KI für Alle 2: Verstehen, Bewerten, Reflektieren

Themenblock Prognosemodelle (Klassifikation und Regression): 02_03Verfahren_BäumeWälder_01

Der decision tree

Erarbeitet von
Dr. Katja Theune

Lernziele 1
Inhalt 2
Einstieg 2
Decision trees: Idee 2
CART Algorithmus: Idee 3
CART Algorithmus: Gini Index 3
CART Algorithmus: Weiteres Vorgehen 5
Diskussion, Vor- und Nachteile 5
Abschluss 6
Quellen 6
Weiterführendes Material 6
Disclaimer 6

Lernziele

- Du kannst die Idee und Vorgehensweise des CART-Algorithmus für einen decision tree erläutern
- Du kannst den Gini Index als Maß für Verunreinigung anhand eines einfachen Beispiels einordnen
- Du kannst mit Hilfe des Gini Index eine passende Aufteilung der Daten auswählen
- Du kannst Vor- und Nachteile eines decision trees erläutern

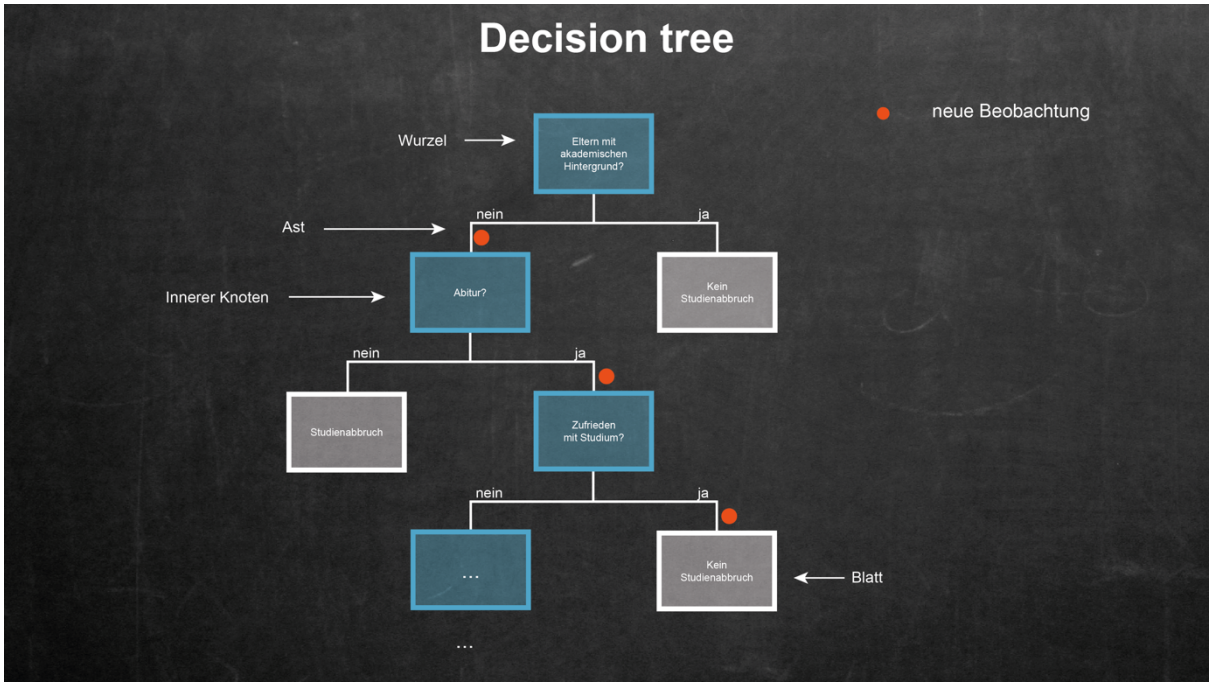
Projekt ai4all „KI für Alle"
HeiCAD –Heine Center for Artificial Intelligence and Data Science

Inhalt

Einstieg

Sogenannte Baum-basierte Verfahren sind wegen ihrer intuitiven Herangehensweise in Wissenschaft und Praxis sehr beliebt. Insbesondere auch, weil sie sich sehr anschaulich darstellen und nachvollziehen lassen. Vor allem, wenn viele Bäume zu einem Wald kombiniert werden, ergeben sich auch sehr leistungsfähige Verfahren zur genauen Vorhersage. Starten wir aber zunächst mit einem einzelnen Baum.

Decision trees: Idee



Bei einem Entscheidungsbaum oder englisch decision tree klassifizieren wir eine neue Beobachtung dadurch, dass wir sie einen Pfad von Entscheidungsregeln, den sogenannten Ästen, von der Wurzel bis hin zu den Blättern entlang schicken. Die Wurzel repräsentiert dabei den Trainingsdatensatz, welcher nach und nach in Teildatensätze bzw. innere Knoten aufgeteilt werden soll. Die untersten Knoten sind die Blätter. Sie sollen möglichst viele Beobachtungen einer bestimmten Klasse enthalten und bestimmen die prognostizierten individuellen Klassenwahrscheinlichkeiten bzw. die konkrete Klassenzugehörigkeit.

Baum-basierte Verfahren können wir auch für eine Regression verwenden. Dann wird der prognostizierte Output für eine neue Beobachtung als Mittel der Output-Werte aller Beobachtungen in dem zugehörigen Blatt berechnet.

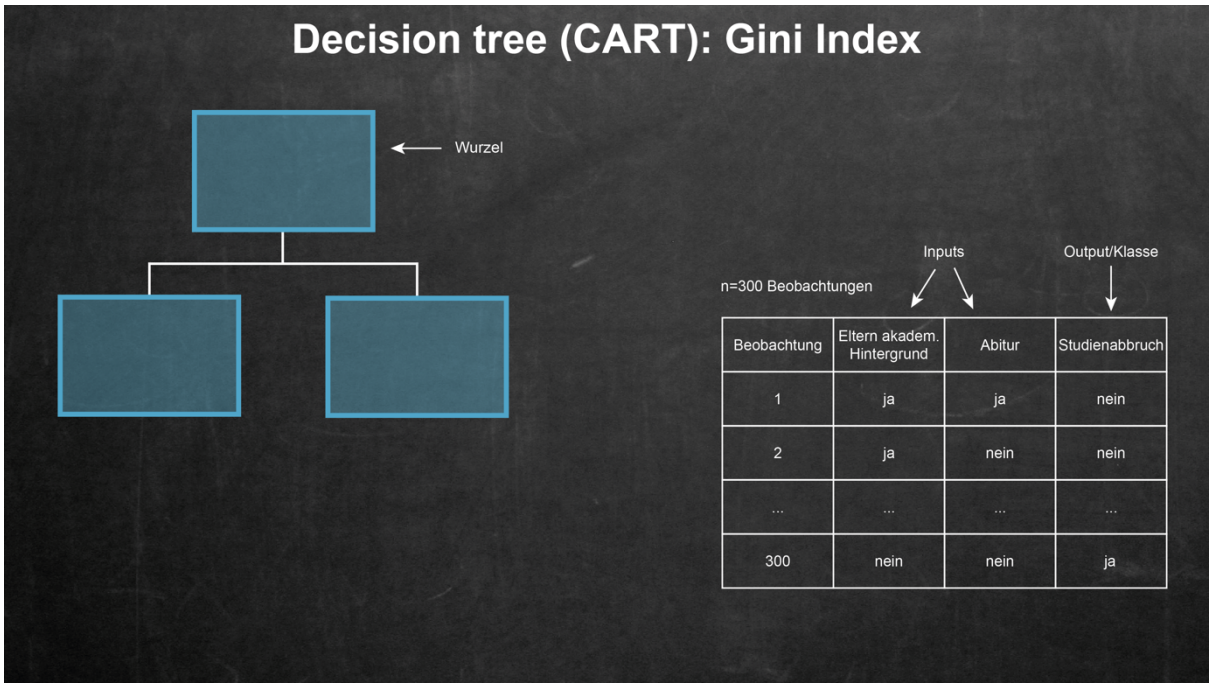
CART Algorithmus: Idee

Wie können wir aber nun die Trainingsdaten in Blätter aufspalten? Hier möchte ich jetzt stellvertretend für verschieden Methoden die Grundidee des sehr beliebten CART-Algorithmus erläutern. CART steht für Classification And Regression Trees. Er zeichnet sich durch eine binäre Aufspaltung aus, d. h. jeder Knoten wird in genau zwei Teildatensätze geteilt. Um genauer zu verstehen, wie das funktioniert, betrachten wir den Fall der Klassifikation.

Quelle [1]

Wir suchen nun denjenigen Input und seine zugehörige Ausprägung, welcher die "Verunreinigung" oder auch Vermischung der Klassen in den nachfolgenden Teildatensätzen am meisten reduziert und damit auch die Prognose optimiert. Hierfür gibt es verschiedene Möglichkeiten der Messung. Beim CART-Algorithmus wird der sogenannte Gini Index G verwendet.

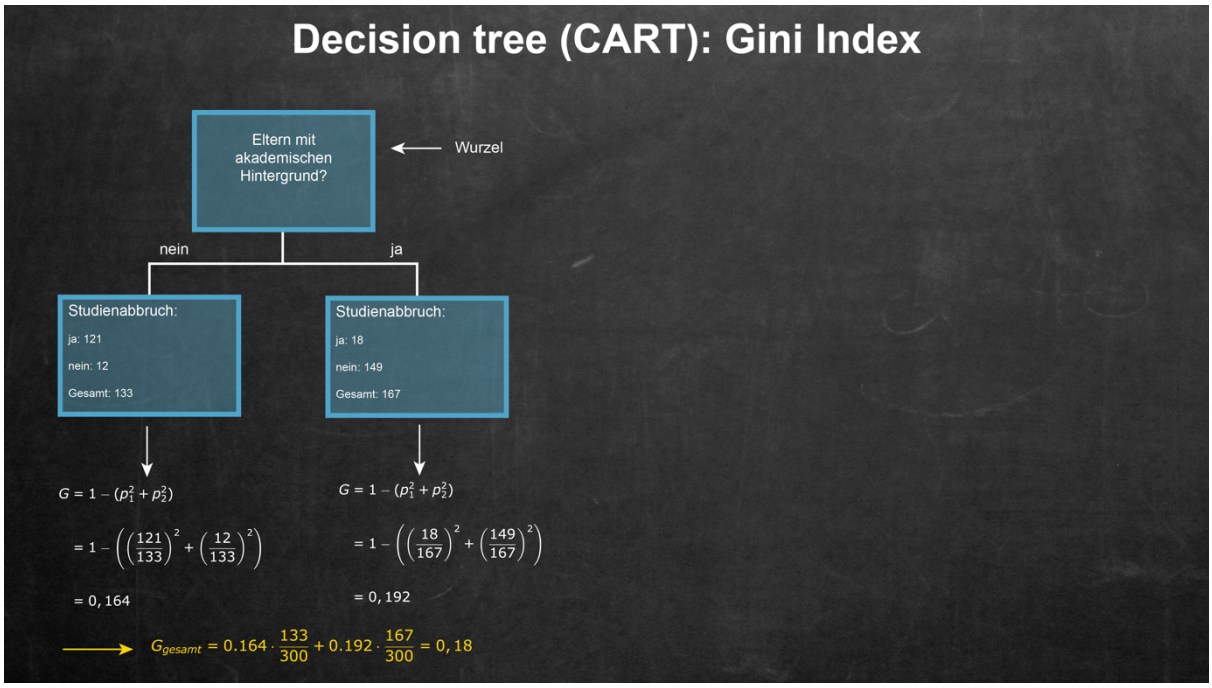
CART Algorithmus: Gini Index



Zur Veranschaulichung verwenden wir wieder einen fiktiven Beispieldatensatz, diesmal mit 300 Studierenden. Die beiden Inputs sind der akademische Bildungshintergrund der Eltern und ob die Studierenden ein Abitur haben. Der Output, also unsere Klassen, beschreibt, ob ein Studium abgebrochen wurde oder nicht. Wir haben also zwei Klassen.

Jetzt wollen wir überprüfen, wie die erste Aufteilung der Wurzel aussehen soll, also welcher Input sich am besten eignet. Nehmen wir zunächst den akademischen Hintergrund der Eltern. Alle mit der Ausprägung „nein" landen im linken und alle mit „ja" im rechten Knoten. Nehmen wir an, das würde im linken Knoten zu einer Klassenverteilung von 121

Studierenden mit Studienabbruch und 12 Studierenden ohne Studienabbruch führen. Im rechten Knoten nehmen wir eine Verteilung von 18 zu 149 an. Wir haben damit eine Gesamtanzahl an Beobachtungen von 133 im linken und 167 im rechten Knoten.



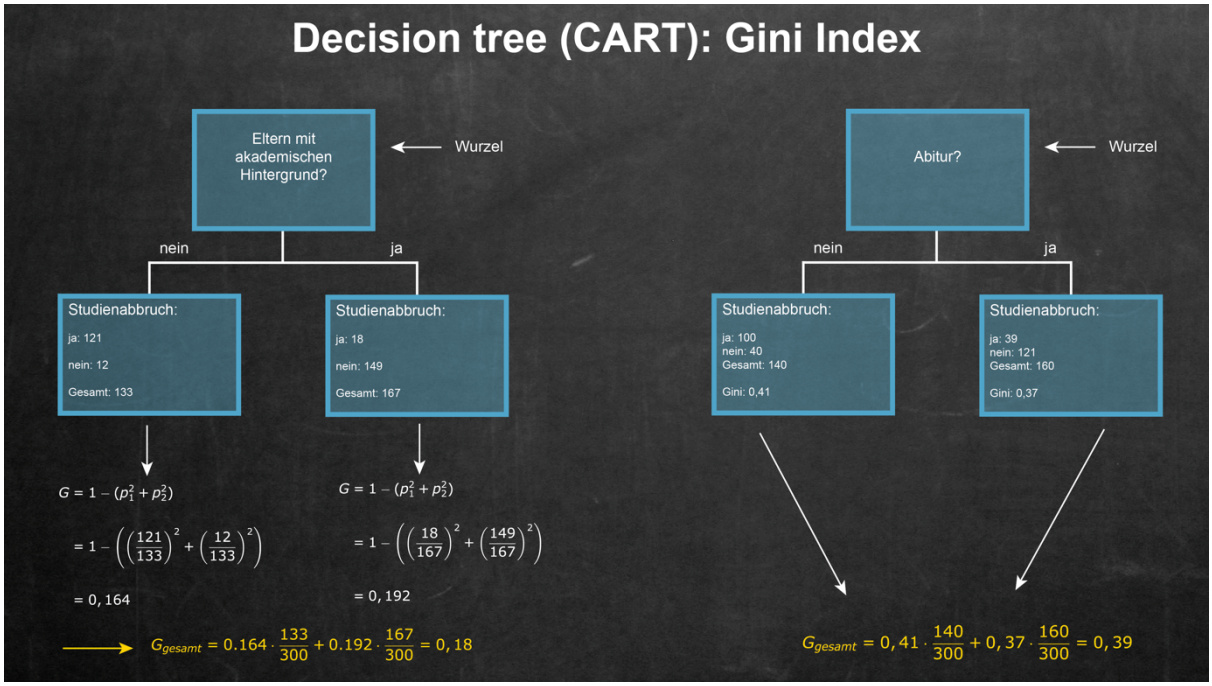
Damit können wir dann die Verunreinigung bzw. die Gini Indizes in beiden Knoten berechnen (siehe Grafik). Wir beginnen mit dem linken Knoten und benötigen dafür die sogenannten Klassenanteile p in diesem Knoten. p_1 ist z. B. der Anteil an Beobachtungen der Klasse 1, bei uns „Studienabbruch", an allen Beobachtungen in diesem Knoten. Wir sagen auch die Wahrscheinlichkeit für einen Studienabbruch.

Im linken Knoten wäre unser p_1 dann 121/133 und p_2 wäre hier 12/133. Diese beiden Anteile werden dann quadriert und aufsummiert und von 1 abgezogen. Wir erhalten dann einen Gini Index von 0,164. Im rechten Knoten wäre der Gini Index dann analog gleich 0,192.

Der Gesamtwert für den Gini Index für diese Aufteilung ergibt sich dann dadurch, dass wir diese beiden Indizes jeweils mit der relativen Knotengröße multiplizieren, also ein gewichtetes Mittel bilden (siehe Grafik). Die relative Knotengröße vom linken Knoten wäre z. B. die Anzahl an Beobachtungen im linken Knoten durch die Gesamtzahl an Beobachtungen, also 133/300. Wir erhalten einen Gesamt-Gini Index von 0,18. Das ist dann die Verunreinigung, die sich durch die Verwendung des Inputs „akademischer Hintergrund der Eltern" ergibt.

Hier noch kurz ein Hinweis zur Interpretation der Formel: der Gini Index wird demnach klein, wenn die Klassenanteile nahe bei 1 oder 0 sind, also viele Beobachtungen in einem Blatt einer bestimmten Klasse angehören, was wir ja wollen. Er ist sogar gleich 0, wenn nur eine Klasse im Blatt vertreten ist, da dann entweder $p_1 = 0$ und $p_2 = 1$ ist oder andersherum. Wir können uns merken: je kleiner der Gini Index, desto besser.

Für den zweiten Input „Abitur" ergibt sich ein gesamter Gini Index von 0,39, welcher größer als der von gerade ist (siehe Grafik). Damit würden wir für die erste Aufteilung der Wurzel den Input wählen, der zu einem kleineren Gini Index führt, also den Bildungshintergrund der Eltern.



CART Algorithmus: Weiteres Vorgehen

Dieser Vorgang wiederholt sich nun an jedem neu entstehenden Knoten, bis wir durch eine weitere Aufteilung keine genügend große Verbesserung des Gini Index mehr erhalten. Wobei wir vorher festlegen müssen, was „genügend groß" ist. Oft ist es aber besser, den Baum zunächst weiter wachsen zu lassen als wir es eigentlich als optimal erwarten würden. Im Nachhinein kann er dann wieder abgeschnitten bzw. gestutzt werden. Das nennt man auch Pruning. Das kann Sinn machen, da wir evtl. spätere lohnenswerte Verbesserungen durch weitere Aufspaltung beurteilen können und nicht vorzeitig abbrechen. Den besten gestutzten Baum kann man z. B. mit einer cross-validation finden.

In einem Regressionssetting mit einem metrischen Output bleibt das Prinzip der Datenaufspaltung gleich. Als Maß der Verunreinigung kann man hier z. B. als Basis die Abweichung zwischen den wahren Outputs und den prognostizierten Outputs in den jeweiligen Knoten berechnen.

Diskussion, Vor- und Nachteile

Decison trees haben den Vorteil, dass sie intuitiver und nachvollziehbarer als andere Verfahren des maschinellen Lernens sind. Zudem können sie gut mit allen Arten von Inputs, also z. B. kategorial oder metrisch, und auch fehlenden Werten umgehen. Sie sind auch robuster gegenüber Ausreißern. Das macht die Datenaufbereitung weniger aufwendig.

Außerdem können sie auch komplexe Zusammenhänge zwischen den Inputs und Outputs abbilden.
Ein bedeutender Nachteil ist, dass tiefe Bäume mit vielen Knoten oft eine hohe Varianz haben, d. h. bei Anwendung auf einen veränderten Datensatz können wir zu ganz anderen Ergebnissen gelangen. Bäume mit wenigen Knoten neigen dagegen zu einer höheren Verzerrung. Häufig ist die Prognosegenauigkeit auch nicht so gut wie bei anderen Verfahren.

Abschluss

Wir haben nun den CART-Algorithmus für einen decision tree und den Gini Index als Maß der Verunreinigung kennengelernt, anhand derer der Trainingsdatensatz aufgespalten wird. Dieses Verfahren ist sehr intuitiv nachvollziehbar und liefert damit Einblicke in die Funktionsweise und Entscheidungsfindung des Prognosemodells. Das geht aber häufig zum Nachteil der Prognosegenauigkeit.

Quellen

Quelle [1] Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Weiterführendes Material

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques* (3. Auflage). Morgan Kaufmann.

James, G., Witten, D., Hastie, T., & Tibshirani, R., & Tylor, J. (2023). *An Introduction to Statistical Learning - with Applications in Python*. Springer.

Lantz, B. (2015). *Machine learning with R* (2. Auflage). Packt Publishing Ltd, Birmingham.

Disclaimer