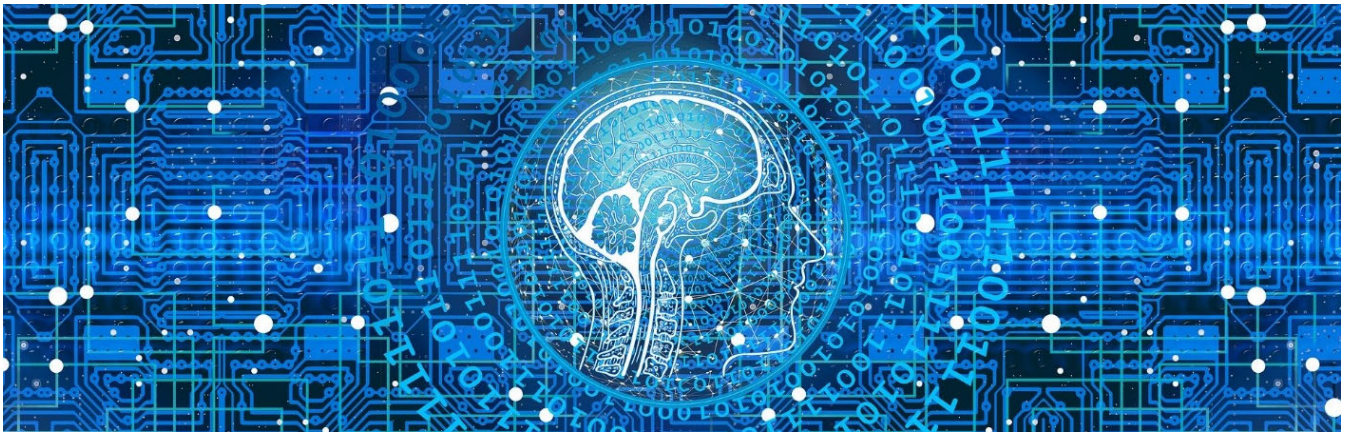
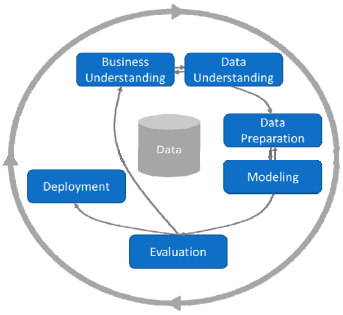
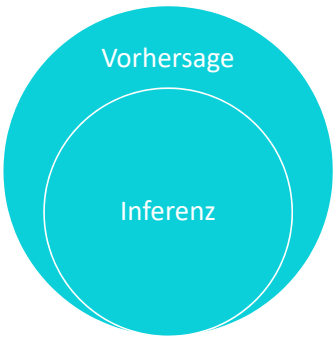
Data Science
6. Teil – State of the Art Maschinelle Lernverfahren

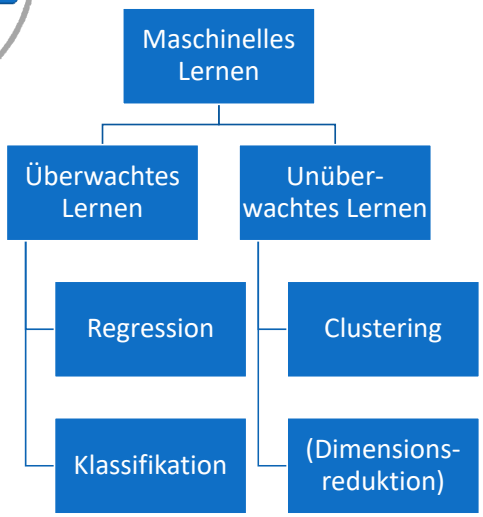
Vorlesung an der DHBW Stuttgart, Prof. Dr. Monika Kochanowski

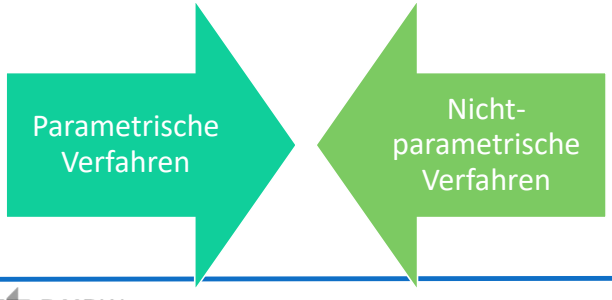


Wiederholung



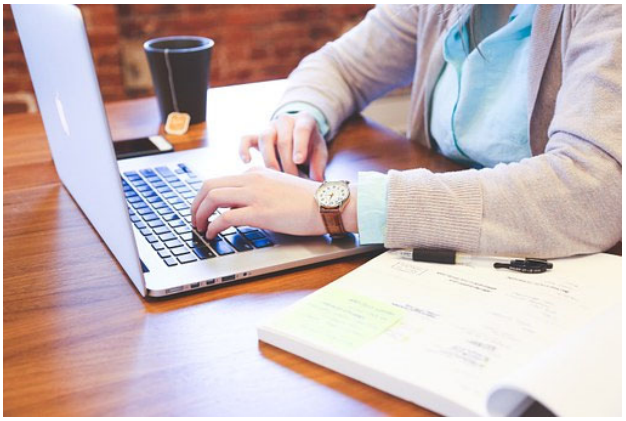




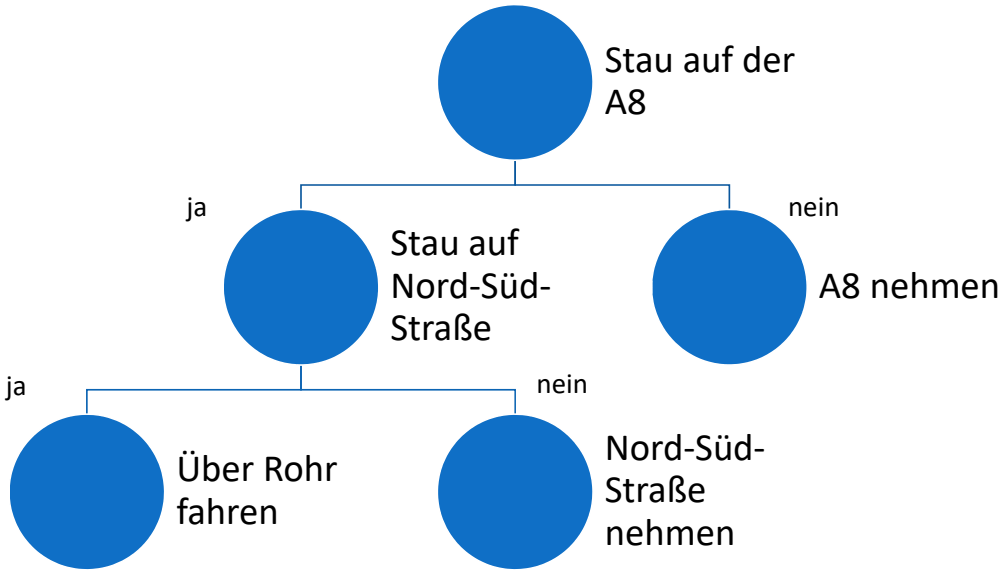


Inhalte der heutigen Vorlesung

- Wiederholung
- Lernverfahren
  - Bäume
  - SVMs
- Ensemble Learning
  - Bagging
  - Boosting



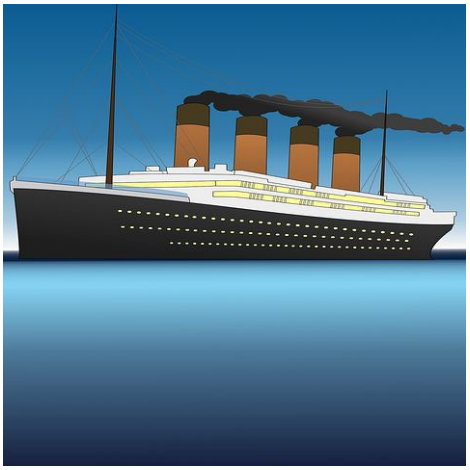
Entscheidungsbäume
Grundlage vieler aktueller Algorithmen



Diskussion

Entscheidungsbäume
Messung der Unreinheit für die Erstellung von Entscheidungsbäumen

- **Entropie**
  - Aus der Informationstheorie
  - $\Phi_m = -\sum_{i=1}^{K} p_m^i \log_2 p_m^i$
  - Zweiklassenproblem: $\Phi(p, 1-p) = -p \log_2 p - (1-p) \log_2 (1-p)$
  - **Beispielberechnung** anhand des Titanic-Beispiels für Klasse und Geschlecht
  - **Information Gain**: Differenz aus Entropie des übergeordneten Knoten und gewichteten Entropie der untergeordneten Knoten
- Alternative: **Gini-Index**
  - $\Phi(p, 1-p) = 2p(1-p)$
- Alternative: **Fehlklassifikationsfehler**
  - $\Phi(p, 1-p) = 1 - \max(p, 1-p)$
- »Die Forschung hat gezeigt, dass es keine signifikanten Unterschiede […] gibt.« [Alpaydin 2008]



Titanic: mehrere Datensätze verfügbar, z. B. Kaggle, hier noch Infos:
http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/titanic3.xls
https://bigml.com/user/czuriaga/gallery/model/52c0cf160c0b5e6fcb000345/tree

Vorwegnahme: Klassifikationsbewertung
Wahrheitsmatrix (binäre Klassifikation)

Macht Beispiele einfacher..

		Wirklichkeit	Wirklichkeit
	Alle	Ist wirklich erkrankt	Ist wirklich gesund
Vorhersage	Test sagt erkrankt	Richtig-positiv	Falsch-positiv
Vorhersage	Test sagt gesund	Falsch-negativ	Richtig-negativ

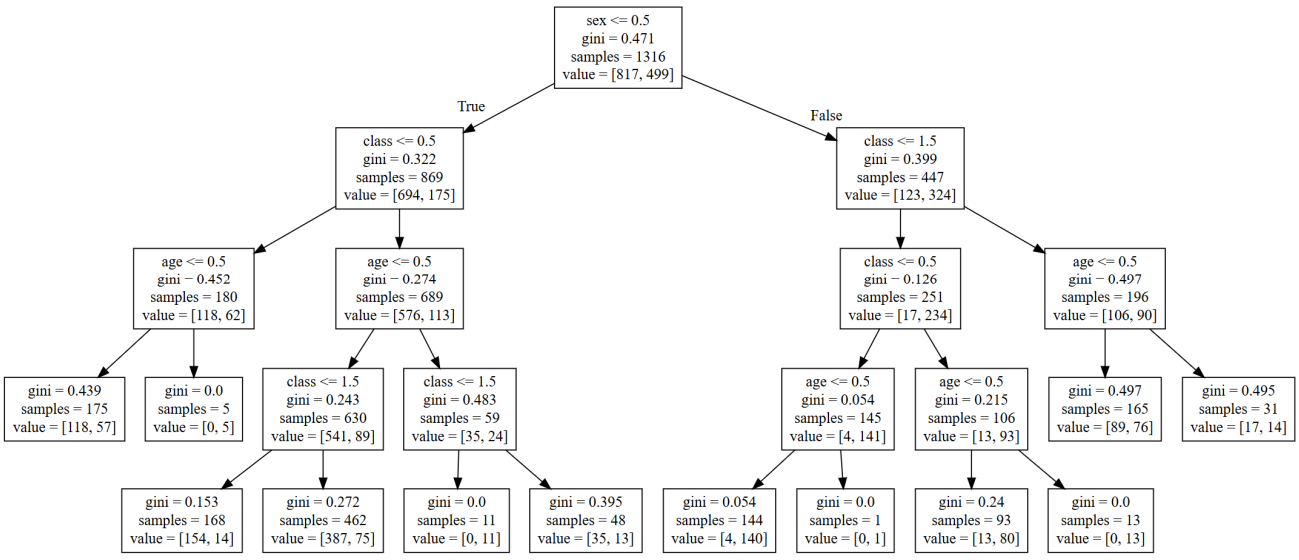
Entscheidungsbäume mit ID3-Algorithmus

Wichtig für gute Algorithmen wie Random Forest oder GBT

```
GeneriereBaum(X)
    If KnotenEntropie(X) < Schwellwert
        Erstelle Blatt mit Label der häufigsten Klasse in X
        Return
    i = Aufspaltungsattribut (X)
    Für jede Verzweigung von x_i
        Finde X_i, das in Verzweigung liegt
        GeneriereBaum(X_i)
AufspaltungsAttribut(X)
    MinimaleEntfernung = MAX
    Für alle Attribute i = 1, .., d
        Teile X in X_1, X_2
        e = AufspaltungsEntropie (X_1, X_2)
        If e < MinimaleEntfernung; MinimaleEntfernung = e; bestf = i
    Return best
```

Quelle des Algorithmus: [Alpaydin 2008]**; Hinweise zum Code: [Alpaydin 2008], [James et al. 2013], Lösung: [Grues 2016], vereinfacht

Interpretation der Visualisierung

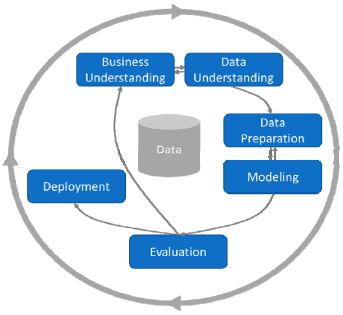


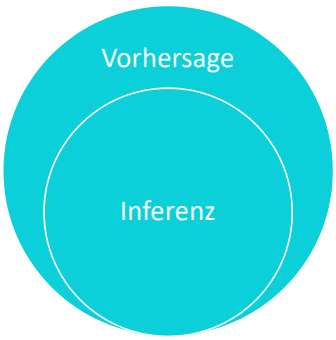
Entscheidungsbäume
Konfigurationsmöglichkeiten und Einsatzmöglichkeiten

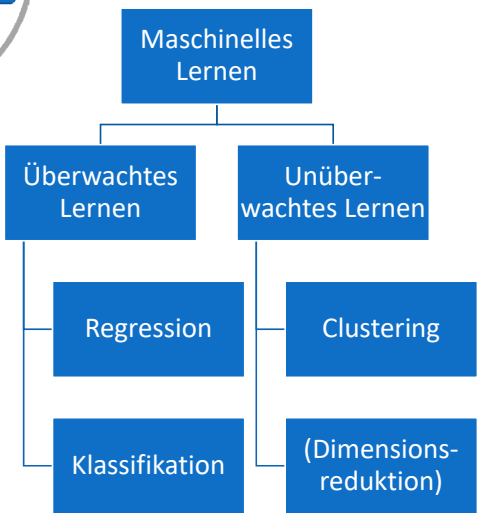
- Generell sind Entscheidungsbäume auch für **Regression** verwendbar
  - Oder für Klassifikation unter Berücksichtigung numerischer Attribute
  - Gut beschrieben in der bereits genannten Literatur
- ID3 Algorithmus mit greedy-Ansatz hat wesentliche Nachteile
  - **Welche?**
- **Pruning**
  - Dt. Beschneiden / Stutzen
  - Prepruning (in der Konstruktion)
  - Postpruning (nach der Konstruktion)

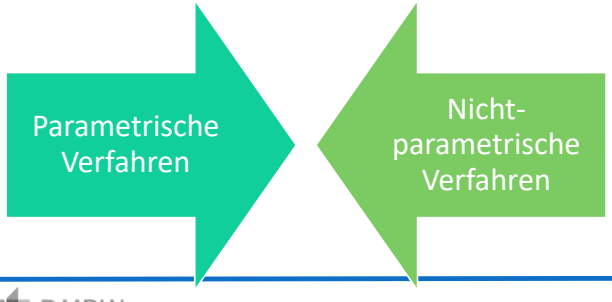
- http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Wiederholung

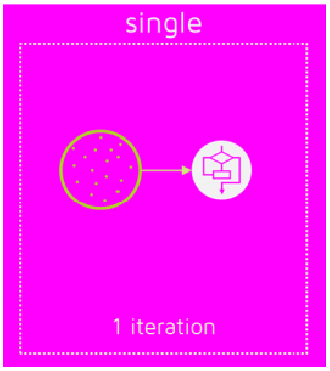




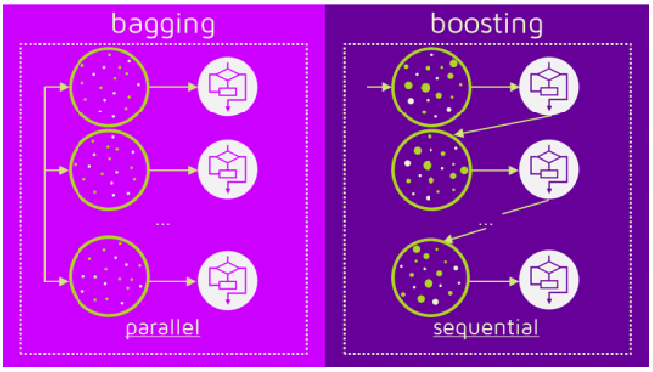




Bootstrap und Ensemble Learning: Bagging und Boosting

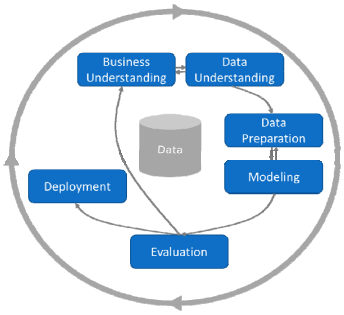


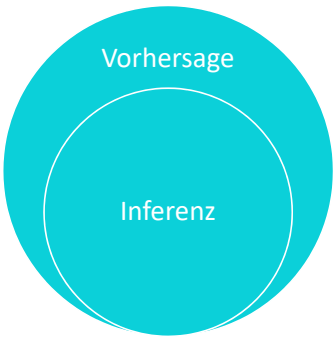
- **Bootstrap**
  - Bootstrap: Um eine unbekannte Varianz abzuschätzen, werden aus Testsets neue Testsets generiert
  - Methode: Ziehen mit Zurücklegen aus den Testdaten
  - Nebeneffekt: Damit kann man Modelle variieren
- **Bagging**: steht für „**B**ootstrap **Agg**regation“
  - Aggregierung über unter Hilfe von Bootstrap erzeugte Modelle
- **Boosting**: Residuen nutzen
  - Trainiere nicht auf den Wert, sondern auf den „Rest“, der nicht gut funktioniert hat
  - Verbessere kontinuierlich
  - Gewichtung von Teilmengen beim Lernen möglich

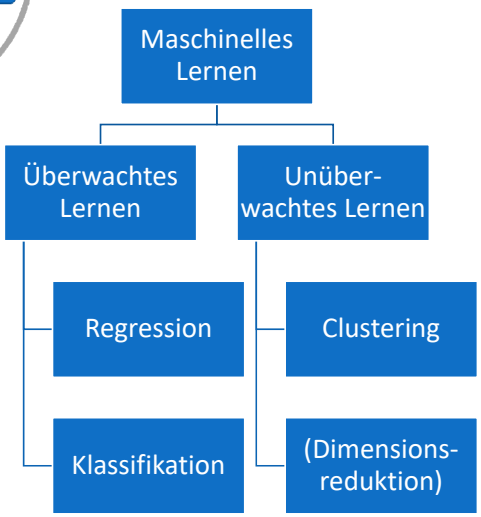


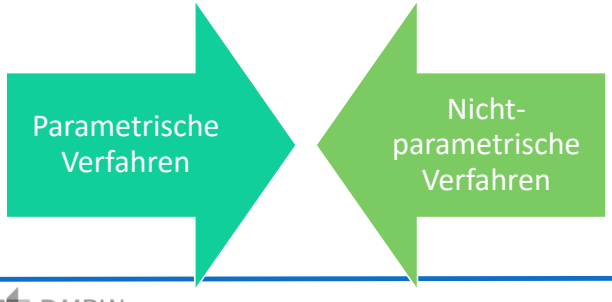
https://quantdare.com/what-is-the-difference-between-bagging-and-boosting/

Wiederholung

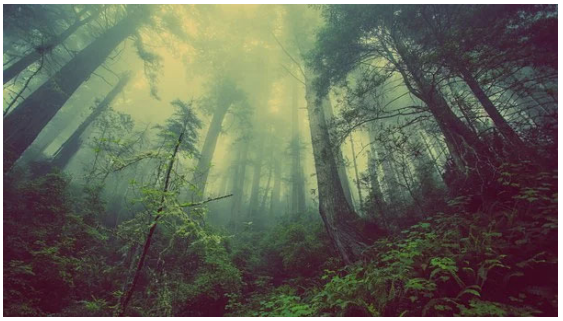








Random Forests
Mehrere unkorrelierte Entscheidungsbäume

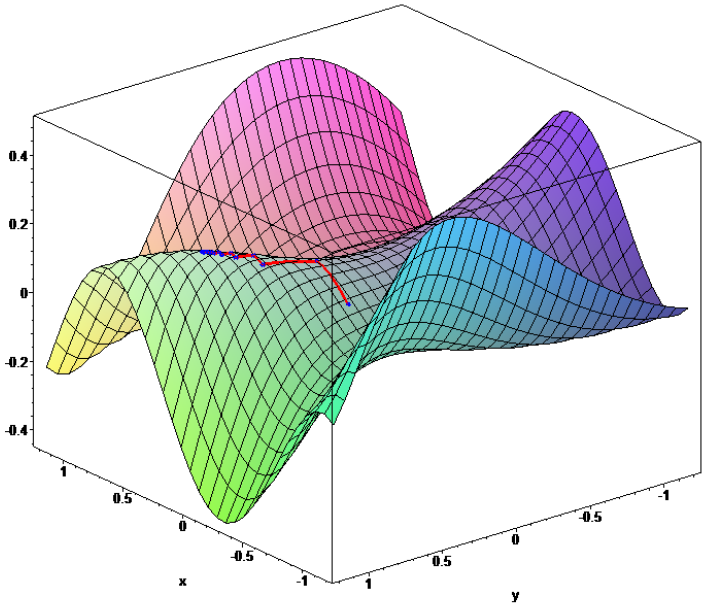


- Exkurs: **Bootstrap** und Bootstrap Aggregation (**Bagging**)
  - Bootstrap: Um eine unbekannte Varianz abzuschätzen, werden aus Testsets neue Testsets generiert
  - Methode: Ziehen mit Zurücklegen aus den Testdaten
  - Für Entscheidungsbäume: Erstellung einer Zahl B von Bäumen
  - Abstimmung mehrerer Bäume mit Durchschnitt / Mehrheitsvotum (Ensemble Learning)
- **Random Forest** baut *unkorrelierte* Entscheidungsbäume
  - Auswahl von $m \approx \sqrt{p}$ Attributen für einen Knoten im Entscheidungsbaum (z. B. 4 aus 13)
  - Zufällig – ***nur*** aus diesen Attributen wird gewählt
  - Baue viele Bäume (in dem Beispiel mit den 4 Attributen: 400)
  - Dann: Durchschnitt / Mehrheitsvotum

Exkurs: Gradientenabstieg

- „Minimierungsverfahren für eine reell-wertige, differenzierbare Funktion“
- Abstiegsrichtung, steilster Abstieg
- Schrittweite, exakt/inexakt

- **Welche Anwendungen fallen Ihnen für Gradientenabstieg ein?**



Bildquelle: Wikipedia, Bild ist public domain

Gradient Tree Boosting
Ein State of the Art Algorithmus

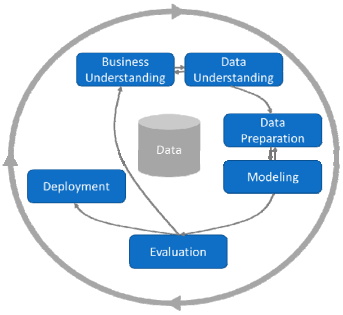
- Woran erkennen die Autoren dass der Tree 1 überangepasst ist?
- Warum ist der Tree 2 das NICHT ist? (WICHTIG!)
- Was ist die Kernidee in 3 Schritten in Draft 1?
  Haben wir bisher was ähnliches gemacht?
- Was wird in Draft 2 zu Draft 1 angepasst? Warum?
- Was bedeutet Boosting (im Gegensatz zu Bagging) generell?
- Wozu wird Gradientenabstieg (Gradient Descent) in diesem Zusammenhang genutzt?
- Warum wird noch Sampling eingeführt?
- **Übungsaufgabe (freiwillig) mit** https://www.gormanalysis.com/blog/gradient-boosting-explained/
- **https://towardsdatascience.com/machine-learning-part-18-boosting-algorithms-gradient-boosting-in-python-ef5ae6965be4**

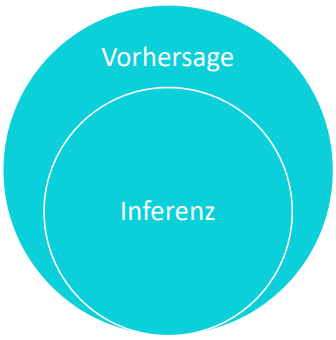
GBT – Zusammenfassung

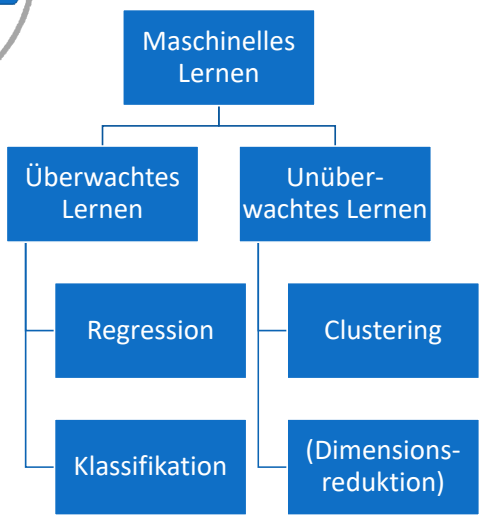
Funktioniert für eine ganze Reihe von Anwendungen extrem gut / am Besten	Overfitting-anfälliger als z. B. Random Forest
Sehr flexibel	Rechenaufwändig im Vergleich zu vielen anderen (sequentiell)
Kann mit NULL-Werten und kategorischen Werten umgehen	Hyperparameter-Tuning ist aufwendig

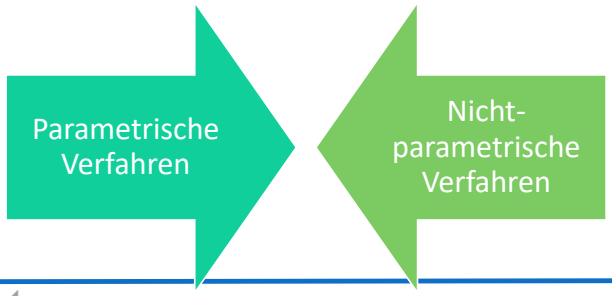
- Wichtige Hyperparameter
  - Angaben zur Höhe des Baumes
  - Anzahl / Teilmenge in Blättern
  - Anzahl der Features (analog Random Forests)
  - Einige weitere, die meistens recht gut voreingestellt sind
  - Quelle: https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gradient-boosting-gbm-python/

Wiederholung

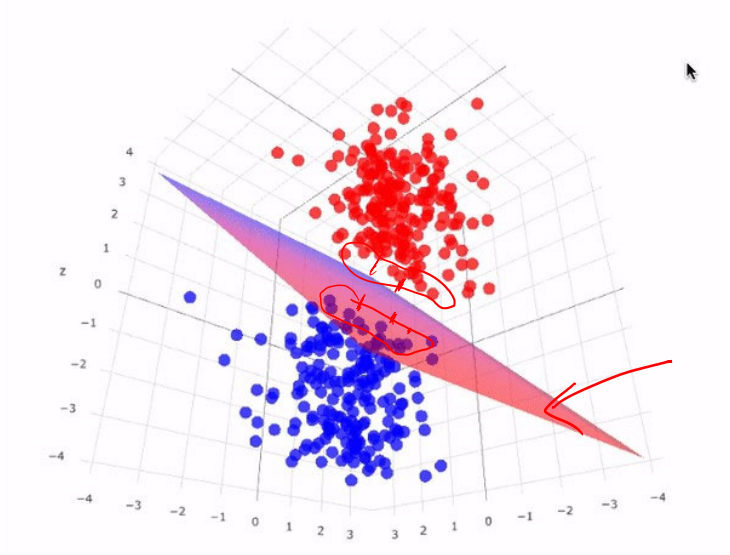








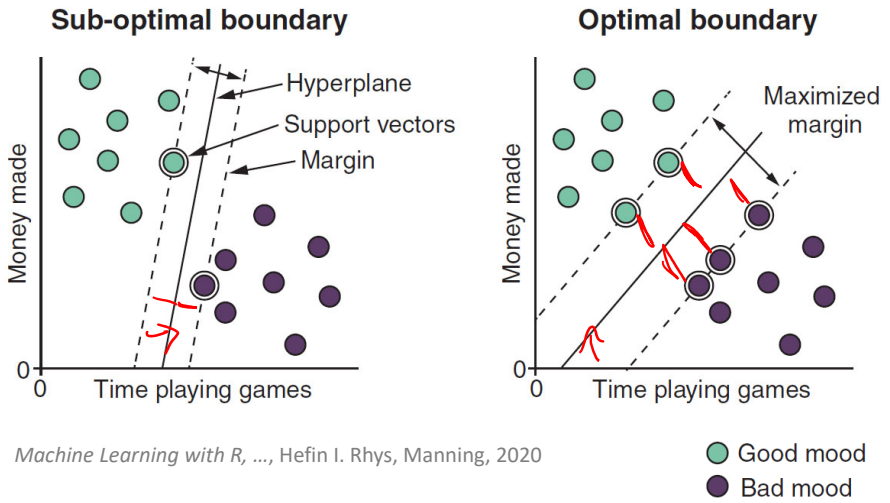
Support Vector Machine (SVM)



- Sehr gute Klassifikationsleistung für viele Anwendungsbereiche, aber oft nur schwer interpretierbar und damit mitunter nicht so einfach nachvollziehbar

Bildquelle: https://mc.ai/support-vector-machine-svm-algorithm-in-a-fun-easy-way/

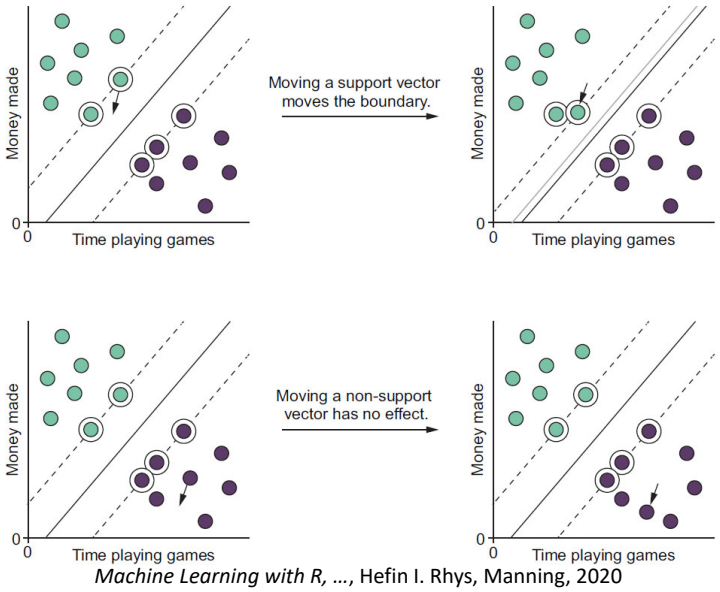
SVM Grundlagen



Machine Learning with R, …, Hefin I. Rhys, Manning, 2020

- Prinzipiell klassifiziert die SVM nur zwischen zwei verschiedenen Klassen
- Die SVM sucht eine optimal separierende Hyperebene im n-dimensionalen Raum der Prädikatoren, die die Ausprägungen der beiden Klassen trennt
- Die Frage ist, welche der möglichen Hyperebenen ist die beste?
- Die mit dem größten Abstand zu beiden Gruppen!
  - Die Hyperebene die am besten generalisiert und bei unbekannten Daten möglichst auch noch funktioniert

SVM – was ist ein Support Vector?

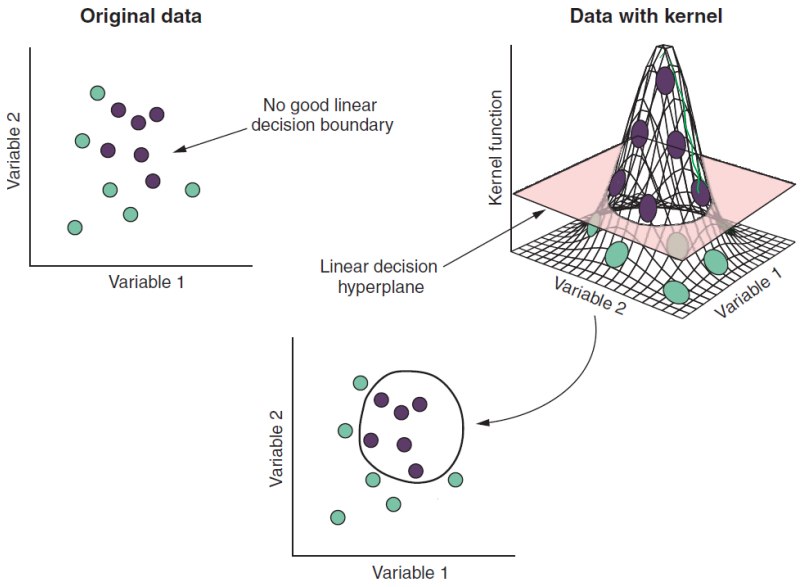


Machine Learning with R, …, Hefin I. Rhys, Manning, 2020

- Support Vector: Definieren die Lage der Hyperebene mit dem größten Margin – Support-Vektoren, das sie der „Support“ für Lage der gewählten Hyperebene sind
- Am Ende sind eigentlich nur sie relevant die relevanten Fälle der Trainingsmenge und man kann problemlos alle anderen Punkte aus den Trainingsdaten entfernen
- Die Mathematik hinter der Berechnung ist ein komplexes Optimierungsproblem.

- Hard Margin: separierbar
- Soft Margin: nicht komplett separierbar, penalty für Punkte auf der „falschen“ Seite

SVM – was ist ein Kernel?



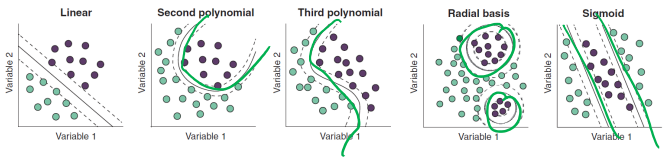
Machine Learning with R, …, Hefin I. Rhys, Manning, 2020

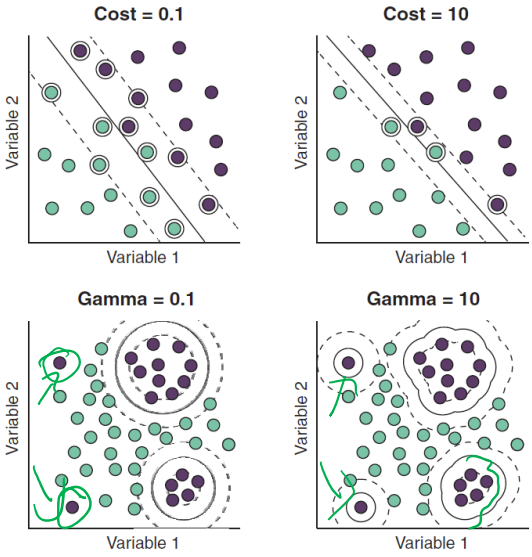
Die Wahl des Kernel der SVM ist ein eigener Hyperparameter, z.B.:

- **Linearer** Kernel (kein Kernel)
- **Polynomialer** Kernel
- Gauß'sche Radiale Basisfunktionen (**RBF**)
- **Sigmoider** Kernel

Wichtige Hyperparameter der SVM

- **kernel** für die Art des verwendeten Kernels
- **degree** für den Grad des polynomiellen Kernel, der damit die „Welligkeit" der Entscheidungsgrenze bestimmt → Kompromiss zwischen Over- und Underfit
- **cost** oder C für die Kontrolle wie „soft" oder „hard" der Margin sein soll
- **Gamma** kontrolliert den Einfluss einzelner Datenpunkte auf die Lage der Entscheidungsgrenze (mehr Einfluß → komplexere Grenzen)





Machine Learning with R, …, Hefin I. Rhys, Manning, 2020

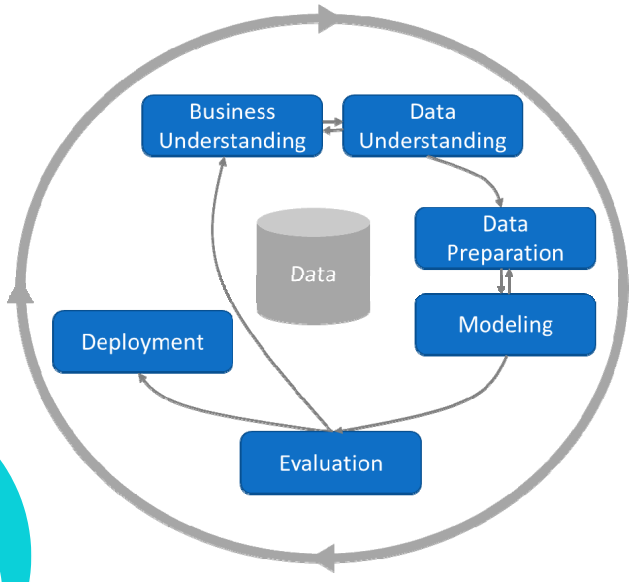
SVMs

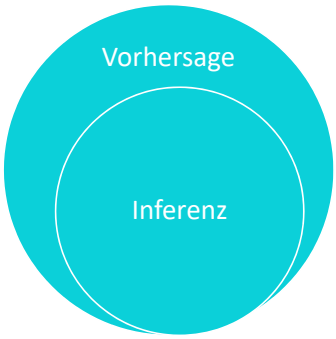
Kann sehr gut sehr komplexe nichtlineare Entscheidungsebenen lernen	Rechenaufwändig (nicht im Vergleich zu DL;)
Funktioniert für eine ganze Reihe von Anwendungen sehr gut	Hyperparameter-Tuning ist (wirklich) aufwendig
Setzt keinerlei Verteilung bezüglich der Prädiktoren voraus	Nur kontinuierliche Parameter
Kann auch Regression (scikit: SVR, support vector Regression)	

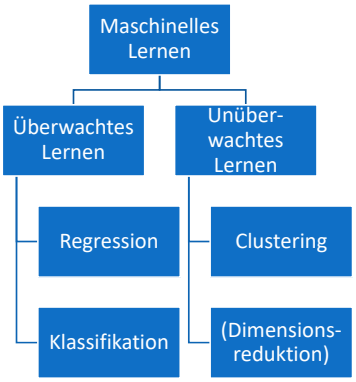
Übung

- Sagen Sie für den Pokémon-Datensatz den Angriffswert vorher.
- Nutzen Sie SVMs und GBTs.
- Verwenden Sie Encoder, wo notwendig.

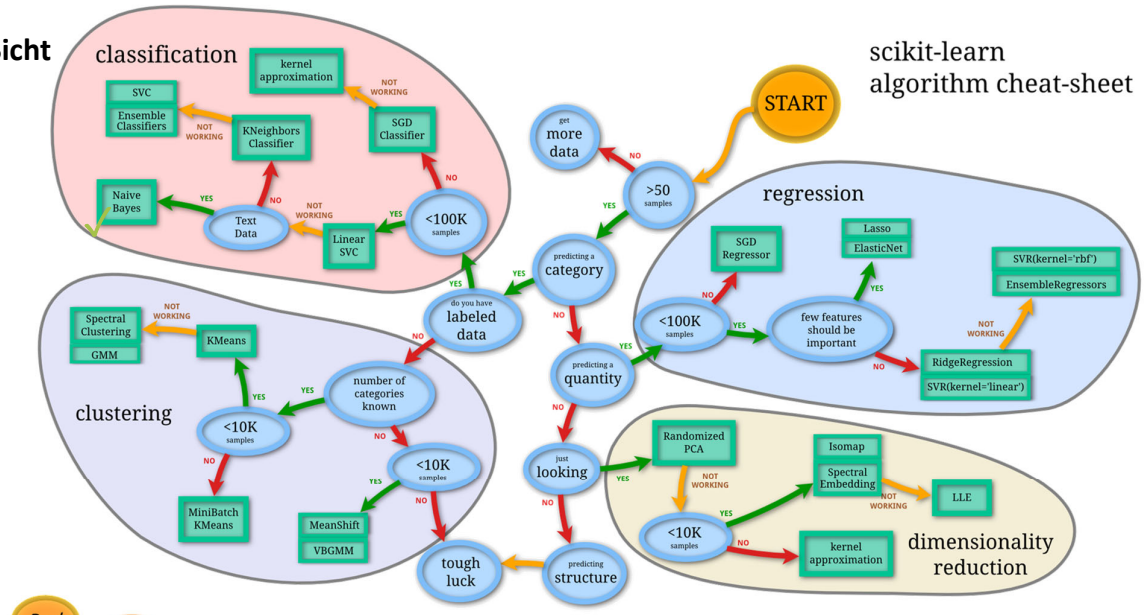
Wiederholung





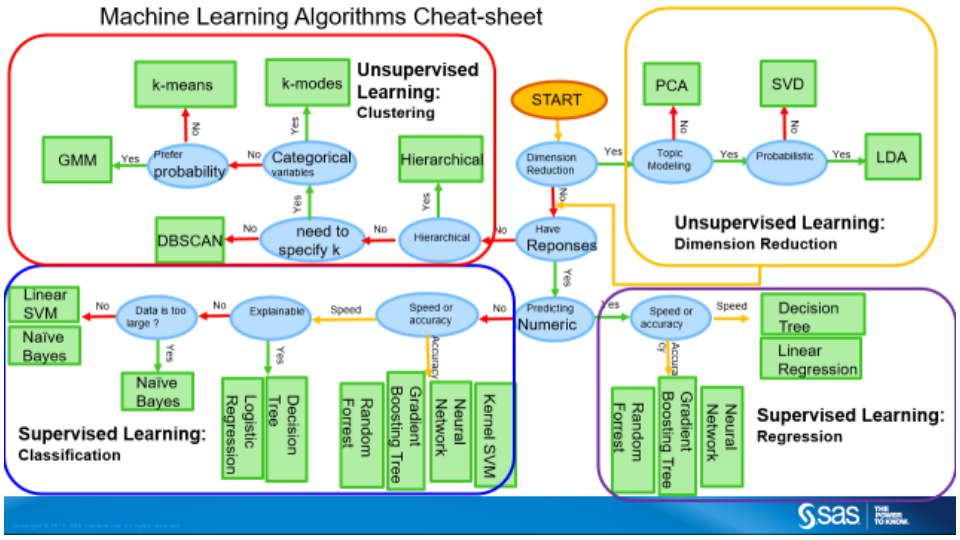


Stand
scikit-Sicht



Bildquelle: http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html
Link: https://scikit-learn.org/stable/supervised_learning.html (Hilfreich für Programmentwurf!)

Stand der Vorlesung
SAS Cheat Sheet Sicht



Bildquelle: https://whatsthebigdata.com/2017/05/02/types-of-machine-learning-algorithms-and-when-to-use-them/, am 10.02.2018

Literaturliste

- [James et al. 2013] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani: An introduction to statistical learning
  - Favorit: Sehr gut gemachte Einführung, jedoch Beispiele in R, verständlich mit Mathematik, als pdf frei erhältlich
- [Hastie et al. 2008] Trevor Hastie, Robert Tibshirani, Jerome Friedman: The elements of statistical learning
  - DIE Referenz, für Mathematiker geschrieben, als pdf frei erhältlich
- [O'Neil and Schutt 2013] Cathy O'Neil and Rachel Schutt: Doing Data Science
  - Spannend zu lesen, teilweise Erfahrungsberichte (durch Drittautoren)
- [Mueller and Guido 2017] Andreas C. Müller & Sasha Guido: An Introduction to Machine Learning with Python
  - Interessant da Python 3 tatsächlich genutzt wird für die Einführung inklusive der üblichen Bibliotheken
- [Grues 2016] Joel Grues (übersetzt von Kristian Rother): Einführung in Data Science
  - Auf deutsch gut übersetzt, nutzt Python für grundlegendes Verständnis ohne die üblichen Bibliotheken, extrem leicht lesbar
- [Alpaydin 2008]: Ethem Alpaydin (übersetzt von Simone linke): Maschinelles Lernen
  - Auf deutsch gut übersetzt, relativ viel Mathematik, in Deutschland scheint das weit verbreitet zu sein
- [Bruce et al. 2020]: Peter Bruce, Andrew Bruce, Peter Gedeck: Practical Statistics for Data Scientists
  - Das einzig wahre Statistikbuch was keines ist
- [Reinhart 2016]: Alex Reinhart (übersetzt von Knut Lorenzen): Statistics done wrong
  - Bevor man wirklich Konfidenzintervalle oder p-Werte angibt und über „Signifikanz" spricht, sollte man das gelesen haben

Literaturliste contd.

- Online-Ressource zu Visualisierung
  - https://www.visualisingdata.com/
- Storytelling with Data [Buch]: Klassiker für Überzeugungsarbeit in Präsentationen von Ergebnissen
  - http://www.bdbanalytics.ir/media/1123/storytelling-with-data-cole-nussbaumer-knaflic.pdf
- Show Me the Numbers [Buch]: Ganz konkrete Tipps für die Praxis
  - https://courses.washington.edu/info424/2007/readings/Show_Me_the_Numbers_v2.pdf
- Now you see it [Buch]: Ebenfalls ganz konkrete Inhalte