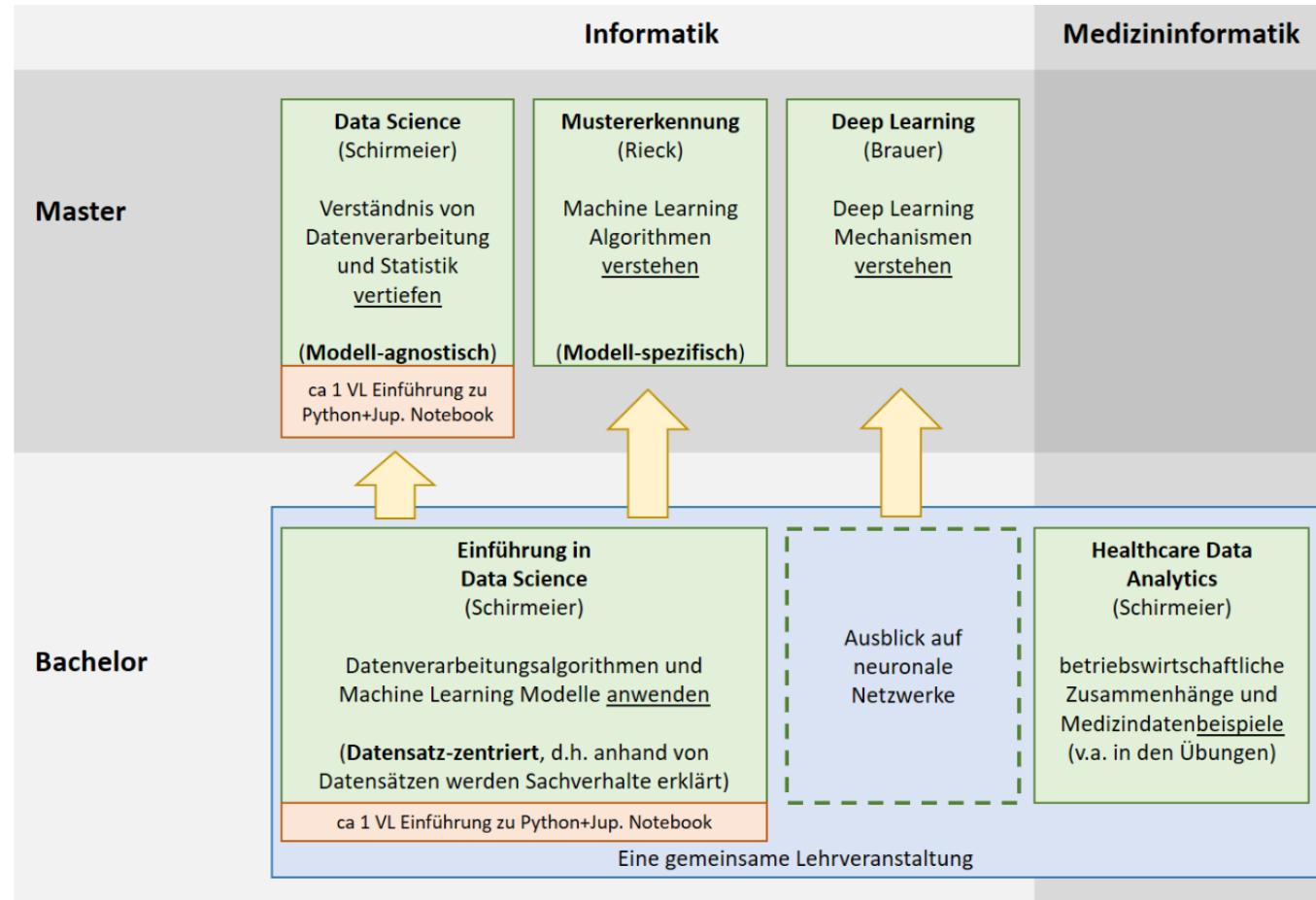
Es sollen mind. 60% der Übungsaufgaben "gut" abgegeben werden.

Jede/r Student/in soll eine Aufgabe im Praktikum präsentieren.

Moodle: Fakultät Informatik > Bachelor of Science > Medizininformatik > "HealthCare Data Analytics / Einführung in Data Science" (Passwort: HDA_EDS) <https://moodle.hs-kempten.de/course/view.php?id=3654>



1.1 Einordnung in das Curriculum

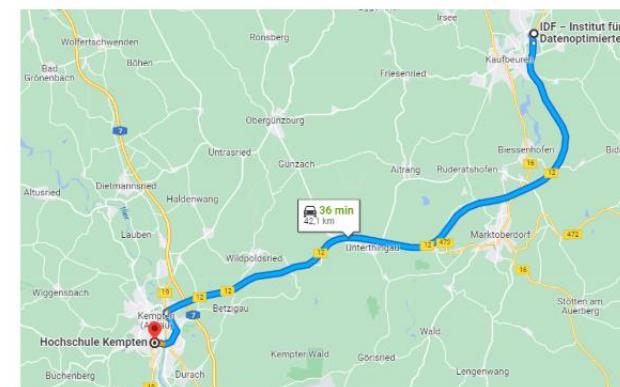


1.2 Über den Dozenten



Prof. Dr. Frank Schirmeier
Mathematiker

- Berufserfahrung als Data Scientist (u.a. Roche, QIAGEN, freiberuflich und in Festanstellung)
- Forschungsprofessur Industrial Data Science
- an der HS Kempten seit 1.6.2021
- Leitung des Instituts für Datenoptimierte Fertigung (IDF) in Kaufbeuren



1.3 Institut für Datenoptimierte Fertigung (IDF)

Ein Forschungsinstitut der Hochschule Kempten mit Schwerpunkt **KI und Data Science in der Industrie 4.0**



Möglichkeit zur Mitarbeit/ Abschlussarbeiten

- ca. 20 wissenschaftliche Mitarbeiter/innen
- Forschungsprojekte u.a. KI Span, DIONE-X, SynErgie (Volumen > 1 Mio €)
- "KI Regionalzentrum Südschwaben" (gemeinsam mit IKR) im Rahmen des Programms "KI Transfer+" des Digitalministeriums

Kooperationen mit renommierten Unternehmen:



Forschungsprojekt **KI Span**: *Den Zerspanungsprozess mit KI optimieren*



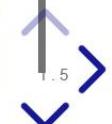
/berger/

Blaser.
SWISSLUBE

CERATIZIT
GROUP

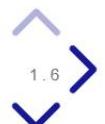
DMG MORI
AKTIENGESELLSCHAFT

promicron
wireless solutions



Gefördert durch das Bayerische Staatsministerium
für Wirtschaft, Landesentwicklung und Energie

Forschungsprojekt **DIONE-X**: Neuartige Datengetriebene Geschäftsmodelle



Forschungsprojekt SynErgie: Energieflexibilisierung in der Fertigung

Samstag, 16. September 2023 | Nr. 214

Wirtschaft | 11

Forscher fordern neuen Kurs in der Energiepolitik

Das Problem hoher Strompreise lässt sich mildern, sagt der Wissenschaftler Hans Ulrich Buhl. Die Industrie müsse dann produzieren, wenn Windräder und Solarparks auf Hochtouren laufen – und die Politik die Weichen richtig stellen.

Von Michael Kierer

Augsburg. In der Bundesregierung steht gerade die Debatte, ob ein subventioniertes Industriestrompreis die Bedingungen für die Wirtschaft verbessern kann. Befürwort wird natürlich, dass mehr erneute Energie erzeugt werden muss, um das Stromangebot auszuweiten und die Preise zu senken. Große Straßenstrassen – für die derzeit der Spannweite erfolgt ist – sollen Windstrom aus dem Norden in die Industriezentren in Süddeutschland liefern. Aus Südbayerischer Perspektive werden diese Ansätze aber kritisiert: reine erneute Energie ist die Industrie unbillig anzusehen. Die bis 2027/2028 geplanten vier Gigawatt Leistung der Stromtrasse SüdLink und die zwei Gigawatt von Stadtolektra werden nicht genügen", warnt Professor Hans Ulrich Buhl, Gründer des Forschungsinstituts für Informationsmanagement (FIM) mit Standorten in Bayreuth und Augsburg.

Aus Sicht der Forscher müssen stärker als bisher Wege gefunden werden, die Produktion in der Industrie mit der schwächeren Energieerzeugung von Sonne und Wind abzustimmen. „Die Schlüsselkompetenz der Energiewende wird es sein, erneutbare Energien dann zu nutzen, wenn sie im Überfluss vorhanden und günstig sind“, betont Buhl.

Augsburg. In der Fabrik des Papierstellers UPM ist Schritt eins des Potenzial für Flexibilisierung bewiesen. Dort werden naerst aus Hochleistungszellulosefasern als Vierprodukt für die Papierproduktion im Werk hergestellt. Diese Herstellung ist sehr energieintensiv. Die Anlage benötigt rund 35 Megawatt Leistung – groß gerichtet so viel, wie ein großer Supermarkt bei voller Sonne produziert. In Rahmen eines Synergiespektrums UPM, die Zellulosefaser Herstellung auf eine energieflexible Produktion um: Die Maschine wird dann betrieben, wenn gerade viel an erneuerbaren Energien erzeugt wird oder der Strom besonders günstig ist.

Nun geht das Projektteam eine Studie weiter. Es untersucht, ob auch ein ganzes Industriequartier das Stadtviertel der Fertigung in Augsburg – auf eine klimafreundliche

In einem großen Projekt haben die Forscher bereits gezeigt, dass es möglich ist, den Energiebedarf in der Industrie mit einem schwankenden Energieangebot zu synchronisieren. Das Projekt läuft unter dem Kürzel SynErgie, seit 2016 wird es bis 2026 mit über 100 Millionen Euro vom Bundesministerium für Bildung und Forschung gefördert. Der Gründersitz Augsburg ist dabei die Modellregion für Energieflexibilität. Allein hier beteiligen sich 26 Partnerunternehmen und 11 Forschungseinrichtungen.

Unter anderen haben die Prozesskettenhöheren in der Fabrik des Papierstellers UPM die Schritte des Potenzials für Flexibilisierung bewiesen. Dort werden naerst aus Hochleistungszellulosefasern als Vierprodukt für die Papierproduktion im Werk hergestellt. Diese Herstellung ist sehr energieintensiv. Die Anlage benötigt rund 35 Megawatt Leistung – groß gerichtet so viel, wie ein großer Supermarkt bei voller Sonne produziert. In Rahmen eines Synergiespektrums UPM, die Zellulosefaser Herstellung auf eine energieflexible Produktion um: Die Maschine wird dann betrieben, wenn gerade viel an erneuerbaren Energien erzeugt wird oder der Strom besonders günstig ist.

Nun geht das Projektteam eine Studie weiter. Es untersucht, ob auch ein ganzes Industriequartier das Stadtviertel der Fertigung in Augsburg – auf eine klimafreundliche

sicher sichert, dass sie viel fossile Energie einsparen. „Das Stadtviertel erzeugt rund die Hälfte der CO₂-Emissionen der Stadt Augsburg“, erklärt Buhl. „Die Stadt wird niemals ihre Klimaziele erreichen, wenn nicht die Industrie senkt ihre Emissionen sichtbar.“ Das SynErgie-Konzept soll immer detaillierteren Stufen bis Jahr 2026 stehen.

Bisher aber, kritisieren die Forscher, hat die Politik erst begonnen, in den Privathaushalten die Weichen für Energieflexibilität zu stellen. Die Bundesnetzagentur hat Pläne vorgelegt, dass die Netzbetreiber in den nächsten Jahren in etwa 100 Ladestellen in den Privathaushalten „zähmen“ können, wenn es der Netzbetrieb erfordert. Die Pläne für die Reform seien „zu begrüßen“, schreibt Buhl zusammen mit Professor Martin Weibelzahl vom Institutsteil Wirtschaftsinformatik der Fraunhofer FIT in einer Stellungnahme an die Redaktionen. Sie hoffen allerdings, dass die Industrie nicht gleichzeitig berücksichtigt wird. „Wesentlich für das schnellere Vorankommen der Energiewende ist es, neben dem Nachfrage-Flexibilisierungspotential der Haushalte auch jenes des Gewerbes sowie der Industrie zu haben.“ Dieses beträgt 91 Tiefwasserstunden – ein Vielfaches dessen, was bei Haushalten erwartbar sei.

Die Politik müsse deshalb schnell die Bedingungen für mehr

Produktion umgestellt werden kann. Dort sind große Industrieunternehmen angesiedelt, unter anderem ebenfalls UPM, der Großmotorenhersteller MAN Energy Solutions und der Raketenbauer MT Aerospace. Aufgefallen ist den Forschern, dass in diesen Betrieben gleichermaßen erneutbare Stoffe wie Wärme und Strom erzeugt wird. Bisher geht ein großer Teil der Energie als Abwärme verloren, be-

schreibt Buhl die Situation. Die Produktion ließe sich weit effizienter gestalten. Die Idee ist es, die Abwärme in der Produktion eines anderen Betriebes zu nutzen, statt dort mit Gas eigene Wärme zu erzeugen. Speicher und industrielle Wärmenutzung könnten helfen. Auch die Anbindung an das Fernwärmenetz der Stadtwerke soll eine große Rolle spielen.

Am Ende, da sind sich die For-

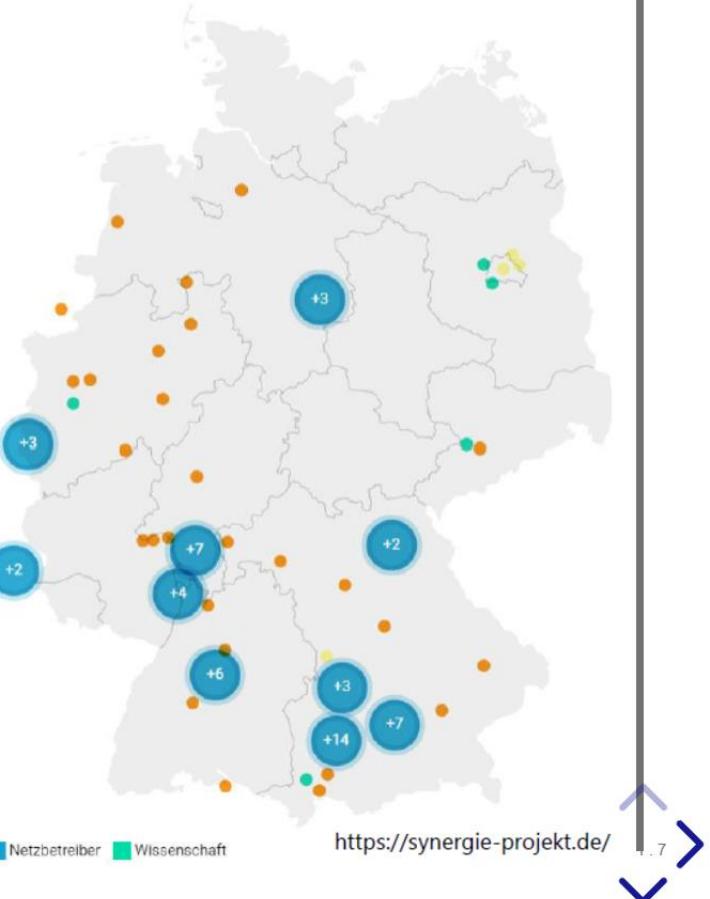
schere sicher, dass sie viel fossile Energie einsparen. „Das Stadtviertel erzeugt rund die Hälfte der CO₂-Emissionen der Stadt Augsburg“, erklärt Buhl. „Die Stadt wird niemals ihre Klimaziele erreichen, wenn nicht die Industrie senkt ihre Emissionen sichtbar.“ Das SynErgie-Konzept soll immer detaillierteren Stufen bis Jahr 2026 stehen.

Zudem bestehen im Energiesystem zu viele Fehlentwicklungen. Ein Beispiel sei, dass die Strom-Netzvergleichsverordnung noch immer Nachlässe für Betriebe gewährt, wenn sie Strom sehr kontinuierlich in einem Netzteil erzeugen. „Diese Beobachtung kann in unserer Welt, in der Kohle- und Atomkraftwerke durchlaufen, wie sie ist aber in einem flexiblen Energiesystem überholt“, sagt Buhl. Er schlägt vor, die Nachlässe stattdessen Unternehmen zu gewähren, die im Sinne eines stabilen Netzes energieflexibel handeln.

„Bereits heute lassen sich mit den zur Verfügung stehenden Möglichkeiten große Mengen CO₂ einsparen, der Instrumentenkanien ist da“, betont Alexander Gaudling von den Technologieförderlinien für den Strukturwandel im Süden (TOS). „Wir brauchen ein politisches Signal, dass Firmen durch Energieflexibilität hier zu wirtschaftlichen Profiten profitieren können.“ Die Technologieförderlinie will die Erkenntnisse der SynErgie-Projekte weiter realisieren.



Augsburgs Industriegebiete brauchen viel Energie. Foto: Uwe Wagner



https://www.ki-transfer-plus.de/regionalzentren

KI TRANSFER + Programm KI-Regionalzentren Projekte 2021 Projekte 2022

Bayerisches Staatsministerium für Digitales

The map displays the state of Bavaria with several regional centers highlighted by dashed lines connecting them to their respective logos and names:

- thws** Technische Hochschule Würzburg-Schweinfurt
- ohm** Technische Hochschule Nürnberg
- HNU** Hochschule Neu-Ulm University of Applied Sciences
- RCAI** REGensburg CENTER FOR ARTIFICIAL INTELLIGENCE
- aci** initiative for applied artificial intelligence
- Hochschule Kempten** University of Applied Sciences

A blue circular icon with three nodes connected by lines is positioned in the top left corner of the map area.

.8

1.4 Vorlesungsinhalt heute



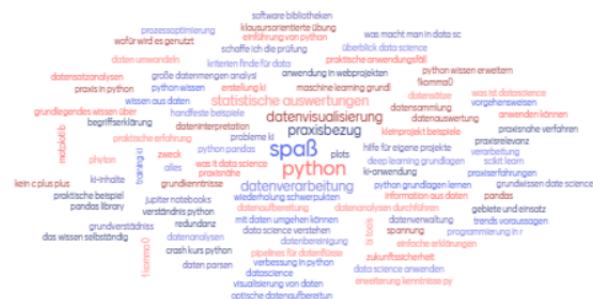
Erkennung von Knochenentwicklung bei Kindern, <https://www.bonexpert.com>

1. Einführung und Überblick

2. Abgrenzung Data Science und Data Analytics

3. Zusammenfassung und Semesterplanung

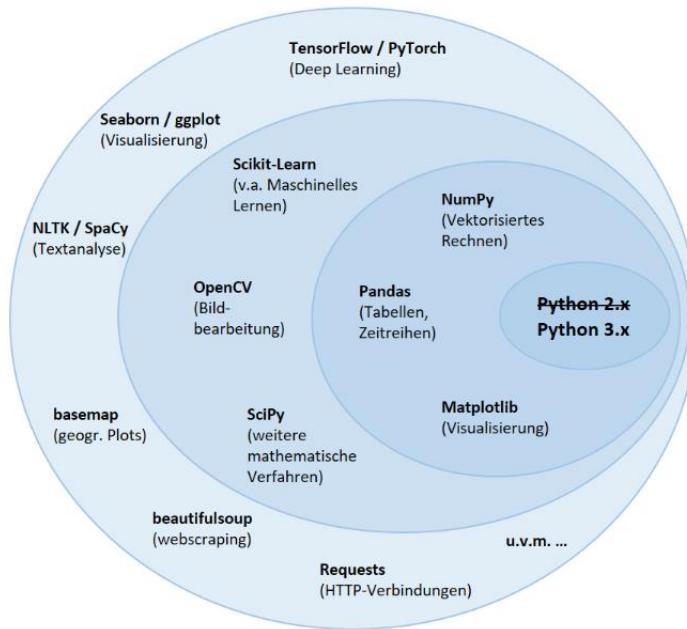
1.5 Ergebnis der Umfrage



1.5 Ergebnis der Umfrage 2024



1.6 Einleitung in Python und Jupyter Notebook



Python-Installation (auf Version >= 3.6 achten):

- z.B. via Anaconda:
<https://docs.anaconda.com/free/anaconda/install/index.html>
- Jupyter Notebook für interaktive Auswertungen

Einrichten der Arbeitsumgebung (in Anaconda Prompt):

- Verwendung von Environments (optional):

```
conda create -n my_env_name  
conda activate my_env_name
```

- Installation von Paketen:

```
conda install jupyter numpy -c conda-forge
```

- Jupyter Notebook starten: `jupyter notebook`

Ab dem 1. Januar 2020 hat Python offiziell die Unterstützung für `python2` eingestellt. Wir werden Python 3.6 oder höher voraussetzen.

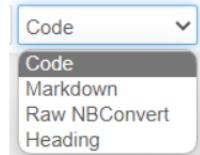
```
In [4]: 1 # mit ! wird ein Konsolenbefehl ausgeführt,  
2 # z.B. !dir  
3 !python --version
```

Python 3.9.20

```
In [5]: 1 # In Jupyter Notebook können Python-Befehle  
2 # im Code-Fenster eingegeben werden, deren  
3 # Ergebnisse direkt darunter angezeigt werden:  
4 x = 5  
5 y = 3  
6 x + y
```

Out[5]: 8

Im Jupyter Notebook können Code-Elemente ("Code") mit Text-Elementen ("Markdown") gemischt werden, hierzu gibt es ein Dropdown-Menü (o.ä.):



Nicht notwendig für Sie für diese Vorlesung:

- Zweisäugigkeit, Zeilenummern usw. ist möglich über sog. nbextensions (Notebook-Version < 7).
- Die Anzeige in einer Slideshow ist möglich über die Erweiterung "RISE".



Python ist eine hochentwickelte, dynamisch typisierte Multiparadigma-Programmiersprache.

Beispiel: Implementierung des klassischen Quicksort-Algorithmus:

```
In [43]: 1 def quicksort(arr):
2     if len(arr) <= 1:
3         return arr
4     pivot = arr[len(arr) // 2]
5     left = [x for x in arr if x < pivot]
6     middle = [x for x in arr if x == pivot]
7     right = [x for x in arr if x > pivot]
8     return quicksort(left) + middle + quicksort(right)
9
10
11 print(quicksort([3, 6, 8, 10, 1, 2, 1]))
12
[1, 1, 2, 3, 6, 8, 10]
```

```
In [7]: 1 meineliste = [1,2,3,4.0,5.0,6.0]
2 meineliste
```

Out[7]: [1, 2, 3, 4.0, 5.0, 6.0]

```
In [25]: 1 # List comprehension: Dynamische Erstellung
2 # einer neuen Liste
3 [x*x for x in meineliste if type(x) is int]
```

Out[25]: [1, 4, 9]

Zahlentypen:

```
In [11]: 1 x = 3
2 print(x)
3 print(type(x))
4
3
<class 'int'>
```

```
In [12]: 1 y = 2.5
2 print(y)
3 print(type(y))
4
2.5
<class 'float'>
```

Ints und Floats verhalten sich so, wie man es erwarten würde:

```
In [17]: 1 print(x + 1)    # Addition (x++ nicht implementiert)
2 print(x - 1.0)   # Subtraktion
3 print(x * 2)    # Multiplikation
4 print(x**2)     # Exponent
5
4
2.0
6
9
```

```
In [16]: 1 print(y + 1 )  # Addition (eines Ints)
2 print(y - 1.0)  # Subtraktion (eines Floats)
3 print(y * 2 )  # Multiplikation
4 print(y**2 )  # Exponent
5
3.5
1.5
5.0
6.25
```

Hier sieht man dynamische Typ-Umwandlungen (3-1.0 -> 2.0).

Booleans: Python implementiert alle üblichen Operatoren für die boolesche Logik, verwendet aber englische Wörter anstelle von Symbolen (`&&`, `||`, etc.):

```
In [49]: 1 t, f = True, False  
2 print(type(t))  
3  
<class 'bool'>
```

```
In [50]: 1 print(t and f) # Logisches UND;  
2 print(t or f) # Logisches ODER;  
3 print( not t) # Logisches NICHT;  
4 print(t != f) # Logisches EXKLUSIVES ODER;  
  
False  
True  
False  
True
```

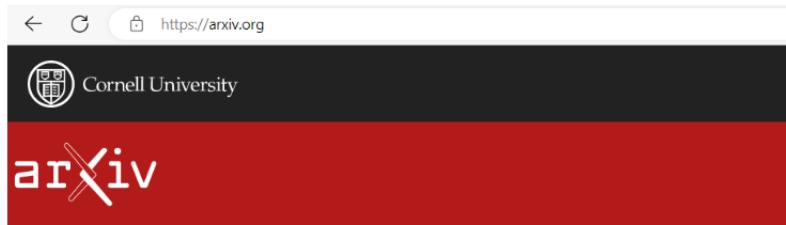
Für vektorisierte Operationen wird `|` für OR, `&` für AND, `~` (Tilde) für NOT verwendet:

```
In [9]: 1 import numpy as np # Details zu numpy später  
2  
3 vec1 = np.array ([True, True, False] )  
4 vec2 = np.array ([True, False, True] )  
5 print ( vec1 & vec2 ) # Ergebnis: [True, False, False]  
6  
7 # Anwendungszenario "Boolsches Filtering"  
8 vec3 = np.array ( ["A", "B", "C"] )  
9 print( vec3[ vec1 & vec2 ] )  
  
[ True False False]  
['A']
```

```
In [26]: 1 # Ergebnis?  
2 vec1 | vec2  
  
Out[26]: array([ True,  True,  True])
```

2. Abgrenzung "Data Science" und "Data Analytics"

2.1. Auswertung von Fachartikeln: "Data Science" vs. "Data Analytics"

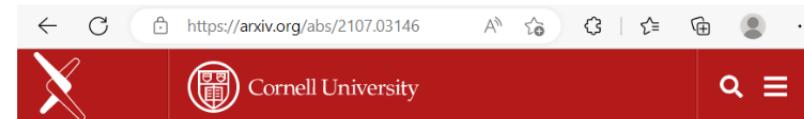


arXiv is a free distribution service and an open-access archive for 2,222,729 scholarly articles in the fields of physics, mathematics, computer science, quantitative biology, quantitative finance, statistics, electrical engineering and systems science, and economics. Materials on this site are not peer-reviewed by arXiv.

Subject search and browse:
Physics Form Interface

News

Read about recent news and updates on [arXiv's blog](#). (View the former "what's new" pages here).
Read [robots beware](#) before attempting any automated download.



Computer Science > Neural and Evolutionary Computing

arXiv:2107.03146 (cs)

Model-agnostic multi-objective approach for the evolutionary discovery of mathematical models

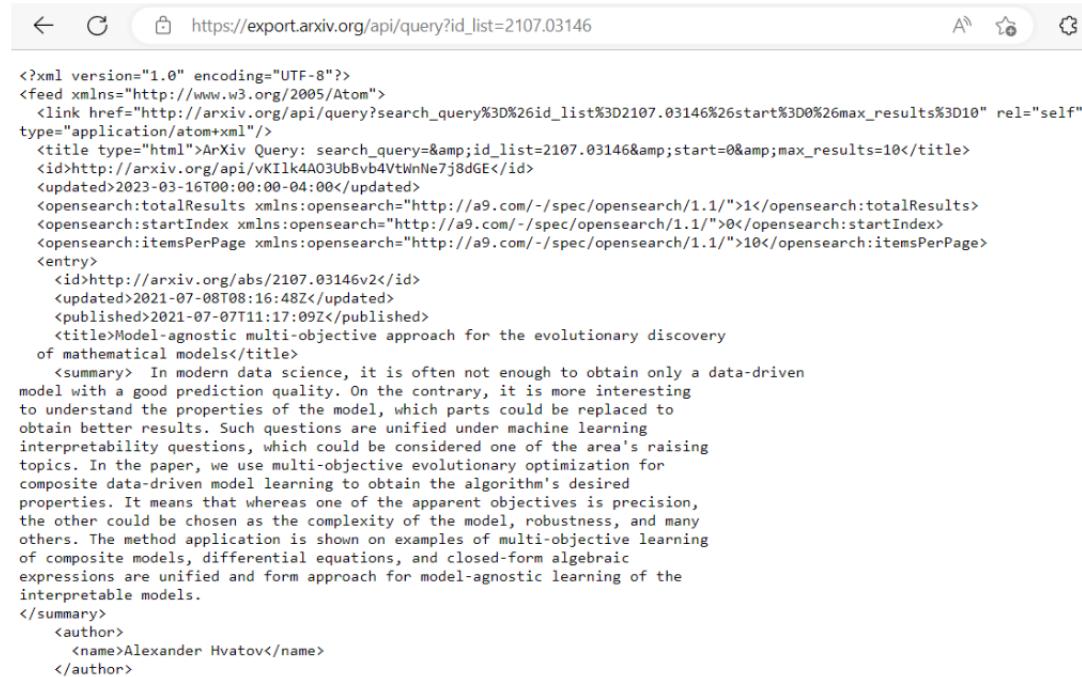
Alexander Hvatov, Mikhail Maslyaev, Iana S. Polonskaya, Mikhail Sarafanov, Mark Merezhnikov, Nikolay O. Nikitin

[Download PDF](#)

In modern data science, it is often not enough to obtain only a data-driven model with a good prediction quality. On the contrary, it is more interesting to understand the properties of the model, which parts could be replaced to obtain better results. Such questions are unified under machine learning interpretability questions, which could be considered one of the area's raising topics. In the paper, we use multi-objective evolutionary optimization for composite data-driven model learning to obtain the algorithm's desired properties. It means that whereas one of the apparent objectives is precision, the other could be chosen as the complexity of the model, robustness, and many others. The method application is shown on examples of multi-objective learning of composite models, differential equations, and closed-form algebraic expressions are unified and form approach for model-agnostic learning of the interpretable models.



Nach maschinen-lesbarer API suchen (bei Web-Scraping wird der Zugriff sonst schnell geblockt):



The screenshot shows a browser window with the URL https://export.arxiv.org/api/query?id_list=2107.03146. The page displays an XML feed. The XML content includes a feed header, a single entry for a paper, and a summary of the paper's content.

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <link href="http://arxiv.org/api/query?search_query%3D2107.03146%26start%3D0%26max_results%3D10" rel="self" type="application/atom+xml"/>
  <title type="html">ArXiv Query: search_query=&id_list=2107.03146&start=0&max_results=10</title>
  <id>http://arxiv.org/api/vKI1k4AO3UbBvb4VtWnNe7j8dGE</id>
  <updated>2023-03-16T00:00:00-04:00</updated>
  <opensearch:totalResults xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/">1</opensearch:totalResults>
  <opensearch:startIndex xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/">0</opensearch:startIndex>
  <opensearch:itemsPerPage xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/">10</opensearch:itemsPerPage>
<entry>
  <id>http://arxiv.org/abs/2107.03146v2</id>
  <updated>2021-07-08T08:16:48Z</updated>
  <published>2021-07-07T11:17:09Z</published>
  <title>Model-agnostic multi-objective approach for the evolutionary discovery of mathematical models</title>
  <summary> In modern data science, it is often not enough to obtain only a data-driven model with a good prediction quality. On the contrary, it is more interesting to understand the properties of the model, which parts could be replaced to obtain better results. Such questions are unified under machine learning interpretability questions, which could be considered one of the area's raising topics. In the paper, we use multi-objective evolutionary optimization for composite data-driven model learning to obtain the algorithm's desired properties. It means that whereas one of the apparent objectives is precision, the other could be chosen as the complexity of the model, robustness, and many others. The method application is shown on examples of multi-objective learning of composite models, differential equations, and closed-form algebraic expressions are unified and form approach for model-agnostic learning of the interpretable models. </summary>
  <author>
    <name>Alexander Hvatov</name>
  </author>
```



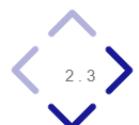
2.1 Abstracts zum Suchstring "Data Science" bzw. "Data Analytics"

```
In [10]: 1 query_arxiv ( search_phrases=["data science"],  
           2             fpath_result="abstracts_ds.txt" )  
           3 with open ( "abstracts_ds.txt", "r" ) as f:  
           4     print ( f.read(1000).splitlines() )  
           5
```

['Although data science builds on knowledge from computer science, mathematics, statistics, and other disciplines, data science is a unique field with many mysteries to unlock: challenging scientific questions and pressing questions of societal importance. This article starts with meta-questions about data science as a discipline and then elaborates on ten ideas for the basis of a research agenda for data science.', 'Increasingly larger number of software systems today are including data science components for descriptive, predictive, and prescriptive analytics. The collection of data science stages from acquisition, to cleaning/curation, to modeling, and so on are referred to as data science pipelines. To facilitate research and practice on data science pipelines, it is essential to understand their nature. What are the typical stages of a data science pipeline? How are they connected? Do the pipelines differ in the theoretical representations and that in the practice? Today we do not fully']

```
In [12]: 1 query_arxiv ( search_phrases=["data analytics"],  
           2             fpath_result="abstracts_da.txt" )  
           3 with open ( "abstracts_da.txt", "r" ) as f:  
           4     print ( f.read(1000).splitlines() )  
           5
```

['With the worldwide emergence of data protection regulations, how to conduct law-regulated big data analytics becomes a challenging and fundamental problem. This article introduces the principle of least sensing, a promising sensing paradigm toward law-regulated big data analytics.', 'On the rise of distributed computing technologies, video big data analytics in the cloud have attracted researchers and practitioners' attention. The current technology and market trends demand an efficient framework for video big data analytics. However, the current work is too limited to provide an architecture on video big data analytics in the cloud, including managing and analyzing video big data, the challenges, and opportunities. This study proposes a service-oriented layered reference architecture for intelligent video big data analytics in the cloud. Finally, we identify and articulate several open research issues and challenges, which have been raised by the deployment of big data technologies in the']



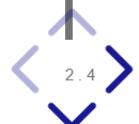
2.2 Einlesen der zwischengespeicherten Abstracts

```
In [59]: ▾ 1 # read data and store as a list of abstracts
  ▾ 2 abstracts_ds = read_data("abstracts_ds.txt",
  3                               ["data science"])
  4 abstracts_ds[:2]
  5
```

Out[59]: ['although data_science builds on knowledge from computer science , mathematics , statistics , and other disciplines , data_science is a unique field with many mysteries to unlock : challenging scientific questions and pressing questions of societal importance . this article starts with meta-questions about data_science as a discipline and then elaborates on ten ideas for the basis of a research agenda for data_science .',
'increasingly larger number of software systems today are including data_science components for descriptive , predictive , and prescriptive analytics . the collection of data_science stages from acquisition , to cleaning / curation , to modeling , and so on are referred to as data_science pipelines . to facilitate research and practice on data_science pipelines , it is essential to understand their nature . what are the typical stages of a data_science pipeline ? how are they connected ? do the pipelines differ in the theoretical representations and that in the practice ? today we do not fully understand these architectural characteristics of data_science pipelines . in this work , we present a three-pronged comprehensive study to answer this for the state-of-the-art , data_science in-the-small , and data_science in-the-large . our study analyzes three datasets : a collection of 71 proposals for data_science pipelines and related concepts in theory , a collection of over 105 implementations of curated data_science pipelines from kaggle competitions to understand data_science in-the-small , and a collection of 21 mature data_science projects from github to understand data_science . . .']

```
In [60]: ▾ 1 # read data and store as a list of abstracts
  ▾ 2 abstracts_da = read_data("abstracts_da.txt",
  3                               ["data analytics"])
  4 abstracts_da[:2]
  5
```

Out[60]: ['with the worldwide emergence of data protection regulations , how to conduct law-regulated big data_analytics becomes a challenging and fundamental problem . this article introduces the principle of least sensing , a promising sensing paradigm toward law-regulated big data_analytics .',
'"on the rise of distributed computing technologies , video big data_analytics in the cloud have attracted researchers and practitioners ' attention . the current technology and market trends demand an efficient framework for video big data_analytics . however , the current work is to o limited to provide an architecture on video big data_analytics in the cloud , including managing and analyzing video big data , the challenges , and opportunities . this study proposes a service-oriented layered reference architecture for intelligent video big data_analytics in the cloud . finally , we identify and articulate several open research issues and challenges , which have been raised by the deployment of big data technologies in the cloud for video big data_analytics . this paper provides the research studies and technologies advancing video analyses in the era of big data and cloud computing . this is the first study that presents the generalized view of the video big data_analytics in the cloud to the best of our knowledge ."]



2.3 Document-Term-Matrizen

```
In [32]: 1 docterms_ds = document_term_matrix ( abstracts_ds )  
2 docterms_ds  
3
```

Out[32]:

| | , | data_science | a | and | of | on | questions | . | with | for | ... | oblivio |
|------|-----|--------------|---|-----|-----|-----|-----------|-----|------|-----|-----|---------|
| 0 | 4 | 4 | | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 0 |
| 1 | 13 | 14 | | 5 | 8 | 10 | 2 | 0 | 7 | 0 | 3 | 0 |
| 2 | 4 | 6 | | 2 | 2 | 2 | 0 | 0 | 4 | 1 | 2 | 0 |
| 3 | 0 | 2 | | 0 | 1 | 1 | 0 | 0 | 2 | 0 | 1 | 0 |
| 4 | 0 | 1 | | 2 | 1 | 2 | 0 | 0 | 1 | 0 | 0 | 0 |
| ... | ... | ... | | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1530 | 5 | 1 | | 6 | 12 | 13 | 0 | 0 | 12 | 1 | 1 | 0 |
| 1531 | 19 | 1 | | 4 | 8 | 6 | 2 | 0 | 10 | 2 | 1 | 0 |
| 1532 | 11 | 1 | | 2 | 17 | 7 | 1 | 2 | 11 | 3 | 6 | 0 |
| 1533 | 12 | 1 | | 12 | 8 | 13 | 1 | 0 | 11 | 2 | 4 | 1 |
| 1534 | 14 | 1 | | 16 | 5 | 27 | 6 | 0 | 13 | 6 | 5 | 0 |

1535 rows × 16748 columns

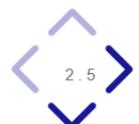
```
In [34]: 1 docterms_da = document_term_matrix ( abstracts_da )  
2 docterms_da  
3
```

Out[34]:

| | a | law-regulated | of | sensing | data_analytics | , | . | big | the | introduces | | |
|------|-----|---------------|----|---------|----------------|-----|---|-----|-----|------------|-----|-----|
| 0 | 2 | 2 | | 2 | 2 | | | 2 | 2 | 2 | 2 | 1 |
| 1 | 1 | 0 | | 5 | 0 | 6 | | 7 | 7 | 9 | 16 | 0 |
| 2 | 4 | 0 | | 7 | 0 | 6 | | 5 | 7 | 7 | 5 | 0 |
| 3 | 0 | 0 | | 3 | 0 | 4 | | 3 | 6 | 0 | 6 | 0 |
| 4 | 5 | 0 | | 12 | 0 | 7 | | 14 | 12 | 9 | 21 | 0 |
| ... | ... | ... | | ... | ... | ... | | ... | ... | ... | ... | ... |
| 1105 | 9 | 0 | | 8 | 0 | 1 | | 10 | 11 | 1 | 28 | 0 |
| 1106 | 8 | 0 | | 11 | 0 | 1 | | 9 | 10 | 0 | 10 | 0 |
| 1107 | 4 | 0 | | 15 | 0 | 1 | | 14 | 15 | 0 | 17 | 1 |
| 1108 | 6 | 0 | | 8 | 0 | 1 | | 11 | 11 | 1 | 4 | 0 |
| 1109 | 7 | 0 | | 8 | 0 | 0 | | 4 | 9 | 0 | 16 | 0 |

1110 rows × 13708 columns

Hier wird also für jeden Abstract gezählt, wie oft die jeweiligen Wörter vorkommen.
(Jede Zeile entspricht einem Abstract.)



2.4: Python-Einschub: Einblicke in die Funktion query_arxiv (1/2)

Erinnerung: `query_arxiv` lädt Abstracts zu einem Suchwort von <https://arxiv.org> und speichert diese in einer .txt-Datei.

```
In [7]: 1 # String Definition
2 a = "Hallo"
3 b = "Welt"
4
5 # String Concatenation
6 c = a + " " + b
7
8 c
9
```

Out[7]: 'Hallo Welt'

```
In [9]: 1 # List Comprehension: Python-Idiom als Alternative zu (expliziten) Schleifen
2 # new_list = [f(i) for i in old_list if condition(i)] , where "if condition(i)" is optional
3
4 [ i*i for i in [0,1,2,3,4,5,6,7,8,9,10] if i%2 == 0 ]
5
```

Out[9]: [0, 4, 16, 36, 64, 100]

```
In [8]: 1 # Listen
2 L = [1,2,3,a,b,c]
3
4 # (zero-based) Indizierung und Manipulation
5 L[0] = 100
6
7 L
8
```

Out[8]: [100, 2, 3, 'Hallo', 'Welt', 'Hallo Welt']

2.4: Python-Einschub: Einblicke in die Funktion query_arxiv (2/2)

Erinnerung: `query_arxiv` lädt Abstracts zu einem Suchwort von <https://arxiv.org> und speichert diese in einer .txt-Datei.

In der Funktion `query_arxiv`:

```
In [14]: 1 # String Concatenation mit "+"
2 batchsize = 5000
3 base_url="http://export.arxiv.org/api/query?"
4 url = base_url + "search_query=" + search_string + \
      "&max_results=" + str(batchsize)
5
6 url
7
8
9 # Der Python-Parser arbeitet Zeilen-basiert;
10 # verwende \ für Umbruch auf mehrere Zeilen
11
```

Out[14]: 'http://export.arxiv.org/api/query?search_query=abs:%22Machine+Learning%22+OR+abs:%22Deep+Learning%22&max_results=5000'

```
In [12]: 1 search_phrases = ["Machine Learning", "Deep Learning"]
2
3 # List Comprehension zur Vorverarbeitung
4 search_phrases = ["abs:%22"+s.replace(" ", "+")+"%22" \
5                   for s in search_phrases]
6 search_phrases
7
```

Out[12]: ['abs:%22Machine+Learning%22', 'abs:%22Deep+Learning%22']

```
In [13]: 1 # String Concatenation mit variabler Anzahl Strings
2 search_string = "+OR+".join ( search_phrases )
3 search_string
4
```

Out[13]: 'abs:%22Machine+Learning%22+OR+abs:%22Deep+Learning%22'

2.5 Python-Einschub: Einblicke in die Funktion `read_data`

Erinnerung: `read_data` liest die .txt-Datei ein, in der je Zeile ein Abstract gespeichert ist.

```
In [ ]: 1 # Funktionsdefinition (Einrückung ist wichtig)
2 def read_data ( fpath, search_phrases ):
3     ...
4     # Rückgabewert (Liste von Abstracts)
5     return abstracts
6
```

```
In [ ]: 1 # Aufruf
2 abstracts_ds = read_data ( "abstracts_ds.txt",
3                             ["Data Science"] )
4
5 # oder mit "named arguments":
6 ... = read_data ( search_phrases=["Data Science"],
7                   fpath="abstracts_ds.txt" )
8
```

In der Funktion `read_data`:

```
In [ ]: 1 with open ( fpath, "r" ) as f:
2     # Einlesen der .txt-Datei in Liste mit Abstracts
3     abstracts = f.read().splitlines()
4
5     # "Vereinfache" den ersten Abstract:
6     abstracts[0] = abstracts[0].lower()
7
```

```
In [ ]: 1 # "Vereinfache" alle Abstracts
2 for i in range(len(abstracts)):
3     abstracts[i] = abstracts[i].lower()
4
5 # Alternativ: "Vektorisiere"
6 abstracts = pd.Series(abstracts)
7 abstracts = abstracts.str.lower()
8 # (die native Liste erlaubt keine Vektorisierung)
9
```

2.6 Python-Einschub: Einblicke in die Funktion document_term_matrix (1/2)

Erinnerung: `document_term_matrix` erzeugt eine Matrix, die für Abstract i und Wort j angibt, wie oft j in i vorkommt.

```
In [66]: 1 # Zerlege einen Abstract in Wörter
2 a = abstracts_ds[0]
3 a.split(" ")
4
```

```
Out[66]: ['although',
'data_science',
'builds',
'on',
'knowledge',
'from',
'computer',
'science',
',',
'mathematics',
',',
'statistics',
',',
'and',
'other',
'disciplines',
',',
'data_science',
'is',
'a',
'unique',
'field',
'with',
'many',
'mysteries',
'to',
',',
```

```
In [73]: 1 # Zähle Wörter in einem Abstract
2 pd.Series(a.split(" ")).value_counts()
3
```

```
Out[73]: ,
data_science      4
a                  3
and                 3
of                  2
on                  2
questions            2
.
with                 2
for                  2
starts                1
builds                1
knowledge              1
computer              1
from                  1
article                1
to                   1
importance              1
ideas                 1
pressing                1
societal                1
discipline              1
unlock                 1
the                   1
is                   1
many                  1
scientific              1
~th~                1
```

2.6 Python-Einschub: Einblicke in die Funktion document_term_matrix (2/2)

Erinnerung: `document_term_matrix` erzeugt eine Matrix, die für Abstract i und Wort j angibt, wie oft j in i vorkommt.

```
In [76]: 1 # Alternativ: Vektorisiere und wende das Wörterzählen auf alle Abstracts gleichzeitig an:  
2 pd.Series(abstracts_ds)[:3].apply(  
3     lambda x: pd.Series(x.split(" ")).value_counts()  
4 )
```

Out[76]:

| | , | data_science | a | and | of | on | questions | . | with | for | ... | level | irrelevant | overview | have | familiar | risk | outlines | could | should | developing |
|---|------|--------------|-----|-----|------|-----|-----------|-----|------|-----|-----|-------|------------|----------|------|----------|------|----------|-------|--------|------------|
| 0 | 4.0 | 4.0 | 3.0 | 3.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| 1 | 13.0 | 14.0 | 5.0 | 8.0 | 10.0 | 2.0 | NaN | 7.0 | NaN | 3.0 | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| 2 | 4.0 | 6.0 | 2.0 | 2.0 | 2.0 | NaN | NaN | 4.0 | 1.0 | 2.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |

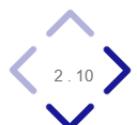
3 rows × 163 columns

```
In [77]: 1 # Finaler Befehl, wie in der Funktion verwendet:  
2 pd.Series(abstracts_ds)[:3].apply( lambda x: pd.Series(x.split(" ")).value_counts() ).fillna(0.0).astype(int)  
3
```

Out[77]:

| | , | data_science | a | and | of | on | questions | . | with | for | ... | level | irrelevant | overview | have | familiar | risk | outlines | could | should | developing |
|---|----|--------------|---|-----|----|----|-----------|---|------|-----|-----|-------|------------|----------|------|----------|------|----------|-------|--------|------------|
| 0 | 4 | 4 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 13 | 14 | 5 | 8 | 10 | 2 | 0 | 7 | 0 | 3 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 4 | 6 | 2 | 2 | 2 | 0 | 0 | 4 | 1 | 2 | ... | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

3 rows × 163 columns



2.7 Interpretation der Document-Term-Matrizen (1/6)

```
In [78]: 1 # Document-Term-Matrix für "data science"  
2 docterms_ds  
3
```

Out[78]:

| | data_science | a | and | of | on | questions | . | with | for | ... | oblivio |
|------|--------------|-----|-----|-----|-----|-----------|-----|------|-----|-----|---------|
| 0 | 4 | 4 | | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 0 |
| 1 | 13 | 14 | | 5 | 8 | 10 | 2 | 0 | 7 | 0 | 3 |
| 2 | 4 | 6 | | 2 | 2 | 2 | 0 | 0 | 4 | 1 | 2 |
| 3 | 0 | 2 | | 0 | 1 | 1 | 0 | 0 | 2 | 0 | 1 |
| 4 | 0 | 1 | | 2 | 1 | 2 | 0 | 0 | 1 | 0 | 0 |
| ... | ... | ... | | ... | ... | ... | ... | ... | ... | ... | ... |
| 1530 | 5 | 1 | | 6 | 12 | 13 | 0 | 0 | 12 | 1 | 1 |
| 1531 | 19 | 1 | | 4 | 8 | 6 | 2 | 0 | 10 | 2 | 1 |
| 1532 | 11 | 1 | | 2 | 17 | 7 | 1 | 2 | 11 | 3 | 6 |
| 1533 | 12 | 1 | | 12 | 8 | 13 | 1 | 0 | 11 | 2 | 4 |
| 1534 | 14 | 1 | | 16 | 5 | 27 | 6 | 0 | 13 | 6 | 5 |

1535 rows × 16748 columns

```
In [79]: 1 # Document-Term-Matrix für "data analytics"  
2 docterms_da  
3
```

Out[79]:

| | a | law-regulated | of | sensing | data_analytics | , | . | big | the | introduces |
|------|-----|---------------|----|---------|----------------|-----|---|-----|-----|------------|
| 0 | 2 | 2 | | 2 | 2 | 2 | | 2 | 2 | 2 |
| 1 | 1 | 0 | | 5 | 0 | 6 | | 7 | 7 | 9 |
| 2 | 4 | 0 | | 7 | 0 | 6 | | 5 | 7 | 7 |
| 3 | 0 | 0 | | 3 | 0 | 4 | | 3 | 6 | 0 |
| 4 | 5 | 0 | | 12 | 0 | 7 | | 14 | 12 | 9 |
| ... | ... | ... | | ... | ... | ... | | ... | ... | ... |
| 1105 | 9 | 0 | | 8 | 0 | 1 | | 10 | 11 | 1 |
| 1106 | 8 | 0 | | 11 | 0 | 1 | | 9 | 10 | 0 |
| 1107 | 4 | 0 | | 15 | 0 | 1 | | 14 | 15 | 0 |
| 1108 | 6 | 0 | | 8 | 0 | 1 | | 11 | 11 | 1 |
| 1109 | 7 | 0 | | 8 | 0 | 0 | | 4 | 9 | 0 |

1110 rows × 13708 columns

2.7 Interpretation der Document-Term-Matrizen (2/6)

Idee: Verwende die absoluten Häufigkeiten als Indikator für die Relevanz unterschiedlicher Wörter (links: data science, rechts: data analytics)

```
In [85]: 1 # Absolute Häufigkeiten
2 docterms_ds.mean(axis=0)
3 # or: docterms_ds.sum(axis=0)/len(docterms_ds)
4
```

```
Out[85]: , 9.373941
data_science 1.291205
a 4.048860
and 6.327687
of 6.631270
...
1-norm 0.000651
reformulated 0.000651
moment-sos 0.000651
sole 0.000651
meager 0.000651
Length: 16748, dtype: float64
```

```
In [87]: 1 # Absolute Häufigkeiten
2 docterms_da.mean(axis=0)
3 # or: docterms_da.sum(axis=0)/len(docterms_da)
4
```

```
Out[87]: a 3.926126
law-regulated 0.001802
of 6.665766
sensing 0.035135
data_analytics 1.081081
...
rein 0.000901
communication-and-computing 0.000901
multiple-input-multiple-output 0.000901
over-the-air 0.000901
eigen-gaps 0.000901
Length: 13708, dtype: float64
```

2.7 Interpretation der Document-Term-Matrizen (3/6)

Idee: Verwende die absoluten Häufigkeiten als Indikator für die Relevanz unterschiedlicher Wörter

Python-Einschub: Slicing

```
In [103]: 1 # Slicing zum Extrahieren [:bis (exkl.)]
          2 [1,2,3,4,5][:3]
          3
```

Out[103]: [1, 2, 3]

```
In [104]: 1 # Slicing zum Extrahieren [von (inkl.):bis (exkl.)]
          2 [1,2,3,4,5][1:3]
          3
```

Out[104]: [2, 3]

Der optionale dritte Parameter gibt die Schrittweite an:

```
In [105]: 1 # zum Überspringen [von (inkl.):bis (exkl.):Schrittwei
          2 [1,2,3,4,5][::2]
          3
```

Out[105]: [1, 3, 5]

```
In [106]: 1 # zum Umdrehen mit negativer Schrittweite
          2 [1,2,3,4,5][::-1]
          3
```

Out[106]: [5, 4, 3, 2, 1]



2.7 Interpretation der Document-Term-Matrizen (4/6)

Idee: Verwende die absoluten Häufigkeiten als Indikator für die Relevanz unterschiedlicher Wörter

```
In [118]: 1 # Absolute Häufigkeiten, absteigend sortiert
2 docterms_ds.mean(axis=0).sort_values()[:-1][:50]
3
```

```
Out[118]: , 9.373941
the 9.250814
. 7.876873
of 6.631270
and 6.327687
to 4.627362
in 4.092508
a 4.048860
we 2.385668
for 2.358306
data 2.117915
is 2.022150
that 1.704886
)
this 1.569381
(
on 1.376547
as 1.343974
with 1.336808
data_science 1.291205
are 1.117915
from 0.884691
an 0.873616
by 0.859283
learning 0.779805
can 0.723779
be 0.695114
our 0.681020
```

```
In [119]: 1 # Absolute Häufigkeiten, absteigend sortiert
2 docterms_da.mean(axis=0).sort_values()[:-1][:50]
3
```

```
Out[119]: the 9.723423
,
9.484685
.
8.472973
and 6.723423
of 6.665766
to 4.826126
a 3.926126
in 3.821622
data 3.145946
for 2.418919
we 2.123423
is 2.098198
)
1.770270
(
1.698198
that 1.674775
this 1.626126
on 1.594595
as 1.368468
with 1.300000
are 1.263964
data_analytics 1.081081
by 0.973874
an 0.878378
from 0.836036
can 0.779279
big 0.708108
be 0.685586
our 0.615915
```

2.7 Interpretation der Document-Term-Matrizen (5/6)

Idee: Verwende die absoluten Häufigkeiten als Indikator für die Relevanz unterschiedlicher Wörter

```
In [137]: 1 freqs_ds = docterms_ds.mean(axis=0)
2 freqs_da = docterms_da.mean(axis=0)
3 quot = (freqs_ds+1).divide(freqs_da+1,fill_value=1).sort_values() # Absoluten Häufigkeiten (normiert), durcheinander geteilt
4
```

```
In [139]: 1 quot[-30:][::-1] # Typische Begriffe für "data science"
2
```

```
Out[139]: data_science    2.291205
\           1.153612
methods      1.131109
problems     1.109587
learning     1.096937
scientists   1.096029
$             1.087117
students     1.084371
we            1.083961
models        1.082986
statistics   1.081893
:              1.081551
quantum       1.076444
theory        1.072907
neural         1.067502
statistical  1.065075
structure     1.065047
problem       1.063391
field          1.062823
community     1.062617
python         1.062310
```

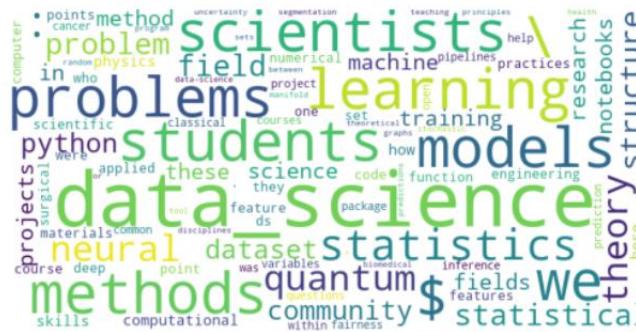
```
In [140]: 1 quot[:30] # Typische Begriffe für "data analytics"
2
```

```
Out[140]: data_analytics    0.480519
big            0.654476
data           0.752040
iot             0.812408
system          0.824410
processing      0.828644
analytics       0.852658
privacy          0.856708
distributed      0.858446
smart            0.859935
systems          0.861617
cloud            0.863325
performance      0.863435
computing         0.863479
applications     0.879491
devices          0.889183
memory           0.891205
spark             0.901306
analytic          0.906080
edge              0.909834
such              0.913152
storage           0.913545
```

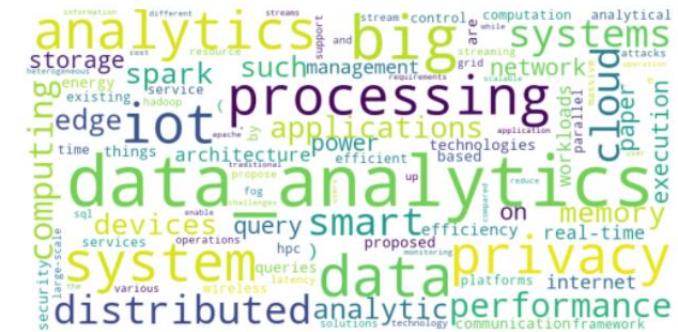
2.7 Interpretation der Document-Term-Matrizen (6/6)

Idee: Verwende die absoluten Häufigkeiten als Indikator für die Relevanz unterschiedlicher Wörter

```
In [159]: 1 from wordcloud import WordCloud
2 wc = WordCloud ( background_color="white",
3                  width=800, height=400, random_state=1
4                  )
5 wc.generate_from_frequencies ( quot[-100:][::-1] )
6 wc.to_file ( "VL01_Material/wc_ds.png" )
7 plt.axis("off")
8 plt.imshow(wc);
```



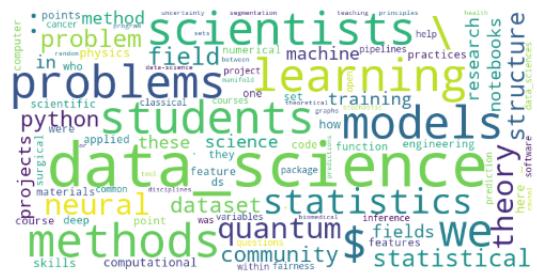
```
In [160]: 1
2 wc = WordCloud ( background_color="white",
3                  width=800, height=400, random_state=1
4                  )
5 wc.generate_from_frequencies ( 1/quot[:100] )
6 wc.to_file ( "VL01_Material/wc_da.png" )
7 plt.axis("off")
8 plt.imshow(wc);
```



2.8 Zusammenfassung

Auswertung von Publikations-Abstracts: Was unterscheidet Data Science von Data Analytics?

- **Methodik:** Sog. Bag-of-Words-Ansatz (Reihenfolge spielt keine Rolle)
- **Möglicher nächster Schritt:** Zähle Bigramme ("and the") oder n-Gramme
- **Perspektivisch:** Natural Language Processing mit Neuronalen Netzwerken



Tendenz: eher "Forschung"



Tendenz: eher "Implementation" und "Produkt"

2.9 Epilog: Wir können auch einfach ChatGPT fragen (1/2)

Kannst du mir den Unterschied zwischen Data Science und Data Analytics erklären?

Ja, gern!

Data Science und Data Analytics sind eng verwandte Bereiche, die sich mit der Verarbeitung und Analyse von Daten beschäftigen. Allerdings gibt es einige Unterschiede zwischen ihnen.

Data Analytics bezieht sich im Allgemeinen auf die Analyse von Daten, um Einblicke und Erkenntnisse zu gewinnen. Es konzentriert sich darauf, Fragen zu beantworten und Entscheidungen auf der Grundlage von Daten zu treffen. Data Analytics kann in der Regel in zwei Hauptkategorien unterteilt werden: Deskriptive Analyse und Prädiktive Analyse.

Deskriptive Analyse beschreibt, was in der Vergangenheit passiert ist, während prädiktive Analyse versucht, zukünftige Ereignisse vorherzusagen. Data Analytics konzentriert sich normalerweise auf eine bestimmte Fragestellung und versucht, eine bestimmte Hypothese zu beweisen oder zu widerlegen.

Data Science hingegen umfasst mehrere Disziplinen wie Mathematik, Statistik, Informatik und maschinelles Lernen. Es geht darum, Daten zu sammeln, zu bereinigen, zu modellieren und zu analysieren, um Muster und Erkenntnisse zu gewinnen. Data Science kann als ein Prozess betrachtet werden, der sich mit der Extraktion von Wissen aus Daten befasst.

Ein Data Scientist erstellt und trainiert Modelle, um Vorhersagen zu treffen und Entscheidungen zu treffen. Data Science ist nicht nur auf die Beantwortung spezifischer Fragen beschränkt, sondern konzentriert sich auf die Entdeckung von Erkenntnissen und die Identifizierung von Mustern, die zur Lösung komplexer Probleme beitragen können.

Zusammenfassend lässt sich sagen, dass Data Analytics sich darauf konzentriert, Fragen zu beantworten, während Data Science sich darauf konzentriert, Erkenntnisse und Muster zu entdecken, die zur Lösung komplexer Probleme beitragen können.

Bitte erstelle einen Vergleich in Tabellenform.

Gern! Hier ist eine Tabelle, die den Vergleich zwischen Data Science und Data Analytics darstellt:

| Eigenschaften | Data Science | Data Analytics |
|------------------------|---|---|
| Zielsetzung | Entdeckung von Mustern und Erkenntnissen | Beantwortung spezifischer Fragen |
| Fokus | Problemlösung | Entscheidungsfindung auf der Grundlage von Daten |
| Schwerpunkt | Mathematik, Statistik, Informatik und maschinelles Lernen | Mathematik, Statistik und explorative Datenanalyse |
| Umfang | Umfasst mehrere Disziplinen und Methoden | Konzentriert sich auf Analyse und Interpretation von Daten |
| Prozess | Extraktion von Wissen aus Daten | Analyse von Daten zur Beantwortung von Fragen |
| Ziele der Analyse | Identifizierung von Mustern und Erkenntnissen | Deskriptive Analyse zur Beschreibung von Daten |
| Art der Analyse | Explorative Analyse, Modellierung und Vorhersage | Deskriptive Analyse und prädiktive Analyse |
| Anwendungen | Vorhersage, Klassifikation, Mustererkennung, usw. | Kundenanalyse, Marktsegmentierung, Betragssanalyse, usw. |
| Werkzeuge und Methoden | Maschinelles Lernen, Data Mining, Big Data, usw. | Statistische Analysen, Datenvizualisierung, Datenbankabfragen |

Ich hoffe, diese Tabelle hilft Ihnen, die Unterschiede zwischen Data Science und Data Analytics besser zu verstehen.

2.9 Epilog: Wir können auch einfach ChatGPT fragen (2/2)

 Bitte erkläre mir den Unterschied nun in maximal zwei Sätzen.



Data Science geht darum, Erkenntnisse und Muster in Daten zu entdecken und komplexe Probleme zu lösen, während Data Analytics sich auf die Beantwortung spezifischer Fragen und die Entscheidungsfindung auf der Grundlage von Daten konzentriert.



Data Science

Einsatzszenario: (eher) Forschung und Entwicklung
Zielgruppe: (eher) Wissenschaftler



Foto von [Alejandro Piñero Amerio](#) auf [Unsplash](#)

Data Analytics

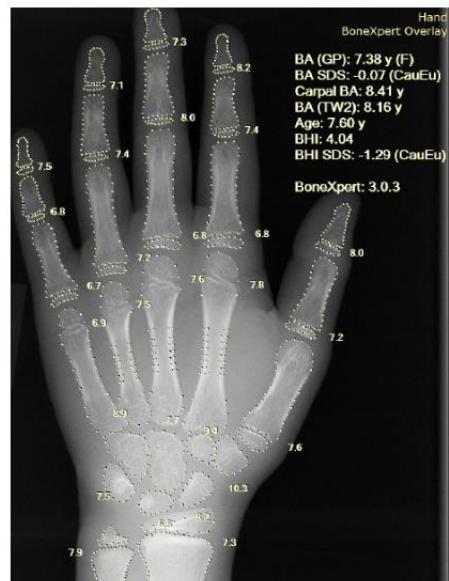
Einsatzszenario: (eher) im laufenden Betrieb
Zielgruppe: (eher) Nicht-Wissenschaftler



Foto von [Stephen Dawson](#) auf [Unsplash](#)

3. Zusammenfassung und Semesterplanung (1/2)

Inhalt der heutigen Vorlesung: Kennenlernen + Organisatorisches + Einblicke in Python und Datenauswertung



Erkennung von Knochenentwicklung bei Kindern, <https://www.bonexpert.com>

Inhalte (Grundlagen):

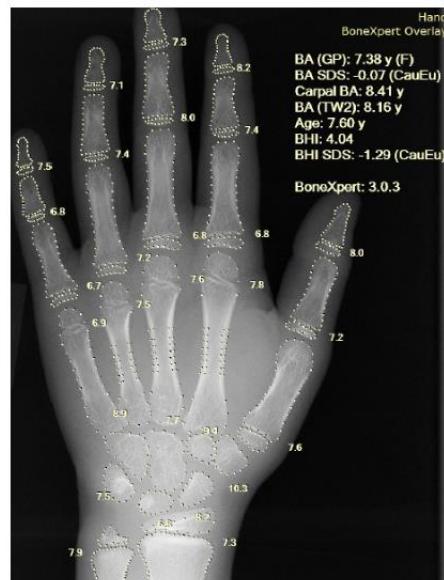
- Python und Jupyter Notebook
- Datenvisualisierung
- Deskriptive Statistik (Mittelwert, Varianz, ...)

Inhalte (erweitert):

- Unsupervised Learning: Clustering
- Supervised Learning: Regression und Klassifikation
- Grundlagen zum Datenmanagement

3. Zusammenfassung und Semesterplanung (2/2)

Inhalt der heutigen Vorlesung: Kennenlernen + Organisatorisches + Einblicke in Python und Datenauswertung



Erkennung von Knochenentwicklung bei Kindern, <https://www.bonexpert.com>

Lernziel:

Sie sollen befähigt werden, Datenprobleme in der Praxis zu lösen und datengestützte Entscheidungen zu erarbeiten, aber auch die Grenzen der Methoden kennenzulernen.

Im Rahmen dieser Vorlesung:

HDA = EDS + Medizinbeispiele

Vielen Dank für Ihre Aufmerksamkeit!

Anhang: Python-Code

Imports

```
In [ ]: 1 # Standard-Imports
2 import numpy as np
3 import pandas as pd
4 import matplotlib.pyplot as plt
5
6 # Besondere Imports für dieses Einsatzszenario
7 import time
8 import requests
9 import urllib
10 import feedparser
11
```

Funktion query_arxiv

```
In [ ]: 1 def query_arxiv ( search_phrases, fpath_result="abstracts.txt", batchsize=5000, max_batches=50,
2                         base_url="http://export.arxiv.org/api/query?", with_dates=False, silent=True ):
3     # based on https://static.arxiv.org/static/arxiv.markdown
4     # /0.1/help/api/examples/python_arXiv_parsing_example.txt
5
6     # based on the given search_phrases, build the url to connect
7     # to the arxiv API
8     search_string = "+OR+".join(["abs:%22"+s.replace(" ", "+")+"%22" for s in search_phrases])
9     url = base_url + "search_query="+search_string+"&max_results="+str(batchsize)
10
11    # request data in batches and write to the specified file
12    with open(fpath_result,"w") as f:
13        for i in range(max_batches):
14            # open connection
15            data = urllib.request.urlopen(url+"&start="+str(i*batchsize)).read()
16            feed = feedparser.parse(data)
17            totalresults = int(feed.feed.opensearch_totalresults)
18
19            # in the first iteration: report the total number of results; end loop if queue is empty
20            # (note that this differs from len(feed.entries) because of the argument max_results in the query
21            # which limits each query to the batch size)
22            if i == 0:
23                if not silent: print("Number of results: "+str(totalresults))
24            if len(feed.entries) == 0:
25                break
26            if not silent: print("Batch "+str(i))
27
28            # collect lines in a buffer and append the buffer to the output file.
29            # apply rudimentary masking to special characters.
30            buffer = []
31            for entry in feed.entries:
32                if with_dates:
```

Funktion `read_data`

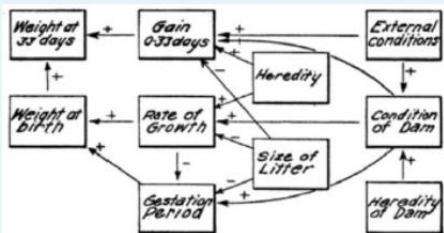
```
In [ ]: 1 def read_data ( fpath, search_phrases ):
2     # read and cleanup; store as a pandas Series for convenience
3     with open(fpath,"r") as f:
4         abstracts = pd.Series( f.read().splitlines() ).str.lower()
5         cleanup = {s: " "+s+" " for s in [")","[",",",".",";","+",":","!","?",",","/","\\","'","~","#"]}
6     {
7         "(": " ( ",
8         ")": " )",
9         "[": " [",
10        "]": " ]",
11        ",": " ,",
12        ".": " .",
13        ";": " ;",
14        "+": " +",
15        "!": " !",
16        "?": " ?",
17        "\\"": " \\" ",
18        "'": " '",
19        "~": " "
20    }
21
22    # add spaces around special characters; remove multiple spaces
23    for k,v in zip(cleanup.keys(), cleanup.values()):
24        abstracts = abstracts.str.replace(k, v, regex=False).str.replace("\s+", " ", regex=True).str.strip()
25
26    # protect search phrases
27    for s in search_phrases:
28        abstracts = abstracts.str.replace(s, s.replace(" ","_"), regex=False)
29
30    # ensure that keywords are kept together: replace spaces by underscores
31    search_phrases_protected = [s.replace(" ","_") for s in search_phrases]
32
```

Funktion document_term_matrix

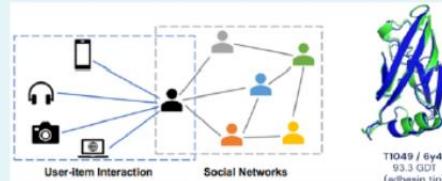
```
In [ ]: 1 def document_term_matrix ( abstracts, batchsize=1000, remove_seldom_quantile=0.0, silent=True ):
2     # re-convert to series for convenience
3     abstracts = pd.Series(abstracts)
4
5     # count word occurrences in each abstract. for memory reasons, do this in batches
6     for b in range( int ( np.ceil ( len(abstracts)/batchsize) ) ):
7         if not silent: print( "Batch "+str(b))
8         _abstracts_countmatrix = abstracts[b*batchsize:(b+1)*batchsize].apply(
9             lambda x:pd.Series(x.split("|")[-1].split(" ")).value_counts()).fillna(0.0).astype(int)
10
11     # throw away very seldom words (unless contained in the search phrases)
12     if remove_seldom_quantile > 0:
13         idx_keep = _abstracts_countmatrix.sum(axis=0).sort_index()
14         idx_keep = idx_keep.index [ (idx_keep > idx_keep.quantile(0.10))
15                                     | idx_keep.index.isin(search_phrases_protected) ]
16         _abstracts_countmatrix = _abstracts_countmatrix.loc[:,idx_keep].copy()
17
18     # in the first iteration, use the temporary matrix as the starting point;
19     # in later iterations, concatenate
20     if ( b == 0 ):
21         abstracts_countmatrix = _abstracts_countmatrix
22     else:
23         abstracts_countmatrix = pd.concat([abstracts_countmatrix,_abstracts_countmatrix],
24                                         axis=0)
25
26     # up to now, the "count" for a term that was not found in a document is NA;
27     # it is more natural to have zeros here
28     abstracts_countmatrix = abstracts_countmatrix.fillna(0.0).astype("int")
29
30 return ( abstracts_countmatrix )
```

Anhang: Abschlussarbeitsthemen (1/3)

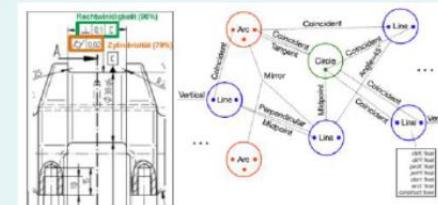
Causal Inference: Modellierung kausaler Zusammenhänge



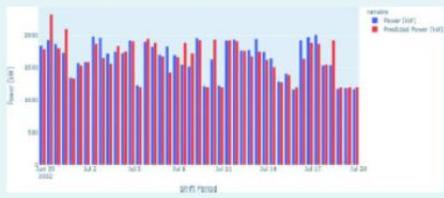
Deep Learning auf Graphen



Deep Learning zur Analyse technischer Zeichnungen



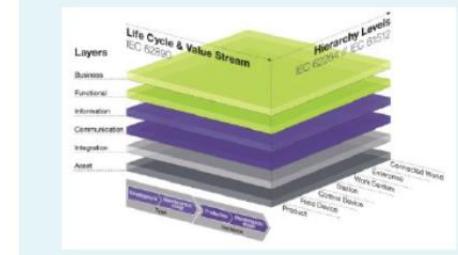
Energieverbrauchsprognose für CNC-Werkzeugmaschinen basierend auf semantischen Datenströmen



KI-basierte Vorhersage von Materialeigenschaften



RAMI 4.0 in der zerspanenden Industrie

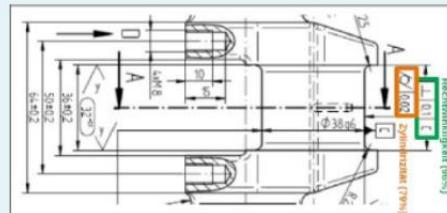


Anhang: Abschlussarbeitsthemen (2/3)

Simulation physikalischer Prozesse
beim Zerspanen



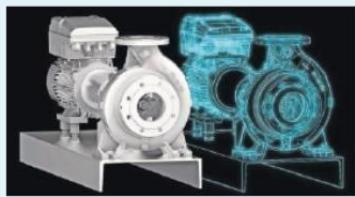
Anforderungsanalyse und
Bewertung von Frä-, Dreh- und
Bohrbearbeitungsszenarien



Reinforcement Learning zur
Rekonstruktion von CAD-Modellen



Entwicklung eines Augmented-
Reality-Prototypen zur Visualisierung
von Kräften und Toleranzen am
Digitalen Zwilling in der Zerspanung



Multimodale Neuronale Netzwerke
in der Fertigung

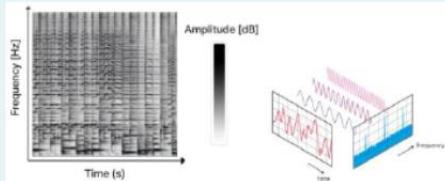


Novelty Detection in der Analyse
großer Textdatenbestände



Anhang: Abschlussarbeitsthemen (3/3)

Stable Diffusion zur Erzeugung synthetischer Sensordaten



KI-basierte Analyse von Konstruktionsanleitungen



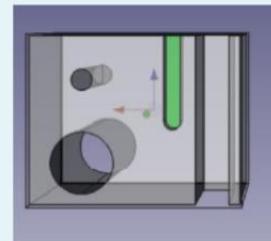
Computer Vision-basierte Automatisierte Bewertung von Werkzeugschneiden



Transformer-basierte Large Language Models



KI-basierte Feature-Erkennung in CAD-Modellen



Reinforcement Learning zur Planung von Fräsbahnen

