

# **Evaluierung, Analyse und Implementierung eines Machine Learning Verfahrens zur Verbesserung von fachlichen Entscheidungen**

**Bachelor-Thesis**

Fakultät Informatik



zur Erlangung des akademischen Grades

**Bachelor of Science**

vorgelegt von: Fabian Cotic

Studiengang: Wirtschaftsinformatik

Matrikelnummer: 288710

Erstgutachter: Prof. Dr. Christian Johner

Zweitgutachter: Dr. Bernd Michelberger



# Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht sind und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

---

Ort, Datum, Unterschrift



# Vorwort

Die vorliegende Arbeit bildet den Abschluss meines Bachelor-Studiums der Fachrichtung Wirtschaftsinformatik an der HTWG Konstanz. Diese Arbeit ist im Zeitraum vom 01. Juni 2017 bis 31. August 2017 unter der Betreuung von Herrn Prof. Dr. Christian Johner und Dr. Bernd Michelberger entstanden. Das Thema der Arbeit entstand einerseits aus meinem persönlichen Interesse an Machine Learning, sowie steigenden Kundenanfragen bezüglich Machine Learning im Vertrieb der ACTICO GmbH. Aufgrund meiner vorigen Tätigkeiten im Unternehmen, kam mir die Ehre zuteil mein Thema selbst zu wählen und Mithilfe meines Betreuers Dr. Bernd Michelberger auszugestalten.

Johner ...

Bernd ...

ACTICO ...

Korrekturleser ...

Zu Letzt möchte ich mich bei meinen Eltern bedanken, die mir während meines Studiums immer liebevoll mit Rat und Tat zur Seite standen und mich finanziell unterstützen um mir so mein Studium erst zu ermöglichen.



# Abstract

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.





# Abkürzungsverzeichnis

**BKM** Business Knowledge Model

**BPMN** Business Process Model and Notation

**CORBA** Common Object Request Broker Architecture

**DM** Decision Management

**DMN** Decision Model and Notation

**DMS** Decision Management Systems

**DS** Decision Service

**DRD** Decision Requirements Diagramm

**OMG** Object Management Group

**ML** Machine Learning

**UML** Unified Modeling Language



# Abbildungsverzeichnis

1.1. Gründe für falsch getroffene fachliche Entscheidungen. . . . .	19
1.2. Aufbau der Arbeit. . . . .	22
2.1. Der Lebenszyklus von automatisierten Entscheidungen. . . . .	25
2.2. Ein beispielhaftes DRD. . . . .	26
2.3. Beispiel einer Entscheidungstabelle zur Berechnung des Zinssatzes. . . . .	27
2.4. Vorgehensweise zur Entwicklung eines Modells. . . . .	30
2.5. Aufbau eines Perzeptrons. . . . .	32
2.6. Aufbau eines Perzeptrons. . . . .	33
2.7. Aufbau eines neuronalen Netzwerkes mit mehreren Schichten. . . . .	33
2.8. Neuronalesnetz mit Sigmoid-Neuronen und Bias. . . . .	35



# Tabellenverzeichnis

1.1. Forschungsfragen entlang der Kapitel. . . . .	21
2.1. Beispielhafter Aufbau der Trainingsdaten. . . . .	31



# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>9</b>
<b>Abbildungsverzeichnis</b>	<b>11</b>
<b>Tabellenverzeichnis</b>	<b>13</b>
<b>1. Motivation</b>	<b>17</b>
1.1. Problemstellung . . . . .	18
1.2. Zielsetzung . . . . .	20
1.3. Abgrenzung . . . . .	21
1.4. Aufbau der Arbeit . . . . .	22
<b>2. Grundlagen</b>	<b>25</b>
2.1. Decision Management . . . . .	25
2.2. Machine Learning . . . . .	29
2.3. Technologien . . . . .	34
2.3.1. Decision Management . . . . .	34
2.3.2. Machine Learning . . . . .	35
<b>3. Konzeption</b>	<b>37</b>
3.1. Definition von Anwendungsfällen . . . . .	37
3.1.1. Erkennung von Kreditausfällen . . . . .	37
3.1.2. Erkennung von Betrugsversuchen bei Banktransaktionen . . . . .	38
3.2. Identifikation relevanter Themenfelder . . . . .	38
3.2.1. Lerndaten . . . . .	38
3.2.2. Evaluationsdaten . . . . .	38
3.2.3. Decision-Engine . . . . .	39
3.2.4. Neuronales-Netzwerk . . . . .	39

<b>4. Implementierung</b>	<b>41</b>
4.1. Datenschema . . . . .	41
4.2. Decision-Engine . . . . .	41
4.3. Neuronales-Netzwerk . . . . .	42
4.4. Optimierung . . . . .	42
<b>5. Analyse</b>	<b>43</b>
5.1. Vorgehensweise . . . . .	43
5.2. Erkennung von Kreditausfällen . . . . .	43
5.3. Erkennung von Betrugsversuchen bei Banktransaktionen . . . . .	43
5.4. Zusammenfassung der Ergebnisse . . . . .	43
<b>6. Diskussion</b>	<b>45</b>
<b>7. Zusammenfassung und Ausblick</b>	<b>47</b>
<b>Literaturverzeichnis</b>	<b>49</b>
<b>A. Anhang</b>	<b>53</b>



# 1. Motivation

Unternehmen treffen unzählige Entscheidungen jeden Tag. Wertschöpfende Entscheidungen tragen, wie ihr Name schon sagt, zur Wertschöpfung bei und beeinflussen somit unmittelbar den wirtschaftlichen Erfolg eines Unternehmens. Eine Bank beispielsweise muss einen Kreditnehmer auf dessen Fähigkeit, Raten zurückzahlen zu können, prüfen. Somit soll sichergestellt werden, dass die Bank den Betrag des Kredites sowie auch die ihr zustehenden Zinsen zurückbekommt. Dadurch steht und fällt der Erfolg einer Bank mit den Krediten, die sie vergibt [7, vgl. S. 1]. Prominentestes Beispiel hierfür ist die ehemalige US-amerikanische Investmentbank Lehmann Brothers, die im Zuge der Finanzkrise 2008 Insolvenz anmelden musste [19, vgl. S. 3]. Boomende Immobilien-Preise resultierten in höheren Sicherheiten von Privathaushalten, wodurch auch Kredite an Darlehnsnehmer mit geringer Bonität vergeben wurden (Subprime-Kredit) [5, vgl. S. 82-83]. Nach dem Platzen der Immobilienblase kam es zu Zahlungsausfällen und -störungen im US-amerikanischen Hypothekenmarkt [1, vgl. S. 6 ff.], was erst in einem Finanzcrash und dann in einer Weltwirtschaftskrise resultierte [8, vgl. S. 109].

Etwa im Juni 2009 waren die schlimmsten Auswirkungen der Finanzkrise überstanden, die amerikanische Wirtschaft begann wieder zu wachsen [13, vgl. S. 21] und Banken vergaben fleißig Kredite. Heutzutage werben Banken mit Kreditentscheidungen innerhalb von wenigen Sekunden. Das Ausfüllen eines Formulars ist ausreichend, um wenige Sekunden später eine Kreditentscheidung zu erhalten. Die Prüfung der Bonität geschieht automatisiert im Hintergrund durch eine zuvor definierte Entscheidungslogik. Die Entscheidungslogik wird von Fach-Experten erstellt, wobei die Fach-Experten im Wesentlichen zwei essentielle Dinge beachten:

1. Verhinderung der Vergabe von Krediten an Personen, die diese nicht zurückzahlen können ("default rate")

2. Verhinderung der Nicht-Vergabe von Krediten an Personen, die eigentlich in der Lage sind, diese zurück zu zahlen, wodurch Einnahmen an Konkurrenten verloren gehen ("acceptance rate")

Banken sind somit angehalten eine gute Balance zwischen Sicherheit und Gewinn zu finden. Dazu muss die fachliche Entscheidungslogik drohende Zahlungsausfälle so genau wie möglich vorhersagen. Studien wie [26] zeigen, dass sieben Prozent der vergebenen Kredite im Euroraum mehr als 90 Tage im Zahlungsverzug sind. Das entspricht einem Volumen von einer Billionen Euro. Bei einem Zahlungsverzug von über 90 Tagen wird generell davon ausgegangen, dass ein Kredit nicht mehr in voller Höhe zurückgezahlt wird und somit zu einem Verlust-Geschäft führt [4, vgl. S. 8]. Könnte man die Entscheidungslogiken bei den Banken dahingehend optimieren, ein Prozent weniger "faule" Kredite zu vergeben, könnten große Summen eingespart werden, die sonst im Zuge einer Banken-Rettung dem Steuerzahler auferlegt werden.

## 1.1. Problemstellung

Das Einsparen der genannten Summen wäre erstrebenswert, allerdings gibt es unzählige Gründe, die den Finanzcrash ab 2007 verschuldet haben. Als Hauptgrund wird in der Literatur die umfangreiche Vergabe von Subprime-Krediten genannt [5, 19]. Bei der Vergabe dieser Kredite wurde das Risiko nicht korrekt identifiziert [19, vgl. S. 3], was zu unzähligen falschen fachlichen Entscheidungen führte. Daraus abgeleitet ergibt sich die Problemstellung dieser Arbeit, dass zu viele fachliche Entscheidungen falsch getroffen werden. Die Gründe hierfür werden nachfolgenden beschrieben und in Abbildung 1.1 zusammengefasst.

Ein Grund für die vielen falschen Entscheidungen, sind fehlerhafte Entscheidungslogiken und falsche Annahmen (Grund G1 - siehe Abbildung 1.1). Bei der Vergabe der Kredite wurden die Entscheidungen verfälscht, da die Entscheidungslogiken mit Eingabedaten befüllt wurden, die nicht der Realität entsprachen. Durch die unrealistisch hohen Immobilien-Preise, waren die Kredite mit Immobilien gesichert, die real weitaus weniger wert waren [11]. Die Kredite wurden dementsprechend auf Basis von nicht vorhandenen Sicherheiten berechnet.

Ein weiterer Grund ist, dass vergangene Entscheidungen nicht mehr reflektiert wurden. (Grund G2 - siehe Abbildung 1.1). Die Kredite wurden gebündelt unter den Banken

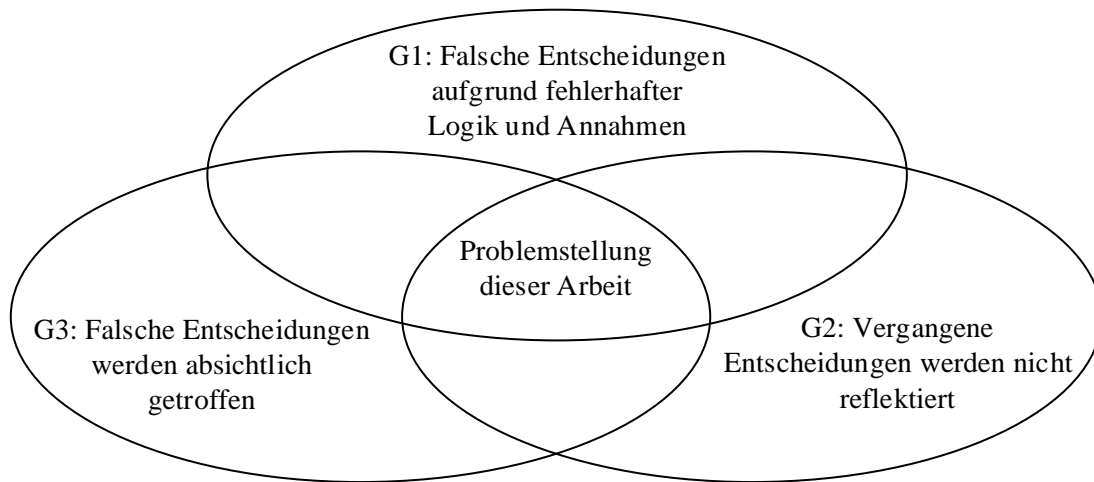


Abbildung 1.1.: Gründe für falsch getroffene fachliche Entscheidungen.

weiterverkauft [19, vgl. S. 2], ein Verlust-Risiko bestand somit nur für die kurze Zeit, in der ein Kredit-Bündel noch nicht weiterverkauft wurde. Saunders et al. sagen "The underwriters [...] had little incentive to screen and monitor the activities of borrowers for whom they originated loans" [22]. Nach der ursprünglichen Vergabe des Kredites, wurden also keine weitere Aktivitäten unternommen, um die damaligen Kredit-Entscheidungen weiterhin zu prüfen oder zu überwachen. Ein Kontrollverlust der Situation war nur eine Frage der Zeit.

Im Falle der Finanzkrise waren allerdings nicht nur fehlerhafte Entscheidungslogik, falsche Annahmen und mangelnde Reflektion schuld am Zusammenbruch der Finanzmärkte. Absichtlich falsch getroffene Entscheidungen sind ebenso als Ursache der vielen falsch getroffenen Entscheidungen anzuführen (Grund G3 - siehe Abbildung 1.1). In einer Pressemitteilung des Vorstandes der US-Notenbank, wurden Bonus-Zahlungen an Bank-Mitarbeiter als weiterer Grund der Finanzkrise aufgezählt: "Flaws in incentive compensation practices were one of many factors contributing to the financial crisis. Inappropriate bonus or other compensation practices can incent senior executives or lower level employees, such as traders or mortgage officers, to take imprudent risks that significantly and adversely affect the firm" [9]. Mitarbeiter übertraten bewusst Risiken, nutzten mangelnde Regulationen zu ihrem Vorteil und ließen keine Möglichkeit aus ihren Gewinn auf Kosten von Stabilität und Sicherheit zu maximieren.

Zusammenfassend soll Abbildung 1.1 nochmals die Gründe aufzeigen die verantwortlich sind, dass zu viele falsche fachliche Entscheidungen getroffen werden.

## 1.2. Zielsetzung

Das übergeordnete Ziel ist die in 1.1 genannte Problemstellung, dass zu viele falsche fachliche Entscheidungen getroffen werden, zu verbessern, indem man die Anzahl falscher fachlichen Entscheidungen reduziert. Diese Arbeit soll hierzu ein Machine Learning (ML) Verfahren evaluieren, analysieren und implementieren, um fachliche Entscheidungen zu verbessern. Aus dieser Zielsetzung können die folgenden Teilziele abgeleitet werden.

Das *erste Teilziel* ist das Erstellen und die kontinuierliche Optimierung von Modellen, anhand von vergangen fachlichen Entscheidungen und deren Korrektheit. Dazu soll im ersten Schritt ein ML-Verfahren implementiert werden, das Modelle aus den Daten vergangener Entscheidungen generiert. Die Datensätze dieser Entscheidungen werden um ein weiteres Attribut ergänzt, die Performance. Die Performance beschreibt den Fall, der sich in der realen Welt als der richtige erwiesen hat. Bei der Kredit-Entscheidungslogik beispielsweise, wäre die Performance ob ein Kunde seinen Kredit zurückgezahlt hat oder nicht. Das ML-Verfahren, soll die Kombination von Input-Daten und Performance auf Gesetzmäßigkeiten prüfen und diese anschließend als Modell zur Verfügung stellen. Ein Modell beschreibt den Output eines ML-Verfahrens und bietet zwei fundamentale Mechanismen. Erstens soll das Modell in der Lage sein, weiter optimiert werden zu können ("lernen"), zweitens soll für einen gegebenen Input-Datensatz ein dazugehöriger Output vorhergesagt werden ("evaluieren").

Das *zweite Teilziel* besteht darin Modelle heranzuziehen, um Entscheidungslogiken so zu verbessern, dass zukünftige fachliche Entscheidungen korrekter getroffen werden. Das erlernte Modell wird dazu verwendet, mehrere Datensätze zu evaluieren und den Ausgang deren Entscheidungen vorherzusagen. Die daraus resultierenden Ergebnisse sollen dann in den Entscheidungsfindungsprozess mit einfließen und somit zu mehr richtig getroffenen Entscheidungen verhelfen.

Auf Basis dieser Teilziele ergeben sich eine Reihe von Forschungsfragen:

- **Forschungsfrage 1:** Können ML-Verfahren verwendet werden, um fachliche Entscheidungen zu verbessern?
  - Was gibt es für einsetzbare ML-Verfahren?
  - Welche ML Verfahren eignen sich am besten?
- **Forschungsfrage 2:** Wie können einsetzbare ML-Verfahren anhand eines fachlichen Anwendungsfall angewendet werden?

- Welche Anwendungsfälle eignen sich hierzu?
- Was für Ergebnisse liefern diese?
- Wie können die Ergebnisse interpretiert werden?
- **Forschungsfrage 3:** Wie kann die Funktionsweise des ML-Verfahrens in ein Modell überführt werden, ohne dass dabei Performance-Daten offen gelegt oder zur Verfügung gestellt werden müssen?
- **Forschungsfrage 4:** Wie kann sichergestellt werden, dass die fachliche Entscheidung verbessert werden können?

Die vier Forschungsfragen werden in Tabelle /reftab:chapters 1.2 zu einem oder mehreren Kapiteln dieser Thesis zu geordnet.

#	Kap. 1	Kap. 2	Kap. 3	Kap. 4	Kap. 5	Kap. 6	Kap. 7
Forschungsfrage 1		×			×		
Forschungsfrage 2		×	×	×			
Forschungsfrage 3			×	×			
Forschungsfrage 4		×	×	×	×		

Tabelle 1.1.: Forschungsfragen entlang der Kapitel.

## 1.3. Abgrenzung

Einhergehend mit den zu beantwortenden Forschungsfragen findet auch folgende Abgrenzung statt: Es ist kein Ziel dieser Arbeit, einen eigenen ML-Algorithmus zu entwickeln. Die Entwicklung eines eigenen Algorithmus ist nicht notwendig, da in der Literatur und Praxis schon vielversprechende Algorithmen entwickelt worden sind. Vielmehr sollen bereits bestehende Algorithmen und Lösungen, die sich in der Praxis als erfolgreich bewiesen haben, intelligent verknüpft werden, um fachliche Entscheidungen zu verbessern. Ziel dieser Arbeit ist ausschließlich das Verbessern von automatisierbaren Entscheidungen. Entscheidungen können operativer und strategischer Natur sein, wobei sich nur operative Entscheidungen zur Automatisierung eignen.

## 1.4. Aufbau der Arbeit

Die vorliegende Bachelorarbeit besteht aus sieben Kapiteln, die sich nochmals in drei Teile (Teil I - Teil III) aufgliedern (siehe Abbildung 1.2).

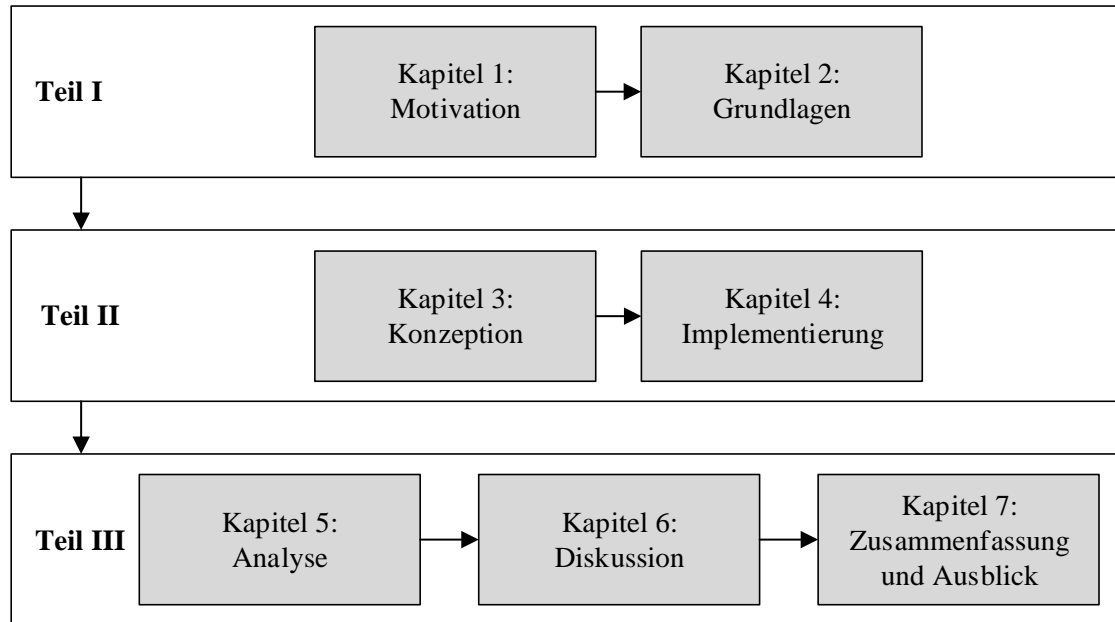


Abbildung 1.2.: Aufbau der Arbeit.

**Teil 1** umfasst neben diesem einleitenden Kapitel die theoretischen Grundlagen, die für das weitere Verständnis benötigt werden. Kapitel 2.1 erläutert wie Entscheidungen abgebildet werden können, anschließend werden die Grundlagen von ML vermittelt (Kapitel 2.2), um im Anschluss verwendbare Technologien für Decision Management und ML zu identifizieren (Kapitel 2.3).

**Teil 2** befasst sich mit der Konzeption des Prototypen (Kapitel 3), sowie dessen Implementierung (Kapitel 4) und stellt damit den Hauptteil dieser Arbeit dar. Hierzu sollen zuerst Anwendungsfälle genannt werden (Kapitel 3.1), wovon anschließend Anforderungen abgeleitet und spezifiziert werden (Kapitel 3.2). Mit dem Abschluss der Konzeption, wird die Implementierung des Prototypen beschrieben. Hierzu wird zunächst die Implementierung des Datenschemas dargelegt (Kapitel 4.1), bevor die Implementierung der zu verbessernden Entscheidungslogik (Kapitel 4.2) erläutert wird. Aufbauend auf der Entwicklung des Datenschemas und der Entscheidungslogik, kann das ML-Verfahren implementiert (Kapitel 4.3) und optimiert werden (Kapitel 4.4).

**Teil 3** bildet den Schlussteil der Arbeit und umfasst die Analyse (Kapitel 5), die Diskussion (Kapitel 6) und die Zusammenfassung inklusive eines Ausblicks (Kapitel 7).





## 2. Grundlagen

Dieses Kapitel beschreibt die grundlegenden Konzepte, die für das weitere Verständnis dieser Arbeit erforderlich sind. Abschnitt 2.1 behandelt Decision Management und gibt Aufschluss darüber, wie Entscheidungen automatisiert und verwaltet werden. In Abschnitt 2.2 folgt eine Einführung in das Machine Learning. Abschließend nennt Abschnitt 2.3 einsetzbare Lösungen zur Umsetzung der genannten Konzepte.

### 2.1. Decision Management

*Decision Management* (DM) adressiert die Verwaltung von automatisierten Entscheidungen, über den kompletten Lebenszyklus von Entscheidungen hinweg. Dieser beinhaltet die Modellierung, Ausführung, Überwachung und die Optimierung, mit dem Ziel, fachliche Entscheidungen zu verbessern, indem der Lebenszyklus kontinuierlich durchlaufen wird. Abbildung 2.1 zeigt die einzelnen Phasen des Lebenszyklus.

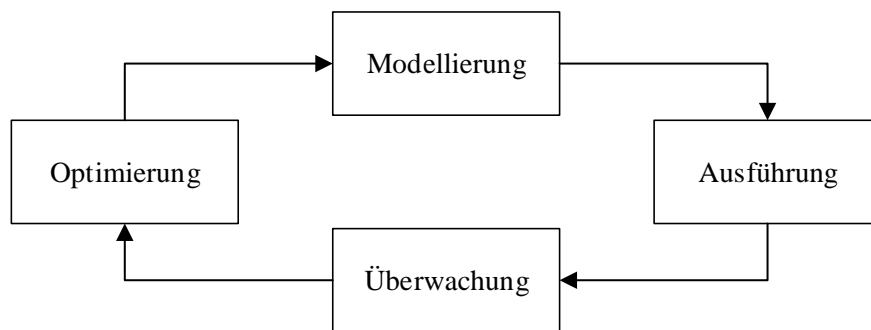


Abbildung 2.1.: Der Lebenszyklus von automatisierten Entscheidungen.

**Modellierung.** Zur Modellierung von Entscheidungen wurde 2015 der Standard Decision Model and Notation (DMN) von der Object Management Group veröffentlicht [10, vgl. S. 7 ff.]. Rücker nennt die Schaffung eines Notationsstandards für Entscheidungen, der

für Fach- und IT-Anwender gleichermaßen verständlich ist, als übergeordnetes Ziel von DMN [20, vgl. S. 40]. Die offizielle DMN-Spezifikation nennt die Erfüllung der folgenden drei Anforderungen als Ziel von DMN [10, vgl. S. 18 ff.]:

1. Modellierung von Entscheidungen, die von Menschen getroffen werden
2. Modellierung von Entscheidungen, die automatisiert getroffen werden
3. Implementierung von automatisierten Entscheidungen

Zur Umsetzung der genannten Anforderungen definiert DMN zwei verschiedene Ebenen: Die Entscheidungsanforderungsebene (deskriptiv) und die Entscheidungslogikebene (präskriptiv), beide zusammen bilden das *Decision Model*. Auf der Entscheidungsanforderungsebene werden die Anforderungen, die eine Entscheidung benötigt, in einem Decision Requirements Diagrams (DRD) modelliert. Ein DRD besteht aus Entscheidungen, Input-Daten, Business Knowledge Models und Knowledge Sources. Abbildung 2.2 [10, vgl. S. 21] zeigt die genannten Elemente eines beispielhaften DRD und deren Notation.

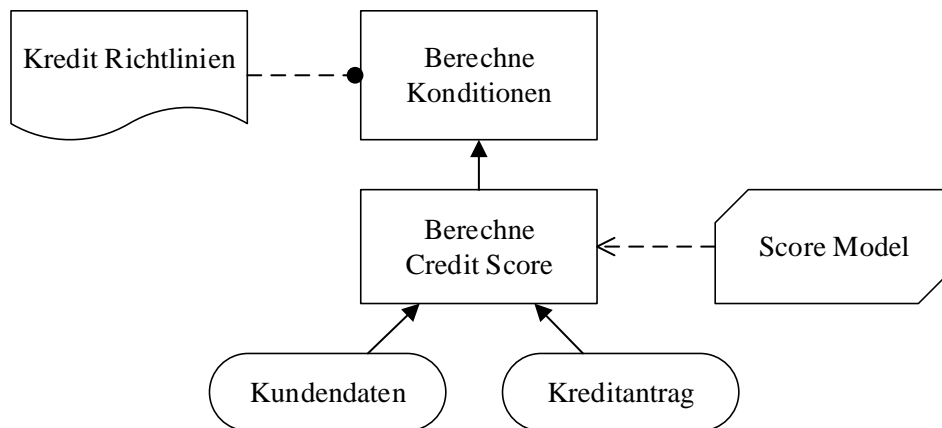


Abbildung 2.2.: Ein beispielhaftes DRD.

- Ein Input-Data-Element beschreibt Informationen, die als Eingabe für eine oder mehrere Entscheidungen dient [10, vgl. S. 30]. Abbildung 2.2 zeigt die beiden Input-Daten-Elemente Kundendaten und Kreditantrag.
- Ein Business Knowledge Model beschreibt eine Funktion die Fachwissen kapselt, beispielsweise als Geschäftsregel, Entscheidungstabelle oder als analytisches Model [10, vgl. S. 30]. In Abbildung 2.2 kapselt das Score Model das Fachwissen zur Berechnung des Credit Scores.

- Ein Decision-Element beschreibt eine Entscheidung. Sie bestimmt einen Output, anhand verschiedener Inputdaten, mithilfe von Entscheidungslogik [10, vgl. S. 20]. In Abbildung 2.2 wird der Credit Score mittels der Inputdaten Kundendaten und Kreditantrag berechnet. Anschließend wird der Credit Score an die darüber liegende Entscheidung, zur Berechnung der Konditionen, weitergegeben.
- Eine Knowledge-Source beschreibt die Quelle einer Entscheidung oder eines Business Knowledge Models. Das kann beispielsweise ein Dokument oder auch ein Fach-Experte sein [10, vgl. S. 18]. Im vorliegenden Beispiel werden die Konditionen des Kredites durch die Vorgaben der Kredit Richtlinien berechnet.

Nun gilt es sich der Entscheidungslogik-Ebene zu widmen, die insbesondere für die automatisierte Ausführung von Entscheidungen essentiell ist [10, vgl. S. 18]. DMN erlaubt den Import von existierenden Entscheidungslogik-Standards. Der wohl wichtigste Entscheidungslogik-Standard im Umfeld von DMN ist die Entscheidungstabelle. Taylor definiert Entscheidungstabellen als: "a look-up table where each cell represents an executable business rule. The conditions of the rule are columns or rows in the decision table and the intersection of the rows and columns shows the consequence of the rule" [25, S. 132]. Abbildung 2.3 zeigt eine Entscheidungstabelle zur Bestimmung des Zinses anhand eines Credit Scores. Wäre der Credit Score beispielsweise 632, würde die erste Zeile der Entscheidungstabelle zutreffen und der Zins 6 % betragen.

Berechne_Konditionen			
U	Input +	Output +	Annotation
	Credit Score	Zinssatz	
	integer	double	
1	[600..650[	6,0	-
2	[650..700[	4,0	-
3	[700..750[	2,0	-
4	[750..800[	1,5	-
5	[800..850]	1,0	-
+			

Abbildung 2.3.: Beispiel einer Entscheidungstabelle zur Berechnung des Zinssatzes.

**Ausführung.** Um Decision Models ausführen zu können, müssen diese in einem geeigneten DMN-Tool modelliert werden. Zur vollständigen Automatisierung der Entscheidungen, muss die Entscheidungslogik in der Lage sein, für jeden möglichen Satz von Input-

Daten einen Entscheidungsausgang bestimmen zu können. Um eine effiziente Ausführung zu ermöglichen wird eine Decision Engine benötigt, die es erlaubt Entscheidungen in einer produktiven Einsatzumgebung auszuführen. "Eine Decision Engine ist ein Stück Software, das eine zustandslose Schnittstelle bereitstellt, um Entscheidungen auf Basis der in DMN definierten Entscheidungslogik zu fällen" [20, S. 41]. Nach der Inbetriebnahme der Decision Engine bedürfen die getroffenen Entscheidungen der Überwachung, um sicherzustellen, dass Entscheidungen so getroffen werden, dass deren Konsequenzen bestmöglich zur Erreichung der betriebswirtschaftlichen Ziele beitragen.

**Überwachung.** Taylor nennt vier Faktoren, die kontrolliert werden sollen und dazu führen, dass Entscheidungen verbessert werden müssen [25, vgl. S. 158]:

1. Änderungen an wirtschaftlichen Zielsetzungen
2. Neue Regularien oder Richtlinien
3. Änderungen an zugrundeliegenden Datenschemata
4. Gesamt Performance der Entscheidungen

Auf die ersten drei Faktoren muss bei deren Eintritt reagiert werden, wohingegen der vierte Faktor einer ständigen Untersuchung bedarf. Proaktive Änderungen an Entscheidungslogiken bieten Möglichkeiten, die Effektivität unserer Entscheidungen zu steigern, um somit den wirtschaftlichen Zielsetzungen näher zukommen. Wie effektiv oder gut eine Entscheidung ist definiert Taylor wie folgt: "The goals and key performance indicators or metrics of your business set a context that defines what a good or an effective decision looks like. The data you have collected over time gives you insight into what works and what doesn't, which is reflected in your current decision-making approach." [25, vgl. S. 159]. Die Analyse der Daten vergangener Entscheidungen ist somit unerlässlich, um proaktiv Entscheidungslogiken zu verbessern. Um eine effektive Überwachung und Verbesserung zu ermöglichen, müssen umfangreiche Datenmengen über die Effektivität von Entscheidungen gesammelt werden. Dazu zählen [25, vgl. S. 164]:

- Ausführungs-Daten: Hierzu zählen alle Daten die während der Ausführung einer Entscheidung erhoben werden können. Beispielsweise IDs, Zeitstempel, Eingabedaten, Ausgabedaten, die Quelle des Aufrufs, oder kontextbezogene Daten die während der Ausführung generiert wurden.
- Antwort-Daten: Meistens ergibt sich aus einer Entscheidung heraus eine Konsequenz für den Empfänger der Entscheidung. Um nochmals das Beispiel der Online-

Kredit-Bewerbung aufzugreifen, nachdem der Kreditantrag evaluiert und das Ergebnis zurückgeliefert wurde, kann die Reaktion des Empfängers gespeichert werden. Der Empfänger kann das zurückgelieferte Kreditangebot ablehnen, zustimmen, mehr Informationen verlangen oder für später abspeichern. Gelingt es die Reaktion des Empfängers mit der Entscheidung zur späteren Analyse abzuspeichern, können daraus wertvolle Rückschlüsse über die Effektivität der Entscheidung gewonnen werden.

- **Andere Unternehmens-Daten:** Neben den Daten die direkt oder indirekt durch die Entscheidung und ihre Konsequenz entstehen, macht es Sinn weitere Unternehmensdaten an vergangene Entscheidungen zu knüpfen, die später in unserer Überwachungsumgebung zu sehen sein sollen. Bei einer Entscheidung zur Berechnung der Bonität beispielsweise würde die Information, ob es zu Zahlungsausfällen kam, verknüpft mit dem entsprechenden Entscheidungsdatensatz erlauben, direkte Aussagen über die Effektivität der Entscheidungslogik zu treffen.

Eine entsprechende Überwachungsumgebung müsste all diese Daten sammeln und aufbereiten, so dass der komplette Entscheidungsfindungsprozess und die daraus resultierenden Ereignisse im Nachhinein noch nachvollzogen werden können.

**Optimierung.** Um bestehende Entscheidungen optimieren zu können, bedarf es einer Simulations-Umgebung in der verschiedene Ansätze gegeneinander getestet werden können. Ein mögliches Szenario könnte so aussehen, dass man die Entscheidungslogik im Produktiveinsatz, in einem A/B Test gegen eine neue Entscheidungslogik testet. Testet man fortlaufend parallel verschiedene Ansätze kann man erkennen, welche sich langfristig am effektivsten auf wirtschaftliche Zielsetzungen auswirken [25, vgl. S. 173]. *Predictive Analytics* eignen sich um neue Entscheidungslogiken zu entwickeln. Predictive Analytics bezeichnet [3, vgl. S. 5] das Verwenden von Verfahren aus der Statistik, sowie dem Machine Learning um vorherzusehen, mit welcher Wahrscheinlichkeit ein ungewisses Ereignis eintritt.

## 2.2. Machine Learning

*Machine Learning* (ML) ist eines der zentralen Themengebiete innerhalb der *Künstlichen Intelligenz* (KI) [27, vgl. S. 2082] und beschreibt das Forschungsgebiet des Computer

befähigt zu lernen, ohne explizit dafür programmiert worden zu sein [18, vgl. S. 1]. Lernen im Kontext von ML definieren Mitchell et al. als: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E" [17]. Soll beispielsweise ein Programm implementiert werden, das aufgrund vergangener Banktransaktionen (Experience E) lernen soll, wie man betrügerische Banktransaktionen erkennt (Task T), wird das Programm, nach erfolgreichem Lernen, besser darin sein betrügerische Banktransaktionen zu erkennen (Performance P). Der Zusammenhang zwischen Ursache und Wirkung soll mit Hilfe eines Modells beschrieben werden. Das Modell soll aus den Erfahrungen der Vergangenheit auf Ergebnisse in der Zukunft schließen können. Dabei wird versucht Eingabe- auf Ausgabewerte abzubilden, so dass die Differenz zwischen dem vorhergesagten Wert und dem realen Wert möglichst klein ist [16, vgl. S. 68]. Abbildung 2.4 veranschaulicht den Lernvorgang.

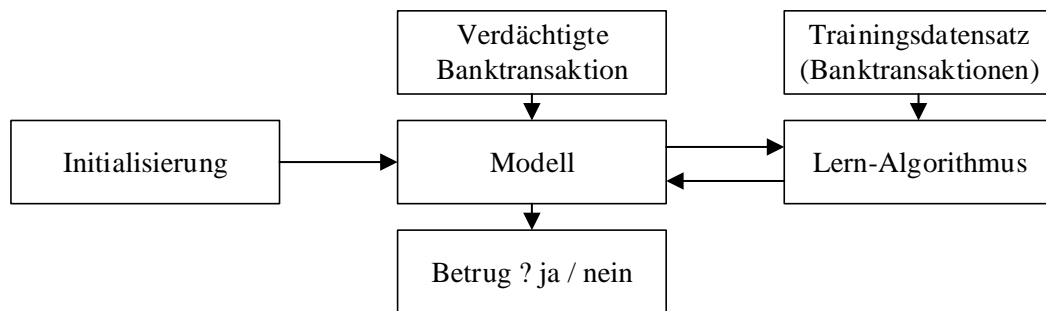


Abbildung 2.4.: Vorgehensweise zur Entwicklung eines Modells.

Die Banktransaktionen, die die Trainingsdaten bilden, besitzen verschiedene Attribute, die im Kontext von ML *Features* (Eingabedaten wie Alter, Geschlecht etc.) genannt werden. Neben den Attributen, die die Banktransaktion beschreiben, besitzen die Trainingsdatensätze ein *Label*. Das Label bildet die Erfahrung aus der gelernt werden soll. Im vorliegenden Beispiel wäre das Label ein boolescher Wert, der angibt, ob die vorliegende Banktransaktion betrügerisch war oder nicht. Tabelle 2.1 zeigt den beispielhaften Aufbau der Trainingsdaten.

Abhängig von der zugrundeliegenden Problemstellung gibt es mehrere Verfahren zur Modellbildung. Zur Auswahl eines geeigneten Lern-Algorithmus, muss die Problemstellung und die vorhandenen Daten validiert und einem der folgenden Verfahren zugeordnet werden:

- *Supervised-Learning*: Die Lerndatensätze bestehen aus Features und Labels. Das

Feature 1	Feature 2	Feature 3	Label
0.86743	0.123	0.34657	1
0.26926	0.756	0.93095	0
...	...	...	...

Tabelle 2.1.: Beispielhafter Aufbau der Trainingsdaten.

Modell wird trainiert, in dem man ihm vorgibt, wie Klassen richtig zugeordnet oder Werte richtig berechnet werden [16, vgl. S. 68].

- *Unsupervised-Learning*: Die Lerndatensätze besitzen keine Labels. Der Lern-Algorithmus versucht nur anhand der Inputvariablen Strukturen abzuleiten, wodurch dann das Modell gebildet wird [21, 16].
- *Semi-Supervised-Learning*: Eine Mischform zwischen Supervised-Learning und Unsupervised-Learning, wobei manche Trainingsdatensätze Labels besitzen können und andere nicht [28, vgl. S. 892].
- *Reinforcement-Learning*: Das Reinforcement-Learning basiert darauf, dass ein Agent lernt, wie er sich in einer bestimmten Umgebung verhalten soll, in dem er eine Belohnung beziehungsweise eine Strafe in Form eines Skalars bekommt, der angibt wie gut oder schlecht sich der Agent verhalten hat [15, vgl. S. 5].

Über die Lernverfahren hinaus, müssen auch die zu berechnenden Ergebnisse der Problemstellung klassifiziert werden, um einen geeigneten Algorithmus zu finden:

- *Klassifikation*: Unbekannte Eingabedaten werden einer Klasse zugeordnet [16, vgl. S. 68].
  - Beispiel 1: Katzenfotos sollen von Hundefotos unterschieden werden.
  - Beispiel 2: Auf MRT-Bildern von Lungen soll erkannt werden, ob ein Tumor bös- oder gutartig ist.
- *Regression*: Bestimmung eines numerischen Wertes für einen gegebenen Eingabedatensatz [21].
  - Beispiel 1: Preisberechnung eines Gebrauchtwagens.
  - Beispiel 2: Vorhersage der verbleibenden Lebensdauer eines Motors.
- *Clustering*: Beschreibt das Gruppieren von Objekten in verschiedene Gruppen. Die Datensätze einer einzelnen Untergruppe sind sich sehr ähnlich in ihren Eigenschaften, wobei die Datensätze der verschiedenen Untergruppen sehr unterschiedlich in

ihren Eigenschaften sind [21].

- Beispiel 1: Smartphone Fotoalben, die Gesichter analysieren und Fotos nach Personen gruppieren.
- Beispiel 2: Analysieren von Kundendaten zur Gruppierung verschiedener Kundenklassen, für gezielte Marketingkampagnen.

Konventionelle Informationssysteme bearbeiten Aufgaben für gewöhnlich schneller als der Mensch. Nichtsdesto trotz handelt der Mensch in vielen Disziplinen noch "intelligenter" als ein Computer, als Beispiel hierfür ist die Gesichts- und Spracherkennung zu nennen. Daher liegt es nahe, dass Forscher sich am menschlichen Gehirn inspirieren lassen beim Versuch intelligente Maschinen zu entwickeln. Die Entwicklung von künstlichen neuronalen Netzen geht zurück in die 40er Jahre, als die notwendigen biologischen und elektrotechnischen Voraussetzungen vorlagen [23, vgl. S. 1]. Grundbaustein der menschlichen Intelligenz bilden Neuronen, die über Ein- und Ausgangsverbindungen (Synapsen) mit bis zu 200000 anderen Neuronen verbunden sein können [24, vgl. S. 5]. Das *Perzeptron* ist ein künstliches Neuron, das in den 60er Jahren zur Mustererkennung entwickelt wurde [2, vgl. S. 86]. Abbildung 2.5 zeigt ein Perzeptron.

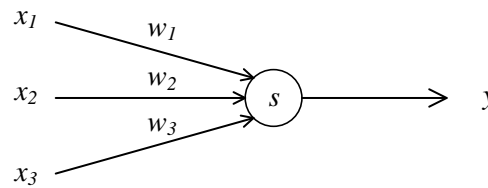


Abbildung 2.5.: Aufbau eines Perzeptrons.

Das Beispiel-Perzeptron besitzt die Inputvariablen  $x_1$ ,  $x_2$ ,  $x_3$ , die Gewichte  $w_1$ ,  $w_2$ ,  $w_3$ , sowie den Schwellenwert  $s$ . Jede Verbindung zwischen zwei Neuronen besitzt ein Gewicht, das den Einfluss eines Neuron A auf ein Neuron B beschreibt [2, vgl. S. 86]. Der Ausgabewert ermittelt sich aus dem Vergleich der gewichteten Summe  $\sum_j w_j \times x_j$  und einem zuvor definiertem Schwellenwert [12, 14, 6]. Formel 2.1 [12, 6] zeigt den Vergleich nochmals als mathematischen Ausdruck:

$$y = \begin{cases} 0 & \text{if } \sum_j w_j \times x_j \leq s \\ 1 & \text{if } \sum_j w_j \times x_j > s \end{cases} \quad (2.1)$$



Wird der Schwellenwert  $s$  überstiegen, gibt das Perzeptron die 1 zurück anderen Falls die 0. Abbildung 2.6 soll die Funktionsweise des Perzeptrons nochmals mit Hilfe von Beispiel-Werten veranschaulichen. Aus der Abbildung kann der Schwellenwert  $s = 6$  entnommen werden und mittels der Inputvariablen und Gewichten errechnet sich  $\sum_j w_j \times x_j = 8$ . Da  $8 > 6$  wird der Schwellenwert überstiegen und das Perzeptron liefert die 1 zurück (vgl. Formel 2.1).

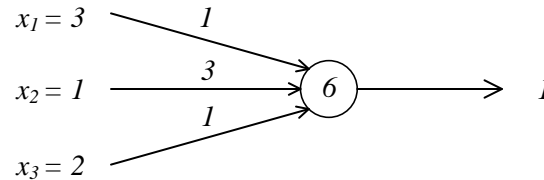


Abbildung 2.6.: Modell eines Perzeptrons mit Beispiel-Daten.

Neben dem Perzeptron gibt es noch weitere künstliche Neuronen, wie z.B. das Sigmoid-Neuron. Zum Verständnis von neuronalen Netzen, ist die Betrachtung des Perzeptrons an dieser Stelle allerdings genügend. Mit einem einzigen künstlichen Neuron ist es jedoch nicht möglich, komplexere Sachverhalte abzubilden. Dies wird erst durch die Kombination vieler künstlicher Neuronen möglich. Üblicherweise besteht ein künstliches neuronales Netz aus einer Eingabe-Schicht (*Input-Layer*), einer Versteckten-Schicht (*Hidden-Layer*) und der Ausgabe-Schicht (*Output-Layer*) [6, vgl. S. 291]. Abbildung 2.7 soll diesen Sachverhalt verdeutlichen.

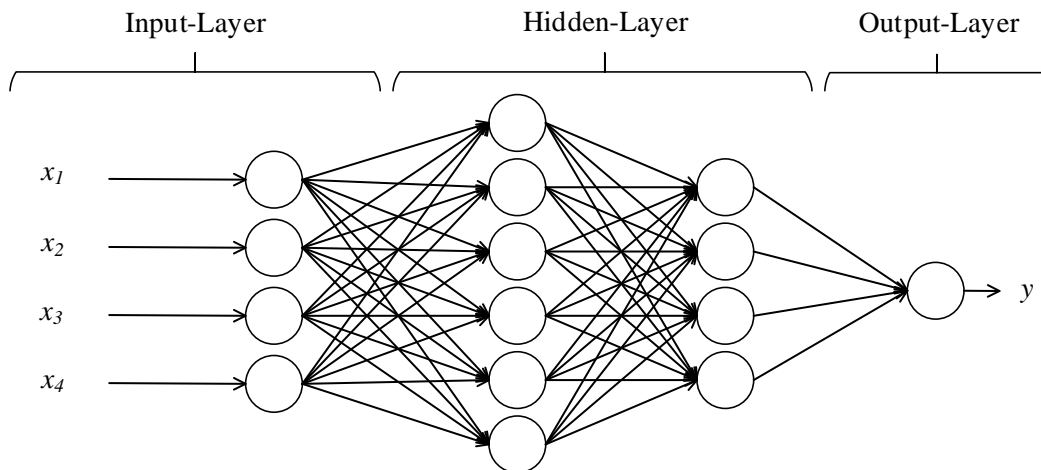


Abbildung 2.7.: Aufbau eines neuronalen Netzwerkes mit mehreren Schichten.

Dabei ist die Anzahl der Neuronen in der Hidden-Layer variabel, wohingegen die Input-Layer so viele Neuronen wie Inputvariablen besitzt. Die Anzahl der Neuronen in der

Output-Layer ergibt sich durch die Anzahl der insgesamt zu berechnenden Werte des neuronalen Netzes. Soll beispielsweise ein neuronales Netz zur Erkennung von Ziffern in Bildern entwickelt werden, gäbe es zehn Neuronen in der Output-Layer, wobei jedes Neuron eine Ziffer aus dem Bereich null bis neun repräsentiert.

Durch das Variieren der Gewichte und der Schwellenwerte, ergeben sich Modelle mit denen verschiedene Entscheidungsprozesse abgebildet werden können. Um ein Modell zu entwickeln muss das neuronale Netz lernen (trainiert werden). Im Kontext von neuronalen Netzen versteht man unter Lernen, die Modifikation der Gewichte [6]. Zur Modifikation dieser Parameter, werden Lern-Algorithmen wie der *Backpropagation-Algorithmus* verwendet. Der Backpropagation-Algorithmus wird für Supervised-Learning Problemstellungen verwendet [2, vgl. S. 89], indem iterativ Datensätze durch das Neuronale-Netz gegeben werden, so dass auf Basis der Abweichung zwischen berechnetem und tatsächlichem Wert (Fehler-Funktion), die Gewichte dahingehend aktualisiert werden das die Fehler-Funktion kleiner wird.

- .....
- Was ist Machine Learning
  - Arten von Algorithmen
  - Neuronale Netze
  - Übergang: Es gibt viele verschiedene Algorithmen etc. allerdings macht es keinen Sinn mehr diese selbst zu implementieren -> Verwendung von Frameworks und somit überleiten auf nächste Kapitel

## 2.3. Technologien

- Verwendung von Frameworks erforderlich um Effizient zu sein.
- Rad nicht neu erfinden evtl. Studie zur Verwendung von Frameworks.

### 2.3.1. Decision Management

- ACTICO - SAS - FICO - Signavio - Camunda

## 2.3.2. Machine Learning

- Neuroph - Aerosolve - Deeplearning4j - Apache Spark

Die nachfolgende Rechnung soll das Gewicht-Update mit dem Backpropagation-Algorithmus erklären. Abbildung 2.8 zeigt ein beispielhaftes neuronales Netz mit Sigmoid-Neuronen. Ein Sigmoid-Neuron ist ein modifiziertes Perzeptron, dessen Ausgabewert jeden Wert zwischen null und eins annehmen kann.

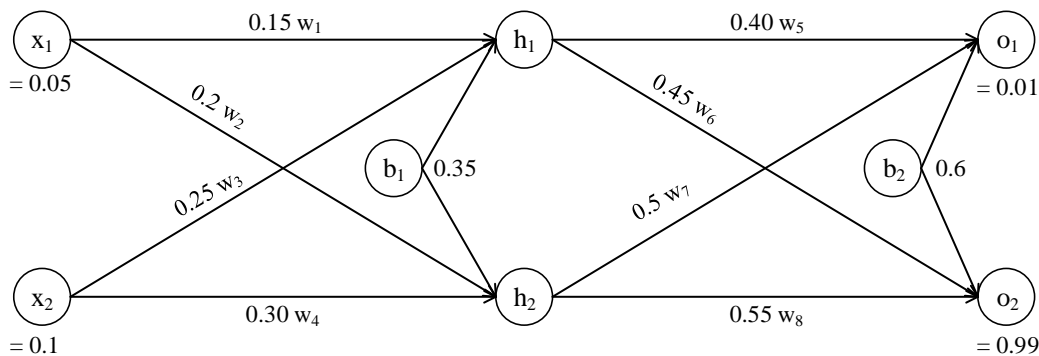


Abbildung 2.8.: Neuronalesnetz mit Sigmoid-Neuronen und Bias.

Das Neuronalesnetz besitzt zwei Inputvariablen  $x_1$  und  $x_2$ , zwei Outputvariablen  $o_1$  und  $o_2$  und zwei Biase  $b_1$  und  $b_2$ . Die Gewichte  $w_n$  können den Kanten entnommen werden. Des weiteren enthält die Zeichnung den ersten Lerndatensatz. Für  $x_1 = 0,05$  und  $x_2 = 0,1$  sollen die Werte  $o_1 = 0,01$  und  $o_2 = 0,99$  vorhergesagt werden. Da das Gewicht-Update aus Basis der Fehlerrate geschieht, muss diese zuerst berechnet werden.

1. Der erste Schritt beinhaltet das Berechnen der Fehlerrate mittels der Squared-Error-Funktion (siehe Formel 2.2).

$$E_{total} = \sum \frac{1}{2} (target-output)^2 \quad (2.2)$$

Dazu muss der Output  $o_1$  für den gegebenen Lerndatensatz berechnet werden, indem wir den Output des Neurons  $h_1$  berechnen:

$$net_{h1} = w_1 * x_1 + w_2 * x_2 + b_1 * 1$$

$$net_{h1} = 0.15 * 0.05 + 0.2 * 0.1 + 0.35 * 1 = 0.3775$$

Einsetzen von  $net_{h1} = 0,3775$  in die Sigmoid-Funktion:

$$out_{h1} = \frac{1}{1+e^{-net_{h1}}} = \frac{1}{1+e^{-0.3775}} = 0,593269992$$

Wiederholen wir den selben Schritt für  $h_2$

2.

$$\frac{\partial E_{total}}{\partial w} = \frac{\partial E_{total}}{\partial out} * \frac{\partial out}{\partial net} * \frac{\partial net}{\partial w} \quad (2.3)$$

## 3. Konzeption

Theoretische Grundlagen geklärt. Zur Erreichung des ersten Teilziels, “Konzepte ermitteln“, fehlt nun die Konzeption eines Prototypen und die Definition von Anwendungsfällen mit denen ich meine Forschungsfrage prüfen möchte.

### 3.1. Definition von Anwendungsfällen

Mit dem Prototypen möchte ich anhand von 2 Anwendungsfällen aus dem Finanzbereich prüfen, ob ich die dort modellierten Entscheidungen verbessern kann. Hierzu soll eine vereinfachte Kreditantragsbewertung in DMN modelliert werden. Anschließend möchte ich prüfen ob auch bei einem anderem Anwendungsfall, das selbe Resultat erwartet werden kann. Hierzu soll die Betrugserkennung von Banktransaktionen modelliert werden.

#### 3.1.1. Erkennung von Kreditausfällen

- Kurze Einordnung des Geschäftszweigs “Lending“ plus Bedeutung des Themas für den Finanzsektor
- Vorstellen des Datenschemas evtl. tabellarisch (Input und Output Data)
- Vorstellen des Decision Tables

### **3.1.2. Erkennung von Betrugsversuchen bei Banktransaktionen**

- Kurze Erläuterung von “Fraud Detection“, sowie dessen Wichtigkeit
- Vorstellen des Datenschemas evtl. tabellarisch (Input und Output Data)
- Vorstellen des Decision Tables

## **3.2. Identifikation relevanter Themenfelder**

- Klären der Methodik des Prototyping um These zu beweisen
- Erläuterung des Prototypen und dessen Komponenten

GRAFIK PROTOTYPE AT A GLANCE

### **3.2.1. Lerndaten**

- Welche Daten werden benötigt um unser Neuronales Netz zu füttern.
- Erklärung des Erzeugungs-Algorithmus
- Erklären von Feature Scaling
- Erklären von Labels
- Grafik mit Beispiel Datensätzen

### **3.2.2. Evaluationsdaten**

- Welche Daten werden benötigt um unser Neuronales Netz zu evaluieren.
- Erklärung des Erzeugungs-Algorithmus
- Erklären von Feature Back Scaling
- Grafik mit Beispiel Datensätzen

### **3.2.3. Decision-Engine**

- Erklärung der theoretischen Funktionsweise der Decision Engine
- Erklärung des Stellenwerts innerhalb des Prototypen

### **3.2.4. Neuronales-Netzwerk**

- Erklärung des Stellenwerts innerhalb des Prototypen
- Funktionale Anforderungen: Lernen können sowie Datensatz evaluieren, sprich Output berechnen können
- Sollte leicht Anpassbar auf verschiedene Daten-Modelle bzw. UseCases sein





## 4. Implementierung

- Dieses Kapitel beschreibt die Implementierung des (name?) Prototypen den wir zuvor konzipiert haben

### 4.1. Datenschema

- Ausgehend von den Zuvor in 3.2.1 und 3.2.2 beschriebenen Konzepten soll nun die Implementierung des Datenschemas beschrieben werden.
- H2-DB
- Sql-Script / Datenbankinhalt
- CSV Files bzw. deren Ersatz (in der Spätren Implementierung)
- Mit der erfolgreichen Implementierung der Datenschicht, haben wir nun eine Grundlage mit der wir später unser ML Verfahren sowie unsere Decison Engine füttern können.

### 4.2. Decision-Engine

- Implementierung Decision Engine
- Nach Implementierung der Decision Engine muss nun das ML Verfahren implementiert werden, das unsere Entscheidungen verbessern soll

### **4.3. Neuronales-Netzwerk**

- Implementierung nach unseren Anforderungen. 1. Lernen können und 2. Evaluieren können.
- Erklären inwiefern DeepLearning4J leicht auf andere UseCases angewendet werden kann (Um das Kriterium der Allgemeingültigkeit zu erfüllen)

### **4.4. Optimierung**

- Die Strukturen des NN wurden gesetzt. Nun müssen noch die Lernparameter bestimmt werden.
- Early Stopping using Best Model
- Trial and Error
- Tabelle aus meiner PPT
- Evaluierungsdaten können auch gelernt werden

## **5. Analyse**

- Nach dem die Problemstellungen Px und Px gelöst wurden. - Kann die eigentliche Forschungsfrage dieser Thesis beantwortet werden

### **5.1. Vorgehensweise**

- Hierzu sollen die Ergebnisse unseres Prototypen mit den Output-Werten der Decision Engine verglichen werden.
- Anhand der zwei Use Cases soll Verglichen werden, welches Verfahren (ML vs. DMN) statistisch bessere Ergebnisse liefert

### **5.2. Erkennung von Kreditausfällen**

- Tabelle mit statistischen Werten

### **5.3. Erkennung von Betrugsversuchen bei Banktransaktionen**

- Tabelle mit statistischen Werten

### **5.4. Zusammenfassung der Ergebnisse**

- Tabelle mit statistischen Werten



## 6. Diskussion

In diesem Kapitel werden die Ergebnisse der vorliegenden Arbeit diskutiert und bewertet

- > Wie aussagekräftig sind die Ergebnisse aus 5.1 bzw. 5.2
- > Was bedeutet das für die Forschungsfrage
- > Welches Nutzen hat das für Anwender
- > Integration ACTICO Plattform
- > White Box <-> Blackbox



## **7. Zusammenfassung und Ausblick**

- Diese Kapitel fasst die vorliegende Arbeit zusammen und soll darüber hinaus einen Ausblick geben wie der Prototyp weiter entwickelt werden könnte, bzw. wie ein Pilot-Projekt aussehen könnte





# Literaturverzeichnis

- [1] Aaron J. Katzel, Thomas A. Russo. *The 2008 Financial Crisis and its Aftermath: Confronting the Next Debt Challenge*. Thomas Russo, 2010. URL: <https://books.google.de/books?id=u515NZjrFuMC>.
- [2] Alpaydin, Ethem. *Machine Learning: The New AI (The MIT Press Essential Knowledge series)*. The MIT Press, 2016. ISBN: 978-0262529518.
- [3] Barga, Roger et al. *Predictive analytics with Microsoft Azure machine learning*. Springer, 2015.
- [4] Bloem, Adriaan M. and Russel Freeman. “The Treatment of Nonperforming Loans”. In: *Statistics Department International Monetary Fund* (June 2005), pp. 1–15. URL: <http://www.imf.org/external/pubs/ft/bop/2005/05-29.pdf>.
- [5] Brunnermeier, Markus K. “Deciphering the Liquidity and Credit Crunch 2007-2008”. In: *Journal of Economic Perspectives* 23.1 (Mar. 2009). URL: <http://www.aeaweb.org/articles?id=10.1257/jep.23.1.77>.
- [6] Ertel, Wolfgang. “Neuronale Netze”. In: *Grundkurs Künstliche Intelligenz: Eine praxisorientierte Einführung*. Wiesbaden: Springer Fachmedien Wiesbaden, 2016, pp. 265–311. ISBN: 978-3-658-13549-2. DOI: 10.1007/978-3-658-13549-2\_9. URL: [http://dx.doi.org/10.1007/978-3-658-13549-2\\_9](http://dx.doi.org/10.1007/978-3-658-13549-2_9).
- [7] Fight, Andrew. *Credit Risk Management*. Essential Capital Markets. Elsevier Science, 2004. ISBN: 9780080472409. URL: <https://books.google.de/books?id=wd15dswnQWgC>.
- [8] Garnaut, R. and D. Llewellyn-Smith. *The Great Crash Of 2008*. Melbourne University Publishing, 2009. ISBN: 9780522860016. URL: <https://books.google.de/books?id=hhayaM-knEIC>.

- [9] Governors of the Federal Reserve System, Board of. “Press Release announcing: ”proposal designed to ensure that incentive compensation policies of banking organizations do not undermine the safety and soundness of their organizations””. In: (2009). URL: <https://www.federalreserve.gov/newsevents/press/bcreg/20091022a.htm>.
- [10] Group, Object Management. *Decision Model and Notation (DMN) - Version 1.1*. Letzter Aufruf: 26. April 2017. URL: <http://www.omg.org/spec/DMN/1.1/PDF/>.
- [11] Hardaway, R.M. *The Great American Housing Bubble: The Road to Collapse*. Praeger, 2011. ISBN: 9780313382284. URL: [https://books.google.de/books?id=%5C\\_UKK41Qh8kcC](https://books.google.de/books?id=%5C_UKK41Qh8kcC).
- [12] Haykin, Simon S. *Neural Networks: A Comprehensive Foundation*. International edition. Prentice Hall, 1999. ISBN: 9780132733502. URL: <https://books.google.de/books?id=bX4pAQAAMAAJ>.
- [13] Hurd, Michael D and Susann Rohwedder. *Effects of the financial crisis and great recession on American households*. Tech. rep. National Bureau of Economic Research, 2010. URL: <http://www.nber.org/papers/w16407.pdf>.
- [14] Lippmann, Richard. “An introduction to computing with neural nets”. In: *IEEE Assp magazine* 4.2 (1987), pp. 4–22.
- [15] Marco Wiering, Martijn van Otterlo. *Reinforcement Learning - State-of-the-Art*. Springer, 2012. ISBN: 978-3-642-27644-6. URL: <https://link.springer.com/book/10.1007/978-3-642-27645-3>.
- [16] Masanori Fujita, Nicolaj Kirchhof. “Data Analytics in der Praxis - Regressionsanalyse”. In: *Entwickler Magazin* 2017/03 (2017).
- [17] Mitchell, Tom M et al. *Machine learning*. WCB. 1997.
- [18] Munoz, Andres. *Machine Learning and Optimization*. 2014. URL: [https://www.cims.nyu.edu/~munoz/files/ml\\_optimization.pdf](https://www.cims.nyu.edu/~munoz/files/ml_optimization.pdf).
- [19] Naudé, Wim. “The financial crisis of 2008 and the developing countries”. In: 2009/01 (2009). URL: <http://hdl.handle.net/10419/84665>.
- [20] Rücker, Bernd. “Decision Model and Notation: Digitalisierung von Entscheidungen mit DMN”. In: *OBJEKTSpektrum* 2016/05 (2016). URL: [https://www.sigs-datacom.de/uploads/tx\\_dmjournals/ruecker\\_OS\\_05\\_16\\_TRBT.pdf](https://www.sigs-datacom.de/uploads/tx_dmjournals/ruecker_OS_05_16_TRBT.pdf).

- [21] “Unsupervised Learning”. In: *Encyclopedia of Machine Learning and Data Mining*. Ed. by Sammut, Claude and Geoffrey I. Webb. Boston, MA: Springer US, 2017, pp. 1304–1304. ISBN: 978-1-4899-7687-1. DOI: 10.1007/978-1-4899-7687-1\_976. URL: [http://dx.doi.org/10.1007/978-1-4899-7687-1\\_976](http://dx.doi.org/10.1007/978-1-4899-7687-1_976).
- [22] Saunders, A. and L. Allen. *Credit Risk Management In and Out of the Financial Crisis: New Approaches to Value at Risk and Other Paradigms*. Wiley Finance. Wiley, 2010. ISBN: 9780470622360. URL: [https://books.google.de/books?id=Rr%5C\\_6y9evvowC](https://books.google.de/books?id=Rr%5C_6y9evvowC).
- [23] Schneider, H. *Historie der neuronalen Netze*. Letzter Auffggruf: 02. Mai 2017. URL: [https://www.fbi.h-da.de/fileadmin/personal/h.schneider/public\\_html/ma-c-hi/arbeitsblaetter/nn-historie.pdf](https://www.fbi.h-da.de/fileadmin/personal/h.schneider/public_html/ma-c-hi/arbeitsblaetter/nn-historie.pdf).
- [24] Strecker, Stefan. *Künstliche Neuronale Netze – Aufbau und Funktionsweise*. Letzter Auffggruf: 02. Mai 2017. Oct. 1997. URL: [http://geb.uni-giessen.de/geb/volltexte/2004/1697/pdf/Apap\\_WI\\_1997\\_10.pdf](http://geb.uni-giessen.de/geb/volltexte/2004/1697/pdf/Apap_WI_1997_10.pdf).
- [25] Taylor, J. *Decision Management Systems: A Practical Guide to Using Business Rules and Predictive Analytics*. IBM Press. Pearson Education, 2011. ISBN: 9780132884440. URL: <https://books.google.de/books?id=X2pJuopB2JsC>.
- [26] Thomas Theobald Silke Tober, Benjamin Lojak. “IMK Finanzmarktstabilitätsreport 2016”. In: *Institut für Makroökonomie und Konjunkturforschung* (Feb. 2017), pp. 1–22. URL: [https://www.boeckler.de/pdf/p\\_imk\\_report\\_121\\_2017.pdf](https://www.boeckler.de/pdf/p_imk_report_121_2017.pdf).
- [27] Wojtusiak, Janusz. “Machine Learning”. In: *Encyclopedia of the Sciences of Learning*. Ed. by Seel, Norbert M. Boston, MA: Springer US, 2012. ISBN: 978-1-4419-1428-6. DOI: 10.1007/978-1-4419-1428-6\_1927. URL: [http://dx.doi.org/10.1007/978-1-4419-1428-6\\_1927](http://dx.doi.org/10.1007/978-1-4419-1428-6_1927).
- [28] Zhu, Xiaojin. “Semi-Supervised Learning”. In: *Encyclopedia of Machine Learning*. Ed. by Sammut, Claude and Geoffrey I. Webb. Boston, MA: Springer US, 2010, pp. 892–897. ISBN: 978-0-387-30164-8. DOI: 10.1007/978-0-387-30164-8\_749. URL: [http://dx.doi.org/10.1007/978-0-387-30164-8\\_749](http://dx.doi.org/10.1007/978-0-387-30164-8_749).



# A. Anhang

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Hier folgt Beispiel A.1 für Quellcode:

Listing A.1: XML-Konfiguration: settings.xml.

```
<servers>
  <server>
    <id>xxx</id>
    <username>(Benutzername)</username>
    <password>(Passwort)</password>
  </server>
  <server>
    <id>yyy</id>
    <username>(Benutzername)</username>
    <password>(Passwort)</password>
  </server>
</servers>
```