Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Thomas Beznoskov

Klassifikation von Produkttypen mit Hilfe von Machine Learning Methoden zur Automatisierung der Navigationspfade in einem Online-Shop

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Thomas Beznoskov

Klassifikation von Produkttypen mit Hilfe von Machine Learning Methoden zur Automatisierung der Navigationspfade in einem Online-Shop

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Ulrike Steffens
Zweitgutachter: Prof. Dr. Chris Biemann

Eingereicht am: 10. Januar 2019

Thomas Beznoskov

Thema der Arbeit
Klassifikation von Produkttypen mit Hilfe von Machine Learning Methoden zur Automatisierung der Navigationspfade in einem Online-Shop

Stichworte
Machine-Learning, Deep-learning, Produktklassifikation, Multi-Label-Klassifikation, Text Mining

Kurzzusammenfassung
Eine Großteil von Verkaufsartikeln in einem Online-Shop wird durch manuelle Klassifizierung katalogisiert, doch das sollte durch automatische Klassifizierung ersetzt werden. In dieser Arbeit wird ein System entworfen und untersucht, welches aus dem Verbund der unterschiedlichen Klassifikationsalgorithmen besteht und nach der Optimierung mit in dieser Arbeit vorgeschlagenen und untersuchten Optimierungsmethoden für die Klassifikation von Produkttypen zur Automatisierung der Navigationspfade in einem Online-Shop eingesetzt werden kann.

Thomas Beznoskov

Title of the paper
Classification of product types using machine learning methods to automate the navigation paths in an online shop

Keywords
Machine Learning, Deep Learning, Product Classification, Multi Label Classification, Text Mining

Abstract
Much of the sales items in an online store are cataloged by manual classification, but this should be replaced by automatic classification. In this thesis, a system is designed and investigated, which consists of the combination of different classification algorithms and can be used after optimization with optimization methods proposed and examined in this thesis for the classification of product types to automate the navigation paths in an online shop.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

1 Einleitung

Die vorliegende Arbeit wurde in Kooperation mit dem Ecommerce-Bereich von OTTO verfasst. OTTO GmbH & Co. KG ist der zweitgrößte Versandhändler Europas und ist seit 1995 mit Otto.de im Online-Handel vertreten. Otto.de zählt zu den erfolgreichsten Online-Shops Deutschlands. (Vgl. Hofacker und Schwandt (2018))

Die Betreuung fand durch das Team PRADA statt, das für die Optimierung von Produktdaten für vertriebliche Präsentation des Angebotes auf der Plattform Otto.de zuständig ist. Bei der Lösung ihrer Aufgaben bevorzugt das Team moderne, automatisierte, selbst-lernende und selbst-optimierende Methoden.

Im ersten Kapitel wird ein Überblick über Motivation, Zielsetzung, Methodik und Aufbau der Arbeit gegeben. Dies sollte den Einstieg in das eigentliche Thema erleichtern.

1.1 Motivation

Die stetige Entwicklung und das Wachstum des Online-Handels führt zu einem Anstieg des Angebotes. Als Folge dieses Umstandes entstehen auch jede Menge neue Herausforderungen für die Produktdatenverwaltung (Vgl. Hofacker und Schwandt (2018)).

Die großen Onlineshops bieten ihre Dienste, den Verkauf und Versand, anderen Händlern an. Während Onlineshops sich um Verkaufsabwicklung und Lieferung kümmern, müssen die Händler durch Formularausfüllung ihre Verkaufsartikel kategorisieren. Da es sich um ein sehr unterschiedliches Sortiment handelt und ständig Änderungen vorgenommen werden, ist es nicht immer möglich alle benötigten Produktdaten in der Formularform abzufragen. Dementsprechend müssen Verkaufsartikel großteils durch manuelle Klassifizierung katalogisiert werden.

Falsch zugeordnete Produkte können kostspielig sein. Auf der einen Seite kann es dazu führen, dass die Onlineshops ihren Kunden verlieren, da falsch katalogisierte Artikeln verwirrend und unprofessionell aussehen. Auf der anderen Seite kann ein Händler seine Produkte nicht verkaufen, weil die Artikel von Kunden nicht gefunden werden.

In den letzten Jahren gewinnt maschinelles Lernen auf dem Markt immer mehr an Interesse. Dies lässt sich mit den rasanten Fortschritten auf dem Gebiet der künstlichen Intelligenz erklären (Vgl. Döbel u. a. (2018); S.10). Die Weiterentwicklung der Hardware und die kontinuierliche Entwicklung der Digitalisierung und Vernetzung unsere Systeme hat dazu beigetragen, dass das maschinelle Lernen und insbesondere Deep Learning, in der automatischen Sprachverarbeitung und Bildanalyse neue Möglichkeiten eröffnet. Aus diesem Grund ist das Problem der automatischen Klassifikation von Produkttypen lösbarer geworden.

1.2 Zielsetzung und Problemstellung

In der OTTO GmbH & Co. KG werden auf der E-Commerce-Plattform Otto.de, auf Grund einer massiv wachsenden Artikelanzahl, neue besser skalierbare Systeme entwickelt. Es hat dazu geführt, dass eine neue vereinfachte Struktur der Navigationspfade mit besserer Abdeckung als heute entwickelt wird. Dabei sollte die manuelle Ermittlung der Kategorie-Information durch automatische Klassifizierung von Produkten ersetzt werden.

Diese Arbeit beschäftigt sich einerseits mit der maschinellen Lernbarkeit der einzelnen Verfahren von Text- und Bildklassifizierung genau so wie mit der Lösbarkeit von Multi-Label-Klassifikationsproblemen andererseits mit dem Entwurf und der Realisierung eines lernfähigen Systems, welches die Ergebnisse der beiden Klassifikatoren nutzt, um die gesamte Klassifikationsleistung zu verbessern. Dies soll durch die Kopplung der modernen Klassifikationsalgorithmen erreicht werden. Dabei ist das Ziel möglichst viele Optimierungsmethoden zu untersuchen, um die Klassifikationsleistung zu steigern.

Das Ergebnis dieser Arbeit soll ein System sein, welches aus dem Verbund der unterschiedlichen Klassifikationsalgorithmen besteht und nach der Optimierung mit in dieser Arbeit vorgeschlagenen und untersuchten Optimierungsmethoden für die Klassifikation von Produkttypen zur Automatisierung der Navigationspfade in einem Online-Shop eingesetzt werden kann. Dies ist in Abbildung 1.1 visualisiert.

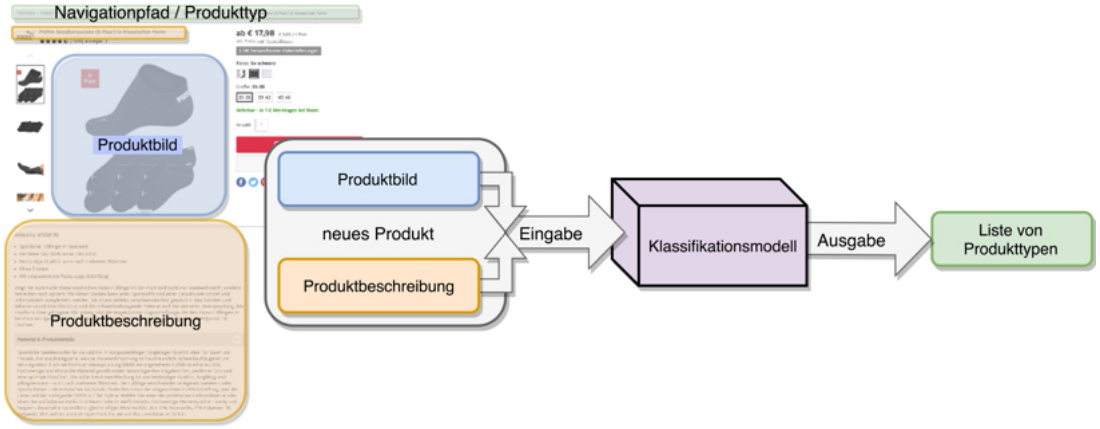

Abbildung 1.1: Klassifikationsmodell

Es gibt eine Vielzahl von Arbeiten, die sich mit der Klassifikation von Texten beschäftigen. Die meisten Lösungen limitieren sich auf Klassifikation von Nachrichten, Blogs und Online-Rezensionen. Auch Bildklassifikation ist ein Gebiet, das immer mehr an Interesse gewinnt, und dadurch auch die Forschung im Bereich antreibt. Der Großteil von diesen Arbeiten klassifiziert geringe Menge von Multi-Klassen, dabei erweisen sich maschinelle Lernmethoden als effektiv und zeigen eine hohe Genauigkeit. Trotz intensiver Forschungsaktivitäten in den Bereichen automatische Bild- und Textklassifikationen, gibt es jedoch eine geringe Anzahl von Arbeiten, die beide Klassifikationsarten in Verbindung setzen. Das ist ein Indikator dafür, dass in dieser Arbeit an einem Problem gearbeitet wird, für das keine Standardlösung existiert, und welchem bisher in der Forschung wenig Aufmerksamkeit gewidmet wurde, obwohl die Notwendigkeit für solche Klassifikation, wie Produkttypenklassifikation besteht. In dieser Arbeit werden die Erkenntnisse aus Forschungen in beiden Bereichen berücksichtigt und auf deren Ergebnisse zurückgegriffen.

Die Methoden, welche heutzutage für die Lösung von Produkttypenklassifizierung eingesetzt werden und deren Erfolge, werden im Kapitel Verwandelte Arbeiten vorgestellt. Die Ergebnisse dieser Forschungen werden ebenfalls berücksichtigt und in die vorliegende Arbeit einfließen.

1.3 Methodik

Nachdem das Ziel dieser Arbeit formuliert wurde, und nach sorgfältiger Datenanalyse, wurde sowohl Literaturrecherche betrieben als auch Experimente mit kleinen Stichproben von Daten durchgeführt. Auf Grundlage dieser Ergebnisse wurde die Entscheidung getroffen, beide Klassifikationsarten zu verbinden. Dabei wird Akzent auf Textklassifikation gesetzt. Es werden unterschiedliche Klassifikationsalgorithmen untersucht und dessen Ergebnisse analysiert.

Allerdings spielt Bildklassifikation eine unterstützende Rolle, um die Stabilität und Leistung der Lernalgorithmen, durch Anreicherung mit zusätzlichen Informationen, zu steigern. Anschließend werden unterschiedliche Ensemblemodelle entworfen und implementiert die auf die Ergebnisse von Bild- und Textklassifikationen zurückgreifen. Es werden alle vorgeschlagenen Optimierungsmethoden untersucht und anhand von durchgeführten Experimente belegt.

1.4 Gliederung und Aufbau der Arbeit

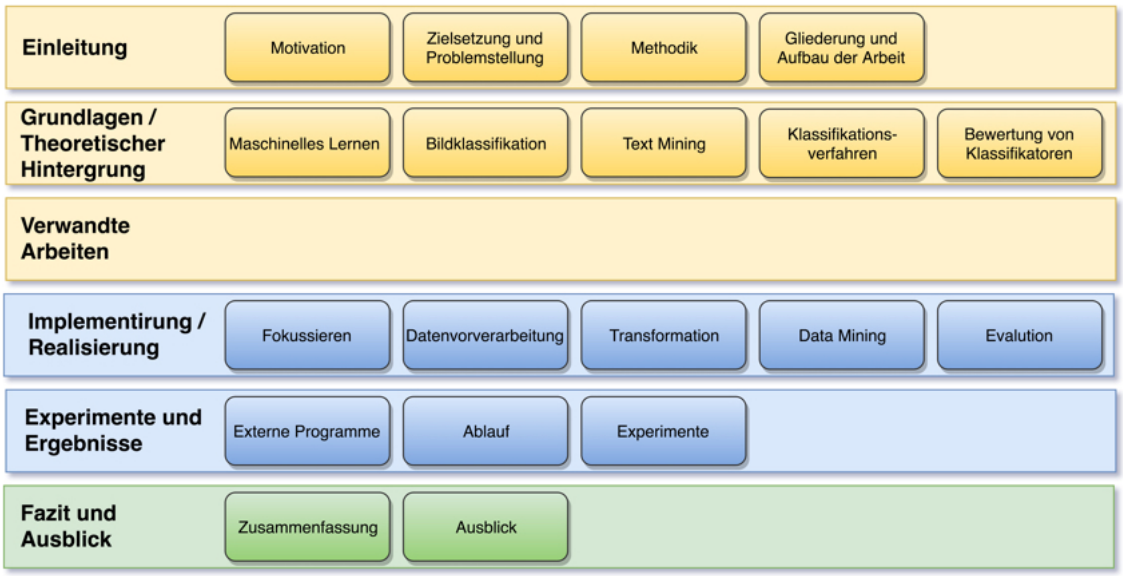

Abbildung 1.2: Aufbau der Arbeit

Die vorliegende Arbeit ist in sechs Kapitel unterteilt, die sich an der vorgestellten Zielsetzung orientieren, wie in Abbildung 1.2 zu sehen ist. Die sechs Kapitel werden in drei Gruppen zusammengefasst: Einführung (orange), Entwicklung (blau), Ergebnisse (grün).

Kapitel 1, Einleitung

Die Einleitung beschreibt die Motivation und Struktur dieser Arbeit. Es werden Zielsetzung und Problemstellung definiert. In der Methodik wird kurz beschrieben, wie es geplant wurde, das Problem zu lösen und die gewählte Vorgehensweise.

Kapitel 2, Grundlagen / Theoretischer Hintergrund

Im zweiten Kapitel werden für die Zielstellung dieser Arbeit relevante Begriffe und Definitionen aus dem Bereich des maschinellen Lernens vorgestellt. Es werden wichtige Verfahren, die später in der Arbeit verwendet werden, erläutert.

Kapitel 3, Verwandte Arbeiten

Im dritten Kapitel werden zuerst verwandte Arbeiten analysiert um einen Ausgangspunkt festzulegen und gewisse Entscheidungen zu begründen.

Kapitel 4, Implementierung / Realisierung

Im viertem Kapitel werden die Umsetzung des Konzepts und die Implementierung sowie die einzelnen Schritte und Optimierungsmöglichkeiten begründet und vorgestellt.

Kapitel 5, Experimente und Ergebnisse

Im fünften Kapitel werden die verwendeten Bibliotheken vorgestellt, genau so werden die Experimente beschrieben und die Ergebnisse ausgewertet.

Kapitel 6, Fazit und Ausblick

Im sechsten Kapitel wird das Ergebnis dieser Arbeit zusammengefasst, präsentiert und zusätzlich wird ein Ausblick gegeben, wie man das Projekt noch verbessern bzw. erweitern könnte.

Eine genauere Beschreibung der einzelnen Kapitel wird jeweils am Anfang des jeweiligen Kapitels gegeben.

1.5 Zusammenfassung

Im ersten Kapitel wurde ein Überblick über Motivation, Zielsetzung, Methodik und Aufbau der Arbeit gegeben. Dies sollte den Einstieg in das eigentliche Thema erleichtern. Im nächsten Kapitel werden für diese Arbeit die relevanten Begriffe und Definitionen vorgestellt.

2 Grundlagen / Theoretischer Hintergrund

In diesem Kapitel werden die Grundlagen und Lernstile des maschinellen Lernens, für ein Verständnis dieser Arbeit erläutert. Des Weiteren werden alle für diese Arbeit relevanten Klassifikationsverfahren in verkürzter Form vorgestellt. Anschließend werden die Metriken für die Bewertung der Klassifikationsmodelle festgelegt, welche in dieser Arbeit eingesetzt werden.

2.1 Maschinelles Lernen

Die Aufgabenstellung, die dieser Arbeit zugrunde liegt, ist in dem Teilbereich der Informatik Künstliche Intelligenz (KI) anzusiedeln. Um noch genauer zu sein, einem Teilgebiet der KI, dem maschinellen Lernen. Die Forschungsrichtung des maschinelles Lernens verfolgt das Ziel bestimmte Muster oder Regeln in vorgegebenen Datenmengen automatisiert zu finden. Die gewonnene Information kann dann auf weiteren Daten mit ähnlicher Struktur und Aufbau angewendet werden um neue oder effizientere Lösungen zu erreichen. Zum Beispiel wäre es nicht möglich einen Computer für Gesichtserkennung oder Spracherkennung zu programmieren, ohne die Hilfe von Lernalgorithmen (Vgl. Russell und Norvig (2012)).

2.1.1 Lernstile des Maschinellen Lernens

Für unterschiedliche Zwecke werden geeignete Lernstile eingesetzt. Es ist abhängig von der Zusatzinformation, die zu der Verfügung steht welche algorithmische Umsetzung von den Lernstilen des maschinellen Lernens verwendet werden. Beim überwachten Lernen müssen die richtigen Antworten zu den Beispielen, in dieser Arbeit Labels genannt, vorhanden sein. Beim unüberwachten Lernen reichen nur die Beispieldaten aus, um daraus Muster in den Daten zu

erkennen für die Gruppierung der Daten und darauf basierten Vorhersagen zu erzeugen (Vgl. Döbel u. a. (2018)).

Überwachtes Lernen

Die Hauptaufgabe beim Überwachten Lernen besteht darin, die Beziehung zwischen bestimmten Merkmalen der Daten zu einem dieser Daten zugeordneten Label zu modellieren. Das daraus resultierende Ergebnis sollte ein Modell sein, das benutzt wird, um für die neuen, unbekannten Daten ein Label vorherzusagen. In der Abbildung 2.1 sind die wesentliche Schritte beim überwachten Lernen dargestellt. Der Algorithmus erhält einen Satz von Trainingsdaten als Eingabedaten und dazugehörige Labels als Ausgabe. Als Erstes werden aus Trainingsdaten Merkmale extrahiert. Anschließend wird ein Modell trainiert indem es zugrunde liegende Zusammenhänge lernt und so sein eigenes Modell anpasst, dass es besser zu den gegebenen Trainingsdaten passt. Das Modell wird evaluiert, beispielsweise mit den Trainingsdaten, die nicht zum Trainieren verwenden wurden. Je nach Genauigkeit des Models wird entschieden ob es noch mal zu Merkmalsextraktion oder Modelltraining zurückspringt, um das Modell zu optimieren oder es mit der Vorhersage von Labels beginnt. Bei der Vorhersage werden zuerst die im Training verwendete Merkmale von den Daten ohne bekannte Labels extrahiert. Auf der Merkmalsmatrix wird das trainierte Modell angewendet, als Ergebnis daraus werden die Labels vorhergesagt (Vgl. Raschka (2015)).

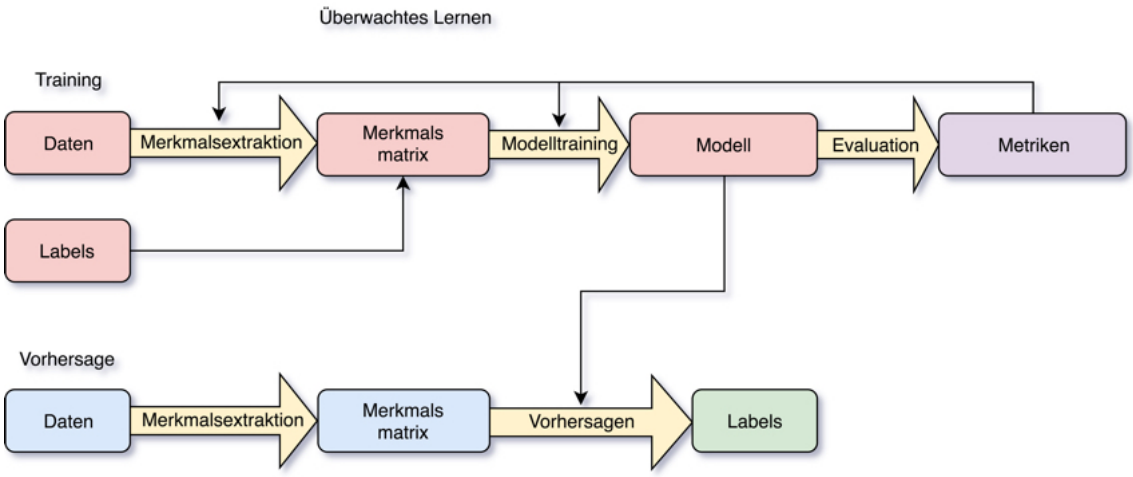

Abbildung 2.1: Ablauf beim überwachten Lernen

Unüberwachtes Lernen

Der wichtige Unterschied zwischen überwachten und unüberwachten Lernen ist, dass beim unüberwachten Lernen Trainingsdaten ohne dazugehörige Labels verwendet werden. Es wird damit begründet, dass es sich meistens um sehr große, unstrukturierte Datenmengen handelt, und es im Vorfeld nicht bewusst ist, wie gut sie beschrieben oder nach welchen Kriterien sie aufgeteilt werden sollen (Vgl. Döbel u. a. (2018); S.26).

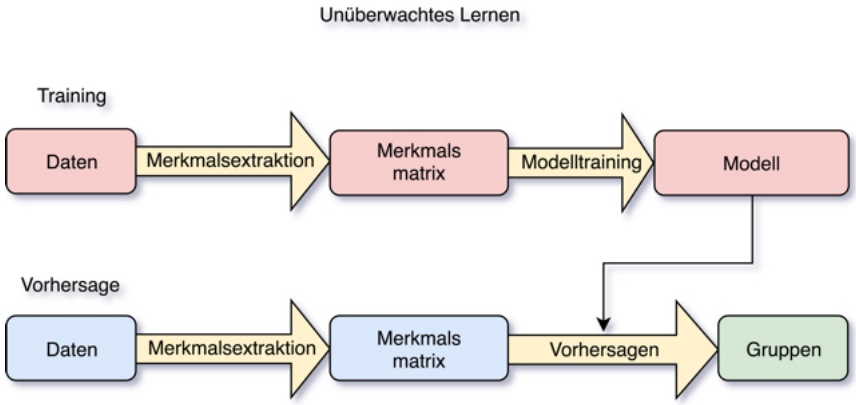

Abbildung 2.2: Ablauf beim unüberwachtes Lernen

In der Abbildung 2.2 ist der Ablauf für unüberwachtes Lernen dargestellt. Der Algorithmus erhält einen Satz von Trainingsdaten ohne Labels als Eingabedaten. Zuerst werden aus Trainingsdaten Merkmale extrahiert. Danach wird ein Modell trainiert indem Daten in unterschiedlichen Gruppen aufgeteilt werden. Bei der Vorhersage werden zuerst die im Training verwendeten Merkmale aus den Daten extrahiert. Auf der Merkmalsmatrix wird das trainierte Modell angewendet, als Ergebnis daraus wird die Zugehörigkeit zur jeweiligen Gruppe vorhergesagt. Beim unüberwachten Lernen entstehende Modelle werden für die Aufgaben wie Clustering und Dimensionsreduktion eingesetzt.

Es wurden wichtige Lernstile des maschinellen Lernens für diese Arbeit vorgestellt. Die Funktionsweisen der weiteren Lernstile sind in folgenden Büchern beschrieben: Raschka (2015) und Russell und Norvig (2012), sind aber für dieser Arbeit nicht relevant.

2.2 Bildklassifikation

Im maschinellen Lernen ist die Bildklassifikation anhand von vielen Bildern ein Standardverfahren. Convolutional Neural Networks (CNN) ist das beliebteste Verfahren bei Bildklassifizierungsaufgaben im Bereich der Bildverarbeitung. Die Kernidee bei CNN besteht darin, viele Schichten von Merkmalsdetektoren aufzubauen, um die räumliche Anordnung von Pixeln in einem Eingangsbild zu berücksichtigen (Vgl. Raschka (2015); S.381). Bildklassifikation mit CNN ist so erfolgreich geworden, dass es für diese Algorithmen einige Wettbewerbe wie den Imagenet[1] gibt. Bei diesem Wettbewerb wurde an einem Klassifikationsproblem gearbeitet, für das Bilder einer von 1000 Klassen zugeordnet werden sollen. Es werden 14 Millionen Bilder zum Training verwendet. Eins der Wettbewerb erfolgreichen CNN, wird in dieser Arbeit für die Merkmalsextraktion bei Produktbildklassifikation eingesetzt.

2.3 Text Mining / Computerlinguistik

Text Mining ist ein Forschungsgebiet, das sich damit beschäftigt das Problem der Informationsüberflutung mit Hilfe von maschinellem Lernen und Computerlinguistik zu lösen. Ähnlich wie bei Data Mining versucht Text Mining, nützliche Informationen aus Datenquellen zu extrahieren, indem interessante Muster identifiziert und untersucht werden. Text und Data Mining Systeme weisen viele architektonische Ähnlichkeiten auf höhere Ebene auf (Vlg. Feldman und Sanger (2006); S.1). Aus diesem Grunde wird in dieser Arbeit versucht Text Mining und Data Mining Algorithmen zusammen zu koppeln, um von beiden Informationsarten zu profitieren und durch Training ein stabileres und leistungsstärkeres Klassifikationsmodell zu erzeugen.

[1] http://image-net.org

2.4 Klassifikationsverfahren

2.4.1 Random Forest

Random Forest Klassifikation gehört zu den sogenannten Ensemble-Methoden. Es werden schwach lernende Entscheidungsbäume zu einem Ensemble kombiniert, um ein robusteres Modell zu entwickeln. Damit die einzelne Bäume nicht genau dasselbe lernen, werden diese mit verschiedenen Teilmengen der Trainingsbeispiele trainiert (Vgl. Döbel u. a. (2018); S.31).

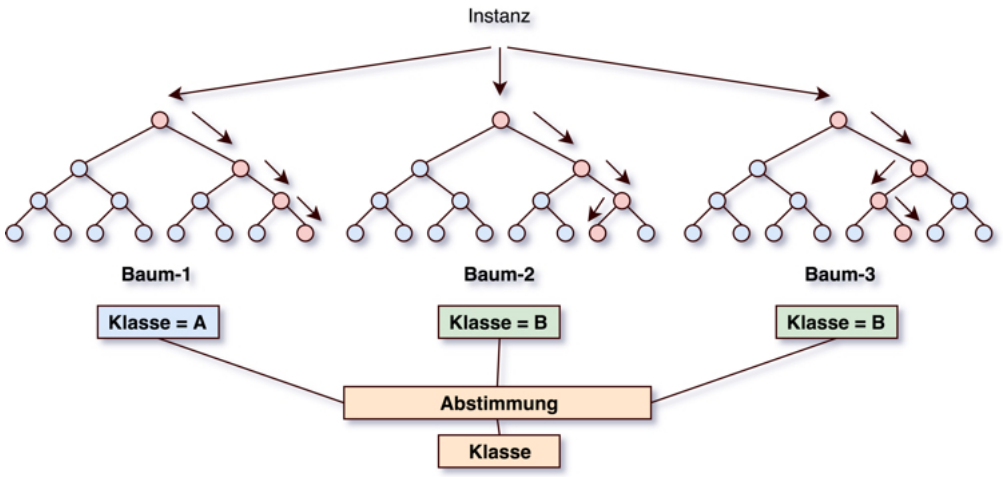

Abbildung 2.3: Random Forest Klassifizierung (Nachbildung (Vgl. Döbel u. a. (2018); S.31)

Das Modell weist einen besseres Verallgemeinerungsfehler und ist weniger anfällig für Überanpassung. Esemble-Methoden haben in den letzten zehn Jahren wegen ihrer guten Klassifikationsleistung, Skalierbarkeit und Benutzerfreundlichkeit eine große Popularität in Anwendungen des maschinellen Lernens bekommen (Vgl. Raschka (2015); S.381). Weitere Information über Vorteile und Nachteile von Entscheidungs- bzw., Klassifikationsbäumen sind bei (Vgl. Raschka (2015); S.381) zu finden. Für die durchgeführten Experimente wurde die Klasse RandomForestClassifier der scikit-learn[2] Bibliothek benutzt.

[2]https://scikit-learn.org

2.4.2 Support-Vektor-Maschine (SVM)

Die Support-Vektor-Maschine ist der populärste Ansatz für überwachtes Lernen. Wenn kein spezialisiertes Wissen über eine Domäne bekannt ist, wird diese als Erstes gewählt, um erste Testversuche durchzuführen (Vgl. Russell und Norvig (2012); S.863).

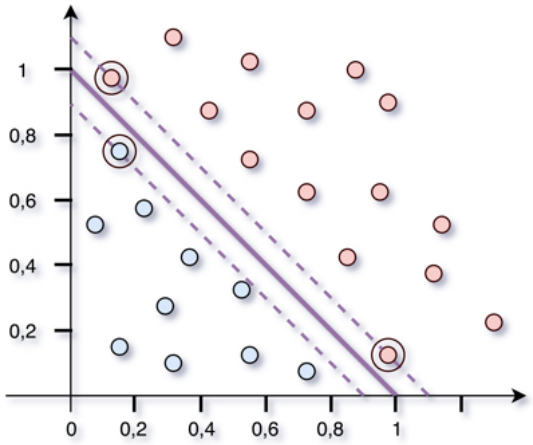

Abbildung 2.4: Suport-Vector-Maschinen-Klassifizierung (Nachbildung (Vgl. Russell und Norvig (2012); S.864)

Drei Eigenschaften machen SVMs attraktiv (Vgl. Russell und Norvig (2012); S.863):

- SVMs bauen einen Maximum-Margin-Separator. Es wird bei der Entscheidungsgrenze die größtmögliche Distanz zu Beispielpunkten gewählt.
- Das Arbeiten in hohen Dimensionen wird ermöglicht. SVMs nutzen die Eigenschaft von vielen Daten, die im ursprünglichen Eingaberaum nicht trennbar sind, sich durch Anwendung von mathematischen Operationen im Raum einer höheren Dimension leichter separieren.
- SVMs sind schnell, denn die benötigten Parameter werden nur auf Support Vektoren und nicht auf den kompletten Trainingsdaten angewendet. Sie besitzen die Flexibilität, komplexe Funktionen darzustellen, sind aber resistent gegen Überanpassung.

2.4.3 Künstliche neuronale Netze (KNN)

Künstliche neuronale Netze zeigen vermehrt Erfolge, vor allem in der Bilderkennung und Verarbeitung natürlicher Sprache (Vgl. Döbel u. a. (2018); S.37). KNN ist ein informationsverarbeitendes System, das aus Schichten besteht welche wiederum aus mehreren einfachen Einheiten, Neuronen zusammengesetzt sind. Die Neuronen sind miteinander gerichtet verbunden und tauschen die Informationen in Form der Aktivierung der Zelle aus, wie Abbildung 2.5 zeigt. Durch ein Gewicht wird die Stärke der Verbindung zwischen zwei Neuronen ausgedrückt. Das wesentliche Element der KNN ist ihre Lernfähigkeit, obwohl Lernen bei KNN als Gewichtsveränderungen zwischen den Einheiten definiert wird und das Wissen ist in seinen Gewichten gespeichert (Vlg. Rey und Rey (2017); S.5).

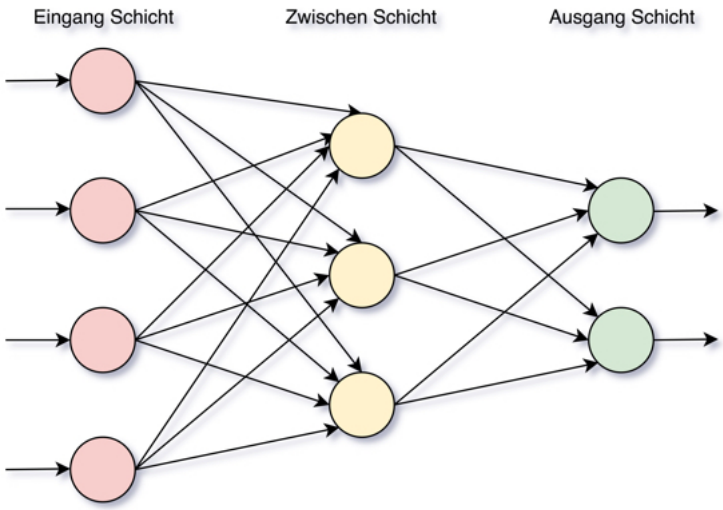

Abbildung 2.5: Schematische Darstellung KNN (Nachbildung Rey und Rey (2017); S.6)

KNN bestehen aus 3 verschiedenen Arten von Neuronen, die eine Schicht bilden. Wie in Abbildung 2.5 gezeigt ist. Die Eingangsschicht kann die Signale (Reize, Muster) von der Außenwelt empfangen. Die Zwischenschicht, die sich zwischen Eingang- und Ausgangsschicht befindet, beinhaltet eine interne Repräsentation der Außenwelt. Die Ausgangsschicht gibt die Signale an die Außenwelt weiter (Vlg. Rey und Rey (2017); S.6). Nach diesem Schema werden in dieser Arbeit KNNs gebaut z. B. mit Eingangsschicht werden beide Klassifikatoren verbunden und jedes Neuron repräsentiert ein Produkttyp.

2.5 Bewertung von Klassifikatoren

Um die Qualität eines Klassifikators zu messen und durch Parameteranpassung zu steigern, ist es notwendig die Vorhersagen des Modell eines Lernverfahrens bewerten zu können.

Die Bewertung von Multi-Label-Klassifikatoren ist deutlich komplexer als die von Single-Label-Klassifikatoren. Das ist u. a. damit zu erklären, dass bei der Single-Label-Klassifizierung in einzelnen Labels die Klassifizierung nur entweder richtig oder falsch sein kann, während bei der Multi-Label-Klassifizierung auch teilweise korrekte Ergebnisse möglich sind. Deswegen können die traditionellen Qualitätsbewertungsmetriken, die bei Single-Label-Klassifizierung verwendet werden, nicht zu der Bewertung der Qualität der Multi-Label-Klassifizierung eingesetzt werden.

Es ist von der Aufgabenstellung abhängig, nach welchen Kriterien die Klassifikationsmodelle bewertet werden sollen. Bei Multi-Label-Klassifizierung gibt es eine wichtige Trennung zwischen der Label-basierten Auswertung, die pro Label, also pro Produkttype durchgeführt wird, und Label-Satz-basiert Auswertung, die einen Label-Satz, ein Produkt bewertet. In dieser Arbeit werden beide Auswertungsverfahren eingesetzt, um einerseits Algorithmen zu identifizieren, die bei vielen unterschiedlichen Bewertungsmaßnahmen gut funktionierten, anderseits um ein vollständigeres Bild der Fähigkeiten eines Modells zu präsentieren (Vgl. Asim u. a. (2017)).

2.5.1 Hamming loss

Bei der Multi-Label-Klassifikation gibt die $Hamming\_loss$ Funktion ein durchschnittliches Maß für die Abweichung zwischen dem vorhergesagten Wert und dem tatsächlichen Wert. Es beschreibt im Intervall [0,1] die falsch zugeordnete Labels an die Gesamtzahl der Labels, wobei ein falsch zugeordnete Labels entweder $false\_negative$ oder $false\_positive$ sein können. Ein niedriger Wert der $Hamming\_loss$ Funktion ist erforderlich, um eine bessere Klassifizierung zu gewährleisten. Bei der idealen Klassifizierung ist der $Hamming\_loss$ Wert gleich 0 (Vgl. Asim u. a. (2017)). Der $Hamming\_loss$ Wert wird nach folgender Formel berechnet:

- N - ist die Anzahl der Instanzen
- L - ist die Anzahl der Labels
- $\tilde{y}_i$ - sind die vorhersagten Labels

- y_i- sind die richtigen Labels

$$Hamming\_loss = \frac{1}{N}\sum_{n=1}^{N}\frac{|\tilde{y}_i \Delta y_i|}{L} \tag{2.1}$$

2.5.2 Accuracy

Die $Accuracy$ Funktion des Multi-Label-Klassifikators ist definiert als der Anteil der vorhergesagten korrekten Labels an der Gesamtzahl der Labels für diese Instanz. Die Gesamtgenauigkeit ist der Durchschnittswert aller Instanzen. Der Wert befindet sich im Intervall [0,1], wobei je näher der $Accuracy$ Wert an 1 liegt, umso genauer ist die Vorhersage des Klassifikators (Vgl. Asim u. a. (2017)). Bei der Multi-Label-Klassifizierung wird der $Accuracy$ Wert anhand der folgenden Formel gemessen:

$$Accuracy = \frac{1}{N}\sum_{n=1}^{N}|\frac{\tilde{y}_i \bigcap y_i}{\tilde{y}_i \bigcup y_i}| \tag{2.2}$$

2.5.3 Precision

$Precision$ misst die Genauigkeit der Vorhersage des Klassifikators. Es ist der Quotient aus der Anzahl der korrekt vorhergesagten Labels und der Gesamtanzahl der vorhergesagten Labels über alle Instanzen gemittelt. Mit anderen Worten, ist es das Verhältnis $true\_positive$ zur Summe $true\_positive$ und $false\_positive$ gemittelt über alle Instanzen. Der Wert befindet sich im Intervall [0,1], wobei je näher der $Precision$ Wert an 1 liegt, umso genauer ist die Vorhersage des Klassifikators (Vgl. Asim u. a. (2017)). Die Formel für $Precision$ ist wie folgt definiert:

$$Precision = \frac{1}{N}\sum_{n=1}^{N}|\frac{\tilde{y}_i \bigcap y_i}{\tilde{y}_i}| \tag{2.3}$$

2.5.4 Recall

$Recall$ misst die Vollständigkeit der Vorhersage des Klassifikators. Es ist der Quotient aus der Anzahl der vorhergesagten korrekten Labels an der Gesamtzahl der Labels, gemittelt über alle Instanzen. Mit anderen Worten, ist es das Verhältnis von $true\_positive$ zu der Summe von $true\_positive$ und $false\_negative$ gemittelt über alle Instanzen. Der Wert bewegt sich im Intervall [0,1], wobei je näher der $Recall$ Wert an 1 liegt, umso vollständiger ist die Vorhersage des Klassifikators (Vgl. Asim u. a. (2017)). $Recall$ wird nach folgender Formel berechnet:

$$Recall = \frac{1}{N}\sum_{n=1}^{N}|\frac{\tilde{y}_i \bigcap y_i}{y_i}| \tag{2.4}$$

2.5.5 F1 - Measure

$F_1 - Measure$ ist ein Maß, das durch das harmonische Mittel von $Precision$ und $Recall$ definiert wird. So gesehen kann es als gewichteter Durchschnitt von $Precision$ und $Recall$ interpretiert werden. $F_1 - Measure$ Wert liegt im Intervall [0,1], obwohl je näher der $F_1 - Measure$ Wert an 1 liegt, umso besser ist die Vorhersage des Klassifikators (Vgl. Asim u. a. (2017)). Um $F_1 - Measure$ zu berechnen, kann folgende Formel benutzt werden:

$$F_1 - Measure = \frac{1}{N}\sum_{n=1}^{N} 2 * \frac{|\tilde{y}_i \bigcap y_i|}{|\tilde{y}_i| + |y_i|} \tag{2.5}$$

2.5.6 Subset_accuracy

$Subset\_accuracy$ beschreibt das genaue Übereinstimmungsverhältnis zwischen der vorhergesagten und der tatsächlichen Menge. Mit anderen Worten, der Durchschnitt einer Menge vorhergesagter Labels, die genau der Menge der tatsächlichen Labels entspricht. Wobei I eine Indikatorfunktion ist, die einen Wertebereich von 0;1 hat. Ein klarer Nachteil von $Subset\_accuracy$ besteht darin, dass sie nicht zwischen komplett und teilweise korrekten Mengen von Labels unterscheiden kann. Der $Subset\_accuracy$ Wert befindet sich im Intervall

[0,1], wobei je näher der $Subset\_accuracy$ Wert an 1 liegt, umso besser ist die Vorhersage des Klassifikators (Vgl. Asim u. a. (2017)). Die Formel für $Subset\_accuracy$ ist wie folgt definiert:

$$Subset\_accuracy = \frac{1}{N}\sum_{n=1}^{N} I(\tilde{y}_i = y_i) \tag{2.6}$$

2.6 Zusammenfassung

In diesem Kapitel wurden die Grundlagen und Lernstile des maschinellen Lernens, für ein Verständnis dieser Arbeit erläutert. Es wurden die relevanten Klassifikationsverfahren und Metriken, die in dieser Arbeit eingesetzt wurden, präsentiert. Im nächsten Kapitel werden Verwandte Arbeiten, die sich mit Klassifizierung von Produkten befassen, analysiert, um den aktuellen Stand der Forschung zu präsentieren.

3 Verwandte Arbeiten

Nachdem die relevanten Methoden und Verfahren vorgestellt wurden, wird in diesem Kapitel der aktuelle Stand der Forschung in Bezug auf Produkttypenklassifizierung präsentiert. Es werden zwei Arbeiten analysiert, die sich mit Klassifizierung von Produkten befassen.

Es gibt immer mehr Bereiche die von ML erobert werden. Der Onlinehandel ist ein Vorreiter. Es wird in den Medien über erfolgreiche Empfehlungssysteme von Onlineshops berichtet. Das ist die Ursache dafür, dass es immer mehr Bemühungen gibt, Produkte automatisch zu kategorisieren.

3.1 Everyone Likes Shopping! Multi-class Product Categorization for e-Commerce

Bei der ersten Arbeit, die vorgestellt wird, widmet sich Zornitsa Kozareva (2015) der Untersuchung eines Multi-Klassifikationsproblems. Es wurden auf Basis von 445.408 Produkten, die aus der Yahoo-Einkaufsplattform stammen, mehrere Klassifikationsalgorithmen, verglichen. Es wurde ein Klassifikation-Algorithmus entworfen, der vorgegebene Produkte in 319 Kategorien, die in 6 Ebenen unterteilt waren, sortieren sollte. Mit dem Word2Vec wurden Merkmale erzeugt, und anschließend mit einem künstlichen neuronalem Netz Produkte klassifiziert. Das KNN erwies sich als das beste Klassifikationverfahren und erreichte F1-Measure von 0.88. Eine weitere Untersuchung der Fehler hat gezeigt, dass die produzierten Ergebnisse der vorhergesagten Kategorien oft spezifischer und feiner waren als die Vorgegebenen.

Wie bereits erwähnt, befassen sich viele Arbeiten mit Klassifikation von geringen Klassenmengen. In der vorgestellten Arbeit waren 319 Kategorien die sich noch in 6 Ebenen unterteilen. Als Schlussfolgerung daraus, waren es durchschnittlich mehr Produktbeispiele pro Produkt als in dieser Arbeit verwendet werden (siehe Tabelle 3.1).

Ebene	Anzahl der Klassen	Produktbeispiele pro Klasse (durchschnittlich)
1	8	55 676
2	31	14 368
3	91	4789
...	...	...

Abbildung 3.1: Überblick über den Produktbeispielen

Die hierarchische Aufteilung von Produkten und die Abbildung von den gesamten Pfaden werden in dieser Arbeit nicht gemacht. Es werden mehr als 1200 Produkttypen klassifiziert und die dafür verwendeten Produktbeispiele liegen bei 100 bis 500 pro Produkttyp, was diese Arbeit deutlich von Kozareva (2015) unterscheidet. Gerade die Klassifikation mit wenigen Produktbeispielen und vielen Produkttypen macht diese Arbeit interessant.

3.2 Improving Product Classification Using Images

Die nächste Arbeit, die vorgestellt wird, befasst sich mit der Produkttypenklassifizierung. Es wurde in dem Vergleich zu der vorherigen Arbeit ein anderer Ansatz gewählt. Die Autoren verwenden bei ihre Untersuchungen nicht nur Texte sondern auch das Produktbild, um die Produkte zu klassifizieren Kannan u. a. (2011). Bei der Klassifizierung nutzen die Autoren das Bild von einem Produkt als ein schwaches Signal, um das Textklassifikationsmodell zu unterstützen. Die Verfasser begründen dies damit, dass es zwei unterschiedliche Produkte mit dem gleichen Beschreibungstext geben kann. Die Idee, wie in dieser Arbeit, zwei unterschiedliche Modelle zu verbinden um die Klassifikationsleistung zu verbessern, wurde auch in der vorliegenden Arbeit erfolgreich eingesetzt, wie in Kapitel 4 zu sehen ist. Der Unterschied ist, dass Kannan u. a. (2011) an einem Multi-Klassenproblem mit 17 Produktklassen und 28 015 Beispielprodukten arbeiten. Es wurden deutlich mehr Beispielprodukte als in der vorliegenden Arbeit verwendet.

Es wurden keine weiteren Arbeiten gefunden, die sich von diesen beiden deutlich unterscheiden und in der über ein Multi-Label-Problem geschrieben worden ist, wo auch bei der Produkttypenklassifikation mit geringen Produktbeispielen, nicht hierarchisch ein Ensemble eingesetzt wurde.

3.3 Zusammenfassung

In dem Kapitel wurden die aktuellen Forschungen mit Bezug auf Produkttypenklassifizierung präsentiert. Die beiden Arbeiten wurden analysiert. Die Ansätze und Klassifikationsmethoden, die sich in den Untersuchungen als erfolgreich erwiesen haben, sind in der vorliegenden Arbeit zu finden. In dem folgenden Kapitel werden die Verfahren genauer erläutert.

4 Implementierung / Realisierung

Das folgende Kapitel befasst sich mit der Implementierung von Klassifikationsalgorithmen. Dieses Kapitels ist nach dem allgemeinen Prozess des Data Mining aufgebaut (siehe Abbildung 4.1). Als erster Schritt wird die Datengrundlage präsentiert, auf deren Basis die Auswahl der relevanten Daten stattfindet. Danach werden die Ergebnisse der Datenanalyse und der darauf folgenden Vorverarbeitung vorgestellt. Dabei werden die relevanten Schritte präsentiert und durch durchgeführte Experimente belegt. Anschließend werden, in der Arbeit benutzte Ensemble-Modellarchitekturen erläutert und einzelne Modelle präsentiert. Am Ende dieses Kapitels werden beide Evaluationsmethoden erklärt und erläutert, warum die beiden Methode eingesetzt wurden.

Prozessmodell

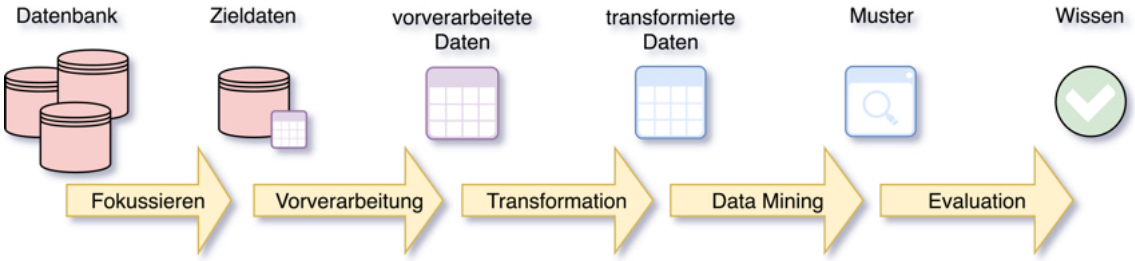

Abbildung 4.1: Prozessmodel nach Fayyad, Piatetsky-Shapiro & Smyth (Vgl. Richert (2013) , S.38); eigene Darstellung

4.1 Fokussieren

4.1.1 Datengrundlage

Für diese Bachelorarbeit wurden Produktdaten von allen Artikeln des Onlineshops Otto.de[1] bereitgestellt. Auf Grund der Tatsache, dass jeder Onlineshop eigene Taxonomien mit hierarchischen und allgemeinen Relationen besitzt, wird in dieser Arbeit keine Gruppierung auf Sortimentsebene stattfinden, stattdessen werden Produkttypen direkt vorhergesagt. Es wurden die Produktdaten für die Produkttypenklassifikation benutzt, die fast jeder Onlineshop besitzt. Es wurden folgende relevante Produktinformationen ausgewählt:

- Produktname
- Produktmarke
- Produktbeschreibung
- Verkaufsargumente
- Bilder
- Produkttypen

Die Daten wurde in Form von mehreren xml-Dateien bereitgestellt. Um die Datenanalyse und anschließende Vorverarbeitung performant durchzuführen, wurden die relevanten Produktinformationen in eine csv-Datei konvertiert. Nach der Auswahl sehen die Zieldaten folgendermaßen aus (siehe Tabelle 4.1):

Marke	Name	Beschreibung	Verkaufsargumente	Bilder	Produkttypen
PUMA	Sport...	Diese sport...	Sportliche Füßl...	1123...	Füßlinge...
...	...	...	...	...	...

Tabelle 4.1: Überblick über einen Beispieldatensatz

[1] https://www.otto.de

4.1.2 Datenanalyse

Um einen Überblick über die Menge und Eigenschaften von den Produktinformationen zu erhalten, wurden vor der Vorverarbeitung der Daten diese analysiert. Zur Analyse wurden alle zur Verfügung gestellten Daten verwendet. Nach der Datenanalyse wird eine Datenreduzierung durchgeführt, mehr darüber in 4.1.4 Datenreduzierung.

Produkttyp

Produkttypen geben die Möglichkeit, alle Produkte zu gruppieren und mit einer Menge mit gleichen Attributen auszustatten. Es ermöglicht Kunden Produkte durch Nutzung der Navigationsleiste oder der Suche im Onlineshop zu finden. Anzumerken ist, dass ein Produkt mehreren Produkttypen zugeordnet sein kann. Aus diesem Grund enthält das Feld eine Liste von Produkttypen, was das folgende Beispiel verdeutlicht.
[Füßlinge|Multipacks|Sneakersocken|Socken|Sportsocken]

Anzahl	Anteil
1	47%
2	33%
3	12%
4	5%
5	3%
6	0.6%
...	...

Tabelle 4.2: Anteil der Produkttypen Anzahl an der Gesamtmenge aller Produkte

In der Tabelle 4.2 wird gezeigt, dass es sich um ein Multi-Label-Klassifikationsproblem handelt. Mehr als die Hälfte der Produkte sind mehr als 2 Produkttypen zugeordnet. Jeder einzelne Produkttyp wird in dieser Arbeit als Label betrachtet. Es ist möglich die gesamte Produkttypenliste als Label zu betrachten und das Problem als Single-Label-Klassifikation zu lösen. Dabei sollte jedoch nicht außer Acht gelassen werden, dass sich die Anzahl der Labels vervielfältigt, und als Folge daraus sich die Anzahl von den Produkten pro Label deutlich reduziert. Dies führt dazu, dass sich die Klassifikationsleistung verschlechtert und deutlich mehr Trainingsdaten benötigt

werden um die Leistung wieder zu verbessern. Die Aussage wird durch folgende Beispiele aus der Datenanalyse verdeutlicht (siehe Tabelle 4.3).

	Multi-Label-Klassifikation	Single-Label-Klassifikation
Anzahl der Labels	5 851	25 325
Anzahl Produkte pro Label	134	31

Tabelle 4.3: Vergleich Multi-Label-Klassifikation und Single-Label-Klassifikation

Name

Die Untersuchung der Produktnamen hat ergeben, dass die Namen zu 83% aus 4 und mehr Wörtern bestehen. Wenn es sich um ein Markenprodukt handelt, wird die Marke in der Regel an erster Position in den Produktnamen geschrieben.

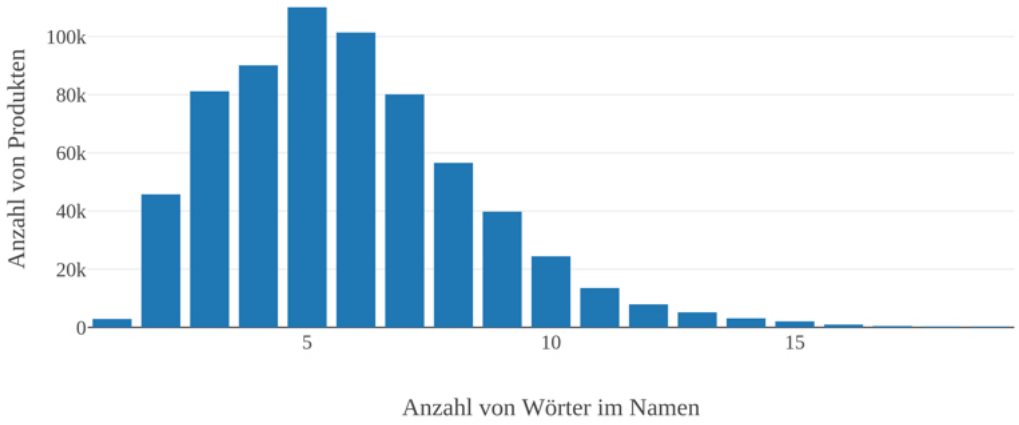

Abbildung 4.2: Bar Graph Anzahl von Wörter in den Produktnamen über alle Produkten

Die Produktnamen enthalten zu 74% einen Produkttyp in Singularform und ein Produktmerkmal, dass das Produkt gut gegenüber anderen Produkten seines Typs abgrenzt (siehe Tabelle 4.4). Des Weiteren wurde festgestellt, dass die Wiederholungsrate im Durchschnitt bei ca. 10% liegt. Daraus lässt sich schließen, dass die Produktnamen viel Informationsgehalt bezüglich

des Produkttyps besitzen. Das wurde im späteren Verlauf durch Experimente bewiesen. Als Ergebnis daraus wurden Produktnamen in der Vorverarbeitung sowohl separat als auch in Kombination mit dem Beschreibungstext behandelt.

Produktnamen	Produkttypen
Adelia´s Diamantring 585 Gold mit Diamant	Diamantringe|Goldringe
Liebeskind Berlin Schultertasche	Handtaschen|Ledertaschen|Schultertaschen
Guess Kids T-Shirt	Langarmshirts|T-Shirts

Tabelle 4.4: Beispiele für Produktnamen und Produkttypen

Marke

Die Produktmarke ist eine Markierung vom Hersteller. Der Hersteller versucht damit sich von konkurrierenden Herstellern abzugrenzen. Es lässt sich kein Produkttyp direkt ableiten. Jedoch ist es möglich bei bestimmten Herstellermarken einen Bereich abzugrenzen.

Samsung => Technik
Adidas => Sport
MAYBELLINE=> Kosmetik
...

Weitere Untersuchungen zeigen, dass alleine die Top 50 Produktmarken ca. 20% von allen Produkten in einem Durchschnitt von 3000 Produkten pro Marke unter sich aufteilen. Das ist ein Zeichen dafür, dass sich die Produktmarken gut für die Gruppierung von Produkten eignen. Das wurde im späteren Verlauf genauer untersucht und durch Experimente gezeigt (Vgl. Tabelle 5.2).

Verkaufsargumente

Die Verkaufsargumente sind eine Liste von den wichtigsten vertrieblichen Fakten. In der Tabelle 4.5 ist zu erkennen, dass es sich bei den Verkaufsargumenten um spezifische und relevante Produktinformationen handelt. Der Informationsgehalt ist sehr hoch. Alleine bei der ersten Betrachtung lässt sich daraus schließen, welche Produkttypen beschrieben werden. Leider wurde durch genauere Datenanalyse festgestellt, dass die Verkaufsargumente aus der

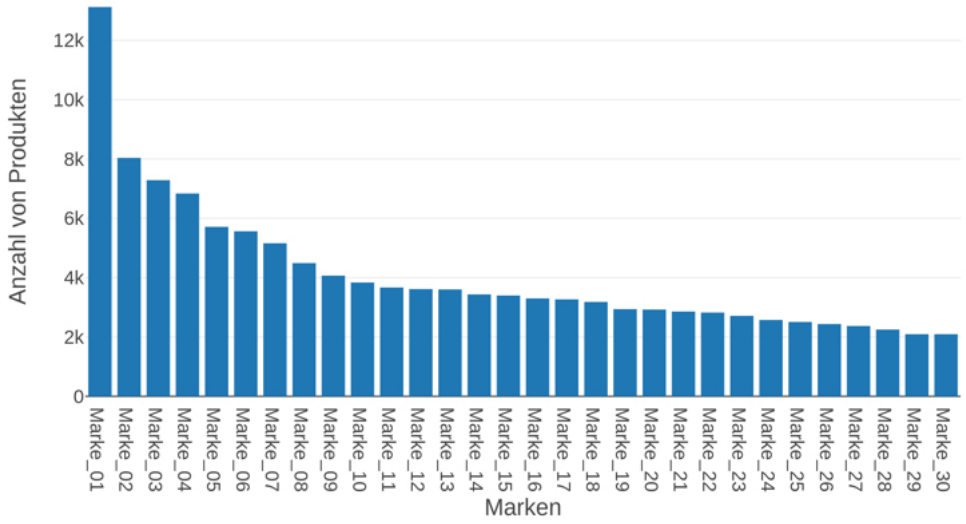

Abbildung 4.3: Top 30 Produktmarken Verteilung über alle Produkten

Produktbeschreibung abgeleitet sind. Trotzdem zeigen die Verkaufsargumente in Kombination mit dem Beschreibungstext gute Klassifikationsergebnisse, da sie durch die Anwendung von N-Gramme informationsreiche Merkmale ermöglichen.

Produkttypen	Verkaufsargumente
Diamantringe|Goldringe	- Prachtvoll eleganter Damenring - Aus teillweise rhodiniertem Gelbgold 585 - Feminine, zeitlose Eleganz!
Handtaschen|Ledertaschen|Schultertaschen	- Extras: Schlüsselhalter, Staubbeutel - Verschlussart: Reißverschluss - Außenmaße (LxBxH): 16cm x 6cm x 20cm - Volumen in L ca.: 0-10 - Anzahl Hauptfächer: 1

Tabelle 4.5: Beispiele von Verkaufsargumenten und dazu gehörigen Produkttypen

Beschreibungstext

Der Beschreibungstext besteht aus teilweise strukturierten Textabschnitten. Bei dem Großteil handelt es sich jedoch um unstrukturierte Texte. Es werden Produkteigenschaften oder technische Daten genauso wie visuelle Merkmale beschrieben. Die detailreiche Beschreibung ist nicht immer vorhanden. Es gibt Produkte, die mit einem Satz oder sogar mit nur wenigen Worten beschrieben werden. Im Beschreibungstext sind HTML-tags zu finden, was natürlich die Ergebnisse der vorläufige Datenanalyse beeinflusst. Nur 4% der Produktbeschreibungen bestehen aus 10 Worten oder weniger.

Produkttyp	Beschreibungstext
Staubsaugerbeutel	Weitere Vorteile: - Exakt angepasst an die jeweiligen Geräte- und Funktionsanforderungen. - Mit Hygiene-Klapp-Verschluss zur sicheren Entsorgung des vollen Beutel. Eigenschaften und Funktionen: Zubehör passend für: - HANSEATIC DUST MASTER 2000 - HANSEATIC VC-H5003 - UFESA AT 9220 Technische Daten: 15 Staubsaugerbeutel & #43; 3 Motorschutzfilter ... ...

Tabelle 4.6: Beispiele für den Beschreibungstext und dazu gehörigen Produkttyp

Auf Grund dessen, dass der Beschreibungstext das Produkt so gut wie möglich beschreiben sollte, um dem Käufer die richtige Auswahl beim Einkaufen zu ermöglichen, sollte es auch möglich sein auf Basis von Produktbeschreibungen die jeweiligen Produkttypen vorherzusagen. Auch das wurde durch ein Experiment belegt.

Bilder

Jedes Produkt besitzt mehrere Bilder. Es wurde für diese Arbeit nur ein Bild pro Produkt verwendet, was natürlich einen großen Nachteil für die Bildklassifikation darstellt. Um ein gutes Ergebnis bei der Bildklassifikation zu erhalten, wird eine ausreichende Menge von Trainingsdaten benötigt. Es werden z. B. bei dem Wettbewerb Imagenet[2] für 1000 Klassen 14

[2] http://image-net.org

Millionen Bilder zum Training verwendet, das ergibt durchschnittlich 1400 Bilder pro Klasse. Es wird in Vorverarbeitung von Bildern an diesem Problem gearbeitet, um trotz weniger Trainingsbilder ein ausreichendes Ergebnis zu bekommen.

Abbildung 4.4: Bilderbeispiele mit jeweiligen Produkttypen

Die Beispielbilder 4.4 zeigen, dass es für einen Mensch nicht immer möglich ist, den richtigen Produkttyp zu erahnen. Es liegt teilweise daran, dass auf den Bildern mehrere Produkte abgebildet sind.

4.1.3 Behandlung von fehlenden Werten

Die Untersuchung hat gezeigt, dass es notwendig ist, fehlende Werte zu behandeln. Wie die Tabelle 4.7 zeigt, handelt es sich um eine unterschiedliche Art und Verteilung fehlender Daten.

Wert	Prozentanteil
Produktyp	14,6%
Produktmarke	11,3%
Verkaufsargumente	0,73%
Bilder	0,48%
Beschreibungstext	0,05%
Name	0.0%

Tabelle 4.7: Anteil der Produkttypen Anzahl an der Gesamtmenge alle Produkte

Es existieren viele Methoden um die fehlenden Werte zu behandeln. In dieser Bachelorarbeit wurden nur zwei Methoden eingesetzt, die sich nach Art des Wertes unterscheiden. Bei den unerwarteten fehlenden Werten, bei dem ein Wert in der Realität existiert, ist aber in dem Datenset nicht vorhanden ist, wurden Entfernungen von Instanzen verwendet. Dementsprechend wurden alle Produkte ohne Produkttyp oder Produktbild entfernt. Bei den erwarteten fehlenden Werten, das sind die Werte, die nicht in der Realität existieren wird ein Signalwort hinzugefügt. Zum Beispiel, wenn die Produktmarke nicht existiert, wurde NoBrand als die Produktmarke eingegeben. Durch ein Signalwort wird deutlich gemacht, dass es um ein Produkt handelt, dass keine Herstellermarke besitzt.

4.1.4 Datenreduzierung

Die gewonnenen Erkenntnisse aus der Datenanalyse deuten darauf hin, dass in Bezug auf die Produkttypen stark unbalancierte Daten vorliegen. Aus folgenden Gründen wurde in dieser Arbeit eine Datenreduzierung durchgeführt:

- Die Lernverfahren liefern oftmals bei Klassifizierung und Validierung, insbesondere hinsichtlich wenig vertretener Klassen kein zufriedenstellendes Ergebnis.

- Es existieren Verfahren, die die Klassifizierung von unbalancierte Daten ermöglichen, solche zu untersuchen, würde den zeitlichen Rahmen dieser Bachelorarbeit sprengen.
- Um die Trainings- und Testzeiten von einzelnen Lernverfahren zu reduzieren, um viele Optimierungsmöglichkeiten in Bezug auf die Zielstellung dieser Arbeit experimentell zu belegen.

Aufgrund der Verteilung der Produkttypen wurden alle Produkte mit den Produkttypen genommen, die öfter als 100 mal und weniger als 500 mal vorkommen. Als Ergebnis daraus wurden im weiteren Verlauf dieser Arbeit zum Vergleich der Performance der verschiedenen Lernverfahren und Klassifikationsmodelle 225 421 Produkte mit 1257 Produkttypen untersucht.

4.2 Datenvorverarbeitung

Nachdem Fokussieren der Zieldaten, also der Beschaffung der Daten und Auswahl der relevanten Daten auf Grundlage der Datenanalyse, folgt der nächste Schritt, die Datenvorverarbeitung (siehe Abbildung 4.1). In dem Vorverarbeitungsprozess werden die Daten so aufbereitet, dass bei dem nächsten Schritt die extrahierten Informationen möglichst bereinigt und einheitlich sind.

4.2.1 Bildvorverarbeitung

Die Bildvorverarbeitung dient dazu, im Bild enthaltene Informationen, für die weitere Mustererkennung, um die Bilder zu klassifizieren, hervorzuheben. Es werden alle Bilder auf eine Größe von 299x299 Pixel komprimiert. Dabei wird eine gewisse Information verloren gehen, ermöglicht aber die Verwendung eines vortrainierten CNN (gemäß Chollet. (2017)). Auf Grund dessen, dass für die Merkmalsextrahierung ein vortrainiertes CNN verwendet wird, das bereits unterschiedliche Filter enthält, wurde in dieser Arbeit auf weitere Bildvorverarbeitung verzichtet.

Um das Problem von nicht ausreichenden Bildbeispielen zu umgehen und Überanpassung zu verhindern, wurde die Funktionalität von dem ImageDataGenerator der Bibliothek Keras [3] erweitert um Bilder für die Multi-Label-Klassifikation zu generieren. Es werden zufällige Bilder

[3] https://keras.io

mit folgenden Parametern generiert:
Streckung: -0.08% bis 0.08%
Verschiebung nach links: 0.8%
Verschiebung nach rechts: 0.8%
Drehung: -8° bis +8°
Spiegelung: vertikal

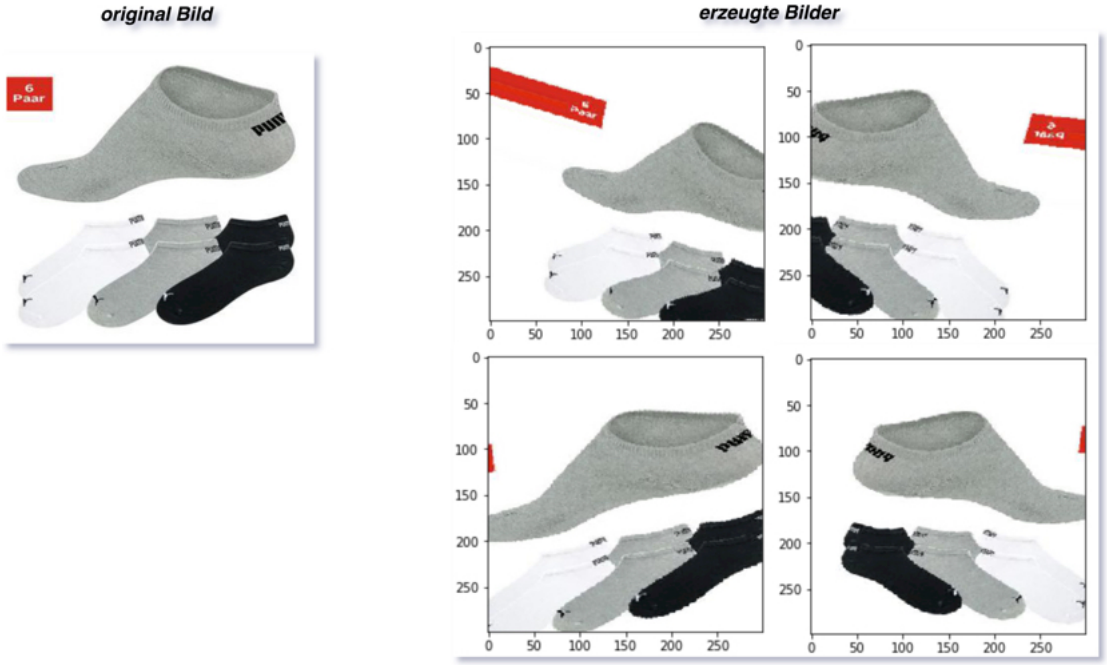

Abbildung 4.5: Bildererzeugung

In der Abbildung 4.5 sind Beispiele für automatisch erzeugte Bilder zu sehen.

4.2.2 Textvorverarbeitung

Eine der wichtigsten Optimierungsmöglichkeiten ist die Vorverarbeitung von Texten. Die Produktklassifikationsleistung des Modells kann nur mit vorverarbeiteten Beschreibungstexten gute Ergebnisse zeigen. Ein Produktbeschreibungstext hat viele störende und irrelevante Elemente, diese müssen gefunden und behandelt werden. Es wurde gezeigt, dass sich Vorhersagen

mit den vorgeschlagenen und untersuchten Methoden deutlich verbessern, wie die Ergebnisse in der Tabelle 5.1 und 5.2 zeigen.

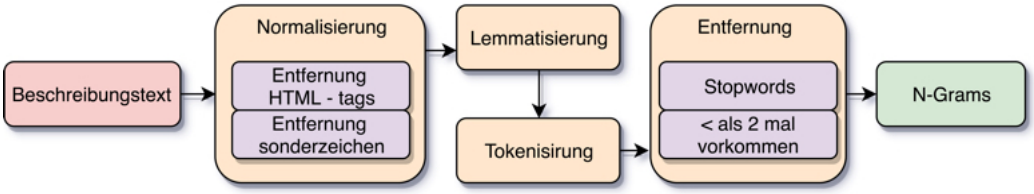

Abbildung 4.6: Ablauf Textvorverarbeitung

Im Abbildung 4.6 sind die einzelne Schritte des Ablaufs dargestellt. Als Erstes wird der Produktbeschreibungstext normalisiert.

Beispieltext: *Anhänger 'Kreuz' aus 14 Karat (585) Weißgold mit einem Diamant-Brillanten, 0,015 ct., W/SI wesselton (weiß), SI=(kleine, mit bloßem Auge nicht erkennbare natürliche Einschlüsse), Feinbearbeitung gut
Höhe ca. 22,4 mm, Breite ca. 13,3 mm, Tiefe ca. 2,1 mm, Innenmaße der Öse ca. 4,1 mm x 2,8 mm, Gewicht ca. 2,0 g
Bitte beachten Sie die Maße! Auf dem Foto kann der Artikel größer wirken*

Normalisierung

Es werden alle HTML-tags und die störende Sonderzeichen entfernt, sodass sich der Wiedererkennungswert von einzelnen Beschreibungen verbessert. Die genaue Untersuchung von Ergebnissen hat gezeigt, dass weitere Normalisierungsschritte möglich sind um das Klassifikationsmodell zu optimieren. Aus zeitlichen Gründen wurde darauf verzichtet. Es wurden alle Satzzeichen bis auf den Punkt entfernt.

Beispieltext normalisiert: *Anhänger Kreuz aus 14 Karat 585 Weißgold mit einem Diamant Brillanten 0 015 ct. W SI wesselton weiß SI kleine mit bloßem Auge nicht erkennbare natürliche Einschlüsse Feinbearbeitung gut Höhe ca. 22 4 mm Breite ca. 13 3 mm Tiefe ca. 2 1 mm Innenmaße der Öse ca. 4 1 mm x 2 8 mm Gewicht ca. 2 0 g Bitte beachten Sie die Maße Auf dem Foto kann der Artikel größer wirken*

Lemmatisierung

Der nächste Vorverarbeitungschritt ist die Lemmatisierung. Dabei werden alle Wörter in Grundform umgewandelt. Für diesen Zweck wurde in der Arbeit die Specy[4] Bibliothek verwendet. Genauso wird kein Unterschied zwischen Groß- und Kleinschreibung gemacht. Dadurch geht gewisse Information verloren. Aber durch Wiedererkennungswert werden mehr Produkttypen richtig klassifiziert (Vgl. Tabelle 5.2).

Beispieltext lemmatisiert: *anhänger kreuzen aus 14 karat 585 weißgold mit einer diamant brillanten 0 015 ct . w si wesselton weiß si kleine mit bloß auge nicht erkennbar natürlich einschlüsse feinbearbeitung gut höhe ca. 22 4 mm breiten ca. 13 3 mm tief ca. 2 1 mm innenmaße der öse ca. 4 1 mm x 2 8 mm gewicht ca. 2 0 g bitten beachten ich der maßen auf der foto können der artikel groß wirken*

Tokenisierung

Bei der Tokenisierung wurde ein zusammenhängender Text in seine Einzelteile zerlegt. Die Texte werden in Sätze oder in einzelne Wörter zerlegt. Der Tokenisierungsschritt, der in der Abbildung 4.6 gezeigt ist, zerlegt die lemmatisierten Sätze in die einzelnen Wörter.

Beispieltext tokenisiert: *'anhänger', 'kreuzen', 'aus', '14', 'karat', '585', 'weißgold', 'mit', 'einer', 'diamant', 'brillanten', '0', '015', 'ct', '.', 'w', 'si', 'wesselton', 'weiß', 'si', 'kleine', 'mit', 'bloß', 'auge', 'nicht', 'erkennbar', 'natürlich', 'einschlüsse', 'feinbearbeitung', 'gut', 'höhe', 'ca.', '22', '4', 'mm', 'breiten', 'ca.', '13', '3', 'mm', 'tief', 'ca.', '2', '1', 'mm', 'innenmaße', 'der', 'öse', 'ca.', '4', '1', 'mm', 'x', '2', '8', 'mm', 'gewicht', 'ca.', '2', '0', 'g', 'bitten', 'beachten', 'ich', 'der', 'maßen', 'auf', 'der', 'foto', 'können', 'der', 'artikel', 'groß', 'wirken'*

Stopwords Entfernung

Stopwörter sind die Wörter, die keine relevante Information für die Klassifizierung beinhalten. Aus diesem Grund können diese gelöscht werden. Stopwörter zu entfernen führt nicht nur zur Dimensionsreduzierung, sondern es entstehen in Verbindung mit der Benutzung von von N-Grammen neue, für die Klassifikation wichtige Tokens, die zur einer Verbesserung

[4] https://spacy.io

der Klassifikationsleistung führt. Wie die Ergebnisse in der Tabelle 5.1 zeigen, gibt es zwei Vorgehensweisen bei der Entfernung von Stopwörtern: Wörter, die am häufigsten vorkommen oder Wörter, die in eine Liste vorgegeben sind zu löschen. Die erste Methode ist schneller zu realisieren, kann aber zu einer Verschlechterung der Vorhersagen des Models führen, da Wörter die oft vorkommen aber trotzdem informativ bezüglich Vorhergesagten Klassen, entfernt werden. Die zweite Methode ist individuell anpassbar, was auch als gute Optimierungsmöglichkeit angesehen werden kann. Zuerst kann die Wortliste beliebige Wörter und Wortkombinationen beinhalten, unabhängig davon wie oft es in Texten vorkam. Zweitens lässt es sich einfach überprüfen, ob sich das Entfernen von einzelnen Wörtern positiv auf die Klassifikationsleistung auswirkt. Auf diese Weise könnte eine Stopwortliste kreiert werden, die bei Produkttypenklassifizierung nicht nur zur Dimensionsreduzierung führt sondern auch die Klassifikationsleistung steigert, wie die Ergebnisse in der Tabelle 5.2 zeigen. Es werden zusätzlich Wörter entfernt die nur einmal in den Trainingstexten vorkommen da die Wahrscheinlichkeit, dass solche Wörter in den Testdaten auftauchen gering ist. In dieser Arbeit wurde die Stopwordlist der Spacy[5] Bibliothek verwendet.

Beispieltext ohne Stopwörter: *'anhänger', 'kreuzen', '14', 'karat', '585', 'weißgold', 'diamant', 'brillanten', '0', '015', 'ct', '.', 'w', 'si', 'wesselton', 'weiß', 'si', 'bloß', 'auge', 'erkennbar', 'einschlüsse', 'feinbearbeitung', 'höhe', 'ca.', '22', '4', 'mm', 'breiten', 'ca.', '13', '3', 'mm', 'tief', 'ca.', '2', '1', 'mm', 'innenmaße', 'öse', 'ca.', '4', '1', 'mm', 'x', '2', '8', 'mm', 'gewicht', 'ca.', '2', '0', 'g', 'bitten', 'beachten', 'maßen', 'foto', 'artikel', 'wirken'*

N-Grams

Nicht nur einzelne Wörter sind Informationsträger. Auch die Wortreihenfolge spielt eine große Rolle. Es besteht ein gravierender Unterschied zwischen den Sätzen: *es ist bruchsicher aber nicht wasserfest* und *es ist wasserfest aber nicht bruchsicher*. Ein Nachteil von N-Grams ist, dass sie dadurch die Anzahl von Dimensionen erhöhen. Die Experimente haben gezeigt, dass es wie erwartet, zu einer Verbesserung der Klassifikationsleistung führt (Vgl. Tabelle 5.2).
Beispieltext 2-Grams: *..., 'anhänger kreuzen', 'kreuzen 14', '14 karat', 'karat 585', '585 weißgold', 'weißgold diamant', 'diamant brillanten', 'brillanten 0', '0 015', '015 ct', 'ct .', '. w', 'w si', ...*

[5] https://spacy.io

4.3 Transformation

Nach der Vorverarbeitung der Produktdaten, wie in Abbildung 4.1 gezeigt, werden diese transformiert, um das Training mit Lernalgorithmen zu ermöglichen.

4.3.1 Bild-Transformation

In dieser Bachelorarbeit wurden die Merkmale aus den Bildern mit der Hilfe des vortrainierten CNN extrahiert; dabei wurde entschieden, das Netz zu nehmen, das sehr gute Ergebnisse in dem Wettbewerb ImageNet[6] gezeigt hat. Die detaillierte Beschreibung des Aufbaus und der Funktionsweise des Xception CNN ist in der Arbeit Chollet. (2017) zu finden. Wichtig für vorliegende Bachelorarbeit ist, dass das CNN nicht komplett verwendet wurde, sondern nur die drei Abschnitte des Merkmal extrahierenden Teils. Diese sind in der Abbildung 4.7 grau gezeigt: Entry flow, Middle flow und Exit flow.

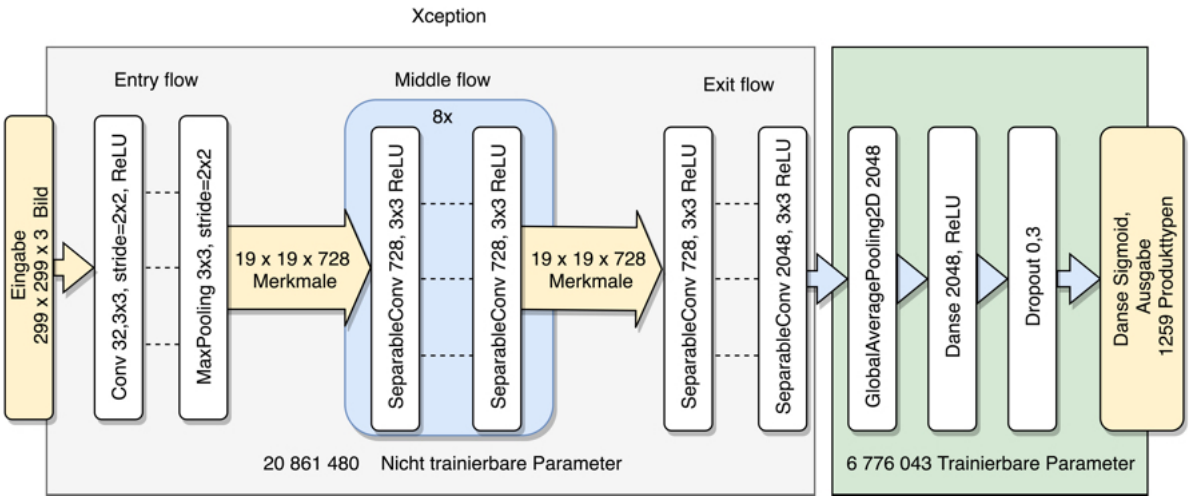

Abbildung 4.7: Bildklassifikationsmodell

Der Abschnitt, der für die Klassifikation von Bilder verantwortlich war, wurde mit dem neuen untrainirtem KNN ersetzt, dessen Aufbau Abbildung 4.7 grün dargestellt ist. So wurde ein CNN konstruiert, das insgesamt 27.637.523 Parameter hat, obwohl der Merkmal extrahierende Teil eingefroren wurde. Das bedeutet, die Gewichte im Netz werden durch Training nicht weiter verändert. Nur der Klassifikationteil mit 6.776.043 Parametern wird trainiert. Dadurch wird die

[6] http://image-net.org

Trainingszeit reduziert. Die Bilder werden zu einem dreidimensionalen Array transformiert (299,299,3). Die erste und zweite Position sind die Koordinaten. Die dritte ist mit einem von drei RGB Kanälen belegt. Danach werden alle Werte in dem Bereich zwischen 1 und 0 normalisiert. Dies beschleunigt den Trainingsprozess des KNN.

4.3.2 Text-Transformation

Für eine Vektordarstellung von Texten werden beide Standardverfahren eingesetzt: Bag of Words und eine gewichtete Darstellung als TfIdf. Beide Transformationen wurden mit Hilfe der scikit-learn[7] Bibliothek realisiert.

4.3.3 Dimensionsreduzierung

Um die Berechnungszeiten für das Training und Vorhersagen des Models zu reduzieren, werden auf die transformierten Daten Dimensionsreduktionsalgorithmen angewendet. Es wird eine kompaktere Repräsentierung der Daten gesucht. In der Arbeit wird mit Hilfe von x^2 eine Statistik berechnet, wie die einzelnen Merkmale mit der Klasse korrelieren. Als Ergebnis daraus wird die Dimension eines Produktbeschreibungsvektors gesenkt. Es wurde festgestellt, dass die Reduzierung der verfügbaren Merkmale zu kürzeren Trainingszeiten führt. Weitere Experimente in der Tabelle 5.2 zeigen, dass eine geringe Verschlechterung der Klassifikationsleistung entsteht.

4.4 Data Mining

Nachdem die Daten vorverarbeitet und transformiert wurden, wird in diesem Unterkapitel die Architektur und Konfiguration der eingesetzten Klassifikationsmodelle und Bewertungsverfahren präsentiert.

[7] https://scikit-learn.org

Architektur Textklassifikationsmodell

Aufgrund dessen, dass in den verwandten Arbeiten KNN sich als erfolgreich erwies, wurde auch in dieser Arbeit KNN für Text- und Bildklassifikation eingesetzt. Alle Textklassifikationsmodelle, die in dieser Arbeit verwendet und auf KNN basieren, sind wie folgt aufgebaut:

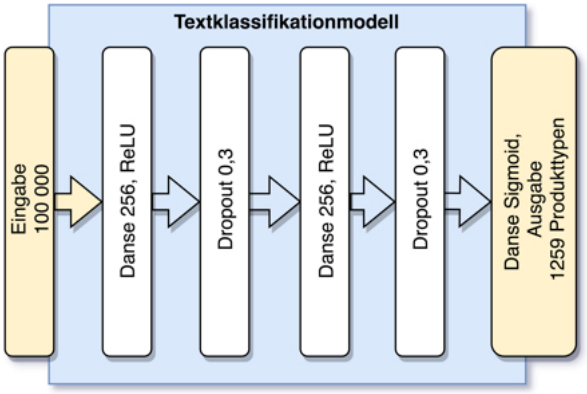

Abbildung 4.8: Textklassifikationsmodell

Die Anzahl der Neuronen variiert nach Text- und Transformationsart. Es werden drei Schichten verwendet, die mit einem Dropout von 30% verbunden wurden, um der Überanpassung entgegen zu wirken. Die Experimente haben gezeigt, dass die KNN in dem Vergleich zu den Anderen in dieser Arbeit eingesetzten Klassifikationsverfahren eine bessere Klassifikationsleistung haben(Vgl. Tabelle 5.3).

Architektur Bildklassifikationsmodell

Genau so wie die Text- wurde auch die Bildklassifikation mit der Hilfe von KNN realisiert. Der Aufbau wurde unter Verwendung von Erfahrungswerten und unter Berücksichtigung verwandter Arbeiten realisiert. Die einzelnen Layer und Aktivierungsfunktionen können aus der Abbildung 4.7 entnommen werden. Für die Klassifikation ist der grün gekennzeichnete Modellabschnitt zuständig.

4.4.1 Ensemble Methoden

Die Ensemble Methoden werden in dieser Arbeit für die Verbesserung der Klassifikationsleistung eingesetzt. Es werden mehrere, am besten auf unterschiedliche Arten, Datenmengen oder Merkmale trainierte Modelle zu einem komplexen Gesamtmodell kombiniert, dadurch wird eine bessere und stabilere Klassifikationsleistung ermöglicht. Die Kopplung von vortrainierten Modellen kann auf unterschiedliche Art stattfinden, es können dazu beliebige Klassifikationsverfahren eingesetzt werden. In dieser Arbeit wurden zuerst die zwei Standardverfahren Soft- und Hard-Voting-Classifier untersucht. Weiterhin wurden mit KNN die vortrainierten Modelle verbunden. Abbildung 4.9 zeigt, wie beide vortrainierten Modelle mit der Hilfe von KNN verbunden wurden. Es wird damit ein Beispielaufbau präsentiert. Auf diese Art wurden auch die anderen Ensemblemodelle aufgebaut. Wie erwartet wurden Verbesserungen in Bezug auf die Klassifikationsleistung beobachtet. Die daraus resultierenden Ergebnisse werden in dem Kapitel 5: Experimente und Ergebnisse präsentiert (Vgl. Tabelle 5.4).

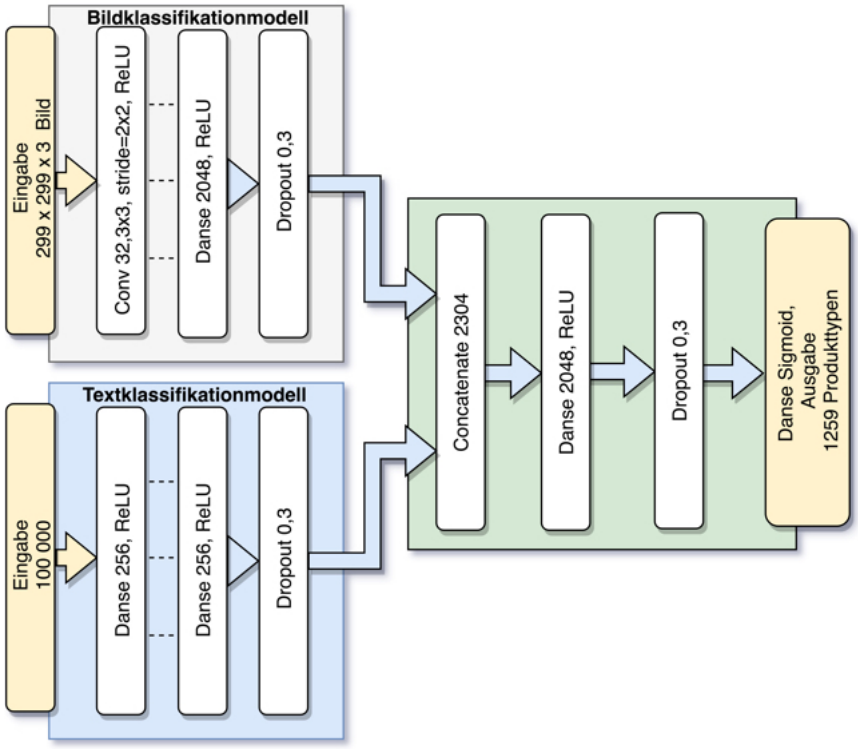

Abbildung 4.9: Ensemblemodell

4.5 Evaluation

Zur Auswertung der Klassifikationsmodelle wurden zwei unterschiedliche Verfahren eingesetzt. Die Holdout-Methode für die Bewertung der Klassifikationsleistungen und die K-fold-Kreuzvalidierungs-Methode beim Modelltraining für die Validierung der Modelle und die Optimierung der Parameter.

4.5.1 Die Holdout-Methode

Bei der Holdout-Methode werden alle relevanten Daten in zwei disjunkte Teilmengen aufgeteilt und zwar in Trainings- und Testmenge. Die Testmenge wird auch Holdout-Menge genannt. Die Trainingsmenge wird zum Modelltraining und zur Optimierung der Parameter verwendet. Die Testmenge wird nur zum Überprüfen der Klassifikationsleistung des Models verwendet und wird während der Trainingsphase und Optimierungsphase nicht benutzt. Das Holdout Verfahren soll vor einer Überanpassung des Models an die Daten schützen und auf Grund des Grades der Übereinstimmung zwischen vorhergesagten und tatsächlichen Labels wird in dieser Arbeit die Bewertung des Klassifikationsmodells gemacht. Die Mengen werden mit Hilfe von scikit-learn[8] Bibliothek zufällig automatisiert in 80% Trainingsmenge und 20% Testmenge aufgeteilt.

4.5.2 K-fold-Kreuzvalidierungsmethode

Zwar wird die Holdout-Methode bei der Validierung des Modells verwendet, also die Trainingsmenge wird in Validierungs- und Trainingsmenge aufgeteilt, aber eine solche Aufteilung hat auch einen Nachteil: Er besteht darin, dass die Klassifikationsleistung nur auf eine bestimmte Testmenge von ausgewählten Beispielprodukten gemessen wird. Die Qualität variiert für unterschiedliche Stichproben der Daten. Dadurch besteht die Gefahr für Überanpassung des Modells. Aus diesem Grunde wurde in dieser Arbeit für die Modellauswahl und Optimierung eine robustere Technik für die Messung der Klassifikationsleistungen eingesetzt. Bei der K-fold-Kreuzvalidierungsmethode wird die Holdout-Methode 5 mal auf 5 Untermengen der Trainingsdaten wiederholt. Wie in der Abbildung 4.10 zusehen ist, wurden alle Trainingsdaten zufällig automatisiert und in fünf gleichgroße Teile aufgeteilt. Es wurden vier Trainingsmengen

[8] https://scikit-learn.org

für das Modelltraining und eine Trainingsmenge zum Testen verwendet. Der Vorgang wurde 5 mal wiederholt. So wurden fünf Modelle trainiert. Dann wurde die durchschnittliche Leistung der Modelle berechnet, die auf den unterschiedlichen, unabhängigen Testmengen basiert. Die dabei durch Optimierung gefundenen Hyperparameterwerte wurden, dann zum Training des Modells auf die vollständigen Trainingsdaten verwendet.

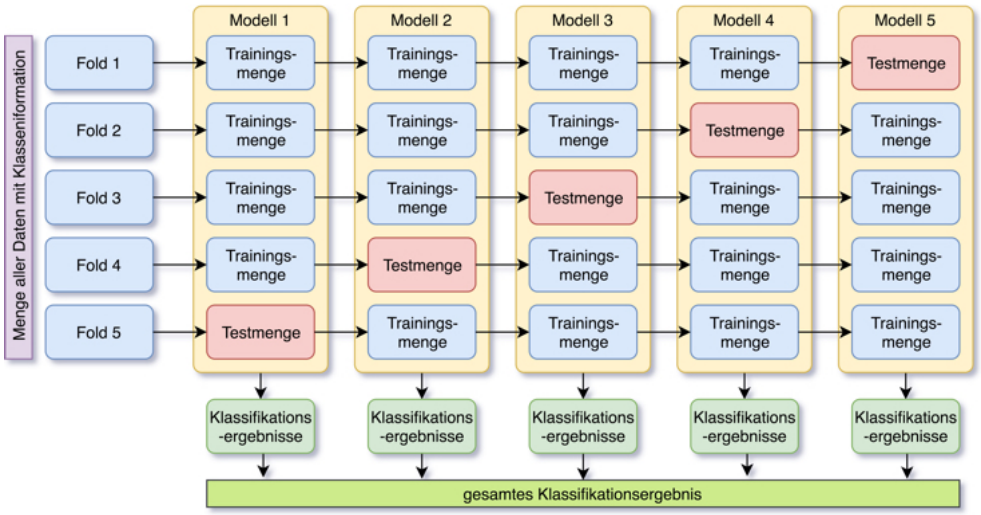

Abbildung 4.10: 5-fold-Kreuzvalidierungsmethode (vgl. Richert (2013) , S.38); eigene Darstellung

4.6 Zusammenfassung

Im diesem Kapitel wurden alle eingesetzten Schritte zum Trainieren des Klassifikationsmodells erläutert und begründet. Es wurden Optimierungsmöglichkeiten untersucht und belegt. Im folgenden Kapitel werden Trainingsabläufe, die aus diesen Schritten bestehen und daraus resultierende Ergebnisse präsentiert.

5 Experimente und Ergebnisse / Auswertung

Im folgenden Kapitel werden die verwendeten Bibliotheken vorgestellt. Danach werden die Trainingsabläufe anhand eines Datenflussdiagramms erklärt. Genauso werden die Experimente beschrieben und die daraus resultierenden Ergebnisse ausgewertet.

5.1 Externe Programme und Hilfsmittel

NLTK

NLTK[1] oder auch Natural Language Toolkit ist eine Sammlung von Hilfsmitteln zur Verarbeitung natürlicher Sprache in der Programmiersprache Python. Es enthält Bibliotheken für die Segmentierung in Sätze und Wörter sowie auch das Part-of-Speech Tagging.

Spacy

Spacy[2] ist eine Open Source Softwarebibliothek für die natürliche Sprachverarbeitung, geschrieben in den Programmiersprachen Python und Cython. Die Bibliothek bietet derzeit statistische neuronale Netzwerkmodelle für Englisch, Deutsch, Spanisch, Portugiesisch, Französisch, Italienisch, Niederländisch und mehrsprachig NER, sowie Tokenisierung für verschiedene andere Sprachen an.

[1] https://www.nltk.org
[2] https://spacy.io

Scikit-learn

Scikit-learn[3] ist eine Open Source Bibliothek für Python zum maschinellen Lernen. Scikit-learn bietet unterschiedliche Klassifikationsalgorithmen, unter anderen Support-Vektor-Maschinen, k-means und Random Forest. Die Bibliothek bietet auch viele nützliche Hilfsmittel für die Bewertung von Klassifikationsalgorithmen.

Keras

Keras[4]ist eine Open Source Deep-Learning-Bibliothek für Python. Keras hat Hilfsmittel für Vorverarbeitung von Text und Bildklassifikation. Keras bietet eine einheitliche Schnittstelle für verschiedene Backends, unter anderem TensorFlow und Theano. Die Bibliothek ist in Python geschrieben, was eine einfache Erweiterung und Modifizierung ermöglicht.

Draw.io

Draw.io[5] ist eine Open Source webbasierte Anwendung zur Erstellung von Diagrammen. Alle Abbildungen in dieser Arbeit wurden mit draw.io gezeichnet.

5.2 Ablauf

In Kapitel 4 wurden alle einzelnen Schritte, genauso wie daraus resultierende Optimierungsmöglichkeiten zu dem gesamten Ablauf beschrieben. In der Abbildung 5.1 wurden die Schritte in einem Datenflussdiagramm präsentiert. Nach diesem Ablauf wurden auch die einzelnen Modelle trainiert. So ermöglicht ein Ablauf, die einzelnen Modelle parallel zu trainieren. Das führt zur Reduzierung der Trainingszeiten. Das KNN wurde in dieser Arbeit unter Verwendung der Keras[6] Bibliothek entwickelt. Auf Grund dessen, dass Keras im Backend das Tensorflow Framework von Google verwendet, ist es möglich alle damit entwickelten KNNs auf Grafikprozessoren zu trainieren, das führte zu einer zwölffachen Beschleunigung des Trainings. Die

[3] https://scikit-learn.org
[4] https://keras.io
[5] https://www.draw.io
[6] https://keras.io

Modelltrainingszeiten variierten stark, bei der Verwendung von unterschiedlichen Modellarchitekturen und den gewählten Hyperparametern. Zum Beispiel wurden für das Training von Modell_Bild im Durchschnitt auf einem Prozessor pro Epoche 16,4 Stunden benötigt. Im Vergleich dazu dauerte es nur 1,3 Stunden pro Epoche auf einer GPU. Um aussagekräftige Ergebnisse zu präsentieren, wurden für diese Arbeit zahlreiche Experimente durchgeführt. Die 100% Klassifikationsleistung zu erreichen war nicht das Ziel dieser Arbeit, trotzdem waren viele Experimente erforderlich, um geeignete Hyperparameterwerte zu einzelnen Modellen zu finden. Die Experimente in dieser Arbeit wurden unter Verwendung einer Nvidia GTX1070Ti mit 8GB Grafikspeicher, 32GB RAM und einem Intel Core i7-8700k Prozessor durchgeführt.

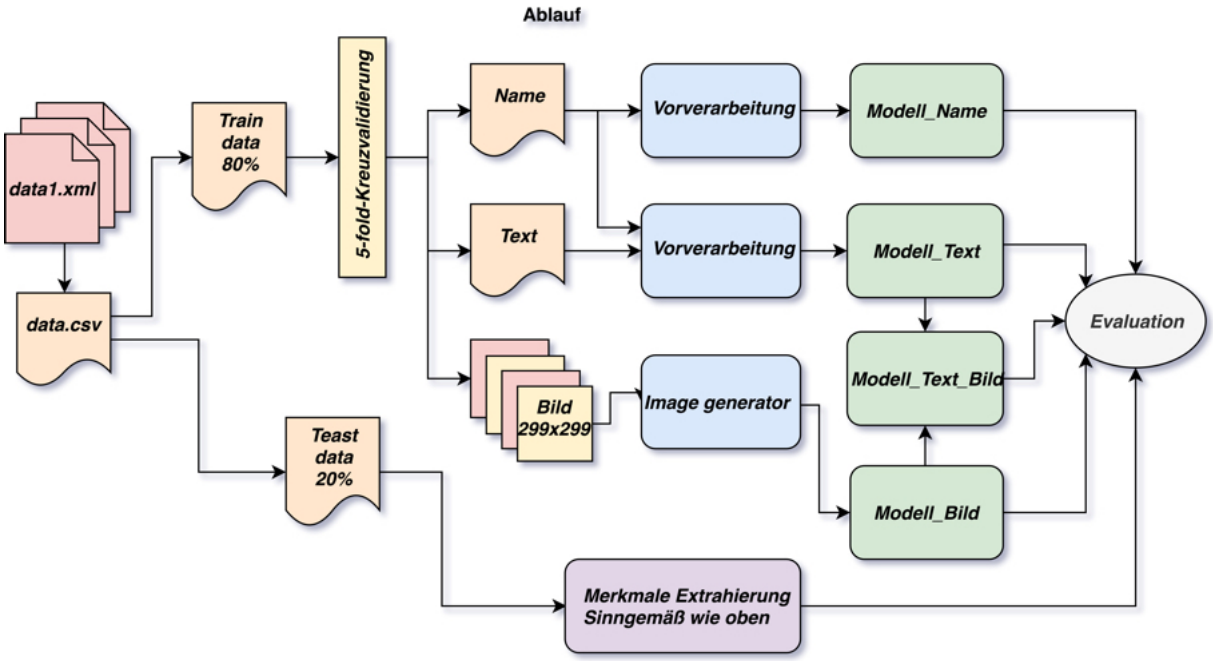

Abbildung 5.1: Trainingsablauf

5.3 Experimente

In Kapitel 4 wurden die einzelnen Schritte der Vorverarbeitung beschrieben. Einige dieser Schritte wurden experimentell evaluiert, die Ergebnisse dieser Experimente sind in den Tabellen 5.1 und 5.2 präsentiert.

5.3.1 Produkttypenklassifikation Vorverarbeitung

Normalisierung:
h - Entfernung HTML-tags k - Kleinschreibung
s - Entfernung Sonderzeichen
sw - Entfernung Stopwörter
n-2 - Verwendung 2-Grams
l -Lemmatisierung

Name

Klassifikationverfahren Normalisierung	SVM ohne	SVM k	SVM s	SVM s,n-2
Hamming loss	0.0002940	0.0002953	0.0002879	0.0002780
Precision	0.7791	0.7787	0.7857	0.7960
Recall	0.7749	0.7744	0.7810	0.7943
F1- Measure	0.7709	0.7704	0.7773	0.7891
Subset_accuracy	0.7209	0.7198	0.7273	0.7393
Accuracy	0.7587	0.7580	0.7650	0.7769

Tabelle 5.1: Name Vorverarbeitung

Es wurden die einzelnen Schritte der Vorverarbeitung von Produktnamen in dem Unterkapitel 4.1.2 erläutert. Die Ergebnisse zeigen, dass die Klassifikation von Produkttypen auf Grund der Beschaffenheit von Produktnamen gut funktioniert. Trotzdem ist es als instabil zu bezeichnen, da weitere Untersuchungen gezeigt haben, dass sie wie erwartet bei der Dimensionsreduktion von 10% eine Genauigkeit von 63.45% hat. Die einzelnen Wörter sind informativ und sobald

sich die Struktur von Produktnamen verändert, verschlechtert sich auch die Klassifikationsleistung. Eine weitere Untersuchung zeigte, dass nicht alle Online-Shops den Produkttyp im Produktnamen besitzen, also um das Klassifikationssystem zu verallgemeinern, werden die Produktnamen in den Produkttext einfließen.

Produkttext

Der Produkttext besteht aus Produktnamen, Produktmarke, Verkaufsargumenten und Beschreibungstext. In der Tabelle 5.2 werden einzelne Vorverarbeitungsschritte von Produkttexten präsentiert. Wie erwartet steigt mit jedem weiteren Schritt die Klassifikationsleistung. Die einzelnen Schritte sind nicht an Beispieldaten angepasst, sondern wurden allgemein gehalten. Trotzdem sollte die Vorverarbeitung am jeweiligen Datenset optimiert werden, um eine stabilere und höhere Klassifikationsleistung zu bekommen, wie in Kapitel 4 schon erklärt wurde.

Klassifikationverfahren Normalisierung	SVM ohne	SVM h,k	SVM h,k,s	SVM h,k,s,sw	SVM h,k,s,sw,l
Hamming loss	0.0002837	0.0002632	0.0002580	0.0002571	0.0002560
Precision	0.7702	0.7912	0.8006	0.8024	0.8043
Recall	0.7598	0.7819	0.7920	0.7936	0.7957
F1- Measure	0.7597	0.7814	0.7909	0.7925	0.7944
Subset_accuracy	0.7160	0.7379	0.7459	0.7466	0.7479
Accuracy	0.7491	0.7708	0.7800	0.7814	0.7831

Tabelle 5.2: Produkttext Vorverarbeitung

Vergleiche beim Training

Nachdem der Produkttext vorverarbeitet wurde, zeigte SVM eine Klassifikationsgenauigkeit von 78,31% , Subset_accuaracy von 74,79%, ohne große Optimierungen von Vorverarbeitung und Hyperparametern. Auch wäre es wichtig zu wissen, ob eine Dimensionsreduktion beim Beschreibungstext die Klassifikationsleistung stark beeinflusst. Es wurden für die weitere Klassifikation nur 6,66% der erzeugten Merkmale benutzt. Die Auswertungen der Ergebnisse belegen, dass wie erwartet die Genauigkeit nur um 3,05% sinkt. Zusammenfassend lässt sich sagen, dass das Klassifikationsmodell basierend auf Produkttext auch ohne Optimierungen schon

eine gewisse Stabilität und Qualität aufweist. Im nächsten Schritt werden unterschiedliche Verfahren verglichen. Durch die Auswertung der Ergebnisse konnte die These der untersuchten Arbeiten, dass KNN für diese Aufgabe besser geeignet ist als andere Lernalgorithmen, bestätigt werden. Das untersuchte KNN erreichte eine Klassifikationsgenauigkeit von 83,42%. Anhand dieses Beispiels wird deutlich, dass KNN im Verbund mit anderen Modellen untersucht werden sollte.

Klassifikationverfahren Normalisierung Dimensionsreduktion	SVM h,k,s,sw,l -	SVM h,k,s,sw,l 100 000	KNN h,k,s,sw,l 100 000	RF h,k,s,sw,l 100 000
Hamming loss	0.0002560	0.0002825	0.0002657	0.0003155
Precision	0.8043	0.7744	0.8496	0.7593
Recall	0.7957	0.7652	0.8772	0.7526
F1- Measure	0.7944	0.7641	0.8539	0.7498
Subset_accuracy	0.7479	0.7169	0.7735	0.6994
Accuracy	0.7831	0.7526	0.8342	0.7375

Tabelle 5.3: Vergleiche beim Training

5.3.2 Untersuchung der Verbindungsarten

Die Ergebnisse der Verbindungsartuntersuchung sind in der Tabelle 5.4 präsentiert. Alle getesteten Verbindungsarten zeigen eine Verbesserung der Klassifikationsleistung. Daraus ergibt sich, dass die Ensemble Methoden dafür geeignet sind, Stabilität und Leistung eines Modells zu steigern. Durch Experimente mit Bildklassifikationen wurde gezeigt, dass auch ein schwaches Modell in der Lage ist, durch eine geeignete Verbindungsart ein starkes Modell zu verbessern. Anhand dieser Ergebnisse wird deutlich, dass alle gewählten Verbindungsarten geeignet sind, Produkttypen zu klassifizieren. Die beiden Voting Methoden sind schnell realisierbar und müssen nicht trainiert werden. Diese können durch Gewichtung einzelner Modelle oder durch Schwellenwertanpassung bei einzelnen Modellvorhersagen weiter optimiert werden. Im Vergleich zu den beiden Voting Methoden muss eine Verbindung durch KNN trainiert werden. Zusammenfassend lassen sich folgende Ergebnisse herausstellen, die Verbindung durch KNN zeigt die beste Klassifikationsleistung und wird in dieser Arbeit als Produkttypenklassifikator gewählt. Es werden weitere Optimierungen notwendig, um das Modell in einem Online-Shop

einzusetzen. Alle Ergebnisse der untersuchten Optimierungsmöglichkeiten deuten darauf hin, dass die vorgeschlagene und untersuchte Modellarchitektur viel Potenzial aufweist.

Klassifikationverfahren Verbindungsart Normalisierung	CNN - Bilder	KNN+CNN KNN Bilder+Text	SVM+CNN+KNN soft Voting Bilder+Text	SVM+CNN+KNN hard Voting Bilder+Text
Hamming loss	0.0005724	0.0002266	0.0002186	0.0002281
Precision	0.5578	0.8917	0.8517	0.8650
Recall	0.6956	0.9340	0.8583	0.8896
F1- Measure	0.5945	0.9016	0.8483	0.8687
Subset_accuracy	0.4307	0.8096	0.7916	0.7965
Accuracy	0.5455	0.8790	0.8345	0.8510

Tabelle 5.4: Untersuchung der Verbindungsarten

5.3.3 Fehleruntersuchung

Textklassifikation

Die Auswertung der Ergebnisse in der Tabelle 5.3 zeigen, dass KNN mit der Vorverarbeitung der Produkttexte eine Accuaracy von 83,42% hat. Die genauere Untersuchung der anderen 16,58% ergeben, dass es sich hauptsächlich um drei Fehlerarten handelt.

- Durch Dimensionsreduktion sind relevante Information verloren gegangen, so dass es nicht mehr möglich ist, einen Produkttyp eindeutig zu klassifizieren.

  Lösungsvorschlag: durch Verbindung von Bild- und Textklassifikation wurde die Klassifikationsleistung erhöht, wie die Tabelle 5.8 zeigt und in der Tabelle 5.5 gezeigte Beispiele wurden richtig klassifiziert

Produktname	Erzeugte Merkmale	Produkttyp	Vorhersage
Bilderwelten Vliestapete Quer »Tosende Wellen«	welle, vliestapete, quer, bilderwelten vliestapete, bilderwelten	Fototapeten Meer	Kindertapeten
IDEALDECOR Fototapete »Architektur weißes Hochhaus«, Vlies, 2 Bahnen, 183 x 254 cm	x 254, x, weiß, vlies 2, vlies, idealdecor fototapete, idealdecor, fototapete, cm, bahn 183, bahn, architektur, 254 cm, 254, 2 bahn, 2, ...	Fototapeten Stadt	Fototapeten Comic, Fototapeten Meer
...	...	...	...

Tabelle 5.5: Textklassifikationsprobleme durch Dimensionsreduktion

- Fehlerhalte Daten: es existieren zwei oder mehrere Produkttypenbezeichnungen für einen Produkttyp oder die Produkttypen sind Synonyme. Aus diesem Grund sind die Ergebnisse von Produkttypenvorhersagen und wirklichen Produkttypen unterschiedlich (siehe die Tabelle 5.6).

  Lösungsvorschlag: Standardisierung von Produkttypen im Vorverarbeitung

Produktbeschreibung	Produkttyp	Vorhersage
name it Blumenprint Sweatkleid Mit langen Ärmeln ...	Sweatkleider	Langarm Kleider, Sweatkleider
Speck HardCase »PRESIDIO Grip iPhone (8/7/6S/6) Black/Black« Die Neuentwicklung der beliebten CandyShell-Hülle von Speck! ...	iPhone Cover	iPhone Case, iPhone Cover
3SIXT Schutzhülle »JellyCase für Apple iPhone 8 Plus« Schutzhülle JellyCase für Apple iPhone 8 Plus ...	iPhone Cover, iPhone Case	iPhone Case
Dreimaster Langmantel Wärmender Herren Parka Mit angesetzter Kapuze ...	Langmantel	Langmantel, Langmäntel
...	...	...

Tabelle 5.6: Textklassifikationsprobleme durch Fehlerhaltedaten

- Die Produkttexte sind zu einzigartig. Es gibt nicht ausreichend Trainingsbeispiele, um eine erforderliche Produkttypenklassifikationsleistung zu erbringen.

  Lösungsvorschlag: Anreicherung durch Zusatzinformationen oder mehr Trainingsbeispiele

Bildklassifikation

Durch Auswertung der Ergebnisse konnten vier Hauptprobleme bei der Bildklassifikation identifiziert werden, wie schon in 4.1.2 erläutertet wurde. Auf Grund dessen, dass bei der Bildklassifikation nur ein Produktbild pro Produkt verwendet wurde, zeigte die Produkttypenklassifizierung nur mit Bilderklassifizierung wie erwartet keine höhere Klassifikationsgenauigkeit.

Auch durch die Verbesserung durch den Imagegenerator konnte nur eine Klassifikationsleistung von 54.55% erreicht werden.

- Gleiche Bilder repräsentieren unterschiedliche Produkttypen. Abbildung 5.2 zeigt ein Bild, das bei vier Produkten vier unterschiedliche Produkttypen repräsentiert: "Bad-Hängeschränke", "Waschbeckenunterschränke", "Bad-Unterschränke", "Bad-Hochschränke".

  Lösungsvorschlag: durch Verbindung von Bild- und Textklassifikation konnten alle vier richtig klassifiziert werden. Ein weiterer Vorschlag wäre alle Bilder eines Produkts für Produkttypenklassifikation zu verwenden.

Abbildung 5.2: Bilderbeispiel für unterschiedliche Produkttypen

- Fehlerhafte Daten: es existieren zwei oder mehrere Produkttypenbezeichnungen für einen Produkttyp oder die Produkttypen sind Synonyme.

  Lösungsvorschlag: Siehe fehlerhafte Daten bei Textklassifikation 5.3.3.

- Die Produktbilder sind zu einzigartig. Zum Beispiel die Bilder einiger Bücher, die zu einem Produkttypen gehören, sind zu verschieden, deswegen ist es problematisch mit ausreichender Klassifikationsleistung vorherzusagen .

  Lösungsvorschlag: Verwendung von mehr Trainingsbeispielen oder zusätzlichen Informationen über Produkttypen erhöht die Produkttypenklassifikationsleistung, wie in der Tabelle 5.8 zu sehen ist.

- Auf einem Bild sind weitere Produkttypen abgebildet die nicht verkauft werden, wie Abbildung 4.4, Abbildung 5.2 und Abbildung 5.3 zeigen. Zum Beispiel Abbildung 5.3 repräsentiert »Bianco Patent Penny Halbschuhe« Produkttype: Loafer. Auf dem Produkt-

bild sind auch andere Kleidungstücke abgebildet. Anhand dieser Beispiel wird deutlich, dass die Bildklassifikation von dem Produkttype: Loafer problematisch ist.

Abbildung 5.3: Bianco Patent Penny Halbschuhe

Lösungsvorschlag: Die Verbindung von Bild- und Textklassifikation verbessert die Leistung des Klassifikationsmodells, wie die Tabelle 5.8 zeigt. Ein weitere Vorschlag wäre alle Bilder eines Produkts für Produkttopklassifikation zu verwenden.

Bild- und Textklassifikation

Eine genauere Untersuchung der Ergebnisse hat gezeigt, dass 18% der Produkttypen 98% oder mehr f1-measure und nur 2% einen f1-measure unter 50% haben. Die Beispiele von diesen 2% werden in der Tabelle 5.7 präsentiert. Da die Beschreibungstexte und Bilder von den Büchern meistens einzigartig sind, ist es mit wenig Produktbeispielen problematisch diese zu klassifizieren.

Produkttyp	precision	recall	f1-measure	support
Geschichtswissenschafts-Bücher	1.00	0.06	0.11	33.0
Schulbücher	0.71	0.17	0.27	30.0
Biologie Bücher	0.50	0.19	0.28	21.0
Humor Bücher	0.52	0.25	0.34	51.0
Psychologie Bücher	0.78	0.23	0.36	30.0
Bildbände	0.60	0.29	0.39	42.0
...	...	...	...	...

Tabelle 5.7: Beispiele von Produkttypen

Weitere Untersuchungen zeigten bei Produkttypen, die eine niedrigere Klassifikationsleistung bei Textklassifikation hatten, eine Leistungssteigerung durch Verbindung von Bild- und Textklassifikation erreichen, wie in der Tabelle 5.8 zu sehen ist.

Produkttyp	f1-measure Text	f1-measure Bild	f1-measure Bild+Text	support
Fototapeten Meer	0.16	0.70	0.86	23.0
Fototapeten Blumen	0.21	0.55	0.68	28.0
Fototapeten Wald	0.45	0.83	0.89	25.0
Loafer	0.46	0.60	0.83	13.0
Fototapeten Stadt	0.58	0.76	0.87	64.0
Blumentapeten	0.50	0.62	0.77	36.0
...	...	...	...	...

Tabelle 5.8: Beispiele von Produkttypen

5.4 Zusammenfassung

Es wurden die verwendete Programme und Hilfsmittel vorgestellt. Der gesamte Trainingsablauf wurde anhand eines Datenflussdiagramms gezeigt und es wurden Experimente beschrieben und die daraus resultierenden Ergebnisse ausgewertet. Im nächsten Kapitel wird das Ergebnis dieser Arbeit zusammengefasst und ein Ausblick gegeben.

6 Fazit und Ausblick

Im sechsten Kapitel wird das Ergebnis dieser Arbeit zusammengefasst präsentiert. Darüber hinaus wird ein Ausblick gegeben, wie das Projekt noch verbessert bzw. erweitert werden könnte.

6.1 Zusammenfassung

Die Zielstellung dieser Arbeit war ein lernfähiges System zu entwerfen und zu realisieren, das nach der Optimierung mit vorgeschlagenen und in dieser Arbeit untersuchten Optimierungsmethoden für die automatische Klassifikation von Produkttypen in einem Online-Shop eingesetzt werden kann. Dabei sollten möglichst viele Optimierungsmethoden untersucht werden, die die Klassifikationsleistung verbessern.

6.1.1 Vorgehensweise

Die sorgfältige Datenanalyse hat gezeigt, dass die vorliegenden Daten nicht nur Beschreibungstexte, sondern auch Bilder, die für die Produkttypenklassifizierung von den Produkten geeignet sind beinhalten, was auch durch Experimente belegt worden ist. Die Literaturrecherche deutete darauf hin, dass die einzelnen Klassifikationarten bei Multi-Class-Klassifikationsproblemen mit geringer Klassenanzahl gute Ergebnisse zeigen. Bei Multi-Label-Klassifikation mit über 1200 Klassen und einer geringen Menge der Trainingsdaten waren die Ergebnisse der einzelnen Klassifikationsarten nicht ausreichend genug. Als Ergebnis daraus wurde die Entscheidung getroffen, beide Klassifikationsarten zu verbinden um die Klassifikationsleistung zu steigern. Dabei wurde der Akzent auf Textklassifikation gesetzt, die Bildklassifikation spielte eine unterstützende Rolle um Stabilität und Leistung der Lehralgorithmen zu steigern. Es konnten durch Experimente nachgewiesen werden, dass der Verbund von zwei schwachen Modellen einen

erhebliche Zuwachs für Klassifikationsleistung zeigt. Es konnte sogar zwei unterschiedliche Kopplungsarten bei den vortrainierten Modellen realisiert und untersucht werden. Wobei sich die hierarchische Verbindung durch KNN als besser geeignete Architektur erwies, um zuverlässige Vorhersagen auf der Grundlage für die beiden vortrainierten Modellen zu ermöglichen. Das haben die durchgeführten Qualitätsmessungen gezeigt.

6.1.2 Erkenntnisse

Es ist anhand der vorliegenden Arbeit demonstriert worden, dass die Anwendung von maschinellen Lernmethoden für die Klassifikation von Produkttypen, auch mit nur 100 bis 500 Trainingsbeispielen und mehr als 1200 Multi-Label-Klassen, geeignet ist. Es sind beide präsentierten Ensemblemodellarchitekturen geeignet, um ähnlich charakterisierte Probleme zu lösen. Die produzierten Klassifikationergebnisse zeigen im Vergleich zu einzelnen Klassifikationsverfahren eine deutlich bessere Leistung. Es ist jedoch die in der Arbeit erwähnte Durchführung der Optimierung notwendig um die Klassifikationsmodelle in einer Produktion einzusetzen zu können, damit sich die Qualität von Produkttypenklassifikation noch weiter verbessern lässt.

Die iterative Vorgehensweise bei der Vorverarbeitung der bereitgestellten Daten hat gezeigt, dass es die Klassifikationsleistung maßgeblich beeinflusst. Die vorgeführten Experimente zeigen, dass die Verbesserung der Vorverarbeitungsprozesse einige Optimierungsmöglichkeiten anbieten. Die Schritte in der Vorverarbeitung sollen jedoch auf den jeweiligen Datenbestand, durch den im Kapitel 2 beschriebenen Ablauf beim überwachten Lernen mit Hilfe von Evaluation angepasst werden. Die durchgeführten Schritte führen speziell für die vorliegenden Trainingsbeispiele zu einer Verbesserung der Klassifikationsleistung. Es ist auch notwendig bei den einzelnen Klassifikationsmodellen die Optimierung durch die Anpassung der Hyperparameter durchzuführen.

Bücher und Multimedia-Artikel wie DVDs, CDs und Blu-rays müssen genauer untersucht werden, da die Beschreibungstexte und Bilder einzigartig sind, ist auch die Klassifikation erschwert. Es ist eine Informationsanreicherung notwendig um bessere Klassifikationsergebnisse zu erzeugen.

6.2 Ausblick

Das Klassifikationssystem hat ein großes Potenzial. Mit dem entworfenen System konnten nicht nur der Produkttyp von einem Produkt bestimmt werden, sondern jede beliebige Produkteigenschaft. Ein weiterer möglicher Forschungsschwerpunkt ist ein System zu realisieren, das automatisch alle relevanten Produkteingenschaften oder einen gesamten Navigationspfad bestimmt. Dafür müsste das System weiter angepasst werden. Durch den Einsatz von ML könnte die Vorverarbeitung von Produkttexten als solche automatisiert werden. Die Literaturrecherche zeigte, dass Spacy[1] viele Möglichkeiten anbietet, um dies zu realisieren. Bei der Bildklassifikation sollte es möglich sein mit allen Produktbildern zu trainieren. Die Literaturrecherche zeigt, dass es viele Arbeiten gibt, welche der Fusion von Farbbildern und Lasern oder Stereokameras untersuchen. Um Produktbilder zu fusionieren sollte die Arbeit von Ioana Gheta (2011) genauer analysiert werden. Der gewählte Ansatz könnte, dazu verwendet werden, um alle Bilder von einem Produkt für die Produkttypenklassifizierung zu verwenden. Anknüpfend an die vorliegende Arbeit ist ein weiterer interessanter Ansatz auch die Gruppierung der Produkte. Auf Grund der existierenden Navigationspfade ist es möglich Produkte zu gruppieren. Danach können die Produkte in ihren Gruppen zur Klassifikation verwendet, oder die Gruppen als ein kategoriales Merkmal verwendet werden.

[1] https://spacy.io

Literaturverzeichnis

[Asim u. a. 2017] Asim, Muhammad N. ; Rehman, Abdur ; Shoaib, Umar: Accuracy Based Feature Ranking Metric for Multi-Label Text Classification. In: *International Journal of Advanced Computer Science and Applications* 8 (2017), Nr. 10. – URL http://dx.doi.org/10.14569/IJACSA.2017.081048. – Letzter Zugriff: 09.01.2019

[Chollet. 2017] Chollet., François: Xception: Deep Learning with Depthwise Separable Convolutions. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. Honolulu, HI, USA : 2017 IEEE, 07 2017, S. 1800–1807. – URL https://doi.org/10.1109/CVPR.2017.195. – Letzter Zugriff: 09.01.2019. – ISBN 978-1-5386-0457-1

[Döbel u. a. 2018] Döbel, Inga ; Leis, Dr. M. ; Vogelsang, Manuel M. ; Neustroev, Dmitry ; Petzka, Dr. H. ; Rüping, Dr. S. ; Voss, Dr. A. ; Wegele, Martin ; Welz, Dr. J.: *Maschinelles Lernen - Kompetenzen, Anwendungen und Forschungsbedarf*. München : Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V. Elektronische Publikation, 2018. – 52 S. – URL https://www.bigdata.fraunhofer.de/content/dam/bigdata/de/documents/Publikationen/Fraunhofer_Studie_ML_201809.pdf. – Letzter Zugriff: 09.01.2019

[Feldman und Sanger 2006] Feldman, Ronen ; Sanger, James: *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press (11. Dezember 2006), 2006. – 424 S. – ISBN 978-0521836579

[Gheta 2011] Gheta, Loana: Fusion multivariater Bildserien am Beispiel eines Kamera-Arrays / von Ioana Gheţa. In: *Fusion Multivariater Bildserien am Beispiel Eines Kamera-Arrays (Schriftenreihe Automatische Sichtprüfung und Bildverarbeitung) Taschenbuch - 14. November 2011*. Karlsruhe : KIT Scientific Publishing (14. November 2011), 2011, S. 226. – URL http://d-nb.info/1014483115. – Letzter Zugriff: 09.01.2019. – ISBN 978-3-86644-684-7

[Hofacker und Schwandt 2018] Hofacker, Lars ; Schwandt, Friedrich D.: *E-Commerce-Markt Deutschland (Marktstudie der 1.000 umsatzstärksten B2C-Onlineshops für physische Güter).* Hamburg : EHI Retail Institute, Statista, 2018. – URL https://www.ehi-shop.de/de/studien/digitale-version/studie-e-commerce-markt-deutschland-2018. – Letzter Zugriff: 09.01.2019. – ISBN 978-3-87257-505-0

[Kannan u. a. 2011] Kannan, Anitha ; Talukdar, Partha P. ; Rasiwasia, Nikhil ; Ke, Qifa: Improving Product Classification Using Images. In: *2011 IEEE 11th International Conference on Data Mining.* Vancouver, Canada : IEEE, December 2011, S. 310–319. – URL https://www.microsoft.com/en-us/research/wp-content/uploads/2011/12/ImageText-ICDM2011.pdf. – Letzter Zugriff: 09.01.2019

[Kozareva 2015] Kozareva, Zornitsa: Everyone Likes Shopping! Multi-class Product Categorization for e-Commerce. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* Denver, CO, USA : The Association for Computational Linguistics, 2015, S. 1329–1333. – URL http://www.aclweb.org/anthology/N15-1147. – Letzter Zugriff: 09.01.2019

[Raschka 2015] Raschka, Sebastian: *Python Machine Learning, 1st Edition (English Edition).* Packt Publishing (23. September 2015), 2015. – 454 S. – ISBN 978-178355-513-0

[Rey und Rey 2017] Rey, Günter D. ; Rey, Günter D.: *Neuronale Netze Eine Einführung in die Grundlagen, Anwendungen und Datenauswertung.* Hogrefe, 2017. – URL http://neuronalesnetz.de/downloads/neuronalesnetz_de.pdf. – Letzter Zugriff: 09.01.2019. – ISBN 978-3-456-84513-5

[Richert 2013] Richert, Willi: *Building Machine Learning Systems with Python.* Packt Publishing (26. Juli 2013), 2013 (Community experience distilled). – 290 S. – ISBN 978-1-782-16141-7

[Russell und Norvig 2012] Russell, Stuart ; Norvig, Peter: *Künstliche Intelligenz: ein moderner Ansatz.* Pearson, Higher Education, 2012 (Always learning). – ISBN 978-3-868-94098-5

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 10. Januar 2019 Thomas Beznoskov