Machine Learning In Microsoft Azure - Ein Vergleich

DIPLOMARBEIT

verfasst im Rahmen der

Reife- und Diplomprüfung

an der

Höheren Abteilung für Informatik

Eingereicht von:
Polleichtner Moritz
Sljivic Emina

Betreuer:
Karpowicz Michał

Projektpartner:
Bammer Patrick, smartpoint IT consulting GmbH

Leonding, April 2022

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Die vorliegende Diplomarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Leonding, April 2022 M. Polleichtner & S. Emina

Zur Verbesserung der Lesbarkeit wurde in diesem Dokument auf eine geschlechtsneutrale Ausdrucksweise verzichtet. Alle verwendeten Formulierungen richten sich jedoch an alle Geschlechter.

Abstract

Brief summary of our amazing work. In English. This is the only time we have to include a picture within the text. The picture should somehow represent your thesis. This is untypical for scientific work but required by the powers that are. Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Zusammenfassung

Zusammenfassung unserer genialen Arbeit. Auf Deutsch. Das ist das einzige Mal, dass eine Grafik in den Textfluss eingebunden wird. Die gewählte Grafik soll irgendwie eure Arbeit repräsentieren. Das ist ungewöhnlich für eine wissenschaftliche Arbeit aber eine Anforderung der Obrigkeit. *Bitte auf keinen Fall mit der Zusammenfassung verwechseln, die den Abschluss der Arbeit bildet!* Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Inhaltsverzeichnis

1 Machine Learning - Theorie

1.1 Künstliche Intelligenz

Küntliche Intelligenz (KI) gewann in den letzten Jahrzehnten immer mehr an Beliebtheit und ist im Alltag stets vertreten (auch in Situationen, wo man sie nicht erwartet), jedoch ist diese Art von Technologie keine Neuheit. Erstmals wurde der Begriff KI von McCarthy im Jahr 1956 erwähnt, welcher auf der Dartmouth Universität ein Forschungsprojekt zu diesem Thema führte. [1] Dazu kommen noch Begrifflichkeiten wie Machine Learning (ML) und Deep Learning (DL), welche oftmals als Synonym verwendet werden, allerdings handelt es sich dabei um eigene Bereiche, die sich dennoch in vielen Aspekten überschneiden.

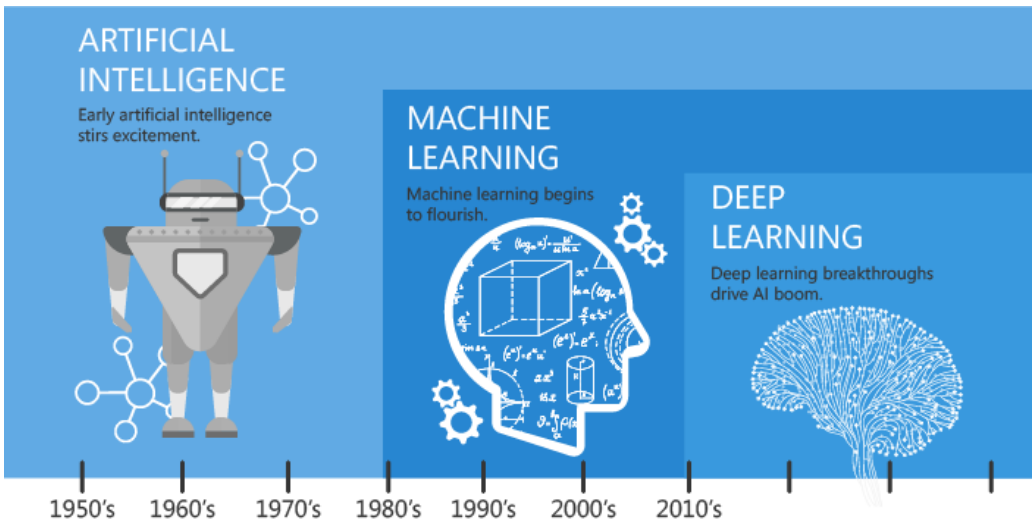


Abbildung 1: Evolution von Künstlicher Intelligenz

Der Begriff KI (Artificial Intelligence (AI)) lässt sich nur schwer genau definieren und umfasst daher ein breites Spektrum der Informatik. Dabei wurden im Verlauf der Geschichte mehrere Definitionen aufgestellt, welche sich auf die unterschiedlichen Interpretationen des Wortes "Intelligenz" fokussieren.

> Artificial intelligence is that activity devoted to making machines intelligent, and intelligence is that quality that enables an entity to function appropriately and with foresight in its environment. [2]

Diese Definition umfasst sowohl komplexe als auch einfach verknüpfte Logiken wie ein Taschenrechner oder vorbestimmte Zuordnungen (Listing 1). Jedoch repräsentieren diese Beispiele nicht den heutigen Fortschritt in diesem Bereich der Technologie.

Listing 1: Zuordnung mit einer if/else-Verzeigung

```
1       if(/* Bedingung */){
2           // Funktionalitaet
3       } else {
4           // Funktionalitaet
5       }
```

Da das Konzept der KI nicht vorgibt, mit welcher Technik ein Problem gelöst wird, bestand die Möglichkeit einer Ausbreitung in allen möglichen Richtungen, welche sowohl Algorithmen als auch Einsatzgebiete inkludieren. Dabei gilt eine Voraussetzung: das Imitieren einer menschlichen Funktion.

1.2 Machine Learning

Bei ML handelt es sich um eine Untergruppe der KI, welche das menschliche Lernen nachahmt. Dabei werden vordefinierte Daten übergeben und die Maschine versucht Strukturen und Muster zu finden, welche dann in Gruppen unterteilt werden. Hierfür ist eine große Menge an Daten nötig, die in der Trainingsphase an das Modell übergeben werden. Im Laufe der Verwendung wird dieses Modell immer präziser, da mit neuen Daten neue Verbindungen geknüpft werden können und vorhandene gestärkt werden.

Modell Dabei ist ein Modell der Algorithmus, der Muster mithilfe von Daten findet.

Zu den Einsatzmöglichkeiten von ML gehören [3]:

- Spamerkennung von Emails und Telefonanrufen
- Dokumentenklassifizierung
- Chatbots
- Persönliche Assistenten (Siri, Alexa, ...)
- Medizinische Diagnostik
- Betrugserkennung bei Banktransaktionen

1.2.1 Arten

Der wichtigste und komplizierteste Teil beim maschinellen Lernen ist die Kunst, einem Computer das selbstständige Lernen beizubringen, dabei orientiert man sich am Lernprozess eines Menschen. Dazu existieren eine Vielzahl an Ansätzen und zu den bekanntesten gehören:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Um diese Ansätze nachvollziehen zu können, muss man zuerst die menschliche Intelligenz verstehen oder genauer gesagt, die Frage "Wie lernt das menschliche Gehirn?".

Menschliche Ursprünge vom Machine Learning

Bereits als Fötus entwickeln sich Neuronen, die sich über die Zeit verknüpfen und gemeinsam ein Netzwerk bilden, welches dem Körper Anweisungen übermittelt. Daher kommt ein Neugeborenes mit etwa 100 Milliarden Neuronen auf die Welt, die zur Zeit der Geburt nur schwach miteinander verbunden sind. Mithilfe des Lernens werden diese Verbindungen gestärkt, und das Kind kann Vorgehensweisen besser verstehen und neue Erkenntnisse gewinnen, damit steigen zusätzlich auch das Gewicht und die Größe des Gehirnes. [4]

Jedoch verschwinden diese Verbindungen, falls sie nicht aufgefrischt werden und Informationen in Vergessenheit verfallen. Weitere Faktoren könnten der Alterungsprozess oder eine neurologische Krankheit sein, die das Phänomen erklären, dass man im höheren Alter Probleme beim Erlernen von Neuem hat. [5]

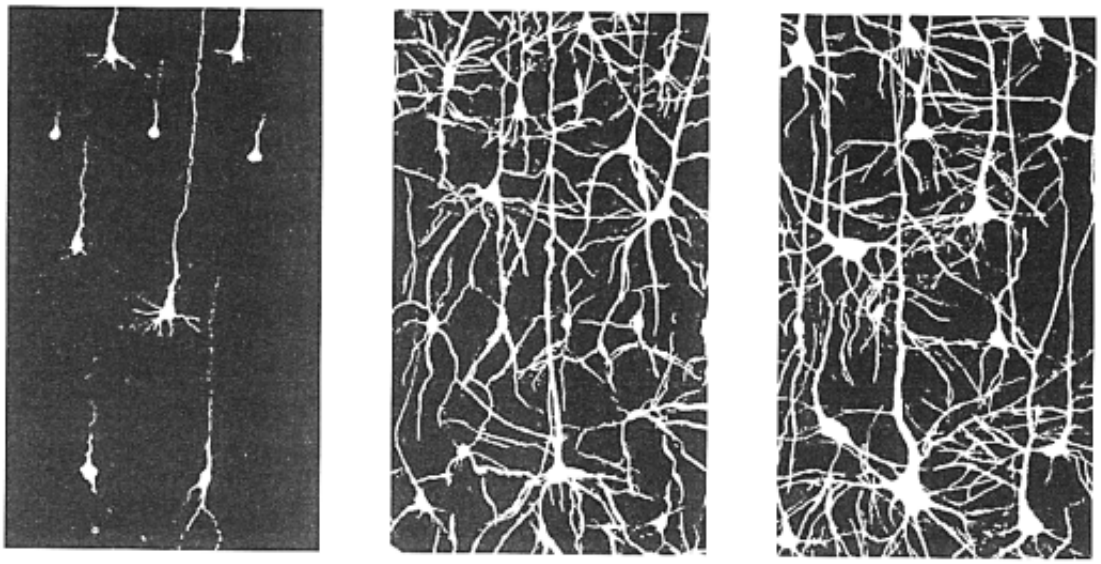


Abbildung 2: Vernetzungen nach der Geburt, nach 3 Monaten und nach 15 Monaten

Neue Informationen verbinden bereits bestehende Neuronen und altes Wissen wird erneuert. Außerdem wird mit oftmaligem Wiederholen das Netzwerk dichter, und man kann das Erlernte leichter abrufen, zugleich werden die neuen Informationen mit dem bereits existierenden Vorwissen besser kombiniert. [4]

Supervised Learning

An einen Algorithmus werden gruppierte Daten übergeben, die neben einer Gruppe noch mehrere Merkmale beinhalten, um danach neue Datensätze zu klassifizieren oder zu regredieren. Die Bezeichnung "Supervised" (im Deutschen "Überwachtes") kommt daher, dass die Gruppen bereits vordefiniert sind und das Programm nicht selber welche erstellen muss. [6]

Farbe	**Form**	**Geschmack**	**Frucht**
rot	herzförmig	süß	Erdbeere
gelb	oval	säuerlich	Zitrone
rot	rund	süß	Apfel
grün	rund	säuerlich	Apfel

Tabelle 1: Beispiel für gruppierte Daten als Tabelle; Merkmale: Farbe, Form, Geschmack; Gruppe: Frucht

Diese Art von Lernen kann man in zwei Typen aufteilen:

- Klassifizierung
- Regression

Klassifizierungs Probleme verwenden Algorithmen, um Daten einer bestimmten Kategorie zuzuteilen. Oft gibt es nur zwei Kategorien wie zum Beispiel Hund/Katze oder Ja/Nein, jedoch gibt es auch Fälle, wo eine Vorhersage mit einer Wahrscheinlichkeit zwischen 0 und 1 getroffen wird. Weiteres gibt es auch Situationen, wo zwischen einer großen Menge an Kategorien ausgewählt wird, zum Beispiel bei der Erkennung von handschriftlichen Ziffern, in diesem Beispiel würde es zehn Möglichkeiten geben.

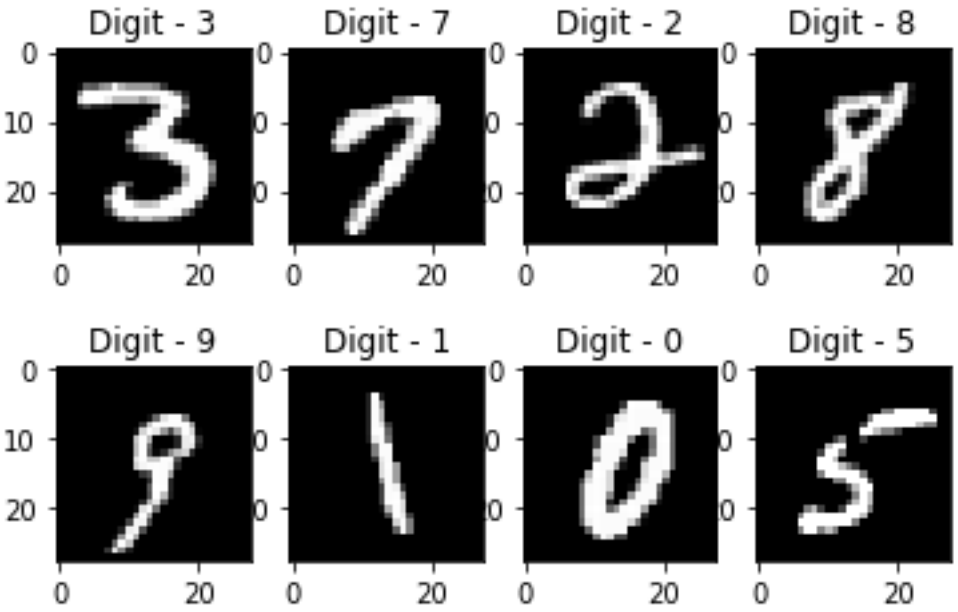


Abbildung 3: Beispiel für eine Klassifizierung von handschriftlichen Ziffern

Zu diesen ”überwachten” Algorithmen gehören Lineare Diskriminanzanalysen, Support Vector Machines (SVM), Random Forests und Entscheidungsbäume.[6]

Entscheidungsbäume (Decision Trees) sind eine hierarchische Abfolge von Bedingungen und sind vergleichbar mit verschachtelten if/else-Statements. Dabei beginnt der Entscheidungsbaum mit einer Bedingung, die auch als ”Root-Node” bezeichnet wird, worauf im Normalfall weitere Bedingungen folgen. Die Blätter dieser Pfade spiegeln die Gruppen oder Klassifizierungen wieder und sind nur über mehrere Pfade erreichbar.

Jedoch haben Entscheidungsbäume das Problem, dass sie sehr gut mit den antrainierten Daten arbeiten und weniger genau mit neuen Daten (Overfitting 1.2.2). Um diese

Genauigkeit zu verbessern, kann man zum Beispiel über Hyperparameter die maximale Länge eines Pfades setzen, wodurch die Bedingungen verallgemeinert werden.

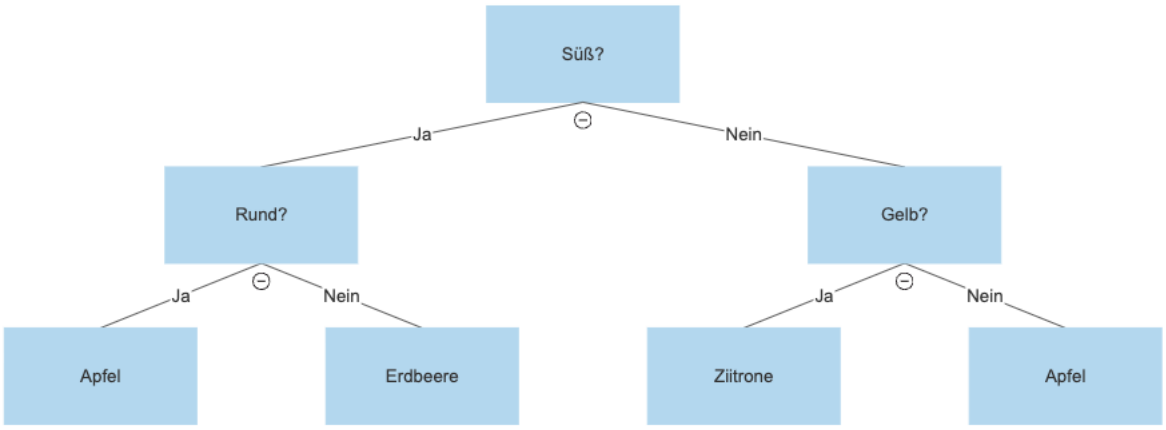


Abbildung 4: Decision Tree an dem Beispiel von Tabelle 1

In diesem Beispiel sieht dieser Entscheidungsbaum noch sehr lesbar aus, jedoch ändert sich dies, wenn Entscheidungen dargestellt werden, bei denen es auf die Nachkommastelle ankommt und wenn die Kategorien sehr schwer differenzierbar sind. Bei dem Supervised Learning erstellt das Programm selbstständig einen Entscheidungsbaum, indem es Muster oder Zusammenhänge findet und analysiert. Nach vielem Lernen kann dieser Entscheidungsbaum optimiert werden und unnötige Verbindungen können entfernt werden.

Fasst man mehrere Entscheidungsbäume zusammen, entsteht ein Random Forest, wobei jeder Entscheidungsbaum nur bestimmte Spalten/Merkmale zugeteilt bekommt. Soll ein neuer Datensatz kategorisiert werden, entscheidet die Mehrheit jener Ergebnisse der Entscheidungsbäume, zu welcher Gruppe dieser Datensatz dazugehört.

Regressionen, Erstellung einer kontinuierlichen Funktion mithilfe von Werten, die auf oder nahe an der Funktion liegen, sind hilfreich, wenn anstatt diskreten kontinuierlichen Werte, wie zum Beispiel die Größe einer Person, festgestellt werden sollen. Dazu gehören lineare Regressionen und polynominale Regression.

Unsupervised Learning

Beim Unsupervised Learning oder unüberwachtes Lernen werden Verbindungen ohne genauere Informationen über den Testdatensatz erzeugt. Dabei muss das Programm

selbst Gruppen definieren und dann die übergebenen Daten in diese Gruppen zuordnen. [6]

Clustering gehört zu den beliebtesten Varianten, um selbstständig Gruppen zu erstellen. Dabei wird jeder Datensatz als Punkt in ein Koordinatensystem mit beliebig vielen Dimensionen eingetragen. Eine Achse stellt ein Attribut dar und je nach Ausprägung ist der Punkt mehr oder weniger vom Ursprung entfernt.

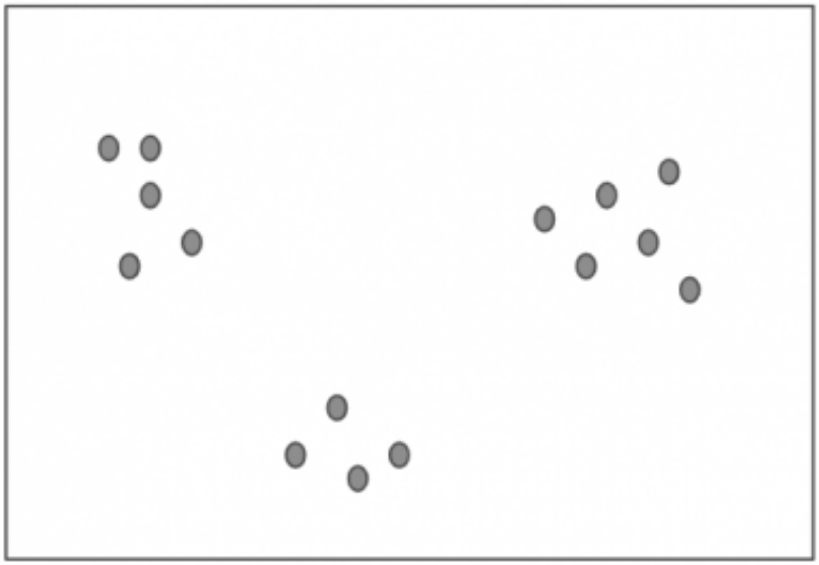


Abbildung 5: Unkategorisierte Daten

Auch für das menschliche Auge ist es möglich, dieses Beispiel (Abbildung 5) in Haufen oder Klumpen zusammenzufassen, genau das gleiche macht ein Programm mit Clustern. Die Interpretation dieser Gruppen muss jedoch wieder durch Menschen erfolgen, da ein Computer nicht im Stande dazu ist, diese Cluster einer Kategorie zuzuteilen.

Diese Vorgehensweise wird oft in sehr komplizierte Einsatzbereiche genutzt und daher ist es sehr schwer, differenzierbare Cluster zu erstellen. Der Prozess, solche Cluster zu definieren, basiert darauf, die Punkte so zu gruppieren, dass der Abstand in diesem Cluster klein ist und zu anderen Clustern groß.

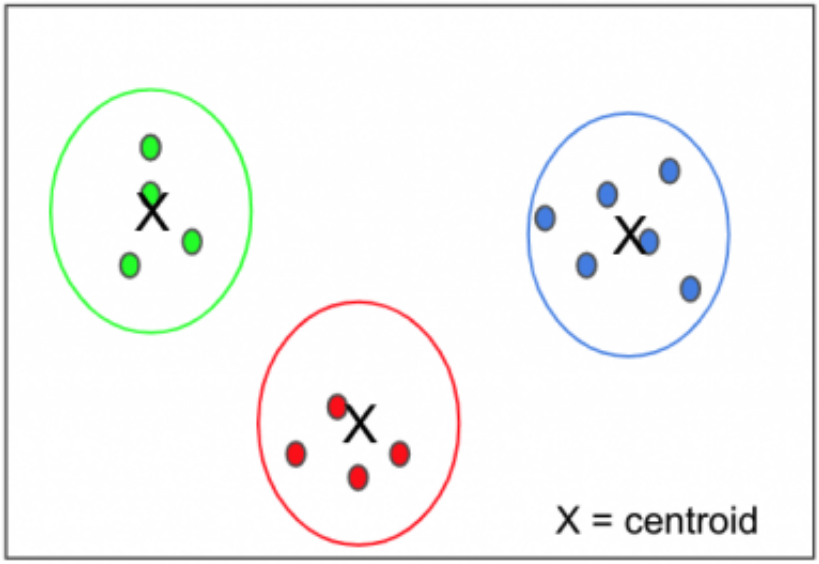


Abbildung 6: Kategorisierte Daten

Reinforcement Learning

Das Reinforcement Learning oder das beschränkte/verstärkte Lernen wird oft mit dem Konzept ”Learning by doing” verglichen, da es sich weniger auf das Ergebnis fokussiert und mehr auf Aktionen oder Vorgängen. Ein Beispiel für diese Vorgehensweise aus der Sicht eines Schülers ist das Üben vor einer Matheschularbeit. Wird während dem Üben ein Fehler gemacht, merkt man sich das Problem und passt sein Verhalten / seinen Rechenweg so an, dass dieser Fehler nicht mehr vorkommt. Diesen Vorgang nennt man auch negative Verstärkung. [6]

Dahingegen führen richtige Ergebnisse oder erwartete Reaktionen zu positiven Verstärkungen und man versucht dieses Verhalten zu wiederholen.

Durch negative und positive Verstärkungen wird das Verhalten verbessert, um den besten Weg zum Ziel zu finden. Bei komplexen Systemen kann dies selbstständig vom Programm gemacht werden, jedoch bei simpleren kann es geschehen, dass ein unnötig komplizierter Weg gefunden wird. In diesen Fällen schaut ein Supervisor dem Programm ”über die Schulter”.

1.2.2 Probleme

Overfitting

bedeutet, dass ein Modell zu sehr an die Trainingsdaten angepasst ist und auf neue Daten falsche Ergebnisse liefert. Dies hat den Grund, dass das Modell zu lange mit dem

Trainingsdatensatz gearbeitet hat und zu spezifisch wurde. Daher versucht man mit einer Verallgemeinerung über die Hyperparameter das Modell zu vereinfachen.

Underfitting

bedeutet, dass ein Modell viel zu allgemein ist und nicht an die Trainingsdaten angepasst ist. Einerseits könnte es sein, dass das Modell nur kurz mit den Trainingsdaten gearbeitet hat, andererseits kann es sein, dass die Klassifizierung innerhalb des Trainingsdatensatzes eine starke Mehrheit zeigt. [7]

1.2.3 Phasen

Das Erstellen von einem ML-Modell wird in vier Phasen unterteilt, wobei jede Phase eine bestimmte Aufgabe erfüllt.

1. Vorbereitung der Daten / Preprocessing

   Jeder Anfang sollte mit einer Analyse der Daten anfangen. Dabei sollte man die Datentypen, die Anzahl der Daten oder Klassifizierungen und die Anzahl der fehlenden Werte zusammenzufassen und überblickbar darstellen (Tabellen oder Diagramme). Dies hat zwei wesentliche Gründe und einer davon ist, dass man sich mit den Daten vertraut macht und danach über ein gewisses Verstehen verfügt. Der zweite Grund ist, dass man im Verlauf der Phase die Daten so verarbeiten kann, dass man am Ende das beste Ergebnis erreicht.

   Diese Vorbereitung sollte mit stichprobenartigen Daten angeführt und später durch verschiedene Strategien weitergeführt werden, die dazu dienen, entweder fehlende Daten zu bereinigen oder Ausreißer zu beseitigen.

   Wie man mit fehlenden Daten umgeht, kommt auf die Situation an. Hat man Zugang zu einer großen Menge an Daten, könnte es sinnvoll sein, diese Daten zu löschen. Dabei bleibt die Frage offen, ob man die ganze Spalte (wenn alle fehlenden Werte in einer Spalte sind) löscht oder nur die betroffenen Zeilen. Falls die Entscheidung getroffen wird, dass die Spalte entfernt wird, muss man sich vergewissern, dass damit potenziell wichtige Daten verloren gehen. Ist die Anzahl der Datensätze jedoch klein, hat man die Möglichkeit, fehlende Werte mit dem Mittelwert oder Ähnlichem zu ersetzen, dieser kann aus allen Daten gebildet werden oder aus den Daten, die über dieselbe Klassifizierung verfügen. Sind diese

Mittelwerte nicht repräsentativ, kann dieser durch einen fixen Wert ersetzt werden. Ein Beispiel dafür wäre die Analyse eines Schlaganfall-Datensatzes, in Fällen eines Kindes ist zu vermuten, dass es nicht raucht und daher ist es vertretbar, dass man dafür den Wert 0/false einsetzt. Die letzte Möglichkeit ist, dass man diese Werte mithilfe eines ML-Modells (Regression) bestimmt. [8]

Das Gegenteil der fehlenden Werte sind doppelte Werte, diese können zur Überrepräsentierung von bestimmten Klassifizierungen führen und sollten daher bereinigt werden. Bei ähnlichen Werten ist dies jedoch eine Interpretationssache, da es in manchen Situationen ein unbedachter Messfehler sein könnten. Ein Vorteil dieses Schrittes ist, dass man am Ende von jeder Klasse gleich viele Datensätze hat.

Als Nächstes wäre eine Analyse jeder einzelnen Spalte sinnvoll, anfangend mit den Datentypen. Falls es sich um Enumerationswerte handelt, müssten diese in numerische Werte umgewandelt werden, außer es handelt sich um die Klassifizierungsspalte. Handelt es sich um eine Spalte, wessen Werte eine Rangfolge (z.B.: gut, mittel, schlecht) darstellen, kann man diese mit einer Nummer zwischen 0 und 1 austauschen. Dabei ist zu beachten, dass der minimalste und maximalste Rang entweder den Wert 0 oder 1 zugeteilt bekommen und jeder Rang dazwischen einen Wert dazwischen. Sind es jedoch unabhängige Enumerationswerte, könnte man mithilfe der One-Hot-Encoding Methode die Daten umwandeln, wo jeder Enumerationswerte eine zusätzliche Spalte bekommt und entweder mit 0/1 (false/true) befüllt ist. Außerdem sollten unterschiedliche Einheiten angeglichen werden und jene textuelle Einheit aus dem Wert entfernt werden.

Das letzte Problem sind Ausreißer, um diese zu identifizieren, müsste man die eigentliche Verteilung der Daten kennen. Danach vergleicht man verdächtigte Werte (meistens der größte oder kleinste Wert) mit dem Durchschnitt und entscheidet, ob es sich wirklich um unerklärliche Werte handelt. Diese können mit den gleichen Funktionen wie bei fehlenden Werten ersetzt werden.

2. Visualisierung der Daten

   Die tabellarische Darstellung von Daten erschwert dem menschlichen Auge, Muster zu erkennen, um dies zu umgehen ist jede Art der Visualisierung hilfreich. Damit kann man schnell Insights und Gruppen identifizieren, außerdem kann sie auch bereits bei der Vorbereitung der Daten helfen, da man zum Beispiel mithilfe eines Boxplots Ausreißer deutlich schneller erkennen kann.

Um dies noch leichter zu machen, ist es wichtig, dass die Farbpalette an die gegebene Situation angepasst ist, da es bei schlechter Repräsentation leicht zu Verwirrung kommen kann. Wie zum Beispiel beim Darstellen des Wetters bietet sich die Farbe blau für Regen an und rot/gelb für Sonnenschein.

Jedes Diagramm hat seine Vorteile und Nachteile, wie zum Beispiel beim Violinplot die Verteilung der Daten relativ zur Anzahl der gleichen Klassifizierung sehr gut dargestellt werden. Jedoch hat es den Nachteil, dass diese Darstellungen oft irreführend sind, da sie relativ anstatt absolut ist.

Um Zusammenhänge besser zu erkennen, kann man zwei Variablen in einem Diagramm darstellen.

3. Training des Modells

   Der erste Schritt beim Trainieren ist die Einteilung von Trainingsdaten und Testdaten, hierzu wird eine Einteilung von 70% zum Trainieren und 30% zum Testen angestrebt.

   Es ist wichtig, dass die Testdaten eine ausgeglichene Anzahl von Daten pro Klassifizierung enthaltet, da es sonst zu einem unausgewogenen Modell kommen könnte. Wie in 1.2.2 (Overfitting) beschrieben wurde, kann sich ein Modell zu sehr an die übergebenen Trainingsdaten anpassen, sodass später die Testdaten nicht korrekt klassifiziert werden. Um dies vorzubeugen, könnte man einen Teil der Testdaten als Validierungsdaten nutzen, welche die Anpassung der Hyperparameter möglich machen [9].

   Es ist außerdem wichtig, dass die Testdaten lediglich zum Testen benutzt werden, denn sobald das Modell diese Daten verarbeitet, werden sie gespeichert und im späteren Verlauf ebenfalls zur Klassifizierung genutzt.

   Am Anfang sollte man sich für einen traditionellen Algorithmus entscheiden, damit man später jegliche Veränderungen mit einem Basiswert vergleichen kann.

4. Evaluierung und Verbesserung

   Um herauszufinden, ob das antrainierte Modell wirklich nutzvoll und genau ist, können folgende Werte berechnet und verglichen werden: Accuracy (Genauigkeit), Precision (Präzision) oder Recall. Je nach Situation sollte man den Fokus auf einen bestimmten Wert legen. Zum Beispiel beim Testen von Wasserqualitäten würde der wichtigste Wert die Precision sein, denn man kann in Kauf nehmen, dass

nicht schädliches Wasser als schädlich gekennzeichnet wird, die entgegengesetzte Situation wäre nicht akzeptabel. [10]

$$accuracy = \frac{correct\ predictions}{all\ predictions}$$

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

$$recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

Abbildung 7: Formeln für Accuracy, Precision und Recall

Nachdem das Modell mit den Testdaten überprüft wurde, kann man beurteilen, ob die Genauigkeit die Erwartungen erfüllt. Falls dies nicht der Fall ist, bestehen drei Möglichkeiten die Genauigkeit zu verbessern.

Als Erstes kann man die ausgewählte Vorgehensweise für fehlende oder ausreißende Daten ändern oder anpassen und die zweite Möglichkeit ist es, den Algorithmus des Modells zu ändern. Am Ende kann man noch selbstständig die Hyperparameter anpassen, was jedoch ein mühseliger Prozess sein kann, um ihn zu verkürzen, kann man einen dieser Ansätze verwenden: Gridsuche, Zufallssuche, Bayessche Optimierung, Gradientenbasierte Optimierung oder Evolutionäre Optimierung. [8]

1.2.4 Optical Character Recognition

In den meisten Fällen erfolgt eine Eingabe über eine Tastatur, jedoch ist dies manchmal weder die beste noch effizienteste Art, Text einzulesen. Mithilfe von Optical Character Recognition (OCR) ist ein automatisiertes Einlesen und Verarbeiten von handschriftlichen oder gedruckten Text möglich und das schon bereits in den 1950er. Am Anfang noch um Verkaufsberichte in Lochkarten zu konvertieren, damit ein Computer mit den Verkaufsdaten arbeiten kann. [11]

Im Bereich von OCR sind bereits momentan gute und präzise Resultate mit Machine Learning (ML) erwartbar, jedoch wie bei allen anderen Problem ist es verbesserbar. Um einen großen Fortschritt zu erreichen, würde die Nutzung von Deep Learning verpflichtend sein, dies ist jedoch in den meisten Situationen nicht notwendig.

Die Präzision hängt von vielen Attributen ab, daher kann ein eingescannter Text viel besser verarbeitet werden als ein im Freien geschossenes Bild mit dem Fokus auf ein Straßenzeichen [12].

- Textdichte

  Es macht einen Unterschied, wie viel Text sich auf einer Fläche oder einem Bild befindet, denn es ist in gewissen Situationen leichter, Text auszulesen, wenn dieser nur spärlich vorkommt.

- Struktur

  Wenn man eine klare Struktur erkennt, zum Beispiel in Tabellen oder in Zeilen, kann man ein besseres Ergebnis erwarten, daher ist es auch wichtig, dass man vor dem Auslese-Prozess das übergebene Bild aufbereitet und als Beispiel die Rotation ändert.

- Schriftart

  Handgeschriebene Texte oder ”laute” Schriftarten sind im Gegensatz zu einfachen und gedruckten viel komplizierter, da sie kaum strukturiert sind. Außerdem könnten Buchstaben Ähnlichkeiten aufweisen, was später zu Verwechslungen führen kann.

- Buchstaben

  Sprachen wie Arabisch, Chinesisch, Russisch oder Japanisch benutzen im Gegensatz zu Deutsch ein anderes Alphabet, dabei kann es zu ähnlichen Buchstaben und Vertauschungen kommen, daher sinkt die Präzision in Texten mit mehreren Sprachen. Dies kann auch der Fall sein, wenn mathematische Formeln vorkommen.

Kyrillisches Alphabet	Ähnelt dem Buchstaben im lateinischem Alphabet
р	p
В	B
Н	H
у	y
С	C

Tabelle 2: Ähnlichkeiten zwischen Buchstaben im Lateinischem und Kyrillischem Alphabet

- Platzierung

  Zentrierte Texte erlauben ein besseres Auslesen als abgeschnittene oder verstreute Wörter.

Die Texterkennung ist generell in zwei Phasen aufgeteilt:

- Text Detection
- Text Recognition

Text detection

ist der Prozess, indem in einem Bild oder einer PDF erkannt wird, wo sich Text befindet. Beim Resultat handelt es sich um Bounding Boxen, diese schließen einen Textblock (Wort, Buchstabe oder Paragraf) ein, dabei werden die genauen Koordinaten der Eckpunkte zurückgegeben. Diese werden später genutzt, um zur Visualisierung Boxen auf dem Bild oder der PDF aufzuzeichnen. Dieser Prozess wird auch in der Objekterkennung genutzt.

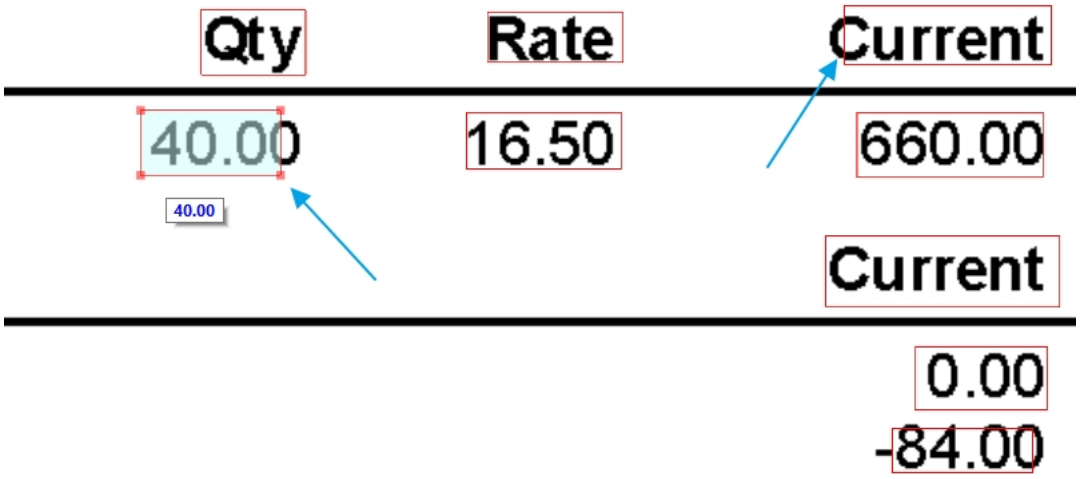


Abbildung 8: Bounding Boxes Beispiel

Die Erkennung kann entweder mittels der auf Regionen basierenden oder Textur basierenden Methode durchgeführt werden.

Bei der auf Regionen basierten Methode werden Pixel verbunden und als Zeichenkandidat markiert, welche später mehrmals gruppiert werden und schlussendlich Wörter oder Textzeilen bilden. Dabei kommt es auf die geometrischen Eigenschaften an, wo es zu Fehlern beim Gruppieren kommen könnte. Wie man bei Abbildung 8 sehen kann, wurde

der Buchstabe ”C” im Wort ”Custom” oder die letzte Nachkommastelle bei ”40.00” nicht ganz als Teil des Wortes oder der Zahl erkannt.

Mit dem Stroke Width Transformer (SWT) wird jedem Pixel eine Strichbreite zugeteilt, indem zwei Kanten gefunden werden mit der gleichen Richtung modulo 180°. Die Entfernung dieser zwei Kanten werden in den Kantenpixeln und alle unterliegenden Pixeln als Strichbreite gespeichert und alle Zusammenliegeden gleich breite Pixel werden zu einem Zeichenkandidaten gruppiert. Danach werden alle benachbarten Zeichenkandidaten untersucht und zu einem Wort gruppiert, falls das mittelwertige Strichbreite-Verhältnis nicht über 2 liegt. Außerdem wird die Höhe und Farbe des Zeichenkandidaten berücksichtigt. Dies kann auch der Grund sein, wieso bei Abbildung 8 das Minuszeichen bei ”-84.00” nicht zur Zahl hinzugefügt wird, das Minuszeichen ist signifikant niedriger als der Rest der Zahl. [13]

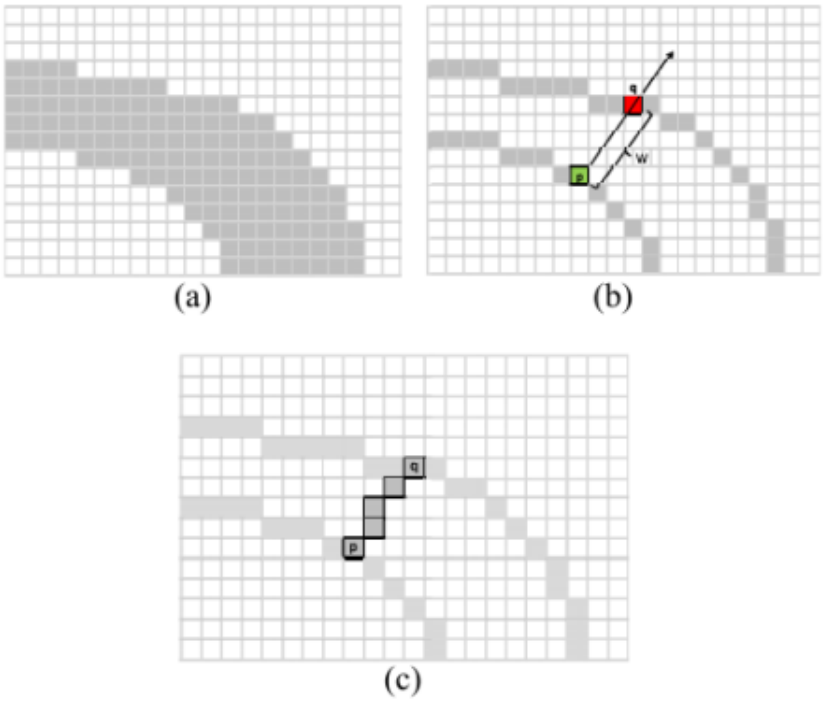


Abbildung 9: Die Kanten des Striches (a) werden solange verglichen, bis zwei gefunden werden mit der mit der gleichen Richtung (b). Alle unterliegenden Pixel erhalten die Strichbreite der Entfernung zwischen der Start- und Endkante (c).

Textur basierte Methoden unterteilen das Bild in Fenster, dessen Höhe wird später mit der geschätzten Textgröße verglichen. Dabei kann es ebenfalls zu Erkennungsfehlern kommen.

Die Mischung dieser beiden Methoden gewann im Jahr 2011 den ICDAR-Wettbewerb mit einem F-Score von 71.28%. Hierbei hat Chunghoon Kim den Vorschlag gegeben, dass als erstes Blöcke mit dem Maximally Stable Extremal Regions (MSER) Verfahren extrahiert und danach benachbarte Blöcke gruppiert werden, falls die Farbe und Größe sich ähnelt. Jedoch werden mit diesem Verfahren ebenfalls eine große Menge an false-postive Blöcken erkannt. Um diese Anzahl zu verringern, wird eine ähnlich Idee zur SWT verwendet. [13]

Wettbewerbe sind ein großer Grund für die Fortschritte in der Texterkennung und im Rahmen des zwei-jährlich stattfindenden International Conference of Document Analysis and Recognition (ICDAR) Wettbewerbs wurden alle oben genannten Ideen verglichen [14].

Text Recognition

ist genau wie bei der Erkennung von Text in zwei Möglichkeiten unterteilt: Regionen basierend und Textur basierend.

Die von MSER generierten und normalisierten Blöcke werden je nach der Orientierung in ein separates Bild extrahiert. Dabei sind acht Orientierungen möglich, welche mit dem Gaußschen Filter bearbeitet werden und auf ein 5 x 5 Bild komprimiert wurden. Mit diesen 5 x 5 x 8 = 200 dimensionalen Vektoren werden dann die bereits erkannten Blöcke klassifiziert. [13]

1.2.5 Natural Language Processing

Natura Language Processing (NLP) gehört zu den Disziplinen der künstlichen Intelligenz, welche konstant...

1.3 Deep Learning

Sowie beim ML sind DL-Algorithmen abhängig von antrainierten Daten und kann daher als Synonym oder Untergruppe gesehen werden. Der grobe Unterschied liegt darin, dass diese Daten meist komplett roh sind und keine Vorarbeit geleistet wurde. Daher kann ein DL-Modell wie ein Mensch noch nie davor gesehene, spezielle Bilder kategorisieren.

Die Einsatzmöglichkeiten überlappen sich sehr mit jenen von ML, jedoch ist das Resultat von DL-Modellen viel präziser. Daher würden die meisten modernen Assistenten ohne DL auch nicht auf dem erwarteten Niveau arbeiten.

Obwohl schon viele Anwendungen auf DL basieren, handelt es sich dabei um eine junge Technologie, die noch weit von ihrem vollen Potenzial entfernt ist.

1.4 ML vs DL

1.4.1 Feature Extraction

In beiden Fällen unterlaufen die Daten denselben Phasen: Feature Extraction und Classification. Jedoch unterscheiden sie sich bei der Ausführung der Feature Extraction.

In der Feature Extraction werden Merkmale zusammengefasst und in einer strukturierten Tabelle dargestellt. Jede Zeile beinhaltet diese Merkmale und die dazugehörige Klassifizierung.

Bei einem ML-Modell ist die Feature Extraction ein separater Vorgang, welcher nicht automatisch durch das Modell vorgenommen wird. Dieser Prozess ist oft sehr kompliziert und beansprucht eine lange Zeit, außerdem ist hier eine Person notwendig, die sich in dem gegebenen Bereich auskennt.

Im Kontrast dazu nutzt ein DL-Modell ein Neuronales Netzwerk, welches sich stark an einem menschlichen Gehirn orientiert und die Feature Extraction übernimmt.

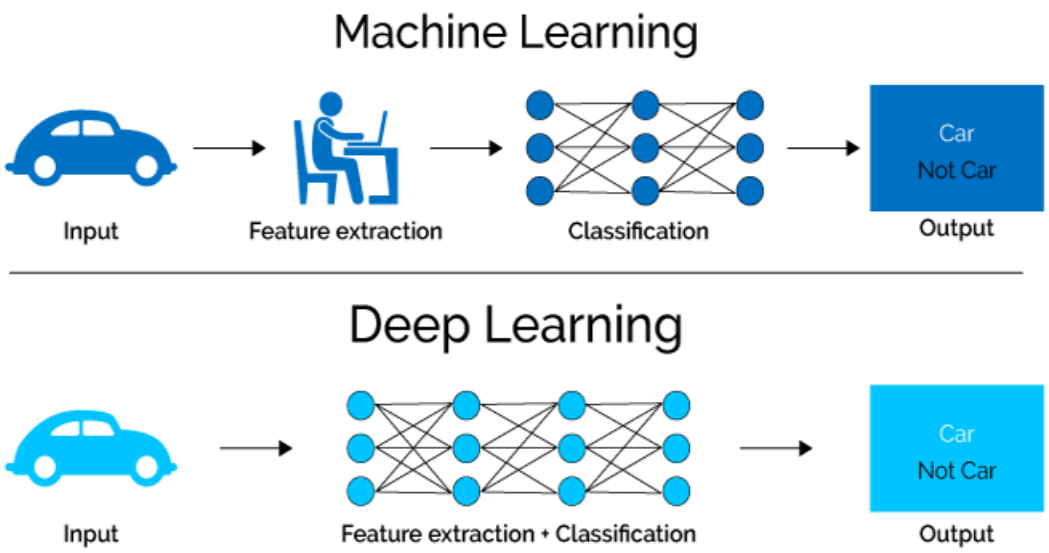


Abbildung 10: Feature Extraction, ML vs DL

Wie der Name ”Neuronales Netzwerk” schon verrät, besteht dieses Konzept aus Tausenden nachgeahmten Neuronen, die über mehrere Schichten zusammen kommunizieren. Dabei kann sich ein Neuron nur mit anderen Neuronen auf derselben oder darüberlegenden Schicht verknüpfen, daher entsteht eine hierarchische Struktur.

Die einzelnen Schichten können mit den fünf Sinnen eines Menschen verglichen werden. Verbindet man zum Beispiel einer Person die Augen und lässt sie dann eine Erdbeere probieren, kann die Person anhand des Geschmackes und des Duftes erkennen, um welche Frucht es sich handelt. Ähnlich würde es in einem Neuronalen Netzwerk ablaufen, wobei der Geschmack und der Duft einzelne Schichten repräsentieren.

Legt die Person daraufhin die Augenbinde ab und sieht, dass die Erdbeere weiß ist, kann sie sich trotzdem sicher sein, dass es sich um eine Erdbeere handelt, da sie aus Erfahrung sagen kann, dass der Geschmack sowie der Duft eindeutiger sind. Dies würde ein Neuronales Netzwerk am Anfang verwirren, da es nicht über diese Erfahrung verfügt. Mit der Zeit würde das Neuronale Netzwerk bestimmten Synapsen (Verbindungen) eine Gewichtung zuteilen, welche die Wichtigkeit einzelner Parameter angibt. Dieses Einordnen ist die Art, wie ein Neuronales Netzwerk ”lernt”.[15]

1.4.2 Vergleich

Ein ML-Modell sollte bevorzugt in Bereichen genutzt werden, wo bereits strukturierte Daten vorhanden sind. Die Menge an Daten ist hierbei nicht sehr relevant, da ein ML-Modell mit wenigen Daten präzise sein kann.

Stehen keine bereits präparierten Daten zur Verfügung, wäre ein DL-Modell die bessere Lösung, da man sich die Zeit und Kosten für die Feature Extraction sparen kann. Jedoch benötigt man leistungsstarke Computer, da Neuronale Netzwerke komplexe Rechnungen durchführen, die mithilfe einer GPU beschleunigt werden können. Außerdem ist die Programmierung und Trainingsphase sehr lang und mühselig. [16]

Bei der Entscheidung, welches Modell sinnvoller ist, sollte man außerdem beachten, dass die Technologie noch nicht voll entwickelt ist und dass manche Probleme mit dem momentanen Stand der Technik nicht lösbar sind.

2 Cloud Computing

2.1 Theorie

2.1.1 Was ist Cloud Computing?

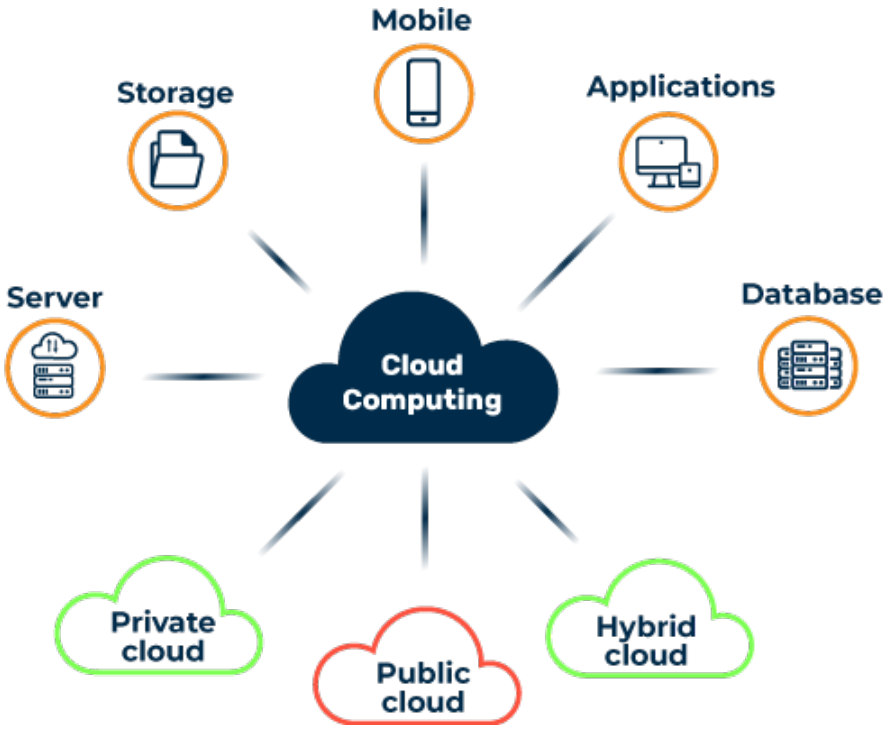


Abbildung 11: CC-Overview

Cloud Computing ist ein Modell, bei dem virtueller Speicher, jegliche Art von Server, Anwendungen usw. nicht physisch, sondern rein über das Internet bereitgestellt werden. Diese werden in der Regel nach Bedarf als Teil eines As-a-Service-Modells angeboten. Üblicherweise wird dieses Modell in Verbindung mit nutzungsbasierter Bezahlung offeriert. Mithilfe der Cloud werden lokale Rechenzentren und interne Systeme mit virtuellen Rechen-, Netzwerk- und Speicherressourcen ausgetauscht. Im Normalfall werden diese Ressourcen von externen Anbietern bereitgestellt. Statt selbst breit gefächerte Rechen- und Speicherressourcen aufzustellen und nebenbei auch noch zu warten, wird eine Vielzahl dieser Aufgaben von einem Cloudservice-Anbieter übernommen.

2.1.2 Vorteile von Cloud Computing

Oft fallen, in Verbindung mit dem Thema Cloud Computing, Stichwörter wie „flexibel“ oder „agil“, das liegt daran, dass gerade in heutigen Zeiten marktseitige und auch technologische Veränderungen schnell und oft vorgenommen werden und deswegen Unternehmen oftmals nicht Schritt halten können. Deshalb betreiben viele Unternehmen sogenanntes Outsourcing, um nicht selbst für die Rechenleistung ihrer eigenen Dienste verantwortlich zu sein. Skalierbarkeit spielt hierbei eine große Rolle, zum Beispiel: je mehr Zugriffe auf einen Webshop, desto mehr Ressourcen müssen im Hintergrund hochgefahren werden, um eine fehlerfrei Nutzung zu garantieren.

Auch die Usability ist simpler bei cloudbasierten Prozessen, da viele dieser Prozesse in den Hintergrund verschoben und von den Cloud Service-Anbietern übernommen werden. Der Aufwand für Wartung, Beschaffung, usw. für Rechenzentren entfällt weitgehend. Somit kann man im Bereich Energie- und Erhaltungskosten einsparen. Im Allgemeinen werden die Kosten für solche Dienste je nach Vereinbarung nutzungsabhängig vereinbart. Im Normalfall fallen diese Kosten monatlich oder jährlich an, wobei diese verhältnismäßig kleiner als die von On-Premise Lösungen sind.

Ebenfalls ein wichtiges Stichwort hier ist Datenkonsistenz. Im Fall von komplexen Prozessen ist die Konsistenz der Daten ein wesentlicher Punkt, um drastische Probleme zu verhindern. Gerade bei dezentraler Speicherung und Verarbeitung der Daten, ist die Synchronität sehr wichtig. Bei Cloud Computing ist dieses Risiko minimiert, da im Normalfall die Daten, auch bei Zugriff von unterschiedlichen Schnittstellen, synchron sind.

2.1.3 Hindernisse für den Einsatz einer Cloud

Da es sich beim Cloud Computing um ein neues Modell handelt, besteht eine gewisse Unsicherheit, wie man auf allen Ebenen eine solide Sicherheit erreichen kann. Deshalb wird die Fähigkeit der Cloud, den Datenschutzbestimmungen gerecht zu werden, in Frage gestellt. Das Grundprinzip der Cloud sieht eine dauerhafte Verfügbarkeit vor und gerade diese Verfügbarkeit kann auch ein großer Nachteil sein. Erst durch den offenen Zugang zu den zur Verfügungen gestellten Rechenleistung und anderen Ressourcen, entfaltet das CC-Model sein volles Potenzial.

Heutzutage müssen Anwendungen dauerhaft erreichbar sein und hierbei kann sich die Abhängigkeit von einem Cloud Service Provider negativ auf dauerhafte Konnektivität

auswirken. Im Fall eines Ausfalls oder von Aussetzern müssen Notfallpläne- und oder Ressourcen gestartet werden um, zum Beispiel Datenverlust zu verhindern, was im Allgemeinen, zu fatalen Fehlern führen kann.

Die Interoperabilität und Übertragbarkeit von Informationen zwischen privaten und öffentlichen Clouds sind entscheidende Voraussetzungen für die breite Einführung von Cloud Computing in Unternehmen. Viele Unternehmen haben erhebliche Fortschritte bei der Standardisierung ihrer Prozesse, Daten und Systeme durch die Einführung von ERPs gemacht. Dieser Prozess wurde ermöglicht durch skalierbare Infrastrukturen, zur Schaffung einzelner Instanzen oder hochintegrierter Verbindungen zwischen Instanzen, um die Konsistenz von Stamm- und Bewegungsdaten zu verwalten und zuverlässige konsolidierte Informationen zu erzeugen.

2.1.4 Cloud Computing Modelle

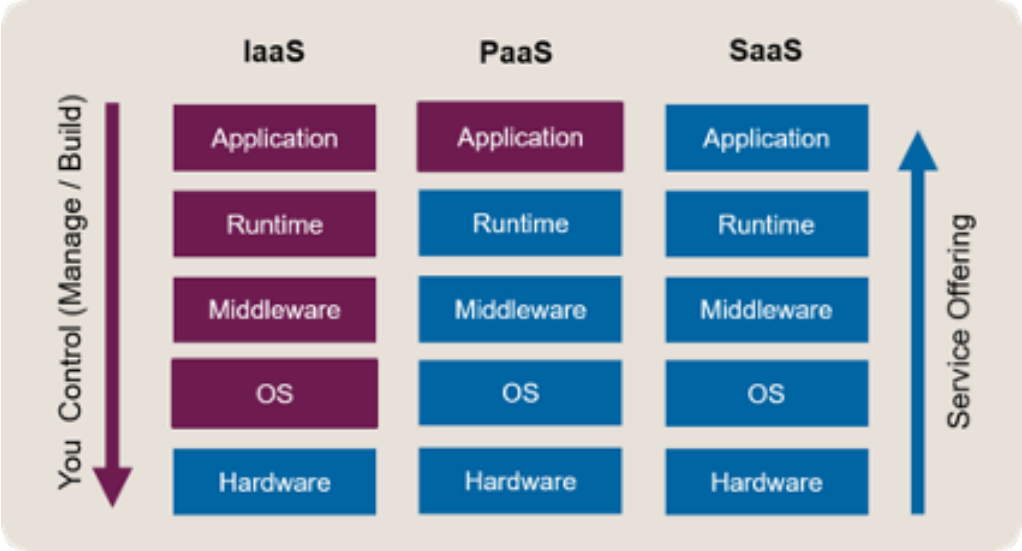


Abbildung 12: Modelle

Infrastructure as a Service (IaaS)

Infrastructure as a Service stellt sowohl virtuelle und physische Server als auch Netzwerk- und Speicherressourcen nach Verlangen des Verbrauchers zur Verfügung. Der Nutzer hat volle Kontrolle über Ressourcen in Bezug auf Speicher und Rechenleistung. Die Auswahl des Betriebssystems wird ebenfalls dem Nutzer überlassen. Beispiel dafür sind Amazon EC2-Cluster und Microsoft Azure. Cloud Storage ist in ähnlicher Weise ein spezieller Fall von IaaS.

Platform as a Service (PaaS)

Diese Form des Cloud Computing gibt den Kunden die Möglichkeit, seine Anwendungen auf einer, vom Dienstleistungsgeber, gehosteten Plattform zu entwickeln, bereitzustellen und zu verwalten. Bei diesem Modell handhabt der Dienstleister sämtliche Ressourcen, wie z.B.: Speicher und Rechenleistung. Die Anwendung wird den potenziellen Nutzern über APIs (Application Programming Interface) zur Verfügung gestellt. Hierbei ist wichtig das der Verbraucher keinerlei Kontrolle über die zugrunde liegende Infrastruktur hat. Bekannte Beispiele hierfür sind Web-Hosting Dienste, wie Microsoft Azure Web und Amazon Web Services.

Software as a Service (SaaS)

Bei SaaS wird den Nutzern die Möglichkeit gegeben auf eine, vom Dienstleister in der Cloud Infrastruktur bereitgestellten Anwendung zu zugreifen und zu benutzen. Benutzer können dann über eine webbasierte Schnittstelle oder andere, wie ftp und clientseitige Schnittstellen darauf zugreifen. Solche Anwendungen werden gegen monatliche oder jährliche Zahlungen dem Benutzer bereitgestellt. Dabei hat der Verbraucher aber keine bis wenig Kontrolle über im Hintergrund geregelte Ressourcen. Beispiele hierfür sind Microsoft Office 365, Microsoft Skype und Google Apps.

2.1.5 Anwendungsfälle von Cloud Computing

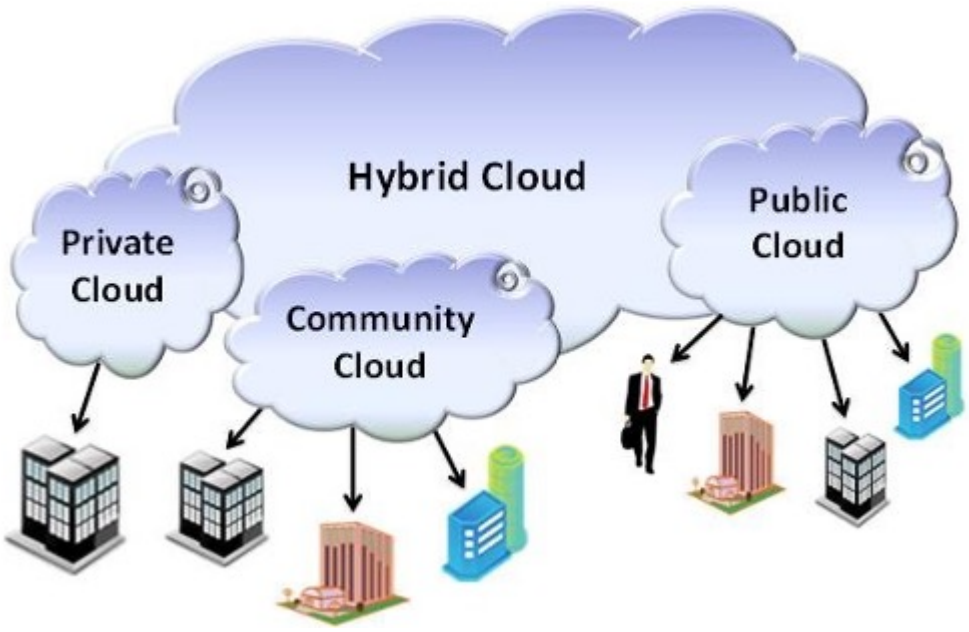


Abbildung 13: Anwendungsfälle

Private Cloud

Dieses Bereitstellungsmodell wird vor allem in Unternehmen verwendet, d.h. diese Dienste sind nicht für die Öffentlichkeit zugänglich, sondern nur Unternehmensintern. Die Nutzer können standortunabhängig von diesen Cloud Computing Diensten Gebrauch machen, aber müssen Teil der gleichen Organisation sein. Die private Cloud ist die sicherste der Bereitstellungsmethoden, da Prozesse innerhalb des Unternehmens kontrolliert und verwaltet werden, ohne jegliche Leistungs- Sicherheitsbeschränkungen, die Dienste in der Public Cloud vielleicht benötigen. In der privaten Cloud ist es möglich, dass die fundamentale Infrastruktur der Cloud vom Unternehmen selbst, von Drittanbietern oder von Beiden verwaltet wird. Generell wird die Private Cloud in zwei unterschiedliche Arten unterteilt:

- On-Premise Private Cloud

  Diese Art wird auch interne Cloud genannt. Sie bietet zusätzliche Sicherheit aber kann jedoch in Größe und Skalierbarkeit stark eingeschränkt sein, da man selbst das Kapital für Hard- und Software, sowie Wartung und Instandhaltung aufbringen muss. Die interne Cloud eignet sich somit für Anwendung, die volle Kontrolle sämtlicher Ressourcen verlangen.

- extern gehostete Private Cloud

  Hier wird das Hosten der Cloud von einem Drittanbieter übernommen. Diese Anbieter fördern eine restriktive Cloud Computing-Umgebung mit vollständiger Vertraulichkeit. Diese Art von der Private Cloud wird für Institutionen empfohlen, die das Kapital für eine interne Private Cloud nicht aufbringen können und die Risiken der gemeinsamen Nutzung physischer Ressourcen der Public Cloud eingehen wollen.

Public Cloud

Eine Public Cloud ist für die Öffentlichkeit zugänglich und kann von einem Unternehmen, einer staatlichen Einrichtung oder Organisation verwendet werden. Die Cloud ist jedoch im Besitz eines Drittanbieters. Hierbei greift der Nutzer über Schnittstellen auf die CC-Dienste des Cloud-Besitzers zu.

Community Cloud

In der Community Cloud wird die Infrastruktur einer Cloud von mehreren Unternehmen oder Institut mit ähnlichen oder gemeinsamen Zielen verwendet. Die Infrastruktur kann entweder von einer oder mehreren Organisationen verwaltet werden.

Hybrid-Cloud

Diese Variante ist eine Kombination aus einer oder mehreren private/n und öffentliche/n Cloud, die aber als getrennte Einheiten fungieren. Diese Einheiten werden durch eine Standardisierung und Protokolle verbunden. Dieses Modell wird hauptsächlich verwendet, wenn man die Vorteile der unterschiedlichen CC-Bereitstellungsoptionen kombinieren möchte, z.B.: ein Unternehmen möchte die Datenspeicherung auf einer privaten Cloud realisieren und andere Aufgaben mithilfe einer öffentlichen Cloud erledigen.

Azure Cloud

Microsoft Azure ist ein Cloud-Computing-Dienst von Microsoft. Azure bietet eine Reihe von Software as a Service (SaaS), Plattform as a Service (PaaS) und Infrastruktur as a Service (IaaS) Optionen für die Bereitstellung von Anwendungen und Diensten auf einer von Microsoft verwalteten Rechenzentrumsinfrastruktur. Mit 50 Betriebsregionen bietet Azure mehr als jeder andere Cloud-Anbieter.

2.2 Praxis: AI Builder

Basierend auf Azure AI Cognitive Service ist AI Builder ein Tool zum Erstellen und Trainieren von Modellen ohne das Schreiben von Code. Die Integration mit Power Apps und Power Automate ist eine integrierte Funktion, die den Nutzern die Möglichkeit bietet, bestehende Geschäftsanwendungen zu erweitern und zu verbessern. Microsoft Power Plattform ist eine Low-Code-Plattform, die es Unternehmen ermöglicht, Geschäftsprozesse zu automatisieren. Power Plattform umfasst drei Hauptprodukte: Power BI, PowerApps und Flow. Der AI Builder ermöglicht es auf einfache Weise Prozesse zu automatisieren und Ergebnisse vorherzusagen. AI Builder ist eine schlüsselfertige Lösung, die die Leistungsfähigkeit der künstlichen Intelligenz von Microsoft mit wenigen Mausklicks nutzbar macht. Mit dem AI Builder können Sie Ihren Anwendungen Intelligenz hinzufügen, auch wenn Sie keine Programmier- oder Data-Science-Kenntnisse haben.

2.2.1 Benutzerdefinierte Modelle

Der erste Schritt bei der Erstellung eines KI-Modells besteht darin, festzustellen, ob für meinen Anwendungsfall bereits vorgefertigte oder bereits trainierte Modell vorhanden sind. Falls dies nicht der Fall ist, stehen in der Benutzeroberfläche des AI Builders fünf Modelle zur Verfügung:

1. Category Classification
2. Entity Extraction
3. Form Processing
4. Object Detection
5. Prediction

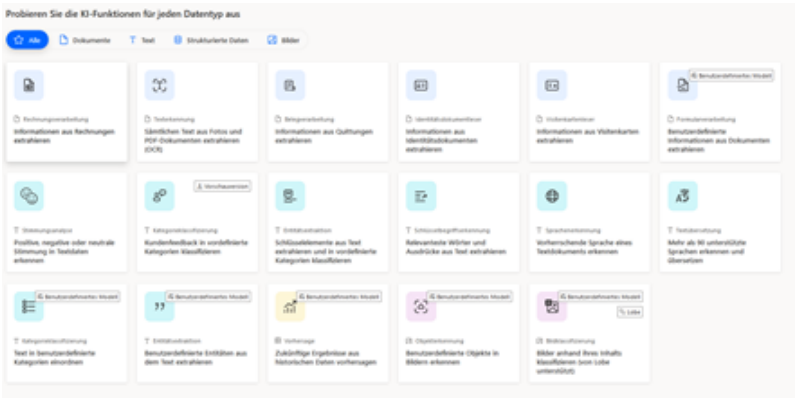


Abbildung 14: AI-Builder Modelle

Category Classification

Bei diesem Ansatz wird ein Modell verwendet bzw. trainiert um große Mengen an Textdaten, Dokumenten oder sonstige Textdatenquellen zu analysieren und den Text zu klassifizieren. Besonders hilfreich ist dieses Modell, um Spam zu identifizieren und entsprechend zu behandeln. Zuerst muss das Modell mit Trainingsdaten trainiert werden, mit dem Text und den Tags in zwei Spalten in der gleichen Tabelle. Das Zeichenlimit für jede Textprobe liegt bei fünftausend Zeichen.

Die Analysen, die dieses Modell liefert, können auch als Input für andere KI-Lösungen verwendet werden. Wichtig hierbei ist das für jeden Tag mindestens zehn Textproben bereitgestellt werden, ansonsten sinkt die Wahrscheinlichkeit ein genaues Ergebnis zu erzielen.

Entity Extraction

Hier werden wichtige Textelemente identifiziert und den definierten Kategorien zugeordnet. Ergebnisse werden dabei, entsprechend den Anforderungen, standardisiert und strukturiert. Auch hier werden wieder mindestens 10 Datensätze benötigt, um mit dem Trainieren des Modells zu beginnen. Das Modell ist anpassbar, indem man neue Entitätstypen mit wenigen Trainingsdaten erstellen oder bestehende Entitätstypen modifizieren. Genauer gesagt verfügt der Ai Builder über vorgefertigte Trainingsdaten, die zur Erweiterung der eigenen Trainingsdaten verwendet werden können.

Form Processing

Die Formularverarbeitung ist das KI-Modell, das Daten aus Formularen, auch aus Papier- oder PDF-Dokumenten, extrahiert. Bei diesem Modell benötigt man fünf Beispielformulare, um es zu trainieren, die Felder eines Dokuments zuzuordnen und eine funktionierende Anwendung zu erstellen. Diese Lösung wird verwendet um Rechnungen, Aufträgen, Reklamationen, etc. zu erfassen. Es ist zum Beispiel möglich, das Modell zu trainieren und einen Ablauf zu erstellen, der automatisch Schlüsselinformationen aus Bestelldokumenten erkennt und extrahiert und anschließend eine E-Mail an den zuständigen Mitarbeiter sendet.

Die empfohlenen Formate für Eingabedaten sind .jpg, .png und .pdf. Die Gesamtgröße der, für das Training verwendeten, Dokumente darf insgesamt 50 MB nicht überschreiten.

Object Detection

Die Objekterkennung wird dazu verwendet, um Objekte auf Fotos oder Videos zu erkennen. Dieses Modell kann verwendet werden, um Produkte oder Maschinen und dazu Informationen zu erhalten. Ebenfalls hilfreich kann dieses Modell bei mobilen Anwendungen sein. Zum Beispiel ein Mitarbeiter will den Bestand eines Produkts überprüfen oder benötigt Einsicht in eine dazu gehörige Betriebsanleitung.

Für das Training werden mindestens 15 Fotos von jedem Objekt benötigt; je mehr Fotos, desto genauer ist das Modell. Die Fotos sollten eine Vielzahl von Hintergründen mit den abgebildeten Objekten in unterschiedlichen Entfernungen und Winkeln enthalten, um die korrekte Identifizierung der Objekte zu verbessern. Es ist zu beachten, dass die Trainingsbilder im .jpg-, .bmp- oder png-Format vorliegen müssen und insgesamt

6 MB pro Training nicht überschreiten dürfen, wobei die Fotos nicht kleiner als 256 x 256 Pixel sein dürfen. Beständigere Ergebnisse sind möglich, wenn das Verhältnis zwischen den Objekten mit den wenigsten und den meisten Bildern mindestens 1:2 beträgt. Anders ausgedrückt: Wenn 500 die höchste Anzahl von Trainingsbildern für ein Objekt ist, dann muss es mindestens 250 Trainingsbilder für das Objekt mit den wenigsten Bildern geben.

Prediction

Bei diesem Modell werden große Mengen an alten Daten analysiert, um darin Muster zu erkennen. Dieses „Wissen“ wird dann verwendet, um diese Muster in neuen Datensätzen zu erkennen und Vorhersagen zu treffen. Diese Mechanismen können Muster aufdecken, die binäre Fragen (ja/nein), Fragen mit mehreren Antworten (eine Liste von Ergebnissen) oder Fragen, die mit einer Zahl beantwortet werden.

Zum Trainieren des Modells werden mindestens 10 Zeilen mit historischen Werten für jede Klasse der Datenspalte "Label"benötigt. Die Mindestanzahl der Zeilen für das Training beträgt 50, aber ein Minimum von 1.000 Zeilen gewährleistet die besten Ergebnisse.

2.2.2 AI-Builder in der Praxis

Um den AI-Builder für den gewünschten Use-Case am besten zu nutzen, hat sich das Diplomarbeitsteam dazu entschieden, das Modell des Form Processing zu verwenden. Der gewünschte Use-Case ist, mehrere im Vorhinein definierte Felder aus einer Eingangsrechnung, im PDF, zu extrahieren und anschließend zu markieren. Folgende Schritte sind notwendig, um ein Modell zu erstellen, dass auf spezielle Eingangsrechnungen (siehe Abbildung 15) trainiert ist:

1. Zu extrahierende Felder definieren
2. Trainingsdaten bereitstellen
3. Trainingsdaten mit Tags versehen
4. Modell trainieren und
5. Modell veröffentlichen

Musterfirma A
4020 Musterstrasse a
Musterstadt A
Musterland A

RECHNUNG
12345

Datum: Oct 21, 2020
Fälligkeitsdatum: Nov 11, 2020
Saldo fällig: 1.270,00 €

Rechnung an:
Musterfirma B
4020 Musterstrasse b
Musterstadt B
Musterland B

Artikel	Menge	Rate	Betrag
Musterprodukt A	3	400,00 €	1.200,00 €
Musterprodukt B	10	7,00 €	70,00 €

Zwischensumme: 1.270,00 €
MwSt (0%): 0,00 €
Gesamt: 1.270,00 €

Abbildung 15: Beispiel Eingansrechnung

Zunächst ist es wichtig zu wissen, welche Felder man aus der ER extrahieren will. In der folgenden Abbildung wird gezeigt welche Felder bei dieser Arbeit ausgewählt wurden. Diese Felder wurden nur zu Veranschaulichung der Fähigkeiten des AI-Builders ausgewählt und nicht zur firmeninternen Nutzung:

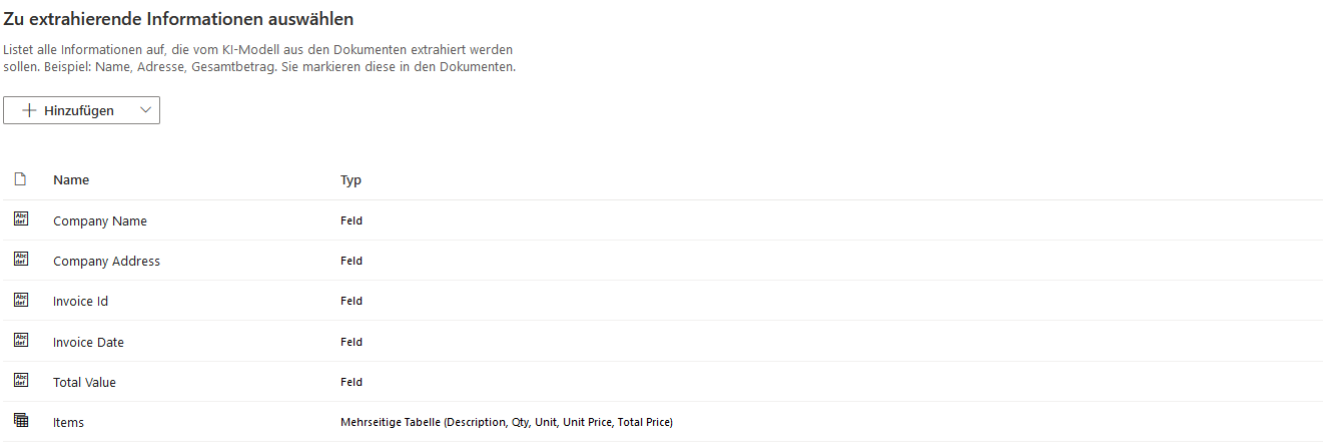


Abbildung 16: AI-Builder Felder

1. Company Name
2. Company Address
3. Invoice Id
4. Invoice Date

5. Total Value
6. Items (Mehrseitige Tabelle)

Items (Mehrseitige Tabelle): Diese Tabelle bietet ein experimentelles Feature, um tabellarische Daten aus einer, in der Rechnung vorhandenen Tabelle zu entnehmen, auch wenn sich die Tabelle über mehrere Seiten zieht. Diese Tabelle enthält wiederrum eigene Daten wie:

1. Description
2. Qty (Quantity)
3. Unit
4. Unit Price
5. Total Price

Um Daten für das Trainieren des Modells bereitzustellen, ist es notwendig mindestens fünf Beispieldokumente hochzuladen. Nichtsdestotrotz ist für eine hohe Genauigkeit des Modells, sprich wie sicher sich der Algorithmus ist, dass er das richtige Feld markiert hat, von Vorteil mehrere ähnliche Dokumente bereitzustellen. Um dem Algorithmus anzutrainieren, welche Felder in dem Dokument vorhanden sind, ist es relevant die Daten in dem Dokument händisch zu markieren und die zuvor definierten Informationsfelder zu zuweisen. Siehe folgende Abbildung (17)

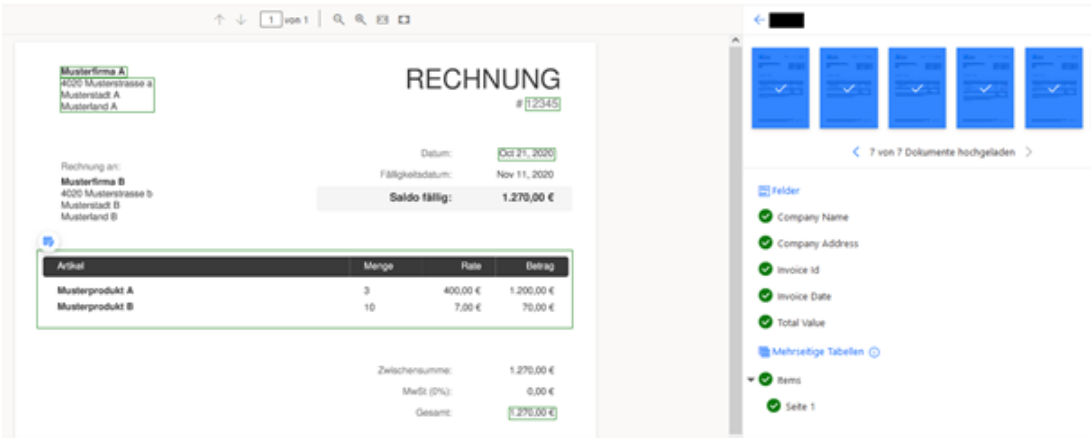


Abbildung 17: AI-Builder Felder Tagging

2.3 Praxis: Power Automate

Power Automate ist ein Teil der Microsoft Power-Platform, mit der man Cloud-basierte Automatisierungsprozesse erstellen und verwalten kann. Der Benutzer hat Zugriff auf verschiedene Dienste von Microsoft und kann ohne oder wenig Code einen Workflow erstellen. Power Automate reagiert auf einen im Vorhinein definierten Trigger und führt eine Reihe von vordefinierten Schritten aus. Dieser Trigger kann eine neue E-Mail, ein neuer Datensatz im CRM eingefügt wird oder einfach die Erstellung eines neuen Dokuments im SharePoint sein. Somit ist es möglich auch ohne jegliche Vorkenntnisse oder fachspezifisches Wissen eine einfache Anwendung zu erstellen. Zum Beispiel kann beim Eingang einer E-Mail, die eine Kundenrechnung enthält, an den AI-Builder weitergegeben werden und dort mit einem vortrainierten Modell, die wichtigsten Daten herauslesen und dann erneut mit einer E-Mail versendet werden.

2.3.1 Cloud Flow

Da bei dieser Arbeit nur der Cloud Flow in Verwendung ist, wird hier nur auf diesen genauer eingegangen. Power Automate bietet hier drei verschiedene Arten von Cloud Flows an:

1. Automated Flow
2. Instant Flow
3. Scheduled Flow

Automated Flow

Dieser Flow wird ausgelöst, wenn eine gewünschte Kondition eintritt. Solche Trigger können, der Eingang einer E-Mail einer speziellen Person, eine Teams-Nachricht oder wenn ein Dokument in OneDrive geändert wird. Sogenannte Connectors werden verwendet, um eine Folge von Schritten zu bilden, um den gewünschten Effekt zu erzielen.

Instant Flow

Ein Instant Flow wird mit dem Klicken eines Buttons gestartet. Dieser Flow läuft sowohl auf Mobile als auch auf Desktop Devices. Er wird verwendet, um ein breites Spektrum von Aufgaben zu erledigen, wie die Beantragung einer Genehmigung via Teams oder SharePoint.

Scheduled Flow

Dieser Flow wird nach einem Zeitplan automatisiert und zu einem gewünschten Datum oder beliebiger Uhrzeit ausgeführt werden. Dieser Ablauf ist hilfreich, um zum Beispiel jeden Tag zur selben Uhrzeit einen Daten-Upload zu machen.

Cloud Flow in der Praxis

Die Diplomanden dieser Arbeit entschieden sich für einen Automated Flow, da dieser die gewünschte Funktion erfüllt. Wie bereits erwähnt, ist der Automated Flow ein Event-basierter Flow und wird durch das Erhalten einer E-Mail ausgelöst (Siehe Abbildung 18).

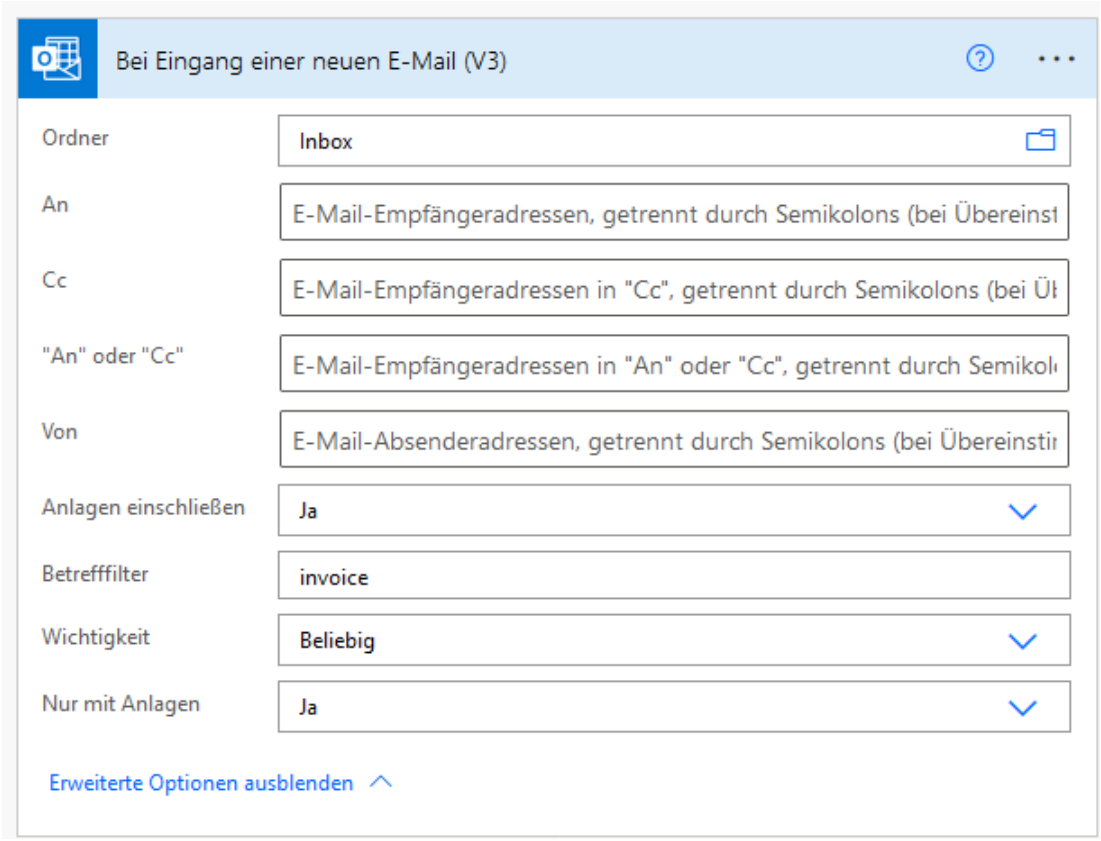


Abbildung 18: Power Automate Flow Trigger

Um alle Anlagen der E-Mail zu verarbeiten, wird eine Schleife um die nächsten Schritte gesetzt. Da es erlaubt ist mehrere Eingangsrechnungen in der E-Mail anzuhängen muss jeder Anhang einzeln verarveitet werden. Das zuvor definierte AI-Builder Modell extrahiert die notwendigen Information aus dem Dokument (Siehe Abbildung 19).

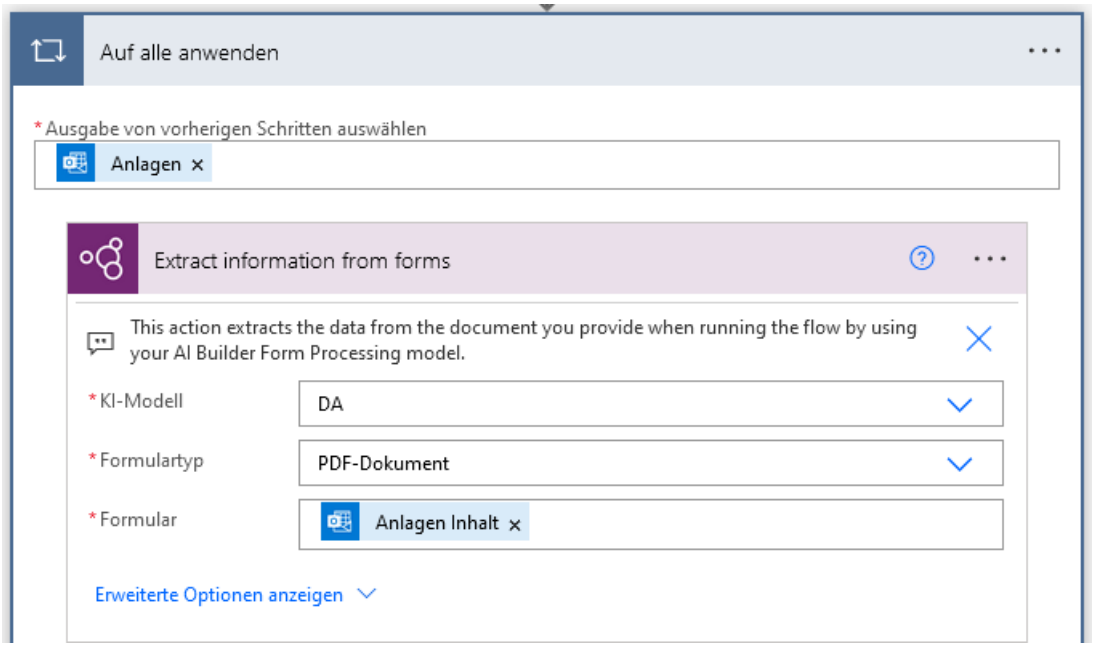
q 

Abbildung 19: Verarbeitung des Dokuments

Um die extrahierten Daten zu persistieren, wurde eine Dynamcis CRM Testumgebung aufgesetzt, um einen ecten Geschäftsfall zu simulieren. Es wurden zwei Entitäten, Invoice und Invoice Items erstellt (20). Da die Entität Invoice Items von der Entität Invoice abhängig ist, muss die die Entität Invoice Items eigens gespeichert werden.

Kurzer Exkurs; Microsoft Dynamics CRM: Customer Relationship Management (CRM) beinhaltet eine Reihe von Software-Tools, die für das Verwalten der drei Pfeiler zwischen Kunde und Unternehmen zuständig sind: Marketing, Verkauf und Dienstleistungen. Neben Sammeln von Kudendaten aus verschiedenen Quellen und Automatisieren von sich wiederholenden Vertriebs-, Marketing- und Kundendienstprozessen, fördert eine CRM-Software auch die Abteilungsübergreifende Zusammenarbeit. Diese CRM-Software lässt sich in zwei Arten unterscheiden:

1. On-Premise CRM-Software
2. Cloud-basierte CRM-Software

On-Premise CRM-Software

Da das Hauptgeschäft von Unternehmen wie Gesundheits- und Finanzeinrichtungen, der Umgang mit sensiblen Kundendaten ist, wird aus Sicherheitsgründen eine On-Premise Lösung, sprich eine CRM-Software vor Ort, bevorzugt. Diese Systeme sind aber mit

hohen Wartungs- und Entwicklungskosten verbunden, und die Unternehmen sind selbst für Sicherheit und Datenpflege zuständig.

Cloud-basierte CRM-Software

Unternehmen haben die Möglichkeit die CRM-Software und die damit verbundenen Aufwände, in die Cloud zu verlagern. Im Gegensatz du der On-Premise Lösung ist die Cloud-Variante flexibler in Hinsicht auf Speicher- und Rechenressourcen.

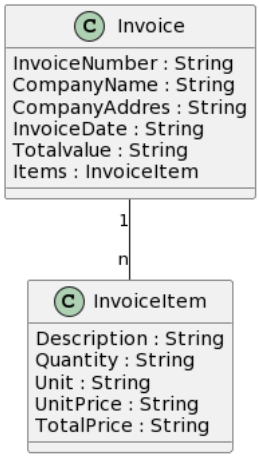


Abbildung 20: Klassendiagram Invoice

Um die Richtigkeit des Datenmodells zu sichern muss die Reihenfolge der Persistierung beachtet werden und alle Invoice Items abhängig von der dazugehörigen Invoice gemacht werden 21.

2.2.2

2.4 Praxis: Cognitive Services

Die Azure Cognitive Services sind ein Teil der Cloud-basierten Dienste von Microsoft. Mithilfe von REST-APIs und Client-SDKs, ist es möglich kognitive Intelligenz in ihre Applikation einzubauen. Ein großer Vorteil ist, dass man dafür wenig bis keine Erfahrung im Bereich der künstlichen Intelligenz und Data Science benötigt. Die, in den Azure Cognitive Services, beinhaltete Sammlung an kognitive Funktionen ermöglicht es Lösungen zu erstellen, die menschliche Fähigkeiten wie sehen, sprechen und hören nachahmen.

Kurzer Exkurs; REST-API, SDK:

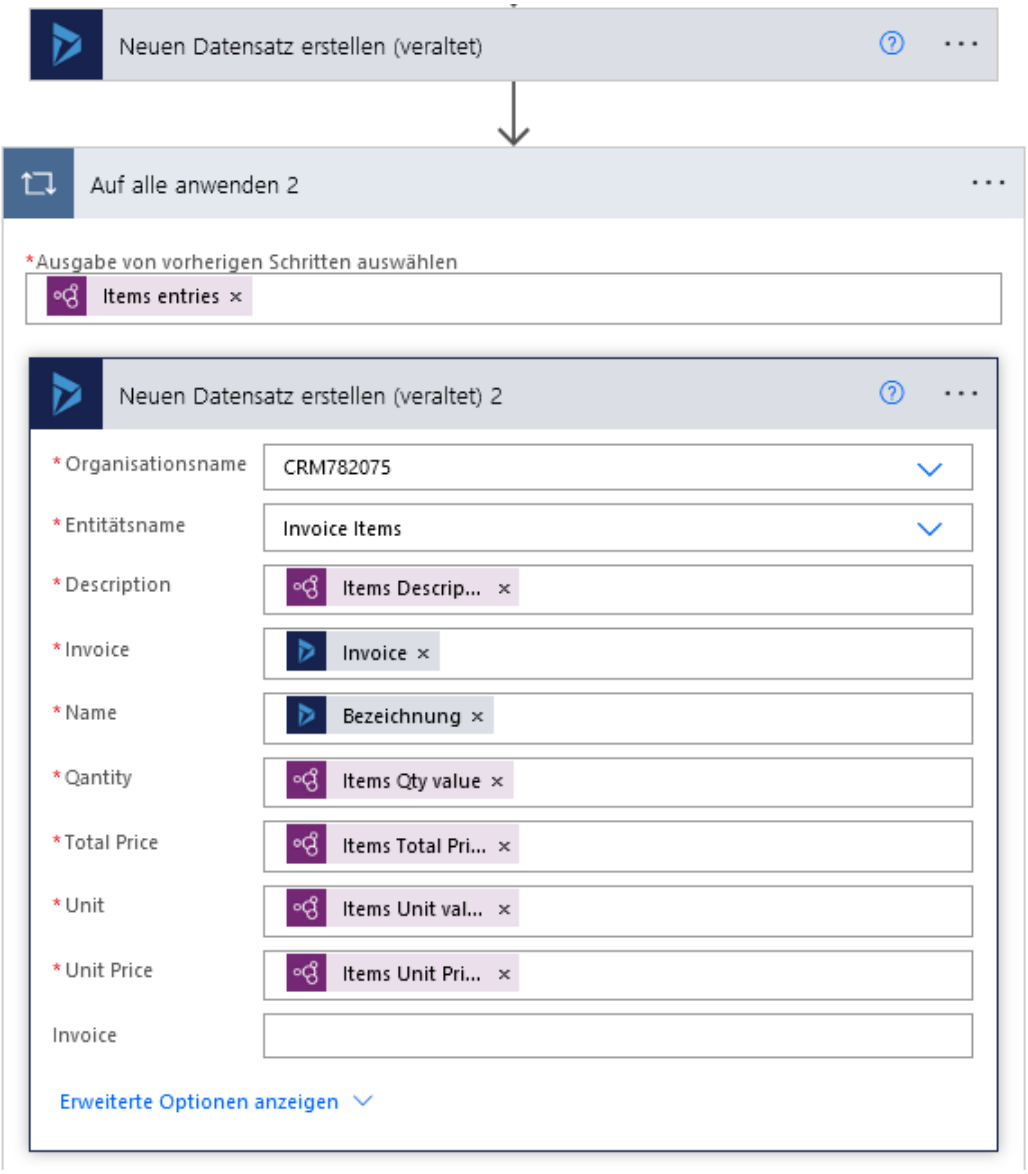


Abbildung 21: Persistierung der Entitäten

1. Stabile und interaktive Anwendungen setzten voraus, dass Programme untereinander barrierefrei kommunizieren können. Eine API, Application Programming Interface, definiert regeln, die diese Kommunikation erleichtert. Unter diesen Programmen kann man entweder Softwarebibliotheken, ein Betriebssystem oder einen Webserver verstehen. Der entscheidende Vorteil von APIs sind, dass das anfragende Programm keinerlei Informationen über das dahinterstehende System des Antwortgebers haben muss.

   REST-API: Einer der bekanntesten Architekturen einer API ist REST. Representational State Transfer ist ein Stil für die Entwicklung Anwendungen, die untereinander auf irgendeine Weise vernetzt sind. Es wird oft für die Entwicklung von Web-APIs verwendet. Wichtig zu wissen ist, dass REST zustandslos ist. In diesem Kontext bedeutet zustandslos, dass Anfragen des Clients alle notwendigen

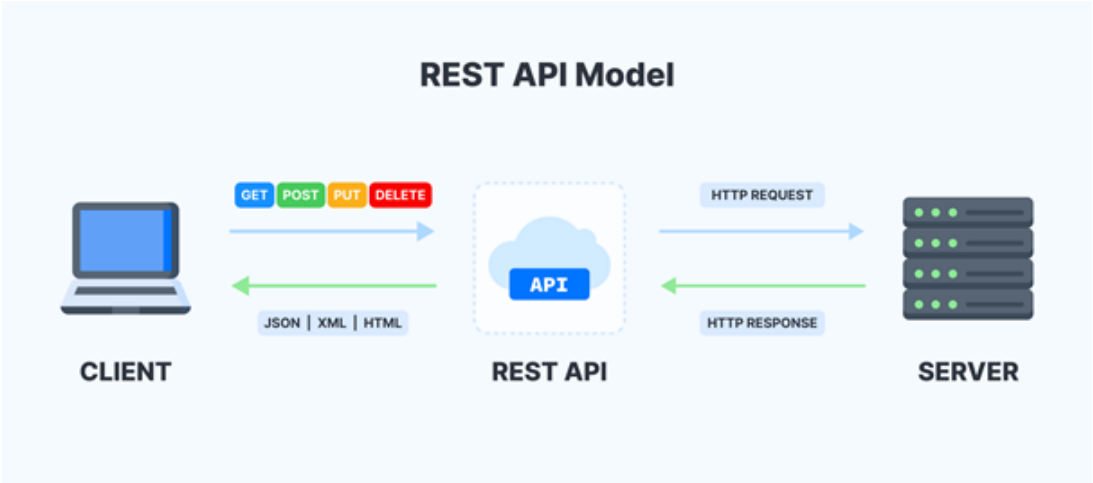


Abbildung 22: Funktionsweise von REST

Informationen für den Server besitzen muss. Zum Beispiel kann der Client nicht davon ausgehen, dass sich der Server an die vorherigen Requests des Clients erinnern kann. REST-APIs senden im Normalfall HTTP-Anfragen wie GET, POST, PUT, etc.

2. Ein **Software Development Kit**, kurz SDK, ist eine Sammlung aus mehreren Tools, die von einem Hersteller einer Programmiersprache, eines Betriebssystems oder einer Hardwareplatform zur Verfügung gestellt werden. Diese Tools können Debugger, Frameworks oder eine Sammlung aus Codebibliotheken sein.

2.4.1 Cognitive Computing

Cognitive Computing ist ein intelligentes System, das durch umfangreiches Lernen und zielgerichtetes Denken mit Menschen in ihrer natürlichen Form spricht und diese nachahmt. Cognitive Computing ist die dritte Ära der Informatik, und gleichzeitig hat Cognitive Computing sowohl in der Wissenschaft als auch in der Industrie breite Aufmerksamkeit auf sich gezogen. Die Kombination aus maschineller und menschlicher Intelligenz kann die komplexesten Probleme der Welt lösen. Die Verarbeitung natürlicher Sprache mit Sentimentanalyse, künstlicher Intelligenz (KI), maschinellem Lernen und neuronalen Netzen ist der Eckpfeiler von Cognitive-Computing-Prozessen, die Probleme wie Menschen lösen können. Heutzutage wenden fortschrittliche Technologien Cognitive Computing in vielen Bereichen an. Angesichts der heutigen Datenexplosion und der sich schnell ändernden Geschäftsumgebungen können kognitive Systeme intelligente, fließende und verbesserte Mensch-Maschine-Interaktionen effektiv angehen. Künstliche Intelligenz wird in vielen Anwendungen wie Alexa, dem Sprachassistenten von Amazon,

Netflix und den Algorithmen von Amazon verwendet, um den nächsten Film oder Kauf vorzuschlagen. Einige Beispiele für persönliche Assistenten, die Cognitive Computing verwenden, sind Alexa, Siri, Google Assistant und Cortana. Die Weiterentwicklung dieser Technologie und ihre Übernahme im öffentlichen und privaten Sektor wird die Zukunft des Cognitive Computing aufgrund technologischer Entwicklungspfade und -trends stark beeinflussen. Kognitive Systeme müssen in Geschäfts- und breiten Anwendungen anpassungsfähig, interaktiv, iterativ, zustandsbehaftet und kontextbezogen sein. Zu den Anwendungen, die von solchen Technologien mit Cognitive Computing profitieren können, gehören Finanz- und Investmentfirmen, Gesundheitswesen und Veterinärmedizin, Reisen und Tourismus, Gesundheit und Wellness, Bildung und Lernen, Landwirtschaft, Kommunikations- und Netzwerktechnologie.

2.4.2 Cognitive Computing und Cloud Computing

Cloud Computing virtualisiert die Datenverarbeitung, den Speicher und die Bandbreite. Dadurch werden die Kosten für die Bereitstellung von Softwarediensten gesenkt und die Industrialisierung sowie die Förderung der Anwendung des Cognitive Computings unterstützt. Darüber hinaus bietet die hohe Rechen- und Speicherkapazität des Cloud Computing dynamische, flexible, virtuelle, gemeinsam genutzte und effiziente Rechenressourcendienste für das Cognitive Computing.

Nachdem die Big-Data-Analyse auf der CC-Plattform durchgeführt wurde, werden Technologien wie maschinelles Lernen eingesetzt, um Daten zu analysieren und die Ergebnisse in verschiedenen Bereichen anzuwenden. Die verschiedenen Kategorien von Informationen entsprechen unterschiedlichen Verarbeitungstechnologien. So entsprechen beispielsweise die wörtlichen Informationen der natürlichen Sprachverarbeitung und die bildlichen Informationen dem maschinellen Sehen. Der kognitive Dienst von IBM für Sprache und die kognitive Computing-Anwendung von Google legen den Schwerpunkt auf die Verwirklichung von gehirnähnlicher Wahrnehmung und Urteilsfähigkeit durch den Einsatz eines Cloud-Service-Modells, um präzise Entscheidungshilfen zu bieten. Cloud Computing und das Internet of Things (IoT) bieten eine Software- und Hardwarebasis für Cognitive Computing, während die Big-Data-Analyse Methoden und Denkweisen zur Entdeckung und Erkennung neuer Möglichkeiten und neuer Werte in Daten bereitstellt.

3 Ausgangslage

3.1 Ausgangssituation

3.2 Istzustand

3.3 Problemstellung

3.4 Ziele

3.5 Aufgabenstellung

3.5.1 Funktionale Anforderungen

3.5.2 Nicht funktionale Anforderungen

3.6 Systemarchitektur

3.7 Ablauf

4 Umsetzung

4.1 Frontend

4.1.1 Verwendete Technologien

HTML

Mit der steigenden Nachfrage von Webseiten in den 1990er wurde Hypertext Markup Language (HTML) als strukturierte Möglichkeit vorgestellt, um grafische Webseiten zu erstellen.

Die Basis jeder Webseite ist ein HTML-Gerüst, welches die Struktur und Einzelteile angibt, dabei umfasst der HTML-Standard Überschriften, Paragrafen, Tabellen und Listen. Zusätzlich ermöglicht HTML das Inkludieren von Fotos, Videos als auch von anderen Dokumenten (PDF-Dateien, Webseiten). [17]

Um das Arbeiten mit HTML zu vereinfachen, wurden genau wie bei anderen Technologien für jedes Programm unterschiedliche Plugins eingebunden. Mit der Zeit wurden die Plugins in den Standard eingefasst und damit entwickelten sich über die Jahre unterschiedliche Versionen. Seit Oktober 2014 ist HTML5 als allgemeiner Grundsatz definiert und erhielt seit jeher keine weitere Versionsnummer, da man Änderungen als ”living standard” weiterentwickelt. [18]

Ein HTML-File (.html, .htm) besteht aus HTML-Tags, welche mit einem Kleiner-Zeichen ”<” beginnen und einem Größer-Zeichen ”>” enden. Der Inhalt dazwischen unterscheidet sich je nach Komponente (p, h1, a, ...).

Neben dem eigentlichen Tag können HTML-Elemente noch zusätzlich Attribute beinhalten, welche zusätzlich Informationen liefern. Diese werden normalerweise als Name/Value-Paar im Starting-Tag aufgelistet. Dabei sind die meisten freiwillig, jedoch gibt es dafür auch Ausnahmen (src-Attribut im img-Tag). [19, 20]

In der Regel besteht jede Komponente aus zwei HTML-Tags, einem Starting-Tag und einem Ending-Tag, welches noch einen zusätzlichen Schrägstrich ”/” beinhaltet. Jedoch gibt es Self-Closing-Tags als Ausnahme, wo eine Komponente aus einem einzelnen

HTML-Tag besteht. Dazu gehört das `<img>`-Tag, da es nur aus einem Starting-Tag und Attributen besteht.

Dieses HTML-File nimmt die Struktur eines Baumes an, das heißt, dass es sich um eine hierarchische Abfolge von Komponenten handelt. Diese Struktur ähnelt der von Extensible Markup Language (XML), jedoch befolgt es nicht alle Richtlinien von XML. Wird diese Kompatibilität vorausgesetzt, muss man auf Extensible Hypertext Markup Language (XHTML) zurückgreifen. [17]

Zu den wichtigsten HTML-Tags gehören: [21]

`<h1>` bis `<h6>`	Überschrift in unterschiedlicher Reihung (1 = Hauptüberschrift, 2-6 = Unterüberschriften)
`<p>`	Paragraph
`<a>`	Hyperlink
`<img>`	Bild
`<ol/ul>`	Geordnete und ungeordnete Liste
`<li>`	Element einer Liste
`<strong>`	Hebt den Text in Fettschrift hervor
`<div>`	Block-Element
`<table>`	Tabelle
`<tr>`	Tabellenzeile

Tabelle 3: Beispiele für HTML-Tags

Zusätzlich könnten HTML-Elemente diese Attribute enthalten (Auszug):

`id`	Mit der Id kann ein HTML-Element direkt referenziert werden
`title`	Gibt Zusatzinformationen zu einem HTML-Element, welche als Tooltip dargestellt werden
`disabled`	Gibt an, ob ein HTML-Element benutzbar ist (benötigt keinen Value)

Tabelle 4: Beispiele für HTML-Attribute

Ein gewöhnliches HTML-File beginnt mit `<!DOCTYPE html>`, um den Browser mitzuteilen, welcher Datentyp zu erwarten ist. Diese Information unterscheidet sich je nach Version und bei XHTML. [22]

Ein HTML-File, welches einen Auszug aus den wichtigsten HTML-Tags beinhaltet, könnte so aussehen:

Listing 2: Beispiel für ein HTML-File

```
<!DOCTYPE html>
<html>

<head>
    <title>Titel - DA</title>
</head>

<body>
    <!-- Dies ist ein Zeilenkommentar, der sich ueber
       mehrere Zeilen ziehen kann. -->
    <h1>Ueberschrift 1</h1>
    <p>Ein Paragraph, der mit Abstand unter der
       Ueberschrift angezeigt wird.</p>
    <ul>
        <li>Elemente einer</li>
        <li>ungeordneten Liste.</li>
    </ul>
    <p>
        Ein Paragraph mit
        <br /> <!-- Dies ist ein Beispiel fuer ein
           Self-Closing Tag -->
        einer Unterbrechung.
    </p>
</body>

</html>
```

Wird das vorherige HTML-File in einem gewöhnlichen Browser geöffnet, wird die Struktur folgendermaßen interpretiert:

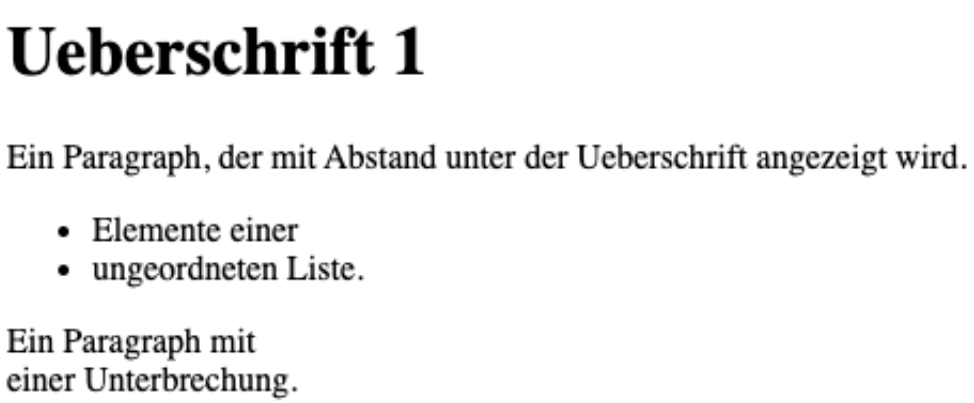


Abbildung 23: Beispiel für ein angezeigtes HTML-File

CSS

Um Webseiten zu gestalten, wird Cascading Style Sheets (CSS) verwendet, damit kann das Design von der generellen Struktur und dem Inhalt getrennt werden. Wobei CSS immer mit HTML assoziiert wird, kann es mit jeder auf XML basierenden Ausdruckssprache benutzt werden. [17]

Generell besteht eine CSS-Regel aus einem Key/Value-Pair, welches die Styling-Property (was soll geändert werden) und den Styling-Wert (auf was soll es geändert werden) setzt. Dabei wird der Styling-Wert je nach Styling-Property eingeschränkt.

Zusätzlich können mehrere CSS-Regeln zu einer zusammengefasst werden, diese Regeln werden als Shorthand bezeichnet (z.B.: `background`, `border`). Abhängig von der Werteanzahl wird ein bestimmtes Shorthand referenziert. Die Styling-Property `margin` kann folgendermaßen angegeben werden:

- Ein Wert: `margin: 10px`

  Dieser Wert wird für alle Seiten genutzt.

- Zwei Werte: `margin: 10px 5px`

  Der erste Wert wird für die obere und untere Seite genutzt und der zweite für die linke und rechte.

- Drei Werte: `margin: 10px 5px 15px`

  Der erste Wert wird für die obere Seite genutzt, der zweite Wert für die linke und rechte Seite und der letzte Werte für die untere.

- Vier Werte: `margin: 10px 5px 15px 20px`

  Die Werte werden im Uhrzeigersinn beginnend mit der oberen Seite vergeben.

Beispiele für gültige Properties:

`width/height`	`100px / 100%`	Gibt die Breite/Höhe eines HTML-Elements an (als Länge oder Prozentzahl)
`display`	`block / flex`	Gibt an wie die HTML-Elemente angezeigt werden
`margin/padding`	`10px`	Gibt den inneren/äußeren Abstand an
`background`	`#fff`	Gibt die Farbe/das Bild für den Hintergrund an
`color`	`red`	Gibt die Farbe/Strichstärke an

Tabelle 5: Beispiele für CSS-Properties

Um die Struktur mit dem Styling zu verbinden, bietet HTML drei Möglichkeiten an: [23]

- Direkt im HTML-File innerhalb vom `<head>`-Tag, mithilfe von einem `<style>`-Tag.

```
<head>
    <style>
        <!-- CSS-Regeln -->
    </style>
</head>
```

- Als Referenz zu einem separatem CSS-File (Dateinamenserweiterung: `.css`) wird ein `<link>`-Tag im `<head>`-Tag gesetzt. Wobei diese Möglichkeit es ermöglicht, die CSS-Regeln in mehreren HTML-Files zu nutzen und man damit Codeverdoppelung vermeidet.

```
<head>
    <link rel="stylesheet"
       href="pfad/zum/css-file.css">
</head>
```

- Direkt im HTML-Tag mit dem `style` Attribut.

```
<div style="<!-- CSS-Regeln -->">
```

Um eine CSS-Regel zuweisen zu können, werden sogenannte Selektoren angegeben, welche direkt ein HTML-Element ansprechen oder mehrere zusammenfassen. Gültige Selektoren sind:

*	*	Selektiert alle HTML-Elemente
element	h1	Selektiert alle HTML-Elemente mit dem angegebenen Elementnamen
.klasse	.nav	Selektiert alle HTML-Elemente mit dem angegebenen class-Attribut
#id	#main	Selektiert alle HTML-Elemente mit dem angegebenen id-Attribut

Tabelle 6: CSS-Selektoren

Um diese Spezifikation noch genauer zu definieren, können Selektoren mit unterschiedlichen Operatoren zusammengefügt werden.

,	h1, p	Selektiert alle <p> und <h1>-Elemente
" " (Leerzeichen)	div p	Selektiert alle <p>-Elemente in einem -Elemente
>	div > p	Selektiert alle <p>-Elemente, die direkt in einem -Elemente sind
(2 Selektoren ohne Operator)	div.main	Selektiert alle -Element, die auch die Klasse .main haben

Tabelle 7: CSS-Operatoren

SCSS Da die Funktionalitäten von CSS sehr eingeschränkt sind, wird oftmals als Weiterentwicklung Sassy Cascading Style Sheets (SCSS) genutzt. Dies ermöglicht das Vereinfachen von Regeln und macht diese generell besser überschaubar.

Zusätzlich zu den normalen Funktionen von CSS stellt SCSS zum Beispiel folgende zur Verfügung: [24]

- Variablen

  Um über mehrere Regeln einen gleichen Wert zu setzen, kann dieser Wert in einer Variable gespeichert werden. Dies ermöglicht ein leichtes Updaten an mehreren Stellen gleichzeitig.

- Nesting

  Da mit der Zeit die Selektoren sehr kompliziert werden, ermöglicht SCSS das Erstellen von Regeln in einer hierarchischen Abfolge.

- Modules

  Damit man Codeverdoppelung vermeidet, kann man weitere CSS-Files importieren.

Angular

Angular ist ein auf TypeScript basierendes Frontend-Framework, welches neben React, Express und Vue.js das beliebteste Framework auf dem Markt ist. [25] Dabei wird Angular sowohl für kleine Projekte in der Schule als auch große Projekte in weltbekannten Plattformen verwendet.

Components Das Frontend wird in Komponenten (Components) aufgeteilt und diese stellen unterschiedliche Elemente der Darstellung dar, welche auch mehrmals untereinander verwendet werden können, um repetitiven Code zu verhindern.

Jedes Component besteht aus

- einer Component-Klasse (TypeScript-File),
- einem HTML-Template und
- einem Stylesheet (CSS/SCSS-File).

Listing 3: Beispiel eines Angular-Components

```
1   @Component({
2       selector: 'custom-app',
3       templateUrl: './app.component.html',
4       styleUrls: ['./app.component.scss']
5   })
6   export class AppComponent {
7
8   title: string = "Meine eigene App!";
9
10  items: any[] = [
11      {
12          "text": "Das ist meine"
13      },
14      {
15          "text": "erste eigene Angular-App."
16      }
17  ];
18
19  // Restlicher Code
20  }
```

Wie man im obigen Beispiel sieht, müssen bei einem Component Felder gesetzt werden, damit die unterschiedlichen Files miteinander assoziierbar sind:

selector	verpflichtend	Gibt den HTML-Tag an, mit welchem man das Component in anderen aufrufen kann (Bsp.: <app-custom-app></app-custom-app>)
template/templateUrl	verpflichtend	Gibt entweder direkt den HTML-Code (template) an oder referenziert zu einem HTML-File (templateUrl)
styles/styleUrls	freiwillig	Gibt entweder direkt einen oder mehrere CSS-Codes (styles) an oder referenziert zu einem oder mehreren CSS-Files (styleUrls)

Tabelle 8: Parameter einer Komponente

HTML-Templates Neben der generellen Struktur des Frontends kann ein HTML-Template auch Daten aus dem Component anzeigen.

Listing 4: Beispiel fuer ein HTML-Template

```
1 <h1>{{title}}</h1>
2
3 <ul>
4     <li *ngFor="let item of items">{{item.text}}</li>
5 </ul>
```

Generell besteht die Struktur aus üblichen HTML-Tags, jedoch kann es zusätzlich noch Component-Tags oder auch programmierähnliche Operationen beinhalten. Hierbei unterscheidet man zwischen:

- Interpolation
- Property Binding
- Event Binding
- Two-Way Binding

Interpolation Bei einer Interpolation werden zwei geschwungene Klammern "{{}}" genutzt, um den Wert einer Variable im Frontend anzuzeigen. Dabei muss darauf geachtet werden, dass die Interpolation immer den String-Wert der Variable nutzt. [26]

Property Binding Mit einem Property Binding können genau wie bei der Interpolation Daten im Frontend angegeben werden, jedoch kann jeder Datentyp genutzt werden.

Event Binding Damit kann das Backend auf gewisse Aktionen (Klicken, Bewegung der Maus, Hover) im Frontend reagieren.

Two-Way Binding Die Mischung aus den Vorherigen ergibt das Two-Way Binding. Sowohl das Backend als auch das Frontend reagiert auf Änderungen der jeweils anderen Komponente.

CLI Eine Command Line Interface (CLI) ermöglicht das Nutzen von Funktion über die Kommandozeile, spezifisch für Angular wird darüber das Erstellen, Generieren und Verwalten gesteuert. Das Kommandowort für Angular lautet `ng` und darauf können unterschiedliche Funktionswörter folgen.

TypeScript

Laut der jährlichen Entwicklerumfrage von StackOverflow landet TypeScript nach Python auf dem zweiten Platz (mit 15,29%) der meistgesuchten Programmiersprachen. Damit liegt TypeScript mit 0,7% vor JavaScript, die Sprache auf der TypeScript basiert. [25]

TypeScript versucht JavaScript zu verbessern, indem es eine strikte Typisierung fordert. Damit beugt man dem einfachen Fehler vor, dass Werte einen anderen Typ enthalten, als der gefordert ist. Dies führt dazu, dass Typenfehler bereits in der Entwicklung gefunden werden und verhindert, dass diese Fehler überhaupt in der Produktivumgebung aufkommen.

Abgesehen davon sind die zwei Sprachen ident, daher ist es auch möglich, bereits bestehende Tools von JavaScript mit TypeScript zu verwenden. Das heißt, dass alle JavaScript-Libraries weiterhin genutzt werden können. Die beliebtesten Libraries sind zum Beispiel jQuery und Chart.js.

4.2 Backend

4.2.1 Verwendete Technologien

4.2.2 Python als Progrmmiersprache

Nach dem Scheitern seiner ersten Programmiersprache, entwickelte Guido van Rossum die Sprache Python, dabei wollte er alle Fehler, die er beim Entwickeln von ABC gefunden hat, verbessern. Daher basieren auch die Stukturen und Konventionen von Python auf Unix, ohne an Unix gebunden zu sein.

Python unterscheidet sich in vielen Punkten zu anderen Sprachen, unteranderem dass sie viel Wert auf Lesbarkeit gibt. Die auffälliste davon ist, dass Einrückungen Codeblöcke unterteilt anstatt eine Art von Klammern. Dafür gibt es zwei Gründe:

- Es macht den Code kürzer und er wirkt nicht unnötig ausgeschmückt, daher braucht man eine kürzere Aufmerksamkeitsspanne um den Sinn einer Codestelle nachvollziehen zu können.
- Die Stuktur des Codes ist vereinhaltlicht, was es einfacher macht Projekte von anderen zu verstehen.

Außerdem werden dem Entwickler in vielen Entscheidungen leichter gemacht, da unnötige Möglichkeiten entfernt wurden, das heißt, dass es meisten nur eine offensichtliche Art und Weise gibt, etwas zu implementieren. Dazu kommt noch die Nutzung von Spezialzeichen, es werden nur Zeichen unterstützt, die den meisten bereits bekannt sind und dessen Operation Sinn machen. [27]

Dies beantwortet jedoch nicht die Frage ”Wieso ist Python die beliebteste Sprache für ML?”. Die Antwort darauf ist, dass die Sprache nicht nur triviale Aufgaben bereits vorimplementiert, sonder auch, dass die meisten ML Funktionen in Python Libraries zusammengefasst sind. Daher muss man als Neuanfänger oder Fortgeschrittener keine bereits gelösten Probleme von Grund auf noch einmal angehen.

4.2.3 Notebooks

Da es bei ML es öfters dazu kommt, dass bestimmte Codeblöcke oft wiederholt ausgeführt werden, kann man mit Hilfe von Notebooks diese einzelen ausführen. Zum Beispiel beim Analysieren eines Datensatzes oder schnell kleine Änderung am geplanten Vorgehen vorzunehemen.

Diese Codeblöcke können entweder Code, Texte oder Grafiken beinhalten, diese werden fortlaufend mit einer Nummer versehen und die Ausgabe/Grafik erscheint direkt unter dem Code.

```
[1]: from matplotlib import pyplot as plt
     import numpy as np

     # Generate 100 random data points along 3 dimensions
     x, y, scale = np.random.randn(3, 100)
     fig, ax = plt.subplots()

     # Map each onto a scatterplot we'll create with Matplotlib
     ax.scatter(x=x, y=y, c=scale, s=np.abs(scale)*500)
     ax.set(title="Some random data, created with JupyterLab!")
     plt.show()
```

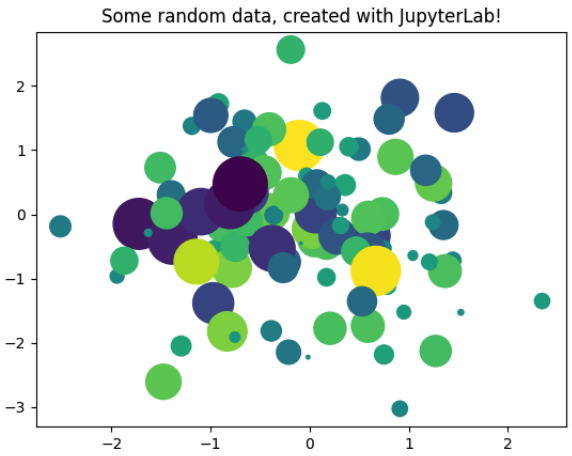


Abbildung 24: Beispiel für ein Notebook mit Code und Ausgabe mit Jupyter Notebook

Diese Notebooks kann man entweder lokal starten oder auf diversen Webseiten online benutzten, dann wird auch der Code auf den Servern dieser Notebookseiten ausgeführt. Dies kann von großem Vorteil sein, da oft generierte Grafiken sehr viel Leistung in Spruch nehmen. Außerdem werden die Ergebnisse/Ausgaben gespeichter und man muss nicht bei jedem Neuladen jeden Codeblock neu ausführen.

4.2.4 Libraries

Pandas Bevor der eigentliche ML-Prozess beginnen kann, müssen die Daten als Erstes analysiert werden und dafür ist die Library ”pandas” geeignet . Neben der Analyse ist pandas auch in der Datenmanipulation sehr hilfreich, da die vordefinierten Funktionen nicht nur leicht und verständlich sind, sondern auch für echte Daten aus der Welt gedacht.

Zu den zwei Hauptdatentypen von pandas gehörten Dataframes und Series.

Series sind eindimensionale Arrays, welche aus verschiedenen Datentypen bestehen können. Mithilfe von einer ID kann man auf jeden Eintrag in der Series zugreifen, diese sind entweder mitgegeben worden oder von pandas generiert (aufsteigende Zahl beginnend bei 0).

Mit der Methode `pandas.Series(data)` kann man diese Datentypen in Series umwandeln:

- Dictionary
- ndarray
- Skalarwert

Dataframes sind zweidimensionale Tabllen mit verschiedenen Datentypen als Spalten. Sie sind vergleichbar mit Datenbanktabellen oder mit einem Dictionary mit dem Datentyp Series als Value.

Jede Zeile beinhaltet eine ID, falls diese nicht vergeben ist, wird von pandas eine generiert und mittels dieser ID kann man dann auf bestimmte Zeilen zugreifen. Das gleiche Prinzip gilt bei den Spalten, wo man mit der Spaltenbezeichnung auf jeden Wert der Zeilen assoziiert mit dieser Spalte zugreifen kann.

pandas bietet beim Erstellen von Dataframes mehrere Möglichkeiten an:

- Einlesen einer Datei (*.json, *.html, *.sql, *.xlsx, ...)

  Am Beispiel einer .csv-Datei: `pandas.read_csv(FILENAME)`

- Dictionaries

  `pandas.DataFrame.from_dict(DICT_VARIABLE)`

- Series

  `pandas.Series.to_frame(SERIES_VARIABLE)`

- Anderes Dataframe

Das Benutzen von pandas ersparrt sehr viel Zeit und ermöglicht den schnellsten Weg zu einem Ergebniss mit diesen Funkionalitäten:

- Löschen und Ersetzen von fehlenden Daten, welches in
- HInzufügen und Löschen von Spalten in Dataframes

- Gruppierung von Daten
- Koventierung in NumPy oder andere Python Objekte
- ...

Seaborn ist eine auf Matplotlib basierende Visualisierungslibrary.

Auch wenn das Analysieren der Daten eine grobe Einschätzung ermöglicht, ist es für das menschliche Auge besser diese Daten in Grafiken und Diagrammen visualisiert zu sehen. Auch bei größeren Datensätzen wird mit der Visualisierung ermöglicht, dass Zusammenhänge, Muster oder Ausreißer besser erkannt werden.

Bei der Auswertung ist es wichtig die Datenverbreitung zu beachten, da die Auswahl von Daten ein Diagramm stark beeinflussen kann.

Da jede Art von Diagrammen Vor- und Nachteile hat, ist es wichtig die Daten in verschiedenen Diagrammtypen darzustellen. Eine Auswahl von möglichen Plots ist hier aufgelistet:

- Scatter Plot
- Box Plot, ermöglicht das Ablesen von Ausreißern
- Violin Plot
- Swarm Plot
- Histogram
- Bar Plot
- Dist Plots

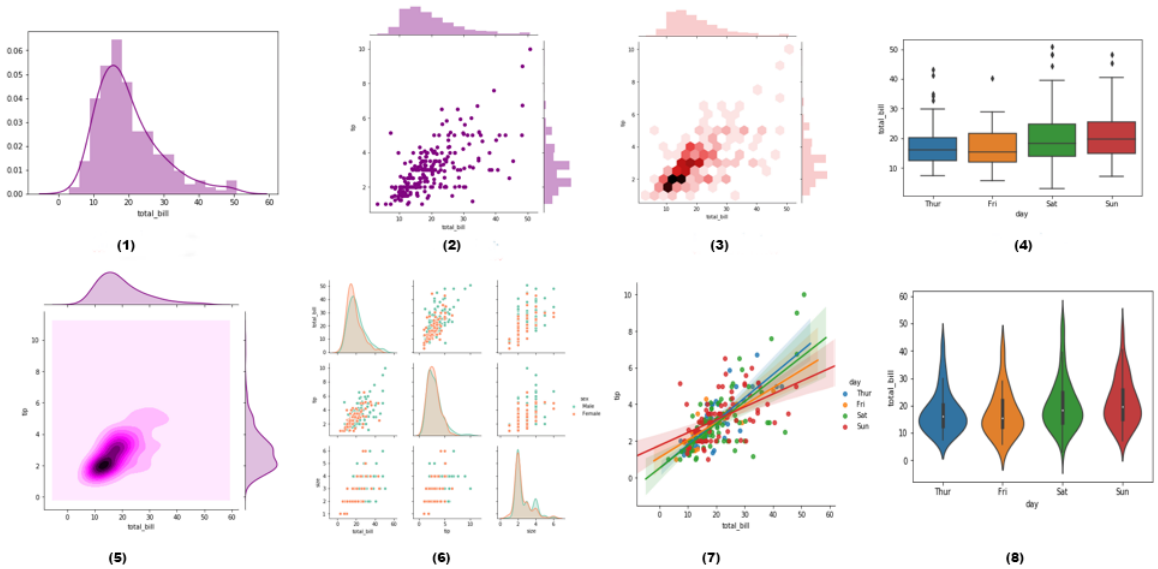


Abbildung 25: Beispiele für (1) Dist Plots, (2) Hex Plots, (3) Box Plots, (4) KDE Plots, (5) Pair Plots und (6) Violin Plots

5 Gegenüberstellung und Conclusio

5.1 Konfigurations- und Entwicklungszeit

5.2 Performance und Präzision

5.3 Benutzerfreundlichkeit

5.4 Kosten

Glossar

AI Artificial Intelligence

CLI Command Line Interface

CSS Cascading Style Sheets

DL Deep Learning

HTML Hypertext Markup Language

ICDAR International Conference of Document Analysis and Recognition

KI Küntliche Intelligenz

ML Machine Learning

MSER Maximally Stable Extremal Regions

NLP Natura Language Processing

OCR Optical Character Recognition

SCSS Sassy Cascading Style Sheets

SWT Stroke Width Transformer

XHTML Extensible Hypertext Markup Language

XML Extensible Markup Language

Literaturverzeichnis

[1] W. P. u. A. S. u. J. S. u. M. T. A. T. Peter Stone und Rodney Brooks und Erik Brynjolfsson und Ryan Calo und Oren Etzioni und Greg Hager und Julia Hirschberg und Shivaram Kalyanakrishnan und Ece Kamar und Sarit Kraus und Kevin Leyton-Brown und David Parkes, „Artificial Intelligence and Life in 2030." 2016. Online verfügbar: https://ai100.stanford.edu/sites/g/files/sbiybj18871/files/media/file/ai100report10032016fnl_singles.pdf

[2] N. J. Nilsson, *The Quest for Artificial Intelligence*. Cambridge, UK: Cambridge University Press, 2009.

[3] B. B. und Daniel Klemm und Dr. Carlo Velten, „Machine Learning im Unternehmenseinsatz - Künstliche Intelligenz als Grundlage digitaler Transformationsprozesse," 2017. Online verfügbar: https://cloudflight.io/app/uploads/2022/02/studie_machine-learning-im-unternehmenseinsatz.pdf

[4] D. E. Ullmann, „Lernen aus neurobiologischer Perspektive," 2015. Online verfügbar: https://www.vhs-st.de/fileadmin/user_upload/Eigene_Daten/Projekte/PUYB/OEsterreich/20161006_WS_04_Neurobiologie.pdf

[5] „Entwicklung von Gehirn und Nervensystem," letzter Zugriff am 12.09.2022. Online verfügbar: https://www.neurologen-und-psychiater-im-netz.org/gehirn-nervensystem/entwicklung

[6] J. Delua, „Supervised vs. Unsupervised Learning: What's the Difference? | IBM," 2021, letzter Zugriff am 03.27.2022. Online verfügbar: https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning

[7] A. Biswal, „The Complete Guide on Overfitting and Underfitting in Machine Learning," 2021, letzter Zugriff am 13.09.2022. Online verfügbar: https://www.simplilearn.com/tutorials/machine-learning-tutorial/overfitting-and-underfitting

[8] C. N. N. und Oliver Zeigermann, *Machine Learning kurz und gut*, 1. Aufl. Heidelberg: dpunkt.verlag, 2018.

[9] L. Wuttke, „Training-, Validierung- und Testdatensatz," 2021, letzter Zugriff am 01.04.2022. Online verfügbar: https://datasolut.com/wiki/trainingsdaten-und-testdaten-machine-learning/

[10] J. JORDAN, „Evaluating a machine learning model." 2017, letzter Zugriff am 13.09.2022. Online verfügbar: https://www.jeremyjordan.me/evaluating-a-machine-learning-model/

[11] L. Eikvil, „OCR - Optical Character Recognition," 1993, letzter Zugriff am 03.27.2022. Online verfügbar: http://home.nr.no/~eikvil/OCR.pdf

[12] G. Shperber, „A gentle introduction to OCR." 2018, letzter Zugriff am 03.27.2022. Online verfügbar: https://towardsdatascience.com/a-gentle-introduction-to-ocr-ee1469a201aa

[13] B. E. und Eyal Ofek und Yonatan Wexler, *Detecting Text in Natural Scenes with Stroke Width Transform.* Microsoft Corporation, 2010, letzter Zugriff am 01.04.2022. Online verfügbar: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/201020CVPR20TextDetection.pdf

[14] „ICDAR," letzter Zugriff am 03.30.2022. Online verfügbar: https://icdar2021.org

[15] V. Meel, „Artificial Neural Network: Everything you need to know," letzter Zugriff am 12.09.2022. Online verfügbar: https://viso.ai/deep-learning/artificial-neural-network/

[16] L. Wuttke, „Machine Learning vs. Deep Learning: Wo ist der Unterschied?" letzter Zugriff am 12.09.2022. Online verfügbar: https://datasolut.com/machine-learning-vs-deep-learning

[17] „HTML & CSS," letzter Zugriff am 14.11.2022. Online verfügbar: https://www.w3.org/standards/webdesign/htmlcss

[18] „HTML/Tutorials/Entstehung und Entwicklung," letzter Zugriff am 14.11.2022. Online verfügbar: https://wiki.selfhtml.org/wiki/HTML/Tutorials/Entstehung_und_Entwicklung

[19] „HTML Attributes," letzter Zugriff am 17.11.2022. Online verfügbar: https://www.w3schools.com/html/html_attributes.asp

[20] „Übersicht: Attribute der HTML-Tags," 2022, letzter Zugriff am 17.11.2022. Online verfügbar: https://www.mediaevent.de/html/HTML-Attribute.html

[21] „Die 10 wichtigsten HTML-Tags," 2018, letzter Zugriff am 14.11.2022. Online verfügbar: https://code-crowd.de/blog/die-10-wichtigsten-html-tags/

[22] „HTML <!DOCTYPE> Declaration," letzter Zugriff am 14.11.2022. Online verfügbar: https://www.w3schools.com/tags/tag_doctype.asp

[23] „CSS/Tutorials/Einstieg/Stylesheets einbinden," letzter Zugriff am 22.11.2022. Online verfügbar: https://wiki.selfhtml.org/wiki/CSS/Tutorials/Einstieg/Stylesheets_einbinden#Einbindung

[24] „Sass Basics," letzter Zugriff am 22.11.2022. Online verfügbar: https://sass-lang.com/guide

[25] „2021 Developer Survey," 2021, letzter Zugriff am 13.09.2022. Online verfügbar: https://insights.stackoverflow.com/survey/2021#technology-most-popular-technologies

[26] „Displaying values with interpolation," 2022, letzter Zugriff am 13.11.2022. Online verfügbar: https://angular.io/guide/interpolation

[27] G. van Rossum, „Foreword for "Programming Python"(1st ed.)," 1996, letzter Zugriff am 01.04.2022. Online verfügbar: https://legacy.python.org/doc/essays/foreword

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

Anhang

Hi