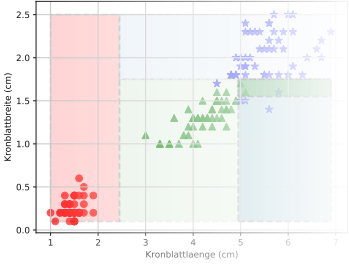
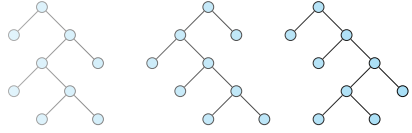
Ensemble Learning und Random Forest

Prof. Dr. Jörg Frochte

Maschinelles Lernen





Ensemble Learning

- Den Ansatz, mehrere Lerner zusammenzuschalten, nennt man **Ensemble Learning**.
- Ziel: Mehrere (schwache) Lerner zu kombinieren, um einen stärken Lerner zu bilden.
- Grob gibt es im Ensemble Learning zwei Ansätze: **Boosting** und **Bagging**.
- Ein typisches Boosting-Verfahren ist Gradient-Boosting, welches wir später behandeln.
- Der Random Forest den wir nun betrachten gehört zu der Gruppe der Bagging-Verfahren.
- Der Ausdruck *Bagging* setzt sich zusammen aus der englischen Beschreibung *Bootstrap Aggregating*.
- Das *Bootstrapping* in *Bootstrap Aggregating* geht auf Begriffe der Statistik und nicht der Informatik zurück.
- In der Statistik ist *Bootstrapping* eine Methode des Resampling von Daten mit **Zurücklegen**.

Zurücklegen in der Statistik

- Um zu verstehen bzw. sich zu erinnern, was *Zurücklegen* hier bedeutet, stellen wir uns eine Urne mit vier roten und zwei schwarzen Kugeln vor: 
- Wenn man nun blind in die Urne greift, um eine Kugel zu ziehen, hat man eine Wahrscheinlichkeit von $2/3$, eine rote Kugel zu ziehen: 
- Nehmen wir an, wir haben eine rote Kugel gezogen. Wenn wir diese jetzt nach dem Zug beiseite legen, dann steigt die Wahrscheinlichkeit eine schwarze Kugel zu ziehen, während es unwahrscheinlicher wird eine rote zu ziehen. 
- Legen wir die gezogene Kugel hingegen zurück, so bleibt es bei $2/3$ zu $1/3$. 
- In diesem Sinne ist *Zurücklegen* beim Bilden von Mengen im Rahmen des Baggings bzw. in der Statistik gemeint.

Bagging und Subagging

- Wie steht dies nun im Zusammenhang mit dem *Ensemble Learning*?
- Nehmen wir an, wir arbeiten mit einem Trainingsset D, welches l Datensätze enthält.
- Mittels Bagging werden nun N Mengen $\tilde{D}_i$ generiert, wobei jede die Größe $\tilde{l}_i \leq l$ besitzt.
- Die Probenentnahme aus D geschieht dabei mit gleicher Wahrscheinlichkeit für jede Probe und mit Zurücklegen. $\Rightarrow$ Datensätze können in $\tilde{D}_i$ doppelt auftauchen.
- Eben diese Art von Datenzusammenstellung wird als **Bootstrap-Sample** bezeichnet.
- Der Fall $\tilde{l}_i < l$ wird auch als **Subagging** bezeichnet.
- Jeweils eine der N Trainingsmengen wird nun jeweils einem Lerner (hier CART) zum Training zugeführt.
- Durch diesen Ansatz erzeugt man Lerner, die sich leicht unterschiedlich auf demselben Problem verhalten, was wir ja wollen.
- Bei der Auswertung werden deren Ergebnisse gemittelt (bei Regression) bzw. ein Mehrheitsentscheid durchgeführt (bei Klassifizierung).

Random Forest

- Die unter dem Namen **Random Forest** bekannte Methode wurde von Leo Breiman in im Jahr 2001 publiziert.
- Wie unterschiedet sich nun der *Random Forest* vom reinen Bagging mit verschiedenen CART-Bäumen?
- In gewisser Weise kann man sagen, dass das reine Bagging oft nicht genug Variationen von Bäumen erzeugt. Folglich geht der Ansatz des *Random Forest* über das Zuweisen von durch Bagging erzeugten Trainingsmengen hinaus und fügt weitere Zufallselemente dem Algorithmus zur Erzeugung des Entscheidungsbaums hinzu.
- Das Zufallselement, was neu eingebracht wird, ist, dass der Algorithmus nicht mehr unter allen Merkmalen auswählen darf, sondern nur noch auf einer zufällig gewählten Teilmenge. Diese Teilmenge wird bei jeder Entscheidung im Baum neu zusammengestellt.

Pseudocode für den Random Forest

Sei m die Anzahl an Merkmalen.
Wähle die Anzahl N von Bäumen, welche der Random Forest umfassen soll

1: Sei m die Anzahl an Merkmalen
2: Wähle die Anzahl N von Bäumen, welche der Random Forest umfassen soll
3: **for** $i = 0$ **to** N **do**
4: Erzeuge neue Bootstrap-Trainingsmenge D_i
5: Beim Lernen des Entscheidungsbaums wähle zufällig an jedem Knoten $\tilde{m} \leq m$ Merkmale aus, die zur Aufteilung verwendet werden dürfen
6: Füge den so auf D_i trainieren CART dem Random Forest hinzu
7: **end for**

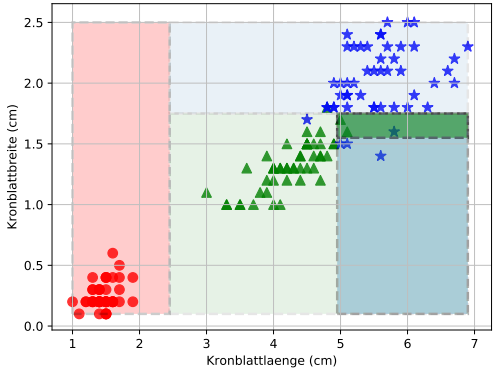
Wahl von $\tilde{m}$

In der Literatur wird als recht robuster Ansatz oft die Wahl $\tilde{m} = \text{int}(\sqrt{m})$ oder $\tilde{m} = \text{int}(\log_2(m))$ angegeben. Das ist aber oft eher an Klassifikationsproblemen orientiert. Bei der Regressionen hingegen wird als Default oft $\tilde{m} = m$ genutzt.

- Nun bleibt noch die Frage, wie groß unser Wald sein soll, also die Wahl von N.
  1. Man kann die Tatsache nutzen, dass der Random Forest für viele Aspekte eine Art eingebaute Validierungsmenge besitzt. Jeder Baum enthält nur einen Teil der Trainingsdaten, sodass ein anderer Teil für das Parameter-Tuning zur Verfügung steht. Das erlaubt uns den **Out-of-Bag-Error** zu berechnen. Hierzu nehmen wir ein $x_i \in D$ und lassen dies von allen Bäumen auswerten, die x_i nicht in ihren Trainingsdaten hatten. Das tun wir für alle Datensätze und bilden dann den mittleren Vorhersagefehler. Um nun N zu bestimmen, fügen wir solange Bäume dem Wald hinzu, wie dies der Qualität bzgl. des Out-Of-Bag-Errors zuträglich ist.
  2. Verschiedene N ausprobieren und testen wie gut das Modell geworden ist.
  3. Als letztes können wir natürlich auch einfach auf Grund unserer **Erfahrung** einen Wert für N festlegen. Oft ist das nicht der schlechteste Ansatz.
- Eine weitere gute Eigenschaft des Random Forests ist, dass die Notwendigkeit zum Pruning bzgl. der Qualität der Ergebnisse geringer wird.

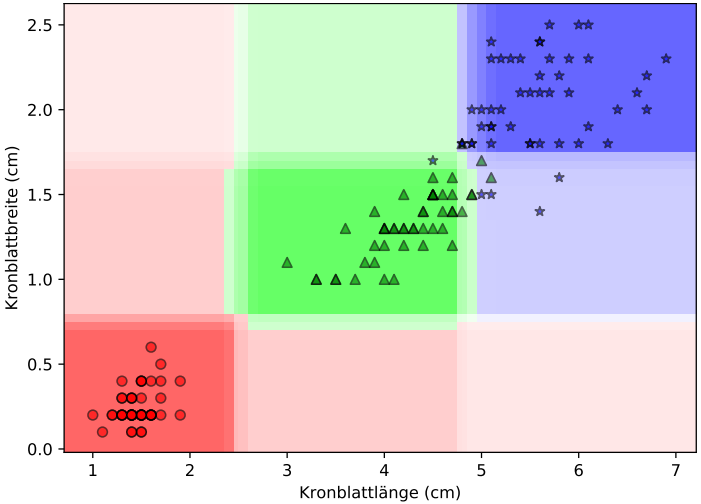
Realistischere Wahrscheinlichkeiten

- Der erste Schnitt könnte auch entlang der Kronenblattbreite (y-Achse) erfolgen und würde ebenfalls die rote Gruppe separieren.
- In einem Random Forest werden beide Schnitte vorkommen, da die Reihenfolge der Features immer zufällig ist bzw. auch nicht immer alle zur Verfügung stehen.
- Das bedeutet, die Menge unterschiedlicher Bäume hilft uns das Problem zu adressieren, das Merkmale implementierungsabhängig als erstes ausgewählt werden.



- Bei einem einzelnen Baum würde der gesamte rote Bereich immer die rote Klasse mit einer 100%igen Wahrscheinlichkeit wählen. Das ist bei einem Random Forest nicht so.

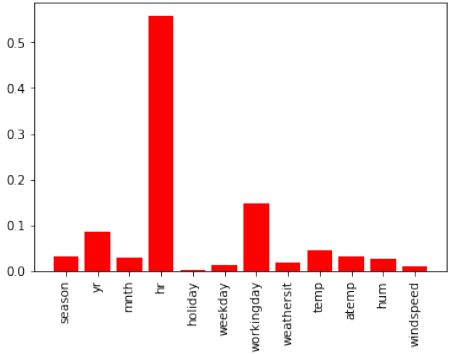
Ergebnis aller Bäume



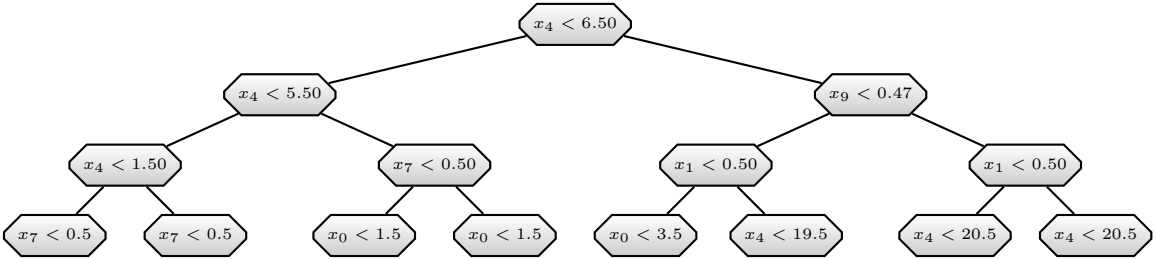
- Wenn alle Bäume verwendet werden lassen sich Unsicherheiten besser Greifen.
- Links dargestellt sind die Gini-Impurities der Ergebnisse eines Punktes. Bei N Bäumen gibt es N Ergebnisse welcher Klasse der Punkt angehört.
- Alternativ lassen sich die Klassenwahrscheinlichkeiten der Bäume mitteln.

Random Forest zur Feature Analyse

- Bei einer geeigneten Implementierten wie z. B. in *scikit-learn* kann man den Random Forest auch zur Feature Analyse nutzen
- Wenn man auswertet, wie oft und an welcher Stelle ein Merkmal in den unterschiedlichen Bäumen ausgewählt wurde, kann man deren Bedeutung bewerten
- Welchen Vorteil gibt es bzgl. der Bedeutung eines Merkmals, wenn man den Random Forest einsetzt statt dem einzelnen CART-Baum?
- Egal ob es um einen einzelnen Baum geht oder einen ganzen Wald. Wenn wir uns nur auf „oben im Baum“ als Kriterium verlassen würden, wäre dies zu kurz gegriffen.
- Können Sie sich vorstellen warum?



Ausschnitt eines Baums



- Nehmen wir einmal die zweite Ebene des bekannten Baumes.
- Es ist keinesfalls klar, dass der Baum als Ganzes balanciert ist oder dass genauso viele Trainingsbeispiele rechts wie links angefertigt werden.
- Das bedeutet, dass die Knoten unabhängig von ihrer Lage im Baum unterschiedlich viel dazu beitragen das Fehlermaß zu verringern bzw. die Impurity zu senken.

- Die Standardmethode zur Berechnung der *feature importance* ist die Reduktion des Fehlermaßes, also z.B. Gini-Bedeutung oder MSE
- Die Grundidee ist, bei jeder Teilung im Baum die Verbesserung bzgl. dieses Maßes dem Merkmal zuzuschreiben, welches verwendet wurde.
- Abschließend werden die Ergebnisse über alle Bäume im Wald gemittelt.
- Ein Ansatz um die Bedeutung eines Knoten zu berechnen lautet daher:

$$\mathrm{ni}_j = w_j \cdot C_j - w_{\text{links}}(j) \cdot C_{\text{links}}(j) - w_{\text{rechts}}(j) \cdot C_{\text{rechts}}(j)$$

- Es geht zunächst hierbei um das Merkmal i welches zum Splitting verwendet wird.
- $w(j)$ ist dabei die Anzahl der Elemente im Knoten j und $c(j)$ ist das Qualitätsmaß an diesem Knoten.
- Links und Rechts meint dabei die Kinder-Knoten vom Knoten j aus betrachtet.

- Nun addieren wir die Bedeutung eines Merkmals auf für den ganzen Baum

$$\tilde{\text{ni}}_i = \sum_{j \text{ mit } i \text{ Entscheidungskriterium}} \text{ni}_j$$

- Diese Werte sind nicht normiert, sodass wir schwer die Bedeutung erfassen können. Daher normieren wir es in dem Baum auf 1 in dem wir die gewichtete Verbesserung an allen Knoten addieren

$$\bar{\text{ni}} = \sum_{\text{alle Knoten}} \text{ni}_j$$

- Abschließend normieren wir die Bedeutung jeweils

$$\text{ni}_i = \frac{\tilde{\text{ni}}_i}{\bar{\text{ni}}}$$

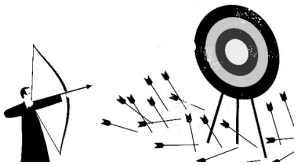
- Diese Werte pro Baum werden abschließend über den ganzen Wald gemittelt und erzeugen so ein gutes Maß für die Bedeutung eines Merkmals.
- Ein Vorteil ist, dass die Arbeit für die *feature importance* quasi automatisch eh beim erstellen der Bäume bzw. des Random Forest anfällt.
- Aussage von Breiman & Cutler als Erfinder des Random Forest

  Adding up the Gini decreases for each individual variable over all trees in the forest gives a ***fast variable*** *importance that is often* ***very consistent*** *with the permutation importance measure.*

- *feature importance* ist also eine leicht zu berechnende Größe, die man jedoch trotzdem immer kritisch hinterfragen muss[1].
- Ein bekanntes Problem ist durch den Algorithmus selber es eine Tendenz gibt unterschiedliche Typen von Merkmalen zu bevorzugen.

[1] u. a. Altmann, A. et al 2010 *Permutation importance: a corrected feature importance measure*

- Nehmen wir an zu einem Modell gehören sowohl kontinuierliche Merkmale, als auch solche mit vielen kardinalen Kategorien und solche mit wenigen.
- In jedem Knoten sucht der Algorithmus nun den besten Schnittpunkt.



- Kontinuierliche Variablen haben viele mögliche Schnittpunkte und unterliegen daher einem statistischen Problem, welches als *Multiples Testen* bekannt ist.

Multiples Testen

Jeder Test auf einen Zusammenhang zwischen X und Y kann trotz Signifikanz falsch sein. Typische Wahrscheinlichkeiten, die für einen einzelnen Test akzeptiert werden, liegen bei 1% bis 5%. Das bedeutet, wenn ein Test durchgeführt wird, ist die Wahrscheinlichkeit einen Zusammenhang zu sehen, der nicht da ist, z. B. bei 2%. Führen wir aber 100 Tests durch erhalten wir

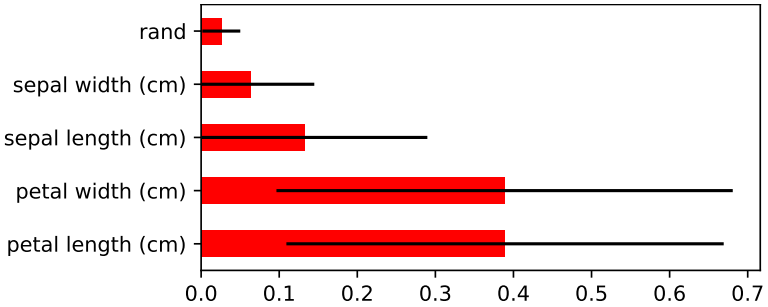
$$1 - 0.98^{100} \approx 0.867$$

als Sicherheit bzw. über 13% Wahrscheinlichkeit etwas zu sehen, was nicht da ist.

- Durch diesen Effekt wird die Feature Importance von Merkmalen, die viele mögliche Schnittpunkte beinhalten aus statistischen Gründen tendenziell überbewertet gegenüber solchen mit nur wenigen Kategorien oder im Extremfall binären Merkmalen.
- Ein weiteres Problem ist, dass die Feature Importances verschiedener Merkmale nichts darüber aussagen, ob diese untereinander korrelieren bzw. abhängig sind.
- Beispielsweise werden ggf. die Variabel Gewicht, Länge und Breite eines Schiffes für eine Aussage als besonders wichtig eingeschätzt.

- Diese Werte sind jedoch eher gemeinsam ein Merkmal für *Schiffsgröße*. Wir werden später darauf eingehen.
- Die Merkmale sind allein betrachtet wichtig, würden aber zusammen nicht unbedingt die optimale Basis für unseren Merkmalsraum darstellen.

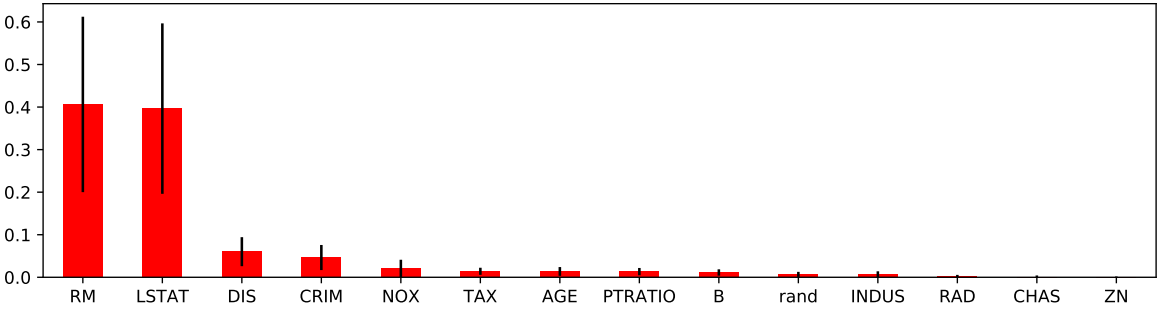
Eigene Aufnahme Wellington 2017



- Um den Effekt zu zeigen, wenden wir diese Technik auf das *Iris flower data set* an.
- Dabei fügen wir als weiteres Merkmal einfach zufällige Zahlen zwischen 0 und 1 ein.
- Natürlich hat dieses Merkmal keine Relevanz für die Frage um welche Blumenart es sich handelt.
- Wie man sieht, stellt auch der Random Forest dies so fest.

Boston housing price dataset (1978)

Name	Beschreibung
CRIM	Pro-Kopf-Kriminalitätsrate
ZN	Anteil an Wohnbauland für Grundstücke über 25.000 qm
INDUS	Anteil an nicht zum Einzelhandel gehörenden Gewerbeflächen pro Stadt
CHAS	Charles River Dummy-Variable (1, wenn die Fläche an den Fluss grenzt; sonst 0)
NOX	Stickoxidkonzentration
RM	durchschnittliche Zimmeranzahl pro Wohnung
AGE	Anteil selbst genutzter Einheiten, die vor 1940 gebaut wurden
DIS	Gewichtete Entfernungen zu fünf Bostoner Arbeitsämtern
RAD	Index der Zugänglichkeit zu den Autobahnen
TAX	Grundsteuersatz pro 10.000 Dollar
PTRATIO	Schüler-Lehrer-Verhältnis
B	$1000 \cdot (Bk - 0.63)^2$, wobei Bk der Anteil der farbigen Bevölkerung angibt
LSTAT	% der Bevölkerung mit niedrigem Status



- Im Fall des *Bosten House Price Data Set* überholt der Zufall reale Merkmale.
- Er erscheint bedeutender als
  - INDUS: proportion of non-retail business acres per town
  - RAD: index of accessibility to radial highways
  - etc.
- Das bedeutet, dass wir i. W. nur die ersten Aussagen bis TAX oder AGE wirklich berücksichtigen sollten.