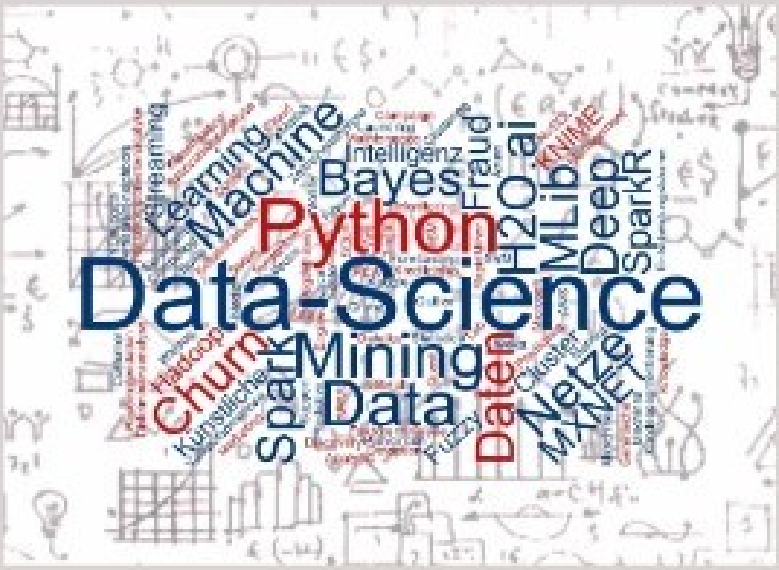
MICHAEL OETTINGER

DATA SCIENCE UND AI

Eine praxisorientierte Einführung im Umfeld von Machine Learning, künstlicher Intelligenz und Big Data



3. erweiterte Auflage

Michael Oettinger

Data Science und AI

Eine praxisorientierte Einführung im Umfeld von Machine Learning, künstlicher Intelligenz und Big Data

3. erweiterte Auflage

Druck und Distribution im Auftrag des Autors:
tredition GmbH, Heinz-Beusen-Stieg 5, 22926 Ahrensburg, Germany

Inhalt

1 Einleitung

Ich habe im Jahr 2017 die erste Auflage dieses Buches geschrieben, weil ich meinen Job im Softwarevertrieb für ein kleineres US-Softwareunternehmen gekündigt hatte und mich als Data-Scientist selbstständig machen wollte. Ich hatte keine Referenzen und Kunden und dachte mir, dass man einem Autor eines Fachbuches die Kompetenz für das Fachgebiet unterstellen kann und ich so den Eintritt in die Welt der Freelancer und vor allem in meine ersten Projekte finden kann.

Spoiler: Es hat funktioniert und ich arbeite seit nunmehr sieben Jahren in unterschiedlichen Projekten für unterschiedliche Kunden (übrigens bei Vollauslastung und vor allem bei 'Vollzufriedenheit').

Nach drei Jahren war es Zeit für ein Update, sodass im Jahr 2020 eine zweite aktualisierte Auflage des Buches veröffentlicht wurde.

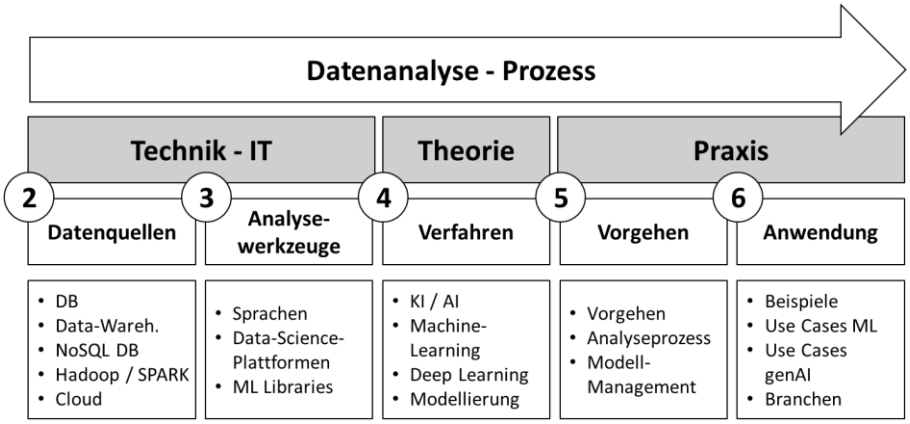
Die Gründe für die Erstellung der hier vorliegenden dritten Auflage sind:

- Data-Science und vor allem die dazugehörende Softwaretechnologie haben sich weiterentwickelt.
- Spätestens mit der Veröffentlichung von ChatGPT ist das Thema künstliche Intelligenz in aller Munde und eine Einordnung von Data-Science, Machine Learning und Artificial Intelligence scheint dringend notwendig.
- Mit der Projekterfahrung der letzten sechs Jahre würde ich das Buch heute anders gestalten. Die Grundstruktur und der Aufbau passen, aber an vielen Stellen würde ich heute deutlichere Aussagen machen. Weniger Abwägen und wissenschaftliches Erarbeiten und mehr deutliche, pragmatische Empfehlungen geben.
  Also habe ich mich hingesetzt und das Buch aufgefrischt. Wenn man so will, ist es nun polemischer geworden, da ich in allen Kapiteln 'meinungsstarke' und deutliche Kommentare eingefügt habe. Aber es ist auch pragmatischer, da es nun Code-Beispiele in Python bzw. SQL enthält und einige Cheat-Sheets angefügt wurden.

Bedanken möchte ich mich an dieser Stelle bei allen Kunden und Kollegen, die mir das Vertrauen geschenkt und mir so ermöglicht haben, als Data Scientist zu arbeiten. Außerdem möchte ich mich bei meiner Frau und meiner Tochter bedanken. Ohne meine Familie wäre zwar dieses Buch wahrscheinlich einen oder zwei Monate früher fertig geworden, aber mein Leben wäre sinnloser gewesen.

Gliederung des Buches

Das Buch ist folgendermaßen gegliedert:



Nach der Einführung in **Kapitel 1** orientiert sich die Gliederung des Buches am Prozess der Datenanalyse. Von der Datenquelle geht es über die verwendeten Werkzeuge und die eingesetzten Verfahren bis hin zum konkreten Vorgehen und Beispielen in der Praxis.

Kapitel 2 enthält Erläuterungen zu den ‘Datentöpfen‘ aus einer technischen Perspektive. Wo und wie werden die Daten bereitgestellt, die als Quelle für die Datenanalyse herangezogen werden? Konkret werden die am weitesten verbreiteten Arten von Datenbanken vorgestellt:

- Flatfiles,
- ODBC-Datenbanken,
- Data-Warehouse,
- NoSQL-Datenbanken,
- Hadoop- und Spark-Plattformen sowie
- Cloud-Speicher.

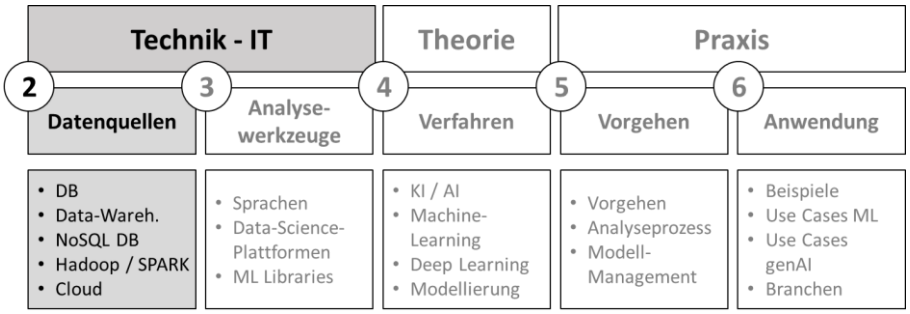
In **Kapitel 3** wird auf die Werkzeuge – d. h. die Softwarelösungen – eingegangen, mit denen die Daten analysiert werden. Dabei wird zwischen den wichtigsten Sprachen (SQL, Python, R), den Data-Science-Plattformen und den Machine Learning-Librarys unterschieden. Zu diesen Softwareanwendungen gibt es sowohl Open-Source- als auch kommerzielle Angebote.

In **Kapitel 4** wird auf die gebräuchlichsten Analyseverfahren eingegangen. Dabei handelt es sich um Verfahren aus den Bereichen Statistik, Mathematik, Machine Learning, künstliche Intelligenz und Computer-Science. Es wird versucht, die Verfahren zu strukturieren und im Einzelnen so darzustellen, dass ein Grundverständnis für ihre Möglichkeiten und Grenzen aufgebaut werden kann.

Kapitel 5 ist der Praxis gewidmet, indem erläutert wird, wie Analytics-Projekte in Unternehmen oder Forschungseinrichtungen durchgeführt werden. Die bewährten Vorgehensmodelle werden vorgestellt. Außerdem wird auf das Thema Modellmanagement eingegangen. Dies ist vor allem dann wichtig, wenn in größeren Teams zusammengearbeitet wird und über die Zeit eine Vielzahl von Analysemodellen erstellt, getestet, angepasst und wieder verworfen wird.

In **Kapitel 6** werden Use-Cases – d. h. Anwendungsfälle – für die besprochenen Verfahren und Techniken vorgestellt. Dabei geht es nicht nur um konkrete Einzelfälle, sondern auch um den Versuch, ein Bild über mögliche Einsatzszenarien zu geben. Die Use-Cases werden vorgestellt und die Besonderheiten ausgewählter Branchen werden diskutiert.

2 Daten bereitstellen



Data-Science bezeichnet den Prozess, durch die Analyse von Daten mit geeigneten Verfahren Erkenntnisse zu gewinnen. Die erste Frage, die sich stellt, ist diejenige nach der Quelle der Daten. Woher kommen die zu analysierenden Daten und wo und wie werden sie bereitgestellt? Im Folgenden wird auf diese *Datenquellen* näher eingegangen. Konkret handelt es sich dabei um:

- Flatfiles
- Relationale Datenbanken
- Data-Warehouses
- NoSQL-Datenbanken
- Hadoop
- Cloud-Datenbanken

2.1 Flatfiles

Die einfachste Form der Datenbereitstellung sind Flatfiles, also Tabellen und strukturierte Textdateien, die man aus operativen Systemen wie z. B. ERP-Systemen exportiert oder über Befragungen gewonnen hat. Die Dateien werden in unterschiedlichen Formaten zur Verfügung gestellt. Die gebräuchlichsten sind:

- csv
- xls
- xml
- produktspezifische Formate (SPSS, SAS, Stata, ARFF, DBase …)

Bei dieser Form der Datenanalyse handelt es sich meist nicht um ‘Big Data’ (auch wenn die Größe der Files grundsätzlich nahezu unbegrenzt sein kann), aber dennoch spielen Flatfiles nach wie vor eine wichtige Rolle in Data-Science-Projekten. Es muss z.B. kein Zugang zur Datenbank eines Produktivsystems eingerichtet werden, was meist einen höheren Aufwand im Bereich Berechtigungen und Netzwerkzugang bedeutet. Stattdessen werden die Daten aus dem Quellsystem exportiert und dann in das Analysesystem eingelesen, wo die eigentliche Analyse bzw. Modellierung stattfindet. Liegt eine sehr hohe Anzahl an Flatfiles vor, bietet es sich an, den Prozess des Einlesens und Zusammenfassen der Daten z. B. durch ein Programm in Python zu automatisieren.

2.2 Relationale Datenbanksysteme

Relationale Datenbanksysteme dienen der Datenverwaltung und beruhen auf einem tabellenbasierten, relationalen Datenbankmodell. Sie werden auch als RDBMS (Relational Database Management System) bezeichnet. Zum Abfragen und Manipulieren der Daten wird überwiegend die Datenbanksprache SQL (Structured Query Language) eingesetzt.

Relationale Datenbanken folgen einem grundsätzlichen Schema. Daten werden in Tabellen gespeichert, wobei die Spalten die Variablen darstellen und die Zeilen die einzelnen Datensätze. Datenbanken werden dadurch ‘relational’, dass es Relationen – also Verbindungen – zwischen den Tabellen gibt. Diese werden eingeführt, um eine redundante Speicherung der gleichen Daten

zu vermeiden. Damit wird Speicherplatz gespart und inkonsistente Datenhaltung vermieden. Beispielsweise werden bei einer Datenbank für *Kunden* nicht für jeden einzelnen Kunden die Unternehmensdaten angegeben, sondern die Kategorie *Unternehmen* wird als eigenständige Tabelle ausgelagert und über eine Relation den einzelnen Kunden zugeordnet. Ändert sich etwas an der Adresse des Unternehmens, muss dies nur an einer Stelle geändert werden – durch die Relation wird den einzelnen Kunden automatisch das entsprechende Unternehmen zugeordnet.

Tabelle	Kunde	
Name	**Vorname**	**Unternehmen**
Karl	Mustermann	U1
Peter	Müller	U2
Claudia	Maier	U1
…		

Tabelle	Unternehmen		
UntNr	**Unternehmen**	**Strasse**	**Ort**
U1	ACME	Goethestr. 1	Berlin
U2	Müller GmbH	Haupttsr. 2	Hamburg
U3	ABC AG	Schillerstr. 1	Essen
…			

Trotz neuerer Entwicklung (siehe den folgenden Abschnitt) stellen relationale Datenbanken nach wie vor die große Mehrzahl der Datenspeicher in Unternehmen dar und sind zentraler Bestandteil der meisten operativen Anwendungen (ERP, CRM, HCM, SCM, Fachsysteme …).

Die wichtigsten Anbieter sind:

- Oracle (Marktführer nach Umsatz)
- Microsoft SQL Server (Marktführer in bestimmten Märkten und auf bestimmten Plattformen)
- MySQL (Open Source, von Oracle erworben, höchste Anzahl an Implementierungen)
- PostgreSQL (Open Source)
- IBM DB2
- SAP Adaptive Server / SQL Anywhere / SAP MaxDB
- Amazon RDS (Cloud-Angebot für RDBS)

2.3 Data-Warehouse

Ein Data-Warehouse (DW oder DWH) ist eine zentrale Sammlung von Daten, die sich aus verschiedenen Quellen speist und vor allem für den Zweck der Analyse und der betriebswirtschaftlichen Entscheidungshilfe dauerhaft gespeichert wird.

Meistens wird ein Data-Warehouse aus zwei Gründen aufgebaut:

- Es soll eine **Integration** von Daten aus verteilten und unterschiedlich strukturierten Datenbeständen erfolgen. Im Data-Warehouse können die Daten konsistent gesichtet und datenquellenübergreifend ausgewertet werden. Die zeitaufwendigen und technisch anspruchsvollen Aufgaben der Datenextraktion und -integration aus verschiedenen Systemen erfolgt damit (im Ideal) einmalig und an zentraler Stelle. Die Daten stehen dann für Analysen und Reporting für die Fachabteilungen ‘konsumbereit’ zur Verfügung.
- Durch eine **Trennung** der (oft ‘sensiblen’) Daten in den operativen Systemen von den für das Reporting genutzten Daten im Data-Warehouse soll sichergestellt werden, dass durch die Datenabfragen für Analysen und Reporting keine operativen Systeme ‘gestört’ werden. Niemand möchte, dass der Azubi in der Vertriebsabteilung durch eine Abfrage der kompletten, weltweiten Produktverkäufe, nach Wochen und Postleitzahl gegliedert, das Buchhaltungssystem für eine halbe Stunde lahmlegt.

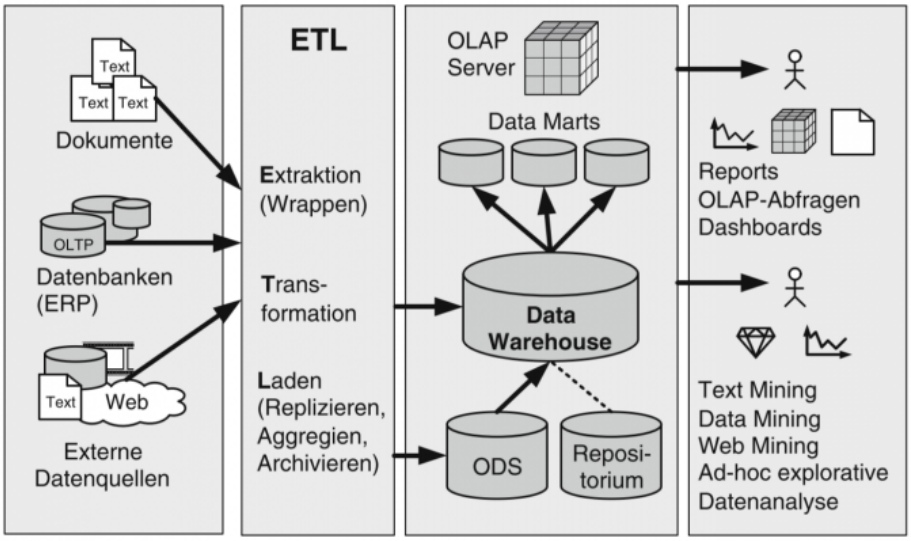


Abbildung 1: Architektur eines Data Warehouses (Müller & Lenz, 2013, S. 19)

Technisch gesehen sind Data-Warehouse-Systeme eine Sammlung von Softwarekomponenten, die die Umsetzung des Data-Warehouse-Konzeptes ermöglichen. Sie bestehen aus:

- **ETL-Komponenten**, die den ETL-Prozess (also die Extraktion, Transformation und das Loading der Daten) unterstützen,
- dem **Core-Data-Warehouse**, also einer Sammlung von gemanagten Datenbanksystemen, die auf Parallelisierung und Performance für das Handling riesiger Datenmengen optimiert sind,
- den 'vorbereiteten' **Aggregationen** (Star-Schemas), die Auswertungen beschleunigen.
- einem **User Interface**, das die Verwaltung und die Auswertung der Datenbestände ermöglicht.

Die wichtigsten Anbieter von Data-Warehouse-Systemen sind:

- Oracle
- Teradata

- Microsoft
- IBM
- SAP

Data Lake

In letzter Zeit wird immer häufiger der Begriff ‘Data Lake’ verwendet. Es handelt sich dabei um ein Konzept, das als eine Erweiterung des Data-Warehouse-Gedankens gesehen werden kann, der dann aber technisch mit Hadoop- oder NoSQL-Mitteln umgesetzt wird (siehe die folgenden zwei Abschnitte).

Im Unterschied zum Data-Warehouse, wo die Daten aus verschiedenen Quellen bezogen und dann so aufbereitet werden, dass sie vergleichbar sind und damit aggregiert werden können (ETL-Prozess), werden beim Data Lake die Daten erst einmal im ursprünglichen Format und unbearbeitet gesammelt. Eine Bearbeitung bzw. Transformation der Daten erfolgt dann erst bei Bedarf vor der eigentlichen Analyse (ELT-Prozess). Diese Vorgehensweise eignet sich also vor allem für

- eher unstrukturierte Daten, z. B. aus sozialen Medien, Blogbeiträgen, Bild- und Videodateien,
- strukturiertere XML- bzw. HTML-Daten,
- oder für Sensor-Daten.

Damit sind wir nun wirklich im Bereich Big Data angekommen. Die große Herausforderung ist es an dieser Stelle, diesen erstmal unbearbeiteten ‘Datensee’ tatsächlich für Analysen und damit einhergehend für den Erkenntnisgewinn zu nutzen. Ein Datentümpel, der ständig mit unnützen Datenmengen ergänzt wird und wächst und wächst, ist wertlos.

Die klassischen Analyseverfahren (siehe Abschnitt 4.5) sind für strukturierte Daten konzipiert. Eine Analyse der unstrukturierten Daten setzt also voraus, dass diese in irgendeiner Form strukturiert werden, um sie im Anschluss mit

den vorhandenen Verfahren analysieren zu können. Nur durch eine integrierte Datenstrategie, die die strukturierten und unstrukturierten Daten miteinbezieht, können die Schätze des Big Data tatsächlich gehoben werden.

2.4 NoSQL

Unter dem Begriff NoSQL werden unterschiedliche Arten von Datenverwaltungssystemen zusammengefasst. Ganz wichtig vorneweg: NoSQL steht **nicht** für ‘no SQL’, also ‘kein SQL’! Das ‘No’ bedeutet vielmehr ‘not only’. NoSQL ist also keine Anti-SQL-Bewegung, sondern soll eine Alternative bzw. Bereicherung zur SQL-Welt darstellen.

Den unterschiedlichen Ausprägungen von NoSQL-Datenbanken ist gemeinsam, dass sie für Anwendungsfälle geschaffen wurden, in denen die verfügbaren SQL-basierten Datenbanken an ihre Grenzen stießen und daher nicht oder nur mit sehr großem Aufwand einsetzbar waren.

Die Architektur vieler NoSQL-Datenbanken setzt auf den Einsatz einer großen Anzahl kostengünstiger Rechnersysteme zur Datenspeicherung, wobei die meisten Knoten gleichrangig sind. Eine Skalierung erfolgt dann einfach durch Hinzufügen weiterer Knoten.

NoSQL-Datenbanken unterscheiden sich hinsichtlich der Art der ‘Verschlüsselung’. Es gibt ‘Key-Value-Stores’ oder komplexere, dokumentenorientierte Ansätze, die zusätzlich zu Dokumenten noch Verknüpfungen zwischen Dokumenten bieten.

NoSQL-Datenbanken werden vor allem dann eingesetzt, wenn SQL-Datenbanken an ihre Grenzen stoßen. In NoSQL-Systemen lassen sich z. B. wesentlich größere Mengen an Daten performant ablegen und aufrufen. Bei komplexen Abfrageanforderungen, etwa im Bereich unstrukturierter Daten wie Video-, Audio- oder Bilddateien, erlauben einige NoSQL-Datenbanken baumförmige Strukturen der Metadaten ohne ein fest definiertes Datenschema und deren flexible Abfrage. Bei Daten mit schwankendem Typ und Inhalt eignen

sich NoSQL-Datenbanken besser, weil sich die Daten nicht länger in das ‘SQL-Korsett’ von Tabellen und Relationen pressen lassen müssen.

Man muss sich aber bewusst darüber sein, dass die Verfahren, mit denen aus Daten Erkenntnisse für eine Prognose gewonnen werden, auf strukturierte Daten angewiesen sind. Das bedeutet nicht, dass das ‘SQL-Korsett’ für die Rohdaten eingehalten werden muss, aber die Aufbereitung vor der Analyse erfordert eine Strukturierung. Bei der Verwendung von NoSQL-Datenbanken müssen daher die ja immer vorhandenen Strukturen der Datenhaltung beachtet und die entsprechende Aufbereitungsschritte angewendet werden.

Wichtige Anbieter von NoSQL-Datenbanken sind:[1]

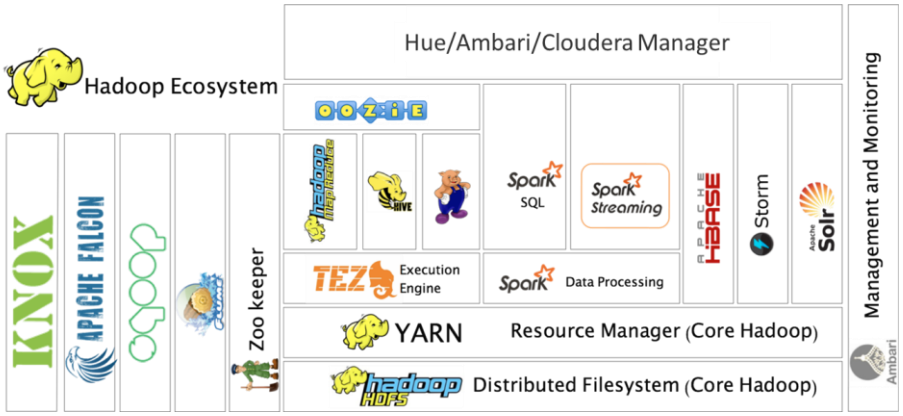
- MongoDB
- Cassandra
- Redis
- HBase
- Couchbase
- NoSQL-Angebote der Cloudanbieter wie AWS und MS Azure

2.5 Hadoop/Spark

Apache Hadoop ist ein Software-Framework, mit dessen Hilfe rechenintensive Prozesse mit großen Datenmengen auf Server-Clustern bearbeitet werden können. Anwendungen können mit der Unterstützung Hadoops komplexe Aufgaben auf Tausende von Rechnerknoten verteilen und Datenvolumina im Petabyte-Bereich verarbeiten. Es basiert ursprünglich auf dem MapReduce-Algorithmus und Grundideen des Google-Dateisystems. Hadoop wird von der Apache Software Foundation – einer Gemeinschaft von Entwicklern, die Open-Source-Softwareprodukte entwickeln – als Top-Level-Projekt vorangetrieben.

[1] Vgl.: http://nosql-database.org/

Hadoop besteht aus vier Kernmodulen und weiteren Komponenten, die zum Hadoop Ecosystem gerechnet werden.



Die vier Kernmodule sind:

- **Hadoop Common**: Hilfswerkzeug, das die Hadoop-Komponenten verwaltet bzw. unterstützt.
- **Hadoop Distributed File System** (HDFS): HDFS ist ein hochverfügbares Dateisystem zur Speicherung sehr großer Datenmengen auf den Dateisystemen mehrerer Rechner (Knoten). Dateien werden in Datenblöcke mit fester Länge zerlegt und redundant auf die teilnehmenden Knoten verteilt. Dabei gibt es Master- und Slave-Knoten. Ein Master-Knoten, der sogenannte NameNode, bearbeitet eingehende Datenanfragen, organisiert die Ablage von Dateien in den Slave-Knoten und speichert anfallende Metadaten. HDFS unterstützt dabei Dateisysteme mit mehreren 100 Millionen Dateien.
- **Hadoop YARN**: Eine Softwarelösung, die die Verwaltung der Ressourcen (also das Job-Scheduling) eines Clusters übernimmt.

- **Hadoop MapReduce**: Ein auf YARN basierendes System, das paralleles Prozessieren großer Datenmengen realisiert. Hadoop beinhaltet den MapReduce-Algorithmus, dieser gilt aber zunehmend als veraltet und wird durch graphenbasierte Verfahren (Spark, Tez) ersetzt.

Insbesondere **Spark** hat mittlerweile die größere Verbreitung im Hadoop-Umfeld und hat MapReduce als Prozess-Engine abgelöst.

Im Rahmen von Apache werden weitere Projekte als zum **Hadoop Ecosystem** zugehörig gezählt:

- **Ambari**: Ambari ist eine Managementplattform, die die Verwaltung (Provisionierung, Management, Monitoring) der Hadoop-Cluster vereinfachen soll. Unterstützt werden: HDFS, Hadoop MapReduce, Hive, HCatalog, HBase, ZooKeeper, Oozie, Pig and Sqoop.
- **Avro**: Avro ist ein System zur Serialisierung von Daten.
- **Cassandra**: Cassandra ist ein skalierbares NoSQL-Datenbanksystem für Hadoop-Cluster.
- **Chukwa**: Chukwa ermöglicht die Datensammlung und Echtzeitüberwachung sehr großer verteilter Systeme.
- **HBase**: HBase ist eine skalierbare Datenbank zur Verwaltung großer Datenmengen innerhalb eines Hadoop-Clusters. Die HBase-Datenbank basiert auf Googles BigTable. Diese Datenstruktur ist für Daten geeignet, die selten verändert, dafür aber häufig ergänzt werden. Mit HBase lassen sich Milliarden von Zeilen verteilt und effizient verwalten.
- **Hive**: Hive ist eine Data-Warehouse-Infrastrukturkomponente, die Hadoop-Cluster um Data-Warehouse-Funktionalitäten erweitert. Mit HiveQL wird eine SQL-Sprache zur Abfrage und Verwaltung der Datenbanken bereitgestellt.
- **Mahout**: Mahout ist eine skalierbare Machine Learning- und Data-Mining-Library, die aber nicht mehr weiterentwickelt wird.

- **Pig**: Pig ist einerseits eine Hochsprache für Datenfluss-Programmierung, andererseits ein Framework, das die Parallelisierung der Rechenvorgänge unterstützt.
- **Spark**: Spark ist eine performante In-Memory-Batch-Prozess-Engine für Hadoop-Daten. Spark unterstützt ETL-, Machine Learning-, Streaming- und Graphenprozesse.
- **Tez**: Apache Tez ist ein allgemeines Datenfluss-Programmier-Framework. Die ursprünglich von Hortonworks entwickelte Anwendung unterstützt Directed Acyclic Graph (DAG). Tez baut auf YARN auf und wird auch durch YARN gesteuert. Tez kann jeden MapReduce-Job ohne Modifikationen ausführen. MapReduce-Jobs können in einen Tez-Job überführt werden, was die Leistung steigert.
- **ZooKeeper**: ZooKeeper ist ein performantes System zur Koordination und Konfiguration verteilter Systeme.

Aus der Aufzählung und kurzen Beschreibung der Hadoop-Komponenten wird deutlich, dass es sich bei Hadoop nicht um ein einfaches Datenmanagement-Tool handelt. Es ist vielmehr ein komplexes und sich dynamisch veränderndes Sammelsurium an Projekten und Softwareprodukten, die der Idee der verteilten Datenhaltung von Big Data folgen.

Die unterschiedlichen Hadoop-Komponenten können von der Homepage der Apache Foundation kostenlos heruntergeladen werden. Unternehmen greifen aber bei Hadoop auf die Dienstleistungen kommerzieller Hadoop-Distributoren zurück. Diese bieten vorgefertigte Pakete mit z. T. zusätzlichen Komponenten an. In der Regel fallen keine Lizenzkosten an, es wird aber eine Subscription-Fee verlangt, also eine Mietgebühr für die Wartung, Pflege und den Support der Software. Wichtige Anbieter sind:

- Cloudera (Fusioniert mit Hortonworks im Q1 2019)
- MapR
- IBM
- Pivotal

Daneben haben die Cloud-Anbieter eigene Hadoop-Angebote:

- Amazon Web Services EMW
- Microsoft Azures HDInsight

Cloudera hat mit der Data-Platform ein Angebot, das über die Distribution der Apache Hadoop/Spark-Komponenten hinausgeht (Data-Hub) und diese mit weiteren Komponenten zu einer integrierten Datenplattform bündelt. Die Installation kann on-premises, in der Cloud oder hybrid erfolgen.

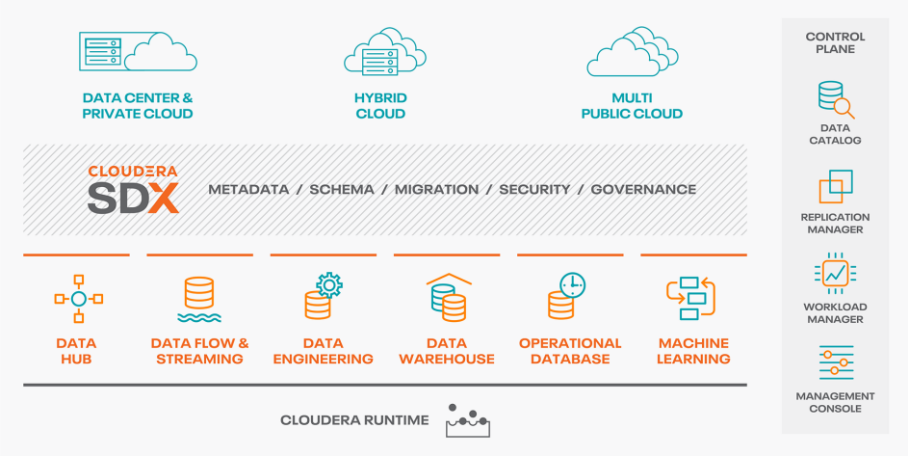


Abbildung 2: Cloudera-Data-Platform

Im vorangegangenen Abschnitt zum Thema Data-Lake wurde schon auf die Herausforderung eingegangen: Für die exponentiell wachsende Datenmenge an strukturierten und unstrukturierten Daten soll eine integrierte technische Plattform bereitgestellt werden. Sowohl strukturierte (HBase) als auch weniger strukturierte Daten (HDFS, Cassandra) können mit extremer Skalierbarkeit in einem gemeinsamen Framework verwaltet und bereitgestellt werden. Daher ist Hadoop die ideale technische Plattform, um eine integrierte Datenstrategie des Unternehmens umzusetzen.

Das Thema hat in den letzten Jahren einen Hype erfahren, wobei einige bereits das Ende klassischer Data-Warehouse-Produkte vorhersagten. Die Gründe zu diskutieren, warum das schnelle Ende klassischer Data-Warehouse-Systeme nicht bevorsteht, würde den Rahmen dieses Buches überschreiten. Etwas vereinfacht kann das Thema aber wie folgt zusammengefasst werden:

- Aufgrund der Komplexität und Dynamik der Hadoop-Projekte ist eine Hadoop-Installation alles andere als kostenlos. Auch wenn keine Lizenzkosten anfallen, sind z. B. Wartung, Hardware, Personal und Schulung mit hohem Aufwand verbunden.
- Die Installationen können als laufende Projekte gesehen werden, da die Hadoop-Komponenten einem dauernden Wandel unterzogen sind. Was heute angesagt ist, kann morgen schon wieder als veraltet gelten.
- Die Performance für bestimmte Arten von Queries in Data-Warehouse-Systemen ist mit Hadoop-Komponenten nicht erreichbar. Den kommerziellen Data-Warehouse-Produkten liegen Hunderte von Entwicklerjahren zugrunde, die für die Optimierung der verteilten Speicherung und Abfrage von Daten verwendet wurden. Unternehmen müssen genau definieren, wo und wie sie ihre Daten speichern wollen. Businesskritische, strukturierte Daten sind wahrscheinlich in einem Data-Warehouse am besten aufgehoben. Hingegen können große Mengen von unstrukturierten Facebook- und Twitter-Daten, bei denen noch gar nicht klar ist, wie sie verwendet werden sollen, in einem *Hadoop-Lake* abgelegt werden. Es zeigt sich, dass für Unternehmen eine umfassende Datenstrategie notwendig ist.

Ein Brückenschlag zwischen dem ‘unstruktierten’ Data Lake- und dem ‘strukturiertem’ Data-Warehouse-Ansatz kommt z.B. vom Anbieter **Databricks**, der sein Konzept unter dem Begriff **Data Lakehouse** vermarktet**.**

Ein Data Lakehouse ist ein Datenarchitektur-Konzept, das Elemente eines Data Lakes und eines Data Warehous vereint. Es integriert die Flexibilität und

Skalierbarkeit eines Data Lakes mit der Struktur und Leistung eines Data Warehous. Daten werden zuerst im Rohformat im Data Lake gespeichert und dann in einem Zwischen-Layer strukturiert und optimiert, um sie in einem Data Warehouse ähnlichen Format zu organisieren. Abfragen auf die dann strukturierten Daten können dann in einer strukturierten Sprache – z.B. Hive SQL erfolgen.

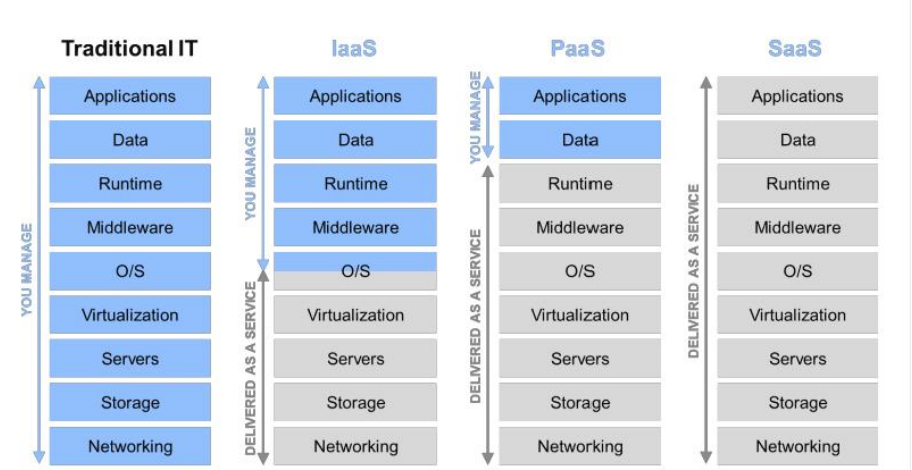
2.6 Cloud-Computing

In den vorangegangenen Abschnitten wurden die unterschiedlichen Datenquellen vorgestellt. Beim Cloud-Computing handelt es sich nicht um einen eigenen Typ von Datenquelle, sondern um die Art der Bereitstellung von Computing-Services. Es ist hier also von einer anderen Ebene die Rede. Die vorher beschriebenen Datenquellen können grundsätzlich On-Premises (also in den eigenen Räumlichkeiten) oder in der Cloud betrieben werden. Dennoch soll in diesem Abschnitt auf das Cloud-Computing eingegangen werden, da dieses derzeit in der betrieblichen Praxis häufig zeitgleich mit Big Data diskutiert wird.

Der unklare Begriff Cloud-Computing wird von der Öffentlichkeit mittlerweile mehr und mehr verstanden und die Diskussionen entsprechend angemessen geführt. Beim Thema Cloud-Computing – also der Bereitstellung von Computing-Leistung über das Internet – lassen sich drei Arten von Angeboten unterscheiden:

- **IaaS (Infrastructure as a Service):** Der Cloud-Dienstleister stellt die Server samt Netzwerk, Speicher, Virtualisierungstechnologie und gegebenenfalls inkl. Betriebssystem zur Verfügung. Der Kunde verwaltet die Anwendungen und die Daten in eigener Regie.
- **PaaS (Platform as a Service):** Hier stellt der Dienstleister zusätzlich Betriebssystem, Middleware und Laufzeitumgebung zur Verfügung, während der Kunde sich nur noch um Anwendungssoftware und Daten kümmert.

- **SaaS (Software as a Service):** Bei SaaS wird die gesamte Anwendung inclusive der Datenhaltung als Service bereitgestellt.



Source: Microsoft.

Darüber hinaus ist zu unterscheiden, wer die Cloud-Lösung betreibt. Bei einer **Public Cloud** wird also die *öffentliche* Infrastruktur des Cloud-Anbieters gemeinsam von den unterschiedlichen Kunden genutzt. Eine **Private Cloud** nutzt die Technologien des Cloud-Computings, aber die Infrastruktur wird exklusiv für einen Kunden zur Verfügung gestellt oder sogar aufgebaut. Die Abgrenzung zu einem traditionellen On-Premises-Betrieb mit flexiblen Virtualisierungs-Technologien ist nicht immer ganz klar und manchmal auch eher marketing- denn technologiegetrieben.

Eine Zwischenform ist die **Hybrid-Cloud**, in der Teile der Computing-Leistung on-premises durchgeführt werden, während die anderen Teile auf die (Public-)Cloud ausgelagert werden.

Die bekanntesten und größten Anbieter von Cloud-Dienstleistungen sind:

- AWS (amazon cloud services)
- Microsoft Azure

- Google
- IBM

Auch die Anbieter von Hadoop-Distributionen – allen voran Cloudera – ergänzen ihre Angebote um Cloud-Dienstleistungen.

Die unterschiedlichen Arten von Datenquellen, wie sie in den vorangegangenen Abschnitten beschrieben worden sind, können auf den Cloud-Plattformen betrieben werden. Egal ob Oracle-Datenbank, MongoDB, Hadoop-Cluster oder Teradata-Data-Warehouse, all diese Produkte – und noch viele mehr – sind auf den Marketplaces der großen Cloud-Anbieter als SaaS-Angebot verfügbar oder können bei einer PaaS- oder IaaS-Cloud-Lösung als *selbst mitgebrachte* Software installiert werden.

Darüber hinaus stellen die Cloud-Anbieter auch eigene Datenbankangebote zur Verfügung:

Zu den AWS-Datenbankservices gehört der Amazon Relational Database Service (Amazon RDS) mit Unterstützung von häufig verwendeten Datenbank-Engines, z. B:

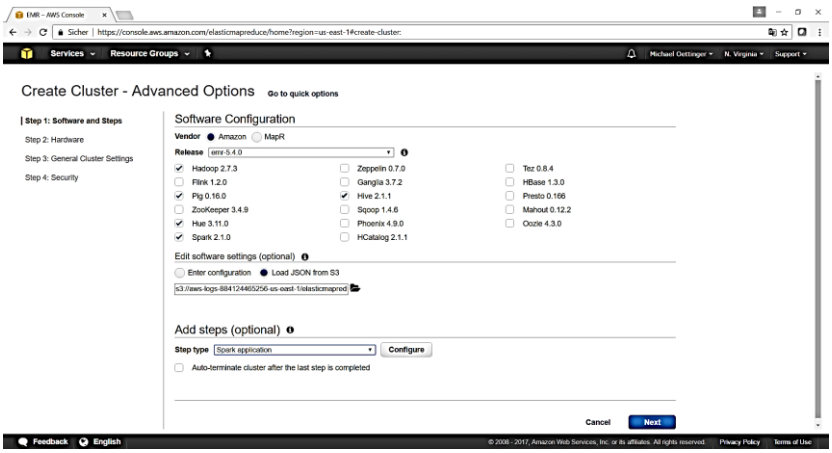
- Amazon Aurora, eine MySQL-kompatible relationale Datenbank,
- Amazon DynamoDB, ein NoSQL-Datenbankservice,
- Amazon Redshift, ein Warehouse-Service,
- Amazon EMR, das ein verwaltetes Hadoop-Framework bietet.

Das entsprechende Microsoft-Azure-Angebot besteht aus den folgenden Elementen:

- Azure SQL-Datenbank
- Document DB, eine NoSQL-Datenbank
- Azure SQL Data-Warehouse
- Microsoft Azure Data Lake Store, eine Hadoop-Anwendung

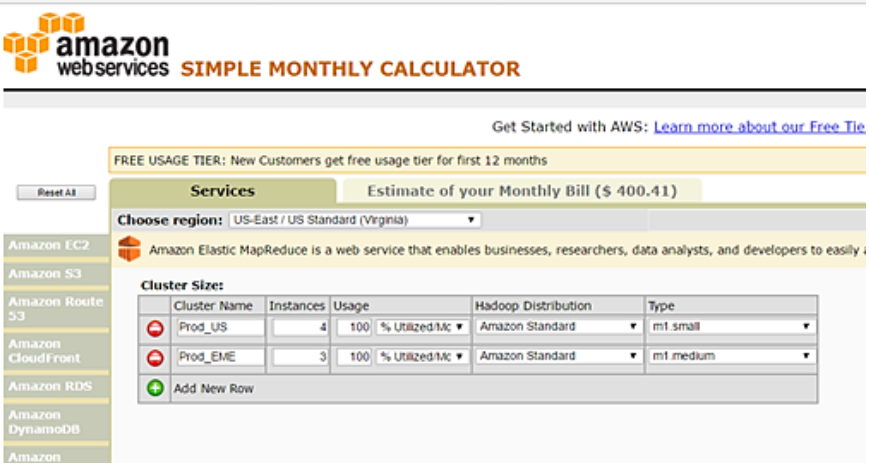
Die Cloud-Anbieter konkurrieren mit den etablierten Datenbanklösungs-Anbietern auf ihren Marktplätzen, was zu ungewohnten Konkurrenz- bzw. Kooperationssituationen führt. Es wird spannend sein, inwieweit es den Cloud-Anbietern (allen voran AWS und Azure) gelingt, daraus Kapital zu schlagen und ihren Marktanteil in den Softwarebereichen auszubauen.

Bei der Einrichtung bspw. eines Hadoop-Clusters werden die Vorteile der Cloud-Technologie deutlich. Ohne sich Gedanken über Hardware, deren Konfigurationen und Betrieb, Kompatibilitäten von Software-Komponenten, Skalierung und Nutzungsprognosen etc. zu machen, kann mit wenigen Mausklicks ein Hadoop-Cluster eingerichtet werden.



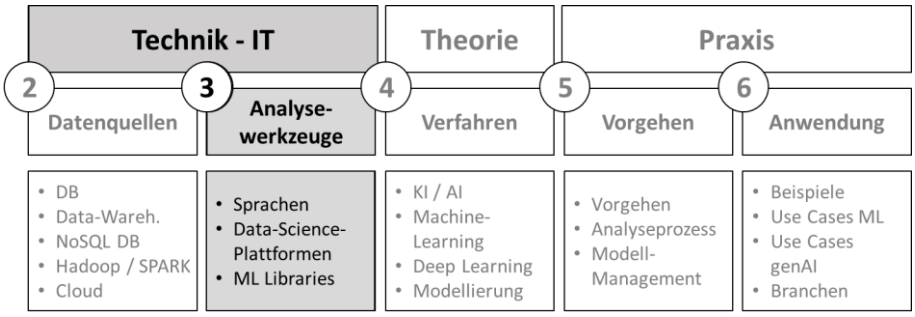
Das Einrichten des Hadoop-Clusters stellt sich dabei zunächst in etwa so schwierig dar wie eine Pizzabestellung auf lieferando.de. Aus einem Menü an Optionen lässt sich ein individuelles Cluster konfigurieren.

Die monatlichen Kosten können dabei im Voraus geschätzt werden.



Die entsprechenden Software-Komponenten werden damit tatsächlich zum Service, der nach Bedarf eingerichtet, skaliert, betrieben und auch *retired* werden kann.

3 Datenanalyse



Im vorangegangenen Kapitel ging es um die Datenquellen, d. h. die Softwarekomponenten, in denen die zu analysierenden Daten gespeichert werden. In diesem Kapitel stehen nun die Komponenten im Fokus, mit denen die Analysen durchgeführt werden können. Sie sind die wichtigsten Werkzeuge für die Arbeit der Data-Scientists. Die Softwareprodukte und -services lassen sich unterscheiden in:

- Programmiersprachen,
- Data-Science-Plattformen,
- Machine Learning-Bibliotheken bzw. -Librarys sowie
- Machine Learning-Angebote der Cloud-Anbieter.

3.1 Programmiersprachen

Im Jahr 2018 hat Python Java als beliebteste Programmiersprache überholt. Das ist praktisch für Data-Scientists, denn Python ist die meistgenutzte Sprache für Machine Learning und die bedeutendste Schnittstelle zu allen relevanten Librarys. Alle anderen Sprachen kann man getrost vergessen!

Rank	Language	Share
1	**Python**	**28,1 %**
2	Java	15,8 %
3	JavaScript	8,9 %
4	C/C++	6,8 %
5	C#	6,6 %
6	PHP	4,6 %
7	**R**	**4,5 %**
8	TypeScript	2,8 %
9	Swift	2,8 %
10	Objective-C	2,3 %
11	**Rust**	**2,1 %**
12	Go	2,0 %
13	Kotlin	1,8 %
14	**Matlab**	**1,7 %**
15	Ada	1,0 %
16	Ruby	1,0 %
17	Dart	1,0 %
18	Powershell	0,9 %
19	VBA	0,9 %
20	Lua	0,7 %
21	Abap	0,6 %
22	**Scala**	**0,6 %**

Abbildung 3: PYPL PopularitY of Programming Language Dez. 2023.

Kurz vor der Rente stehende ‘Data-Miners‘ werden hier widersprechen und irgendwas von **R** faseln. Das Thema kann man aber aussitzen, es sei denn, im Unternehmen existiert noch ein Schrank voller Disketten mit geschäftskritischem R-Code. Dann sollte man den schleunigst in Python überführen, solange noch einer der Angestellten R-Kenntnisse aufweist.

Den historischen Verdienst von R als herstellerunabhängige Sprache für Machine Learning möchte ich damit nicht verleugnen, aber in die Zukunft gerichtet ist R der Programmiersprache Python bei Weitem unterlegen und wird daher noch an Bedeutung verlieren.

Rust wird derzeit (u. a. von Elon Musk) gehypt, aber ich wage die Prognose, dass es – ähnlich wie **Scala** zuvor – in einer Techie-Nische verbleiben und im Machine Learning-Mainstream nie hohe Bedeutung erreichen wird. **Matlab** ist eine herstellerexklusive Sprache und wird im Beliebtheitsranking mit Sicherheit nicht signifikant aufsteigen.

Nicht im PYPL-Ranking enthalten ist die Standarddatenbanksprache **SQL**. In der praktischen Arbeit in der Datenanalyse ist SQL aber wahrscheinlich sogar wichtiger als Python. Etwa 95 % des Zeitaufwandes in analytischen Aufgaben entstehen durch die Vor- und Aufbereitung der Daten, die meist SQL-Querys erfordert.

Fazit: Als Data-Scientist muss man SQL und Python können. Basta!

Im Folgenden soll übersichtsartig auf Python, R und SQL eingegangen werden.

3.1.1 Python

Python ist eine allgemeine höhere Programmiersprache, die seit Anfang der 1990er Jahre entwickelt wird. Das Ziel dabei war es, eine einfach zu erlernende, klar strukturierte, funktionale und objektorientierte Programmiersprache zu schaffen. Python ist weitgehend plattformunabhängig und als Open-Source-Produkt kostenlos verfügbar.

Die Strukturierung des Quellcodes erfolgt durch Einrückungen und nicht durch Klammern, wodurch eine Übersichtlichkeit des Programmcodes gegeben ist. Es handelt sich um eine Interpreter-Sprache, sodass einerseits die Entwicklung von Programmcode schneller abläuft (da kein Compiling notwendig

ist), die Performance aber andererseits etwas geringer ausfällt als bei Compiler-Sprachen bzw. nur über Umwege verbessert werden kann.

Die Sprache kommt mit relativ wenigen Schlüsselwörtern aus und die Syntax ist vergleichsweise minimalistisch auf Übersichtlichkeit hin optimiert. Dadurch lassen sich Python-Programme oft knapper formulieren als in anderen Sprachen. Von der Idee her ist Python eine kompakte Sprache, die durch Bibliotheken übersichtlich aber funktional praktisch unbegrenzt erweitert werden kann.

Genau darin liegt ein maßgeblicher Grund, warum Python eine so große Bedeutung im Rahmen der Data-Science besitzt. Es sind die **Programm- und Funktionsbibliotheken**, die aus Python ein universelles Werkzeug für Data-Scientists machen. Durch die hohe Verbreitung von Python bieten auch viele der Standardsoftwarepakete eine **Schnittstelle zu Python** an. Dadurch lassen sich Python-Programme als Module in Data-Science-Plattformen einbinden. **Apache Spark** nutzt zur Steuerung und Parametrisierung neben seiner *Muttersprache* Scala noch Java und Python. Auch dadurch wird die Bedeutung von Python betont.

Python-Bibliotheken für Data-Science

Es gibt allgemeine Programmbibliotheken mit den grundlegenden mathematischen und datenmanipulierenden Verfahren. Dazu zählen vor allem die Folgenden:

- **NumPy** ist eine Programmbibliothek für die Handhabung von Vektoren, Matrizen oder generell großen multidimensionalen Arrays. Neben den Datenstrukturen bietet NumPy auch Funktionen für numerische Berechnungen.
- **SciPy** ist eine Python-Bibliothek mit mathematischen Werkzeugen wie Algorithmen zur numerischen Integration und Optimierung.

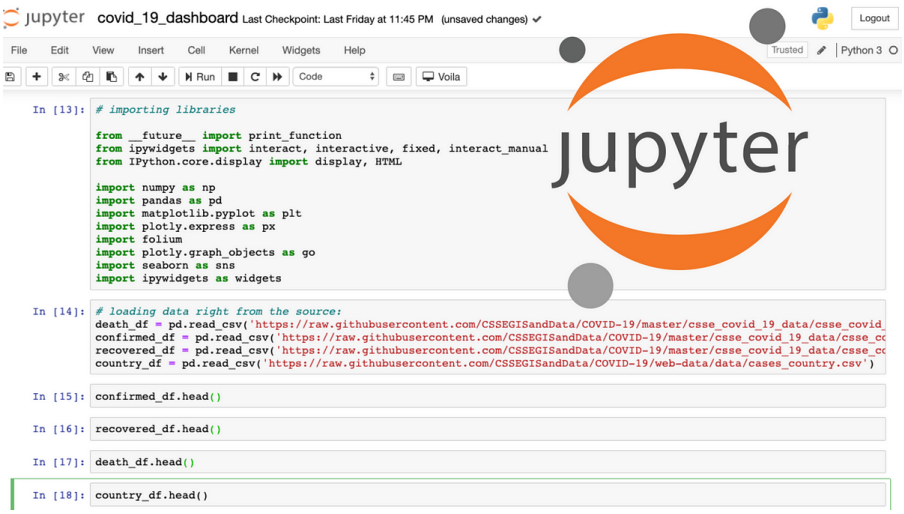
- **Matplotlib** ist eine Plotting-Library, die die grafische Darstellung mathematischer Funktionen erlaubt.

Neben diesen allgemeineren Bibliotheken sind zahlreiche Module speziell für die Bereiche Data-Mining bzw. Machine Learning verfügbar. Oft setzen die Module eine der grundlegenden Bibliotheken voraus. Bedeutend in diesem Bereich sind:

- **Pandas (Python Data Analysis Library)** umfasst Funktionen zur Manipulation und Analyse von Daten.
- **Scikit-learn** umfasst Klassifikations-, Regressions- und Clustering-Algorithmen. Unterstützt werden z. B. Random Forests, Gradient-Boosting, k-Means und DBSCAN.
- **TensorFlow** ist eine aus einem Google-Projekt entstandene Programmbibliothek für Graphenalgorithmen und neuronale Netze (s. Kapitel 3.3.1).
- **Keras** ist eine Open-Source-Bibliothek zum Thema Deep Learning, einerseits eine eigenständige Bibliothek aber auch Teil der TensorFlow-Core-API.
- **PySpark** ist eine Bibliothek und gleichzeitig die Schnittstelle zum Spark-Universum, mit dem die Vorteile des Parallel Computing in Spark genutzt werden können.
- **Mlpy (Machine Learning Python)** umfasst Machine Learning-Algorithmen u. a. für Regression, Klassifikation, Clustering und Dimensionsreduzierung.
- **Statsmodels** bietet grundlegende statistische Funktionen wie Tests, Datenexploration und Gütekriterien.
- **NetworkX** ist ein Softwarepaket für die Erstellung, Manipulation und Analyse komplexer Netzwerkstrukturen und Graphen.
- **PyBrain** unterstützt die Erstellung neuronaler Netzwerke.
- **MDP 3.5** ist eine Bibliothek verbreiteter ML-Algorithmen, die als Module für Pipelines (Prozessketten) bzw. als Knoten eines Netzwerkes genutzt werden können.

- **Theano** und **Caffe** sind Deep-Learning-Bibliotheken für Python.
- **Pattern** ist eine Bibliothek, die Funktionalitäten in den Bereichen Web-Mining, Spracherkennung, Sentiment-Analyse und Machine Learning bietet.
- **Vaex** ermöglicht die Visualisierung, Exploration, Analyse und Machine Learning tabellarischer Datensätze, die nicht von der Größe des Arbeitsspeichers, sondern jener der Festplatte begrenzt sind. Dadurch können also auf einem üblichen Arbeitsplatzrechner Datensätze mit einem Umfang in der Größenordnung mehreren Hundert Gigabyte in vertretbarer Geschwindigkeit verarbeitet werden.[2]

Eine der bedeutendsten Arbeitsumgebungen für Data-Scientists ist das Jupyter-Notebook.



Ein Jupyter-Notebook ist ein versionierbares Dokument, das Eingabezellen (Python-Programmiercode sowie Beschreibungstext) und Ausgabezellen (z. B. Graphen, Diagramme und Ergebnisse) kombiniert und sich daher für das iterative Vorgehen bei analytischen Aufgaben hervorragend eignet.

[2] Jovan Veljanoski, How to analyse 100 GB of data on your laptop with Python

In Kapitel 5.4 sind beispielhafte Python-Snippets aufgeführt und in Kapitel 5.5 wird anhand eines Jupyter-Notebooks ein typisches Machine Learning-Projekt skizziert.

3.1.2 R

R ist eine Programmiersprache für statistische Berechnungen und Grafiken, die sowohl in der Wissenschaft als auch in der Industrie als bedeutendste Standardsprache für statistische Problemstellungen galt. Sie ist Teil des GNU-Projekts und als Open-Source-Lösung für Windows, Mac OS, Linux und Unix erhältlich. Es handelt sich um eine Interpreter-Sprache, die nicht kompiliert werden muss.

Die einfachste Datenstruktur in R ist ein Vektor. Die Elemente von **Vektoren** (eindimensional), **Matrizen** (ein- oder zweidimensional) und **Arrays** (beliebig dimensional) müssen den gleichen Datentyp aufweisen, damit auf diese Rechenoperationen angewendet werden können. Neben diesen homogenen Datenstrukturen werden oft **Data-Frames** verwendet, um Daten als Datensatz darzustellen. Sie sind matrizenförmig, können jedoch aus Spalten unterschiedlicher Datentypen bestehen. Darüber hinaus stehen **Listen** zur Verfügung, in denen Daten beliebiger R-Strukturen und Datentypen enthalten sein können. Objekte verschiedener Datenstrukturen können gemeinsam in der Arbeitsumgebung existieren und gleichzeitig in Analysen verwendet werden.

Der Funktionsumfang wird durch eine **Standardbibliothek** erweitert, die aus 29 Paketen besteht. Darüber hinaus existieren Tausende weitere Packages, die den Funktionsumfang fast beliebig ergänzen. Allein auf CRAN (Comprehensive R Archive Network) waren im Dezember 2023 über 20.200 Packages verfügbar. Eine strukturierte Übersicht über die in den Packages enthaltenen Verfahren findet sich ebenfalls auf CRAN.[3]

[3] https://cran.r-project.org/web/views/MachineLearning.html

Damit ist R die Software mit dem bei Weitem größten Funktionsumfang für Data-Science. Schnittstellen von R zu den gängigen Datenquellen, Programmen und Sprachen sind vorhanden.

Informationen zu den beliebtesten Paketen finden sich auf www.rdocumentation.org/trends oder auf www.r-pkg.org/downloaded.

Der wohl wesentlichste Nachteil von R ist im Bereich der Performance insbesondere für *Big Data* zu sehen. Grundsätzlich werden in R die Daten in den Hauptspeicher geladen, wo auch die Prozesse durchgeführt werden. Daraus ergeben sich Restriktionen der Anzahl der Variablen und Datensätze, die verarbeitet werden können.

Außerdem sind R als Interpreter-Sprache insgesamt und einzelne Algorithmen im Speziellen nicht auf das Maximum an Performance ausgelegt.

3.1.3 SQL

SQL steht für ‘Structured Query Language‘ (strukturierte Abfragesprache) und ist eine Programmiersprache, die für die Verwaltung und Abfrage relationaler Datenbanken entwickelt wurde. Letztere sind eine effiziente Möglichkeit, strukturierte Daten zu organisieren und zu speichern. SQL ermöglicht es, Daten in diesen Datenbanken zu erstellen, abzufragen, zu ändern und zu löschen.

SQL bietet eine standardisierte Möglichkeit, auf Datenbanken zuzugreifen, unabhängig von der spezifischen Datenbanksoftware. Es existieren verschiedene Implementierungen der Sprache, deren Syntax i. d. R. ähnlich ist, während Unterschiede in den unterstützten Funktionen und spezifischen Eigenschaften auftreten können.

Die am häufigsten bei der Auswertung von Daten verwendeten SQL-Anweisungen sind:

- Die grundlegende **SELECT**-Anweisung wird verwendet, um Daten aus einer oder mehreren Tabellen abzurufen.
- **DISTINCT** wird verwendet, um eindeutige Werte in einer Spalte auszuwählen.
- Mit **JOIN** können Zeilen aus mindestens zwei Tabellen basierend auf einer verwandten Spalte kombiniert werden.
- Die **WHERE**-Klausel wird verwendet, um die Ergebnisse basierend auf bestimmten Bedingungen zu filtern.
- Durch **ORDER BY** werden die Ergebnisse nach mindestens einer Spalte aufsteigend oder absteigend sortiert.
- Mit **GROUP BY** können die Ergebnisse basierend auf den Werten in mindestens einer Spalte gruppiert werden.

In Kapitel 5.4 ist eine Liste von SQL-Befehlen aufgeführt, die im Rahmen der Datenbeschaffung für ein Machine Learning-Projekt hilfreich sein können.

3.2 Data-Science-Plattformen

Als Data-Science-Plattformen werden Softwarepakete bezeichnet, die die Prozesse der Datenanalyse und des Machine Learnings unterstützen. Meist enthalten die Plattformen Funktionen:

- zur Extraktion, Aufbereitung und Manipulation von Daten aus unterschiedlichen Quellen und
- bieten eine Bibliothek an Algorithmen bzw. Funktionen für die Datenanalyse.

Es bestehen sowohl kommerzielle Angebote und lizenzkostenfreie Open-Source-Produkte als auch gemischte Angebote (z. B. eine eingeschränkte kostenlose Version und eine lizenzkostenpflichtige *Professional*-Version).

Gartner hat im Jahr 2021 seine Einschätzungen zu den Plattformen in Form eines Magic Quadrant veröffentlicht.

Abbildung 4: www.gartner.com/doc/reprints

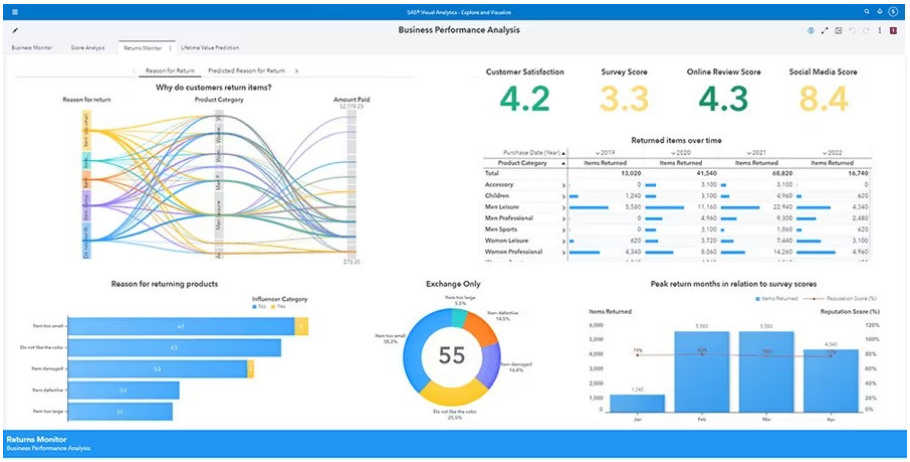
Im Folgenden werden die führenden Data-Science-Plattformen kurz vorgestellt und im Anschluss wird eine – auf den praktischen Erfahrungen beruhende – subjektive Empfehlung des Autors gegeben.

3.2.1 SAS Institute

SAS ist die *Mutter*, wenn nicht sogar die *Großmutter* aller Data-Science-Plattformen. Erste Entwicklungen erfolgten 1966 an der North Carolina State University und führten 1976 zur Gründung des SAS Institute durch Anthony Barr und James Goodnight. Auf den Umsatz bezogen ist SAS klarer Marktführer im Bereich analytischer Data-Science-Plattformen.

SAS bietet über 200 Softwareprodukte bzw. -komponenten an. Kernbereich ist die SAS Software Suite, die um zahlreiche branchen- und funktionsbezogene Module bzw. eigenständige Produkte ergänzt wird.

Als Data-Science-Plattform können der SAS Enterprise Miner und die SAS Visual-Data-Mining-and-Machine Learning(VDMML)-Produktreihe und seit einigen Jahren die viya Plattform genannt werden. Der Enterprise Miner ist das ursprünglichere Produkt, das neben einer grafischen Benutzeroberfläche auch über eine eigene, befehlsorientierte Sprache verfügt, die lange Zeit (vor dem Aufkommen von R) als der Data-Mining-Standard galt. Daher ist die Anzahl der Nutzer mit SAS-Kenntnissen noch groß aber im auf dem absteigenden Ast.



Die Stärken von SAS liegen eindeutig im Funktionsumfang und dem Mind-Share der Produkte, der auf der langen Geschichte des SAS Institute basiert. Die Fähigkeiten des Produkts, nicht nur in der eigentlichen Modellerstellung, sondern auch in den vorbereitenden Aufgaben (Datenextraktion, -bereinigung, -qualitätssicherung, -aufbereitung etc.), setzen den Branchen-Standard. Mit der viya Plattform öffnet sich SAS gegenüber open source Technologien und den aktuellen Herausforderungen aus Cloud- und verteilten Open-Source-Architekturen.

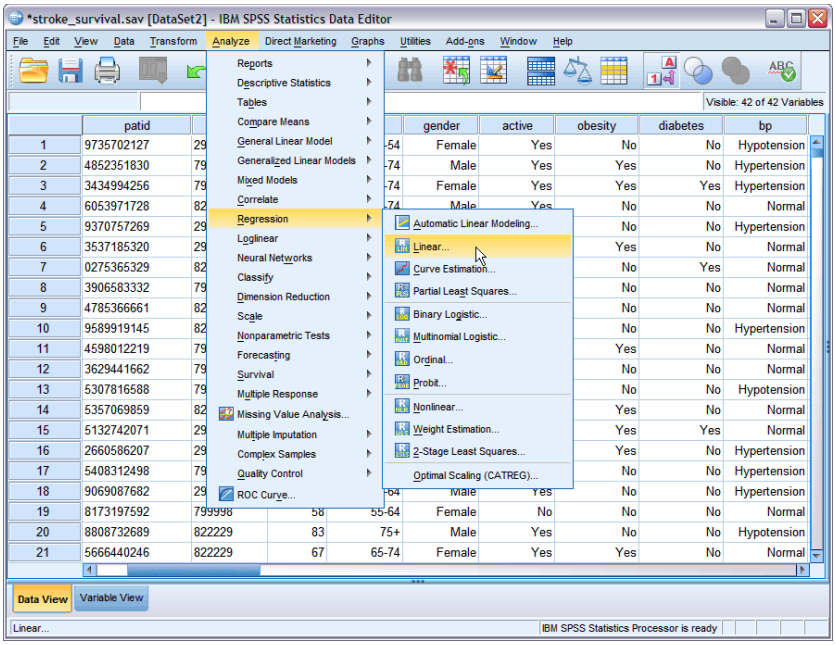
Kritisch werden häufig die Lizenzkosten (v. a. in Hinblick auf die aufkommende Open-Source-Konkurrenz) und die anspruchsvolle Systemverwaltung und -pflege gesehen. Auch sind die Nutzer (und manchmal auch die eigene Vertriebsmannschaft) von der großen Menge an sich teils funktional überschneidenden Produkten schlicht überfordert.

3.2.2 IBM

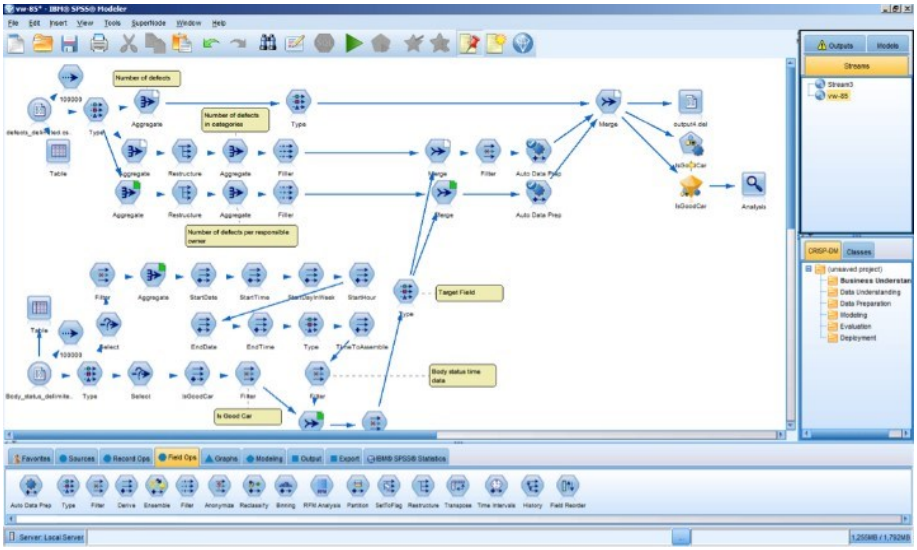
IBM wird als führender Anbieter von Data-Science-Plattformen angesehen, hat aber in den letzten Jahren etwas an Marktanteil verloren. Das Produktportfolio umfasst dabei neben BI-Anwendungen von Cognos vor allem die folgenden Data-Science-Plattformen bzw. analytischen Anwendungen:

- IBM SPSS Statistics
- IBM SPSS Modeler
- IBM Watson Studio
- IBM Watson Discovery

IBM SPSS Statistics ist das Urgestein der Analytics-Anwendungen und existiert seit 1968. Das *Look and Feel* erinnert auf den ersten Blick an Excel. Ausgangspunkt ist eine Datentabelle mit den zu analysierenden Daten, auf die die Verfahren angewendet werden können. Der Umfang der enthaltenen Verfahren ist groß, wobei diese in unterschiedlichen Bundles angeboten werden. Der Ursprung der Anwendung liegt in der sozialwissenschaftlichen Forschung, bei der Daten aus Befragungen ausgewertet wurden. Mittlerweile wird das Produkt jedoch auch im unternehmerischen Umfeld eingesetzt. SPSS läuft als Einzelplatzversion auf unterschiedlichen Betriebssystemen und bietet auch eine Client-Server-Version, die für größere Datenmengen geeignet ist und die Performance erhöht. Der Datenanalyse-Prozess kann über ein Menü oder über eine Scripting-Sprache gesteuert werden. Eine Schnittstelle zu R existiert.



IBM SPSS Modeler (früher SPSS Clementine) ist eine Data-Science-Plattform, mit der der gesamte analytische Prozess in einer grafischen Benutzeroberfläche gesteuert werden kann. Die Oberfläche bzw. Herangehensweise kann als Vorbild für ähnliche Plattformen (z. B. KNIME, Rapid Miner) gesehen werden.



Als Datenquellen können unterschiedliche Quellen (Relationale DB, Hadoop, NoSQL, Flatfiles) angebunden werden. Die Funktionen im Bereich Datenaufbereitung und -qualität sind sehr umfangreich. Die enthaltenen analytischen Verfahren sind ein Ausschnitt aus den SPSS-Statistics-Funktionalitäten, ergänzt um neuronale Netze und bestimmte Entscheidungsbäume.

IBM Watson Studio ist ein Projekt, dessen erste Produktversion 2016 (unter dem Namen Data-Science-Experience) veröffentlicht wurde. Es kann als IBMs Antwort auf die Herausforderung der Open-Source-Aktivitäten rund um Hadoop und Spark gesehen werden. Es handelt sich um eine offene Plattform, die im Freemium-Modell (Teile sind lizenzkostenfrei oder mit Lizenzkosten verbunden) angeboten wird. Sie bietet eine Umgebung, die es Data-Scientisten ermöglicht, unterschiedliche Tools aus Open-Source-Paketen und proprietären Lösungen einzusetzen. Zudem unterstützt die Plattform die Zusammenarbeit bei Analyseprojekten. Ziel ist es, als zentrale Drehscheibe für alle Arten von Data-Science-Werkzeugen zu dienen und den Workflow der Projekte zu unterstützen.

IBM Watson ist ein Oberbegriff für das Angebot im Bereich des kognitiven Computings. Bekanntheit erlangte IBM Watson 2011 durch die erfolgreiche Teilnahme am Quiz Jeopardy, als Watson in der Lage war, Fragen in natürlicher Sprache zu verstehen und die richtige Antwort zu finden. Das Angebot umfasst heute Dienstleistungen, die im Paket mit der Leistung des Watson-Rechenzentrums angeboten werden. Für den Anwender stellt Watson eine Black Box dar, die Ergebnisse liefert, aber wenig bis keine Einfluss- und Einsichtnahme in die zugrundeliegenden Verfahren ermöglicht. Es kann daher nicht als Data-Science-Plattform gesehen werden, sondern ist eher eine KI-Cloud-Lösung.

Andererseits wird mit Watson Analytics ein Softwarepaket bzw. ein softwaregestützter Service angeboten. Watson Analytics ist ein Service für intelligente Datenanalyse und -visualisierung, mit dem Muster und Erkenntnisse in Daten gefunden werden sollen. Zielgruppe ist dabei nicht der *Power-Data-Scientist*, sondern der versierte *Business-User*. Der Service führt durch die Datenermittlung, automatisiert Vorhersageanalysen und bietet kognitive Funktionen, z. B. für einen Dialog in natürlicher Sprache.

Das Gesamtangebot von IBM ist umfangreich, wobei die Zielgruppen und Anwendungsfälle unterschiedlich sind. Marktanalysten, die IBM als Lieferant von Data-Science-Plattformen betrachten, richten also ihr Augenmerk vor allem auf SPSS und auf Watson. Als positiv werden die Marktstärke und die hohe Kundenzahl von IBM gesehen. Außerdem steht mit Watson Studio eine vielversprechende Plattform zur Verfügung. Damit beweist IBM ein starkes Engagement gegenüber der Open-Source-Community.

SPSS wird für seinen Funktionsumfang, die große Erweiterbarkeit, die umfangreichen Anbindungsmöglichkeiten an Datenquellen und seine Fähigkeiten im Bereich Modell-Management gelobt.

Als negativ werden bei IBM oft die Bürokratie und das Pricing bemängelt. Außerdem werden die Marktkommunikation, Produktnamen und unterschiedlichen Angebote als verwirrend und nicht immer nachvollziehbar wahrgenommen.

3.2.3 MathWorks – Matlab

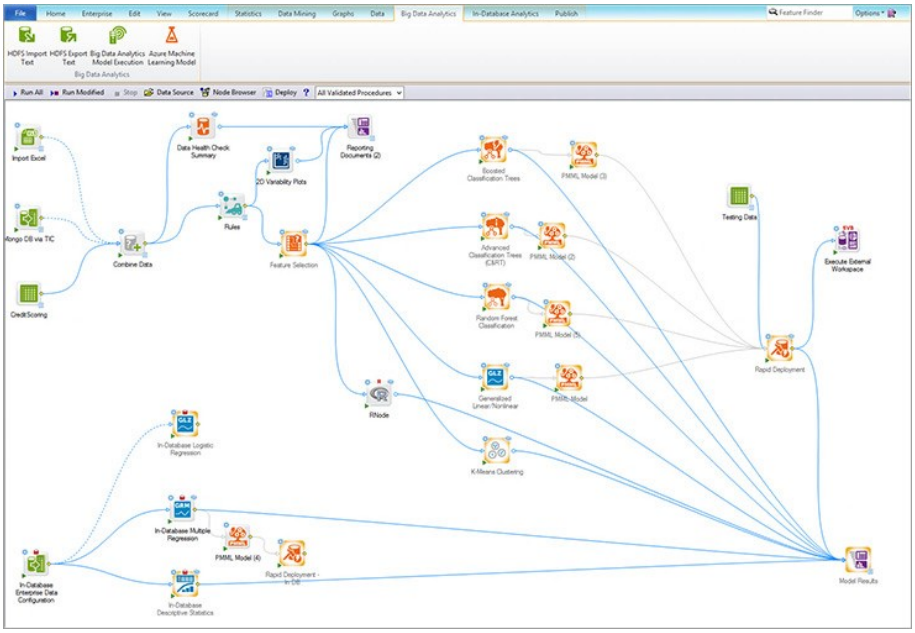
Matlab ist eine kommerzielle Software des US-amerikanischen Unternehmens MathWorks zur Lösung mathematischer Probleme und zur grafischen Darstellung der Ergebnisse. Matlab ist vor allem für numerische Berechnungen mithilfe von Matrizen ausgelegt und verfügt über eigene Programmier- und Scriptsprachen. Die Verbreitung in *ingenieurnahen* Kreisen ist sehr groß und die Software wird vor allem in den Bereichen Simulation, Datenanalyse, mathematische Lösungsalgorithmen und Prototyping eingesetzt. Die Anzahl der vorhandenen Data-Science-Methoden ist begrenzt, da der Schwerpunkt eher auf den obengenannten Bereichen liegt. Die Relevanz von Matlab im Data-Science-Bereich ergibt sich also hauptsächlich aus der großen Verbreitung und Bekanntheit des Produktes und der Programmiersprache aus dem Ingenieurbereich heraus sowie der großen Anzahl an Simulations-, Signalverarbeitungs-, Kontroll- und Machine Learning-Optimierungsverfahren.

3.2.4 TIPCO – Statistica

Statistica ist eine Software für statistische und grafische Datenanalyse, die seit Mitte der 80er Jahre von StatSoft entwickelt wird und damit – nach SAS und SPSS – zu den Pionieren im Bereich Datenanalyse gehört. Es bedarf schon eines gewissen Spürsinns, um zu verstehen, wer aktuell der Eigentümer der Software ist. StatSoft wurde 2014 von Dell aufgekauft. Schon 2012 hatte Dell die Softwarefirma Quest übernommen und die Softwareaktivitäten im Bereich Dell Software zusammengefasst. 2016 erfolgte der Verkauf von Dell Software an eine Investorengruppe, die den Bereich als Quest Software wiederbelebte.

Dieser Teil wurde mittlerweile an TIBCO weiterveräußert, das sich damit ein umfangreiches BI- und Analytics-Portfolio zusammengekauft hat.

Statistica ist eine Data-Science-Plattform für Windows, die neben ETL, Datenvisualisierung und -aufbereitung auch zahlreiche statistische und analytische Verfahren enthält. Eine grafische Oberfläche unterstützt den Data-Science-Prozess.



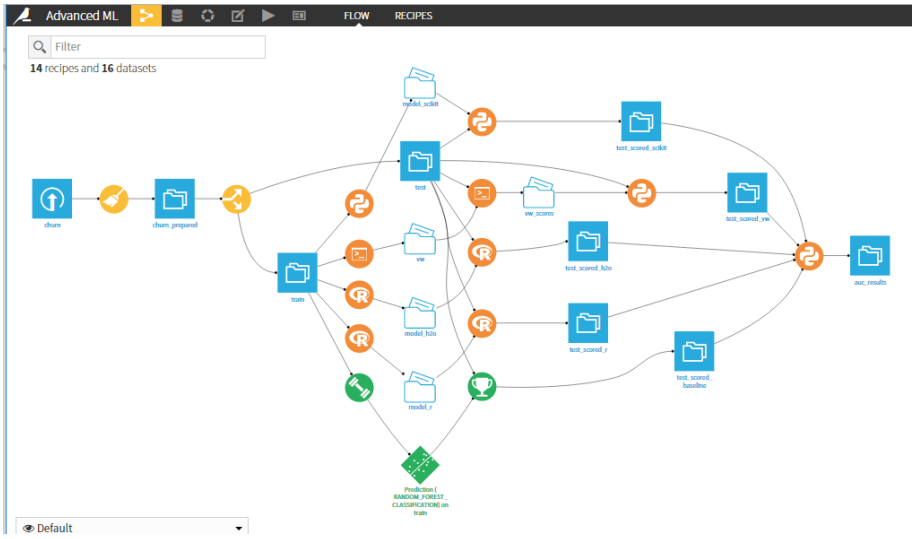
Die Plattform ist flexibel und erweiterbar, z. B. durch Schnittstellen zu R, Python und zu ML-Bibliotheken wie H_20. Ihre Fähigkeiten im Bereich Modell-Management und Deployment von Modellen werden gelobt.

Als kritisch werden z. T. die Erlernbarkeit, Performance und mangelnde Unterstützung von Spark gesehen. Durch den mehrmaligen Managementwechsel in den letzten Jahren und die dadurch hervorgerufene Unsicherheit bezüglich der Weiterentwicklung des Produktes wurde Statistica schon 2016 von Gartner vom ‘Leader’ zum ‘Challenger’ herabgestuft.

3.2.5 Dataiku

Dataiku wurde 2013 in Paris gegründet und hat seinen Hauptsitz mittlerweile in New York. Das Hauptprodukt ist das Dataiku-Data-Science-Studio (DSS), das in einer kostenlosen Ausführung und einer erweiterten kommerziellen Ausgabe (inkl. Multi-User-Zusammenarbeit und Realtime-Scoring) angeboten wird. Das DSS bietet eine flexible und offene Plattform und unterstützt:

- Proprietäre Dataiku-Verfahren
- Machine Learning-Bibliotheken (z. B. H_2O.ai, MLlib)
- Sprach-Plugins (R, Python, Scala)
- Spark



Aufgrund seiner Offenheit und Flexibilität hat Dataiku-DSS schnell eine hohe Aufmerksamkeit unter Data-Scientists erlangt.

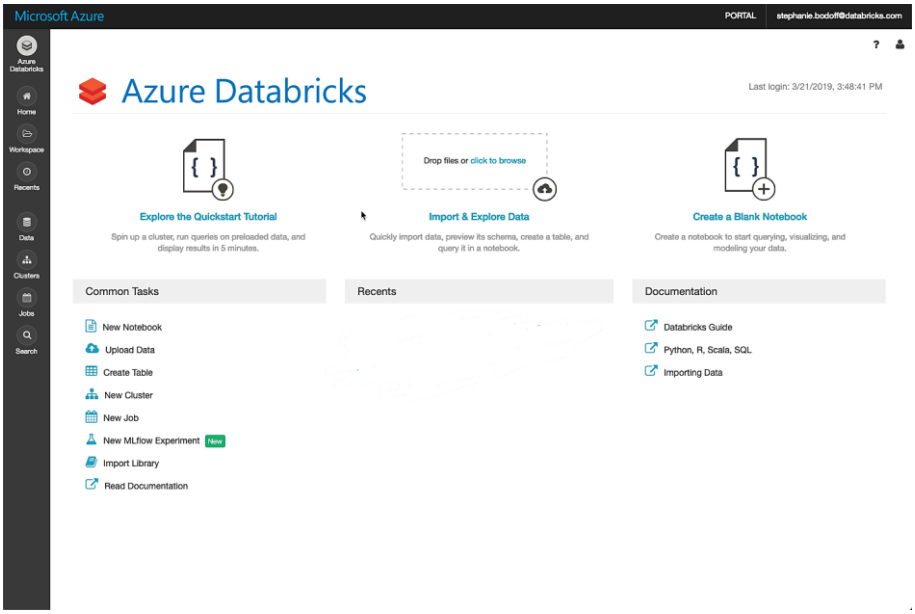
Neben Lob für die Unterstützung der Kollaboration, die Benutzerfreundlichkeit und leichte Erlernbarkeit wurde allerdings auch Kritik bezüglich des geringen funktionalen Umfangs im Bereich Datenzugriff und Datenaufbereitung geäußert. Außerdem sind die für sehr junge Unternehmen üblichen Probleme

(wenige Partnerschaften mit Systemintegratoren, Supportmängel und *Wachstumsschmerzen*) zu erwarten.

3.2.6 Databricks

Databricks ist ein Unternehmen, das von den Entwicklern von Apache Spark gegründet wurde und Kunden bei der cloudbasierten Verarbeitung von Big Data mithilfe von Spark unterstützen soll. Databricks bietet eine webbasierte Plattform für die Arbeit mit Spark, die automatisiertes Cluster-Management und Notebooks im IPython-Stil enthält.

Mit Databricks ist es möglich, Apache-Hadoop- und Spark-Komponenten unter einer Oberfläche zu nutzen. Es unterstützt beim Aufbau eines Hadoop-Clusters, bei der Vorbereitung, bei der Auswertung der Daten und bei der Produktivsetzung der Lösung. Durch die Kooperation mit den zwei führenden Cloud-Anbietern AWS und Azure wird das Management der Infrastruktur in die Plattform miteinbezogen.



Die Architektur ist dreigeteilt:

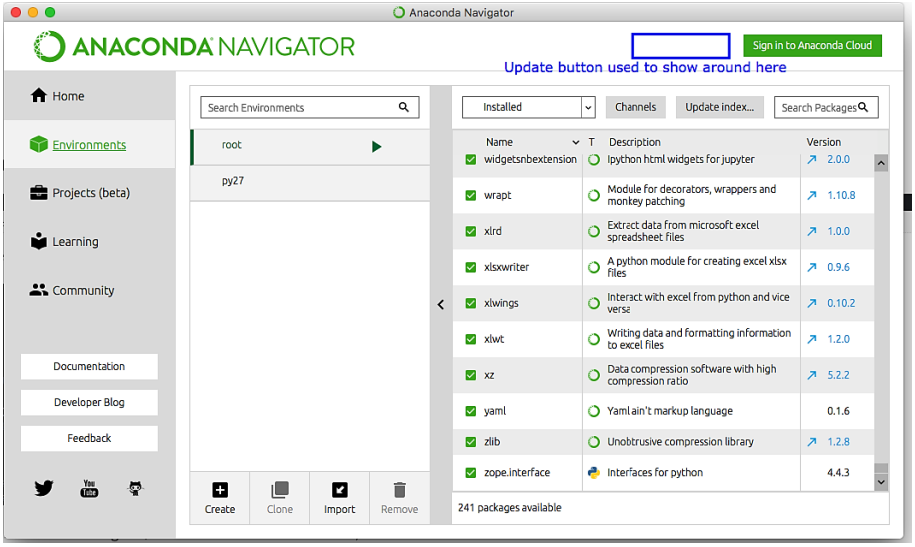
1. Der Workspace – d. h. das Benutzerinterface – nutzt die APIs, die über Python, R, Scala, SQL oder Java aufgerufen werden können.
2. Die Analyseplattform nutzt die Spark-Komponenten und ergänzt diese um die populärsten Machine Learning-Librarys (TensorFlow, Keras, XGBoost, PyTorch, scikit-learn) und weitere Datenkomponenten (mlflow).
3. Die Infrastruktur wird von Azure oder AWS bereitgestellt und per DBU (Databricks-Unit) – d. h. der sekundengenau abgerechneten Prozessorleistung – bepreist.

Kernelement der Philosophie von Databricks ist das Konzept des **Data-Lakehouse**. Dies stellt einen Brückenschlag zwischen dem ‘unstrukturierten‘ Data-Lake- und dem ‘strukturierten‘ Data-Warehouse-Ansatz dar. Ein Data-Lakehouse ist ein Datenarchitektur-Konzept, das Elemente eines Data Lakes und eines Data Warehouse vereint. Es integriert die Flexibilität und Skalierbarkeit eines Data Lakes mit der Struktur und Leistung eines Data Warehouse. Daten werden zuerst im Rohformat im Data Lake gespeichert und dann in einem Zwischen-Layer strukturiert und optimiert, um sie in einem Data Warehouse-ähnlichen Format zu organisieren. Abfragen auf die dann strukturierten Daten können dann in einer strukturierten Sprache – z.B. Hive SQL erfolgen.

3.2.7 Anaconda

Anaconda ist vor allem eine Data-Science-Entwicklungsumgebung für Python-User und weniger eine Data-Science-Plattform wie die anderen Softwarelösungen, die von Gartner in den Quadranten aufgenommen wurden. Anaconda ermöglicht die komfortable Verwaltung von IDEs bzw. Notebooks basierend auf Python (und R) und den dazugehörenden Packages/Libraries. Insbesondere die Unterstützung im Update-Prozess bei den Libraries wird von

den Nutzern geschätzt. Aufgrund des Open-Source-Ansatzes und der mit der Installation verbundenen Arbeitserleichterung beim Download der Libraries konnte Anaconda rasch eine hohe Nutzerzahl gewinnen.

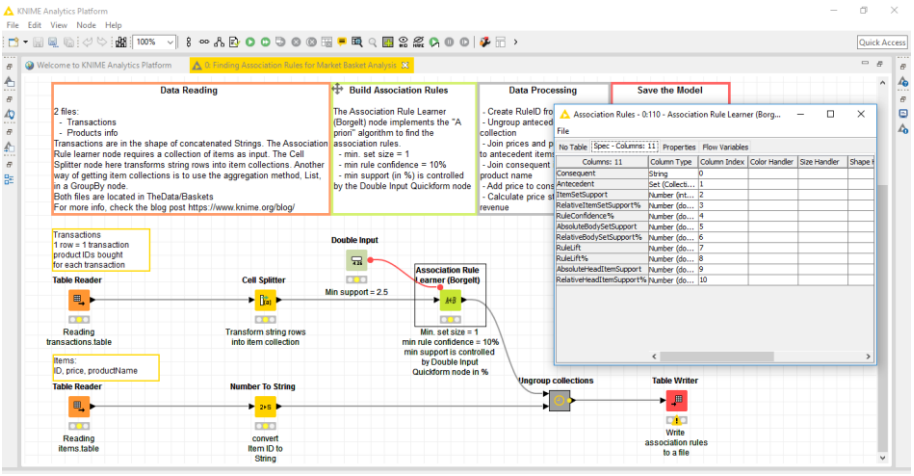


Neben der lizenzkostenfreien Distribution gibt es eine Enterprise-Version, die im Bereich Model-Management, Team-Kooperation und Model-Deployment zusätzliche Funktionen bietet.

3.2.8 KNIME

KNIME, der *Konstanz Information Miner*, ist eine Open-Source-Software-Plattform für die Analyse von Daten. KNIME ermöglicht durch ein modulares Konzept die Integration zahlreicher Verfahren des maschinellen Lernens und des Data-Mining. Die grafische Benutzeroberfläche ermöglicht das einfache und schnelle Aneinandersetzen von Modulen für die Datenvorverarbeitung (ETL: Extraction, Transformation, Loading), Modellierung, Analyse und Visualisierung.

KNIME wurde 2004 von einer Gruppe von Softwareentwicklern aus dem Silicon Valley unter der Leitung von Prof. Dr. Michael Berthold an der Universität Konstanz konzipiert und entwickelt. Mitte 2006 erschien die erste öffentliche Version. Seit Juni 2008 bietet die in Zürich ansässige Firma KNIME.com GmbH Support-Leistungen für das Open-Source-Produkt, lizenzpflichtige Erweiterungen und Beratungsdienstleistungen an.



Die KNIME-Plattform wird in unterschiedlichen Branchen genutzt, die stärkste Verbreitung ist aber in den Bereichen der Fertigungs- und Pharmaindustrie auszumachen.

Die Stärken der Plattform liegen in ihrer Beliebtheit in der Data-Scientist-User-Community. Als Open-Source-Software sind die Eintrittsbarrieren niedriger, was zur Verbreitung der Software beigetragen hat. Die grafische Benutzeroberfläche unterstützt den kompletten Analyseprozess. Dieser wird als Workflow aus einzelnen *Knoten* dargestellt. Die Knoten stellen dabei die einzelnen Analyseverfahren dar, die in der internen Bibliothek bereitgestellt werden und durch zahlreiche *Extensions* ergänzt werden können.

Schwachpunkte von KNIME werden in den Bereichen Modell-Management, Skalierbarkeit und Performance gesehen. Auch die Bereiche der Datenexploration und -visualisierung werden nicht als Stärken von KNIME betrachtet.

3.2.9 RapidMiner

RapidMiner ist eine Data-Science-Software-Plattform, die von einem Unternehmen gleichen Namens entwickelt, vertrieben und supported wird. Der Programmcode ist quelloffen, es gibt eine im Datenumfang beschränkte kostenlose Version (Basic Edition) und Versionen mit Support ohne Datenlimits für den kommerziellen Einsatz.

Das Vorgängerprodukt von RapidMiner mit dem Namen YALE wurde 2001 am Lehrstuhl für künstliche Intelligenz der Technischen Universität Dortmund entwickelt. Mittlerweile hat das Unternehmen RapidMiner seinen Sitz in Boston.

Als Produkte werden das RapidMiner Studio (Einzelplatzversion) und der RapidMiner Server (Server Version) angeboten. RapidMiner unterstützt alle Schritte des Datenanalyse-Prozesses mit mehr als 1.500 Operatoren.

Der RapidMiner verfügt über eine grafische Oberfläche, die den Datenanalyse-Prozess unterstützt und eine Programmierung nicht notwendig macht. Programmierer und fortgeschrittene User können aber eine Scripting-Sprache verwenden.

Die Stärken des RapidMiners liegen

- in der Größe der User-Community,
- im Umfang der Plattform (sowohl was die Anzahl der enthaltenen Verfahren bzw. Modelle als auch die Unterstützung der vorbereitenden Aufgaben im Analyseprozess wie Datenaufbereitung und Qualitätsprüfungen etc. betrifft),
- in der vergleichsweise leichten Erlernbarkeit,
- in der guten Einbindung von Datenquellen (Cloud und on-premises) sowie
- in der Einbindung von Open-Source-Ressourcen und Programmiersprachen wie R, Python oder Weka.

Über eine Ergänzung (RapidMiner Radoop) ist es möglich, die im RapidMiner kreierten Workflows in einen in der Hadoop-Umgebung lauffähigen Code zu übersetzen. Dabei werden Hive-Operatoren und Spark-MLlib-Algorithmen genutzt.

Kritisch angemerkt wird, dass einige Schwächen im Bereich der Dokumentation bestehen und dort nur wenige Beispiele gezeigt werden. Im Gegensatz zur großen User-Community ist das Unternehmen RapidMiner relativ klein und besitzt eine geringe Marktpräsenz (nur ein Büro in den USA und drei in Europa), was die Supportfähigkeit für globale Unternehmen begrenzt.

3.2.10 DataRobot

Bei DataRobot handelt es sich um eine Plattform, mit der das Ziel verfolgt wird, die Arbeit von Data-Scientisten zu *industrialisieren*, also zu automatisieren. Es ist daher keine Data-Science-Plattform, wie sie die in den vorangegangenen Abschnitten vorgestellt wurden, sondern vielmehr eine übergeordnete Schicht, die beim Erstellen von Machine Learning-Modellen unterstützt.

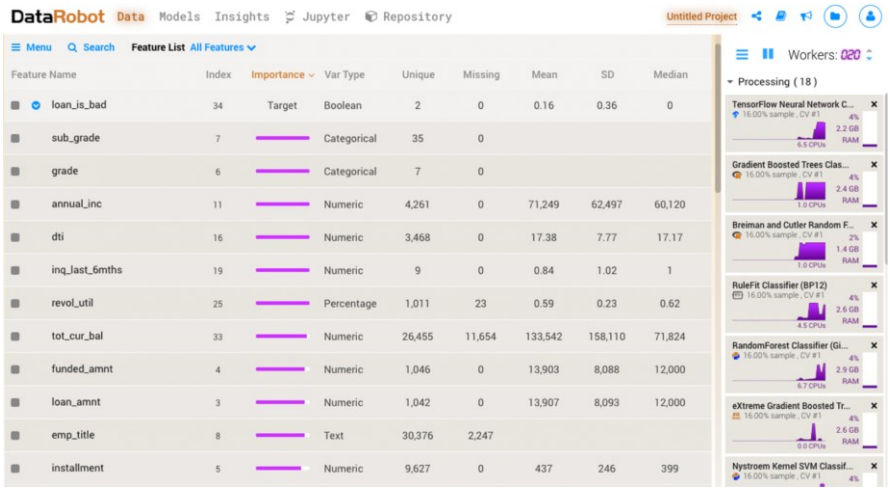


Abbildung 5: Beispiel DataRobot

Insbesondere die Arbeitsschritte

- Auswahl der Zielvariablen,
- Modellbildung mit verschiedenen Modellen und Softwaretools sowie
- Hyperparameter-Tuning

können automatisiert werden und erhöhen dadurch die Produktivität des Data-Scientists. Ebenso werden das Deployment und die Pflege des Modells unterstützt.

3.3 Machine Learning-Bibliotheken

Bei den Machine Learning-Bibliotheken handelt es sich nicht um Data-Science-Plattformen, sondern um Sammlungen von Algorithmen, die Machine Learning-Funktionalitäten bereitstellen und über eine API ausgeführt werden können. Die mit Abstand meistgenutzte API ist bei fast allen Bibliotheken die jeweilige Python-API, was die herausragende Bedeutung von Python weiter unterstreicht.

3.3.1 TensorFlow

TensorFlow ist eine Open-Source-Plattform für Machine Learning und neuronale Netzwerke, die von der Google-Brain-Abteilung entwickelt wurde. Es ermöglicht Entwicklern, komplexe Machine Learning-Modelle zu erstellen und zu trainieren, um Aufgaben wie Bilderkennung, Sprachverarbeitung oder natürliche Sprachverarbeitung zu lösen. Seit November 2023 ist TensorFlow in der Version 2.15 verfügbar.

TensorFlow ist besonders bekannt für seine Unterstützung bei der Entwicklung **neuronaler Netzwerke**. Es bietet Funktionen und Tools für den Aufbau künstlicher neuronaler Netzwerke verschiedener Komplexitätsgrade.

Keras ist seit der TensorFlow-Version 1.10 die bevorzugte High-Level-API für die Definition neuronaler Netzwerke. Sie bietet eine benutzerfreundliche Schnittstelle für das Erstellen, Trainieren und Auswerten neuronaler Netzwerke. Es ermöglicht Entwicklern, Modelle mit wenigen Codezeilen zu erstellen, ohne sich um die komplexen Details der tieferliegenden Ebenen von TensorFlow kümmern zu müssen. Am nachfolgenden Python-Code-Beispiel kann die Einfachheit der Definition eines neuronalen Netzwerkes verdeutlicht werden.

```
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train),(x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10, activation='softmax')])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

TensorFlow unterstützt unterschiedliche Plattformen (Windows, Linux, MacOS und Android) und Prozessortypen (neben CPUs auch GPUs von NVIDIA). Insbesondere die Anwendung auf GPUs mit zahlreichen Prozessorkernen ermöglicht eine Parallelisierung der Prozesse und damit verbunden eine hohe Zeitersparnis, insbesondere beim Training von Modellen. Erst dadurch wird es möglich, Deep-Learning-Anwendungen – d. h. neuronale Netze mit vielen Layers und Knoten – in annehmbarer Zeit zu berechnen.

3.3.2 H_2O.ai

H_2O.ai ist ein 2011 in Mountain View, Kalifornien, gegründetes Unternehmen, das Open-Source-Software für Big-Data-Analysen entwickelt. H_2O bietet Algorithmen aus den Bereichen Statistik, Data-Mining und maschinelles Lernen mit dem Ziel, analytische Algorithmen und Modelle performant auf dem Hadoop Distributed File System auszuführen.

Die wichtigsten Produkte sind:

- **H2O-3** ist ein Open-Source-Framework für maschinelles Lernen, das Funktionen für die Datenanalyse, Modellentwicklung und -bereitstellung bereitstellt. Es unterstützt eine Vielzahl von Algorithmen für Supervised und Unsupervised Learning und ist auf die Verarbeitung von großen Datensätzen ausgelegt.
- **Driverless AI** ist eine automatisierte Plattform für maschinelles Lernen, die von H2O.ai entwickelt wurde. Diese Plattform ermöglicht es Benutzern, komplexe maschinelle Lernmodelle ohne tiefgreifende Kenntnisse der Algorithmen oder Programmierung zu erstellen. Driverless AI automatisiert viele Aspekte des maschinellen Lernens, einschließlich Feature Engineering, Modellauswahl und Hyperparameter-Optimierung.
- **H2O Sparkling Water** ist die Integration von H2O mit Apache Spark.
- **H2O-4GPU**: Diese Version ist darauf ausgerichtet, maschinelles Lernen auf Grafikprozessoren (GPUs) zu beschleunigen, um die Verarbeitungsgeschwindigkeit von Modellen zu erhöhen.

Schnittstellen zu Python, R, Java, Scala, Excel u. a. existieren.

3.3.3 Spark ML

Spark ML (früher als MLlib oder Spark MLlib bekannt) ist eine Komponente des Apache Spark-Frameworks, das für die Verarbeitung von Big Data entwickelt wurde. Spark ML bietet eine skalierbare, verteilte Maschinenlernbibliothek, die darauf abzielt, die Entwicklung von Machine Learning-Modellen auf großen Datensätzen zu erleichtern. Spark ML bietet eine standardisierte Schnittstelle für eine Vielzahl von maschinellen Lernalgorithmen, einschließlich Klassifikation, Regression, Clustering und Dimensionalitätsreduktion.

Spark ML ist darauf ausgelegt, Machine Learning-Modelle auf verteilten Spark-Clustern auszuführen, um die Verarbeitung von Big Data zu ermöglichen.

Beispiele für Algorithmen, die in Spark ML enthalten sind, umfassen lineare Regression, logistische Regression, Entscheidungsbäume, Random Forests, k-Means-Clustering und mehr.

3.3.4 PySpark

PySpark ist die Python-Bibliothek für Apache Spark und ermöglicht Spark-Funktionalitäten von der Python-Programmiersprache aus zu nutzen.

PySpark ermöglicht damit die Verarbeitung von Daten in großem Maßstab auf verteilten Spark-Clustern. Es ist nahtlos in das breitere Apache Spark-Ökosystem integriert, und ermöglicht so die Nutzung der anderen Spark-Komponenten wie Spark SQL, Spark MLlib (Machine Learning), Spark Streaming und Spark GraphX.

3.3.5 MXNet

MXNet ist eine Open-Source-Bibliothek, die Deep-Learning-Modelle wie neuronale Netzwerke mit Faltungscodierung (Convolutional Neural Networks – CNNs) und Lang- oder Kurzzeitgedächtnisnetzwerke (Long Short-Term Memory Networks – LSTMs) unterstützt. Die Software ist skalierbar und erlaubt dadurch eine schnelle Erstellung von Modellen. Es werden unterschiedliche Programmschnittstellen unterstützt (C++, Python, Julia, Matlab, JavaScript, Go, R, Scala).

Das Framework hat seinen Ursprung in der akademischen Welt und ist das Ergebnis der Zusammenarbeit von Wissenschaftlern verschiedener Universi-

täten. Es wurde entwickelt, um v. a. in den Bereichen Bilderkennung, Sprachverarbeitung und -verständnis Lösungen bereitzustellen. Mittlerweile sind zahlreiche Unternehmen an der Weiterentwicklung des Frameworks beteiligt.

Mithilfe von MXNet können Netzwerke in vielen verschiedenen Anwendungsbereichen definiert, trainiert und bereitgestellt werden. Eine Skalierung über mehrere Prozessoren (sowohl CPUs als auch GPUs) und Maschinen hinweg ist möglich. Amazon Web Services hat sich für MXNet als bevorzugtes Deep-Learning-Framework entschieden und bietet die entsprechenden Services an. Auch in Microsoft Azure ist MXNet verfügbar.

3.3.6 Weitere Deep-Learning-Frameworks

Neben den oben aufgeführten Bibliotheken für Deep-Learning existieren noch weitere Software-Lösungen bzw. Bibliotheken, meist als Open Source Software.

Software	Year	Platform	Interface
BigDL	2016	Apache Spark	Scala, Python
Caffe	2013	Linux, macOS, Windows	Python, MATLAB, C++
Chainer	2015	Linux, macOS	Python
Deeplearning4j	2014	Linux, macOS, Windows, Android	Java, Scala, Clojure, Python, Kotlin
Dlib	2002	Cross-Platform	C++
Intel Data Analytics Acceleration Library	2015	Linux, macOS, Windows	C++, Python, Java
Keras	2015	Linux, macOS, Windows	Python, R
Microsoft Cognitive Toolkit (CNTK)	2016	Windows, Linux	Python (Keras), C++, Command line, BrainScript
Apache MXNet	2015	Linux, macOS, Windows, AWS, Android, iOS, JavaScript	C++, Python, Julia, Matlab, JavaScript, Go, R, Scala, Perl
OpenNN	2003	Cross-platform	C++
PyTorch	2016	Linux, macOS, Windows	Python, C++
Apache SINGA	2015	Linux, macOS, Windows	Python, C++, Java
Theano	2007	Cross-platform	Python (Keras)

Table 1: https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software

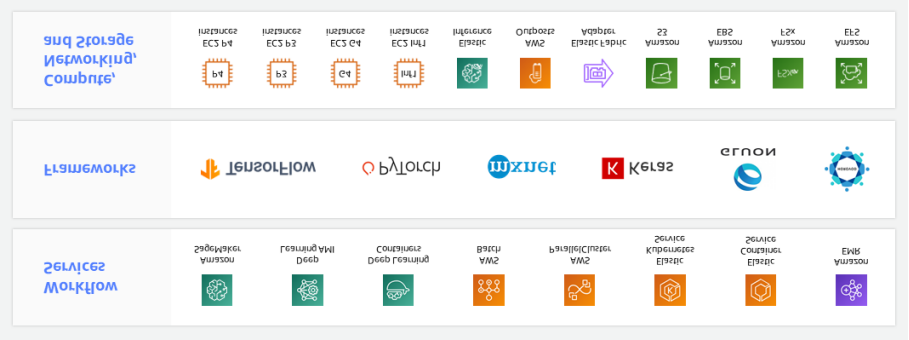
3.4 Cloud-Angebote

In den vorangegangenen Abschnitten wurden Softwarelösungen (Plattformen und Bibliotheken) vorgestellt, die Nutzer herunterladen und auf einem eigenen System betreiben können. Manche der Lösungen sind darüber hinaus auch als Cloud-Angebot verfügbar. Darüber hinaus bieten die Cloud-Anbieter Services für Machine Learning und künstliche Intelligenz an, die *out of the box*

genutzt werden können. Die Angebote bieten in der Regel weniger Parametrisierungs-Möglichkeiten zugunsten einer einfachen Handhabung.

Neben AWS, Microsoft Azure und IBM stellen auch z. B. Google oder Oracle Machine Learning-Angebote aus der Cloud bereit.

3.4.1 Amazon Web Services AWS



Amazon Web Services (AWS) bietet eine breite Palette von Machine Learning-Komponenten und Diensten, die es Entwicklern ermöglichen, maschinelles Lernen und künstliche Intelligenz in ihren Anwendungen zu integrieren. Hier sind einige wichtige Machine Learning-Komponenten von AWS:

- **Amazon SageMaker** ist eine vollständig verwaltete Plattform für das End-to-End-Management von maschinellen Lernmodellen. Es ermöglicht das Trainieren, Bereitstellen und Skalieren von Modellen in der AWS-Cloud.
- **Amazon Machine Images (AMIs)** bietet vortrainierten Deep-Learning-Frameworks wie TensorFlow, PyTorch, Apache MXNet und andere. Diese AMIs erleichtern die Einrichtung und Verwendung von Deep-Learning-Plattformen in der Cloud.

- **Amazon Comprehend**: Ein Dienst für natürliche Sprachverarbeitung, der Texte analysiert und Einblicke in Stimmung, Schlüsselphrasen, benannte Entitäten und mehr bietet.
- **Amazon Polly**: Ein Text-to-Speech-Dienst, der Text in natürliche Sprache umwandelt, sodass Anwendungen Sprachausgabe generieren können.
- **Amazon Rekognition**: Ein Dienst für Bild- und Videoverarbeitung, der Objekterkennung, Gesichtserkennung, Texterkennung und weitere Funktionen bietet.
- **Amazon Forecast** ist ein Service für maschinelles Lernen, der auf Zeitreihendaten spezialisiert ist. Er ermöglicht Vorhersagen und Prognosen für zukünftige Ereignisse.
- **Amazon Personalize** ermöglicht die Erstellung von personalisierten Empfehlungen für Benutzer basierend auf ihren Aktivitäten und Präferenzen.
- **Amazon Lex** unterstütz die Entwicklung von Chatbots und Konversationsanwendungen mit automatischer Spracherkennung und natürlicher Sprachverarbeitung.
- **Amazon Augmented AI (A2I)** ist ein Service, der menschliche Überprüfungen in maschinellen Lernmodellen integriert, um die Genauigkeit und Qualität der Modelle zu verbessern.

3.4.2 Microsoft – Azure

Microsoft Azure bietet eine Vielzahl von Diensten und Komponenten im Bereich maschinelles Lernen und künstliche Intelligenz. Wichtigste Komponenten sind:

- Der **Azure Machine Learning Service** ist eine umfassende Plattform für das Entwickeln, Trainieren und Bereitstellen von maschinellen Lernmodellen. Es bietet Tools für Datenwissenschaftler, Entwickler

und Ingenieure, um End-to-End-Workflows für maschinelles Lernen zu erstellen.

- Das **Azure Machine Learning Studio** ist eine webbasierte integrierte Entwicklungsumgebung (IDE) für maschinelles Lernen. Es ermöglicht das Erstellen, Trainieren und Bereitstellen von Modellen ohne Programmierkenntnisse.
- **Azure Cognitive Services**: Diese Dienste bieten vorgefertigte KI-Funktionen, darunter Bilderkennung, Sprachverarbeitung, Übersetzung, Textanalyse und mehr. Sie ermöglichen es Entwicklern, kognitive Funktionen in ihre Anwendungen zu integrieren, ohne tiefgreifende Kenntnisse in maschinellem Lernen zu haben.
- **Azure Databricks** ist eine Analytics-Plattform, die auf Apache Spark basiert. Es bietet integrierte Tools für maschinelles Lernen und fortschrittliche Analyse in einer kollaborativen Umgebung.
- **Azure Cognitive Search** ermöglicht die Integration von Suche in Anwendungen mit maschinellem Lernen, um Inhalte zu verstehen und relevante Informationen zu extrahieren.
- **Azure Computer Vision** ermöglicht die Integration von Bilderkennungsfunktionen in Anwendungen, um Objekte, Szenen und Text in Bildern zu identifizieren.
- **Azure Speech Services** bietet Sprachverarbeitungsfunktionen, darunter Spracherkennung und Sprachsynthese.
- **Azure Text Analytics** ermöglicht die Analyse von Texten, einschließlich Sentimentanalyse, Erkennung von Schlüsselphrasen und benannten Entitäten.

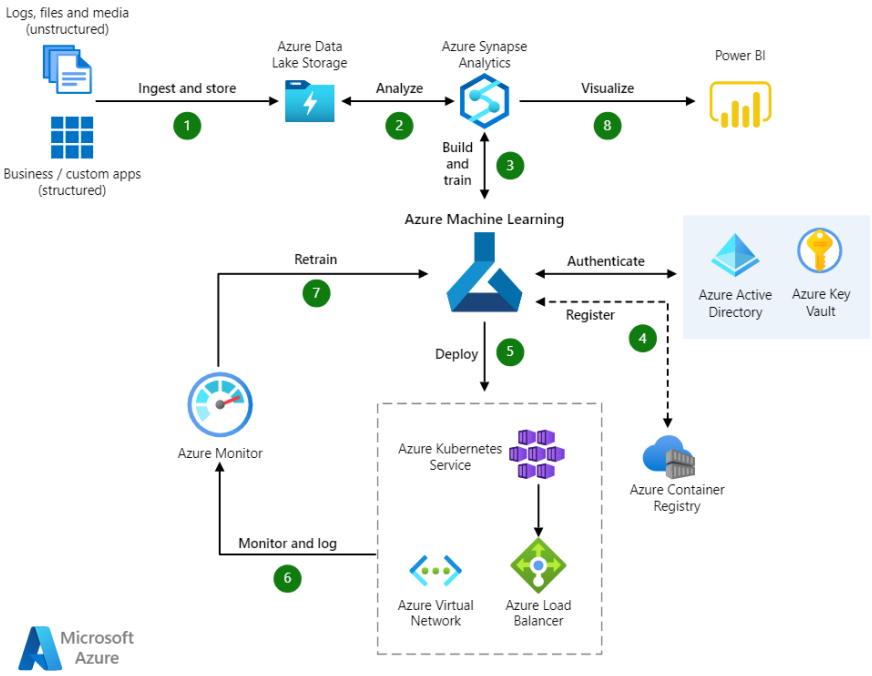


Abbildung 6: Beispielhafte Architektur(Quelle: learn.microsoft.com)

3.4.3 Google

Das Machine Learning-Angebot von Google lässt sich in die folgenden Bereiche unterteilen:

- Die **Google Cloud AI Platform** ist eine umfassende Plattform für das Entwickeln, Trainieren, Bereitstellen und Verwalten von maschinellen Lernmodellen. Sie unterstützt gängige Frameworks wie TensorFlow und scikit-learn.
- **Google Cloud AutoML** ermöglicht es, maschinelle Lernmodelle zu erstellen, ohne tiefgreifende Kenntnisse im Bereich des maschinellen Lernens zu haben. Es gibt spezifische AutoML-Produkte für Vision, Natural Language, Tables und Video.

- **Google Cloud Vision API**: ist ein Dienst für die Bildverarbeitung, der Funktionen wie Bilderkennung, Objekterkennung und Texterkennung bietet.
- Die **Google Cloud Natural Language API** stellt Funktionalität für die natürliche Sprachverarbeitung, der Funktionen wie Sentimentanalyse, Entitätsanalyse und Syntaxanalyse bereit.
- **Google Cloud Translation API** ist der Service für maschinelles Übersetzen von Texten in verschiedene Sprachen.
- **Google Cloud Speech-to-Text** und **Text-to-Speech**: Die Umwandlung von gesprochener Sprache in Text und umgekehrt.
- **Google Cloud Video Intelligence API**: Ein Dienst zur Analyse von Videos, der Funktionen wie Objekterkennung, Aktivitätserkennung und Texterkennung in Videos bietet.
- **Google Cloud AI Building Blocks** ist eine Sammlung von vorgefertigten KI-Bausteinen, die spezifische Aufgaben wie Bildklassifizierung, Objekterkennung, Spracherkennung und mehr erleichtern.

3.5 Entscheidungshilfe für die Softwareauswahl

In den vorangegangenen Abschnitten wurde deutlich, dass es nicht an Angeboten von Softwarelösungen für die Arbeit eines Data-Scientisten mangelt. Im Gegenteil stellt die wachsende Auswahl an Data-Science-Plattformen, Data-Mining-Software und ML-Libraries eine Herausforderung dar. Welche Lösung ist die vielversprechendste? Wie können Nutzer sicherstellen, den zukünftigen Aufgaben gewachsen zu sein?

In den früheren Auflagen dieses Buches folgte an dieser Stelle eine längere und ausgewogene Abwägung der Entscheidungskriterien mit einer ‘vorsichtigen‘ Empfehlung. Mittlerweile ist meine Empfehlung kurz, drastisch und eindeutig:

Databricks!

Werfen sie alle Vertriebler der kommerziellen Softwareanbieter raus. Das führt Sie nur in eine technische Sackgasse und lizenzgebührentechnische Abhängigkeit. Die Softwarelösungen der hier genannten Hersteller sind allesamt gut und brauchbar, aber sie sind meist Opfer ihrer eigenen 'Legacy'. Proprietäre Verfahren der Analyse und Datenhaltung leiden unter ihrer technologischen Vergangenheit. Und die Öffnung gegenüber Open Source Komponenten manifestiert eigentlich nur das Eingeständnis dieser Unterlegenheit.

Vereinfacht gesprochen bestehen alle hier besprochenen Plattformen aus zwei Komponenten.

- Komponente 1: Datenhaltung
- Komponente 2: Datenanalyse und Ergebnis-Präsentation.

Bei der Datenhaltung ist Spark allen proprietären Technologien, was Skalierbarkeit und Parallelisierungsmöglichkeiten überlegen – und das mit Open Source Technologie.

Bei der Analyse-Komponente gibt es ebenso keinen Grund auf ein proprietäres System zu setzen. Es kann nur Python geben. Python ist eine Machine Learning Sprache und gleichzeitig produktive Programmiersprache in einem. 99% der *ernsthaften* Data Scientisten nutzen Python.

Der Ansatz von Databricks ist es, die Datenhaltung auf Sparks mit all seinen Vorteilen direkt mit SQL und Python für die Analyse nutzbar zu machen und das innerhalb eines Frameworks, umsetzbar On-Premises oder direkt auf Azure oder AWS.

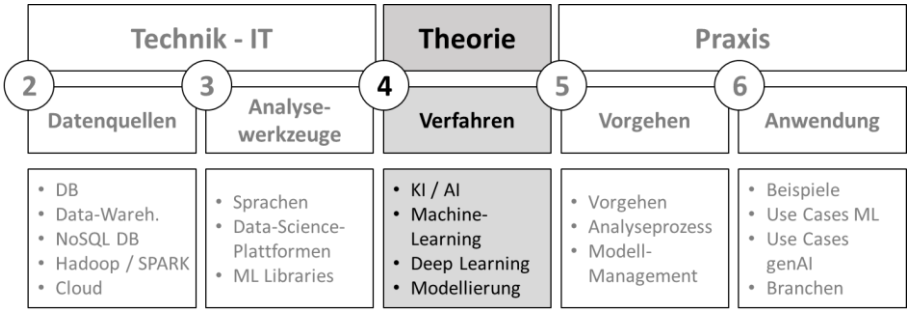
Disclaimer:

Nein, ich arbeite nicht für Databricks und werde für diese Empfehlung hier auch nicht bezahlt. Und ja es stimmt, für Databricks fallen ebenso Lizenzgebühren an und man begibt sich in eine Abhängigkeit von einem kommerziellen Anbieter.

Es ist einfach eine persönliche Empfehlung basierend auf der Erfahrung aus zwanzig Jahren Analytics-Projekten, in denen IBM SPSS, SAS, KNIME, Rapid-Miner, Teradata, Oracle, Microsoft, SAP, Hadoop, Spark, Databricks, Azure ML, AWS ML etc. eingesetzt wurde.

Es handelt sich hier also um eine subjektive Empfehlung, die aber andere geeignete Lösungen nicht ausschließen soll.

4 Verfahren der Datenanalyse



Der Begriff Data-Science enthält den Anspruch, dass es sich bei diesem Fachgebiet um eine *Wissenschaft* handelt. Die in den vorangegangenen Kapiteln beschriebenen Aspekte waren jedoch eher praxisorientiert. Im Fokus stand die Technik, insbesondere die Software, die Data-Scientists bei ihrer Arbeit einsetzen bzw. von denen sie abhängig sind (Datenlieferanten).

Im folgenden Kapitel wird nun der wissenschaftliche Kern betrachtet und die wichtigsten Verfahren im Rahmen der maschinellen Datenanalyse werden vorgestellt. Dabei ist das Ziel, beim Aufbau eines Verständnisses für den ‘Geist‘ der Verfahren zu helfen. Die statistischen und mathematischen Grundlagen und besonders die Feinheiten sowie die Ausprägungen der Methoden können im Rahmen dieses Buches nicht ausführlich behandelt werden. Bevor die Verfahren in Kapitel 4.4 dargestellt werden, erfolgt zuerst deren Einordnung in den Gesamtzusammenhang der künstlichen Intelligenz. Außerdem sollen zentrale Begriffe und Grundlagen erläutert werden.

4.1 Künstliche Intelligenz

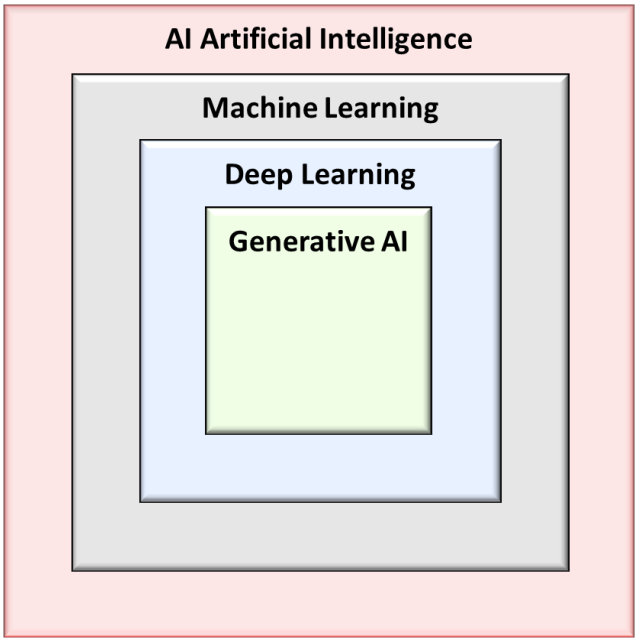
Seit ChatGPT im November 2022 wie aus dem Nichts in den Fokus der Öffentlichkeit gerückt ist, ist der Begriff künstliche Intelligenz bzw. Artificial Intelligence in aller Munde. Die Diskussion nimmt fast schon hysterische Formen an, in der entweder die baldige Weltherrschaft der künstlichen Intelligenz

bzw. der Verlust aller Arbeitsplätze prophezeit wird oder aber sehr lautsprechende ‘Crypto-Bros‘ auf meiner LinkedIn-Timeline erklären, wie ich dank künstlicher innerhalb von einem Monat zum Millionär werde.

Eine sachliche Beschreibung des Konzeptes der künstlichen Intelligenz ist notwendig, um die Chancen und Gefahren fundierter einschätzen zu können und darüber hinaus zu verstehen, wie die Aufgaben eines Data-Scientists und somit der Inhalt dieses Buches in den Gesamtkontext einzuordnen sind.

4.1.1 Abgrenzung künstlicher Intelligenz

Es bestehen unterschiedliche Ansätze, wie die Konzepte zu künstlicher Intelligenz abgegrenzt werden, wobei mir die folgende hierarchische Darstellung am meisten zusagt.



Künstliche Intelligenz (englisch: Artificial Intelligence, AI) ist ein Teilgebiet der Computer-Science bzw. Informatik und ein Oberbegriff, der sich auf die Fähigkeit von Maschinen bezieht, menschenähnliche Aufgaben wie Spracherkennung, Bilderkennung und Entscheidungsfindung auszuführen.

Machine Learning ist ein Teilbereich von künstlicher Intelligenz. Unter Anwendung fortschrittlicher Algorithmen soll aus großen Datenmengen gelernt werden, indem Muster erkannt werden. Das Ziel ist es, Vorhersagen zu treffen und damit Entscheidungen zu unterstützen.

Deep Learning bezieht sich auf einen Ausschnitt der Machine Learning-Verfahren, insbesondere auf die neuronalen Netze mit mehreren Hidden Layers (s. Kapitel 4.4.7) und Graphenverfahren (z. B. Multi-Class Logistic Regression). Es sind Verfahren, die in der Grundidee die Arbeitsweise des menschlichen Gehirns simulieren und besonders in den Bereichen Bild- und Spracherkennung eingesetzt werden.

Generative Artificial Intelligence ist eine Teilmenge von Deep-Learning-Modellen, die dazu verwendet werden, eigenständig Content wie Texte, Bilder, Videos oder Programmiercode anhand eines gegebenen Inputs zu generieren. Sie werden an riesigen Datenmengen trainiert und bedürfen keiner besonderen Instruktion.

Wenn in der Presse von künstlicher Intelligenz die Rede ist, muss somit differenziert werden, ob damit die Gesamtheit der computergestützten Lernmethoden (das ‘Gesamtkästchen‘ in der obigen Abbildung) oder der Einsatz von Generative Artificial Intelligence gemeint ist (das innerste ‘Kästchen‘).

In den folgenden Abschnitten werden die Verfahren der Generative Artificial Intelligence (genAI) erläutert.

4.1.2 ChatGPT und LLMs

Der Auslöser für den Hype um künstliche Intelligenz war die öffentliche Verfügbarkeit von ChatGPT im November 2022. Die Qualität der Antworten auf frei formulierte Fragen bzw. Aufforderungen stellte alles in den Schatten, was vergleichbare Sprachmodelle zuvor leisten konnten.

ChatGPT ist eine von OpenAI entwickelte Chatbot-Software, die auf künstlicher Intelligenz basiert und mit Menschen kommunizieren kann. Sie beruht auf einem Modell für Machine Learning, das Texte verarbeitet und Vorhersagen macht. ChatGPT lernt aus den Gesprächen mit den Nutzern und nutzt bereits gelesene Texte, um Antworten zu generieren.

Um die Grundlagen von ChatGPT zu verstehen, ist der Begriff ‘Large Language-Model‘ (großes Sprachmodell) zu klären. ChatGPT beruht auf einem GPT (Generative Pre-trained Transformer), der die Fähigkeit aufweist, große Mengen an Textdaten zu verstehen und darauf basierend qualitativ hochwertige Texte zu generieren.

Transformer-Modelle ermöglichen es, komplexe Abhängigkeiten in Daten zu modellieren und sind besonders effektiv bei der Verarbeitung langer Sequenzen, wie sie in Texten vorkommen.

Wer die grundsätzliche Arbeitsweise von Transformer-Modellen detaillierter verstehen möchte, sei auf die ausführliche Webseite der Financial Times verwiesen: ig.ft.com/generative-ai.

Abbildung 7: Auszug aus ig.ft.com/generative-ai.

Dort werden die grundlegenden Konzepte der Transformer (wie Token, Embeddings, Vector und Self-Attention) anschaulich erklärt.

Ein entscheidender Aspekt von ChatGPT ist das Konzept des unüberwachten Lernens. Das Modell wird mit riesigen Mengen an Textdaten vortrainiert, ohne dabei spezifische Anweisungen für eine bestimmte Aufgabe zu erhalten. Durch dieses Verfahren kann ChatGPT eine breite Vielfalt von Sprachmustern und Kontexten erlernen.

ChatGPT zeichnet sich durch seine Fähigkeit aus, kontextsensitive Vorhersagen zu treffen. Das bedeutet, dass es nicht nur auf den vorherigen Satz eines Textes reagiert, sondern auf den gesamten bisherigen Kontext. Dadurch kann das Modell präzisere und kohärentere Antworten generieren und Informationen über lange Textabschnitte hinweg berücksichtigen. Somit werden komplexe Zusammenhänge verstanden und darauf basierend adäquate Antworten generiert.

Durch adaptives Lernen kann ChatGPT während der Interaktion mit Benutzern neue Informationen aufnehmen und in den Kontext einbinden. Auf diese Weise ist das Modell flexibel und anpassungsfähig.

Weitere Large Language-Models

ChatGPT ist der prominenteste Vertreter der Transformer-based Large Language-Models, aber nicht der einzige. Der zeitliche Verlauf der Entwicklung und die Verwandtschaft der bedeutenden Modelle sind nachfolgend in einem Stammbaum dargestellt.

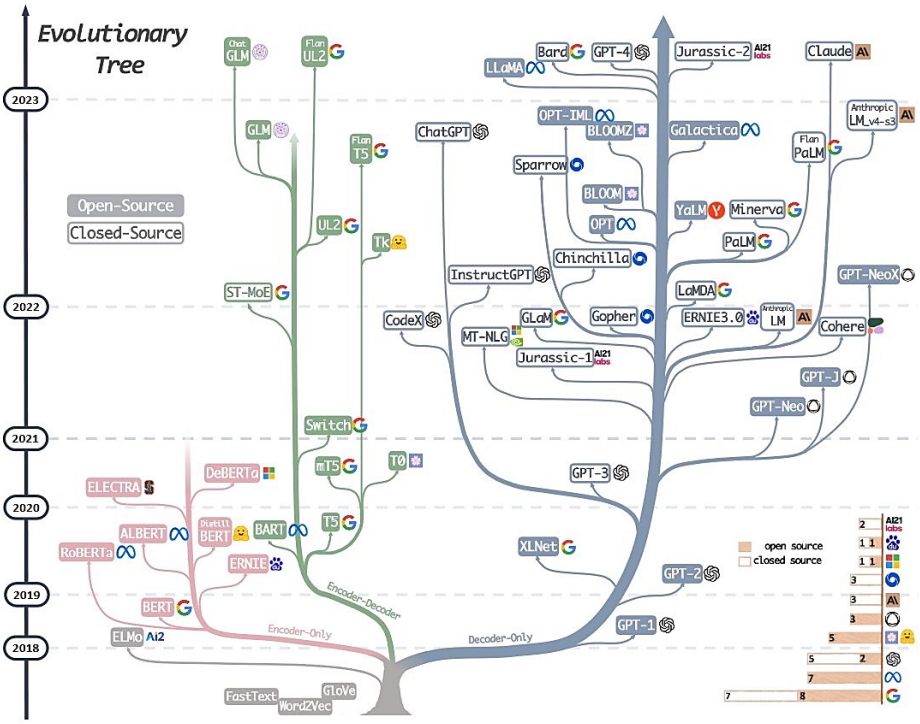


Fig. 1. The evolutionary tree of modern LLMs traces the development of language models in recent years and highlights some of the most well-known models. Models on the same branch have closer relationships. Transformer-based models are shown in non-grey colors: decoder-only models in the blue branch, encoder-only models in the pink branch, and encoder-decoder models in the green branch. The vertical position of the models on the timeline represents their release dates. Open-source models are represented by solid squares, while closed-source models are represented by hollow ones. The stacked bar plot in the bottom right corner shows the number of models from various companies and institutions.

Abbildung 8: https://samim.io/p/2023-04-30-evolutionary-tree-of-llms/.

Die Wurzeln reichen zurück zu früheren Modellen wie GPT-1 und GPT-2 von OpenAI. Diese bildeten den Grundstein für die Entwicklung leistungsstarker Sprachmodelle.

GPT-3, die dritte Iteration der Generative Pre-trained Transformer-Modelle, wurde im Jahr 2020 von OpenAI vorgestellt. Mit 175 Milliarden Parametern ist GPT-3 das bisher größte und leistungsfähigste Sprachmodell. Seine Fähigkeit, komplexe Zusammenhänge zu verstehen und qualitativ hochwertige Texte zu generieren, bildete die Grundlage für die Entwicklung von ChatGPT.

ChatGPT entstand durch Fine-Tuning von GPT-3 für die spezielle Anwendung im Dialog. Die Idee war, ein Sprachmodell zu schaffen, das natürliche und kohärente Gespräche mit Benutzern führen kann.

Die führenden Marktteilnehmer im Bereich Large Language-Models sind neben OpenAI Microsoft (das an OpenAI beteiligt ist), Meta und Google.

Eine Übersicht über die bedeutendsten Large Language-Models findet sich auf Wikipedia. Die Entwicklung ist dynamisch, sodass monatlich neue Versionen oder Modelle veröffentlicht werden.

Name	Release date	Developer	Number of parameters	Notes
GPT-1	June 2018	OpenAI	117 million	First GPT model, decoder-only transformer.
BERT	October 2018	Google	340 million	An early and influential language model, but encoder-only and thus not built to be prompted or generative.
XLNet	June 2019	Google	340 million	An alternative to BERT, designed as encoder-only.
GPT-2	February 2019	OpenAI	1.5 billion	general-purpose model based on transformer architecture
GPT-3	May 2020	OpenAI	175 billion	A fine-tuned variant of GPT-3, termed GPT-3.5, was made available to the public through a web interface called ChatGPT
GPT-Neo	March 2021	EleutherAI	2.7 billion	The first of a series of free GPT-3 alternatives released by EleutherAI.
GPT-J	June 2021	EleutherAI	6 billion	GPT-3-style language model
Megatron-Turing NLG	October 2021	Microsoft and Nvidia	530 billion	Standard architecture but trained on a supercomputing cluster.
Ernie 3.0 Titan	December 2021	Baidu	260 billion	Chinese-language LLM. Ernie Bot is based on this model.
Claude[144]	December 2021	Anthropic	52 billion	Fine-tuned for desirable behavior in conversations.
GLaM (Generalist Language Model)	December 2021	Google	1.2 trillion	Sparse mixture of experts model, making it more expensive to train but cheaper to run inference compared to GPT-3.
Gopher	December 2021	DeepMind	280 billion	Further developed into the Chinchilla model.
LaMDA (Language Models for Dialog Applications)	January 2022	Google	137 billion	Specialized for response generation in conversations.
GPT-NeoX	February 2022	EleutherAI	20 billion	based on the Megatron architecture
Chinchilla	March 2022	DeepMind	70 billion	Reduced-parameter model trained on more data. Used in the Sparrow bot. Often cited for its neural scaling law.
PaLM (Pathways Language Model)	April 22	Google	540 billion	Trained for ~60 days on ~6000 TPU v4 chips.
OPT (Open Pretrained Transformer)	May 2022	Meta	175 billion	GPT-3 architecture with some adaptations from Megatron
YaLM 100B	June 2022	Yandex	100 billion	English-Russian model based on Microsoft's Megatron-LM.
Minerva	June 2022	Google	540 billion	Minerva is based on PaLM model, further trained on mathematical and scientific data.
BLOOM	July 2022	Large collaboration led by Hugging Face	175 billion	Essentially GPT-3 but trained on a multi-lingual corpus (30% English excluding programming languages)
Galactica	November 22	Meta	120 billion	Trained on scientific text and modalities.
AlexaTM (Teacher Models)	November 22	Amazon	20 billion	bidirectional sequence-to-sequence architecture
Neuro-sama	December 2022	Independent	Unknown	A language model designed for live-streaming on Twitch.
LLaMA (Large Language Model Meta AI)	February 2023	Meta	65 billion	Trained on a large 20-language corpus to aim for better performance with fewer parameters.
GPT-4	March 2023	OpenAI	Exact number unknown	Available for ChatGPT Plus users and used in several products.
Cerebras-GPT	March 2023	Cerebras	13 billion	Trained with Chinchilla formula.
Falcon	March 2023	Technology Innovation Institute	40 billion	
BloombergGPT	March 2023	Bloomberg L.P.	50 billion	LLM trained on financial data from proprietary sources.
PanGu-Σ	March 2023	Huawei	1.085 trillion	
OpenAssistant	March 2023	LAION	17 billion	Trained on crowdsourced open data
Jurassic-2	March 2023	AI21 Labs	Exact size unknown	Multilingual
PaLM 2 (Pathways Language Model 2)	May 2023	Google	340 billion	Used in Bard chatbot.
Llama 2	July 2023	Meta	70 billion	Successor of LLaMA.
Claude 2	July 2023	Anthropic	Unknown	Used in Claude chatbot.
Falcon 180B	September 23	Technology Innovation Institute	180 billion	
Mistral 7B	September 23	Mistral AI	7.3 billion	
Claude 2.1	November 23	Anthropic	Unknown	Used in Claude chatbot. Has a context window of 200,000 tokens, or ~500 pages.
Grok-1	November 23	x.AI	Unknown	Used in Grok chatbot. Grok-1 has a context length of 8,192 tokens and has access to X (Twitter).
Gemini	December 2023	Google DeepMind	Unknown	Multimodal model, comes in three sizes. Used in Bard chatbot.
Mixtral 8x7B	December 2023	Mistral AI	46.7B total,	Mixture of experts model, outperforms GPT-3.5 and Llama 2 70B on many benchmarks. All weights were released via torrent.
Phi-2	December 2023	Microsoft	2.7B	So-called small language model, trained on "textbook-quality" data based on the paper "Textbooks Are All You Need".

Abbildung 9: https://en.wikipedia.org/wiki/Large_language_model.

Auf den Punkt gebracht sind ChatGPT und vergleichbare Large Language-Models lediglich *Sprachmodelle*, die in einer bisher nicht gekannten Qualität auf sprachlichen Eingaben basierende sprachliche Ausgaben erzeugen können. Sie basieren auf Methoden, denen ‘zufälligerweise‘ in der Wissenschaft der Begriff *künstliche Intelligenz* zugewiesen wurde. Sie sind aber keine intelligenten Wesen. Sie haben keine künstliche Intelligenz, sie verfügen über kein Bewusstsein und sie sind nicht zu echter Kreativität fähig, die über den vorgegebenen Rahmen hinaus geht.

Hier ein lustiges Beispiel, was passiert, wenn ein Bildredakteur durch ‘künstliche Intelligenz‘ ersetzt wird:

Abbildung 10: https://www.swr3.de/aktuell/schimmel-vermieter-mietwohnung-rechte-108.html.

Cheat-Sheet zu ChatGPT

Nach meiner Meinung besteht somit nicht die Gefahr, dass die '*künstliche Intelligenz*' in Zukunft sämtliche Aufgaben übernehmen, uns Menschen ersetzbar und damit arbeitslos machen wird. Unbestreitbar wird sich durch den Einsatz von Generative Artificial Intelligence die Produktivität in zahlreichen Branchen deutlich erhöhen lassen. Somit werden Menschen, die Tools auf Basis von Generative Artificial Intelligence intelligent nutzen, Personen ersetzen, die diese Tools nicht bedienen können. Es ist daher sinnvoll, sich mit den Eigenheiten der Tools vertraut zu machen. An erster Stelle ist in diesem Kontext ChatGPT zu nennen. Es folgt daher eine Anleitung, wie Fragen bzw. Aufforderungen – sogenannte Prompts – an ChatGPT formuliert werden können.

Grundsätzlich kann man in natürlicher Sprache mit ChatGPT sprechen (d. h. wie meine Mutter schon seit Jahren Google Suchen formuliert). Sowohl die Prompts als auch die Ausgaben können – auch unabhängig voneinander - in zahlreichen Sprachen erfolgen.

Da die Sprachmodelle primär mit englischsprachigen Texten trainiert wurden, wird bei einer Ausgabe in einer anderen Sprache eine Übersetzung integriert. Damit dieser Schritt für die Eingabe nicht notwendig ist, habe ich es mir angewöhnt, die Prompts auf Englisch zu formulieren und als Ausgabesprache ggf. Deutsch zu wählen. Ob dadurch die Ergebnisse tatsächlich besser werden, kann ich nicht beurteilen, aber auf Englisch ist eine größere Anzahl an Beispielen und Tutorials verfügbar.

Bei der Formulierung eines Prompts kann man:

- eine **Rolle** benennen,
- den **Bedarf** formulieren,
- die **Aufgabe** spezifizieren, auch mit **Details,**
- und das **Ausgabeformat** festlegen.
- Man kann Dinge **ausschließen**
- und **Beispiele** für die gewünschte Ergebnisse anfügen.

*Act like a **[ROLE]**, I need a **[NEEDS]**, you will **[TASK]**, in the process, you should **[Details]**, **[DO NOT...]**, present the result as **[FORMAT]**. Here is an example: **[Example].***

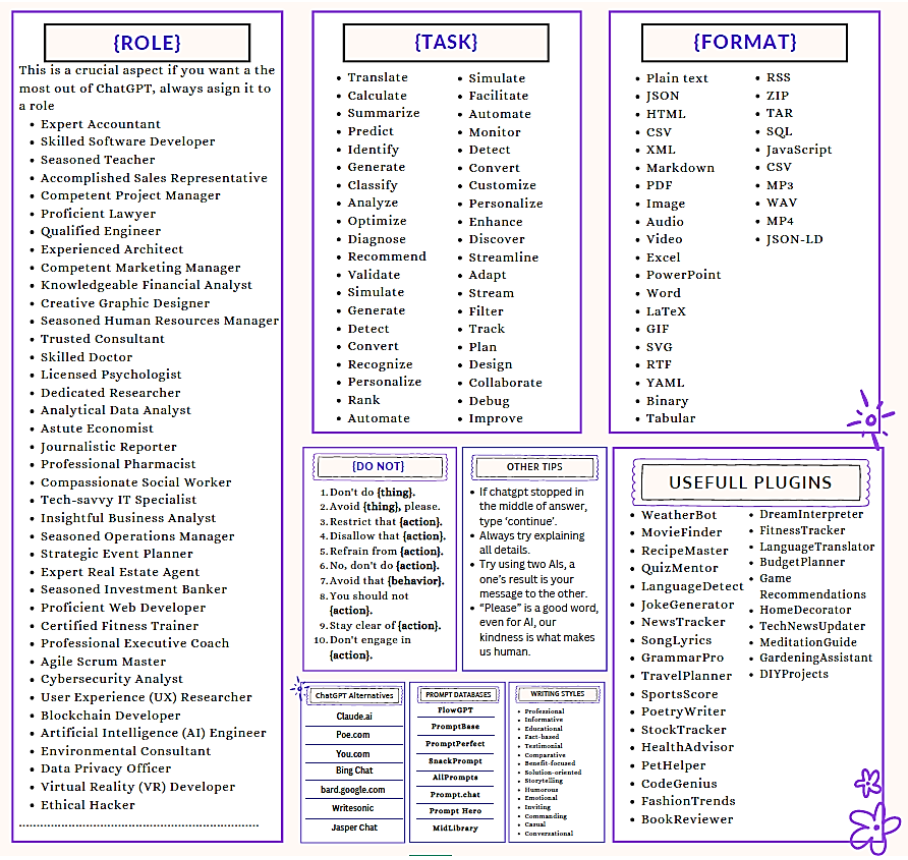


Abbildung 11: gptai.gumroad.com

Hinweise und Beispiele zu ChatGPT-Prompts in deutscher Sprache finden sich z. B. auf blogkurs.de/chatgpt-prompts/.

4.1.3 Weitere GenAI-Anwendungen

ChatGPT ist der prominenteste Vertreter der neuartigen Tools im Bereich der künstlichen Intelligenz, mitnichten jedoch der einzige. Seit der Zugriff auf das OpenAI-GPT-Sprachmodell über eine offene API auch Drittentwicklern zur Verfügung steht, werden täglich neue Tools veröffentlicht. Diese Drittanwendungen nutzen die ‘Text-in-Text-out‘-Funktionalität der OpenAI-API, um einen Mehrwert zu generieren. So schnell wie sie entstehen geraten jedoch auch viele dieser Anwendungen wieder in Vergessenheit, vermutlich da der angestrebte Mehrwert nicht so nachhaltig ist wie versprochen.

Um einen Überblick über die tatsächlich nützlichen Tools zu behalten, ist es sinnvoll, diese nach ihrer grundsätzlichen Aufgabe zu gruppieren. Im Anschluss können die bedeutendsten Vertreter der Klassen beleuchtet werden.

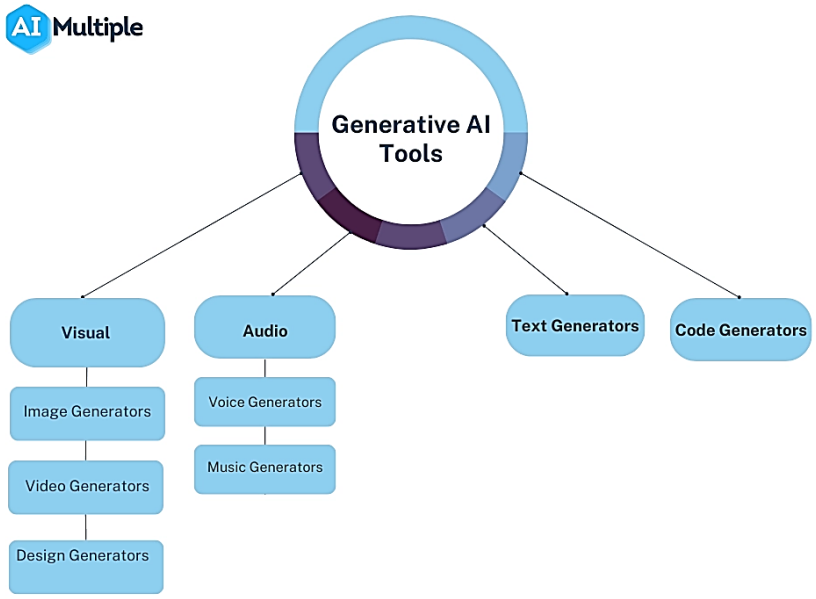


Abbildung 12: aimultiple.com

Die ‘Erzeugnisse‘ der Generative-Artificial-Intelligence-Tools können folgende Formen annehmen:

- visuell, in Form von Bildern, Videos oder Designelementen,
- Audio, d. h. gesprochene Sprache oder Musik,
- Text oder
- Code in einer Programmiersprache.

Visuell

Midjourney kreiert aus Textbeschreibungen Bilder und wird durch Befehle an einen Discord-Bot gesteuert. Das Programm erschafft zunächst eine Vorschau mit vier möglichen Entwürfen, die zur Weiterverarbeitung genutzt werden können.

Ein von OpenAI entwickeltes vergleichbares Programm ist **DALL-E**. Hier werden ebenso Bilder aus Textbeschreibungen erzeugt.

CleanupPictures fällt in die Kategorie Fotobearbeitung, speziell in den Bereich Fotobereinigung. Benutzer können mit dieser Anwendung unerwünschte Elemente wie Wasserzeichen oder Mängel automatisiert aus Fotos entfernen.

Zum Erstellen von Präsentationen eignet sich **Tome.** Darin können Templates ausgewählt werden, aus denen anhand der Anweisungen eine Präsentation erzeugt wird. Für die Texterstellung wird ChatGPT, für die Bebilderung DALL-E 2 genutzt.

Der Videogenerator **Runway Gen-1** erzeugt aus hochgeladen Bildern, Videodateien und Textprompts Videos.

Videobearbeitung auf der Basis künstlicher Intelligenz ermöglicht **Pictory**. Das Tool verhilft zu einer schnellen Nachbearbeitung von Videos einschließlich dem Herausschneiden unliebsamer Sequenzen oder Kürzungen. Aus langen Videos lassen sich auf Knopfdruck kurze Trailer generieren, die die Highlights des Videos aufgreifen. Es können auch Skripte oder Blogposts in Videos umgewandelt werden und Texte automatisiert in Videos eingeblendet werden.

Audio

Das Noise-Cancelling-Tool **Krisp** kann störende Hintergrundgeräusche in Videokonferenzen mit Programmen wie Skype, Microsoft Teams und Zoom per Mausklick herausfiltern. In puncto Google-Suchvolumen kann Krisp zwar nicht mit ChatGPT oder Copy.ai mithalten, es rangiert aber immerhin im Mittelfeld der zehn populärsten Künstliche-Intelligenz-Tools.

Mit **otter.ai** kann gesprochene Sprache in Text transkribiert werden. Dadurch können z. B. in Videos automatisch Untertitel bzw. Texte zu Podcasts erzeugt werden. Außerdem können damit die Inhalte von Audiodateien für die Textanalyse zugänglich gemacht werden.

Die umgekehrte Richtung der Umwandlung von Text in Sprache ermöglicht z. B. **Lovo.ai**.

VocalRemover ist in der Lage, Stimme und Musik zu trennen, beispielsweise zur Erstellung von Remixes oder Karaokeversionen eines Songs.

Text

Neben dem Platzhirsch ChatGPT bestehen weitere nützliche Tools zur Textverarbeitung mit künstlicher Intelligenz, die nachfolgend genannt werden.

Google Bard ist ein von Google entwickelter Chatbot auf der Basis von künstlicher Intelligenz, der als direkte Reaktion auf den Erfolg von ChatGPT entwickelt und im März 2023 in eingeschränkter Kapazität veröffentlicht wurde, bevor er im Laufe des Sommers in weiteren Ländern verfügbar wurde.

Character.ai ist ebenfalls eine Chat-Anwendung, die eine größere Verbreitung gefunden hat.

QuillBot ist eine im Jahr 2017 entwickelte Software, die künstliche Intelligenz nutzt, um Texte neu zu schreiben und zu paraphrasieren. Damit lassen sich beispielsweise E-Mails, Social-Media-Posts und Essays automatisiert erstellen und Texte umschreiben. Quillbot.ai nutzt dazu u. a. eine Grammatikprüfung und Funktionen für die Zusammenfassung von Texten. Es stellt sich hier die Frage, ob es möglich ist, z. B. Referate für das Fach Geschichte in der neunten Klasse zuerst in ChatGPT zu erstellen und danach in QuillBot umformulieren zu lassen, um sie damit für automatische genAI-Erkennungssoftware unverdächtig zu machen.

Copy.ai wird von Unternehmen als Textgenerator eingesetzt, beispielsweise für Blogs, Social-Media-Posts oder E-Mails.

Generierung von Programmiercode

Zur Erzeugung von Programmiercode bestehen neben ChatGPT spezialisierte Anwendungen.

OpenAI Codex ist der auf OpenAI-GPT basierende Programmiercode-Generator, der anhand von Milliarden Codezeilen trainiert wurde und derzeit 14 Programmiersprachen unterstützt.

Copilot verwendet öffentlich verfügbaren Code aus GitHub-Repositorys. Das Tool erkennt Fehler im Code und empfiehlt Änderungen. Copilot kann in die üblichen Programmierumgebungen integriert werden.

Der von Google unterstützte **AlphaCode** von DeepMind ermöglicht Entwicklern Zugriff auf Quellcode aus Sprachbibliotheken. Mit AlphaCode können Tausende vorgefertigte Bibliotheken genutzt und so Schnittstellen zu APIs von Drittanbietern schnell und einfach erstellt werden.

GenAI Stack ist ein Framework für die Integration von Large Language Models in jede Anwendung. Damit können eigene Anwendungen schnell um deren mächtige Funktionalitäten erweitert werden.

Ein Blick auf die Häufigkeit der Website-Besuche der Generative-Artificial-Intelligence-Anwendungen hilft, die Übersicht zu bewahren und die bedeutendsten Tools zu identifizieren.

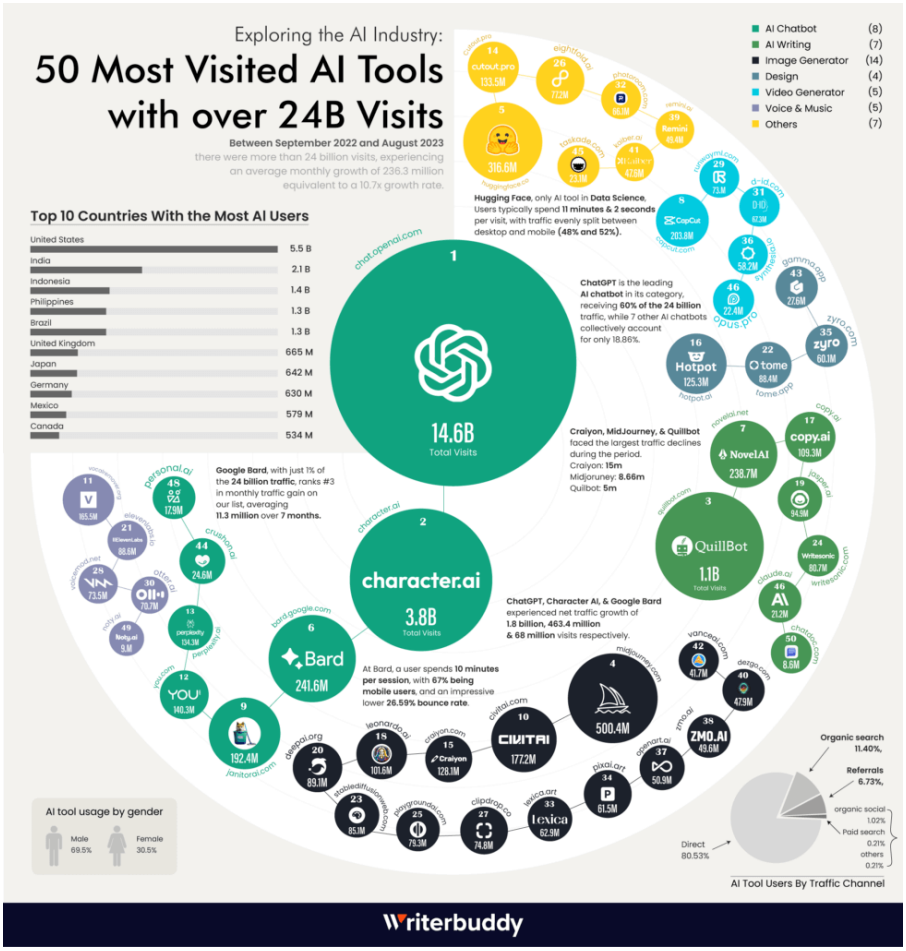


Abbildung 13: writerbuddy.ai.

4.2 Weitere Begriffe im Rahmen der Datenanalyse

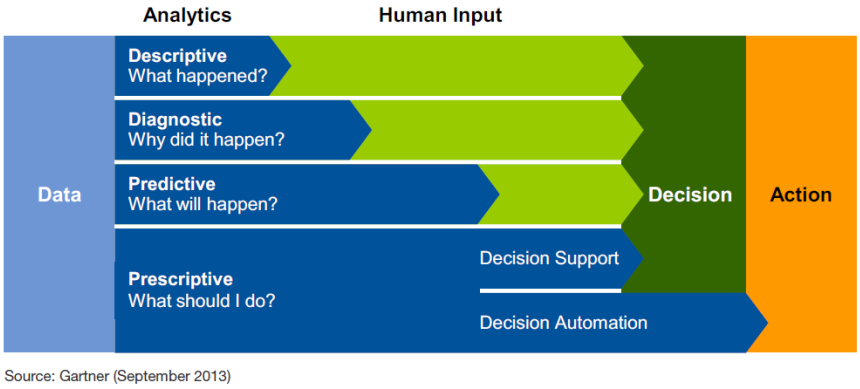
Neben den im vorangegangenen Abschnitt beschriebenen Begriffen gibt es weitere Ausdrücke, die immer wieder auftauchen. Da wird von Big Data Analytics, Data-Mining, Predictive Analytics, BI etc. gesprochen. Anbieter von

Dienstleistungen und Softwarelösungen tragen mit immer neuen Buzz Words auch nicht gerade dazu bei, das begriffliche Durcheinander zu vereinfachen. Im Folgenden sollen daher die unterschiedlichen Begriffe anhand des Datenanalyse-Prozesses eingeordnet und erklärt werden.

Bei dem Prozess geht es darum:

1. Daten zu analysieren,
2. um daraus Erkenntnisse zu gewinnen.
3. Diese Erkenntnisse dienen dann als Grundlage (oder Unterstützung) einer Entscheidung und lösen
4. Handlungen aus.

Gartner hat den Prozess im Zusammenhang mit unterschiedlichen Begriffen in folgender Grafik zusammengefasst.[4]



Werden Daten zusammengefasst und mit deskriptiven statistischen Methoden (also Summieren, Gruppieren, Teilen etc.) dargestellt, spricht man von **Descriptive Analytics**. Die Blickrichtung ist also auf die Vergangenheit ge-

[4] Gartner (2013)

richtet und beschreibt, was passiert ist. Der Anteil, den ein Mensch noch beitragen muss, um zu einer Entscheidung und einer Handlung zu kommen, ist groß.

Eine Bank analysiert die Daten von Kunden, die ihre Kreditraten nicht bezahlt haben. Sie gruppiert sie vielleicht oder ordnet sie geografischen oder sozioökonomischen Faktoren zu.

Diagnostic Analytics: Will man diagnostizieren, wieso bestimmte Sachverhalte eingetreten sind, können mit der Anwendung entsprechender Verfahren (Kausalanalysen) die Gründe gesucht werden. Die Blickrichtung ist zwar immer noch auf die Vergangenheit gerichtet, aber das Ziel ist es, nicht nur Sachverhalte darzustellen, sondern auch Gründe für deren Eintreten zu finden. Der menschliche Anteil bei der Entscheidungsfindung ist schon etwas geringer, da die Analyse eine weitere Entscheidungsgrundlage liefert.

Durch die Analyse der Kreditausfalldaten erkennt die Bank, welche Faktoren die Zahlungsausfälle begünstigt haben.

Die **Predicitve Analytics** nutzt die Erkenntnisse aus der Datenanalyse, um eine Prognose für zukünftige Fälle zu geben. Die Blickrichtung ist also auf die Zukunft gerichtet.

Die Bank erstellt anhand der vorangegangenen Analysen ein Scoring-Modell. Ein neuer Kreditantragsteller wird mit diesem Modell bewertet. Der Scoringwert ist ein Entscheidungskriterium für die Frage, ob und zu welchen Konditionen der Kunde den Kredit bekommt. Es muss aber nicht unbedingt das einzige Kriterium sein. Die 'Nase' des Antragstellers wird bei einem persönlichen Gespräch mit dem Bankmitarbeiter eventuell auch noch bewertet.

Die **Prescriptive Analytics** geht noch einen Schritt weiter und lässt den menschlichen Anteil an einem Entscheidungsprozess überflüssig werden. Es werden anhand der Daten automatische Entscheidungen oder sogar Handlungen ausgelöst.

Für kleinere Konsumentenkredite bietet die Bank im Web die komplette Abwicklung eines Kreditantrages an. Anhand der eingegebenen und vorhandenen Daten über den Antragsteller fällt die Kreditentscheidung automatisch und die Auszahlung wird veranlasst.

Im Hinblick auf die hier vorgenommene Einordnung des Analytics-Begriffes sollen auch die weiteren Begriffe, die im Zusammenhang mit dem Thema Data Science häufig fallen, definiert werden. Es geht dabei nicht um eine wissenschaftlich korrekte Definition der Begriffe, sondern es sollen die Unterschiede und Gemeinsamkeiten der Termini verdeutlicht werden.

- **Business Intelligence**: Der Begriff Business Intelligence (BI) wurde in den 1990er Jahren populär und bezeichnet Verfahren und Prozesse zur systematischen Analyse (Sammlung, Auswertung und Darstellung) von Daten. Ziel ist die Gewinnung von Erkenntnissen, die in Hinsicht auf die Unternehmensziele bessere operative oder strategische Entscheidungen ermöglichen. In der Praxis handelt es sich bei BI-Systemen in der Regel um Software, die es Business Usern ermöglichen soll, schnell und einfach auf aggregierte Zahlen zugreifen zu können. Der vorausschauend analytische Teil spielt eine untergeordnete Rolle und wir bewegen uns im Bereich der *deskriptiven* Analytics, also auf der ersten Stufe der oben gezeigten Abbildung. Mit Intelligenz (im Sinne von Verfahren der künstlichen Intelligenz) hat das noch wenig zu tun.

- **Knowledge Discovery**: Als man damit begann, systematisch Daten zu analysieren, um damit Erkenntnisse zu gewinnen – also ab den 1980er Jahren – wurde oft noch der Begriff Knowledge Discovery (Process) verwendet. Damit wurde der gesamte analytische Prozess des Erkenntnisgewinns aus Daten gemeint, der die *Diagnostic*- und *Predictive*-Aspekte miteinbezog. Eine der bekanntesten und ältesten DataScience-Informationsseiten im Web (kdnuggets.com) hat die Abkürzung KD (Knowledge Discovery) noch in ihrem Namen.

- **Data-Mining**: Eng verbunden mit der Knowledge Discovery ist der Begriff ‘Data-Mining”, also das Schürfen nach Goldschätzen in Datensätzen. Damit sind vor allem die *Diagnostic*- und *Predictive*-Verfahren gemeint, aber auch der gesamte analytische Prozess. Data-Mining ist damit eigentlich nur ein ‘fancy’ Begriff für die Knowledge Discovery und wird zum Teil synonym, zum Teil nur auf den eigentlichen, analysierenden Teil des Gesamtprozesses bezogen, verwendet.
- **Advanced Analytics**: Um sich gegenüber den meist deskriptiven Verfahren im Rahmen von Business-Intelligence-Systemen abzugrenzen, haben v. a. die Data-Mining-Software-Anbieter den Begriff Advanced Analytics geprägt. Auch hier bewegen wir uns im Bereich der diagnostizierenden und vorhersagenden (*Predictive*-) Verfahren.
- **Big Data Analytics**: Mit diesem Begriff soll dem exponentiell wachsenden Datenaufkommen Rechnung getragen werden. Die analytischen Verfahren und Vorgehensweisen sollen also auch auf Big Data angewendet werden. Und mit Big Data sind neben den strukturierten Daten (z. B. Sensordaten) auch die ‘unstrukturierten’ oder weniger strukturierten Daten (z. B. Texte aus sozialen Medien, Sprach- und Bilddateien etc.) gemeint.

4.3 Datentypen und Skalentypen

Da es bei den in den folgenden Abschnitten beschriebenen Verfahren von Bedeutung ist, von welchem Typ die Daten sind, werden an dieser Stelle die unterschiedlichen Datentypen und deren Skalen kurz vorgestellt. Es gibt drei bzw. vier Skalentypen:

- Nominalskala
- Ordinalskala
- Kardinalskala, mit den Ausprägungen Intervallskala und Verhältnisskala

Die Unterschiede sind in folgender Tabelle aufgeführt:

Datentyp/Skala	Beschreibung	Beispiele	Messbarkeit
Nominal	Rein qualitative Merkmalsausprägungen ohne natürliche Ordnung	Geschlecht, Berufsstatus, ja/nein-Fragen	Häufigkeit
Ordinal	Qualitative Merkmalsausprägungen mit natürlicher Ordnung	Benotung (sehr gut, gut, mittel, schlecht, sehr schlecht)	Häufigkeit, Reihenfolge
Kardinal - Intervallskala	Merkmalsausprägungen, die in einer Zahl bestehen und eine Dimension besitzen	Datum	Häufigkeit, Reihenfolge, Abstand,
Kardinal - Verhältnisskala	Merkmalsausprägungen, die in einer Zahl bestehen und eine Dimension und einen Nullpunkt besitzen	Einkommen in Euro, Alter in Jahren	Häufigkeit, Reihenfolge, Abstand, natürlicher Nullpunkt

4.4 Einordnung der Verfahren

Für die Gewinnung von Erkenntnissen aus Daten werden unterschiedliche Methoden verwendet. Diese stammen aus den Bereichen der künstlichen Intelligenz, des maschinellen Lernens, des Data-Mining, der Statistik und der Mathematik.

Unterteilung nach Aufgaben
Die Verfahren lassen sich nach ihren Aufgaben bzw. deren Absicht unterscheiden. Die Abgrenzung ist zwar nicht immer eindeutig, aber sie hilft dabei, grundsätzliche Unterschiede der Verfahren zu berücksichtigen:

Klassifikation	Zuordnung zu vorgegebenen Klassen
Prognose / Vorhersage	Berechnung von unbekannten Werten anhand bekannter Werte
Segmentierung	Bildung von möglichst gleichartigen Gruppen anhand von Ähnlichkeiten
Abhängigkeitsanalyse	Assoziationsanalysen: Entdeckung und Quantifizierung von Abhängigkeiten
Abweichungsanalyse	Entdecken von „Ausreißern“

Bei der **Klassifikation** werden vor der Datenanalyse Klassen oder Kategorien festgelegt, denen dann einzelne Elemente zugeordnet werden. Dies geschieht aufgrund von Vergleichen zwischen Klasseneigenschaften und Objektmerkmalen. Als Beispiel kann das Thema Kreditwürdigkeit herangezogen werden. Anhand von Ausprägungen verschiedener Variablen (Einkommen, Alter, Wohnort, Familienstand, Bildung etc.) kann eine Zuordnung zu den beiden Klassen ‘kreditwürdig’ oder ‘nicht kreditwürdig’ vorgenommen werden.

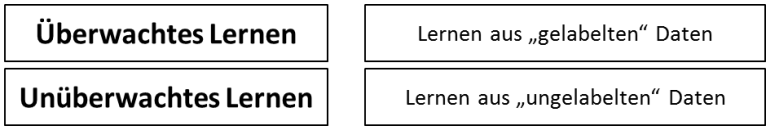
Die **Vorhersage** oder Prognose eines Wertes (einer abhängigen Variablen) auf Basis der Werte anderer Merkmale (unabhängiger Variablen), bzw. anhand von Werten der gleichen Variable aus früheren Perioden, wird als Prognose bezeichnet. Streng genommen kann die Klassifikation auch eine Prognose sein, da ja die Zuordnung zu einer Klasse anhand der Werte der untersuchten Variablen eine Prognose bedeuten kann (ob der Kunde kreditwürdig ist). Der Unterschied der Vorhersage-Methoden ist der, dass dabei eine Berechnung eines stetigen Wertes vorgenommen wird. Der exaktere Begriff für diese Art der Verfahren wäre daher die Vorhersageberechnung. Telefongesellschaften prognostizieren beispielsweise den jährlichen Umsatz eines Kunden mit den entsprechenden Vorhersagemodellen.

Bei der **Segmentierung** werden Objekte in Gruppen zusammengefasst. Die Gruppen sind – im Unterschied zur Klassifizierung – nicht vorgegeben oder bekannt. Es ist vielmehr die Aufgabe der Segmentierung, die Gruppen (Cluster) herauszuarbeiten, mit dem Ziel, die Ähnlichkeit der Gruppenmitglieder möglichst groß und die Gruppen untereinander möglichst unterschiedlich zu wählen. Konsumgüterhersteller versuchen z. B. Ihre Kunden zu segmentieren, um sie dann unterschiedlich anzusprechen.

Die **Abhängigkeitsanalyse** hat das Ziel, Beziehungen zwischen verschiedenen Objekten oder zwischen Merkmalen eines Objektes zu finden. Dies kann sich auf einen bestimmten Zeitpunkt oder verschiedene Zeitpunkte beziehen. Häufig wird das z. B. in der Warenkorbanalyse eingesetzt, um Abhängigkeiten, also Assoziationen zwischen Produkten, aufzudecken, um entsprechend darauf reagieren zu können.

Bei der **Abweichungsanalyse** sollen Ausreißer identifiziert werden, deren Eigenschaften von denen der anderen Objekte signifikant abweichen. Ziel ist es, die Ursachen für diese Abweichungen auszumachen. Dies kann z. B. in der Qualitätskontrolle in der Fertigung eingesetzt werden.

Einteilung nach Lernmethode
Eine weitere Dimension, nach der die Verfahren eingeteilt werden können, ist die Frage, ob es sich bei dem Verfahren um ‘überwachtes‘ oder ‘unüberwachtes‘ Lernen handelt (Supervised Learning versus Unsupervised Learning).

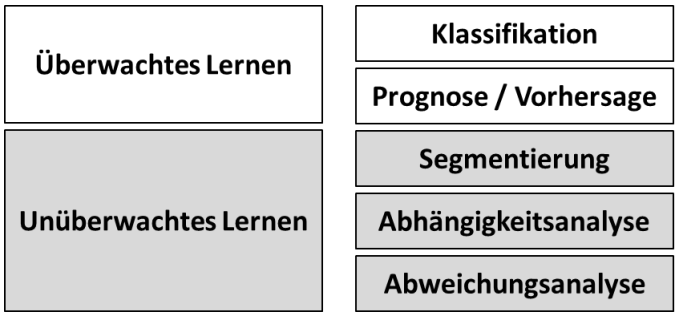


Der Begriff stammt aus dem Bereich des maschinellen Lernens. Dabei unterscheidet man hauptsächlich – aber nicht ausschließlich – zwischen zwei Arten von Lernmethoden: Supervised (überwachtes) und Unsupervised Learning (unüberwachtes Lernen).

Beim überwachten Lernen haben wir Daten vorliegen, die schon ein Ergebnis oder ‘Label’ enthalten. In anderen Worten: jeder Datenpunkt in der vorhandenen Datenmenge besteht aus Eingabe- und Ausgabewerten. Beim unüberwachten Lernen handelt es sich um das allgemeine Verstehen der vorliegenden Daten und die Entdeckung versteckter Strukturen. Die Daten sind also nicht durch ‘Labels’ gekennzeichnet.

Die Frage nach überwachtem oder unüberwachtem Lernen ist also weniger ‘dramatisch’ als es die Begriffe vermuten lassen. Es geht nicht um eine echte Überwachung des Lernprozesses (als Vater denke ich da unwillkürlich an die Hausaufgabenüberwachung meiner Tochter in der ersten Klasse), sondern lediglich darum, ob die Daten gelabelt sind oder nicht. Gibt es in der Datentabelle eine Spalte mit Ergebniswerten (z. B. Betrugsfall ja/nein oder ein Umsatzwert, der prognostiziert werden soll) oder soll aus der Gesamtheit der Daten ein Ergebnis ermittelt werden (z. B. verschiedene Cluster oder Abhängigkeiten)?

Auch wenn es nicht immer ganz eindeutige Zuordnungen gibt, kann man sagen, dass sich die Klassifikations- und Vorhersageverfahren dem überwachten Lernen zuordnen lassen. Segmentierung, Abhängigkeitsanalyse und Abweichungsanalyse gehören dagegen meist zum unüberwachten Lernen.



Die hier beschriebenen Verfahrensarten beziehen sich grundsätzlich auf Verfahren für strukturierte Daten. Bildlich gesprochen sind das also Daten, die man in Form einer Tabelle darstellen kann, wobei die Spalten die Variablen und die Zeilen die einzelnen ‘Fälle’ darstellen.

Daneben gibt es im Rahmen der künstlichen Intelligenz noch weitere Verfahren bzw. Konzepte, die sich auf eher unstrukturierte Daten beziehen. Unter dem Begriff ‘Struktur’ kann man hier die Möglichkeit verstehen, die Daten z. B. in der Form einer Tabelle darzustellen. Unstrukturierte Daten – also z. B. Bild- und Sounddateien oder Fließtexte – haben natürlich auch in irgendeiner Form eine Struktur, diese lässt sich aber nicht direkt in Tabellenform überführen.

Die weiteren Gebiete im Rahmen der künstlichen Intelligenz beziehen sich vorwiegend auf ‘unstrukturierte’ Daten. Im Folgenden wird auf fünf dieser Begriffe kurz eingegangen:

Text Mining	Entdeckung von Bedeutungs-strukturen aus Textdaten
Sprachverarbeitung NLP	Erkennen und Verarbeiten natürlicher Sprache
Bilderkennung	Erkennen und Verstehen von Bildinformationen
Expertensysteme	Bündelung von Spezialwissen zur Entscheidungsunterstützung
Selbstlernende Systeme	Ohne menschlichen Input sich selbst verbessernde Systeme

Beim **Text Mining** handelt es sich um ein Verfahren, das das Ziel hat, Bedeutungen und Strukturen von Texten zu entdecken. Es handelt sich dabei in der Regel um ein zweistufiges Vorgehen. In einem ersten Schritt werden die *unstrukturierten* Texte strukturiert. Dies kann durch das Erzeugen von sogenannten Tokens erfolgen. Dabei wird der Text in logisch zusammenhängende Tokens, also Texteinheiten (z. B. Wörter oder auch zusammengesetzte Ausdrücke) zerlegt, die dann weiterverarbeitet werden können. Bei der Weiterverarbeitung werden z. B. Worthäufigkeiten oder die Entfernung von Wortpaaren bzw. -gruppen ermittelt. Die Texte werden quantifiziert und strukturiert, sodass im zweiten Schritt die klassischen multivariaten Verfahren für strukturierte Daten (siehe detaillierte Beschreibung der Verfahren im folgenden Abschnitt) angewendet werden können. Beliebte Libraries sind in Python NLTK oder text.tokenizer in Keras.

Um natürliche Sprache verarbeiten zu können, reicht die Strukturierung der Texte nicht aus. Es muss der Sinn des Textes verstanden werden. Die **Sprachverarbeitung** natürlicher Sprache (**NLP=Natural Language Processing**) setzt also voraus, dass die Bedeutung von Worten und Sätzen trainiert wurde. Da dies ein extrem aufwendiges Vorhaben ist und eine große Menge an Trainingstexten voraussetzt, empfiehlt es sich, APIs mit trainierten NLP-Modellen zu verwenden bzw. auf diesen aufzubauen. Die APIs haben unterschiedliche

Schwerpunkte (Übersetzung, Sentimentanalyse, Textzusammenfassung, Topic Tagging etc.) und sind oft frei verfügbar. Beispielsweise können folgende APIs genannt werden:

- Systran.io API
- Aylien API
- Text Summarization API
- Twinword Text Analysis Bundle API
- IBM Watson Alchemy API
- RxNLP API
- Linguakit API

Google verfügt mit **BERT** (Bidirectional Encoder Representations from Transformers) über ein vortrainiertes *Sprachverstehen-Modul*, das den Sinn von ganzen Sätzen erkennen soll. Google setzt dieses für Suchanfragen in der Google-Suchmaschine ein, stellt aber auch eine API als Open-Source-Modell zur Verfügung.

Ist der Ausgang der Sprache kein geschriebener Text, sondern gesprochene Sprache, so muss dem NLP-Prozess eine Speech2Text-Transformation vorgeschaltet sein. Aus einer *Sounddatei* wird also eine Textdatei erzeugt. Auch dafür empfiehlt es sich, auf vorhandene APIs zurückzugreifen. Anbieter sind v. a. die BIG-AI-Player:

- Google Speech-To-Text
- Microsoft Cognitive Services
- Dialogflow
- IBM Watson
- Speechmatics

In der **Bilderkennung** wird versucht, Objekte in einem Bild zu erkennen und diesen eine Bedeutung zuzuordnen. Dabei kann es sich um Gesichtserkennung, Identifizierung von Gegenständen, Erkennen von Zuständen (z. B. ein

Nudity-Detector bei Facebook), Texten oder bestimmter Muster handeln. Bilder von Texten können einem Texterkennungsprozess (OCR=optical character recognition) unterzogen werden, der daraus Text extrahiert, der dann weitergehend analysiert werden kann.

Auch bei der Bilderkennung empfiehlt es sich, auf vortrainierte APIs zuzugreifen. Bedeutende Anbieter sind:

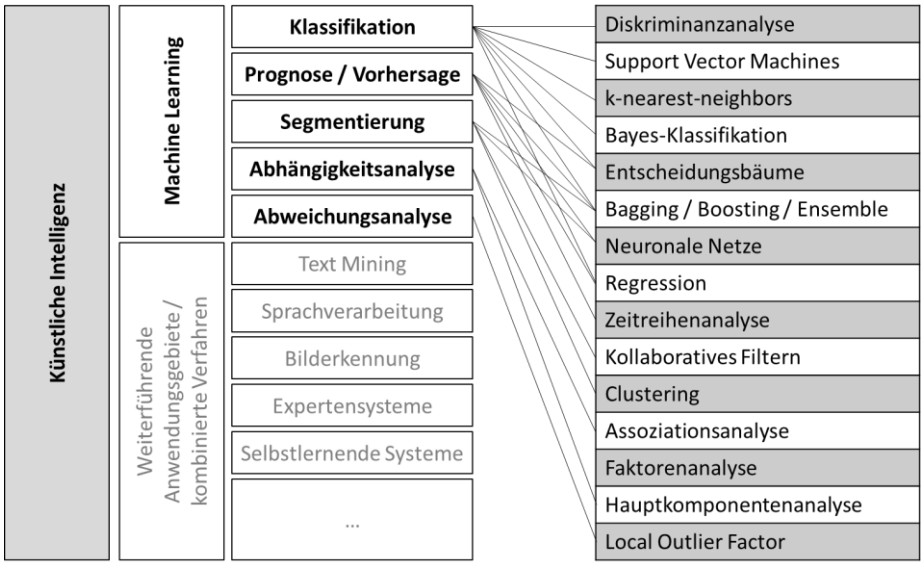
- Google CloudVision API
- Amazon Rekognition
- IBM Watson Visual Recognition
- Microsoft Image Processing API
- Clarifai
- Imagga

In **Expertensystemen** werden das Spezialwissen und die Schlussfolgerungsfähigkeit qualifizierter Fachleute auf einem eng begrenzten Anwendungsgebiet im Computer nachgebildet. Die so entstandenen Systeme sollen Fachleute bei ihren Entscheidungen unterstützen.

Bei **selbstlernenden Systemen** geht es darum, (Computer-)Systeme zu entwickeln, die in der Lage sind, durch die Verarbeitung von Informationen neues Wissen zu generieren und vorhandenes Wissen zu verbessern, ohne dass ein programmierendes Eingreifen eines Menschen notwendig wäre.

4.5 Analyseverfahren – Machine Learning-Algorithmen

In diesem Kapitel werden die wichtigen Verfahren vorgestellt. Es kann zwar keine detaillierte Beschreibung der Verfahren erfolgen und es kann auch nicht auf jede (statistische) Feinheit eingegangen werden, aber dieser Abschnitt soll zumindest als Einstieg dienen und ein 'Gefühl' für die Möglichkeiten und Grenzen der einzelnen Verfahren vermitteln. Die Abbildung stellt zusammenfassend die im Folgenden beschriebenen Verfahren, in Zusammenhang mit den Gliederungen aus dem vorangegangenen Kapitel, dar. Diese Zuordnungen sind nicht immer eindeutig. In der Abbildung wurden die wichtigsten bzw. 'üblichen' Zusammenhänge zwischen Verfahren und Zweck abgebildet.



In den folgenden Unterkapiteln wird jeweils ein Verfahren besprochen. Dabei wird am Anfang jedes Abschnitts in einer Grafik das entsprechende Verfahren einer der oben vorgestellten Aufgaben und Lern-Arten zugeordnet.

Verfahrensname: Unterschiedliche Ausprägungen des Verfahrens, …			
Absicht / Zweck	**Lernen**	**Analys. Daten**	**Ergebnis**
Klassifikation	Überwachtes Lernen	Nominalskala	Nominalskala
Prognose / Vorhersage		Ordinalskala	Ordinalskala
Segmentierung	Unüberwachtes Lernen	Kardinalskala	Kardinalskala
Abhängigkeitsanalyse			
Abweichungsanalyse			

Die Anforderung an das Skalenniveau der Variablen (unabhängige Variablen) der untersuchten Daten ist angegeben. Bei Verfahren, bei denen auch abhängige Variablen existieren (überwachtes Lernen), wird das Skalenniveau des Ergebnisses (der abhängigen Variable) angegeben, ansonsten die Art des Ergebnisses.

4.5.1 Diskriminanzanalyse

<table>
<tr><td colspan="4">Diskriminanzanalyse: Allgemeine lineare Diskriminanzanalyse,
Fischersche Dka, Quadratische Dka, Regularisierte Dka</td></tr>
<tr><th>Absicht / Zweck</th><th>Lernen</th><th>Analys. Daten</th><th>Ergebnis</th></tr>
<tr><td>Klassifikation</td><td rowspan="2">Überwachtes Lernen</td><td>Nominalskala</td><td>Nominalskala</td></tr>
<tr><td>Prognose / Vorhersage</td><td>Ordinalskala</td><td>Ordinalskala</td></tr>
<tr><td>Segmentierung</td><td rowspan="3">Unüberwachtes Lernen</td><td>Kardinalskala</td><td>Kardinalskala</td></tr>
<tr><td>Abhängigkeitsanalyse</td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>Abweichungsanalyse</td></tr>
</table>

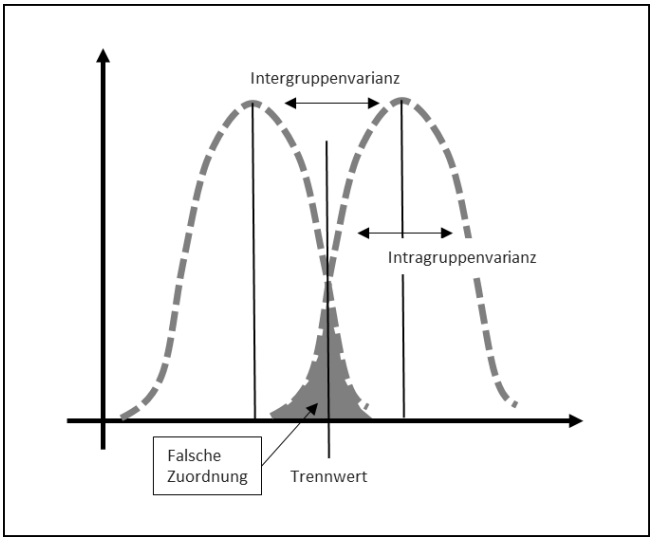
Die Diskriminanzanalyse ist ein Verfahren zur Analyse von Gruppenunterschieden. Es werden die ‘diskriminierenden’ Variablen erkannt, die für die Gruppenzugehörigkeit sorgen. Dadurch kann anhand der vorliegenden Ausprägung der unabhängigen Variablen auf die Gruppierung der abhängigen Variable geschlossen werden (z. B. bei der Kreditwürdigkeitsprüfung).

Die Diskriminanzanalyse erstellt ein Vorhersagemodell für Gruppenzugehörigkeiten. Dieses Modell besteht aus einer Diskriminanzfunktion (oder bei mehr als zwei Gruppen einem Set von Diskriminanzfunktionen), die auf der Grundlage derjenigen linearen Kombinationen der Prädiktorvariablen bestimmt wird, die die beste Diskriminanz zwischen den Gruppen ergeben. Die Funktionen werden aus einer Stichprobe der Fälle generiert, bei denen die Gruppenzugehörigkeit bekannt ist. Diese Funktionen können dann auf neue Fälle mit Werten für die Prädiktorvariablen zur Bestimmung der Gruppenzugehörigkeit angewandt werden.

Unabhängige Variablen					Abhäng. Variable
Einkommen	**Alter**	**Wertpapier-bestand**	**Kunde seit Jahren**	**…**	**Kreditausfall**
45.000	58	50.000	25		ja
28.888	27	232.000	5		ja
26.000	31	0	1		nein
…					

Die Bildung der Diskriminanzfunktion erfolgt unter folgenden Bedingungen:

- Die Varianz zwischen den Gruppenmittelwerten (Intergruppenvarianz) sollte möglichst groß sein.
- Die Varianz innerhalb einer Gruppe (Intragruppenvarianz) sollte möglichst klein sein.
- Die sich 'überlappende' Fläche (die eine Falschklassifikation bedeutet) sollte möglichst klein sein.



Neben der **allgemeinen linearen Diskriminanzanalyse** (Annahme Gleichverteilung, gleiche Gruppengröße) werden weitere Arten der Diskriminanzanalyse verwendet. Die unterschiedlichen Ausprägungen der Diskriminanzanalyse unterscheiden sich bezüglich ihrer Annahmen (Normalverteilung, gleiche Häufigkeit der Gruppenmitglieder), der Anzahl der Gruppen, der Anzahl der eingeschlossenen Variablen, der Art der Diskriminanzfunktion und dem Vorgehen zur Auswahl der Variablen.

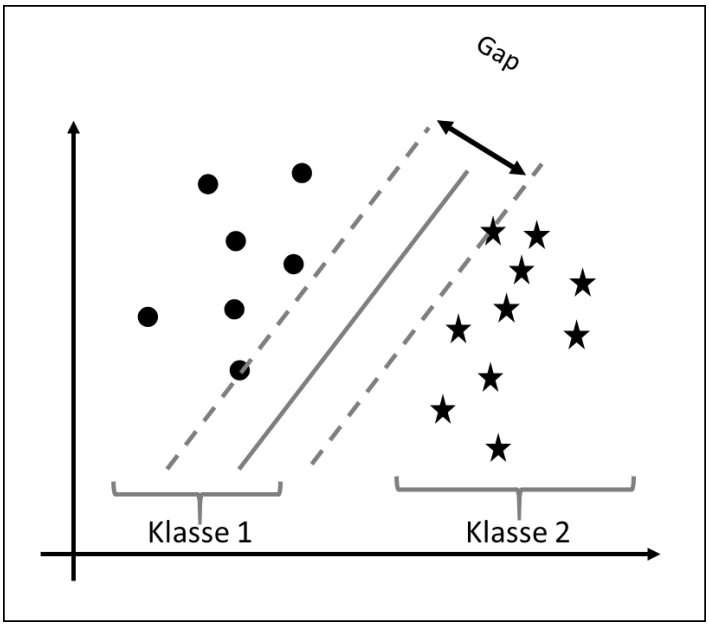
Weitere Beispiele für wichtige Verfahren sind:

- Bayes'sche Diskriminanzanalyse (keine gleiche Gruppengröße)
- Fischer'sche Diskriminanzanalyse (Ziel der Dimensionsreduktion)
- Quadratische Diskriminanzanalyse (unterschiedliche Erwartungswerte in den Gruppen und unterschiedliche Kovarianzmatrizen)
- Regularisierte Diskriminanzanalyse (Regularisierungs- oder Glättungsmethoden zur Verringerung der zu schätzenden Parameter)

4.5.2 Support Vector Machine – SVM

Support Vector Machine (SVM):			
Absicht / Zweck	**Lernen**	**Analys. Daten**	**Ergebnis**
Klassifikation	Überwachtes Lernen	Nominalskala	Nominalskala
Prognose / Vorhersage		Ordinalskala	Ordinalskala
Segmentierung	Unüberwachtes Lernen	Kardinalskala	Kardinalskala
Abhängigkeitsanalyse			
Abweichungsanalyse			

Support Vector Machines sind statistische Verfahren, die vor allem zur Klassifizierung von Datenobjekten verwendet werden. Bei dem Verfahren geht es darum, eine Grenzfläche zu finden, die einen möglichst breiten Bereich zwischen den Grenzen der einzelnen Klassen bildet. Dazu wird eine Trainingsdatenmenge benötigt, bei der die Klassenzugehörigkeit bekannt ist.



Die Grenzfläche kann entweder einer linearen Funktion folgen, oder auch von nicht linearem Charakter sein. Dazu werden die Daten so lange in einen höherdimensionalen Raum transformiert, bis eine lineare Trennung möglich ist. Diese lineare Trennung aus einem höherdimensionalen Raum wird dann wieder in die Ausgangsdimension zurücktransformiert und erscheint dann als ‘krumme Linie’, die die Gruppen trennt. Zur Verringerung des Rechenaufwandes können dabei diverse Verfahren (z. B. Kernel-Trick) Anwendung finden.

Da in der Praxis meist keine vollständige, eindeutige Klassifizierung möglich ist, werden sog. Schlupfvariablen eingeführt. Diese bewerten einzelne ‘Abweichler’ aus den Gruppen, ermöglichen so ein einfacheres Klassifizierungsmodel und vermeiden Überqualifizierung.

4.5.3 Nächste-Nachbar-Klassifikation - *k*-Nearest Neighbor

<table>
<tr><th colspan="4">Nächste-Nachbar-Klassifikation / k-Nearest Neighbor</th></tr>
<tr><th>Absicht / Zweck</th><th>Lernen</th><th>Analys. Daten</th><th>Ergebnis</th></tr>
<tr><td>Klassifikation</td><td rowspan="2">Überwachtes Lernen</td><td>Nominalskala</td><td>Nominalskala</td></tr>
<tr><td>Prognose / Vorhersage</td><td>Ordinalskala</td><td>Ordinalskala</td></tr>
<tr><td>Segmentierung</td><td rowspan="3">Unüberwachtes Lernen</td><td>Kardinalskala</td><td>Kardinalskala</td></tr>
<tr><td>Abhängigkeitsanalyse</td><td rowspan="2">Transformation nominal oder ordinal skalierter Daten möglich</td><td rowspan="2"></td></tr>
<tr><td>Abweichungsanalyse</td></tr>
</table>

Der nächste Nachbar (Nearest Neighbor) ist ein Maß der Entfernung multidimensionaler Datenpunkte mit kardinalen Variablenwerten. Ordinale und nominale Daten können transformiert werden, um so die Entfernung zu ermitteln. Die Entfernung kann gewichtet oder ungewichtet berechnet werden.

Dieses Entfernungsmaß ist die Grundlage für die Nächste-Nachbar-Klassifikation (*k*-Nearest Neighbor oder *k*-NN), ein Verfahren, das vor allem als Klassifikationsverfahren Anwendung findet.

Die Nächste-Nachbar-Analyse ist eine Methode für die Klassifikation von Fällen nach ihrer Ähnlichkeit mit anderen Fällen. Es handelt sich um eine Vorgehensweise für die Mustererkennung in Daten, ohne dass ‘gelabelte’ Daten vorliegen. Ähnliche Fälle liegen nah beieinander und Fälle mit geringer Ähnlichkeit sind weit voneinander entfernt. Der Abstand zwischen zwei Fällen kann als Maß für ihre Unähnlichkeit herangezogen werden. Fälle, die nahe

beieinanderliegen, werden als Nachbarn bezeichnet. Ein neuer Fall wird entsprechend seinem Abstand zu den Fällen im Modell berechnet. Der Fall wird in die Kategorie eingeordnet, die die größte Anzahl an nächstgelegenen Nachbarn aufweist. Der Wert k bedeutet die Anzahl der Nachbarn.

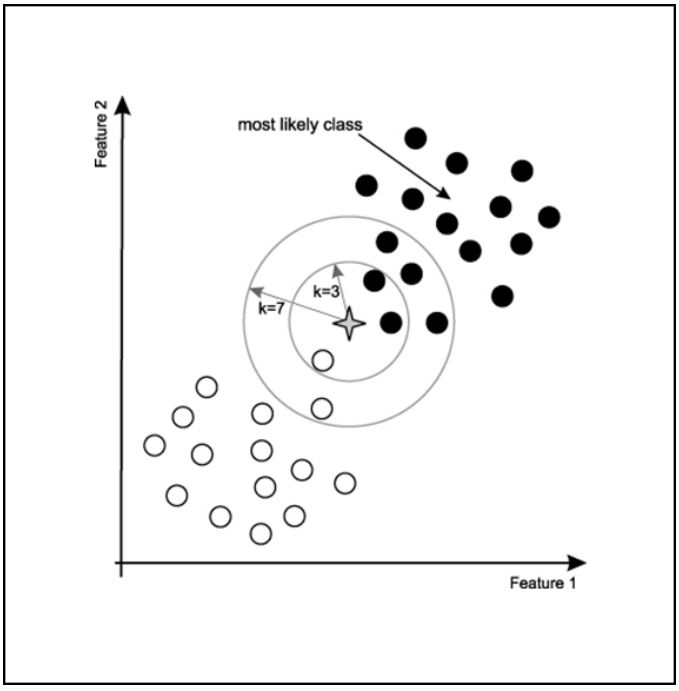


Abbildung 14: Quelle: https://mayuresha.wordpress.com/category/uncategorized/page/3/

Das Verfahren ist eine vergleichsweise einfache Methode der Klassifikation. Es wird kein Wissen (gelabelte Daten) über die Ausprägung der Trainingsdaten benötigt. Das Verfahren liefert die Klasse, wobei diese Einordnung einer Interpretation durch den Nutzer bedarf. Bei großen Datensätzen steigt der Rechenaufwand exponentiell, was zum Einsatz von vereinfachenden Algorithmen geführt hat.

Eine Anmerkung zur Begrifflichkeit: Nearest Neigbor wird auch in Regressions- und Clusterverfahren eingesetzt, dabei aber als Maß der Distanz. Es handelt sich dann logischerweise nicht mehr um ein k-NN-Klassifikationsverfahren, sondern entsprechend um Regressionen bzw. Clusterverfahren.

4.5.4 Bayes-Klassifikation

<table>
<tr><td colspan="4">Bayes-Klassifiaktion: Optimale-Bayes-Klassifikation, Naive-Bayes-Klassifikation</td></tr>
<tr><th>Absicht / Zweck</th><th>Lernen</th><th>Analys. Daten</th><th>Ergebnis</th></tr>
<tr><td>Klassifikation</td><td rowspan="2">Überwachtes Lernen</td><td>Nominalskala</td><td>Nominalskala</td></tr>
<tr><td>Prognose / Vorhersage</td><td>Ordinalskala</td><td>Ordinalskala</td></tr>
<tr><td>Segmentierung</td><td rowspan="3">Unüberwachtes Lernen</td><td>Kardinalskala</td><td>Kardinalskala</td></tr>
<tr><td>Abhängigkeitsanalyse</td><td rowspan="2">Texte</td><td rowspan="2"></td></tr>
<tr><td>Abweichungsanalyse</td></tr>
</table>

Die Bayes-Klassifikation ist eine statistische Klassifikationsmethode, die die Wahrscheinlichkeit vorhersagt, mit der ein Objekt zu einer bestimmten Gruppe gehört. Sie basiert auf der Formel von Bayes, mit der die bedingte Wahrscheinlichkeit eines Ereignisses unter Bedingungen berechnet werden kann.

Bayes' Klassifikationsverfahren gehören zu den überwachten Klassifikatoren, da sie erst durch Trainingsdaten mit bekannter Klassifikation trainiert und dann auf neue Instanzen angewendet werden können. Die Entscheidungsregel funktioniert nach dem Prinzip, eine neue Instanz der Klasse zuzuordnen, bei der die berechnete Wahrscheinlichkeit für diese Klasse am größten ist.
Dabei wird eine A-priori-Wahrscheinlichkeit mit den gewichteten bedingten Wahrscheinlichkeiten berechnet.

Bei vielen Variablen wird die Berechnung der (optimalen) Bayes-Klassifikation sehr aufwendig, sodass als Näherung die **Naive Bayes-Klassifikation** als Verfahren angewendet werden kann. Alle Attribute werden dabei so behandelt als wären sie statistisch unabhängig. Damit entfällt die Notwendigkeit der Berechnung der bedingten 'Kreuzwahrscheinlichkeiten' und der Berechnungsaufwand steigt nicht exponentiell mit der Anzahl der Variablen. Obwohl die Annahme der Unabhängigkeit der Variablen in der Praxis häufig verletzt wird, liefert die Naive Bayes-Klassifikation trotzdem gute Ergebnisse (zumindest für den Fall, dass sich die Korrelationen in Grenzen halten).

Die Funktion sieht dann so aus:

$$\text{argmax}_c\, P(C=c) \prod_{i=1}^{k} P(F_i=f_i \mid C=c)$$

Da so eine Formel kein Mensch versteht, soll das Prinzip der Naiven Bayes-Klassifikation an einem Beispiel erläutert werden.[5]

Wir haben Früchte, die mit drei Variablen beschrieben werden (Länge, Geschmack, Farbe). In der Tabelle sind die 1.000 Trainingsdaten dargestellt, mit denen das Klassifizierungsmodell erstellt werden soll.

	Länge		**Geschmack**		**Farbe**		**Gesamt**
	lang	kurz	süß	nicht süß	gelb	and. Farbe	
Banane	400	100	350	150	450	50	**500**
Orange	0	300	150	150	300	0	**300**
Sonst. Frucht	100	100	150	50	50	150	**200**
Gesamt	**500**	**500**	**650**	**350**	**800**	**200**	**1000**

[5] In Anlehnung an http://stackoverflow.com/questions/10059594/a-simple-explanation-of-naive-bayes-classification

Die A-priori-Wahrscheinlichkeiten sind:

$P_{(Banane)} = 0{,}5 \quad P_{(Orange)} = 0{,}3 \quad P_{(Sonst)} = 0{,}2$

$P_{(lang)} = 0{,}5 \quad P_{(süß)} = 0{,}65 \quad P_{(gelb)} = 0{,}8$

Likelihood – Ausprägungswahrscheinlichkeiten:

$P_{(lang \mid Banane)} = 0{,}8$ (400/500) $\quad P_{(lang \mid Orange)} = 0$ (es gibt keine langen Orangen)

…..

$P_{(and.\ Farbe \mid Orange)} = 0 \quad P_{(and.\ Farbe \mid Sonst.)} = 0{,}75$ (150/200)

Soll nun eine neue Frucht klassifiziert werden, die **lang, süß und gelb** ist:

1. So berechnet man zuerst die bedingte Wahrscheinlichkeit, dass die Frucht eine Banane, eine Orange oder eine andere Frucht ist:

   $P_{(Banane \mid lang,\ süß,\ gelb)} = 0{,}252$

   $P_{(Orange \mid lang,\ süß,\ gelb)} = 0$

   $P_{(Sonst. \mid lang,\ süß,\ gelb)} = 0{,}01875$

   Die Formel für die Berechnung ist das Produkt aus den Ausprägungswahrscheinlichkeiten mit der A-priori-Wahrscheinlichkeit, geteilt durch die A-priori-Wahrscheinlichkeiten der Variablen. Muss man nicht verstehen, sieht aber z. B. für die Banane so aus:

   $$P_{(Banane \mid lang,süß,gelb)} = (P_{(lang \mid Banane)} * P_{(süß \mid Banane)} * P_{(gelb \mid Banane)} * P_{(Banane)}) / (P_{(lang)} * P_{(süß)} * P_{(gelb)})$$

2. Danach wählt man den wahrscheinlichsten Wert aus, sofern man mit dem Unterschied der Wahrscheinlichkeit zufrieden ist. Da die lange,

süße, gelbe Frucht mit dem Wert 0,252 mehr als 10-fach so wahrscheinlich eine Banane als eine sonstige Frucht ist (0,01875), kann man die unbekannte neue Frucht ‘guten Gewissens’ als Banane klassifizieren.

Die naive Bayes-Klassifikation wird häufig für die Klassifikation von Texten verwendet, beispielsweise in Spam-Filter, die E-Mails als Spam bzw. kein Spam kategorisieren.

4.5.5 Entscheidungsbäume

Entscheidungsbaum: Klassifikationsbäume, Regressionsbäume, Entscheidungswälder

Absicht / Zweck	Lernen	Analys. Daten	Ergebnis
Klassifikation	Überwachtes Lernen	Nominalskala	Nominalskala
Prognose / Vorhersage		Ordinalskala	Ordinalskala
Segmentierung	Unüberwachtes Lernen	Kardinalskala	Kardinalskala
Abhängigkeitsanalyse			*Entscheidungs-regeln / -baum*
Abweichungsanalyse			

Entscheidungsbäume sind Baumstrukturen, die der Darstellung von Entscheidungsregeln dienen. Sie veranschaulichen hierarchisch aufeinanderfolgende Entscheidungen. Ihre Anwendungsgebiete umfassen dabei die automatische Klassifizierung und die Herleitung von formalen Regeln aus Erfahrungswissen. Ein Entscheidungsbaum besteht immer aus einem Wurzelknoten und beliebig vielen inneren Knoten sowie mindestens zwei Blättern. Dabei repräsentiert jeder Knoten eine logische Regel und jedes Blatt eine Antwort auf das Entscheidungsproblem.

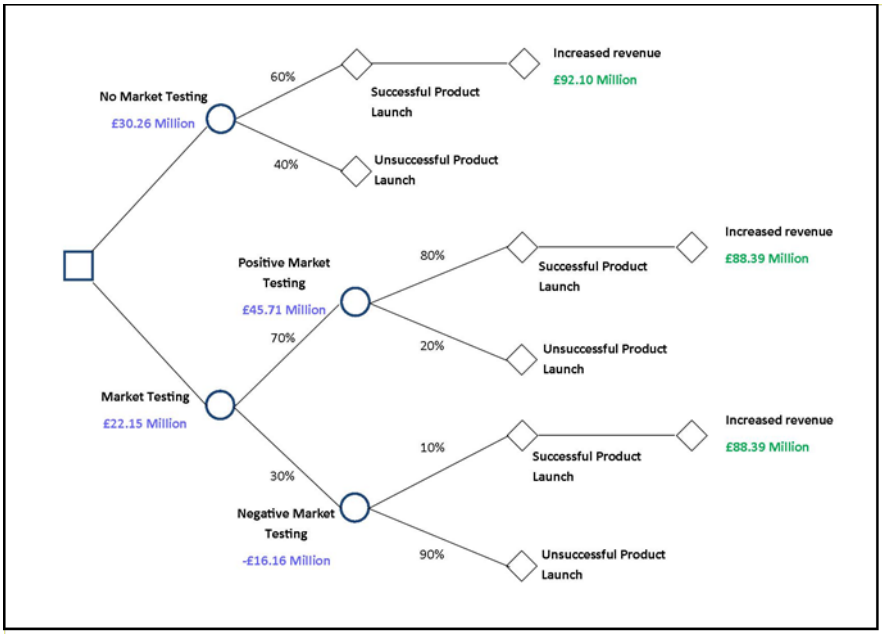


Abbildung 15 Quelle: http://mosaic.cnfolio.com/B302CW2011B135

Grundsätzlich lassen sich Entscheidungsbäume in zwei Varianten unterteilen: die Klassifikationsbäume und die Regressionsbäume.

- **Klassifikationsbäume** zeigen eine Auswahl von diskreten Klassen und deren Beziehungen untereinander.
- **Regressionsbäume** dienen der Prognose eines stetigen Wertes der abhängigen Variable.

Entscheidungsbäume können entweder von Experten manuell aufgestellt werden oder sie werden mit Techniken des maschinellen Lernens automatisch aus gesammelten Daten erstellt. Hierzu gibt es unterschiedliche Algorithmen.

- **CHAIDs** (Chi-square Automatic Interaction Detectors) konstruieren Entscheidungsbäume anhand von diskreten Attributen. Für die Wahl der Attribute wird beim CHAID-Algorithmus der Chi-Quadrat-Unabhängigkeitstest verwendet. CHAIDs kommen zur Anwendung, wenn

eine Aussage über die Abhängigkeit zweier Variablen gemacht werden muss. Zur Begrenzung der Größe der Bäume kommen ‘Pruning’-Verfahren (zurechtstutzen) zum Einsatz. Die Erstellung der Bäume kann Bottom-Up oder Top-Down erfolgen.

- **CARTs** (Classification and Regression Trees) erzeugen Binärbaume, d. h. bei den Verzweigungen gibt es immer genau zwei Abzweigungen. Bei den CART-Entscheidungsbäumen sind die Attribute mit dem höchsten Informationsgehalt in Bezug auf die Zielgröße am weitesten oben im Baum zu finden. Die Entscheidungsschwellwerte ergeben sich jeweils durch die Optimierung der Entropie (Informationsgehalt) der Spalte.
- **ID3** wird verwendet, wenn bei großer Datenmenge viele verschiedene Attribute von Bedeutung sind und deshalb ein Entscheidungsbaum ohne große Berechnungen generiert werden soll. Somit entstehen meist einfache Entscheidungsbäume. Das Attribut mit dem höchsten Informationsgewinn bzw. der kleinsten Entropie wird gewählt und daraus ein neuer Baum-Knoten generiert. Das Verfahren endet, wenn alle Trainingsinstanzen klassifiziert wurden, d. h. wenn jedem Blattknoten eine Klassifikation zugeordnet ist.
- **C4.5 und C5.0** sind Nachfolger des ID3-Algorithmus. Die Algorithmen sind vergleichbar mit dem CART-Verfahren, es ist aber möglich, die Bäume in mehr als zwei Abzweigungen zu unterteilen. Dadurch werden die Bäume breiter (mehr Äste) und weniger tief (weniger Knoten).

Die Vorteile der Entscheidungsbäume liegen darin, dass sie in der Regel gut zu verstehen und zu interpretieren sind. Beispielsweise kann der Entscheidungsbaum zu einer Kreditentscheidung in Fließtext interpretiert werden: Einem nicht berufstätigen Studenten, der keine Bürgschaft seiner Eltern vorweisen kann, wird kein Kredit gewährt.

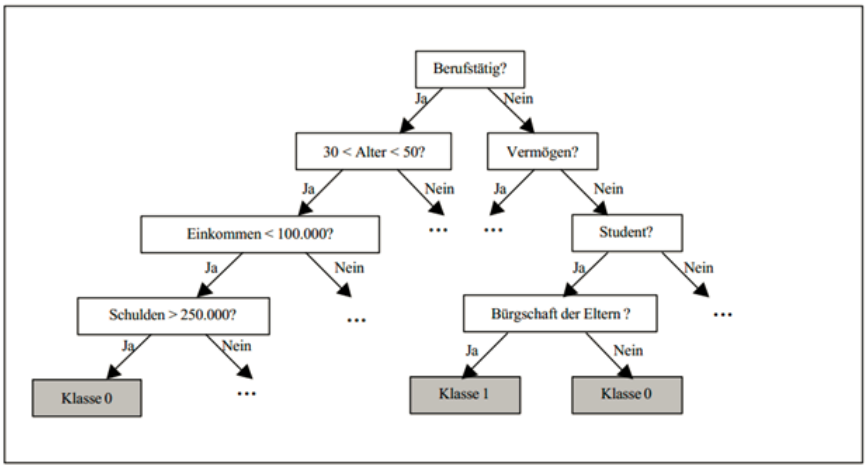


Abbildung 16: http://winf-wiki.fhslabs.ch/index.php?title=Data_Mining

Als Nachteil der Verfahren kann gesehen werden, dass Entscheidungsbäume häufig zu schlechteren Klassifikationsergebnissen führen als andere Verfahren (z. B. neuronale Netzwerke). Die Verständlichkeit der Regeln geht auf Kosten der Klassifikationsgüte. Außerdem besteht – je nach Verfahren und Art der Daten – die Gefahr, dass die Bäume zu groß und damit schwerer verständlich bzw. weniger aussagekräftig werden.

Eine Erweiterung der Entscheidungsbäume sind die Entscheidungswälder (Decision Forests). Dabei handelt es sich um den kombinierten Einsatz mehrerer Entscheidungsbäume. Über eine Kombination der Entscheidungsbäume anhand von Mehrheitsentscheidungen soll die Klassifikationsgüte erhöht werden. Im folgenden Abschnitt wird auf diese kombinierten Verfahren genauer eingegangen.

4.5.6 Ensemble-Methoden

Ensemble-Methoden			
Absicht / Zweck	**Lernen**	**Analys. Daten**	**Ergebnis**
Klassifikation	Überwachtes Lernen	Nominalskala	Nominalskala
Prognose / Vorhersage		Ordinalskala	Ordinalskala
Segmentierung	Unüberwachtes Lernen	Kardinalskala	Kardinalskala
Abhängigkeitsanalyse			
Abweichungsanalyse			

Bei den Ensemble-Methoden[6] handelt es sich um keine eigenständigen statistischen Verfahren, sondern um die Zusammenfassung mehrerer Verfahren. Ziel ist es, aus mehreren (schwachen) Verfahren ein (starkes) Verfahren zu kombinieren, das bessere Ergebnisse liefert als die einzelnen Prozesse alleine. Die Grundidee lässt sich z. B. anhand des Jokers bei *Wer wird Millionär?* verdeutlichen. Die Wahrscheinlichkeit einer richtigen Antwort wird dadurch erhöht, dass der Meinung eines Einzelnen (des Kandidaten) die Meinung von vielen (Publikumsjoker) hinzufügt wird, wobei eventuell weitere Joker genutzt werden.

Grundsätzlich lassen sich Ensemble-Verfahren für alle Fragestellungen einsetzen, aber meistens werden sie für überwachtes Lernen (Klassifikationen/Vorhersagen) verwendet. Je nachdem, ob gleiche oder unterschiedliche Verfahren kombiniert werden, ist von homogenen oder heterogenen Verfahren die Rede.

Die Ursache von falschen Ergebnissen von Modellen liegt im Bereich Noise (Rauschen), Variance und Bias (Verzerrung). Oder die Modelle funktionieren

[6] Ein sehr guter Überblick auf: https://www.datavedas.com/ensemble-methods/

zwar perfekt bei den Trainingsdaten, sind aber nicht für die Test- bzw. Realitätsdaten geeignet (Overfitting des Modells).

Sowohl in Theorie als auch Praxis hat sich gezeigt, dass sich mit Ensemble-Methoden die Ergebnisse deutlich verbessern lassen, wenn auch nicht immer für alle Parameter gleichzeitig. Viele Gewinner von Kaggle-Wettbewerben haben Ensemble-Methoden – dabei vor allem XGBoost – eingesetzt.

Die wichtigsten Ensemble-Methoden sind Bagging, Boosting und Stacking.

Bagging	Methode, bei der mehrere Teilmengen von Daten aus dem Trainingssatz nach dem Zufallsprinzip ausgewählt und gleichgewichtet bewertet werden.
Boosting	Eine Variante des Bagging, bei der die Ausgabe des vorherigen Modells das nächste Modell beeinflusst. Sequentielles lernen des Modells
Stacking	Die Ergebnisse verschiedener Modelle warden auf einer Meta-Ebene durch ein weiteres Modell kombiniert.
Bayesian model combination	Verfahren der Gewichtung der Einzelmodelle
...	

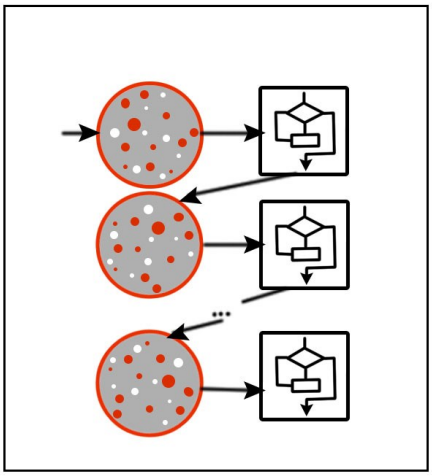
Bagging (Bootstrap aggregating): Beim Bagging werden die Trainingsdaten zufällig in eine größere Anzahl von Teilmengen an Trainingsdaten zerlegt, wobei *Zurücklegen* vorgesehen ist, sodass also Datensätze in mehreren Teilgruppen der Trainingsdaten vorkommen können.

Bag 1				Bag 2				Bag 3				Bag 4				Bag 5			
X1	X2	X3	X4	X1	X2	X3	X4	X1	X2	X3	X4	X1	X2	X3	X4	X1	X2	X3	X4
245	26	416	5	564	26	131	55	453	67	357	15	453	67	357	15	189	47	251	45
453	67	357	15	505	39	226	50	803	41	188	20	453	67	357	15	505	39	226	50
245	26	416	5	564	26	131	55	803	41	188	20	245	26	416	5	189	47	251	45

Abbildung 17: vgl. https://www.datavedas.com/bagging/

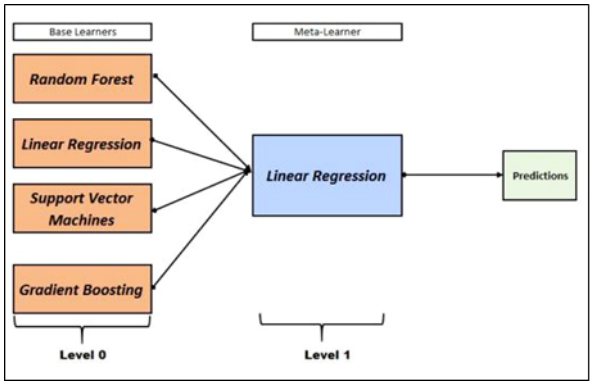
Für jede Teilmenge wird nun ein Modell trainiert und die Ergebnisse der Einzelmodelle werden gleichgewichtet zusammengefasst. Dies erfolgt bei Klassifikationen per Mehrheitsentscheidung und bei Regressionen z. B. durch das arithmetische Mittel.

Random Forests sind eine Variante des Bagging-Algorithmus, der speziell für Entscheidungsbäume entwickelt wurde. Es handelt sich um eine Kombination aus Bootstrapping und zufälligem Teilen der Daten.



Boosting ist eine Variante des Baggings, bei der die Ausgabe des vorherigen Modells das nächste Modell beeinflusst. Hierbei wird die Funktion sequentiell gelernt. Es gibt verschiedene Arten von Boosting, z. B. Adaptive Boosting, Gradient Boosting und Extra Gradient Boosting (XGBoost). Bei Adaptive Boosting wird eine Reihe von schwachen Lernern verwendet, bei denen die Funktion nacheinander durch Zuweisen von Gewichten zu den falsch vorhergesagten Werten gelernt wird. Gradient Boosting ist eine Erweiterung des Adaptive Boosting, Extra Gradient eine weiterentwickelte Variante von Gradient Boosting.

Stacking: Beim Stacking werden die Ergebnisse verschiedener Modelle auf einer Meta-Ebene durch ein weiteres Modell kombiniert und zu einem Kombinationsergebnis zusammengefasst. Durch Hinzufügen weiterer Ebenen und Variation der Hyperparameter der einzelnen Modelle können bei den Stacking-Modellen die Komplexität und damit verbunden der Rechenaufwand beliebig erhöht werden.



Unter der Voraussetzung, dass die Trainingsdatenmenge groß genug ist, führen Ensemble-Methoden eigentlich immer zu einer Verbesserung der Modellqualität, ohne dabei ein Overfitting einzugehen. Der wahrscheinlich größte Nachteil der Verfahren ist der Verlust der Erklärbarkeit bzw. Nachvollziehbarkeit des Modells. Ein einzelner Entscheidungsbaum kann als Regelsatz im Fließtext interpretiert werden, ein Ensemble aus Entscheidungsbäumen (oder auch aus anderen Modellen) wird zur Black Box.

4.5.7 Neuronale Netze

Künstliche neuronale Netze (KNN):

Absicht / Zweck	Lernen	Analys. Daten	Ergebnis
Klassifikation	Überwachtes Lernen	Nominalskala	Nominalskala
Prognose / Vorhersage		Ordinalskala	Ordinalskala
Segmentierung	Unüberwachtes Lernen	Kardinalskala	Kardinalskala
Abhängigkeitsanalyse			
Abweichungsanalyse			

Jetzt erreichen wir den heiligen Gral der Verfahren. Für Data-Science-Verhältnisse wird es fast schon esoterisch, denn das Thema neuronale Netze umgibt eine geheimnisvolle Aura der Macht, mit dem tatsächlich eine künstliche Intelligenz erschafft wird. Dabei sind neuronale Netze einerseits viel banaler als es ihnen unterstellt wird. Es handelt sich lediglich um eine Kombination aus mathematischer Matrizenrechnung und einem Ablaufschema (Lernalgorithmus), mit dem iterativ Verbesserungen ermöglicht werden sollen.

Ich wiederhole: Ein bisschen Matrizenrechnung und einige Vorgehensregeln, wie man die Matrizenrechnung anwendet. Das war's. Keine 'Rocket Science' und keine Frankenstein'sche Alchemie.

Auf der anderen Seite werden neuronale Netze aber tatsächlich im Bereich der künstlichen Intelligenz eingesetzt und ermöglichen – gerade in den letzten Jahren – Entwicklungen, die zu Erstaunen geführt haben.

Aus diesem Grund wird im folgenden Abschnitt etwas ausführlicher auf das Thema eingegangen, sodass die Möglichkeiten, aber auch die Grenzen des Verfahrens, richtig eingeordnet werden können.

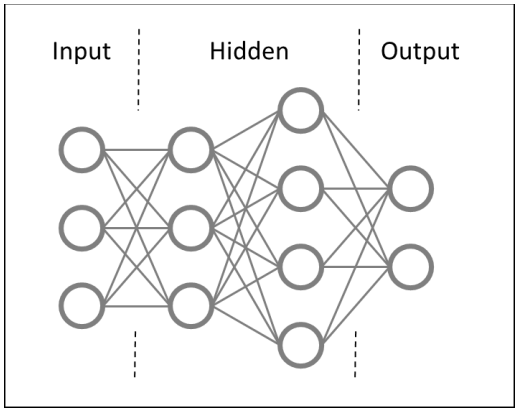
Mit künstlichen neuronalen Netzen (KNN) wird versucht, mit Computern Mechanismen nachzubilden, wie sie im menschlichen Gehirn vorkommen. Ziel ist es, dem Computer das Lernen beizubringen. Der Computer wird nicht programmiert, und folgt dann dem Programmablauf, sondern er soll selbständig lernen.

Das menschliche Gehirn besteht aus ca. 80 bis 100 Milliarden Nervenzellen (Neuronen), die über sogenannte Synapsen verbunden sind und damit ein riesiges Netzwerk bilden. In einem KNN wird versucht, die Grundidee des Gehirns ‘im Kleinen’ nachzubilden. Eine Definition des künstlichen neuronalen Netzes lautet: [7]

> *„Ein Neuronales Netz ist ein sortiertes Tripel (N, V, w) mit zwei Mengen N, V sowie einer Funktion w, wobei N die Menge der Neurone bezeichnet und V eine Menge {(i, j)|i, j ∈ N} ist, deren Elemente Verbindungen von Neuron i zu Neuron j heißen. Die Funktion w : V → R definiert die Gewichte, wobei w((i, j)), das Gewicht der Verbindung von Neuron i zu Neuron j, kurz mit wi,j bezeichnet wird.“*

Alles klar? Sollte es doch noch nicht ganz klar sein, dann tasten wir uns an das Thema langsam ran. Ein KNN besteht aus:

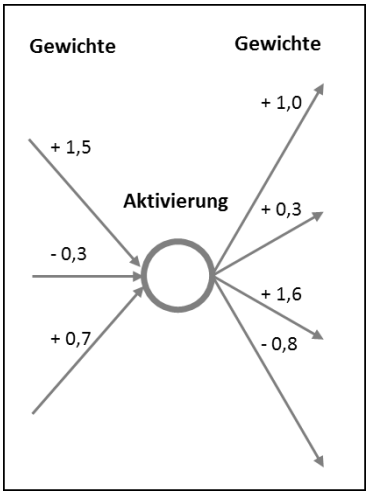
[7] Kriesel, S. 36



- **Einheiten** (Units). Die Einheiten lassen sich in verschiedene Ebenen (**Layers**) zusammenfassen. Neben den obligatorischen Input und Output Layers gibt es oft noch einen oder mehrere Hidden Layers.
- Verbindungen (**Kanten**) der Einheiten. Die Stärke der Verbindung wird durch ein **Gewicht** ausgedrückt.
- Die Einheiten verfügen über einen Wert der **Aktivierung**, der beschreibt, ob und wie aktiv eine Einheit ist. Das ist sozusagen der Schaltzustand der Einheit. Der Aktivierungsgrad hat Einfluss darauf, wie eine Einheit auf den Input der anderen Einheiten reagiert und bestimmt den Output der Einheit.
- Die Struktur des Netzwerkes wird als **Topologie** bezeichnet.
- Die **Aktivierungsfunktion** beschreibt den funktionalen Zusammenhang zwischen dem Input und dem Output einer Einheit.
- Der gesamte Input einer Unit wird **Netzinput** genannt. Dieser wird über die sog. **Propagierungsfunktion bzw. Übertragungsfunktion** bestimmt. Die verbreitetste Übertragungsfunktion ist eine Linearkombination, bei der sich der Netzinput additiv aus sämtlichen gewichteten Inputs zusammensetzt, die das Neuron von anderen Neuronen erhält.

- Das **Wissen** des Netzwerks besteht aus der Gesamtheit der Gewichte der einzelnen Kanten. Als Ausgangswerte werden in der Regel Zufallswerte angenommen.
- **Lernen** erfolgt durch iteratives Anpassen der Gewichtungen nach unterschiedlichen Lernalgorithmen.

Schauen wir uns dies zur Erläuterung einmal an einem Beispiel für eine beliebige Einheit an:



Ausgangspunkt sind gegebene Werte für die Gewichte der Kanten. Der Wert für die Aktivierung und den Output der Einheit in Abhängigkeit vom Input in das Netzwerk ergibt sich über folgenden Zusammenhang:

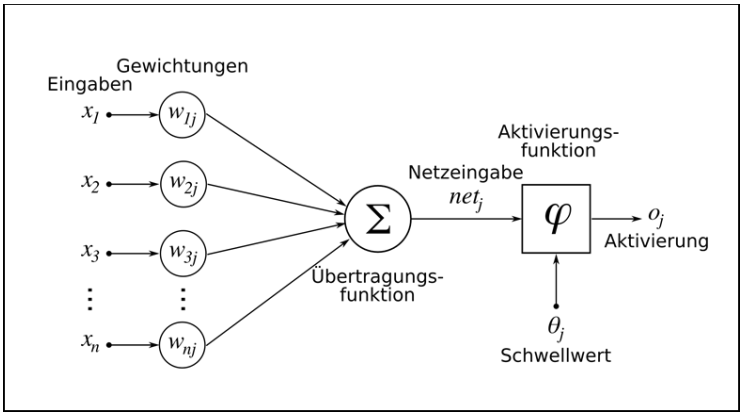
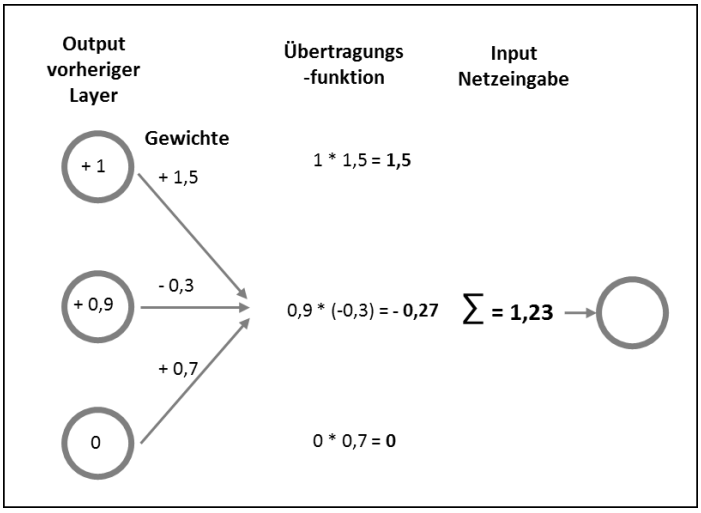


Abbildung 18: https://de.wikipedia.org/wiki/Künstliches_neuronales_Netz

Der Input x_1 bis x_n entspricht dem Output der entsprechenden vorangegangen Knoten. Dieser wird mit den Kantengewichten multipliziert und summiert (die Übertragungsfunktion ist also annahmegemäß einfach die Summe der gewichteten Eingaben x_1 bis x_n).

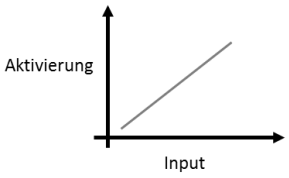
Im Beispiel sähe das so aus:



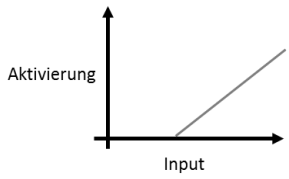
Die Werte für den Output der Einheiten des vorherigen Layers sind im Beispiel willkürlich gewählt und sollen an dieser Stelle nicht interessieren. Die Übertragungsfunktion summiert hier die gewichteten Outputs und so erhalten wir den Input (die Netzeingabe) für die Einheit in Höhe von 1,23.

Aus diesem Input wird über die **Aktivierungsfunktion** der Aktivierungswert berechnet. Es handelt sich um eine funktionale Zuordnung des Inputs mit der Aktivierung der Zelle. Die Aktivierungsfunktion kann unterschiedliche Ausprägungen einnehmen:

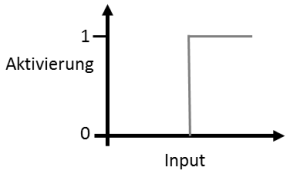
- Lineare Funktion



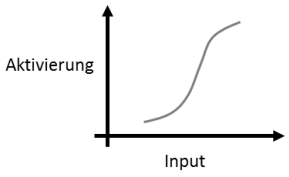
- Lineare Funktion mit Schwellenwert



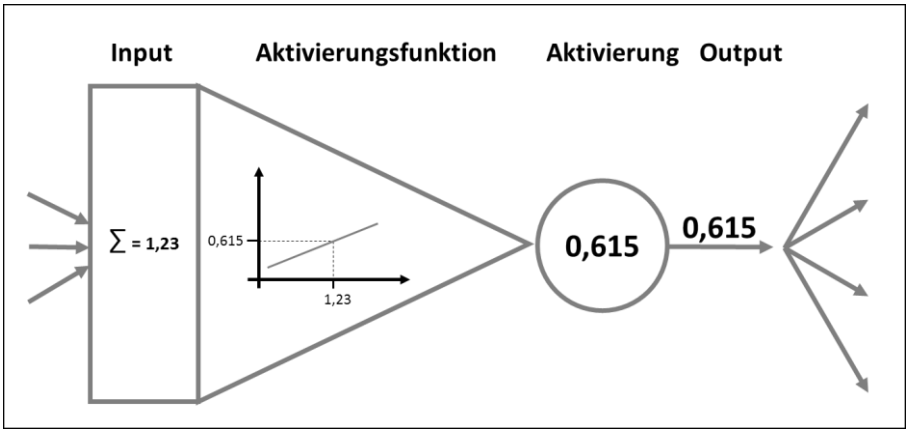
- Binäre Funktion



- Sigmoide (z. B. logistische) Funktion



Nehmen wir für unser Beispiel an, dass die Aktivierungsfunktion eine lineare Funktion ohne Schwellenwert mit der ‘Steigung’ 0,5 ist und die Aktivierung dem Output entspricht: Aktivierung = 0,5 * Input = Output = 0,615



Es kann aber auch festgelegt werden, dass die Aktivierung nicht dem Output entspricht, sondern z. B. den Faktor darstellt, mit dem die Aktivierung aus dem vorangegangenen ‘Durchgang’ multipliziert wird. Alternativ könnte auch der Mittelwert aus den beiden Werten ermittelt werden.

Damit wurde die Berechnung einer Einheit erläutert. In der Regel geht man dann von ‘links nach rechts’ vor, d. h. ausgehend vom Input in die Input-Elemente werden die Aktivierungen der einzelnen Elemente in den Hidden Layers und zum Schluss im Output Layer berechnet.

Aus diesem Grundprinzip des KNN ergeben sich nun mehrere Fragen bzw. Handlungsoptionen:

- Soll das KNN für ein **überwachtes** oder **unüberwachtes Lernen** eingesetzt werden, oder anders ausgedrückt: Liegen gelabelte Lerndaten vor (also Trainingsdaten, für die Input- und Output-Werte vorliegen) oder soll mit dem KNN eine Datenmenge mit nicht vorhandenen Output-Werten zur Erkennung von Mustern verwendet werden? Im ersten

Fall müssen über einen **Lernalgorithmus** die Gewichtungen der Kanten solange angepasst werden, bis das Netz die Trainingsdaten hinreichend gut wiedergibt. Im zweiten Fall muss der Lernalgorithmus die Gewichtungen im Netz solange anpassen, bis das Lernziel (z. B. die Mustererkennung oder die Clusterbildung) hinreichend gut erfüllt ist. Es müssen also die für das Ziel des Lernens geeigneten Lernregeln ausgewählt werden. Auf die Lernregeln wird weiter unten eingegangen.

- Welche **Topologie** soll mein KNN haben? Wie viele Knoten und wie viele Hidden Layers soll das Netz haben? Neuronale Netze ohne Hidden Layers werden als **kompetitive Netze** bezeichnet.
- Wie ist die **Richtung** der Verbindungen? Die Verbindungen können ausschließlich in eine Richtung (von 'links nach rechts') und immer nur zwischen zwei benachbarten Layers vorkommen (**Feedforward Netze**), oder aber es sind **Rückkopplungen** in die andere Richtung (indirekte Rückkopplung), im selben Layer (seitliche Rückkopplung), oder sogar mit derselben Unit (direkte Rückkopplung) möglich.
- Welche **Aktivierungsfunktion** soll ausgewählt werden? Soll die 'alte' Aktivierung einer Unit verwendet werden oder soll der Output der Unit der Aktivierung entsprechen?

Aus dieser Aufstellung wird deutlich, dass die Schwierigkeit der Anwendung neuronaler Netze auch in der schieren Anzahl an Ausprägungen und 'Einstellungen' liegt, die möglich sind. Die Auswahl der richtigen Netztopologie, der richtigen Aktivierungsfunktionen und des richtigen Lernalgorithmus lässt sich nicht allgemeingültig bewerkstelligen. Darüber hinaus gibt es zahlreiche weitere 'Stellregler', die angepasst werden können.

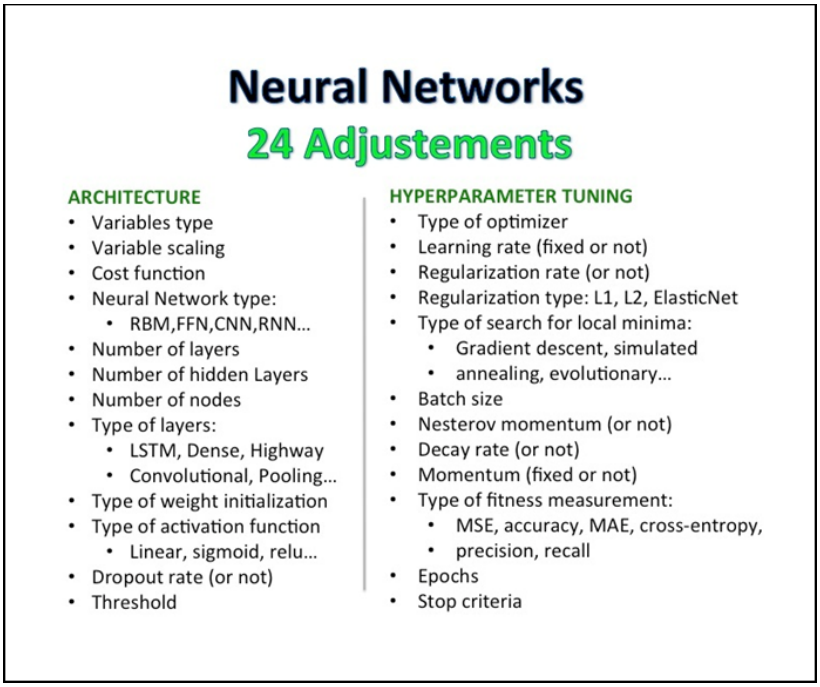


Abbildung 19: http://www.datasciencecentral.com/profiles/blogs/24-neural-network-adjustements

Es liegt dann in der Erfahrung, aber auch in der Bereitschaft des Anwenders, mit dieser Unsicherheit umzugehen.

In den folgenden zwei Abschnitten sollen zuerst unterschiedliche Lernalgorithmen kurz vorgestellt und danach das Thema Netztopologie beleuchtet werden.

Lernalgorithmen – Lernregeln

Das Wissen des neuronalen Netzes liegt in den Gewichtungen der Kanten. Lernen bedeutet also, solange die Gewichte anzupassen, bis das Ergebnis ‘passt’. Um die Gewichte in der Trainingsphase zu modifizieren, benötigt man

folglich eine Lernregel, die angibt, wie die Veränderungen vorgenommen werden sollen. Eine Lernregel stellt dabei einen Algorithmus dar, der darüber Auskunft gibt, welche Gewichte des neuronalen Netzes wie stark erhöht oder reduziert werden sollen.

Es gibt unterschiedliche Lernregeln, in Abhängigkeit davon, ob es sich um ein KNN mit **Hidden Layer** handelt oder nicht.

Außerdem können Lernalgorithmen in die beiden Klassen der überwachten und der nicht überwachten Verfahren eingeteilt werden. Beim **überwachten Lernen** wird die vom Netz erzeugte Ausgabe betrachtet und ihre Abweichung von der gewünschten Ausgabe gemessen. Danach werden die Gewichte entsprechend der Größe der Abweichung angepasst. Eine Untergruppierung des überwachten Lernens ist das **bestärkende Lernen** (Reinforced Learning). Hierbei liegen keine Informationen über die Höhe der Abweichung vom gewünschten Wert vor, sondern es ist lediglich die Abweichung bekannt.

Das **unüberwachte Lernen** setzt keinen Lehrer voraus, d. h. für die Eingabewerte der Trainingsdaten ist die richtige Ausgabe nicht bekannt. Das Netz muss sich selber organisieren, d. h. die Gewichtungen anpassen, um Muster zu erkennen und z. B. Gruppierungen (Cluster) der Datensätze vorzunehmen.

In der folgenden Tabelle sind wichtige Lernregeln für die unterschiedlichen Netzarten aufgeführt.

	Zweistufiges Netz	Mehrstufiges Netz (Hidden Layer)
Überwachtes Lernen	- Delta-Regel	- Gradientenverfahren - Backpropagation - BFGS - Levenberg-Marquardt - CG-Verfahren
Bestärkendes Lernen	- Temporal Difference Learning - SARSA	
Unüberwachtes Lernen	- Adaptive Resonanztheorie - Hebb'sche Lernregel	- Competitive Learning - Kohonen-Netze

Netztopologie

Es gibt unzählige Ausprägungen an Netzwerken. Ein wichtiges Merkmal ist die Frage, ob und wie viele Hidden Layers vorhanden sind. Bei mehr als einem Hidden Layer spricht man häufig von **Deep Learning,** ein Begriff, der zwar meiner Meinung nach einen falschen Eindruck vermittelt, aber sich einfach durchgesetzt hat.

Eine sehr umfassende Übersicht über verschiedene Arten von neuronalen Netzen hat Fjodor van Veen vorgenommen.

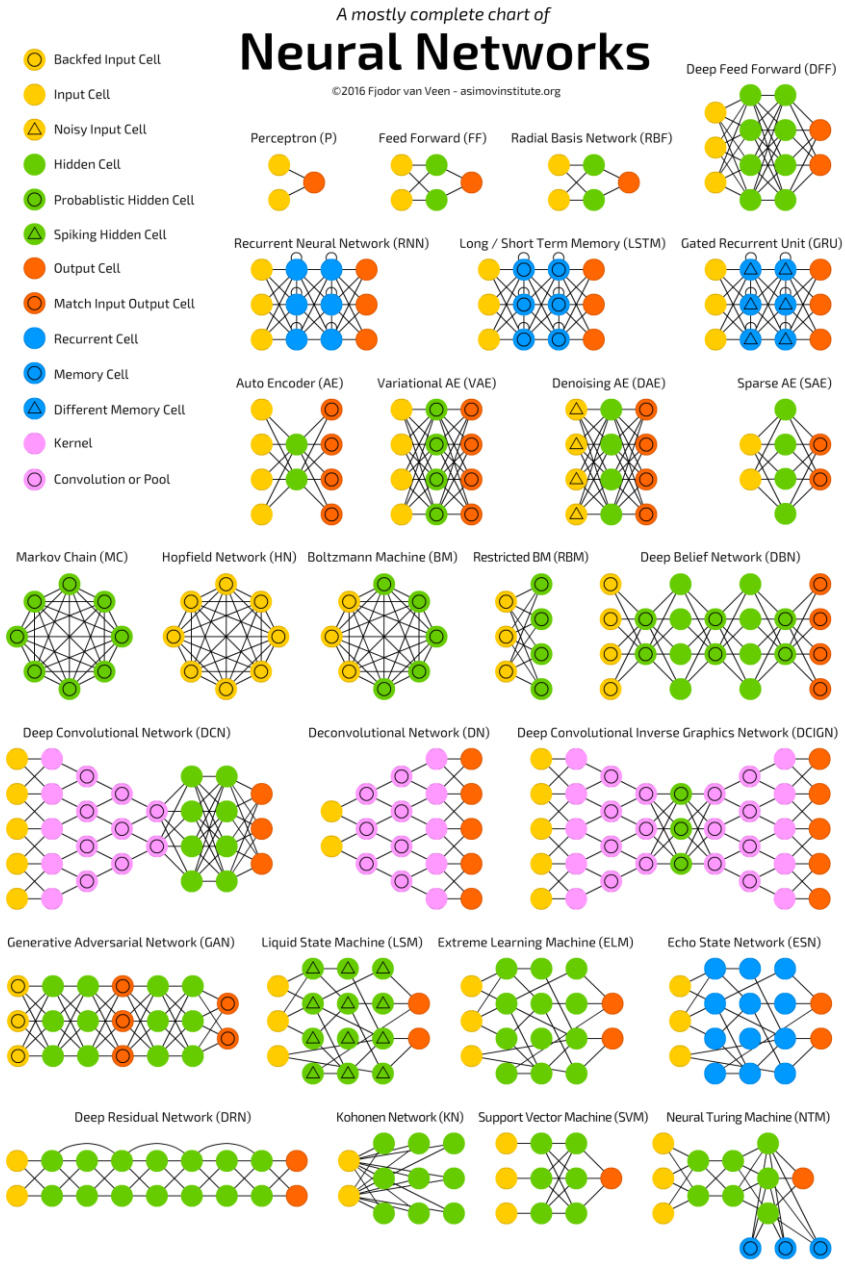


Abbildung 20: http://www.asimovinstitute.org/wp-content/uploads/2016/09/neuralnetworks.png

Neuronale Netze sind in ihrem Einsatzgebiet und in ihrer Ausgestaltung sehr flexibel. Sie werden für Klassifizierungen, Prognosen und Clustering-Aufgaben angewendet. Insbesondere im Bereich der Bild-, Schrift- und Spracherkennung werden sie erfolgreich eingesetzt.

Bei der Bilderkennung hat sich der Einsatz von **Convolutional Neural Networks** bewährt. Diese *gefalteten* Netzwerke haben die Besonderheit, dass die Eingabeknoten aus zwei- bzw. dreidimensionalen Matrizen bestehen können, was sich zur Abbildung von 2-D- bzw. 3-D Bildern anbietet.

Insgesamt kann gesagt werden, dass sich neuronale Netze als Machine Learning-Verfahren für vielfältige Einsatzgebiete eignen. Die Rechenleistung moderner Computer und der Einsatz von Graphikprozessoren, die aufgrund ihrer sehr hohen Anzahl von *Cores* Rechenprozesse massiv parallelisieren können, ermöglichen heute Anwendungen, die vor ein paar Jahren noch an mangelnder Computerleistung gescheitert wären. Dennoch sind sie kein Allheilmittel, das ohne eigenen Verstand eingesetzt werden kann. Die Güte der Modelle unterliegt in vielen Fällen immer noch klassischeren Ensemble-Methoden (allen voran XGBoost), wie an vielen Kaggle-Wettbewerben gesehen werden kann.

Es wird außerdem als kritisch gesehen, dass neuronale Netze eine Black Box darstellen, deren Ergebnisse manchmal nur schwer zu erklären sind. Die Gestaltung eines Netzwerkes ist oft eher willkürlich und es besteht die Gefahr der Überanpassung (Overfitting) des Netzes an die Trainingsdaten. Das Netz liefert dann perfekte Ergebnisse für die Trainingsdaten, stellt sich aber als ungeeignet für *neue* Daten heraus. Je komplizierter (tiefer) die Struktur der Netze ist, umso rechenaufwendiger werden die Verfahren.

Dank der gestiegenen Rechenleistung aktueller Computer und immer neuer Deep-Learning-Bibliotheken erfahren neuronale Netze aber derzeit geradezu eine Renaissance in ihrer Anwendung. Die Ergebnisse in vielen Einsatzgebieten sind sehr erfolgsversprechend.

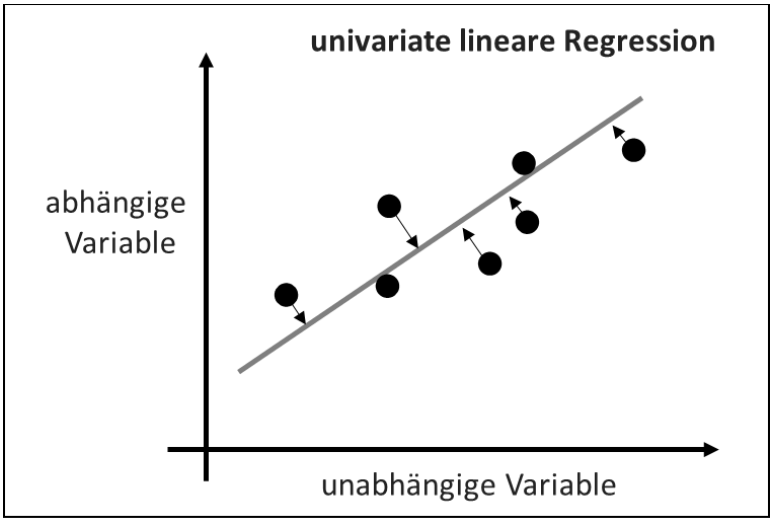
Bei aller berechtigten Euphorie sollte die Erwartungshaltung aber nicht zu hoch sein. So darf nicht vergessen, werden, dass ein menschliches Gehirn aus 80 bis 100 Milliarden Neuronen besteht. Davon sind künstliche neuronale Netze noch weit entfernt.

4.5.8 Regression

<table>
<tr><td colspan="4">Regressionsanalyse: Lineare Regression (univariat, multivariat), logistische Regression, nichtparametrische Regression</td></tr>
<tr><th>Absicht / Zweck</th><th>Lernen</th><th>Analys. Daten</th><th>Ergebnis</th></tr>
<tr><td>Klassifikation</td><td rowspan="2">Überwachtes Lernen</td><td>Nominalskala</td><td>Nominalskala</td></tr>
<tr><td>Prognose / Vorhersage</td><td>Ordinalskala</td><td>Ordinalskala</td></tr>
<tr><td>Segmentierung</td><td rowspan="3">Unüberwachtes Lernen</td><td>Kardinalskala</td><td>Kardinalskala</td></tr>
<tr><td>Abhängigkeitsanalyse</td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>Abweichungsanalyse</td></tr>
</table>

Die Regression kann als die *Mutter aller Verfahren* bezeichnet werden. Das liegt einerseits daran, dass sie in ihrer einfacheren Ausprägung als lineare Regression gut verständlich und damit nachvollziehbar ist, aber auch daran, dass die im Verfahren eingesetzten Methoden in vielen anderen Verfahren eine analoge Anwendung finden. Als Prognoseverfahren wird sie in vielen Anwendungsfällen eingesetzt und ist sicher das am häufigsten verwendete Schätzmodell in der Praxis.

Der einfachste Fall ist die lineare univariate Regression, also mit jeweils einer metrischen abhängigen und unabhängigen Variablen. Es wird ein linearer Zusammenhang zwischen den Variablen vermutet.



Als Beispiel kann man die Abhängigkeit des Umsatzes eines Produktes von den Ausgaben für Werbung anführen. Die Punkte stellen Ergebnisse aus verschiedenen Zeitpunkten dar. Die Y-Achse (abhängige Variable) ist der Umsatz, auf der X-Achse (unabhängige Variable) ist die Höhe der Werbeausgaben abgetragen.

Der Zusammenhang der beiden Variablen kann mit einer linearen Funktion dargestellt werden (auf die statistischen Annahmen wie z. B. einer zugrunde gelegten Normalverteilung der Zufallskomponente soll hier nicht eingegangen werden):

$$y = \beta_0 + \beta_1 x_1$$

Man versucht nun, die Regressionsgerade so in die Punktewolke einzupassen, dass die Abstände der Punkte von der Gerade minimiert werden (wie so häufig wird hierbei meist das Quadrat der Abstände genommen).

Wenn man mit den statistischen Gütezahlen zufrieden ist (also z. B. den normalisierten quadrierten Abweichungsdistanzen), erhält man dann ein gut verständliches Modell, das für die Prognose verwendet werden kann.

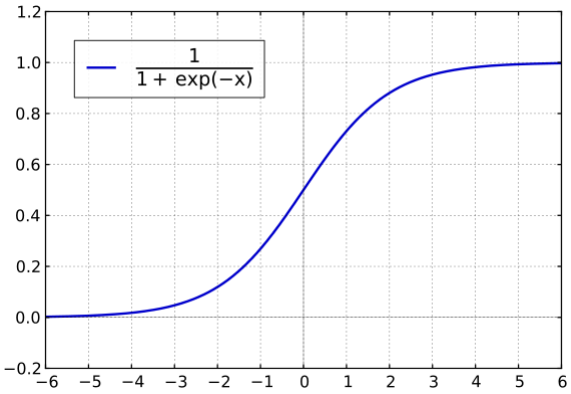
$$\text{Umsatz} = 5.274 + 3{,}76\ x$$

Neben dem ‘Grundrauschen’ von etwa 5.300 € bringt jeder in Werbung investierte Euro ein Umsatzplus von 3,76 €.

An dem Beispiel wird schnell deutlich, dass sich die Realität selten mit einer so einfachen univariaten Gleichung abbilden lässt. Man wird daher:

- Weitere erklärende Variablen miteinbeziehen wollen. Dadurch bekommt man eine nicht mehr so einfach zu visualisierende multivariate Regression.
  $$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$
- Das Schätzverfahren ändern. Durch das Quadrieren der Abstände haben einzelne Ausreißer eine relativ große Auswirkung auf die Regressionsfunktion, die man eventuell begrenzen möchte (z. B. ‘M-Schätzer’).
- Keinen linearen Zusammenhang annehmen, sondern versuchen, die Art des funktionalen Zusammenhangs anhand der Daten herzuleiten.

Eine häufig verwendete Form der Regression ist die **logistische Regression** oder Logit-Modell. Darunter versteht man eine Klassifizierungsverfahren zur Modellierung der Verteilung diskreter abhängiger Variablen, z. B. im Falle von Ja-Nein-Entscheidungen (z. B. ob der Kunde kreditwürdig ist oder nicht).



Man könnte derartige Entscheidungsmodelle auch mit einer linearen Regression lösen, wobei man dann einen Wert für y als Grenze zwischen ‘Ja’ und ‘Nein’ definiert. Das führt dazu, dass Werte knapp an diesem Wert relativ willkürlich in die eine oder andere Klasse zugeordnet würden. Die logistische Verteilfunktion mit ihrer hohen Steigung im ‘Mittelteil’ der Kurve teilt sozusagen das ‘Ja’ besser von dem ‘Nein’, da der Bereich in der Nähe des Cut-offs sehr steil ist und damit nur wenige Werte ‘betrifft’.

Eine Verallgemeinerung der logistischen Regression ist die **Soft Max Regression / Multi-class Logistic Regression.** Hier wird von mehreren Klassen ausgegangen, wobei die Wahrscheinlichkeit errechnet wird, dass ein Wert zu einer bestimmten Klasse gehört. Die Summe der Wahrscheinlichkeiten ist 1.

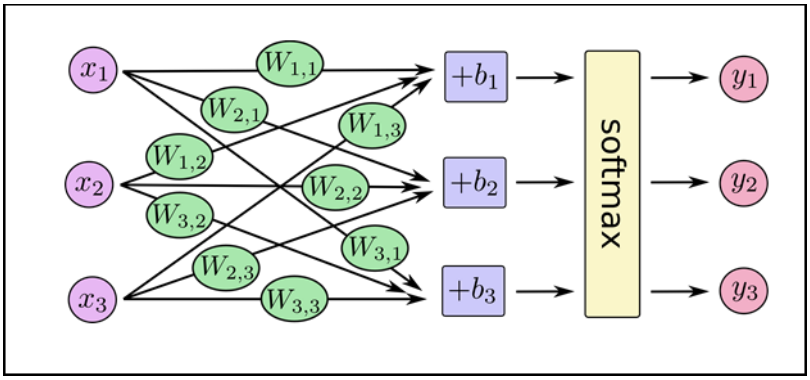


Abbildung 21: Quelle: https://www.tensorflow.org/get_started/mnist/beginners

Dieses Verfahren wird z. B. in der Handschrifterkennung angewendet. Der Input (x_1, x_2 und x_3) wären dann im Beispiel die Bildpunkte eines Sensors und die Klassen (y_1, y_2 und y_3) die Ziffern. Die Soft-Max-Regressionsfunktion gibt dann die Wahrscheinlichkeit an, ob ein Bild eine bestimmte Ziffer darstellt. Die Summe der Wahrscheinlichkeiten ist 1. Das Verfahren weist eine Verwandtschaft zu einfachen neuronalen Netzen auf.

Nichtlineare Regression

Durch die nichtlineare Regression wird der Anwendungsbereich der Regressionsanalyse erweitert. Es lassen sich nahezu beliebige Modellstrukturen bilden. Eine Schätzung der Regressionskoeffizienten ist nur iterativ möglich, wodurch sich nicht nur der Rechenaufwand im Vergleich zur linearen Regression erhöht. Ein deutlicher Nachteil der nichtlinearen Regression ist es auch, dass keine statistischen Tests zur Prüfung der Güte eines Modells oder der Signifikanz der Parameter zur Verfügung stehen.[8]

[8] Vgl. Backhaus (2015), S. 24f

4.5.9 Zeitreihenanalyse

Zeitreihenanalyse:			
Absicht / Zweck	**Lernen**	**Analys. Daten**	**Ergebnis**
Klassifikation	Überwachtes Lernen	Nominalskala	Nominalskala
Prognose / Vorhersage		Ordinalskala	Ordinalskala
Segmentierung	Unüberwachtes Lernen	Kardinalskala	Kardinalskala
Abhängigkeitsanalyse			
Abweichungsanalyse			

Eine Zeitreihe ist eine zeitabhängige Folge von Datenpunkten. Die Zeitreihenanalyse beschäftigt sich mit der statistischen Analyse von Zeitreihen und der Prognose ihrer künftigen Entwicklung. Sie ist eine Spezialform der Regressionsanalyse.

Die Vorgehensweise im Rahmen der Zeitreihenanalyse lässt sich in folgende Phasen einteilen:

- **Identifikation**: Identifikation eines geeigneten Modells für die Modellierung der Zeitreihe.
- **Schätzung**: Schätzung von geeigneten Parametern für das gewählte Modell.
- **Diagnose**: Diagnose und Evaluierung des geschätzten Modells.
- **Prognose**: Einsatz des Modells zu Prognosezwecken.

Die Zeitreihe ähnelt der Regression, da versucht wird, eine gegebenen Datenmenge so durch eine Gerade (oder eine andere Funktion) zu vereinfachen, dass damit die zukünftige Entwicklung vorausgesagt werden kann. Im Gegensatz zur Regression geht man aber davon aus, dass die Abweichungen von der Geraden (der Regressionsfunktion) nicht nur auf Modellfehler und Zufallsabweichungen zurückzuführen sind, sondern dass es zusätzlich z. B. saisonale

Schwankungen gibt. Das Zeitreihenmodell versucht dann, diese Saisonalität zu beachten und aus dem Trend herauszurechnen.

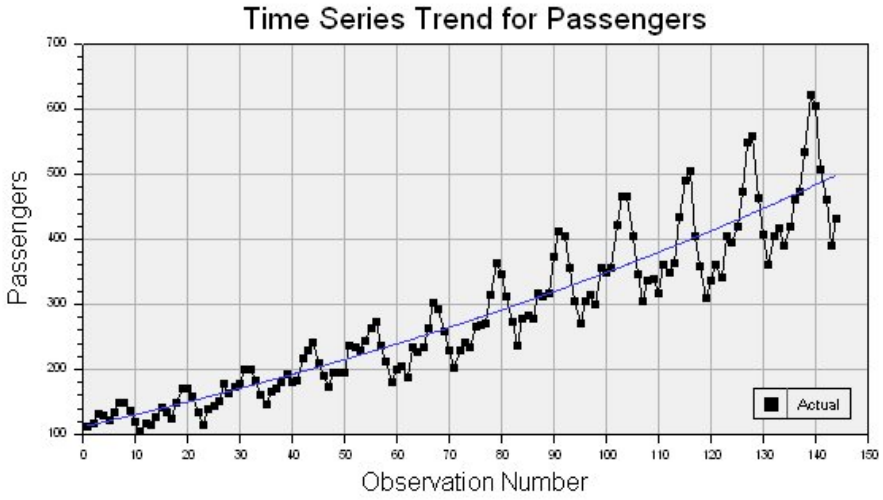


Abbildung 22: Quelle: https://www.dtreg.com/uploaded/pageimg/TsTrend_1.jpg

Bei der Erstellung des Modells sollte man daher Wissen bzw. eine Vermutung haben bzgl.:

- Saisonalität (wie lange ist eine Saison?)
- Trend (handelt es sich beispielsweise um einen linearen Trend, um eine logistische Wachstumskurve oder gar um einen 'Wachstumsbuckel' mit anschließendem Rückgang?)

Die Werte, die vom Trend und der Saisonalität abweichen, sind dann der Restwert, der die statistische Streuung bzw. Modellungenauigkeit darstellt.

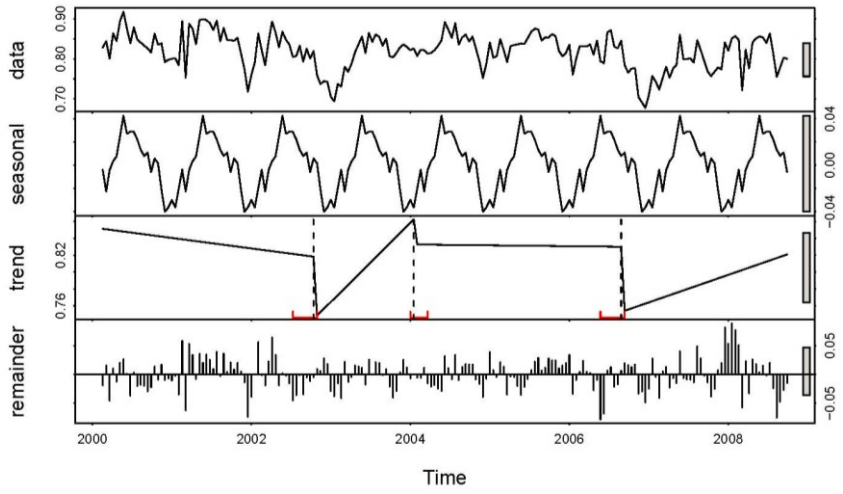


Abbildung 23: Quelle: http://bfast.r-forge.r-project.org/seasonalbreak_TreeMort.jpg

Mit dem so erstellten Zeitreihenmodell können dann Prognosen abgegeben und Abweichungen der tatsächlich eintretenden Werte im Zeitverlauf mit der Prognose erkannt werden.

4.5.10 Kollaboratives Filtern

Kollaboratives Filtern			
Absicht / Zweck	**Lernen**	**Analys. Daten**	**Ergebnis**
Klassifikation	Überwachtes Lernen	Nominalskala	Nominalskala
Prognose / Vorhersage		Ordinalskala	Ordinalskala
Segmentierung	Unüberwachtes Lernen	Kardinalskala	Kardinalskala
Abhängigkeitsanalyse			*Empfehlung*
Abweichungsanalyse			

Das kollaborative Filtern (Collaborative Filtering) ist ein Verfahren, das oft von Empfehlungsdiensten (Recommendation Engines) angewendet wird. Online-Shops empfehlen z. B. einem Kunden ein weiteres passendes Produkt, oder Streaming-Portale empfehlen einen Film oder einen Musiktitel. Dabei werden Verhaltensmuster von Benutzern ausgewertet, um auf die Interessen Einzelner zu schließen und damit eine Vorhersage von Interessen für andere Benutzer zu ermöglichen.

Dabei gibt es grundsätzlich zwei Betrachtungsweisen:

- **Personenbezogen**: Anhand der Aktionen vergleichbarer Personen werden Empfehlungen gegeben (‘Kunden die xy kaufen, kaufen auch yz’). Man schaut dabei nach Personengruppen, die ein vergleichbares Bewertungs- bzw. Kaufmuster wie der aktive Benutzer haben, um dann anhand der Ratings bzw. Käufe dieser Benutzergruppe eine Empfehlung zu geben und damit implizit eine Prognose über das Verhalten des aktiven Benutzers zu machen.

- **Objektbezogen**: Ausgehend vom Produkt wird ein ähnlich bewertetes Produkt empfohlen. Wird in einem Videostreamingdienst ein Film positiv bewertet, so wird ein vergleichbarer Film empfohlen (‘Sie haben sich xy anschaut, deshalb empfehlen wir yz’). Dazu wird eine Objekt-Objekt-Matrix erstellt, um Zusammenhänge von Objektpaaren aufzudecken. Der aktive Nutzer wird dieser Matrix zugeordnet, um daraus eine Empfehlung für ein ‘passendes’ Objekt abzuleiten.

In beiden Fällen werden Informationen über das Verhalten und Vorlieben von möglichst vielen Nutzern gesammelt. Die zu Grunde liegende Annahme des kollaborativen Filtern ist, dass wenn zwei Personen dieselben Vorlieben zu ähnlichen Produkten haben, sie sich auch in anderen Produkten einig sein sollten. Daher auch der Begriff der Kollaboration.

Die Verfahren, die dabei angewendet werden, um vergleichbare Nutzer bzw. Objektgruppen zu finden, gehen von einfachen Distanzbetrachtungen (z. B.

Nearest Neighbour) bis zur Kombinationen von mehreren Verfahren, wie Faktorenanalyse (oder andere dimensionsreduzierende Verfahren), Bayes-Netzwerke, Clustering, Markov'sche Prozessketten und andere heuristikbasierte Verfahren.

Eine Herausforderung für die Anwendung der Verfahren stellt die Tatsache dar, dass sich die Bewertungsmatrizen ständig ändern und dass es über neue Nutzer noch kein Erfahrungswissen gibt, auf dem dessen Einordnung basieren könnte. Dem kann dadurch Rechnung getragen werden, dass mit 'Default-Werten' gearbeitet wird und für Erstnutzer einfachere objektbasierte Empfehlungen angewendet werden, während bekannten Nutzern, über die viele Daten vorliegen, auf Basis von personenbezogenen Verfahren eine Empfehlung gegeben wird.

4.5.11 Clustering

<table>
<tr><td colspan="4">Clusteranalyse: hierarchisch, partitionierend, graphentheoretisch, optimierend</td></tr>
<tr><th>Absicht / Zweck</th><th>Lernen</th><th>Analys. Daten</th><th>Ergebnis</th></tr>
<tr><td>Klassifikation</td><td rowspan="2">Überwachtes Lernen</td><td>Nominalskala</td><td>Nominalskala</td></tr>
<tr><td>Prognose / Vorhersage</td><td>Ordinalskala</td><td>Ordinalskala</td></tr>
<tr><td>Segmentierung</td><td rowspan="3">Unüberwachtes Lernen</td><td>Kardinalskala</td><td>Kardinalskala</td></tr>
<tr><td>Abhängigkeitsanalyse</td><td rowspan="2"></td><td rowspan="2">Cluster / Gruppen</td></tr>
<tr><td>Abweichungsanalyse</td></tr>
</table>

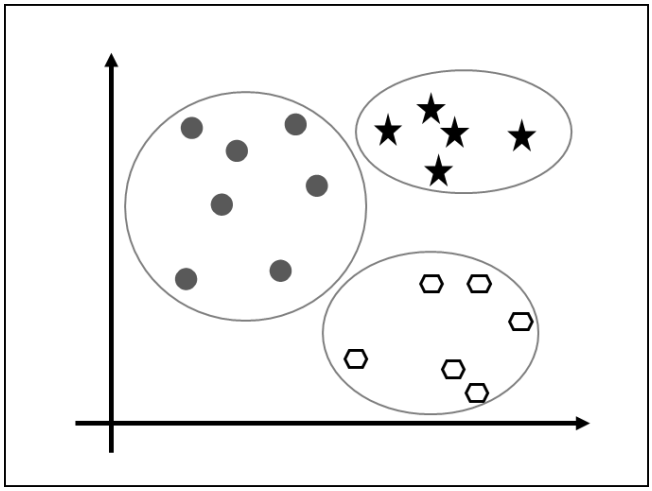
Die Clusteranalyse ist ein Verfahren zur Bündelung bzw. Gruppierung von Objekten. Das Ziel ist dabei, die Objekte so zu Gruppen (Clustern) zusammenzufassen, dass die Objekte innerhalb einer Gruppe sich möglichst ähnlich und die Gruppen als Ganzes sich möglichst unähnlich sind. Da im Gegensatz zu klassifizierenden Verfahren (z. B. Diskriminanzanalyse) die Gruppen vor

der Anwendung der Analyse nicht bekannt sind, sondern stattdessen das Ergebnis der Verfahrensanwendung darstellen, handelt es sich um ein Verfahren des unüberwachten Lernens.

Clusteranalysen werden häufig eingesetzt als:

- Grundlage einer Markt- bzw. Kundensegmentierung,
- Grundlage zur anschließenden, automatischen Klassifizierung von Daten,
- Im Bereich der Bilderkennung.

Bildlich kann man sich die Clusteranalyse als ein Verfahren vorstellen, das Punktewolken (die Punkte stellen die einzelnen Datensätzen in einem n-dimensionalen Raum dar, wobei die n-Dimensionen den Variablen entsprechen) zu ähnlichen Gruppen zusammenfasst. Ein Cluster stellt dann eine Punktewolke dar, die ähnliche Punkte zusammenfasst. Vereinfacht auf zwei Dimensionen stellt sich das so dar:



Es gibt sehr viele unterschiedliche Verfahren der Clusteranalyse. Die Verfahren unterscheiden sich unter anderem nach den folgenden Fragestellungen:

- Wie erfolgt die **Gruppeneinteilung?** Sind die Zuteilungen zu Gruppen eindeutig, überlappend oder ‘fuzzy’, d. h. ‘schwammig’ mit Wahrscheinlichkeiten versehen.

- Welches **Ähnlichkeitsmaß** wird angewendet? Die Gruppen werden so gebildet, dass die Ähnlichkeit *in* den Gruppen und die Unterschiede *zwischen* den Gruppen möglichst groß sind. Ähnlichkeit kann bedeuten, dass der Abstand (im Quadrat, logistisch oder linear) der einzelnen Punkte vom Gruppenmittelpunkt minimiert wird, während gleichzeitig der Abstand der Mittelpunkte maximiert wird (k-Means Clustering). Die Proximitätsbestimmung kann aber auch nach anderen Kriterien erfolgen, sodass – bildlich gesprochen – die Cluster nicht nur Kreise, sondern auch andere Formen darstellen können (dichtebasiertes Clustering).

- Welche **Skalierung** haben die Daten? Je nach Skalenniveau der untersuchten Daten werden andere Proximitätsmaße angewendet: Bei binären Skalen z. B. der Jaccard-Koeffizient oder das Lance-Williams-Maß; bei Nominal-Skalierung das Chi-Quadrat-Maß und bei metrischer Skalierung der Pearson-Korrelationskoeffizient oder euklidische Metrik.

- Welches **Verfahren** wird angewendet? Es gibt **hierarchische Verfahren**. Die hierarchisch agglomerativen Verfahren gehen zuerst von genau so vielen Clustern wie Datensätzen aus und agglomerieren (fusionieren) die Cluster so lange, bis die Clusteranzahl und die Kennwerte den Anforderungen entsprechen. Bei den divisionalen Verfahren wird umgekehrt vorgegangen. Startend mit einem einzigen Cluster werden Schritt für Schritt neue Gruppen gebildet. Bei den **partitionierenden Clusterverfahren** wird von einer gegebenen Anzahl an Clustern ausgegangen, die so lange ‘verschoben’ werden, bis die o. g. Optimierungskriterien (Abstand in der Gruppe versus Abstand der Gruppen untereinander) gegeben sind. Daneben gibt es **dichtebasierte Verfahren**, die Gruppen nach der Dichte der Punktewolke

gruppieren und **kombinierte Verfahren,** die die anderen Verfahren kombinieren.

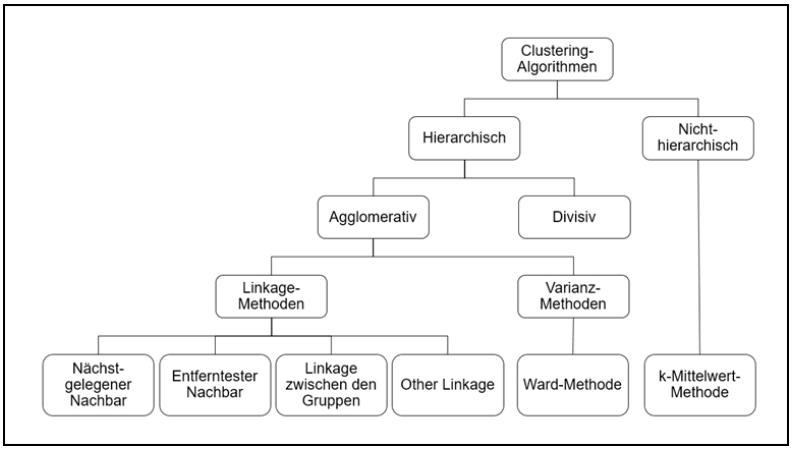


Abbildung 24: http://www.methodenberatung.uzh.ch/de/datenanalyse/interdependenz/gruppierung/cluster.html

Den Clusterverfahren ist gemeinsam, dass die Ergebnisse nicht ohne Sachverständnis analysiert und interpretiert werden sollten. Die Ergebnisse können in einem Dendrogramm dargestellt werden. Dabei wird aber deutlich, dass dies nur im Fall einer begrenzten Anzahl an Daten-sätzen sinnvoll ist.

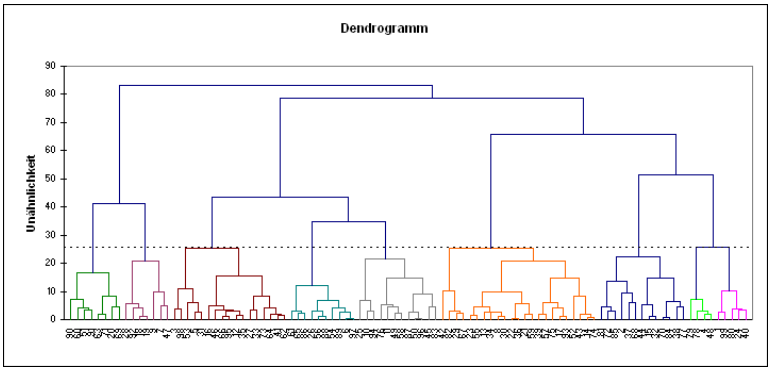


Abbildung 25: https://cdn.xlstat.com/img/tutorials/cahmx6d.gif

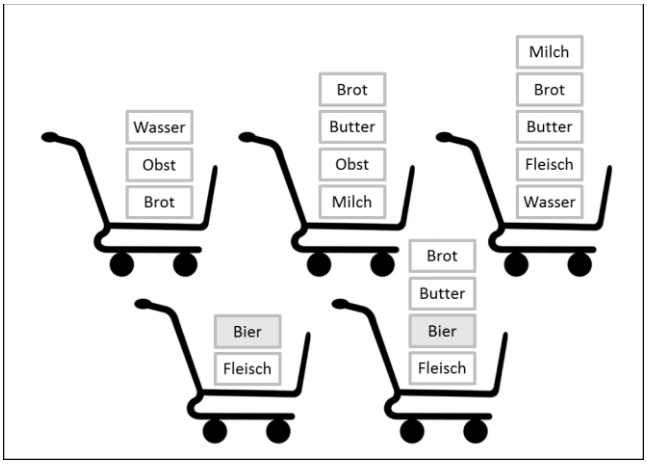
4.5.12 Assoziationsanalyse

Assoziationsanalyse - Association rule learning:			
Absicht / Zweck	**Lernen**	**Analys. Daten**	**Ergebnis**
Klassifikation	Überwachtes Lernen	Nominalskala	Nominalskala
Prognose / Vorhersage		Ordinalskala	Ordinalskala
Segmentierung	Unüberwachtes Lernen	Kardinalskala	Kardinalskala
Abhängigkeitsanalyse			
Abweichungsanalyse			

Die Assoziationsanalyse hat das Ziel, Zusammenhänge und Abhängigkeiten in einer Datenbasis zu entdecken. Es geht um die Aufdeckung von 'Wenn-Dann'-Zusammenhängen aus v. a. transaktionalen Daten. Durch die Analyse z. B. von Warenkörben, also der Produkte, die innerhalb eines Einkaufs erworben wurden, sollen Muster erkannt werden.

Als Gütekriterien, die die Stärke der Zusammenhänge zwischen den Objekten wiedergeben, werden verschiedene Maßzahlen verwendet. Der Support, die Konfidenz und der Lift als Bedeutungsindikator.

- **Support**: Relative Häufigkeit der Beispiele, in denen ein Produkt vorkommt
- **Konfidenz**: Relative Häufigkeit der Beispiele, in denen die Regel richtig ist.
- **Lift**: Der Lift gibt an, wie hoch der Konfidenzwert für die Regel den Erwartungswert übertrifft, er zeigt also die generelle Bedeutung einer Regel an.



Am Beispiel der oben dargestellten Warenkörbe soll die Regel

'Wenn ein Kunde Bier kauft → dann kauft er auch Fleisch'

dargestellt werden:

- Der **Support** beträgt 40 % für Bier (2 von 5 Warenkörbe enthalten Bier), 60 % für Fleisch (3 von 5) und 40 % für die Kombination Bier und Fleisch (2 von 5 Warenkörben)
- Die **Konfidenz** beträgt 100 % (2 von 2 Warenkörbe mit Bier enthielten auch Fleisch)
- Der **Lift** ist 1,66, d. h. der Zusammenhang ist deutlich größer als 1 und damit nicht zufällig. Er berechnet sich nach dem Support der Kombination geteilt durch das Produkt aus den Einzel-Supports (0,4/(0,4*0,6)).

Die Assoziationsanalyse ist ein zweistufiges Verfahren. Aus der gesamten Datenbasis werden mithilfe entsprechender Algorithmen im ersten Schritt die Regeln entdeckt, die einen vorgegebenen Supportwert übersteigen. Im zweiten Schritt werden danach die Regeln 'ausgesiebt', deren Konfidenzwert nicht ausreichend ist.

Die Ergebnisse der Assoziationsanalyse sind dann einfach zu verstehende Regeln, die für Empfehlungen, Sortimentsgestaltungen, Werbeaktionen etc. genutzt werden können.

4.5.13 Faktorenanalyse

<table>
<tr><th colspan="4">Faktorenanalyse</th></tr>
<tr><th>Absicht / Zweck</th><th>Lernen</th><th>Analys. Daten</th><th>Ergebnis</th></tr>
<tr><td>Klassifikation</td><td rowspan="2">Überwachtes Lernen</td><td>Nominalskala</td><td>Nominalskala</td></tr>
<tr><td>Prognose / Vorhersage</td><td>Ordinalskala</td><td>Ordinalskala</td></tr>
<tr><td>Segmentierung</td><td rowspan="3">Unüberwachtes Lernen</td><td>Kardinalskala</td><td>Kardinalskala</td></tr>
<tr><td>Abhängigkeitsanalyse</td><td rowspan="2"></td><td rowspan="2">Reduktion der Variablen auf Faktoren</td></tr>
<tr><td>Abweichungsanalyse</td></tr>
</table>

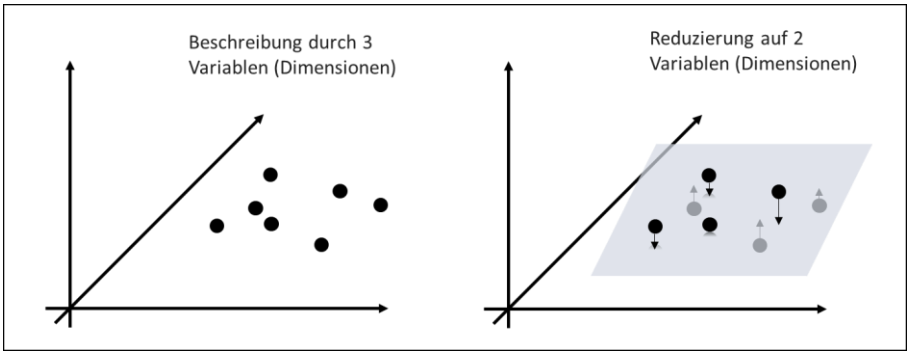
Bei der Faktorenanalyse handelt es sich um ein Verfahren zur Zusammenfassung von Variablen. Ziel ist es, die Anzahl der erklärenden Variablen zu verringern und ‘ähnliche’ Variablen zu Faktoren zusammenzufassen, die dann weitgehend voneinander unabhängig sind. Die Bedeutung der Faktoren ist dabei nicht immer offensichtlich und muss interpretiert werden.

Die Faktorenanalyse findet also dann Anwendung, wenn eine Vielzahl von Variablen vorhanden sind und davon ausgegangen werden kann, dass diese Variablen oft das Gleiche oder Ähnliches ‘aussagen’ und sich deshalb auf eine deutlich kleinere Anzahl aussagekräftiger Faktoren reduzieren lassen. Ein einfaches Beispiel hierzu bildet die Verdichtung der zahlreichen technischen Eigenschaften von PKWs auf wenige Dimensionen, wie z. B. Größe, Leistung, Prestige und Sicherheit.

Traditionell	Zuverl.	Ordentlich	Faktor 1	Spontan	Frei	Kreativ	Modern	Faktor 2
2	2	1	2	5	4	5	3	5
4	5	3	4	2	3	2	2	2
1	2	3	1	4	4	2	5	3
5	4	4	5	1	1	1	1	1
2	4	1	2	2	4	4	3	3
3	4	2	3	4	3	3	3	3
3	3	4	3	1	2	2	1	2
2	2	1	1	2	5	4	5	4
5	4	3	1	1	4	3	5	4
3	4	4	4	2	3	1	3	2
1	2	1	1	5	5	5	5	5
5	4	3	4	2	3	1		1
4	5	5	5	3	2	2	1	2
3	5	4	4	3	1	1	2	2

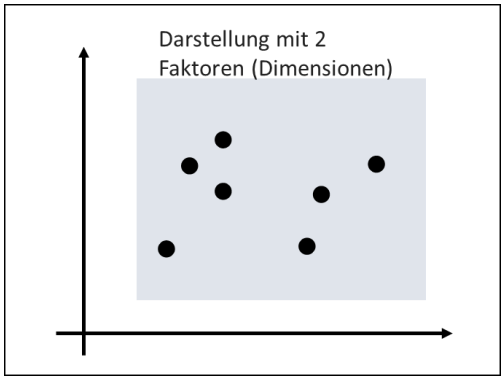
Dem Ziel der Faktorenanalyse – nämlich die Reduktion der Variablen und damit die Reduktion der Komplexität des Modells – steht auf der anderen Seite die Gefahr des Informationsverlustes gegenüber. Diese Aspekte müssen gegeneinander abgewogen werden.

Bildlich kann man das Prinzip der Faktorenanalyse an diesem Beispiel veranschaulichen:



Die Daten werden durch drei Variablen beschrieben. Hier im Beispiel wären die Variablen drei orthogonale Dimensionen (also mit der Korrelation 0). In der Realität ist die Anzahl der Variablen höher und sie sind miteinander mehr oder weniger stark korreliert. Das Beispiel dient lediglich zur Veranschaulichung des Prinzips der Variablenzusammenfassung.

Mit der Faktorenanalyse wird nun also versucht, eine zweidimensionale Fläche (zwei Faktoren) so in den Raum einzupassen, dass die Punkte mit möglichst wenig Verlust (Minimierung der Pfeillängen) auf dieser Fläche dargestellt werden können. Im Ergebnis können die Punkte auf der zweidimensionalen Fläche dargestellt werden, wobei die Reduktion der Komplexität mit einem gewissen Informationsverlust einhergeht.



Das Vorgehen der Faktorenanalyse erfolgt in der Regel in diesen Stufen:

- Im **ersten Schritt** werden alle Variablen ausgewählt, die in die Faktorenanalyse eingehen sollen. Für alle ausgewählten Variablen wird anschließend eine Korrelationsmatrix erstellt, die den Zusammenhang der einzelnen Variablen miteinander darstellt. Damit können einzelne Variablen schon vor der eigentlichen Faktorenanalyse ausgeschlossen werden.
- Der **zweite Schritt** ist die eigentliche Faktorextraktion. Aufgrund verschiedener statistischer Kennzahlen kann entschieden werden, ob das

gefundene Faktorenmodell geeignet ist, die vorliegenden Variablen in Faktoren zusammenzufassen.

- Im **dritten Schritt** werden die Faktoren einer Transformation unterzogen, die als Faktorrotation bezeichnet wird, um damit besser interpretierbare Faktoren zu erhalten.
- Im **vierten Schritt** wird ermittelt, welche Werte die untersuchten Variablen hinsichtlich der Faktoren annehmen. Dies dient dazu, die Faktoren inhaltlich zu interpretieren.

4.5.14 Hauptkomponentenanalyse PCA

<table>
<tr><th colspan="4">Hauptkomponentenanalyse</th></tr>
<tr><th>Absicht / Zweck</th><th>Lernen</th><th>Analys. Daten</th><th>Ergebnis</th></tr>
<tr><td>Klassifikation</td><td rowspan="2">Überwachtes Lernen</td><td>Nominalskala</td><td>Nominalskala</td></tr>
<tr><td>Prognose / Vorhersage</td><td>Ordinalskala</td><td>Ordinalskala</td></tr>
<tr><td>Segmentierung</td><td rowspan="3">Unüberwachtes Lernen</td><td>Kardinalskala</td><td>Kardinalskala</td></tr>
<tr><td>Abhängigkeitsanalyse</td><td rowspan="2"></td><td rowspan="2">Haupt-komponenten</td></tr>
<tr><td>Abweichungsanalyse</td></tr>
</table>

Die Hauptkomponentenanalyse (Principal Component Analysis – PCA) ist eine variablenorientierte Methode, mit der – vergleichbar zur Faktorenanalyse – versucht wird, einen hochdimensionalen Datensatz in einen niederdimensionalen Raum zu projizieren. Dabei versucht die PCA, die Varianzen der Objekte im ursprünglichen Raum möglichst gut mit dem neuen niederdimensionalen Raum abzudecken. Die Hauptkomponentenanalyse besteht darin, eine orthogonale Transformation der ursprünglichen Variablen in eine neue Menge unkorrelierter Variablen, die Hauptkomponenten, vorzunehmen. Im Gegensatz dazu sind die Faktoren bei der Faktorenanalyse nicht zwingend orthogonal.

Die Hauptkomponenten werden nacheinander in absteigender Bedeutung konstruiert. Die Hauptkomponenten sind Linearkombinationen der ursprünglichen Variablen. Die erste Hauptkomponente wird so konstruiert, dass sie für den größten Teil der Variation verantwortlich ist.

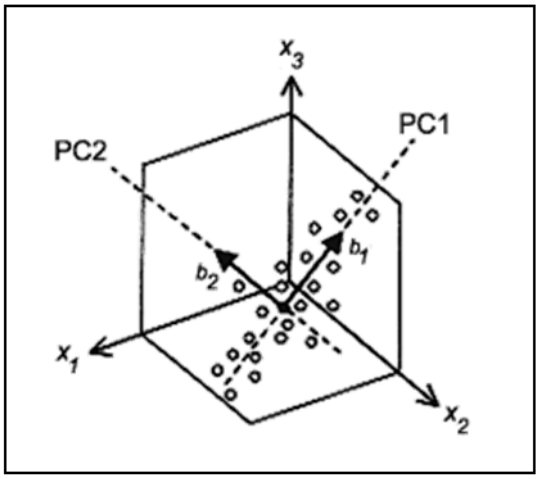


Abbildung 26: Quelle: http://www2.chemie.uni-erlangen.de/projects/vsc/chemoinformatik/erlangen/datenanalyse/bilder/pca.gif

In der Abbildung ist ein Beispiel veranschaulicht, in dem die Datenpunkte eines dreidimensionalen Raumes durch die zwei Hauptkomponenten (PC1 und PC2) dargestellt werden. Die zwei Hauptkomponenten werden solange in dem dreidimensionalen Raum rotiert, bis der Informationsverlust am geringsten ist.

Die Hauptkomponentenanalyse verfolgt also die folgenden Ziele:

- **Repräsentation** multidimensionaler Daten mit einer geringeren Anzahl an Variablen (unter Beibehaltung der Hauptmuster des Datensatzes),
- **Projektion** multidimensionaler Daten in einen niederdimensionalen Raum (unter bestmöglicher Beibehaltung der Variabilität der Daten),

- **Identifikation** versteckter Muster in einem Datensatz, und deren Klassifikation hinsichtlich der Frage, wie viel diese Muster in den Daten versteckte Information erklären (beschreiben) können.

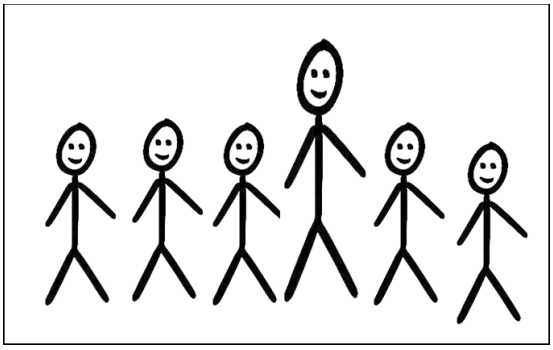
Die Interpretation und Visualisierung der Daten in einem niederdimensionalen Raum ist im Allgemeinen einfacher (insbesondere bei Reduktion auf zwei oder drei Dimensionen). Die Hauptkomponentenanalyse wird daher hauptsächlich als Vorbereitung für die grafische Analyse und Visualisierung von Daten verwendet, wenn diese im zwei- oder maximal dreidimensionalen Raum dargestellt werden.

Hauptkomponenten können auch als Input für weitere Analysen verwendet werden. Das macht jedoch bei vielen multidimensionalen Verfahren keinen Sinn, da diese Verfahren eigene Methoden besitzen, um den erklärenden Anteil der einzelnen Dimensionen zu ermitteln.

4.5.15 Local Outlier Factor

<table>
<tr><th colspan="4">Local Outlier Factor</th></tr>
<tr><th>Absicht / Zweck</th><th>Lernen</th><th>Analys. Daten</th><th>Ergebnis</th></tr>
<tr><td>Klassifikation</td><td rowspan="2">Überwachtes Lernen</td><td>Nominalskala</td><td>Nominalskala</td></tr>
<tr><td>Prognose / Vorhersage</td><td>Ordinalskala</td><td>Ordinalskala</td></tr>
<tr><td>Segmentierung</td><td rowspan="3">Unüberwachtes Lernen</td><td>Kardinalskala</td><td>Kardinalskala</td></tr>
<tr><td>Abhängigkeitsanalyse</td><td rowspan="2"></td><td rowspan="2">Ausreißer</td></tr>
<tr><td>Abweichungsanalyse</td></tr>
</table>

Um in einem eindimensionalen Datensatz einen Ausreißer zu erkennen, benötigt man kein besonders ausgefallenes Verfahren. Das Ergebnis ist in der Regel direkt erkennbar.



In multidimensionalen Datensätzen ist diese Aufgabe deutlich komplexer und nicht mehr intuitiv lösbar. Als Verfahren haben sich Algorithmen bewährt, die die Entfernung bzw. Dichte der Datenpunkte betrachten.

Der Local Outlier Factor (LOF) ist ein Verfahren zur Erkennung von Ausreißern. Die Kernidee von LOF besteht darin, die Dichte eines Punktes mit den Dichten seiner Nachbarn zu vergleichen. Unter der Dichte kann man die Anzahl von ‘nahen Nachbarn’ verstehen. Ein Punkt mit hoher Dichte befindet sich in einer Gruppe, ein Punkt mit geringer Dichte ist ein Ausreißer.

Man berechnet dabei die k-Distanz, also die Distanz des Objektes zu seinen k-nächsten Nachbarn. Daraus wird die Erreichbarkeitsdistanz errechnet, aus der die Erreichbarkeitsdichte ermittelt werden kann. Die Erreichbarkeitsdichte des Punktes wird mit dem seiner Nachbarn verglichen. Ist die Dichte geringer als die der Nachbarn, wird es sich um einen Ausreißer handeln.

Neben dem LOF können für die Ausreisererkennung auch Verfahren verwendet werden, die eigentlich einem anderen Zwecke dienen. Dies sind z. B. Clustering, neuronale Netze, Nächste-Nachbar-Klassifizierungen, Support Vector Machines oder Assoziationsregeln.

4.5.16 Genetische Algorithmen

Das Thema genetische Algorithmen passt nicht ganz zu den vorhergehenden Verfahren, da es sich nicht um ein (multivariates) Analyseverfahren handelt, sondern um ein heuristisches Optimierungsverfahren. Da das Verfahren im Bereich Optimierung eine gewisse Bedeutung hat, wurde es hier aufgenommen.

Die Lösung von Optimierungsaufgaben über mathematische Gleichungssysteme hat vor allem zwei Probleme:

- Mit zunehmender Komplexität (Variablenanzahl) kann der Aufwand für die Lösung der Probleme exponentiell steigen.
- Es besteht die Gefahr, kein globales Optimum zu finden, sondern in einem lokalen Optimum 'steckenzubleiben'. Anschaulich bedeutet das, dass der Flensburger davon ausgeht, dass er auf dem Deich an der höchsten Stelle steht; er vergisst dabei aber die entfernten Alpen.

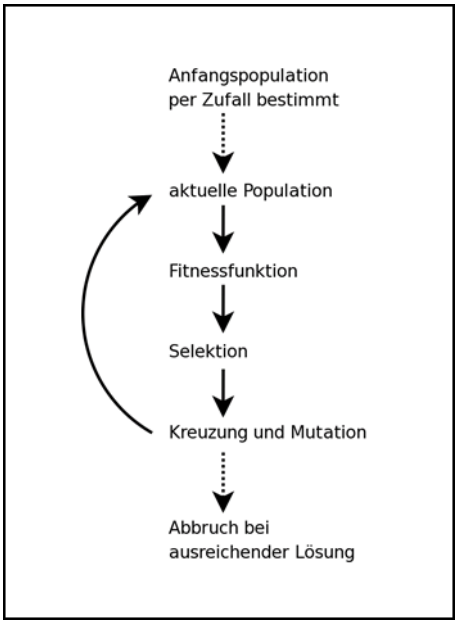
Mit genetischen Algorithmen wird ein Weg gewählt, diesem 'blinden' Teiloptimieren durch die Einbindung einer Zufallskomponente vorzubeugen. Man orientiert sich dabei an den Prinzipien der Evolution.

Die Evolution hat viel Zeit. Über den Zeitverlauf erfolgt eine 'Selection of the Fittest'. Dabei gibt es die Prinzipien:

- Selektion,
- Rekombination,
- Mutation.

Es gibt eine 'Fitness-Funktion' (im dem Sinne, dass die Funktion ermitteln kann, welche Ausprägung den besten 'Fit' hat). Anhand dieser Funktion können die besten Kandidaten ausgewählt werden. Durch Rekombination werden die Eigenschaften der besten Kandidaten kombiniert. Im biologischen Fall ist das die Geninformation von Vater und Mutter, die im Kind kombiniert wird. Im mathematischen Falle werden also die Variablenwerte zweier Kandidaten gemischt, um dann erneut den Fitness-Wert zu berechnen. Dies alleine würde

aber dazu führen, dass die Lösungswerte immer noch auf ein lokales Optimum hin konvergieren könnten. Es bestünde quasi die Gefahr mathematischer Inzucht. Daher wird das Zufallskonstrukt Mutation eingeführt, das dafür sorgt, dass sich einzelne Gene (Werte) zufällig ändern können.



Wird dieser Grundalgorithmus – Generation um Generation – lang genug wiederholt, erreicht man eine zufriedenstellende Lösung des Optimierungsproblems. Eine Garantie für den Optimalwert ist allerdings noch nicht gegeben, da man die Iterationen beenden wird, sobald man mit dem erreichten Fitnesswert zufrieden ist, ohne sicher zu sein, das Optimum tatsächlich erreicht zu haben.

4.5.17 Weitere Verfahren

In den vorangegangenen Abschnitten wurden wichtige Verfahren des maschinellen Lernens dargestellt. Darüber hinaus gibt es zahlreiche weitere Verfahren, die teils eine Weiterentwicklung der bestehenden Verfahren darstellen, oder aber einen eigenständigen Ansatz verfolgen. Erwähnenswert sind in diesem Zusammenhang z. B. noch:

- **Kontingenzanalyse**: Mithilfe der Kontingenzanalyse ist es möglich, die Abhängigkeit bzw. Unabhängigkeit von zwei oder mehreren nominalskalierten Variablen zu untersuchen. Es geht also darum, den Zusammenhang zwischen den Variablen statistisch zu überprüfen. Die Überprüfung erfolgt dabei auf der Basis von Daten, die in Form einer Kreuztabelle (Kontingenztabelle) angeordnet sind.

- **Mehrdimensionale Skalierung** (MDS): Der Hauptanwendungsbereich der multidimensionalen Skalierung ist die Positionierungsanalyse, d. h. die Positionierung von Objekten im Wahrnehmungsraum von Personen. Diese erfolgt in der Regel auf Grund von Befragungen bezüglich der Ähnlichkeit von Objekten (z. B. Produkte in der Marktforschung). Man bildet dann die Dimensionen, mit deren Hilfe diese Objekte dargestellt werden können (vergleichbar der Faktorenanalyse, die aber aus multivariaten Ausgangsdaten basiert).

- **Kausalanalyse**: Bei einer Kausalanalyse werden erhobene Daten auf vermutete Ursache-Wirkungs-Beziehungen zwischen den Merkmalen überprüft. Dazu werden drei Verfahren miteinander verknüpft: Mit der Faktorenanalyse wird überprüft, ob die ermittelten Merkmale die dahinterliegenden (nicht direkt messbaren) Dimensionen und Konstrukte widerspiegeln. Aus der Pfadanalyse leitet sich die Idee einer prüfbaren Ursache-Wirkungs-Beziehung zwischen den einzelnen Dimensionen und Konstrukten ab. Die Regressionsanalyse testet die Wirkungsrichtung zwischen den ursprünglich nicht messbaren Dimensionen.

- **Bayes'sches Netzwerk**: Ein Bayes'sches Netz ist ein gerichteter, azyklischer Graph, in dem die Knoten Zufallsvariablen und die Kanten bedingte Abhängigkeiten zwischen den Variablen beschreiben. Es steht also im Gegensatz zu kausalen Netzwerken, die kausale Zusammenhänge darstellen und nimmt das Konzept der bedingten Wahrscheinlichkeit unter Anwendung der Bayes'schen Regel mit auf.

4.6 Auswahl des richtigen Verfahrens

In den vorangegangenen Abschnitten wurden die wichtigsten Verfahren vorgestellt. Die Frage aller Fragen, die sich für den konkreten Einsatz in der betrieblichen Praxis aufdrängt, ist:

> *Welches Verfahren soll ich einsetzen?*
> *Welches ist das richtige Verfahren für meine Fragestellung?*

Gerade der Anfänger fühlt sich möglicherweise von der großen Anzahl an Algorithmen überfordert bzw. verunsichert, dabei wurde hier ja nur ein Teil der tatsächlich existierenden Verfahren vorgestellt.

Im Netz findet man sog. ‘Cheat Sheets‘ – also Spickzettel, die den Anwender in einer Entscheidungsbaumstruktur zum richtigen Verfahren führen sollen.

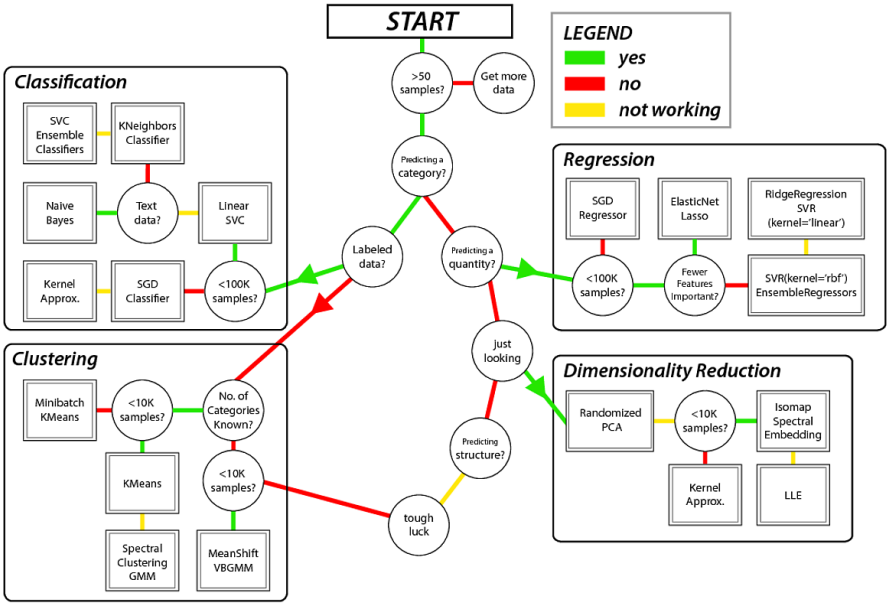
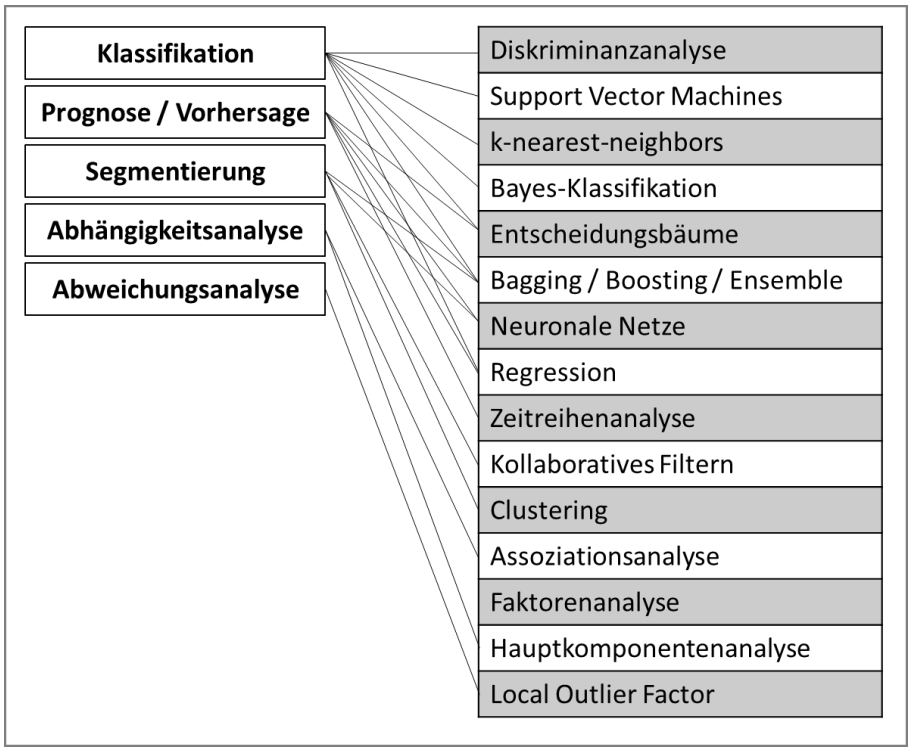


Abbildung 27: https://www.pinterest.de/pin/556827941413198289/

Diese ‘Cheat Sheets’ taugen aber nur bedingt, da die Verfahren nur unvollständig abgedeckt sind und häufig eine Vermischung aus grundsätzlichen Verfahren mit den von den Verfahren verwendeten Algorithmen stattfindet.

Für die Microsoft Azure Machine Learning-Bibliothek gibt es von Microsoft ein ‘Cheat Sheet’, das schon sehr differenziert Einsatzgebiete und Verfahren auswählen lässt. Die Verfahrensauswahl ist aber begrenzt, bedingt durch den Umfang der dahinterliegenden Bibliothek.[9]

Um dem Thema den Schrecken zu nehmen, empfehle ich ein dreistufiges Vorgehen anhand der folgenden Abbildung.



[9] https://docs.microsoft.com/de-de/azure/Machine Learning/Machine Learning-algorithm-cheat-sheet

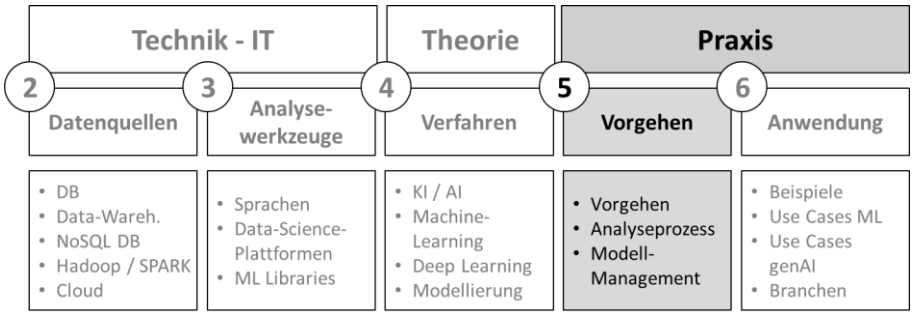
- Im **ersten Schritt** sollte man sich darüber im Klaren werden, was man eigentlich mit der Datenanalyse bezweckt. Will man etwas prognostizieren oder gruppieren? Will man einen Ausreißer identifizieren oder Abhängigkeiten erkennen? Daraus schränkt sich automatisch die Anzahl der geeigneten Verfahren ein.

- Im **zweiten Schritt** überprüft man, welche Forderungen die vorausgewählten Verfahren an den Skalentyp der Daten stellen. Dadurch kann sich eine weitere Einschränkung der Verfahren ergeben.

- Im **dritten Schritt** bleibt einem nichts Anderes übrig, als in die Details der ausgewählten Verfahren zu gehen. Meist stehen verschiedene Unterarten bzw. Algorithmen oder Lernverfahren alternativ zur Verfügung. Hieraus ergibt sich dann eine weitere Konkretisierung des 'richtigen Verfahrens'. Es spricht grundsätzlich nichts dagegen – bzw. ist es sogar wünschenswert – wenn mehrere Verfahren für die Modellbildung genutzt und anschließend verglichen werden (siehe Abschnitt 5.1).

In der Praxis ist die Auswahl des Verfahrens aber nicht so kompliziert. Insbesondere der dritte Schritt verliert, wenn man konkret mit z.B. Python die Modellierung vornimmt, seinen Schrecken. Je Verfahren sind oft nur ein paar Zeilen Programmiercode notwendig und die Syntax ist für die Verfahren oft gleich. Es spricht also nichts dagegen, 'alle' verfügbaren und passenden Verfahren zu verwenden.

5 Vorgehensmodell für ML-Projekte



In diesem Kapitel wird darauf eingegangen, wie beim Einsatz der in den vorangegangenen Abschnitten beschriebenen Verfahren in konkreten Projekten vorzugehen ist. Wie soll ein Machine Learning-Projekt idealerweise ablaufen? Zuerst wird in Abschnitt 5.1 eine bewährte Vorgehensweise vorgestellt. Im darauffolgenden Abschnitt wird dann das Thema Modell-Management behandelt. Abschnitt 5.4 enthält ein 'Cheat-Sheet'. Es werden dabei beispielhaft praktische Code-Snippets in SQL und Python aufgeführt, die häufig in der Arbeit an Data Science Vorhaben genutzt werden können. Im Abschnitt 5.5 ist anhand eines Jupyter Notebooks ein komplettes Analyse-Projekt skizziert.

5.1 Vorgehensweise – Methode

Es gibt verschiedene Vorgehensmodelle, die als Empfehlung für die erfolgreiche Durchführung eines Datenanalyse-Projektes verwendet werden können. Das mit Abstand bekannteste und am weitesten verbreitete Modell ist CRISP-DM (Cross Industry Standard Process for Data-Mining).[10] Daneben ist das von SAS entwickelte SEMMA-Modell noch von gewisser Bedeutung, hat aber nicht den breiten Anwendungsbereich von CRISP.

[10] http://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html

Es soll daher im Folgenden, in starker Anlehnung an das CRISP-Modell, eine Empfehlung für die Vorgehensweise bei Data-Science-Projekten gegeben werden.

CRISP-DM wurde als gemeinsames Projekt unterschiedlicher Unternehmen (SPSS, Teradata, NCR Corporation, Daimler AG und OHRA) ab 1996 entwickelt. Es gingen die Erfahrungen aus Data-Mining-Projekten der beteiligten Unternehmen ein. IBM, das 2009 SPSS übernommen hatte, entwickelt CRISP noch heute weiter, zum Teil als Erweiterung unter dem Namen ASUM-DM (Analytics Solutions Unified Method for Data-Mining / Predictive Analytics).

CRISP beschreibt den Analytics-Prozess in verschiedenen Phasen, die insgesamt einen Kreislauf darstellen und unterschiedliche Rückkopplungen zulassen.

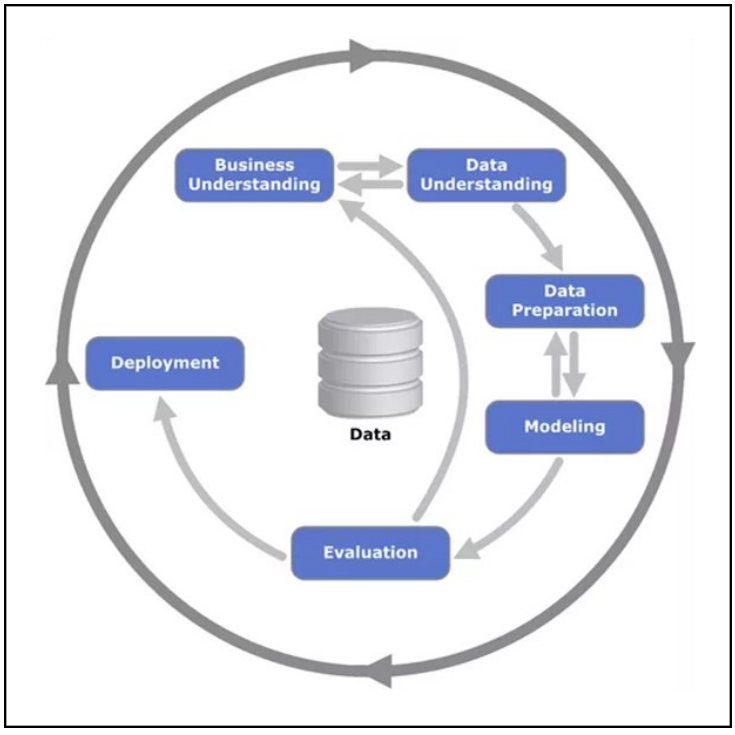


Abbildung 28: http://statistik-dresden.de/archives/1128

Die Phasen sind:

- **Business Understanding**: Verständnis des geschäftlichen Hintergrunds der Fragestellung,
- **Data Understanding**: Verständnis der Daten,
- **Data Preparation**: Vor- und Aufbereitung der Daten,
- **Modeling**: Modellieren mithilfe der unterschiedlichen Verfahren,
- **Evaluation**: Bewertung und Überprüfung der Ergebnisse,
- **Deployment**: Bereitstellung und Anwendung der Ergebnisse im produktiven Betrieb.

Das Grundmodell von CRISP ist gut verständlich und selbsterklärend, es sollen aber zu den einzelnen Phasen einige Anmerkungen gemacht werden.

Business Understanding
In dieser Phase soll eine Business-Fragestellung in eine Daten-Fragestellung übersetzt werden. Das kann nur erfolgreich geschehen, wenn im Data-Science-Team ein Verständnis für das den Daten zugrundeliegende Geschäft besteht. Die Frage, wie beispielsweise der Erfolg einer Marketingaktion verbessert werden kann, setzt bei den im Projekt beteiligten Data Scientisten ein anderes 'Domänen'-Wissen voraus als die Optimierung von Wartungsarbeiten bei Kraftwerkturbinen. Die Verminderung der Abwanderungsrate von Mobilfunkkunden erfordert eine andere Denkweise als es bei der Aufdeckung von Betrugsfällen im Gesundheitswesen der Fall wäre. Es ist daher meist notwendig, dass der Datenanalyst seinen 'Computerkeller' verlässt und sich im Feld 'die Hände schmutzig' macht.

Das ist aber sicher auch ein Grund dafür, warum die Arbeit von Data Scientisten so spannend ist: das Überbrücken von Schnittstellen zwischen Fach- und IT-Bereichen, zwischen Business und Statistik. Auf der Business-Seite herrscht häufig eine falsche Vorstellung von den Möglichkeiten der analytischen Verfahren. Einerseits fehlt die Phantasie, was alles möglich ist – welche Erkenntnisse aus den Daten gewonnen werden können. Auf der anderen Seite

werden aber die Verfahren auch überschätzt, nach dem Motto: ich gebe diesem mir suspekten ‘Mathe-Guru’ Zugang zu meinen Daten und nach zwei Tagen Analyse wird er mit völlig neuen Erkenntnissen zurückkommen. Er wird Antworten auf Fragen liefern, die ich gar nicht gestellt hatte. Genau hier liegt aber die Krux, zu Beginn des analytischen Projektes. Nur wer die richtigen, realistischen Fragen stellt, wird auch zufriedenstellende Antworten erhalten.

Data Understanding

Eng verbunden mit der Business-Understanding-Phase ist die Data-Understanding-Phase. Diese Phase kann noch einmal aufgeteilt werden in die Frage nach der Datenverfügbarkeit und nach dem Datenverständnis.

- **Datenverfügbarkeit**: Dabei geht es um Fragen wie: Welche Daten sind vorhanden? Welche Daten wären noch verfügbar? Welche Daten könnten extern bezogen werden? Welche Daten sollten zusätzlich erhoben werden? Zwei Beispiele dazu: Ein Hersteller von Alkoholika wollte wissen, wie sich Preisaktionen auf den Verkauf eines Produktes auswirken. Im Verlauf des Projektes kam man zur Erkenntnis, dass es Sinn macht, den Abstand (in cm) des eigenen Produktes von dem des Hauptkonkurrenten im Verkaufsregal der Läden zu erheben. Diese Daten lagen natürlich nicht vor. Es ergab sich dann aus der Analyse, dass Preisreduzierungen nur erfolgreich bzw. nötig waren, wenn das Konkurrenzprodukt nahe beim eigenen platziert war. Ansonsten verpuffte die Aktion. Ein anderes Beispiel ist eine Supermarktkette, die den Verkauf von Frischeprodukten u. a. mithilfe von Wetterdaten prognostizierte (siehe Abschnitt 6.2.6). Die Wetterdaten mussten extern dazugekauft werden, da sie in den eigenen Daten nicht vorlagen.

- **Datenverständnis**: Der andere Aspekt ist das Verständnis der Daten im engeren Sinne: Was bedeuten die Daten tatsächlich? Was sich trivial anhört, sieht im Detail alles andere als belanglos aus. Um beim obigen Beispiel zu bleiben, kann man die Frage stellen, was z. B. der Datensatz des Supermarktes bedeutet:

Tagesumsatz eines Produktes x: 1.207,23 €

War das der Gesamtumsatz dieses Produktes, weil der Bedarf nicht höher war, oder weil das Produkt ausverkauft war? Wann war der letzte Umsatz mit diesem Produkt (nachmittags um 14 Uhr oder zwei Minuten vor Ladenschluss)? Es kann Fragestellungen geben, wo diese Unterscheidung irrelevant ist. In anderen Fällen – z. B. wenn es um die Prognose von maximal möglichen Verkäufen geht – ist sie entscheidend. An diesem Beispiel wird deutlich, wie eng das Datenverständnis mit dem Business-Verständnis zusammenhängt. Eine stumpfe, unreflektierte Anwendung beliebiger statistischer Verfahren auf die – eigentlich gar nicht verstandenen – Daten kann zu falschen Ergebnissen führen. Die Weichenstellung dazu fällt in den beiden Anfangsphasen des Projektes.

Data Preparation

Die Datenaufbereitungsphase ist oft die unangenehmste Aufgabe für den Data Scientisten im Projekt. Die Daten können noch so gut im Data-Warehouse aufbereitet worden sein, es wird jedes Mal doch wieder mehr Aufwand notwendig sein als geplant. Es geht einerseits darum, die zu analysierenden Variablen auszuwählen, andererseits darum, die Daten für die entsprechenden Analyseverfahren vorzubereiten. Es müssen Daten aggregiert, transformiert, normalisiert etc. werden. Tabellen werden eventuell erstellt, verändert oder pivotisiert (Spalten und Zeilen getauscht). Unterschiedliche Schreibweisen der gleichen Sachverhalte müssen zusammengefasst werden, Nullwerte evtl. bereinigt oder ergänzt werden. Außerdem müssen die Daten entsprechend den Anforderungen der einzusetzenden Verfahren in der Modellierungsphase vorliegen und entsprechend angepasst werden.

Modeling
Die Modellierungsphase umfasst die eigentliche Analyse der Daten mit den entsprechenden Verfahren.

Der wichtigste Schritt ist zunächst die Auswahl des geeigneten Verfahrens. In diesem Zusammenhang ist hier – analog zum Business- und Datenverständnis in den ersten Phasen – ein **Modell-Verständnis** nötig. Nicht jedes Verfahren eignet sich für jede Fragestellung und nicht alle Datentypen können mit allen Verfahren analysiert werden. Das größte Risiko bergen Verfahren, die zu scheinbar validen Ergebnissen führen, aber inhaltlich sinnlos oder einfach falsch sind. Als vereinfachtes Beispiel kann die Analyse des Zusammenhangs zwischen Monatseinkommen und PS-Anzahl des gefahrenen Autos genannt werden. Eine Regressionsanalyse würde zu dem Ergebnis führen, dass, je höher die PS-Anzahl des eigenen Autos ist, umso höher sich dann das Monatseinkommen darstellt. Die Folgerung aus diesem Modell wäre, dass der Kauf eines größeren Autos zu höheren Verdiensten führt. Der klassische Fehler besteht darin, aus einer Korrelation der Daten auf einen kausalen Zusammenhang (in die falsche Richtung) zu schließen.

Diese Phase ist nun tatsächlich die wissenschaftlichste Aufgabe des Data-Scientisten. Üblicherweise werden in dieser Phase folgende Aktivitäten durchgeführt – nicht zwingend in dieser Reihenfolge, sondern oft im iterativen Vorgehen mit mehreren Schleifen:

- Es wird ausgewählt, welche Verfahren Anwendung finden sollen.
- Daten werden – je nach Verfahren – aufgeteilt in Trainingsdaten (mit denen das Modell erstellt wird), Test- und Validierungsdaten.
- Mit den Daten wird *gespielt*, d. h. es wird versucht, z. B. mit Visualisierungsmethoden erste Erkenntnisse zu gewinnen.
- Modelle werden erstellt und bewertet. Manche Ergebnisse müssen aufgrund statistischer Kennzahlen als unbrauchbar verworfen und das Modell entsprechend angepasst werden (andere Variablen, anderes Verfahren, andere Daten usw.)

- Finetuning der Modelle: Die ausgewählten Modelle können einer Verbesserung unterzogen werden. Die Hyperparameter – also die Parameter, die zur Steuerung des Trainingsalgorithmus verwendet werden – können verändert werden. Hyperparameter-Optimierung kann in manchen Softwarelösungen für bestimmte Verfahren automatisiert werden, um so die möglichst beste Kombination an Parametereinstellungen zu finden. Die Kriterien, nach denen die Güte der Modelle bewertet wird, können unterschiedlich sein. Je nach Zweckmäßigkeit wird nach unterschiedlichen Kennzahlen (z. B. Accuracy, Recall, Precision, Konfidenzniveau, GINI-Koeffizienz) oder anhand einer ROC-Kurve bewertet. Letztere stellt z. B. den Zusammenhang bei Klassifizierungsmodellen zwischen *false positives* und *false negatives* dar.

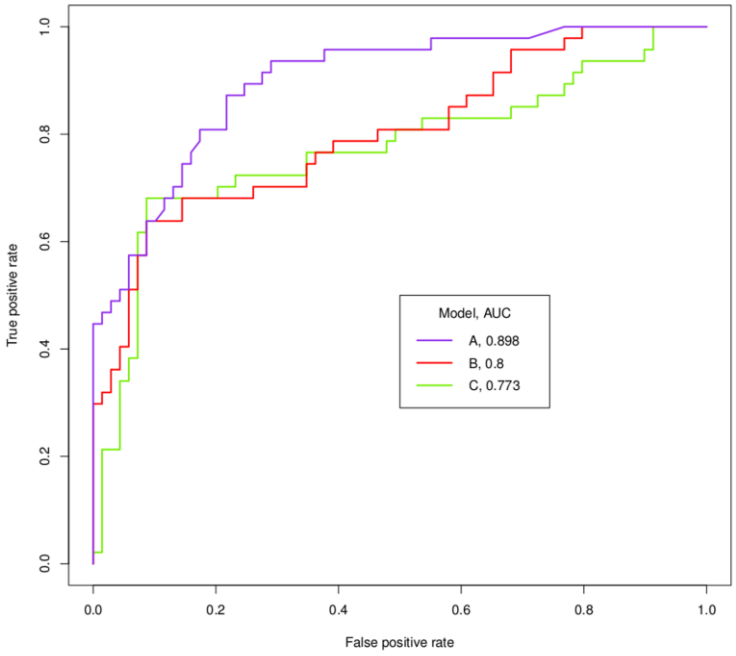


Abbildung 29: Beispiel einer ROC-Kurve

- Eventuell muss ein Rücksprung in die Datenbearbeitungsphase erfolgen, da sich neue Anforderungen an die Daten ergeben haben.

Als Ergebnis dieser Phase liegen ein oder auch mehrere konkurrierende Modelle vor, die die statistischen Qualitätskriterien erfüllen und grundsätzlich geeignet sind, die Fragestellung des Projektes zu beantworten.

In der Praxis wird gerade beim letzten Punkt der ein oder andere Statistik-Purist beim Blick auf die statistische Solidität der Modelle irritiert sein. Die Konfidenzniveaus oder andere Parameter stellen oft die Robustheit der Modelle infrage. Dennoch funktionieren viele dieser *falschen* Modelle in der Praxis und führen zu (geldwerten) Verbesserungen der Ergebnisse. Vor diesem Hintergrund ist auch folgender vielzitierter Satz entstanden:

> 'Essentially, all models are wrong, but some are useful.'[11]

Evaluation

Die Evaluationsphase dient dazu, die Ergebnisse – also das Modell – ein letztes Mal zu überprüfen. Nicht aus statistischer bzw. Datensicht, sondern unter Erwägung der *Business*-Aspekte. Wurden beim Analyseprozess eventuell relevante Gesichtspunkte übersehen oder im Laufe der Prozessschritte vernachlässigt? Ist das Modell tatsächlich geeignet, einen Mehrwert für das Unternehmen zu generieren?

Bevor das Ergebnis *deployed* wird, muss also die konsensuale Entscheidung erfolgen, dass man als Nutzer mit dem Ergebnis zufrieden ist und es nun umgesetzt werden soll. Die Präsentation der Ergebnisse in der geeigneten zielgruppengerechten Form ist ein essenzieller – oft etwas vernachlässigter – Teil des Analyseprozesses.

[11] Box, George E. P.; Norman R. Draper (1987). Empirical Model-Building and Response Surfaces, p. 424, Wiley.

Deployment
Wie diese Phase aussieht, hängt von der Fragestellung des Projektes ab. Wenn es ausschließlich darum ging, (einmalig) Erkenntnisse aus Daten zu gewinnen, um damit z. B. eine Entscheidung zu begründen, dann entfällt in diesem Sinne die Deployment-Phase. Wenn aber das Ergebnis der Modellierung z. B. ein Prognosemodell ist, so wird die Absicht darin bestehen, dieses Modell nun in die operativen Prozesse zu integrieren – meist gleichbedeutend mit einem Deployment in einem operativen System. Ein Modell, mit dem z. B. die Kreditwürdigkeit eines Antragstellers anhand einer Scoring-Funktion ermittelt wird, wird in den Webshop integriert, über den ein Kreditantrag eingegeben werden kann. Aufgrund der hohen Relevanz dieser Phase – erst hier wird aus einer Spielerei von Datenspezialisten eine produktive Business-Anwendung – wird im übernächsten Abschnitt 5.3 näher auf die Ausgestaltung und mögliche Alternativen der Deployment-Szenarien eingegangen.

Phase 0 – *Statement of work*
Eine Phase, die in keinem der Vorgehensmodelle explizit vorkommt, aber die vom Zeitaufwand her einen großen Anteil der Arbeit von Data-Scientisten ausmacht, sind die vorbereitenden bzw. abstimmenden Tätigkeiten. Es sind die Meetings, Telefonate, Gespräche mit der Fachseite, dem Management, mit Softwareherstellern, der IT-Abteilung oder dem Datenschutz etc., die viel Zeit kosten. Viele Menschen müssen erst überzeugt werden, bevor mit der eigentlichen Arbeit begonnen werden kann. Es bedarf schon reichlich Vorbereitungs- und Überzeugungsarbeit, bis die tatsächliche Analysetätigkeit erfolgen kann. Das zu bedauern ist müßig, da es einfach die betriebliche Realität darstellt und zum Job-Profil eines Data-Scientisten dazugehört, sich und seine Arbeit zu verkaufen.

5.2 Modell-Management

Im vorangegangenen Abschnitt ging es darum, den Prozess eines einzelnen Machine Learning- bzw. Data-Mining-Projektes darzustellen. Als Ergebnis erhält man ein oder mehrere analytische Modelle. In Unternehmen und insbesondere in den großen, datengetriebenen Organisationen sammeln sich dann mit der Zeit zahlreiche solcher Modelle an. An den entsprechenden Prozessen sind sehr viele Mitarbeiter in unterschiedlichen Organisationseinheiten beteiligt. Ein Telco-Unternehmen z. B. wird mit der Zeit hunderte unterschiedlicher Churn-Modelle für unterschiedliche Märkte und Kundengruppen mit verschiedenen Versionsständen entwickelt haben.

Das Managen dieser Modelle ist eine komplexe Aufgabe, die häufig am Anfang der analytischen Aktivitäten in den Unternehmen unterschätzt wurde.

- Welches ist das aktuelle Modell?
- Welche Änderungen gab es? Werden diese getrackt?
- Wie sahen die unterschiedlichen Ergebnisse der Modelle aus?
- Wer war an der Erstellung beteiligt, was sind die Rahmenbedingungen?
- Wie ist die Gültigkeitsdauer einzelner Modelle? Nicht nur das Deployment eines Modells, sondern auch das Retirement des Models muss geregelt werden.
- Wie können verschiedene Teams (z. B. aus verschiedenen Ländern) zusammenarbeiten und von den Modellen der anderen Teams profitieren?
- Wie werden Verbesserungen realisiert?
- …

All diese Fragen können auf Dauer nicht mehr nur in verteilten, zufälligen Dokumentationen oder gar allein in den Köpfen der beteiligten Mitarbeiter festgehalten und verwaltet werden. Analytische Plattformen bieten z. T. Unterstützung für das Model Management an. Es geht dabei um ein – im Idealfall – unternehmensweites Repository, das in der Lage ist, die entsprechenden

Metadata zusammen mit den Modellen zu verwalten. Große Unternehmen legen daher bei der Auswahl einer Data-Science-Plattform zunehmend auf das Thema Model Management gesteigerten Wert. Das Thema bekommt eine höhere Priorität als das Vorhandensein des 555-igsten Algorithmus in der Verfahrensbibliothek. In fortgeschrittenen Umgebungen wird ein Model Management Framework realisiert, das auch die ständige Weiterentwicklung der Modelle und das Scheduling von in Realtime verwendeten Modellen übernimmt.

5.3 Deployment

Das Ergebnis eines Data-Mining-Prozesses ist in der Regel ein Modell innerhalb einer Data-Science-Umgebung, mit dessen Hilfe sich eine Entscheidung treffen bzw. unterstützen lässt. Handelt es sich dabei nicht um einen einmaligen Vorgang, muss idealerweise das Modell in die tägliche Praxis überführt werden. Das ist gleichbedeutend mit dem Deployment des Modells in einer außerhalb der Data-Science-Umgebung liegenden (Software-)Umgebung. Wie lässt sich also beispielsweise das in Python erarbeitete Kredit-Scoring-Modell in den Webshop des Unternehmens integrieren? Grundsätzlich lassen sich dabei folgende Ansätze unterscheiden:

- Nachprogrammieren
- Webservices – multi component
- Webservices – single framework

Nachprogrammieren

Dieser Ansatz ist teilweise noch in der Praxis zu finden, obwohl er die arbeitsaufwendigste und fehleranfälligste Alternative darstellt und die Umsetzung auf wenige Modellierungsergebnisse begrenzt ist. Die Ergebnisse aus der Data-Science Modellierung werden der Anwendungsentwicklung der produktiven Fachanwendung übermittelt. Dann wird versucht, diese Modelle in der entsprechenden Umgebung nachzubilden. Konkret heißt das z. B., dass ein

Python-Modell in Java nachprogrammiert wird. Für Regressionsmodelle ist dies noch möglich, stößt aber bei komplizierteren Modellen (z. B. einem Random Forest oder einem neuronalen Netz) an die Grenzen der Programmiersprache des Produktivsystems. Bei Änderungen im Modell müssen die Änderungen in der Anwendung entsprechend *nachgezogen* werden, was sehr aufwendig und fehleranfällig ist.

Webservices – multi component

Aus den Nachteilen der Nachprogrammier-Alternative wird schnell klar, dass vernünftigerweise ein anderer Ansatz verfolgt werden muss. Das Modell aus der Data-Science-Anwendung sollte also idealerweise direkt in der produktiven Anwendung verwendet werden können. Dies kann dadurch erreicht werden, dass aus dem Modell ein Webservice erstellt wird, der von der Anwendung aufgerufen wird und das entsprechende Ergebnis übermittelt. Modell und Anwendung sind entkoppelt und können unabhängig voneinander weiterentwickelt werden, solange die vereinbarte Schnittstelle genutzt wird. Der dabei verwendete Standard ist meist die REST API. Für die Umsetzung dieses Ansatzes werden unterschiedliche Komponenten – hier am Beispiel für Python – benötigt: eine Komponente, die die Erstellung des Webservices unterstützt (z. B. Flask), eine Webserver-Komponente, die den Service bereitstellt und die Kommunikation mit dem anfragenden Produktivsystem organisiert (z. B. django), eine Kapselungstechnologie (z. B. docker) und eine Komponente, die diese gekapselten Umgebungen orchestriert (z. B. kubernetes).

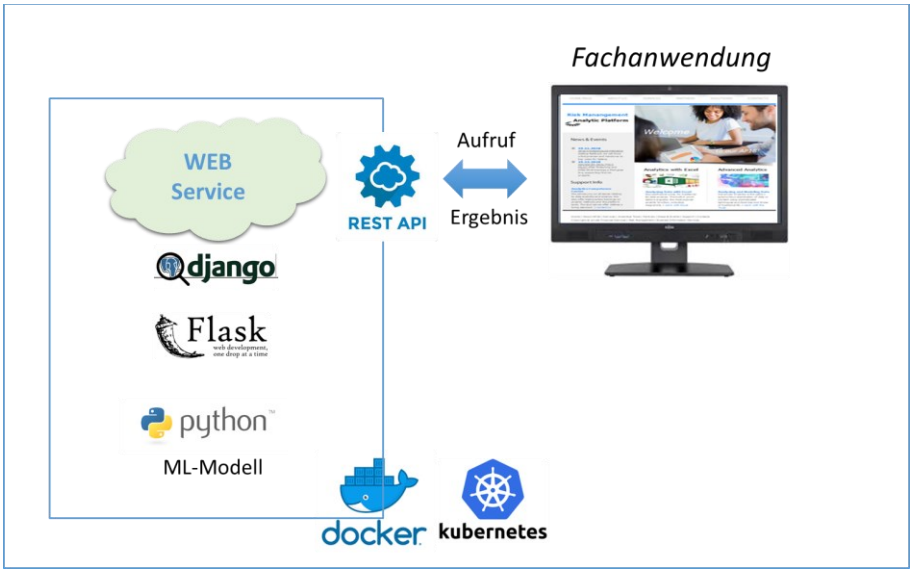


Abbildung 30: Schematische Webservices-Architektur

Bei diesem Ansatz gilt es zu bedenken, dass durch den Einsatz unterschiedlicher Komponenten die Komplexität und Anforderungen an die DevOps-Kollegen steigen. Außerdem ist zu berücksichtigen, dass das von einem Data-Scientist erstellte Modell Teil einer produktiven Anwendung wird. Die Ansprüche an die Programmierung, insbesondere was Fehlerhandling und Stabilität angeht, steigen dadurch und überfordern daher mithin die Skills des Data-Scientisten. Eine Einbeziehung und die Qualitätskontrolle durch einen Programmierer sind daher meist empfehlenswert.

Webservices – single framework

Idealerweise sollte ein ML-Modell auf Knopfdruck deployed werden können. Diesem Ideal möglichst nahe zu kommen, versuchen (oder kommunizieren) verschiedene Anbieter mit ihren integrierten Frameworks. Der grundsätzliche Ansatz entspricht der im vorangegangenen Abschnitt erläuterten Webservice-

Idee, nur dass die Bereitstellung des Webservices unter dem Mantel einer einheitlichen Oberfläche geschieht und weitgehend automatisiert funktioniert. Solche Lösungen bieten sowohl die großen Cloud-Anbieter (Amazon, Azure, Google) als auch spezialisierte Anbieter von Data-Science-Plattformen bzw. Big-Data-Infrastrukturen. Beispielhaft seien genannt:

- Databricks (on-Premises oder in Verbindung mit Azure bzw. AWS)
- Cloudera Data Science Workbench

Databricks stellt eine Oberfläche zur Verfügung, über die z. B. ein – in einem Jupyter-Notebook erstelltes – Python-Modell weitgehend automatisiert als Webservice in der Cloud-Umgebung deployt werden kann.

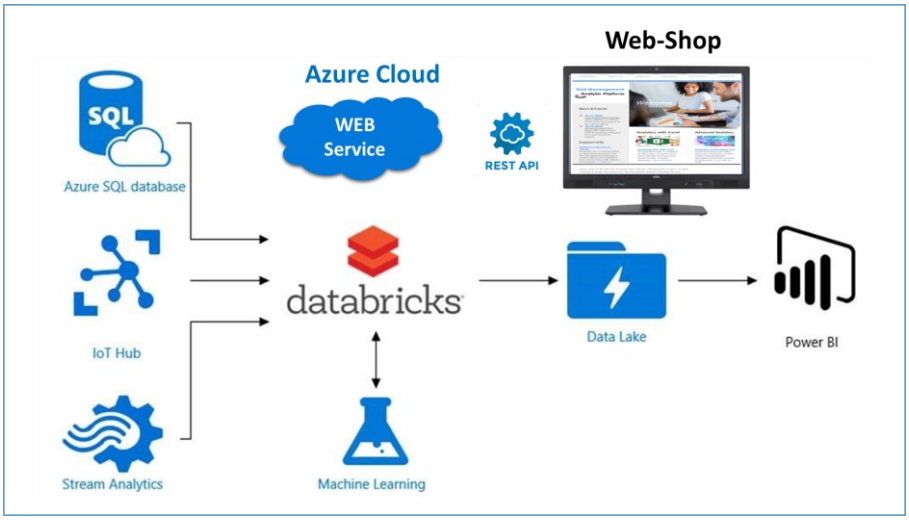


Abbildung 31: Databricks

Für SPARK-Umgebungen gibt es ein vergleichbares Angebot von Cloudera in Form einer integrierten Workbench.

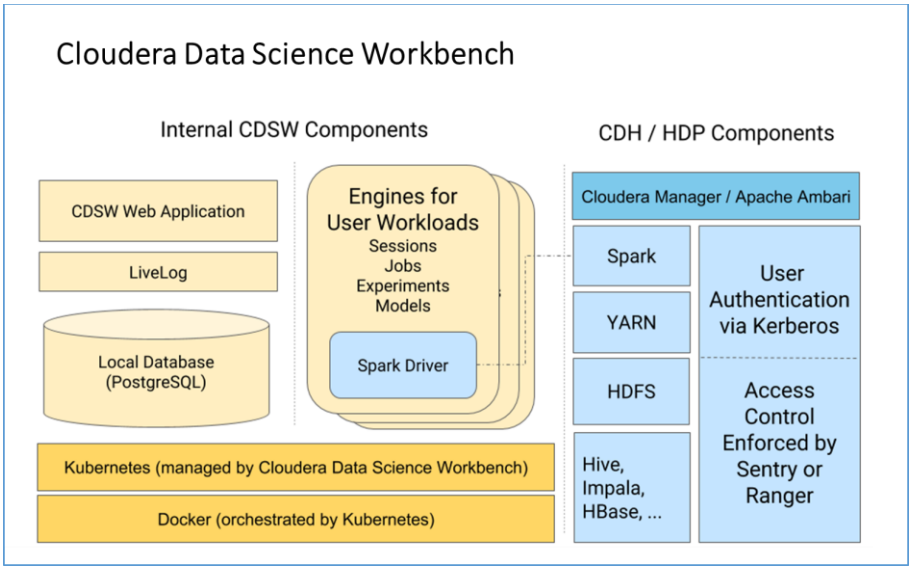


Abbildung 32: Ansicht der Cloudera Data Science Workbench.

Multimodellumgebungen

In Unternehmen, die eine große Anzahl Machine Learning-Anwendungen aus unterschiedlichen Umgebungen gleichzeitig verwenden, kann sich der Ausbau eines **NVIDIA Triton Inference Servers** anbieten. Es handelt sich um eine von NVDIA gepflegte Open-Source-Software, die trainierte Modelle für Machine Learning oder Deep Learning auf jedem beliebigen Prozessor (z. B. GPU oder CPU) ausführen und den Workload bei gleichzeitig laufenden Services bedarfsgerecht steuern kann.

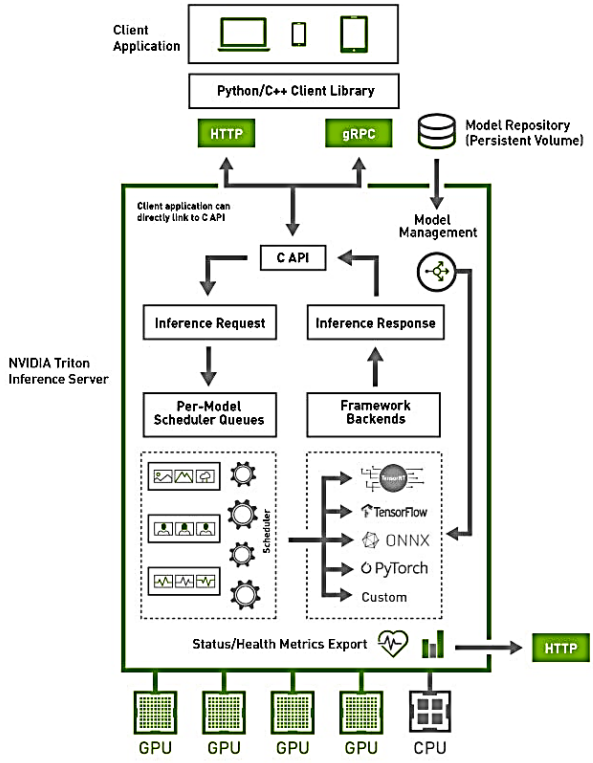


Abbildung 33: NVIDIA Triton Inference Servers

Die dabei unterstützten Frameworks umfassen TensorFlow, PyTorch, Python, ONNX, NVIDIA TensorRT, RAPIDS cuML, XGBoost, scikit-learn RandomForest, OpenVINO und benutzerdefinierte C++-Modelle. Triton ermöglicht Hochleistungsinferenz auf Plattformen wie NVIDIA-Grafikprozessoren, x86- und Arm-CPUs sowie AWS Inferentia.

DevOps- und MLOps-Lösungen wie Kubernetes zur Skalierung und Prometheus zur Überwachung können in die Triton-Umgebung integriert werden.

Die Lösung kann auf den bekannten Cloud- und auf lokalen KI- und MLOps-Plattformen verwendet werden.

5.4 Cheat-Sheet SQL, Python und PySpark

Wie in Kapitel 4 erläutert, sind SQL, Python und Spark die empfohlenen ‘Umgebungen‘ für Data-Science-Aufgaben. Im Folgenden sind Code-Snippets aufgeführt, die bei Datenanalysen häufig zum Einsatz kommen.

Zur Arbeitserleichterung kann die Sammlung auf der Website zu diesem Buch www.data-science-buch.de/cheat-sheat kopiert werden.

SQL-Code	Beschreibung	Beispiel
date(SpalteTimestamp)	Wandelt Timestamp bzw. Datetime in ein Datum um.	2024-01-07
month(SpalteDatum) as Monat	Gibt den Monat eines Datumsfeldes als Integer aus.	11
case when isnull(TabellenspalteA) then 0 else 1 end as SpalteA_hat_Wert	Wenn Tabellenspalte A leer ist, dann wird eine 0 ausgegeben.	0
case when (TabellenspalteB like'%Begriff%') then 1 else 0 end as Begriff_in_Spalte	Wenn in der Tabellenspalte B die Zeichenfolge 'Begriff' vorkommt (egal an welcher Stelle), dann wird eine 1 ausgegeben.	1
when SpalteX in ('A', 'B', 'C')	Bedingung, ob in Spalte X die Zeichen A, B **oder** C vorkommen.	
substring(SpalteY, 4, 3)	Gibt den Teil der Zeichen einer Spalte wieder (Spalte, Start, Länge).	Bei**spi**el spi
weekofyear(DatumSpalte)	Kalenderwoche eines Datums	32
datediff(month,DatumSpalte, current_date()) as Alter_in_Monaten	Gibt die Differenz zweier Datumsfelder in Monaten aus.	73
date_format(DatumSpalte, 'yyyy-MM')	Wandelt ein Datumsfeld entsprechend um.	2024-01
int(round((length(SpalteZ)/4),0))	Gibt die Zeichenlänge einer Spalte an (length), teilt sie durch vier, rundet das Ergebnis (round) auf einen Integer-Wert (Int).	68
join TabelleA as ta on ta.SpalteA = tb.SpalteA and ta.SpalteB = tb.SpalteB and ta.SpalteC='Kriterium'	Join zweier Tabellen, in denen zwei einander entsprechende Spalten die gleichen Werte haben und ein Kriterium erfüllt ist.	

Bei der Arbeit in einer Spark-Umgebung (z. B. mit Databricks) wird meist in einem Python-Notebook PySpark genutzt. Dadurch können die Vorteile der Spark-Architektur (wie Skalierbarkeit und parallele Verarbeitung) bei der Manipulation der Tabellen genutzt werden.

PySpark	Beschreibung
df1.join(df2,df1.SpalteA == df2.SpalteA,'INNER')	Inner Join der Tabellen df1 und df2
df.crosstab('ColA', 'ColB').display()	Kreuztabelle
df.drop('ColA')	Entfernung der Spalte ColA
df.orderBy(['ColA', 'ColB'], ascending = [0, 1])	Sortierung nach ColA absteigend und nach ColB aufsteigend
df.dropDuplicates(['ColA', 'ColB'])	Löschung von Duplikaten (d. h. Einträgen mit gleichen Werten) in ColA und ColB

PySpark bietet jedoch nicht den Funktionsumfang der Python-Bibliothek Pandas. Zur Nutzung von Pandas muss der PySpark-Dataframe erst in einen Pandas-Dataframe umgewandelt werden. Dies wirkt sich auf die Performance aus, da Pandas nur auf einem Single Node ausgeführt wird. Daher sollte die Verarbeitung sehr großer Tabellen vorzugsweise in PySpark vorgenommen werden.

Der Übergang von PySpark zu Pandas und umgekehrt erfolgt durch die Befehle, die in der nachfolgenden Tabelle aufgelistet sind.

PySpark	Beschreibung
df = _sqldf.toPandas()	Wandelt den PySpark-Dataframe in einen Pandas-Dataframe df um.
spark.createDataFrame(df).display()	Wandelt den Pandas-Dataframe in einen PySpark-Dataframe um.

Wichtige Python-Befehle zur Bearbeitung von Pandas-Dataframes sind in der nachfolgenden Tabelle aufgelistet.

Python (Pandas) Code	**Beschreibung**
df.sort_values(by = ['SpalteA', 'SpalteB', 'SpalteC'], ascending = True)	Sortierung nach den Spalten A, B und C aufsteigend
df.drop_duplicates(subset = ['SpalteA', 'SpalteB'], keep = 'last')	Löscht Duplikate in Spalte A und B und behält den letzten Wert bei.
df.drop(columns = ['SpalteA', 'SpalteB'])	Löscht die Spalten A und B.
pd['ColX'] = np.where((pd['ColA'] == 'Bedingung'), 'erfüllt' ,'nicht erfüllt')	Erstellt die neue Spalte mit dem Namen ‘ColX‘, in der steht, ob ColA die ‘Bedingung‘ erfüllt.
pd.crosstab(index = [pd['ColA'], pd['ColB'], pd['ColC'], pd['ColD']], columns = 'Anzahl', margins = False, values = pd['ColZ'], aggfunc = 'count').fillna(0).reset_index()	Kreuztabelle nach den Spalten A bis D. In der Spalte ‘Anzahl‘ wird die Anzahl (‘count‘) der Ausprägungskombinationen der Spalten A bis D gezählt.
pd.to_datetime(pd['DatumID'].astype(str), format = '%Y%m%d', errors='coerce')	Wandelt ein numerisches Datumsfeld (im Format 20240304) in ein ‘echtes‘ Datumsfeld um: 2024-03-04.
df.rename(columns = {'Alt': 'Neu', 'Alt2': 'Neu2'}, axis = 1)	Umbenennung von Spalten
pd.concat([df1,df2], axis = 0)	Zusammenfassen zweier Tabellen ‘untereinander‘ (axis = 0) bzw. ‘nebeneinander‘ (axis = 1)
df1.merge(df2, how = 'left', left_on = ['ColA','ColB'], right_on = ['ColA','ColB'])	Vergleichbar mit dem Join-Befehl aus SQL

5.5 Cheat-Sheet Machine Learning im Python-Notebook

In diesem Kapitel soll beispielhaft ein Machine Learning-Projekt anhand einzelner Codezellen eines (I)Python-Jupyter-Notebooks erläutert werden.

Im ersten Schritt werden die notwendigen **Python-Bibliotheken** geladen. Diese Librarys werden eigentlich immer gebraucht, da sie grundlegende Funktionen für DataFrames, mathematische Verfahren und Datums-Formaten bereitstellt.

```
import pandas as pd
import numpy as np
from datetime import datetime
from datetime import date
```

Laden der Daten

Nach dem Import der Librarys werden die für die Erstellung des Machine Learning-Modells notwendigen Daten geladen. Häufig liegen diese als CSV-Datei vor. In einer lokalen Umgebung auf einem PC erfolgt der Upload durch einen einfachen Befehl.

```
import os

df = pd.read_csv('yourdata.csv', low_memory = False)
```

In einer Spark-Umgebung lautet der entsprechende Befehl:

```
df = spark.read.csv('dbfs:/FileStore/document.csv', sep = ',', header = True)
df = df.toPandas()
```

Bei der Nutzung eines Databricks-Lakehouse besteht die Möglichkeit, direkt aus dem Notebook mit SQL auf Lakehouse-Tabellen zuzugreifen.

```
%sql
select
  Variable1, Variable2 …
from
  table1 as t1
  join table2 as t2 on t1.ID = t2.ID
  join table3  …
where
  condition1 and (condition2 or condition3)

df = _sqldf.toPandas()
```

Externe Datenquellen können – soweit verfügbar – über APIs angesprochen werden. Beispielsweise bietet **Yahoo Finanzdaten** zu Aktienkursen und Aktienindexverläufen an. Die entsprechenden Python-Befehle sind nachfolgend dargestellt.

```
import yfinance as yf

dax = yf.Ticker('^GDAXI')    #<-- das entsprechende Ticker-Sympol
dow = yf.Ticker('^DJI')
nick = yf.Ticker('^N225')
hang = yf.Ticker('^HSI')

dax = dax.history(period = 'max').reset_index()
…
```

Für den Zugriff auf Daten bezüglich der relativen Anzahl der in **Google** eingegebenen **Suchbegriffe** in einem zeitlichen Verlauf kann die Python-Google-API genutzt werden.

```
from pytrends.request import TrendReq

pytrend = TrendReq()
suchworte = ['Suchwort1', 'Suchwort2', 'Suchwort3', 'Suchwort4']
trends = pd.DataFrame()

for k in suchworte:
   pytrend.build_payload(kw_list = [k], cat = 0,timeframe = '2023-01-01 2023-03-31',
geo = 'US')
   temp = pytrend.interest_over_time()
   temp = temp[[k]]
   trends = pd.concat([trends, temp], axis = 1)
```

Bearbeitung der Daten

Nachdem die zu analysierenden Daten geladen wurden, sind Manipulationen notwendig. Dabei werden z. B. Tabellen zusammengeführt, Werte angepasst und Variablen verändert oder erzeugt. Hierzu können die im vorangegangenen Abschnitt aufgeführten Code-Snippets verwendet werden.

```
pd['ColX'] = np.where((pd['ColA'] == 'Bedingung'), 'erfüllt' ,'nicht erfüllt')

df.rename(columns = {'Alt': 'Neu', 'Alt2': 'Neu2'}, axis = 1)

pd.concat([df1,df2], axis = 0)

df1.merge(df2, how = 'left', left_on = ['ColA','ColB'], right_on = ['ColA','ColB'])
```

Mit dem folgenden Code kann überprüft werden, ob in der Spalte df['Textfeld'] bestimmte Worte vorkommen. Für jedes Wort wird eine neue Spalte erzeugt, die in einer Zeile den Wert 1 enthält, wenn das Wort in der entsprechenden Zeichenfolge vorkommt, und anderenfalls den Wert 0.

```
liste = ['Zeichenfolge1', 'Zeichenfolge2', 'Zeichenfolge3']
for a in liste:
   df['Wert_'+a] = 0
   df['Wert_'+a] = np.where((df['Textfeld'].str.contains(a)), 1 , df['Wert_'+a])
```

Machine Learning-Verfahren erfordern häufig Variablen mit kardinalem Skalenniveau. Ein lineares Regressionsverfahren funktioniert z. B. nicht mit der Variable ‘Farbe‘ und den Ausprägungen ‘Rot‘, ‘Grün‘ und ‘Blau‘. Daher müssen solche nominal skalierten Variablen in entsprechend viele **Dummy-Variablen** umgewandelt werden. Dies sind Variablen mit den Ausprägungen 1 und 0, die als Indikator für das Vorhandensein einer Ausprägung einer mehrstufigen Variable dient. Im o. g. Beispiel wird die Variable ‘Farbe‘, in die drei Variablen ‘Rot‘, ‘Grün‘ und ‘Blau‘ mit einer jeweils binären Codierung umgewandelt.

Der automatisierte Python-Befehl hierzu lautet:

```
df = pd.get_dummies(df, columns = ['Farbe', 'Model'], prefix = ['C_‘,'M_‘])
```

Mit diesem werden automatisch die entsprechenden Variablen bzw. Spalten erzeugt. Anschließend können die ursprünglichen Spalten gelöscht werden.

```
df.drop(columns=['Farbe', 'Model'])
```

Für manche Verfahren ist es erforderlich bzw. im Hinblick auf die Interpretierbarkeit der Ergebnisse ratsam, dass die Variablen gleich **skaliert** sind. Wenn z. B. die meisten Variablen Werte zwischen 0 und 1 annehmen, aber die Variable ‘Kaufpreis‘ in Euro Werte zwischen 207,45 und 23.088,12 aufweist, so sollte auch diese Variable auf Werte zwischen 0 und 1 transformiert werden. Das kann entweder manuell pro Variable erfolgen oder über einen entsprechenden Python-Befehl.

```
# Manuell:
X_std = (X - X.min(axis=0)) / (X.max(axis = 0) - X.min(axis = 0))
X_scaled = X_std * (max - min) + min

# Python-Befehl
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
model = scaler.fit(data)
scaled_data = model.transform(data)
```

Erzeugung des Machine Learning-Modells

Die Erzeugung des Machine Learning-Verfahrens ist mit vergleichsweise geringem Programmieraufwand verbunden.

Zuerst werden die Daten in die Zielvariable und die beschreibenden Variablen aufgeteilt. Anschließend werden die Daten in einen Trainings- und einen Testdatensatz aufgeteilt (20 % der Daten werden im u. g. Beispiel zufällig dem Testdatensatz zugeordnet).

```
from sklearn.model_selection import train_test_split

Ziel = df['target']
Daten = df.drop(['target'], axis = 1)
X_train, X_test, y_train, y_test = train_test_split(Daten, Ziel,test_size = 0.2)
variablen = list(X_test)
```

Im nächsten Schritt muss ein für das gewünschte Modell geeignetes Verfahren ausgewählt werden. Für einen **Random-Forest-Classifier** sieht der Python-Code folgendermaßen aus:

```
from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier(max_depth=2, random_state=0)
clf.fit(X_train,y_train)
clf.score(X_test, y_test)
```

Diese drei Befehlszeilen enthalten die gesamte ‘Magie‘ der künstlichen Intelligenz:

1. Das Machine Learning-Modell wird parametrisiert,
2. darauffolgend wird es mit den Trainingsdaten trainiert
3. und anschließend wird das durch das Modell prognostizierte Ergebnis für die Target-Variable anhand der Testdaten mit dem tatsächlichen Wert verglichen. Als Ergebnis wird die Accuracy ausgegeben, die den Prozentsatz der Prognosen des Modells widerspiegelt, die für den Testdatensatz korrekt waren.

Die **Accuracy** ist jedoch nicht in jedem Fall das geeignete Gütemaß für die Bewertung eines Modells. Stattdessen können **Recall, Precision** oder **FScore** herangezogen werden. Die entsprechenden Verfahren in Python sind nachfolgend dargestellt.

```
from sklearn.metrics import precision_recall_fscore_support as score

predict = pd.DataFrame(clf.predict(Daten))
precision_clf, recall_clf, fscore_clf, support_clf = score(y_test, predict_clf)
```

Weitere Erkenntnisse können aus der Feature-Importance der Variablen gezogen werden. Diese gibt die ‘Wichtigkeit‘ (mit einem Wert zwischen 0 und 1) der Variablen für die Modellerstellung wieder. Bei einem Wert von 0 hatte die Variable keinen Einfluss auf das Modell. Jeder andere Wert gibt den Anteil der Variablen an der Erklärbarkeit des Modells wieder.

```
featureimp = clf.feature_importances_
featureimp = pd.DataFrame(data=featureimp)
variablen = pd.DataFrame(data=variablen)
featureimp = pd.concat([variablen, featureimp], axis= 1)
pd.concat([variablen, featureimp], axis = 1)
```

Neben der Vorhersage des Modells für einen Testdatensatz kann auch die Wahrscheinlichkeit angegeben werden, die das Modell für diese Vorhersage ermittelt hatte. Damit kann die Vorhersagegenauigkeit des Modells genauer analysiert werden, über die Einteilung in Klassen hinaus. Im hier dargestellten Beispiel wird eine Exceldatei erstellt, die für jeden Datensatz die Prognose für die Klasseneinteilung und deren Wahrscheinlichkeit enthält.

```
probab_clf = pd.DataFrame(clf.predict_proba(Daten))
predict = pd.DataFrame(clf.predict(Daten))
pd.concat([Ziel, predict, probab_clf], axis = 1).to_excel('predict_clf.xlsx')
```

Es erfordert kaum Aufwand, mehrere Verfahren für die Modellerstellung zu nutzen. Denn die Codes sind analog anzuwenden.

Logistische Regression

```
from sklearn.linear_model import LogisticRegression

logistic = LogisticRegression()
logistic.fit(X_train,y_train)
logistic.score(X_test, y_test)
```

XGBoost

Anhand des folgenden Beispiels mit einem XGBoost-Verfahren soll gezeigt werden, dass neben den Standardeinstellungen auch alle Parameter des Modells individuell angepasst werden können. Dadurch kann das Ergebnis (z. B. die Accuracy) des Modells ‘getunt‘ werden. Man kann dann entweder ‘nach Gefühl‘ an den Parametern drehen oder strukturiert und automatisiert Veränderungen vornehmen, bis eine optimale Einstellung gefunden wird. Das Stichwort dafür lautet ‘Grid-Search‘ für das Tuning der Hyperparameter. Darauf kann aber an dieser Stelle aus Platzgründen nicht eingegangen werden.

```
from xgboost import XGBClassifier

# xgb = XGBClassifier() – Alternativ dazu Finetuning der Parameter:

xgb = XGBClassifier(learning_rate = 0.2,
 n_estimators = 34,
 max_depth = 3,
 min_child_weight = 12,
 gamma = 0,
 subsample = 0.8,
 colsample_bytree = 0.8,
 nthread = 4,
 seed = 27)

xgb.fit(X_train,y_train)
xgb.score(X_test, y_test)
```

Neuronale Netze mit TensorFlow und Keras

Für Deep-Learning-Modelle mit TensorFlow ist der Python-Code geringfügig umfangreicher, aber im Grunde genauso aufgebaut wie die scikit-learn-Modelle.

```
import tensorflow as tf

model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape = (28, 28)),
  tf.keras.layers.Dense(128, activation = 'relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10)
])

model.compile(optimizer = 'adam',
            loss = loss_fn,
            metrics = ['accuracy'])

model.fit(x_train, y_train, epochs = 5)
model.evaluate(x_test,  y_test, verbose = 2)
```

Export des Modells

Wurden nun in der vorangegangenen Phase das ‘beste‘ Modell gefunden und die Parameter für dieses optimiert, so stellt sich häufig die Frage, wie dieses Modell gespeichert und weiterverwendet werden kann. Denn das Modell ist in einer laufenden Session trainiert worden und geht verloren, sobald das Notebook geschlossen wird. Es müsste somit bei jeder Ausführung von Neuem trainiert werden. Um dies zu vermeiden, kann die Speicherung trainierter Modelle mit **Pickle** erfolgen.

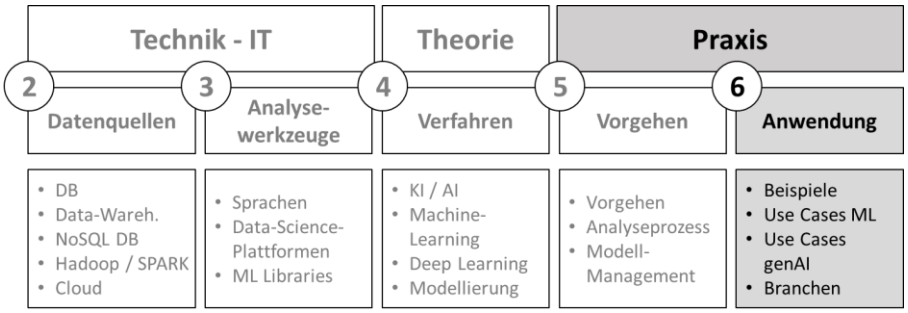
Dazu wird ein Pickle-File erzeugt, das in einer anderen Umgebung und zu einer anderen Zeit geladen werden kann, ohne dass dazu die Trainingsdaten notwendig sind.

Der Code ist übersichtlich, wie in der nachfolgenden Abbildung erkennbar.

```
# Speichern des trainierten XGB-Modells mit dem Namen ‘Model-Name‘:
import pickle
pickle.dump(xgb, open('Model-Name.sav', 'wb'))

# Laden des trainierten Modells an anderer Stelle:
import pickle
xgb_geladen = pickle.load(open('Model-Name.sav', 'rb'))
```

6 Anwendungsfälle – Use-Cases



In diesem Kapitel geht es darum, Anwendungsfälle vorzustellen. Im ersten Abschnitt (6.1) wird auf Besonderheiten ausgewählter Branchen eingegangen. Im zweiten Abschnitt (6.2) werden dann konkrete Fallbeispiele von Machine Learning Projekten beschrieben. (6.3) zeigt Beispiele der Anwendung von beszehenden genAI Tools.

6.1 Use Cases nach Branchen

Machine Learning-Anwendungsfälle gibt es in jeder Branche. In diesem Abschnitt soll nun auf die Besonderheiten einiger Branchen eingegangen werden. In manchen Bereichen ist der Einsatz von Datenanalyse historisch schon weit fortgeschritten, in anderen sind ganz spezielle Einsatzszenarien erkennbar. In einer Studie von McKinsey wurden die Branchen herausgestellt, die eine besonders große Auswirkung von ‘Big Data’ erwarten sollten.[12]

[12] McKinsey (2011)

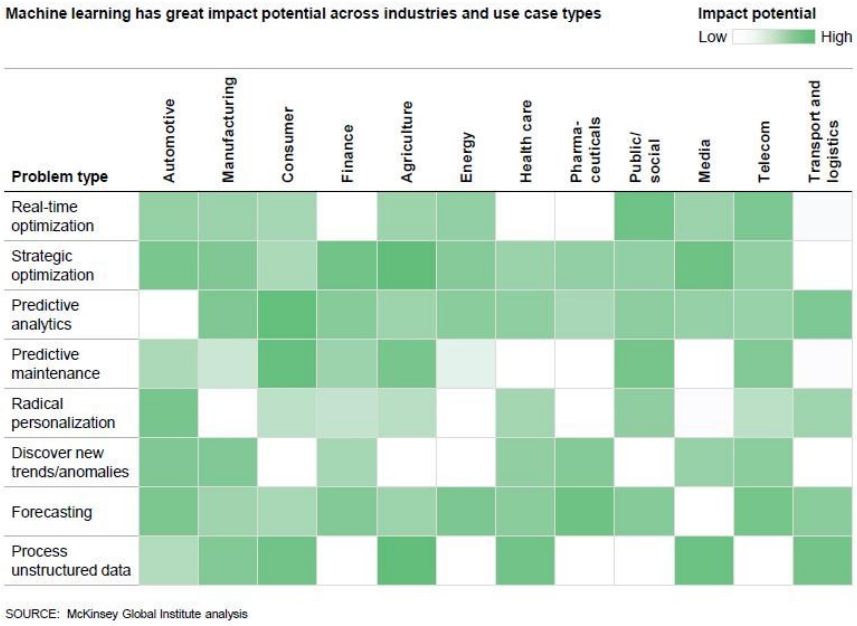
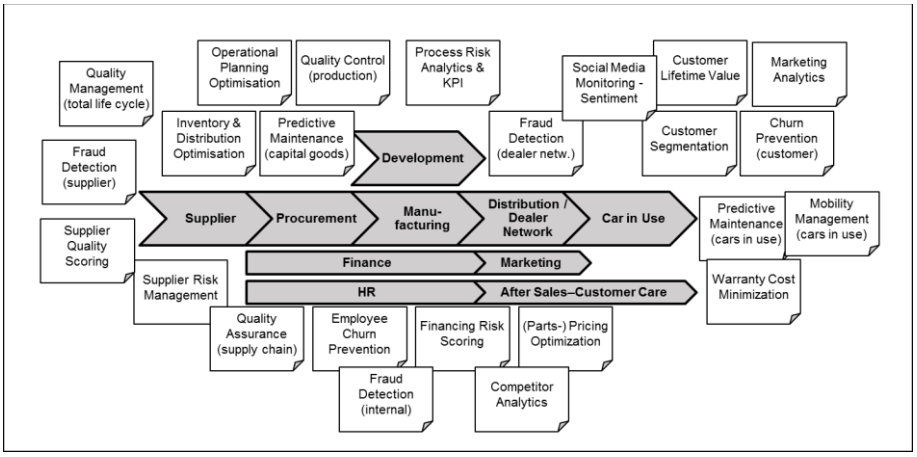


Abbildung 34: http://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/big-data-the-next-frontier-for-innovation

6.1.1 Automobilindustrie

Die Automobilbranche ist geprägt davon, dass über 80 % der Wertschöpfung außerhalb der eigentlichen Automobilhersteller bei Zulieferern erbracht wird. Man kann also nicht nur von Automobilherstellern, sondern genauso gut von Zulieferer-Managern sprechen. Dementsprechend sind neben den für Produktionsbetrieben üblichen Analytics-Einsatzgebieten Projekte mit der Zielrichtung Lieferanten-Management von besonderer Bedeutung.

In der folgenden Grafik sind typische Anwendungsbeispiele entlang der vereinfachten Wertschöpfungskette aufgezeigt.



Über die Wertschöpfungskette der Hersteller hinaus können die im Gebrauch befindlichen Autos eine riesige Menge an Daten liefern (telemetrische Daten, Fehlerdaten, Daten aus dem Motormanagement, GPS-Daten), die Grundlage für weitere Use Cases sind. Die Einsatzgebiete analytischer Methoden – nicht nur für die Hersteller, sondern auch für ganz neue Anbieter – sind dabei noch lange nicht ausgeschöpft. Die Daten aus dem Kauf und der Nutzung von digitalen Diensten der Fahrzeuge bieten zahlreiche Ansatzpunkte für Analysen.

6.1.2 Energieversorgung

Energieversorgungsunternehmen sind einem extremen Wandel unterworfen, der das Geschäftsmodel verändert, völlig neue Prozesse erfordert und die Steuerung des Betriebes datenmäßig revolutioniert.

Von einem eher einfachen Geschäftsmodell, in dem Strom, der billig in wenigen Großkraftwerken produziert wurde, über stabile Netzstrukturen verteilt und in monopolistischen Märkten verteilt wurde, haben sich alle Elemente der Wertschöpfungskette der Energieversorger revolutionär verändert.

- Es muss die **Energiewende** (gleichzeitiger Ausstieg aus der Stromerzeugung durch Atomkraftwerke und der Anstieg von alternativen Energiequellen) gemeistert werden.
- Die **Stromproduktion** erfolgt in einer stark steigenden Anzahl von Einheiten (Solardächer auf Einfamilienhäusern, Windräder, Kraftwärmekopplungsanlagen etc.) mit stark schwankenden Produktionsmengen (z. B. Wind- und Wetterabhängigkeit).
- Die **Netze** müssen erneuert werden, um dem Auseinanderdriften von Produktion und Verbrauch (Windräder in der Nordsee, Verbraucher im Süden) und den starken Produktionsschwankungen gerecht zu werden.
- Die **Märkte** werden liberalisiert. Neue Anbieter, neue Marktmechanismen (Energiebörsen) und die internationale Energiepolitik sorgen für eine notwendige Marktausrichtung der Energieanbieter.

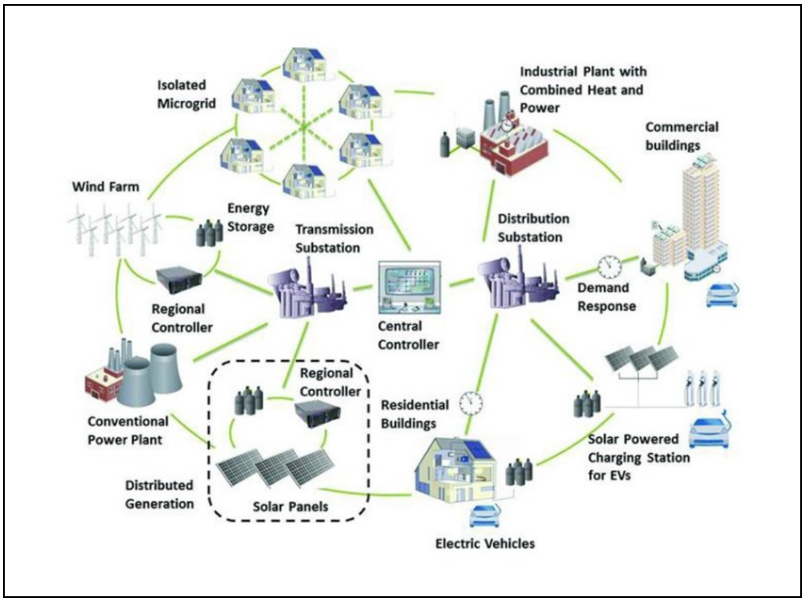


Abbildung 35: Quelle Utegration

Vor diesem Hintergrund eines sich ändernden Geschäftsmodells ergeben sich zahlreiche neue Einsatzgebiete für Machine Learning und Big Data Analytics. Im Folgenden werden einige Beispiele für neue, aber auch 'klassische' Anwendungsfälle genannt:

Energieproduktion:

- Wetterabhängige Produktionsprognose von Energie aus alternativen, dezentralen Quellen (Solarzellen, Windräder)
- Produktionssteuerung des gesamten Produktionsgrids
- Predictive Maintenance von Anlagen (Kraftwerke, Windräder …)
- Realtime Monitoring und Steuerung
- Ausfallerkennung und -Prävention
- Kapazitätsplanung
- Energieverbrauchsprognose
- Sicherheitsanalyse für kritische Infrastruktur

Netzwerk

- Network loss prevention
- Netzwerkplanung und -steuerung

Marketing – Kunden

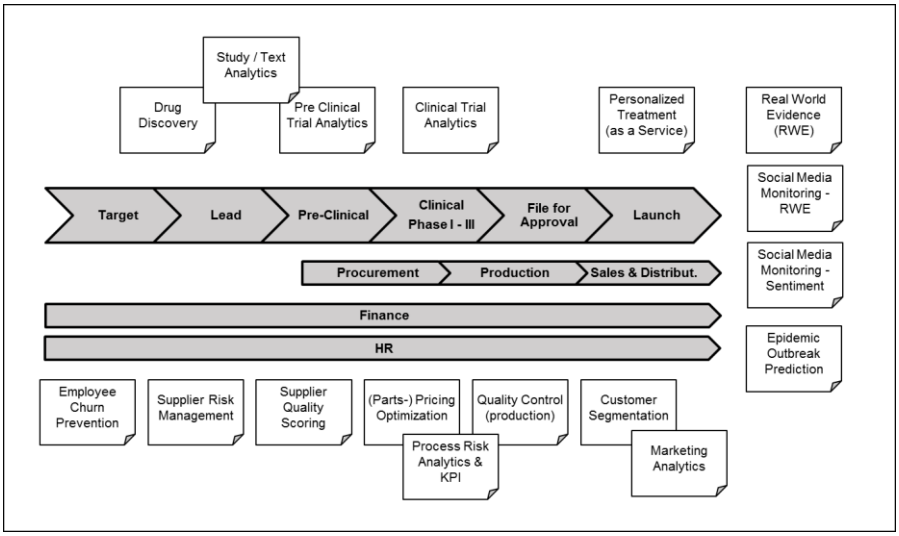
- Fraud-Detection
- Trading-Optimierung
- Tarifsimulation und -optimierung
- Marketing Analytics
- Churn-Prevention (Verhinderung der Abwanderung von Kunden)

Allein aus der vereinfachenden Zusammenfassung der Herausforderung für Energieunternehmen lässt sich erkennen, dass es einen riesigen Bedarf an Ein-

satzgebieten für analytische Anwendungen gibt. Energieunternehmen der Zukunft werden nur erfolgreich sein können, wenn sie die steigende Anzahl an Daten entsprechend managen.

6.1.3 Pharmaindustrie und Biotechnologie

Pharma- und Biotechnologieunternehmen gehören traditionell zu den Hauptnutzern analytischer Verfahren. Das liegt daran, dass neben den üblichen Einsatzgebieten produzierender Unternehmen (z. B. Qualitätskontrolle in der Produktion, Marketing Analytics, HR-Analytics) zusätzlich in den Kernprozessen der Medikamentenentwicklung, mit ihren streng regulierten Entwicklungs- und Zulassungsphasen, sehr viele Daten anfallen, deren Analyse diesen Prozess deutlich beschleunigen, kostengünstiger und sicherer machen kann.



- **Drug Discovery**: Im Rahmen der ersten Stufen bei der Entwicklung eines neuen Medikamentes können Datenanalyse-Verfahren eingesetzt werden. Beispiele sind das initiale Screening von Inhaltsstoffen, um die Erfolgswahrscheinlichkeit vom Zusammenwirken mehrerer

Komponenten zu prognostizieren, oder in der Simulation von Next Generation Sequencing (also die Sequentialisierung des menschlichen Genoms).

- **Präklinische Studien**: Präklinische Studien sind ohne eine adäquate Analyse und Interpretation von Daten nicht denkbar. Mit den Erkenntnissen daraus können unter anderem gesundheitsschädliche Nebenwirkungen von Medikamenten und Behandlungsmethoden früh identifiziert und reduziert werden.

- **Klinische Studien**: Das Gleiche gilt für alle Phasen der klinischen Studien. Daten über Wirkungen, Neben- und Wechselwirkungen von Medikamenten fallen an und stehen für die Analyse bereit.

- **Personalisierte Medikamente**: Darunter versteht man eine maßgeschneiderte Medikation, die zusätzlich zum speziellen Krankheitsbild die individuelle physiologische Konstitution und geschlechtsspezifische Wirkeigenschaften von Medikamenten berücksichtigt. Durch Datenanalyse können individualisierte Therapien entwickelt werden. Diese kann sich sowohl auf einzelne individualisierbare Medikamente, von denen es derzeit in Deutschland 51 Wirkstoffe gibt, als auch auf die individualisierte Kombination von Medikamenten beziehen. Es ist auch denkbar, dass Pharma-Unternehmen diesen Dienst als Service anbieten.

- **Real World Evidence (RWE):** RWE ist die Analyse des tatsächlichen Einsatzes eines zugelassenen Medikamentes. Die ‘Kunst’ besteht darin, aus internen und externen Daten Erkenntnisse über die Wirksamkeit und Nebenwirkungen von Medikamenten im ‘Echtbetrieb’ zu gewinnen.

- **Social Media Monitoring:** Pharmaunternehmen können Textquellen aus dem Internet automatisch scannen und auswerten, um Erkenntnisse über das Sentiment (die Meinung) der Konsumenten bezüglich

des Medikamentes oder des Unternehmens, oder aber über Wirkungen und Nebenwirkungen des Mittels zu gewinnen.

- **Epidemic Outbreak Prediction**: Im Zusammenhang mit der Markt- und Produktionsplanung kann es für Pharmaunternehmen auch von Interesse sein, die Ausbreitung von (epidemischen) Krankheiten zu prognostizieren.

6.1.4 Telekommunikation

Normalerweise wissen Unternehmen darüber Bescheid, wer ihre Kunden sind und wie, wann und welche Produkte und Dienstleistungen des Unternehmens erworben wurden. Was aber nach dem Kauf geschieht, entzieht sich meistens dem Wissen der Unternehmen, mal abgesehen von relativ wenigen Informationen über Wartung und Serviceaktivitäten. In der Telekommunikation ist das anders. Es gibt wenige andere Branchen, in denen die Unternehmen so viele Daten über den tatsächlichen Gebrauch ihrer Dienstleistung durch die Kunden zur Verfügung haben. Jedes Telefonat kann mit Dauer, Ort und Kommunikationspartner einem Kunden zugeordnet werden.

Dementsprechend sind Telekommunikationsunternehmen Pioniere, was den Einsatz von Datenanalyse angeht.

- Um Abwanderungen von Kunden zu verhindern, werden schon seit langem Churn-Prevention-Projekte durchgeführt (s. Kapitel 6.2.5).
- Loyalitätsmodelle werden gebildet.
- Kundengruppen werden geclustert und entsprechend individualisierte Angebote entwickelt.
- Umsatz- und Preisoptimierung durch Tarifvariationen wird simuliert.
- Mit Credit Scoring werden die Kunden vor Vertragsvergabe bewertet.
- Fraud-Analyse soll Betrugsfälle im Netz aufdecken.
- Das Netzwerk wird überwacht (Monitoring) und optimiert.

6.1.5 Handel

Im Handel wurden schon immer große Datenmengen verarbeitet und diese werden auch in Zukunft noch weiter anwachsen. In den Kassen- und Backend-Systemen fallen riesige Datenmengen an. Sie bilden die ideale Basis, um Entscheidungen zur Warendisposition und zur Preisgestaltung auf Basis von Machine Learning-Algorithmen weiter zu beschleunigen und zu automatisieren.

Man muss beim Handel unterscheiden, ob es sich um stationären oder Online-Handel handelt. Sind beide Vertriebskanäle verfügbar, so ist es anzustreben, eine einheitliche Kundensicht unabhängig von der Wahl des Kanals zu gewähren. Wichtige Use Cases für den Einsatz von Machine Learning sind:

Online-Handel

- **Empfehlungen**: Dem Nutzer müssen individualisierte Empfehlungen gemacht werden (siehe Abschnitt Empfehlungen 6.2.8).
- **Preisoptimierung**: Abhängig von der Tageszeit, der Konkurrenzsituation, dem gewählten Zugangsgerät (Computer, Smartphone, Tablet) und den Kundenpräferenzen kann eine individualisierte Preisgestaltung gewählt werden.
- **Forecasting** (Lageroptimierung): Zur Optimierung der Lagerbestände können Verkaufsprognosen erstellt werden.
- **Personalisierung von Kommunikation und Produktangebot**: Werbemaßnahmen und der Aufbau des Online-Shops können kundenindividuell erfolgen.

Stationärer Handel

- **Warenkorbanalyse – Platzierung**: Über Warenkorbanalysen kann die Platzierung von Produkten optimiert werden.
- **Store-Design**: Die Warenplatzierung kann sich auf das gesamte Design des Ladens auswirken.

- **Sortimentsoptimierung**: Die Sortimente einer Verkaufsstelle können anhand der Verkaufsdaten gestaltet werden.
- **Personalisierte Kommunikation**: Mithilfe von Kundenkarten können Informationen über die Kaufgewohnheiten der einzelnen Kunden gesammelt werden, um diese Information für personalisierte Werbung bzw. Angebote zu nutzen.
- **Verkaufsprognose**: Die Prognose der Verkaufsmenge kann die Verkaufs- und Verlustmengen optimieren (siehe Abschnitt 6.2.6)

6.1.6 Banken – Finanzdienstleistungen

Die Finanzdienstleistungsbranche bietet zahlreiche Einsatzmöglichkeiten von Machine Learning und Datenanalyse. Die Einsatzgebiete lassen sich in unterschiedliche Bereiche untergliedern:

Marketing – Sales

- **Kunden-Segmentierung**: Um erfolgreich Cross- und Upsellingaktionen durchzuführen und die Kundenerfahrung entsprechend der unterschiedlichen Anforderungen zu differenzieren, ist eine Segmentierung von Kunden im Finanzdienstleistungsbereich besonders erfolgsversprechend.
- **Prognose Kundenwert**: Ein Teilaspekt der Kundensegmentierung ist die vorausschauende Prognose des Kundenwertes (Lifetime Value) von Kunden. Die Segmentierung der Kunden wird daher nicht nur als statische Aufteilung der Kunden, sondern dynamisch im Zeitverlauf gesehen. Der studentische Girokonto-Inhaber ist z. B. der Hypothekenkredit-Nachfrager der näheren Zukunft.

Kunden-Service

- **Spracherkennung**: Über die Verarbeitung von natürlicher Sprache ist es (zukünftig) möglich, erweiterte Routinetätigkeiten im Kunden-Service-Bereich zu unterstützen. E-Mail-, Chat- oder auch telefonische Anfragen können von 'maschinellen Agenten' (Chat Bots) angenommen und dann entweder vollständig bearbeitet oder zumindest vorverarbeitet werden, um sie dann an einen 'menschlichen' Mitarbeiter weiterzuleiten.
- **Sentiment-Analyse**: Soziale Netzwerke können ausgewertet werden, um die 'Stimmung' der Kunden gegenüber der Bank als Organisation, oder gegenüber einzelnen Produkten automatisch auszuwerten.

Compliance – Risiko – Betrug

- **Risiko-Management**: Sowohl das Risiko einer einzelnen Kreditvergabe, als auch das Risiko des gesamten Kreditportfolios, kann über die entsprechenden Modelle bewertet und die entsprechenden Entscheidungen damit unterstützt werden.
- **Fraud Detection**: Die Aufdeckung von betrügerischen Aktivitäten durch die Erkennung der entsprechenden Muster in den Transaktionsdaten.
- **Geldwäsche – Anti Money Laundering (AML):** Analog zur Fraud Detection werden illegale Geldwäscheaktivitäten aufgedeckt (siehe Abschnitt 6.2.11).
- **Compliance – Abnormal Trading**: Die Compliance-Anforderungen an Banken steigen ständig. Beispielsweise gibt es zahlreiche Überwachungs- und Anzeigepflichten für außergewöhnliche Handelsaktionen, sowohl von Kunden als auch von Mitarbeitern der Geldinstitute.

Produkte

- **Algorithm Trading**: Investitionsportfolios können einem modellorientierten Tradingansatz folgen. Die Investitionsentscheidung erfolgt dabei nicht von Portfolio-Managern, sondern 'automatisch' gemäß einem Tradingmodell, das anhand der Analyse von Vergangenheitswerten erstellt wurde.
- **Personal Finance – Personalisierung**: Den Kunden werden nicht Standard-Produkte 'von der Stange' angeboten, sondern die Produkte sind individualisiert auf den einzelnen Kunden zusammengestellt. Die Auswahl der Kunden und die Erstellung der Produkte erfolgt daten- und modellgetrieben.
- **Product Engineering – Pricing**: Das Entwickeln neuer Produkte und die Bepreisung dieser erfolgt mit der Unterstützung von Machine Learning-Verfahren.

6.1.7 Öffentlicher Sektor

Bei einer Staatsquote von etwa 49 % in Deutschland[13] ist der Public Sector die mit Abstand größte Branche in Deutschland. So gesehen ist Deutschland kein Auto- oder Maschinenbau-Land, sondern ein 'Public-Sector-Land'. Das Potenzial für den Einsatz von Data Science ist riesig. Der öffentliche Sektor ist aber – abgesehen von einigen Ausnahmen – wahrlich kein 'Early Adopter', was Machine Learning angeht. Auf technologischer Ebene schlägt man sich noch – seit nun bald zwanzig Jahren – mit Themen wie eGoverment, elektronischen Akten und Vorgangsbearbeitungssystemen herum; also Themen, die die Industrie längst abgearbeitet hat. An Big-Data-Projekte ist da nicht zu denken. Die Beispiele und Leuchtturmprojekte sind aber vielversprechend und zeigen das vielfältige Potenzial auf. Hier folgen einige Beispiele für tatsächliche bzw. mögliche Anwendungsszenarien:

[13] Quelle: BMF Monatsbericht „Übersichten zur finanzwirtschaftlichen Entwicklung

- **Crime Prevention**: Senkung der Kriminalitätsrate durch verbesserten Einsatz der Polizeikräfte (siehe Abschnitt 6.2.12).
- **Fraud Detection im Gesundheitswesen**: Der Missbrauch im Gesundheitswesen durch Betrug und Verschwendung wird für Deutschland auf jährlich 20 Milliarden Euro geschätzt. Die Aufdeckung von Missbrauch mithilfe von Datenanalyse der Abrechnungsdaten könnte zur Senkung dieses Betrages beitragen.
- **Verkehrsflusssteuerung – Parkraumverwaltung**: Die Realtime-Steuerung von Verkehrsflüssen (insbesondere im Straßenverkehr) über datengetriebene Modelle könnte zur Einsparung von Staukosten beitragen.
- **Betriebsprüfungsoptimierung – Steuerbetrugserkennung**: Zur Verminderung von Steuerbetrug kann die Betriebsprüfung optimiert werden (siehe Abschnitt 6.2.13).
- **Predictive Maintenance der öffentlichen Infrastruktur**: Die öffentlichen Bauten und Geräte könnten über Predictive Maintenance kostengünstiger in Schuss gehalten werden.

6.2 Beschreibung einzelner Use Cases

Im Folgenden sollen einzelne Use Cases für den Einsatz von analytischen Verfahren skizziert werden. Dabei kann es aus Platzgründen nicht um eine ausführliche Beschreibung eines konkreten Beispiels mit allen Details gehen. Es soll aber zumindest die Grundidee des Anwendungsfalles verdeutlicht werden. Der Phantasie sind keinen Grenzen gesetzt. Überall da, wo größere Datenmengen anfallen, ist ein Einsatz von analytischen Verfahren denkbar und häufig auch lohnend. Insbesondere die exponentiell wachsende Datenmenge, die unter das Schlagwort IoT ('Internet of Things') fällt, lässt neue Einsatzgebiete erwarten.

Es handelt sich hier also wie gesagt um eine strukturierte Sammlung von Beispielen für den Einsatz **Datenanalysen-Verfahren**. Es geht ausdrücklich nicht um Beispiele für den Einsatz von generativer AI. Darauf wird im anschließenden Abschnitt eingegangen.

Hier in Abschnitt 6.2 sind Anwendungsfälle beschrieben, deren Ziel es ist durch Analyse von vorhandenen Daten selbst ein Machine Learning / AI Modelle zu erstellen, die dann für den beschriebenen Einsatz genutzt werden können. Im Abschnitt 6.3 folgen Beispiele, wie *bestehende* genAI Tools im Unternehmensumfeld eingesetzt werden können.

6.2.1 Marketing Analytics – Campaign Management

Bei Marketing Analytics geht es darum, die Wirksamkeit und den Erfolg von Marketingaktionen zu verbessern. Es wird anhand vorangegangener Aktionen, bei denen die Ergebnisse bekannt sind, versucht, die Erfolgsfaktoren zu ermitteln und damit den Erfolg zukünftiger Aktionen zu erhöhen. An einem Beispiel soll das verdeutlicht werden:

Es ist geplant einen Werbebrief zu verschicken, der zum Kauf eines elektronischen Gerätes auffordert. Aus der Vergangenheit weiß man, dass mit einer Responsequote von 0,7 % zu rechnen ist. Eine beliebige Adresse

– Michael Müller, Mozartstr. 35, 87654 Musterhausen –

führt also mit 0,7 % Wahrscheinlichkeit zu einer positiven Antwort. Aus der Analyse der Kunden weiß man, dass das Produkt vor allem von Männern mittleren Alters und mit höherem Einkommensniveau gekauft wird. Außerdem gibt es Erfahrungswerte über die Erfolgsquote verschiedener Marketingaktivitäten bei den Käufern in Form eines statistischen Modells.

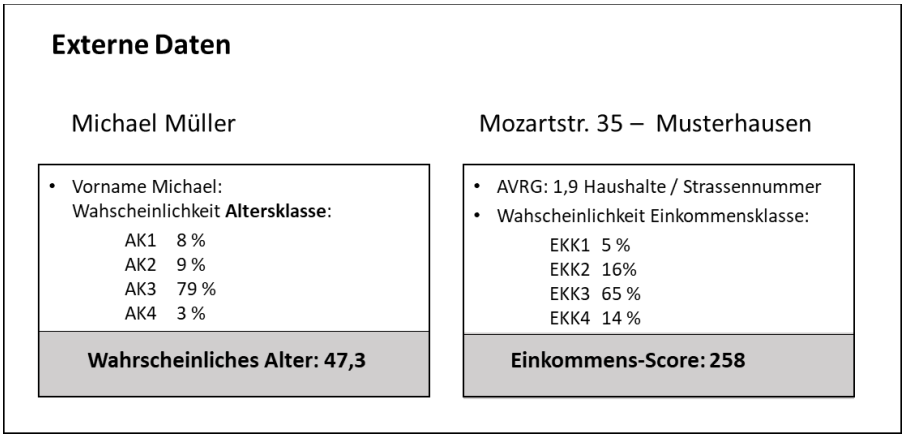
- Verteilung Kunden Produkt xy:

Altersklasse		**Einkommensklasse**	
AK1	7 %	EKK1	12 %
AK2	16 %	EKK2	27 %
AK3	65 %	EKK3	51 %
AK4	12 %	EKK4	10 %

- Kaufwahrscheinlichkeit in Abhängigkeit von Marketingsmaßnahme x_i

$$BP = 0.1254 + 4.5762x_1 + 0{,}125x_2 - 1{,}127x_3$$

Aus einer externen Datenquelle beschafft man sich soziodemografische Daten. Anhand von Namen und Adressen können Aussagen über Alter und Kaufkraft einer Person gemacht werden. Der Name Michael tritt z. B. am häufigsten bei Personen im Alter von 40 bis 55 Jahren auf. Unter anderem anhand der Anzahl von gemeldeten Personen pro Hausnummer kann man auf die Kaufkraft der Wohngegend schließen (wenige Haushalte pro Hausnummer = Einfamilienhaushalte = höhere durchschnittliche Kaufkraft).



Fasst man die Erkenntnisse aus den analysierten eigenen und den dazugekauften sozioökonomischen Daten zusammen, kann man die Zielgruppe der Kampagne und die Marketingmaßnahme so optimieren, dass der Erfolg deutlich steigt. Man verschickt den Brief also beispielsweise nur noch an Personen mit einer bestimmten Alterswahrscheinlichkeit und einem Einkommens-Score-Wert im gewünschten Rahmen. Die Responsequote erhöht sich dadurch auf 2,5 %.

An diesem Beispiel kann man das Prinzip des Verfahrens gut erkennen: Mit größter Wahrscheinlichkeit (97,5 %) wird Michael Müller den Werbebrief nach wie vor wegwerfen. Das Verhalten der Kunden kann immer noch nicht vorhergesagt werden. Durch die Nutzung externer Daten und einem auf vergangene Kampagnen basierenden Prognosemodell, konnte aber die Kaufwahrscheinlichkeit von 0,7 % auf 2,5 % mehr als verdreifacht werden. Wenn man bisher 50.000 € für die Kampagne geplant hatte, reichen jetzt 14.000 € aus, um den gleichen Erfolg zu erzielen. Das ‘bisschen Rechnen’ hatte eine Produktivität von 36.000 €.

6.2.2 Vorausschauende Wartung – Predictive Maintenance

Unter das Stichwort ‘Predictive Maintenance’ fallen Anwendungsfälle, bei denen mithilfe von Prognosemodellen versucht wird, Verschleißteile in Anlagen oder Produkten, die demnächst kaputtgehen werden, rechtzeitig zu erkennen und vor dem Verschleiß auszutauschen. Der Ausfall von Anlagen oder Produkten (z. B. Produktionsanlagen, Verkehrsmitteln) oder auch der eigentliche Austauschprozess von Verschleißteilen (z. B. bei einem Windrad) kann sehr teuer sein. Daher wird versucht, durch die Analyse von Fehlerfällen in der Vergangenheit Verursachungsmuster bzw. zeitlich vorlaufende Indikatoren zu erkennen. Die Analyse dieser Daten führt zu einem Prognosemodell, das die auszutauschenden Teile vorhersagt.

Die Kunst besteht darin, den optimalen Zeitpunkt zu finden, wann ein Teil ausgetauscht werden soll. Theoretisch betrachtet, sollte ein Teil dann ausgetauscht werden, wenn die Summe aus vorsorglichen Austauschkosten und korrigierenden Reparaturkosten minimal sind. Dieses Optimum kann nur ermittelt werden, wenn man die Lebensdauer der auszutauschenden Teile – eben mithilfe der Prognosemodelle – vorhersagen kann und eine Vorstellung der Kostenfunktionen hat.

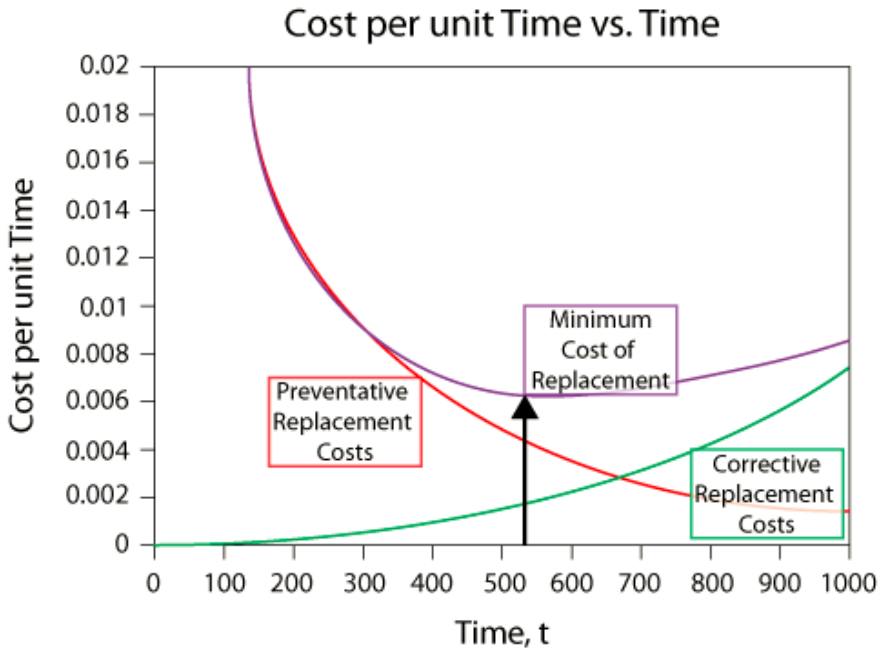


Abbildung 36: Quelle http://reliawiki.com/index.php/Introduction_to_Repairable_Systems

Ein Hersteller von Zügen bietet seinen Kunden (i. d. R. Eisenbahngesellschaften) einen Service an, der Ausfälle von Zügen durch Predictive Maintenance (vorausschauende Wartung) verringert.

Den Kunden wird eine Verfügbarkeit der Züge garantiert, die über Predictive Maintenance sichergestellt wird. In modernen Zugsystemen fallen unzählige Daten im Betrieb der Züge an. Neben den Betriebsdaten werden auch Sensordaten aus den Wagons und den Antriebsteilen gesammelt. Über die Temperaturmessungen können Ausfälle vorhergesagt werden. In einem Beispiel korrelierte ein Abfall der Motorentemperatur von mittel auf niedrig, gefolgt von einem erneuten Anstieg auf mittel, mit dem Ausfall des Motors drei Tage spä-

ter. Dadurch kann der Triebwagen rechtzeitig in die Werkstatt geschickt werden, bevor es zu einem teuren und ärgerlichen Ausfall auf der Strecke kommt.[14]

6.2.3 Prognose der Stromproduktion

Mit dem Aufkommen alternativer Energiequellen wird es für Energieversorger schwieriger, eine bedarfsgerechte Stromproduktion sicherzustellen. Wind- und Sonnenenergie sind sehr stark von lokalen Wetterbedingungen abhängig und decken sich selten mit dem Bedarfsverlauf. Die Stromproduzenten müssen Angebot und Nachfrage über konventionelle Kraftwerke und über Speichermöglichkeiten ausgleichen. Mithilfe von Daten zu Sonnenstunden und Windstärken können historische Daten analysiert und als Grundlage für die Prognose der Stromproduktion und der sich daraus ergebenden Netzbelastung erstellt werden.

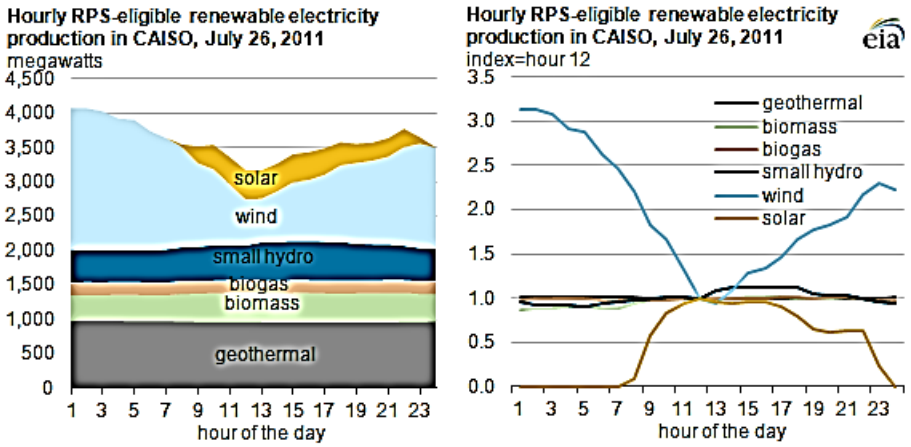


Abbildung 37: https://www.eia.gov/todayinenergy/detail.php?id=4870

[14] IT Production Online

Sowohl die historischen Daten als auch die Prognosedaten müssen möglichst auf sehr granularer Ebene zur Verfügung stehen. Dadurch steigt der Datenumfang und die Komplexität der Modelle, aber eben auch die Qualität der Prognoseergebnisse.

6.2.4 Präventive Medizin

Am Beispiel einer Krankenversicherung in den USA soll gezeigt werden, wie im Gesundheitswesen Kosten gespart und letztendlich Menschenleben gerettet werden können:[15] Diabetes ist in den USA eine Volkskrankheit. Für die Krankenversicherungen bedeutet das sehr hohe Kosten; für die Erkrankten eine massive Einschränkung der Lebensqualität bis hin zu einem erhöhten Todesrisiko. Durch die Analyse der zur Verfügung stehenden Daten über die an Diabetes Erkrankten – auch im Zeitverlauf, als die Erkrankung noch nicht aufgetreten war – können die Versicherer Muster erkennen. Dies kann für zweierlei Fragestellungen verwendet werden:

- Einerseits können die Faktoren identifiziert werden, die den Ausbruch der Krankheit begünstigen bzw. vorhersagen. Dadurch können Risikopatienten, die aktuell noch gesund sind, aber aller Voraussicht nach erkranken werden, frühzeitig identifiziert und durch entsprechende Maßnahmen gesund erhalten werden. Das funktioniert nicht bei jedem Einzelnen, aber die statistische Zahl der Verbesserungen war signifikant.
- Andererseits kann der Verlauf der Krankheit von schon erkrankten Patienten verbessert werden. Es können Risikofaktoren ermittelt werden, die eine Verschlechterung der Krankheit voraussagen. Diabetes-Patienten nutzen z. B. Medikamente falsch bzw. nicht ausreichend. Der Versicherer erkennt das daran, dass er zu wenige Rechnungen für das Medikament bezahlen musste. Andere Risikofaktoren bestanden aus Kombinationen von anderen Erkrankungen, Medikamenten und

[15] Fuzzy Logix (2017)

Krankheitsepisoden, die auf eine Verschlimmerung in nächster Zeit hindeuteten. In allen Fällen kann der Versicherer Maßnahmen anregen, die die Verschlimmerung der Krankheit verhindern helfen.

Das Beispiel lässt sich zwar nicht eins zu eins auf die Situation bei deutschen Krankenversicherungen übertragen, da die Datengrundlage für die hiesigen Versicherungen sehr unterschiedlich ist. Dennoch wird es genügend Einsatzgebiete geben, in denen vergleichbare Erkenntnisse aus Daten gewonnen werden können.

6.2.5 Kundenabwanderung - Customer Churn

Ein klassischer Anwendungsfall von Datenanalyse-Verfahren sind Churn-Prevention-Projekte, also der Versuch, die Abwanderung von Kunden vorherzusagen, um sie dann noch mit geeigneten Maßnahmen verhindern zu können. Dies findet häufig bei Telefongesellschaften statt, insbesondere im Mobilfunkbereich. Das Nutzungsverhalten von abgewanderten Kunden wird in den Monaten vor der Vertragskündigung analysiert, um entsprechende Muster zu erkennen. Stellt man ein Nutzungsverhalten von aktiven Kunden fest, das diesem ‘Churn-Muster’ entspricht, kann man diese Kunden als abwanderungsgefährdet identifizieren. In der Regel werden entweder Scoringmodelle erstellt, wobei der Scoringwert des einzelnen Kunden die Abwanderungswahrscheinlichkeit angibt. Oder es erfolgt über Klassifizierungsmodelle eine Einteilung in ‘gefährdete’ und ‘sichere’ Kunden. Zur Verhinderung der Abwanderung können dann Maßnahmen veranlasst werden, wie z. B. das Anbieten eines neuen Tarifes, der Anruf durch einen Mitarbeiter, das Angebot eines neuen Mobiltelefons etc.

Die Logik der Churn-Prevention kann analog auch in anderen Branchen, oder auf Mitarbeiter bzw. Lieferanten angewendet werden.

6.2.6 Verkaufsprognose

Für Produkte, die in irgendeiner Form transportiert und bereitgestellt werden müssen, ist es wichtig, den Bedarf vorherzusehen. Insbesondere für verderbliche Waren gilt das umso mehr, da eine Überschätzung des Bedarfs bedeutet, dass man Waren wegwerfen muss. Eine Unterschätzung ist verschenkter Umsatz für den Händler.

Eine beispielhafte Anwendung ist die Prognose von Frischewaren in einer Supermarktkette aus Großbritannien.[16] Anhand von Verkaufsdaten aus der Vergangenheit, die mit weiteren externen Informationen (z. B. Wetterdaten, Feiertage, Standort, besondere Ereignisse, Verkaufsaktionen, Werbemaßnahmen etc.) ergänzt wurden, sind Prognosemodelle für den Verkauf der Waren gebildet worden.

Idealerweise wird pro Produktgruppe und Filiale ein eigenes Modell gebildet. Im konkreten Beispiel ergaben sich dabei Größenordnungen von 15 Millionen einzelnen Prognosemodellen (bei 3.000 Filialen und 5.000 Produktkategorien). Bei etwa 150 Variablen pro Datensatz, die bei der Modellbildung berücksichtigt wurden, wird die Anforderung an die Performance der analytischen Plattform deutlich. Im konkreten Fall wurde dabei eine In-Database-Technologie verwendet, da das übliche Vorgehen – Extraktion der Daten aus dem Data-Warehouse in ein externes Analysesystem, Erstellen der Modelle im Analysesystem, Rückspielen der Modellergebnisse in die Datenbank – an systematische Kapazitätsgrenzen stieß. Die Berechnungen würden ansonsten Tage dauern und wären dann schlichtweg zu spät verfügbar.

Die Erkenntnisse bei der Supermarktkette waren im Einzelnen durchaus trivial und leicht nachvollziehbar. Ein Beispiel: Der Anstieg der Temperatur um 10 Grad bedeutet 300 % mehr Umsatz an Grillfleisch und 45 % an Salat.

[16] Fuzzy Logix (2016)

Die Erkenntnisse aus den 15 Millionen Modellen werden anhand aktueller Wetterprognosen für die Prognose der Verkäufe genutzt. Über eine Verbindung zum Logistiksystem erfolgen nun die Bestellungen für die Filialen automatisch. Die 300 % mehr Umsatz an Grillfleisch können ja auch nur realisiert werden, wenn die Ware zur richtigen Zeit am richtigen Ort vorhanden ist. An den Vergangenheitsdaten hat man das übrigens nicht unbedingt nur durch einen Umsatzanstieg in Abhängigkeit von der Temperatur erkennen können, sondern z. B. auch an der Tatsache, dass der letzte Verkauf einer Produktgruppe an einem Tag schon um 15 Uhr erfolgte. Das bedeutet: 'Sold out' am Nachmittag und den ganzen Abend über Umsatz verschenkt. Bei der Analyse der Daten gilt es also immer kreativ zu sein und versuchen zu verstehen, was die Daten bedeuten können.

Natürlich hatte nicht nur das Wetter einen Einfluss auf den Absatz. Auch andere Faktoren spielten eine Rolle: Die Filiale in der Nähe eines Strandes reagiert am Montag nach einem Pokalendspiel anders auf einen Temperaturanstieg als eine City-Filiale. Die Erkenntnisse aus den 15 Millionen Modellen sind im Einzelnen möglicherweise also trivial und nachvollziehbar, aber in Summe dann weit jenseits aller menschlichen Intuition. Das können nur Maschinen.

6.2.7 Warenkorbanalyse

Die Warenkorbanalyse dient dazu, mithilfe des Einsatzes von statistischen Analysemethoden Kundenprofile zu erstellen. Unter einem Warenkorb versteht man in diesem Zusammenhang die Menge aller innerhalb eines bestimmten Zeitraums gekauften Produkte. Die Warenkorbanalyse dient u. a. dazu:

- die Kaufwahrscheinlichkeit für ein Produkt in Abhängigkeit vom Kauf anderer Produkte zu ermitteln,
- Kundentypen nach ihren Kaufpräferenzen zu bilden,
- die Sortimentsgestaltung und Warenpräsentation im Hinblick auf die 'Warenkörbe' der Kunden hin zu optimieren.

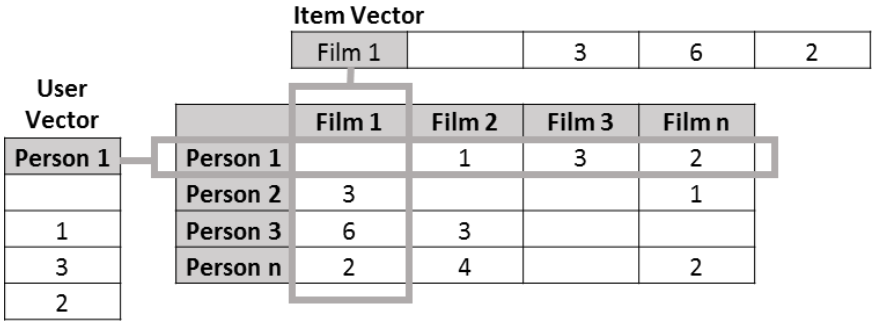
Als Verfahren wird häufig die Assoziationsanalyse eingesetzt, die Aussagen über die Wahrscheinlichkeit eines Kaufes in Abhängigkeit einer schon getätigten Auswahl vornimmt. Eine Empfehlungs-Engine eines Online-Händlers z. B. empfiehlt einem Kunden anhand der vorgenommenen Produktsuchen und der Warenkorbanalyse weitere Produkte zum Kauf.

6.2.8 Empfehlungen – Recommendation Engine

Anbieter vor allem von Online-Angeboten (Online-Shops, Film- oder Musik-Anbieter) nutzen Recommendation Engines, um den Nutzern bzw. Kunden anhand der bekannten Auswahl eines Produktes weitere Produkte zu empfehlen. Grundsätzlich gibt es zwei Parameter, nach denen eine Empfehlung erfolgen kann:

- **Produkt**: Ausgehend vom Produkt wird ein ähnliches Produkt empfohlen. Wird in einem Videostreamingdienst ein Film positiv bewertet, so wird ein vergleichbarer Film empfohlen (‘Sie haben sich xy anschaut, deshalb empfehlen wir yz’).
- **Person**: Anhand der Aktionen vergleichbarer Personen werden Empfehlungen gegeben (‘Kunden, die xy kauften, kauften auch yz’).

Bei Recommendation Engines muss ebenfalls beachtet werden, dass die Daten ständig ergänzt werden. Starre Empfehlungsmodelle funktionieren also nicht.



Moderne Empfehlungs-Engines tragen dem Rechnung und wenden als Verfahren eine Matrix-Faktorisierung an. Die Matrix bezieht sich auf die Tabelle mit den zwei Dimensionen Produkt und Person. Daraus können sowohl Personen- als auch Produkt-Vektoren abgeleitet werden. Die Faktorisierung bedeutet, dass inhärente Faktoren (siehe Faktorenanalyse 4.5.13) bei Bewertungen gebildet werden. Über die entsprechenden Algorithmen kann dann in einer gegebenen Situation (Person x bewertet Film y positiv) eine konkrete Empfehlung gegeben werden. Empfehlungen auf Basis der Matrix-Faktorisierung haben sich gegenüber rein personenbezogenen Empfehlungen (Collaborative Filtering siehe Abschnitt 4.5.10) als überlegen erwiesen.[17]

6.2.9 Betrugserkennung – Fraud Detection

Bei der Fraud Detection geht es darum, durch die Analyse von Daten bestimmte Muster zu erkennen und betrügerisches Verhalten aufzudecken. Die Erstellung der Modelle erfolgt in der Regel mit gelabelten Daten – also mit Daten über Fälle, von denen man weiß, ob sie betrügerisch waren oder nicht. Eine Krankenkasse schaut sich die Abrechnungsdaten eines betrügerischen Arztes genauer an. Die Muster, die man aus der Analyse der Abrechnungsdaten der betrügerischen Ärzte im Vergleich zu den nicht betrügerischen Ärzten

[17] Vgl. Yehuda

erkennt, werden dazu verwendet, das Betrugserkennungs-Modell zu trainieren. Damit können dann laufende Abrechnungen bewertet werden, indem man für diese einen Betrugswahrscheinlichkeits-Scoringwert errechnet. Ab einem gewissen Schwellenwert werden dann Abrechnungen einer genaueren Untersuchung unterzogen.

6.2.10 Kreditrisiko-Bewertung

Bei der Ermittlung des Kreditausfallsrisikos eines Kredit-Antragstellers geht man ähnlich wie im obigen Fraud-Detection-Beispiel vor.

Auch hier geht es darum, durch die Analyse von Daten bestimmte Muster zu erkennen, in diesem Fall eben Muster von Kreditnehmern, die Ihren Zahlungen nicht mehr nachkommen. Die Erstellung der Modelle erfolgt ebenso mit gelabelten Daten – also mit Daten über vergangene Fälle, von denen man weiß, ob sie Kreditausfälle waren oder nicht. Zusätzlich zu den internen Daten werden Scoring-Daten von Kreditagenturen (Schufa, Creditreform etc.) und weitere sozioökonomische Daten herangezogen.

Daraus werden Kreditrisikomodelle ermittelt, mit denen für jeden neuen Antragsteller ein Risiko-Score ermittelt werden kann. Schlechtere Scoringwerte bedeuten dann entweder höhere Zinsen, weitere Sicherungsmaßnahmen (z. B. verpflichtender Abschluss einer Versicherung) oder gar die Ablehnung des Antrages.

6.2.11 Geldwäscheerkennung – Anti Money Laundering

Unter Geldwäsche versteht man die kriminelle Aktivität, illegal erworbenes Geld (aus Drogenhandel, Betrug, Diebstahl, Steuerhinterziehung etc.) in den legalen Geldkreislauf einzuschleusen, sodass es nicht mehr als illegal erkennbar ist. Das Geld wird dadurch ‘sauber’. Laut Geldwäschegesetzen sind Finanzinstitute verpflichtet, Maßnahmen zu unternehmen, damit Geldwäscheaktivitäten aufgedeckt und den Behörden mitgeteilt werden.

Die Erstellung der Modelle erfolgt anhand von gelabelten Daten, also von Kundentransaktionsdaten, von denen man weiß, dass sie geldwaschende Aktivitäten umfassen. Die Modelle werden dann auf die aktuellen Transaktionen angewendet und schlagen bei verdächtigen Aktivitäten an. Diese Fälle werden anschließend in der Regel einer weiteren, manuellen Prüfung durch die Geldwäschebeauftragten der Banken unterzogen.

Die Anzahl der Geldwäschefälle im Verhältnis zu den regulären Transaktionen ist sehr gering. Dadurch ist die Modellbildung erschwert und die Anzahl der Fehlalarme bzw. der übersehenen echten Fälle wird erhöht. Wenn der Anteil der Geldwäschefälle an den Transaktionen im Promillebereich liegt und die Güte (Accuracy) des statistischen Modells bei vielleicht 99 % liegt, wird der Graubereich der ‘False Positives’ bzw. ‘False Negatives’ hoch sein. Dementsprechend wichtig ist in diesem Bereich die Qualität des Modells.

6.2.12 Crime Prevention – Verbrechensbekämpfung

Predictive Analytics können für die Verbrechensbekämpfung eingesetzt werden. Polizeibehörden verfügen über eine große Anzahl von Daten über begangene Straftaten. Die Daten umfassen z. B. Datum, Wochentag, Uhrzeit, Höhe des Schadens, Art des Verbrechens, Ort etc. Die Daten können darüber hinaus angereichert werden. Der geografische Ort eines Verbrechens z. B. kann ergänzt werden um Angaben wie die Entfernung zu einer Autobahn oder Landesgrenze, die Bebauungsart, die sozialen Merkmale des Ortes. Zusätzlich können externe Daten wie z. B. Veranstaltungen, besondere Ereignisse, Wetter und Urlaubszeit hinzugefügt werden.

Abbildung 38: http://www.ithome.com.tw/news/97006

Aus der Analyse der Daten lassen sich Muster erkennen, die eine Prognose von Verbrechensschwerpunkten in Bezug auf Zeit und Ort zulassen. Dementsprechend können die Polizeibehörden ihre Einsatzpläne abstimmen und so durch Präsenz an potenziellen Verbrechensorte zu einer Verhinderung von Verbrechen beitragen.

6.2.13 Optimierung von Betriebsprüfungen

Um die Einhaltung von Regeln zu überprüfen, nehmen Organisationen Compliance-Prüfungen unterschiedlichster Ausprägungen vor. Die Ressourcen für die Durchführungen dieser Untersuchungen sind begrenzt und sollten deshalb auf ihre Ergebniserzielung hin optimiert werden. Ein Beispiel ist die Betriebsprüfung, die die Finanzverwaltung bei Unternehmen durchführt. In Abhängigkeit der Betriebsgröße schickt das Finanzamt mit unterschiedlicher Häufigkeit den Betriebsprüfer ins Haus. Große Unternehmen werden fast ständig untersucht, kleine Unternehmen im Schnitt nur alle 50 Jahre. Anhand der Ergeb-

nisse (Höhe der Steuernachzahlung) und der Analyse der Daten aus den vergangenen Betriebsprüfungen kann die Finanzverwaltung die Kandidaten ermitteln, bei denen die höchste Steuerrückzahlung zu erwarten ist. Die knappe Ressource Betriebsprüfer kann damit für die lohnenden Fälle eingesetzt werden.

6.2.14 Absatzprognose zur Zolloptimierung

Produzierende Unternehmen mit weltweit verteilten Produktionsstätten und Absatzmärkten stehen vor der Herausforderung, dass in Abhängigkeit des Produktionsortes und des Absatzmarktes unterschiedliche Kosten für Importzölle entstehen können, die naheliegender Weise minimiert werden sollen. Ein Schmuckhersteller z. B. kann das gleiche Produkt in seiner asiatischen oder aber in der europäischen Produktionsstätte herstellen. Wird ein Produkt z. B. in Hong Kong verkauft, fällt für das in Asien produzierte Produkt kein Importzoll an. Das identische Produkt aus europäischer Produktion würde zu hohen Zollgebühren führen.

Durch die Prognose der Absatzzahlen in den unterschiedlichen Absatzmärkten und die darauf abgestimmte, bedarfsgerechte Produktion der Produkte an der ‘richtigen’ Stelle, kann ein Unternehmen Zollgebühren in Millionenhöhe einsparen.

Dazu werden die historischen Daten der Verkäufe, aber auch externe Daten und Erfahrungswissen für neue Kollektionen analysiert, um damit für die entsprechenden Produkte eine Prognose erstellen zu können.

6.2.15 Social Media Monitoring – Sentiment Analysis

Unternehmen sind mit ihren Marketingaktivitäten zunehmend in sozialen Medien unterwegs. Einerseits mit aktiver Kommunikation, andererseits als Beobachter. Durch die Sentiment-Analyse von z. B. Twitter versuchen sie mit ihrem ‘Ohr am Kunden zu bleiben’. Es kann nicht jeder Tweet, der sich mit einem Unternehmen beschäftigt, von einem Mitarbeiter gelesen und eventuell beantwortet werden. Dazu ist die schiere Menge einfach zu groß. Was aber gemacht wird, ist, dass alle Tweets, die Unternehmens-, Produkt- oder Management-Namen enthalten, gesammelt und einer automatischen Textanalyse unterzogen werden, mit dem Ziel, eine geäußerte Haltung z. B. als positiv oder negativ zu erkennen und sie dann über den Zeitverlauf ins Verhältnis zu besonderen Ereignissen zu stellen.

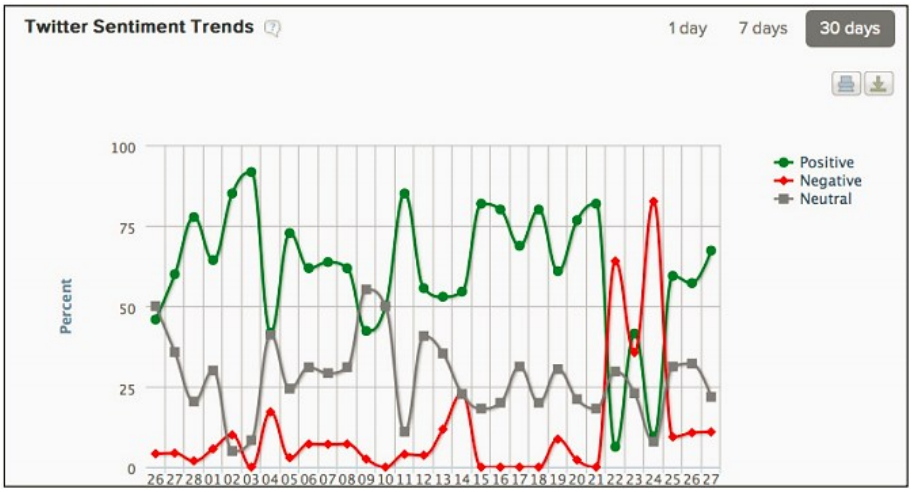


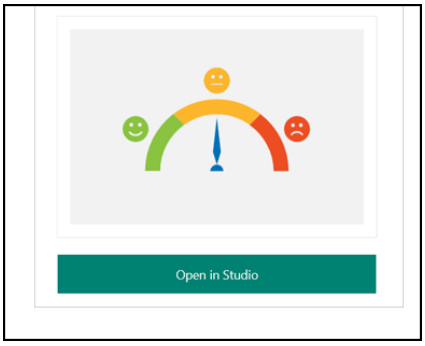
Abbildung 39: www.cision.com/us/resources/white-papers/understanding-social-media-sentiment/

Es werden dabei verschiedene Text-Mining-Verfahren eingesetzt. Im ersten Schritt wird anhand der Analyse von gelabelten Texten (also Texte, bei denen man z. B. weiß, ob sie positiv oder negativ sind) gelernt, wie die Klassifizierung stattzufinden hat. Danach werden die Texte anhand des Modells entspre-

chend zugeordnet. Die Modelle beachten meist das Vorkommen von Begriffen, Kombinationen von Begriffen und den Abstand von Begriffen in Sätzen und setzen diese mit bestimmten Sentiments in Verbindung. Aus den 'unstrukturierten' Texten werden dadurch strukturierte Daten, die dann mit weiteren quantitativen Verfahren analysiert werden können.

6.2.16 Analyse von Streaming-Daten

Ein Sonderfall der hier aufgeführten Use Cases stellt das Thema Echtzeitanalyse von Streaming-Daten dar. Anwendungsfälle sind z. B.:



- Die Stimmungsanalyse von Social Media in Echtzeit. Dabei werden veröffentlichte Texte in quasi Realtime analysiert und zu einem Stimmungsindex zusammengefasst. Microsoft bietet mit Azure Stream Analytics ein entsprechendes Angebot, das als Webdienst innerhalb von Azure auf die entsprechenden Datenquellen angewendet werden kann.
- Finanztransaktionen von Börsen oder Banken werden in Echtzeit auf Geldwäsche- oder Betrugsaktivitäten hin analysiert.
- Sensordaten von Maschinen, Fahrzeugen oder Anlagen werden in Realtime ausgewertet, um entsprechende Maßnahmen einleiten zu können.

Diese Anwendungsfälle unterscheiden sich aber von den vorherigen Beispielen, da es sich bei Streaming Analytics lediglich um die Anwendung eines vorher erstellten Modells auf Streaming-Daten handelt. Die eigentliche Modellerstellung erfolgt dabei nicht in Echtzeit mit Streaming-Daten, sondern ganz konventionell mit statischen Daten in der Modellbildungsphase. Das

Deployment eines Modells in Systeme mit Streaming-Daten ist also grundsätzlich bei allen Anwendungsfällen denkbar und stellt daher keine eigene Kategorie von Analytics-Fällen dar.

6.2.17 Bilderkennung – Arbeitssicherheit

Die Bilderkennung ist ein wichtiges Einsatzgebiet des Machine Learnings bzw. der künstlichen Intelligenz. Zum überwiegenden Teil liegen den Systemen Modelle auf Basis von neuronalen Netzen zugrunde, die auf das Erkennen von Inhalten in Bildern trainiert wurden. Ein mögliches Einsatzgebiet ist z. B. die Überwachung von Fertigungsprozessen. Über Kameras werden Werkstätten überwacht. Die Bilderkennungslogik erkennt Personen, Werkzeuge, Maschinen, Produkte und Arbeitsgänge. Es kann somit z. B. sichergestellt werden, dass sich nicht unbefugte Personen in einem sensiblen Arbeitsbereich aufhalten, bestimmte Werkzeuge von Personen genutzt werden, die dafür keine Ausbildung haben oder Arbeitsgänge in falschen Reihenfolgen durchgeführt werden. Das System kann entsprechenden Alarm auslösen oder Sofortmaßnahmen veranlassen.[18]

Dieses Einsatzszenario ruft natürlich ‘Big Brother’-Assoziationen hervor: Die Maschine, die den Menschen ununterbrochen überwacht und gegebenenfalls korrigiert. Einsätze im Bereich Arbeitsschutz sind – zumindest in Europa – unter diesen Vorbehalten sicher nur in sehr engen Grenzen denkbar.

[18] Quelle: https://www.digitaltrends.com/computing/microsoft-build-2017-first-keynote-covered-ai-cloud-cortana/

6.2.18 Künstliche Intelligenz für die Malware-Erkennung

Konventionelle Sicherheitssoftware wie Antivieren-Programme oder Firewalls sind signaturbasiert, d. h. die Abwehr von Gefahren erfolgt anhand einer Liste (Signatur) von bekannten Gefahren. Da man damit aber keine neuen Angriffsarten (sog. Zero Day Threats) erkennen kann, gehen Sicherheitsfirmen darauf über, mithilfe von Machine Learning Muster zu erkennen, die auch unbekannte Angriffsarten aufdecken. Die Modelle, die in diesem Rahmen entwickelt werden, werden in den entsprechenden Produkten 'verpackt', aber nicht offengelegt. Sie stellen die Intellectual Property für die Sicherheitsunternehmen dar und werden daher als Betriebsgeheimnis gehütet. Die Erstellung der entsprechenden Modelle in den Softwareunternehmen entspricht dem üblichen Ablauf der Datenanalyse und Modellerstellung. Der Einsatz der Software in den Anwendungsunternehmen hat dann mit der Arbeit eines Data Scientisten nichts mehr zu tun, da ja nur ein 'fertiges' Produkt eingesetzt wird.

6.2.19 Autonomes Fahren

Eine der Königsdisziplinen des Einsatzes von Machine Learning-Methoden und künstlicher Intelligenz ist das autonome Fahren. Autonome Fahrzeuge erkennen Verkehrsschilder, halten Abstand zu anderen Fahrzeugen, bremsen vor Hindernissen rechtzeitig und finden ihren Weg zum Ziel ohne menschliches Zutun. Dabei müssen sie auch mit unvorhergesehenen Situationen umgehen können. Die entsprechenden Machine Learning-Modelle müssen in der Lage sein,

- in Echtzeit große Datenmengen zu verarbeiten, auszuwerten und dann die entsprechenden Entscheidungen zu treffen,
- ständig dazuzulernen,
- in Abstimmung mit anderen Modellen und den Fahrzeugsystemen zu arbeiten und
- Ausfallsicherheit und Fallbackszenarien zu bieten.

- Sie müssen zudem eine extrem hohe Zuverlässigkeit aufweisen. Eine Accuracy von 99,9 Prozent bedeutet beispielsweise einen Fehler pro tausend Fälle. Der durchschnittliche Fahrzeuginsasse eines autonom fahrenden PKWs dürfte wahrscheinlich nicht glücklich damit sein, alle 1000 Sekunden an einem Autounfall zu sterben. Das bedeutet nicht, dass jedes der eingesetzten Machine Learning-Modell eine (in der Höhe unmögliche) Zuverlässigkeit aufweisen muss. Aber das Gesamtsystem Auto muss in Kombination aller Systeme eine extrem hohe Zuverlässigkeit bieten. Dies lässt sich wie folgt veranschaulichen: Auch wenn die Bilderkennung einen über die Straße rennenden Fuchs nicht als Fuchs erkennt, muss die Kombination aus Radar-, Video-, Laser- und Fahrerfahrungssystemen zur richtigen Entscheidung *Bremsen und nach rechts ausweichen*, oder bei Gegenverkehr an einer Bergstraße am Abhang zu *Spurhalten, Hindernis überfahren und langsam abbremsen* führen.

Zahlreiche Fragen müssen beim autonomen Fahren geklärt werden. Als Beispiel seien folgende genannt:

Wo soll das Weiterlernen der Modelle stattfinden – im Auto oder an einem zentralen Cloud-Rechner? Wie findet der Abgleich der Modelle statt? Soll ein Austausch mit anderen Fahrzeugen stattfinden, und wenn ja, wie? Wer hat die Schuld bei Unfällen? Wie soll bei moralischen Fragen entschieden werden? Es kommt ein 20 Tonnen Lastwagen mit hoher Geschwindigkeit ohne Ausweichmöglichkeit entgegen. Soll das Leben der schwangeren Insassen gerettet werden, indem der Wagen in die Gruppe Rentner am Straßenrand ausweicht? Wie viele Rentner dürfen es höchstens sein? Wie alt müssen die Rentner mindestens sein? Was passiert, wenn ein Rentner sein Enkelkind dabeihat? Gibt es dann also einen Aufrechnungsalgorithmus, der die Wertigkeit aufrechnet, oder sollte vielleicht das menschliche Verhalten des Fahrers nachgebildet werden, indem eine Zufallsentscheidung unter extremen Stress nachgeahmt wird...?

Es gibt auch profanere Fragen nach der Datenhoheit. Sollen Dritte Zugang zu den Daten aus den Fahrzeugen bekommen? Ist noch ein TÜV erforderlich, wenn die OEM alle notwendigen Daten zum Thema Sicherheit ohnehin schon haben? Werden freie Werkstätten vom Markt verschwinden, da der Autohersteller ein defektes Fahrzeug schon in die Werkstatt gelenkt hat?

Mit der kurzen Darstellung einiger Aspekte des Themas autonomes Fahren wurde deutlich, dass es weit über einen einfachen Data-Science-Use-Case hinausgeht.

6.2.20 Datenanalyse bei einer Pandemie

Pandemien sind BI-Zeiten, d. h. Zeiten, in denen vor allem ein Bedarf an deskriptiven Datenanalysen besteht. Jeder hat wahrscheinlich während der Corona-Krise täglich die entsprechenden Dashboards des RKI oder der Hopkins Universität betrachtet. Aber es gibt auch genügend Anwendungsfälle für prediktive Analytics. Die Verbreitung von Viren erfolgt gemäß einer logarithmischen Kurve, die zuerst exponentiell ansteigt und sich dann langsam einer Obergrenze annähert. Ohne vorhandene Impfstoffe und eingreifende Maßnahmen, die die Verbreitung einschränken (z. B. Ausgangsbeschränkungen), entspricht diese Obergrenze der Herdenimmunität, die im schlimmsten Fall bei ca. 60 Prozent der Bevölkerung liegt. Im Falle der Corona-Epidemie musste für ein Prognosemodell also der idealtypische Verlauf (der ein absolutes Worst-Case-Szenario für das Gesundheitssystem und die Anzahl der Todesfälle bedeutet hätte) auf eine deutlich niedrigere Obergrenze gedeckelt werden. Die Prognose stellt also eine mathematische Kurve mit *manuellem* Eingriff anhand von Annahmen bzw. Expertenwissen dar.

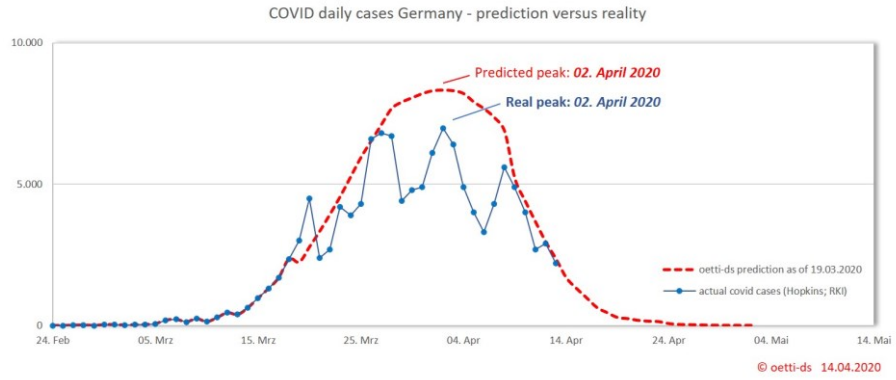


Abbildung 40: Prognose von oetti-ds zur Ausbreitung von Corona

Zu Beginn der massiven Ausbreitung des Corona-Virus hat der Autor eine Prognose des Verlaufs der täglichen Neuerkrankungen veröffentlicht. Die prognostizierte Kurve verlief zwar insgesamt etwas flacher als der tatsächliche Verlauf, aber der *peak* der neu gemeldeten Fälle wurde auf den Tag genau vorhergesagt.

6.3 Use Cases für genAI

Im Unterschied zu den Use-Cases im vorangegangenen Abschnitt folgen nun beispielhafte Anwendungsfälle in Unternehmen für genAI Tools. Die Abgrenzung ist nicht immer eindeutig, aber der Unterschied kann wie folgt erklärt werden:

- Abschnitt 6.2 zeigt Anwendungsfälle, deren Ziel es ist durch Analyse von vorhandenen Daten *selbst* ein Machine Learning / AI Modelle zu erstellen, die dann für den beschrieben Einsatz genutzt werden können.
- Hier in Abschnitt 6.3 folgen Beispiele, wie *bestehende* genAI Tools im Unternehmensumfeld eingesetzt werden können.

Zur Gliederung der insgesamt 23 Use-Cases wurden zwei Ansätze verwendet.

Einerseits kann eine Zuordnung der (nummerierten und im Anschluss kurz skizzierten) Anwendungsfälle an Funktionen in der unternehmerischen **Wertschöpfungskette** erfolgen. Hierzu werden die Fälle aus der Sicht des Unternehmens nach dem Bereich der Wertschöpfungskette geordnet.

In einem anderen Ansatz erfolgt die Zuordnung anhand der sog. **Produktkontaktkette** aus der Sicht des Kunden. Die Produktkontaktkette ist ein konsequent kundenzentrierter Ansatz, der bei oetti-ds entwickelt wurde, um ergänzendes Servicepotenzial um ein Kernprodukt herum zu ermitteln. Zusammengefasst kann die Grundidee wie folgt beschrieben werden:

- Ein Produkt (oder eine Dienstleistung) hat den Zweck, ein Problem des Kunden zu lösen.
- Im Zusammenhang mit dem Produkt können für den Kunden jedoch ‘Nebenprobleme‘ entstehen. Dies kann bereits vor der Produktnutzung beginnen und erst nach der Nutzung enden. Beispielsweise verursacht ein PKW das Nebenproblem der Finanzierung oder der Entsorgung bzw. des Weiterverkaufs nach der Nutzung. Produkthersteller können dafür Services anbieten, um die gesamte Produktkontaktzeit für den Kunden ‘problemloser‘ zu gestalten. Selbst zu Produkten mit geringer Komplexität kann mit dieser Sichtweise Dienstleistungspotenzial ermittelt werden. Das Produkt Brötchen kann am Sonntagvormittag das Nebenproblem ‘Anreise zum Vertragsabschluss‘ erzeugen. Ein Bäcker, der einen Lieferservice anbietet, hilft, dieses Nebenproblem zu lösen (und kann damit mehr Brötchen verkaufen).
- Unternehmen können nun für Produkte die Bereiche in der Produktkontaktkette identifizieren, die relevant sind. Anschließend kann systematisch überlegt werden, wo Potenzial für zusätzliche produktergänzende Serviceleistungen liegen. Dabei können auch Beispiele aus anderen Branchen als Inspiration dienen.

Die grundsätzliche Idee der Produktkontaktkette kann aber auch verwendet werden, um Ansätze für den Einsatz von Werkzeugen aus dem Bereich Generative Artificial Intelligence zu finden.

Einige der folgenden Use-Cases wurden deshalb der entsprechenden Stelle in der Produktkontaktkette zugeordnet.

(1) Softwareentwicklung: Generative Artificial Intelligence kann unterstützend genutzt werden, um Softwarecode zu schreiben, zu vervollständigen und zu überprüfen. Das gesamte Programmierung anhand der Eingabe natürlicher Sprache ist auf diesem Weg nur in einfachen Fällen möglich, da die natürliche Sprache zu unpräzise ist. Dennoch ergeben sich ein signifikanter Produktivitätsgewinn aus der Automatisierung von Standardprogrammiertätigkeiten und eine Steigerung der Qualität, wenn generative KI-Modelle Fehlerbehebungen, Testgenerierung und Dokumentationsfunktionen übernehmen.

(2) Produktentwicklung: Im Bereich Produktentwicklung bzw. Wartung und Weiterentwicklung von Produkten können die Tools genutzt werden, um Daten aus mehreren Quellen auszuwerten, zu konsolidieren und erzählerisch in leicht verständliche Empfehlungen umzuwandeln.

(3) Produktinnovation, Ideenfindung und Prototyping: Mit Generative AI können durch neue Ideen originelle Produkte, Dienstleistungen oder Erlebnisse entworfen werden, die den Bedürfnissen und Vorlieben der Kunden entsprechen. Assistenzprogramme aus diesem Bereich können auf der Grundlage

des bestehenden Portfolios und des Marktes Ideen für innovative Produkte und Dienstleistungen generieren. Visualisierungen oder Prototypen dieser Produktideen erleichtern die Auswahl und die Verbesserung relevanter Ideen, sodass der Innovationsprozess beschleunigt wird.

(4) Arzneimittelforschung: KI-Technologie wird eingesetzt, um die Entwicklung neuer Arzneimittel effizienter zu gestalten und damit auch die Möglichkeit einer Individualisierung von Medikamenten zu erreichen. Mit dieser neuen Entwicklung beginnen Wissenschaftler, neuartige Moleküle zu erzeugen, ungeordnete Proteine effektiver zu entdecken und Ergebnisse klinischer Studien vorherzusagen. Generative AI beschleunigt den Arzneimittelentwicklungsprozess durch die Optimierung molekularer Strukturen. Die Algorithmen analysieren umfangreiche Datensätze zu chemischen Verbindungen, sagen molekulare Konfigurationen voraus und schlagen optimierte Strukturen für potenzielle Medikamente vor. Dieser Anwendungsfall beschleunigt die frühen Phasen der Arzneimittelentwicklung erheblich.

(5) Business-Intelligence und Reporting: Da Generative AI riesige Text- und Datenmengen schnell auf die wichtigsten Punkte zusammenfassen kann, stellt sie ein nützliches Werkzeug für Business-Intelligence und Leistungsberichte dar. Dies ist besonders hilfreich für unstrukturierte und qualitative Datenanalysen, da die hierfür verwendeten Informationen normalerweise mehr Verarbeitung erfordern, bevor Erkenntnisse gewonnen werden können. Es bestehen Ansätze, in denen die kontextualisierte Interpretation von Daten mithilfe von KI-Methoden erfolgt. Dies geht über typische Visualisierungen und Dashboards hinaus und umfasst Erklärungen in natürlicher Sprache.

(6) Automatisierte Kreditentscheidung: Die Entscheidung über die Gewährung von Krediten kann über Werkzeuge aus dem Bereich Generative AI unterstützt werden. Ebenso kann eine verbale Begründung für die positive oder negative Entscheidung generiert werden.

(7) Projektmanagement: Projektmanagementsoftware mit integrierter genAI-Komponente kann Benutzer bei der Aufgaben- und Unteraufgabengenerierung, bei der Dokumentation bis hin zur Projektrisikovorhersage unterstützen. Sie kann dabei helfen, Assets wie Dokumente und Datensätze zu verwalten und zusammenzufassen, sodass sowohl interne Ressourcen als auch vom Kunden übermittelte Informationen effizienter verarbeitet und auf Projekte angewendet werden können.

(8) Dokumentation von Meetings: Powerpoint-Präsentationen für Besprechungen werden durch genAI-Tools schneller erstellt. Ebenso können strukturierte Besprechungsprotokolle aus Stichpunkten automatisiert erzeugt werden.

(9) Wissensmanagement - Retrieval-Augmented Generation: Large Language-Models werden anhand öffentlich zugänglicher Daten und Texte trainiert. Bei Retrieval-augmented Generation (RAG) wird die Ausgabe des Modells optimiert, indem eine Wissensbasis außerhalb der Trainingsdatenquellen einbezogen wird, bevor eine Antwort generiert wird. RAG erweitert somit die bereits leistungsstarken Funktionen der Large Language-Models auf ausgewählte Domains oder die interne Wissensbasis einer Organisation, ohne dass das Modell neu trainiert werden muss.

(10) Individuelle (Produkt-)Schulung: Trainingsmaterial (z. B. Dokumente oder Videos) können mit genAI-Werkzeugen schneller und individualisierter

erzeugt und an den Kontext (vorhandene Informationen über den Nutzer) angepasst werden. Dies ist sowohl für die interne Schulung der Mitarbeiter als auch für die Anleitung der Kunden in Bezug auf die Nutzung des Produktes möglich.

(11) Performance-Management und Coaching: Generative AI kann im Bereich Mitarbeiterführung eingesetzt werden. Beispielsweise liefert die Dokumentation und Zusammenfassung von Contact-Center-Anrufen in Kombination mit einer Stimmungsanalyse den Vorgesetzten die Informationen, die sie benötigen, um aktuelle Anrufe von Kundendienstmitarbeitern zu bewerten und die Arbeitskräfte hinsichtlich Verbesserungsmöglichkeiten zu coachen.

(12) Supply-Chain-Management: GenAI-Tools können z. B. die Dokumentation zu Lieferantenverträgen analysieren und entscheidende Bedingungen, Konditionen sowie Leistungskennzahlen identifizieren. Diese Informationen können Unternehmen dabei helfen, die Leistung ihrer Lieferanten zu bewerten, bessere Konditionen auszuhandeln und potenzielle Risiken oder Engpässe in der Lieferkette zu identifizieren.

(13) Medizinische Diagnostik: Werkzeuge zur Bilderzeugung und -bearbeitung werden zur Optimierung und Vergrößerung medizinischer Bilder eingesetzt, sodass Mediziner den menschlichen Körper besser und realistischer betrachten können. Einige Tools führen sogar selbstständig medizinische Bildanalysen und grundlegende Diagnosen durch. Bei Hautkrebsscreenings werden z. B. verdächtige Bereiche automatisch markiert, die von der medizinischen Fachkraft anschließend genauer untersucht werden können.

(14) Vorausschauende Wartung von Produktionsmaschinen: GenAI hilft produzierenden Unternehmen, den Betrieb zu optimieren, indem z. B. Sensordaten von Geräten und Maschinen interpretiert werden. Dadurch können ungeplante Ausfallzeiten reduziert, die Betriebseffizienz gesteigert und die Auslastung maximiert werden. Wenn ein Problem erkannt wird, können die Tools mögliche Lösungen und einen Serviceplan empfehlen, um Wartungsteams bei der Behebung des Problems zu unterstützen. Fertigungsingenieure können über natürliche Sprache und allgemeine Anfragen mit dieser Technologie interagieren.

(15) Erstellen von Social-Media-Inhalten: LLMs sind in der Lage, geeignete und kreative Inhalte für Blogs, Social-Media-Beiträge, Produktseiten und Unternehmenswebsites zu erstellen. Bestehende Inhalte können mit generativen Tools geändert, gekürzt oder erweitert werden. Auch die Erstellung völlig neuer Inhalte ist möglich.

(16) Inbound- und Outbound-Marketing: Inbound- und Outbound-Marketingkampagnen erfordern häufig, dass Mitarbeiter täglich kontextbezogene E-Mails und Chat-Threads an potenzielle und bestehende Kunden senden. GenAI-Lösungen können genutzt werden, um die Inhalte für diese Kommunikation zu erstellen und zu versenden.

(17) Grafikdesign und Videomarketing: GenAI ist in der Lage, realistische Bilder, Animationen und Audiodaten zu erzeugen, die für Grafikdesign- und Videomarketingprojekte verwendet werden können. Einige Angebote umfassen auch Sprachsynthese und KI-Avatare an, sodass die Kunden Marketingvideos ohne Schauspieler, Videoausrüstung oder Videobearbeitungskenntnisse erstellen können.

(18) Erstellen von Produktbeschreibungen: Produktbeschreibungen für Prospekte oder Webshops können anhand von Stichworten und formlosen Beschreibungen automatisch erzeugt sowie beliebig übersetzt werden.

(19) Erstellung von Produktbildern: GenAI-Tools können aus Textbeschreibungen realistische Produktbilder erzeugen. So können z. B. im Rahmen eines individuellen Produktkonfigurators Bilder für verschiedene Produktvarianten erstellt werden.

(20) Intelligente Produktsuche: GenAI kann intelligente und benutzerfreundliche Produktsuchmaschinen unterstützen. Durch das Verständnis der Bedeutung hinter Kundensuchanfragen werden relevantere und präzisere Ergebnisse präsentiert.

(21) Personalisierung des Kundenerlebnisses: Einer der wirkungsvollsten Anwendungsfälle für genAI ist die Personalisierung des Kundenerlebnisses. Kundenpräferenzen werden anhand vorhandener Daten verstanden und genutzt, um relevante Inhalte und Produktvorschläge zu erstellen.

(22) Betrugserkennung bei Kundenbewertungen: GenAI kann Textdaten wie Kundenbewertungen, E-Mails und Finanztransaktionen analysieren, um verdächtige Muster, betrügerische Aktivitäten und potenzielle Risiken aufzudecken. Die Effektivität dieser Tools bei der Betrugserkennung liegt in der Fähigkeit, große Mengen an Textdaten schnell und genau zu verarbeiten sowie zu interpretieren. Durch die Analyse der in Kundenbewertungen verwendeten Sprache kann genAI beispielsweise Muster betrügerischer Bewertungen oder gefälschtes Feedback erkennen.

(23) Sentiment-Analysis im Kundenservice: Interaktionen mit Kunden können in Echtzeit oder im Nachhinein durch genAI einer Stimmungsanalyse unterzogen werden. Die Stimmung des Kunden kann durch Sprachanalyse oder Bilderkennung ermittelt werden, sodass die Reaktion des Kundenservices entsprechend angepasst werden kann.

(24) Kundenbetreuung und Kundenservice: Standardkundenanfragen können durch genAI-Chatbots und virtuelle Assistenten rund um die Uhr bearbeitet werden. (Meist nur mäßig funktionierende) Chatbots werden seit über 10 Jahren für den Kundenservice eingesetzt, jedoch gibt die Weiterentwicklung von genAI Hoffnung auf eine Qualitätssteigerung dieser Prozesse. Gleichzeitig ermöglicht sie es, die vorhandenen Ressourcen von den Standardaufgaben zu entlasten und für die individuelle Fallbearbeitung freizusetzen. Anstatt sich auf vordefinierte Skripte zu verlassen, generieren KI-Modelle Antworten in Echtzeit basierend auf den Benutzeranfragen und dem Kontext. Diese Anpassungsfähigkeit erhöht die Vielseitigkeit virtueller Assistenten und versetzt sie in die Lage, ein breites Spektrum an Benutzerinteraktionen zu bewältigen.

(25) Nutzungsassistent: Produkte können mit Assistenten ausgestattet werden, die anhand der bei der Nutzung des Produktes anfallenden Daten Hinweise auf die korrekte Anwendung und notwendige Wartungsarbeiten oder Warnungen bei fehlerhafter Nutzung ausgeben. Automatisiert kann auch der Kontakt zum Hersteller aufgenommen werden.

(26) End-of-Life-Assistent: Produkte können mit Assistenten ausgestattet werden, die anhand der bei der Nutzung des Produktes anfallenden Daten Hinweise auf das Ende der Lebensdauer des Produktes geben. Dazu könnten sie z. B. Empfehlungen für eine Ersatzbeschaffung geben oder Kontakt zu anderen Nutzern oder dem Hersteller aufbauen.

7 Abschluss

Zum Abschluss des Buches sollen noch einmal einige grundsätzliche Themen beleuchtet werden. Das Buch war als Einführung in und als Übersicht über das weitumfassende Themengebiet Data-Science gedacht. Allzu sehr in die Tiefe konnte daher nicht gegangen werden. Jedes der vorgestellten Verfahren ist umfassend genug für ein eigenes Buch und die Data-Science-Softwareplattformen könnten auf Tausenden von Dokumentations- und Trainingsseiten behandelt werden. Programmierleitfäden und Bibliotheksdokumentationen zu R oder Python würden ausgedruckt ganze Schränke füllen. Die spannendsten Erkenntnisse liegen in den Details der Use-Cases, die hier nicht aufgezeigt werden konnten.

Es ist jedoch nicht der Anspruch des Buches, alle Themen abschließend zu behandeln. Stattdessen sollte ein grundsätzliches Verständnis für dieses spannende und abwechslungsreiche Feld aufgebaut und ein Einstieg zur weiterführenden Vertiefung gegeben werden.

Folgende Punkte sollen an dieser Stelle noch einmal zusammenfassend erwähnt werden:

- **Gespür für die Verfahren**. Mit den in Kapitel 4.5 aufgeführten Beschreibungen der Verfahren soll ein ‘Gefühl‘ für die Möglichkeiten und Grenzen der Verfahren vermittelt werden. Meine persönliche Einstellung zu den Methoden ist etwas zwiespältig. Eigentlich sind sie im Kern recht einfach. Meist werden lediglich willkürliche quadrierten Distanzen aufsummiert und anschließend wird versucht, diese Summe zu minimieren. Auf der anderen Seite steckt statistisches bzw. mathematisches Know-how in den Verfahren und Algorithmen. Allzu leichtfertig sollte daher nicht über die statistisch-mathematischen Grundlagen hinweggesehen werden. Die Softwarepakete machen die Anwendung so einfach, dass die theoretischen Grundlagen schnell in den Hintergrund treten. Nur durch ein grundsätzliches Verständnis

der Funktionsweise und des Zweckes der Verfahren kann man die statistischen Details und Kennzahlen einordnen sowie sein Wissen über die Details vertiefen.

- **Grenzen und Grenzenlosigkeit verstehen**: Die Verfahren sind oft banale Mathematik, mit ein bisschen Heuristik kombiniert. Ein künstliches neuronales Netzwerk mit mehreren Tausend Hidden Layers ist meilenweit von den 80 bis 100 Milliarden Neuronen eines echten Gehirns entfernt. Folglich können damit auch nicht die gesamten kognitiven Fähigkeiten eines Menschen nachgebildet werden. Auch darf man nicht vergessen, dass jedes echte Gehirn jahrelang trainiert wird, bevor seine Fähigkeiten vollends entwickelt sind. Alle Eltern können nachvollziehen, wie langwierig und anstrengend dieses ‘überwachte‘ Lernen tatsächlich ist. Daher ist nicht zu erwarten, dass echte menschliche Kognition bald vollständig durch Maschinen ersetzt wird. Diesen Einschränkungen stehen die fast grenzenlosen Einsatzgebiete von Machine Learning und genAI gegenüber. Der Computer ist in manchen Bereichen dem Menschen so überlegen, dass es geradezu fahrlässig ist, diese Fähigkeiten nicht einzusetzen. Ein Muster in einer Tabelle mit zweihundert Spalten (Variablen) für 50 000 Datensätze zu erkennen, ist eine für Menschen unlösbare Aufgabe. Das Gleiche gilt für die Berechnung von 15 Millionen Prognosemodellen für den Verkauf von Frischeprodukten in Supermärkten (s. Kapitel 6.2.6). Die noch nicht ausgeschöpften Potenziale der Datenanalyse vor dem Hintergrund der Big-Data-Entwicklung sind praktisch grenzenlos.

- **No Excuse**: Es gibt keine Entschuldigung dafür, nicht morgen schon mit Machine Learning-Projekten zu beginnen. Für jeden Geschmack und jeden Wissenshintergrund existieren die entsprechenden Data-Science-Werkzeuge. Ob es sich um ein grafikorientiertes Klickprogramm, an Kommandozeilen orientiertes Scripting oder flexible Programmiersprachen, die Anwendung auf dem Notebook, einem Server oder aus der Cloud handelt: Die meisten Programme sind in der Zeit,

in der Sie diese eine Seite lesen, schon heruntergeladen und auf einem Rechner installiert. Die Arbeit kann beginnen.

- **Mut zur Lücke**: Man muss ein pragmatisches Gefühl dafür aufbauen, wann Mut zur Lücke angemessen ist. Denn viele Verfahren können strenggenommen nie angewendet werden, da die inhärenten Annahmen und Voraussetzungen (z. B. über die Normalverteilung einer Zufallsvariable) meist nicht erfüllt sind. Manche Daten sind nicht in der erforderlichen Qualität verfügbar oder Daten aus unterschiedlichen Quellen sind nicht vollständig vergleichbar. Einem traditionellen Statistiker dreht sich der Magen um, wenn er einem unbekümmerten Data-Scientist dabei zusieht, wie dieser an den Parametern einer nichtlinearen Regression 'herumschraubt', bis er mit dem Ergebnis zufrieden ist (ohne den Unterschied zwischen einem F-Test und einem T-Test zu kennen oder, schlimmer noch, Korrelation und Kausalität nicht unterscheiden zu können). Dennoch kann ein Modell unter diesen Voraussetzungen funktionieren. Man muss eine gewisse 'Fuzziness' zulassen, sofern man sich ihrer bewusst ist. Das altbekannte Motto sei an dieser Stelle nochmal wiederholt: „Die meisten Modelle sind falsch, aber einige funktionieren."

- **Fantasie bezüglich der Datenquellen**: Oft ist die Auswahl der zu analysierenden Daten wichtiger als die Optimierung des hundertsten Parameter des Verfahrens. Zu Beginn des Analyseprojektes wird überlegt, welche Daten miteinbezogen werden sollen. Dies ist ein entscheidender Erfolgsfaktor für das Projekt. Denn wenn bei der Verkaufsprognose für einen Supermarkt Wetterdaten einbezogen werden oder der Abstand der Regalposition im Vergleich zum Konkurrenzprodukt erhoben wird, kann eine andere Qualität an Erkenntnissen gewonnen werden als ohne diese Daten. Die Empfehlung lautet daher, sich in diesem ersten Prozessschritt Zeit zu nehmen und der Fantasie freien Lauf zu lassen.

- **Datenschutz ernst nehmen**: Belange des Datenschutzes und der Wahrung der Persönlichkeitsrechte sind ernst zu nehmen. Das betrifft zwei Aspekte. Einerseits signalisiert es schlichtweg mangelnden Respekt gegenüber den entsprechenden Abteilungen und Kollegen im Unternehmen, wenn das Thema Datenschutz nicht berücksichtigt wird. Ein Data-Scientist mag stolz darauf sein, dass er ein Modell entwickelt hat, das den Mitarbeiter-Churn anhand von Daten wie Pausen-, Telefon- und Chatzeiten prognostizieren kann. Die Personal- und Datenbeauftragten werden hierauf jedoch zu Recht empört reagieren, wenn sie davon erfahren. Es führt auch zu einer Blockadehaltung, wenn die entsprechenden Kollegen nicht von Anfang an einbezogen werden. Man kann ja durchaus mit ‘offenem Visier‘ diskutieren. Wenn unüberbrückbare Unterschiede in den Auffassungen bestehen, muss schließlich das Management entscheiden. Entscheidend ist die Berücksichtigung des Datenschutzes von Anfang an, sodass es nicht zwei Tage vor dem Go-Live-Termin zu einem unerwünschten ‘Showstopper‘ kommt.

  Der zweite Aspekt des Datenschutzes ist die sachliche Notwendigkeit einer kritischen Überprüfung. Nicht alles, was möglich ist und dem Fachmann gefallen würde, ist richtig. Amerikaner gehen damit vielleicht anders um, wenn Google die Inhalte von Gmail-Konten analysiert und mit Daten aus anderen Google-Diensten kombiniert. Europäer und insbesondere Deutsche sind diesbezüglich jedoch weitaus sensibler und verzichten lieber auf Bequemlichkeit zugunsten von Datenschutz bzw. Datensouveränität.

- **Einfach machen**: Das schlägt nochmal in die Kerbe des Punktes weiter oben. Mehr machen, weniger planen und absichern! Es ist sinnvoller zwanzig Projekte durchzuführen, von denen zwar acht schiefgehen, aber zwölf sehr wertvolle Erkenntnisse zu gewinnen, als in der gleichen Zeit nur ein perfekt geplantes Projekt durchzuführen. Bild-

lich gesprochen: Der Prozessor der Analyseplattform muss heiß laufen und nicht die Kaffeemaschine im Besprechungsraum für das fünfzigste Vorbereitungsmeeting. Let’s start!

8 Informationsquellen

Das Web ist voll mit Informationen zum Thema. Eigentlich findet man alles, was man wissen muss, im Netz. Nützlich für mich haben sich die folgenden Seiten erwiesen, da sie helfen, einen strukturierten Zugang zu Wissen und aktuellen Diskussionen zu bekommen:

- **KDnuggets**: Die Großmutter aller Informationsseiten. Man erkennt schon am Namen, dass die Seite schon existierte, als Data Science noch KD = Knowledge Discovery in Databases hieß. Aber dennoch immer noch jung und aktuell und eine wichtige Informationsquelle.
  www.kdnuggets.com

- **Date Science Central**: Eine Online-Plattform für Big-Data-Praktiker. Viele Blogbeiträge, Webinare und Wissenszusammenfassungen.
  www.datasciencecentral.com

- **kaggle**: Eine Plattform für Analytics-Wettbewerbe. Unternehmen, Organisationen oder private Mitglieder können Wettbewerbe ausrufen. Eine interessante Lernplattform und wertvolle Quelle für Datensätze. Die Lösungsansätze der anderen Mitglieder können eingesehen werden. Google hat kaggle 2017 akquiriert.
  www.kaggle.com

- **User Groups in LinkedIn und XING**: Die entsprechenden Gruppen innerhalb der sozialen Berufsnetze eignen sich zum Netzwerken und um sich über aktuelle Trends zu informieren. Viele Beiträge sind zwar werblicher Natur, aber man findet immer wieder interessante Posts von Usern. Über E-Mail-Benachrichtigungen kann man sich nach Wunsch auf dem Laufenden halten.

  LinkedIn: z. B. ‘Big Data and Analytics’; ‘Big Data Analytics on Hadoop’; Xing: ‘Data Science Germany’; ‘Predictive Analytics and Big Data’

- **Towards data science**: Plattform für die Veröffentlichung von Beiträgen zum Thema Data Science.
  www.towardsdatascience.com

- **Datanami**: Ein Nachrichtenportal zum Thema Big Data und Analytics, produziert von Tabor Communications.
  www.datanami.com

- **Data Science Blog**: Ein vorwiegend deutschsprachiger Blog über Data Science, verantwortet von Benjamin Aunkhofer, Geschäftsführer von Datanomiq, einer Beratungsgesellschaft. Nicht hunderte Beiträge am Tag, aber immer wieder interessante Interviews, Use Cases und Fachbeiträge.
  www.data-science-blog.com

- **Meetups**: Meetup ist ein soziales Online-Netzwerk, das offline Meetings von Interessengruppen organisiert. Zu den Themen Data Science, künstliche Intelligenz, Big Data etc. gibt es in verschiedenen deutschen Städten die entsprechenden Gruppen. Die mehr oder weniger regelmäßig stattfindenden 'Meetups' bestehen in der Regel aus Vorträgen und Diskussionen und dienen dem *Netzwerken*.

Autor

Michael Oettinger ist Gründer und Geschäftsführer der oetti-ds GmbH. Er berät Unternehmen im Bereich AI, Data Science und Machine Learning. Nach einem Studium der Betriebswirtschaft mit Schwerpunkt auf mathematischen Verfahren und Marktforschung in Augsburg und Oviedo, Spanien füllte er unterschiedliche Rollen bei PwC, IBM (u. a. SPSS), Fuzzy Logix und weiteren Softwareunternehmen aus. Als Mitglied bei MENSA beschäftigt er sich sowohl mit der menschlichen als auch mit der *künstlichen* Intelligenz. Schwerpunkt seiner Aktivitäten ist der konkrete und pragmatische Einsatz der existierenden analytischen Modelle in der betrieblichen Praxis mit den entsprechenden Softwaretools.

Literaturverzeichnis

Das Literaturverzeichnis mit direkten Links auf im Internet verfügbare Literaturquellen findet sich auch auf der Webseite zum Buch.

www.data-science-buch.de/literatur.html.

Alby, Tom (2023), Data Science in der Praxis: Data Science in der Praxis - Eine verständliche Einführung in alle wichtigen Data-Science-Verfahren.

Backhaus, Klaus; Erichson, Bernd; Weiber, Rolf (2015), Fortgeschrittene Multivariate Analysemethoden.

Backhaus, Klaus; Erichson, Bernd; Plinke, Wulff; Weiber, Rolf (2023), Multivariate Analysemethoden - Eine anwendungsorientierte Einführung.

Bali, Raghav; Sarkar, Dipanjan (2016), R Machine Learning By Example.

Bengfort, Benjamin; Bilbro, Rebecca; Ojeda, Tony (2018), Applied Text Analysis with Python.

Box, George E. P.; Norman R. Draper (1987), Empirical Model-Building and Response Surfaces.

Brownlee, Jason (2019), A Tour of Machine Learning Algorithms, http://machinelearningmastery.com/a-tour-of-Machine Learning-algorithms/, (abgerufen am 05.05.2020).

Cook, Darren. (2016), Practical Machine Learning with H_2O.ai - Powerful, Scalable Techniques for Deep Learning and AI.

Dua, Rajdeep; Singh Ghotra, Manpreet; Pentreath, Nick (2017 - 2. Auflage), Machine Learning with Spark.

Fuzzy Logix (2016), U.K. retailer embraces fresh thinking, www.fuzzylogix.com/solutions/supply-chain-optimization/, (abgerufen am 05.05.2020).

Fuzzy Logix (2017), Fighting Diabetes with Data www.fuzzylogix.com/solutions/chronic-illness-predictive-modelling/, (abgerufen am 05.05.2020).

Gartner (2013), Extend Your Portfolio of Analytics Capabilities.

Gartner (2020), Magic Quadrant for Data Science Platforms, www.gartner.com/reviews/market/data-science-Machine Learning-platforms, (abgerufen am 05.05.2020).

Gallatin, Kyle; Albon, Chris (2023), Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning.

Géron, Aurélien (2022), Hands-On Machine Learning with Scikit-Learn and TensorFlow.

Harvard Business Review (Oktober 2012) Data Scientist: The Sexiest Job of the 21st Century. https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century (abgerufen am 05.05.2020).

Haneke, Haneke (Hrsg); Trahasch, Stephan (Hrsg); Zimmer, Michael (Hrsg); Felden, Carsten (Hrsg), Data Science: Grundlagen, Architekturen und Anwendungen (2019).

Herbold, Steffen (2022), Data-Science-Crashkurs: Eine interaktive und praktische Einführung.

Hueske, Fabian; Kalavri, Vasiliki (2019), Stream Processing with Apache Flink.

Joshi, Prateek; Hearty, John; Sjardin, Bastiaan; Massaron, Luca; Boschetti, Alberto (2016), Python: Real World Machine Learning.

Kapil, Archish Rai (2018), Data Vedas: An Introduction to Data Science

Karim, Rezaul; Kaysar, Mahedi (2016), Large Scale Machine Learning with Spark.

Kapoor, Amita; Gulli, Antonio (2022), Deep Learning with TensorFlow and Keras: Build and deploy supervised, unsupervised, deep, and reinforcement learning models, 3rd Edition

Kriesel, David (2007), Ein kleiner Überblick über Neuronale Netze, www.dkriesel.com, (abgerufen am 05.05.2020).

Lantz, Brett (2019 – 3. Auflage), Machine Learning with R.

Liu, Yuxi (2017), Python Machine Learning By Example.

Mahayar, David (2016), Kollaborative Empfehlungssysteme im E-Commerce, www.ke.tu-darmstadt.de/lehre/arbeiten/master/2016/Davari_Mahyar.pdf, (abgerufen am 05.05.2020).

McKinsey Global Institute (2011), Big data: The next frontier for innovation, competition, and productivity.

Moroney, Laurence (2020), AI and Machine Learning for Coders: A Programmer's Guide to Artificial Intelligence

Müller, Andreas C.; Guido, Sarah (2017), Einführung in Machine Learning mit Python - Praxiswissen Data Science.

Müller, R. M. & Lenz, H.-J. (2013). Business Intelligence. Berlin: Springer Vieweg.

Ng, Annalyn Ng; Soo, Kenneth (2018), Data Science – was ist das eigentlich?!: Algorithmen des maschinellen Lernens verständlich erklärt.

Papp, Stefan; Weidinger, Wolfgang (2022), Handbuch Data Science und KI: Mit Machine Learning und Datenanalyse Wert aus Daten generieren

Polak, Adi (2023), Scaling Machine Learning With Spark: Distributed ML With MLlib, TensorFlow, and PyTorch.

Ramsundar, Bharath; Bosagh Zadeh, Reza (2018), TensorFlow for Deep Learning.

Raschka, Sebastian; Liu, Yuxi et al. (2022), Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python

Sjardin, Bastiaan; Massaron, Luca; Boschetti, Alberto (2016), Large Scale Machine Learning with Python.

SparkR http://spark.apache.org/docs/latest/sparkr.html.

Veljanoski, Jovan (2019), How to analyse 100 GB of data on your laptop with Python - https://towardsdatascience.com/how-to-analyse-100s-of-gbs-of-data-on-your-laptop-with-python-f83363dda94, (abgerufen am 05.05.2020).

Wickham, Hadley; Grolemund, Garrett (2017), R for Data Science.

Witten, Ian H.; Frank, Eibe; Hall, Mark A. (2005 – 2. Auflage), Data-Mining: Practical Machine Learning Tools and Techniques.

Stichwortverzeichnis