

**Hochschule Worms**  
**Fachbereich Informatik**

Sommersemester 2021

Thesis zur Erlangung des akademischen Grades Bachelor of Science

**Prof. Dr. Eberhard Kurz**

**Untersuchung zur möglicher Implementierung von künstlicher  
Intelligenz in eine Low-Code Entwicklungsumgebung**

Fabian Mohs  
Ahseufer 5  
59063 Hamm

Matr.-Nr.: 673618

8. Semester

Datum der Abgabe: 31.05.2021

---

## INHALTSVERZEICHNIS

<b>Abkürzungsverzeichnis .....</b>	<b>IV</b>
<b>Abbildungsverzeichnis.....</b>	<b>V</b>
<b>Abstract.....</b>	<b>VII</b>
<b>1 Einleitung .....</b>	<b>1</b>
<b>2 Grundlagen .....</b>	<b>2</b>
2.1 Allgemeine Begriffsklärung .....	2
2.1.1 User Experience .....	2
2.1.2 Low-Code.....	2
2.1.3 Versionskontrolle .....	3
2.2 Grundlagen der Webentwicklung .....	3
2.2.1 Hypertext Markup Language (HTML).....	4
2.2.2 Cascading Style Sheets (CSS).....	4
2.2.3 Syntactically Awesome Stylesheets (Sass) .....	4
2.2.4 JavaScript.....	4
2.2.5 Javascript-Frameworks.....	5
2.2.6 Vue.....	5
2.2.7 Quasar.....	7
2.2.8 Typescript.....	7
2.2.9 Application Programming Interface (API).....	7
2.3 Grundlagen Künstliche Intelligenz .....	7
2.3.1 Machine Learning.....	8
2.3.2 Neuronale Netze .....	11
2.3.3 Deep Learning .....	12
2.3.4 Trainingsdaten .....	14
2.3.5 Transfer Learning .....	15

---

2.3.6	Natural Language Processing .....	15
2.3.7	Cloud und Edge KI.....	16
<b>3</b>	<b>Künstliche Intelligenz heute .....</b>	<b>17</b>
3.1	Aktuelle Nutzung von KI.....	18
3.2	Deutschland als KI Standort .....	19
3.3	Anforderungen an KI-Projekte .....	19
3.4	KI/AI as a Service.....	21
<b>4</b>	<b>Entwicklung von KI Systemen .....</b>	<b>22</b>
4.1	Entwicklungsumgebung.....	23
4.2	Entwicklungsprozess .....	24
<b>5</b>	<b>Künstliche Intelligenz und Low-Code .....</b>	<b>29</b>
5.1	Vectorsoft AG.....	29
5.2	Softwareentwicklung .....	29
5.3	Ausgangssituation und Identifikation der Rahmenbedingungen.....	30
5.4	Mögliche Problemstellungen .....	31
5.5	Prototypische Verwendung eines ML Modells .....	32
5.6	Weitere Szenarien für Implementierungsmöglichkeiten .....	37
5.6.1	Kundenkommunikation.....	37
5.6.2	Projektverwaltung .....	38
5.6.3	Bedienung/Steuerung der Anwendung.....	39
5.7	Unterstützung im Design .....	41
5.8	Entwicklung von Schnittstellen für Anwender .....	41
<b>6</b>	<b>Fazit .....</b>	<b>43</b>
<b>7</b>	<b>Ausblick.....</b>	<b>46</b>
<b>8</b>	<b>Literaturverzeichnis .....</b>	<b>49</b>
<b>9</b>	<b>Anhang .....</b>	<b>54</b>

---

## ABKÜRZUNGSVERZEICHNIS

AI	Artificial Intelligence
API	Application programming interface
CPU	Central Processing Unit
CSS	Cascading Style Sheets
DOM	Document Object Model
ERP	Enterprise Ressource Planning
GPU	Graphics processing unit
GUI	Graphical User Interface
HTML	Hypertext Markup Language
JS	JavaScript
KI	Künstliche Intelligenz
ML	Machine Learning
NLP	Natural Language Processing
NPM	Node Package Manager
ONNX	Open Neural Network Exchange
PPS	Produktionsplanungs- und Steuerungssystem
RAD	Rapid Application Development
RAM	Random-Access-Memory
Sass	Syntactically Awesome Stylesheets
Scss	Sassy CSS
SVG	Scalable Vector Graphics
UI	User Interface
UX	User Experience
VS	Visual Studio
XML	Extensible Markup Language
Y	yeet

---

## ABBILDUNGSVERZEICHNIS

Abbildung 1: Darstellung des Verzweigungsmodells von Git.....	3
Abbildung 3: Gegenüberstellung einer Scss Syntax und einer klassischen CSS Schreibweise. ....	4
Abbildung 4: Nutzungshäufigkeit der meist genutzten Web Frameworks ab Anfang 2020 .....	5
Abbildung 5: Beispielhafter Aufbau einer Vue Komponente.....	6
Abbildung 6: Bereiche Künstliche Intelligenz.....	8
Abbildung 7: Aufbau eines neuronalen Netzes.....	11
Abbildung 8: Vergleich zwischen klassischem M.L. und Deep Learning.....	13
Abbildung 9: Performance Vergleich zwischen "klassischem" M.L. und Deep Learning .....	13
Abbildung 10: Umsatzprognose für KI Unternehmensanwendungen .....	18
Abbildung 11: Preistabelle der Natural Language API von Google.....	21
Abbildung 12: Zeigt abstrahiert den Ablauf bei der Integration.....	25
Abbildung 13: Beispielhafte Rollenverteilung im Entwicklungsprozess .....	27
Abbildung 14: Anfallende Prozesse bei der Entwicklung eines ML Modells .....	28
Abbildung 15: Der Designer - Auszug aus einem Entwicklungsstand, zeigt nicht die finale Version. ....	30
Abbildung 16: Art der Nutzung von KI .....	32
Abbildung 17: Aufbau der Chat Komponente .....	33
Abbildung 18: Import von Tensorflow .....	34
Abbildung 19: Laden des Modells aus dem Internet .....	34
Abbildung 20: Struktur der Metadaten .....	34
Abbildung 21: Formatierung der Eingabe.....	35
Abbildung 22: Eingabe einer Nachricht.....	35
Abbildung 23: Numerische Darstellung der Eingabe .....	35
Abbildung 24: Aufgefülltes Array .....	35
Abbildung 25: Verarbeitung der Eingabe durch das Modell .....	36
Abbildung 26: Blockierung eine "unerlaubten" Nachricht .....	36
Abbildung 27: Einsatz von Sprachassistenten .....	39
Abbildung 28: Erfolgsmessung, ab wann Nutzen aus KI Projekten gezogen wird .....	44

---

Abbildung 29: Prognostiziertes Marktumsatz Weltweit bis 2027.....	47
Abbildung 30: Offene Stellen von deutschen Unternehmen im Bereich KI.....	47

**ABSTRACT**

In dieser Arbeit mit dem Thema „Untersuchung zur möglicher Implementierung von künstlicher Intelligenz in Low-Code Entwicklungsumgebung“ werden zunächst Begriffe der Softwareentwicklung und grundlegende Funktionsweisen von Machine Learning, welches die Basis von künstlicher Intelligenz darstellt, erläutert. Dabei wird Bezug auf den heutigen Einsatz von künstlicher Intelligenz genommen und die verschiedenen Betriebsarten, Anforderungen und Risiken werden dargestellt. Verschiedene Frameworks werden vorgestellt, die in der Entwicklung genutzt werden können, um abschließend zu untersuchen, wie diese Techniken genutzt werden können, um die Funktionalität einer Low-Code Entwicklungsumgebung zu verbessern oder zu erweitern. Das Ergebnis dieser Untersuchung ist, dass es eine ganz unterschiedliche Möglichkeiten gibt, künstliche Intelligenz in eine Low-Code Entwicklungsumgebung zu implementieren, es dabei aber die Schwierigkeit ist die individuell richtige Lösung zu finden.

## **1 EINLEITUNG**

Low-Code spielt eine immer wichtiger werdende Rolle, vor allem im Bezug auf die digitale Transformation von Unternehmen. Das US-amerikanische Analyse Unternehmen Gartner prognostiziert für das Jahr 2024, dass mehr als 65% aller Software mit Low-Code Entwicklungsplattformen entwickelt worden sind.<sup>1</sup>

Da immer mehr Möglichkeiten existieren, künstliche Intelligenz in Softwareprodukte zu integrieren und die Nutzung von Low-Code Entwicklungsumgebungen stark ansteigt, soll diese Arbeit die Frage beantworten, welche Implementierungsmöglichkeiten von künstlicher Intelligenz es im Bezug auf Low-Code Entwicklungsumgebungen gibt.

In dieser Arbeit werden zuerst allgemeine Grundlagen und die der Webentwicklung erläutert. Darauf folgend werden grundlegende Funktionsweisen von künstlicher Intelligenz und die damit verbundenen Begriffe erklärt. Dies sorgt für ein technisches Verständnis, um anschließend auf die unterschiedlichen Methoden bei dem Betrieb von KI-Systemen einzugehen und die Anforderungen an KI-Projekte darzustellen. Darauf folgt die Untersuchung, wie künstliche Intelligenz in einer Low-Code Entwicklungsumgebung genutzt werden kann und ob die Verwendung sinnvoll ist. Im Zuge dieser Untersuchung werden auch Herausforderungen und Anforderungen solcher Projekte verdeutlicht.

---

<sup>1</sup> (Vgl. Peham, 2019)

## **2 GRUNDLAGEN**

Das folgende Kapitel untergliedert sich in drei Kapitel in denen allgemeine Begriffe der Softwareentwicklung, Begrifflichkeiten aus der Webentwicklung und die Grundlagen aus dem Bereich der künstlichen Intelligenz, im Folgenden mit KI abgekürzt, erklärt werden.

### **2.1 Allgemeine Begriffsklärung**

Im Folgenden werden grundlegende Begriffe erläutert, die in der Softwareentwicklung verwendet werden und wichtig für das Verständnis der Low-Code Anwendung sind.

#### **2.1.1 User Experience**

„User-Experience-Design oder UX-Design befasst sich mit der Analyse, Kreation und Optimierung der Nutzererfahrung. Nutzererfahrung ist ein sehr komplexes Thema, denn sie beinhaltet die komplette Erfahrung (z.B. Gedanken, Emotionen und Bedürfnisse) eines Nutzers bei der Interaktion mit einem digitalen Produkt.“<sup>2</sup> UX-Design hat in der Entwicklung eine wichtige Bedeutung und kann laufend durch Feedback von Anwendern, z.B. aus User-Tests, verbessert werden.

#### **2.1.2 Low-Code**

Der Begriff Low-Code wurde durch das US-amerikanische Marktforschungsunternehmen Forrester Research 2014 geprägt.<sup>3</sup> Low-Code beschreibt die Erstellung von Software, ohne viele Funktionen selbst durch Code schreiben zu müssen. Vielmehr werden grafische Oberflächen und Modellierungssprachen genutzt, um Funktionen zu implementieren. Durch Low-Code Entwicklungen können auch Nicht-Informatiker oder Mitarbeiter, die über nur wenig entwicklungsspezifisches Wissen verfügen, eine Anwendung erstellen. Dadurch entsteht ein Zeitersparnis und die Möglichkeit der Reduzierung von Entwicklungskosten. Durch den Low-Code Ansatz vereint man den Vorteil einer schnellen Entwicklung, die trotzdem über die Möglichkeit verfügt, komplexere individuelle Anpassungen vorzunehmen.<sup>4</sup>

---

<sup>2</sup> (Mayerhofer, 2021)

<sup>3</sup> (Vgl. Mundt, 2019)

<sup>4</sup> (Vgl. Kukut & Human, 2020)

---

Neben dem Low-Code Ansatz gibt es den No-Code Ansatz. Bei Anbietern dieser Variante hat der Anwender z.B. nicht, oder nur stark eingeschränkt, die Möglichkeit, eigenen Code einzufügen und damit die Anwendung stäker auf spezifische Bedürfnisse anzupassen.

### 2.1.3 Versionskontrolle

Versionsverwaltung ist ein System, welches Änderungen an einer oder einer Reihe von Dateien über die Zeit hinweg protokolliert, sodass man später auf eine bestimmte Version zurückgreifen kann.<sup>5</sup> Git ist ein Open Source System zur Versionskontrolle, welches die Möglichkeit gibt, für jede neue Entwicklung eine neue Verzweigung (Branch) zu erstellen. Ein Verzweigungsmodell ist in Abbildung 1 zu sehen. Diese Verzweigung ermöglicht es, mehrere Entwicklungsstände zu verwalten.



Abbildung 1: Darstellung des Verzweigungsmodells von Git<sup>6</sup>

Weit verbreitete, auf Git basierende Plattformen für eine Versionsverwaltung sind beispielsweise GitLab und GitHub.

## 2.2 Grundlagen der Webentwicklung

Nachdem allgemeine Begriffe der Softwareentwicklung erläutert wurden, folgen nun grundlegende Begriffe und Technologien der Webentwicklung. Diese sind für das Verständnis der Low-Code Applikation notwendig.

---

<sup>5</sup> (Vgl. Git, kein Datum)

<sup>6</sup> (Vgl. Git, kein Datum)

### 2.2.1 Hypertext Markup Language (HTML)

Durch HTML kann der strukturelle Aufbau einer Internetseite bestimmt werden.<sup>7</sup> Durch ein HTML Gerüst kann der Browser bei der Ausführung das sogenannte Document Object Model (DOM) erstellen.

### 2.2.2 Cascading Style Sheets (CSS)

Mithilfe von CSS können HTML-, SVG- und XML-Dokumente formatiert und gestylt werden.<sup>8</sup>

### 2.2.3 Syntactically Awesome Stylesheets (Sass)

Sass ist ein CSS Präprozessor und genau wie CSS eine Stylesheet Sprache. Die Verwendung bietet gegenüber klassischem CSS einige Vorteile, wie z.B. verschachtelte Regeln, die für eine bessere Lesbarkeit sorgen.<sup>9</sup> Abbildung 2 zeigt links ein Beispiel für die Scss Syntax und rechts eines für die klassische Schreibweise in CSS.



Abbildung 2: Gegenüberstellung einer Scss Syntax und einer klassischen CSS Schreibweise.

### 2.2.4 JavaScript

JavaScript ist eine Skriptsprache, die weit verbreitet ist und bei der Erstellung von Internetseiten genutzt wird. JavaScript kann auch außerhalb des Browsers genutzt

---

<sup>7</sup> (Vgl. selfhtml wiki, 2021)

<sup>8</sup> (Vgl. selfhtml wiki, 2020)

<sup>9</sup> (Vgl. Sass, kein Datum)

werden, z.B. für das Erstellen eines Webservers mit der JavaScript Laufzeitumgebung NodeJs.<sup>10</sup>

### 2.2.5 Javascript-Frameworks

Ein JavaScript-Framework bietet ein Grundgerüst und nützliche Funktionen für die Entwicklung. Es gibt viele verschiedene JavaScript Frameworks, die sich stark vom Funktionsprinzip und grundlegenden Aufbau unterscheiden können. Für die Webentwicklung ist man nicht gezwungen, ein Framework zu verwenden. Durch den großen Vorteil der Zeitersparnis, und damit auch Kostenersparnis, ist es aber in vielen Fällen in der heutigen Entwicklung nicht wegzudenken. Die am häufigsten genutzten Frameworks sind in Abbildung 3 dargestellt.

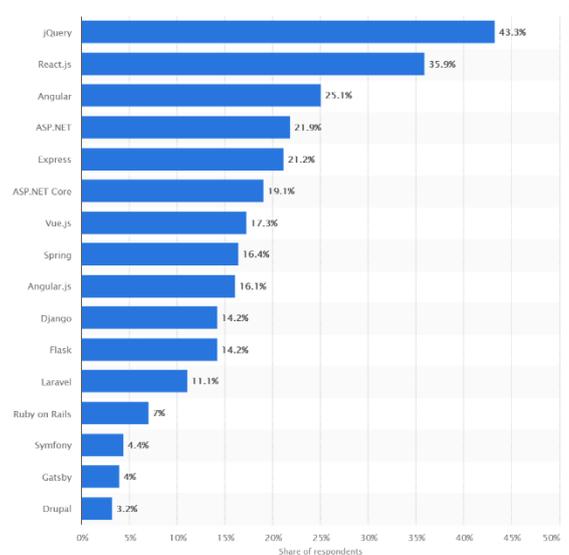


Abbildung 3: Nutzungshäufigkeit der meist genutzten Web Frameworks ab Anfang 2020<sup>11</sup>

### 2.2.6 Vue

Vue ist ein von Evan You entwickeltes<sup>12</sup> clientseitiges JavaScript Framework. Mit Vue lassen sich Internetseiten und grafische Oberflächen, auch User Interfaces (UI) genannt,

<sup>10</sup> (Vgl. Mozilla Developer Network, 2020)

<sup>11</sup> (Shanhong Liu Statista, 2020)

<sup>12</sup> (Vgl. You, kein Datum)

die immer dynamischer und somit komplexer werden, leichter und schneller erstellen. Das Framework ist aufgeteilt in drei Module: Core, Vuex und Vue-Router.

Vue ist ein progressives Framework. Es bietet die Möglichkeit, das Framework auch nur für einen Teil einer Anwendung zu nutzen, man ist nicht gezwungen, alles zu verwenden (all or nothing). Vue ist komponentenbasiert, was bedeutet, dass jedes UI Element eine eigene Komponente mit eigenem HTML, JS und CSS Teil sein kann. Ein Beispiel für den Aufbau einer Vue Komponente ist in Abbildung 4 dargestellt.

Durch diese Aufteilung erhöht sich die Zugänglichkeit, Testbarkeit und die Wartbarkeit der Anwendung. Durch das Verwenden des Frameworks besteht die Möglichkeit, bessere Performanceergebnisse zu erzielen, da nur einzelne Teile der Webseite neu geladen oder ausgetauscht werden. Vue ist ein datenbasiertes Framework. Durch Änderung der Daten aktualisiert sich die Darstellung automatisch. Die data binding Funktion von Vue baut darauf auf und erzeugt eine Reaktivität, d.h. sobald sich Daten ändern, die von Komponenten genutzt werden, werden diese neu gerendert.

```
<template>
  <q-page class="row items-center justify-evenly class1">
    <example-component
      title="Example component"
      active
      :todos="todos"
      :meta="meta"
    ></example-component>
  </q-page>
</template>

<script lang="ts">
import ExampleComponent from 'components/CompositionComponent.vue';
import { defineComponent, ref } from '@vue/composition-api';

export default defineComponent({
  name: 'PageIndex',
  components: { ExampleComponent },
  setup() {
    return {};
  }
});
</script>

<style lang="scss" scoped>
.class1 {
  background-color: red;
}
</style>
```

Abbildung 4: Beispielhafter Aufbau einer Vue Komponente

Eine Vue Komponente hat immer die gleiche Struktur: Zuerst gibt es einen `<template>` Block, indem durch HTML das Layout aufgebaut wurde. Im `<script>` Block folgt die eigentliche Logik der einzelnen Komponente. Als letztes wird über den `<style>` Block Scss Code eingefügt zum Editieren des Styles von HTML Elementen im Template Block.

### **2.2.7 Quasar**

Quasar ist ein Frontend Framework, welches auf Vue aufbaut und vorgefertigte UI Komponenten bietet. Durch die Nutzung von Quasar kann Zeit eingespart werden, da Komponenten nicht von Grund auf neu entwickelt werden müssen.

### **2.2.8 Typescript**

Typescript ist eine von Microsoft entwickelte Open-Source-Sprache, die JavaScript um mehrere Funktionen erweitert.<sup>13</sup> Typescript wird auch als ein JavaScript Superset bezeichnet. Ein Feature von Typescript ist die Typdefinition und damit einhergehende Typsicherheit. Typsicherheit gibt Entwicklern die Möglichkeit, mögliche Laufzeitfehler während der Erstellung zu erkennen. Durch die Nutzung von Typescript erreicht man eine bessere Codequalität und damit vereinfacht man auch die Fehlerfindung. Durch diese Vorteile verbessert sich die Wartbarkeit. Typescript ist in einer JavaScript Laufzeitumgebung nicht ausführbar, daher wird jeder Typescript Code am Ende immer zu JavaScript compiliert.

### **2.2.9 Application Programming Interface (API)**

Eine API ist eine Schnittstelle zwischen Programmen, welche einen Austausch von Daten ermöglicht oder Funktionen bereitstellen kann.

## **2.3 Grundlagen Künstliche Intelligenz**

Der Begriff Künstliche Intelligenz (KI) umfasst viele unterschiedliche Begriffe und Funktionsweisen, die sich in unterschiedliche Teilbereiche untergliedern. Obwohl schon lange an Künstlicher Intelligenz geforscht wird, hat sich bis jetzt keine einheitliche Definition des Begriffs durchgesetzt. Ein Grund dafür mag sein, dass sich Begriffe wie

---

<sup>13</sup> (Vgl. Microsoft, kein Datum)

---

Denken und Intelligenz auch jenseits der IT einer genauen Definition entziehen. Oft wird Künstliche Intelligenz wie folgt definiert: “Artificial Intelligence is the Ability to acquire and apply knowledge”<sup>14</sup> und bedeutet übersetzt, dass KI die Fähigkeit darstellt, Wissen zu erwerben und anzuwenden.

In der folgenden Darstellung (Abb. 5) sieht man eine grobe Aufgliederung des Bereiches Künstliche Intelligenz. Machine Learning, Neuronale Netze und Deep Learning bieten die Basis für eine Künstliche Intelligenz. Der immer wichtiger werdende Bereich des Deep learning ist ein Teilbereich des Machine learnings, auf dieses wird in Kapitel XX genauer eingegangen.

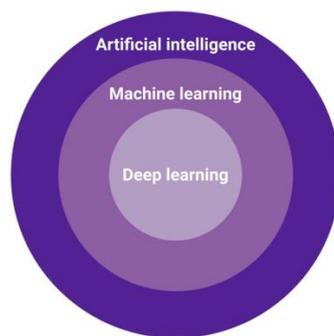


Abbildung 5: Bereiche Künstliche Intelligenz<sup>15</sup>

Eine Künstliche Intelligenz wird als „schwache künstliche Intelligenz“ bezeichnet, wenn das System in einer Tätigkeit oder einem Bereich besser als der Mensch ist, z.B. bei der Bilderkennung oder dem Autofahren. Wenn eine künstliche Intelligenz menschenähnliches Denken abbildet, spricht man von einer „starken künstlichen Intelligenz“.

### 2.3.1 Machine Learning

Wie bereits in der Abb. 6 vorweg dargestellt, ist Machine Learning ein Teilgebiet der Künstlichen Intelligenz. Das im Deutschen übersetzte maschinelles Lernen steht für das Erkennen von Mustern in vorliegenden Datenbeständen und bringt IT-Systeme in die Lage eigenständig Lösungen für Probleme zu finden.<sup>16</sup>

---

<sup>14</sup> (Vgl. Copeland, kein Datum)

<sup>15</sup> (Gerard, 2020)

<sup>16</sup> (Vgl. Luber & Litzel, BIGDATA INSIDER, 2016)

Wie der Begriff der künstlichen Intelligenz, umfasst auch der Begriff Machine Learning viele unterschiedliche Teilbereiche und Umsetzungsmöglichkeiten. Abhängig von den Gegebenheiten unterscheidet sich die Funktionsweise, wie ein Computersystem lernt, grundlegend werden jedoch immer Trainingsdaten, welche erzeugt oder vorhanden sind, genutzt, um ein System mit Trainingsdaten zu trainieren. Im Folgenden werden vier verschiedene Lernmethoden erläutert.

### **1. Überwachtes Lernen (supervised learning)**

Bei dieser Variante werden Trainingsdaten verwendet, die eine vollständige Zuordnung zwischen bestimmten Daten und ihrer Bedeutung haben. Als Beispiel wird ein Bild, auf dem ein Hund abgebildet ist, als Hund gelabelt. Diese Daten werden als labeled data bezeichnet.

### **2. Unüberwachtes Lernen (unsupervised learning)**

Beim unüberwachten Lernen werden keine konkreten Vorhersagen getroffen, sondern es wird eine Klassifizierung auf Basis von Daten erstellt, die keine Zuordnung zu ihrer Bedeutung haben (unlabeled data). Beispiele sind hier das Klassifizieren von Kundengruppen anhand der gekauften Produkte und ihrem Nutzerverhalten.

### **3. Teilüberwachtes Lernen (semi-supervised learning)**

Bei dieser Variante werden Trainingsdaten genutzt, die keine vollständige Zuordnung von Daten zu ihrer Bedeutung haben. Das System benötigt nur einige Zuordnungen, die manuell erstellt worden sind. Um den Rest der Zuordnungen selbständig zu erstellen, sucht das System dann nach Ähnlichkeiten. Der Vorteil bei dieser Methode ist, dass man sehr schnell einen Trainingsdatensatz erstellen kann.

### **4. Bestärkendes Lernen (reinforcement learning)**

Reinforcement learning wird benötigt, wenn Programme und Maschinen komplexe Aufgaben erfüllen müssen, die sich entweder nur schwer oder gar nicht auf typischem Wege codieren lassen oder keine Trainingsdaten verfügbar sind.

Bei dieser Methode wird durch Versuch und Irrtum (oder Erfolg) versucht herauszufinden, welche Aktionen gut und welche nicht gut sind. Aktionen, die nicht gut waren, werden bestraft und andere bestärkt. Dieses Vorgehen findet Parallelen in der Konditionierung von Tieren, z.B. wird ein gewünschtes Verhalten, das auf Kommando gezeigt wird, mit Futter belohnt.

In der Robotik wird das reinforcement learning stark eingesetzt. Anwendungsbeispiele für das reinforcement learning sind z.B. selbstfahrende Autos oder ein virtueller Computergegner in einem Computerspiel.

Grundlegend besteht reinforcement learning aus den folgenden fünf Bestandteilen: Agent, Environment, Action, Observation und Reward.

Der Agent beschreibt dabei die KI, welche trainiert wird. Dies könnte z.B. ein Computergegner in einem Spiel sein. Dieser Agent bewegt sich in der Umgebung (Environment) und führt Aktionen (Actions) aus. Die Beobachtung (Observation) stellt die Daten dar, welche als Input für das Training verwendet werden. Nachfolgend auf die vom Agent ausgeführte Aktion erhält dieser den sogenannten Reward, also eine Belohnung, welche allerdings auch eine Bestrafung sein kann. Der Reward ist die einzige Information, die ein Agent auf seine durchgeführte Aktion bekommt. Wie oft der Reward an den Agenten weiter gegeben wird, ist frei wählbar.

Die Ergebnisse eines Lernprozesses werden in einem Modell repräsentiert, welches die Entscheidungswege enthält. Als „white box models“ werden Modelle bezeichnet, die einfach nachzuvollziehen und zu verstehen sind, dagegen sind „black box models“ komplexer und die Entscheidungsfindung kann nicht einfach so nachvollzogen werden.<sup>17</sup>

---

<sup>17</sup> (Thamm, Gramlich, & Borek, 2020)

### 2.3.2 Neuronale Netze

Eine Möglichkeit, ein Machine Learning Modell zu erstellen, ist die Entwicklung neuronaler Netze. Ein neuronales Netz besteht aus einzelnen künstlichen Neuronen, die miteinander verknüpft und in mehreren Schichten (Layer) hintereinander geschaltet sind (siehe Abb. 6). Wie genau die einzelnen Neuronen und Schichten miteinander verknüpft sind, kann sich stark unterscheiden. Eine vollständige Ausführung der unterschiedlichen Vernetzungsarten ist jedoch für das grundlegende Verständnis über den Aufbau nicht nötig und daher hier auch nicht weiter ausgeführt. Neuronale Netze werden in drei Schichten unterteilt:<sup>18</sup>

#### 1. Input Layer

Die als Basis für eine Vorhersage dienenden Daten werden durch diese Schicht dargestellt.

#### 2. Hidden Layer

Der Hidden Layer stellt die verborgenen Schichten dar, also alle Schichten, die zwischen dem Input und dem Output Layer liegen. Die Anzahl dieser verborgenen Schichten ist variabel und nicht fix festgelegt.

#### 3. Output Layer

Im Output Layer wird das Ergebnis oder auch die Vorhersage dargestellt.

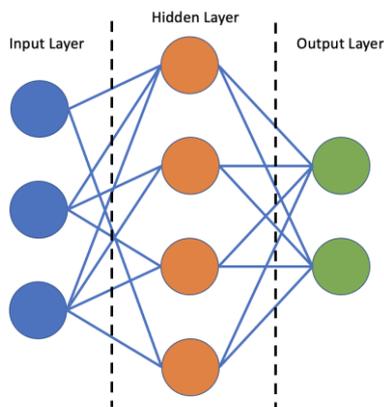


Abbildung 6: Aufbau eines neuronalen Netzes

<sup>18</sup> (Vgl. Jannis, 2021)

Je mehr Schichten ein neuronales Netz besitzt, desto höher ist auch die Möglichkeit der Abstraktion, also die Funktion z.B. bei einer Gesichtserkennung möglichst viele Details zu erkennen, nimmt zu.

Für die einzelnen Neuronen werden Gewichte festgelegt, die bestimmen wie stark der Input eines Neurons die „Aktivierung“ des folgenden Neurons beeinflusst. Für die Aktivierung werden mathematische Aktivierungsfunktionen genutzt.

### **2.3.3 Deep Learning**

An dem sogenannten Deep learning wird seit ca. 1995 gearbeitet.<sup>19</sup> Deep Learning ist eine spezielle Art (subset) des Machine learnings, dabei werden neuronale Netze mit künstlich erzeugten Layers (mit künstlich erstellten Neuronen) erstellt und verwendet. Dabei zeichnet die große Anzahl an Layers, also die Verschachtelung, Deep Learning aus.

Deep Learning benötigt durch die hohe Anzahl an Schichten eine viel höhere Rechenleistung und wird durch die immer größere Verfügbarkeit von mehr Rechenleistung (GPU`s) leichter nutzbar. Daneben erhöht sich auch die Trainingsdauer des Neuronalen Netzes. Diese kann, nicht wie bei „normalem“ Machine Learning Minuten und Stunden, sondern Wochen bis Monate betragen. In der folgenden Abbildung 7 wird der Unterschied von „klassischem“ Machine Learning und Deep Learning dargestellt.

---

<sup>19</sup> (Vgl. Ertel, 2016)

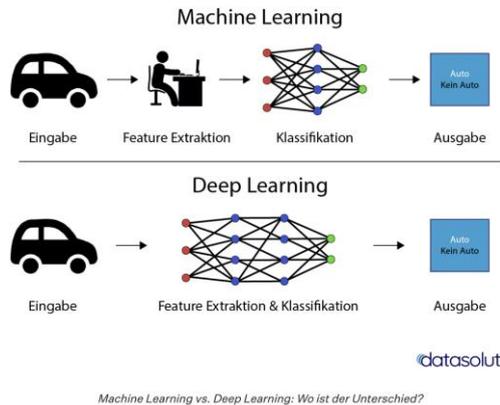


Abbildung 7: Vergleich zwischen klassischem M.L. und Deep Learning<sup>20</sup>

Beim Deep Learning ändern sich auch die Anforderungen an die Trainingsdaten und es besteht die Möglichkeit, eine viel höhere Anzahl an Datensätzen für ein Training zu verwenden. In Abb. 8 wird beispielhaft der Unterschied zwischen Deep Learning und anderen Lernmethoden dargestellt.

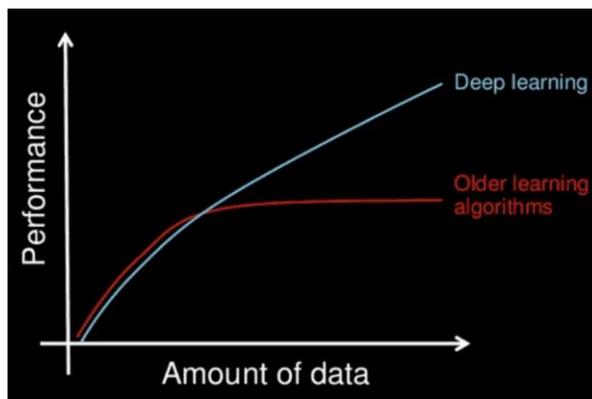


Abbildung 8: Performance Vergleich zwischen "klassischem" M.L. und Deep Learning<sup>21</sup>

Es gibt viele verschiedene Arten von neuronalen Netzen, die sich z.B. anhand ihres Aufbaus unterscheiden. Neben den neuronalen Netzen gibt es auch andere Techniken, wie Machine Learning umgesetzt werden kann. Diese werden jedoch hier nicht weiter ausgeführt.

<sup>20</sup> (Vgl. Wuttke, datasolut, kein Datum)

<sup>21</sup> (Vgl. Ng, kein Datum)

### 2.3.4 Trainingsdaten

Wie gut das Ergebnis des Lernprozesses eines Modells ist, hängt von den verwendeten Trainingsdaten ab. An Trainingsdaten werden bestimmte Anforderungen gestellt, diese müssen von der Menge her ausreichend sein, ausreichend differenziert und vollständig sein.<sup>22</sup>

Je größer und umfangreicher die Testdaten sind, desto länger dauert es, ein Modell zu trainieren. Trainingszeiten wachsen oft exponentiell mit der steigenden Komplexität der Eingabedaten.

Je besser Trainingsdaten den Anforderungen gerecht werden, desto besser ist auch das Ergebnis des trainierten Modells. Bei der Entwicklung von einer Künstlichen Intelligenz kann es aufgrund von ungenügenden Trainingsdaten, z.B. wenn die oben genannten Kriterien nicht erfüllt werden, zu einem Bias kommen. Ein Bias ist eine Verzerrung der Realität und sorgt z.B. bei einer Vorhersage für ein „falsches“ Ergebnis. Es gibt viele unterschiedliche Arten von einem Bias wie z.B. ein Confirmation, Selection (sample bias) oder auch Prejudice Bias.

Dabei ist anzumerken, dass ein Bias minimiert, aber nicht vollständig vermieden werden kann.

Bei einer hohen Anzahl an Trainingsdatensätzen besteht die Möglichkeit einer Aufteilung dieser Daten. Dabei werden die aufgeteilten Daten in mehreren Durchläufen oder auch Epochen für das Training verwendet.

Neben den Trainingsdaten benötigt man zum Testen des Modells auch Evaluationsdaten. Mit diesen Daten wird geprüft, ob das Modell auch korrekte Ergebnisse liefert. Die Evaluationsdaten dürfen dem Modell nicht „bekannt“ sein, also nicht für das Training verwendet worden sein.

Die Datenqualität kann durch verschiedene Vorgehensweisen und Methoden gesteigert werden.

---

<sup>22</sup> (Gerard, 2020)

Als Quelle für Trainingsdaten gibt es viele Open Source Datensätze, die man, neben den eigenen Daten, als Trainingsdaten verwenden kann, dazu zählt z.B. auch die MNIST Datenbank. In dieser befinden sich 60.000 Trainingsdatensätze, welche vollständig manuell gelabelt wurden.<sup>23</sup> Ein weiteres Beispiel ist z.B. die Fashion-MNIST Datensammlung von Zalando<sup>24</sup>, welche wie am Namen schon erkennbar ist, an die MNIST Datenbank angelehnt ist.

Neben den Open Source Trainingsdaten gibt es auch viele ML Modelle, die unter einer Open Source Lizenz frei verfügbar sind. Trainingsdatensätze, welche nicht selbst erstellt worden sind, sollten genau geprüft und auf deren Nutzbarkeit untersucht werden.

Trainingsdaten zu generieren zählt zu den größeren Herausforderungen im Entwicklungsprozess eines Machine Learning Modells.

### **2.3.5 Transfer Learning**

„Unter Transfer Learning versteht man das Übertragen der Ergebnisse eines fertig trainierten neuronalen Netzes auf eine neue Aufgabe“.<sup>25</sup> Dadurch lassen sich schon fertig entwickelte komplexe Modelle mit eigenen Daten erweitern. So kann z.B. ein Modell für Spracherkennung, welches mit dem normalen Wortschatz trainiert worden ist, genutzt und mit eigenen Wörtern erweitert werden. Transfer Learning kann eine wichtige Rolle bei der Entwicklung eigener Modelle spielen, da dadurch Zeit eingespart werden kann. Einheitliche Formate wie das „Open Neural Network Exchange (ONNX)“, welches von Amazon, Facebook und Microsoft entwickelt wurde, könnten bei dem Austausch von Modellen oder bei der Übertragung in andere Tools oder Umgebungen hilfreich sein.<sup>26</sup> Dieses Format dient zum Austausch von Deep Learning Modellen.

### **2.3.6 Natural Language Processing**

Ein weiterer großer Bereich von künstlicher Intelligenz ist die Verarbeitung von Sprache. Dies wird als Natural Language Processing, folgend mit NLP abgekürzt, bezeichnet. Im

---

<sup>23</sup> (Vgl. LeCun, Cortes, & Burges, kein Datum)

<sup>24</sup> (Vgl. Zalando, kein Datum)

<sup>25</sup> (Becker, 2019)

<sup>26</sup> (Vgl. Luber & Litzel, BIGDATA INSIDER, 2019)

Folgendes wird ein Überblick über die genutzten Techniken gegeben, die notwendig sind um Texte als Eingabe für ein ML nutzen zu können.

Um z.B. Texte für Trainingsdaten oder für Vorhersagen zu verwenden, muss das Format des Textes generalisiert werden.

Bei der Tokensierung wird ein vorliegender Text in die dort enthaltenen einzelnen Wörter aufgeteilt, die dann als Token bezeichnet werden.<sup>27</sup> Da es je nach Sprache viele Varianten eines Wortes geben kann, wird jedes Wort auf seinen Wortstamm reduziert. Dieser Vorgang wird als Stemming bezeichnet.

Neben dem Stemming gibt es auch die Möglichkeit der Lematisierung, durch die unterschiedliche Wörter der gleichen Bedeutung zugeordnet werden. Für das Verarbeiten von Text müssen nicht beide Methoden verwendet werden.

Füllwörter, also Wörter, die in einem Satz keine Bedeutung haben, werden als stop words bezeichnet und können entfernt werden. Dadurch kann die Größe der Trainingsdaten reduziert werden. Die Vorgehensweise kann sich je nach Verwendungszweck unterscheiden.

### **2.3.7 Cloud und Edge KI**

Es gibt verschiedene Möglichkeiten wie man KI-Systeme betreiben kann. Die erste Möglichkeit ist der Betrieb in der Cloud, also in einem externen oder eigenen Rechenzentrum. Die zweite Möglichkeit ist, das KI-System bzw. das ML Modell direkt auf dem Endgerät oder auf Systemen am gleichen Standort zu betreiben.

Je nach Einsatzgebiet und Anforderungen kommt entweder die eine oder die andere Art in Betracht. Beide Arten haben die grundlegende Funktionsweise von Neuronalen Netzen und Machine Learning gemeinsam, sie unterscheiden sich aber in Geschwindigkeit, Hardware- und Betriebskosten, Datenschutz und Datensicherheit.

Ob ein KI Modell direkt auf dem Gerät oder in der Cloud betrieben wird, hängt stark vom Anwendungsfall ab. Beim Edge Computing läuft das Modell direkt oder in der Nähe des Endgerätes und somit auch an der Stelle, an der die dazugehörigen Daten gesammelt werden.<sup>28</sup> Diese Lösung kann für Produktionen in der Industrie optimal sein, da so Daten

---

<sup>27</sup> (sree017, kein Datum)

<sup>28</sup> (1&1 IONOS, 2020)

über die Supply-Chain nicht weiter übermittelt werden müssen. Beim Edge Computing steht oft weniger Speicherplatz zur Verfügung als beim Cloud Computing. Das hat direkten Einfluss auf die Menge an Trainingsdaten, welche genutzt werden können. Edge Computing benötigt meist weniger Zeit für eine Berechnung, welche allerdings durch die wenigen Trainingsdaten an Genauigkeit verliert. Die schnelle Reaktionszeit ist gerade in der Industrie (Stichwort Industrie 4.0) wichtig, da dort Echtzeiterkennungen notwendig sind. Im Gegensatz dazu hat man beim Cloud Computing praktisch unendlich viel Speicher zur Verfügung und kann beliebige Mengen Daten für den Trainingsprozess und für Berechnungen verwenden, was dann jedoch einen großen Zeitaufwand zur Folge hat. Bei einer hybriden oder auch Fake Edge AI kann das Modell z.B. in der Cloud trainiert und dann auf ein Gerät übertragen und dort verwendet werden.

Cloud und Edge AI können auch in Kombination genutzt werden, z.B. wenn ein Gerät aus lokal gesammelten Daten lernen und diese für eine Vorhersage nutzt, aber bei Bedarf auf das KI-System in der Cloud zurück greifen kann.<sup>29</sup>

### **3 KÜNSTLICHE INTELLIGENZ HEUTE**

Heutzutage findet künstliche Intelligenz in vielen unterschiedlichen Bereichen Anwendung. Neben dem Einsatz für Einparkhilfen im Auto oder Serienvorschlägen finden sich auf viele Einsatzmöglichkeiten bei Logistikrobotern, Prozessen für Betrugserkennung oder im Medizinischen Bereich z.B. für die Krebserkennung.

---

<sup>29</sup> (Hempleton, 2021)

Jedoch gibt es keine universell einsetzbare Künstliche Intelligenz, KI Systeme sind immer auf einen bestimmten Zweck ausgelegt, wie z.B. die Klassifizierung von Bildern oder die Textverarbeitung.

### 3.1 Aktuelle Nutzung von KI

Gemäß einer Studie von McKinsey könnte sich das Bruttoinlandsprodukt von Deutschland durch den Einsatz von selbstlernenden Systemen um bis zu vier Prozent steigern bis zum Jahre 2030.<sup>30</sup> Im Vergleich dazu könnte der weltweite Markt für KI-basierte Dienstleistungen auf 130 Milliarden US-Dollar steigen, was einer Steigerung von 25% jährlich entsprechen würde bis zum Jahr 2025.<sup>31</sup>

Wie in der unten stehenden Abb. 9 zu erkennen ist, wird ein starker Anstieg des Umsatzes erwartet, der durch Unternehmensanwendungen, die in dem Bereich der KI angesiedelt sind, generiert wird.

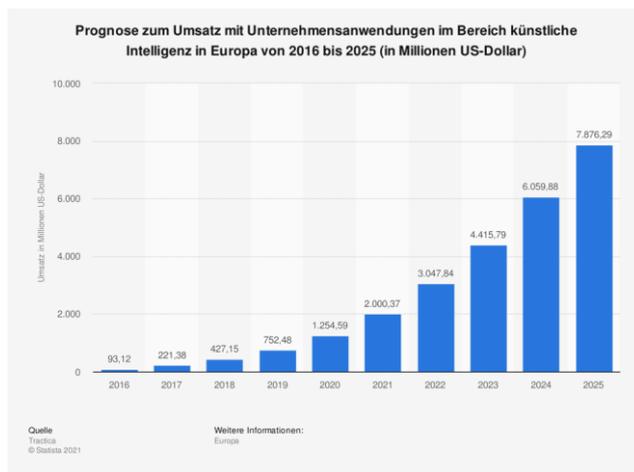


Abbildung 9: Umsatzprognose für KI Unternehmensanwendungen<sup>32</sup>

<sup>30</sup> (Vgl. Buxmann & Schmidt)

<sup>31</sup> (Vgl. Buxmann & Schmidt)

<sup>32</sup> (Tractica, 2016)

### **3.2 Deutschland als KI Standort**

In Deutschland wird aktive Forschung im Bereich der Künstlichen Intelligenz betrieben, einige der bedeutendsten Standorte dafür sind, Berlin, Dortmund, St. Augustin, Saarbrücken, Tübingen, Leipzig, Dresden und München.<sup>33</sup> Um Deutschland als KI-Standort attraktiver zu machen, wurde von der Bundesregierung die Nationale KI-Strategie initiiert. Diese Strategie umfasst insgesamt 12 Handlungsfelder<sup>34</sup> und hat unter anderem das Ziel, „Artificial Intelligence (AI) made in Germany“ als weltweit anerkanntes Gütesiegel zu platzieren.<sup>35</sup>

### **3.3 Anforderungen an KI-Projekte**

KI-Projekte müssen mehrere Anforderungen aus unterschiedlichen Bereichen erfüllen. Anforderungen aus dem Bereich Sicherheit sind beispielsweise der Schutz gegen ungewollte Zugriffe von außen und innen, durch die möglicherweise Trainingsdaten verändert werden und somit den Output des Systems beeinflussen. Da auch bereits kleinste Änderungen an Trainingsdaten zu fehlerhaftem Output führen können<sup>36</sup>, muss der Zugriff auf das System kontrolliert werden und das System selbst muss regelmäßigen Kontrollen unterliegen. Durch Angriffe auf KI-Systeme und der damit verbundenen Manipulation von Daten ist es möglich, Änderungen vorzunehmen, die von Menschen nur schwer bemerkt werden können. Ein Beispiel dafür ist die Manipulation von Bilddateien, bei denen das dargestellte Bild sich für den Menschen nicht sichtbar verändert. Neben solchen Sicherheitsanforderungen müssen auch rechtliche Anforderungen erfüllt werden. Diese werden unter anderem auch durch die Datenschutzgrundverordnung definiert. Es muss nachvollziehbar sein, wo und welche Daten gesammelt werden, wie diese verarbeitet werden und zu welchen externen Unternehmen die Daten möglicherweise weitergegeben werden. Dies kann bei KI-Systemen und AI as a Service Lösungen zu größerem administrativen Aufwand führen oder zu neuen Prozessen, in denen diese Daten erst anonymisiert werden müssen.

---

<sup>33</sup> (Vgl. Bundesministerium für Bildung und Forschung, kein Datum)

<sup>34</sup> (Vgl. Bundesministerium für Bildung und Forschung, kein Datum)

<sup>35</sup> (Vgl. Bundesministerium für Bildung und Forschung, 2018)

<sup>36</sup> (Vgl. PwC, 2019)

Bei KI-Systemen, die in Unternehmen eingesetzt werden oder von diesen genutzt werden und sich nicht ausschließlich auf Forschung beschränken, kommt die Anforderung der wirtschaftlichen Rentabilität hinzu. Dabei ist das Ziel bei dem Einsatz von KI eine Kostenersparnis zu erreichen, Prozesse zu optimieren und über einen Wettbewerbsvorteil zu verfügen.

Durch die in Kapitel 2.3.7 erwähnten unterschiedlichen Möglichkeiten bei der Betreuung eines ML Modells, lässt sich die Komplexität der Datenschutzanforderungen verringern. Durch Edge Computing reduziert man die Daten, die übertragen werden müssen und kann die zu übertragenden personenbezogene Daten, auf ein Minimum reduzieren.

Neben den technischen Anforderungen und den Anforderungen, die im Entwicklungsprozess entstehen, gibt es auch den Aspekt der Akzeptanz, der bei der Entwicklung und Implementierung berücksichtigt werden sollte. Um die Akzeptanz und somit auch das Vertrauen von Mitarbeitern und Kunden in KI Lösungen herzustellen, sollten folgende Bereiche beachtet werden:

**1. Fairness**

Wie können Verzerrungen vermieden werden und der Bias von einem Modell reduziert werden? Dies betrifft die genutzten Daten und das genutzte Modell.

**2. Interpretierbarkeit**

Wie können die Entscheidungen, also das Ergebnis eines ML Modells nachvollzogen werden?

**3. Robustheit & Sicherheit**

Wie kann die Sicherheits und auch die Performance sichergestellt werden?

**4. Governance**

Unterliegt das Modell regelmäßigen Kontrollen und wird die Qualität der Trainingsdaten geprüft?

**5. Ethische Fragen**

Wie werden die Entscheidungen von den Mitarbeitern von dem KI System beeinflusst?<sup>37</sup>

---

<sup>37</sup> (Vgl. PwC, 2019)

### 3.4 KI/AI as a Service

Durch den immer größer werdenden „as a Service“ Trend, gibt es auch immer mehr Möglichkeiten, fertige KI-Systeme bzw. ML Modelle „as a Service“ direkt zu verwenden. Alternativ wird auch die komplette Entwicklung eines Modells in der Cloud abgewickelt und dieses dann nach der Fertigstellung exportiert, wahlweise in die extern verwaltete Cloud oder in die eigene Umgebung. Dadurch reduziert sich der eigene Entwicklungsaufwand und es besteht die Möglichkeit der besseren Skalierung durch einen bekannten Kostenplan. In der folgenden Abb. 10 ist ein Beispiel einer Kostenaufstellung dargestellt. Diese gehört zu der Natural Language API von Google.

Feature	0 bis 5.000	Über 5.000 bis 1 Mio.	Über 1 Mio. bis 5 Mio.	Über 5 Mio. bis 20 Mio.
Entitätsanalyse	Kostenlos	1,00 \$	0,50 \$	0,25 \$
Sentimentanalyse	Kostenlos	1,00 \$	0,50 \$	0,25 \$
Syntaxanalyse	Kostenlos	0,50 \$	0,25 \$	0,125 \$
Sentimentanalyse pro Entität	Kostenlos	2,00 \$	1,00 \$	0,50 \$

Feature	0 bis 30.000	Über 30.000 bis 250.000	Über 250.000 bis 5 Mio.	Über 5 Mio.
Inhaltsklassifizierung	Kostenlos	2,00 \$	0,50 \$	0,10 \$

Abbildung 10: Preistabelle der Natural Language API von Google<sup>38</sup>

Die Anzahl der Einheiten werden bei diesem Dienst durch die Anzahl an Unicode-Zeichen berechnet, bei der 1000 Zeichen eine Einheit ergeben.

Wie stark der Trend ist, KI-Lösungen als Service aus der Cloud zu nutzen, wird durch eine Studie von Deloitte Global verdeutlicht. In dieser Studie wird geschätzt, dass bis 2022 Unternehmen rund die Hälfte ihres IT-Budgets für Subscription-Services ausgeben werden.<sup>39</sup>

Nur 16 % der KI einsetzenden Unternehmen haben die KI-Anwendungen selbst entwickelt. In 24 % erfolgte die Entwicklung sowohl durch das Unternehmen selbst als auch durch Dritte. 60 % griffen auf KI Entwicklungen durch Dritte zurück.<sup>40</sup>

<sup>38</sup> (Google, kein Datum)

<sup>39</sup> (Vgl. Rottstedt & Human, kein Datum)

<sup>40</sup> (Bundesministerium für Wirtschaft und Energie, 2020)

Dabei kann man die angebotenen Cloud Dienste in weitere Kategorien unterteilen. Komplette Anwendungen, in denen z.B. KI integriert ist (Software as a Service), eigene Entwicklung von Modellen (Plattform as a Service) oder nur die Abfrage des fertigen Modells über Schnittstellen abfragen zu können (Infrastruktur as a Service).<sup>41</sup>

#### **4 ENTWICKLUNG VON KI SYSTEMEN**

Da sich die Anforderungen und Funktionsweisen von KI-Systemen zu „klassischen“ Softwarelösungen unterscheiden, sind andere Entwicklungsprozesse notwendig. Durch die Anforderungen, dass ein System skalierbar, verfügbar, überwacht und wiederverwendbar sein soll, können zusätzliche Herausforderungen entstehen.

##### **Herausforderungen in der Entwicklung:**

###### **1. Skalierbarkeit**

Je nach Anwendung muss ein Machine Learning Modell viele Anfragen gleichzeitig verarbeiten können, dies benötigt viel Rechenleistung und Speicher. Die eingesetzte Hardware muss genügend CPUs, GPUs und RAM besitzen, um die angeforderte Rechenleistung stellen zu können.

---

<sup>41</sup> (Vgl. 1&1 IONOS, 2021)

## **2. Verfügbarkeit**

Da Webanwendungen ständig erreichbar sind, müssen auch die damit verbundenen Machine Learning Modelle eine Erreichbarkeit aufweisen, damit das Modell aktualisiert und für Vorhersagen genutzt werden kann.

## **3. Überwachbarkeit**

Das Modell muss überwachbar sein, um die Möglichkeit zu haben, Input und Output zu überprüfen und Fehler zu verhindern.

## **4. Wiederverwendbarkeit**

Da das Erstellen und Testen eines Machine Learning Modells sehr aufwendig sein kann, bietet es sich an, das Modell auch in anderen Bereichen für andere oder ähnliche Aufgaben zu verwenden.<sup>42</sup>

### **4.1 Entwicklungsumgebung**

Oft werden Modelle in einer anderen Umgebung, mit anderen Tools, Bibliotheken und in anderen Programmiersprachen als die eigentliche Applikation erstellt und dann in andere Umgebungen exportiert. Durch die Vielzahl an Bibliotheken und Frameworks für die Programmiersprache Python eignet sich diese hervorragend für die Entwicklung. Es bietet sich an, Modelle in Python zu erstellen, zu trainieren und dann wahlweise in eine Cloud oder in das eigene System zu exportieren. Python bietet viele Möglichkeiten, Daten interaktiv zu bearbeiten und zu analysieren. Die Machine Learning Plattform Anaconda bietet sich, durch Tools wie Jupyter Notebook oder Jupyter Labs, für die Entwicklung von Machine Learning Modellen in Python an.

Neben Tools wie beispielsweise „Open AI Gym“, welches von der Non-Profit Organisation Open AI veröffentlicht wurde und bei der Entwicklung von reinforcement learning Lösungen hilfreich sein kann<sup>43</sup>, gibt es auch Frameworks wie „Scikit-Learn“, welches viele der grundlegenden Machine Learning Algorithmen enthält. Bibliotheken wie „spaCy“ bieten Funktionen für die Entwicklung von NLP an.

---

<sup>42</sup> (Vgl. Gerard, 2020)

<sup>43</sup> (Schnellbacher, 2016)

**Keras** vereinfacht das Trainieren und Ausführen von neuronalen Netzen und bietet dafür z.B. auch eine High Level API für Deep learning an. AutoKeras vereinfacht die Wahl eines Modells für einen vorliegenden Datensatz.<sup>44</sup>

Die Open-Source Plattform **Tensorflow** ist der direkte Nachfolger von Googles Plattform Dist Belief.<sup>45</sup> Tensorflow bietet z.B. auch die Möglichkeit, mit Keras erstellte Modelle zu nutzen und direkt im Browser auszuführen. Keras ist seit Tensorflow 2.0 fester Bestandteil von Tensorflow.

Tensorflow nutzt sogenannte Tensoren. Dies sind Datenstrukturen zum Speichern von Daten, welche leichter durch TensorflowJs verarbeitet werden können.

Tensorflow bietet verschiedene Bibliotheken, um das Entwickeln von Machine Learning Modellen zu vereinfachen. Beispielsweise bietet Tensorflow Data Validation die Möglichkeit, die Qualität der Daten zu verbessern, indem z.B. fehlende oder unvollständige Daten gefunden werden.<sup>46</sup>

Außerdem bietet Tensorflow viele fertige ML Modelle für die direkte Verwendung oder die Weiterentwicklung an. Zu diesen fertigen Modellen gehören beispielsweise Modelle für die Bildklassifizierung, Gesichtserkennung, Handposenerkennung und viele weitere. Je nach Projekt können diese Modelle als Grundlage genutzt werden, um z.B. Techniken aus dem Transfer learning für die Weiterentwicklung zu nutzen.

Eine weitere Open Source Bibliothek ist „spaCy“. Diese Bibliothek bietet nützliche Funktionen für den Bereich Natural Processing Language (siehe Kap. 2.3.6) also der natürlichen Sprachverarbeitung.

## 4.2 Entwicklungsprozess

Wie schon erwähnt, unterscheidet sich der Entwicklungsprozess von KI-Projekten im Vergleich zu „klassischen“ Softwareprodukten. Bei KI Systemen ändern sich im wesentlichen drei Bestandteile, Data, Model und Code.<sup>47</sup>

---

<sup>44</sup> (Wuttke, kein Datum)

<sup>45</sup> (Vgl. Geißler & Litzel, 2018)

<sup>46</sup> (Google, kein Datum)

<sup>47</sup> (Vgl. Fowler, 2019)

In der folgenden Abb. 11 ist ein beispielhafter Ablauf der Entwicklung und Integration in einer Webapplikation dargestellt.

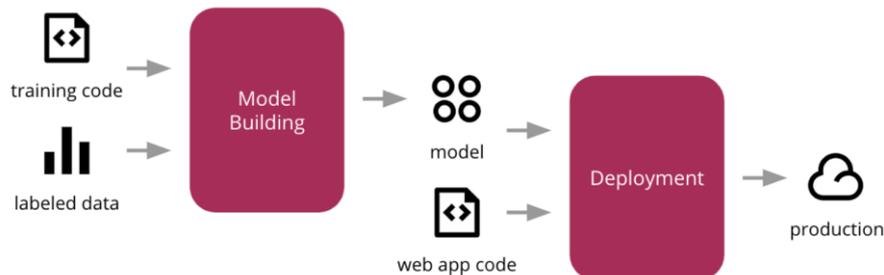


Abbildung 11: Zeigt abstrahiert den Ablauf bei der Integration<sup>48</sup>

Ein von Martin Fowler entwickeltes Schema „Continuous Delivery for Machine Learning“ beschreibt den Ablauf des Entwicklungsprozesses von Machine Learning Anwendungen und wird wie folgt beschrieben.

„Continuous Delivery for Machine Learning (CD4ML) is a software engineering approach in which a cross-functional team produces machine learning applications based on code, data, and models in small and safe increments that can be reproduced and reliably released at any time, in short adaptation cycles.“<sup>49</sup> Das Schema beschreibt also, wie funktionsübergreifende Teams durch Code, Daten und Modelle Machine Learning Lösungen entwickeln können, welche reproduzierbar und zuverlässig sind und in kleinen Anpassungsintervallen veröffentlicht werden können. Dabei wird in kleinen und „sicheren“ Schritten vorgegangen. Dieser Prozess besteht aus den nachfolgenden sechs Schritten:

### 1. Model building

Zuerst wird das Modell erstellt und mit den zur Verfügung stehenden Trainingsdaten trainiert.

### 2. Model evaluation and experimentation

<sup>48</sup> (Fowler, 2019)

<sup>49</sup> (Fowler, 2019)

Nachdem die Trainingsphase abgeschlossen ist, muss das Modell evaluiert werden. Dazu könnten z.B. wie schon in Kapitel 2.3.4 erwähnt, zurückgehaltene Trainingsdaten genutzt werden, um Korrektheit von Vorhersagen zu prüfen.

### **3. Production the Model**

Dieser Schritt folgt der Evaluation und bereitet das Modell vor, um es dann z.B. in eine Produktionsumgebung nutzen zu können

### **4. Testing**

In diesem Schritt wird überprüft, ob das für die Überführung in die Produktionsumgebung fertige Modell wie erwartet funktioniert und die Ergebnisse denen des zweiten Schrittes (evaluation und experimentation) entsprechen.

### **5. Deployment**

Nach der erfolgreichen Testphase kann das Modell dann z.B. in eine Produktionsumgebung genutzt werden.

### **6. Monitoring**

Da sich jederzeit Änderungen ergeben können, müssen die Daten, welche als Input genutzt werden, stetig überprüft werden.<sup>50</sup>

In den verschiedenen Schritten kommen unterschiedliche Rollen zum Einsatz, da andere Fachkenntnis benötigt wird. Abb. 12 verdeutlicht, dass möglicherweise ganz unterschiedliche Teams in den Entwicklungsprozess miteingebunden sind. In diesem Beispiel ermöglicht der Data Engineer den Zugriff auf die Daten und stellt den Datenfluss sicher. Diese Daten können dann vom Data Scientist, welcher sich um die Entwicklung und Verbesserung des Modells kümmert, genutzt werden. Im Anschluss muss das Modell in die Applikation eingebunden werden. Diesen Prozessabschnitt übernehmen dann ML engineers.

---

<sup>50</sup> (Vgl. Fowler, 2019)

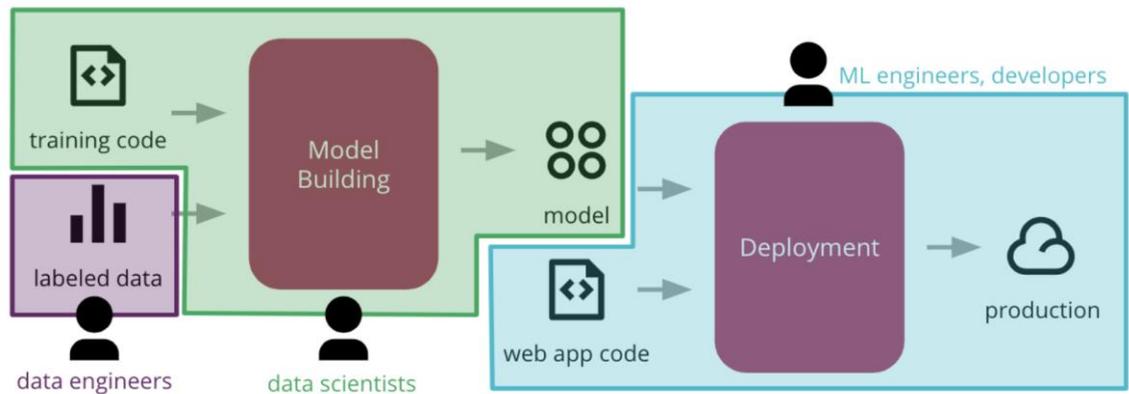


Abbildung 12: Beispielhafte Rollenverteilung im Entwicklungsprozess<sup>51</sup>

Eine Herausforderung in diesem Prozess ist die Abstimmung unter den unterschiedlichen Teams, die unterschiedlichen Arbeitsabläufen folgen und andere Tools einsetzen.<sup>52</sup> Die Abstimmung unter den Teams hat eine zentrale Bedeutung, da auch dies direkten Einfluss auf die Qualität des Modells und Projektes hat.

Bei der Entwicklung von ML-Systemen liegt der größte Arbeitsaufwand nicht zwangsläufig bei der eigentlichen Erstellung des Modells, sondern in den Prozessen, die durch die Entwicklung und Aufbereitung benötigter Daten entstehen. Dies wird mit der folgenden Abb. 13 verdeutlicht, in der die Prozesse rund um die eigentliche Erstellung des Codes für ein Modell dargestellt werden.

<sup>51</sup> (Fowler, 2019)

<sup>52</sup> (Vgl. Fowler, 2019)

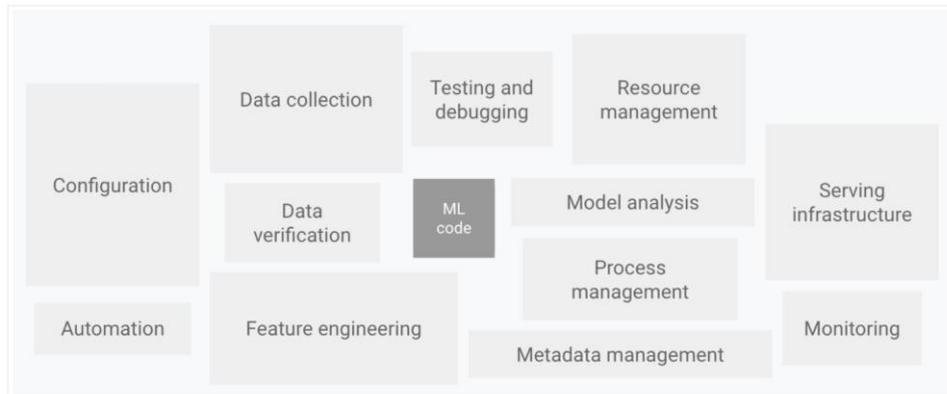


Abbildung 13: Anfallende Prozesse bei der Entwicklung eines ML Modells<sup>53</sup>

Nachdem grundlegende Begrifflichkeiten der Webentwicklung und künstlicher Intelligenz erläutert wurden und ein Überblick über den heutigen Stand und allgemeine Entwicklungsprozesse gegeben wurde, folgt nun die Untersuchung zu möglichen Implementierungsmöglichkeiten von künstlicher Intelligenz in Low-Code Entwicklungsumgebungen. Dies wird anhand von praktischen Szenarien, die das Portfolio der Vectorsoft AG betreffen, aufgezeigt.

---

<sup>53</sup> (Google, kein Datum)

## **5 KÜNSTLICHE INTELLIGENZ UND LOW-CODE**

In den vorangegangenen Kapiteln wurde aufgezeigt, wie breit die Einsatzmöglichkeiten von künstlicher Intelligenz sind und welche grundlegenden Ansätze es gibt, KI in ein Unternehmen oder Produkt einzubauen. Im Folgenden wird spezifisch untersucht, wie KI in eine Webanwendung eingebaut werden kann und welche Vorteile dadurch entstehen können. Bei der Webanwendung, auf welche sich diese Untersuchung bezieht, handelt es sich um die webbasierte Low-Code Entwicklungsplattform, die zurzeit von der Vectorsoft AG entwickelt wird. Im Folgenden Kapitel wird die Vectorsoft AG kurz vorgestellt, um die Grundlage der anschließenden Ausführungen zu bilden.

### **5.1 Vectorsoft AG**

Die Vectorsoft AG wurde 1979 gegründet und hat ihren Sitz in Heusenstamm, südlich von Frankfurt am Main. Das Unternehmen entwickelt und vertreibt das Softwareentwicklungswerkzeug mit integrierter Datenbank CONZEPT 16. Dieses ermöglicht eine effiziente Programmierung von Software per Rapid Application Development (RAD).

CONZEPT 16-Applikationen lassen sich unabhängig von der Userzahl in maximal einer halben Stunde installieren und sind vollständig administrationsfrei. Softwarepartner und -entwickler weltweit erstellen Branchen- und Standardapplikationen auf Basis von CONZEPT 16. Die Anwendungen reichen von einfachen Adressverwaltungen über Warenwirtschafts- und Informationssysteme bis hin zu komplexen PPS- und Finanzmanagement-Lösungen.

### **5.2 Softwareentwicklung**

Die Vectorsoft AG entwickelt zurzeit ein neues Produkt mit dem Namen „yeet“. Yeet wird mit dem Ziel entwickelt, eine neue, innovative und webbasierte Low-Code Entwicklungsumgebung mit einer einzigartigen Nutzererfahrung zu bieten. Für die schnellere Entwicklung baut yeet zunächst auch auf bestehender CONZEPT 16 Infrastruktur auf.

Die neue Low-Code Entwicklungsumgebung soll eine möglichst breite Masse an Nutzern ansprechen. Damit bestehen verschiedene Herausforderungen, wie z.B. die Erfüllung der Bedürfnisse unterschiedlicher Nutzertypen zu erfüllen. Auf der einen Seite gibt es

Bestandskunden, die sich Kontinuität und den Erhalt der bisherigen hohen Funktionalität wünschen. Auf der anderen Seite gibt es potenzielle neue Nutzer, die noch keine Kenntnisse in CONZEPT 16 haben und sich eine moderne Software wünschen, die einfach, intuitiv und ohne großen Lernaufwand zu bedienen ist.

Integraler Bestandteil des Softwareangebots der Vectorsoft AG ist der direkte Support durch CONZEPT 16-Spezialisten sowie individuelles Software-Consulting.<sup>54</sup>

Ein Hauptelement der Plattform yeet ist der sogenannte Designer. Der Bereich, in dem der Low-Code Anwender eine grafische Oberfläche (GUI) bauen kann. In der folgenden Abb. 14 ist der Designer dargestellt. Auf diesen kann, wie bei einer „klassischen“ Internetseite über den Browser zugegriffen werden. Im linken Bereich sieht man den Navigationsbereich und die dem Anwender zur Verfügung stehenden UI Komponenten. Im Zentrum steht die Arbeitsfläche, in der dann eine vollständig benutzerdefinierte GUI erstellt werden kann.

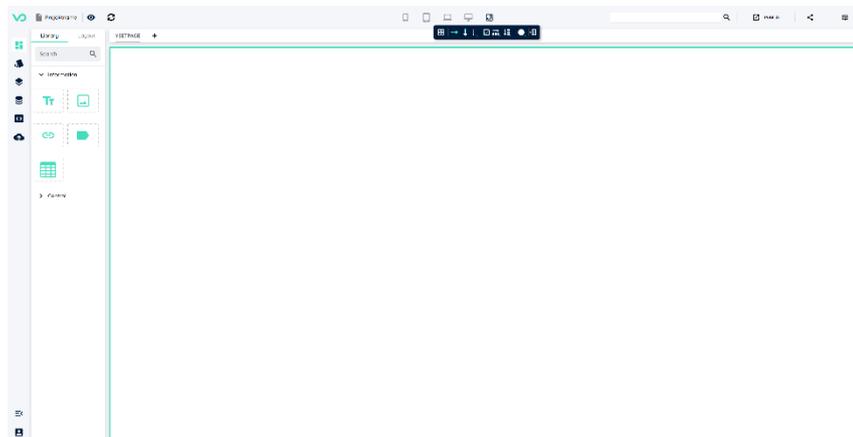


Abbildung 14: Der Designer - Auszug aus einem Entwicklungsstand, zeigt nicht die finale Version.

### 5.3 Ausgangssituation und Identifikation der Rahmenbedingungen

Es sollte zunächst untersucht werden, in welchen Prozessen, ob intern im Unternehmen oder in der eigentlichen Low-Code Entwicklungsumgebung, also beim Anwender, Verbesserungspotenziale oder Engpässe bestehen und welche Problemstellungen

---

<sup>54</sup> (Vectorsoft AG, kein Datum)

vorliegen. Sobald ein oder mehrere Problemstellungen identifiziert sind, kann geprüft werden, wie diese behoben werden können und ob der Einsatz von KI sinnvoll ist.

Beispielsweise kann eine Sprachverarbeitung die Bedienung für den Anwender erleichtern oder durch eine automatische Klassifizierung von Daten die Analyse und Auswertung von Kundendaten erleichtert werden, um fehlende Funktionalitäten zu erkennen oder bessere Vorschläge bei Suchen zu ermöglichen. Zunächst kann geprüft werden über welche bestehenden Daten man verfügt oder durch welche Methoden welche Art von Daten gesammelt werden können. Dies muss unter Berücksichtigung der in Kapitel 3.3 genannten Datenschutzanforderungen geschehen. Bei der Wahl einer lokalen oder Cloud KI Lösung sollten auch Latenzzeiten berücksichtigt werden, gerade bei Implementierungen, in denen z.B. Input von Anwendern in Echtzeit verarbeitet werden muss. Wartezeiten, die durch fehlende oder langsame Rückmeldungen vom System entstehen, können beim Anwender für Frustration und Abbrüche sorgen.

#### **5.4 Mögliche Problemstellungen**

Es ist schwierig zu kalkulieren, wie groß die spätere Auswirkung eines ML Modells ist und daher ist es auch schwierig abzuschätzen, ob sich die Umsetzung für ein Unternehmen lohnt.<sup>55</sup> Desweiteren stellt sich die Frage, ob das Unternehmen, gerade bei selbst entwickelten Modellen, über die nötige Fachkenntnis und die nötigen Ressourcen verfügt oder sich diese aneignen oder einkaufen kann. Durch die in Kapitel 3.4 erwähnten vielen Möglichkeiten, KI durch externe Service Anbieter einzukaufen, ist es möglicherweise sinnvoller auf einen von diesen zurück zu greifen. Wie in der folgenden Abb. 15 dargestellt ist, entwickeln die meisten Unternehmen KI Lösungen nicht vollständig selbst sondern nutzen vorgefertigte Lösungen.

---

<sup>55</sup> (Vgl. Kunze, 2021)

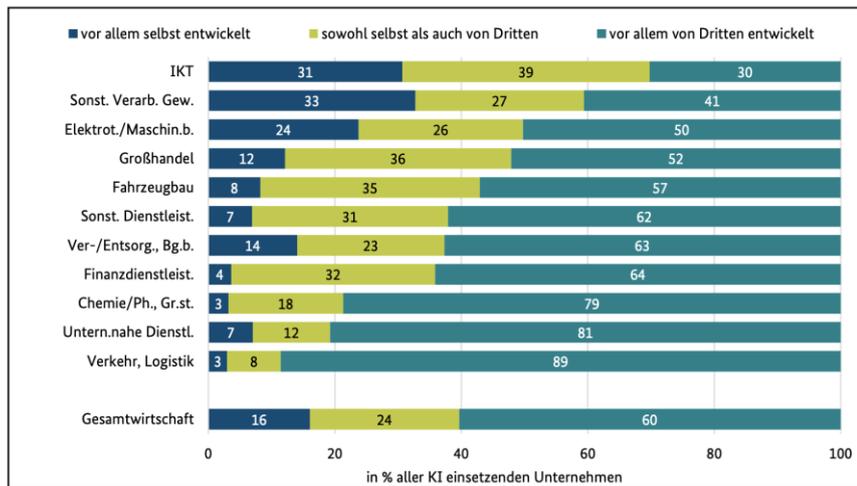


Abbildung 15: Art der Nutzung von KI<sup>56</sup>

Je nach gewünschtem Anwendungsfall kann es Probleme mit der Kompatibilität zwischen Frameworks oder Programmiersprachen geben. Beispielsweise ist das Frontend von yeet in Typescript geschrieben und daher werden Typendefinitionen für verwendete Bibliotheken und Frameworks benötigt. Diese kommen entweder direkt aus der Bibliothek, aus einem zweiten Package oder aus der eigenen erstellten Typendefinition.

## 5.5 Prototypische Verwendung eines ML Modells

In diesem Kapitel wird aufgezeigt wie die spezifische Implementierung von einem ML Modell in eine Webanwendung erfolgen kann. Dazu wird ein Modell von Tensorflow (siehe Kap. 4.1) verwendet. Bei diesem Beispiel wird also kein neues Modell erstellt, sondern ein Vorgefertigtes aus einer externen Quelle genutzt. Die Berechnung des Modells wird lokal im Browser und nicht von einem zusätzlichen Server durchgeführt. Dazu wird das Modell in eine Chat Komponente eingebaut, um eine Sentimentanalyse für die eingegebene Nachricht durchzuführen und Nachrichten zu blockieren, welche beispielsweise beleidigende oder rassistische Wörter enthalten. Dies könnte im Supportbereich für die Kundenkommunikation zum Einsatz kommen. Das verwendete Modell gehört zu dem Natural Language Processing Bereich (siehe Kap. 2.3.6).

<sup>56</sup> (Bundesministerium für Wirtschaft und Energie, 2020)

Das Modell ist unter der Apache 2.0 Lizenz veröffentlicht, daher ist die Nutzung und eine mögliche Weiterentwicklung des Modells rechtlich zugelassen.<sup>57</sup> In Abb. 16 ist der Aufbau der Chat Komponente dargestellt. Es gibt drei Bestandteile. Oben werden Details über den Chatpartner dargestellt. In der Mitte werden alle Nachrichten angezeigt. Unten gibt es eine Eingabemöglichkeit für neue Nachrichten.

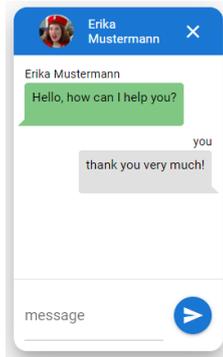


Abbildung 16: Aufbau der Chat Komponente

Der grundlegende Ablauf bei der Klassifizierung einer Nachricht soll wie folgt aussehen:

1. User wird mit einer Nachricht begrüßt, sobald er den Chat öffnet.
2. User gibt eine Nachricht ein
3. Eingabe wird als Input für das Modell verwendet, um die Eingabe zu klassifizieren

Angenommen die Eingabe des Anwenders ist gegenüber dem Mitarbeiter rassistisch oder beleidigend, so wird das Versenden einer solchen Nachricht blockiert und der Kunde kann auf die Problemstellung hingewiesen werden.

Für die Umsetzung wird zunächst der strukturelle HTML Aufbau erstellt (siehe Anhang Kap. 8.1). Für die Eingabemöglichkeit wird die „input“ Komponente von Quasar verwendet. Sobald sich die Eingabe verändert, wird eine Funktion aufgerufen, die als Parameter die eingegebene Nachricht erhält. An dieser Stelle soll nun die Klassifizierung der Eingabe durch das Modell von Tensorflow erfolgen. Um das Modell nutzen zu

---

<sup>57</sup> (Vgl. Tensorflow, kein Datum)

können, muss zunächst Tensorflow importiert werden (vgl. Abb. 17). Nach der Installation von Tensorflow kann der Import wie folgt erfolgen:

```
//Tensorflow
import * as tf from '@tensorflow/tfjs';
```

Abbildung 17: Import von Tensorflow

```
const getModel = async () => {
  model.value = await tf.loadLayersModel(modelUrl.value);
};
```

Abbildung 18: Laden des Modells aus dem Internet

Das zu nutzende Modell, sowie die dazugehörigen Metadaten, werden in diesem Beispiel aus dem Internet beim Start der Anwendung, also beim Aufruf der Internetseite geladen (Abb. 19). Die Metadaten bieten zusätzliche Informationen über das Modell wie z.B. die Anzahl an genutzten Trainingsdaten, die maximale Länge einer Eingabe oder die Anzahl der Epochen. In der folgenden Abb. 20 ist die Struktur der zu diesem Modell gehörigen Metadaten abgebildet.

```
interface TextMetadata {
  batch_size: number;
  embedding_size: number;
  epochs: number;
  index_from: number;
  max_len: number;
  model_type: string;
  vocabulary_size: 20000;
  word_index: { [key: string]: number };
}
```

Abbildung 19: Struktur der Metadaten

Ein wichtiger Inhalt der Metadaten ist die Eigenschaft „word\_index“. Dieses Objekt enthält eine Zuordnung von den Wörtern, die dem Modell bekannt sind, zu einer Nummer.

Da das von Tensorflow importierte Modell als Eingabe nur Tensoren verarbeiten kann, die numerische Werte enthalten, müssen die einzelnen Wörter aus der Eingabe zuerst mit Hilfe der Metadaten der zugehörigen Nummer zugeordnet werden. Dafür wird zunächst jede Eingabe formatiert. Dabei werden z.B. alle Buchstaben in Kleinbuchstaben umgewandelt. In der folgenden Darstellung (Abb. 20) sieht man die Formatierung der Eingabe, welche in der Funktion durchgeführt wird, die bei jeder Änderung der Eingabe aufgerufen wird.



formatierte Liste übergeben bekommt. Mithilfe der Funktion `tensor2d()` wird ein Tensor erstellt welcher dann der `predict()` Funktion übergeben werden kann. Das Ergebnis daraus wird in der Variable `score` gespeichert.

```
const toxicScan = (wordList: number[]) => {  
  if (metadata.value && model.value) {  
    const inputTensor = tf.tensor2d(wordList, [1, metadata.value.max_len])  
  };  
  let prediction = model.value.predict(inputTensor) as tf.Tensor;  
  let score = prediction.dataSync()[0] * 100; //performance?  
  if (score < 50) warning.value = 'toxic';  
  else warning.value = '';  
}  
};
```

Abbildung 24: Verarbeitung der Eingabe durch das Modell

Das Endergebnis ist eine Fließkommazahl zwischen null und eins, je näher die Zahl an der eins ist, desto höher ist die Wahrscheinlichkeit, dass der Satz keine negativen Wörter enthält.

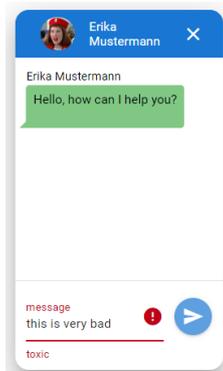


Abbildung 25: Blockierung eine "unerlaubten" Nachricht

Die Funktionalität ließe sich durch eine genauere Klassifizierung, um welche Art von Negativität es sich handelt, erweitern.

In diesem gezeigten Beispiel läuft das Modell direkt im Browser und somit lokal auf dem Client. Es ist daher nicht notwendig neue Rechenleistung in der eigenen IT-Infrastruktur zu schaffen. Die Berechnung läuft in diesem Beispiel zwar schnell und es entstehen praktisch keine relevanten Ladezeiten, dies kann jedoch bei anderen Modellen nicht der Falls sein. Daher muss für jedes Szenario individuell geprüft werden, ob die Berechnung auf dem Client möglich und sinnvoll ist, um den oben genannten Anforderungen an die Ladezeit gerecht zu werden.

## **5.6 Weitere Szenarien für Implementierungsmöglichkeiten**

Neben dem zuvor beschriebenen Beispiel der Implementierung eines fertigen Modells über Tensorflow zur Texterkennung, bestehen viele weitere Implementierungsmöglichkeiten für unterschiedliche Verwendungszwecke.

Die im vorherigen Kapitel verwendete Methode, fertige Modelle von Tensorflow einzubinden, beschränkt sich nicht nur auf die Erkennung von „verbotenen“ Wörtern, sondern kann z.B. auch für Gestensteuerung oder für weitere NLP Funktionen genutzt werden. Auch bietet Tensorflow Modelle für die automatisierte Antwort auf Fragen.

### **5.6.1 Kundenkommunikation**

Modelle aus dem Natural Language Processing Bereich, dem auch das vorgestellte Modell in Kapitel 5.5 zugeordnet wird, können genutzt werden, um Kundenanfragen zu filtern, Anfragen zu klassifizieren oder um aus dem Kontext heraus automatisch passende Antworten zu finden. Kundenanfragen, die unfreundlich oder beleidigend sind, können automatisch herausgefiltert und redundante Anfragen können erkannt werden. So werden die Support-Mitarbeiter entlastet und die Qualität ihrer Arbeitsumgebung verbessert. Kundenanfragen über Ticket-Systeme, E-Mails, Foren, Chats und weitere Kanäle können schnell für eine große Menge an Daten sorgen, KI-Systeme können die Aufbereitung und Strukturierung dieser Daten und Anfragen optimieren und effizienter machen, um im Anschluss zusätzlich die automatische Weiterleitung an den richtigen Mitarbeiter zu übernehmen. Tensorflow bietet verschiedene vorgefertigte Modelle an. Wie schon in Kapitel 2.3.5 erwähnt, besteht zudem die Möglichkeit, Modelle aus anderen Quellen zu nutzen.

Für die Verbreitung der Low-Code Anwendung und das Erschließen neuer Märkte durch Expansion in weitere Länder kann eine Übersetzung in die jeweilige Landessprache wichtig sein. Die Übersetzung erfolgt entweder manuell, automatisch oder in einer Mischung, indem vorerst automatisiert übersetzt und dies in einem weiteren Schritt manuell geprüft und angepasst wird. Da eine manuelle Übersetzung sehr aufwendig ist, kann eine externe Lösung dafür verwendet werden. Für eine automatische Übersetzung bieten sich Schnittstellen zu Service Anbietern an, wie z.B. die in Kapitel 3.4 vorgestellte

Translation-API von Google. So könnte beispielsweise jeder Anwender eine automatisch generierte Übersetzung von seiner Anwendung anfertigen lassen.

### **5.6.2 Projektverwaltung**

Es bestehen weitere Implementierungsmöglichkeiten für KI in anderen Prozessen oder für andere Problemstellungen, die eine höhere Komplexität aufweisen und dementsprechend eine aufwendigere Planung, Konzeptions- und Umsetzungsphase benötigen als die oben gezeigten Beispiele. Dabei bleiben jedoch die Möglichkeiten der technischen Implementierung bestehen. Neben der Nutzung von Bilderkennung wäre die Importierung von bestehenden Konzept16 Projekten in die neue Webanwendung ein möglicher Anwendungsfall für KI.

Die bestehende Entwicklungsumgebung Konzept16 arbeitet bei dem Aufbau eines Layouts mit der Positionierung anhand von x und y Werten. Im Gegensatz dazu wird das Layout bei yeet durch sogenannte Flexboxen erstellt. Beim Flexbox Layout wird keine feste Position für ein Element anhand von Koordinaten bestimmt, das Element wächst je nach verfügbarem Platz und Inhalt.

Das bedeutet, dass Kunden keine einfache Lösung zur Verfügung steht, ein fertiges Layout aus der Konzept16 Anwendung in der neuen Anwendung yeet zu nutzen.

Dies kann für das Ziel, Bestandskunden den Einstieg in die neue Applikation möglichst einfach zu gestalten, hinderlich sein. Neben der Möglichkeit jedes Layout manuell im neuen System nachzubauen, könnte dies mithilfe eines ML Modells möglicherweise auch automatisiert passieren. Neben der Nutzung der Layoutdaten in dem gleichen Format wie diese auch gespeichert werden, wäre eine weitere Möglichkeit Bilder als Trainingsdaten zu nutzen. Dabei könnten Screenshots von einem Layout erstellt werden und diese dann als Eingabe für das Modell verwendet werden.

Ein weiterer Verwendungsmöglichkeit ist die automatische Klassifizierung von Bildern. Die Bilderkennung könnte z.B. für die, vom Anwender hochgeladenen, Bilder genutzt werden.

### 5.6.3 Bedienung/Steuerung der Anwendung

Durch den Einsatz einer Spracherkennung kann die Bedienung einer Webanwendung und so auch die der Low-Code Entwicklungsplattform zusätzlich erweitert und möglicherweise vereinfacht werden. Der Einsatz im Designer, um den Nutzer beim Erstellen eines Layouts zu unterstützen, könnte mehrere Vorteile haben, wie z.B. das Steigern der Geschwindigkeit bei der Erstellung und das Vereinfachen von bestimmten Prozessen. Zudem könnte körperlich eingeschränkten Anwendern die Möglichkeit gegeben werden, die Anwendung uneingeschränkt nutzen zu können. Eine KI-gestützte Sprachsteuerung könnte auch für eine Suchfunktion verwendet werden oder in der Unterstützung bei der generellen Bedienung der Anwendung für den Nutzer. Desweiteren könnte eine Sprachsteuerung den Nutzer bei Prozessen wie z.B. dem Hinzufügen und Entfernen von Komponenten, der Navigation in der Anwendung und bei der Suche nach Funktionen unterstützen. Aus der folgenden Statistik (Abb. 23) ergibt sich, dass die Nutzung digitaler Sprachassistenten steigt.

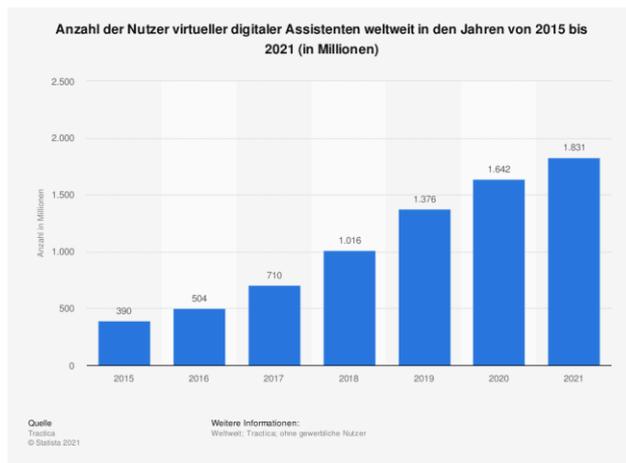


Abbildung 26: Einsatz von Sprachassistenten<sup>58</sup>

Eine Sprachsteuerung muss die Eingabe schnell verarbeiten können, über hohe Genauigkeit bei der Vorhersage verfügen und optimalerweise auch mehrere Sprachen verarbeiten können. Außerdem kann es notwendig sein, bestimmte Schlüsselwörter aus

<sup>58</sup> (Tractica, 2016)

gesprochenen Sätzen herauszufiltern. Für die Implementierung kann man bereits existierende Modelle, die auf die Verarbeitung von Spracheingaben ausgerichtet sind, um zusätzliche Sprachbefehle erweitern. Ein Vorteil dieser Variante besteht in der schon vorhandenen Erkennung von Stör- oder Hintergrundgeräuschen. Dies würde bei einer vollständigen eigenen Entwicklung zu einem Mehraufwand führen.

Eine Sprach- oder Texterkennung kann außerdem genutzt werden, um neue Programmlogik in eine Anwendung einzufügen. Durch die Eingabe von Befehlen oder Abläufen und die nachfolgende Verarbeitung können dem Anwender Vorschläge von möglichen Funktionen unterbreitet werden. Ein Beispiel für eine ähnliche Lösung ist in der von Microsoft entwickelten Low-Code Entwicklungsumgebung Power-Apps zu sehen. Dort wird das ML Modell GPT-3 genutzt, um Funktionen z.B. durch folgende Befehle einzufügen:

„Finde Produkte, deren Name mit ‚Kinder‘ beginnt.“<sup>59</sup> Diese Eingabe kann dann in richtige Programmlogik automatisch umgewandelt werden.

Im Anschluss einer Implementierung könnte das System beispielsweise durch Usertests evaluiert und getestet werden.

Anstatt die Berechnung direkt im Browser und somit auf dem Client auszuführen, könnte man auch eine Schnittstelle eines externen Anbieters nutzen, die man z.B. über einen http Request anfragen und ein Ergebnis von dem eigenen Server oder z.B. von einem der in Kap. 3.4 erwähnten externen Cloud Dienste erhalten könnte. Dies würde allerdings bedeuten, dass man die Eingaben des Nutzers an z.B. Anbieter wie Google senden müsste. Dies kann, wie in Kap. 3.4 beschrieben, zu einem erhöhten rechtlichen und administrativen Aufwand führen durch Datenschutzvorgaben von bspw. der DSGVO. Außerdem verliert man möglicherweise einen Teil der Datensouveränität. Da Microsoft sowie Google auch Anbieter von Low-Code Plattformen sind, sollte genau analysiert werden, welche Daten man diesen Anbietern, welche zugleich Marktbegleiter sind, zur Verfügung stellt.

---

<sup>59</sup> (Vgl. Petereit, 2021)

Auch jetzt wird in der Entwicklung schon KI eingesetzt. Ein naheliegendes Beispiel dafür ist die Erweiterung „Visual Studio IntelliCode“ für den Code Editor Visual Studio Code. Diese Erweiterung nutzt Machine Learning, um passende Vorschläge oder Autovervollständigungen zu liefern.

### **5.7 Unterstützung im Design**

Künstliche Intelligenz kann auch bei der Erstellung eines Designs unterstützen. Durch Analyse der Farben des Logos beispielsweise und durch die Beschreibung des Inhalts können Layout- und Designvorschläge generiert werden.

### **5.8 Entwicklung von Schnittstellen für Anwender**

Die oben genannten Implementierungsmöglichkeiten haben direkten Einfluss auf die Funktionsweise der Low-Code Entwicklungsumgebung oder betreffen interne Prozesse des Unternehmens. Neben diesen Einsatzmöglichkeiten besteht auch die Möglichkeit, Schnittstellen für die Anwender der Low-Code Entwicklungsumgebung zu entwickeln und anzubieten, sodass die entwickelten Anwendungen der Kunden direkt ML Techniken nutzen können.

Um passende und auf die Bedürfnisse ausgerichtete Schnittstellen anzubieten, kann zunächst untersucht werden, welche Art von Software mit der Low-Code Entwicklungsumgebung oft oder hauptsächlich entwickelt wird. Da sich die neue webbasierte Entwicklungsumgebung noch in der Entwicklung befindet, können bestehende Umsetzungen mit der Conzept16 Entwicklungsumgebung als Ausgangspunkt für die Orientierung dienen.

In Softwarelösungen, die im Bereich der Wartung und Instandhaltung eingesetzt werden, können ML Modelle genutzt werden, um auf der Basis von historischen Daten den optimalen Zeitpunkt für eine Wartung z.B. von Maschinen zu berechnen. Dadurch bietet sich auch die Möglichkeit, mögliche Spitzen bei anstehenden Wartungen zu vermeiden, um eine gleichmäßige Auslastung zu erzielen. Daraus kann eine Optimierung und somit eine bessere Ressourcen- und Personalplanung entstehen. Durch automatische Erkennung von Auffälligkeiten in großen Datenmengen ergeben sich auch Anwendungsfälle in der Betrugserkennung.

Durch die Analyse vorhandener gesammelter Daten, können weitere konkrete Einsatzmöglichkeiten erkannt werden.

Entwickelte und angebotene Schnittstellen unterliegen neben den in Kap. 3 erwähnten Anforderungen noch weiteren. Die Schnittstellen müssen für Anwender einfach zugänglich und nutzbar sein. Um rechtliche Anforderungen zu erfüllen, sollte eine Transparenz der genutzten Technologien sichergestellt werden, gerade wenn die Daten des Anwenders zu weiteren externen Dienstleistern weitergeleitet wird.

Durch die Einbindung von KI in die erstellten Softwarelösungen können somit nicht nur direkt Vorteile für den Anwender der Low-Code Entwicklungsumgebung entstehen, sondern auch für die Endanwender.

Durch die Integration von Schnittstellen für die Textverarbeitung oder Audioverarbeitung, lassen sich die Eingabemöglichkeiten, die der Entwickler in seine Anwendung einbauen kann, erweitern.

## 6 FAZIT

Die Ausführungen zeigen, wie groß das Einsatzgebiet von KI Systemen generell ist und wie viele Möglichkeiten sich bieten, ein KI-System in eine Low-Code Entwicklungsumgebung zu implementieren. ML Modelle, die z.B. der Bilderkennung, Sprachverarbeitung, Textanalyse oder Datenklassifizierung dienen, bieten eine Vielzahl von Implementierungsmöglichkeiten, die dabei nicht nur für Vorteile für das eigentliche Produkt, sondern auch für Verbesserungen bei internen Prozessen eines Unternehmens sorgen können. Die Entwicklung von Schnittstellen zu ML Funktionen für den Anwender einer Low-Code Entwicklungsplattform kann den Funktionsumfang des Produktes erheblich erweitern.

Neben den vielen Vorteilen, die eine Implementierung von künstlicher Intelligenz bieten kann, entstehen Risiken und neue Anforderungen an den gesamten Entwicklungsprozess. Auch bringt der Einsatz von KI nicht immer eine Verbesserung und ist somit nicht immer generell sinnvoll, daher müssen individuell für jeden Einzelfall die Ergebnisse evaluiert werden.

Neben den externen Cloud Lösungen besteht auch die Möglichkeit, eigene ML Modelle zu entwickeln. Die Entwicklung eigener Modelle kann sehr kostenaufwendig und zeitintensiv sein, hat aber durch die Datenkontrolle und Passgenauigkeit große Vorteile.

Eine Studie von Deloitte zeigt, dass sich KI-Projekte im Durchschnitt schon nach zwei Jahren amortisieren<sup>60</sup>. Zusätzlich dazu zeigt eine von IDG Research Services, durchgeführte Studie, dass viele Unternehmen schon nach drei Monaten erste positive Effekte eingesetzter Machin-Learning-Lösungen verzeichnen können. Laut dieser Untersuchung können gerade Großunternehmen auch direkt nach Implementierung einer ML Lösung eine Verbesserung feststellen.<sup>61</sup> Detailliertere Daten zu der Erfolgsmessung aus der Studie von IDG Research Services werden in der unten stehenden Abb. 28 aufgezeigt.

---

<sup>60</sup> (Deloitte, kein Datum)

<sup>61</sup> (Vgl. Hill, 2020)

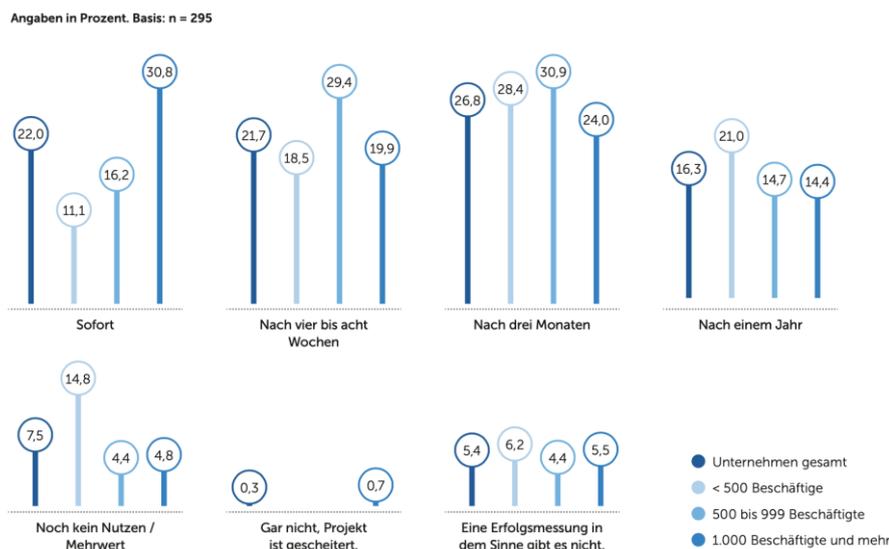


Abbildung 27: Erfolgsmessung, ab wann Nutzen aus KI Projekten gezogen wird<sup>62</sup>

Da es zu Schwierigkeiten bei der Kalkulation kommen kann, ob sich die Entwicklung oder Implementierung eines KI-Systems wirtschaftlich rentiert, können die vorher bekannten Preise von externen Cloud Anbietern eine Arbeitserleichterung darstellen. Die Nutzung von KI as a Service ist von der technischen Seite einfacher als die vollständige Eigenentwicklung, kann aber zu einem nochmal größeren Aufwand durch die Datenschutzanforderungen führen und zu einer Abhängigkeit zu dem jeweiligen Anbieter.

Durch die vielen unterschiedlichen Möglichkeiten der Implementierungs- und Nutzungsart ist es schwierig, die beste Methode für den jeweiligen Einsatz zu finden. Durch den sehr frühen, aber schnell voranschreitenden Entwicklungsstand fehlt es auch an Daten, die beispielsweise für genauere Analysen des Nutzerverhaltens verwendet werden könnten.

Für die Ent- und Weiterentwicklung ist es notwendig, standardisierte Prozesse zu entwickeln, um den Qualitätsanforderungen gerecht zu werden. Durch die Notwendigkeit Daten zu sammeln, zu verarbeiten und zu analysieren, verstärken sich die Anforderungen an den Datenschutz und der dadurch resultierende rechtliche und administrative Aufwand muss berücksichtigt werden.

<sup>62</sup> (Hill, 2020)

Daher ist es wichtig, nach neuen Wettbewerbsvorteilen und Marktfeldern Ausschau zu halten. Potenzial für einen Wettbewerbsvorteil und Umsatzsteigerung durch den Einsatz von KI ist groß. Da sich yeet allerdings noch in einer frühen Entwicklungsphase befindet, ist es möglicherweise noch zu früh, um mit einer konkreten Umsetzung eines KI-Projektes zu beginnen, aber mit voranschreitendem Entwicklungsfortschritt ergeben sich auch bessere Möglichkeiten, um mit einer konkreten Planung zu beginnen. Bei Entwicklung oder Implementierung einer KI ist es wichtig die grundlegenden Anforderungen zu erfüllen, um ein robustes System zu schaffen. Für die Akzeptanz, auch vom eigenen Team, ist eine große Transparenz im gesamten Prozess von Vorteil.

## 7 AUSBLICK

„Künstliche Intelligenz gilt in der deutschen Wirtschaft als Zukunftstechnologie und immer mehr Unternehmen sehen die Technologie als eine Chance für das eigene Geschäft.“<sup>63</sup>

Künstliche Intelligenz und dem verbundenen Bereich des Machine Learnings, sind Bereiche, die zurzeit und in Zukunft stark in Bewegung sind und in denen viel Forschung betrieben wird.

Durch schnell wachsende Datenquellen und Datenbestände und dadurch, dass man immer mehr Rechenleistung zur Verfügung hat, stehen immer neue Anwendungsfelder für KI zur Verfügung, so dass bessere Ergebnisse erzielt werden können.<sup>64</sup>

Es ergeben sich große Chancen für einen Produktivitäts- und Wachstumsschub<sup>65</sup>, laut einer Untersuchung von PwC liegt der größte Vorteil bei dem Einsatz von KI momentan in der Prozessautomatisierung und der Unterstützung von Mitarbeitern bei der Entscheidungsfindung. Jedoch werden zukünftig auch qualitativ hochwertige, personalisierte und datengetriebene Produkte KI als Basis nutzen.<sup>66</sup>

Der benötigte Zeitaufwand für die Einbindung und Nutzung von externen KI-Lösungen wird durch die steigende Anzahl an externen Anbietern geringer, aber auch die eigene Entwicklung wird durch ein großes Angebot an Open Source Bibliotheken sowie Tools immer einfacher gestaltet.

---

<sup>63</sup> (Industrial AI, kein Datum)

<sup>64</sup> (Buxmann & Schmidt)

<sup>65</sup> (Buxmann & Schmidt)

<sup>66</sup> (Vgl. PwC, 2019)

Durch die Kombination mit dem voraussichtliche zukünftigen Anstieg bei dem Umsatz und Nutzung von Low-Code Plattformen, ist dieser Bereich für KI Projekte sehr interessant.

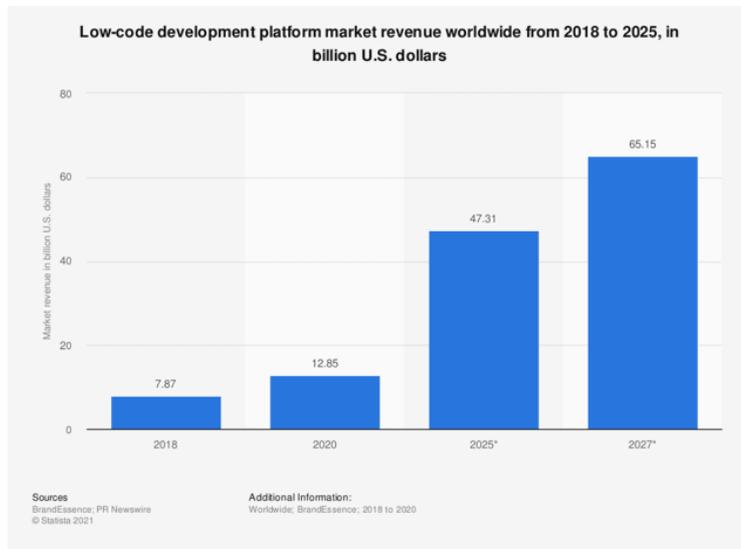


Abbildung 28: Prognostiziertes Marktumsatz Weltweit bis 2027<sup>67</sup>

Bei der Planung einer Integration von KI-Lösungen sollte auch die Zahl, der an diesem Standort verfügbaren Fachkräfte, bekannt sein. In der folgenden Abb. 29 sieht man die Anzahl an offenen Stellen von Unternehmen in Deutschland für den KI Bereich.

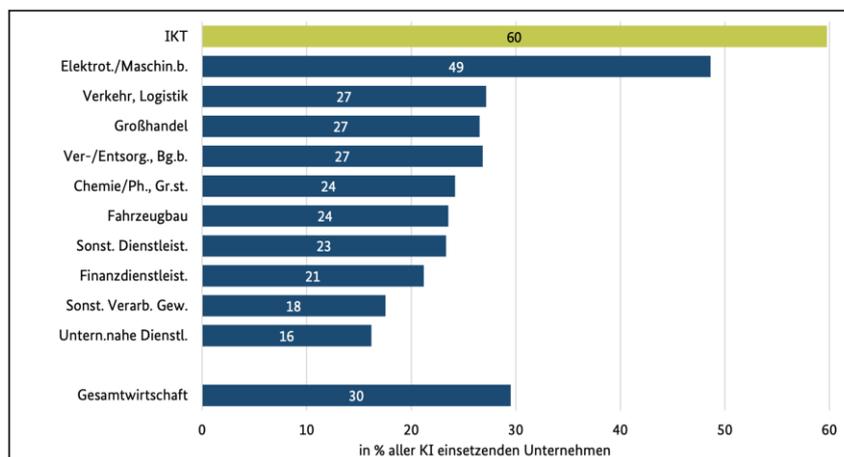


Abbildung 29: Offene Stellen von deutschen Unternehmen im Bereich KI<sup>68</sup>

<sup>67</sup> (Statista, 2021)

<sup>68</sup> (Bundesministerium für Wirtschaft und Energie, 2020)

Da für die Umsetzung von IT-Projekten oft immer weniger Zeit zur Verfügung steht und bestehende Anwendungen häufigen Änderungen, z.B. durch neue Anforderungen, unterliegen können, bietet sich die Nutzung von Low-Code Plattformen an. Durch die zusätzliche Unterstützung von KI-Lösungen kann die Entwicklungszeit weiter reduziert werden und für eine Reduzierung der Komplexität sorgen. Das sorgt für eine breitere Anwenderschaft, da auch Nicht-Informatiker leichter Projekte umsetzen können.

---

## 8 LITERATURVERZEICHNIS

- 1&1 IONOS. (27. Juli 2020). *Digital Guide IONOS*. Abgerufen am 30. Mai 2021 von <https://www.ionos.de/digitalguide/server/knowhow/edge-computing-erklaerung-und-definition/>
- 1&1 IONOS. (20. Mai 2021). *Digital Guide*. Abgerufen am 30. Mai 2021 von <https://www.ionos.de/digitalguide/server/knowhow/iaas-infrastructure-as-a-service/>
- Becker, R. (12. Februar 2019). *JAAI*. Abgerufen am 30. Mai 2021 von <https://jaai.de/transfer-learning-1739/>
- Bundesministerium für Bildung und Forschung. (kein Datum). Abgerufen am 30. Mai 2021 von <https://www.bmbf.de/de/kuenstliche-intelligenz-5965.html>
- Bundesministerium für Bildung und Forschung. (November 2018). Abgerufen am 30. Mai 2021 von [https://www.bmbf.de/files/Nationale\\_KI-Strategie.pdf](https://www.bmbf.de/files/Nationale_KI-Strategie.pdf)
- Bundesministerium für Bildung und Forschung. (kein Datum). *KI-Strategie-Deutschland*. Abgerufen am 30. Mai 2021 von <https://www.ki-strategie-deutschland.de/home.html>
- Bundesministerium für Wirtschaft und Energie. (März 2020). Abgerufen am 30. Mai 2021 von [https://www.bmwi.de/Redaktion/DE/Publikationen/Wirtschaft/einsatz-von-ki-deutsche-wirtschaft.pdf?\\_\\_blob=publicationFile&v=8](https://www.bmwi.de/Redaktion/DE/Publikationen/Wirtschaft/einsatz-von-ki-deutsche-wirtschaft.pdf?__blob=publicationFile&v=8)
- Buxmann, P., & Schmidt, H. (kein Datum). *Künstliche Intelligenz Mit Algorithmen zum wirtschaftlichen Erfolg*. Springer Gabler.
- Cognizant. (kein Datum). *cognizant.com*. Abgerufen am 13. Mai 2021 von <https://www.cognizant.com/glossary/human-centered-ai>
- Copeland, B. (kein Datum). *Britannica*. Abgerufen am 29. Mai 2021 von <https://www.britannica.com/technology/artificial-intelligence>
- Deloitte. (kein Datum). Abgerufen am 30. Mai 2021 von <https://www2.deloitte.com/de/de/pages/technology-media-and-telecommunications/articles/ki-studie-2020.html>
- Ertel, W. (2016). *Grundkurs Künstliche Intelligenz*. Springer Vieweg.
- Fowler, M. (19. September 2019). *martinFowler*. Von <https://martinfowler.com/articles/cd4ml.html> abgerufen

- 
- Geißler, O., & Litzel, N. (29. Januar 2018). *BIGDATA INSIDER*. Von <https://www.bigdata-insider.de/so-funktioniert-google-tensorflow-a-669777/> abgerufen
- Gerard, C. (2020). *Practical Machine Learning in JavaScript: TensorFlow.js for Web Developers*.
- Git. (kein Datum). Abgerufen am 23. Januar 2021 von <https://git-scm.com/about>
- Git. (kein Datum). Abgerufen am 23. Januar 2021 von <https://git-scm.com/book/de/v2/Erste-Schritte-Was-ist-Versionsverwaltung%3F>
- Google. (kein Datum). Abgerufen am 30. Mai 2021 von <https://cloud.google.com/natural-language/pricing>
- Google. (kein Datum). *Google Cloud*. Abgerufen am 30. Mai 2021 von <https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning>
- Google. (kein Datum). *TensorFlow*. Abgerufen am 30. Mai 2021 von <https://www.tensorflow.org/tfx/guide/tfdv>
- Hempleton. (13. Mai 2021). *Hampleton*. Von <https://www.hampletonpartners.com/de/news/newsdetail/announce/cloud-oder-edge-wo-liegt-die-zukunft-der-ki/> abgerufen
- Hill, J. (2020).
- Industrial AI*. (kein Datum). Abgerufen am 30. Mai 2021 von <https://www.ind-ai.net/zahlenfutter/kuenstliche-intelligenz-kommt-voran/>
- Jannis, S. (Februar 2021). *Udemy*. Abgerufen am 30. Mai 2021 von <https://www.udemy.com/course/deep-learning-und-ai/learn/lecture/11524292?start=0#overview>
- Kukat, F., & Human, S. (22. Juni 2020). *Industry OF Things*. Abgerufen am 12. Mai 2021 von <https://www.industry-of-things.de/low-code-vs-no-code-gemeinsamkeiten-unterschiede-und-einsatzszenarien-a-937195/>
- Kunze, S. (3. März 2021). *elektrotechnik AUTOMATISIERUNG*. Abgerufen am 30. Mai 2021 von <https://www.elektrotechnik.vogel.de/wann-sich-deep-learning-in-der-produktion-lohnen-kann-a-1003300/#:~:text=In%20der%20Produktion%20gibt%20es,kann%20Deep%20L earning%20sinnvoll%20sein>

- 
- LeCun, Y., Cortes, C., & Burges, C. (kein Datum). *THE MNIST DATABASE*. Von <http://yann.lecun.com/exdb/mnist/> abgerufen
- Luber, S., & Litzel, N. (01. September 2016). *BIGDATA INSIDER*. Abgerufen am 29. Mai 2021 von <https://www.bigdata-insider.de/was-ist-machine-learning-a-592092/#:~:text=Machine%20Learning%20ist%20ein%20Teilbereich,k%C3%BCnstliches%20Wissen%20aus%20Erfahrungen%20generiert.>
- Luber, S., & Litzel, N. (6. August 2019). *BIGDATA INSIDER*. Abgerufen am 30. Mai 2021 von <https://www.bigdata-insider.de/was-ist-onnx-open-neural-network-exchange-a-851510/>
- Mayerhofer, J. (14. 02 2021). *empatic-ux.com*. Abgerufen am 23. Januar 2021 von <https://empatic-ux.com/de/was-ist-ux/#:~:text=User%20Experience%20Design%20oder%20UX,Interaktion%20mit%20einem%20digitalen%20Produkt>
- Microsoft. (kein Datum). *github*. Abgerufen am 13. Mai 2021 von <https://github.com/microsoft/TypeScript>
- Mozilla Developer Network*. (16. September 2020). Abgerufen am 23. 01 2021 von <https://developer.mozilla.org/de/docs/Web/JavaScript>
- Mundt, E. (30. Januar 2019). *industry-of-things.de*. Abgerufen am 23. Januar 2021 von <https://www.industry-of-things.de/was-ist-low-code-definition-anwendung-und-beispiele-a-794509/>
- Ng, A. (kein Datum). *slideshare*. Abgerufen am 30. Mai 2021 von <https://www.slideshare.net/ExtractConf>
- Peham, T. (15. November 2019). *Computerwelt*. Abgerufen am 23. Januar 2021 von <https://computerwelt.at/news/gartner-bis-2024-werden-65-aller-unternehmen-mit-low-code-entwickeln/>
- Petereit, D. (25. Mai 2021). *t3n*. Abgerufen am 30. Mai 2021 von <https://t3n.de/news/entwickeln-fuer-jedermann-gpt-3-1380692/>
- PwC. (30. Januar 2019). *PwC Deutschland*. Abgerufen am 30. Mai 2021 von <https://www.pwc.de/de/digitale-transformation/kuenstliche-intelligenz/sechs-wichtige-prioritaeten-um-ki-2019-voranzubringen.html>

- 
- Rottstedt, H., & Human, S. (kein Datum). *Industry of Things*. Abgerufen am 30. Mai 2021 von <https://www.industry-of-things.de/cpiaas-der-as-a-service-trend-verstaerkt-sich-a-858659/>
- SAS. (13. Mai 2021). *sas blog*. Von <https://blogs.sas.com/content/sasdach/2019/04/08/bessere-datenqualität-dafur-gibts-7-analytische-methoden/> abgerufen
- Sass. (kein Datum). Abgerufen am 25. Januar 2021 von <https://sass-lang.com/guide>
- Schnellbacher, M. (28. April 2016). *entwickler.de*. Abgerufen am 13. Mai 2021 von <https://entwickler.de/online/tools/openai-gym-open-source-243685.html>
- selfhtml wiki*. (27. 12. 2020). Abgerufen am 23. Januar 2021 von <https://wiki.selfhtml.org/wiki/CSS>
- selfhtml wiki*. (18. Januar 2021). Abgerufen am 23. 01. 2021 von <https://wiki.selfhtml.org/wiki/HTML>
- Shanhong Liu *Statista*. (11. Juni 2020). Abgerufen am 23. Januar 2021 von <https://www.statista.com/statistics/793843/worldwide-developer-survey-most-wanted-framework/>
- sree017. (kein Datum). *Cheatography*. Abgerufen am 13. Mai 2021 von <https://cheatography.com/sree017/cheat-sheets/nlp/>
- Statista. (März 2021). *statista*. Abgerufen am 13. Mai 2021 von <https://www.statista.com/statistics/1226179/low-code-development-platform-market-revenue-global/>
- Tensorflow. (kein Datum). *GitHub*. Abgerufen am 30. Mai 2021 von <https://github.com/tensorflow/tfjs-models/blob/master/LICENSE>
- Thamm, A., Gramlich, M., & Borek, A. (2020). *The Ultimate Data and AI Guide*. München: Data AI Press.
- Tractica. (August 2016). *statista*. Abgerufen am 30. Mai 2021 von <https://de.statista.com/statistik/daten/studie/620321/umfrage/nutzung-von-virtuellen-digitalen-assistenten-weltweit/#:~:text=F%C3%BCr%20das%20Jahr%202019%20wird,rund%201%2C4%20Milliarden%20prognostiziert.>

- Tractica. (September 2016). *Statista*. Abgerufen am 30. Mai 2021 von <https://de.statista.com/statistik/daten/studie/620443/umfrage/umsatz-mit-unternehmensanwendungen-im-bereich-kuenstliche-intelligenz-weltweit/>
- Vectorsoft AG. (kein Datum).
- Wuttke, L. (kein Datum). *datasolut*. Abgerufen am 13. Mai 2021 von <https://datasolut.com/einfuehrung-in-keras/>
- Wuttke, L. (kein Datum). *datasolut*. Abgerufen am 30. Mai 2021 von <https://datasolut.com/machine-learning-vs-deep-learning/>
- You, E. (kein Datum). *Github*. Abgerufen am 23. Januar 2021 von <https://github.com/yyx990803>
- Zalando. (kein Datum). *GitHub*. Abgerufen am 30. Mai 2021 von <https://github.com/zalandoresearch/fashion-mnist>

## 9 ANHANG

### HTML Quellcode Chatkomponente

```
<template>
  <div class="chat">
    <q-btn
      :icon="chatIsOpen ? 'close' : 'chat'"
      round
      color="primary"
      @click="chatIsOpen = !chatIsOpen"
    ></q-btn>
    <div v-if="chatIsOpen" class="chat-container shadow-19">
      <q-toolbar class="chat-container__toolbar">
        <q-item>
          <q-item-section side>
            <q-avatar>
              
            </q-avatar>
          </q-item-section>
          <q-item-section>
            <q-item-label>{{ name }}</q-item-label>
          </q-item-section>
        </q-item>
        <q-btn
          icon="close"
          round
          flat
          color="white"
          @click="chatIsOpen = !chatIsOpen"
        ></q-btn>
      </q-toolbar>
      <div class="chat-container__content">
        <q-chat-message
          v-for="(message, idx) in chatMessageList"
          :key="idx"
          :name="message.sent ? 'you' : name"
          :text="message.message"
          :sent="message.sent"
        />
      </div>
      <q-separator />
      <div class="chat-container__input-container">
        <q-input
          v-model="message"
          label="message"
          :error-message="warning"
        >
      </div>
    </div>
  </div>
</template>
```

```
:error="warning === '' ? false : true"
hide-bottom-space
@input="changeMessage"
>>template v-slot:after>
  <q-btn
    icon="send"
    round
    color="primary"
    :disable="warning === '' ? false : true"
    @click="sendMessage"
  >>/q-btn
  >>q-spinner-cube
    v-if="!metadataLoaded"
    size="md"
    color="primary"
  /> </template
>>/q-input>
</div>
</div>
</div>
</template>
```

---

## **Ehrenwörtliche Erklärung**

"Ich erkläre hiermit ehrenwörtlich, dass

1. ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe,
2. die aus fremden Quellen direkt oder indirekt übernommenen Gedanken als solche kenntlich gemacht sind,
3. dieses Exemplar mit der beurteilten Arbeit übereinstimmt und
4. diese Arbeit bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht wurde."

Worms, den

---