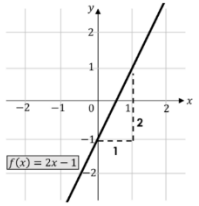
Machine Learning -> S.159

Law of the Hammer:
Wenn ich als Bauarbeiter nur einen Hammer als Werkzeug habe, muss ich alles so behandeln, als wäre es ein Nagel.

Ziel von Machine Learning

Aus einem möglichst grossen Datensatz einen Erkenntnisgewinn gewinnen.
Daten an sich sind keine Informationen, sie können unstrukturiert sein. ML-Methoden sind eine spezielle Form der Statistik.

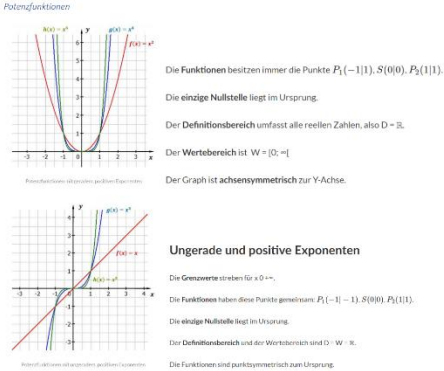
Basics Mathe für Machine Learning



$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

Steigung m berechnen

f(x) = y = m*x + c -> Wenn ich ein x da reinstecke, sollte ein y kommen.
Beispiel hier: f(x) = 2x -1 -> 2x0 = 0 -1 = -1 -> y = -1



Vektoren
Sie sind “Richtungsweiser” und zeigen, wieviel ich in einer Dimension gehen soll. -> Wähle einen Punkt aus auf dem Koordinatensystem und verschiebe ihn in irgendeine Richtung. -> Dann gibt es eine Veränderung der x- und y-Koordinate. Das ist ein Vektor.

Wie kommen bei Machine Learning, Vektoren zum Zug?
Ich erstelle ein Koordinatensystem mit x Achse -> Klarheit, y Achse -> Alkoholgehalt.
Ich möchte nun anhand der Mathematischen Daten (nur den zwei gegebenen) der verschiedenen Getränke wissen können, welche Art / welcher Alkohol dass es ist und welche Position dass dieser auf dem Koordinatensystem hat.
Neuen unbekannten Punkt mit den zwei Daten -> Was ist das für Alkohol?

Länge eines Vektors:

2D die Länge von $\vec{a} = \begin{pmatrix} x \\ y \end{pmatrix}$ ergibt sich aus der Formel : Länge = $\sqrt{x^2 + y^2}$

3D: die Länge von $\vec{a} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ ergibt sich aus der Formel: Länge = $\sqrt{x^2 + y^2 + z^2}$

Korrelation und Kausalität

Korrelation:
A geschieht im Zusammenhang von B
Kausalität:
Wenn zwischen zwei Merkmalen ein Zusammenhang aus Ursache und Wirkung besteht

Korrelationskoeffizient $r_{xy} = \frac{s_{xy}}{s_x \times s_y}$

Gibt an, wie stark eine Variable A die Variable B beeinflusst.

Regeln bei ML

Ohne saubere Daten = kein Machine Learning (GIGO -> Garbage In, Garbage Out)

No Free Lunch Theorem (NFL) -> Keis Gratis Habbru Theorem
"There is no free lunch!"

Es gibt für jedes Problem **einen** optimalen Algorithmus.
Es gibt nicht einen universell besten Optimierungsalgorithmus, der für alle Probleme gleichermassen gut funktioniert.

Daten Daten Daten!

Es braucht genügend Daten, um ML Algorithmen anwenden zu können.

Machine can	Machine cannot
Forecast	Create something new
Memorize	Get smart really fast
Reproduce	Go beyond their task
Choose best item	Kill all humans

War with the Machines
Wir stellen uns immer die Frage: Wann werden die Maschinen schlauer als wir? Das ist aber falsch.

Es sieht so aus als würden die Menschen eine Grenze für Intelligenz aufstellen.

On Top sind die Menschen, Hunde sind etwas blöder und die blödesten sind die Tauben. -> **Falsch** -> **Denn dann müsste der Mensch in allem besser sein als Tiere**

Singularität

Statistics -> Machine Learning -> General AI -> Singurality

Zuerst wurden Wetter Daten von Hand analyiert und Vorhersagen getroffen.
-> Machine Learning ist die Technologie die es erlaubt, dass die Wetter Daten automatisch analysiert werden und dann Vorhergesagt wird.
--> General AI ist eine AI die selbstständig neue Konzepte erlernen kann und verschiedene Probleme lösen kann.
---> Singularity ist schon fast eine eigene Spezies die «schlauer» als die Menschen ist und sich selbst reproduzieren und verbessern kann.

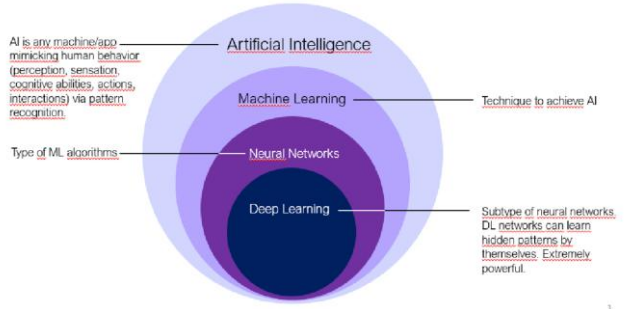
Matrizen
Addieren Zahl mal Matrix

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}; \quad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

$$A \pm B = \begin{pmatrix} a_{11} \pm b_{11} & a_{12} \pm b_{12} \\ a_{21} \pm b_{21} & a_{22} \pm b_{22} \end{pmatrix}$$

$$r \cdot \underbrace{\begin{pmatrix} 3 & 2 \\ 4 & 5 \end{pmatrix}}_{A} = \begin{pmatrix} 3 \cdot r & 2 \cdot r \\ 4 \cdot r & 5 \cdot r \end{pmatrix}.$$

Definition of AI:
Die Wissenschaft und Art intelligente Maschinen, vorallem intelligente Computer Programme zu machen. Einfach gesagt: Künstliche Intelligenz ist ein Feld, welches Informatiktechniken mit robusten Datensätzen verbindet, um Probleme zu lösen.
Siehe die Grafik.



Turing Test:
Ein Menschlicher Befrager versucht, zwischen einer Computer- und einer menschlichen Textantwort zu unterscheiden.

Reinforcement-Learning
Maschine probiert und lernt etwas, scheitert und weiss das es falsch ist. Startet neu und fällt wieder auf die Schnauze, bis es klappt. Und zwar immer. (Super Mario durchspielen von alleine)

Daten:

Definition -> Darstellung von Fakten, Konzepten, Anweisungen in einer formalisierten Weise, die für die Kommunikation, Interpretation oder Verarbeitung durch Menschen oder Maschinen geeignet sein sollten.

Informationen sind organisierte oder klassifizierte Daten, die für den Empfänger einen bestimmten Wert. Reine Daten selbst sind NICHT die Information. Erst wenn man Zusammenhänge bzw. Korrelationen, sowie bestenfalls Kausalitäten erkennt, kann man aus den Daten Informationen und damit Nutzen ziehen.

Daten Arbeitsschritte

1. Daten Bezug
   a. Daten ggf in Firma sammeln
      i. Daten standardisieren
      ii. Daten speichern
      iii. Daten verwalten
   b. Open Data
   c. Daten herunterladen als CSV Datei(en)
      i. Daten über SQL beziehen
      ii. Daten über API (http request) beziehen.
      iii. Feature Extraction I -> Nur Spalten die relevant sind exportieren + Primärschlüssel
2. Data Preprocessing
   a. Daten cleaning I
      i. Daten in Notepad++ sichten
      ii. Daten in Notepad++ bereinigen
      iii. Daten als neue Datei speichern, Originaldaten behalten
   b. Daten in Jupyter Notebook importieren
      i. Daten Import CSV mit Pandas
      ii. Daten sichten -> Head
      iii. Datentypen anzeigen
      iv. Datentypen ggf anpassen
      v. Statistik über Daten anzeigen Pandas
      vi. Daten ggf anonymisieren
   c. Fehlende Daten
      i. Fehlende Daten als NaN füllen
      ii. REGEX bei Text Daten falls nötig
      iii. Fehlende Daten ggf interpolieren oder extrapolieren
      iv. Oder Fehlende Daten ggf löschen (Zeilen löschen)
      v. Daten sichten und Statistik nochmals anzeigen
   d. Features extrahieren II (neue Pandas Datenbank mit relevanten Features erstellen)
      i. Alte Datenbank ggf löschen
      ii. Daten plotten (matplotlib) -> Daten grafisch Darstellen und visuell inspizieren
      iii. Daten ggf numerisieren -> encoding
      iv. Daten ggf normalisieren -> Bsp. anstatt 1-100 -> 0.00 bis 1.00
   e. Daten aggregieren -> Datenbanken zusammenführen über Primärschlüssel

ab ca. 1990:

George Hinton, Andrew Ng, ...

Faktoren für die Rennaissance von Neuronalen Netzen:

a) grosse Rechenkraft günstig zur Verfügung (GPU)
b) grosse Menge an Trainingsdaten
c) (open source) libraries für die schnelle Entwicklung neuer Algorithmen

Datentyp	Definition	Beispiele
Nominal	Rein qualitative Merkmalsausprägungen ohne natürliche Ordnung	Geschlecht, Berufsstatus, dichotome Antwort vom Typ „ja/nein“
Ordinal	Qualitative Merkmalsausprägungen mit natürlicher Ordnung	Qualitätseinschätzung („sehr gut“, „gut“, „mittel“, „schlecht“, „sehr schlecht“)
Metrisch (auch: rational)	Merkmalsausprägungen, die in einer Zahl besteht und eine Dimension und einen Nullpunkt besitzt	Einkommen (in Euro), Alter (in Jahren), Leistung (in Stück pro Stunde, in km/h)

Provides:	Nominal	Ordinal	Interval	Ratio
The "order" of values is known		✔	✔	✔
"Counts," aka "Frequency of Distribution"	✔	✔	✔	✔
Mode	✔	✔	✔	✔
Median		✔	✔	✔
Mean			✔	✔
Can quantify the difference between each value			✔	✔
Can add or subtract values			✔	✔
Can multiple and divide values				✔
Has "true zero"				✔

Metrische (quantitive, numerische) Daten

- **Diskrete Daten**

  Können nur bestimmte numerische Werte haben. -> Sie entstehen nahezu ausnahmslos durch Zählungen. Es sind Intervalle, im Zusammenhang mit Messungen auf der Zeit achse entsprechen sie Punkten. (Bsp: Anzahl Kinder)

- **Kontinuierliche Daten**

  Werden durch Messungen gesammelt. Die Daten können alle möglichen Werte annehmen. Nur die Messung kann sie einschränken. Im Zusammenhang mit Messungen auf der Zeitachse entsprechen sie Linien. (Bsp: Körpergrösse)

Geschichte Machine Learning

1943
Warren McCulloch und Walter Pitts beschreiben eine Art neurologischer Netzwerke, bauen Schwellwertschalter die jede logische oder auch arithmetische Funktion berechnen können.

«Elektronengehirne» unterstützt von Konrad Zuse

1949
Donald O. Hebb formuliert die klassische Hebb'sche Lernregel, welche die Basis fast aller neuronalen Lernverfahren darstellt.

1951
Marvin Minsky entwickelt Neurocomputer für Dissertation (Snark)

1957 – 1958
Frank Rosenblatt, Charles Wightman und Mitarbeiter entwickeln erfolgreichen Neurocomputer am MIT (Mark 1 Perceptron)

1969
Marvin Minsky und Seymour Papert veröffentlichen eine genaue Analyse des Perceptron und dass dieser viele wichtige Probleme g nicht repräsentieren kann. -> Grosse Überschätzung

1982
Teuvo Kohonen beschreibt die nach ihm benannten selbstorganisierten Karten auf der Suche nach den Mechanismen des Gehirns

1983
Fukushima, Miyake und Ito stellen das neuronale Modell Neocognitron zur Erkennung handgeschriebener Zeichen vor. (Pioneere von OCR)

1986
Lernverfahren «Backpropagation of Error» wird als Verallgemeinerung der Delta-Regel durch die Parallel Distributed Processing-Group separat entwickelt und weit publiziert.

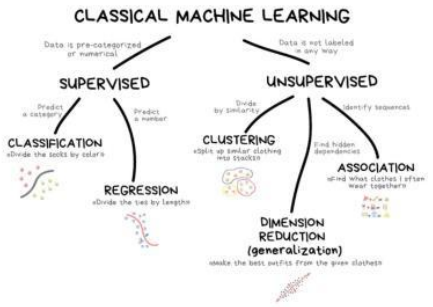
Philosophie:
Aristoteles formulierte Gesetze für logisches Denken. Hobbes verglich Denken mit Rechnen. Leibniz träumte von Maschinen, die alle Probleme löse
Mathematik und Statistik:
Mathematik bietet Regeln für logische Schlüsse. Bayes' Regel hilft bei unsicherem Wissen. Turing untersuchte, was berechenbar ist.
Wirtschaft (Spieltheorie):
Entscheidungstheorie kombiniert Wahrscheinlichkeiten und Nutzen. Spieltheorie zeigt, wie rationale Entscheidungen unter Unsicherheit getroffen werden.
Neurowissenschaften:
Das Gehirn verarbeitet Informationen über neuronale Netzwerke.
Psychologie:
Untersucht, wie Menschen und Tiere denken und handeln.

Computertechnik:
Fokussiert auf den Bau effizienter Computer und Softwareentwicklung.

Elektronik und Elektrotechnik:
Entwickelt Hardware wie Rechenmaschinen, z. B. die Turing-Maschine.

Kybernetik:
Wiener definierte Regelkreise und negative Rückkopplung. Fokus liegt auf Steuerung und Zielerreichung.

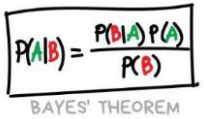
Linguistik:
Untersucht die Beziehung zwischen Sprache und Denken.



Supervised Learning (Classical ML): Die Maschine hat einen "Aufseher" oder „Lehrer", der der Maschine alle Antworten gibt. Er hat die Daten bereits gelabelt.

Klassifikation: Objekte aufgrund im Vor-hinein bekannter Attribute einteilen. **Algorithmen:** Naive Bayes, Decision Tree, Logistische Regression, KNN, SVM. Daten sollten immer mit Features gelabelt sein.

Naive Bayes: Beispiel Spam: Die Maschine zählt die Anzahl Erwähnungen von „Viagra" in Spam-Mails und Normalen Mails. Danach werden die Wahrscheinlichkeiten mittels der Bayes-Gleichung multipliziert und die Ergebnisse zusammengezählt.



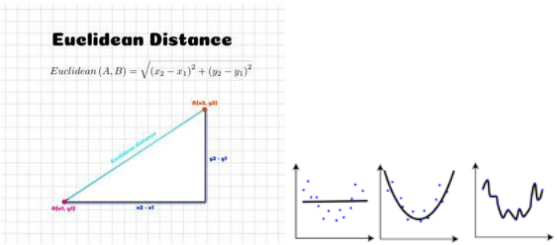
Decision Tree: Die gesamten Daten werden automatisch in Ja / Nein-Fragen (Binary Trees) aufgeteilt. Je höher der Zweig, desto allgemeiner die Frage.

kNN: Bei kNN wird einem Punkt eine Klasse zugewiesen, indem die Klassen der nächstge-legenen Punkte berücksichtigt werden. Bestimmung der Entfernung: Distanz messen (Euklidisch: $\sqrt{(\Delta x)2+(\Delta y)2}$) kNN 5: 5 nächste Punkte. Nachteile: Kann nicht auf Unvorhergesehenes reagieren / langsam und verbraucht viel Speicher / nicht für grosse Datensätze geeignet

kNN: Der kNN-Klassifikator identifiziert die Klasse eines Datenpunkts mittels des Mehrheitswahlprinzips. Wenn k auf 5 gesetzt ist, werden die 5 nächsten Punkte geprüft. Die Vorhersage wird aufgrund der häufigsten Klasse durchgeführt. (kNN-Regression nimmt den Medianwert der 5 nächsten Punkte) **/** Die Entfernung wird mittels der Distanz bestimmt (Euklidisch: $\sqrt{(\Delta x)2+(\Delta y)2}$).

Bestes kNN finden k = 1: Modell ist zu eng gefasst und nicht ordentlich verallgemeinert. Empfindlichkeit auf Rauschen ist hoch. Neue, vorher unbekannte Daten werden sehr genau auf den Datensatz vorhergesagt, allerdings ist die Vorhersage auf neue, ungesehene Daten schlecht. Resultat: Overfit. **k = 100:** Modell ist zu allgemein und sowohl auf die Test als auch auf die Trainings-Datensätze unzuverlässig. Resultat: Underfitting

kNN-Limitationen: Kann nicht auf unvorhergesehene Dinge reagieren / Langsam und teuer in Zeit und Speicher / Benötigt viel Speicher zum Speichern des gesamten Trainings-Datensets / Euklidische Distanz ist empfindlich auf Grössen → Charakteristiken mit grossen Grössen werden diese mit kleinen Grössen überschatten. / Nicht für grossdimensionale Datensätze geeignet.



Support Vector Machine SVM: Es wird versucht, eine Linie zu zeichnen, bei dem der Abstand (Margin) zwischen den Datensätzen möglichst gross ist.

Vorteil: Detektion von Anomalien (Wenn man dem Computer beibringt, was stimmt, lernt er automatisch auch das, was falsch ist). **Faust-regel:** Je grösser der Datensatz, desto komple-xer der Algorithmus

Auswahl des richtigen Algorithmus: Für Text, Zahlen und Tabellen sollte man den **klassischen Weg gehen**, da die Modelle kleiner sind, diese schneller lernen und klarer arbeiten. Für Bilder, Video and andere komplizierte Big-Data-Sachen, sollte man **definitiv neuronale Netzwerke verwenden.**

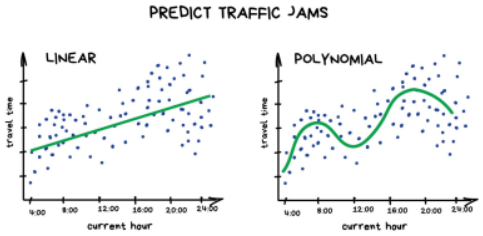
Regression:
Versuch, eine Zahl statt einer Kategorie vorherzusagen.

Gerade Linie: Lineare Regression.
Ungerade Linie: Polynomiale Regression.
Achtung: Logistische Regression ist eine **Klassifikationsmethode**!

Lineare Regression: Statistisches Modell, in dem die lineare Beziehung zwischen zwei oder mehr Variablen untersucht wird – einer abhängigen und deren unabhängigen Variablen.
Bedeutet: Wenn eine unabhängige Variable steigt oder sinkt, steigt oder sinkt auch die abhängige Variable.

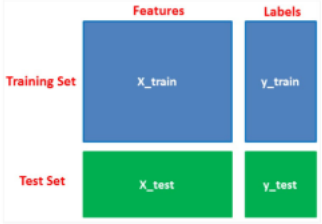
$f(x)=m\cdot x+c$S

Polynomiale Regression: Der Graph ist nicht gerade, sondern gekurvt.

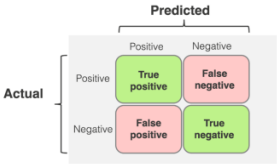


Ganz wichtig

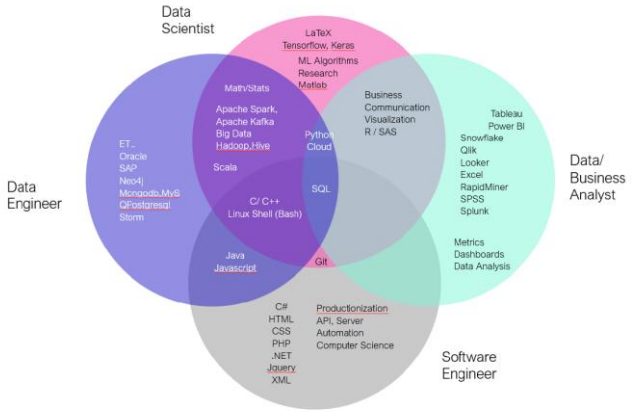
Die Daten müssen in ein Testset und ein Trainingsset aufgeteilt werden. Trainingsset: 20-30%. Testset: 70-80%. Dabei müssen die Labels und die Features getrennt werden, weil es sonst zu y = y kommt, und das natürlich nicht gewollt ist.



Nicht zu verwechseln mit der Confusion Matrix!



Interpolation & Extrapolation: Ergänzung von fehlenden Daten



Open Data

Open Data ist ein Konzept, das auf der Grundidee beruht, dass Daten für alle BürgerInnen frei zugänglich gemacht werden sollen. Dies betrifft insbesondere Verzicht auf Urheberrechte, Patente oder andere Nutzungsausschlüsse.

Beispiel Forst: interne Daten mit Flächennutzungsdaten des Kantons ergänzen

Daten sammeln

- Alle Daten mit einem Primärschlüssel versehen
- Originaldaten (Rohdaten) falls möglich getrennt in Datenbank speichern
- Nie Rohdatenbank selbst bearbeiten, sondern Daten kopieren und dort bearbeiten.
- Daten Features (Variablen) einheitlich und erkennbar kennzeichnen
- Text Daten (zB. Reports) mit einheitlicher Sprache schreiben
- Standardisierte Formulare verwenden
- Daten aufbereiten und nach Datenbankschema speichern
- Daten aggregieren und in Datenbanken speichern
- Data Pooling: Daten von verschiedenen Quellen an einer Stelle speichern. Ggf speziell dedizierten Server verwenden
- Auf Interoperability achten! Daten nach Standards erheben und speichern (zB ISO Standards)
- Daten verwalten und Datenbanken aktuell halten.
- Wenn neue Software zum Datenbankmanagement installiert wird, überprüfen, dass diese weiter genutzt werden können.
- Daten, die nur analog vorhanden sind, digitalisieren und standardisieren.

Tipp: Daten die extern gespeichert oder berechnet werden, immer zuerst intern anonymisieren!

Python

Strings integers zuweisen und ausprinten.

```
[14]: d = {'a': 1, 'b': 2, 'c': 3}

[16]: d['a']

[16]: 1
```

Pandas Series aus Dictionary

```
<varname> = {<index>: <int64>, <index>: <int64>, <index>: <int64>}
<series_name> = pd.Series(data=<varname>, index=[<index>, <index>, <index>])
```

```
Wert an Stelle n einfügen
<series_name>.iloc[<n>] = <int64>
```

Pandas Dataframe

```
<df_name> = pd.DataFrame()
```

Pandas Dataframe aus CSV (Bei grossen Datensätzen low_memory auf false setzen)

```
<df_name> = pd.DataFrame(pd.read_csv(<path>, sep=<sep>, low_memory=<bool>))
```

```
Erste Zeilen des Dataframes anzeigen
<df_name>.head(<opt:int64>)

Alle Spalten anzeigen
<df_name>.columns

Bestimmte Spalte anzeigen
<df_name>.loc[:,"<row_name:string>"]

Spalten als Liste:
<df_name>.columns.tolist()

Spalten umbenennen
<df_name> =
<df_name>.rename(columns={'<old>':"<new>"})

Spalte kopieren
<df_name>[‘<copy>’] =
<df_name>[‘<og>’].copy()

Neue Spalte mit 0 bzw np.NaN
<df_name>['<new>'] = <0 / np.nan>

Statistik und Infos
<df_name>['<key>'].unique()
<df_name>['<key>'].min()
<df_name>['<key>'].max()
<df_name>.info()
<df_name>.describe()

Alle Zeilen anzeigen
pd.set_option('display.max_rows', None)

Alle Spalten anzeigen
pd.set_option('display.max_columns', None)
```

Product Cycle Ablauf

Faustregel: Kosten für Entwicklung und Trainieren des Modells sind hoch und zeitauf-wändig, Kosten für laufenden Betrieb relativ billig.

1.-Zieldefinition (Was soll mit Projekt erreicht werden, welche Informationen sollen gewonnen werden, was ist erfolgreich, Wichtig: klares Ziel)

2. Kostenrechnung und Risikoanalyse (Kosten und Risikoanalyse des Projekts, Kosten für laufenden Betrieb)

3. Datenbeschaffung (Daten beschaffen oder sammeln)

4. Aufbereitung (NP++ und Pandas)

5. Training

a. Daten in Trainset und Testset aufteilen.
b. Faustregel: 20-25% Trainingsdaten & 75-80% Testdaten
c. Daten aufteilen in Features und Labels. Denn es macht keinen Sinn, mit der Antwort zu trainieren, sonst wäre y = y. Man versucht die Labels vorherzusagen

6. Testing

7. Interpretation, ob Ergebnisse Sinn machen; 8. Verifikation;
9. Produktiver Einsatz (LINUX Server)

Arten von Machine Learning

Grundsätzlich: ML setzt voraus, dass Daten in Features und Labels getrennt werden. Features sind die Charakteristiken der Sache, die man sich anschaut (unabhängige Variable). Labels sind das, was man versucht, vorauszu-sagen (abhängige Variable).