DataScience1

Grundlagen der Prognosemodellierung

Sebastian Sauer

2022-03-11 10:17:06

Contents

Chapter 1

Überblick

from Imgflip Meme Generator

1.1 Was Sie hier lernen und wozu das gut ist

Alle Welt spricht von Big Data, aber ohne die Analyse sind die großen Daten nur großes Rauschen. Was letztlich interessiert, sind die Erkenntnisse, die Einblicke, nicht die Daten an sich. Dabei ist es egal, ob die Daten groß oder klein sind. Natürlich erlauben die heutigen Datenmengen im Verbund mit leistungsfähigen Rechnern und neuen Analysemethoden ein Verständnis, das vor Kurzem noch nicht möglich war. Und wir stehen erst am Anfang dieser Entwicklung. Vielleicht handelt es sich bei diesem Feld um eines der dynamischsten Fachgebiete der heutigen Zeit. Sie sind dabei: Sie lernen einiges Handwerkszeugs des "Datenwissenschaftlers". Wir konzentrieren uns auf das vielleicht bekannteste Teilgebiet: Ereignisse vorhersagen auf Basis von hoch strukturierten Daten und geeigneter Algorithmen und Verfahren. Nach diesem Kurs sollten Sie in der Lage sein, typisches Gebabbel des Fachgebiet mit Lässigkeit mitzumachen. Ach ja, und mit einigem Erfolg Vorhersagemodelle entwickeln.

1.2 Lernziele

Nach diesem Kurs sollten Sie

- grundlegende Konzepte des statistischen Lernens verstehen und mit R anwenden können
- gängige Prognose-Algorithmen kennen, in Grundzügen verstehen und mit R anwenden können
- die Güte und Grenze von Prognosemodellen einschätzen können

1.3 Voraussetzungen

Um von diesem Kurs am besten zu profitieren, sollten Sie folgendes Wissen mitbringen:

- grundlegende Kenntnisse im Umgang mit R, möglichst auch mit dem tidyverse
- grundlegende Kenntnisse der deskriptiven Statistik
- grundlegende Kenntnis der Regressionsanalyse

1.4 Hinweise zu diesem Projekt

- Die URL zu diesem Projekt lautet <test.io>.
- Lesen Sie sich die folgenden Informationen bitte gut durch: Hinweise
- Den Quellcode finden Sie in diesem Github-Repo.
- Sie haben Feedback, Fehlerhinweise oder Wünsche zur Weiterentwicklung? Am besten stellen Sie hier einen *Issue* ein.
- Dieses Projekt steht unter der MIT-Lizenz.

1.5 Lernhilfen

1.5.1 Software

- Installieren Sie R und seine Freunde.
- Installieren Sie die folgende R-Pakete:
  - tidyverse
  - tidymodels
  - weitere Pakete werden im Unterricht bekannt gegeben (es schadet aber nichts, jetzt schon Pakete nach eigenem Ermessen zu installieren)
- R Syntax aus dem Unterricht findet sich im Github-Repo bzw. Ordner zum jeweiligen Semester.

1.5.2 Online-Zusammenarbeit

- Frag-Jetzt-Raum zum anonymen Fragen stellen während des Unterrichts. Der Keycode wird Ihnen vom Dozenten bereitgestellt.

- Padlet zum einfachen (und anonymen) Hochladen von Arbeitsergebnissen der Studentis im Unterricht. Wir nutzen es als eine Art Pinwand zum Sammeln von Arbeitsbeiträgen. Die Zugangsdaten stellt Ihnen der Dozent bereit.

1.6 Modulzeitplan

Nr.	Kalenderwoche	Datum	Thema
1	11	14.-18.3.22	Grundkonzepte
2	12	21.3.-25.3.	tidyverse, 2. Blick
3	13	28.3.-1.4.	tidymodels
4	14	4.4.-8.4.	kNN
5	15	11.4.-15.4.	Statistisches Lernen
6	16	18.4.-22.4	Wiederholung
7	17	25.4.-29.4	Logistische Regression
8	18	2.5.-6.5.	Naive Bayes
9	19	9.5.-13.5.	Entscheidungsbäume
10	20	16.5.-20.5.	Zufallswälder
11	21	23.5.-27.5.	Fallstudie
12	23	6.6.-10.6.	Wiederholung
13	24	13.6.-17.6.	GAM
14	25	20.6.-24.6.	Lasso und Co
15	26	27.6.-1.7.	Vertiefung

1.7 Literatur

Zentrale Kursliteratur für die theoretischen Konzepte ist **?**. Bitte prüfen Sie, ob das Buch in einer Bibliothek verfügbar ist. Die praktische Umsetzung in R basiert auf **?**; das Buch ist frei online verfügbar. Eine theoretische Konzepte sind **?** entnommen; dieser Test ist frei online verfügbar. In einigen Punkten ist **?** hilfreich; das Buch ist über SpringerLink in Ihrer Hochschul-Bibliothek verfügbar.

1.8 FAQ

- *Folien*
  - Frage: Gibt es ein Folienskript?

  - Antwort: Wo es einfache, gute Literatur gibt, gibt es kein Skript. Wo es keine gute oder keine einfach zugängliche Literatur gibt, dort gibt es ein Skript.

- *Englisch*
  - Ist die Literatur auf Englisch?
  - Ja. Allerdings ist die Literatur gut zugänglich. Das Englisch ist nicht schwer. Bedenken Sie: Englisch ist die lingua franca in Wissenschaft und Wirtschaft. Ein solides Verständnis englischer (geschriebener) Sprache ist für eine gute Ausbildung unerlässlich. Zu dem sollte die Kursliteratur fachlich passende und gute Bücher umfassen; oft sind das englische Titel.

- *Anstrengend*
  - Ist der Kurs sehr anstrengend, aufwändig?
  - Der Kurs hat ein mittleres Anspruchsniveau.

- *Mathe*
  - Muss man ein Mathe-Crack sein, um eine gute Note zu erreichen?
  - Nein. Mathe steht nicht im Vordergrund. Schauen Sie sich die Literatur an, sie werden wenig Mathe darin finden.

- *Prüfungsliteratur*
  - Welche Literatur ist prüfungsrelevant?
  - Die Prüfung ist angewandt, z.B. ein Prognosewettbewerb. Es wird keine Klausur geben, in der reines Wissen abgefragt wird.

- *Nur R?*
  - Wird nur R in dem Kurs gelehrt? Andere Programmiersprachen sind doch auch wichtig.
  - In der Datenanalyse gibt es zwei zentrale Programmiersprachen, R und Python. Beide sind gut und beide werden viel verwendet. In einer Grundausbildung sollte man sich auf eine Sprache begrenzen, da sonst den Sprachen zu viel Zeit eingeräumt werden muss. Wichtiger als eine zweite Programmiersprache zu lernen, mit der man nicht viel mehr kann als mit der ersten, ist es, die Inhalte des Fachs zu lernen.

Chapter 2

Modulüberblick

2.1 Grundkonzepte

2.1.1 Datum

- 14.-18.3.22

2.1.2 Lernziele

- Sie können erläutern, was man unter statistischem Lernen versteht.
- Sie wissen, war Overfitting ist, wie es entsteht, und wie es vermieden werden kann.
- Sie kennen verschiedenen Arten von statistischem Lernen und können Algorithmen zu diesen Arten zuordnen.

2.1.3 Vorbereitung

- Lesen Sie die Hinweise zum Modul.
- Installieren (oder Updaten) Sie die für dieses Modul angegeben Software.
- Lesen Sie die Literatur.

2.1.4 Literatur

- Rhys, Kap. 1
- evtl. Sauer, Kap. 15

2.1.5 Aufgaben

- Machen Sie sich mit 'Kaggle' vertraut
- Arbeiten Sie diese Regressionsfallstudie (zum Thema Gehalt) auf Kaggle auf
- Werfen Sie einen Blick in diese Fallstudie auf Kaggle zum Thema Hauspreise
- Wiederholen Sie unser Vorgehen in der Fallstudie zu den Flugverspätungen

2.1.6 Vertiefung

- Verdienst einer deutschen Data Scientistin
- Weitere Fallstudie zum Thema Regression auf Kaggle
- Crashkurs Data Science (Coursera, Johns Hopkins University) mit 'Star-Dozenten'

2.1.7 Hinweise

- Bitte beachten Sie die Hinweise zum Präsenzunterricht und der Streamingoption.
- Bitte stellen Sie sicher, dass Sie einen einsatzbereiten Computer haben und dass die angegebene Software (in aktueller Version) läuft.

2.2 tidyverse, 2. Blick

2.2.1 Datum

- 21.3.-25.3.

2.2.2 Lernziele

- Sie können Funktionen, auch anonyme, in R schreiben.
- Sie können Datensätze vom Lang- und Breit-Format wechseln.
- Sie können Mapping-Funktionen anwenden.
- Sie können eine dplyr-Funktion auf mehrere Spalten gleichzeitig anwenden.

2.2.3 Vorbereitung

- Lesen Sie die Literatur.

2.2.4 Literatur

- Rhys, Kap. 2

2.3 tidymodels

2.3.1 Datum

- 28.3.-1.4.

2.3.2 Literatur

- TMWR

2.4 kNN

2.4.1 Datum

- 4.4.-8.4.

2.4.2 Literatur

- Rhys, Kap. 3

2.5 Statistisches Lernen

2.5.1 Datum

- 11.4.-15.4.

2.5.2 Literatur

- Rhys, Kap. 3

2.5.3 Vertiefung

- Fields arranged by purity, xkcd 435

2.5.4 Hinweise

- In dieser Woche fällt die Übung aus (Ostern).

2.6 Wiederholung

2.6.1 Datum

- 18.4.-22.4

2.6.2 Hinweise

- In dieser Woche fällt die Vorlesung aus (Ostern).

2.7 Logistische Regression

2.7.1 Datum

- 25.4.-29.4

2.7.2 Literatur

- Rhys, Kap. 4

2.8 Naive Bayes

2.8.1 Datum

- 2.5.-6.5.

2.8.2 Literatur

- Rhys, Kap. 6

2.9 Entscheidungsbäume

2.9.1 Datum

- 9.5.-13.5.

2.9.2 Literatur

- Rhys, Kap. 7

2.10 Zufallswälder

2.10.1 Datum

- 16.5.-20.5.

2.10.2 Literatur

- Rhys, Kap. 8

2.11 Fallstudie

2.11.1 Datum

- 23.5.-27.5.

2.11.2 Literatur

- Rhys, Kap.9

2.11.3 Hinweise

- Nächste Woche ist Blockwoche; es findet kein regulärer Unterricht statt.
- Diese Woche fällt die Übung aus.

2.12 Wiederholung

2.12.1 Datum

- 6.6.-10.6.

2.12.2 Hinweise

- In dieser Woche fällt die Vorlesung aus (Pfingsten).

2.13 GAM

2.13.1 Datum

- 13.6.-17.6.

2.13.2 Literatur

- Rhys, Kap. 10

2.14 Lasso und Co

2.14.1 Datum

- 20.6.-24.6.

2.14.2 Literatur

- Rhys, Kap. 11

2.15 Vertiefung

2.15.1 Datum

- 27.6.-1.7.

2.15.2 Literatur

- Rhys, Kap. 12

2.15.3 Vertiefung

- Wie man eine Data-Science-Projekt strukturiert

2.15.4 Hinweise

- Nach dieser Woche endet der Unterricht.

Chapter 3

Prüfung

3.1 tl;dr: Zusammenfassung

Vorhersagen sind eine praktische Sache, zumindest wenn Sie stimmen. Wenn Sie den DAX-Stand von morgen genau vorhersagen können, rufen Sie mich bitte sofort an. Genau das ist Ihre Aufgabe in dieser Prüfungsleistung: Sie sollen Werte vorhersagen.

Etwas konkreter: Stellen Sie sich ein paar Studentis vor. Von allen wissen Sie, wie lange die Person für die Statistikklausur gelernt hat. Außerdem wissen Sie die Motivation jeder Person und vielleicht noch ein paar noten-relevante Infos. Und Sie wissen die Note jeder Person in der Statistikklausur. Auf dieser Basis fragt sie ein Student (Alois), der im kommenden Semester die Prüfung in Statistik schreiben ~~muss~~ will: "Sag mal, wenn ich 100 Stunden lerne und so mittel motiviert bin (bestenfalls), welche Note kann ich dann erwarten?". Mit Hilfe Ihrer Analyse können Sie diese Frage (und andere) beantworten. Natürlich könnten Sie es sich leicht machen und antworten: "Mei, der Notendurchschnitt war beim letzten Mal 2.7. Also ist 2.7 kein ganz doofer Tipp für deine Note." Ja, das ist keine doofe Antwort, aber man genauere Prognose machen, wenn man es geschickt anstellt. Da hilft Ihnen die Statistik (doch, wirklich).

Kurz gesagt gehen Sie so vor: Importieren Sie die Daten in R, starten Sie die nötigen R-Pakete und schauen Sie sich die Daten unter verschiedenen Blickwinkeln an. Dann nehmen Sie die vielversprechendsten Prädiktoren in ein Regressionsmodell und schauen sich an, wie gut die Vorhersage ist. Wiederholen Sie das ein paar Mal, bis Sie ein Modell haben, das Sie brauchbar finden. Mit diesem Modell sagen Sie dann die Noten der neuen Studis (Alois und Co.) vorher. Je genauer Ihre Vorhersage, desto besser ist Ihr Prüfungsergebnis.

3.2 Vorhersage

Neben der erklärenden, rückwärtsgerichteten Modellierung spielt insbesondere in der Praxis die *vorhersageorientierte* Modellierung eine wichtige Rolle: Ziel ist es, bei gegebenen, neuen Beobachtungen die noch unbekannten Werte der Zielvariablen y *vorherzusagen*, z.B. für neue Kunden auf Basis von soziodemographischen Daten den *Kundenwert* – möglichst genau – zu prognostizieren. Dies geschieht auf Basis der vorhandenen Daten der Bestandskunden, d.h. inklusive des für diese Kunden bekannten Kundenwertes.

Ihnen werden *zwei Teildatenmengen* zur Verfügung gestellt: Zum einen gibt es die Trainingsdaten (auch *Lerndaten* genannt) und zum anderen gibt es Anwendungsdaten (auch *Testdaten* genannt), auf die man das Modell anwendet.

1. Bei den Trainingsdaten (Train-Sample) liegen sowohl die erklärenden Variablen $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ als auch die Zielvariable y vor. Auf diesen Trainingsdaten wird das Modell $y = f(x) + \epsilon = f(x_1, x_2, \ldots, x_n) + \epsilon$ gebildet und durch $\hat{f}(\cdot)$ geschätzt. Es ist also die Variable y vorherszusagen.

2. Dieses geschätzte Modell ($\hat{f}(\cdot)$) wird auf die Anwendungsdaten $\mathbf{x_0}$, für die (Ihnen) die Werte der Zielvariable y unbekannt sind, angewendet, d.h., es wird $\hat{y}_0 := \hat{f}(\mathbf{x}_0)$ berechnet. Der unbekannte Wert y_0 der Zielvariable y wird durch $\hat{y}_0$ prognostiziert.

Liegt zu einem noch späteren Zeitpunkt der eingetroffene Wert y_0 der Zielvariable y vor, so kann die eigene Vorhersage $\hat{y}_0$ evaluiert werden, d.h. z.B. kann der Fehler $e = y_0 - \hat{y}_0$ zwischen prognostiziertem Wert $\hat{y}_0$ und wahrem Wert y_0 analysiert werden.

In der praktischen Anwendung können zeitlich drei aufeinanderfolgende Schritte unterschieden werden (vergleiche oben):

1. die *Trainingsphase*, d.h., die Phase für die sowohl erklärende ($\mathbf{x}$) als auch die erklärte Variable (y) bekannt sind. Hier wird das Modell geschätzt (gelernt): $\hat{f}(\mathbf{x})$. Dafür wird der Trainingsdatensatz genutzt.

2. In der folgenden *Anwendungsphase* sind nur die erklärenden Variablen ($\mathbf{x_0}$) bekannt, nicht y_0. Auf Basis der Ergebnisses aus dem 1. Schritt wird $\hat{y}_0 := \hat{f}(\mathbf{x}_0)$ prognostiziert.

3. Evt. gibt es später noch die *Evaluierungsphase*, für die dann auch die Zielvariable (y_0) bekannt ist, so dass die Vorhersagegüte des Modells überprüft werden kann.

Im Computer kann man dieses Anwendungsszenario *simulieren*: man teilt die Datenmenge *zufällig* in eine Lern- bzw. Trainingsstichprobe (Trainingsdaten;

$(\mathbf{x}, \mathbf{y})$) und eine Teststichprobe (Anwendungsdaten, ($\mathbf{x_0}$)) auf: Die Modellierung erfolgt auf den Trainingsdaten. Das Modell wird angewendet auf die Testdaten (Anwendungsdaten). Da man hier aber auch die Zielvariable (y_0) kennt, kann damit das Modell evaluiert werden.

3.3 Hauptziel: Genaue Prognose

Ihre Aufgabe ist: Spielen Sie den Data-Scientist! Konstruieren Sie ein Modell auf Basis der Trainingsdaten $(\mathbf{x}, \mathbf{y})$ und sagen Sie für die Anwendungsdaten ($\mathbf{x_0}$) die Zielvariable möglichst genau voraus ($\hat{y}_0$).

Ihr(e) Dozent*in kennt den Wert der Zielvariable (y_0).

Von zwei Prognosemodellen zum gleichen Datensatz ist dasjenige Modell besser, das weniger Vorhersagefehler aufweist, also genauer vorhersagt. Kurz gesagt: Genauer ist besser.

3.4 Zum Aufbau Ihrer Prognosedatei im CSV-Format

1. Die CSV-Datei muss aus genau zwei Spalten mit (exakt) folgenden Spaltennamen bestehen:

a) `id`: Den ID-Wert jedes vorhergesagten Wertes
b) `pred`: Der vorhergesagte Wert.

3. Umlaute sind zu ersetzen (also `Süß` wird `Suess` etc.).

4. Die CSV-Datei muss als *Spaltentrennzeichen* ein *Komma* verwenden und als *Dezimaltrennzeichen* einen *Punkt* (d.h. also die *Standardformatierung* einer CSV-Datei; *nicht* die deutsche Formatierung).

5. Die CSV-Datei muss genau die Anzahl an Zeilen aufweisen, die der Zeilenlänge im Test-Datensatz entspricht.

6. Prüfen Sie, dass Ihre CSV-Datei sich problemlos lesen lässt. Falls keine (funktionstüchtige) CSV-Datei eingereicht (hochgeladen) wurde, ist die Prüfung nicht bestanden. Tipp: Öffnen Sie die CSV-Datei mit einem Texteditor und schauen Sie sich an, ob alles vernünftig aussieht. Achtung: Öffnen Sie die CSV-Datei besser nicht mit Excel, da Excel einen Bug hat, der CSV-Dateien verfälschen kann auch ohne dass man die Datei speichert.

3.5 Einzureichende Dateien

1. Folgende* Dateiarten* sind einzureichen:
    1. Prognose: Ihre *Prognose-Datei* (CSV-Datei)
    2. Analyse: Ihr *Analyseskript* (R-, Rmd- oder Rmd-Notebook-Datei)
2. Weitere Dateien sind nicht einzureichen.
3. Komprimieren Sie die Dateien *nicht* (z.B. via *zip*).
4. Der Name jeder eingereichnte Datei muss wie folgt lauten: `Nachname_Vorname_Matrikelnummer_Da`
   Beispiel: `Sauer_Sebastian_0123456_Prognose.csv` bzw. `Sauer_Sebastian_0123456_Analyse.R`

3.6 Tipps

3.6.1 Tipps für eine gute Prognose

- Schauen Sie in die Literatur.
- Evtl. kann eine Datenvorverarbeitung (Variablentransformation, z.B. log() oder die Elimination von Ausreißern) helfen.
- Überlegen Sie sich Kriterien zur Modell- und/ oder Variablenauswahl. Auch hierfür gibt es Algorithmen und R-Funktionen.
- Vermeiden Sie Über-Anpassung (Overfitting).
- Vermeiden Sie viele fehlende Werte bei Ihrer Prognose. Fehlende Werte werden bei der Benotung mit dem Mittelwert (der vorhandenen Prognosewerte Ihrer Einreichung) aufgefüllt.

3.6.2 Tipps zur Datenverarbeitung

- Ein "deutsches" Excel kann Standard-CSV-Dateien nicht ohne Weiteres lesen. Online-Dienste wie Google Sheets können dies allerdings.

3.6.3 Tipps zum Aufbau des Analyseskripts

- Zu Beginn des Skripts sollten alle verwendeten R-Pakete mittels `library()` gestartet werden.
- Zu Beginn des Skripts sollten die Daten von der vom Dozenten bereitgestellten URL importiert werden (*nicht* von der eigenen Festplatte, da das Skript sonst bei Dritten, wie Ihrem Prüfer, nicht lauffähig ist).

3.6.4 Sonstiges

- Legen Sie regelmäßig Sicherheitskopien Ihrer Arbeit an (ggf. auf einem anderen Datenträger).
- Achten Sie darauf, dass Sie nicht durcheinander kommen, in welcher Datei der aktuelle Stand Ihrer Arbeit liegt.

3.7 Bewertung

3.7.1 Kriterien

- Es gibt drei Bewertungskriterien:
  - *Formalia*: u.a. Reproduzierbarkeit der Analyse, Lesbarkeit der Syntax, Übersichtlichkeit der Analyse.
  - *Methode*: u.a. methodischer Anspruch und Korrektheit in der Explorativen Datenanalyse, Datenvorverarbeitung, Variablenauswahl und Modellierungsmethode.
  - *Inhalt*: **Vorhersagegüte**.
- Das zentrale Bewertungskriterium ist *Inhalt*; die übrigen beiden Kriterien fließen nur bei besonders guter oder schlechter Leistung in die Gesamtnote ein.
- Die quantitative Datenanalyse in Durchführung und Interpretation ist der Schwerpunkt dieser Arbeit. Zufälliges identisches Vorgehen, z.B. im R Code, ist sehr unwahrscheinlich und kann als **Plagiat** bewertet werden.
- Die Gesamtnote muss sich nicht als arithmetischer Mittelwert der Teilnoten ergeben.
- Es werden keine Teilnoten vergeben, sondern nur eine Gesamtnote wird vergeben.

3.7.2 Kennzahl der Modellgüte

Die Güte der Vorhersage wird anhand des *mittleren Absolutfehlers* (mae) bemessen:

$$\text{mae} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)$$

3.7.3 Notenstufen

Zur Vorhersagegüte: Die Vorhersagegüte eines einfachen Minimalmodells entspricht einer $4,0$, die eines Referenzmodells des Dozenten einer $2,0$.

Ihre Bewertung erfolgt entsprechend Ihrer Vorhersagegüte, d.h., sind Sie besser als das Referenzmodell erhalten Sie hier in diesem Teilaspekt eine bessere Note als $2,0$!

3.7.4 Bewertungsprozess

Der Gutachter legt im Nachgang der Prüfung alle Teilnehmis ihre jeweilige Wert der Kennzahl der Modellgüte offen. Außerdem werden die vorherzusagenden Daten veröffentlicht sowie die Grenzwerte für jede Notenstufe. Auf dieser Basis ist es allen Teilnehmis möglich, die Korrektheit Ihrer Note zu überprüfen.

3.8 Hinweise

Sie haben freie Methodenwahl bei der Modellierung und Vorverarbeitung. Nutzen Sie den Stoff wie im Unterricht gelernt; Sie können aber auch auf weitere Inhalte, die nicht im Unterricht behandelt wurden, zugreifen.

Eine Einführung in verschiedene Methoden gibt es z.B. bei Sebastian Sauer (2019): *Moderne Datenanalyse mit R*[1] aber auch bei Max Kuhn und Julia Silge (2021): *Tidy Modeling with R.*[2]. Die Bücher beinhalten jeweils Beispiele und Anwendung mit R.

Auch ist es Ihnen überlassen, welche Variablen Sie zur Modellierung heranziehen – und ob Sie diese eventuell vorverarbeiten, d.h., transformieren, zusammenfassen, Ausreißer bereinigen o.Ä.. Denken Sie nur daran, die Datentransformation, die Sie auf den Trainingsdaten durchführen, auch auf den Testdaten (Anwendungsdaten) durchzuführen.

Hinweise zur Modellwahl usw. gibt es auch in erwähnter Literatur, aber auch in vielen Büchern zum Thema Data-Science.

Alles, was Sie tun, Datenvorverarbeitung, Modellierung und Anwenden, muss transparent und reproduzierbar sein. Im Übrigen lautet die Aufgabe: Finden Sie ein Modell, von dem Sie glauben, dass es die Testdaten gut vorhersagt. $\widehat{y} = 42$ ist zwar eine schöne Antwort, trifft die Wirklichkeit aber leider nicht immer. Eine gute Modellierung auf den *Trainingsdaten* (z.B. hohes R^2) bedeutet nicht zwangsläufig eine gute Vorhersage (*Test-Set*).

[1] https://link.springer.com/book/10.1007/978-3-658-21587-3
[2] https://www.tmwr.org/

3.9 Formalia

1. Es sind nur Einzelarbeiten zulässig.

2. In der Analyse muss als Ausgangspunkt der vom/von der Dozenten/in bereitgestellten Datensatz genutzt werden.

3. Alle Analyseschritte bzw. alle Veränderungen an den Daten müssen im (eingereichten) *Analyseskript* nachvollziehbar (transparent und reproduzierbar) aufgeführt sein. Das Analyseskript ist als R-Skript, Rmd-Datei oder Rmd-Notebook-Datei abzugeben. Sie können die bereitgestellte Vorlage als Analyseskript nutzen (`Template-Dokumentation-Vorhersagemodellierung.Rmd`).

4. Das Analyseskript muss funktionstüchtig für den Prüfer sein: Alle Befehle müssen ohne Fehlermeldung durchlaufen (abgesehen von etwaiger Installation fehlender Pakete).

5. Es dürfen keine weiteren Informationen (Daten) als die vom Dozenten ausgegebenen verwendet werden. Sonstige Hilfe (z.B. von Dritten) ist ebenfalls unzulässig.

6. Nichtbeachtung der für dieses Modul formulierten Regeln kann zu Nichtbestehen oder Punkteabzug führen.

7. Der Schwerpunkt dieser Hausarbeit liegt auf der quantitativen Modellierung, der formale Anspruch liegt daher unter dem von anderen Hausarbeiten.

8. Es muss keine Literatur zitiert werden.

9. Ein ausgedrucktes Exemplar muss nicht abgegeben werden.

10. Während der Prüfungsphase werden keine inhaltlichen Fragen ("wie macht man nochmal eine Log-Transformation?") und keine technischen Fragen ("wie installiert man nochmal ein R-Paket?") beantwortet.

3.10 Wo finde ich Beispiele?

Eine Beispiel-Modellierung finden Sie in der Datei `Beispielanalyse-Prognose-Wettbewerb.Rmd`. Eine beispielhafte Vorlage (Template), die Sie als Richtschnur nutzen können, ist mit der Datei `Template-Vorhersagemodellierung.Rmd` hier bereitgestellt. Im Internet finden sich viele Fallstudien, von denen Sie sich inspirieren lassen können.

3.11 Plagiatskontrolle

Die eingereichten Arbeiten werden automatisiert auf Plagiate überprüft. Gibt es substanzielle Überschneidungen zwischen zwei (oder mehr) Arbeiten, werden alle diese Arbeiten mit *ungenügend* bewertet.

Chapter 4

Grundkonzepte

4.1 Was ist Data Science?

Es gibt mehrere Definitionen von *Data Science*, aber keinen kompletten Konsens. **?** definieren Data Science wie folgt (S. 4):

The science of extracting meaningful information from data

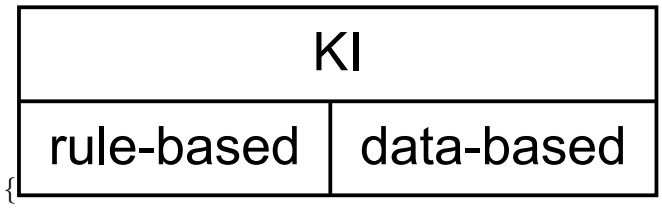
Auf der anderen Seite entgegen viele Statistiker: "Hey, das machen wir doch schon immer!".

Eine Antwort auf diesen Einwand ist, dass in Data Science nicht nur die Statistik eine Rolle spielt, sondern auch die Informatik sowie - zu einem geringen Teil - die Fachwissenschafte ("Domäne"), die sozusagen den Empfänger bzw. die Kunden oder den Rahmen stellt. Dieser "Dreiklang" ist in folgendem Venn-Diagramm dargestellt.

4.2 Was ist Machine Learning?

Maschinelles Lernen (ML), oft auch (synonym) als *statistisches Lernen* (statistical learning) bezeichnet, ist ein Teilgebiet der *künstlichen Intelligenz* (KI; artificial intelligence, AI) (**?**). ML wird auch als *data-based* bezeichnet in Abgrenzung von *rule-based*, was auch als "klassische KI" bezeichnet wird, vgl. Abb. **??**.

\begin{figure}[H]



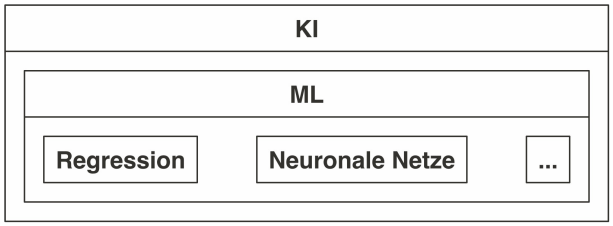
{

In beiden Fällen finden Algorithmen Verwendung. Algorithmen sind nichts anderes als genaue Schritt-für-Schritt-Anleitungen, um etwas zu erledigen. Ein Kochrezept ist ein klassisches Beispiel für einen Algorithmus.

Hier findet sich ein Beispiel für einen einfachen Additionsalgorithmus.

Es gibt viele ML-Algorithmen, vgl. Abb. **??**.

\begin{figure}[H]



{

4.2.1 Rule-based

Klassische (ältere) KI implementiert Regeln "hartverdrahtet" in ein Computersystem. Nutzer füttern Daten in dieses System. Das System leitet dann daraus Antworten ab.

Regeln kann man prototypisch mit *Wenn-Dann-Abfragen* darstellen:

```
lernzeit <- c(0, 10, 10, 20)
schlauer_nebensitzer <- c(FALSE, FALSE, TRUE, TRUE)

for (i in 1:4) {
  if (lernzeit[i] > 10) {
    print("bestanden!")
  } else {
    if (schlauer_nebensitzer[i] == TRUE) {
      print("bestanden!")
    } else print("Durchgefallen!")
  }
}
```

```
## [1] "Durchgefallen!"
## [1] "Durchgefallen!"
## [1] "bestanden!"
## [1] "bestanden!"
```

Sicherlich könnte man das schlauer programmieren, vielleicht so:

```
## # A tibble: 4 x 3
##   lernzeit schlauer_nebensitzer bestanden
##      <dbl> <lgl>                <lgl>
## 1        0 FALSE                FALSE
## 2       10 FALSE                FALSE
## 3       10 TRUE                 TRUE
## 4       20 TRUE                 TRUE
```

4.2.2 Data-based

ML hat zum Ziel, Regeln aus den Daten zu lernen. Man füttert Daten und Antworten in das System, das System gibt Regeln zurück.

? definieren ML so: Nehmen wir an, wir haben die abhängige Variable Y und p Prädiktoren, $X_1, X_2, \ldots, X_p$. Weiter nehmen wir an, die Beziehung zwischen Y und $X = (X_1, X_2, \ldots, X_p)$ kann durch eine Funktion f beschrieben werden. Das kann man so darstellen:

$$Y = f(X) + \epsilon$$

ML kann man auffassen als eine Menge an Verfahren, um f zu schätzen.

Ein Beispiel ist in Abb. 4.1 gezeigt (**?**).

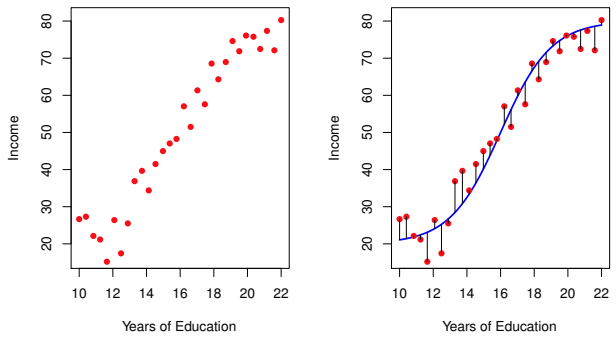


Figure 4.1: Vorhersage des Einkommens durch Ausbildungsjahre

Natürlich kann X mehr als eine Variable beinhalten, vgl. Abb. 4.2 (**?**).

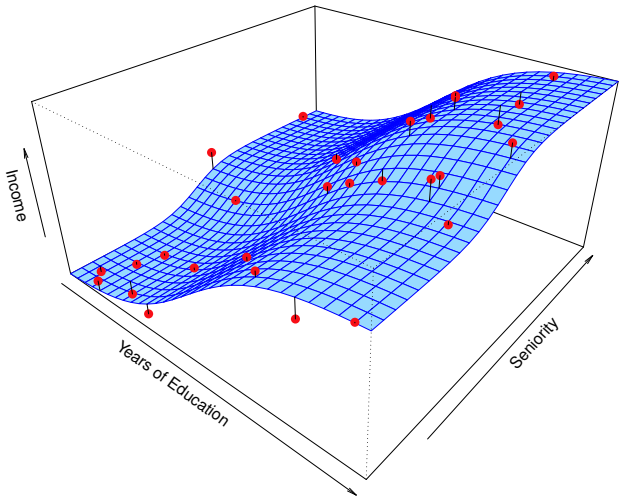
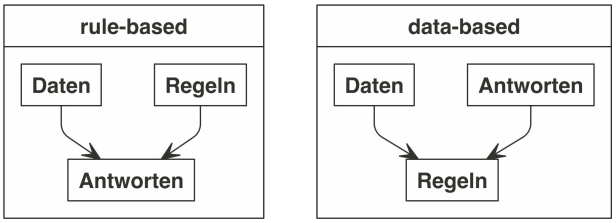


Figure 4.2: Vorhersage des Einkommens als Funktion von Ausbildungsjahren und Dienstjahren

Anders gesagt: traditionelle KI-Systeme werden mit Daten und Regeln gefüttert und liefern Antworten. ML-Systeme werden mit Daten und Antworten gefüttert und liefern Regeln zurück, vgl. Abb. **??**.

\begin{figure}[H]



{

4.3 Modell vs. Algorithmus

4.3.1 Modell

Ein Modell, s. Abb. 4.3 (**?**)!

Figure 4.3: Ein Modell-Auto

Wie man sieht, ist ein Modell eine vereinfachte Repräsentation eines Gegenstands.

Der Gegenstand definiert (gestaltet) das Modell. Das Modell ist eine Vereinfachung des Gegenstands, vgl. Abb. 4.4.

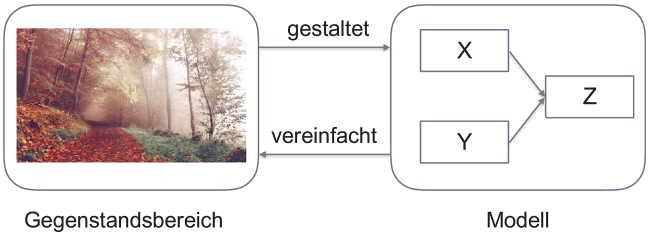


Figure 4.4: Gegenstand und Modell

Im maschinellen Lernen meint ein Modell, praktisch gesehen, die Regeln, die aus den Daten gelernt wurden.

4.3.2 Beispiel für einen ML-Algorithmus

Unter einem ML-Algorithmus versteht man das (mathematische oder statistische) Verfahren, anhand dessen die Beziehung zwischen X und Y "gelernt" wird. Bei **?** (S. 9) findet sich dazu ein Beispiel, das kurz zusammengefasst etwa so lautet:

Beispiel eines Regressionsalgorithmus

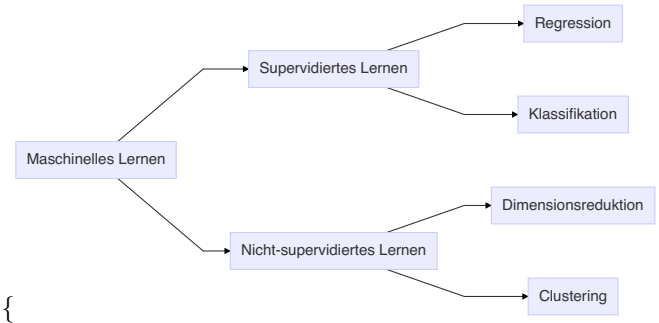
1. Setze Gerade in die Daten mit $b_0 = \hat{y}, b_1 = 0$
2. Berechne $MSS = \sum(y_i - \hat{y}_i)^2$
3. "Drehe" die Gerade ein bisschen, d.h. erhöhe $b_1^{neu} = b_1^{alt} + 0.1$
4. Wiederhole 2-3 solange, bis $MSS <$ Zielwert

Diesen Algorithmus kann man "von Hand" z.B. mit dieser App durchspielen.

4.4 Taxonomie

Methoden des maschinellen Lernens lassen sich verschiedentlich gliedern. Eine typische Gliederung unterscheidet in *supervidierte* (geleitete) und *nicht-supervidierte* (ungeleitete) Algorithmen, s. Abb. **??**.

\begin{figure}[H]

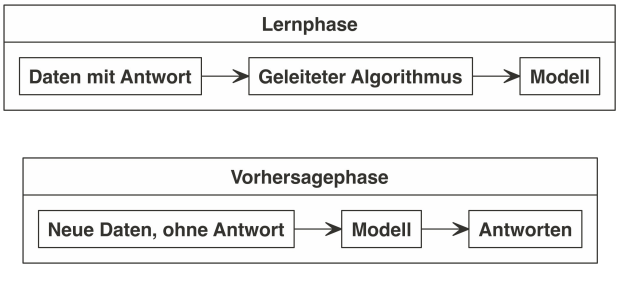


{

4.4.1 Geleitetes Lernen

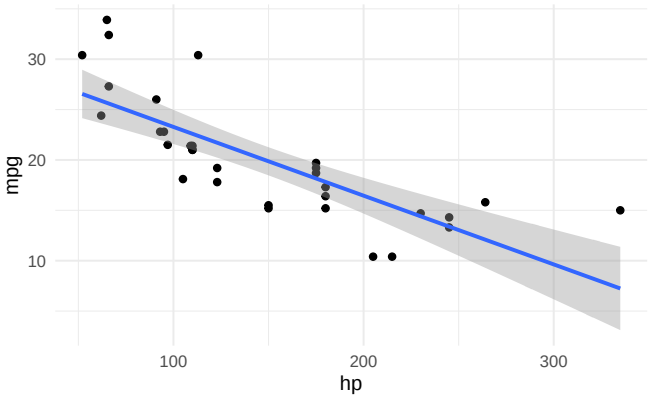
Die zwei Phasen des geleiteten Lernens sind in Abb. **??** dargestellt.

\begin{figure}[H]



{

4.4.1.1 Regression: Numerische Vorhersage



Die Modellgüte eines numerischen Vorhersagemodells wird oft mit (einem der) folgenden *Gütekoeffizienten* gemessen:

- Mean Squared Error (Mittlerer Quadratfehler):

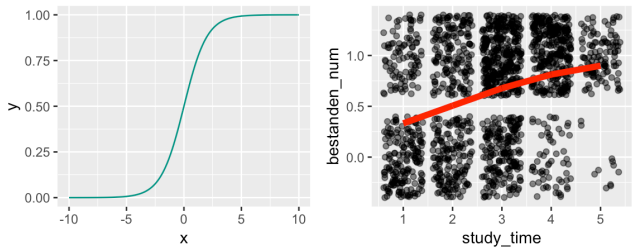
$$MSE := \frac{1}{n} \sum (y_i - \hat{y}_i)^2$$

- Mean Absolute Error (Mittlerer Absolutfehler):

$$MAE := \frac{1}{n} \sum |(y_i - \hat{y}_i)|$$

Wir sind nicht adaran interessiert die Vorhersagegenauigkeit in den bekannten Daten einzuschätzen, sondern im Hinblick auf neue Daten, die in der Lernphase dem Modell nicht bekannt waren.

4.4.1.2 Klassifikation: Nominale Vorhersage



Die Modellgüte eines numerischen Vorhersagemodells wird oft mit folgendem *Gütekoeffizienten* gemessen:

- Mittlerer Klassifikationfehler e:

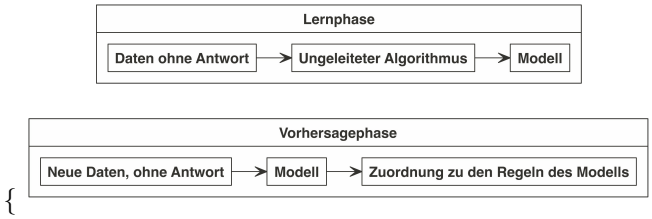
$$e := \frac{1}{n} I(y_i \neq \hat{y}_i)$$

Dabei ist I eine Indikatorfunktion, die `1` zurückliefert, wenn tatsächlicher Wert und vorhergesagter Wert identisch sind.

4.4.2 Ungeleitetes Lernen

Die zwei Phasen des ungeleiteten Lernens sind in Abb. **??** dargestellt.

\begin{figure}[H]



{

Ungeleitetes Lernen kann man wiederum in zwei Arten unterteilen, vgl. Abb. 4.5:

1. Fallreduzierendes Modellieren (Clustering)
2. Dimensionsreduzierendes Modellieren (z.B. Faktorenanalyse)

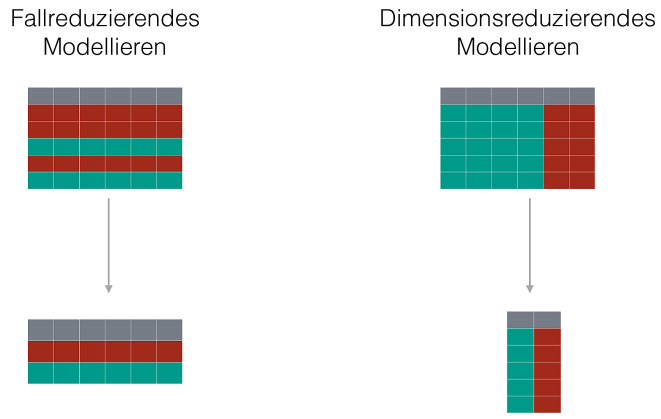
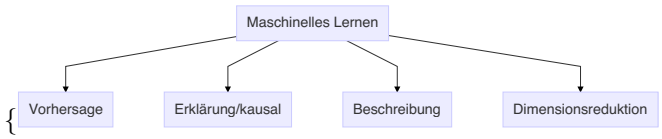


Figure 4.5: Zwei Arten von ungeleitetem Modellieren

4.5 Ziele des ML

Man kann vier Ziele des ML unterscheiden, s. Abb. **??**.

\begin{figure}[H]



Vorhersage bezieht sich auf die Schätzung der Werte von Zielvariablen (sowie die damit verbundene Unsicherheit). *Erklärung* meint die kausale Analyse von Zusammenhängen. *Beschreibung* ist praktisch gleichzusetzen mit der Verwendung von deskriptiven Statistiken. *Dimensionsreduktion* ist ein Oberbegriff für Verfahren, die die Anzahl der Variablen (Spalten) oder der Beobachtungen (Zeilen) verringert.s

Wie "gut" ein Modell ist, quantifiziert man in verschiedenen Kennzahlen; man spricht von Modellgüte oder *model fit*. Je schlechter die Modellgüte, desto höher der *Modellfehler*, vgl. Abb. 4.6.

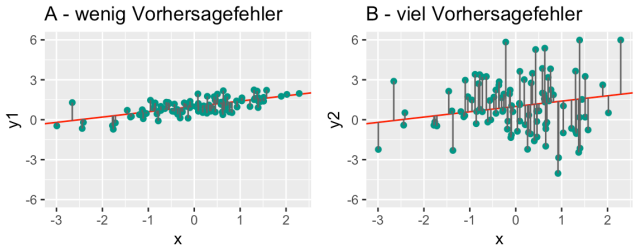


Figure 4.6: Wenig (links) vs. viel (rechts) Vorhersagefehler

Die Modellgüte eines Modells ist nur relevant für *neue Beobachtungen*, an denen das Modell nicht trainiert wurde.

4.6 Über- vs. Unteranpassung

Overfitting: Ein Modell sagt die Trainingsdaten zu genau vorher - es nimmt Rauschen als "bare Münze", also fälschlich als Signal. Solche Modelle haben zu viel *Varianz* in ihren Vorhersagen.

Underfitting: Ein Modell ist zu simpel (ungenau, grobkörnig) - es unterschlägt Nuancen des tatsächlichen Musters. Solche Modelle haben zu viel *Verzerrung* (Bias) in ihren Vorhersagen.

Welches der folgenden Modelle (B,C,D) passt am besten zu den Daten (A), s. Abb. 4.7, vgl. (**?**), Kap. 15.

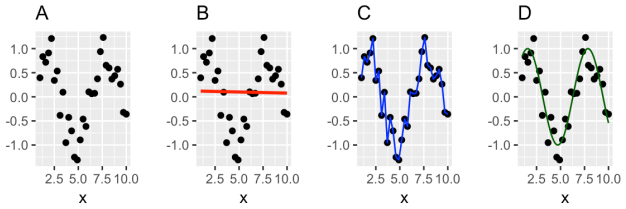


Figure 4.7: Over- vs. Underfitting

Welches Modell wird wohl neue Daten am besten vorhersagen? Was meinen Sie?

Modell D zeigt sehr gute Beschreibung ("Retrodiktion") der Werte, anhand derer das Modell trainiert wurde ("Trainingsstichprobe"). Wird es aber "ehrlich" getestet, d.h. anhand neuer Daten ("Test-Stichprobe"), wird es vermutlich *nicht* so gut abschneiden.

Es gilt, ein Modell mit "mittlerer" Komplexität zu finden, um Über- und Unteranpassung in Grenzen zu halten. Leider ist es nicht möglich, vorab zu sagen, was der richtige, "mittlere" Wert an Komplexität eines Modells ist, vgl. Abb. 4.8 aus (**?**).

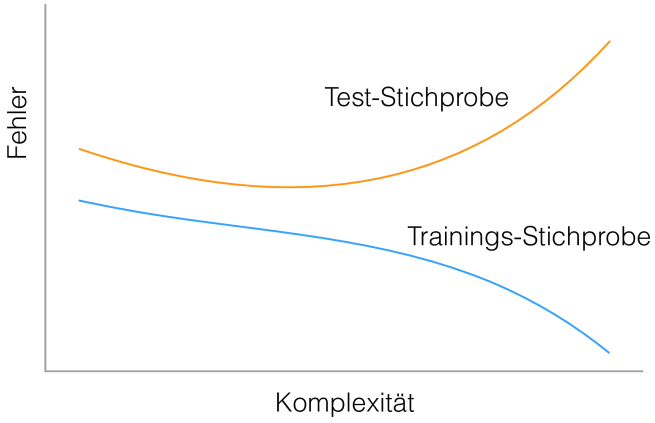


Figure 4.8: Mittlere Modellkomplexität führt zur besten Vorhersagegüte

4.7 No free lunch

from Imgflip Meme Generator

Wenn f (die Beziehung zwischen Y und X, auch *datengenerierender Prozess* genannt) linear oder fast linear ist, dann wird ein lineare Modell gute Vorhersagen liefern, vgl. Abb. 4.9, dort zeigt die schwarze Linie den "wahren Zusammenhang", also f an. In orange sieht man ein lineares Modell, in grün ein hoch komplexes Modell, das sich in einer "wackligen" Funktion - also mit hoher Varianz - niederschlägt. Das grüne Modell könnte z.B. ein Polynom-Modell hohen Grades sein, z. B. $y = b_0 + b_1 x^{10} + b_2 x^9 + ... + b_1 1 x^1 + \epsilon$. Das lineare Modell hat hingegen wenig Varianz und in diesem Fall wenig Bias. Daher ist es für dieses f gut passend. Die grüne Funktion zeigt dagegen Überanpassung (overfitting), also viel Modellfehler (für eine Test-Stichprobe).

Die grüne Funktion in Abb. 4.9 wird neue, beim Modelltraining unbekannte Beobachtungen (y_0) vergleichsweise schlecht vorhersagen. In Abb. 4.10 ist es umgekehrt.

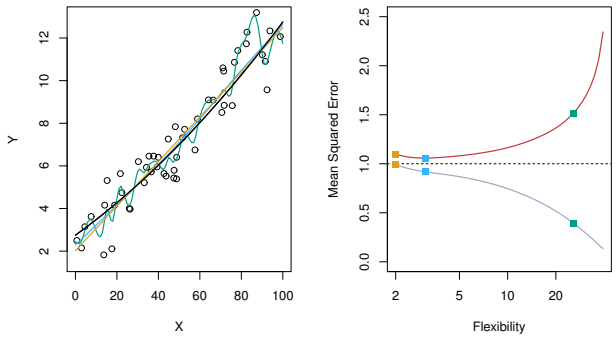


Figure 4.9: Ein lineare Funktion verlangt ein lineares Modell; ein nichtlineares Modell wird in einem höheren Vorhersagefehler (bei neuen Daten!) resultieren.

Betrachten wir im Gegensatz dazu Abb. 4.10, die (in schwarz) eine hochgradig *nichtlineare* Funktion f zeigt. Entsprechend wird das lineare Modell (orange) nur schlechte Vorhersagen erreichen - es hat zu viel Bias, da zu simpel. Ein lineares Modell wird der Komplexität von f nicht gerecht, Unteranpassung (underfitting) liegt vor.

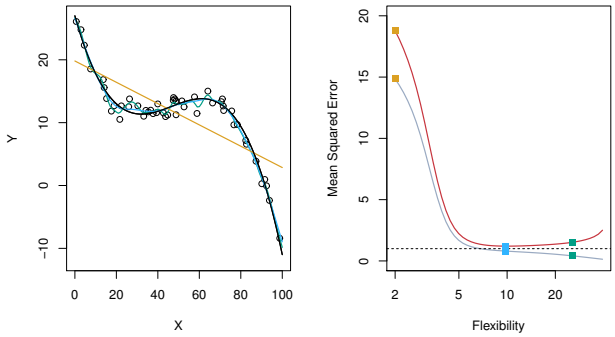


Figure 4.10: Eine nichtlineare Funktion (schwarz) verlangt eine nichtlineares Modell. Ein lineares Modell (orange) ist unterangepasst und hat eine schlechte Vorhersageleistung.

4.8 Bias-Varianz-Abwägung

Der Gesamtfehler E des Modells ist die Summe dreier Terme:

$$E = (y - \hat{y}) = \text{Bias} + \text{Varianz} + \epsilon$$

Dabei meint ϵ den *nicht reduzierbaren Fehler*, z.B. weil dem Modell Informationen fehlen. So kann man etwa auf der Motivation von Studentis keine perfekte Vorhersage ihrer Noten erreichen (lehrt die Erfahrung).

Bias und Varianz sind Kontrahenten: Ein Modell, das wenig Bias hat, neigt tendenziell zu wenig Varianz und umgekehrt, vgl. Abb. 4.11 aus (**?**).

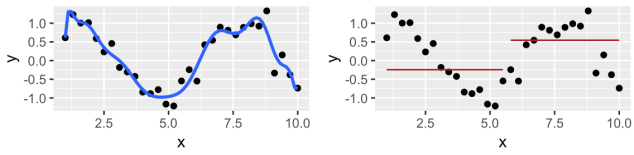


Figure 4.11: Abwängung von Bias vs. Varianz

Bibliography

Baumer, B. S., Kaplan, D. T., and Horton, N. J. *Modern Data Science with R (Chapman & Hall/CRC Texts in Statistical Science)*. Chapman and Hall/CRC.

James, G., Witten, D., Hastie, T., and Tibshirani, R. *An introduction to statistical learning: with applications in R*. Springer texts in statistics. Springer, second edition edition.

Rhys, H. (2020). *Machine Learning with R, the tidyverse, and mlr*. Manning publications, Shelter Island, NY. OCLC: on1121083327.

Sauer, S. *Moderne Datenanalyse mit R: Daten einlesen, aufbereiten, visualisieren und modellieren*. FOM-Edition. Springer, 1. auflage 2019 edition.

Silge, J. and Kuhn, M. (2022). *Tidy Modeling with R*.

Spurzem, L. *VW 1303 von Wiking in 1:87*.