Noname manuscript No.
(will be inserted by the editor)

Feature Engineering

Aaron Braatz

28. Februar 2020

Zusammenfassung Feature Engineering ist Teil des Data Mining Prozesses und dient dazu die Rohdaten zu Featuren zu verarbeiten. Hierbei ist das Ziel die Daten so aufzubereiten, dass sie eine möglichst hohe Informationsdichte haben, um einerseits das Training der Data Mining Modelle zu optimieren und andererseits die Qualität der Resultate zu verbessern. In dieser Arbeit wird zu erst das Feature Engineering im Data Mining Prozess eingeordnet. Im Anschluss werden die einzelnen Schritte des Feature Engineerings beschrieben und daraufhin Methoden in den Bereichen: Zahlen, Texten und kategorische Daten erläutert. Abgeschlossen wird die Arbeit mit einem Fazit und Ausblick.

Schlüsselwörter Feature Engineering · Feature Enrichment · Feature Selection · Data Mining · Machine Learning

1 Einleitung

Machine Learning und Data Mining (DM) sind schon seit längerer Zeit von hohem Interesse und wird bereits in diversen Domänen eingesetzt. Feature Engineering (FE) ist dabei der Part, welcher aus den Rohdaten Feature generiert die eine möglichst hohe Informationsdichte haben und in ein passendes Format für DM-Modelle überführt. Dadurch kann das Training der Modelle verbessert und die Qualität der Ergebnisse verbessert werden. Allgemein ist bekannt, dass ein großer Teil der Zeit und des Aufwands im DM-Prozess in FE und Datenvorverarbeitung investiert werden muss. Dennoch gibt es nur wenig Literatur die sich mit dem reinen Thema FE beschäftigen [Zheng and Casari, 2018].

A. Braatz
Hamburg, Germany
Tel.: +123-45-678910
Fax: +123-45-678910
E-Mail: aaron.braatz@haw-hamburg.de

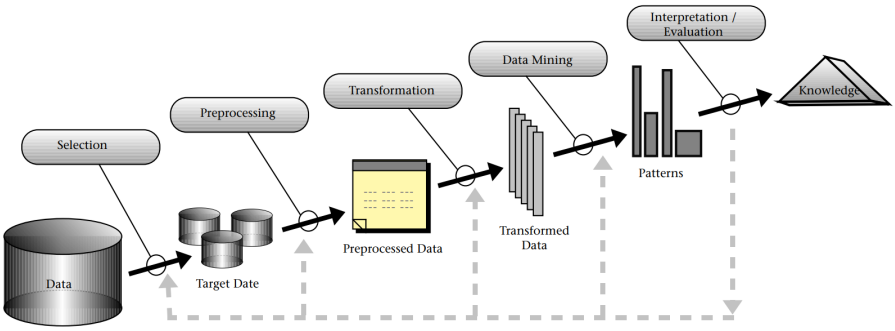

Abb. 1 Ablauf des KDD-Prozesses [Fayyad et al., 1996, Abb. 1]

In dieser Arbeit wird zunächst FE in den DM Prozess eingeordnet. Daraufhin werden die einzelnen Schritte Feature Engineering vorgestellt. Im Anschluss werden einige grundlegende FE-Methoden in den Bereichen Zahlen, Texte und Kategorien erläutert. Zum Abschluss wird noch ein Fazit gezogen und Ausblick gegeben.

2 Einordnung in den Data Mining Prozess

Fayyad et al. beschreibt 1996 den allgemein bezeichneten DM-Prozess als Knowledge Discovery in Databases (KDD). Dieser iterative Prozess besteht aus fünf Schritten. In dem ersten Schritt wird aus der allgemein vorhandenen Menge an Daten, unter Berücksichtigung des Anwendungsziels, die relevanten Daten selektiert oder wenn nötig erhoben. In dem zweiten Schritt werden die Daten vorverarbeitet. Hierbei werden fehlenden Daten gegebenenfalls ergänzt und Ausreißer behandelt. In dem dritten Schritt, der Transformation, werden die Daten in ein Format überführt, welches von dem DM-Verfahren genutzt werden kann. In dem vierten Schritt wird das eigentliche DM-Verfahren angewendet. Im letzten Schritt werden die Ergebnisse interpretiert und evaluiert. Ist das Anwendungsziel noch nicht erreicht kann in jedem Schritt des KDD-Prozesses neu angesetzt und Optimierungen vorgenommen werden.

FE ordnet sich in dem Prozess bei den Schritten Vorverarbeitung und Transformation ein. Das Ziel ist es Rohdaten in Features mit einer hohen Informationsdichte zu verarbeiten. Ein großer Teil der FE-Verfahren lassen sich hauptsächlich in dem Bereich der Transformation einordnen. Die Vorverarbeitung ist dabei häufig eher ein Nebeneffekt.

3 Ablauf

3.1 Visualisierung

Zu Beginn des FE werden die Daten visualisiert. Um die passenden FE-Verfahren zu wählen muss zu erst verstanden werden in welcher Form die Daten vorliegen und auf welche Besonderheiten eingegangen werden muss. Bei der Visualisierung wird das Fehlen von Daten offensichtlich und auch Ausreißer sind gut zu identifizieren. Durch eine grafische Darstellung lässt sich auch gut die Verteilung der Daten veranschaulichen und Entscheidungen zu Transformationen getroffen werden. Auch kann es interessant sein, in welchem Format die Daten vorliegen. Zum Beispiel kann der Datensatz ein Datum als *String* enthalten. Das kann dazu führen, dass die Werte diskret interpretiert werden. Eine Umwandlung in einen *Timestamp* führt zu stetigen Werten.

3.2 Feature "Processing"

Nachdem ein Überblick geschaffen ist in welcher Form und mit welchen Besonderheiten die Daten vorliegen, können nun unterschiedliche Verfahren angewendet um die Daten zu Features mit einer hohen Informationsdichte aufzuwerten. Hierbei gibt es unterschiedliche Arten von Verfahren. Bei einer ungleichen Verteilung von Daten können diese transformiert werden um eine gleichmäßigere Verteilung zu erhalten. Bei unterschiedlichen Wertebereichen kann es sinnvoll sein diese zu skalieren. Menschen-sprachliche kategorische Daten müssen in ein Maschinen-verständliches Format überführt werden. Beispiele für diese Verfahren werden im Abschnitt Anwendungsmöglichkeiten vorgestellt.

3.3 Feature Enrichment

Bei dem Feature Enrichment geht es nicht darum die bisherigen Features mit neuen Daten zu ergänzen, sondern aus den vorhandenen Features weitere Features zu generieren. Zum einen kann aus einem reinen Datumsattribut auch ein Feature Jahreszeiten generiert werden, was Vorteile bei Anwendungen im Agrarbereich haben kann. Aber auch die Entfernung von einem Koordinatenpunkt einer Messstation zur nächsten Küste kann eine zusätzliches Feature sein in der Meteorologie oder in der Klimawissenschaft. Hierbei werden vorhandene Informationen von einem Menschen ,,vorinterpretiert". Für ein DM-Verfahren ist es schwierig dieses ,,Verständnis" erlernen, hingegen ist es für einen Menschen naheliegend.
Eine weitere Möglichkeit des Feature Enrichment ist die logische Kombination von Featuren (Interaction Feature). Zum Beispiel bei einem Anwendungsfall mit Autos kann die Farbe eines Autos mit der Marke zu einem Feature kombiniert werden. So kann eine besondere Kombination wie ein *roter Ferrari*

einen besonderen Mehrwert für das DM-Verfahren haben. Interaction Feature können besonders vorteilhaft bei der Nutzung von linearen Modellen sein. [Kotsiantis et al., 2006]

3.4 Feature Selection

Bei der Feature Selection geht es darum inhaltsvollere Feature zu identifizieren und andere zu entfernen um wiederum die Informationsdichte zu erhöhen. Hierfür gibt es drei Herangehensweisen:

3.4.1 Filtering

Bei der Filterung wird die Korrelation zwischen den Featuren und dem Ergebnis berechnet. Feature mit einer geringen Korrelation können daraufhin aussortiert werden. Der Vorteil dieser Methode ist der geringe Aufwand mit dem ein Ergebnis erzielt werden kann. Allerdings werden hierbei auch Feature entfernt, die für das DM-Modelle einen spezifischen Mehrwert bieten können.

3.4.2 Wrapper Method

Bei der Wrapper Methode wird das DM-Modell bei der Selektion mit einbezogen. Dabei werden Teilmengen der Feature in das Training eingebunden und das Ergebnis wird als Referenz genutzt. Durch die Ergebnisse unterschiedlicher Kombinationen lässt sich die optimale Menge an Features identifizieren. Im Gegensatz zur Filterung ist diese Variante mit einem hohen Aufwand verbunden. Dafür erhält man die optimale Kombination an Feature für ein bestmögliches Ergebnis.

3.4.3 Embedded Method

Ein Kompromiss der vorhergehenden Varianten ist die Embedded Methode. Hier wird die Feature Selection zum Teil des Trainingsprozesses. Dabei werden Modelle bevorzugt, die weniger Feature benötigen. Bei Entscheidungsbäumen ist diese Eigenschaften implizit. Bei linearen Modellen kann der L1-Regularizer genutzt werden, um diese Variante der Feature Selection umzusetzen. [Guyon and Elisseeff, 2003]

4 Anwendungsmöglichkeiten

4.1 Zahlen

4.1.1 Binarization

Die Binarization kann bei Featuren genutzt werden, die eine Anzahl beschreiben. Hierbei wird, anstatt der genauen Anzahl, eine ,,1" für ,,vorhanden sein"

genutzt. Auch wenn es scheint, dass dabei Informationen verloren gehen, kann das Feature dadurch aussagekräftiger werden. Veranschaulicht kann das an einem Beispiel für Song-Empfehlungen. Die Information, wie häufig ein Titel gehört wurde hat nicht zwangsläufig zur Folge, dass dieser empfehlenswert wird. Der Datensatz *Million Song Dataset* besteht aus Tripeln mit Song, Nutzer und wie oft der Song gehört wurde. 99% der Tripel haben 24 oder weniger Anhörungen. Allerdings hat das letzte Prozent teilweise Anhörungen in den Tausendern. Das kommt daher, das einige Songs zeitweise in Schleife gehört werden oder in mehreren Playlists vorkommen. Aufgrund dieses Ungleichgewicht von Anhörungen zu Vorkommen in dem Datensatz, ist es schwer eine umfassende Empfehlung auf Basis der Anzahl der Anhörungen zu geben. Stattdessen reicht die Tatsache, dass ein Titel gehört wurde.

4.1.2 Binning

Binning wird genutzt um Feature mit stetigen Zahlen in Intervalle (Bins) zu diskretisieren Zum einen gibt es die Einteilung in Intervalle mit fester Breite. In einem kleinen Zahlenbereich wie einem Alter kann der Wertebereich noch sinnvoll in gleichgroße Intervalle geteilt werden. Erstreckt sich der Wertebereich allerdings über mehrere Zehnerpotenzen, wird der Bereich in exponentiell wachsende Intervalle eingeteilt. Dadurch wird der Detailgrad für kleinere Zahlen gewahrt, welcher bei größeren Zahlen an Relevanz verliert.
Diese Verfahren sind anwendbar solange keine Lücken in den Daten sind. Andernfalls bieten *Quantiles* eine Alternative. Hierbei werden die Daten anhand ihrer Häufigkeit in gleich große Intervalle geteilt. Als Metrik für die Teilung wir zum Beispiel der Median genutzt, da er eine Menge in gleich große Teile trennt. [Weir, 1993]

4.1.3 Transformation

Transformationen werden genutzt um eine ,,Heavy-tailed"-Verteilung der Daten zu korrigieren. Eine solche Verteilung hat viele Datenpunkte mit einer geringen Anzahl, aber auch einige wenige mit einer sehr hohen Anzahlen. Dadurch liegt die Wahrscheinlichkeitsmasse am Ende der Verteilung im Gegensatz zu einer Gaußverteilung. Mit Hilfe einer Log-Transformation kann dem entgegengewirkt werden. Ähnlich wie beim Binning mit exponentiell wachsenden Intervallen, wird so mehr Wert auf die Datenpunkte mit geringerer Anzahl gelegt. Mit Hilfe der *Box-Cox/Power*-Transformation kann die Verteilung noch besser einer Gauß-Verteilung angenähert werden. Das wird durch eine verallgemeinerte, parametrisierbare Form der Log-Transformation erreicht, bei der die Varianz stabilisiert wird. Dabei können aber nur positive Werte transformiert werden, was durch einen Offset in den Daten erreicht werden kann. [Hossain, 2011]

4.1.4 Skalieren

Bei der Skalierung (engl. Feature Scaling) wird sowohl das Skalieren, als auch das Normalisieren von Daten behandelt.
Bei der Normalisierung (engl. Min-Max-Scaling) (1) wird ein potenziell grenzenloser Wertebereich auf einen Bereich zwischen [0...1] abgebildet. Das wird genutzt, um unterschiedliche Wertebereiche von Featuren vergleichbar zumachen. Zum Beispiel hätte der Preis in Euro eines Autos indirekt ein höheres Gewicht bei einem DM-Verfahren im Gegensatz zu einem Felgendurchmesser in cm, da die Zahl des Preises grundsätzlich größer als die des Durchmessers ist. Bei der Standardisierung (engl. Variance-Scaling) (2) wird die Verteilung der Werte angepasst, sodass sie einen Mittelwert von 0 und eine Varianz von 1 hat. Die l^2-*Normalisierung* (3) normalisiert die Werte aller Feature mit Hilfe der Euklidischen Norm. Dadurch haben alle Feature eine Norm von 1 im Datenraum. [Krishna et al., 2016]

$$\hat{x} = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

$$\hat{x} = \frac{x - mean(x)}{\sqrt{var(x)}} \tag{2}$$

$$\hat{x} = \frac{x}{\sqrt{x_1^2 + x_2^2 + ... + x_n^2}} \tag{3}$$

4.2 Texte

4.2.1 Bag-of-Words

Um Texte in ein Maschinen-verständliches Format zu übertragen wird *Bag-of-Words* (BoW) genutzt. Hierbei wird Vektor vergleichbar mit einem Wörterbuch generiert. Die Anzahl der Dimensionen des Vektors entspricht der Anzahl unterschiedlicher Wörter in einem Datensatz. Dabei steht eine Position in dem Vektor für ein Wort. Um einen Text zu beschreiben wird für jedes Wort in dem Text an der dazugehörigen stelle im Vektor die Häufigkeit des Worts notiert.

4.2.2 Bag-of-n-Grams

Bag-of-n-Grams ist die Erweiterung zu BoW um Wortsequenzen zu erfassen. Dabei beschreibt n wie viele Worte zu einer Sequenz gehören. BoW ist dementsprechend ein Bag-of-1-Gram. Der Satz ,,Er geht zur Tür." wird zu: ,,Er geht", ,,geht zur", ,,zur Tür". Die Wortsequenzen werden ähnlich wie bei BoW in einem Vektor erfasst. Um sinnvolle Sequenzen zu bilden ist darauf zu achten, dass Sequenzen zum Beispiel nicht über Satzgrenzen hinaus gebildet werden. Je länger die betrachteten Sequenzen definiert werden, desto schneller

wächst die Anzahl der Dimensionen des Vektors. Daher wird meistens maximal ein Bag-of-3-Grams genutzt, damit der Rechenaufwand verhältnismäßig zum Nutzen bleibt.

4.2.3 Filterung

Filterung wird genutzt, um die Menge von Wörtern, beispielsweise in einem BoW, auf die Wörter mit einer hohen Aussagekraft zu reduzieren. Eine Möglichkeit ist die Nutzung von *Stopwords*. Das sind sprachenspezifische Listen mit Füllwörtern, welche üblich sind in natürlicher Sprache aber einen geringen Informationsgehalt zum Beispiel für Klassifikationen bieten. Diese Listen ändern sich allerdings über die Zeit und müssen manuell gepflegt werden, da sich Sprache entwickelt. Das führt insbesondere bei einem langen Anwendungszeitraum zu einem großen Aufwand.
Eine weitere Möglichkeit ist die frequenzbasierte Filterung. Hierbei können Füllwörter durch die hohe Häufigkeit in Sprache und somit auch in einem BoW identifiziert werden. Allerdings ist dabei zu beachten das in einigen Themenspezifischen Datensätzen auch relevante Wörter häufig vorkommen können. In einem Datensatz mit Bewertungen von Restaurants kann das Wort ,,lecker" häufig vorkommen und hat trotzdem eine wichtige Bedeutung. Daher sollten die häufigsten Wörter noch einmal überprüft werden, bevor sie aussortiert werden.
Mit der gleichen Methode lassen sich auch seltene Wörter ermitteln. Auch wenn seltene Wörter potenziell einen Informationsgehalt haben können, ist ihr Einfluss auf das Lernverhalten vernachlässigbar. Das Wort ,,traumatisch" in Bezug auf einen Restaurantbesuch ist vermutlich selten, aber hat für einen Menschen eine klare Aussage. Das einzelne Wort wird aber im Trainingsprozess für ein Modell keinen großen Einfluss haben. Daher lassen sich diese seltenen Wörter sicher entfernen oder in nur einem Platz im Wortvektor zusammenfassen.

4.2.4 Stemming

Stemming wird genutzt, um grammatische Variationen eines Wortes auf einen Wortstamm zu reduzieren. Die Worte ,,geht" und ,,gehen" sind technisch unterschiedliche Worte, lassen sich aber sinnhaft zusammenfassen. Stemming wird genutzt, um möglichst passend die Ränder von Wörtern zu beschneiden, sodass sie auch technisch identisch sind. In dem genannten Beispiel wäre es ,,geh". Dieses Verfahren ist, ähnlich wie Stopwords aus der Filterung, sprachenspezifisch. Auch lassen sich nicht immer die Wörter sinnvoll beschneiden. Das Wort ,,ging" gehört sinnhaft auch zu dem genannten Beispiel, hat aber keine passende Buchstabensequenz, um mit Stemming zu den anderen Formen vergleichbar zu werden.

4.2.5 Collocation Extraction

Bei der *Collocation Extraction* wird versucht, Phrasen zu identifizieren die einen Mehrwert haben, um die Menge der Wortsequenzen zu reduzieren. Ein Beispiel für eine relevante Phrase ist „starker Kaffee". Eine physische Beschreibung und ein Getränke haben erstmal keinen Zusammenhang. Gemeinsam haben die Wörter, aber eine größere Bedeutung als die einzelnen Worte. Identifiziert werden solche Phrasen in dem Überprüft wird, ob die Kombination der beiden Worte häufiger in dem Datensatz vorkommt, als die einzelnen mit anderen Wörtern. [Bouma, 2009]

4.3 Kategorien

4.3.1 One-Hot-Encoding

One-Hot-Encoding wird genutzt um kategorische Daten in ein Maschinenverständliches Format zu überführen. Ähnlich wie bei BoW wird dafür ein Vektor genutzt. Die Anzahl der Dimensionen des Vektors entspricht dabei der Anzahl unterschiedlicher Werte, die in der Kategorie möglich sind. Zum Beispiel kann die Kategorie Jahreszeiten vier unterschiedliche Werte annehmen, daher hat der Vektor vier Dimensionen. Hat ein Datenpunkt in der Kategorie Frühling, so wird die entsprechende Position im Vektor auf 1 gesetzt (4).

$$\text{Jahreszeiten} = \begin{bmatrix} \text{Frühling} \\ \text{Sommer} \\ \text{Herbst} \\ \text{Winter} \end{bmatrix} \quad \text{Frühling} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4}$$

4.3.2 Dummy-Coding

Dummy-Coding funktioniert grundsätzlich ähnlich wie One-Hot-Encoding, allerdings wird auf eine Dimension bei dem Vektor verzichtet. Das wird erreicht durch das Einführen eines Referenzvektors. Zuvor wurde für jeden möglichen Wert einer Kategorie eine 1 an der zugehörigen Position gesetzt. In dem Konstrukt hat der Nullvektor keine Bedeutung, da er nicht vorkommen kann. Beim Dummy-Coding wird der Nullvektor zur Referenzklasse und kann semantisch für den Wert „normal" stehen.

4.3.3 Effect-Coding

Effect-Coding ist wiederum ähnlich zu Dummy-Coding. Die Anzahl der Dimensionen ergibt sich genauso, aber die Referenzklasse ist statt des Nullvektors ein Vektor gefüllt mit -1. Das hat zur Folge, dass die mit linearer Regression erlernte Konstante der Mittelwert, der Zielwerte ist. Veranschaulicht kann das an einem Preisvorhersage-Beispiel werden, wie in den Tabellen 4.3.3, 4.3.3 und

Tabelle 1 Preise pro Mahlzeit

Essen	Preis/€
Suppe	4
Suppe	6
Auflauf	9
Auflauf	11
Steak	14
Steak	16

Tabelle 2 Datenstrukturbeispiel für One-Hot-Encoding, Dummy-Coding, Effect-Coding

Essen	Preis/€	OHE			DC		EC	
		e0	e1	e2	e0	e1	e0	e1
Suppe	4	1	0	0	1	0	1	0
Suppe	6	1	0	0	1	0	1	0
Auflauf	9	0	1	0	0	1	0	1
Auflauf	11	0	1	0	0	1	0	1
Steak	14	0	0	1	0	0	-1	-1
Steak	16	0	0	1	0	0	-1	-1

Tabelle 3 Mit Linear Regression erlernte Koeffizienten

Methode	x0	x1	x2	b
One-Hot-Encoding	-4	-1	5	10
Dummy-Coding	0	-9	-6	15
Effect-Coding	0	-4	-1	10

4.3.3 dargestellt. Hierbei beträgt der Mittelwert des Preises über alle Gerichte 10€.[eff, 2011]

4.3.4 Feature Hashing

Die drei zuvor genannten Methoden eignen sich vor allem für eine geringe Anzahl möglicher Werte je Kategorie, da die Vektoren linear mitwachsen. Mit *Feature Hashing* kann eine beliebige Menge an möglichen Werten auf eine definierte Anzahl von Dimensionen für den Vektor abgebildet werden. Hierfür wird eine Hash-Funktion genutzt. Die Funktion bildet beliebige Werte auf Zahlen ab. Die Zahlen können dann als „neue Kategorien" betrachtet werden und mit den zuvor genannten Methoden in Vektoren überführt werden. Die Anzahl der Dimensionen lässt sich einstellen, in dem die resultierende Zahl aus der Hash-Funktion modulo der gewünschten Anzahl an Dimensionen gerechnet wird. Hierdurch wird allerdings jegliche Art der Nachvollziehbarkeit verloren. [McMahan et al., 2013]

4.3.5 Bin Counting

Bin Counting verfolgt den Ansatz anstatt die Kategorie einen Vektor zu überführen, wo nur an einer passenden Stelle eine 1 gesetzt wird einen numerische Repräsentation aus anderen Featuren eines Datenpunktes zu berechnen.

Ein Beispiel dafür ist ein Datensatz zu Website-Werbung mit User, Werbungs-ID, den Häufigkeiten wie oft eine Werbung angeklickt und nicht angeklickt wurde. Anstatt nun alle Namen der User mit One-Hot-Encoding in eine hochdimensionalen aber nur spärlich gefüllten Vektor zu überführen wird stellvertretend die Wahrscheinlichkeit genommen, mit der der User auf eine Werbung klickt. [mic, 2017]

5 Fazit und Ausblick

In dieser Arbeit wurde ein genereller Überblick über Feature Engineering gegeben. Zuerst wurde die Motivation hinter Feature Engineering erläutert. Daraufhin wurde betrachtet wo Feature Engineering im Data Mining-Prozess angesiedelt ist. Der Aufbau des Feature Engineering Prozesses wurde genauer betrachtet und wie dabei, neben der Transformation der Daten, die Feature erweitert werden können, um danach die beste Kombination aus Featuren für eine Data Mining Modell zu identifizieren. Im Anschluss wurden unterschiedliche Methoden erläutert, wie Daten in Form von Zahlen, Texten und kategorischer Form zu Featuren aufgewertet werden können. Feature Engineering wird vermutlich noch weiter an Aufmerksamkeit erlangen, da dadurch die Trainingsdauer und Qualität von Data Mining Modellen gesteigert werden kann. Das hat besondere Relevanz, wenn die Modelle größer und komplexer werden wie zum Beispiel im Bereich der Sprachmodelle. Hier enthalten die Modelle mittlerweile 17 Miliarden Parametern [mic, 2020].
Beim Feature Engineering ist es verhältnismäßig leicht mit neuen Datensätzen oder kreativen Kombinationen neue Daten oder Feature zu generieren, wie auch im Abschnitt Feature Enrichment beschrieben. Im Gegensatz dazu ist es schwierig die Menge der Feature auf die relevanten beziehungsweise diejenigen mit der höchsten Informationsdichte zu reduzieren. Um redundante Information in den Featuren zu beseitigen, wird auch die *principal component analysis* (PCA) genutzt. Hierbei werden werden lineare Abhängigkeiten in den Daten identifiziert und ein neues Feature generiert, welches die vorhergehenden ohne Redundanz repräsentiert. Dieses Verfahren wird mittlerweile auch in der Gesichtserkennung genutzt. [Song et al., 2010]

Literatur

(2011) Faq: What is effect coding? https://stats.idre.ucla.edu/other/multpkg/faq/general/faqwhat-is-effect-coding/, (28.02.2020)
(2017) Big learning made easy – with counts! https://docs.microsoft.com/de-de/archive/blogs/machinelearning/big-learning-made-easy-with-counts, (28.02.2020)
(2020) Turing-nlg: A 17-billion-parameter language model by microsoft. https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/, (28.02.2020)

Bouma G (2009) Normalized (pointwise) mutual information in collocation extraction. Proceedings of GSCL pp 31–40

Fayyad U, Piatetsky-Shapiro G, Smyth P (1996) From data mining to knowledge discovery in databases. AI Magazine 17(3):37, DOI 10.1609/aimag.v17i3.1230, URL https://wvvw.aaai.org/ojs/index.php/aimagazine/article/view/1230

Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3(null):1157–1182

Hossain M (2011) The use of box-cox transformation technique in economic and statistical analyses. Journal of Emerging Trends in Economics and Management Sciences 2:32–39

Kotsiantis S, Kanellopoulos D, Pintelas P (2006) Data preprocessing for supervised learning. International Journal of Computer Science 1:111–117

Krishna R, Yu Z, Agrawal A, Dominguez M, Wolf D (2016) The 'bigse' project: Lessons learned from validating industrial text mining. In: 2016 IEEE/ACM 2nd International Workshop on Big Data Software Engineering (BIGDSE), pp 65–71, DOI 10.1109/BIGDSE.2016.019

McMahan HB, Holt G, Sculley D, Young M, Ebner D, Grady J, Nie L, Phillips T, Davydov E, Golovin D, et al (2013) Ad click prediction: A view from the trenches. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA, KDD '13, p 1222–1230, DOI 10.1145/2487575.2488200, URL https://doi.org/10.1145/2487575.2488200

Song F, Guo Z, Mei D (2010) Feature selection using principal component analysis. In: 2010 International Conference on System Science, Engineering Design and Manufacturing Informatization, vol 1, pp 27–30, DOI 10.1109/ICSEM.2010.14

Weir BS (1993) Independence tests for vntr alleles defined as quantile bins. American journal of human genetics 53(5):1107–1113, URL https://pubmed.ncbi.nlm.nih.gov/8105690, 8105690[pmid]

Zheng A, Casari A (2018) Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists, 1st edn. O'Reilly Media, Inc.