NLP Klassifizierungsoptimierung

Jannik Hock
jannik.hock@hotmail.de
6065007

Johannes Türk
jo.uni@icloud.com
6205197

Data Science 1
Goethe Universität Frankfurt am Main

1 Einleitung

Das Ziel von *Machine Learning (ML)* besteht darin, hochwertige Vorhersagen und Entscheidungen auf Basis von gelerntem Wissen treffen zu können [1]. Um dies zu gewährleisten, ist ein korrekter und konsistenter Datensatz unabdingbar [9]. Diese Arbeit orientiert sich an der *Machine Learning Pipeline* und beschäftigt sich mit verschiedenen *Natural language processing (NLP)* Konzepten, untersucht deren Wirkungsweise und dessen Einfluss, den sie auf die Qualität eines Datensatzes nehmen.

2 Datensatz

Der Datensatz ist eine Gruppe inhaltlich zusammenhängender, aber eigenständiger Datenfelder und bildet die Grundlage im *ML* [2].

2.1 Amazon Review Dataset (ARD)

Das *ARD* befasst sich mit der Multilevel-Textklassifizierung von Produktbewertungen der Website Amazon [3] und umfasst 40.000 Instanzen. Für diese Arbeit wird der Fokus auf die Klassifizierung des ersten Levels gesetzt, um sich verstärkt auf das *NLP* zu konzentrieren.
Das Ziel ist es, anhand von Produktbewertungen zu entscheiden, welcher Kategorie beziehungsweise Klasse ein bewerteter Artikel angehört. Die Artikel können den Klassen *toys games*, *grocery gourmet food, beauty*, *baby products*, *pet supplies*, *health personal care* zugeordnet werden.

2.2 Dataset Preparation

Data Preparation ist der Prozess der Rohdatenaufbereitung [7]. Findet die Verarbeitung und Analyse auf Basis falscher oder inkonsistenter Daten statt, kann dies zu mangelhaften Ergebnissen und falschen Schlussfolgerungen führen [8].

2.2.1 Data Merging

Das *ARD* ist für eine optimale Klassifizierung nicht ausreichend balanciert und dementsprechend um zwei weitere Datensätze [4, 5] erweitert. Außerdem wurde Instanzen der Klasse *toys games* und *health personal care* entfernt.

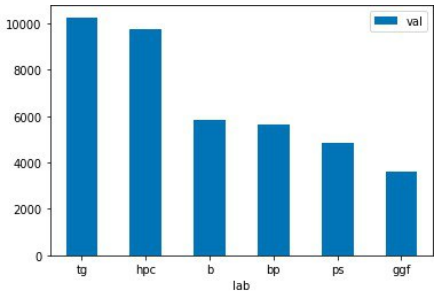

Abb. 1: Unbalanced Dataset

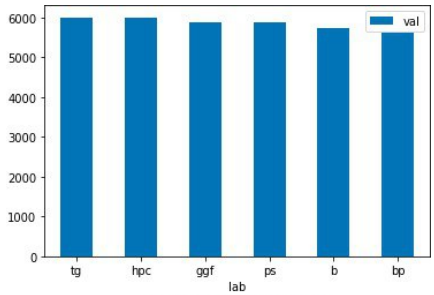

Abb. 2: Balanced ARD

2.2.2 Data Cleaning

Das *Data Cleaning* befasst sich mit der Erkennung und Beseitigung oder Aufarbeitung fehlerhafter und inkonsistenter Daten, um dessen Qualität zu verbessern. Das hat einen positiven Einfluss auf die Aussagekraft der Ergebnisse und die Größe des Datensatzes. [9]

2.2.2.1 Übersetzung

Ein Datensatz der unterschiedlichen Sprachen enthält, kann die Ergebnisse der Klassifizierung verfälschen. Auf den Datensatz dieser Arbeit hat das Übersetzen keinen Einfluss genommen, da alle Produktbewertungen in Englischer Sprache vorliegen.

2.2.2.2 Satzzeichen & Abkürzungen

Damit Abkürzungen wie *couldn't* und deren ausgeschriebenen Pendants wie *could not* bei der Klassifikation identisch behandelt werden, werden Abkürzungen in deren ausgeschriebene Form transformiert. Darüber hinaus werden Satzzeichen aus den Texten entfernt, da sie in diesem Fall keinen Mehrwert für die Qualität der Klassifikation bieten.

2.2.2.3 Lower case

Die Transformation der Produktbewertungen in *lower case* hat eine weniger umfangreiche Zeichencodierung zur Folge [10] und ist notwendig für die spätere Entfernung der *Stopwords*, da die verwendete Funktion lediglich klein geschriebene *Stopwords* berücksichtigt.

2.2.2.4 Stopwords

Das Ziel der Entfernung von *Stopwords* wie *the* ist es, Wörter mit geringem Informationsgehalt nicht in Betracht zu ziehen, um sich auf die wichtigen Wörter des Textes konzentrieren zu können [10].

2.2.2.5 Lemmatization

Lemmatization entfernt grammatikalische Beugungen und führt ein Wort auf dessen Stammform zurück. Dabei werden "sinnvolle" Worttransformationen wie *worked* in dessen Wurzel *work* vorgenommen (Siehe Abb. 3 - pink). [18]

2.2.2.6 Stemming

Stemming entfernt ebenfalls grammatikalische Beugungen und ordnet ein Wort seiner Stammform zu, ohne "kluge" Transformationen vorzunehmen (Siehe Abb. 3 - gelb) [10]. Da Stemming lediglich Wortendungen "abhackt" wird es *Lemmatization* nachgesellt, um "kluge" Transformationen nicht zu verhindern. *Lemmatization* und *Stemming* führen dazu, dass Klassifizierungs-Modelle auf Basis "logisch" gleicher Wörter, unabhängig von deren Beugung, klassifizieren.

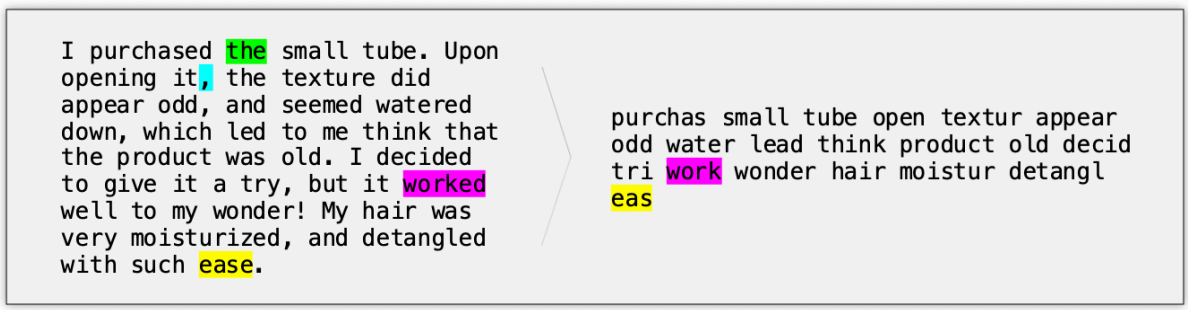

Abb. 3: Data Cleaning Resultat einer Beispielinstanz

Keine der NLP Methoden stellt eine Lösung für Sarkasmus innerhalb des *ARD* dar, was sich negativ auf die Klassifizierungsergebnisse auswirken kann.

2.2.3 Data Quality

Da insbesondere *Stopwords* vereinzelt dazu führen, dass leere Strings entstehen, wurden im Sinne der *Completeness* Anforderung, die NaN-Values und leeren String am Ende des *Data Cleaning* Prozesses entfernt. Dadurch konnte die *Completeness* von 100% beibehalten werden. Darüber hinaus wurden Dupliakte entfernt, um für eine 100% *Uniquness* zu sorgen. Da die Datensätze [4] und [5] keine *Labels* hatten, wurden sie um die entsprechenden Klassen erweitert (*Consistency*). Die Voraussetzung für die *Accuracy* innerhalb des Datensatzes ist es, dass identische Produktbewertungen der gleichen Klasse angehören.

Quality Dimension		Vor Data Cleaning	Nach Data Cleaning
Completeness		100%	100%
Uniqueness		98,76%	100%
Timeliness	Datensatz [3]	62 Tage	62 Tage
	Datensatz [4]	1129 Tage	1129 Tage
	Datensatz [5]	1499 Tage	1499 Tage
Validity		10,6% valid to invalid (nach Merging)	0% valid to invalid (nach Merging)
Accuracy		99,96%	100%
Consistency		89,41%	100%

Tabelle 1: Data Quality Ergebnisse

2.2.4 Data Splitting

Das *ARD* wird im Verhältnis 80/20 in ein Training- und Test-Datensatz unterteilt. Der Trainingsdatensatz wird genutzt, um das spätere *Machine Learning Modell (MLM)* zu trainieren, um Vorhersagen treffen zu können. Das Testdatenset dazu, das trainierte Modell mit Hinsicht auf die Qualität zu validieren. [11]

3 Models

MLMs sind mathematische Modelle, die auf Basis der Trainingsdaten in der Lage sind, Muster wieder zu erkennen und so Vorhersagen oder Entscheidungen über noch nicht gesehen Daten, wie das Testdatenset, zu treffen [12].

3.1 Random Forest Classifier (RFC)

Der *RFC* kombiniert mehrere Entscheidungsbäume und wählt mittels Votingverfahren die beste Lösung der einzelnen Entscheidungsbäume aus [13].
RFCs sind schnell, relativ gut interpretierbar und für Textdatensätze gut geeignet [14]. Diese Eigenschaften helfen, das Model häufig zu trainieren und den Einfluss des *Data Cleanings* überprüfen zu können.

3.2 Gradient Boosting Classifier (GBC)

Der *GBC* nutzt viele schwache Lernmodelle und kombiniert diese zu einem Starken [15], indem er iterativ Bäume kreiert, die die Fehler der vorangegangenen Bäume korrigieren, um die Aussagekraft des Modells fortlaufend zu steigern [16]. *GBCs* sind komplexer zu trainieren als *RFCs*, weisen dafür aber eine bessere *Performance* auf [17]. Daher eignen sich die beiden

Modelle sehr gut, den Einfluss des Data Cleaning auf unterschiedlich *Performance*-starke Modelle zu veranschaulichen.

4 Ergebnisse

Im Folgenden werden die Einflüsse der *NLP* Methoden auf die *MLM* gegenübergestellt und die *Data Quality Dimensions* visualisiert.

NLP Methode	RFC (10)		RFC (100)		GBC (10)		GBC (100)	
	A	T	A	T	A	T	A	T
Keine NLP Methode	0,61	14,66	0,75	101,4	0,67	96,93	0,77	662,3
Übersetzung	0,61	14,66	0,75	101,4	0,67	96,93	0,77	662,3
Satzzeichen & Abkürzungen	0,62	17,47	0,75	114,4	0,69	99,14	0,77	690,7
Lower case	0,65	13,12	0,77	95,61	0,68	92,28	0,79	657,5
Stopwords	0,7	12,58	0,78	93,8	0,67	55,4	0,8	375,84
Lemmatization	0,62	14,17	0,76	94,6	0,68	86,91	0,78	685,4
Stemming	0,63	12,92	0,77	83,1	0,69	90,65	0,79	641,7
Alle NLP Methoden	0,73	12,2	0,79	89,77	0,7	52,24	0,81	375,78

GBC(X) & RFC(X), wobei X ≙ estimators. A ≙ Accuracy. T ≙ Time in sekunden. Alle Ergebnisse wurden auf demselben Computer berechnet (1,6GHz Dual Core Intel Core I5, 4GB 1600 MHz DDR3)

Tabelle 2: Klassifizierungsergebnisse

Besonders auffällig ist, dass fast jede *NLP* Methode eine positive Auswirkung auf die *Performance* bzw. Laufzeit des jeweiligen MLM hat. Hervorzuheben sind die Einflüsse der *NLP* Methoden *Stopwords* und *Stemming*.
Stemming ist im Vergleich zu den anderen *NLP* Methoden stets eine der Besten und sticht bei der Zeit des *RFC(100)* besonders heraus. Darüber hinaus hat *Stemming* gegenüber *Lemmatization* ausnahmslos einen positiveren Einfluss auf die Ergebnisse, obwohl sie sich in der Funktionsweise nur wenig unterscheiden.
Stopwords hat bei dem *ARD* den positivsten Einfluss auf die *Accuracy* und Zeitkomponente. Bei dem *RFC* belaufen sich die positiven Auswirkungen der NLP Methoden in erster Linie auf die *Accuracy*. Wohingegen die *NLP* Methoden bei dem *GBC* vor allem Laufzeitverbesserungen mit sich bringen.
Die *NLP* Methoden *Übersetzung*, *Satzzeichen & Abkürzungen* sowie *Lower case* hatten scheinbar wenig bis garkeinen positiven Einfluss auf das Gesamtergebnis der Klassifizierung.

5 Fazit

NLP Methoden haben im Großen und Ganzen einen enorm positiven Einfluss auf die hier genutzten *MLM*s und deren Ergebnisse genommen.
Je schlechter das *MLM* für einen Datensatz geeignet ist, desto mehr Einfluss scheint das *Data Cleaning* auf die *Accuracy* zu haben. Bei sehr guten *MLM*s nimmt das *Data Cleaning* scheinbar vorrangig einen positiven Einfluss auf die Laufzeit.

GitHub:

https://github.com/TUCK-goethe/DataScience1

Sources:

[1] Reck, Franziska. (7. Januar 2019). Was ist Machine Learning?. Abgerufen 10. Juni 2020, von https://relevanzmacher.de/was-ist-machine-learning/

[2] MCCREA, NICK. (5. Mai 2020). An Introduction to Machine Learning Theory and Its Applications: A Visual Tutorial with Examples. Abgerufen 23. Juni 2020, von https://www.toptal.com/machine-learning/machine-learning-theory-an-introductory-primer

[2] Ivanov, T. (5. Mai 2020). Project Description. Abgerufen 22. Juni 2020, von http://www.bigdata.uni-frankfurt.de/wp-content/uploads/2020/03/Project_AI_Tools_SS2020.pdf

[3] Kashnitsky, Y. (März 2020). Hierarchical text classification. Abgerufen 3. Juni 2020, von https://www.kaggle.com/kashnitsky/hierarchical-text-classification

[4] Stanford Network Analysis Project. (1. Mai 2017). Amazon Fine Food Reviews. Abgerufen 3. Juni 2020, von https://www.kaggle.com/snap/amazon-fine-food-reviews

[5] Stanford. (26-Apr-2016). reviews_Pet_Supplies_5.json.gz. Abgerufen 3. Juni 2020, von http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Pet_Supplies_5.json.gz

[6] Escalona, Todd. (26. Januar 2018). Detect sentiment from customer reviews using Amazon Comprehend. Abgerufen 21. Juni 2020, von https://aws.amazon.com/blogs/machine-learning/detect-sentiment-from-customer-reviews-using-amazon-comprehend/

[7] Pearlman, Shana. (27. Mai 2020). What is Data Preparation?. Abgerufen 19. Juni 2020, von https://www.talend.com/resources/what-is-data-preparation/

[8] Elgabry, Omar. (28. Februar 2019). The Ultimate Guide to Data Cleaning. Abgerufen 23. Juni 2020, von https://towardsdatascience.com/the-ultimate-guide-to-data-cleaning-3969843991d4

[9] Rahm, Erhard & Do, Hong Hai. Data Cleaning: Problems and Current Approaches. Abgerufen 20. Juni 2020, von http://dc-pubs.dbs.uni-leipzig.de/files/Rahm2000DataCleaningProblemsand.pdf

[10] Ganesan, Kavita. All you need to know about text preprocessing for NLP and Machine Learning. Abgerufen 21. Juni 2020, von https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html

[11] Dataset splitting. Abgerufen 16. Juni 2020, von https://www.cl.cam.ac.uk/teaching/1617/MLRD/handbook/dataset-splits.pdf

[12] Microsoft. (1. April 2019). What is a machine learning model?. Abgerufen 17. Juni 2020, von https://docs.microsoft.com/en-us/windows/ai/windows-ml/what-is-a-machine-learning-model

[13] Breiman, Leo & Cutler, Adele. Random Forests. Abgerufen 21. Juni 2020, von https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

[14] Zakariah, Mohammed. (3. September 2014). Classification of large datasets using Random Forest Algorithm in various applications: Survey. Abgerufen 24. Juni 2020, von https://fac.ksu.edu.sa/sites/default/files/classification_of_large_datasets_using_random.pdf

[15] Nelson, Dan. Gradient Boosting Classifiers in Python with Scikit-Learn. Abgerufen 16. Juni 2020, von https://stackabuse.com/gradient-boosting-classifiers-in-python-with-scikit-learn/

[16] Singh, Harsheep. (3. November 2018). Understanding Gradient Boosting Machines. Abgerufen 18. Juni 2020, von https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab

[17] Glen, Stephanie. (28. Juli 2019). Decision Tree vs Random Forest vs Gradient Boosting Machines: Explained Simply. Abgerufen 19. Juni 2020, von https://www.datasciencecentral.com/profiles/blogs/decision-tree-vs-random-forest-vs-boosted-trees-explained#:~:text=Random%20Forest%20vs%20Gradient%20Boosting,one%20tree%20at%20a%20time.

[18] Seo, Jae Duk. (28. Mai 2018). [Basic Data Cleaning/Engineering Session] Twitter Sentiment Data. Abgerufen 18. Juni 2020, von https://medium.com/@SeoJaeDuk/basic-data-cleaning-engineering-session-twitter-sentiment-data-b9376a91109b