Woche 08 Theorie: Bäume und Wälder

Erarbeitet von
Katja Theune

Lernziele 1
Inhalt 2
Einstieg 2
Decision tree – Beispiel 2
Decision tree – Der Name ist Programm 2
Decision tree – Aufteilung der Trainingsdaten 3
Decision tree – Beispielbaum 4
Decision forest – Random forest 7
Abschluss 8
Weiterführendes Material 9
Disclaimer 10

Lernziele

- Erläutern der Idee und Vorgehensweise eines decision trees und random forests
- Anwenden der Vorgehensweise der Verfahren auf ein neues Beispiel
- Beispiele nennen, wozu man decision trees und random forests verwendet

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

Inhalt

Einstieg

Vermutlich hat jede und jeder von uns schon mal im Alltag versucht, eine Entscheidung zu treffen, indem wir – vielleicht auch unbewusst – nach und nach bestimmte Kriterien geprüft haben. Z. B. bei der Wahl der Freizeitaktivitäten, ob es regnen soll oder trocken bleibt, ob es kalt oder warm wird, usw.

Einblendung einfacher Entscheidungsbaum mit Freizeitaktivitäten

Auf diesem Prinzip beruhen auch decision trees und forests oder auf Deutsch Entscheidungsbäume und -wälder. Sie gehören zum supervised learning und sind sogenannte Baum-basierte Verfahren der Klassifikation. Wir können sie auch für Regressionsprobleme verwenden. Hier beschäftigen wir uns jetzt aber erstmal nur mit der Klassifikation. Die Idee lässt sich aber ganz einfach auf ein Regressionsproblem übertragen.

Decision tree – Beispiel

Zum besseren Verständnis schauen wir uns hier einmal ein klassisches Beispiel aus der Praxis an. Und zwar möchte ein Kreditinstitut eine Entscheidung darüber treffen, ob ein Kreditantrag von einem Kunden oder einer Kundin genehmigt wird oder nicht. Unsere Beobachtungen sind hier Antragstellende, die wir entweder der Klasse „kreditwürdig“ oder der Klasse „nicht kreditwürdig“ zuordnen wollen. Wir haben also ein Klassifikationsproblem mit zwei Klassen.

Mit Hilfe unserer Trainingsdaten wollen wir nun einen decision tree erstellen. Die Trainingsdaten beinhalten Informationen, sagen wir über 20 bereits abgewickelte Kreditanträge und ihre Antragstellenden. Wir könnten uns z. B. folgende relevante features vorstellen, mit denen die Kreditwürdigkeit geprüft werden soll: die Kredithistorie, also ob jemand bereits positiv oder negativ bei Kreditaufnahme aufgefallen ist, das Einkommen und die Dauer des bestehenden Beschäftigungsverhältnisses. Wir kennen zudem die jeweilige Klasse der Antragstellenden.

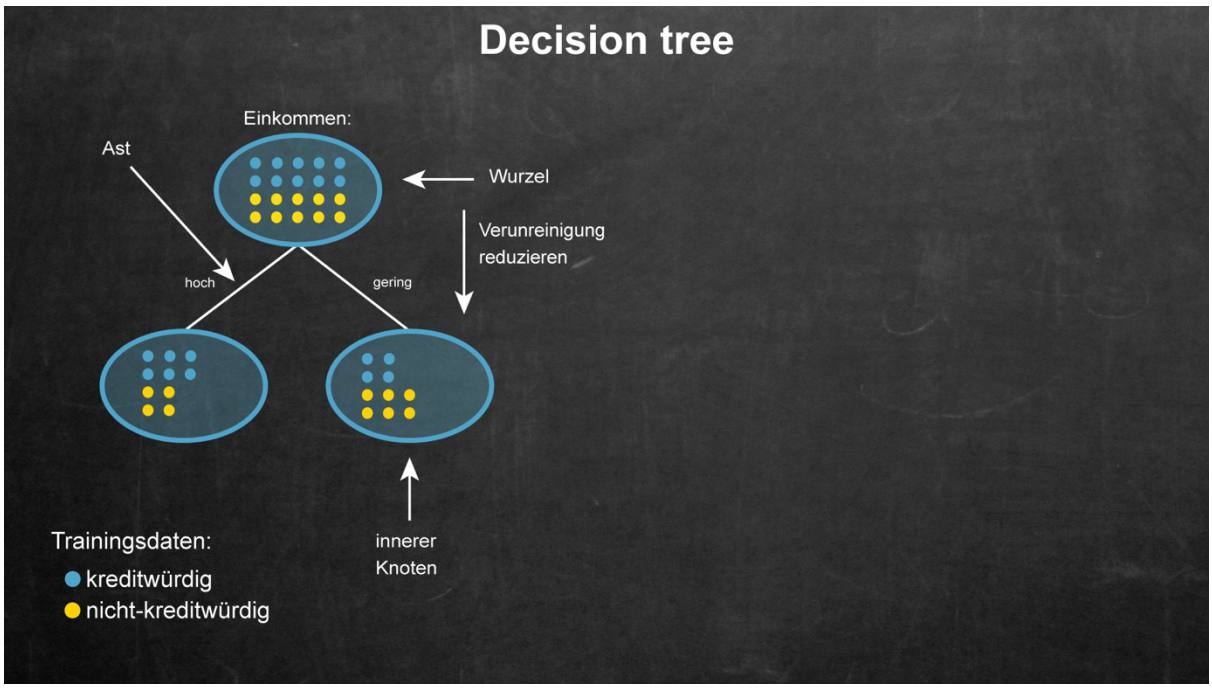
Decision tree – Der Name ist Programm

Aber wie genau funktioniert jetzt so ein decision tree? Wir können ihn uns tatsächlich wie einen echten, nur umgedrehten Baum vorstellen, bei dem man sich von der Wurzel bis zu den Blättern vorarbeitet. Man nennt das auch einen Top-Down-Ansatz. Die Wurzel repräsentiert dabei unseren Trainingsdatensatz, in welchem alle Beobachtungen enthalten sind. Der Algorithmus dahinter versucht kurz gesagt, die Wurzel – also die Trainingsdaten – nach und nach in Blätter aufzuspalten und so sogenannte Entscheidungsregeln abzuleiten. Die Blätter bestimmen die Klassen und die einzelnen Entscheidungen werden durch die Äste repräsentiert.

Tauchen wir mal ein wenig tiefer in die Funktionsweise eines decision trees ein. Es gibt sehr viele verschiedene Methoden, wie genau so ein Baum erzeugt werden kann, aber im Grunde verfolgen sie alle eine ähnliche Idee.

Decision tree – Aufteilung der Trainingsdaten

Zu Beginn betrachten wir den uns vorliegenden Trainingsdatensatz bzw. die Wurzel mit unseren 20 Beobachtungen. Sie werden durch die Punkte repräsentiert. Die 10 blauen Beobachtungen haben die Klasse „kreditwürdig“ und die 10 gelben die Klasse „nicht-kreditwürdig“. Die Wurzel soll nun in Teildatensätze, sogenannte innere Knoten, aufgeteilt werden. Die Pfade dazwischen sind unsere Äste.



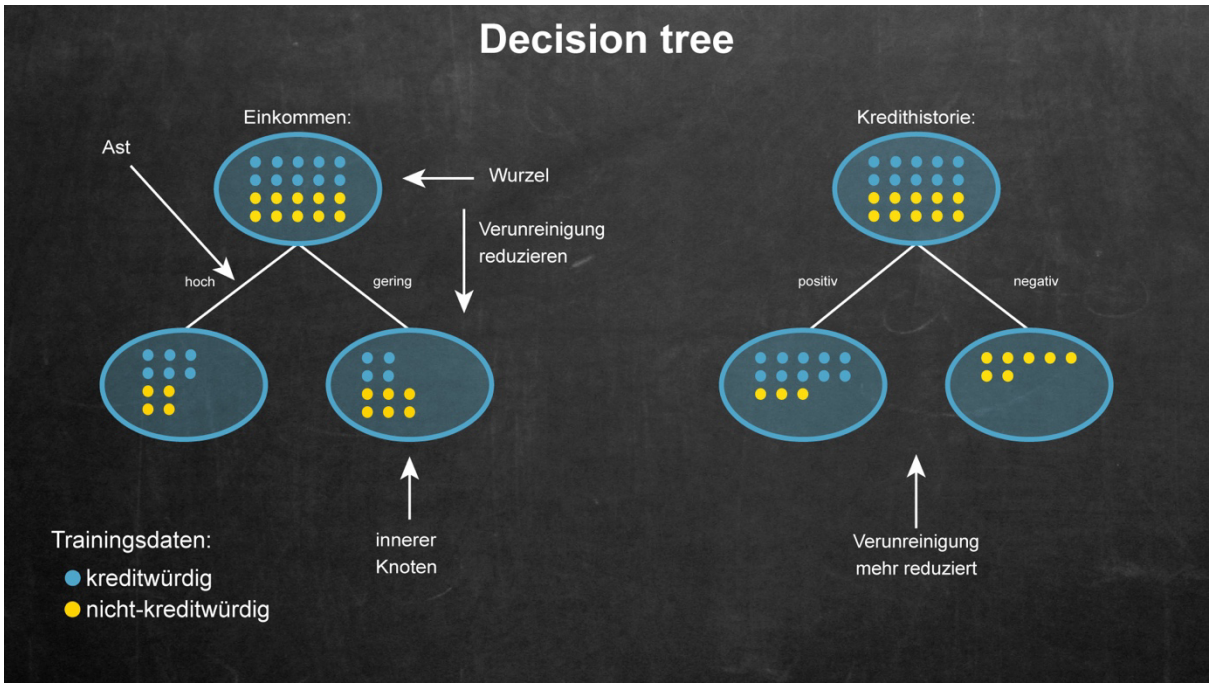
Wir sagen hier der Einfachheit halber, dass der Trainingsdatensatz zunächst in zwei Teildatensätze geteilt werden soll. Aber es sind natürlich auch andere Verfahren möglich. Aber wie erfolgt nun diese erste Aufteilung? Wir suchen im Prinzip diejenige Aufteilung, welche die sogenannte „Verunreinigung“ in den Teildatensätzen am meisten reduziert.

Mit Verunreinigung ist die Vermischung der beiden Klassen gemeint. Unser Ziel ist es, dass wir diese Vermischung möglichst gut verringern. D. h. wir wollen, dass möglichst viele Beobachtungen in den entstehenden Teildatensätzen nur einer bestimmten Klasse angehören.

Z. B. könnten wir uns hier anschauen, wie der Anteil an kreditwürdigen und nicht-kreditwürdigen Antragstellenden in den Teildatensätzen aussehen würde, wenn wir alle Beobachtungen mit einem hohen Einkommen in den linken und alle mit einem geringen Einkommen in den rechten Teildatensatz packen. Wir sehen hier, dass sich die

Verunreinigung schon ein wenig verringert, denn im Gegensatz zur Wurzel gibt es hier jeweils schon eine dominierende Klasse.

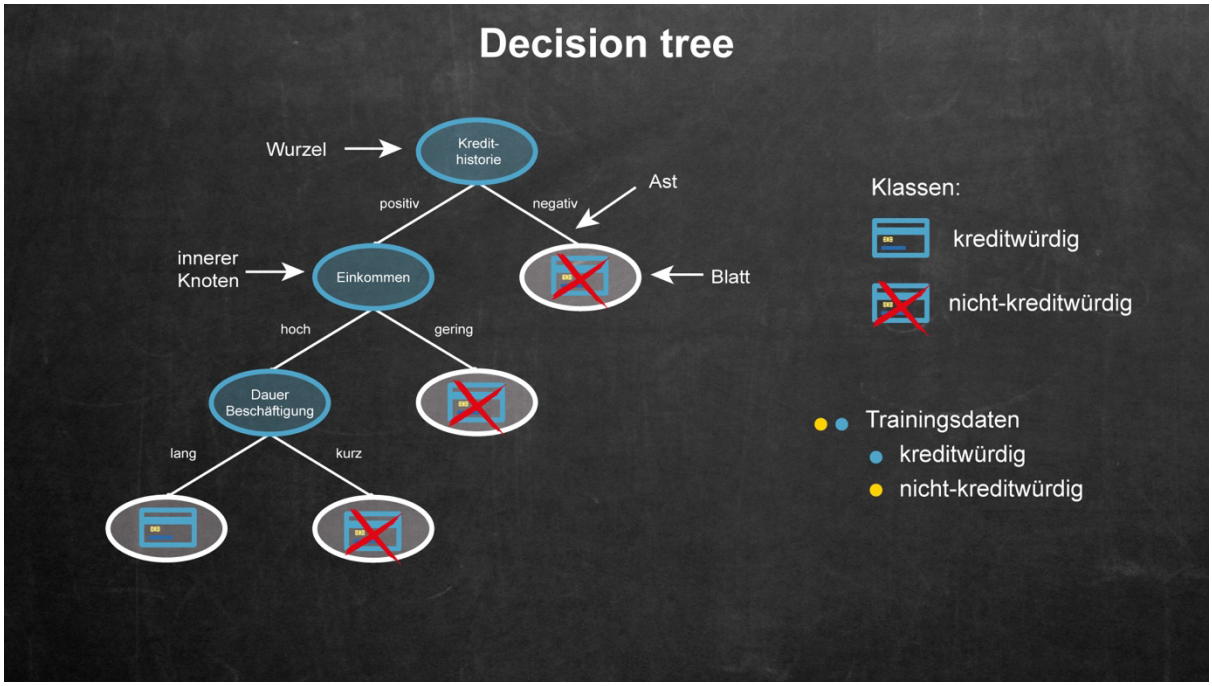
Schauen wir uns jetzt mal eine Aufteilung der Trainingsdaten bzgl. der Kredithistorie an. Wir sehen, dass sich hier die Vermischung der Klassen noch mehr verringert als bei einer Aufteilung bzgl. des Einkommens. Eine Klasse überwiegt hier jeweils deutlich. Das machen wir dann für jedes feature und schauen, wo die Verunreinigung sich am meisten reduziert. Das ist dann unsere erste Aufteilung der Trainingsdaten. An den so neu entstandenen Teildatensätzen bzw. inneren Knoten wiederholen wir diesen Vorgang und suchen wieder die beste Aufteilung. Die untersten Knoten, also die, die nicht weiter aufgeteilt werden, sind dann die Blätter. Unser Ziel ist, dass möglichst viele Beobachtungen in einem Blatt nur einer Klasse angehören. In unserem Beispiel hieße das, dass unser Trainingsdatensatz möglichst gut in kreditwürdige und nicht-kreditwürdige Beobachtungen getrennt wird.



Jetzt kann man sich noch fragen, wie oft wir solche Aufteilungen vornehmen wollen bzw. wie weit wir den Baum „wachsen“ lassen wollen. Hierzu kann man z. B. ein Kriterium wählen, wann das Wachstum gestoppt werden soll, der Baum also „gestutzt“ wird. Ein mögliches Kriterium wäre eine erreichte Minimalanzahl an Beobachtungen in den Knoten, also z. B. mindestens 10 Beobachtungen.

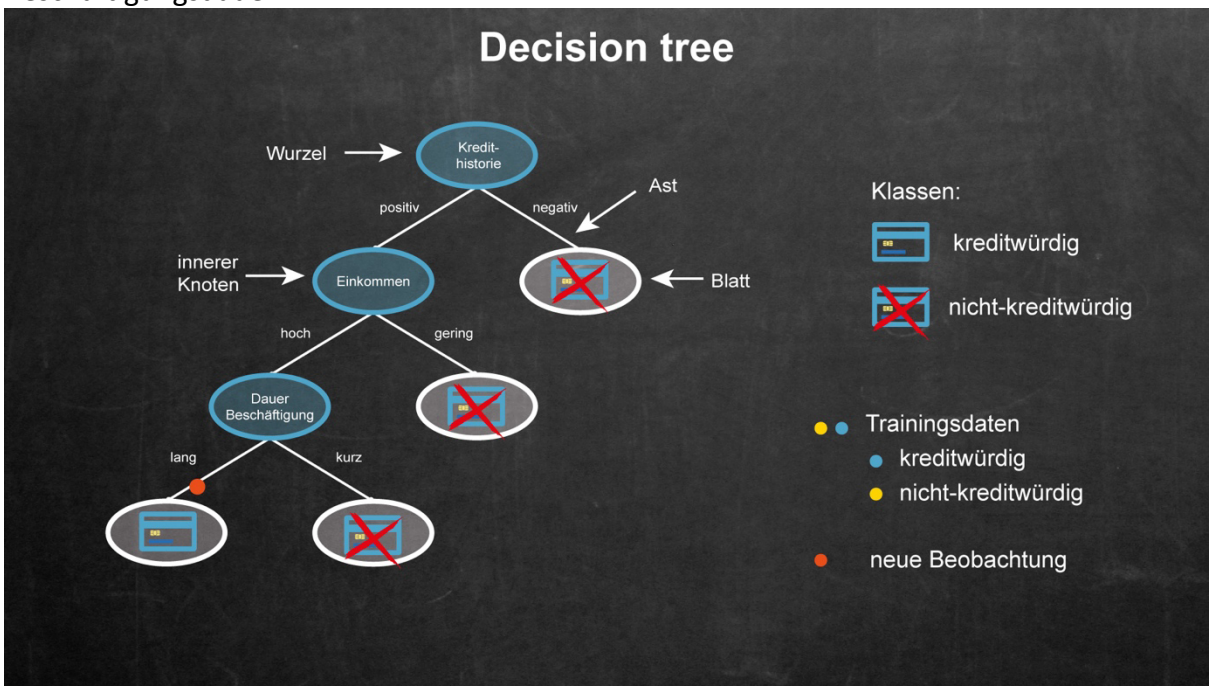
Decision tree – Beispielbaum

Zur Veranschaulichung sehen wir hier beispielhaft einen möglichen decision tree.

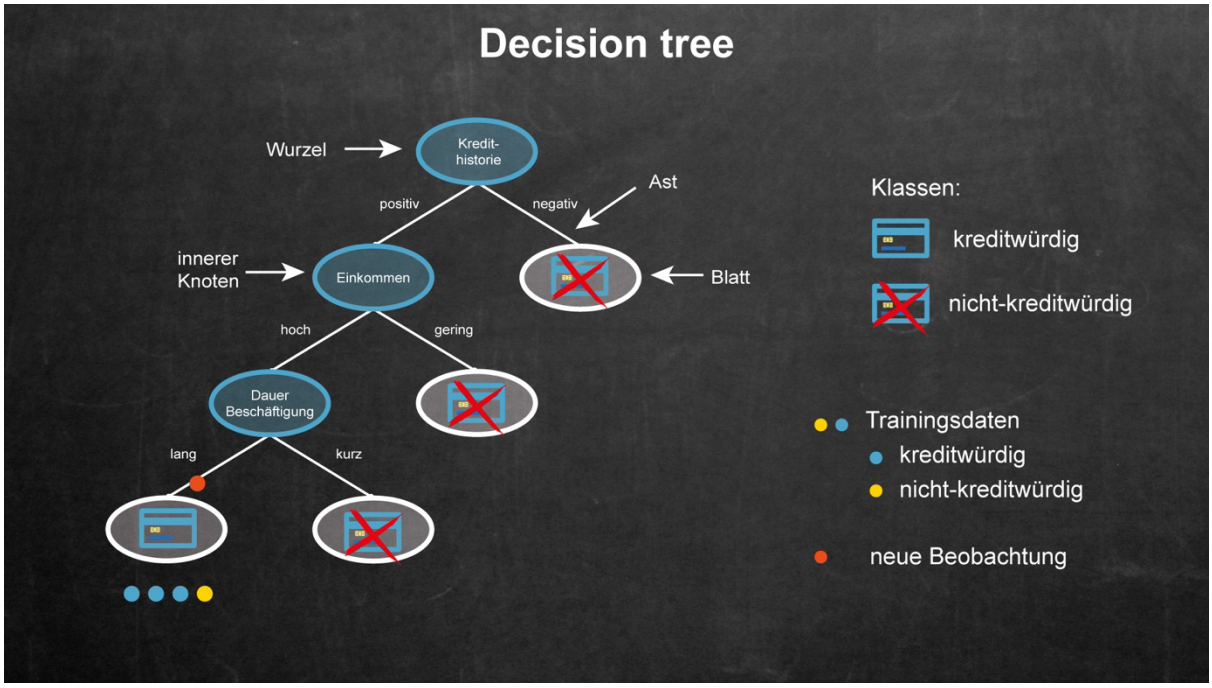


Als Erstes wurde hier das feature „Kredithistorie“ zur Aufteilung verwendet. Beispielsweise befinden sich in den ersten beiden neu entstandenen Knoten jetzt auf der linken Seite nur noch Personen mit positiver und auf der rechten Seite nur noch Personen mit negativer Kredithistorie. Danach wurde das Einkommen und zuletzt die Dauer des bestehenden Beschäftigungsverhältnisses für die weitere Aufteilung ausgesucht.

Aber was genau wollen wir jetzt eigentlich mit dem so entstandenen decision tree machen? Wir wollen neue Beobachtungen – hier neue Antragstellende – einer der beiden Klassen zuordnen. Eine neue Beobachtung ist hier durch den orangenen Punkt gekennzeichnet. Diese hat, sagen wir, eine positive Kredithistorie, ein hohes Einkommen und eine lange Beschäftigungsdauer.



Wir lassen sie jetzt dem Pfad von Entscheidungen, also quasi der Verästelung, von der Wurzel bis hin zu den Blättern folgen. An jedem Knoten überprüfen wir, welchem Ast sie folgen muss. Im ersten Knoten schicken wir sie den linken Ast hinunter, da ihre Kredithistorie positiv ist. Im zweiten Knoten schauen wir uns ihr Einkommen an. Da dieses hoch ist, schicken wir unsere Beobachtung wieder den linken Ast hinunter. Das Gleiche gilt für die Frage nach der Beschäftigungsdauer, welche für unsere Beobachtung lang ist. Unsere Beobachtung landet also im linken Blatt. Die Prognose für diese Beobachtungen, also z. B., ob sie kreditwürdig ist oder nicht, richtet sich nach dem Blatt, in welchem sie am Ende gelandet ist. Dies könnte dann z. B. die in diesem Blatt vorherrschende Klasse sein.



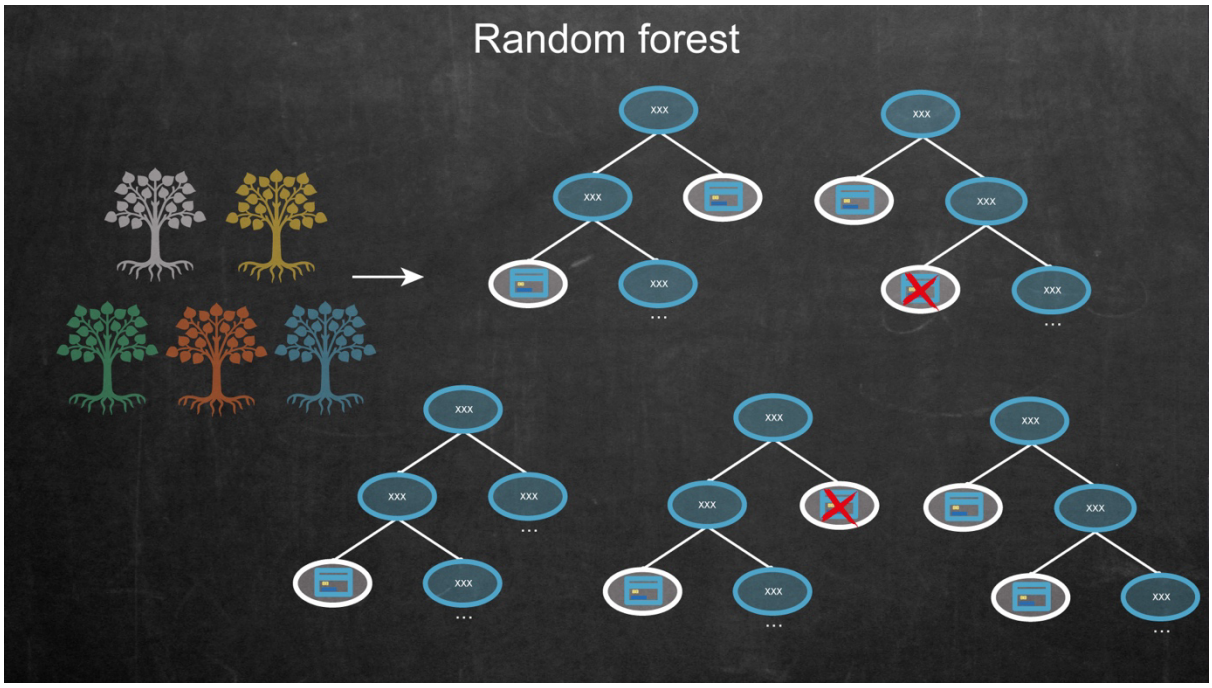
Im linken Blatt sind drei kreditwürdige und eine nicht-kreditwürdige Beobachtung enthalten. Die vorherrschende Klasse ist also „kreditwürdig“. Für unsere neue Beobachtung prognostizieren wir also auch die Klasse „kreditwürdig“. Wir können unserer Beobachtung aber auch eine individuelle Klassen-Wahrscheinlichkeit zuordnen. Sie entspricht dann dem Anteil an Beobachtungen dieser Klasse im jeweiligen Blatt. Hier wäre es z. B. eine Wahrscheinlichkeit von 75 % zur Klasse „kreditwürdig“ und 25 % zur Klasse „nicht-kreditwürdig“ zu gehören.

Decison trees haben den Vorteil, dass sie sehr intuitiv und anschaulich darstellbar sind. Man kann sie im Vergleich zu anderen Methoden im Machine Learning einfach interpretieren. Ein Nachteil ist, dass decision trees dazu tendieren, sehr instabil zu sein. D. h. wenn sich die Daten ein wenig ändern, könnten wir zu komplett anderen Ergebnissen gelangen. Sie machen auch oft weniger treffende Vorhersagen als andere Klassifikationsmethoden.

Decision forest – Random forest

Eine Möglichkeit, das zu verbessern, ist mehrere unterschiedliche Bäume zu erzeugen und sie dann zu aggregieren. Man nennt das decision forests. Es gibt viele verschiedene Methoden der Aggregation. Beliebt ist hier der sogenannte random forest.

Die Idee ist hier, jeden Baum innerhalb des Waldes auf Basis von unterschiedlichen Teildaten unseres Trainingsdatensatzes zu bilden. Wir verwenden demnach für jeden Baum zum einen eine veränderte Menge an Beobachtungen und zusätzlich für die Knotenaufteilung auch nur eine zufällige Auswahl an features.



Eine Beobachtung wandert nun durch alle Bäume und erhält von jedem Baum eine Prognose. Um eine individuelle Klassenwahrscheinlichkeit für unsere Beobachtung zu erhalten, können wir einfach das Mittel über die Klassenwahrscheinlichkeiten aller Bäume bilden. Eine konkrete Klassenzugehörigkeit für eine Beobachtung ergibt sich als einfacher Mehrheitsentscheid über alle Bäume.

Durch die Aggregation so vieler ganz unterschiedlicher Bäume verbessern wir die Prognosegüte für unsere neuen Beobachtungen. Ein Nachteil ist, dass wir einen ganzen Wald nicht mehr so einfach nachvollziehen bzw. interpretieren können wie einen einzelnen Baum.

Einblendungen Bäume

Wir kennen jetzt Baum-basierte Verfahren der Klassifikation, die man sich glücklicherweise auch tatsächlich wie einen echten, nur umgedrehten Baum vorstellen kann. Sie bilden ganz intuitiv unsere eigene Entscheidungsfindung im Alltag ab. Häufig werden sie auch für medizinische Diagnosen oder auch für Prognosen von Bildungserfolgen eingesetzt.

Weiterführendes Material

Fachbücher:

Guter Einstieg ins Thema, anschaulich erläutert, keine Formeln oder tiefergehende methodische Erläuterungen:

Aberham, J., & Kossen, J. (2019). Klassifikation - Schubladendenken. In K. Kersting, C. Lampert, & C. Rothkopf (Hrsg.), *Wie Maschinen lernen* (S. 45-52). Springer, Wiesbaden.

Kossen, J., Müller, M. E., & Ruckriegel, M. (2019). Entscheidungsbäume – Der Eisberg schwimmt nicht weit vorm Schiff. In K. Kersting, C. Lampert, & C. Rothkopf (Hrsg.), *Wie Maschinen lernen* (S. 111-118). Springer, Wiesbaden.

Auch wenn dieses Buch mit R anstatt Python arbeitet, anschauliche Erklärung der Methoden, tiefergehende methodische Erläuterungen:

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

- Chap. 5: Divide and Conquer – Classification Using Decision Trees and Rules
- Chap. 11: Improving Model Performance

Klassisches Werk für Statistisches/Maschinelles Lernen, tiefergehende methodische Erläuterungen:

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning* (2. Auflage). Springer.

- Chap. 8: Tree-Based Methods

Videos/Kurse:

Kurzer Einstieg ins Thema, anschaulich erläutert:

So lernen Maschinen: #4 Überwachtes Lernen - Klassifikation.
https://ki-campus.org/videos/solernenmaschinen

Etwas weitergehender Einstieg ins Thema, anschaulich erläutert:

AMALEA - Angewandte Machine Learning Algorithmen, Woche 3, Kap. 2: Willkommen in der Baumschule.
https://learn.ki-campus.org/courses/amalea-kit2021/items/5UUY37DVePDJ0QULOETks4

Disclaimer

Transkript zu dem Video „Woche 08 Theorie: Bäume und Wälder“, Katja Theune.