GDS2020\_Bericht\_Flavio\_Müller

Entscheidungsbäume



Flavio Müller

Windisch, 20.11.2020

**Inhaltsverzeichnis**

[1 Abstract 3](#_Toc56963620)

[2 Einsatzgebiet 3](#_Toc56963621)

[3 Entwicklung für die Anwendung 3](#_Toc56963622)

[4 Funktionsweise des Algorithmus 4](#_Toc56963623)

[4.1 Den Baum wachsen lassen 4](#_Toc56963624)

[4.2 Den Baum zurückschneiden (Pruning) 4](#_Toc56963625)

[4.2.1 Pre Pruning / Hyperparameter tuning 4](#_Toc56963626)

[4.2.2 Post Pruning 4](#_Toc56963627)

[4.3 Testen 4](#_Toc56963628)

[4.4 Brauchen 4](#_Toc56963629)

[5 Vor und Nachteile des Algorithmus 4](#_Toc56963630)

[5.1 Vorteile 4](#_Toc56963631)

[5.2 Nachteile 4](#_Toc56963632)

[5.3 Verbesserungen 5](#_Toc56963633)

[5.4 Vergleich mit anderen ML-Algorithmen (weglassen wenn zu viel) 5](#_Toc56963634)

[6 Anwendungsgebiete 5](#_Toc56963635)

[6.1 Optimale Anwendungsgebiete 5](#_Toc56963636)

[6.2 Nicht optimale Anwendungsgebiete 5](#_Toc56963637)

[7 Fazit 5](#_Toc56963638)

[8 Quellenverzeichnis 5](#_Toc56963639)

[9 Anhänge 6](#_Toc56963640)

# Abstract

tbd

# Einsatzgebiet

Maschinelles Lernen wird grundsätzlich in drei verschiedene Kategorien unterteilt: Supervised Learning, Unsupervised Learning und Reinforced Learning. Die Meisten Algorithmen, darunter auch Entscheidungsbäume werden hauptsächlich oder ausschliesslich in einem dieser drei Gebiete gebraucht. Entscheidungsbäume sind klar im Supervised Learning (deutsch: Überwachtes Lernen) anzusiedeln. Dabei können sie für Klassifikations- und Regressionsprobleme angewandt werden. Die Funktionsweise bleibt dabei die gleiche, es werden Regeln gesucht, welche die zu Verfügung stehenden Daten möglichst homogen trennen. Der Unterschied liegt nur in der Art der Zielvariable (qualitativ oder quantitativ).

Ein weiteres Einsatzgebiet von Entscheidungsbäumen liegt in der explorativen Datenanalyse. Dabei veranschaulich der Entscheidungsbaum die Abhängigkeiten der Zielvariable zu den anderen Variablen. Die damit gewonnenen Erkenntnisse können in weiteren Analysen gezielt eingesetzt werden.

# Entwicklung für die Anwendung

Um ein Entscheidungsbaummodel zu entwickeln, kann man wie in anderen Data Science Projekten nach dem CRISP-Data Mining Modell (Abbildung n) vorgehen. Dabei müssen zuerst Daten gesammelt, exploriert, gereinigt und transformiert werden. Am Ende dieser Prozesse sollten gereinigte und klassifizierte Daten vorliegen. Damit Supervised Learning funktioniert muss eine Zielvariable definiert sein. Diese wird anhand des Anwendungsfalls definiert. Wenn zum Beispiel die Qualität eines Rotweines auf einer Skala von 1-10 bewerten werden soll. Müssen bereits Daten von Rotweinen vorliegen, welche nach dieser Skala bewertet wurden.

Sobald diese klassifizierten Daten vorliegen, werden sie in drei Gruppen aufgeteilt. Die erste und grösste Gruppe (meistens ca. 70%) bilden die Trainingsdaten. Aus Ihnen wächst später der Entscheidungsbaum. Die übrigen Daten werden meist zu gleichen Teilen in Validations- und Testdaten aufgeteilt. Dabei werden die Validationsdaten gebraucht, um die Hyperparameter (z.B. minimale Anzahl Observationen in einem Blattknoten oder maximale Tiefe des Baumes) zu tunen oder aus verschiedenen Machine Learning Modellen das Beste auszuwählen. Die Testdaten werden dem Model nur einmal gezeigt, um die definitive Genauigkeit zu beurteilen.

Wenn das Model trainiert, ein Optimum für die Hyperparameter gefunden wurde und der Baum mittels den Testdaten getestet wurde, wird anhand der Genauigkeit und der Geschwindigkeit des Baumes beurteilt, ob er bereit ist für den Einsatz. Wenn dies der Fall sein sollte, wird das Model in eine Pipeline oder eine Applikation integriert, um es automatisiert nutzen zu können. Während es im Einsatz ist, wird es dauernd überwacht, um es später mit den gesammelten Daten noch weiter zu verfeinern.

Für die Benutzung in der explorativen Datenanalyse, ist müssen die Daten nicht unterteilt werden, da der Entscheidungsbaum zu diesem Zeitpunkt nicht für maschinelles Lernen genutzt wird.

# Funktionsweise des Algorithmus

## Den Baum wachsen lassen

* Verschiedene Algorithmen
* Target Variable
* Error Algorithmen, Metriken um splits zu bewerten
* Evtl. einen Algorithmus genau analysieren

## Den Baum zurückschneiden (Pruning)

### Pre Pruning / Hyperparameter tuning

### Post Pruning

## Testen

## Brauchen

# Vor und Nachteile des Algorithmus

## Vorteile

Der grösste Vorteil eines Entscheidungsbaums als Machine Learning Algorithmus, liegt in der Nachvollziehbarkeit seiner Vorhersagen. Denn diese basieren auf klar definierten und für Menschen nachvollziehbaren Regeln. Weiter sind Entscheidungsbäume sehr beliebt, weil sie für Anfänger und Fachfremde Personen einfach verständlich und anwendbar sind.

Zudem sind die trainierten Modelle der Entscheidungsbäume eher klein und dadurch sehr schnell darin, neue Daten zu klassifizieren. Während bei neuronalen Netzen Millionen von Parametern abgeglichen werden müssen, operieren Entscheidungsbäume nach ein paar definierten Regeln, und sparen so extrem an Laufzeit ein.

Ausserdem müssen die Daten vorgängig weder normalisiert noch skaliert werden, wie das bei anderen Algorithmen der Fall ist. Dies verkürzt die vorgängig nötigen Arbeitsschritte und spart Zeit. Zusätzlich wird der zeitliche Aufwand verkleinert, indem Entscheidungsbäume sehr gut mit fehlenden Werten umgehen können und diese nicht vorgängig bereinigt werden müssen.

## Nachteile

Durch den simplen Aufbau eines Entscheidungsbaumes, ist er anfällig für Bias. Da bei den meisten Algorithmen lediglich eine Variable für einen neuen Split in Betracht gezogen wird, können kombinierte Abhängigkeiten zur Zielvariable nicht repräsentiert werden, was häufig in einem zu simplen Model endet. Ein weiterer grosser Nachteil bei nicht korrekter Handhabung ist die Anfälligkeit für Variance oder Overfitting. Dies geschieht, wenn die Entscheidungsbäume ohne Beschränkungen wachsen, und danach nicht mehr zurückgeschnitten werden (Pruning).

Ein weiterer Nachteil von Entscheidungsbäumen liegt in der Stabilität. Eine kleine Änderung im Trainingsdatensatz könnte eine grosse Änderung in der Struktur des Baumes bewirken, was zu sehr unterschiedlichen Genauigkeiten führen kann.

## Verbesserungen

Um die genannten Nachteile auszugleichen wurden Algorithmen entwickelt, welche zwar auf Entscheidungsbäumen basieren, jedoch entscheidende Vorteile in der Genauigkeit liefern. Eine populäre Methode ist der Random Forest. Wie es der Name bereits vermuten lässt, handelt es sich hierbei um eine Kombination von Entscheidungsbäumen, die zusammen einen Wald repräsentieren.  
Dafür wird für jeden Baum im Wald nur ein Teil des Trainingsset verwendet. Zusätzlich wird bei beim Wachsen des Baumes bei jedem Split nur eine bestimmte Anzahl Variablen als mögliche Split-variable in Betracht gezogen. Dies hat den Effekt, dass auch kleinere und weniger ausgeprägte Strukturen in den Daten erkannt werden können.  
Laut Thais Mayumi Oshiro, Pedro Santoro Perez und José Augusto Baranauskas von der Univerity of Sao Paulo liegt die optimale Anzahl der Entscheidungsbäume in einem Random Forest zwischen 64 und 128. Da mehr Entscheidungsbäume die Komplexität erhöhen, sich aber die Genauigkeit, wenn überhaupt nur gering verbessert. Um nun neue Daten zu klassifizieren, werden die neuen Daten von allen Bäumen im Wald klassifiziert. Danach wird die Entscheidung mit den meisten Stimmen als Vorhersage des gesamten Waldes ausgegeben.

## Vergleich mit anderen ML-Algorithmen (weglassen wenn zu viel)

* SVM
* KNN
* DNN

# Anwendungsgebiete

## Optimale Anwendungsgebiete

Durch ihre Schlichtheit kommen Entscheidungsbäume überall da zum Einsatz, wo die Vorhersage erklärbar und nachvollziehbar sein muss. Ein

* Versicherungswesen -> Einfachheit
* E-Commerce
* Banking
* Meist als Randomforest oder Boosted Trees
* Generell «einfache», für den Menschen verständliche Daten

## Nicht optimale Anwendungsgebiete

* Sehr komplexe Daten (Bilder, Audio usw.)
* Daten welche Menschen schlecht zuordnen können (Pixelwerte von Bildern)

# Fazit

# Quellenverzeichnis

<https://www.cc.gatech.edu/~hic/CS7616/pdf/lecture5.pdf>

https://www.researchgate.net/publication/230766603\_How\_Many\_Trees\_in\_a\_Random\_Forest

# Anhänge

Terminologie Anhand von Beispiel Tree beschreibenA picture containing text

Description automatically generated