Duale Hochschule Baden-Württemberg Mannheim

Seminararbeit

Autorenklassifikation

Studiengang Wirtschaftsinformatik

Studienrichtung Data Science

Verfasser(in):	Anabel Lilja (4279481)
	Bjarne Gerdes (8608827)
	Johannes Deufel (5610649)
	Simone Marx (6147264)
Kurs:	WWI18DSB
Studiengangsleiter:	Prof. Dr. Bernhard Drabant
Wissenschaftlicher Betreuer:	Mannfred Preisendörfer
Bearbeitungszeitraum:	18.11.2020 - 26.01.2020

Ehrenwörtliche Erklärung

Wir versichern hiermit, dass wir die vorliegende Arbeit mit dem Thema: *<Titel Ihrer Arbeit>* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ort, Datum Anabel Lilja

Ort, Datum Bjarne Gerdes

Ort, Datum Johannes Deufel

Ort, Datum Simone Marx

Kurzfassung (Abstract)

Hier können Sie die Kurzfassung (engl. Abstract) der Arbeit schreiben. Beachten Sie dabei die Hinweise zum Verfassen der Kurzfassung.

Inhaltsverzeichnis

Abbildungsverzeichnis **iv**

Tabellenverzeichnis **v**

Quelltextverzeichnis **vi**

Algorithmenverzeichnis **vii**

Abkürzungsverzeichnis **viii**

1 Einleitung **1**
1.1 Relevanz von Autorenklassifikation 1
1.2 Zielsetzung und Aufbau der Ausarbeitung 2

2 Theoretische Grundlagen **4**
2.1 Ausgewählte Konzepte des Machine Learning 4
2.2 Related Work 6

3 Praktische Umsetzung **8**
3.1 Vorverarbeitung 8
3.2 Linear Support Vector 10
3.3 Stochastic Gradient Descent 11
3.4 Extreme Gradient Boosting 11

4 Diskussion der Ergebnisse **13**
4.1 Bewertung der Modelle für das Feature Age 14
4.2 Bewertung der Modelle für das Feature Gender 14
4.3 Bewertung der Modelle für das Feature Sternzeichen 14

5 Schlussbetrachtung **15**
5.1 Fazit 15
5.2 Ausblick 15

A Tabellen **16**
A.1 Feature Engineering 16

B Beispiel-Anhang: Noch ein Testanhang **17**

Literaturverzeichnis **18**

Abbildungsverzeichnis

Tabellenverzeichnis

Quelltextverzeichnis

Algorithmenverzeichnis

Abkürzungsverzeichnis

BoW	Bag of Words
DHBW	Duale Hochschule Baden-Württemberg
KNN	K-Nearest Neighbors
NLP	Natural Language Processing
SVM	Support Vector Machine
TF-IDF	Term Frequency - Inverse Document Frequency

1 Einleitung

Die vorliegende Ausarbeitung beschäftigt sich mit der Thematik der Autorenklassifikation im Bereich des Machine Learning. Im Rahmen dieser Ausarbeitung werden Machine Learning Modelle erstellt, welche Anhand eines Input-Textes das Alter, Geschlecht und Sternzeichen eines Autors, sowie das Jahr der Veröffentlichung und das Genre seines Textes bestimmen können. Hierfür werden verschiedene Klassifikatoren verwendet, sodass die Vorgehensweisen, die Ergebnisse und die Erkenntnisse dieser miteinander verglichen werden können. Ebenfalls wird die bestehende Literatur dieses Forschungsgebiets untersucht, wodurch die entwickelten Ansätze dieser Ausarbeitung in den aktuellen Forschungsstand eingeordnet werden können.

1.1 Relevanz von Autorenklassifikation

Die Basis dieser Ausarbeitung findet sich in der Wissenschaft der Stilometrie. Die Stilometrie ist eine Disziplin, welche Untersuchungen von Sprachstilen mithilfe statistischer Mittel durchführt. Sie vertritt dabei die Annahme, dass jeder Autor einen individuellen Schreibstil besitzt, welcher durch die unbewusste, aber konsistente Verwendung bestimmter Wortmuster entsteht. Die Verwendung solcher Muster erstreckt sich dabei über alle verfassten Dokumente eines Autors über die Zeit hinweg [8]. Dies führt dazu, dass sich verschiedene Schreibstile anhand statistischer Methoden numerisch abgrenzen lassen und ermöglicht die Zuordnung von Autoren und Texten mithilfe gängiger Indikatoren wie die Wort- und Satzlänge, der Wortschatz, die Häufigkeit von Worten und der Zusammenhang zwischen den verwendeten Wörtern. Es ist ebenfalls möglich, gewisse Eigenschaften über Autoren und Texte zu bestimmen [3, 2].

Die Kenntnis des Profils eines Autors und die Zuordnung zu Texten kann in den unterschiedlichsten Bereichen von hoher Bedeutung sein. So bedient sich die forensische Linguistik den Mitteln und Erkenntnissen der Stilometrie, um beispielsweise das sprachliche Muster einer verdächtigen Textnachricht zu analysieren und dieses mit verdächtigen Personen oder einer Datenbank abzugleichen [11]. Hierbei lassen sich außerdem Rückschlüsse auf bestimmte Charakteristiken wie das Alter, Geschlecht, familiärer und gesellschaftlicher Hintergrund, Umgangs- und Muttersprache oder Interessen ziehen, welche bei polizeilichen Ermittlungen von erheblichem Vorteil sein können [10].

Charakteristiken dieser Art über Einzelpersonen oder Personengruppen sind auch aus Marketingperspektive interessant. Unternehmen benötigen solche Informationen über ihre Kunden sowohl für die Kundenbindung als auch für die Kundengewinnung. Sie ermöglichen

unter anderem die Bereitstellung personalisierter Produkte und Dienstleistungen und das damit einhergehende personalisierte Marketing [6, 12].

Ebenfalls hilfreich und notwendig ist die Autorenklassifikation in Büchereien und Bibliotheken, sowie für Historiker. Die Zuordnung von Autoren und Werken, sowie die Klassifikation dieser in bestimmte Genres oder Jahrgänge kann eine große Hilfe bei der Verwaltung von Büchereien und Bibliotheken darstellen [13, 9]. Des Weiteren kann diese Klassifikation Historikern einen hohen Mehrwert bieten, da zum Beispiel unbekannte Werke in eine bestimmte Epoche oder eine bestimmte Region eingeordnet werden können, wodurch historische Rückschlüsse über diese Zeit oder Gesellschaft gezogen werden können. Dies kann auch in der heutigen Zeit einen hohen Wert haben, da Sentimentanalysen und Klassifikationen ebenfalls Rückschlüsse über soziales und psychologisches Verhalten in der Gegenwart bringen können [1].

Aus akademischer Sicht kann die Autorenklassifikation vor allem bei der Erkennung von Plagiaten behilflich sein [14]. Die Übernahme fremder geistiger Leistung verstößt in der Wissenschaft gegen Prüfungsordnungen, Arbeitsverträge oder Universitätsrechte und kann durch Machine Learning Klassifikatoren zur Autorenzuordnung leichter erkennbar gemacht werden.

1.2 Zielsetzung und Aufbau der Ausarbeitung

Das Ziel dieser Ausarbeitung ist es, das Alter, Geschlecht und Sternzeichen eines Autors, sowie das Jahr der Veröffentlichung und das Genre seines Textes anhand eines Textinputs mithilfe unterschiedlicher Machine Learning Modelle zu bestimmen.

Kapitel 2 dieser Ausarbeitung dient der Definition der theoretischen Grundlagen. Hier wird insbesondere auf ... eingegangen, da diese Verfahren und Methoden im folgenden Kapitel 3 Anwendung finden. Im Zuge dessen wird außerdem die bestehende Literatur zu dieser Thematik analysiert und verschiedene Ansätze beschrieben, sodass aktuelle Vorgehensweisen, Ergebnisse und Erkenntnisse mit unseren verglichen werden können.

Die Modellerstellung erfolgt in Kapitel 3 dieser Ausarbeitung. Zur Erstellung wird der *Blog Authorship Corpus* verwendet, welcher im Jahr 2017 auf der Website der Online-Community Kaggle veröffentlicht wurde. Der Datensatz umfasst insgesamt 681.288 Blogeinträge von 19.320 Verfassern aus dem August 2004.

Die tatsächliche Umsetzung kann in sieben Schritte eingeteilt werden. Als erster Schritt wird Feature Engineering betrieben, um Merkmale aus den Rohdaten zu extrahieren. Im Rahmen dieser Ausarbeitung werden 16 Merkmale untersucht (siehe Anhang A1), darunter

Aspekte wie die Textlänge, die Anzahl der Emails im Text und der Anteil der Großbuchstaben an allen Zeichen. Diese Merkmale können anschließend verwendet werden, um die Leistung der zu erstellenden Machine Learning Algorithmen zu verbessern.

Der zweite Schritt des Cleaning beinhaltet die Entfernung jeglicher Unreinheiten in den Daten, sodass im weiteren Verlauf mit einem bereinigten Datensatz gearbeitet werden kann.

Anschließend werden die Daten im dritten Schritt transformiert, um sie in ein für die Modelle lesbares Format zu bringen. Im Rahmen dieser Ausarbeitung wird dabei zwischen Textdaten und anderen Daten unterschieden und je nach vorhandenem Datentyp eine geeignete Form der Datentransformation vorgenommen.

Anschließend an die Datentransformation werden sowohl supervised als auch unsupervised Machine Learning Verfahren verwendet, um unterschiedliche Modelle zu trainieren und diese mit weiteren Features anzureichern.

Im sechsten Schritt wird Parametertuning angewendet, um die optimalen Parameter für die erstellten Modelle zu finden. Anschließend werden die Modelle im siebten Schritt anhand unterschiedlicher Evaluationsmetriken bewertet und gewichtet anhand ihrer Verluste kombiniert.

Die finalen Modelle werden dann mittels einer flask-API an ein im Rahmen dieser Ausarbeitung entwickeltes Frontend angebunden. Über ein Eingabefeld auf der Website kann somit Text eingelesen werden, anhand dessen die angebundenen Modelle Vorhersagen über das Alter, Geschlecht und Sternzeichen eines Autors, sowie das Jahr der Veröffentlichung und das Genre des Textes treffen können.

Anschließend an die Entwicklung der Modelle beinhaltet Kapitel 4 die Diskussion der Entwicklungsergebnisse. Zum einen wird hier auf die Performance der Modelle eingegangen, welche anhand verschiedener Evaluationsmetriken bewertet werden kann. Die erstellten Modelle werden dabei miteinander, aber auch mit anderen Modellen aus der Literatur verglichen und in den aktuellen Forschungsstand eingeordnet. Zum anderen wird ein besonderer Fokus auf die Analyse der Vorgehensweise und die daraus gewonnen Erkenntnisse gelegt, da diese einen hohen Mehrwert für zukünftige Arbeiten dieser Art haben können.

Abschließend bietet die Schlussbetrachtung eine Zusammenfassung der Ergebnisse und Erkenntnisse, sowie einen zukunftsorientierten Ausblick über die Entwicklung des Forschungsobjektes.

2 Theoretische Grundlagen

Als Basis für die in Kapitel 3 folgende Erstellung von Machine Learning Modellen zur Klassifikation von Autoren werden in diesem Kapitel ausgewählte theoretische Konzepte des Machine Learnings und der Autorenklassifikation erläutert. Außerdem werden Vorgehensweisen und Forschungsergebnisse aus der bestehenden Literatur untersucht. Hierbei werden sowohl ähnliche, als auch von unserer Vorgehensweise abweichende Methoden beleuchtet.

2.1 Ausgewählte Konzepte des Machine Learning

Test und Train Split:
Ein Datensatz wird überlicherweise in Test- und Trainingsdaten unterteilt. Mit den Trainingsdaten wird ein Modell erstellt, welches bestimmte Parameter erlernt. Die Testdaten werden verwendet, um die Qualität des Modells zu bewerten, indem versucht wird, die Zielvariable der Testdaten mit dem Modell vorherzusagen. Es zeigt also, wie gut das erstellte Modell Vorhersagen auf Daten treffen kann, die nicht im Training verwendet wurden. Die Verwendung dieses Splits ist nicht empfohlen, wenn ein besonders kleiner oder unausgeglichener Datensatz vorliegt.

Supervised und unsupervised learning:
Supervised und unsupervised learning sind zwei unterschiedliche Lernansätze des Machine Learning. Beim supervised learning handelt es sich um das sogenannte überwachte Lernen. Hier sind die Trainingsdaten mit einem Label versehen, was bedeutet, dass Funktionen auf Basis von bereits bekanntem Output ermittelt werden. Anwendungsfälle für diese Lernart sind Neuronale Netze, Decision Trees und Lineare Regressionen. Im Gegensatz dazu ist es das Ziel des unsupervised learnings, also des unüberwachten Lernens, aus Daten unbekannte Muster zu erkennen und Regeln aus diese abzuleiten. Die vorhandenen Daten sind hierbei also nicht mit einem Label versehen. Anwendungsfälle für diese Lernart sind das Clustering und die Density Estimation.

Regression:
Die Regression beschreibt im Bereich des Machine Learning den Versuch, Beziehungen zwischen einer Abhängigen und einer oder mehreren unabhängigen Variablen zu modellieren. Ziel ist die quantitative Beschreibung von Zusammenhängen, sowie die Prognose von Werten der abhängigen Variable. Bei den Zielvariablen handelt es sich um kontinuierliche oder numerische Variablen. Bekannte Regression-Methoden sind die Lineare Regression, Lineare Support Vector Machine (SVM) und Regression Trees.

Classification:
Im Bereich des Machine Learning versucht die Classification anhand einer Funktion von Eingangsvariablen auf diskrete oder kategoriale Zielvariablen zu schätzen. Man unterscheidet je nach Anzahl an Eingangsvariablen in binäre und multivariate Classification. Bekannte Classification-Methoden sind Logistic Regression, Naive Bayes, Decision Trees und K-Nearest Neighbors (KNN).

Hyperparameter Optimization:
Hyperparamter sind alle Parameter, die vom Benutzer vor Beginn des Trainings willkürlich eingestellt werden können und bestimmen die grundlegende Struktur des Modells. Ziel ist es, die optimale Kombination der Parameterwerte zu finden, sodass entweder der Verlust der Funktion minimal oder die Accuracy der Funktion maximal wird. Dies ist vor allem für den Vergleich der Güte von verschiedenen Modellen auf einen Datensatz hilfreich. Es gibt verschiedene Verfahren zur Optimierung der Hyperparameter, wie beispielsweise Manual Search, Random Search oder Grid Search.

Cross Validation
Die Cross Validation ist eine statistische Methode, die zur Einschätzung der Fähigkeiten von Modellen des maschinellen Lernens verwendet wird. Dabei wird der Datensatz in k gleichgroße Partitionen unterteilt. Für jede Partition wird die Partition als Testdatensatz und der Rest der Partitionen als Trainingsdatensatz verwendet. Es wird ein Modell an den Trainingsdatensatz angepasst und anhand des Testdatensatzes evaluiert. Die Evaluationsergebnisse der einzelnen Modelle fassen die Fähigkeit des Modells zusammen, indem die Gesamtfehlerquote als Durchschnitt aus den einzelnen Fehlerquoten der k Partitionen berechnet wird. Diese Art der Validation ist vor allem bei kleinen Datensätzen von Vorteil, da sie diese optimal nutzt.

Grid Search:
Grid Search ist ein Verfahren zur Optimierung der Hyperparameter eines Machine Learning Modells. Hierbei wird ein Raster an Hyperparameter gebildet. Das Modell wird dann auf jede mögliche Kombination von Hyperparametern in diesem Raster trainiert und getestet. Die wichtige Komponente ist dabei die Auswahl der Grid Parameter.

Content-based und syntactical features:
Im Allgemeinen kann man bei der Autorenklassifikation in zwei Kategorien an Features unterscheiden, die unabhängig voneinander oder aber kombiniert untersucht werden können. Hierbei handelt es sich um inhaltsbasierte (content-based) und syntaktische Methoden. Inhaltsbasierte Methoden legen den Fokus auf lexikalische Merkmale, also den Inhalt eines Textes. Es wird vor allem auf die Semantik von Wörtern und Sätzen geachtet. Im Gegensatz dazu fokussiert sich die syntaktische Methode auf die Wörter und Zeichen in einem Text, sowie ihre Relation zueinander.

2.2 Related Work

Aufgrund des hohen Nutzens, den eine erfolgreiche Autorenklassifikation in den unterschiedlichsten Anwendungsbereichen erbringen kann, gibt es eine umfassende Anzahl an wissenschaftlichen Veröffentlichungen mit Lösungsansätzen im Bereich des Machine Learning.

Fatima et al. [4] erstellten zur Bewältigung dieser Thematik den *SMS-AP-18 Corpus*. Der Corpus enthält 810 Autorenprofile, wobei jedes Profil aus einer Aggregation von SMS-Nachrichten eines einzelnen Autors und sieben demographischen Merkmalen besteht. Die SMS-Nachrichten sind in Englisch und Roman Urdu verfasst. Der Datensatz wurde unter Anwendung von stilometrie- und inhaltsbasierten Methoden untersucht, um das Alter und Geschlecht der Autoren zu bestimmen. Die stilistischen Verfahren konzentrierten sich auf lexikalische Wörter, sowie die Anzahl von Absätzen und Verben. Die inhaltsbasierte Methode nutze N-Grams zur Identifikation von Alter und Geschlecht. Die Ergebnisse zeigen, dass das inhaltsbasierte Methode mit 5-N-Grams einen F1-Score von 0,947 und eine Accuracy von 0,975 erreichte und damit alle anderen Verfahren dieser Studie übertreffen konnte.

Devi et al. [15] haben ebenfalls eine Autorenklassifikation mit den Merkmalen Alter und Geschlecht durchgeführt und nutzten dafür 500 Autorenprofile aus dem *SMS-AP-18 Corpus*, wovon 150 zum Testen und 350 zum Trainieren der Modelle verwendet wurden. Das Merkmal Geschlecht hatte die Ausprägungen *männlich* oder *weiblich*. Das Alter wurde in drei Kategorien eingeteilt: 15-19 jährige, 20-24 jährige und 25-xx jährige. Inhaltsbasierten Methoden wie Bag of Words (BoW) und Term Frequency - Inverse Document Frequency (TF-IDF) wurden verwendet, um Features aus dem Datensatz zu extrahieren. SVM wurde für die Klassifizierung genutzt. Eine 10-fold cross-validation diente der Indentifikation der besten Modelle. Die Perfomance der erstellten Modelle wurde anhand ihrer Accuracy gemessen. Das beste Modell hat für das Merkmal Alter eine Accuracy von 0,857 und für das Merkmal Geschlecht von 0,643. Die gemeinsame Accuracy liegt bei 0,536.

Azmi und Al-Ghadir [5] untersuchten das Geschlecht von arabischen Social Media Nutzern. Hierfür wurden Posts von den beliebtesten Social Media Seiten in Saudi Arabien extrahiert. Zunächst wurden die Daten im Preprocessing bereinigt. Dabei wurden alle Ziffern, Sonderzeichen, diakritischen Markierungen nicht-arabische Wörter entfernt. Außerdem wurden alle Wörter normalisiert und tokenisiert. TF-IDF-Maßnahmen wurden verwendet, um die Gewichtung einzelner Wörter in den Posts zu bestimmen. Anschließend wurde anhand dieser Gewichtung eine top-k Liste für Wörter und eine top-k Liste für Stämme erstellt. Top-k ist also ein Merkmalsvektor, der die Einträge mit der höchsten k-Rangfolge enthält. Zur Bestimmung des Geschlechts nutzten sie SVM und KNN. Zur Evaluierung der Performance wurde ein Subdatensatz von 1200 Einträgen genutzt. Eine 10-fold cross-validation wurde zur Identifikation der besten Modelle verwendet. Das KNN-Modell hat mit 0,9316 eine höhere Accuracy als das SVM-Modell mit 0,8733 erreicht.

Modaresi et al. [7] fokussierten sich im Rahmen der PAN Author Profiling Challenge 2016 auf die genreübergreifende Alters- und Geschlechtsidentifikation. Der bereitgestellte Trainingsdatensatz umfasste 463 Dokumente an Englischen Tweets, 250 Dokumente an Spanischen Tweets und 384 Dokumente an Niederländischen Tweets, wobei ein Dokument alle Tweets eines Autors umfasste. Um die erstellten Modelle auf andere Genres als Twitter zu evaluieren wurde der PAN2014 verwendet, welcher Posts aus anderen Social Media Plattformen enthält. Da das Genre der Trainings- und Testdaten nicht identisch ist, wurden die Trainingsdaten so bearbeitet, dass die meisten genrespezifischen Informationen eliminiert wurden. So konnte das Risiko von Overfitting auf andere Genres als Twitter reduziert werden. Für die Feature extraction wurden Word Unigrams, Word Bigrams, Character 4-Grams, der Average Spelling Error und Punctation Feautres verwendet. Das finale Modell wurde mit einer Logistic Regression trainiert, da dieser Klassifikator einen vergleichsweise hohen Bias bzw. eine geringe Varianz hat. Da es sich um eine multivariate Klassifikation handelte, wurde die one-vs-rest Methode verwendet. Modaresi et al. haben außerdem Gradient Boosting und Random Forests als Klassifikatoren ausprobiert. Die mit der Logistic Regression erzielten Ergebnisse waren diesen beiden Klassifikatoren überlegen. Zur Evaluation der Modelle wurde eine 10-fold cross-validation auf dem Twitterdatensatz durchgeführt. Die höchste Accuracy für Geschlecht wurde mit 0,7564 für englische Posts erreicht. Die höchste Accuracy für das Alter wurde mit 0.5179 für spanische Posts erreicht. Als besonders erfolgreich konnte die gemeinsame Accuracy von 0,4286 für das Geschlecht und Alter von spanischen Posts gewertet werden.

Im Allgemeinen kann festgehalten werden, dass das Vorgehen bei allen betrachteten Studien sehr ähnlich ist. Unabhängig von der Größe des Datensatzes, der Sprache und der zu bestimmenden Merkmale begannen alle betrachteten Vorgehen mit einem Pre-Processing der Daten. Im nächsten Schritt wurden unterschiedliche Arten des Feature Engineering betrieben, wobei die Nutzung von TF-IDF besonders häufig vergekommen ist. Im Rahmen der Modellerstellung wurde der Datensatz in Trainings- und Testdaten unterteilt. Unterschiede sind vor allem bei den verwendeten Klassifikatoren ersichtlich. Genutzte Klassifikatoren waren SVM, Logisitc Regression, KNN, Gradient Boosting und Random Forests.

3 Praktische Umsetzung

Um die zuvor theoretisch erläuterten Konzepte im Folgenden praktisch zur Anwendung bringen zu können, ist es zunächst notwendig, den Datensatz zu beleuchten und die darin enthaltenen Daten vorzuverarbeiten.

Der in diesem Projekt vorwiegend verwendete *Blog Authorship Corpus* enthält 681.288 Blogeinträge aus dem Forum *blogger.com*. Über den reinen Textinhalt des Posts hinaus enthält er zudem Metadaten über 19.320 Verfasser. Diese Metadaten umfassen das Geschlecht, Alter und Sternzeichen, sowie das Genre und Veröffentlichungsdatum des Posts. Jeder Blogeintrag kann dabei eindeutig einem Autor zugeordnet werden. Dieser Datensatz wurde im Jahr 2004 zusammengetragen und gilt seither als renommierte Datenbasis für verschiedene Natural Language Processing (NLP)- und Klassifizierungsprojekte. //

Noch vor der Vorverarbeitung des Datensatzes muss jedoch das Feature Engineering erfolgen. Hierbei wurden in Kollaboration, die in Anhang 1 XXX beigefügten und erläuterten Features erarbeitet. Dabei waren einerseits persönliches Interesse, andererseits aber auch die theoretischen Grundlagen der Stilometrie von Texten die hauptausschlaggebenden Faktoren.

3.1 Vorverarbeitung

Für dieses Projekt ist eine individuelle Vorverarbeitung unabdinglich, um die Basis für den folgenden Entwicklungsprozess zu legen.

Im Schritt des Cleaning werden dafür zum einen Verlinkungen aus den Blogeinträgen entfernt, um lediglich die Inhalte der verschiedenen Einträge eigenständig betrachten zu können. Zum anderen wird auch die Zeichencodierung von der ursprünglich bestehenden HTML-Codierung zu einer UTF-8-Codierung abgeändert. UTF-8 ist eine der gängigsten Unicode Codierungen und daher für die folgende Verarbeitung optimal geeignet.
Als nächstes werden so genannte Stopwords wie zum Beispiel „me“, die vordefiniert sind und sich an den gängigen Standards orientiert und auch die Sonderzeichen aus dem Text entfernt, da diese nicht relevant für die Informationsgewinnung sind. Sie kommen sehr oft und auch in verschiedensten Kontexten vor und sind vor alle der grammatikalischen Richtigkeit dienlich, die für die folgenden Schritte nicht relevant ist. Sie würden die Verarbeitung daher vor allem verzerren. Zur Unterstützung dieser Textbereinigung wird die Bibliothek TextBlob verwendet, um die unerwünschten Inhalte zu entfernen.

Die Python Bibliothek spaCy unterstützt im letzten Schritt des Pre-Processing die Lemmatisierung der Wörter in den Blogeinträgen. Hierbei werden sie in ihre Grundform zurückgeführt und grundsätzlich klein geschrieben. Das dient der Verringerung der Heterogenität der Blogeinträge und steigert somit die Vergleichbarkeit, da sie nun eine gleichförmige, erkennbare Form haben. Im folgenden Verlauf wird diese Einheitlichkeit z.B. die Verwendung des Sprachfilters ermöglichen, der XXX ebenfalls durch die Voreinstellung der Sprache Englisch in der Bibliothek spaCy problemlos umgesetzt wird.

Die zuvor beschriebenen Schritte sind jedoch nicht ausreichend, um mit der Modellerstellung zu beginnen. Daher wird im Folgenden der Vorgang der Transformation beschrieben. Grundsätzlich wird bei der Modellerstellung auf zwei grundsätzlich verschiedenen Datensätzen gearbeitet, um eine größere Diversität zu gewährleisten und dadurch unterschiedliche Vorgehensweisen im späteren Verlauf umsetzten zu können. Dies ist nötig, um sowohl numerische als auch textuelle Features sachgerecht durch unterschiedliche Classifier in die verschiedenen Modelle miteinzubeziehen. Bevor diese grundsätzlich verschiedenen Datensätze erstellt werden, wird noch der vorverarbeitete Trainingsdatensatz zufällig in je einen Trainings- und Testdatensatz verteilt. Danach kommen verschiedene so genannte „Transformer“ zum Einsatz, um die heterogenen Daten nach numerischen bzw. textuellen Features der Metadaten trennt. Der textuelle Transformer ist definiert durch die Methode TfidfVectorizer, um die nominalskalierten Daten verarbeiten zu können.
Der sogenannte Term-Frequency-Invers-Document-Frequency (TF-IDF) ist bestimmt durch die beiden Methoden TF und IDF. TF setzt dabei die Wortanzahl eines gewählten Wortes mit allen Wörtern im Blogeintrag ins Verhältnis und stellt somit die relative Wahrscheinlichkeit dar, das ein zufälliges Wort mit dem zuvor gewählten Wort übereinstimmt. IDF setzt dagegen logarithmisch die Anzahl an Blogeinträgen mit der Gesamtanzahl der Blogeinträge, die ein bestimmtes Wort enthalten ins Verhältnis und zeigt somit die Häufigkeit der Wörter Blogeintrag übergreifend auf. TF-IDF wird durch TF*IDF berechnet und ist folglich ein Indikator für die Relevanz der Worte in den verschiedenen Blogeinträge in dem gesamten Datensatz. Diese gesamten Informationen werden in einem Vector gespeichert.
Dieser Transformator ist aufgrund seiner umfassenden Betrachtung der gesamten Kollektion der Blogeinträge, jedoch auch der Betrachtung jedes einzelnen Blogeintrags in Bezug auf die verwendeten Wörter optimal für den Anwendungsfall geeignet. Andere Methoden wie z.B. ein Count Vectorizer betrachtet einzig und allein die Verwendung der verschiedenen Wörter in einem einzelnen Datensatz. Dadurch wäre es entweder nicht möglich ohne weitere Methoden die verschiedenen Blogeinträge übergreifend zu bewerten oder eben nach den einzelnen Einträgen zu differenzieren. Angesichts des wissenschaftlichen Anspruchs dieser Arbeit wurde jedoch auch in Teilen der IDF-Faktor des TF-IDF Transformators ausgeschlossen, um nachzuvollziehen, ob dieser überhaupt einen nützlichen Einfluss auf das Ergebnis hat. Darauf wird in der Bewertung weiter eingegangen.

Der numerische Transformator bedient sich bei kategorialen Features des OneHotEncoders, bei z.B. fortlaufenden numerischen Ausprägungen des Standardscalars. Diese Unterschei-

dung ist nötig, da kategoriale Features entweder ordinal- oder kardinalskaliert sein können und dadurch eine Unterscheidung in der weiteren Verarbeitung unabdinglich ist. One Hot Encoder ordnen die verschiedenen ordinalskalierten Ausprägungen des jeweiligen Features ihren entsprechenden Kategorien zu. Standardscalar hingegen standardisiert die verschiedenen kardinalskalierten Ausprägungen so. dass sich der Mittelwert bei 0 befindet und die Standardabweichung 1. Dadurch ergibt sich eine bessere Vergleichbarkeit und dadurch eine erleichterte Bewertbarkeit der Ausprägungen, da sie sich im Bereich von -1 bis 1 befinden.

Diese Transformer werden durch die Sklearn Methode make.column.transformer zu Tupeln mit dem Inhalt (Transformer, Datenanteil) ihrem Datenanteil zugewiesen und auf diesen angewendet. So erhalten wir Datensätze getrennt nach numerischen oder textuellen Features. Diese Teildatensätze X bzw. Y können nun auf die verschiedenen Arten der Features passend weiterverarbeitet und in Modellen berücksichtigt werden.

3.2 Linear Support Vector

Im Folgenden wird erklärt, wie die verschiedenen Modelle erstellt werden. Dabei wird die zunächst die Methode LinearSV und ihre Anwednung auf textuelle und numerische Features erklärt. Grundsätzlich ist es wie auch bei der klassischen linearen Regression das Ziel der LSV XXX, die gegebenen Datenpunkte durch eine lineare Funktion darzustellen. Dazu wird jedoch nicht der OLS verwendet, sondern die L2-Norm des Koeffizientenvektors XXX. Da der Error nun nicht direkt zur Erstellung der Funktion verwendet wird, kann er als vorausgesetzter Wert angegeben werden, um die Accuracy des Modells zu verbessern. Somit kann man durch den angegebenen Error die Datenpunkte mit einer gewissen Toleranz betrachten. Das führt zu einer Modellverbesserung, da die Datenpunkte durch diese Toleranz besser beschrieben werden, als durch eine einfache lineare Funktion. Durch verschiedene Hyperparameter wie z.B. Schlupfvariablen, können auch noch die Datenpunkte miteinbezogen werden, die nicht in den Toleranzbereich um die Funktion fallen. Schlupfvariablen erfassen die Informationen dieser Datenpunkte durch ihre Abstände vom tolerierten Bereich bzw. der nächsten Grenze dieses. Zusätzlich zu der L2-Norm wollen wir auch diese Schlupfvariablen minimieren. Zudem wird auch der zuvor beschriebene angegebene Error bzw. die daraus entstehenden beiden Toleranzgrenzen als Hyperparameter betrachtet. Dieser und die Schlupfvariable werden in Abhängigkeit zu einander folglich optimiert, sodass der Bereich um die lineare Funktion die Datenpunkte bestmöglich abdeckt, ohne zu ungenau zu werden. Wird der Parameter des Toleranzbereichs zu groß gewählt, ist dieser sehr groß und der Wert der Schlupfvariable sehr klein und es geschieht ein underfitting. Sind die Grenzen des Toleranzbereichs sehr klein gewählt, ist der Wert der Schlupfvariable sehr groß und es passiert ein overfitting. Dazwischen muss ein passender Kompromiss gefunden werden, um die Vorhersage des Modells zu optimieren. Diese Optimierung kann z.B. durch

eine Gridsearch durchgeführt werden. Somit ist eine LSVR eine komplexere Erweiterung der klassischen Regression durch Toleranzen und einen festgesetzten Error und dadurch gut als Algorithmus für dieses Projekt geeignet.
Support Vectors können entweder als Regressor dienen, indem sie für numerische Daten das oben beschriebene Modell als Funktion für den Output eines fortlaufenden Wertes berechnen, oder als Klassifikator. Als Klassifikator dient das Modell als „Trennlinie“ zwischen den verschiedenen Klassen. Bei einer binären Klassifikation wird dabei nur eine Funktion mit Toleranzbereich berechnet, bei einer muticlass Klassifikation demensprechend mehrere. Je ein optimiertes Modell wird für jedes der numerischen und textuellen Features auf dieser Grundlage trainiert und getestet. Dies dient folgend dem Vergleich mit den beiden anderen Modellarten, die durch andere Methoden trainiert werden.
(https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2)

3.3 Stochastic Gradient Descent

Die zweite Methodik zur Erstellung der Modelle ist der SGD. Grundsätzlich minimiert der Gradient Descent eine Funktion. Über verschiedene Iterationen wird ein neuer Funktionswert, der näher am Minimum liegt berechnet. Dies geschieht, indem die Ableitung der zu minimierenden Funktion an einem gegebenen Punkt multipliziert mit der definierten Learningrate alpha, als „Schrittweite“ von dem vorherigen Punkt abgezogen wird. Die letzte Iteration ist dadurch definiert, dass das Minimum erreicht ist oder nicht besser erreicht werden kann. Dabei kann die Funktion nahezu beliebig viele Feature Ausprägungen haben, die dann als Dimensionen bzw. Funktionswerte verstanden werden.
Der Stochastic Gradient Descent setzt dabei die Besonderheit um, dass nur ein kleiner Anteil der Datenpunkte zufällig gewählt und durch den beschriebenen Algorithmus verarbeitet wird. Dies wird so oft durchgeführt, bis das Ergebnis der verschiedenen Durchführungen auf der kleinen Datenmenge gegen ein Ergebnis konvergiert.
Der SGD kann sowohl als Regressor als auch als Klassifikator verwendet werden und erstellt somit für jedes numerische und textuelle Feature ein nach dieser Methode optimales Modell, das wiederum mit den Modellen der anderen Methoden im nächsten Kapitel evaluiert wird. (https://venali.medium.com/conventional-guide-to-supervised-learning-with-scikit-learn-stochastic-gradient-descent-sgd-14068f286a7f)

3.4 Extreme Gradient Boosting

XGBoost ist eine erweiterte Version des Gradient Boosting. Das Gradient Boosting grundsätzlich erzeugt eine große Anzahl verschiedener „einfacher“ Modelle auf kleinen Anteilen

des Datensets und kombiniert diese wie in einem Regressiontree bzw. Decisiontree. Diese Modelle werden dann nach ihren Vorhersagen und ihrem Loss gewichtet. Die schlechter Abschneidenden Modelle werden in den folgenden Schritten intensiver trainiert, als diese die von Anfang an gute Vorhersagen treffen. Über den Gradienten wird auch hier wie im Gradient Descent die Lossfunction der verschiedenen Modelle bestmöglich minimiert. Schlussendlich werden die verschiedenen Modelle kombiniert, um ein bestmögliches Ergebnis bei der Vorhersage zu erzielen.
XGBoost ist eine spezielle Umsetzung dieses Ansatzes, die statt des Gradienten Verfahrens das Newton Verfahren verwendet und dadurch über mehr Informationen wie z.B. die Annäherung an die Lossfunction über zweite Ableitung verfügt und dadurch effektiver das Minimum findet. Zudem verwendet XGBoost die Regulierungsterme L1 und L2. Das führt zu einer besseren Verallgemeinerung des schlussendlichen Modells, da nicht nur der Error betrachtet wird.
Hier kann je nach Feature über den Regressiontree bzw. Decisiontree eine Vorhersage über den Wert bzw. die Klasse getroffen werden. Je ein Regressiontree wird für alle numerischen Features und je ein Decisiontree wird für alle zu klassifizierenden features als Modell erstellt, um wiederum ein weiteres Modell zur Auswahl vergleichen zu können. (https://www.shirin-glander.de/2018/11/ml_basics_gbm/)

4 Diskussion der Ergebnisse

Allgemein lässt sich festhalten, dass dieses Projekt vom grundsätzlichen Vorgehen her der Related Work XXX sehr ähnelt. Vor allem das Preprocessing und Data Cleaning als Grundlage für die folgenden Modellberechnungen sind sehr ähnlich. Ein großer Unterschied liegt in der Menge der Daten. So wurde in vergleichbaren Projekten eine wesentlich kleinere Datenbasis verwendet. Jedoch führt ein größerer, diverserer Trainingsdatensatz meist zu besseren Modellen. Ein overfitting oder underfitting kann verhindert werden, da dem Modell ausreichende und sehr heterogene Daten zum Training bereitstehen. Es kann somit eine bessere Qualität der Vorhersagen erzeugt werden.
Die verwendete Datenquelle hat jedoch auch einige Nachteile, denn sie ist veraltet und kann somit den Sprachlichen Wandel von mehr als der letzten 15 Jahre nicht berücksichtigen. So kann es sein, dass die Verwendung verschiedener Textcharakteristika eher einem Geburtenzeitraum als einem pauschalen Alter zugeordnet werden kann. Möglicherweise „altern“ die Gewohnheiten mit den Verfassern. So ist es möglich, dass Charakteristika, die dem Modell nach ca. 15-Jähriger zuzuordnen sind, nun eher in der Verwendung ca. 30-Jähriger. Diese Möglichkeit wird in der hier ausgeführten Arbeit nicht näher beleuchtet, da der Test ebenfalls mit Daten des gleichen Zeitraums stattfindet.
Ein weiterer Punkt ist, dass nur Daten von 13- bis 47-Jährigen im Datensatz enthalten sind und somit eine Vorhersage für jüngere oder ältere Personen kaum möglich ist.
Zudem wird das Ergebnis dieses Projekts nicht problemlos Merkmale aller Texttypen vorhersagen können, da die Modelle nur auf Blogeinträgen erstellt wurden. Blogeinträge haben in vielerlei Hinsichten grundsätzlich andere Charakteristika wie z.B. eine informellere Sprache als z.B. Sachbücher, sie haben eine höhere Informationsdichte als z.B. Kinderbücher und sind vor allem auch deutlich kürzer als diese, um nur einige Punkte zu nennen. Daraus folgt, dass das Ergebnis dieses Projekts nicht für jeden Texttyp, jedoch gut für Blogeinträge geeignet ist.
Abschließend ist auch noch zu bemerken, dass sowohl das Feature Engineering als auch die verschiedenen Methodiken zum Erstellen und Validieren der Modelle deutlich ausgeprägter stattgefunden haben, als in vergleichbaren Projekten. Das Feature Engineering wurde stärker beleuchtet, wodurch sowohl Klassifizierung, Regression aber auch Clustering Anwendung auf den Datensatz finden. Es wurden verschiedene Methoden bei jeder dieser Vorgehensweisen verwendet und die Ergebnisse gegeneinander abgewogen, statt die Umsetzung einer Methodik zu demonstrieren.

Problem: alter Datensatz, kein Update seit 2004

Problem: nicht alle Altersgruppen sind vertreten: nur 13-47

Problem: nur Blogeinträge, keine zuverlässige Klassifizierung von anderen Texttypen

Pro: viel mehr daten als vergleichbare, vergleichbare oft relativ schlechte ergebisse (Acuracy von ca 0.5 kann auch random erreicht werden) Ähnliches Vorgehen, aber Kombination und Vergleiche

4.1 Bewertung der Modelle für das Feature Age

Text 1

4.2 Bewertung der Modelle für das Feature Gender

Text 1

4.3 Bewertung der Modelle für das Feature Sternzeichen

Text 1

5 Schlussbetrachtung

Dieses Kapitel enthält die Zusammenfassung der Arbeit mit Fazit und Ausblick.

5.1 Fazit

...

5.2 Ausblick

...

A Tabellen

A.1 Feature Engineering

Feature	Berechnung	Intention
Text Length	Anzahl der Zeichen im Text	Indikator für Textumfang
Number URLs	Anzahl der Worte, die eines oder mehrere der folgenden Wörter enthalten: „urlLink“, „http“, „www“	Indikator dafür, ob auf andere Quellen verlinkt wird.
Number emails	Anzahl der Emails im Text	Sind Ansprechpartner genannt worden?
Uppercase ratio	Anteil der Großbuchstaben an allen Zeichen	Quantifizierung der Zeichenwahl
Lowercase ratio	Anteil der Kleinbuchstaben an allen Zeichen	Quantifizierung der Zeichenwahl
Number ratio	Anteil der Zahlen an allen Zeichen	Quantifizierung der Zeichenwahl
Symbol ratio	Anteil der Sonderzeichen an allen Zeichen	Quantifizierung der Zeichenwahl
Average letter per words	Durchschnitt der Buchstaben pro Wort	Quantifizierung der Wortlänge
Variance of letters per words	Varianz der Wortlänge	Quantifizierung der Heterogenität der Wortlänge
Unique words ratio	Anzahl der verschiedenen verwendeten Wörter im Text	Quantifizierung der Heterogenität der Wortlänge
Average letters per sentence	Durchschnitt der Buchstaben pro Satz	Quantifizierung der Satzlänge
Average letters per sentence	Durchschnitt der Buchstaben pro Satz	Quantifizierung der Satzlänge
Variance of letters per sentence	Varianz der Buchstaben pro Wort	Quantifizierung der Heterogenität der Satzlänge
Average words per sentence	Varianz der Buchstaben pro Wort	Quantifizierung der Worte pro Satz
Variance of words per sentence	Varianz der Worte pro Satz	Quantifizierung der Heterogenität der Worte pro Satz
Maximum uppercase ratio per sentence	Anteil der Großbuchstaben in dem Satz mit dem höchsten Anteil an Großbuchstaben	Zeigt, ob Sätze mit hohem Anteil an Großbuchstaben vorhanden sind
Length of the max. uppercase ratio sentence	Anzahl der Zeichen in dem obigen Feature	Indikator für die Zuverlässigkeit des obigen Features

B Beispiel-Anhang: Noch ein Testanhang

Literaturverzeichnis

[1] Nick Anstead und Ben O'Loughlin. „Social media analysis and public opinion: The 2010 UK general election". In: *Journal of Computer-Mediated Communication* 20.2 (2015), S. 204–220.

[2] Shlomo Argamon et al. „Automatically profiling the author of an anonymous text". In: *Communications of the ACM* 52.2 (2009), S. 119–123.

[3] M. Sudheep Elayidom et al. „Text Classification For Authorship Attribution Analysis". In: *Advanced Computing: An International Journal* 4 (Okt. 2013). DOI: 10.5121/acij.2013.4501.

[4] Mehwish Fatima et al. „Multilingual SMS-based author profiling: Data and methods". In: *Natural Language Engineering* 24.5 (2018), S. 695–724. DOI: 10.1017/S1351324918000244.

[5] Abdulrahman I Al-Ghadir und Aqil M Azmi. „A study of arabic social media users—posting behavior and author's gender prediction". In: *Cognitive Computation* 11.1 (2019), S. 71–86.

[6] Seifeddine Mechti et al. „A decision system for computational authors profiling: From machine learning to deep learning". In: *Concurrency and Computation: Practice and Experience* (2020), e5985.

[7] Pashutan Modaresi, Matthias Liebeck und Stefan Conrad. „Exploring the Effects of Cross-Genre Machine Learning for Author Profiling in PAN 2016." In: *CLEF (Working Notes)*. 2016, S. 970–977.

[8] Todd K Moon, Peg Howland und Jacob H Gunther. „Document author classification using generalized discriminant analysis". In: *Universidad del Estado de Utah. Estados Unidos* (2006).

[9] Tadashi Nomoto. „Classifying library catalogue by author profiling". In: *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. 2009, S. 644–645.

[10] Francisco Rangel und Paolo Rosso. „Use of language and author profiling: Identification of gender and age". In: *Natural Language Processing and Cognitive Science* 177 (2013).

[11] Francisco Rangel et al. „Overview of the 2nd author profiling task at pan 2014“. In: *CEUR Workshop Proceedings*. Bd. 1180. CEUR Workshop Proceedings. 2014, S. 898–927.

[12] K. Santosh et al. „Author Profiling: Predicting Age and Gender from Blogs Notebook for PAN at CLEF 2013“. In: *CLEF*. 2013.

[13] Fabrizio Sebastiani. „Machine learning in automated text categorization“. In: *ACM Computing Surveys* 34.1 (März 2002), S. 1–47. ISSN: 1557-7341. DOI: 10.1145/505282.505283. URL: http://dx.doi.org/10.1145/505282.505283.

[14] Efstathios Stamatatos. „A survey of modern authorship attribution methods“. In: *Journal of the American Society for information Science and Technology* 60.3 (2009), S. 538–556.

[15] Sharmila Devi V et al. „KCE_DAlab@MAPonSMS-FIRE2018: Effective word and character-based features for Multilingual Author Profiling“. In: *Working Notes of FIRE 2018 - Forum for Information Retrieval Evaluation, Gandhinagar, India, December 6-9, 2018*. Hrsg. von Parth Mehta et al. Bd. 2266. CEUR Workshop Proceedings. CEUR-WS.org, 2018, S. 213–222. URL: http://ceur-ws.org/Vol-2266/T4-2.pdf.