ÜK 259 Machine Learning

Datenschutz und Recht

Datenschutzgesetz

Drei arten von Daten: Personendaten, Besonders schützenswerte Personendaten, Persönlichkeitsprofile

Bestimmt vs Bestimmbar: Wenn z.B. eine Tabelle von Namen zu ID (generiert) umbenennt wird, wird der Datensatz bestimmbar, vorher war er bestimmt. (Bestimmt ist direkt anhand den Daten erkennbar)

Besonders Schützenswert und Persönlichkeitsprofile (strengere Bedingungen als "normalen" Personendaten), dürfen nicht:

- Ohne Rechtfertigung an Dritte bekannt gegeben werden
- Gesammelt werden ohne die Person klar darüber zu informieren
- Vom Bund bearbeitet werden ohne das ein Gesetz dies vorsieht

Allgemeine Regelungen:

- Bearbeitung hat nach Treu und Glauben zu erfolgen und muss verhältnismässig sein
- Pers.-Daten dürfen nur mit Zweck, welcher bei der Beschaffung angegeben wurde, bearbeitet werden
- Beschaffung Pers.-Daten; Zweck ihrer Bearbeitung muss für betroffene Person erkennbar sein
- Ist für Bearbeitung von Pers.-Daten Einwilligung der betroffenen Personen erforderlich, muss diese nach anemessener Information freiwillig erfolgen.

Datensicherheit

Beschreibt wie sicher die Daten vor Diebstahl, Ausfall und Verlust sind.

Code of Ethics

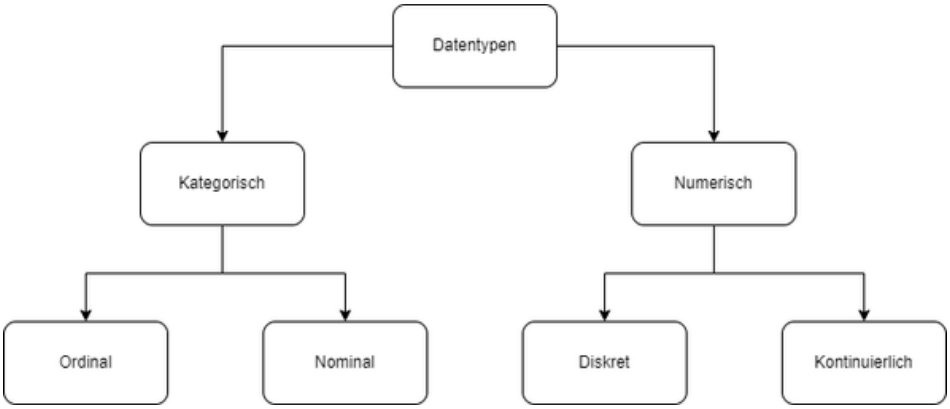
Beschreibt wie Algorithmen und Daten-Modelle fair und Transparent entwickelt werden können.

Asimov's Law of Robotics

- Roboter darf kein menschliches Wesen (wissentlich) verletzen (allgemein wissentlich keinen Schaden einem menschlichen Wesen zugeführt wird.
- Roboter muss den ihm von einem Menschen gegebenen Befehlen gehorchen.

- Ein Roboter muss seine Existenz beschützen, solange dieser Schutz nicht mit Regel eins oder zwei kollidiert.

Datentypen und Umgang



Kategorische Datentypen

Daten welche eine gewisse Charakteristik besitzen. (Haarfarbe, Sprache, Lieblingsfilmgenre.

Ordinal: Haben Ordnung, Werte sind voneinander unterschiedlich und diskret, können jedoch in Reihenfolge angeordnet werden. (XS, S, M, L, XL)

Nominal: Keine Ordnung, Werte sind voneinander unterschiedlich und diskret, können nicht Eingeordnet werden. Sprachauswahl (Deutsch, Englisch, Französisch, Italienisch)

Numerische Datentypen

Nummern

Diskret: Daten sind Eindeutig voneinander getrennt. Können nicht gemessen, nur gezählt werden. Z.B. Anzahl Lernende in Klasse

Kontinuierlich: Messbare Daten. Grösse einer Person oder die Temeperatur.

Datenquellen

Um Datenmodelle zu erstellen benötigt man Daten aus einer Datenquelle.

Lerndatasets: Daten, welche auf Lernplattformen zur Verfügung gestellt werden

Open Data: Frei zugängliche Datensets aus Reierung, Forschun und Entwicklung

Data Science Process / CRISP-DM

Cross Industry Standard Process for Data Mining; Hilft vollständige und gute Datenmodelle zu bauen

6 Phasen von CRISP-DM

Business Understanding (Aufgabendefinition)

Dieser Schritt wird mit dem Kunden oder Fachanwendern zusammen gemacht.

- Problemstellung, welche mit Daten gelöst werden soll?
- Welche Prozesse betrifft diese Problemstellung
- Welche Daten sind vorhanden (und nicht)
- Was soll das Modell ausgeben?
- Wo soll das Resultat des Modells hin?

Data Understanding (Auswahl der relevanten Datenbestände)

Verfügbare Daten werden genau analysiert.

- Welche Daten sind zur Verfügung
- Welche Daten müssen noch organisiert werden
- Explorative Datenanalyse mit vorhandenen Daten (EDA)

Data Preparation (Datenaufbereitung)

Anhander der Explorativen Datenanalyse werden Daten aufbereitet.

Modeling (Auswahl und Anwendung von Data Mining Methoden)

Mehrere Modelle und Parameter werden ausprobiert. Das Modell wird Erstellt und aufbereitet.

Evaluation (Bewertung und Interpretation der Ergebnisse)

Erstellte Modelle werden anhand passenden Kriterien evaluiert und getestet

Deployment (Anwendung der Ergebnisse)

Modell wird in eine Datapipeline gepackt und in der Realität angewendet.

EDA Explorative Datenanalyse

Ziel hier ist, ein Grundverständnis über den Datenbestand zu schaffen.

Hier werden **Anomalien** und **Ausreisser** erkennt.

Es werden **Hypothesen** augestellt

count: Wir sehen, dass wir 270 komplette Zeilen (rows) haben. Denn für jeden der Variablen / Spalte hat es 270 Werte gezählt.

mean: Mean ist der Durchschnitt pro Variable. Bei gewissen Variablen macht diese Messung keinen Sinn. z.B. interessiert es uns wahrscheinlich nicht was der Durchschnitt der "replicate" also der Versuchsgänge ist. Jedoch interessiert uns die Durchschnittstemperatur sehr wohl.

std: std bedeutet Standardabweichung (standard deviation). Diese beschreibt um wie viel sich ein Datenpunkt vom Durchschnitt durchschnittlich abweicht.

min: Min zeigt den Minimum Datenpunkt pro Variable.

25%: Zeigt wo der erste viertel der Daten endet.

50%: Zeigt wo die Mitte der Daten endet (Auch der Median)

75%: Zeigt wo drei viertel der Daten endet.

max: Zeigt wo 100% der Daten endet, der Maximalwert.

Korrelation

Die Korrelation kann Werte von -1 bis +1 annehmen

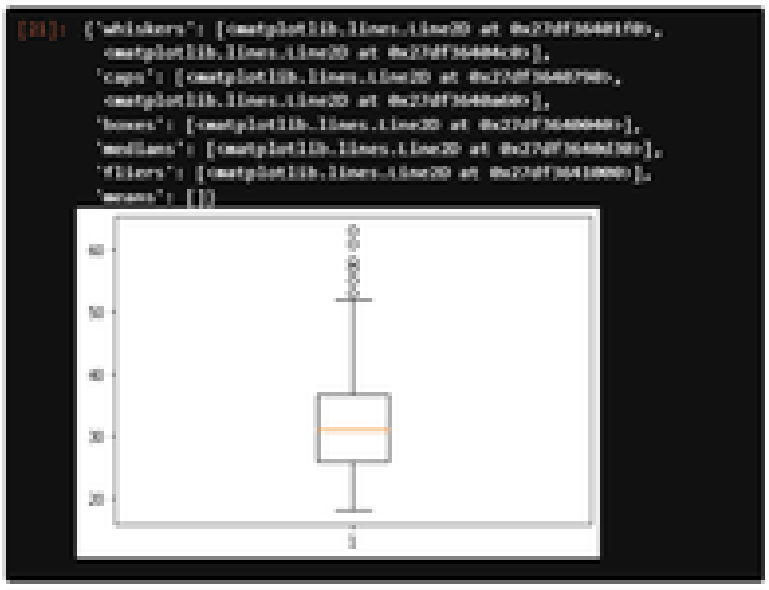
Ist eine Beziehung zwischen zwei oder mehreren Merkmalen, Zustände, Funktionen. Die Beziehung muss keine kausale Beziehung sein.

Unvariate Visualisierungen

Histogramm

Zeigt die Häufigkeit eines Datenpunkts an.

Boxplot



Ein Boxplot ist das ideale Tool für EDA. Es beschreibt die Daten schnell und visuell.

- Der unterste Strich beschreibt das Minimum der Daten.
- Der Anfang der Box ist die 25% Marke.
- Der rote Strich der Median.
- Das Ende der Box die 75% Marke.
- Der letzte Strich das Maximum ohne mögliche Ausreisser.
- Die Punkte sind mögliche Ausreisser.

Feature Engineering

Train / Test / Val Split

Ideal: 70% Trainingsdaten / 30% Testdaten

Feature Engineering wird NUR auf Trainingsdaten gemacht

Wieso splitten?

Bei dem Bau eines Modells werden Trainings- und Testdaten benötigt. Diese kommen meist aus dem **gleichen Datenset**.

```
Grundsätzlich hätten wir hier schon ein Modell welches wir anwenden können.
Jedoch besteht hier die Gefahr des Overfitting. Das Modell ist nur auf die
Trainingsdaten trainiert und wurde noch nicht in der "Realität" getestet.
```

Darum wird das Modell als nächstes auf dem Test Datensatz angewendet. Nun kann die Genauigkeit des Modells ausgewiesen werden.

Worauf ist zu achten?

Es ist sehr wichtig, dass man **gleichwertige Datensätze** hat und diese vor dem Trainieren **überprüft** hat.

Zu wenig Daten über Kalssifikation, kann das Modell unkorrekt werden. z.B. Viele Hunde, wenig Katzen in Daten. Möglichkeit ist gross dass er keine Katzen erkennt.

Algorithmen

fit(X, Y)

```
fit() dient dazu, das Model mit den Vektoren x und dem zugehörigen Resultats-
Vektor Y zu trainieren. Das Modell lernt also anhand der vorhandenen
Resultate.
```

predict(X)

```
predict() dient dazu, anhand der Trainingsdaten aus dem fit()-Schritt, neue
Y-Werte mit unbekannten X-Vektoren zu bestimmen.
```

Klassifikation

Die Klassifikatoren dienen dazu, Datensätzen in unterschiedliche Klassen einzuteilen.

k-Nearest-Neighbours (kNN)

Das Prinzip von k-Nearest-Neighbors (KNN) ist, dass die k nächsten Nachbarn eines Punktes betrachtet werden. Der Punkt wird der Klasse zugewiesen, welche am meisten bei den k nächsten Nachbarn vorkommt.

Decision Tree

Im Decision Tree wird verschachtelt Entschieden, in welche Klasse ein Wert getan werden soll. Es wird im obersten Knoten angefangen und in jedem Knoten wird anhand einer Bedingung ein Ast gewählt. Wenn ein Knoten keine Kinder hat, genannt Blatt, wird die Klasse des Knotens gewählt. Diese entspricht jeweils der Mehrheit der Werte in diesem Knoten.

Lineare Regression

Regressoren dienen dazu, numerische Werte einzuschätzen.

Das Prinzip der linearen Regression ist es, eine lineare Funktion durch die gegebenen Punkte zu ziehen. Das Ziel der Funktion ist es, im Durchschnitt über alle Punkte den vertikalen Abstand von Punkt zu Funktion zu minimieren.

Decision Tree

Das Prinzip des Decision Trees der Regression ist dasselbe wie bei der Klassifikation, jedoch wird anstatt einer Klasse ein bestimmter Wert ausgewählt. So kann ein Knoten sich für den Durchschnitt aller Punkte, welche in diesem Knoten landen, entscheiden.

Feature Space

Mathematischer Raum, der ein Objekt durch dessen Messwerte in Bezug auf dessen besondere Eigenschaften bzw. Merkmale bestimmt.

- Die Target Variable Y ist die Spalte, die wir einschätzen möchten
- Der Feature Space X ist eine Zahl von Spalten, genannt Feature, welche wir für diese Schätzung verwenden möchten.
    - Es wird oft X=[x1,x2,...,xn] geschrieben, wobei jedes xi ein Feature ist.
    - Der Feature Space hat genau so viele Dimensionen wie Features

Daten aufbereiten

Daten müssen meistens aufbereitet werden, bevor diese in einem Modell verarbeitet werden können. Gründe:

- **Falsch geschriebene Werte**: “Schule” und “Shule”, Modell behandelt diese einzeln als unterschiedliche
- **Werte in falschen Feldern**: Im Feld “Stadt” den Wert “Schweiz”
- **Irregularitäten**: Im Feld “Lohn” mehrheit als CHF eingetragen, die minderheit als EUR
- **Fehlende Werte (Null Values):** Machine Learning Modelle sind reine Mathematik. Somit kann ein Modell nur schlecht mit NULL umgehen.
- **Duplikate**: Zeilen wurden doppelt erfasst, erhalten mehr bedeutung im Modell.

Fehlende Werte behandeln

- Spalten löschen
    - `dataframe.drop('Age', axis=1)`
- Zeilen löschen
    - `dataframe = dataframe[dataframe['Age'].notnull()]`
- Zeilen ersetzen
    - `mean = dataframe['Age'].mean()`
    - `dataframe['Age']Da.fillna(mean, inplace=True)`

Duplikate entfernen

```
# Duplikate anzeigen
dataframe[dataframe.duplicated() == True]

# Duplikate löschen
dataframe.drop_duplicates()
```

Ausreisser erkennen

Können logisch ermittelt werden oder mathematisch. Mathematisch wäre z.B. mit einem Boxplot

Daten modellieren

Modellieren

Im generellen ist ein Modell eine Abbildung der Realität. Dabei meist eine vereinfachte Abbildung.

Bei Machine Learning Modellen sind die meisten mathematische Modelle.

Mathematische Modelle haben einen Input und einen Output. Das Modell selber liegt dazwischen und rechnet was mit dem Input und dem Output zu erhalten ist. ==> **f(x) = y**

```
Das Resultat der Funktion bei welcher x (Input Variable / Unabhängige
Variable) übergeben wird hat als Resultat y (Output Variable / Abhängige
Variable)
```

Feature Selection

Beschreibt die Auswahl von Inputvariablen welche die Output-Variablen beeinflussen. Diese werden meist im Prozessschritt EDA schon eingeschränkt.

Feature Engineering

Meistens hat man nicht zu viele Features sondern zu wenig. Hierbei müssen oft Features neu erzeugt werden. Bei einem Spam-Filter-Algorithmus für E-Mails haben wir bis jetzt folgende Features:

- E-Mail Absender
- Betreff
- Text

Dies reicht uns noch nicht um eine E-Mail als Spam zu klassifizieren. Nützlich hier wäre aus diesen bestehenden Features folgendes zu extrahieren:

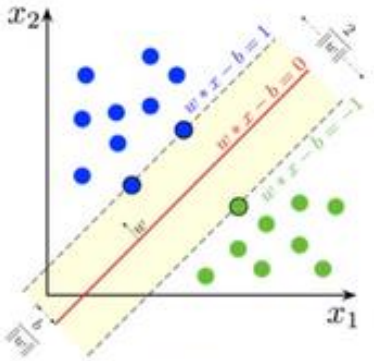
- Domäne des Absenders

- Kommen Stichwörter wie "Bitcoin", "Darlehen", "Lotto", ... vor? (0 oder 1)

Algorithmen → Support Vector Machines

Support Vector Machines sind Algorithmen, welche durch Iterationen die beste Funktion eines Modells ermitteln.

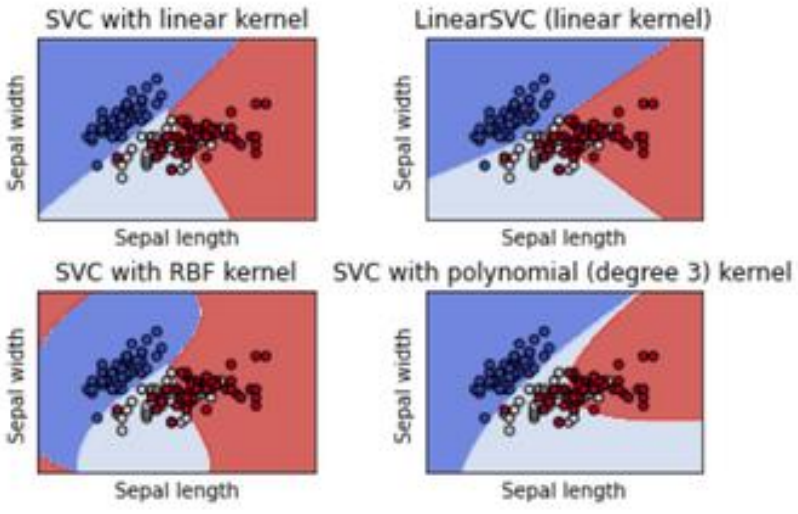
Dies geschieht mit sogenannten “Support Vectors”



Quelle: Wikipedia

Wir sehen hier, dass mit den drei Supportvektoren (zwei blaue, ein grüner Punkt) eine Trennlinie definiert wurde. Ziel von SVM ist hierbei, dass der Abstand (Margin) möglichst gering ist. Auch hier kann es aufgrund von Ausreissern zu Fehlklassifikationen kommen, diese müssen aber (wie bei anderen Algorithmen auch) gewissermassen in Kauf genommen werden.

Kernel Tuning



SVM Kernel (Auswahl)

Kernel haben unterschiedliche Funktionsabbildungen im Hintergrund. Anhand der Grafik erkennt man auch, dass diese sich in der Trennung unterscheiden.

SVC mit linearen Kernel

Bildet die Standartimplementierung der Support Vector Classification ab.

LinearSVC

Wird mittels svc.LinearSVC instanziert und unterscheidet sich zum SVC mit linearem Kernel dahingehend, dass LinearSVC One-vs-All Ansatz verwendet.

SVC mit RBF Kernel

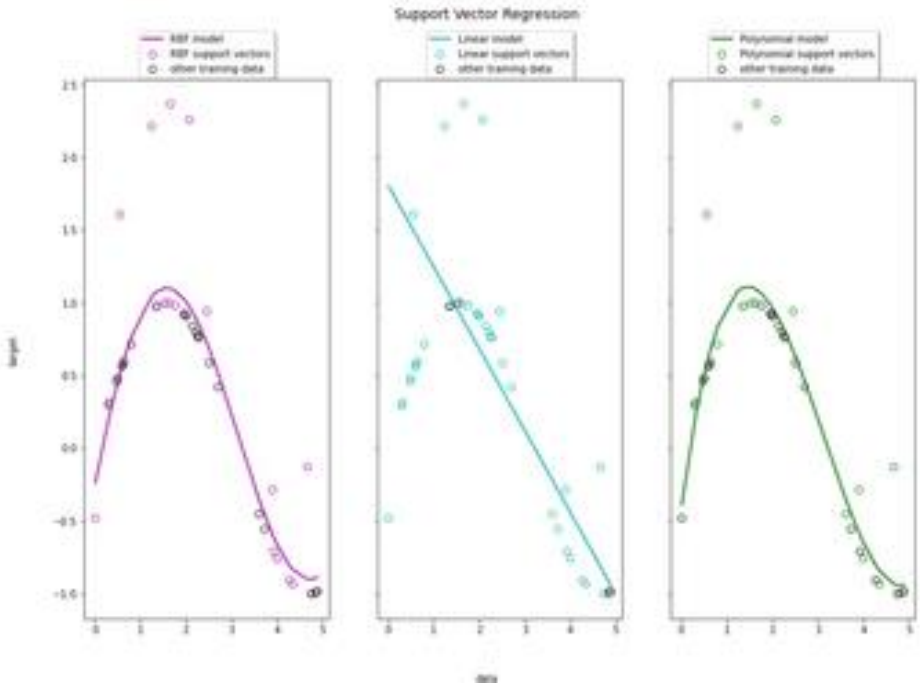
Verwendet die Radiale Basisfunktion, welche für Approximationen gängig ist. Bei vielen Features sind RBFs robust.

SVC mit polynominalen Kernel

Hier können polynomale Funktionen mit Machine Learning abgebildet werden.

Kernel Tuning

```
Wie bei den Support Vector Classifications können die Regressions auch mit
verschiedenen Kernel trainiert werden.
```



Evaluation

Beurteilung von Klassifizierungen

Wahrheitsmatrix

Die Wahrheitsmatrix, auch Konfusionsmatrix genannt, dient dazu, die Vorhersageergebnisse eines Klassifikators zu analysieren. In diesem Fall wird angenommen, dass der Klassifikator zwischen Katzen und Hunden unterscheidet und das Ziel ist, Katzen von Hunden zu unterscheiden. Die Matrix berücksichtigt vier mögliche Ergebnisse:

1. Richtig positiv (True Positive): Das Tier ist eine Katze und wurde als Katze erkannt.
2. Falsch negativ (False Negative): Das Tier ist eine Katze, wurde aber als Hund erkannt.
3. Falsch positiv (False Positive): Das Tier ist ein Hund, wurde aber als Katze erkannt.
4. Richtig negativ (True Negative): Das Tier ist ein Hund und wurde als Hund erkannt.

Die Zielsetzung besteht darin, möglichst viele "True Positives" und "True Negatives" zu erzielen, da diese bedeuten, dass die Klassifikation korrekt war. "False Positives" und "False Negatives" sollten vermieden werden, da sie Fehler in der Klassifikation darstellen. In anderen Kontexten werden diese Begriffe auch als true positive, false positive, false negative und true negative bezeichnet. Die Wahrheitsmatrix hilft dabei, die Leistung des Klassifikators zu bewerten und zu verbessern.

Statistische Grundkriterien

Sensitivität/Recall und Falsch-Negativ-Rate

Die Sensitivität, auch als Recall bezeichnet, misst die Wahrscheinlichkeit, dass in unserem Beispiel eine Katze als Katze korrekt klassifiziert wird. Dabei werden Katzen, die fälschlicherweise als Hunde erkannt werden, als Falsch-Negativ bezeichnet.

Specifity und Falsch-Positiv-Rate

Die Spezifität, auch als Richtig-Negativ oder kennzeichnende Eigenschaft bezeichnet, misst die Wahrscheinlichkeit, dass in unserem Beispiel ein Hund als Hund korrekt klassifiziert wird. Dabei werden Hunde, die fälschlicherweise als Katzen erkannt werden, als Falsch-Positive bezeichnet.

Positiver und negativer Vorhersagewert

1. Sensitivität und Spezifität: Sensitivität und Spezifität sind wichtige Kennzahlen zur Bewertung der Fähigkeit eines Klassifikators, Katzen zu erkennen. Sie messen, wie gut der Klassifikator wahre positive und wahre negative Ergebnisse liefert.

2. Positiver Vorhersagewert (Precision): Der positive Vorhersagewert bezieht sich auf den Anteil der als Katzen klassifizierten Tiere, die tatsächlich Katzen sind. Er misst die Genauigkeit der Katzenidentifikation und steht in Verbindung mit der Sensitivität.
3. Negativer Vorhersagewert: Der negative Vorhersagewert gibt an, wie viele Hunde als Hunde erkannt werden. Er misst die Fähigkeit des Klassifikators, Hunde korrekt zu identifizieren und steht in Beziehung zur Spezifität.
4. Falscherkennungsrate und Falschauslassungsrate: Das Komplement des positiven Vorhersagewerts ist die Falscherkennungsrate, während das Komplement des negativen Vorhersagewerts die Falschauslassungsrate ist. Diese Raten messen, wie oft der Klassifikator fälschlicherweise Katzen oder Hunde identifiziert.
5. Zusammenhang: Die positiven und negativen Vorhersagewerte addieren sich nicht zu 1 (oder 100%), da sie verschiedene Aspekte der Klassifikatorleistung betrachten. In der Medizin sind diese Maßnahmen wichtig, um sicherzustellen, dass Tests zur Krankheitserkennung zuverlässige Ergebnisse liefern.

Korrekt- und Falschklassifikationsrate

Abschliessend ist wichtig zu bestimmen, welcher Anteil aller Objekte auch korrekt klassifiziert wurde. In unserem Beispiel also, welcher Anteil Katzen auch wirklich als Katzen klassifiziert wurde. Andere Begriffe hierbei sind die Vertrauenswahrscheinlichkeit oder die Treffergenauigkeit. Die restlichen Klassifikationen bilden die Falschklassifizierungsrate.

Over-/Underfitting

Ein sehr wichtiges Konzept in Machine Learning ist das Over- / Underfitting. Eine grosse Gefahr besteht, dass wir das Modell zu fest an den bestehenden Daten anpassen und so ein Overfitting entsteht. Das heisst, unser Modell ist zwar perfekt für die Daten die wir haben, jedoch passt es bei neuen Daten nicht mehr. Beim Modellieren müssen wir versuchen ein allgemeingültiges Modell zu finden.
Underfitting hingegen beschreibt wenn das Modell überhaupt nicht auf unsere Daten passt.

Massnahmen Underfitting

Sollte ein Modell zu Underfitting neigen, wäre es ratsam, mehr Datensätze (oder auch mehr Features) zu organisieren. Ebenfalls wäre es sinnvoll (wie auch im Beispiel erwähnt), den eingesetzten Algorithmus zu überprüfen. Wenn beispielsweise eine Polynominale Funktion verwendet wurde, könnte der Grad zu klein sein. Oder es wurde eine Lineare Funktion verwendet, und die Daten sprechen eher für eine polynominale (oder Quadratische) Funktion.

Massnahmen Overfitting

Bei Overfitting ist es ebenfalls ratsam, die eingesetzte Funktion zu überprüfen. Wurde eine polynominale Funktion verwendet, ist eventuell der eingesetzte Grad der Funktion zu hoch. Ebenfalls sollten die Daten überprüft werden. Möglicherweise sind zu viele Ausreisser vorhanden und verzehren so das Modell.