GDS2020\_Bericht\_Flavio\_Müller

Entscheidungsbäume



Flavio Müller

Windisch, 20.11.2020

**Inhaltsverzeichnis**

[1 Abstract 3](#_Toc57059095)

[2 Einsatzgebiet 3](#_Toc57059096)

[3 Entwicklung für die Anwendung 3](#_Toc57059097)

[4 Funktionsweise des Algorithmus 4](#_Toc57059098)

[4.1 Den Baum wachsen lassen 4](#_Toc57059099)

[4.1.1 Quantifizierung der Reinheit 4](#_Toc57059100)

[4.2 Den Baum zurückschneiden (Pruning) 4](#_Toc57059101)

[4.3 Testen 4](#_Toc57059102)

[5 Vor und Nachteile des Algorithmus 4](#_Toc57059103)

[5.1 Vorteile 4](#_Toc57059104)

[5.2 Nachteile 5](#_Toc57059105)

[5.3 Verbesserungen 5](#_Toc57059106)

[5.4 Vergleich mit anderen ML-Algorithmen (weglassen wenn zu viel) 5](#_Toc57059107)

[6 Anwendungsgebiete 6](#_Toc57059108)

[6.1 Optimale Anwendungsgebiete 6](#_Toc57059109)

[6.2 Nicht optimale Anwendungsgebiete 6](#_Toc57059110)

[7 Fazit 6](#_Toc57059111)

[8 Quellenverzeichnis 6](#_Toc57059112)

[9 Anhänge 7](#_Toc57059113)

# 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. Tbd bewertung von neuen DAten

# Funktionsweise des Algorithmus

Im Folgenden werde Ich die Funktionsweise eines Entscheidungsbaums anhand des CART Algorithmus aufzeigen.

## Den Baum wachsen lassen

Gegeben, dass ein klassifiziertes Trainingsdatenset vorliegt, kann der Algorithmus anfangen die Daten zu Teilen. Dabei werden alle Variablen als mögliche Teilvariable betrachtet.

Da der CART Algorithmus nur binäre Teilungen machen kann (genau 2 Kinder pro Teilung) müssen kategorische Variablen mit mehr als 2 Levels gruppiert werden. Dabei werden alle möglichen Kombinationen gebildet und die Reinheit der Zielvariable verglichen.

Bei numerischen Teilvariablen wird ein Schwelwert definiert, welcher alle Werte in zwei Gruppen aufteilt. Dabei ist immer grösser als der Kleinste und kleiner als der Grösste Wert. Nun wird für jeden möglichen Wert von die Reinheit der Kinder berechnet.

### Quantifizierung der Reinheit

Die Reinheit wird abhängig vom Typ der Zielvariable (kategorisch oder numerisch) mit unterschiedlichen Algorithmen berechnet. Bei einem numerischen muss eine Fehlerfunktion definiert werden. In unserem Beispiel nutzen wir die empirische Varianz (Formel 1). Dabei wird für beide Untergruppen der Mittelwert berechnet und die Summe der Quadratischen Abweichung zum Mittelwert durch die Anzahl Beobachtungen in der Untergruppe geteilt. Dies wird nun für alle möglichen Schwellwerte (numerische Teilvariable) bzw. Kombinationen von Klassifikationsgruppen (kategorische Teilvariable) berechnet. Dabei wird der Schwellwert, die Kombination als Teilkriterium gewählt, welche die kleinste Varianz erzeugt.

Bei einer kategorischen Zielvariable wird häufig der Gini-Index (Formel 2) als Quantifizierung der Reinheit gewählt, da er einfach zu berechnen ist. Dafür werden bei beiden möglichen Kindern die summierte, quadrierte Wahrscheinlichkeit für das Vorkommen der jeweiligen Klasse berechnet. Danach werden diese Werte nach Anzahl Beobachtung pro Kind gewichtet und summiert. Der resultierende Wert wird zwischen 0 und 1 sein. Hier wird die Kombination bzw. der Schwellwert als Kriterium ausgewählt, welcher den Gini-Index minimiert, da ein tieferer Gini-Index eine höhere Reinheit impliziert.

## Den Baum zurückschneiden (Pruning)

Entscheidungsbäume sind sehr anfällig für Überanpassung. Das heisst, wenn sie nicht zurückgehalten werden, passen sie sich so gut den Trainingsdaten an, dass dies negative Auswirkungen auf die Genauigkeit des Models hat. Um dies zu verhindern gibt es zwei Ansätze:

* Prepruning bzw. Hyperparameter tuning
* Postpruning

Beim Prepruning werden die zur Verfügung stehenden Hyperparamter (maximale Tiefe, mindeste Anzahl von Beobachtungen pro Blatt etc.) vor dem Wachsen des Baumes so angepasst, dass eine möglichst hohe Genauigkeit erzielt wird. Dies geschieht anhand des Validation Dataset. Dabei werden mehrere Testläufe mit unterschiedlichen Hyperparametern gemacht und so iterativ eine optimale Kombination der Hyperparamter gesucht.  
Beim Postpruning (die beiden Varianten werden meist in Kombination gebraucht) wird nach dem Wachsen des Baumes anahnd der Alphafehler-Kummulierung mögliche Subtrees gesucht, welche einen Teil des gesamten Baumes beinhalten. Nun werden diese Subtrees mit dem gesamten Trees verglichen und der Baum mit der besseren Genauigkeit im Validationset als endgültiger Baum verwendet.

## Testen

Um die Genauigkeit des Entscheidungsbaumes zu testen, wird die Zielvariable des Testdatensatzes durch das trainierte Model vorhergesagt. Danach wird bei Klassifikationsbäumen die Fehlerrate (Formel 3) berechnet. Bei Regressionsbäumen wird wiederum ein Streumass (z.B. mittlere quadratische Abweichung oder Varianz) berechnet, um die Fehlerrate zu quantifizieren. So können verschiedene Modelle und Algorithmen miteinander verglichen werden.

# Vor und Nachteile des Algorithmus

Tbd generisches Vergleichsraster

## 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 mögliches Einsatzszenario ist z.B. im Marketing, wenn zum Beispiel soll vorhergesagt werden, ob ein Kunde anfällig für einen Wechsel zur Konkurrenz ist. Weiter findet der Entscheidungsbaum (meist in einem Random Forst o.ä.) eine gute Anwendung bei der Vorhersage von Versicherungsansprüchen. Generell sind Entscheidungsbäume überall dort stark, wo eine falsche Vorhersage keine gravierenden Schäden anrichten würden.

## Nicht optimale Anwendungsgebiete

Ein weniger optimaler Anwendungsfall ist z.B. die Klassifizierung von Objekten in Bildern. Da der grösste Vorteil von Entscheidungsbäumen ihre Schlichtheit ist, macht es wenig Sinn, Pixelwerte von Bildern als Ausgangsdaten zu brauchen, da diese von Menschen nur sehr schwer, wenn überhaupt interpretiert werden können. Weiter sind alle Anwendungsfälle, in denen man sehr genaue Vorhersagen braucht, schlecht für Entscheidungsbäume geeignet. So zum Beispiel in der Medizin oder in der Pharmaindustrie. Dort können für einige Fallstudien auch Entscheidungsbäume angewendet werden, wenn jedoch die Genauigkeit oberste Priorität hat, ist man mit einem neuronalen Netz meist besser bedient.

# Fazit

Entscheidungsbäume sind einfache und nachvollziehbare Machine Learning Algorithmen, welche Regressions- und Klassifikationsprobleme im Supervised Learning lösen können. Nebst dieser Anwendung glänzen sie in der explorativen Datenanalyse. Im Vergleich zu anderen Algorithmen sind sie einfach anzuwenden, da die Daten weder normalisiert noch skaliert werden müssen. Entscheidungsbäume als Machine Learning Algorithmen können auch wie andere Algorithmen mit dem CRISP Datamining Model gebraucht werden. Sie gehen nach dem «Teile & Herrsche» Prinzip vor, wobei bei jeder neuen Teilung die Option gewählt wird, welche die grösste Reinheit in den geteilten Daten bewirkt. Da Entscheidungsbäume eine eher schlechte mittlere Genauigkeit aufweisen, gibt es diverse Verbesserungen, welche auf Entscheidungsbäumen aufbauen, so z.B. Random Forests. Optimale Anwendungsgebiete sind alle, in welchen die Genauigkeit eine untergeordnete Rolle spielt, und die Vorhersage nachvollziehbar sein muss.

# 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