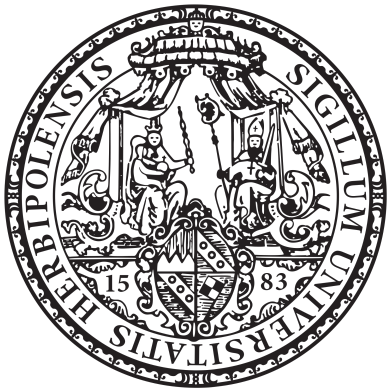
Spotify Mining

Fabian Kochsiek

2199486



Seminararbeit

Lehrstuhl für Wirtschaftsinformatik
und Systementwicklung
Universität Würzburg

Betreuer: Prof. Dr. Frédéric Thiesse

Würzburg, den 17.04.2021

Bearbeitungszeit: 22.02.2021 - 18.04.2021

Inhaltsverzeichnis

Zusammenfassung

In der vorliegenden Arbeit soll ein möglichst präzises empirisches Modell zur Vorhersage der Popularität eines Songs entwickelt werden. Auf Grundlage eines von Spotify bereitgestellten Datensatzes wird durch Vergleich verschiedener Verfahren des maschinellen Lernens der optimierte Decision Tree Algorithmus als zielführendste Methode ermittelt. Aus dem gewonnenen Erklärungsgehalt wird mithilfe des SHAP Frameworks das Modell interpretiert und die Wichtigkeit einzelner Features für die Popularität eines Tracks herausgearbeitet. Abschließend werden auch die erfolgversprechendsten Werte der jeweiligen Features zu bestimmen versucht.

Abstract

The aim of this paper is to develop the most accurate empirical model for predicting the popularity of a song. Based on a dataset provided by Spotify, an optimized decision tree algorithm is determined as the most promising method by comparing different machine learning methods. From the obtained explanatory content, the model is interpreted using the SHAP framework and the importance of individual features for the popularity of a track is highlighted. Finally, an attempt is made to determine the most promising values for the respective features.

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

CNN Convolutional Neural Networks

EDA Explorative Daten Analyse

HSS Hit Song Science

KNN K-Nearest Neighbors

MAE Mean Absolute Error

ML Machine Learning

MSE Mean Squared Error

NLP Neuro-Linguistisches Programmieren

RMSE Root Mean Squared Error

SHAP SHapley Additive exPlanations

1 Einleitung

Was macht einen Song zu einem Hit? Die Beantwortung dieser Frage wird im Spannungsfeld der Interessen von Musikindustrie, Künstlern und Publikum austariert: Die Musikindustrie strebt nach einer möglichst hohen Rendite, die Künstler möchten einen Namen und für sie akzeptable Einnahmen erzielen, und das Publikum möchte sich mit der Musik identifizieren und sich von ihr begeistern lassen. Eine von allen Seiten akzeptierte und eindeutige Definition von Popularität scheint nicht zu existieren. Der Wandel der Musikbranche hat durch Streaming-Dienste dazu geführt, dass jeder immer und überall seine Lieblingsmusik mit seinem Smartphone oder anderen Endgeräten hören kann.
Der Marktführer Spotify, den ich selbst nutzte, ist mit mehr als 345 Millionen aktiven Nutzern (Statista, 2020) eine Marktmacht die Einfluss auf den Publikumsgeschmack nehmen kann. Hier steht vor allem der sogenannte Spotify-Algorithmus in der Kritik, das Hörverhalten der User zu lenken. Durch den Zukauf der Firma Echo Nest können seit 2014 auch intelligente Playlisten anhand von individuellen Nutzervorlieben generiert werden, indem Musik durch neue Algorithmen noch besser klassifiziert werden kann. Doch welche Faktoren bestimmen die Popularität eines Tracks und welche sind dabei die wichtigsten? Lässt sich die Popularität eines Songs überhaupt vorhersagen? Und wenn dem so ist, welche Werkzeuge und Modelle sind dann für eine möglichst genaue Vorhersage am besten geeignet? Auch die Wissenschaft hat sich dieser Fragen angenommen, und diese Seminararbeit soll sich in die bereits bestehende Forschung einreihen.
Ziel dieser Arbeit ist, ein möglichst effektives empirisches Modell für die Vorhersage der Popularität eines Songs zu erstellen und dieses zu interpretieren. Zunächst wird eine Einführung in die Methodik des maschinellen Lernen gegeben und die verwendeten Algorithmen näher beleuchtet. Anschließend wird das für die Interpretation der Ergebnisse genutzte SHapley Additive exPlanations (SHAP) Framework vorgestellt. In einem weiteren Schritt wird der von Spotify bereitgestellte Datensatz präsentiert und mit Werkzeugen der explorativen Statistik untersucht und bereinigt. Um das Modell mit dem höchsten Erklärungsgehalt für die Popularität eines Songs ermitteln zu können, wird durch den Vergleich der Algorithmen und durch Zuhilfenahme von Performance-Kennzahlen das zielführendste Verfahren bestimmt. Die durch die praktische Anwendung herausgearbeiteten Faktoren (Features) werden bezüglich ihrer Wichtigkeit für die Popularität eines Tracks untersucht und unter Verwendung der SHAP Kennzahlen und Diagramme eingeordnet. Zugleich wird durch die Analyse der Frage nachgegangen, welche Ausprägung die Features idealerweise aufweisen sollten.
Abschließend werden die gewonnenen Erkenntnisse dargelegt, auf die bestehenden Limitierungen hingewiesen und weitere Forschungsansätze beschrieben.

2 Methodik

Bei dieser Arbeit handelt es sich um eine empirische Arbeit, bei der die nachfolgenden Forschungsfragen durch die Bildung eines empirischen Modells beantwortet werden sollen. Im Folgen wird die Vorgehensweise kurz vorgestellt.

2.1 Forschungsfragen

Um die bereits in der Einleitung aufgeworfenen Fragen näher eingrenzen und in den Fokus dieser Arbeit rücken zu können, werden zwei Forschungsfragen gestellt, die in dieser Seminararbeit beantwortet werden sollen und auch das Ziel des empirischen Modells definieren. Sie lauten:

1. Welcher Algorithmus des maschinellen Lernens kann am präzisesten die Popularität eines Songs vorhersagen?
2. Welche Features beeinflussen in welcher Ausprägung die erwartbare Popularität eines Songs?

2.2 Vorgehensweise

In der vorliegenden Arbeit werden zunächst mit dem Thema vergleichbare wissenschaftliche Studien vorgestellt, um die Forschung in den bestehenden Kontext einordnen zu können. Anschließend wird eine Einführung in die verwendeten Methoden des maschinellen Lernens und des erklärbaren maschinellen Lernens (engl. explainable AI/machine learning) gegeben.
Im Hauptteil der Seminararbeit wird der Streaming-Anbieter Spotify sowie der von ihm bereitgestellte Datensatz präsentiert. Die Vorgehensweise zur Erstellung des empirischen Modells, das die Popularität der Songs vorhersagen soll, folgt den Vorschlägen von Shmueli und Koppius (2011, S. 563) die in Abbildung 1 zusammengefasst sind. Die

Abbildung 1: Vorgehen zur Bildung des empirischen Modells

(Abbildung im Sinne von Shmueli und Koppius (2011, S. 563))

Daten werden mit Werkzeugen aus der explorativen Statistik untersucht. Basierend auf den so gewonnenen Erkenntnissen wird der Datensatz zur weiteren Verwendung bereinigt und die eingesetzte Performance-Messung der Machine Learning (ML) Verfahren erklärt. Damit die oben gestellten Forschungsfragen auch bestmöglich beantwortet werden können, wird im Anschluss die Untersuchung mit den beschriebenen Methoden durchgeführt und das Modell mit dem höchsten Erklärungsgehalt ermittelt. Dieses Modell wird sodann mit Hilfe der SHAP Kennzahlen und Diagramme genauer untersucht,

um den Einfluss und den Erklärungsfaktor der Features auf das Modell nachvollziehen zu können.
Abschließend werden die Ergebnisse dieser Arbeit zusammenfassend dargelegt und auf Limitierungen und weitere Forschungsmöglichkeiten eingegangen.

3 Grundlagen

In diesem Kapitel werden zunächst vergleichbare Arbeiten aus dem Bereich der Hit Song Science (HSS) vorgestellt. Anschließend wird eine Einführung in die Themen des maschinellen Lernens gegeben und die in der vorliegenden Arbeit verwendeten Algorithmen vorgestellt. Abschließend wir das SHAP Framework präsentiert, welches in dieser Arbeit verwendet wird, um die Ergebnisse der Algorithmen evaluieren und interpretieren zu können.

3.1 Vergleichbare Arbeiten

Um sich in dieser Arbeit dem Stand der aktuellen Forschung annähern zu können, soll im Folgenden untersucht werden, welche Features und Verfahren bereits verwendet wurden, um die Popularität von Songs vorherzusagen.
Dieses Thema wird in der Literatur unter dem Begriff der *Hit Song Science* verzeichnet (Karydis u. a., 2018, S. 5). Dabei ist besonders die Arbeit von Herremans, Martens und Sörensen (2014) zu hervorzuheben. Die Wissenschaftler konzentrierten sich in ihrer Untersuchung auf das Genre der Dance Hits und testeten mit den Features *Duration, Tempo, Time Signature, Mode, Key, Loudness, Dancability, Energy* mehrere Vorhersagemodelle wie "C4.5 Decision Tree, RIPPER, Native Bayes, Logistic regression, SVM Polynomial und RBF". Ihr Ziel war, eine Prognose abgeben zu können, ob ein Song in die Top Ten des Landes aufsteigen würde oder nur in die Top 30-40. Als Vorhersagemodell schnitt die logistische Regression in den Testdaten am besten ab. Die Autoren zeigten, dass es grundsätzlich möglich ist, die Popularität von Songs vorherzusagen, schränkten aber ein, dass ihre signifikanten Ergebnisse ggf. aus der Fokussierung auf das Genre der Dance Music resultieren könnten. Diese würde sich über die Zeit wenig verändern. Außerdem wurden im Test-Set nur aktuelle Songs verwendet.
Einen anderen Ansatz zur Bestimmung der Popularität von Tracks verfolgten Jakubowski u. a. (2017): Sie klassifizierten die Popularität als "Involuntary musical imagery"*(=INMI* oder umgangssprachlich auch *Ohrwurm)*. Aus einer Anzahl von 12 Features wurden mit Hilfe des Random Forest Verfahrens jene Features bestimmt, die den höchsten Erklärungsgehalt für die Popularität boten. Sie wurden angegeben mit *Stimmung, Tempo, Melodische Richtung*. In einem weiteren Schritt wurde ein logistisches Regressionsmodell mit diesen Variablen gefüttert. Auf diese Weise erzielten sie vor allem das Ergebnis, dass Songs aus dem Genre *Pop* eine hohe zu erwartende Popularität aufwiesen. Die Begründung sahen sie darin, dass diese Tracks häufig zu ihrer Zielsetzung passende Werte in allen drei Variablen hervorbrachten.

Eine ähnliche Vorgehensweise verfolgten auch Léveillé Gauvin (2018). In ihrer Studie wurde die Regression mit den Features *Number of words in title, Main tempo, Time before the voice enters, Time before the title is mentioned, Self-focused lyrical content* vorgenommen. Dabei waren *Number of Words in title, Main tempo, Self-focused lyrical content* die wichtigsten Charakteristika für die Erkennung von Hits, allerdings waren keine der drei Variablen signifikant in dem Modell. Für die Untersuchung wurde ebenfalls ein Datensatz von Spotify verwendet. Interessant ist vor allem das Ergebnis der Forscher, dass die Intro's (Zeitraum bis der Gesang im Stück einsetzt) über die Zeit immer kürzer wurden.
In ihrem Beitrag „Musical track popularity mining dataset: Extension & experimentation" zeigen die Autoren Karydis u. a. vor allem die Wichtigkeit einer standardisierten Datenbank und die Wichtigkeit von Feature-Definitionen auf. Diese Erkenntnisse sind für die weitere Forschung wichtig. Daneben werden auch mögliche Vorgehensweisen theoretisch beleuchtet, um die Popularität mit ML-Verfahren vorherzusagen, allerdings werden sie nicht praktisch eingesetzt.
Anders als die vorher genannten Forschungsarbeiten, in denen das Problem binär (Hit/ Kein Hit) beschrieben ist, wurde in der Studie von Martin-Gutiérrez u. a. (2020) ein Regressionsverfahren auf einen kombinierten Datensatz von Spotify und Genius angewendet. Dabei wurden zusätzlich noch weitere Features entwickelt, um mit den Werkzeugen aus dem Neuro-Linguistischen-Programmieren weitere Implikationen aus den Titeln und Songtexten zu gewinnen. Gerade durch diese Untersuchungen hat sich herausgestellt, dass durch die Verwendung eines Convolutional Neural Networks (CNN) für die Gewinnung weiterer Charakteristika eine Verbesserung der Vorhersagequalität der Modelle erreicht werden kann.
Yu u. a. (2019) und Yang u. a. (2017) kamen in ihren Studien ebenfalls zu dem Ergebnis, dass die Anwendung von Deep Learning Algorithmen bezüglich der Popularitätsvorhersage von Songs der Verwendung von klassischen Machine Learning Algorithmen überlegen ist.

3.2 Einführung in maschinelles Lernen

Maschinelles Lernen beschreibt die „künstliche" Generierung von Wissen aus Erfahrung . Um dieses "Wissen" (bzw. Modell) zu generieren, nutzen Computer Algorithmen zur Erkennung von Mustern in Datensätzen (Data Mining). Aus den so gewonnenen Mustern wird ein allgemeingültiges Modell aufgebaut, dass der Vorhersage einer oder mehrerer spezifischer Lösungsvariablen dienen kann (Carbonell, Michalski und Mitchell, 1983, S. 3). Um den Rahmen dieser Arbeit einhalten zu können, werden auf den folgenden Seiten nur die in dieser Arbeit verwendeten Algorithmen vorgestellt. Für einen umfangreicheren Überblick über das Thema, wird an dieser Stelle auf den englischsprachigen Wikipedia-Artikel Machine Learning hingewiesen.
Grundsätzlich wird bei ML der gesamte Datensatz in einen Trainingsdatensatz und einen Testdatensatz unterteilt - meistens im Verhältnis 80/20. Mit den Trainingsdaten kann der Algorithmus die Inputparameter mit den Outputparametern vergleichen und daraus ein Muster errechnen. Dieses Muster wird in ein mathematisches Modell über-

führt. Anschließend wird das Modell überprüft und kann durch Anpassung der Gewichte trainiert werden, sodass die gewünschte Vorhersage genauer auf die Outputparameter zutrifft (siehe dazu Abbildung 2).

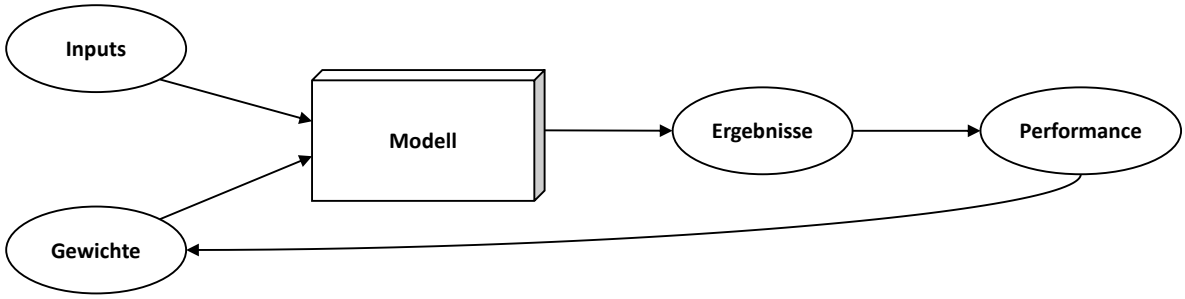


Abbildung 2: Maschinelles Lernen

(Abbildung im Sinne von Howard (2020))

Abschließend kann das Modell anhand der Testdaten auf seine Vorhersagegenauigkeit überprüft werden.
Ein solches Vorgehen wird unter dem Oberbegriff des *Supervised Learning* zusammengefasst, weil die Daten bereits vorher „gelabelt" sind, die Ergebnisse folglich vorher den Inputparametern zugeordnet sind. Allgemein ist wünschenswert, dass der vom Modell vorhergesagte Wert der Zielvariablen möglichst genau dem realen Wert entspricht. Dabei macht es aber je nach Anwendungsbereich einen Unterschied, ob eine möglichst geringe durchschnittliche Abweichung bevorzugt wird oder ob eine hohe Abweichung einzelner Werte möglichst vermieden werden soll (Reduktion der Ausreißer). Aus diesem Grund erscheint es sinnvoll, das Modell mit einer individuellen Verlustfunktion (engl. loss function) auszustatten (siehe Abbildung 3). Beispielsweise kann eine quadratische Formel zur Bestrafung von Ausreißern festgelegt werden. Mögliche Verlustfunktionen, die auch als Zielmetrik eingesetzt werden können, werden zu einem späteren Zeitpunkt in Unterabschnitt 3.2 beleuchtet. Eine weitere Möglichkeit

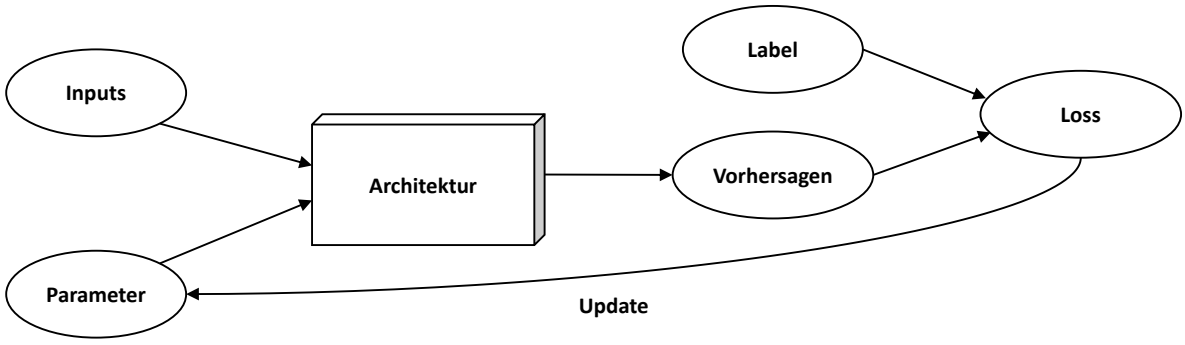


Abbildung 3: Maschinelles Lernen mit Loss Function

(Abbildung im Sinne von Howard (2020))

zur Verbesserung der Vorhersagemodelle ist der Einsatz von Hyperparameteroptimierung (engl. grid search). Dabei werden die Hyperparameter, z.B. die Tiefe des Entscheidungsbaums, die Anzahl an Blättern, die Sensitivität der Klassifizierungsregulation, etc. von einem Algorithmus so ausgewählt, dass die verwendeten ML-Algorithmen zu einem optimalen Ergebnisse kommen, da die Parameter "besser" gewählt worden sind (Bergstra, Yamins und Cox, 2013, S. 1). Um zu solchen zufriedenstellenden Ergebnissen zu kommen, gibt es verschiedene Methoden. In dieser Arbeit kommt die Implementierung von SciKit-Learn GridSearchCV [1] zur Anwendung. Als ML-Algorithmen werden die Linear Regression, K-Nearest Neighbors (KNN), Decision Trees, Random Forest und Gradient-Boosted-Tree-Algorithmus (in Form von XGBoost) genutzt und daher im nächsten Kapitel kurz vorgestellt.

3.3 Verwendete Machine Learning Algorithmen

Die lineare Regression ist eines der einfachsten Werkzeuge aus dem Bereich des Maschinellen Lernens. Wichtig ist die für die Zielsetzung jeweils richtige Zielmetrik, bzw. die loss function zu wählen, um das Modell bestmöglich anpassen zu können (Bishop, 2006, S. 158). Alternativ kann auch eine logistische Regression (Klassifizierung statt Gleitkommazahlen) verwendet werden, wenn eine kategorische Einordnung der Daten angestrebt wird (zu ersehen aus den Arbeiten in Unterabschnitt 3.1). Der KNN Algorithmus ordnet einen zu klassifizierenden Wert durch bereits klassifizierte Werte in der sogenannten "Nachbarschaft" ein. Dabei zeigt k die Grenze der "Nachbarn" an (vgl. Abbildung 4). Ein k, das zu gering gewählt wird, kann zu Overfitting führen, während ein zu groß gewähltes k eine ungenaue Kategorisierung ergeben kann.

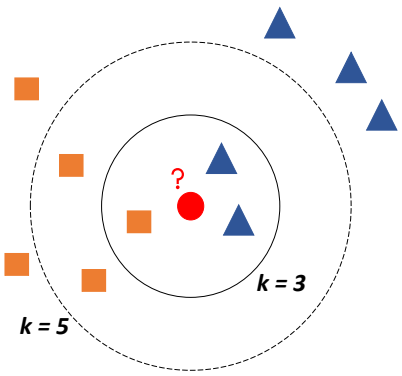


Abbildung 4: KNN
(Wikipedia KNN (2021))

Der Decision Tree Algorithmus folgt einer Baumstruktur, an dem Entscheidungsknoten die Daten in Unterkategorien aufteilen wie in Abbildung 5 zu sehen ist. Die Unterkategorien wird dann wiederum weiter unterteilt bis alle Daten final kategorisiert sind. Der Algorithmus kann auch zur Regression verwendet werden. Bei einem solchen Vorgehen ist das Ergebnis der Unterteilung allerdings keine Kategorie, sondern eine Zahl. Die Unterteilung wird dann meist nicht linear nach Gleitkommazahlen vorgenommen, sondern jeweils nach einer einzelnen Zahl, die größere Intervalle abdecken soll. Ähnlich der KNN Methode kann eine zu geringe Anzahl an Knoten zu Underfitting führen, eine zu hohe Anzahl aber das Modell overfitten. Ein großer Vorteil des Decision Tree Algorithmus liegt darin, dass der Algorithmus automatisch eine Features Selection vornimmt, indem er Features mit niedrigem Erklärungsgehalt schon frühzeitig einem Endknoten zuordnet und sich mit mehreren Knoten

[1] https://scikit-learn.org/stable/modules/grid_search.html

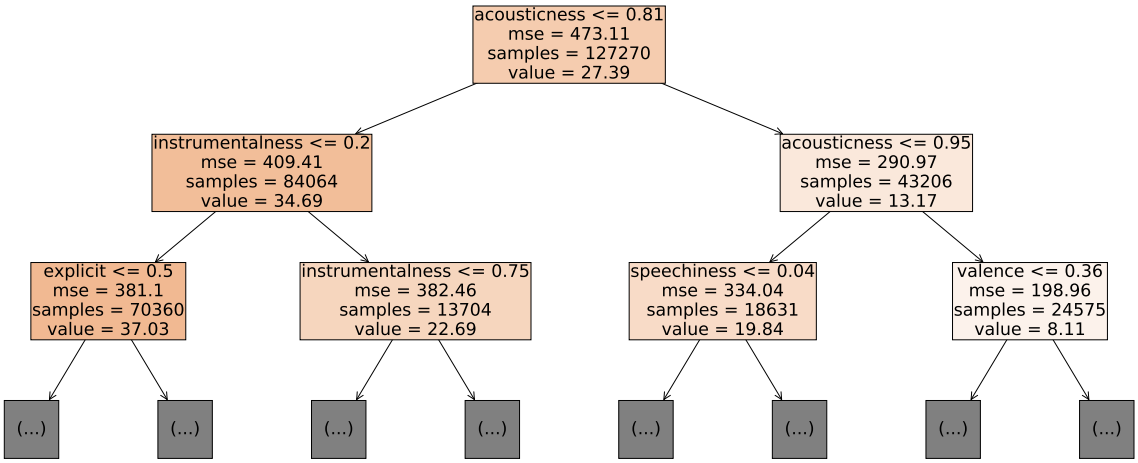


Abbildung 5: Ausschnitt Decision Tree

Zunehmende Farbe steht für zunehmende Wichtigkeit der Knoten
Datensatz ohne das Feature *Jahr*

auf die statistisch wichtigen Charakteristika konzentriert (Kwak und Choi, 2002, S. 1). Der Random Forest Algorithmus basiert auf mehreren zufälligen (unkorrelierten) Entscheidungsbäumen. Jeder Baum kann selbstständig und unabhängig von den anderen Bäumen seine eigene Struktur aufbauen. Am Ende werden die von den meisten Bäumen gewählten Entscheidungsstrukturen zusammengefasst und zu einem einzelnen Entscheidungsbaum verschmolzen.
Als weitere Möglichkeit wird der Gradient-Boosted-Tree-Algorithmus verwendet. In dieser Seminararbeit wird er in Form der XGBoost-Implementierung genutzt. Der Gradient-Boosted-Algorithmus identifiziert schwache Entscheidungsbäume und fügt diese zu einem starken Baum zusammen. Damit ist er laut Chen und Guestrin (2016, S. 786) in den meisten Fällen dem Random Forest Algorithmus überlegen.

3.4 Einführung in SHAP

Die Interpretation einer Vorhersage, die von einem ML-Algorithmus getroffen wurde, sowie eine genaue Evaluierung sowie Validierung auf welche Weise dieser zu seiner Lösung kommt, spielt eine sehr wichtige Rolle bei der Verwendung solcher Methoden (Lundberg und Lee, 2017, S. 1). Einerseits gilt es, die Vorgehensweise des Modells zu verifizieren und zu überprüfen, ob auch exakt das gewünschte Ziel vorhergesagt wird. Andererseits sollte auch das Verständnis vorhanden sein, welche Inputparameter in welchem Maße eine Rolle spielen. Das Modell könnte sonst ohne eine quantitative Erklärung lediglich eine Korrelation aufzeigen, die nur in bestehenden Daten vorliegt, aber nicht mehr signifikant in neuen Daten ist. Außerdem lässt sich ohne ein tieferes Verständnis des Modells dessen Robustheit gegenüber Änderungen in den Inputdaten nur schwer nachvollziehen.
Aus diesen Gründen entwickelten Lundberg und Lee einen allgemeinen Ansatz zur

Erklärung von maschinellem Lernen bzw. KI-Modellen, das sogenannte SHAP Framework. Dieses Umfasst neben den SHAP Values auch eine Sammlung von Diagrammen[2], die bei der Interpretation der Ergebnisse unterstützen sollen.
Die SHAP Values berechnen den Einfluss der Features auf das Modell und leiten sich von den Shapley Values ab, von Kennzahlen, die ursprünglich auf die Spieltheorie zurück gehen (Lundberg und Lee, 2017, S. 5). Diese lassen sich mithilfe folgender Formel errechnen, wobei x die Zielvariable ist und z die Features sind:

$$f(h_x(z')) = E[f(z)|z_S] \quad \text{SHAP Modell mit vereinfachtem Inputmapping} \quad (1a)$$
$$= E_{z_S|z_S}[f(z)] \quad \text{Erwartungswert über } z_{\bar{S}}|z_S \quad (1b)$$
$$\approx E_{z_S}[f(z)] \quad \text{Angenommene Unkorreliertheit der Features} \quad (1c)$$
$$\approx f([z_S, E[z_{\bar{S}}]]) \quad \text{Angenommene Modelllinearität} \quad (1d)$$

Eins der möglichen Diagramme (Abbildung 6) verdeutlicht somit den Einfluss der Features auf den Erwartungswert der Vorhersage ($E[f(z)|z_{1,2,3} = x_{1,2,3}$). Die Funktion $f(x)$ ist der Wert der Zielvariable, wenn die Features unbekannt sind.

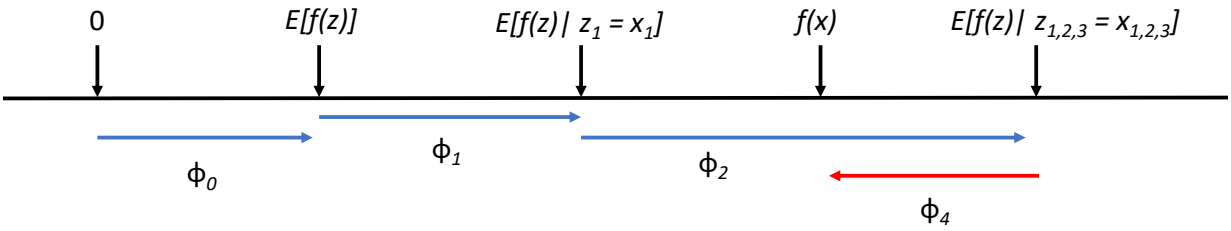


Abbildung 6: Diagramm aus dem SHAP Framework

(Abbildung im Sinne von Lundberg und Lee (2017, S. 5))

4 Hauptteil

Zunächst werden die in dieser Arbeit verwendeten Daten und Spotify als Quelle der Daten präsentiert. Im Anschluss wird der Datensatz explorativ untersucht und die Performance-Kennzahlen zum Training der ML-Algorithmen vorgestellt. Diese Algorithmen werden sodann mit den vorhandenen Daten und zunächst allen Features, später ohne das Erscheinungsjahr verarbeitet und der beste Algorithmus mit der Hyperparameteroptimierung verbessert. Die Ergebnisse werden abschließend mit dem SHAP Framework analysiert und interpretiert.

4.1 Daten

Die in dieser Arbeit verwendeten Daten wurden vom Streaming-Dienst Spotify bereitgestellt und sind auf der Website *Kaggel.com*[3] abgerufen worden. Gemessen an der An-

[2] https://github.com/slundberg/shap
[3] https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks

zahl der Nutzer ist Spotify mit über 320 Millionen Usern im ersten Quartal 2020 der größte Audio-Streaming-Anbieter der Welt (Statista, 2020). Sein Geschäftsmodell besteht darin, Nutzern zu ermöglichen durch Streaming immer und überall auf eine der größten Musiksammlungen weltweit zugreifen zu können. Durch den Kauf von Echo Nest 2014 hat Spotify einen Spezialisten in der Analyse von Musiktracks erworben, der durch verschiedene Verfahren u.a. des maschinellen Lernens und der künstlichen Intelligenz die Songs auf einer neuen Ebene interpretieren kann. Echo Nest ging aus dem MIT Media Lab hervor, sodass es nicht verwundert, dass einige Features aus dem Spotify-Datensatz auf den Analysen der Algorithmen aus dem MIT Labor basieren [4].
Der in dieser Arbeit verwendete Datensatz umfasst 174.389 Tracks von 1920-2021 mit 16 Features (inklusive Erscheinungsjahr). Es gibt keine leeren Datensätze, weshalb fehlende Daten vermutlich durch eine Null ersetzt wurden. Kritisch zu hinterfragen ist an dieser Stelle, dass nicht unterschieden werden kann zwischen Features mit einer tatsächlich vorhandenen 0 und einem fehlendem Wert, der mit 0 angegeben wird. Dieses Vorgehen wird in Unterabschnitt 4.2 in der Explorative Daten Analyse (EDA) weiter untersucht, zunächst wird aber auf jeglichen Einsatz von Interpolation verzichtet.
Die Features im Datensatz sind in Tabelle 1 zusammengefasst, die dazugehörigen Erklärungen sind der API Dokumentation von Spotify[5] entnommen. Zuallererst wird der Datensatz untersucht, um auffällige Daten im Datensatz zu identifizieren. In einem ersten Schritt werden dabei Duplikate entfernt. Auffallend ist, dass in den Daten mehrere Tracks fälschlicherweise als Playlisten (Mixes) gelabelt sind. Diese Mixes werden in einem zweiten Schritt durch das Kennwort "Mix" UND eine Längenbeschränkung von 20 min. herausgefiltert. Außerdem werden sehr kurze, stumme Tracks in einem Album ausfindig gemacht, die als *Pausen Track* definiert sind. Sie werden ebenfalls aus den Daten entfernt.
Da Spotify neben Musiktracks auch Hörbücher und -spiele anbietet, sind diese ebenfalls im Datensatz enthalten. Allerdings wird, wie aus Tabelle 1 zu ersehen ist, keine Kategorisierung in Musik oder Hörbuch vorgenommen. Stattdessen wird auf das Features *speechiness* verwiesen. Bei einem Wert über 0,66 handelt es sich mit großer Wahrscheinlichkeit um ein Hörbuch. Die Unterteilung ist nicht ganz eindeutig, da Genres wie Rap mit einem hohen Sprachanteil aufwarten können, und Hörspiele neben dem gelesenen Text auch Musikabschnitte enthalten können. Da Lesungen nicht Bestandteil der Untersuchung sind, werden diese ebenfalls aus dem Datensatz ausgeschlossen.
Wie bereits zu Beginn des Kapitels erwähnt, wurden fehlende Daten durch eine 0 ersetzt. Allerdings lässt sich nur im Feature *Tempo* eine 0 als fehlender Wert identifizieren, da ein BPM (Beats per Minute) von Null in einem Musikstück nicht plausibel ist: Der auf Tempo Null nächstfolgende niedrigste Wert ist erst oberhalb von 30 angesiedelt. Daher erscheint das Herausfiltern aller Daten bezüglich des Tempos, die einen Wert von 0 besitzen, eine angemessene Vorgehensweise zu sein. Nach vorgenommener Redukti-

[4] https://www.br.de/puls/musik/aktuell/spotify-the-echo-nest-discover-Dweekly-100.html

[5] https://developer.spotify.com/documentation/web-api/reference/#category-tracks

Features	Range	Erklärung
acousticness	0-1 (Float)	Dieser Wert ist ein Schätzwert, der angibt, ob ein Track akustisch ist. 1 steht für eine hohe Wahrscheinlichkeit, dass der Track akustisch (nicht elektronisch) ist.
danceability	0-1 (Float)	Dieser Wert beschreibt, wie gut sich ein Track zum Tanzen eignet. Er basiert auf den Kriterien Tempo, Rhythmusstabilität: Beatstärke und allgemeine Regelmäßigkeit. Ein Wert von 1 ist am tanzbarsten.
duration_ms	5 sec bis 1,48 h (Integer)	Dieser Wert gibt die Dauer eines Tracks in Millisekunden an.
energy	0-1 (Float)	Dieser Wert stellt ein Wahrnehmungsmaß für Intensität und Aktivität eines Tracks dar. Er basiert auf den Kriterien Dynamik, wahrgenommene Lautheit, Klangfarbe, Einsetzgeschwindigkeit und allgemeine Entropie.
explicit	0, 1 (Interger)	Dieser Wert gibt an, ob ein Track explizite (nicht jugendfreie) Inhalte enthält, 1 = explizit.
instrumental-ness	0-1 (Float)	Dieser Wert sagt aus, ob ein Track Stimmen oder Gesang enthält. Rap- oder Spoken-Word-Tracks sind eindeutig "vokal 1 bedeutet, dass ein Track ausschließlich mit Musikinstrumenten produziert wurde.
key	0-11 (Integer)	Dieser Wert gibt die Tonart, in welcher der Track komponiert ist, an. Zugrunde liegendes Schema: 0 = C, 1 = C#, 2 = D, usw.
liveness	0-1 (Float)	Dieser Wert beurteilt die Anwesenheit eines Publikums bei der Aufnahme eines Tracks. Ein Wert über 0,8 stellt eine hohe Wahrscheinlichkeit dar, dass der Track live ist.
loudness	-60 bis 3,85 (Float)	Dieser Wert gibt die Gesamtlautstärke eines Tracks in Dezibel (dB) an. Die Lautheitswerte werden über den gesamten Track gemittelt.
mode	0, 1 (Integer)	Dieser Wert gibt die Modalität (Dur oder Moll) eines Tracks an. Dur = 1 und Moll = O.
speechiness	0-1 (Float)	Dieser Wert bewertet inwieweit gesprochene Wörter in einem Track vorhanden sind. Werte über 0,66 beschreiben Tracks, die wahrscheinlich vollständig aus gesprochenen Wörtern bestehen.
tempo	0 bis 243,51 (Float)	Dieser Wert gibt das geschätzte Gesamttempo eines Tracks in Beats pro Minute (BPM) an.
valence	0-1 (Float)	Dieser Wert beschreibt die musikalische Positivität. Tracks mit hoher Valenz klingen positiver (z. B. glücklich: fröhlich, euphorisch), während Tracks mit niedriger Valenz eher negativ klingen (z. B. traurig, deprimiert; frustriert).
year	1920 - 2021 (Integer)	Dieser Wert gibt das Ersterscheinungsjahr eines Tracks an.
popularity	0-100 (Integer)	Dieser Popularitätswert wird durch einen Algorithmus berechnet und basiert zum größten Teil auf der Gesamtzahl der Wiedergaben des Titels und wie aktuell diese Wiedergaben sind. Im Allgemeinen besitzen Tracks, die aktuell viel gespielt werden, eine höhere Popularität als Songs, die in der Vergangenheit viel gespielt wurden. Doppelte Titel (z. B. derselbe Titel von einer Single und einem Album) werden unabhängig voneinander bewertet.

Tabelle 1: Übersicht und Erklärung über die Features

on der Datenmenge enthält der Datensatz im Ergebnis noch 159.088 einzelne Songs, es wurden also 15.301 entfernt.
Im folgenden Schritt soll nun der Einfluss der einzelnen Features auf die Popularität näher beleuchtet werden.

4.2 Explorative Datenanalyse

Um den Einfluss der Features auf die Popularität zu untersuchen, werden die aufbereiteten Daten auf ihre Korrelation hin untersucht. Dabei zeigt sich, dass vor allem das Erscheinungsjahr eines Songs einen großen Einfluss auf dessen Popularität besitzt (vgl. Abbildung 7). Dieser Zusammenhang ist zunächst mit der Definition von Popularität zu erklären, die Spotify allgemein bei Tracks zugrunde legt: Je aktueller ein abgerufener Song ist, desto größer ist seine Popularität (siehe Tabelle 1). Um dieses Phänomen näher

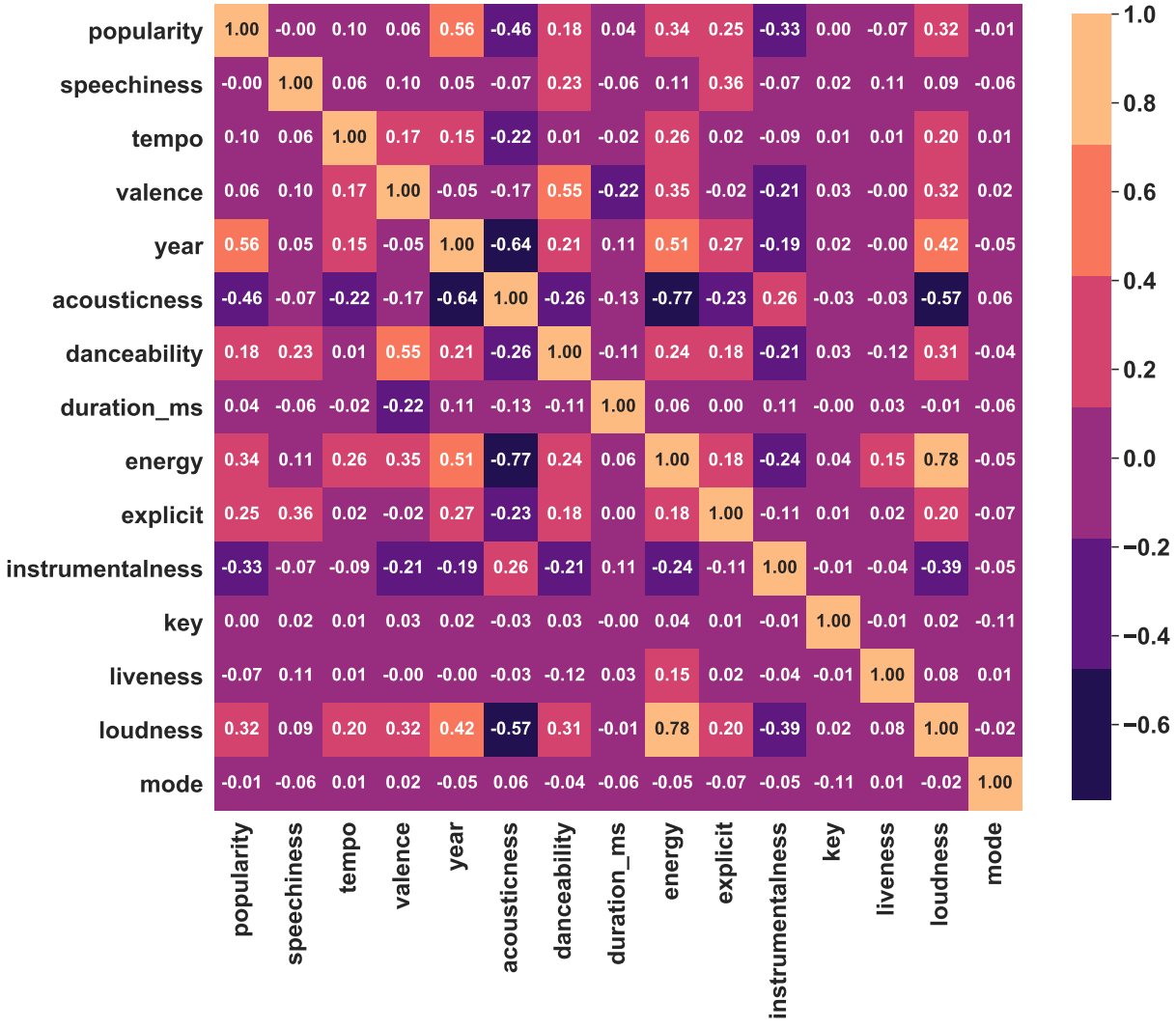


Abbildung 7: Korrelationsmatrix der Features

zu analysieren, wird in Abbildung 8 die Veränderung der Features auf einer Zeitachse gezeigt. Hier ist in erster Linie zu beobachten, dass Musikstücke zwischen 1920-1955 allgemein eine sehr geringe Popularität genießen im Vergleich zu später erschienenen. Als Grund kann hier nicht nur die Definition von Spotify angeführt werden, sondern auch das Aufkommen des Rock 'n' Roll. Diese damals neue US-amerikanische Musikrichtung hat ihren Ursprung in den 50er Jahren, einer Zeit der Protestkultur, die die Jugend antrieb. Prägend für das damalige Lebensgefühl waren noch heute bekannte Künstler wie Elvis Presley, James Dean und Jonny Cash (Rock'n'Roll, 2021).
Dass das Interesse an der Musik aus den 70er, 80er, 90er und 2000er Jahren anhält, liegt auch an dem Einfluss von zeitlosen Musiklegenden wie den Beach Boys, die Beatles, Tina Turner und vielen weiteren Künstlern. Dieser Trend hält bis in die frühen 2000er Jahre an, die durchschnittliche Popularität eines Songs sinkt erst ab 2010 wieder ab. Darüber hinaus ist in dieser Aufstellung auch zu sehen, dass vor Beginn der 50er Jahre

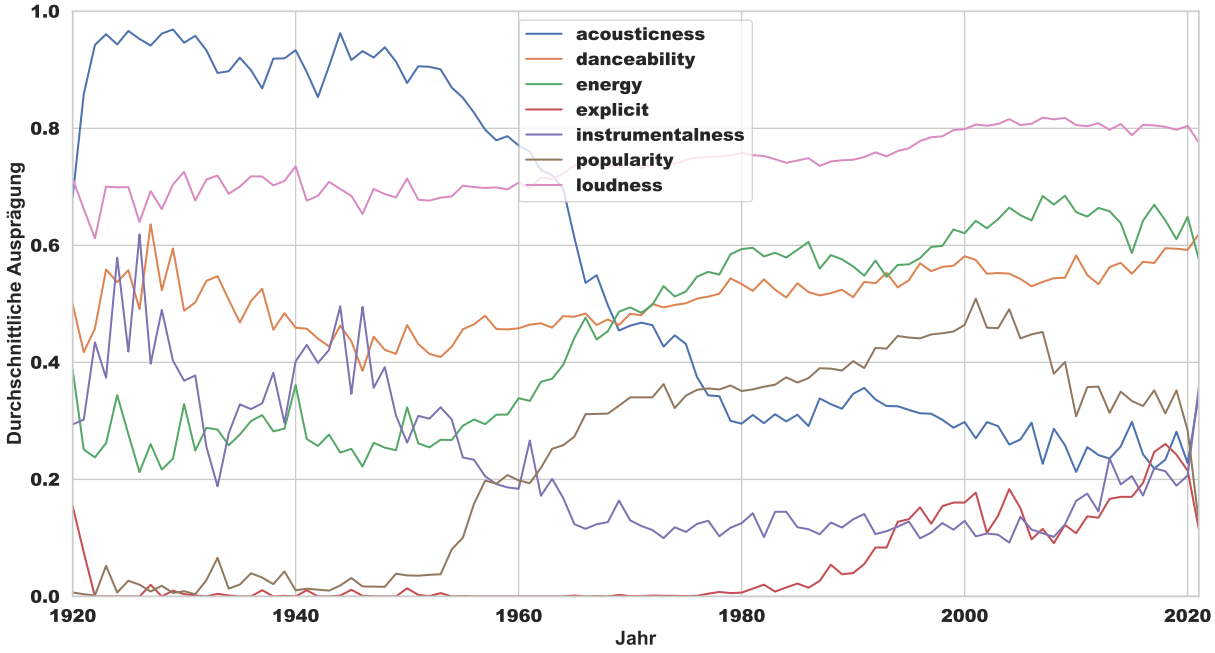


Abbildung 8: Ausgewählte Features und Popularität im Verlauf der Jahre

mehr als 90% aller Titel als rein akustische Musik aufgeführt werden. Fast ausschließlich unverstärkte Instrumente werden gespielt. Nach dieser Zeit fällt der Anteil der akustischen Musik stark ab und wird durch einen Trend hin zu immer mehr elektronisch aufbereiteter Musik ersetzt. Des Weiteren kann beobachtet werden, dass die Anzahl von Instrumentalstücken ohne Gesang abnimmt.
Zusätzlich erhöht sich der durchschnittliche *Energy*-Wert der Musik erheblich, die Ausprägung der *Loudness* steigt im gleichen Betrachtungszeitraum ebenfalls an, allerdings nur moderat. Auffällig ist vor allem die Zunahme von expliziten Inhalten bei den Songtexten. Diese Beobachtung lässt sich durch das Aufkommen von Hip-Hop - insbesondere der Rap Musik - erklären, die von den afroamerikanischen Ghettos der USA aus-

gehend in den letzten Jahrzehnten deutlich an Popularität auch in Europa gewonnen hat (Rap, 2021). Hier sind vor allem nicht jugendfreie Inhalte vorherrschend.
Das Feature *Danceability* der Tracks ist in dem beobachteten Zeitraum relativ konstant, mit einem kleinen Knick in den 1940er - 1960er Jahren. Auftakt für den Einbruch könnte der Ausbruch des Krieges sein. Während und nach dem zweiten Weltkrieg erholte sich weltweit die Tanzkultur verständlicherweise nur sehr langsam, bis in den 60er Jahren Musik mit afroamerikanischen Wurzeln auf dem Vormarsch war, die zum Tanzen einlud.
Zum besseren Verständnis des Zusammenhangs zwischen der Ausprägung der Features und der Popularität über die Korrelation hinaus, werden im nächsten Schritt ausgewählte Charakteristika und ihre durchschnittliche Popularität in einem weiteren Diagramm in Abbildung 9 dargestellt. Die Werte sind normiert, um eine einfache Vergleichbarkeit in einem Schaubild zu ermöglichen. Das Erscheinungsjahr des Songs wird auf der Y-Achse so standardisiert, dass 0 als Ausgangspunkt von 1920 zu interpretieren ist, und 1 das Jahr 2021 angibt. Die X-Achse reicht nur bis zu einem Popularitätswert von 90, da nur 31 Songs mit einer Popularität von über 90 im Datensatz vorhanden sind. Um zu keinen verfälschten Ergebnissen durch diese unterrepräsentierten Songs zu kommen, wurden diese aus der Abbildung entfernt.
Zunächst zeigt sich erneut, dass modernere Stücke eine höhere Beliebtheit aufweisen.

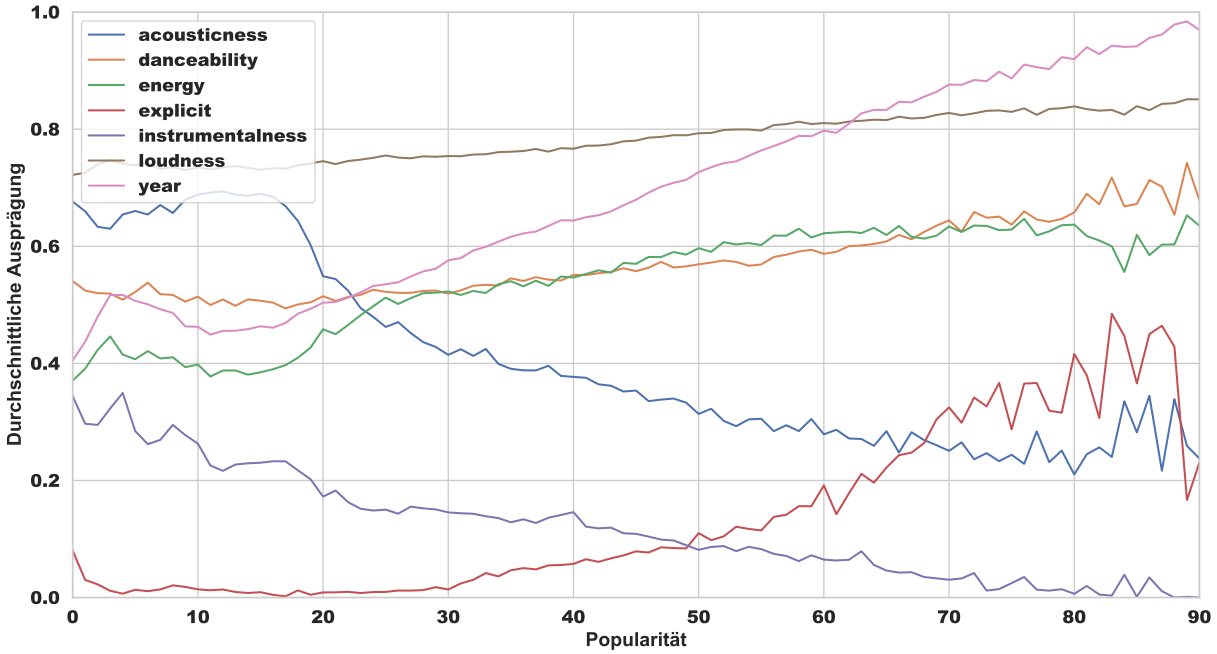


Abbildung 9: Ausgewählte Features und Zusammenhang zur Popularität

Das Feature *Acousticness* fällt fast linear ab, scheint aber eine Schwelle von 0,2 nicht zu unterschreiten. Um einen möglichst populären Song zu kreieren, sollte daher nicht rein elektronisch erzeugte Musik verwendet werden, sondern eher eine Mischung mit Instrumentalmusik, die elektronisch bearbeitet wurde.

Zusätzlich zeigt sich, dass Tracks Gesang beinhalten sollten, da ansonsten die zu erwartende Popularität sehr gering ausfällt. Der *Energy*-Wert eines Stückes steigt mit zunehmender Popularität ebenfalls an, pendelt sich aber auf einem Niveau um die 60% ein. Das Charakteristikum *Loudness* steigt ebenfalls linear zur Popularität auf über 80% an. Ein ähnliches Bild beim Feature *Explicit*: Ein Anstieg, wobei fast 50% der Stücke mit einem Popularitätswert von über 85 explizite Inhalte verwenden. Eine nicht jugendfreie Ausdrucksweise scheint somit kein Negativkriterium für eine hohe Popularität zu sein, sondern vielmehr nötig zu sein, um eine hohe durchschnittliche Beliebtheit zu erzielen. Als Beleg dienen Songs mit einem Popularitätswert von 30 und weniger, die keinerlei explizite Inhalte im Text enthalten. Das Feature *Danceability* steigt ebenfalls, fast parallel mit dem Feature *Loudness*, linear an, überschreitet aber den Wert von 0,7 nicht.
Interessant ist hier beim Vergleich von Abbildung 8 und 9, dass sich die Feature-Werte von Songs mit einer höheren Popularität im Lauf der Zeit auf einem bestimmten Niveau durchsetzten (siehe dazu jeweils die Graphen von z.B. *Acousticness* und *Loudness*). Der digitale Musikmarkt, beziehungsweise die Künstler, scheinen dem Geschmack des Publikums zu folgen.
Aus diesen Erkenntnissen sollen nun Modelle erstellt werden, die den genaueren Zusammenhang der untersuchten Features zielführend abbilden.

4.3 Performancemessung des maschinellen Lernens

Zur Bestimmung, wie gut die Vorhersage der Machine Learning Algorithms bei der gestellten Aufgabe ist, werden einige klassische statistische Kennzahlen verwendet. Diese werden im Folgenden kurz vorgestellt.
Mittlerer absoluter Fehler *(engl. Mean Absolute Error (MAE))* zeigt die durchschnittliche absolute Abweichung. Es wird nicht zwischen positiver oder negativer Abweichung unterschieden. Zur Interpretation müssen die anderen Modelle daneben betrachtet werden, da die Kennzahl alleine nicht aussagekräftig ist. Ob ein Wert gut oder schlecht bezüglich der Popularität eines Songs ist, kann nur mittels der Abgleichung von Modellen und Kennzahlen beurteilt werden.
Mittlere quadratische Abweichung *(engl. Mean Squared Error (MSE)* zeigt die durchschnittliche quadratische Abweichung. Diese Kennzahl zeigt im Vergleich zum MAE vor allem an, wie hoch die Varianz des Schätzverfahrens ist. Dabei wird eine hohe Abweichung besonders stark "bestraft" (Loss Function). Eine niedrige Varianz wird besonders präferiert, da hier die geschätzten Werte weniger extrem von den realen Werten abweichen (Ausreißer).
Die Wurzel der mittleren quadratischen Abweichung *(engl. Root Mean Squared Error (RMSE))* stellt die Mischung der beiden vorher genannten Kennzahlen dar. Mittels dieser Kennzahl wird die absolute Abweichung dargestellt mit einer besonderen Gewichtung auf eine hohe Abweichung. Das RMSE ist die am weitesten verbreitete Kennzahl für die Performance-Messung eines ML-Algorithmus und wird häufig auch als Zielmetrik verwendet, die vom Algorithmus minimiert wird. In der folgenden Arbeit werden alle drei Kennzahlen verwendet. Der beste ML-Algorithmus wird anhand des RMSE

bestimmt.

4.4 Praktische Anwendung ausgewählter Algorithmen

Die Ergebnisse der verschiedenen Algorithmen sind in Tabelle 2 aufgeführt. Auffällig

	Lineare Regression		**KNN**	**Decision Tree**			**Random Forest**	**XGBoost**
MAE	*13,40*	12.72	16.66	*9,55*	9.68	9.16*	10.44	12.75
MSE	*305,14*	289.07	400.51	*198,89*	199.30	187.23*	336.41	314.81
RMSE	*17,47*	17.00	20.01	*14,10*	14.12	13.68*	18.34	17.74

Tabelle 2: Ergebnisse der Algorithmen

Kursive Werte sind Ergebnisse aus dem rohen Datensatz
* sind Ergebnisse mit Hyperparameteroptimierung

ist, dass die lineare Regression ebenso wie KNN von dem bereinigten Datensatz profitieren, die anderen Algorithmen aber eher schlechter abschneiden. Aus Übersichtsgründen sind in der Tabelle nur lineare Regression und Decision Tree stellvertretend mit rohen und bereinigten Daten aufgeführt.
Am besten performt der Decision Tree Algorithmus, wohingegen der KNN sich als das schlechteste Werkzeug herausstellt. Interessant ist, dass der Random Forest Algorithmus sowie der XGBoost Algorithmus sehr viel negativer als der Decision Tree abschneiden, obwohl beide Algorithmen aufeinander aufbauen. Dieses Verhalten wird in Abbildung 10 näher untersucht. In dieser Abbildung ist die Wichtigkeit der einzelnen Features (engl. feature importance) basierend auf den Ergebnissen der jeweiligen Algorithmen dargestellt.
Der Random Forest gewichtet die einzelnen Features fast gleich, während XGBoost die Wichtigkeit des *Explicit* Features extrem stark gewichtet. Nur der Decision Tree Algorithmus scheint eine ausgewogene Gewichtung der Charakteristika mit Fokussierung auf die auch intuitiv wichtigen Features *Speechiness, Intrumentalness, Acousticness* vorzunehmen. Die Feature Importance wird im nächsten Kapitel mithilfe der SHAP Analyse weiter vertieft.
Um das Ergebnis des Decision Tree Algorithmus noch weiter zu verbessern wird die in Unterabschnitt 3.2 angesprochene Hyperparameteroptimierung angewendet. Dabei werden zunächst die Parameter des ursprünglichen Decision Trees analysiert und dann Bereiche für die Hyperparameter gewählt, die um die ursprüngliche Lösung herum liegen. Alternativ kann auch eine zufallsbasierte Grid Search verwendet werden. Da die Optimierung allerdings recht rechenintensiv ist, wird an dieser Stelle darauf verzichtetet und nur die bereits besseren Ergebnisse präsentiert um die Möglichkeiten der Hyperparameteroptimierung aufzuzeigen. So wurde von der ursprünglichen Tiefe des Baumes mit 13 Knoten ein optimale Tiefe von 12 Knoten bestimmt. Die minimalen Blätter von 31 wurden durch die Grid Search auf 46 erhöht. Das Ergebnis des optimierten Decision Tree ist in Tabelle 2 mit einem * markiert.

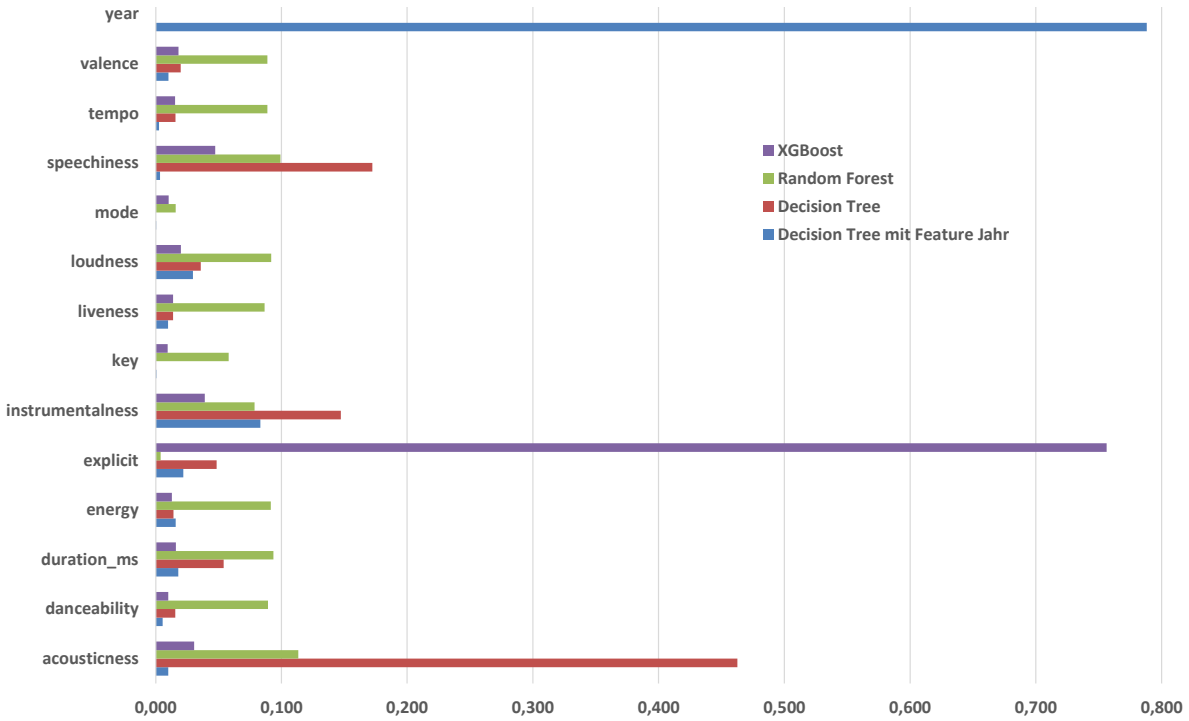


Abbildung 10: Feature Bedeutung

Wie der Korrelationsmatrix in Abbildung 7 zu entnehmen ist, ist die Popularität zu einem großen Teil mit dem Erscheinungsjahr positiv korreliert. Dies unterstreicht die Wichtigkeit dieses Features für die lineare Regression. In Abbildung 10 wird die Wichtigkeit der Features mit und ohne Erscheinungsjahr basierend auf dem Decision Tree Algorithmus aufgezeigt. Festzustellen ist, dass sich diese Methode noch mehr auf das Feature *Year* zur Vorhersage der Popularität verlässt als das Korrelationsmodell der lineare Regression. Da das Ziel dieser Arbeit jedoch ist, dem Künstler eine wissenschaftlich fundierte Vorschlag zu präsentieren, den er zur Popularitätssteigerung seiner Musik nutzen kann, ist aus dieser Analyse heraus nur zu raten, sich an der Musik der 70er, 80er und 90er zu orientieren. Allerdings ist diese Empfehlung nicht an den aktuellen Musikgeschmack angepasst.
Im nächsten Schritt wird das Feature *Year* ausgeschlossen, da es der Künstler nicht verändern kann. Dadurch sinkt zwar die Vorhersagegenauigkeit der Modelle wie in Tabelle 3 zu erkennen ist, jedoch ist klar festzustellen, dass der Decision Tree Algorithmus noch immer der zielführendste Algorithmus ist. Daher erscheint es plausibel zu sein, die Ergebnisse dieses vielversprechenden Algorithmus im nächsten Kapitel genauer zu untersuchen.

4.5 Erklärung mit SHAP

Die Berechnung der SHAP Values zeigt noch einmal die Wichtigkeit der Features in Abbildung 11, sortiert sie allerdings nach absteigender Relevanz. Außerdem wird deut-

	Lineare Regression	KNN	Decision Tree		Random Forest	XGBoost
MAE	14.61	17.36	13.44	12.74*	17.72	15.87
MSE	331.70	433.08	298.44	281.41*	649.55	422.88
RMSE	18.21	20.81	17.28	16.77*	25.49	20.56

Tabelle 3: Ergebnisse der Algorithmen ohne Feature *Year*

* sind für Ergebnisse mit Hyperparameteroptimierung

lich, welche Werte die Features schätzungsweise annehmen sollten, um einen positiven Einfluss auf die Popularität zu haben. Zugleich ist aber auch erkennbar, welche Feature-Ausprägungen vermieden werden sollten, um keinen negativen Einfluss auf die Popularität zu nehmen. Dabei repräsentieren rote Werte einen hohen Wert des Features und blaue einen niedrigen. Die Punkte rechts der Y-Achse haben einen positiven Einfluss auf die Popularität. Je weiter links sie von dieser Achse entfernt sind, desto schlechter ist dieser Wert für die zu erwartende Popularität. In dieser Abbildung ist

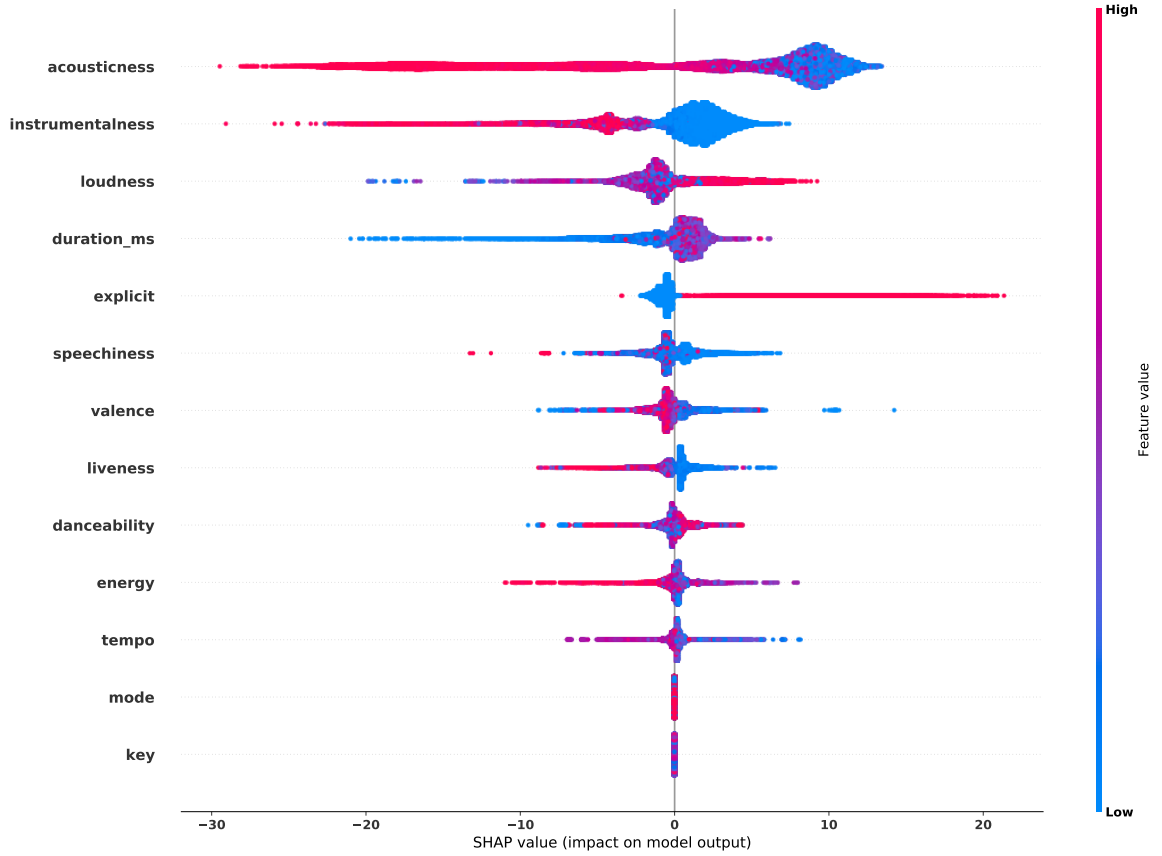


Abbildung 11: Feature Einfluss im SHAP Framework

deutlich zu erkennen, dass eine niedrigere *Accousticness* zu einer höheren Popularität

führt, während viele Songs mit einer hohen *Accousticness* eine niedrige Popularität besitzen. Dem selben Prinzip folgt auch das Feature *Instrumentalness*. Hier sind die Daten jedoch schon näher um die Y-Achse verteilt und es bildet sich eine kleine Datensammlung links der Y-Achse. Das Charakteristikum *Loudness* zeigt, dass höhere Werte zu einer hohen erwarteten Popularität führen, wobei der Einfluss einer mittleren *Loudness* sich nur leicht negativ auswirkt. Bezüglich der Länge eines Tracks ist festzustellen, dass überlange Tracks ein strikt negatives Kriterium darstellen. Des Weiteren sticht deutlich hervor, dass ein gehäuftes Vorkommen von expliziten Inhalten in einem Track einen strikt positiven Einfluss auf die erwartbare Popularität aufweist. Das Feature *Speechiness* hat weniger Bedeutung und sollte eher niedrige Werte aufweisen, da hohe Werte einen negativen Einfluss auf die Popularität besitzen. Ähnliches ist beim *Valence* Feature zu beobachten: Niedrige Werte führen entweder zu einer bipolaren Entscheidung gut oder schlecht für die Popularität, hohe Werte sorgen dagegen für einen negativen Einfluss. Das Charakteristikum *Liveness* zeigt ein trennschärferes Bild, bei dem ein niedriger Wert die Popularität stützt, und hohe Werte negativ zu beurteilen sind. Hinsichtlich des Features *Danceability* lässt sich kein klares Muster erkennen, mit Ausnahme eines negativen Einflusses von sehr niedrigen Werten. Das Feature *Energy* lässt erkennen, dass moderate Werte sich positiv auswirken. Außerdem kann beim *Tempo* erkannt werden, dass niedrige Werte zu einer hohen Popularität führen. Die Features *Mode* und *Key* haben keine Auswirkung auf die Popularität und können daher vernachlässigt werden.
Das SHAP Modell ermöglicht ebenfalls, einen mit dem ML-Algorithmus analysierten Track dahingehend zu untersuchen, welche Features einen positiven und welche einen negativen Einfluss auf die Popularität haben. Daher sind in Tabelle 4 einige Ergebnisse zur Vorhersage der Popularität eines Songs mit den dazugehörigen Features dargestellt. Der erste Track, *Even Now* von Barry Manilow, wurde nun in Abbildung 11 mit

artists	name	popularity	*predicted*	acoustic.	dur_min	explicit	instru.	loudness	speech.	tempo
Barry Manilow	Even Now	29	*36,15*	0.48	3.49	0	0.00	−11.64	0.03	133.93
Ray Anthony	Out Of Nowhere	13	*15,82*	0.96	2.73	0	0.92	−14.78	0.04	75.99
New Kids On The Block	Popsicle	28	*38,14*	0.12	4.79	0	0.00	−12.00	0.06	103.63
Tommy Olivencia	Periquito Pin-Pin	35	*40,98*	0.55	4.44	0	0.00	−6.48	0.04	91.90
Signum	Push Through	0	*10,67*	0.00	4.78	0	0.79	−10.65	0.03	138.99

Tabelle 4: Ausschnitt aus der Vorhersage des Decision Tree Modells

(Tabelle zeigt nur ausgewählte und gerundete Features)

einem weiteren Diagramm aus dem SHAP Portfolio analysiert. In dieser Grafik ist zu sehen, dass sich die Features *Acousticness, Instrumentalness, Duration_ms, Speechiness* in ihrer jeweiligen Ausprägung positiv, die Features *Loudness, Explicit* aber negativ auf die zu erwartende Popularität auswirken. Zusätzlich wird durch die Länge des jeweiligen Balkenabschnitts die Wichtigkeit des Features bestimmt. Vor diesem Hintergrund befinden sich die erstgenannten Features bezogen auf die Popularität im positiven Bereich, während die letztgenannten nicht in einer geeigneten Ausprägung vorliegen.

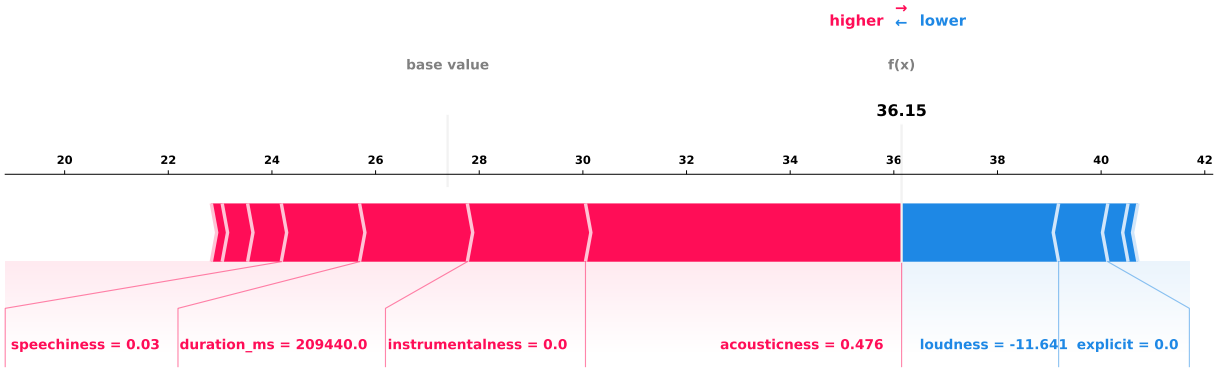


Abbildung 12: Songbeispiel im SHAP Framework

5 Schlussbetrachtung

In diesem Kapitel werden die Ergebnisse der Arbeit, sowie die Grenzen und Limitierungen zusammengetragen. Zudem werden Anregungen und Ausblicke auf die weiteren Forschungsmöglichkeiten im dem Bereich der HSS gegeben.

5.1 Fazit

Ein repräsentativer Datensatz von Spotify wurde explorativ in Unterabschnitt 4.2 untersucht und mit mehreren Verfahren des maschinellen Lernens so bearbeitet, dass nun eine möglichst genaue Vorhersage der zu erwartenden Popularität eines Songs getroffen werden kann. Folgende Ergebnisse können nun präsentiert werden:

1. Die in dieser Arbeit getesteten Algorithmen sind die lineare Regression, der KNN, Random Forest, XGBoost und der Decision Tree. Ihre Performance-Kennzahlen sind in Tabelle 2 zusammengefasst, ebenso in Tabelle 3. Hier wird das Features *Year* jedoch, aus den Gründen, die in Unterabschnitt 4.4 genannt wurden, ausgeschlossen. Die Aussagen beider Tabellen bestätigen, dass der Decision Tree Algorithmus die Popularität am präzisesten vorhersagt. Zudem sind auch die Kennzahlen MAE, MSE und RMSE in beiden Szenarien bei ihm jeweils am niedrigsten, sodass er deutlich am besten von allen in dieser Arbeit verwendeten Algorithmen abschneidet. Durch Hyperparameteroptimierung konnte die Präzision der Vorhersage des Decision Tree Algorithmus noch weiter verbessert werden.
2. Das mit dem Decision Tree Algorithmus errechnete empirische Modell zeigt, dass die Features *Acousticness, Instrumentalness* und *Loudness* den größten Einfluss auf die Vorhersage der Popularität eines Songs nehmen. In dem SHAP Framework in Abbildung 11 sind die einzelnen Feature nach absteigender Wichtigkeit aufgelistet, wobei *Mode* und *Key* keinen Einfluss mehr auf die Popularität aufweisen. Die Frage nach der Ausprägung der jeweiligen Features lässt sich nicht abschließend mit konkreten Zahlen beantworten. Allgemein kann aber festgestellt werden, dass in einem Song explizite Inhalte und höhere *Loudness*-Werte angestrebt und für die Track-Länge sowie das Feature *Energy* ein moderater Mittelwert gewählt werden

sollte. Für die Features *Speechiness, Valence, Liveness* und *Tempo* empfiehlt sich eine eher niedrigere Ausprägung. Für das Charakteristikum *Danceability* kann keine klare Aussage getroffen werden. Für einen Künstler bedeutet dieses Ergebnis, dass ein beabsichtigter Hit eine Mischung aus Gesang und elektronisch bearbeiteter Instrumentalmusik und explizite Anteile aufweisen sollte, nicht zu schnell sein und eine höhere Lautheit besitzen.

5.2 Limitierung

Grundsätzlich ist die Frage, ob eine Vorhersage zur Popularität eines Songs überhaupt mit den bestehenden Features möglich ist, in der Forschung umstritten. Dies zeigen die beiden Artikel „Hit Song Science Is Not Yet a Science“ aus dem Jahr 2008 sowie der Artikel „Hit song science once again a science“ aus dem Jahr 2011. Dabei wird von den Autoren die Featureauswahl und auch Kausalitäten durch verzerrte Experimente kritisch hinterfragt.
Kritisch bei der Verwendung des Datensatzes von Spotify ist die fehlende Klassifizierung in ein Genre, da einzelne Genres einen großen Unterschied in ihren Featureausprägungen aufweisen (Klassische Musik vs. Rock Musik). Spotify bietet bisher nur eine Klassifizierung des gesamten Albums an, aber nicht jedes einzelnen Tracks. Der in dieser Arbeit verwendete Datensatz ist bereits relativ groß, allerdings stammen die Daten alle aus einer Quelle, weshalb der Datensatz schon im Vorhinein beeinflusst sein könnte. Hier ist der bereits in der Einleitung angesprochene Spotify-Algorithmus zu bedenken. Schlussendlich wurden nicht alle möglichen Algorithmen aus dem Bereich des maschinellen Lernens sowie der künstlichen Intelligenz verwendet. Es kann daher durch die Gewinnung weiterer Features und die Nutzung anderer Algorithmen - insbesondere von neuronalen Netzen - andere Ergebnisse gewonnen werden.

5.3 Ausblick

Die *Hit Song Science* bleibt ein spannendes Feld der Forschung. Aktuelle Erkenntnisse wie sie bereits von den Autoren in den verwandten Arbeiten in Bezug auf Deep Learning gewonnen wurden, versprechen bei weiterer Erforschung eine erneute Verbesserung der Vorhersagemodelle. Dabei ist vor allem der Einsatz von CNN zu nennen. Zudem bestehen noch weitere Forschungsmöglichkeiten bezüglich der Auswahl der Features und der weiteren Featuregewinnung u.a. durch Neuro-Linguistisches Programmieren (NLP). Eine in der bisherigen Literatur kaum angesprochene Option bieten die Metadaten der Streaming-Anbieter. Da jeder Nutzer über seine ID identifiziert werden kann, ist es möglich, durch Metadaten-Analyse, einzelne Höhrergruppen besser zu klassifizieren. Die Popularität von Musikstücken könnte somit angepasst an spezifische Gruppen vorhergesagt und daraus resultierend von Künstlern passende Musik für diese Gruppen entworfen werden.
Eine weiterer Gewinn der Metadaten könnte die Analyse von anlassbezogener Musik (Sportplaylist, Einschlafplaylist, etc.) sein. Individuell zugeschnittener Musik und

Künstlern, die in diesem Bereich populär sind, könnte durch Data Mining zu einer noch höheren Popularität in diesen Subgenres verholfen werden.

Literatur

Bergstra, James, Daniel Yamins und David Cox (2013). „Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures“. In: *International conference on machine learning*. PMLR, S. 115–123.

Bishop, Christopher M (2006). *Pattern recognition and machine learning*.

Carbonell, Jaime G, Ryszard S Michalski und Tom M Mitchell (1983). „An overview of machine learning“. In: *Machine learning*, S. 3–23.

Chen, Tianqi und Carlos Guestrin (2016). „Xgboost: A scalable tree boosting system“. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, S. 785–794.

Herremans, Dorien, David Martens und Kenneth Sörensen (2014). „Dance hit song prediction“. In: *Journal of New Music Research* 43.3, S. 291–302.

Howard, Jeremy (2020). *Fast.AI Introduction*. https://github.com/fastai/fastbook/blob/8be580737ee0cc17746a5ed68283150d489b3dc4/01_intro.ipynb. [Online; Stand 11. April 2021].

Jakubowski, Kelly, Sebastian Finkel, Lauren Stewart und Daniel Müllensiefen (2017). „Dissecting an earworm: Melodic features and song popularity predict involuntary musical imagery.“ In: *Psychology of Aesthetics, Creativity, and the Arts* 11.2, S. 122.

Karydis, Ioannis, Aggelos Gkiokas, Vassilis Katsouros und Lazaros Iliadis (2018). „Musical track popularity mining dataset: Extension & experimentation“. In: *Neurocomputing* 280, S. 76–85.

Kwak, Nojun und Chong-Ho Choi (2002). „Input feature selection for classification problems“. In: *IEEE transactions on neural networks* 13.1, S. 143–159.

Léveillé Gauvin, Hubert (2018). „Drawing listener attention in popular music: Testing five musical features arising from the theory of attention economy“. In: *Musicae Scientiae* 22.3, S. 291–304.

Lundberg, Scott und Su-In Lee (2017). „A unified approach to interpreting model predictions“. In: *arXiv preprint arXiv:1705.07874*.

Martin-Gutiérrez, David, Gustavo Hernández Peñaloza, Alberto Belmonte-Hernández und Federico Álvarez Garcia (2020). „A multimodal end-to-end deep learning architecture for music popularity prediction“. In: *IEEE Access* 8, S. 39361–39374.

Ni, Yizhao, Raul Santos-Rodriguez, Matt Mcvicar und Tijl De Bie (2011). „Hit song science once again a science“. In: *4th International Workshop on Machine Learning and Music*. Citeseer.

Pachet, François und Pierre Roy (2008). „Hit Song Science Is Not Yet a Science“. In: *ISMIR*, S. 355–360.

Rap, Wikipedia (2021). *Rap — Wikipedia, Die freie Enzyklopädie*. [Online; Stand 30. März 2021]. URL: https://de.wikipedia.org/w/index.php?title=Rap&oldid=209817843.

Rock'n'Roll (2021). *Rock 'n' Roll — Wikipedia, Die freie Enzyklopädie*. [Online; Stand 30. März 2021]. URL: https://de.wikipedia.org/w/index.php?title=Rock_%E2%80%99n%E2%80%99_Roll&oldid=210248821.

Shmueli, Galit und Otto R Koppius (2011). „Predictive analytics in information systems research“. In: *MIS quarterly*, S. 553–572.

Statista (2020). *Marktanteile der einzelnen Anbieter an den zahlenden Abonnenten von Musikstreaming weltweit im 1. Quartal 2020*. https://de.statista.com/statistik/daten/studie/671214/umfrage/marktanteile-der-musikstreaming-anbieter-weltweit/. [Online; accessed 15-April-2021].

Wikipedia KNN (2021). *K-nearest neighbors algorithm — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=K-nearest_neighbors_algorithm&oldid=1016386199. [Online; accessed 11-April-2021].

Yang, Li-Chia, Szu-Yu Chou, Jen-Yu Liu, Yi-Hsuan Yang und Yi-An Chen (2017). „Revisiting the problem of audio-based hit song prediction using convolutional neural networks“. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, S. 621–625.

Yu, Haiqing, Yanling Li, Shujun Zhang und Chunyan Liang (2019). „Popularity Prediction for Artists Based on User Songs Dataset“. In: *Proceedings of the 2019 5th International Conference on Computing and Artificial Intelligence*, S. 17–24.

A Anhang A

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die Zitate deutlich kenntlich gemacht zu haben.
Ich erkläre weiterhin, dass die vorliegende Arbeit in gleicher oder ähnlicher Form noch nicht im Rahmen eines anderen Prüfungsverfahrens eingereicht wurde.

Würzburg, den 17. April 2021 VORNAME NACHNAME