

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

1.1 Theorie

1.1.1 KI, ML, DL - Unterschiede

Im Zusammenhang mit maschinellem Lernen werden oft die Begrifflichkeiten "Künstliche Intelligenz", "Machine Learning" und "Deep Learning" verwechselt, jedoch handelt es sich bei diesen um eigene Bereiche, die sich in vielen Punkten überschneiden.

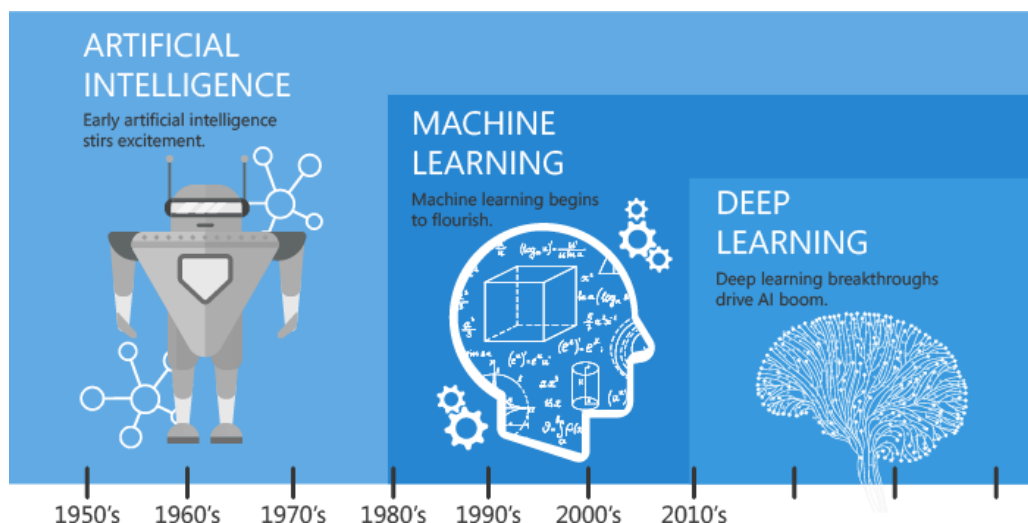


Abbildung 1: Evolution von Künstlicher Intelligenz

KI - Künstliche Intelligenz

Bereits in den 1950er erschien der Begriff "Künstliche Intelligenz", auf englisch "Artificial Intelligence", im Bereich Informatik, um Aufgaben zu lösen. Anfänglich noch mit dem Brettspiel "Dame" und einfachen Logikaufgaben. Jedoch kann es sich bei einer KI nur um eine programmierte Regel handeln, da man nur erwartet, dass sich die KI in gewissen Situationen auf eine bestimmte Art und Weise reagiert. Daher beschreibt es Programme, die humane Funktionen, imitieren und gibt nicht an mit welcher Technik das Problem gelöst wird.

ML - Machine Learning

Der Unterschied zur Künstlichen Intelligenz liegt beim Vorgang des Lernens. Genau wie ein menschliches Gehirn muss ein Machine Learning Model mit Daten trainiert werden, mit welchen das Model dann gewisse Klassifizierungen, Clusterbildungen oder Regressionen durchführen kann. Über die Zeit verbessert sich die Präzision, da diese mit Zuwachs der eingespielten Daten wächst.

DL - Deep Learning

Sowie beim Machine Learning sind Deep Learning Algorithmen abhängig von antrainierten Daten und kann daher als Synonym oder Untergruppe vom Machine Learning gesehen werden. Jedoch kann ein DL Model wie ein Mensch noch nie davor gesehene spezielle Bilder kategorisieren und ist damit einem ML-Modell in vielen Hinsichten überlegen.

Ohne Deep Learning würden die meisten modernen Assistenten nicht auf dem Niveau arbeiten wie erwartet, dabei handelt es sich bei Deep Learning um eine junge Technologie, die auf Neuronale Netzwerke basiert.

ML vs DL

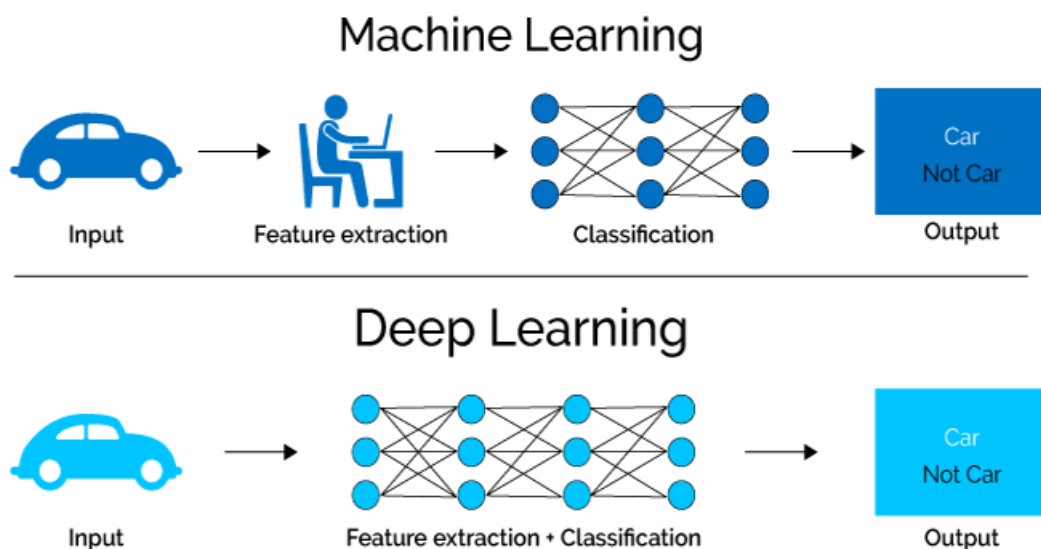


Abbildung 2: Feature Extraction, ML vs DL

Feature Extraction Bevor ein ML-Modell mit der eigentlichen Verarbeitung beginnen kann, müssen die Rohdaten einer sogenannten "Feature Extraction" unterlaufen,

welche die gegebenen Daten abstrakt darstellt. 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 gibt es Neuronale Netzwerke, welche den Feature Extraction Schritt übernehmen und selbstständig Rohdaten verarbeitet. Über mehrere Schichten werden bestimmte Merkmale hierarchisch definiert und später zum Beispiel zur Kategorisierung genutzt, hierbei erhöht sich die Genauigkeit der Extrahierung schon während des Antrainierens.

Welches Verfahren ist sinnvoller? In Bereichen, wo bereits strukturierte Daten vorhanden sind, bietet ein ML-Modell an, ein gutes Beispiel dafür ist die Erstellung von Vorhersehungen von Ereignissen. Außerdem benötigt man keine große Menge an Daten, um genaue Ergebnisse zu bekommen.

Stehen keine bereits vorpreparierte Daten zur Verfügung, wäre ein DL-Modell die bessere Lösung, aber dies auch nur wenn man eine große Menge an Daten hat.

1.1.2 Arten von Machine Learning

Der wichtigste und komplizierteste Teil beim maschinellen Lernen, ist die Kunst einem Computer das selbständige Lernen beizubringen, dabei orientiert man sich am Lernprozess eines Menschen. Dazu besteht es aus einer 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

Ein Neugeborenes kommt mit circa 100 Milliarden Neuronen auf die Welt, diese sind jedoch nur schwach miteinander verknüpft. Mithilfe des Lernens werden diese Verbindungen gestärkt und das Kind kann Vorgehensweisen besser verstehen und neue Erkenntnisse gewinnen.

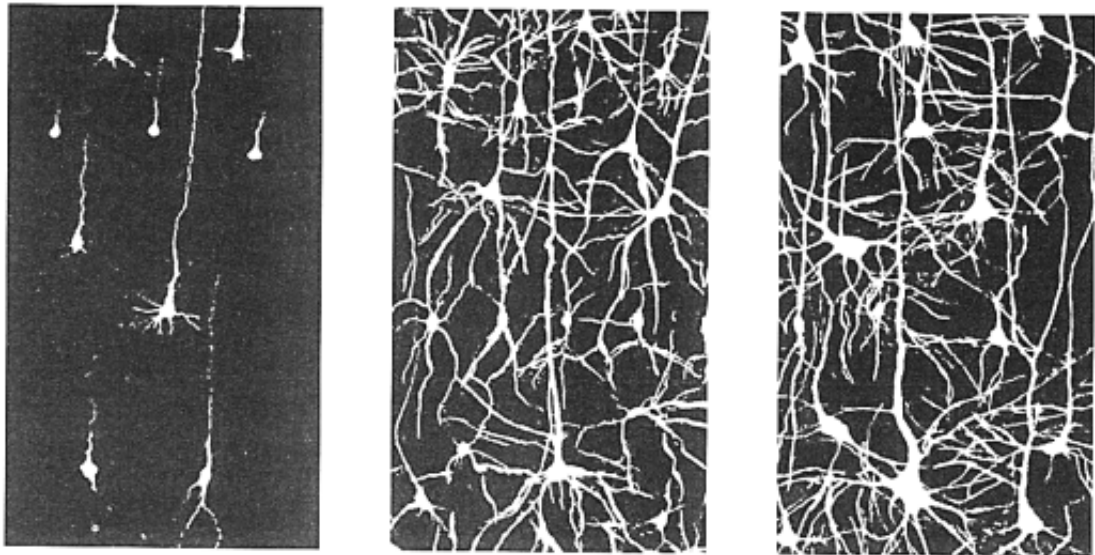


Abbildung 3: Vernetzungen nach der Geburt, nach 3 Monaten und nach 15 Monaten

Die gegenwärtige Neurobiologie, definiert das Lernen als vielschichtigen Prozess, welcher Neuronenverbänder plastisch miteinander verbindet und so bilden sich Netzwerke und System aus Neuronen. Umso mehr Verbindungen entstehen und mit dem Wiederholen des Gelernten gestärkt werden, desto besser kann das Erlernte aufgerufen werden und mit dem bereits existierenden Vorwissen kombiniert werden.

Mithilfe einem zwecksentfermdeten, morphologischem Kasten, auch Zwicky-Box genannt, kann man das menschliche Gehirn vereinfacht darstellen. Dabei gibt die erste Spalte ausgewählte Merkmale an und die darauf folgenden Spalten eine weitere Ausprägung vom Merkmal in der selben Zeile. Die große Anzahl der Ausprägungen ermöglicht später breite Auswahl an Ausgängen der Zwicky-Box.

Merkmal	Ausprägungen		
Augenfarbe	Blau	Grün	Braun
Haarfarbe	Blond	Brünett	Schwarz
Frisur	Lang	Glaze	Kurz
Gesichtsform	Schmal	Rund	Oval
Nase	Klein	Groß	Flach
Brille	Ja	Nein	

Tabelle 1: Beispiel einer Zwicky-Box

In dieser Zwicky-Box wurden Merkmale eines Menschen aufgezählt. Um einen Ausgang darzustellen wird für jedes Merkmal eine Ausprägung ausgewählt und mit einer Linie

verbunden, damit man eine Ausprägung von anderen unterscheiden kann. In diesem Beispiel wird die Linie durch verschiedenen Farben ersetzt.

Merkmal	Ausprägungen		
Augenfarbe	Blau	Grün	Braun
Haarfarbe	Blond	Brünett	Schwarz
Frisur	Lang	Glaze	Kurz
Gesichtsform	Schmal	Oval	Rund
Brille	Ja	Nein	

Tabelle 2: Beispiel einer Ausprägung

Jede eingefärbte Zelle kann man als "aufleuchtendes" Neuron interpretieren, wenn man versucht das Geschlecht einer fremden Person zu identifizieren. Jedoch lernt man durch dieses "Aufleuchten" nicht, da man das eigentliche Geschlecht der Person nicht kennt und man daher keine Verbindungen aufbauen kann. Durch das Kennenlernen von Personen und Merken derer Ausprägungen werden diese oben genannten Neuronenverbindungen aufgebaut, welche später genutzt werden, um das Geschlecht einer fremden Person festzustellen. Diese Art von Lernen ist ein sehr schneller und effektiver Prozess, der mit vielen Nachteilen kommt, denn genau auf diesem Weg werden Vorurteile gebildet. Anhand eines Beispiels: Die meisten würden die obigen Werte aus der Zwicky-Box als Frau wahrnehmen, jedoch könnte jede kleinste Änderung am Aussehen diese Wahrnehmung beeinflussen.

Merkmal	Ausprägungen		
Augenfarbe	Blau	Grün	Braun
Haarfarbe	Blond	Brünett	Schwarz
Frisur	Lang	Glaze	Kurz
Gesichtsform	Schmal	Oval	Rund
Brille	Ja	Nein	
Ergebnis:	Frau		
Sicherheit:	70%		

Tabelle 3: Ergebnis einer Ausprägung

Da das Aussehen im Beispiel dem Aussehen von Bekannten in gewissen Merkmalen ähnelt, kann man mit einer Wahrscheinlichkeit von 70% sagen, dass es sich um eine Frau handelt. Desto mehr Personen man kennt, umso besser und genauer werden die Ausprägungen dem Geschlecht zugeteilt und die Sicherheit wird erhöht.

In dem Fall, dass ein Merkmal nicht erkennbar ist, wird versucht das beste Ergebnis zu finden, jedoch wird die Sicherheit drastisch fallen.

Supervised Learning

Hierbei werden gelabelte (daher auch "supervised" oder im deutsche überwacht) Datensätze genutzt, um Daten zu klassifizieren oder Vorhersagen genau wie möglich aufzustellen. Diese Art von Lernen kann man in zwei Typen aufteilen:

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.

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

Entscheidungsbäume, oder im Englischem Decision Trees, sind Visualisierungen von Entscheidungswegen, oder auch aufgefächerte Zwicky-Boxen, die die Verbindungen visualisieren.

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. Jedoch kann es auch passieren, dass durch Zufälle unberechenbare Verbindungen gefunden werden, dieses Problem wird "Overfitting".

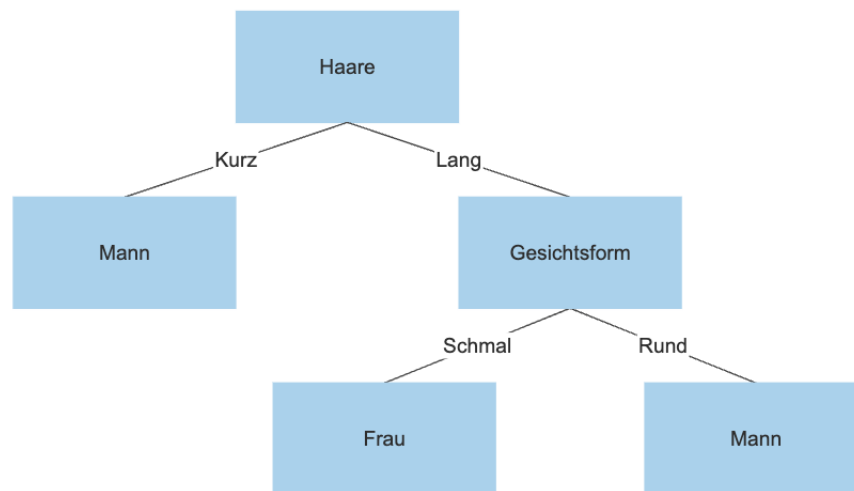


Abbildung 4: Decision Tree an dem Beispiel von ??

Overfitting bedeutet, dass ein Modell perfekt an die Testdaten angepasst ist, jedoch bei unbekannten Daten falsche Verbindungen aufbauen könnte und dadurch falsche Ergebnisse vorhergesagt werden.

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

Unsupervised Learning

Beim Unsupervised Learning, oder Unüberwachtes Lernen, erzeugt Verbindungen ohne genauere Informationen über den Testdatensatz. Dabei muss das Programm selbst Gruppen definieren und dann die übergebenen Daten in diese Gruppen zuordnen.

Clustering ist einer der beliebtesten Variante, 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.

Auch für das menschliche Auge ist es möglich dieses Beispiel ?? 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, dieser Cluster einer Kategorie zuzuteilen.

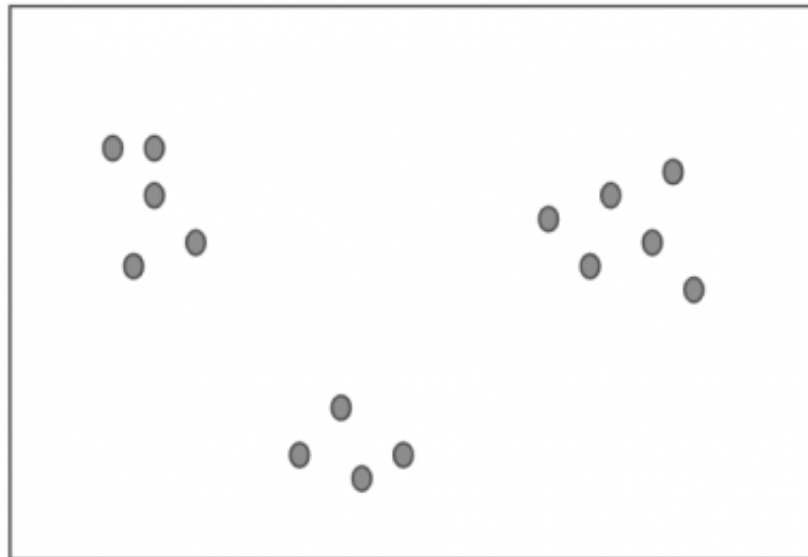


Abbildung 5: Unkategorisierte Daten

Diese Vorgehensweise wird oft in sehr komplizierte Einsatzbereiche genutzt, und daher ist es oft 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.

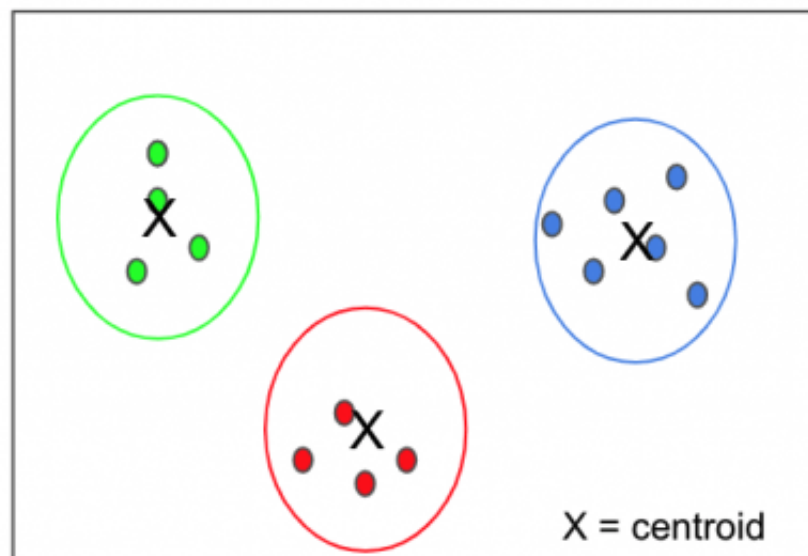


Abbildung 6: Kategorisierte Daten

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. Die nennt man auch negative Verstärkung.

Dahingegen führen richtige Ergebnisse oder erwarteten 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".

1.1.3 Optical Character Recognition

In den meisten Fällen erfolgt eine Eingabe für eine Tastatur, jedoch ist dies manchmal weder die schnellste noch effizienteste Art, Text einzulesen. Mithilfe von Optical Character Recognition ist ein automatisiertes Einlesen und Verarbeiten 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 [?].

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

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

- 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.

- 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.

Lateinisches Alphabet	Kyrillisches Alphabet	Ähnelt dem Buchstaben
r	p	p
B	B	B
I	H	H
U	y	y
S	C	C

Tabelle 4: Ähnlichkeiten zwischen Buchstaben im Lateinischem und Kyrillischem Alphabet

- Platzierung

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

Strategien

Die Texterkennung ist generell in zwei Phasen aufgeteilt:

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 Paragraph) ein, dabei werden die genauen Koordinaten der Eckpunkte mitzurückgegeben. Diese werden später genutzt, um zur Visualisierung Boxen auf dem Bild oder der PDF aufzuzeichnen. Dieser Prozess wird auch in der Objekterkennung genutzt.

Qty	Rate	Current
40.00	16.50	660.00
		Current
		0.00
		-84.00

Abbildung 7: Bouning Boxes Beispiel

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

Bei der **Regions basierden** Methode, werden Pixel verbunden und als Zeichenkandiat makiert, welche später mehrmals gruppiert wernern und schlussendlich Wörter oder Textzeilen bilden. Dabei kommt es auf die geometrischen Eigenschaften, dabei kann es jedoch zu Segmentierungs Fehlern kommen. Wie man bei ?? 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 Zusammenliegenden, gleich breite Pixel werden zu einem Zeichenkanidaten gruppiert. Danach werden alle benachbarten Zeichenkanidaten untersucht und zu einem Wort gruppiert, falls das mittelwertige Strichbreite Verhältnis nicht über 2 liegt. Außerdem wird die Höhe und Farbe des Zeichenkanidaten berücksichtigt. Dies kann auch der Grund sein, wieso bei ?? das Minuszeichen bei "-84.00" nicht zur Zahl hinzugefügt wird, das Minuszeichen ist signifikant niedriger als der Rest der Zahl. [?]

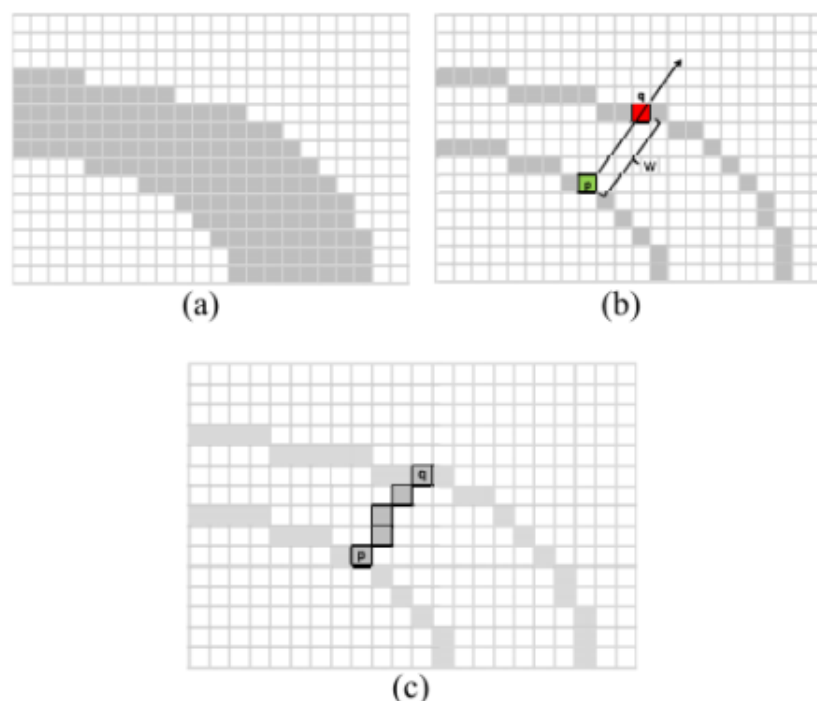


Abbildung 8: Die Kanten des Striches (a) werden solange verglichen, bis zwei gefunden werden mit der mit der gleichen Richtung (b). Alle unterliegenden Pixel erhalten die Strichbreite der Entfernung zwischen der Start- und Endkante (c).

Textur basierende Methoden unterteilen das Bild in Fester, 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, als erstes werden Blöcke extrahiert mit dem Maximally Stable Extremal Regions (MSER) Verfahren, danach werden benachbarte Blöcke gruppiert, falls die Farbe und Größe sich ähnelt. Jedoch werden mit diesem Verfahren ebenfalls eine große Menge an false-positive Blöcken erkannt. Um diese Anzahl zu verringern wird eine ähnlich Idee zur SWT verwendet.

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 [?].

Text recognition ist genau wie bei der Erkennung von Text in zwei Möglichkeiten unterteilt: Regions 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 einem Gaußschen Filter bearbeitet werden und auf ein 5×5 Bild komprimiert wurden. Mit diesen $5 \times 5 \times 8 = 200$ dimensionalen Vektoren werden, dann die bereits erkannten Blöcke klassifiziert.

1.2 Praxis: Python

1.2.1 Python als Programmiersprache

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 Strukturen und Konventionen von Python auf Unix, ohne an Unix gebunden zu sein.

Python unterscheidet sich in vielen Punkten zu anderen Sprachen, unter anderem dass sie viel Wert auf Lesbarkeit gibt. Die auffälligste 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 Struktur des Codes ist vereinheitlicht, 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 meistens 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. [?]

Dies beantwortet jedoch nicht die Frage "Wieso ist Python die beliebteste Sprache für Machine Learning (ML)?" Die Antwort darauf ist, dass die Sprache nicht nur triviale Aufgaben bereits vorimplementiert, sondern 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.

1.2.2 Notebooks

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

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.

2 Cloud Computing

2.1 Theorie

2.1.1 Was ist Cloud Computing?

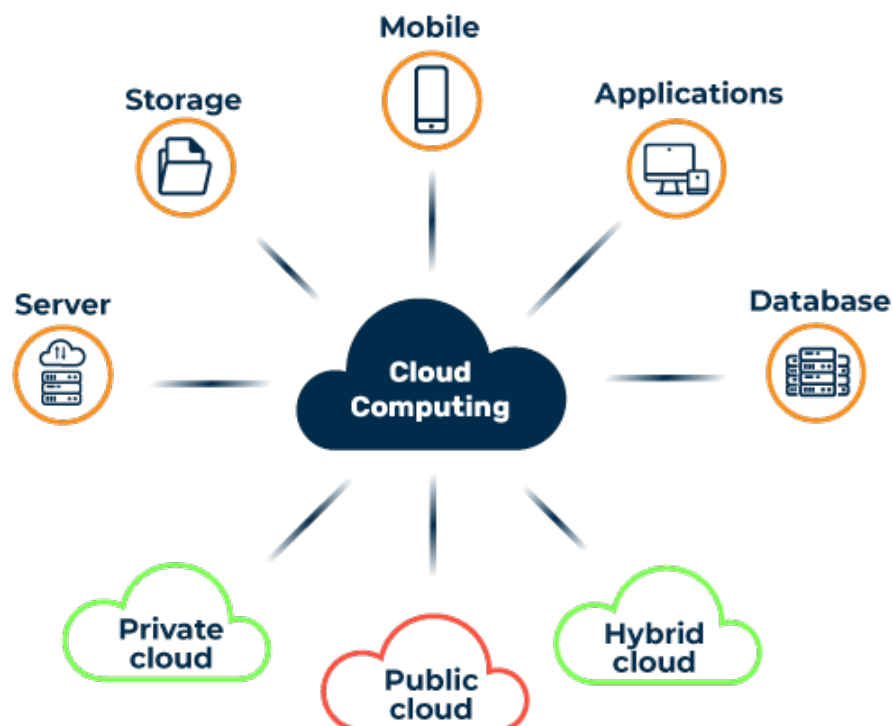


Abbildung 9: CC-Overview

Cloud Computing ist ein Bereitstellungsmodell, bei dem Datenspeicher, Server, Anwendungen usw. 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 ersetzt. 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 einen Absturz des Webshops zu vermeiden.

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 erreicht werden kann. Deshalb wird die Fähigkeit der Cloud die 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

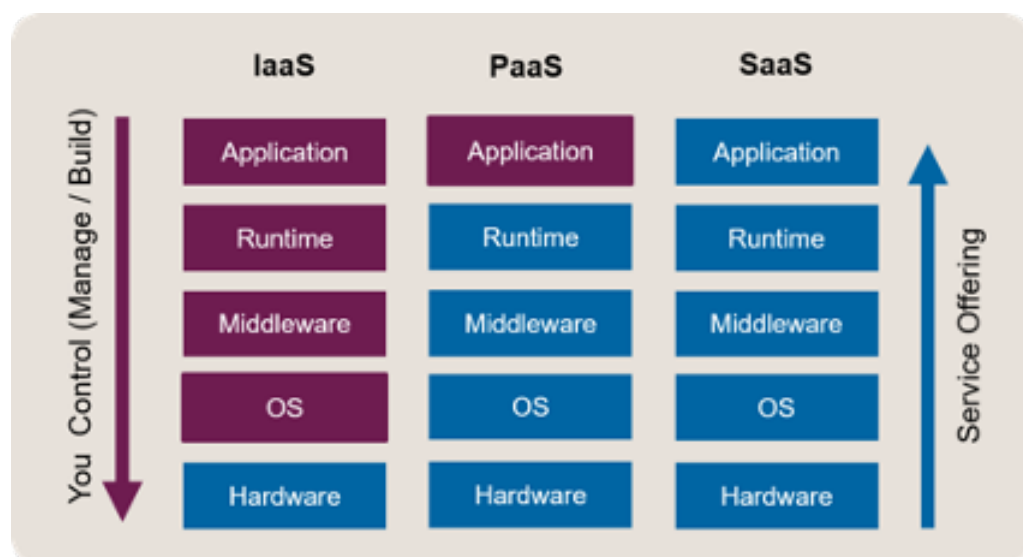


Abbildung 10: Modelle

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, dass 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 bereitgestellte Anwendung zuzugreifen 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

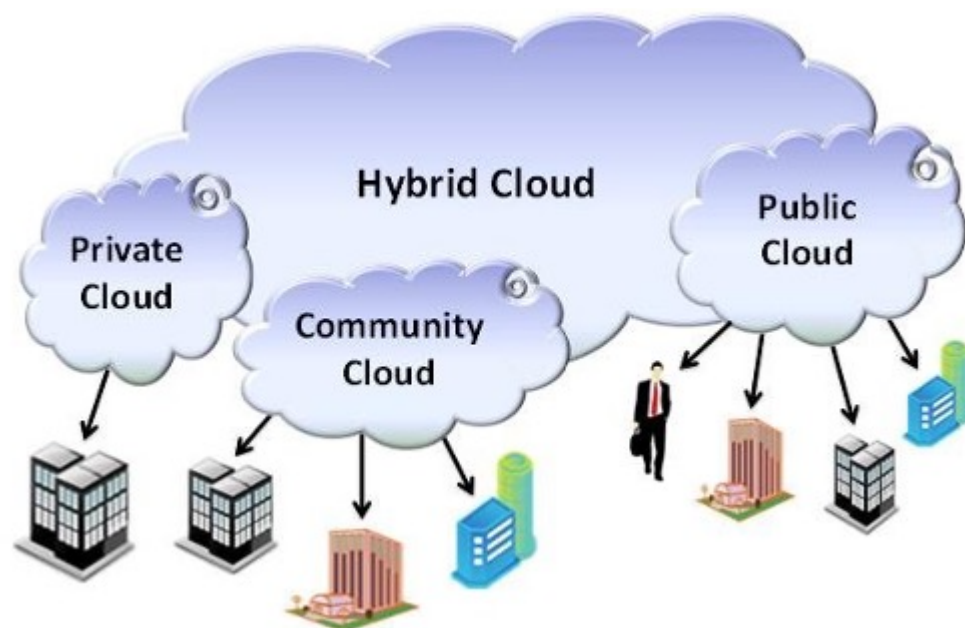


Abbildung 11: Anwendungsfälle

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

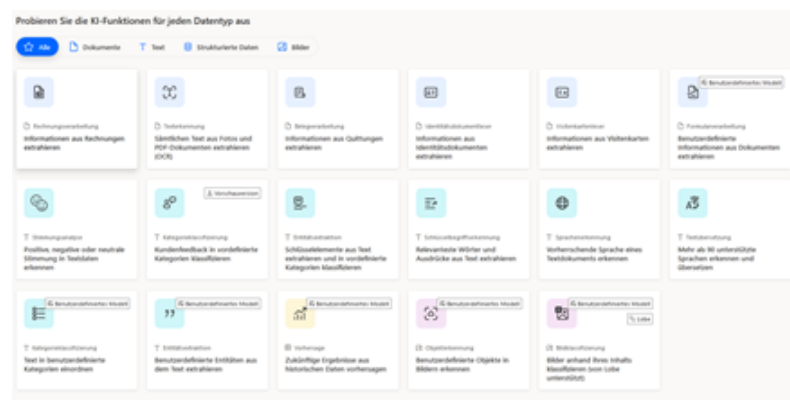


Abbildung 12: AI-BUILDER Modelle

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 ??) 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

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

Datum: Oct 21, 2020
Fälligkeitsdatum: Nov 11, 2020

Saldo fällig: 1.270,00 €

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 13: 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:

Zu extrahierende Informationen auswählen

Wählen Sie alle Informationen aus, die vom AI-Modell aus den Dokumenten extrahiert werden sollen. Beispiel: Name, Adresse, Gesamtbetrag. Sie markieren diese in den Dokumenten.

Name	Typ
<input checked="" type="checkbox"/> Company Name	Text
<input checked="" type="checkbox"/> Company Address	Text
<input checked="" type="checkbox"/> Invoice Id	Text
<input checked="" type="checkbox"/> Invoice Date	Text
<input checked="" type="checkbox"/> Total Value	Text
<input checked="" type="checkbox"/> Items	Mehrseitige Tabelle (Description, Qty, Unit, Unit Price, Total Price)

Abbildung 14: 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 wiederum 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 (??)

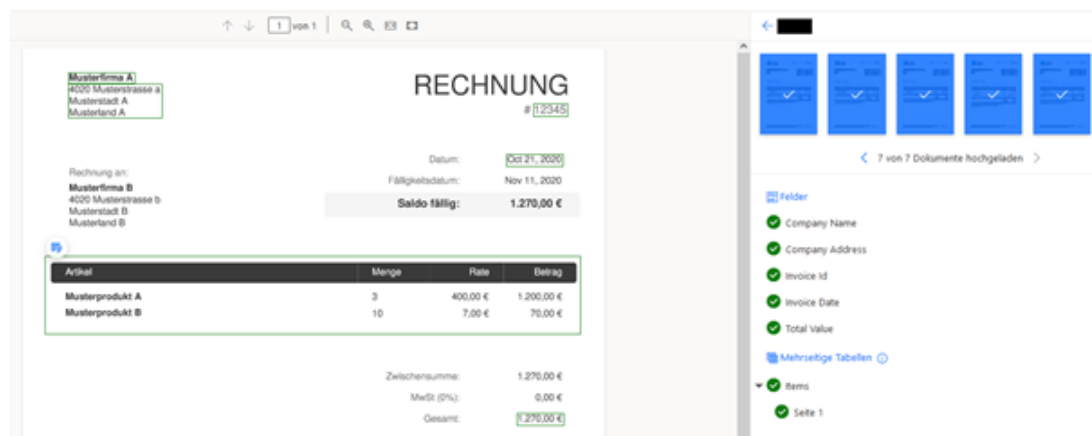


Abbildung 15: AI-Builder Felder Tagging

2.3 Praxis: Power Automate

Power Automate ist ein Teil der Microsoft Power-Plattform, 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 ??).

The screenshot shows the configuration window for the 'Bei Eingang einer neuen E-Mail (V3)' trigger in Power Automate. The window has a blue header bar with the title and a question mark icon. Below the header, there are several configuration fields:

Ordner	Inbox
An	E-Mail-Empfängeradressen, getrennt durch Semikolons (bei Übereinstimmung)
Cc	E-Mail-Empfängeradressen in "Cc", getrennt durch Semikolons (bei Übereinstimmung)
"An" oder "Cc"	E-Mail-Empfängeradressen in "An" oder "Cc", getrennt durch Semikolons (bei Übereinstimmung)
Von	E-Mail-Absenderadressen, getrennt durch Semikolons (bei Übereinstimmung)
Anlagen einschließen	Ja
Betrefffilter	invoice
Wichtigkeit	Beliebig
Nur mit Anlagen	Ja

At the bottom left, there is a link 'Erweiterte Optionen ausblenden' with an upward arrow icon.

Abbildung 16: 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 verarbeitet werden. Das zuvor definierte AI-Builder Modell extrahiert die notwendigen Information aus dem Dokument (Siehe Abbildung ??).

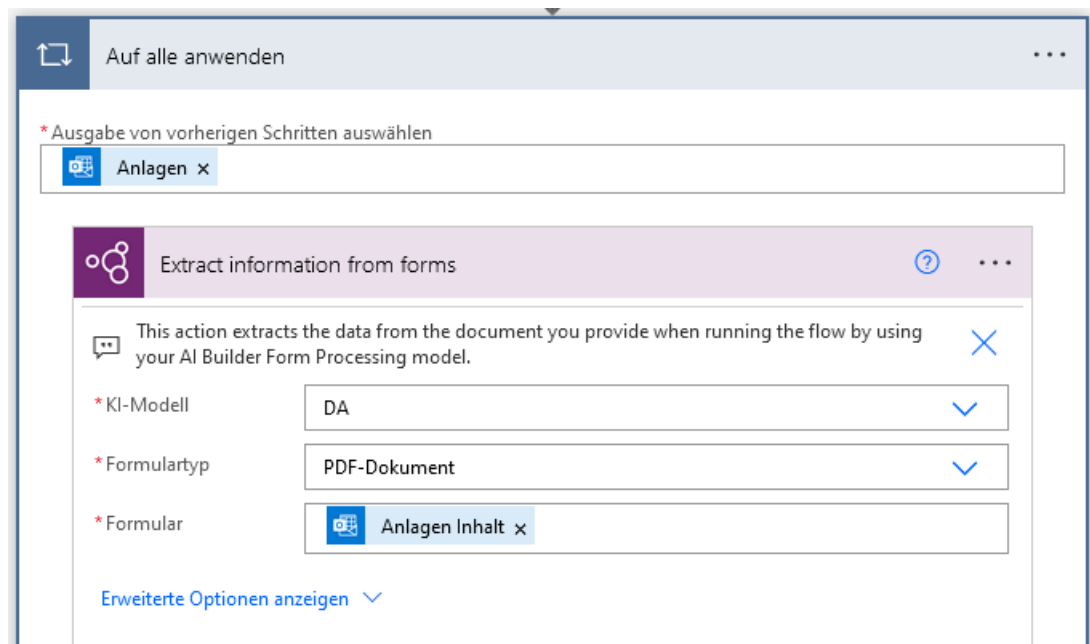


Abbildung 17: Verarbeitung des Dokuments

Um die extrahierten Daten zu persistieren, wurde eine Dynamics CRM Testumgebung aufgesetzt. Es wurden zwei Entitäten, Invoice und Invoice Items erstellt (??). Da die Entität Invoice Items von der Entität Invoice abhängig ist, muss die die Entität Invoice Items eigens gespeichert werden (Siehe Abbildung ...).

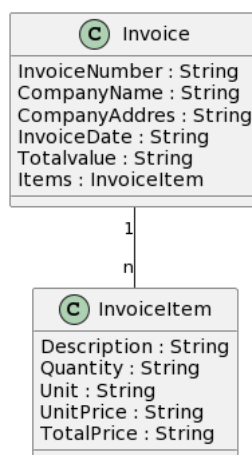


Abbildung 18: Klassendiagramm Invoice

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 ??.

??

Neuen Datensatz erstellen (veraltet)

Auf alle anwenden 2

*Ausgabe von vorherigen Schritten auswählen

Items entries ×

Neuen Datensatz erstellen (veraltet) 2

* Organisationsname CRM782075

* Entitätsname Invoice Items

* Description Items Descrip... ×

* Invoice Invoice ×

* Name Bezeichnung ×

* Quantity Items Qty value ×

* Total Price Items Total Pri... ×

* Unit Items Unit val... ×

* Unit Price Items Unit Pri... ×

Invoice

Erweiterte Optionen anzeigen

Abbildung 19: Persistierung der Entitäten

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:

1. Stabile und interaktive Anwendungen setzen voraus, dass Programme untereinander barrierefrei kommunizieren können. Eine API, Application Programming

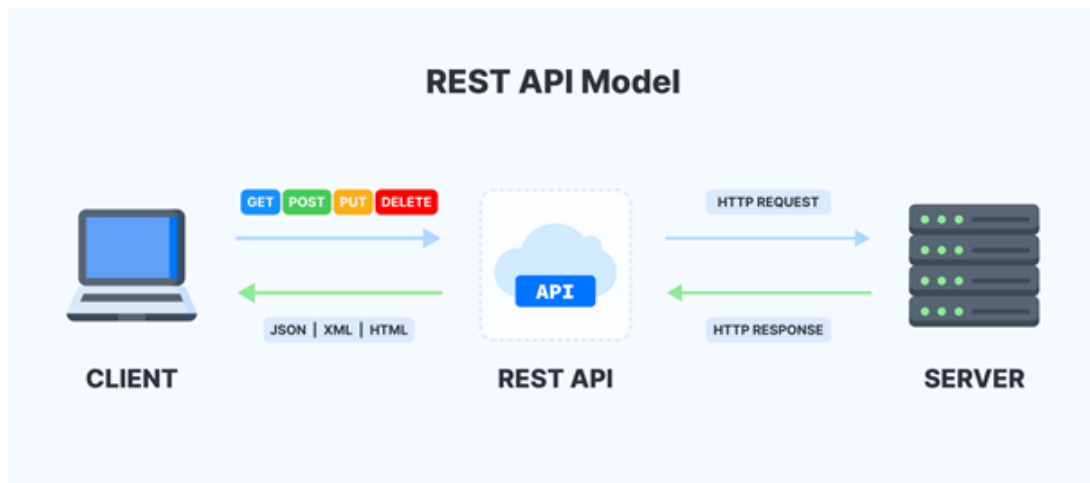


Abbildung 20: Funktionsweise von REST

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 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 Hardwareplattform 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 Konfigurations- und Entwicklungszeit

4.2 Performance und Präzision

4.3 Benutzerfreundlichkeit

4.4 Kosten

5 Gegenüberstellung und Conclusio

5.1 Konfigurations- und Entwicklungszeit

5.2 Performance und Präzision

5.3 Benutzerfreundlichkeit

5.4 Kosten

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

Anhang