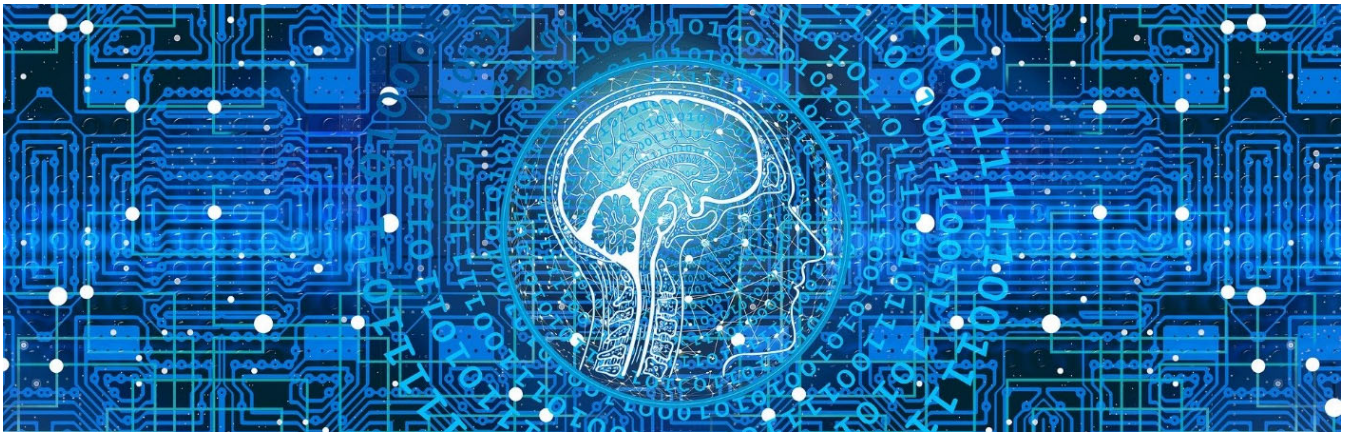
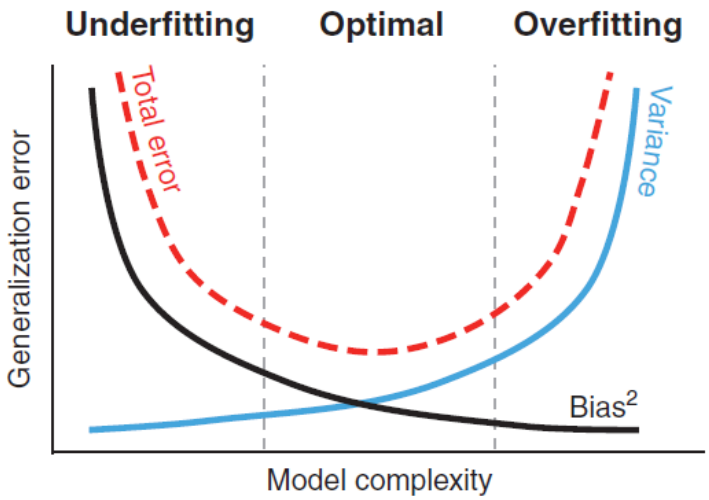
Data Science
5. Teil – Feature Engineering und Validierung

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



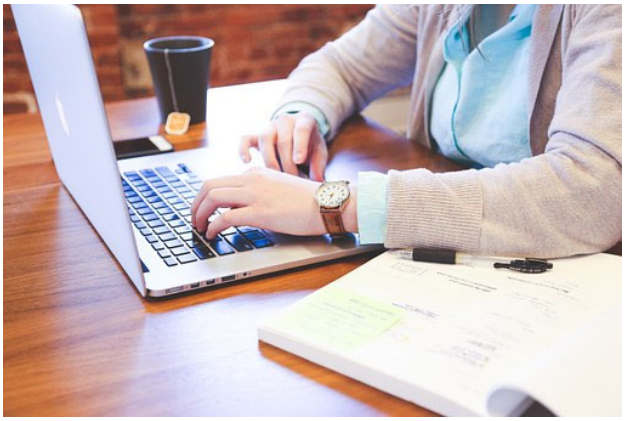
Wiederholung – Bias-Variance-Trade-Off



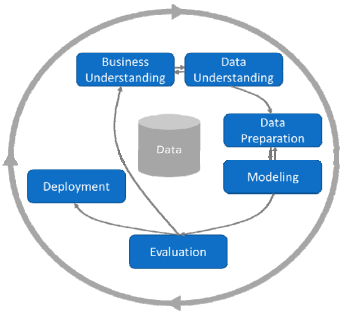
Bildquelle: Machine Learning with R, ..., Hefin I. Rhys, Manning, 2020

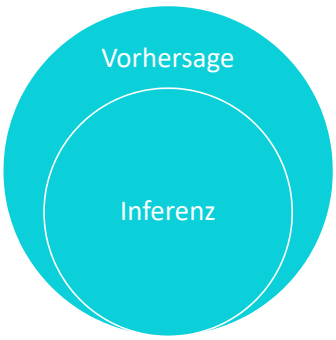
Inhalte der heutigen Vorlesung

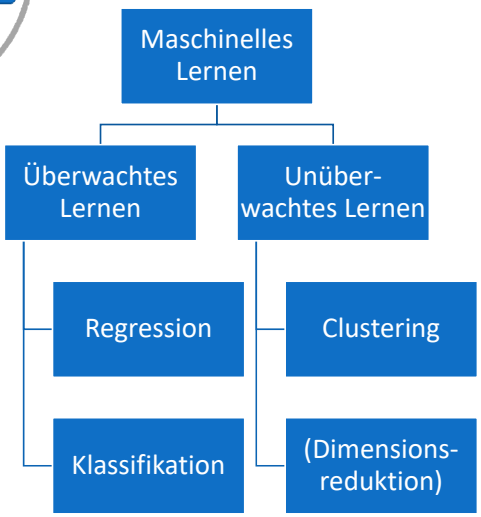
- Wiederholung
- Feature Engineering
- Klassifikationsmetriken
- Übung

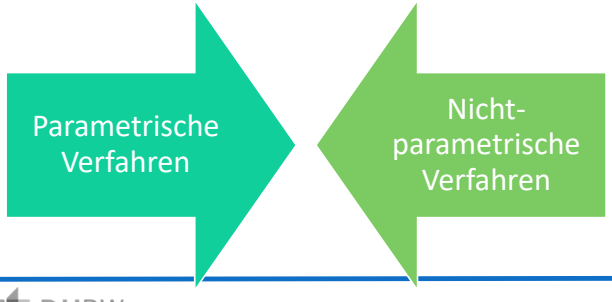


Wiederholung

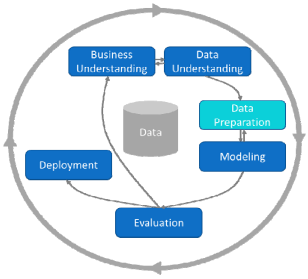




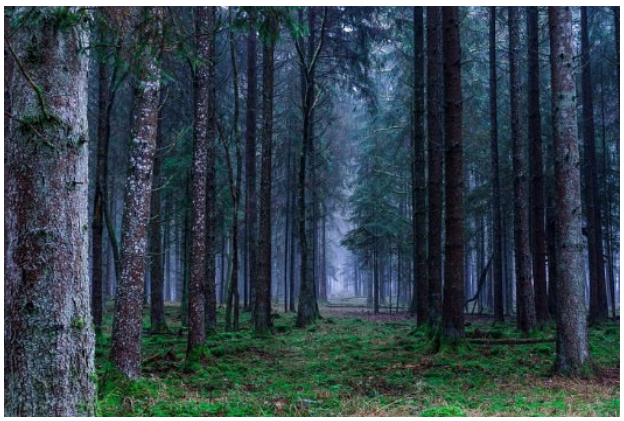




Data Preparation
Einführung und Herausforderungen bei Informationsqualität



- Anzahl der Datensätze
  - Zu wenig Datensätze
  - Zu viele Datensätze
- Anzahl der Attribute
  - Zu wenig Attribute
  - Zu viele Attribute
- Unausgeglichene Datensätze
  - Präklassifikation
  - Oversampling
  - Undersampling
  - Fehlermetriken nutzen
- Erfassung vs. Nützlichkeit
  - **Tafelanschrieb**



Feature Engineering & Feature Extraction
»as much of an art as a science«

- **Was sagt uns das?**
- Allgemeine Brainstorming-Regeln »adaptiert«
- Je mehr desto besser
- Erstmal gibt es keine falschen Features
- Semantische Informationen nutzen

- Später: Reduktion der Dimensionen wenn nötig

dschool-old.stanford.edu/groups/k12/wiki/d3f14/Brainstorming_with_Rules.html

Zitat oben: [O'Neil and Schutt 2013]

Data Preparation
Umgang mit »echten« Daten

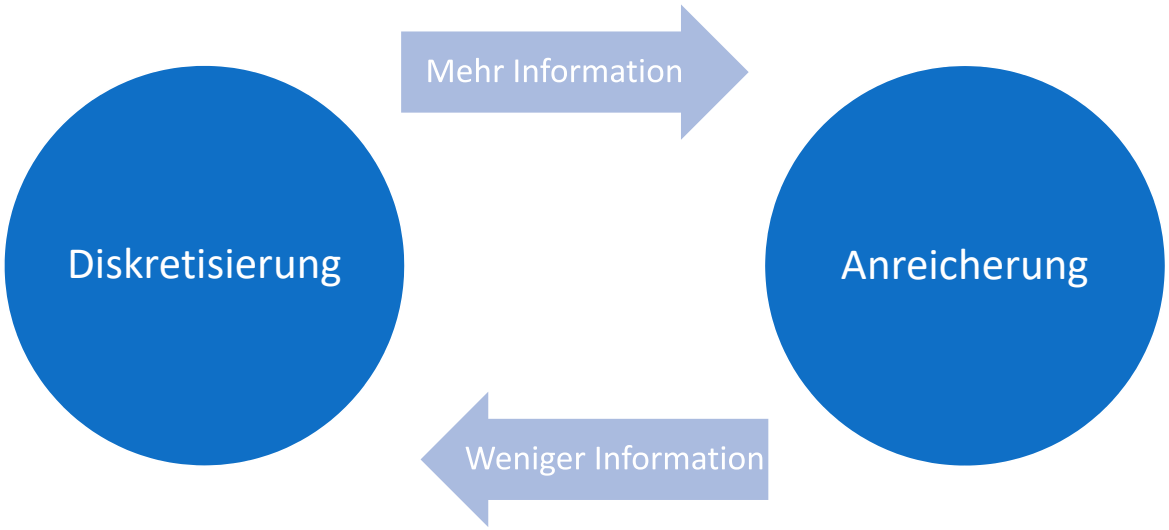
Übung

Automarke	Farbe	Baujahr	Erwerber	Datum	Wohnort	Preis
DAIMLER	Rot	2014	Meier	12.03.2018	Stuttgart	15.000 EUR
Mercedes	Grün	2015	Müller	07.02.2018	Hamburg	10.000 EUR
BMW	Blau	1998	Mustermann	08/08/2017	Stutgart	
VW	Rot	17	Péle	2016	Berlin	5.000 EUR
Audi	Gelb	2014	21	23.5.2016	Paris	5.000,00 EUR
BMW	Grün	1948	NULL	1900-01-01	Stuttgart	170.000 EUR
DAIMLER	Weiss	1856	Adenauer	2017	München	0 EUR
BMW	NULL	1999	Hermann	23.08.2017	Karlsruhe	28.000 EUR
Audi		2012	Haas	12.2.2018	Potsdam	7.000 EUR
VW	Rot	2011	Becker	1.1.2018	Muenchen	5.000 $
NULL	Blau	2014	Heimann	27.12.2017	Berlin	8,000 EUR
VW	Grün	2018	Bertoli	31.11.2017	Stuttgart	08.000 EUR

Datentypen und -skalen

Skala	Alias	Mögliche Operationen	Beispiele und Aussagen
Nominalskala	Kategoriale Daten, qualitatives Merkmal	Gleichheit, Ungleichheit (== / !=), Häufigkeit (Modus)	Zweitstimme bei der Bundestagswahl, Geschlecht (gleiche Wahl wie ..)
Ordinalskala	Rangordnung	Ordnen möglich (==, !=, >, <), Häufigkeit, Reihenfolge, Median	Likert-Skala (stimme zu, stimme eher zu, teils / teils, stimme eher nicht zu, lehne ab), Schulnoten (mehr / weniger als..)
Intervallskala (Kardinalskala)	Quantitative Merkmale, metrische Daten, numerical data	Abstände (Intervalle) besitzen eine Bedeutung (==, !=, >, <, +, -, *, %), Häufigkeit, Reihenfolge, Abstand, arith. Mittel	Temperatur (Celsius), Geburtsjahr (Unterschied ist..), IQ
Verhältnisskala (Kardinalskala)		Mit absolutem Nullpunkt, Häufigkeit, Reihenfolge, Abstand	Einkommen, Alter (doppelt so viel), Geschwindigkeit, Längen, Zeiten, ..

Datentransformationen



Encoding
Kategorische Daten / Nominalskala => Ordinalskala / Zahlen

- Label Encoding

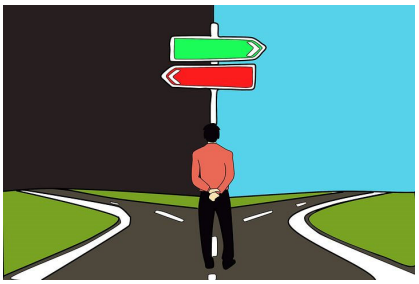
- One-Hot-Encoding

- **Übung: Encoding der Typ 1 im Pokemon Dataset (LabelEncoder, OneHotEncoder, OrdinalEncoder)**

Typische weitere Datentransformationen

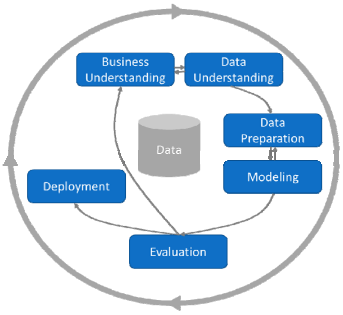
- **Anreicherungen**
  - Zusätzliche Informationen zufügen (z. B. Größe der Stadt zu Stadt)
- **Diskretisierung**
  - Diskrete Werte vergeben für numerische Werte, bzw. gröbere Einteilungen wählen
  - Transformationen von Ordinal- zu Intervallskala (häufig: Durchschnitt von Bewertungen) **(Warum gefährlich?)**
- **Datumstransformationen**
  - Zusammenfassungen (häufig: Quartale), »One-hot-encoding«, zyklische Transformation (**Tafel**)
- **Z-Transformation**
  - Für jeden Wert Mittelwert abziehen und durch Standardabweichung teilen (xls) (**Warum?**)
- Funktion anwenden mit dem Ziel, **normalverteilte Attribute** zu erhalten (**Warum?**)
- **Ausreißererkennung**
  - Viele Algorithmen reagieren »schlecht« auf Ausreißer (**Wie?**) (Stichwort: Multivariate Ausreißer)
- Bei fortgeschrittenen Problemen:
  - Umgang mit fehlenden Werten ist ein großes praktisches Problem
  - **Sparsity** – dünnbesetzt / viele 0-Werte
  - https://web.stanford.edu/~hastie/StatLearnSparsity_files/SLS_corrected_1.4.16.pdf

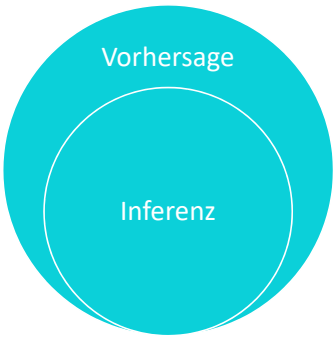
Möglichkeiten zur Auswahl relevanter Attribute
Dimensionen wählen oder reduzieren

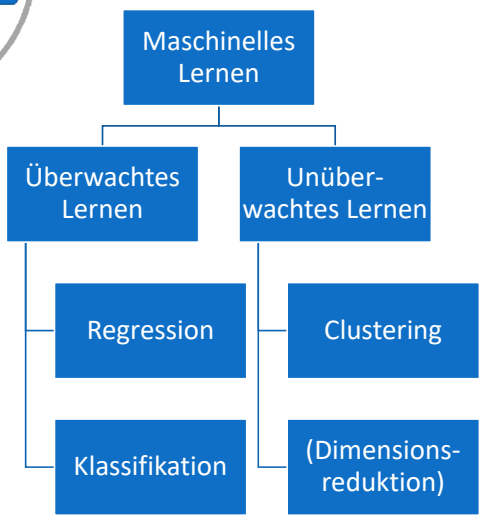


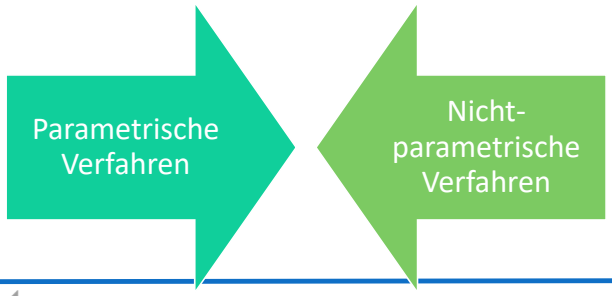
- **Forward**
  - Zufügen von Attributen und Ergebnis prüfen
- **Backward**
  - Streichen von Attributen und Ergebnis prüfen
- **Mixed**
  - Kombination aus Forward und Backward
- **Wrapper**
  - Wähle k Stück, dann »alle« Kombinationen von $\binom{n}{k}$ Attributen zufällig
  - **Effekt?**
- **Algorithmisch oder mathematisch**
  - Mit bestimmten Verfahren, die das bereits für einen übernehmen (z. B. Lasso / Ridge Regression)
  - Dimensionsreduktionsverfahren (Bsp. PCA), evtl. ein Thema für letzte Stunde

Wiederholung





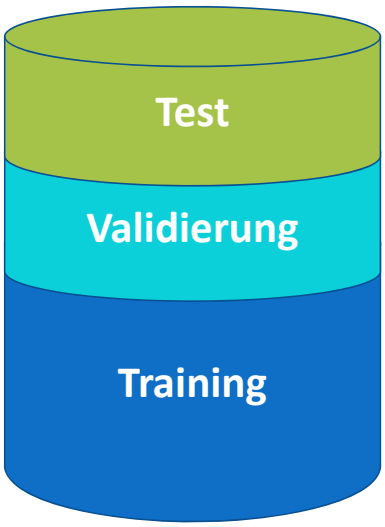




Wiederholung: Validierungs- und Testdatensätze

Unterteilung der Daten in Training-, Validierungs- und Testdaten

- Modelle sollten mittels nicht im Training verwendeter Daten **geprüft** werden
  -> kein Overfitting, Generalisierbarkeit sicherstellen
  -> es ist häufig schlechter auf Testdaten
- **Training**: ~80 % der Daten für Erzeugung der Modelle
  Verschiedene Algorithmen
  Verschiedene Parameter
- **Validierung**: Messung des Fehlers der Modellkandidaten und Auswahl des besten Modells
- **Test**: Überprüfung der Qualität des gewählten Modells am Ende



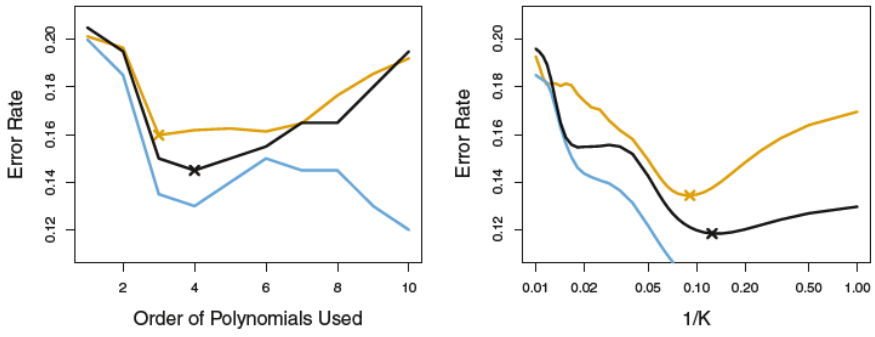
K-fache Kreuzvalidierung
Cross-Validation

Diskussion: Wann braucht man das?
Was bringt das für „Nebeneffekte“?
Alternative: Bootstrapping (später)

- Ca. 80% der Daten in k Teile aufteilen, Validierung auf einem Teil, Nutzung der anderen k-1 Teile für Training
- Typische Werte für k: 5 oder 10
- Berechnung der Gesamtgenauigkeit als Durchschnitt über die k Iterationen
- Berechnung der Testgenauigkeit auf dem Testset (ca. 20 % der Daten, **nicht für Training / Validierung**)

Iteration k	K=1	K=2	K=3	K=4	K=5	Testset
Datenteil 1	Validierung	Training	Training	Training	Training	0,84
Datenteil 2	Training	Validierung	Training	Training	Training	Testscore
Datenteil 3	Training	Training	Validierung	Training	Training	
Datenteil 4	Training	Training	Training	Validierung	Training	
Datenteil 5	Training	Training	Training	Training	Validierung	Cross-V. Score
Metrik	0,9	0,85	0,87	0,89	0,91	0,884

Beispiel: Kreuzvalidierung und Fehlerraten bei Klassifikation



Test error (brown), training error (blue), and 10-fold CV error (black) on the two-dimensional classification data displayed in Figure 5.7. Left: *Logistic regression using polynomial functions of the predictors. The order of the polynomials used is displayed on the x-axis.* Right: *The KNN classifier with different values of K, the number of neighbors used in the KNN classifier.* [James et al. 2013]

Wie finden wir das „richtige“ Modell?

1. Was sind die richtigen **Features**?
2. Was ist das richtige **Modell**?
3. Was sind die richtigen **Parameter**?
4. Was sind die richtigen **Hyperparameter**?

Parameter und Hyperparameter
Ein kleiner, aber feiner Unterschied

- **Parameter**
  - Intrinsisch für dieses Modell
  - Aus den Daten geschätzt
  - Bsp.: Anstieg einer Regressionsgerade, Gleichung bei logistischer Regression

Errechnet scikit-learn automatisch!

- **Hyperparameter**
  - Kann nicht aus den Daten geschätzt werden
  - Kontrolliert wie ein Modell Vorhersagen macht
  - Verschiedene Algorithmen haben verschiedenste Hyperparameter
  - Müssen „extern“ vorgegeben werden
  - Bsp.: k der kNN-Methode, Anzahl der Cluster beim K-Means Verfahren

Müssen wir vorgeben / wählen / optimieren

Hyperparameter finden

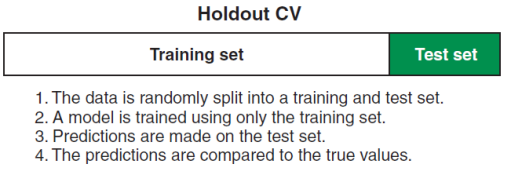
- Auswahl durch Vorkenntnis (ähnliches Problem, im Internet gefunden)
  - Meistens suboptimal
- Einige Werte ausprobieren
  - Dauert, gibt aber guten Startpunkt
- **Hyperparameter-Tuning** als automatisierten Auswahlprozess
  - Maximiert die Wahrscheinlichkeit den wirklich optimalen Wert zu finden
  - Rechenintensiv (ggfs. auskommentieren nachdem man den Wert gefunden hat)
  - Teilweise ist ausprobieren die einzige
  - In scikitlearn: `gridsearch`

Hyperparameter automatisch finden

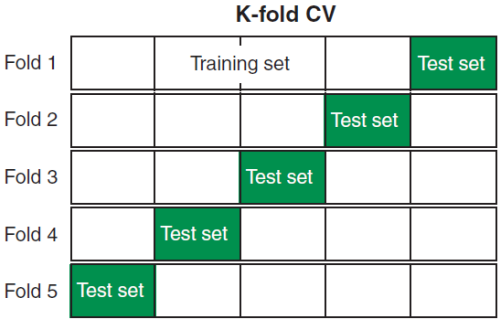
- Oft genutzter Weg für diskrete Hyperparameter ist Grid-Search
- Geeignetes Sampling des Parameterraums
- Einfachste Methode
- Bei kontinuierlichen Hyperparametern ist Grid-Search zu rechenaufwendig, deshalb andere Methoden wie z.B. Random-Search
- Kreuzvalidierung – eine der drei Arten – über alle Parameter des Suchraums hinweg und Auswahl des Hyperparameters mit der besten Performance

Cross-Validation (CV)
Drei Verfahren zur Kreuzvalidierung

1. Holdout Cross-Validation (Einfache Kreuzvalidierung)
2. K-Fold Cross-Validation (Stratifizierte Kreuzvalidierung)
3. Leave-One-Out Cross-Validation (LOOCV, Leave-One-Out-Kreuzvalidierung LOOCV)



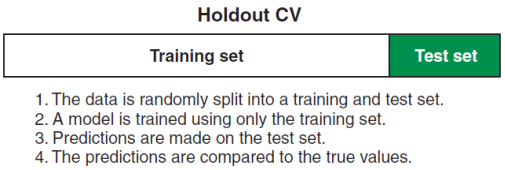
1. The data is randomly split into a training and test set.
2. A model is trained using only the training set.
3. Predictions are made on the test set.
4. The predictions are compared to the true values.



1. The data is randomly split into *k* equal-sized folds.
2. Each fold is used as the test set once, where the rest of the data makes the training set.
3. For each fold, predictions are made on the test set.
4. The predictions are compared to the true values.

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

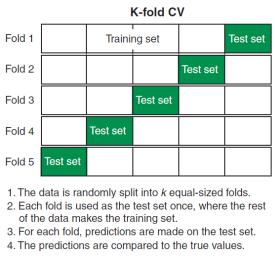
Holdout Cross Validation



1. The data is randomly split into a training and test set.
2. A model is trained using only the training set.
3. Predictions are made on the test set.
4. The predictions are compared to the true values.

- Größe des Test-Sets ist wieder ein Kompromiss:
- Selbst die Abschätzung der Performance durch die CV folgt einen Bias-Variance Trade-Off
- Je kleiner das Test-Set ist, desto mehr Varianz hat die Abschätzung der Performance meines Modells
- Und je kleiner das Trainings-Set ist, desto mehr Bias hat die Performance-Schätzung
- Ein übliches Verhältnis ist 2/3 Trainings-Set und 1/3 Test-Set, was aber von der insgesamt vorhandenen Menge an Daten abhängt
- Zuweisung zu den einzelnen Sets zwar zufällig aber bei Klassifikation stratifiziert nach Klassen
- Sonst könnte im Extremfall eventuell nur eine der Klassen im Test-Set landen
- Performance-Metrik hängt sehr stark vom Aufteilungsver-hältnis zwischen den Datensätzen ab – für verschiedene Durchläufe oft sehr unterschiedliche Ergebnisse
- Vorteil ist der geringere Rechenaufwand im Vergleich zu den anderen CV-Methoden – kann bei rechenintensiven Verfahren eine Rolle spielen

K-Fold Cross Valdiation



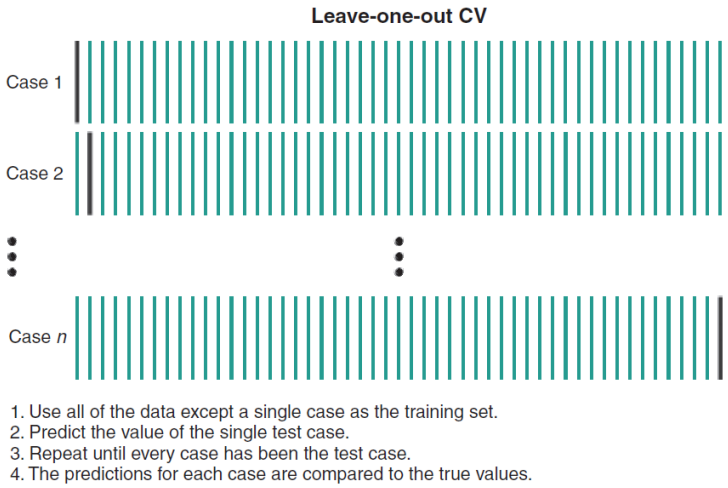
- Wir erzeugen praktisch k „kleine Holdout-Samples“ aus unserem Datensatz
  - Aufspaltung des Datensatzes in k gleich große Teile:
  - Jeweils ein Fold als Holdout/Test-Set
- Am Ende erhalten wir ein k Abschätzungen für die Performance unseres Modells:
  - Durchschnitt aller k Werte als Abschätzung der Performance unseres Modells auf „ungesehenen“ Daten
- Auch Abschätzung der Streuung möglich
- Erweiterung der Methode durch – repeated k-Fold CV:
  - nach dem ersten Durchlauf werden die Werte über die Folds zufällig verteilt („geshuffled“) und die CV für die sich jetzt ergebenden Folds wiederholt
- Ein üblicher Wert für k ist 10, wobei das auch hier wieder von der Größe des vorhandenen Datensatzes abhängt
  - Datensatz in 10 gleichgroße Teile aufteilen, dann CV durchführen
  - Wird das ganze 5 mal wiederholt, ergibt das eine 10-Fold CV 5-mal wiederholt – was nicht das gleiche ist wie eine 50-Fold CV – Abschätzung der Modell-Performance als Schnitt von 50 Durchläufen
- Falls die nötige Rechenleistung verfügbar ist, eine repeated K-Fold CV einer gewöhnlichen vorzuziehen

Repeated k-Fold Cross Validation

- Auswahl von Wiederholungen
- Ziel ist eine möglichst genaue und stabile Schätzung der Modell-Performance zu erhalten
- D.h. je mehr Wiederholungen desto besser – genauere und stabilere Schätzung, das verbessert sich aber nur bis zu einem gewissen Punkt
- Rezept: Mit einer Wiederholungsrate starten, die von den Anforderungen machbar erscheint – variiert die Performance-Schätzung zu sehr, Anzahl an Wiederholungen erhöhen bis sie stabil sind

Leave-One-Out Cross-Validation (LOOCV)

- Praktisch das maximal mögliche k für eine K-Fold CV gewählt:
- Ein einziges Sample bildet das Test-Set, während auf allen restlichen Samples trainiert wird
- Da ein einziges Sample als Test-Set verwendet wird, variieren die Ergebnisse der Durchläufe sehr stark
- Aber es kann im Vergleich zu K-Fold CV auf kleineren Datensätzen weniger variierende Schätzer liefern
- Bei kleinem Datensatz liefert K-Fold CV kleinere Trainingsdatensätze – Varianz des geschätzten Modells wird größer durch das zufällig Sampling von Daten für den Trainingsdatensatz
- Nützlich für kleinere Datensätz
- Vom Rechenaufwand auch weniger anspruchsvoll als *Repeated K-Fold CV*



1. Use all of the data except a single case as the training set.
2. Predict the value of the single test case.
3. Repeat until every case has been the test case.
4. The predictions for each case are compared to the true values.

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

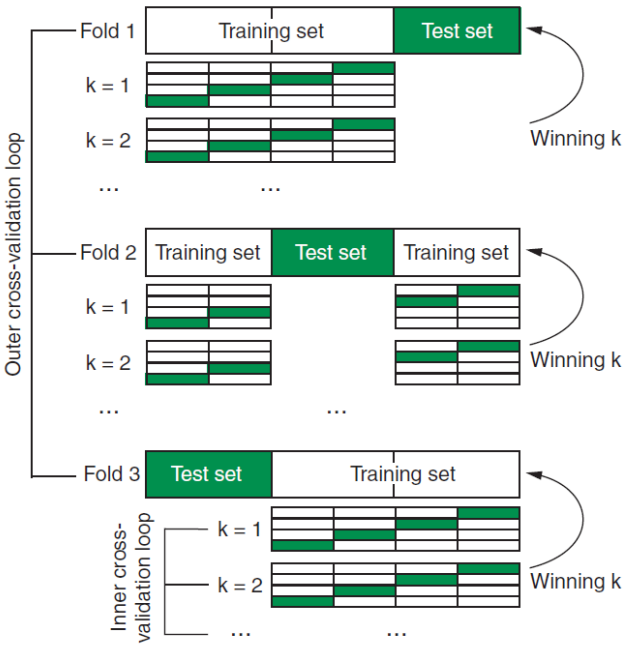
Nested CV

- Innere Schleife kreuzvalidiert verschiedene Hyperparameter und der beste Hyperparameter wird dann in der äußeren Kreuzvalidierungsschleife genutzt, um schließlich das Modell zu validieren
- Aufteilung der Daten in Trainings- und Test-Set – **Outer-Loop**, beliebiges CV-Verfahren
- Trainings-Set wird genutzt um jeden Wert des Hyperparameter-Suchraums zu kreuvalidieren – **Inner-Loop**, auch wieder beliebiges CV-Verfahren
- Der Hyperparameter, der die beste kreuzvalidierte Performance von jeder inneren Schleife hat, wird an die äußere weitergegeben
- Ein Modell mit dem besten Hyperparameter seiner inneren Schleife wird auf jedem Trainings-Set der äußeren Schleife trainiert – und dann auf das Test-Set angewendet
- Die durchschnittlicher Performance dieser Modelle, die in der äußeren Schleife berechnet wurden, sind die Abschätzung für die Performance auf „ungesehenen“ Daten

Nested CV

- Nested CV Example:

- Outer-Loop – 3-Fold CV
- Inner-Loop – 4-Fold CV auf jedem Trainings-Set der Outer-Loop
- 4-Fold-CV für jeden in Frage kommenden Hyperparameter k auf dem Trainings-Set
- Bestes k wird in der Outer-Loop benutzt um die Performance mit 3-Fold CV für dieses k zu bestimmen
- **Diskussion: Unterschied zu vorher vorgestelltem Verfahren**

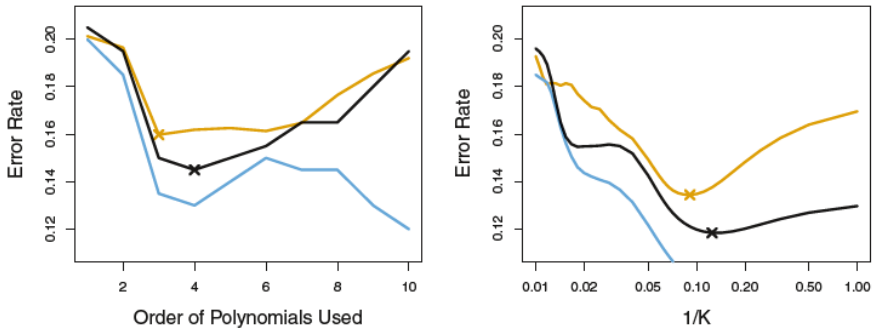


Machine Learning with R, …, Hefin I. Rhys, Manning, 2020

Trick 17

- Ein nicht validiertes Modell ist beim überwachten Lernen quasi wertlos!
- Ist das gefundene bzw. ausgewählte Modell schließlich erfolgreich „cross-validiert“, ***wird dieses Modell mit dem kompletten Datensatz (Trainings-Set + Test-Set) final trainiert***, um „die vorhandene Information bestmöglichst auszunutzen“

Wiederholung: Kreuzvalidierung und Fehlerraten bei Klassifikation



Test error (brown), training error (blue), and 10-fold CV error (black) on the two-dimensional classification data displayed in Figure 5.7. Left: *Logistic regression using polynomial functions of the predictors. The order of the polynomials used is displayed on the x-axis.* Right: *The KNN classifier with different values of K, the number of neighbors used in the KNN classifier.* [James et al. 2013]

Und nächste Stunde sehen Sie..

- Auf jeden Fall: Weitere ML-Verfahren die Sie für die PL nutzen können
- Evtl. Fehlermetriken für Klassifikation
- Evtl. Bootstrapping, Bagging, Boosting, Stratifikation

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