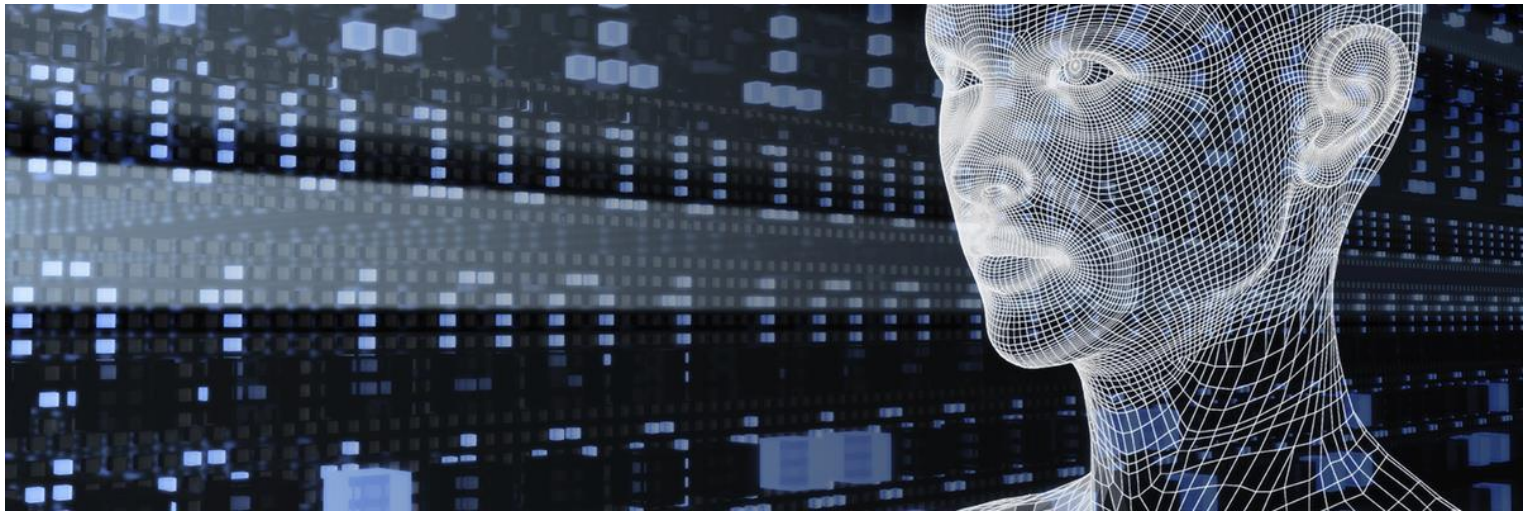
# Künstliche Intelligenz

## **Vorlesung 9**: Maschinelles Lernen und Data Mining



# Wiederholung

Data Mining: finde interessante Strukturen in Daten.

- Regeln
- Graphen oder Netzwerke: Wissensgraphen&Co.
- Bäume
- Gleichungen
- u.v.m.

Data Mining Session: einen oder mehrere Algorithmen.

Induction-based learning: Bilde allgemeine Begriffe aus spezifische Beispiele, aus denen das System lernt.

# Data Mining

- 1995: Statistik, Mathematik, ML, AI, Business.
- 1998: Knowledge Discovery in Databases (KDD) – Wissen wird aus Daten abgeleitet.
- Zusätzlich zu Data Mining enthält KDD:
  - ✓Methoden zur Daten Vorbereitung (sehr Zeitintensiv).
  - ✓Decision Making
- Data Mining vs. Data Science: Große Datenvolumen vs. Stammt aus der Datenbank Community
- Data Mining ist keine Vorbedingung für Data Science (Modelle im analytics Prozess müssen von reellen Daten stammen).

# Zusammenfassung: Maschinelles Lernen

- Was können Computer lernen?
- Vier Lernniveaus:

1. Fakten: eine wahre Aussage
2. Begriffe: Sammlung von Objekte und Eigenschaften
   - Bäume, Regeln, Gleichungen, usw.
3. Prozeduren: schrittweises Herangehen, um eine Aufgabe zu lösen
4. Prinzipien: allgemeine Wahrheiten und Gesetze

# Begriffe: der Klassiker

- Begriffe sind über Eigenschaften erklärt. Diese Eigenschaften ermöglichen uns Individuen, die unter dem Begriff fallen zu identifizieren.
- Beispiel: Vogel, Pinguin, Maus, usw.
- Methoden der Knowledge Discovery: FCA, Knowledge Graphs, Description Logics, Semantic Web, Ontologien, uvm.

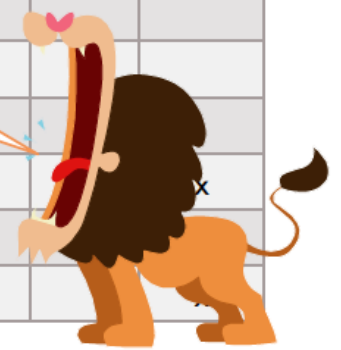
# Begriffe: der Klassiker

| 𝕂 | small | medium | big | 2legs | 4legs | feathers | hair | fly | hunt | run | swim | mane | hooves |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| dove | x | | | x | | x | | x | | | | | |
| hen | x | | | x | | x | | | | | | | |
| duck | x | | | x | | x | | x | | | x | | |
| goose | x | | | x | | | | | | | | | |
| owl | x | | | x | | | | | | | | | |
| hawk | x | | | x | | | | | | | | | |
| eagle | | x | | x | | x | | x | x | | | | |
| fox | | x | | | x | | x | | x | x | | | |
| dog | | x | | | x | | | | | | | | |
| wolf | | x | | | x | | | | | | | x | |
| cat | x | | | | x | | | | | | | | |
| tiger | | | x | | x | | x | | x | | | | |
| lion | | | x | | x | | x | | x | x | | | |
| horse | | | x | | x | | x | | | x | | | x |
| zebra | | | | | x | | x | | | x | | | |
| cow | | | | | x | | x | | | | | | |

set of **attributes** (M)

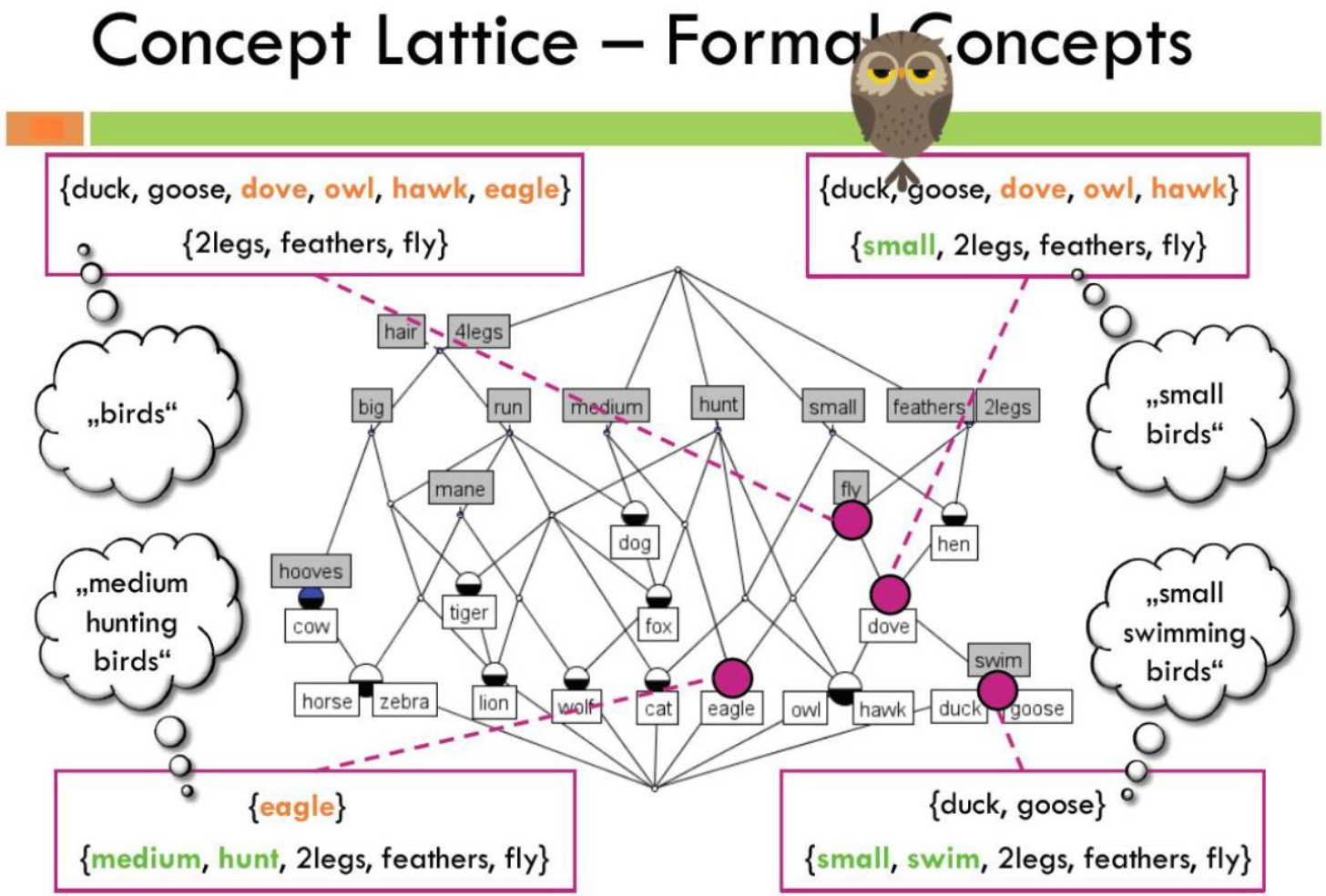
crosses indicate **incidence relation** (I) between G and M

(G,M,I) is called **formal context**



set of **objects** (G)

# Begriffe: der Klassiker

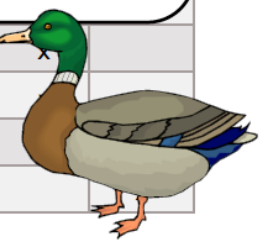


# Begriffe: der Klassiker

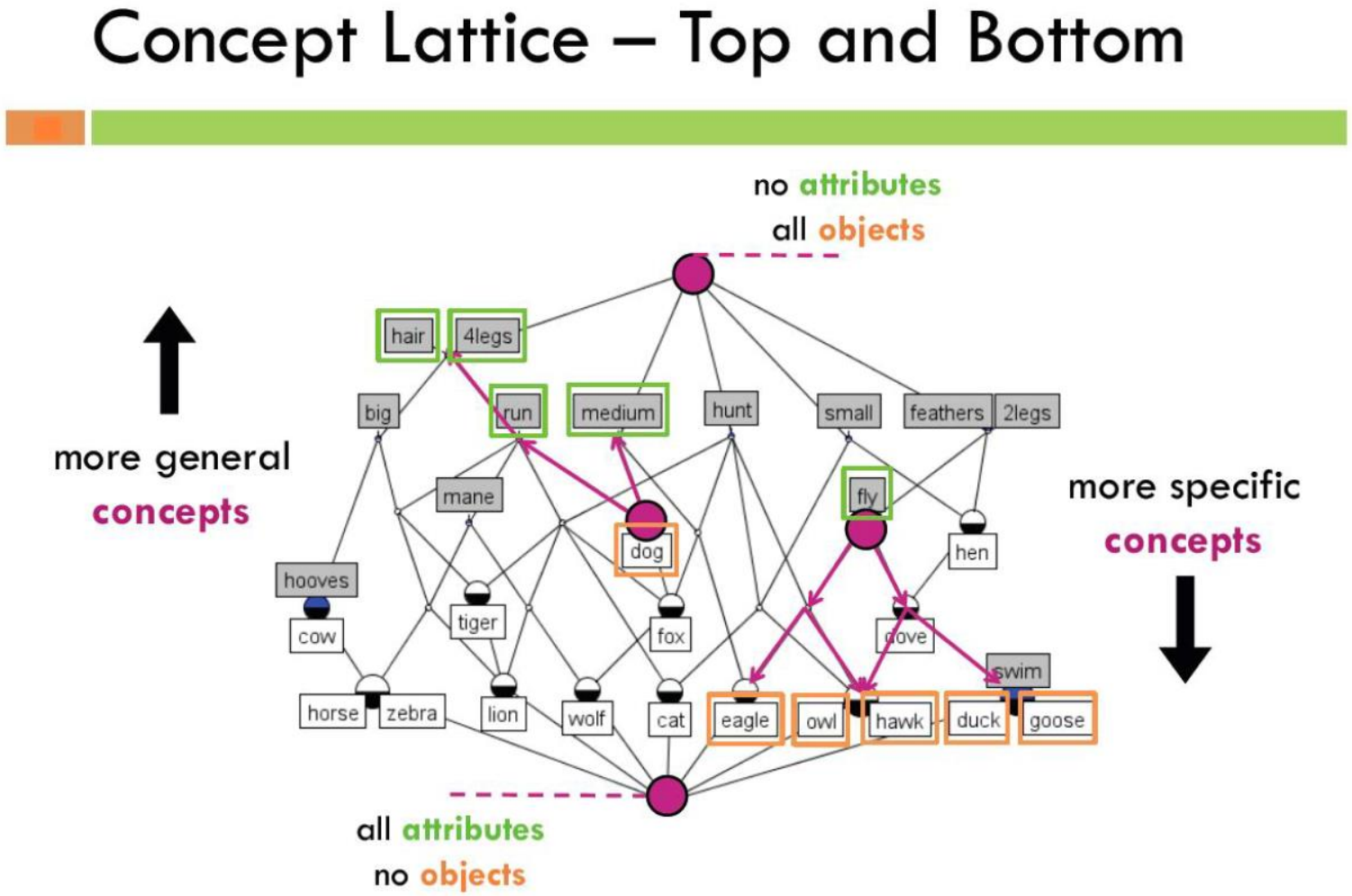
## Calculating Formal Concepts

| 𝕂 | small | medium | big | 2legs | 4legs | feathers | hair | fly | hunt | run | swim | mane | hooves |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| dove | x | | | x | | x | | x | | | | | |
| hen | x | | | x | | x | | | | | | | |
| duck | x | | | x | | x | | x | | | x | | |
| goose | x | | | x | | x | | x | | | x | | |
| owl | x | | | x | | x | | x | x | | | | |
| hawk | x | | | x | | x | | x | x | | | | |
| lion | | | x | | x | | x | | x | x | | x | |
| horse | | | x | | x | | x | | | x | | | |
| zebra | | | x | | x | | x | | | x | | | |
| cow | | | x | | x | | x | | | | | | |

1. Pick a set of objects: A = {duck}
2. Derive attributes: A‘ = {small, 2legs, feathers, fly, swim}
3. Derive objects: (A‘)‘ = {small, 2legs, feathers, fly, swim}‘ = {duck, goose}
4. Formal concept: (A‘‘,A‘) = ({duck, goose}, {small, 2legs, feathers, fly, swim})



# Begriffe: der Klassiker



# Begriffe: Stochastik

- Begriffe müssen nicht mehr über charakteristische Eigenschaften definierbar sein.
- Die Merkmale müssen für die Objekte der Begriffe nur mit einer gegebener Wahrscheinlichkeit zutreffen.
  - Probabilistische Methoden
  - Fuzzy Theorie
  - Fuzzy FCA

# Begriffe: Ähnlichkeitsmaße

- Eine Instanz gehört zu einem Begriff, falls es zu einer Begriffsinstanz ähnlich ist.
- Ähnlichkeitsmaße: Cosinus, Sinus, etc.

# Maschinelles Lernen: Zusammenfassung

- Künstliche Generierung von Wissen aus Erfahrung (Lernen)
- Das System lernt aus Beispielen und kann diese, nach Beendigung der Lernphase verallgemeinern
- Es erkennt neue Muster und Regeln

# Maschinelles Lernen: Zusammenfassung

- Supervised Learning/Überwachtes Lernen
- Über 90% der Algorithmen sind supervised learner
- Gegeben ist eine Menge von Paaren von Input und Output. Ziel ist eine Funktion zu berechnen. Dabei steht während des Lernens ein Lehrer bereit.
- Die Fähigkeit Assoziationen zu bilden wird trainiert (ähnlich wie beim Lernen der Kinder).

# Überwachtes Lernen/supervised learning

Um ein bestimmtes Problem mit Überwachtem Lernen zu lösen, muss man die folgenden Schritte durchführen:

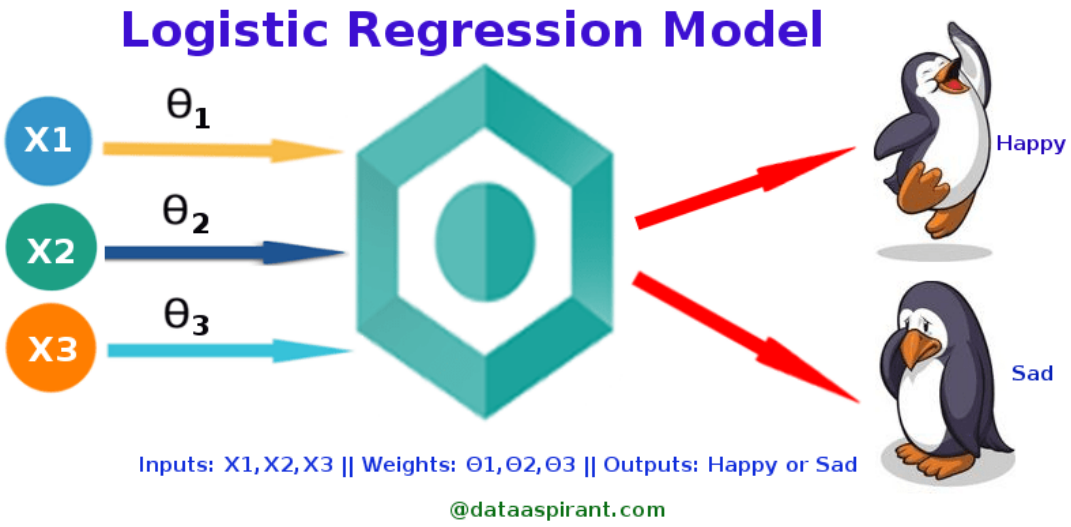
- Die Art der Trainingsbeispiele bestimmen. Das heißt es muss zunächst bestimmt werden, welche Art von Daten der Trainingsdatensatz enthalten soll. Bei der Handschriftanalyse kann es sich z. B. um ein einzelnes handschriftliches Zeichen, ein ganzes handschriftliches Wort oder eine ganze Zeile Handschrift handeln.
- Eine Datenerhebung der vorangegangenen Auswahl entsprechend durchführen. Es müssen sowohl die erklärenden Variablen als auch die erklärten Variablen erhoben werden. Diese Erhebung kann von menschlichen Experten, durch Messungen und andere Methoden vollzogen werden.
- Die Genauigkeit der gelernten Funktion hängt stark davon ab, wie die erklärenden Variablen dargestellt werden. Typischerweise werden diese in einen Vektor transformiert, der eine Reihe von Merkmalen enthält, die das Objekt beschreiben. Die Anzahl der Features sollte nicht zu groß sein; sie sollte aber genügend Informationen enthalten, um die Ausgabe genau vorhersagen zu können.

# Überwachtes Lernen/supervised learning

- Daraufhin muss die Struktur der gelernten Funktion und der dazugehörige Lernalgorithmus bestimmt werden. Bei einem Regressionsproblem zum Beispiel sollte an dieser Stelle entschieden werden, ob eine Funktion mit oder ohne Parameter besser geeignet ist, um die Approximation durchzuführen.
- Anschließend wird der Lernalgorithmus auf dem gesammelten Trainingsdatensatz ausgeführt. Einige überwachte Lernalgorithmen erfordern vom Anwender die Festlegung bestimmter Regelparameter. Diese Parameter können entweder durch die Optimierung einer Teilmenge des Datensatzes (Validierungsdatensatz genannt) oder durch Kreuzvalidierung angepasst werden.
- Als letztes muss die Genauigkeit der gelernten Funktion bestimmt werden. Nach der Parametrierung und dem Erlernen der Parameter sollte die Leistung der resultierenden Funktion an einem Test-Datensatz gemessen werden, der vom Trainingsdatensatz getrennt ist.

# Regressionsprobleme: Logistic Regression

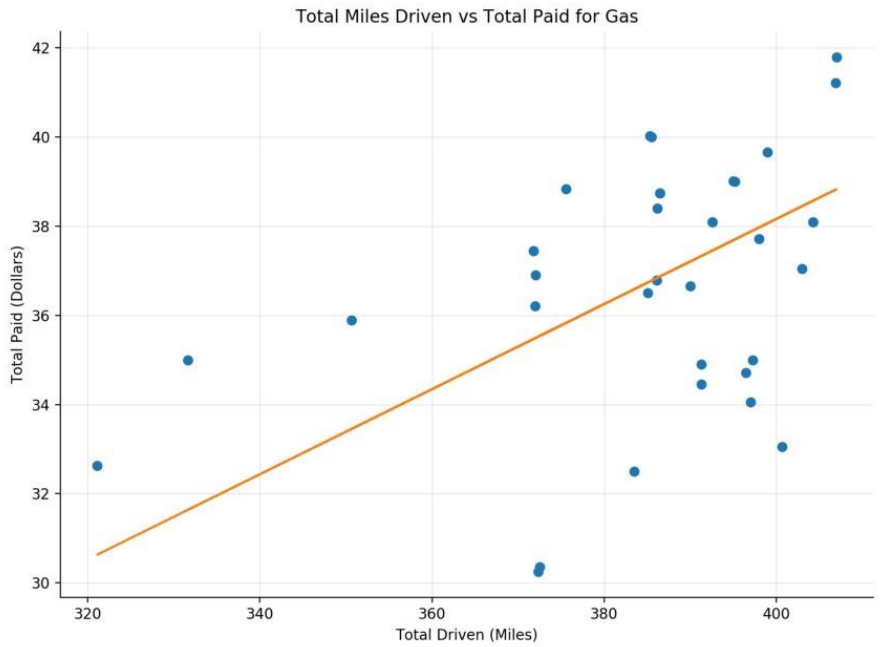
- Logistic Regression wurde im frühen zwanzigsten Jahrhundert in der Biologie verwendet.
- Es wurde dann in vielen sozialwissenschaftlichen Anwendungen verwendet.
- Logistic Regression wird verwendet, wenn die abhängige Variable (Ziel) kategorial ist.
- Zum Beispiel:
  - ❑Um vorauszusagen, ob eine Mail spam ist oder nicht (1) oderr (0)
  - ❑Ob ein Tumor malign (1) ist oder nicht (0)



# Logistic Regression

- Betrachten wir das Szenario, in dem wir klassifizieren müssen, ob eine E-Mail Spam ist oder nicht.
- Wenn wir für dieses Problem eine lineare Regression verwenden, muss ein Schwellenwert festgelegt werden, auf dessen Grundlage die Klassifizierung vorgenommen werden kann.
- Wenn die tatsächliche Klasse bösartig ist, ein Wert von 0,4 vorhergesagt wird und der Schwellenwert 0,5 ist, werden die Daten als nicht bösartig klassifiziert, was in später zu schwerwiegenden Konsequenzen führen kann.

- Lineare Regression?



# Lineare Regression

- Die lineare Regression ist eine grundlegende und häufig verwendete Art der Vorhersageanalyse.
- Die allgemeine Idee der Regression besteht darin, zwei Dinge zu untersuchen :
  1. Leistet eine Reihe von Prädiktorvariablen eine gute Arbeit beim Vorhersagen einer Ergebnisvariablen (abhängigen Variable)?
  2. Welche Variablen haben insbesondere signifikante Prädiktoren für die Ergebnisvariable und wie geben sie die Größe und das Vorzeichen der Beta-Schätzungen an?

- Diese Regressionsschätzungen werden verwendet, um die Beziehung zwischen einer abhängigen Variablen und einer oder mehreren unabhängigen Variablen zu erklären.
- Die einfachste Form der Regressionsgleichung mit einer abhängigen und einer unabhängigen Variablen wird durch die Formel y = c + b*x definiert,

Dabei gilt y = geschätzte abhängige Variable, c = konstant, b = Regressionskoeffizient und x = Bewertung der unabhängigen Variablen.

# Lineare Regression

- Drei Hauptanwendungen für die Regressionsanalyse sind

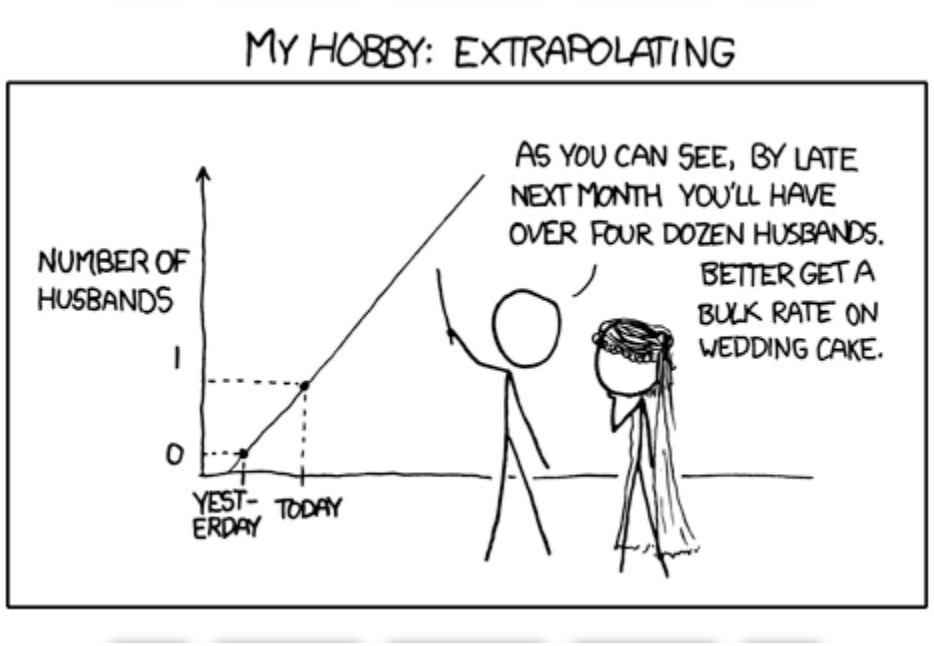
1. Bestimmung der Stärke von Prädiktoren
2. Vorhersage eines Effekts und
3. Trendprognose.

- Die Regression kann verwendet werden, um die Stärke der Auswirkung der unabhängigen Variablen auf eine abhängige Variable zu bestimmen.
- Typische Fragen sind die Stärke der Beziehung zwischen Dosis und Wirkung, Vertriebs- und Marketingausgaben oder Alter und Einkommen.

- Zweitens kann es verwendet werden, um Auswirkungen oder Auswirkungen von Änderungen vorherzusagen.
- Das heißt, die Regressionsanalyse hilft uns zu verstehen, wie viele abhängige Variablen sich mit einer oder mehreren unabhängigen Variablen ändern. Eine typische Frage ist: "Wie viel zusätzliche Einnahmen bekomme ich für jeden 1.000 Dollar, der für Marketing ausgegeben wird?"
- Drittens prognostiziert die Regressionsanalyse Trends und zukünftige Werte. Die Regressionsanalyse kann verwendet werden, um Punktschätzungen zu erhalten. Eine typische Frage ist: "Was wird der Goldpreis in 6 Monaten sein?"

# Regression Analyse

- Simple linear regression: 1 dependent variable (interval or ratio), 1 independent variable (interval or ratio or dichotomous)
- Multiple linear regression: 1 dependent variable (interval or ratio) , 2+ independent variables (interval or ratio or dichotomous)
- Logistic regression: 1 dependent variable (dichotomous), 2+ independent variable(s) (interval or ratio or dichotomous)
- Ordinal regression: 1 dependent variable (ordinal), 1+ independent variable(s) (nominal or dichotomous)
- Multinominal regression: 1 dependent variable (nominal), 1+ independent variable(s) (interval or ratio or dichotomous)
- Discriminant analysis: 1 dependent variable (nominal), 1+ independent variable(s) (interval or ratio)



# Lineare Regression

- Die Kernidee besteht darin, eine Gerade zu erhalten, die am besten zu den Daten passt.
- Die am besten passende Gerade ist die, für die der Vorhersagefehler insgesamt so klein wie möglich ist.
- Der Fehler ist der Abstand zwischen dem Punkt und der Regressionslinie.
- Lineare Regression bedeutet die Bewertungen einer Variablen aus den Bewertungen der zweiten Variablen vorherzusagen.
- Die von uns vorhergesagte Variable wird als Kriteriumvariable bezeichnet und mit **Y** bezeichnet.
- Die Variable, auf der wir unsere Vorhersagen basieren, heißt Prädiktorvariable und mit X bezeichnet.
- Wenn es nur eine Vorhersagevariable gibt, wird die Vorhersagemethode als einfache Regression bezeichnet.
- Ziel der linearen Regression ist es, die am besten passende Gerade durch die Punkte zu finden.
- Diese heißt Regressionsgerade und hat die Gleichung.

$$h_\theta(x) = \theta_0 + \theta_1 x$$

# Lineare Regression

Diese Gleichung heißt **Hypothesen Gleichung**

**wobei:**

- $h_\theta(x)$ hat den Wert Y (den wir vorhersagen wollen) für einzelne Werte x (d.h. Y ist eine lineare Abbildung in x)
- $\theta_0$ ist eine **konstante**
- $\theta_1$ ist der **Regressionskoeffizient**
- X ist der Wert der unabhängigen Variablen

**Die Regressionsgerade hat folgende Eingenschaften:**

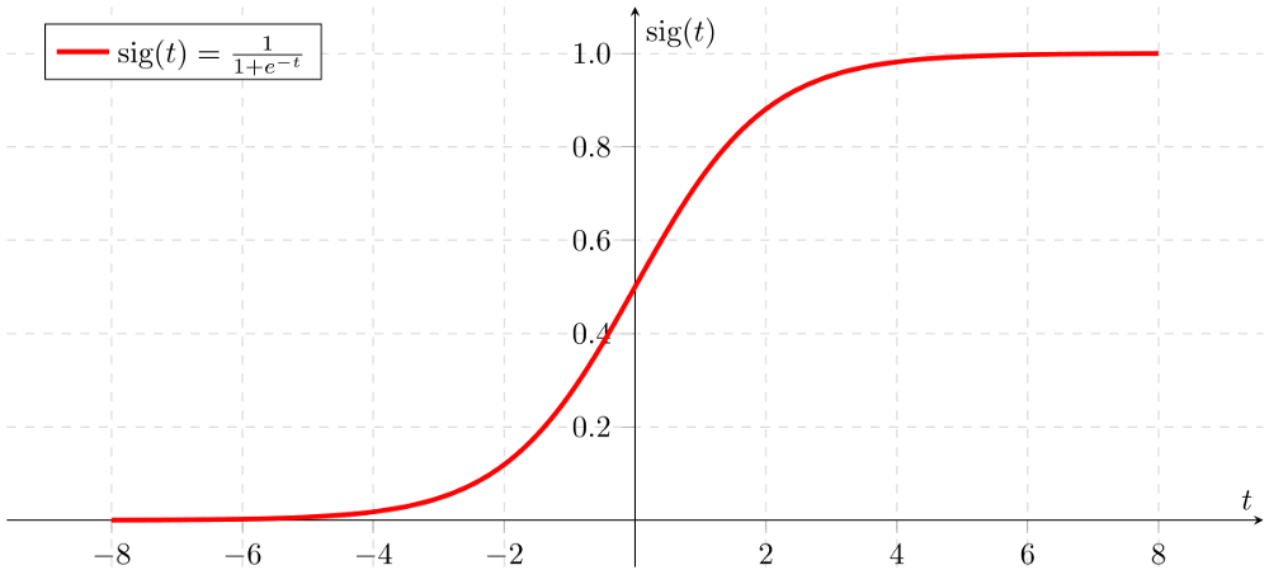
- minimiert die Summe der quadrierten Differenzen zwischen den beobachteten Werten (den y-Werten) und den vorhergesagten Werten (die $h_\theta(x)$ Werte, die aus der Regressiongleichung berechnet werden).
- Die Regressionsgerade geht durch den Mittelwert der X-Werte (x) und durch den Mittelwert der Y-Werte ( $h_\theta(x)$ ).
- Die regressionskonstante($\theta_0$) ist gleich dem y-Achsenabschnitt der Regressionsgeraden.
- Der Regressionskoeffizient($\theta_1$) ist die mittlere Änderung der abhängigen Variablen (Y) für eine Änderung der unabhängigen Variablen (X) um 1 Einheit. Es ist die Steigung der Regressionsgerade.
- Die Regressionsgerade die einzige Gerade, die alle diese Eigenschaften aufweist.

# Logistic Regression

***Modell:***

- Output = 0 or 1
- Hypothese => Z = WX + B
- $h_\Theta(x)$ = sigmoid (Z)

***Sigmoid Funktion***




Falls ‘Z’ Richtung unendlich strebt, Y(predicted) wird den Wert 1 anstreban und falls ‘Z’ Richtung minus unendlich anstrebt, dann Y(predicted) wird den Wert 0 anstreben.

# Logistic Regression

***Analyse der Hypothese***

- Der Output der Hypothese ist die abgeschätzte Wahrscheinlichkeit (estimated probability). Diese wird benutzt um das Vertrauen (confidence) in der predicted value gegeben das Input X abzuschätzen.
- Gegeben folgendes Beispiel:

$$X = [x_0, x_1] = [1, \text{IP-Address}]$$

- Gegeben $x_1$, nehmen wir an, dass die abgeschätzte Wahrscheinlichkeit 0.8 ist. Dies sagt uns, dass die Wahrscheinlichkeit, dass die E-Mail spam ist liegt bei 80%.
- Mathematisch sieht das folgendermaßen aus:

$h_\Theta(x) = P(Y=1 | X; \text{theta})$

Probability that Y=1 given X which is parameterized by 'theta'.

$P(Y=1 | X; \text{theta}) + P(Y=0 | X; \text{theta}) = 1$

$P(Y=0 | X; \text{theta}) = 1 - P(Y=1 | X; \text{theta})$

# Logistic Regression

- Diese Eigenschaft erklärt den Namen ‘logistic regression’.
- Das lineare regressions Modell wird mit Daten gefüttert, das Ergebnis wird dann als Input einer logistic function benutzt, um das Ziel (eine kategorielle abhängige Variable) abzuschätzen.

***Logistic Regression Typen:***

1. Binary Logistic Regression: Binäre Anwort. Beispiel: Spam oder kein Spam
2. Multinomial Logistic Regression: Mindestens drei Kategorien, ohne Ordnung. Beispiel: Vorhersagen der beliebten Mahlzeit (Veg, Non-Veg, Vegan)
3. Ordinal Logistic Regression: Mindestens drei Kategorien mit Ordnung. Beispiel: Movie rating von 1 bis 5

# Logistic Regression

### *Decision Boundary*

- Um vorauszusagen, zu welcher Klasse ein Objekt gehört, können wir ein threshold setzen. Basierend auf dieses threshold, die geschätzte Wahrscheinlichkeit wird dann klassifiziert.
- Beispiel: if predicted_value ≥ 0.5, then classify email as spam else as not spam.
- Decision boundary kann linear oder nicht-linear sein.

### *Cost function*

$$\text{Cost}(h_\Theta(x), Y(\text{actual})) = -\log(h_\Theta(x)) \text{ if } y=1$$

$$-\log(1-h_\Theta(x)) \text{ if } y=0$$

- Linear regression uses mean squared error as its cost function.
- If this is used for logistic regression, then it will be a non-convex function of parameters (theta).
- Gradient descent will converge into global minimum only if the function is convex.

# Logistic Regression

- MNIST handwritten digits database: 95% Treffgenauigkeit nur mit Logistic Regression

Grundidee!

- In der Bearbeitung nicht-linearer Probleme, versuchen wir manchmal die Originaldaten zu linearisieren.

# Supervised Learning: Ziele

Automatische Klassifizierung: Tabelle mit Beispiele und Gegenbeispiele als Trainingsdatensatz

TABLE 1.1 Hypothetical Training Data for Disease Diagnosis

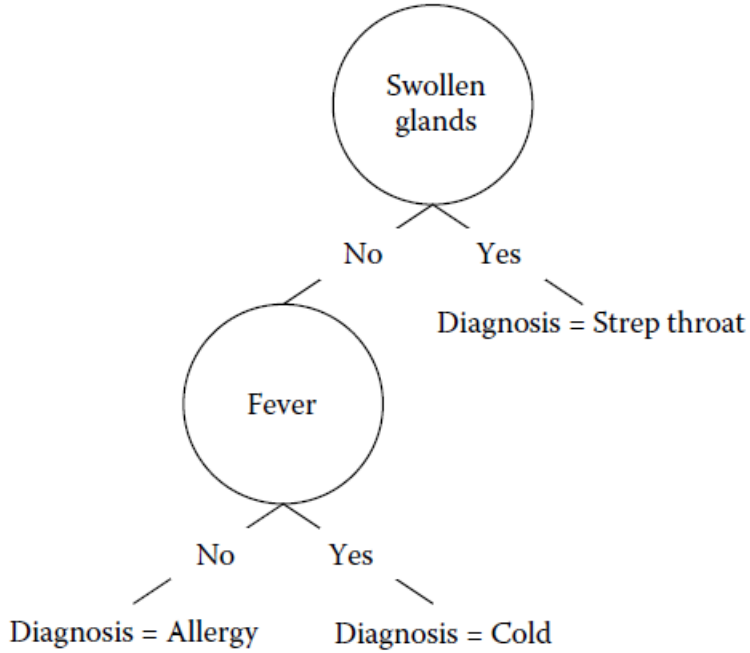
| Patient ID | Sore Throat | Fever | Swollen Glands | Congestion | Headache | Diagnosis |
|---|---|---|---|---|---|---|
| 1 | Yes | Yes | Yes | Yes | Yes | Strep throat |
| 2 | No | No | No | Yes | Yes | Allergy |
| 3 | Yes | Yes | No | Yes | No | Cold |
| 4 | Yes | No | Yes | No | No | Strep throat |
| 5 | No | Yes | No | Yes | No | Cold |
| 6 | No | No | No | Yes | No | Allergy |
| 7 | No | No | Yes | No | No | Strep throat |
| 8 | Yes | No | No | Yes | Yes | Allergy |
| 9 | No | Yes | No | Yes | Yes | Cold |
| 10 | Yes | Yes | No | Yes | Yes | Cold |

# Automatische Klassifizierung: Entscheidungsbäume (C4.5)

- Wenn ein Patient geschwollene Drüsen hat, lautet die Krankheitsdiagnose Halsentzündung.
- Wenn ein Patient keine geschwollenen Drüsen hat und Fieber hat, ist die Diagnose eine Erkältung.
- Wenn ein Patient keine geschwollenen Drüsen hat und kein Fieber hat, lautet die Diagnose eine Allergie

# Automatische Klassifizierung: Entscheidungsbäume (C4.5)

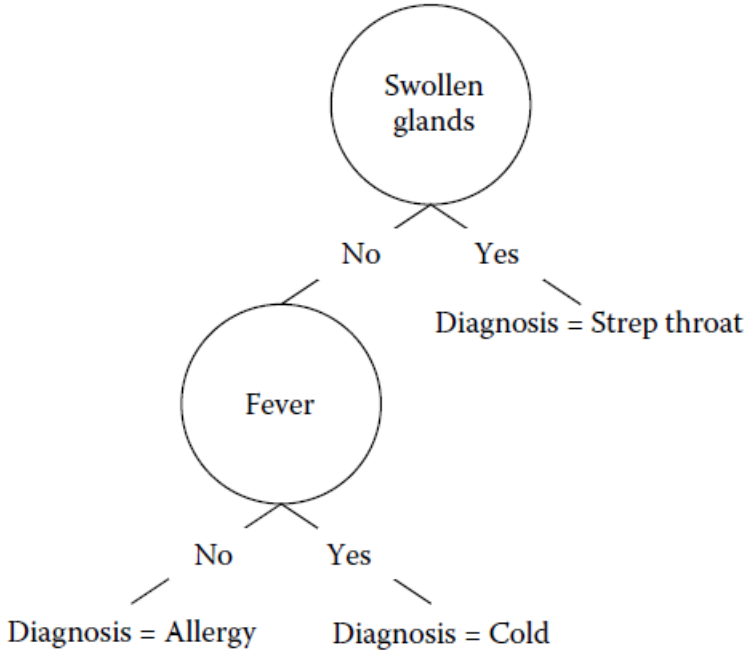
- Der Entscheidungsbaum sagt uns, dass wir den Patienten in diesem Datensatz genau diagnostizieren können, falls die Patienten geschwollene Drüsen und Fieber haben.
- Die Merkmale Halsschmerzen, Anschoppung (congestion) und Kopfschmerzen spielen für die Diagnose keine Rolle.



# Automatische Klassifizierung: Entscheidungsbäume (C4.5)

| Patient ID | Sore Throat | Fever | Swollen Glands | Congestion | Headache | Diagnosis |
|---|---|---|---|---|---|---|
| 11 | No | No | Yes | Yes | Yes | ? |
| 12 | Yes | Yes | No | No | Yes | ? |
| 13 | No | No | No | No | Yes | ? |

ID11: Angeschwolle Drüsen YES => Halsentzündung
ID12: Angeschwolle Drüsen NO => Fieber YES => Erkältung



# Decision Trees => Production Rules

IF *antecedent conditions* THEN *consequent conditions*

1. IF *Swollen Glands* = *Yes*

   THEN *Diagnosis* = *Strep Throat*

2. IF *Swollen Glands* = *No and Fever* = *Yes*

   THEN *Diagnosis* = *Cold*

3. IF *Swollen Glands* = *No and Fever* = *No*

   THEN *Diagnosis* = *Allergy*

# Production Rules for Classification

ID13: Geschwollene Halsdrüsen NO => Fieber YES => Allergie

Mit Regel 3

# Maschinelles Lernen: supervised learning

- Teilüberwachtes Lernen (semi-supervised learning): Nur für einen Teil der Eingaben sind die dazugehörigen Ausgaben bekannt.
- Bestärkendes Lernen (reinforcement learning): Der Algorithmus lernt durch Belohnung und Bestrafung eine Taktik, wie in potenziell auftretenden Situationen zu handeln ist, um den Nutzen des Agenten (d. h. des Systems, zu dem die Lernkomponente gehört) zu maximieren.
- Aktives Lernen (active learning): Der Algorithmus hat die Möglichkeit für einen Teil der Eingaben die korrekten Ausgaben zu erfragen. Dabei muss der Algorithmus die Fragen bestimmen, welche einen hohen Informationsgewinn versprechen, um die Anzahl der Fragen möglichst klein zu halten.

# Unüberwachtes Lernen (unsupervised learning)

- Der Algorithmus erzeugt für eine gegebene Menge von Eingaben ein Modell, das die Eingaben beschreibt und Vorhersagen ermöglicht.
- Clustering-Verfahren, die die Daten in mehrere Kategorien einteilen, die sich durch charakteristische Muster voneinander unterscheiden.
- Das Netz erstellt somit selbständig Klassifikatoren, nach denen es die Eingabemuster einteilt.

# Unüberwachtes Lernen (unsupervised learning)

| Customer ID | Account Type | Margin Account | Transaction Method | Trades/ Month | Gender | Age | Favorite Recreation | Annual Income |
|---|---|---|---|---|---|---|---|---|
| 1005 | Joint | No | Online | 12.5 | F | 30–39 | Tennis | 40–59K |
| 1013 | Custodial | No | Broker | 0.5 | F | 50–59 | Skiing | 80–99K |
| 1245 | Joint | No | Online | 3.6 | M | 20–29 | Golf | 20–39K |
| 2110 | Individual | Yes | Broker | 22.3 | M | 30–39 | Fishing | 40–59K |
| 1001 | Individual | Yes | Online | 5.0 | M | 40–49 | Golf | 60–79K |

# Unüberwachtes Lernen (unsupervised learning)

1. Kann man ein allgemeines Profil eines Online-Investors entwickeln? Wenn ja, welche sind die Merkmale von Online-Anlegern bzw., die von Anlegern, die einen Broker verwenden?

2. Kann man feststellen, ob ein neuer Kunde, der Anfangs kein Margin-Konto eröffnet, dies wahrscheinlich in der Zukunft tun wird?

3. Kann man ein Modell erstellen, mit dem die durchschnittliche Anzahl der Trades pro Monat für einen neuen Anleger genau vorhergesagt werden kann?

4. Welche Merkmale zeichnen weibliche und männliche Anleger aus?

→ Supervised learning!

# Unüberwachtes Lernen (unsupervised learning)

1. Welche sind die Gemeinsamkeiten der Acme Investoren?

2. Welche Unterschiede in den Attributwerten segmentieren die Kundendatenbank?

→Unsupervised Learning

→Wir brauchen eine Abschätzung der Anzahl der Clustern in den Daten.

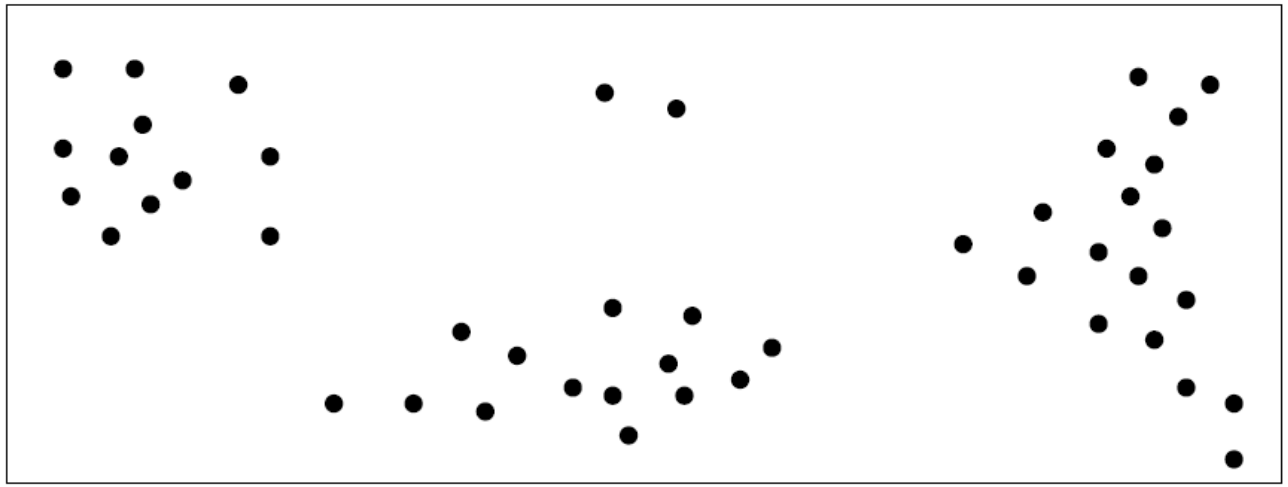
# Clustering: Beispiele

IF *Margin Account* = *Yes and Age* = 20–29 *and Income* = 40–59*K*
THEN *Cluster* = 1
Rule precision: 80%
Rule coverage: 25%
IF *Account Type* = *Custodial and Recreation* = *Skiing and Income* = 80–90*K*
THEN *Cluster* = 2
Rule precision: 95%
Rule coverage: 15%
IF *Account Type* = *Joint and Trades/Month* > 5 *and Transaction Method* = *Online*
THEN *Cluster* = 3
Rule precision: 82%
Rule coverage: 55%

# Clustering

- Suchmaschine: Decke
  - ➢Microfaser-Decke“, „unter der Decke“, etc.
  - ➢ „Stuck an der Decke“, „Wie streiche ich eine Decke“, etc.
  - ➢Textdokumente: zwei deutlich von einander unterscheidbare **Cluster**.
- Mars: Schockoriegel, Planet

# Clustering



- **Clustering:** Trainingsdaten sind nicht klassifiziert.
- Der Lehrer fehlt hier
- Finden von Strukturen
- Häufungen im Raum der Trainingsdaten finden
- Ganz wichtig: Wahl eines geeigneten Abstandsmaßes für Punkte

# Abstandsmaße

der Euklidische Abstand

$$d_e(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}.$$

Summe der Abstandsquadrate

$$d_q(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} (x_i - y_i)^2,$$

Manhattan-Abstand

$$d_m(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|$$

# Abstandsmaße

Abstand der maximalen Komponente

$$d_\infty(\mathbf{x}, \mathbf{y}) = \max_{i=1,\ldots,n} |x_i - y_i|$$

das normierte Skalarprodukt:

$$\frac{\mathbf{x}\ \mathbf{y}}{|\mathbf{x}|\ |\mathbf{y}|}$$

gibt die Ähnlichtkeit von $\mathbf{x}$ und $\mathbf{y}$ an. als Abstandsmaß zum Beispiel der Kehrwert

$$d_s(\mathbf{x}, \mathbf{y}) = \frac{|\mathbf{x}|\ |\mathbf{y}|}{\mathbf{x}\ \mathbf{y}}$$

# Beispiel: Suche in Volltextdatenbanken

Merkmale $x_1, \ldots, x_n$ als Komponenten des Vektors $\mathbf{x}$, werden wie folgt berechnet:

- Wörterbuch mit zum Beispiel 50000 Wörtern
- Vektor hat die Länge 50000
- $x_i$ = Häufigkeit des i-ten Wörterbuch-Wortes im Text.

fast alle Komponenten sind null

# K-means

- Anzahl der Cluster ist bekannt → K-means
- K Cluster werden durch ihre Mittelpunkte initialisiert.

# K-means

- Klassifikation aller Daten zum nächsten Clustermittelpunkt
- Neuberechnung der Clustermittelpunkte

Als Algorithmus ergibt sich das Schema

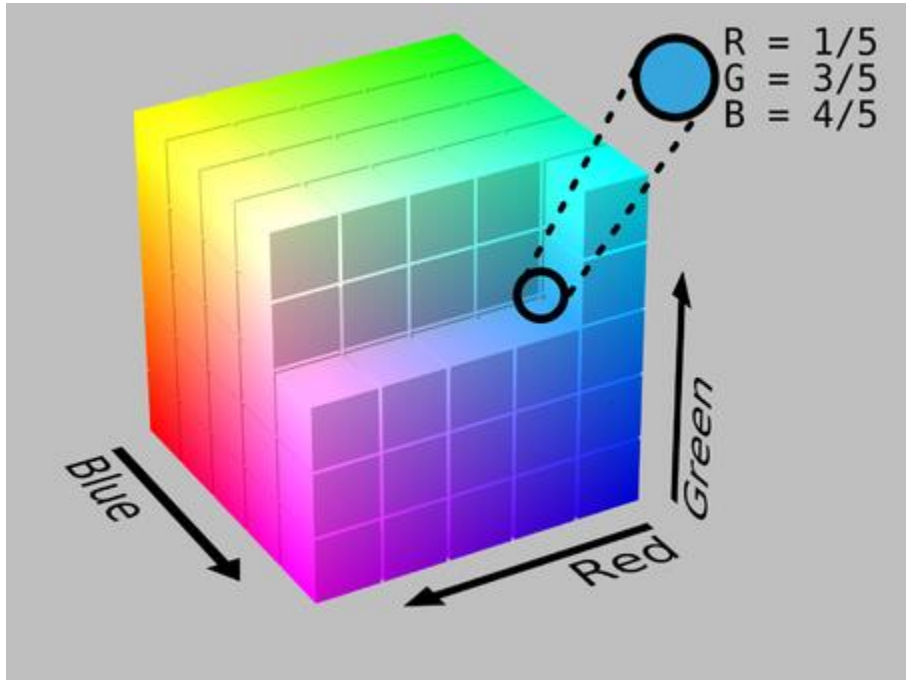
K-MEANS $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, k)$

initialisiere $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ (z.B. zufällig)
**Repeat**
    Klassifiziere $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ zum jeweils nächsten $\boldsymbol{\mu}_i$
    Berechne $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ neu
**Until** keine Änderung in $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$
**Return** $(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k)$

# K-means Beispiel

1. Farben aus einem Bild extrahieren:

- Jeder Datenpunkt ist eine Farbe und kann als Punkt in einem RGB-Farbraum dargestellt werden.
- Die Abstandsfunktion ist die Euklidische Metrik
- Initialisierung: Für die Anzahl der gewünschten Zentroide (k) kann man einen zufälligen ganzzahligen Punkt im Bereich des bereitgestellten Datensatzes generieren und an ein Array anhängen.
- Jeder Punkt im Array repräsentiert die Position eines Schwerpunkts. Die Zentroiden haben natürlich die gleiche Anzahl von Dimensionen wie die Daten.



# K-means Example
# https://github.com/xanderlewis/colour-palettes


# K-means Beispiel

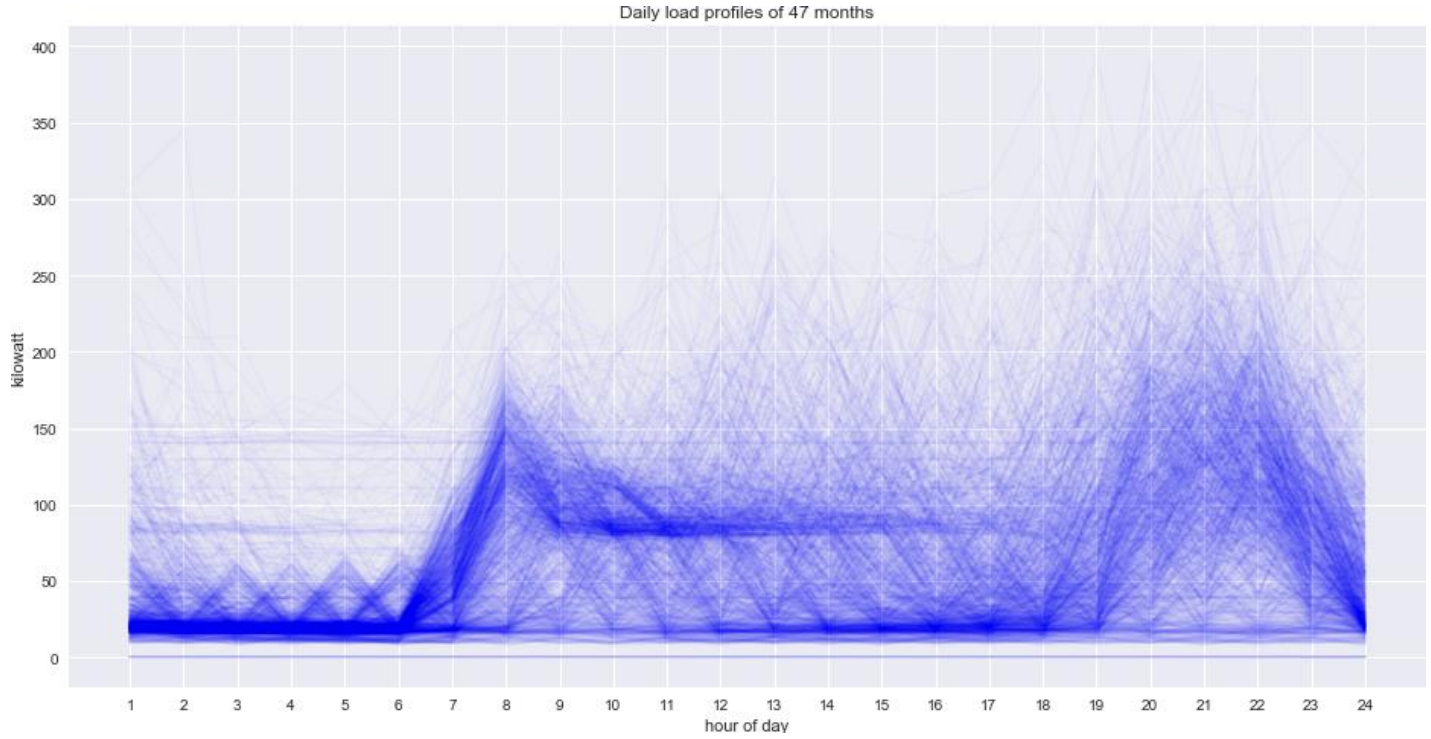
## 2. Stromverbrauchsprofile gruppieren

https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption

Der Datensatz enthält 2.075.259 Messungen, die zwischen Dezember 2006 und November 2010 (47 Monate) gesammelt wurden.

# K-means Beispiel

- Die täglichen Lastprofile von 1456 Tagen wurden zusammen aufgezeichnet.
- Wir können zwei klare Muster des Konsumverhaltens erkennen, wenn wir die dunkleren Regionen betrachten (in denen mehr Kurven konzentriert sind).



# K-means Beispiel

- Wie viele Cluster?
- Eine gängige Methode, um dies zu berechnen, ist die Verwendung der Silhouette-Methode.
- Es ist ein Maß dafür, wie ähnlich ein Punkt seinem eigenen Cluster im Vergleich zu anderen Clustern ist.
- Sie reicht von -1 bis 1, wobei ein hoher Wert angibt, dass ein Punkt gut mit dem Cluster übereinstimmt, zu dem er gehört.
- Die Silhouette kann mit jeder Entfernungsmetrik berechnet werden, z. B. der euklidischen Entfernung oder der Manhattan-Entfernung.

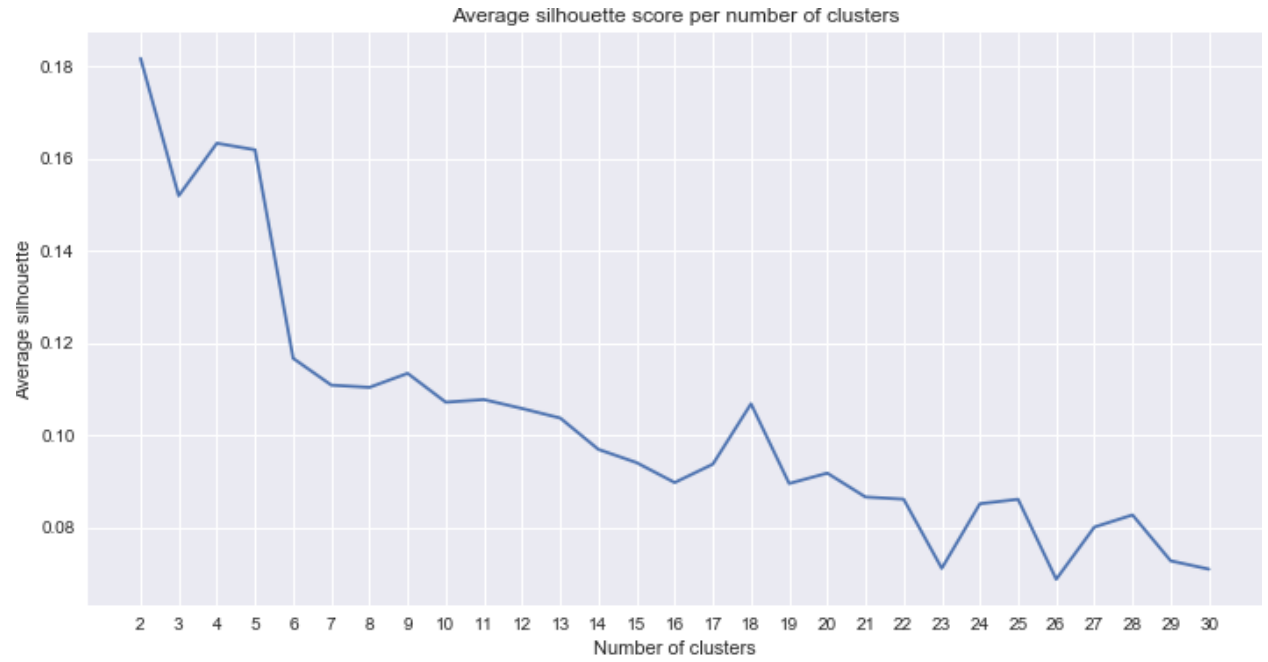
- Sei *b(i)* die kleinste durchschnittliche Entfernung von i zu allen Punkten in einem anderen Cluster, von denen *i* kein Mitglied ist.
- Der Cluster mit dieser niedrigsten durchschnittlichen Unähnlichkeit ist der "Nachbarschaftscluster" von *i* weil es der nächstbeste Cluster für Punkt *i* ist.
- Die Silhouette ist definiert als

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

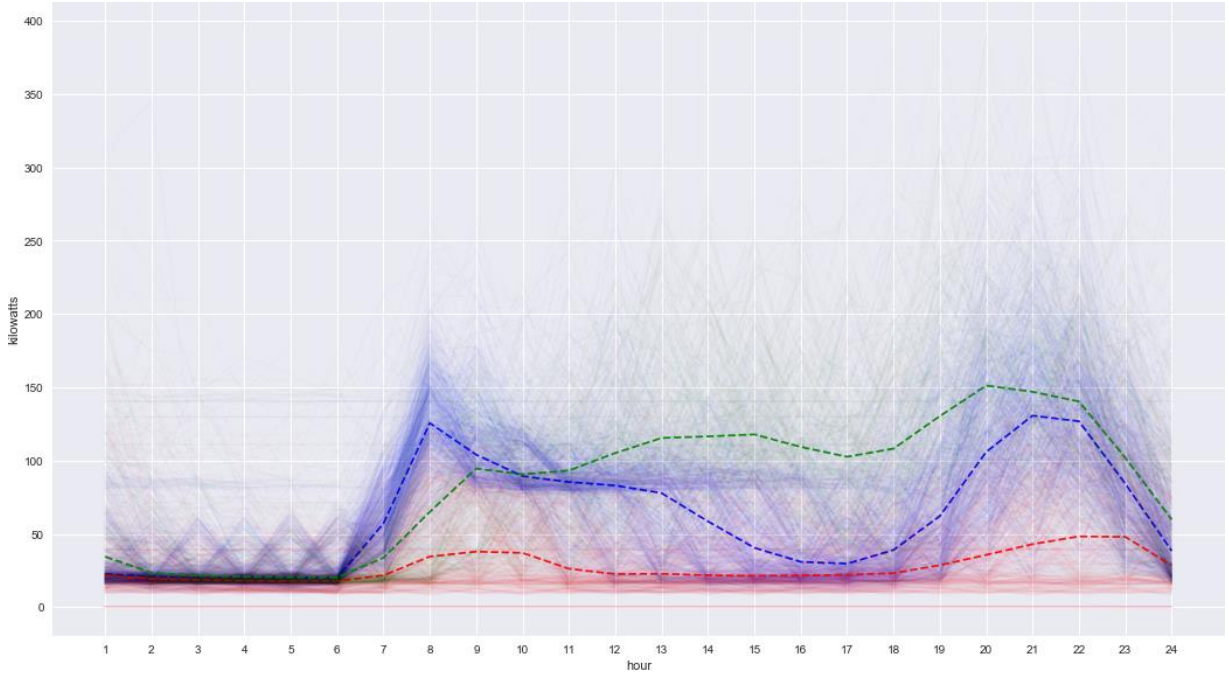
# K-means Beispiel

- Um eine globale Ansicht der Leistung des Algorithmus zu erhalten, kann der Durchschnitt der Silhouetten über alle Lastprofile hinweg ermittelt werden.
- Es ist wichtig, jede Periode innerhalb desselben Bereichs zu skalieren, damit die Größe der Energielast die Auswahl des Clusters nicht beeinträchtigt.



# K-means Beispiel

- Der grüne Cluster enthält Lasten, die den ganzen Nachmittag über Energie verbrauchen. Vielleicht sind dies Tage, an denen die Bewohner zu Hause blieben, wie Wochenenden und besondere Termine.
- Der blaue Cluster hat am Morgen einen hohen Gipfel und am Nachmittag eine Abnahme und in der Nacht wieder einen hohen Verbrauch. Dieses Muster scheint an Werktagen zu passen, wenn die Bewohner zur Arbeit und / oder zur Schule gehen.
- Schließlich zeigt der rote Cluster Tage an, an denen der Verbrauch den ganzen Tag über niedrig ist. Vielleicht Ferien, wenn nur noch wenige Geräte übrig sind?



# K-means

Zahl der Cluster ist bekannt! **k-Means**-Verfahren:

- Zuerst werden die $k$ Clustermittelpunkte $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ zufällig oder manuell initialisiert.
- Dann wird wiederholt:
  - Klassifikation aller Daten zum nächsten Clustermittelpunkt
  - Neuberechnung der Clustermittelpunkte

# K-means Algorithmus

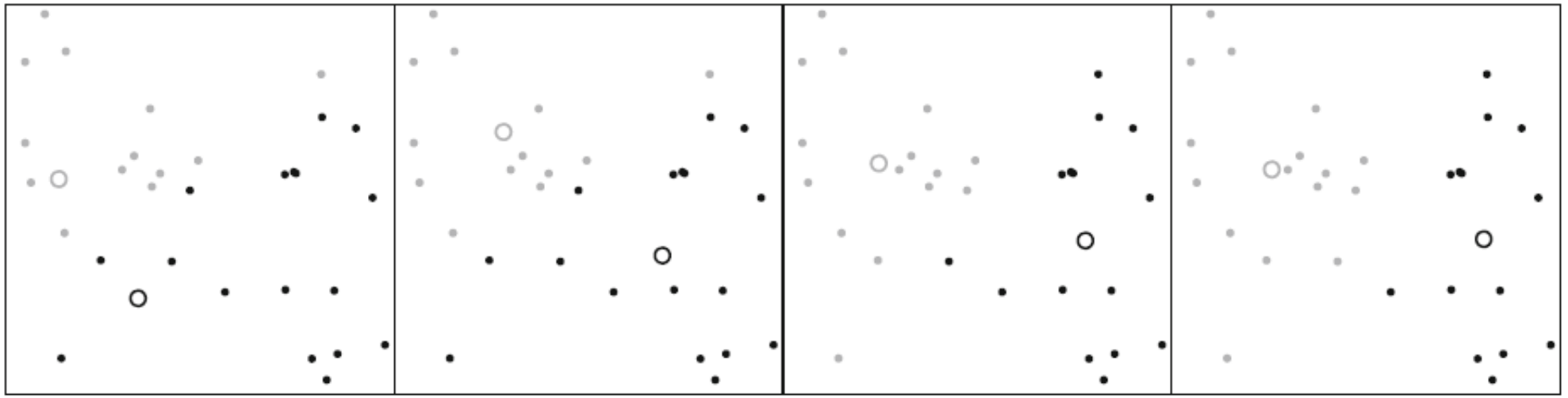
k-Means($\mathbf{x}_1, \ldots, \mathbf{x}_n, k$)

initialisiere $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ (z.B. zufällig)
**Repeat**
    Klassifiziere $\mathbf{x}_1, \ldots, \mathbf{x}_n$ zum jeweils nächsten $\boldsymbol{\mu}_i$
    Berechne $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ neu
**Until** keine Änderung in $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$
**Return**$(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k)$

Berechnung des Clustermittelpunktes:

$$\boldsymbol{\mu} = \frac{1}{l} \sum_{i=1}^{l} \mathbf{x}_i.$$

# K-means




k-Means mit zwei Klassen (k = 2) angewendet auf 30 Datenpunkte. *Ganz links* die Datenmenge mit den initialen Zentren und *nach rechts* die Cluster nach jeder Iteration. Nach drei Iterationen ist die Konvergenz erreicht

# K-means Algorithmus

- keine Konvergenzgarantie, aber meist recht schnelle Konvergenz.
- Zahl der Iterationsschritte meist viel kleiner als Zahl der Punkte.
- Komplexität: O(ndkt), mit
  - ✓n = Gesamtzahl der Punkte,
  - ✓d = Dimension des Merkmalsraumes
  - ✓t = Zahl der Iterationsschritte