



Hintergrund zum Buzzword und Hilfe beim Einstieg

Machine Learning

Miron Kropp, Lisa Töbel

Machine Learning (ML) ist mittlerweile ein viel genannter Begriff. Im Speziellen im Zusammenhang mit Themen wie Big Data, Data Analysis, Data Science, IoT und Industrie 4.0 wird der Begriff häufig verwendet. ML wird dabei als Mittel für Vorhersagen wie Predictive Analysis, aber auch für KI angesehen. AlphaGo hat mit Hilfe von ML einen der besten Go-Spieler der Welt geschlagen. Tesla nutzt ML, um ihren Fahrzeugen autonomes Fahren beizubringen. Doch was verbirgt sich dahinter technisch? Ist es Innovation? Oder doch nur die Anwendung von bekannten Algorithmen. Wie kann es sein, dass auch kleine Start-ups wie Comma.ai (CAI) es wagen können, ein Aftermarket Kit für autonomes Fahren anzukündigen? Der Artikel führt in die Grundlagen von ML ein und bildet die Basis für einen Folgeartikel, der die Leser in die Lage versetzt, selbst ein erstes ML-Projekt durchzuführen.

Das Prinzip von Machine Learning

Um besser zu verstehen, was Machine Learning (oder im Deutschen Maschinelles Lernen, kurz ML) ist und wie es funktioniert, sehen wir uns zunächst die vereinfachten Prozesse des maschinellen Lernens im Vergleich zu der traditionellen Erstellung eines Algorithmus beziehungsweise der traditionellen Art des Programmierens an (s. Abb. 1).

Im traditionellen Ansatz gibt es Eingaben oder Daten. Diese werden mittels eines dafür entwickelten Programms oder Algorithmus durch eine Berechnung verarbeitet. Die Berechnung liefert dann das Ergebnis der Verarbeitung, die Ausgabe.

Im Rahmen von ML haben wir eine Menge von Eingaben und kennen für diese Eingaben (im einfachsten Fall) bereits die Ergebnisse beziehungsweise die Ausgabe. Die Ein- und Ausgabedaten werden dann an den Computer übergeben, der daraus den Algorithmus beziehungsweise das Programm kreiert. Dabei verwendet der Computer entsprechende ML-Algorithmen, auf die wir im späteren Verlauf des Artikels eingehen werden.

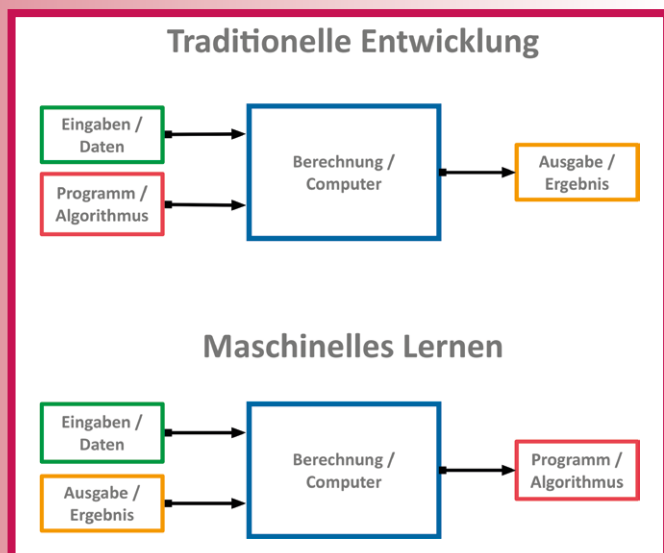


Abb. 1: Traditionelle Entwicklung vs. ML

Das aus dem Lernprozess entstandene Programm kann man nun zum Beispiel für Vorhersagen verwenden (s. Abb. 2). Dabei wird das sich aus dem Maschinellen Lernen ergebende Programm beziehungsweise der Algorithmus als Modell für eine Vorhersage verwendet. Mittels dieses Vorhersagemodells und der neuen Messdaten berechnet der Computer die Vorhersage. Die Daten der Vorhersage können für eine Visualisierung genutzt werden. Man kann noch einen Schritt weitergehen und durch Vergleich der Vorhersage mit tatsächlich gemessenen Ergebnissen die Genauigkeit des Vorhersagemodells ermitteln. Die so gewonnenen Informationen können wieder zurück in das Vorhersagemodell fließen, um dieses zu verbessern.

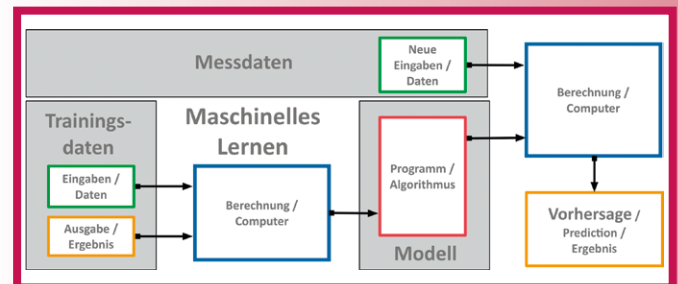


Abb. 2: Einfache Prozessdarstellung für Vorhersagen von Ergebnissen mittels ML

ML-Anwendung in der Praxis und Nutzen

Wie bei anderen Data-Science-Projekten müssen für die Anwendung von ML-Algorithmen und (Teil-)Programmen in der Praxis einige Vorarbeiten durchgeführt werden. ML ist häufig nur ein kleiner, jedoch essenzieller Bestandteil erfolgreicher ML-Projekte. Darüber hinaus sind noch andere Schritte notwendig. Der folgende Ablauf beschreibt die Schritte, die unternommen werden müssen, damit Machine Learning eingesetzt werden kann:

- ▼ **Data Understanding** – verstehen der Domäne, des vorhergehenden Wissens und der Ziele des Projektes: Oft sind Ziele zunächst unklar. Auch muss das implizite Wissen der Domäne erfragt werden, um nicht bei „Null“ anzufangen. Erst wenn man ein gutes Verständnis der Fachlichkeit erlangt hat, kann man die nächsten Schritte durchführen.
- ▼ **Data Collection** – Daten sammeln: Dies ist einer der zentralsten Aspekte, da die Menge und Qualität der Daten maßgeblich Einfluss auf die Ergebnisse des ML haben. Oft sind nicht genügend Daten vorhanden und das Projekt muss so aufgesetzt werden, dass die Daten im Laufe des Projektes gesammelt werden. Je nach Anwendung und Branche/Domäne kann das auch einige Jahre dauern. Auch ist zu definieren, welche Daten aus den Prozessen überhaupt benötigt werden.
- ▼ **Data Cleaning** – Zusammenführung, Auswahl, Säuberung und Vorbereitung der Daten: Die gesammelten Daten liegen oft in verschiedenen Formen und Formaten vor. Diese müssen konsolidiert und zusammengeführt werden. Danach beginnt die Auswahl der tatsächlich nötigen Daten für das Projekt, um die Auswertung möglichst effizient zu gestalten. Dies kann auch beinhalten, zusätzliche externe Daten heranzuziehen, zum Beispiel bei Verkaufsdaten das Wetter. Wenn die richtigen Daten gewählt sind, müssen diese noch gesäubert werden. Darunter versteht man, verzerrende/verfälschende Daten zu entfernen, wie solche von fehlerhaften Sensoren, fehlende Werte zu ergänzen oder auch leere Datensätze zu löschen. Diese aufbereiteten Datensätze müssen zu



guter Letzt für die Verarbeitung mittels der ML-Methoden und -Algorithmen aufbereitet werden. Dieser Teil des Prozesses kann, auch wenn Daten bereits vorhanden sind, bis zu 90 Prozent der Zeit des Projektes in Anspruch nehmen.

- ▼ **Machine Learning** – Modelle des Maschinellen Lernens: Dies ist der Teil, in dem die Modelle des Maschinellen Lernens zum Einsatz kommen: Es muss evaluiert werden, welches Modell mit angepasstem Aufwand ausreichend gute Ergebnisse liefert. Dabei ist zu beachten, dass einfache Algorithmen schon gute Ergebnisse liefern können und der Aufwand, um die Ergebnisse zu verbessern, oft deutlich höher ist als der Nutzen. Als Beispiel kann man die Rossmann Store Sales Kaggle Challenge ansehen [RKC]. Die Sieger schafften eine Genauigkeit von etwa 90 Prozent in der Voraussage der Verkaufszahlen. Dabei wurden ein Zusammenspiel mehrerer relativ aufwendiger Algorithmen und externe Zusatzdaten verwendet. Mit einem einfachen Entscheidungsbaum kommt man nach dem „Data Cleaning“ bereits auf eine höhere Genauigkeit als 75 Prozent. In einigen Zusammenhängen reicht diese Genauigkeit aus. Im Speziellen ist das der Fall, wenn man seinen Stakeholdern überhaupt erst einmal den Nutzen von Machine Learning aufzeigen muss und wenig Mittel und Zeit zur Verfügung hat. Man kann also häufig mit einem kleinen Budget gute Pilotprojekte realisieren, um so die Mittel für größere und aufwendigere Folgeprojekte zu legitimieren.
- ▼ **Generate Knowledge** – Ergebnisse interpretieren: Die gewonnenen Ergebnisse sind je nach verwendetem ML-Modell unter Umständen nicht nachvollziehbar, da diese sich wie eine Blackbox verhalten und Entscheidungen nicht explizit angezeigt werden. In den verschiedenen fachlichen Domänen werden daher, neben den ML-Analysten, fachliche Experten die Richtigkeit der Ergebnisse herausfordern. Um der Herausforderung standzuhalten, müssen die Ergebnisse gut interpretiert und so generiertes Wissen verständlich dargestellt werden. In vielen Fällen ermöglicht ML dabei neue Erkenntnisse oder frühzeitige Information über zukünftige Ereignisse. In einigen Fällen bestätigen die Ergebnisse jedoch nur Bekanntes. Ein positives Beispiel für Knowledge Generation/ML ist eine Bäckereikette, die regelmäßig einen Teil ihrer Produktion von Brötchen wegwerfen musste, da die Kunden anscheinend ohne Zusammenhang mal mehr weiße oder mehr dunkle Brötchen gekauft haben. Mittels ML hat der Algorithmus herausbekommen, dass das Kaufverhalten der Kunden bezüglich der Wahl der Brötchen wetterabhängig ist. Weiße

Brötchen an sonnigen Tagen, dunkle Brötchen an schlechten Tagen. So konnte die Kette die Anzahl der weißen und dunklen Brötchen besser vorhersagen, die Überschussproduktion stark senken und im Zuge dessen auch ihre Kosten [IBM].

- ▼ **Deployment to Production** – Umsetzung in den Betrieb: Nachdem man ein funktionierendes Programm oder einen funktionierenden Algorithmus mit ML realisiert hat, muss dieser noch für die Produktion vorbereitet werden, sodass das System im Dauereinsatz verwendet werden kann. Gegebenenfalls muss auch die oben beschriebene selbstlernende Schleife zur Verbesserung der ML-Vorhersage im Prozess berücksichtigt und realisiert werden. Je nach verwendetem Software-Stack und eingesetztem ML-Stack kann dies zusätzlichen Implementierungsaufwand bedeuten. Auch sollte immer der Wartungsaufwand berücksichtigt werden, den der Produktions-Stack nachher hat. Wenn man mit Java EE-Systemen und WEKA [WDL] arbeiten kann, ist sowohl der Implementierungs- als auch der Wartungsaufwand wahrscheinlich überschaubar. Verwendet man jedoch Python oder R und einen Nicht-Java-Stack, erhöht sich der Aufwand für das korrekte Zusammenspiel der einzelnen Komponenten. Auch steigt der Aufwand für die Wartung, da beim Update verschiedener Technologien, die zu verschiedenen Zeitpunkten in neuen Versionen erscheinen, auch immer auf die Kompatibilität geachtet werden muss. Zusätzlich müssen umfangreichere Tests zur Sicherstellung der Produktivfunktionalität durchgeführt werden. Viele dieser Aspekte und Herausforderungen sind bei einer reinen Java-Lösung deutlich weniger aufwendig.

Arten des Maschinellen Lernens

Es gibt verschiedene Konzepte des Maschinellen Lernens. Zwei Konzepte, überwachtes und unüberwachtes („Supervised“ und „Unsupervised“) Lernen bilden dabei die grundlegenden Konzepte, aus denen verschiedene Algorithmen des Maschinellen Lernens entstanden sind. Zusätzlich gibt es erweiterte ML-Konzepte, die hauptsächlich auf Mischformen der beiden erstgenannten Konzepte beziehungsweise der Anwendung der enthaltenen Algorithmen zurückgehen. Die Algorithmen für jedes der Konzepte können in gleichartige Gruppen eingeteilt werden. Im Folgenden werden einige der Konzepte und Gruppen vorgestellt. Es ist jedoch keine vollständige Liste, da dies den Rahmen dieses Artikels sprengen würde.

„Supervised Learning“ – Überwachtes Lernen

Beim „Supervised Learning“ oder Überwachtem Lernen wird mit Hilfe von Trainingsdatensätzen ein Vorhersage-Algorithmus oder -Programm für eine Fragestellung erzeugt. Der Algorithmus wird dabei mit Hilfe von bekannten Ergebnissen „trainiert“. Dabei wird versucht, eine „mapping function“, eine Abbildungsfunktion zwischen den Eingabe- und den Ausgabedaten zu erstellen. Der Trainingsdatensatz enthält bekannte Datensätze und die zugehörigen, mit späteren Daten vorherzusagenden Ergebnisse. Auf Basis dieses Gesamtdatensatzes (Daten und Ergebnisse) wird das Vorhersageprogramm, also die Abbildungsfunktion, mittels der ML-Algorithmen erstellt. Das sich ergebende Programm oder der Algorithmus wird dann auf die neuen Datensätze (ohne Ergebnisse) für Vorhersagen angewendet.

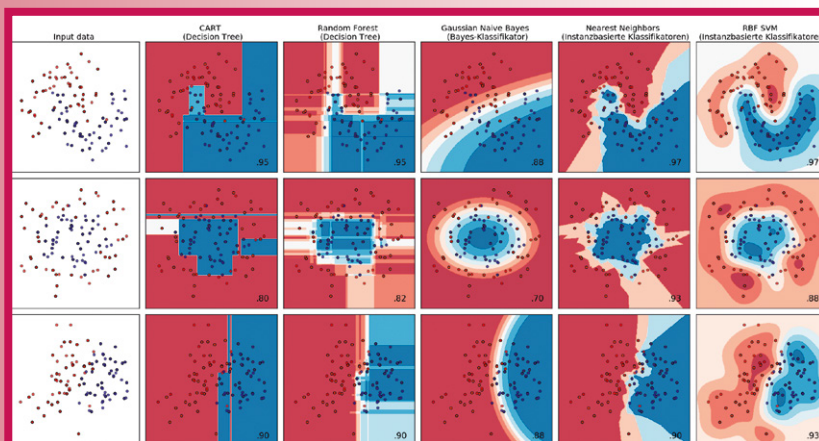


Abb. 3: Vergleich ausgewählter Algorithmen verschiedener Gruppen für drei verschiedene Verteilungen



Überwachtes Lernen wird dabei in zwei Sorten unterteilt: Klassifikation („Classification“) und Regression („Regression“). Bei der Klassifikation ist das Ergebnis eine Kategorie. Dies kann zum Beispiel „blau“, „gelb“ oder „grün“ sein, oder, wie im nachfolgenden Tutorium, einfach Iris-Klassen „setosa“, „versicolor“ oder „virginica“ [Iris]. Hier kann zwischen den einzelnen Kategorien genau unterschieden werden. Bei der Regression hingegen sind die Ergebnisse kontinuierliche reale Werte, zum Beispiel „Euro“ oder „Kilogramm“.

Abbildung 3 zeigt den Vergleich der Anwendung ausgewählter Algorithmen verschiedener Typengruppen für drei verschiedene Verteilungen. Im Allgemeinen können die Algorithmen sowohl für Klassifikation als auch Regression verwendet werden. Im Folgenden betrachten wir vier Gruppen von Algorithmen: Decision Trees, Regressionsalgorithmen, Naive-Bayes-Algorithmen und Instance-Based-Algorithmen.

„Decision Trees“ – Entscheidungsbäume

Algorithmen des ML, die in die Gruppe der Entscheidungsbäume fallen, bauen basierend auf den Trainingsdaten einen Entscheidungsbaum auf. Dieser kann dann für die neuen Daten durchlaufen werden, um so eine Klassifizierung oder eine Voraussage bezüglich der neuen Daten zu ermöglichen.

Ein großer Vorteil der Entscheidungsbäume ist, dass die Ergebnisse des ML-Algorithmus für den Menschen leicht nachvollziehbar sind. Der Baum kann dargestellt und die Entscheidungspunkte können genau benannt werden. Dies ist bei den meisten anderen Algorithmen eher schwer möglich. Daher werden Entscheidungsbäume auch gerne für eine Entscheidungsunterstützung des ML-Problems herangezogen. Aufgrund dieser Eigenschaft verwenden wir auch Entscheidungsbäume in unserem, bereits zu Beginn erwähnten, geplanten Tutorium zum praktischen Einstieg in ML mit WEKA und der Erstellung eines Entscheidungsbaums.

Die Entscheidungsbaum-Algorithmen entscheiden über die einzelnen Knoten des Baums mittels Informationsgehalt. Welcher Wert oder welche Eigenschaft des Datensatzes den höchsten Informationsgehalt hat, wird als Knoten ausgewählt. Um den Informationsgehalt zu bestimmen, werden je nach Algorithmus verschiedene Methoden wie zum Beispiel Entropie (C4.5, C5.0) oder Gini impurity (CART) genutzt. Geteilt wird der Knoten zum Beispiel beim Mittelwert. Abhängig vom Algorithmus kann das Vorgehen etwas abweichen. Das Ergebnis ist jedoch immer ein Entscheidungsbaum. Ausgewählte Algorithmen sind:

- ▼ C4.5 und C5.0, im Tutorium verwenden wir die WEKA-Implementierung J48,
- ▼ Classification and Regression Tree (CART),
- ▼ Chi-squared Automatic Interaction Detection (CHAID) und
- ▼ Random Forest, ein kombinierter Algorithmus, der verschiedene Entscheidungsbäume nutzt und den Mittelwert der Ergebnisse verwendet.

Abbildung 3 zeigt die Ergebnisse der Anwendung von CART und Random Forest in den ersten beiden Spalten. Es ist dabei auffällig, dass die Vorhersagegebiete zu einer bestimmten Klassifizierung rechteckig sind. Dies ist Folge klarer Grenzen innerhalb der Entscheidungsbäume durch das Festlegen von Entscheidungswerten einzelner Eigenschaften.

„Regression Algorithms“ – Regressionsalgorithmen

Regressionsalgorithmen sind iterative Algorithmen, welche die Beziehungen zwischen den Variablen beziehungsweise Daten modellieren. Das Fehlermaß wird für die Verbesserung der Vorhersage des interaktiven Prozesses verwendet. Je nach Algorithmus wird ein anderes zu minimierendes Maß angewandt.

Die Methode stammt aus der Statistik und ist daher eine statistische ML-Methode. Um der Verwirrung vorzubeugen, die Regressionsalgorithmen haben nicht direkt etwas mit der Regression der oben beschriebenen Klassifikation zu tun. Ausgewählte Algorithmen sind:

- ▼ Linear Regression,
- ▼ Logistic Regression und
- ▼ Multivariate Adaptive Regression Splines (MARS).

„Bayesian Algorithms“ – Bayes-Klassifikatoren

Bayes-Klassifikatoren sind Methoden und Algorithmen, die den Satz von Bayes für Klassifikation und Regression anwenden.

Der Satz von Bayes beschreibt mathematisch die Wahrscheinlichkeit für das Eintreten eines Ereignisses basierend auf dem Zusammenspiel einer Zahl unabhängiger Ereignisse, die mit dem zu bestimmenden Ereignis möglicherweise in Beziehung stehen. So kann der Algorithmus Objekte einer Klasse zuordnen, zu der es mit der größten Wahrscheinlichkeit gehört, zum Beispiel, ob eine E-Mail Spam ist oder eine Zuordnung von Maschineneinstellungen, bei der am wenigsten Fehler in der Produktion auftreten. Ausgewählte Algorithmen der Gruppe sind:

- ▼ Naive Bayes,
- ▼ Gaussian Naive Bayes und
- ▼ Bayesian Belief Network (BBN).

Abbildung 3 zeigt in der Spalte 3 eine Gaussian-Naive-Bayes-Klassifikation. Die Grenzen des Vorhersagegebiets sind im Vergleich zu Entscheidungsbäumen fließend und rund, da hier nicht-lineare Funktionen die Grenzen bestimmen.

„Instance Based Algorithms“ – Instanzbasierte Klassifikatoren

Instanz-basierte Algorithmen vergleichen neue Daten mit bereits vorher antrainierten, die als wichtig angesehen und im Speicher vorgehalten werden. Dabei wird typischerweise eine Datenbank mit Beispieldaten angelegt und diese mit neuen Daten verglichen. Der Datensatz mit der größten Ähnlichkeit wird zur Klassifizierung verwendet. Daher nennt man diese Algorithmen auch „Winner-Takes-All“-Methoden, da hier nur der ähnlichste Datensatz zur Klassifizierung reicht. Die Algorithmen unterscheiden sich dabei in der Art, wie die Repräsentation des Datensatzes gespeichert ist und wie die Ähnlichkeitsmessung, eine Art Abstandsmessung zwischen den Ereignissen beziehungsweise Daten, realisiert wird.

Zu der Gruppe der Instanz-basierten Algorithmen gehören unter anderem:

- ▼ k-Nearest Neighbour (kNN),
- ▼ Radial Basis Function Support Vector Machine (RBF SVM) und
- ▼ Learning Vector Quantization (LVQ).

Abbildung 3 zeigt in den Spalten 4 und 5 eine kNN- und eine RBF-SVM-Klassifikation.

„Unsupervised Learning“ – Unüberwachtes Lernen

Unüberwachtes Lernen findet dann statt, wenn der ML-Algorithmus keine Trainingsdaten zur Verfügung hat. Das heißt, es stehen keine zu bestimmende Klassifizierung oder vorherzusagenden Ergebnisse zur Verfügung, sondern nur die Eingabedaten. Ein Modell wird dadurch erstellt, dass Algorithmen des unüberwachten Lernens Strukturen in den Eingabedaten erkennen und die Daten nach diesen Strukturen klassifizieren. Die Herausforderung für die Algorithmen besteht darin, dass diese keine Möglichkeit haben, ihre Ergebnisse zu verifizieren.

Im Weiteren betrachten wir zwei Gruppen von Algorithmen: Clustering und Association Rule Learning. Weitere erwähnenswerte Themen, auf die wir in zukünftigen Veröffentlichungen



eingehen wollen, sind Neuronale Netze, zu denen auch die modernen „Deep Learning“ Methoden gehören, und Dimensions-verringende Algorithmen („Dimensionality Reduction Algorithms“).

„Clustering Algorithms“ – Clusteranalyse

Clusteranalyse ist die Aufgabe, in einem Datensatz die inhärenten Strukturen so zu gliedern, dass die Eigenschaften der gleichen Gruppe eines Clusters auf die eine oder andere Art ähnlicher sind als die Eigenschaften anderer Gruppen. Kurz gesagt, es sollen gleichartige Gruppen gebildet werden. Diese Aufgabe kann von verschiedenen Arten von Algorithmen erreicht werden, die an sich nicht viel Ähnlichkeit miteinander haben. Der Unterschied besteht darin, dass sie andere Vorgaben haben, was ein Cluster ist und was (kleine) Abstände zwischen den Eigenschaften oder Ereignissen sind. Die Abstände können einfache Intervalle sein, aber auch statistische Verteilungen.

Bekannte Algorithmen sind zum Beispiel:

- ▼ k-Means,
- ▼ k-Medians und
- ▼ Hierarchical Clustering.

„Association Rule Learning“ – Assoziationsanalyse

Assoziationsanalyse beschäftigt sich mit Suche nach interessanten Beziehungen zwischen Variablen in großen Datenmengen. Es sollen Elemente gefunden werden, die das Auftreten anderer Elemente innerhalb eines Ablaufs implizieren. Als Beispiel hält meist die Warenkorbanalyse her. Zum Beispiel ist im angloamerikanischen Raum die Auswahl von Zwiebeln und Tomaten ein starkes Indiz dafür, dass auch Hamburger-Fleisch gekauft wird. Daher kann man darauf Werbung und Marketing abstimmen.

Algorithmen, die zu der Gruppe der Assoziationsanalyse gehören, sind:

- ▼ Aprior Algorithm und
- ▼ Eclat Algorithm.

Erweiterte Methoden des maschinellen Lernens

Es gibt neben dem supervised und unsupervised maschinellen Lernen fortgeschrittenere ML-Techniken. Diese sind häufig deutlich aufwendiger in der Umsetzung. Zu diesen Techniken gehören zum Beispiel Semi-Supervised Algorithmen und Algorithmen des Bestärkenden Lernens (Reinforcement Learning), auf die in zukünftigen Artikeln eingegangen werden soll.

Zusammenfassung

Es lässt sich häufig mit relativ einfachen und leicht verständlichen Algorithmen schon einiges, wie Voraussagen von Verkaufszahlen, im Bereich ML erreichen. Hierfür ist es nicht zwingend notwendig, dass man die Algorithmen mathematisch im Detail versteht. Viel wichtiger ist ein logisches Verständnis, sowohl für die Daten als auch für die Algorithmen. So sollte man ein gewisses Verständnis dafür entwickeln, welchen Mehrwert es mit sich bringen kann, bestimmte Ausprägungen von Klassen vorherzusagen und welche Daten für die Vorhersage benötigt werden könnten. Besonders im Bereich Big Data muss eine Vorauswahl der zu analysierenden Daten getroffen werden, um mit den Algorithmen sinnvoll arbeiten zu können.

Auch die Auswahl eines passenden Algorithmus kann ein schwieriges Unterfangen sein. Hierfür wird vor allem Erfahrung im Umgang mit den Daten und den Algorithmen benötigt. Daher sollte man sich nicht scheuen, mit verschiedenen Algorithmen zu experimentieren, um ein Gefühl sowohl für die Daten als auch für die Algorithmen zu entwickeln. Beson-

ders der Decision Tree eignet sich für einen ersten explorativen Umgang mit ML, weil die Entscheidungswege des Algorithmus transparent und leicht verständlich dargestellt werden. Für weitere Informationen zu den Themen Decision Tree und Kreuzvalidierung empfehlen wir das Buch „Data Mining“ [Agg15].

Literatur und Links

[Agg15] C. C. Aggarwal, Data mining: The textbook, Springer, 2015

[CAI] Das kleine Start-up hat ein Aftermarket Kit für autonomes Fahren angekündigt (auch wenn Comma.ai dieses aus regularischen Gründen nicht mehr anbieten wird), <http://comma.ai>

[IBM] IBM Newsroom, Mit IBM Analyse Software den Weihnachtsplätzchen auf der Spur, 15.12.2015, <https://www-03.ibm.com/press/de/de/pressrelease/36255.wss>

[Iris] https://en.wikipedia.org/wiki/Iris_flower_data_set

[RKC] Rossmann Store Sales Kaggle Challenge, <https://www.kaggle.com/c/rossmann-store-sales>

[WDL] Weka 3, Data Mining with Open Source Machine Learning Software in Java, The University of Waikato, <http://www.cs.waikato.ac.nz/ml/weka/downloading.html>



Dr. Miron Kropp hat Physik an der TU Berlin studiert, bevor er bei Siemens in Princeton, NY im Bereich Softwareentwicklung arbeitete. Darauf hat er seine Doktorarbeit im Bereich Sensoren und Halbleiterentwicklung am IMSAS in Bremen abgeschlossen, um anschließend als Projektmanager für Automatisierungstechnik bei BIOTRONIK SE & Co. KG zu arbeiten. Im Anschluss half er als CTO zweier Start-ups, die ersten Wachstumsschmerzen zu überwinden, bevor er seine jetzige Position als Arbeitskreisleiter für Industrie 4.0 bei der akquinet tech@spree GmbH angenommen hat. E-Mail: miron.kropp@akquinet.de

Lisa Töbel hat ihr Diplom an der FU Berlin in Psychologie gemacht. Im Anschluss arbeitete sie an der Universität Konstanz im Fachbereich kognitive Psychologie als wissenschaftliche Mitarbeiterin und steht kurz vor der Promotion. Seit 2016 arbeitet sie bei der akquinet tech@spree GmbH als UX-Beraterin und Data Scientist. E-Mail: Lisa.Toebel@akquinet.de

JavaSPEKTRUM ist eine Fachpublikation des Verlags:

SIGS DATACOM GmbH

Lindlaustraße 2c · 53842 Troisdorf

Tel.: 0 22 41 / 23 41-100 · Fax: 0 22 41 / 23 41-199

E-Mail: info@sigs-datacom.de · www.javaspektrum.de

SIGS DATACOM
FACHINFORMATIONEN FÜR IT-PROFESSIONALS