



# Einführung in das Maschinelle Lernen

Dr. Houssam Jedidi



Automatisierung verschiedener (oft mühsamer) Prozesse



Teilbereich der Künstlichen Intelligenz (KI)



Eines der Ziele hinter dem maschinellen Lernen: die Notwendigkeit der "manuellen" Entwicklung von Computerprogrammen zu ersetzen



Programme wurden entwickelt um Prozesse zu automatisieren → ML ist "Automatisierung der Automatisierung"

Kategorie	Maschinelles Lernen	Statistik
Ziel	Prognose	Kausale Effekte identifizieren
Erklärung	Nutzung von Datenmustern, um zukünftige Ereignisse vorherzusagen	Entwicklung eines Forschungsdesigns, um kausale Beziehungen zwischen Variablen zu bestimmen
Forschungsfrage	Welche Methode liefert die beste Prognose?	Welchen Effekt hat X auf Y?
Beispiel	Mit gegebenen Variablen: Wie können wir das Ergebnis am besten prognostizieren?	Wie beeinflussen unabhängige Variablen die abhängige Variable?
Aufgabe	- Verschiedene Algorithmen ausprobieren und anpassen, um Vorhersagegenauigkeit zu maximieren	- Forschungsdesign basierend auf Theorie entwickeln
	- Theorie über Datenentstehung ist nützlich, aber oft nicht erforderlich	- Zufällige Experimente oder Beobachtungsstudien mit Kontrollvariablen verwenden
	- Fokus auf unbekannte Testdaten und Vermeidung von Überanpassung	- Hypothesen im Voraus festlegen, keine explorative Modellspezifikationssuche
Parameter	Accuracy, Precision, Recall, Sensitivity	Kausale Effektgröße, P-Werte
Hinweise	- Teilmenge der Daten für Validierung/Testdaten reservieren	- Variablen, die den Effekt verdecken/verfälschen könnten, vermeiden
	- Ziel: Überanpassung verhindern	
Überblick	- Alle relevanten Merkmale nutzen, die sich als nützlich für die Prognose erweisen	- Gesamte Stichprobe nutzen, da diese oft repräsentativ für eine Population ist

## Anwendungen des maschinellen Lernens

Email Spam-Erkennung

Gesichtserkennung und  
-abgleich

Websuche

Sports Prognosen

Kreditkartenbetrug

Vorhersagen zur Aktie

Textanalyse (TM, NLP...)

Mitarbeiterfluktuation

Kundenabwanderung

Während wir einige dieser Anwendungen im Unterricht besprechen, ist es eine gute Übung, darüber nachzudenken, wie maschinelles Lernen in den oben genannten Problembereichen oder Aufgaben angewendet werden könnte:

Was ist das gewünschte Ergebnis?

Wie könnte der Datensatz aussehen?

Wie würden Sie den Erfolg messen?

Was sind mögliche Herausforderungen oder Schwierigkeiten?

## Überwachtes Lernen

- Beschriftete Daten
- direkte Rückmeldung
- Vorhersage Ergebnis Zukunft

## Unüberwachtes Lernen

- Keine Kennzeichnungen /Ziele
- Keine Rückmeldung
- Verborgene Struktur in Daten finden

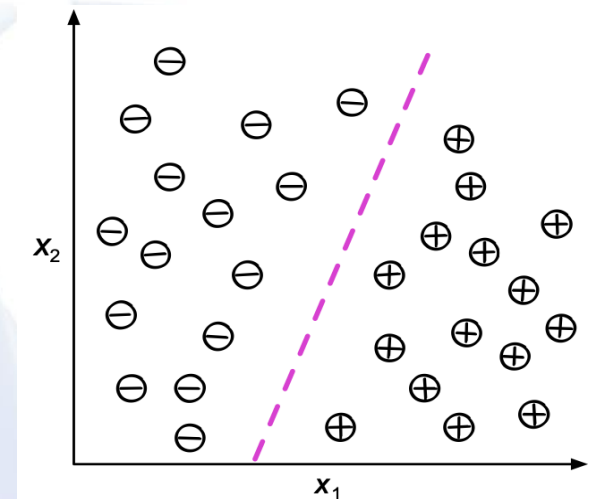
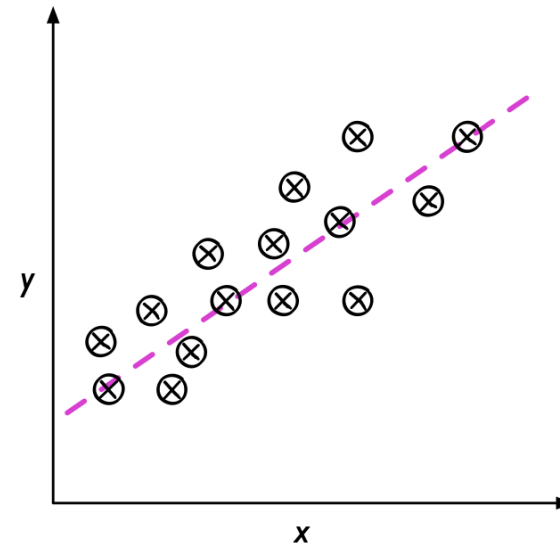
## Verstärkungslernen

- Entscheidungsprozess
- Belohnungssystem
- Lernen Sie eine Reihe von Aktionen



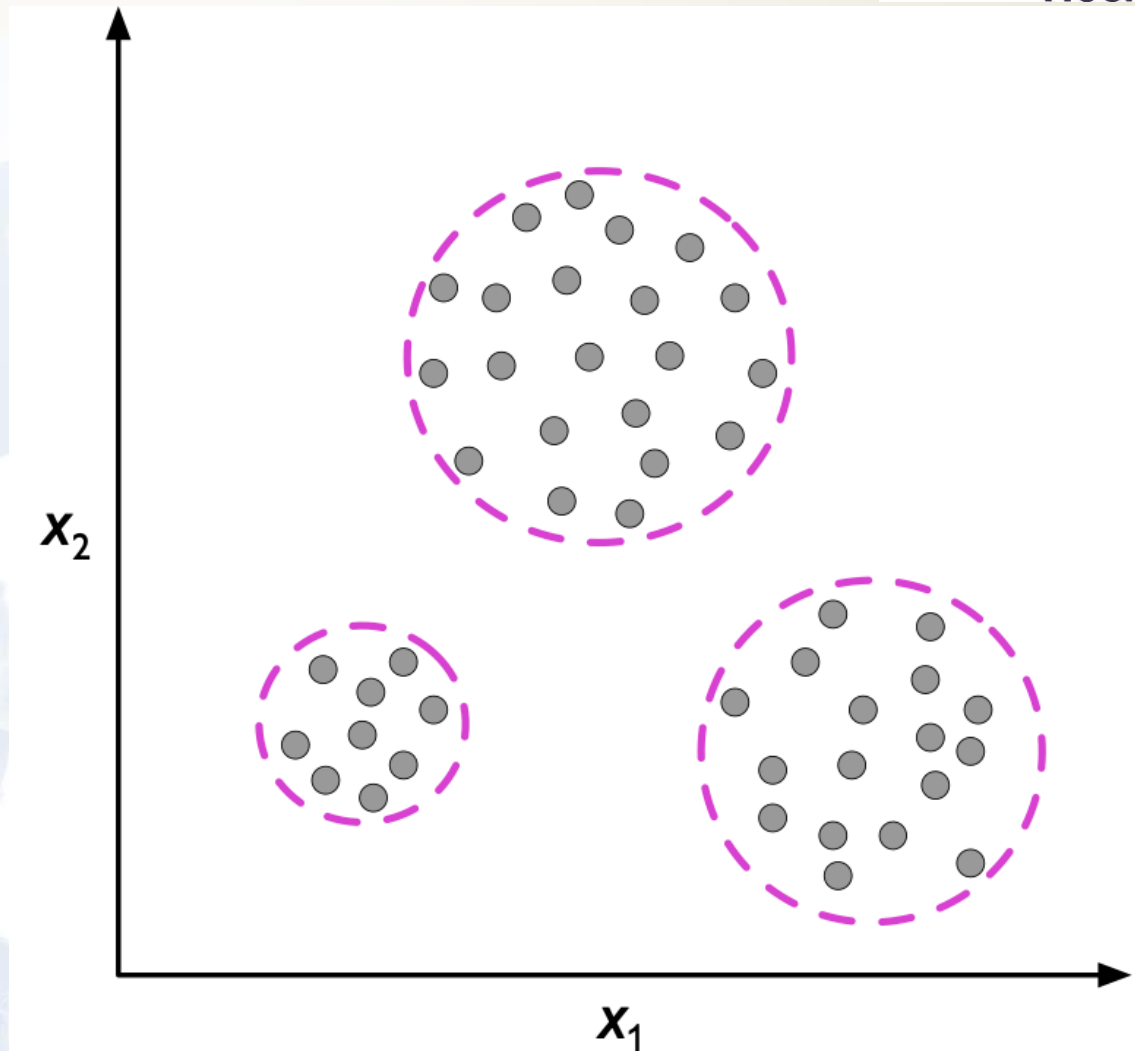
# A- Überwachtes Lernen

Überwachtes Lernen ist die Unterkategorie des maschinellen Lernens, die sich auf das Erlernen einer Klassifizierung (Abbildung 2) oder eines Regressionsmodells (Abbildung 1) konzentriert, d. h. auf das Lernen aus markierten Trainingsdaten (d. h. Eingaben, die auch die gewünschten Ausgaben oder Ziele enthalten; im Grunde "Beispiele" für das, was wir vorhersagen wollen).



# B- Unüberwachtes Lernen

Im Gegensatz zum überwachten Lernen ist das unüberwachte Lernen ein Zweig des maschinellen Lernens, der sich mit nicht beschrifteten Daten befasst. Gängige Aufgaben beim unüberwachten Lernen sind die Clusteranalyse (Zuweisung von Gruppenzugehörigkeiten) und die Multidimensionale Skalierung (Komprimierung von Daten)





## C-

# Verstärkungslernen

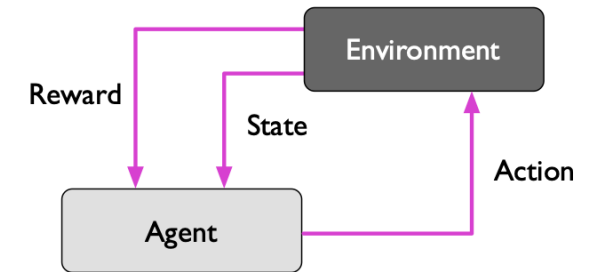
Verstärkung ist der Prozess des Lernens aus Belohnungen bei der Durchführung einer Reihe von Aktionen. Beim Verstärkungslernen sagen wir dem Lernenden oder Agenten (z. B. einem Roboter) nicht, welche Aktion er ausführen soll, sondern weisen jeder Aktion und/oder dem Gesamtergebnis lediglich eine Belohnung zu (für Maschinen, die mit ihrer Umgebung interagieren).

Anstelle von "richtig/falsch"-Etiketten für jeden Schritt muss der Lernende ein Verhalten entdecken oder lernen, das die Belohnung für eine Reihe von Aktionen maximiert.

In diesem Sinne handelt es sich nicht um eine überwachte Umgebung. RL ist in gewisser Weise mit dem unüberwachten Lernen verwandt; das verstärkende Lernen ist jedoch eine eigene Kategorie des maschinellen Lernens.

**Bestärkendes lernen wird in diesem Kurs nicht weiter behandelt.**

Typische Anwendungen des verstärkenden Lernens sind Spiele (Schach, Go, Atari-Videospiele) und irgendeine Form von Robotern, z. B. Drohnen, Lagerroboter und neuerdings auch selbstfahrende Autos.



# D- Semi- Überwachtes Lernen

Semi-Überwachtes Lernen lässt sich grob gesagt als eine Mischung aus überwachtem und unüberwachtem Lernen beschreiben.

Bei halbüberwachten Lernaufgaben enthalten einige Trainingsbeispiele Ausgaben, einige jedoch nicht. Wir verwenden dann die beschriftete Trainingsuntermenge, um den nicht beschrifteten Teil der Trainingsmenge zu beschriften, den wir auch zum Trainieren des Models verwenden.

# 1- Einführung in das überwachte Maschinelle Lernen

Wir werden uns auf überwachtes Lernen und Klassifizierung konzentrieren, die am weitesten verbreitete Form des maschinellen Lernens.

Wir werden jedoch auch einige Regressionsbeispiele in den Vorlesungen sehen, und es wird später in diesem Kurs auch Bsp. zum unüberwachten Lernen geben

Beim überwachten Lernen erhalten wir einen beschrifteten Trainingsdatensatz, aus dem ein Algorithmus für maschinelles Lernen ein Modell lernen kann.

Das gelernte (oder trainierte) Modell kann zur Vorhersage der Bezeichnungen von nicht beschrifteten Datenpunkten verwendet werden.

Bei diesen nicht beschrifteten Datenpunkten kann es sich entweder um Testdaten handeln (für die wir eigentlich Beschriftungen haben, die wir aber zu Testzwecken zurückhalten) oder um nicht beschriftete Daten, die wir bereits gesammelt haben oder in Zukunft sammeln werden.

Bei einem Korpus von Spam- und Nicht-Spam-E-Mails bestünde eine überwachte Lernaufgabe beispielsweise darin, ein Modell zu lernen, das vorhersagt, zu welcher Klasse (Spam oder Nicht-Spam) neue E-Mails gehören. Dies alles setzt natürlich voraus, dass der Trainingsdatensatz und die unmarkierten Datenpunkte für die Vorhersage aus derselben Wahrscheinlichkeitsverteilung gezogen wurden - schließlich können wir nicht erwarten, dass das Modell zuverlässige Vorhersagen für grundlegend unterschiedliche Daten macht.

# 1- Einführung in das überwachte Maschinelle Lernen

Formaler ausgedrückt, definieren wir  $h$  als "Hypothese", eine Funktion, die wir zur Annäherung an eine unbekannte Funktion verwenden:

$$f(x) = y$$

wobei  $x$  ein Vektor von Eingabemerkmalen ist, die mit einem Trainingsbeispiel oder einer Datensatzinstanz verbunden sind (z. B. die Pixelwerte eines Bildes), und  $y$  das Ergebnis ist, das wir vorhersagen wollen (z. B. welche Klasse von Objekt wir in einem Bild sehen). Mit anderen Worten:  $h(x)$  ist eine Funktion, die  $y$  vorhersagt.



Bei der Klassifizierung definieren wir die Hypothesenfunktion als

$$h : X \rightarrow Y$$

mit  $X = \mathbb{R}^m$  und  $Y = \{1, \dots, k\}$  mit den Klassenbezeichnungen  $k$ . Im Spezialfall der binären Klassifikation ist  $Y = \{0, 1\}$  (alternativ kann auch  $Y = \{-1, 1\}$  verwendet werden).

Und bei der Regression besteht die Aufgabe darin, eine Funktion zu lernen:

$$h : \mathbb{R}^m \rightarrow \mathbb{R}.$$

Gegeben eine Trainingsmenge:

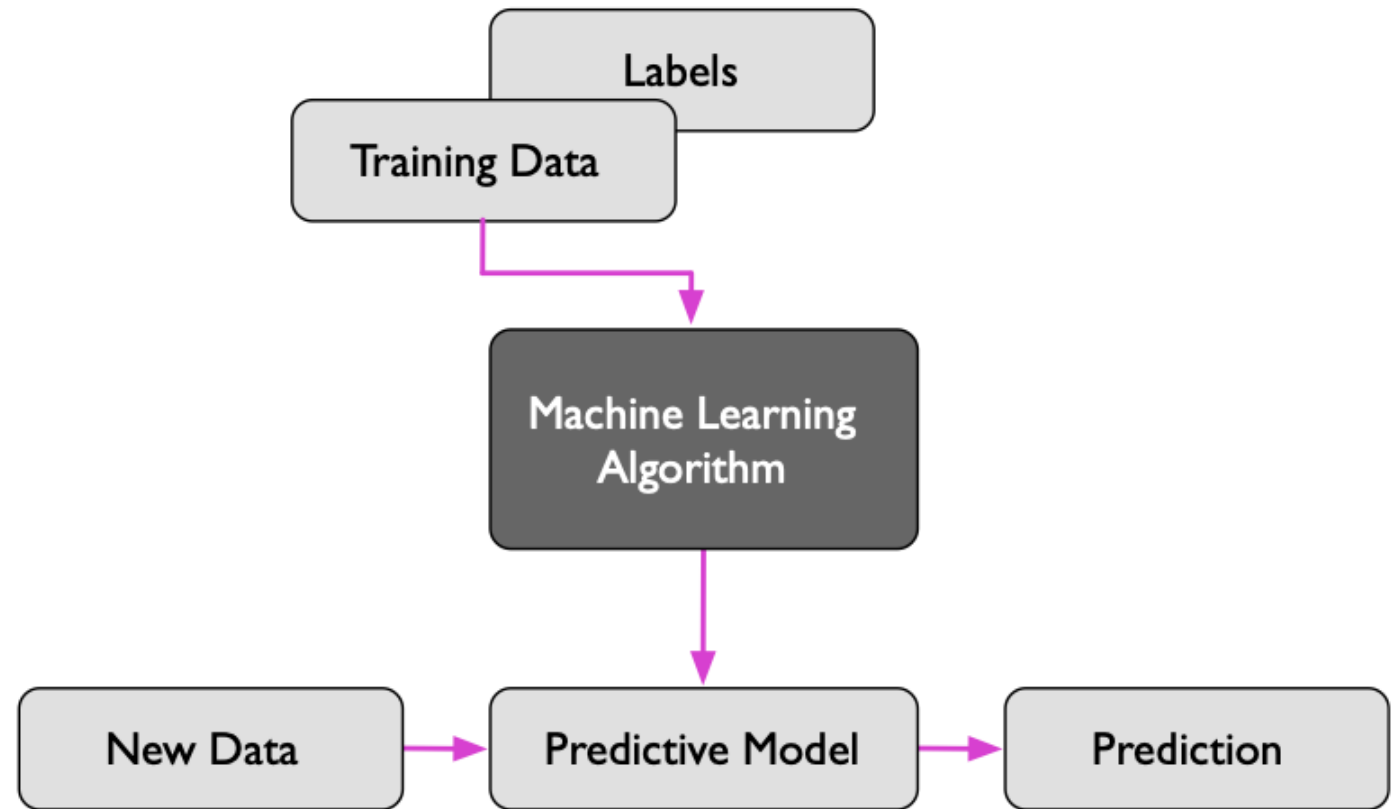
$$\mathcal{D} = \{\langle \mathbf{x}^{[i]}, y^{[i]} \rangle, i = 1, \dots, n\},$$

Eine kritische Annahme für die Theorie des maschinellen Lernens (und die meisten davon) ist, dass die Trainingsbeispiele i.i.d. (unabhängig und identisch verteilt) sind.



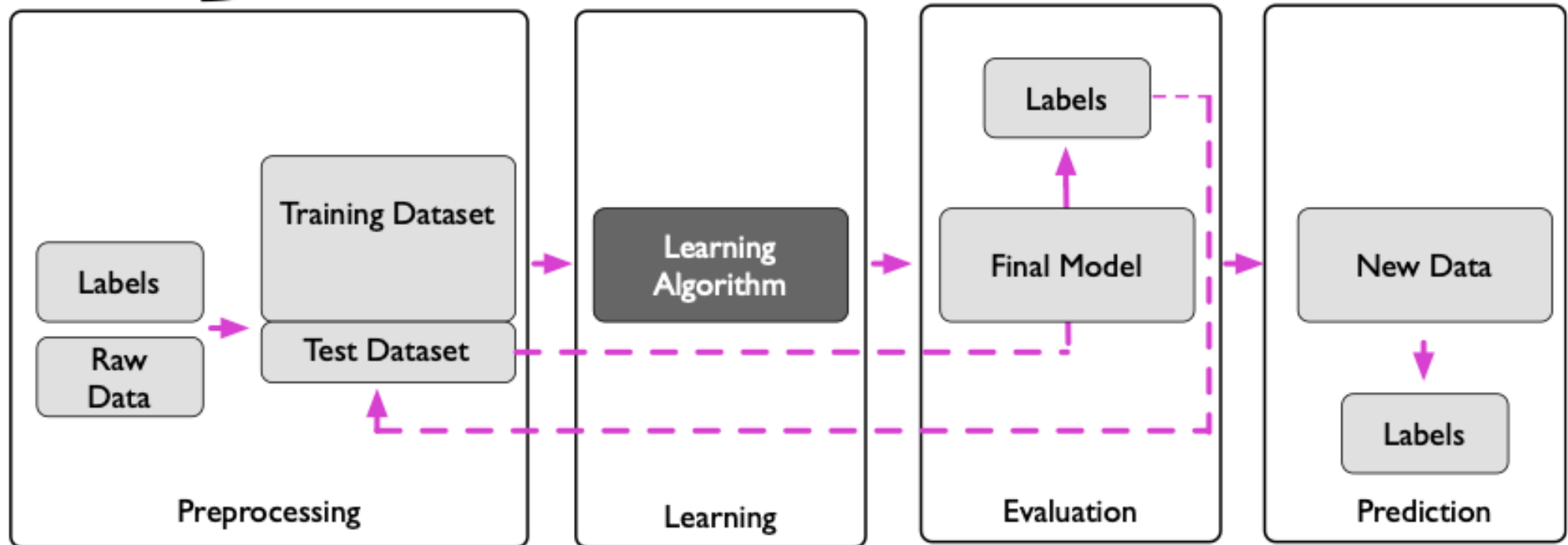
## 1- Einführung in das überwachte Maschinelle Lernen

Grober Überblick über den Prozess des  
überwachten Lernens



Merkmalsextraktion und  
Skalierung  
Merkmalauswahl  
Dimensionalitätsreduktion  
Stichprobenverfahren

Modellauswahl Kreuz-  
Validierung  
Leistungsmetriken  
Hyperparameter-Optimierung



## 1.1 Notation für statistisches Lernen

variiert in der Literatur stark

Datensatz:

$$\mathcal{D} = \{\langle \mathbf{x}^{[1]}, y^{[1]} \rangle, \langle \mathbf{x}^{[2]}, y^{[2]} \rangle \dots, \langle \mathbf{x}^{[n]}, y^{[n]} \rangle\}$$

Instanzen von zwei Zufallsvariablen:

$$X \in \mathcal{X} \quad \text{und} \quad Y \in \mathcal{Y}$$

die Trainingspaare werden aus einer gemeinsamen Verteilung gezogen

$$P(X, Y) = P(X)P(Y|X)$$

Bei einem Fehlerterm  $\epsilon$  können wir dann die folgende Beziehung formulieren:  $Y = f(X) + \epsilon$ .

Das Ziel des statistischen Lernens besteht dann darin,  $f$  zu schätzen, das dann zur Vorhersage von  $Y$  verwendet werden kann:

$$\hat{f}(X) = \hat{Y}.$$

## 1.2 Notation für statistisches Lernen

Im vorigen Abschnitt haben wir das  $i$ -te Paar in einer beschrifteten Trainingsmenge  $\mathcal{D}$  als  $\langle \mathbf{x}^{[i]}, y^{[i]} \rangle$  bezeichnet. Wir werden die Konvention anwenden, kursive Schrift für Skalare, fette Schrift für Vektoren und Fettschrift für Matrizen.

- $x$ : Ein Skalar, der ein einzelnes Trainingsbeispiel mit 1 Merkmal bezeichnet (z. B. die Größe einer Person)
- $\mathbf{x}$ : Ein Trainingsbeispiel mit  $m$  Merkmalen (z. B. könnten wir mit  $m = 3$  die Größe, das Gewicht und das Alter einer Person darstellen), dargestellt als Spaltenvektor (d. h., eine Matrix mit 1 Spalte,  $\mathbf{x} \in \mathbb{R}^m$ )

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}.$$

## 1.2 Notation für statistisches Lernen

Um den Index des Merkmals und den Index des Trainingsbeispiels zu unterscheiden, verwenden wir eine hochgestellte eckige Klammer, um das i-te Trainingsbeispiel zu bezeichnen, und eine normale tiefgestellte Klammer, um das j-te Merkmal zu bezeichnen

Die entsprechenden Ziele werden in einem Spaltenvektor  $y$  dargestellt

$$\mathbf{X} = \begin{bmatrix} x_1^{[1]} & x_2^{[1]} & \cdots & x_m^{[1]} \\ x_1^{[2]} & x_2^{[2]} & \cdots & x_m^{[2]} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{[n]} & x_2^{[n]} & \cdots & x_m^{[n]} \end{bmatrix}.$$

$$\mathbf{y} = \begin{bmatrix} y^{[1]} \\ y^{[2]} \\ \vdots \\ y^{[n]} \end{bmatrix}.$$

### Datensatz:

1000 Mitarbeiter

### Spalten:

- Vor & Nachname
- Alter
- Geschlecht
- Sozialer Status
- Einkommen
- Studium
- Fachgebiet (Studium / Arbeit)
- Einstellungsart (Praktikum, Teilzeit, Vollzeit) & Einstellungsform (un-befristet)
- Verhältnis: noch Mitarbeiter / gekündigt worden / hat gekündigt



# Türen<4	Ps >350	Höhe<1.4m	Länge< 4565 mm	Klassenlabel
Richtig	Richtig	Richtig	Richtig	Sportwagen
Falsch	Richtig	Falsch	Falsch	Kombi
Richtig	Richtig	Richtig	Richtig	Sportwagen
...				

In der Regel ist die Anzahl der erforderlichen Trainingsbeispiele proportional zur Flexibilität der Lernalgorithmen. D.h., wir brauchen mehr Trainingsdaten (markierte Beispiele) für Modelle mit

- einer großen Anzahl von Parametern, die angepasst werden müssen;
- eine große Anzahl von einzustellenden "Reglern" (Hyperparametern);
- eine große Anzahl von Hypothesen.

## 2- Klassen von Algorithmen für maschinelles Lernen

Klasse	Beispiele	Eigenschaften	Stärken	Schwächen
Generalized Linear Models (GLMs)	Logistische Regression, Lineare Regression	Modelliert lineare Beziehungen; benötigt oft Feature-Skalierung	Einfach, effizient, gut interpretierbar	Nicht geeignet für nicht-lineare Beziehungen
Support Vector Machines (SVMs)	Lineare SVM, RBF-Kernel SVM	Findet optimale Trennlinien; Kernel-Funktionen für nicht-lineare Daten	Gute Performance bei hochdimensionalen Daten	Langsam bei großen Datenmengen, Kernel-Auswahl komplex
Artificial Neural Networks (ANNs)	Multi-Layer Perceptrons (MLP), CNN, RNN	Inspiriert von neuronalen Netzen; kann nicht-lineare und komplexe Beziehungen modellieren	Sehr leistungsstark für komplexe Probleme	Hoher Rechenaufwand, schwer interpretierbar
Tree- or Rule-Based Models	Entscheidungsbäume, CART	Hierarchische Entscheidungsregeln; teilt Daten in Teilmengen auf	Einfach zu verstehen, gut interpretierbar	Neigen zur Überanpassung, bei hohen Dimensionen ineffizient
Graphical Models	Bayessche Netzwerke, Markov Random Fields	Modelliert probabilistische Beziehungen zwischen Variablen als Graph	Gut bei Unsicherheiten und probabilistischen Beziehungen	Komplex in der Implementierung
Ensembles	Random Forest, XGBoost, AdaBoost	Kombiniert mehrere Modelle (Bagging/Boosting); verbessert Genauigkeit und Robustheit	Hohe Genauigkeit, robust gegen Überanpassung	Weniger interpretierbar
Instance-Based Learners	K-Nearest Neighbors (KNN)	Nutzt gesamte Trainingsdaten; Vorhersagen basieren auf Ähnlichkeit zu bereits bekannten Instanzen	Einfach, keine Trainingsphase nötig	Langsam bei großen Datenmengen

## 2.1- Kategorisierungsschemata

### A- Eager vs. Lazy (eifrig vs. faul)

Eifrige Lerner sind Algorithmen, die Trainingsdaten sofort verarbeiten, während faule Lerner den Verarbeitungsschritt bis zur Vorhersage aufschieben.

Eifrigen Lernern: Anstatt die gesamten Daten zu speichern, werden die Muster und Zusammenhänge in Form eines Modells erfasst (z. B. Gewichtsmatrizen, Entscheidungsbäume)

Da das Modell bereits abstrahiert wurde, sind Vorhersagen typischerweise schneller als bei Lazy Learning  
Änderungen in den Daten erfordern häufig eine Aktualisierung oder ein erneutes Training des Modells

Faulenzer haben keinen expliziten Trainingsschritt außer der Speicherung der Trainingsdaten.

Ein beliebtes Beispiel für einen „lazy learner“ ist der Nearest Neighbor Algorithmus.

Alle Berechnungen, die für die Vorhersage notwendig sind, finden ad hoc statt.

Hohe Speicheranforderungen, insbesondere bei großen Datensätzen

Bei jeder Anfrage werden die Daten durchsucht, analysiert und verarbeitet, um eine Vorhersage zu generieren

## 2.1- Kategorisierungsschemata

### B- Batch- vs. Online-Lernen

Batch-Lernen bezieht sich auf die Tatsache, dass das Modell mit der gesamten Menge der Trainingsbeispiele gelernt wird.

Neue Daten oder Veränderungen in den Daten erfordern ein erneutes Training mit dem vollständigen, aktualisierten Datensatz.

Da alle Daten gleichzeitig verarbeitet werden, ist der Trainingsprozess oft ressourcenintensiv und zeitaufwendig. Batch Learning wird häufig in Szenarien eingesetzt, in denen die Datenbasis **stabil** und unveränderlich ist.

Online Lerner:

Online-Lernprogramme hingegen lernen jeweils mit einem einzigen Trainingsbeispiel.

In praktischen Anwendungen ist es nicht unüblich, ein Modell durch Batch-Lernen zu erlernen und es dann später durch Online-Lernen zu aktualisieren.

Das Modell kann sich dynamisch an Veränderungen in den Daten (z. B. Konzeptdrift) anpassen.

Geringer Speicherbedarf & Echtzeitfähigkeit

## 2.1- Kategorisierungsschemata

### C- Parametrische vs. nicht-parametrische Modelle

Parametrische Modelle sind "feste" Modelle, bei denen wir eine bestimmte Funktionsform für  $f(x) = y$  annehmen.

Als ein parametrisches Modell mit  $h(x) = w_1x_1 + \dots + w_mx_m + b$ .

Feste Anzahl von Parametern (unabhängig von der Größe des Datensatzes).

Erfordern oft Annahmen über die Form oder Struktur der zugrunde liegenden Daten (z. B. Linearität oder Normalverteilung).

--> In der Regel schnell und Effizient (GLM, SVM mit linearem Kernel, NB)

! Nicht immer flexibel (komplexe oder nicht lineare Muster in den Daten)

Nichtparametrische Modelle sind "flexibler" und haben keine vorgegebene Anzahl von Parametern. Tatsächlich wächst die Anzahl der Parameter in der Regel mit der Größe der Trainingsmenge.

Da das Modell oft die Trainingsdaten (oder eine große Menge davon) speichert, können Speicheranforderungen hoch sein.

Entscheidungsbaum ist ein **nicht-parametrisches Modell**, weil er keine feste Anzahl von Parametern hat, die man vorab festlegt. Stattdessen baut der Baum die Struktur (die Entscheidungsknoten und Verzweigungen) während des Trainingsprozesses basierend auf den Trainingsdaten auf. Die Anzahl und Art der Knoten hängt direkt von den Daten ab, was typisch für nicht-parametrische Modelle ist



## 2.1- Kategorisierungsschemata

### D- Diskriminierend versus generativ

Generative Modelle beschreiben (klassischerweise) Methoden, die die gemeinsame Verteilung  $P(X,Y) = P(Y)P(X|Y) = P(X)P(Y|X)$  für Trainingspaare  $\langle x^{[i]}, y^{[i]} \rangle$  modellieren.

Generative Modelle versuchen, die zugrunde liegende Verteilung der Eingabedaten und die Beziehung zwischen den Variablen zu verstehen.

Sie können neue, plausible Datenpunkte generieren, die ähnliche Eigenschaften wie die Trainingsdaten besitzen (NB, HMM- Hidden Markov Models, GMM- Gaussian Mixture Models, GANs- Generative Adversarial Networks, VAEs Variational Autoencoders).

Diskriminierende Modelle verfolgen einen "direkteren" Ansatz und modellieren  $P(Y|X)$  direkt. Während generative Modelle in der Regel mehr Erkenntnisse liefern und Stichproben aus der gemeinsamen Verteilung ermöglichen, sind diskriminierende Modelle in der Regel einfacher zu berechnen und liefern genauere Vorhersagen.

Diskriminative Modelle können für Aufgaben eingesetzt werden, bei denen es auf präzise Vorhersagen ankommt (KNN, XGBoost , LightGBM, LM, CNN).

Hilfreich für das Verständnis diskriminierende Modelle ist die folgende Analogie:

*"discriminative modeling is like trying to extract information from text in a foreign language without fully learning that language".*



## 2.1- Kategorisierungsschemata

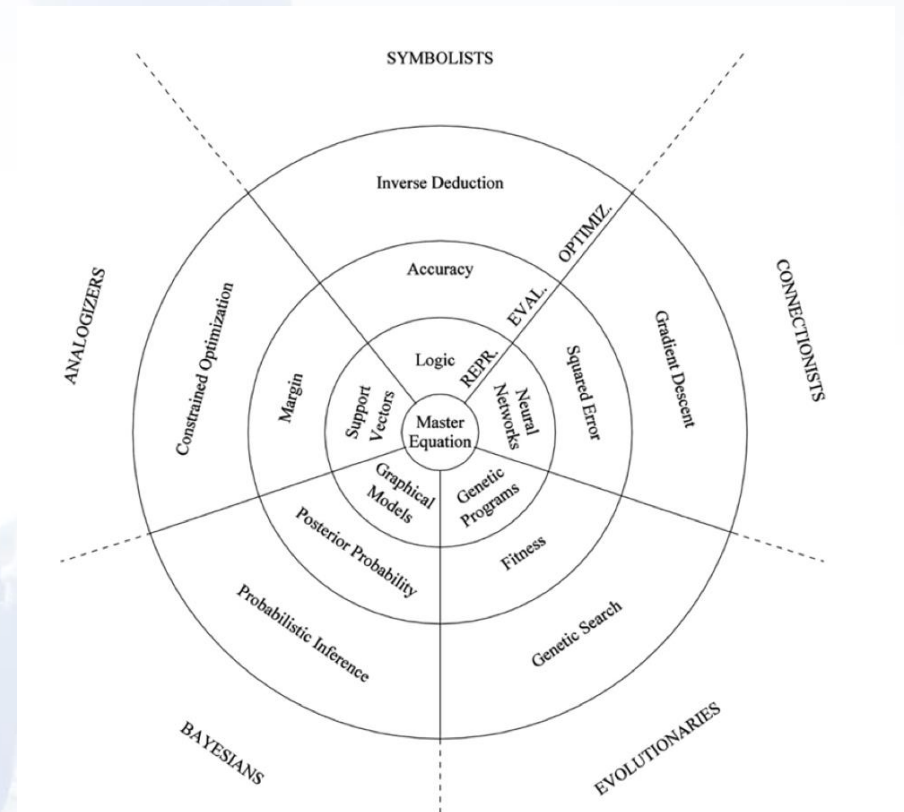
### Zusammenfassung

Kategorie	Erläuterung	Beispiele
A. Eager vs. Lazy Learning	- Eager Learning: Algorithmen verarbeiten die Trainingsdaten sofort und bauen ein Modell, das für alle Daten gilt.	- Eager: Entscheidungsbäume, SVMs, NN, LM
	- Lazy Learning: Kein expliziter Trainingsschritt; Daten werden gespeichert und erst bei der Vorhersage verarbeitet. -Berechnungen finden ad hoc statt	- Lazy: K-Nearest Neighbor (KNN)
B. Batch vs. Online Learning	- Batch Learning: Modell wird mit allen Trainingsdaten auf einmal trainiert.	- Batch: Logistische Regression, Random Forest
	- Online Learning: Modell wird kontinuierlich mit einzelnen Trainingsbeispielen aktualisiert.	- Online: Perceptron, Stochastic Gradient Descent (SGD)
C. Parametrische vs. Nicht-Parametrische Modelle	- Parametrisch: Feste Struktur und begrenzte Anzahl an Parametern, unabhängig von der Trainingsdatengröße.	- Parametrisch: Lineare Regression, SVMs
	- Nicht-Parametrisch: Flexible Struktur; Anzahl der Parameter wächst mit den Daten.	- Nicht-parametrisch: Entscheidungsbäume, KNN

## 2.2- Kategorisierung nach Pedro Domingo

### Evolutionär (Genetische Algorithmen):

- Lernen durch Simulation von natürlichen Evolutionen wie Mutation und Selektion.
- Besonders nützlich bei Optimierungsproblemen, bei denen der Lösungsraum groß und komplex ist.
- Bei der **Optimierung von Robotern** wird ein genetischer Algorithmus verwendet, um verschiedene Modelle von Robotern zu testen. Der Algorithmus simuliert dabei "Mutation" und "Selektion" – Roboter mit besseren Ergebnissen werden für die nächste Generation „vererbt“, um die Lösung kontinuierlich zu verbessern.



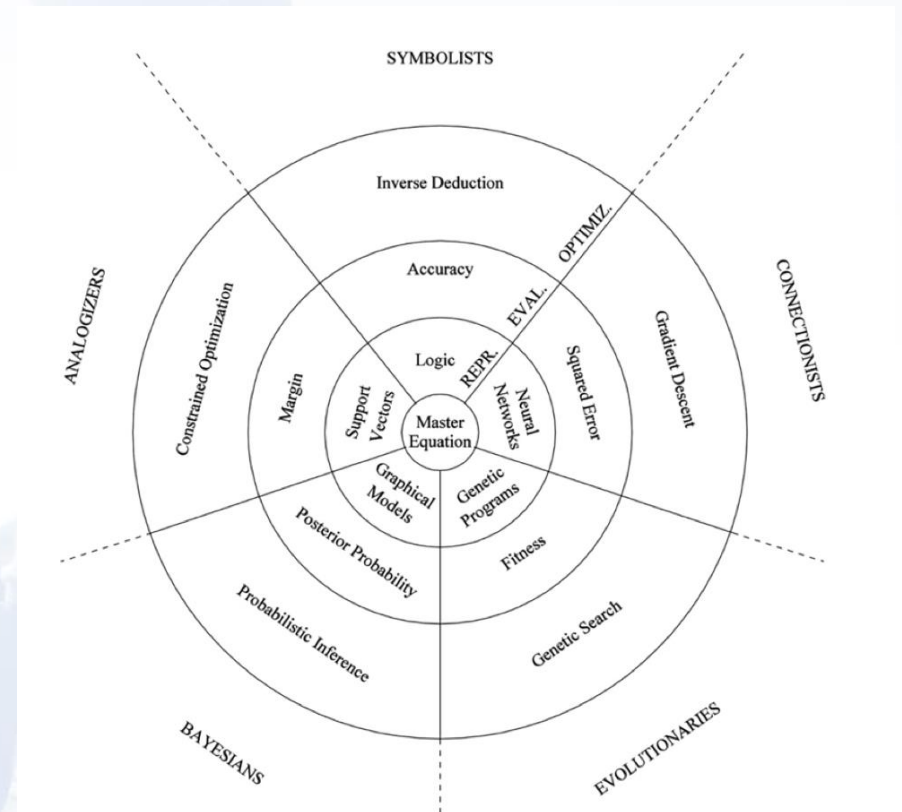
Pedro Domingos. The master algorithm: How the quest for the ultimate learning machine will remake our world. Basic Books, 2015

Dr Houssam Jedidi - Provadis

## 2.2- Kategorisierung nach Pedro Domingo

### Konnektionismus (Neuronale Netzwerke):

- Nachbildung des Gehirns zur Mustererkennung und Problemlösung.
- Besonders stark bei unstrukturierten Daten wie Bildern, Text und Audio.
- Ein **CNN** für **Bilderkennung** lernt von einer Vielzahl von Trainingsbildern, um zwischen verschiedenen Klassen von Bildern (z. B. Katzen und Hunde) zu unterscheiden.
- Diese Modelle sind besonders stark, wenn es um die Erkennung von Mustern in unstrukturierten Daten wie Bildern oder Audio geht.



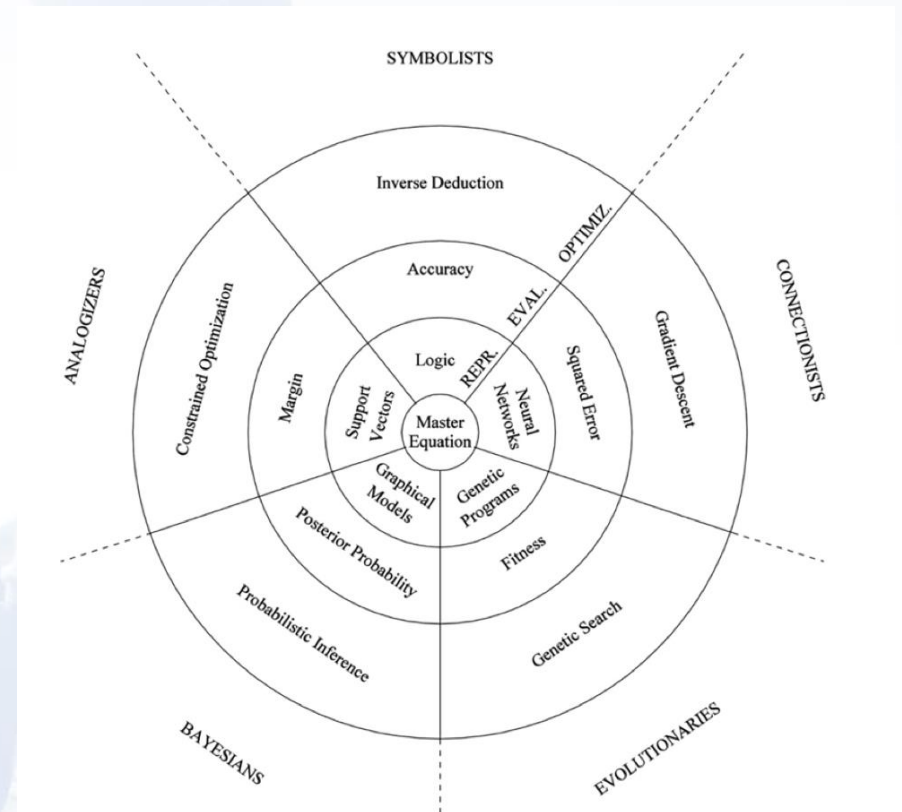
Pedro Domingos. The master algorithm: How the quest for the ultimate learning machine will remake our world. Basic Books, 2015

Dr Houssam Jedidi - Provadis

## 2.2- Kategorisierung nach Pedro Domingo

### Symbolismus (Logic):

- Verwendung von Symbolen und logischen Regeln zur Repräsentation und Verarbeitung von Wissen.
- Ideal für strukturierte und explizite Wissensdomänen.
- Ein **Expertensystem** im medizinischen Bereich basiert auf einer Sammlung von Regeln, die durch Symptome, Diagnosen und Behandlungsmöglichkeiten verknüpft sind.
- Hier werden klare, erklärbare Entscheidungen getroffen, die direkt auf den Regeln basieren.



Pedro Domingos. The master algorithm: How the quest for the ultimate learning machine will remake our world. Basic Books, 2015

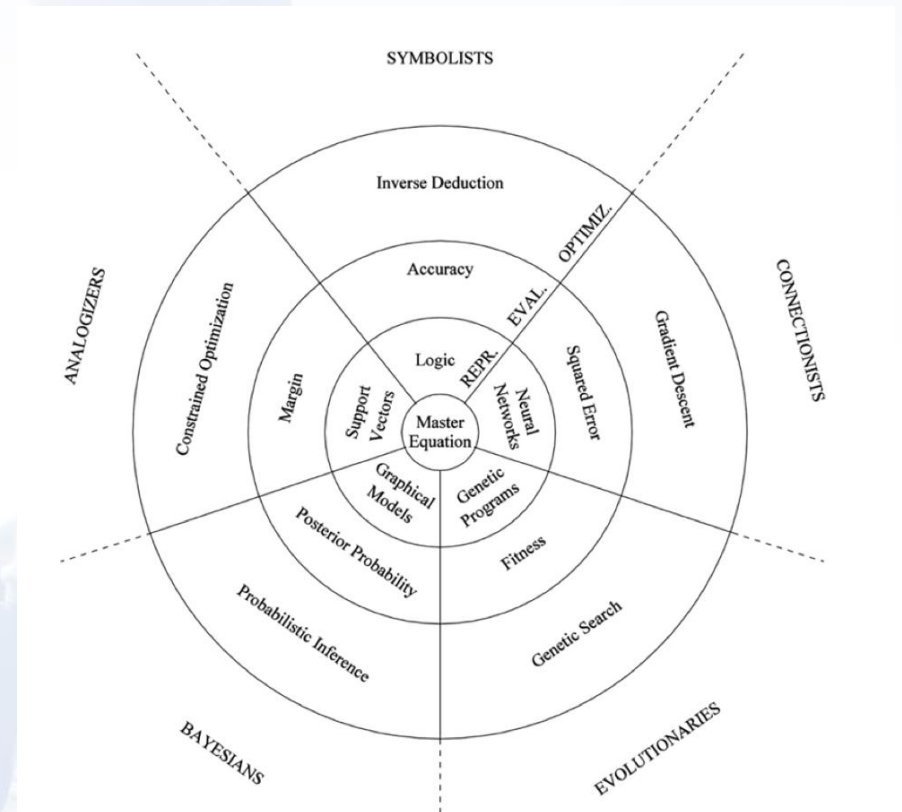
Dr Houssam Jedidi - Provadis



## 2.2- Kategorisierung nach Pedro Domingo

### Stochastische & Bayessche (Graphische Modelle):

- Handhabung von Unsicherheit und Wahrscheinlichkeiten, um zu fundierten Entscheidungen zu kommen.
- Sehr gut bei Problemen mit Unsicherheiten oder bei der Modellierung komplexer probabilistischer Abhängigkeiten.
- In der **Wettervorhersage** kann ein **Bayessches Netzwerk** verwendet werden, um probabilistische Beziehungen zwischen verschiedenen Wetterparametern zu modellieren. Dies ermöglicht eine flexible Vorhersage unter Unsicherheit, indem es kontinuierlich neue Daten integriert.



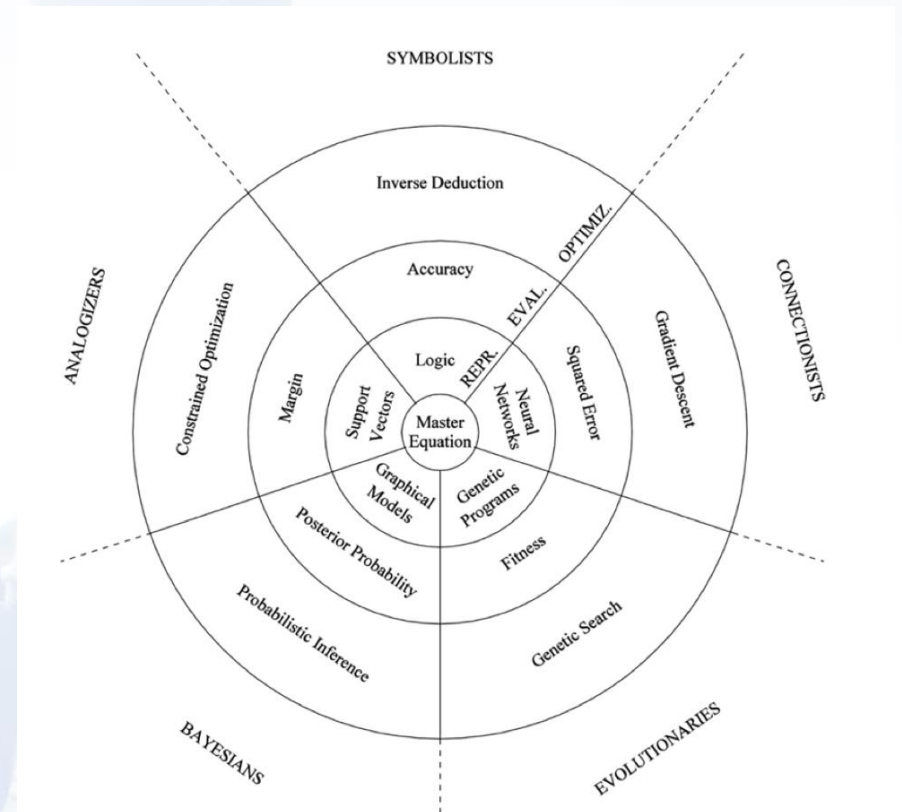
Pedro Domingos. The master algorithm: How the quest for the ultimate learning machine will remake our world. Basic Books, 2015

Dr Houssam Jedidi - Provadis

## 2.2- Kategorisierung nach Pedro Domingo

### Analogistische Modelle (SVM):

- Ähnlichkeitssuche zwischen Instanzen und Ermittlung der besten Trennlinie, um verschiedene Klassen zu unterscheiden.
- Sehr nützlich bei der Klassifikation und Regression, insbesondere bei hochdimensionalen Daten.
- Bei der **Spam-E-Mail-Erkennung** verwendet SVM die **Ähnlichkeit** zwischen E-Mails, um sie korrekt in die Klassen "Spam" oder "Nicht-Spam" zu unterteilen. Das Modell sucht nach der besten Trennlinie (Hyperplane) zwischen den beiden Kategorien, was zu einer hohen Genauigkeit führt.



Pedro Domingos. The master algorithm: How the quest for the ultimate learning machine will remake our world. Basic Books, 2015

Dr Houssam Jedidi - Provadis

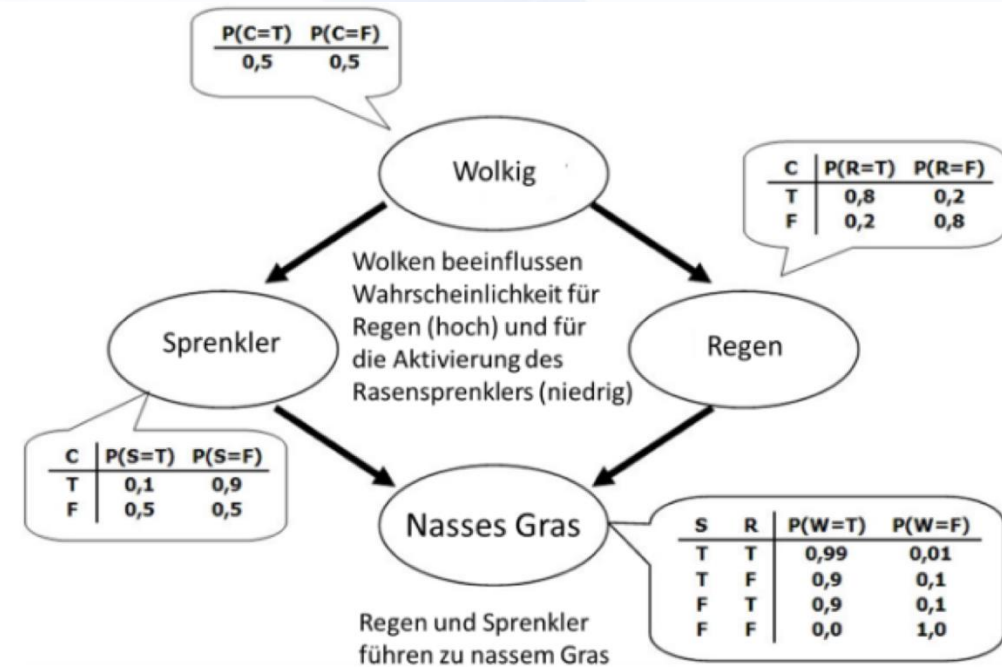


## 2.2.1- Stochastische & Bayes'sche Verfahren

Mit Hilfe des Bayes Theorem das jeweils wahrscheinlichste Modell auf Basis der vorliegenden Datenlage zur Beschreibung und Vorhersage zu generieren  
Kontinuierliche Aktualisierung so bald neu Informationen verfügbar sind.

Bayessches Netz ist ein Mechanismus zur automatischen Anwendung des Satz vom Bayes um die Wahrscheinlichkeit von abhängigen Variablen aus damit zusammenhängenden Beobachtungen zu berechnen.

Vorteilhaft bei geringeren Datenmengen aber  
Reiches Vorwissen

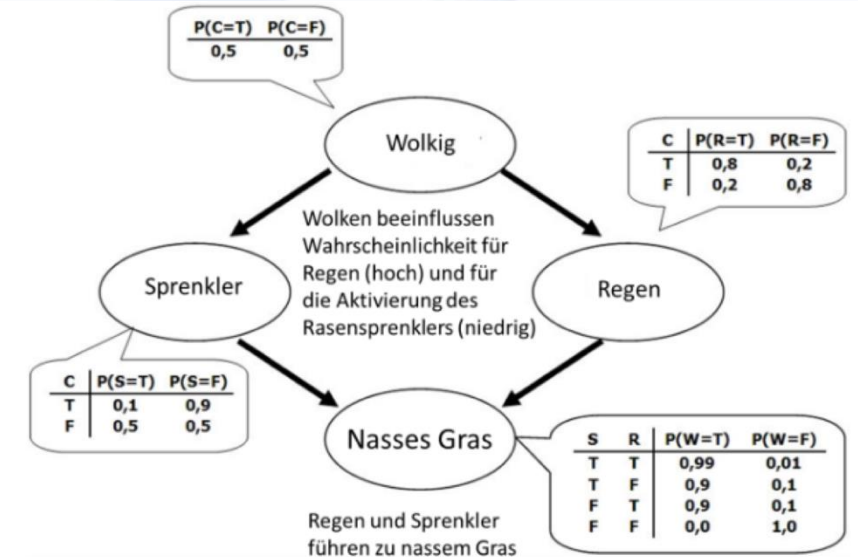


Bayessches Netz

## 2.2.1- Stochastische & Bayes'sche Verfahren

### Knoten und Abhängigkeiten:

- **Wolkig (C):** Dieses Ereignis beeinflusst die Wahrscheinlichkeit von:
  - **Regen (R):** Wolken erhöhen die Wahrscheinlichkeit für Regen.
  - **Sprenkler (S):** Wolken senken die Wahrscheinlichkeit, dass der Sprenkler aktiviert wird.
- **Regen (R) und Sprenkler (S):** Diese beiden Ereignisse beeinflussen gemeinsam:
  - **Nasses Gras (W):** Das Gras wird nass, wenn es regnet oder der Sprenkler läuft.

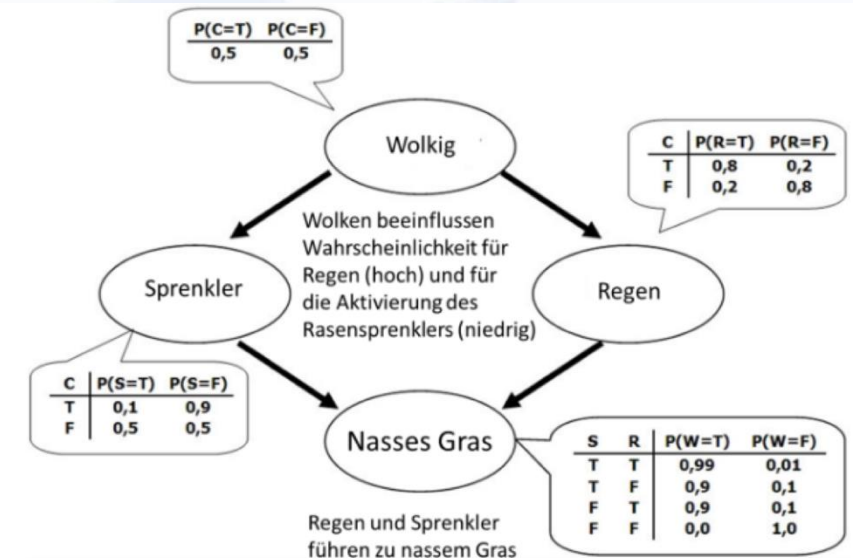


Bayessches Netz

## 2.2.1- Stochastische & Bayes'sche Verfahren

### Wahrscheinlichkeitstabellen:

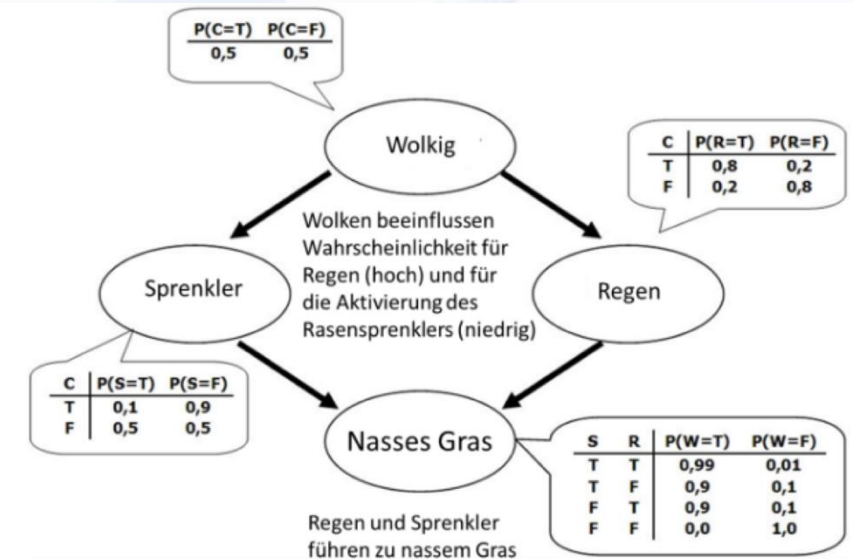
- Die Tabellen an den Knoten geben an, mit welcher Wahrscheinlichkeit ein Ereignis unter bestimmten Bedingungen eintritt.
  - $P(C=T/F)$ :** Wahrscheinlichkeit, dass es wolig ist (50% für wahr und falsch).
  - $P(R=T|C)$ :** Wahrscheinlichkeit für Regen, abhängig davon, ob es wolig ist.
  - $P(S=T|C)$ :** Wahrscheinlichkeit, dass der Sprenkler aktiviert wird, abhängig davon, ob es wolig ist.
  - $P(W=T|R,S)$ :** Wahrscheinlichkeit, dass das Gras nass wird, abhängig davon, ob es regnet oder der Sprenkler läuft.



## 2.2.1- Stochastische & Bayes'sche Verfahren

### Wahrscheinlichkeitstabellen:

- Die Tabellen an den Knoten geben an, mit welcher Wahrscheinlichkeit ein Ereignis unter bestimmten Bedingungen eintritt.
  - $P(C=T/F)$** : Wahrscheinlichkeit, dass es wolig ist (50% für wahr und falsch).
  - $P(R=T|C)$** : Wahrscheinlichkeit für Regen, abhängig davon, ob es wolig ist.
  - $P(S=T|C)$** : Wahrscheinlichkeit, dass der Sprenkler aktiviert wird, abhängig davon, ob es wolig ist.
  - $P(W=T|R,S)$** : Wahrscheinlichkeit, dass das Gras nass wird, abhängig davon, ob es regnet oder der Sprenkler läuft.



Bayessches Netz

### Kausalität:

- Die Pfeile zeigen die kausalen Beziehungen zwischen den Variablen. Zum Beispiel beeinflusst "Wolig" direkt "Regen" und "Sprenkler", während "Regen" und "Sprenkler" gemeinsam "Nasses Gras" beeinflussen.



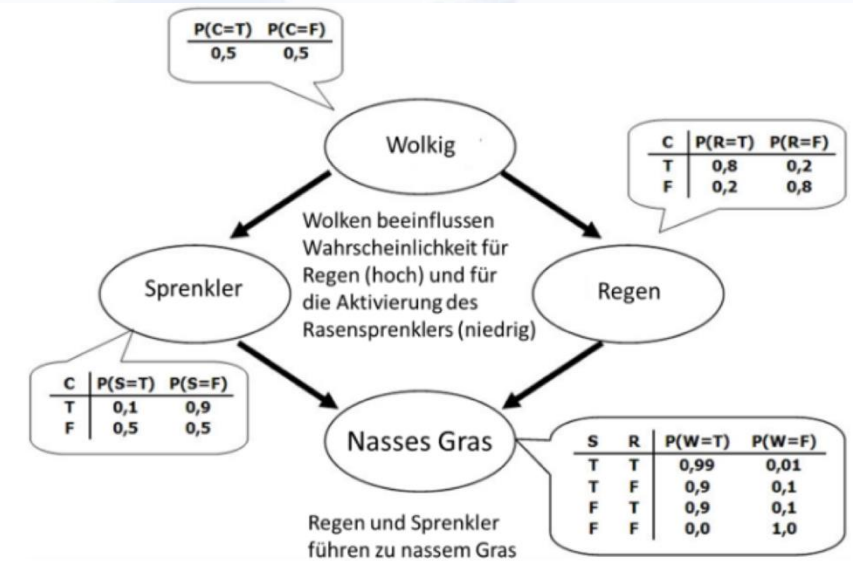
## 2.2.1- Stochastische & Bayes'sche Verfahren

Mit den Wahrscheinlichkeiten (a priori und konditional) kann man Rückschlüsse ziehen, z. B.:

- **Vorwärts:** Wie hoch ist die Wahrscheinlichkeit, dass das Gras nass ist, wenn es regnet?
- **Rückwärts:** Wie wahrscheinlich ist es, dass es regnet, wenn das Gras nass ist?

Angenommen, das Gras ist nass. Wir möchten wissen, ob es am Regen liegt oder am Sprenkler:

- Mit Hilfe der **bedingten Wahrscheinlichkeiten** und des Bayes'schen Satzes können wir berechnen, wie wahrscheinlich jede Ursache ist



Bayessches Netz

$$P(R = T|W = T) = \frac{P(W = T|R = T) \cdot P(R = T)}{P(W = T)}$$

## 2.2.1- Stochastische & Bayes'sche Verfahren

### Schritt 1: Berechnung von $P(W=T|R=T)$

$$P(W=T|R=T) = P(W=T|R=T, S=T) \cdot P(S=T) + P(W=T|R=T, S=F) \cdot P(S=F)$$

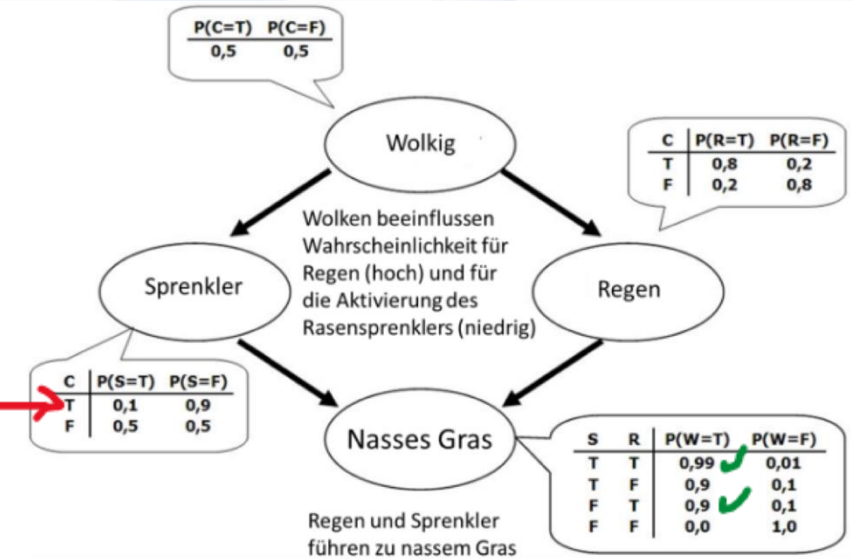
Die Wahrscheinlichkeit für SSS hängt von den Wolken (C) ab:

$$P(S=T) = P(S=T|C=T) \cdot P(C=T) + P(S=T|C=F) \cdot P(C=F)$$

$$\begin{aligned} P(S=T|C=T) &= 0.1 \\ P(S=T|C=F) &= 0.5 \\ P(C=T) &= 0.5 \end{aligned}$$

$$\rightarrow P(S=T) = 0.1 \cdot 0.5 + 0.5 \cdot 0.5 = 0.05 + 0.25 = 0.3$$

$$\rightarrow P(S=F) = 1 - P(S=T) = 1 - 0.3 = 0.7$$



Bayessches Netz



## 2.2.1- Stochastische & Bayes'sche Verfahren

$$P(W=T|R=T) = P(W=T|R=T, S=T) \cdot P(S=T) + P(W=T|R=T, S=F) \cdot P(S=F)$$

Werte aus der Tabell:

$$P(W=T|R=T, S=T) = 0.99$$

$$P(W=T|R=T, S=F) = 0.9$$

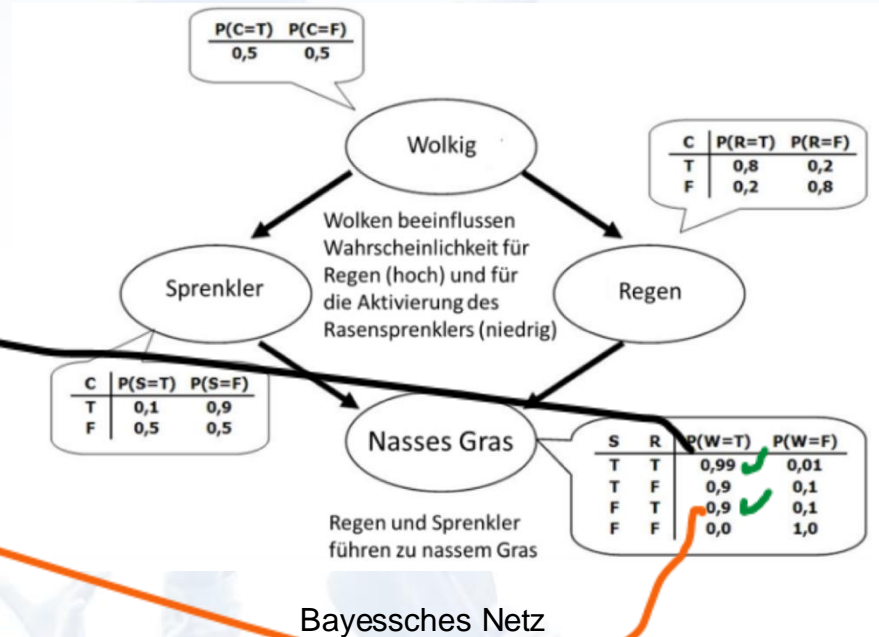
Einsetzen der Zahlen:

$$P(W=T|R=T) = 0.99 \cdot 0.3 + 0.9 \cdot 0.7$$

Berechnung:

$$P(W=T|R=T) = 0.297 + 0.63 = 0.927$$

Die Wahrscheinlichkeit, dass es geregnet hat, gegeben, dass das Gras nass ist, beträgt ungefähr 0.927

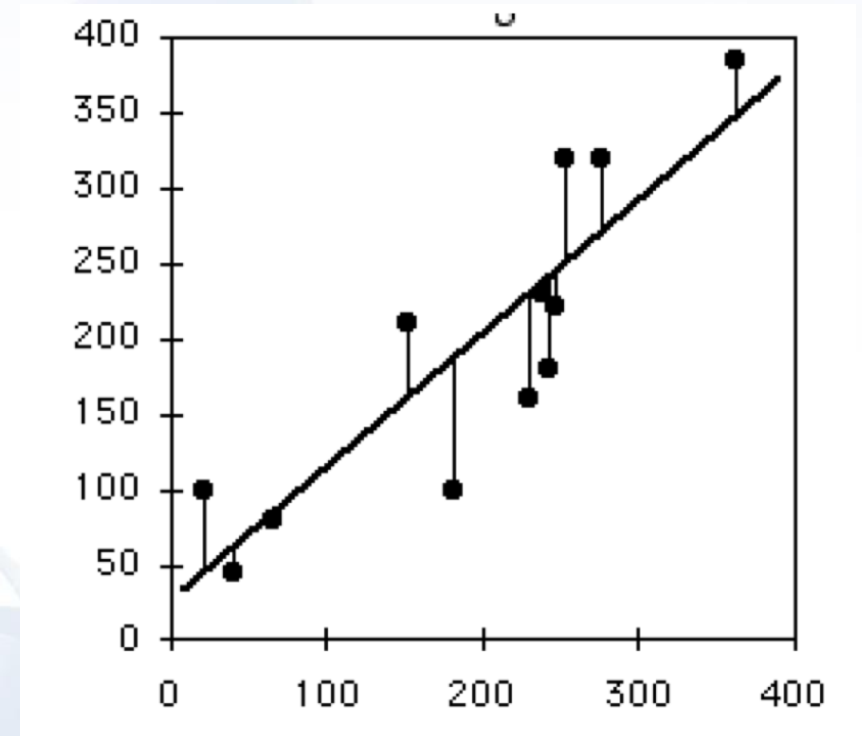


## 2.2.2- Der Analogismus

Annahme: Objekte, die bezüglich bestimmter Merkmale große Ähnlichkeiten aufzeigen → gehören zu einer gemeinsamen Klasse.  
Bauen von Ähnlichkeiten in einem mehrdimensionalen Zahlenraum und Distanzbasiert (Gruppen/ Cluster).

Bsp. Regression:

Bestimmung des Preises eines Hauses auf Basis von Merkmalen (Fläche, Baujahr...) → Nach analogischem Denken führen ähnliche Merkmalswerte zu einem vergleichbaren Preis



## 2.2.3- Der Konnektionismus

Maschinelles Lernen in der Funktionsweise des Gehirns, wo Lernen durch Veränderung in der Stärke der Verbindungen zw. einzelnen Nervenzellen oder Neuronen stattfindet.

In einer Datenstruktur werden „Knoten“ / „Künstliche Neuronen“ Sichtweise zu Künstlichen neuronalen Netzen (KNN) verbunden.  
Dort werden Daten bzw. Signale weiter geleitet.

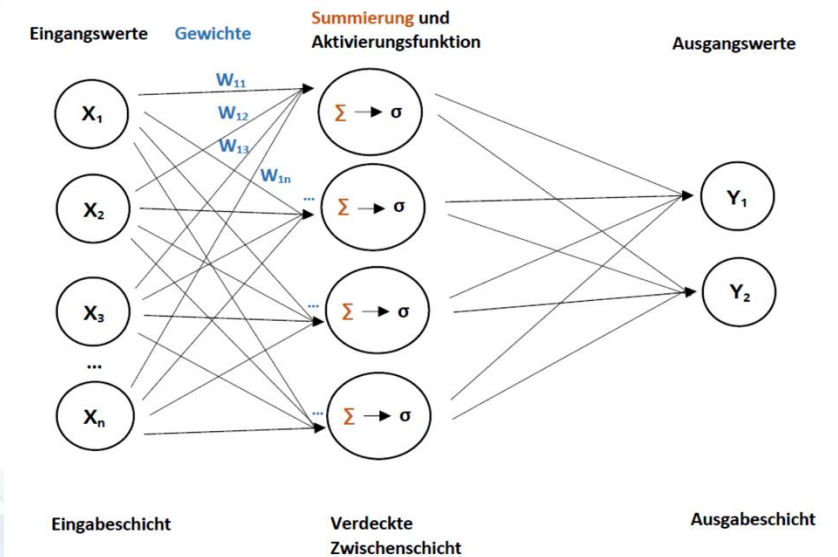
An den Verbindungen liegen mathematische Gewichte, die mit den eintreffenden Signalstärken multipliziert und anschließend aufaddiert werden.

In den Knoten bestimmt eine „Aktivierungsfunktion“, ob und in welcher Stärke ein Signal weitergegeben werden soll.

Aus dem Unterschied zw. Ausgabewerten und richtigen Antworten rückwärts durch die Schichten Korrekturen für die Gewichte werden ermittelt

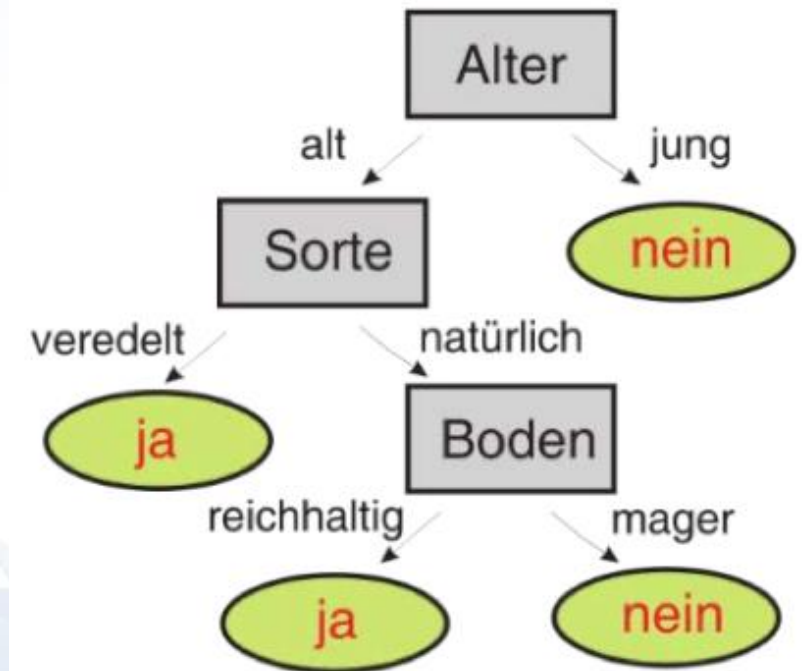
Für Menschen kaum nachvollziehbar

geeignet für Massendaten / Multidimensionale Daten



## 2.2.4- Der Symbolismus

Lokalbeschreibungen von Wissen über Konzepte  
Aus Beobachtungen wird Wissen generiert  
Meist benutztes Verfahren: Entscheidungsbäume  
Lernen in einem Entscheidungsbaum: Knoten von oben nach unten  
möglichst diskriminative Abfragen von Eigenschaften zuweisen  
Lern Effizienz hängt stark von der Präzision der Fragen

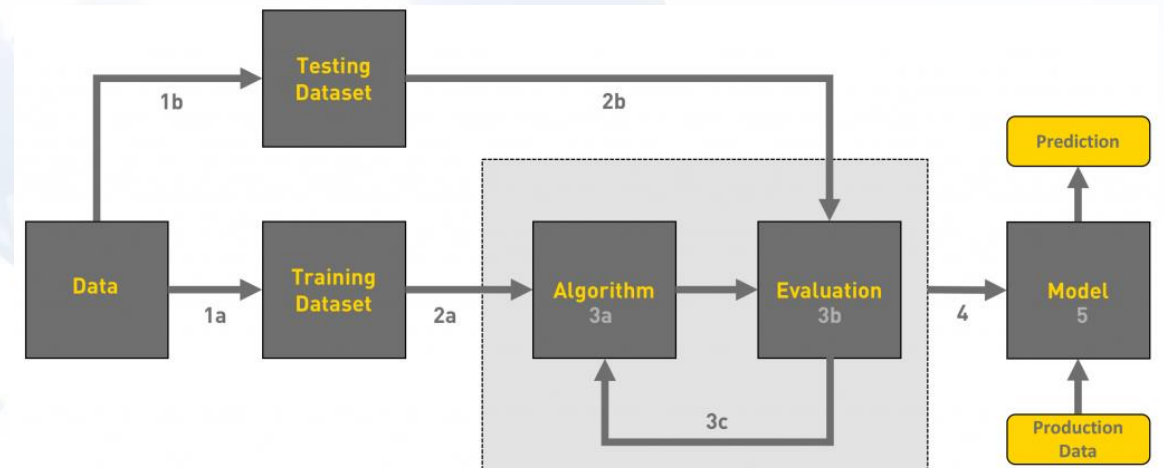


Entscheidungsbaum zur Vorhersage der Apfelernte



# 4- Vorgehensweise des maschinellen Lernen

1. Datenimport
2. Datenaufbereitung
3. Datenteilung/split
4. Modeltraining
5. Modelauswertung (score)- Auswahl



## 4- Vorgehensweise des maschinellen Lernen

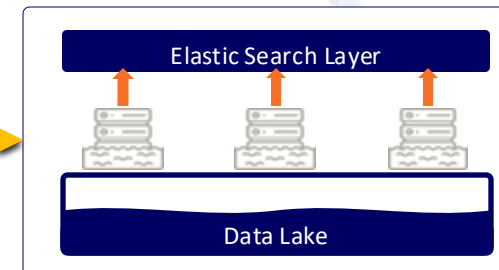
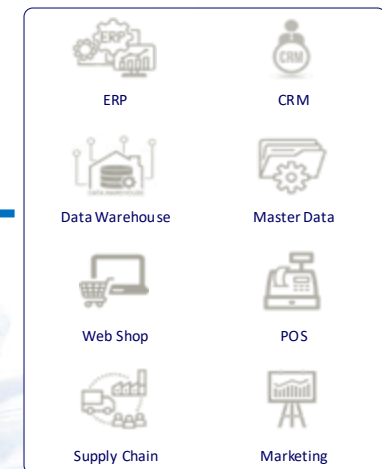
### 4.1 Datenimport

Verschiedene Datenquellen  
Unterschiedliche Datenformate (Un-/Semi-/ Strukturiert)  
Inhalt (Marketing, Logistik, Verkauf, PR, ...)

#### External



#### Internal





## 4- Vorgehensweise des maschinellen Lernen

### 4.2 Datenaufbereitung (Preprocess)

In diesem Schritt werden Daten normalisiert, partitioniert, gefiltert oder transformiert  
Es können neu berechnete Spalten hinzugefügt werden, eine Datenkonvertierung durchgeführt werden usw.  
Wenn die Daten im Quellsystem vorbereitet wurden, ist dieser Schritt nicht zwingend notwendig. Wenn aber z.B. rohe Daten eingelesen werden, können wir in diesem Schritt die notwendigen Aktionen durchführen.

## 4- Vorgehensweise des maschinellen Lernen

### 4.3 Datenaufteilung

Jeder Machine-Learning Algorithmus funktioniert so, dass die Daten in Training und Score (Auswertungs-) Daten aufgeteilt werden müssen.

Splittingstrategie:

80 – 20 Regel: Training – Testing

70 – 20 – 10 Regel: Training – Testing – Validation

Zufällige Teilung der Daten (Shuffle/ Randomness) für bessere Effizienz

Bspw. Produktverkäufe der letzten 5 Jahre. Wenn man den Zufallseffekt ignoriert, wird der Algorithmus auf völlig anderen Daten getestet (die Testdaten stammen nur aus einem Teil des letzten Jahres, in dem neue Parameter wie Krieg und Inflation vorhanden sind)

## 4- Vorgehensweise des maschinellen Lernens

### 4.4 Algorithmen

#### A- Klassifikation:

Dieser Typ des Machine Learning Algorithmus wird dazu verwendet, unbekannte Daten zu klassifizieren, oder zu kategorisieren. Dieser Typ wird oft bei der Bildererkennung oder OCR (Texterkennung) verwendet. Beispielsweise unterteilt das System in Bilder mit Katzen, Hunden oder Personen. Als Ergebnis wird immer ein Ja, oder Nein geliefert. Ist es eine Katze, ist es ein Hund usw.

#### B- Regression:

Regression-Algorithmen liefern immer eine Zahl. Zum Beispiel die Temperatur in den nächsten Tagen, Aktienwerte oder die Anzahl der verkauften Produkte. Es gibt viele Regression Algorithmen:

- Lineare Regression
- Bayesian Regression
- Decision Tree / Forest
- Neural Network Regression
- Poisson Regression

## 4- Vorgehensweise des maschinellen Lernens

### 4.5 Modell Training

Training-Modelle verwenden als Input-Parameter Training Daten und einen ausgewählten Algorithmus, um das Model zu trainieren

Es werden Attribute definiert und das System probiert anschließend, die Zusammenhänge zwischen den Attributen zu erkennen und Ergebnisse zu ermitteln

Beispiel: Mit den Input-Attribute Geschlecht (Mann, Frau), Kaufverhalten (Vielkäufer, Sparfuchs), Ort und Wetterdaten kann das Ergebnis die erwartete Anzahl verkaufter Produkte sein





## 4- Vorgehensweise des maschinellen Lernens

### 4.5 Modell Training

Nach dem das Modell trainiert wurde, werden Ergebnisse mit den bereits existierenden Daten verglichen, um die Genauigkeit der Vorhersage des Modells zu bewerten.

Bsp.: WettererscheinungID, Temperature und ProductCount sind bekannte Parameter (bei bestimmter Temperatur und Wettererscheinung wurde eine bestimmte Anzahl Produkte (ProductCount) verkauft). Scored Labels ist die Zahl, die vom Machine Learning Algorithmus vorhergesagt wurde.

Wenn wir die ProductCount und Scored Labels vergleichen, sehen wir, dass es Unterschiede gibt zwischen dem, was tatsächlich verkauft wurde und dem, was vorhergesagt wurde. Es gibt aber auch gute Treffer

WettererscheinungID	Temperature	ProductCount	Scored Labels
			
8	21.648148	21	8.184381
2	-1.689	64	36.907272
8	12.921384	10	10.739407
2	10.923076	3	8.486647
8	9.915833	3	4.285086
3	-1.810416	24	19.531075
2	10.487083	11	4.228116
10	18.869	9	12.221935
2	10.5	5	1.125671



# Wichtigste Algorithmen des Maschinellen Lernens

## 1- Resgressionsanalyse

Lernaufgabe: Regression

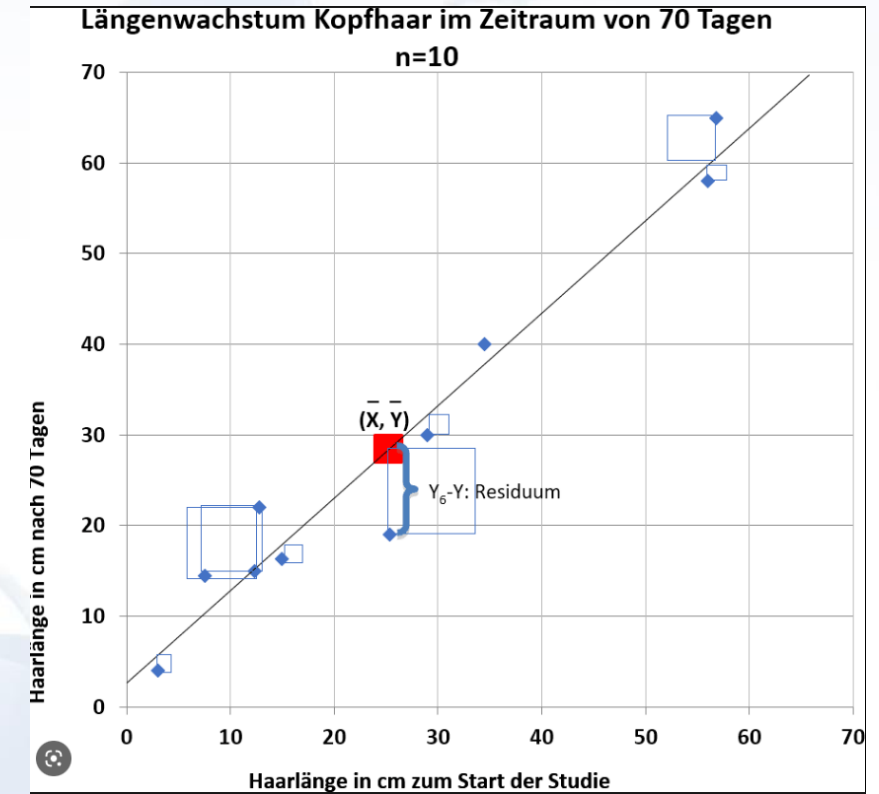
Lernstil: Überwacht, Analogistischer Ansatz

Alle Merkmale numerisch / Kategorisch

Lineare Funktion

Parameter: Abstand vom Nullpunkt & Steigung

Ziel: Minimierung der Abweichung zur Regressionslinie





# Wichtigste Algorithmen des Maschinellen Lernens

## 2- Logistische Regression

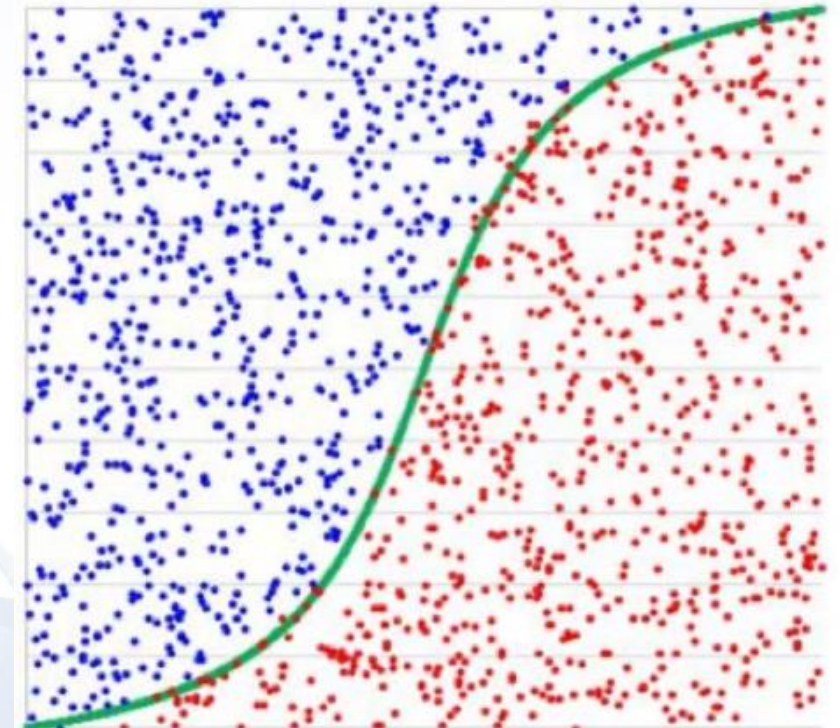
Lernaufgabe: Klassifikation

Lernstil: Überwacht, Analogistischer Ansatz

Alle Merkmale numerisch / Kategorisch

Trennungslinie, die zwei Klassen voneinander trennt (# Klassen könnte erhöht werden)

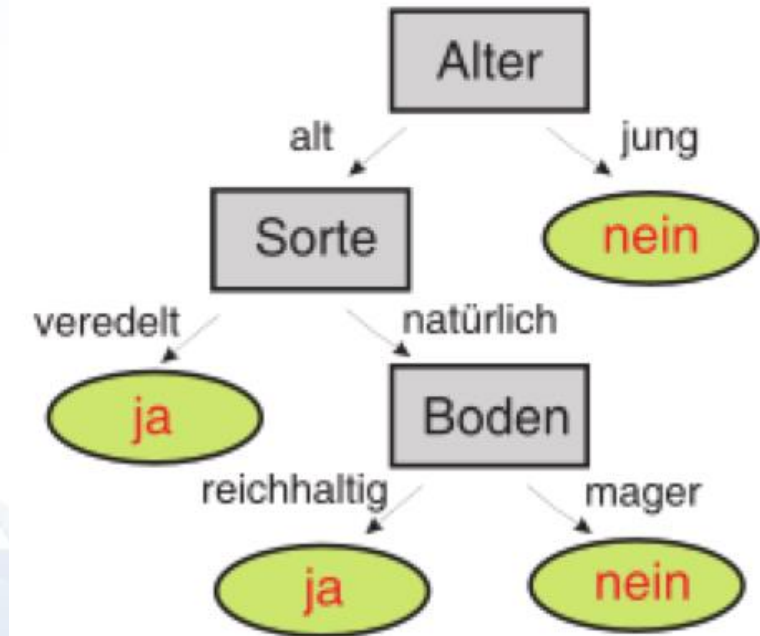
Die logistische Funktion ergibt eine Entscheidungsgerade



# Wichtigste Algorithmen des Maschinellen Lernens

## 3- Entscheidungsbäume

Lernaufgabe: Klassifikation  
Lernstil: Überwacht, Symbolischer & Analogistischer Ansatz  
Hat eine Baumstruktur  
An jedem Verzweigungsknoten wird ein Merkmalswert abgefragt.  
In den Endknoten steht eine Klasse



# Wichtigste Algorithmen des Maschinellen Lernens

## 4- Klassifikations- und Regressionsbäume (CART)

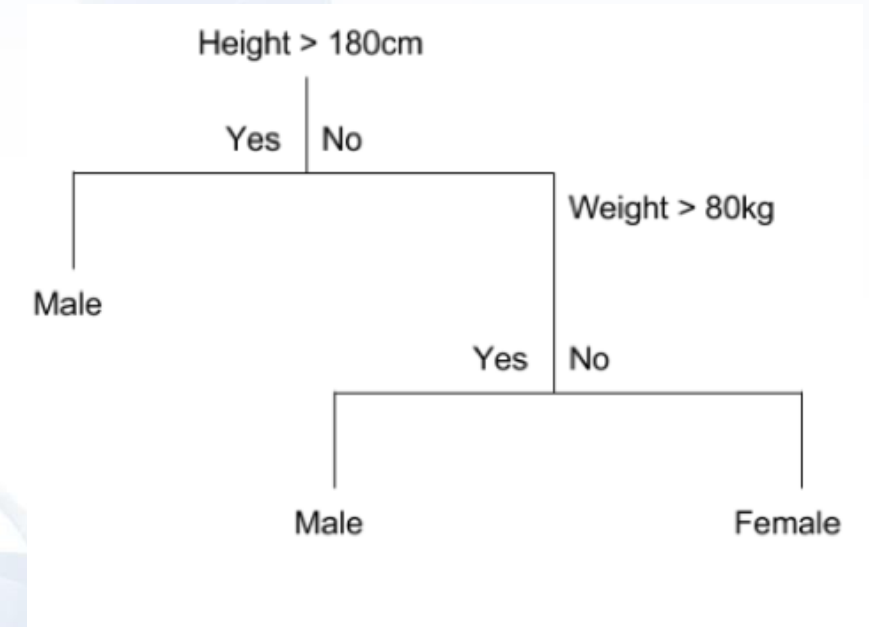
Lernaufgabe: Regression

Lernstil: Überwacht

An Jedem Verzweigungsknoten steht einen Numerischen Wert

Es liegt am Modellentwickler, den Baum bei einer geeigneten Tiefe zu kappen

Es gibt verschiedene Strategien, die Entscheidungskriterien an den Knoten zu wählen



# Wichtigste Algorithmen des Maschinellen Lernens

## 5- Random forest

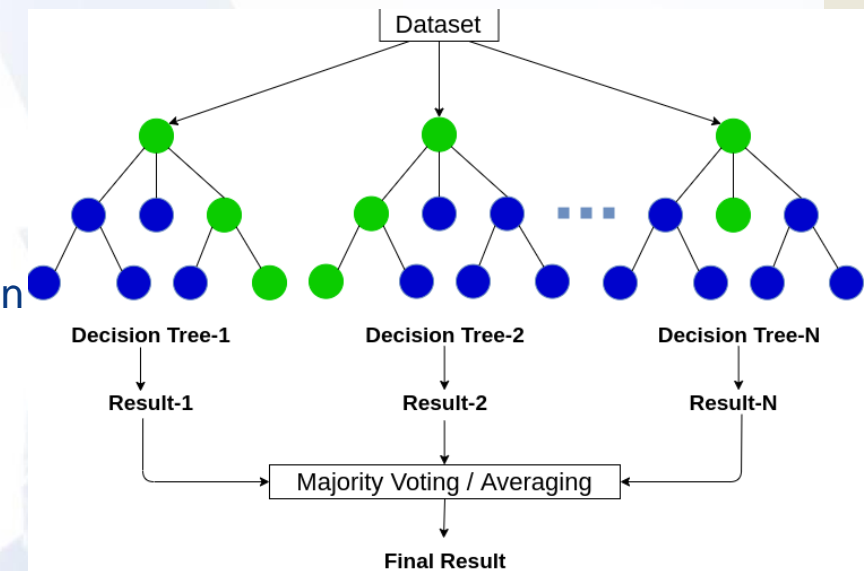
Lernaufgabe: Regression oder Klassifikation

Lernstil: Überwacht

Modell besteht aus einer Gruppe (Wald, Forest) von Entscheidungsbäumen, die parallel laufen

Jeder Baum kommt zu einer Ergebnis und die Klasse, die von den meisten Bäumen gewählt wurde, ergibt die Antwort

Die Bäume werden mit unterschiedlichen Teilmengen der Trainingdaten trainiert  
Gehört zu einer allg. Klasse der Ensemble-Methoden: Nutzung von mehreren Lernalgorithmen um bessere Ergebnisse zu erzielen

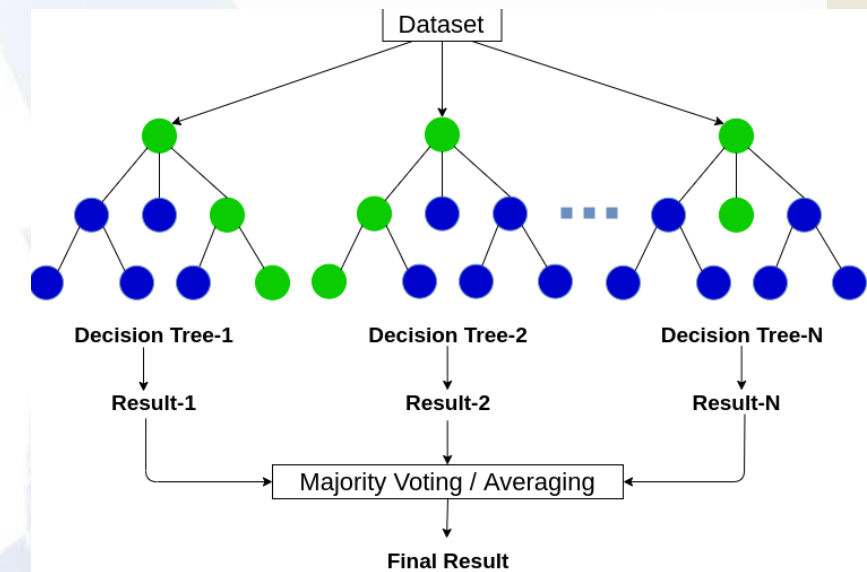




# Wichtigste Algorithmen des Maschinellen Lernens

## 5- Random forest (Fortsetzung)

Die Einzelnen Lernalgorithmen sind als „schwach“ bezeichnet  
Grund: Fehlerhafte Klassifizierung für die einzelnen Entscheidungsbäume tolerierbar sind und im Gesamtmodell ausgeglichen werden können  
Daher, „Boosting“ ist eine Lösung  
Beim „Boosting“ werden die vom Vorgänger fehlklassifizierten Beispiele stärker gewichtet, um dessen Unzulänglichkeiten anzugleichen





# Wichtigste Algorithmen des Maschinellen Lernens

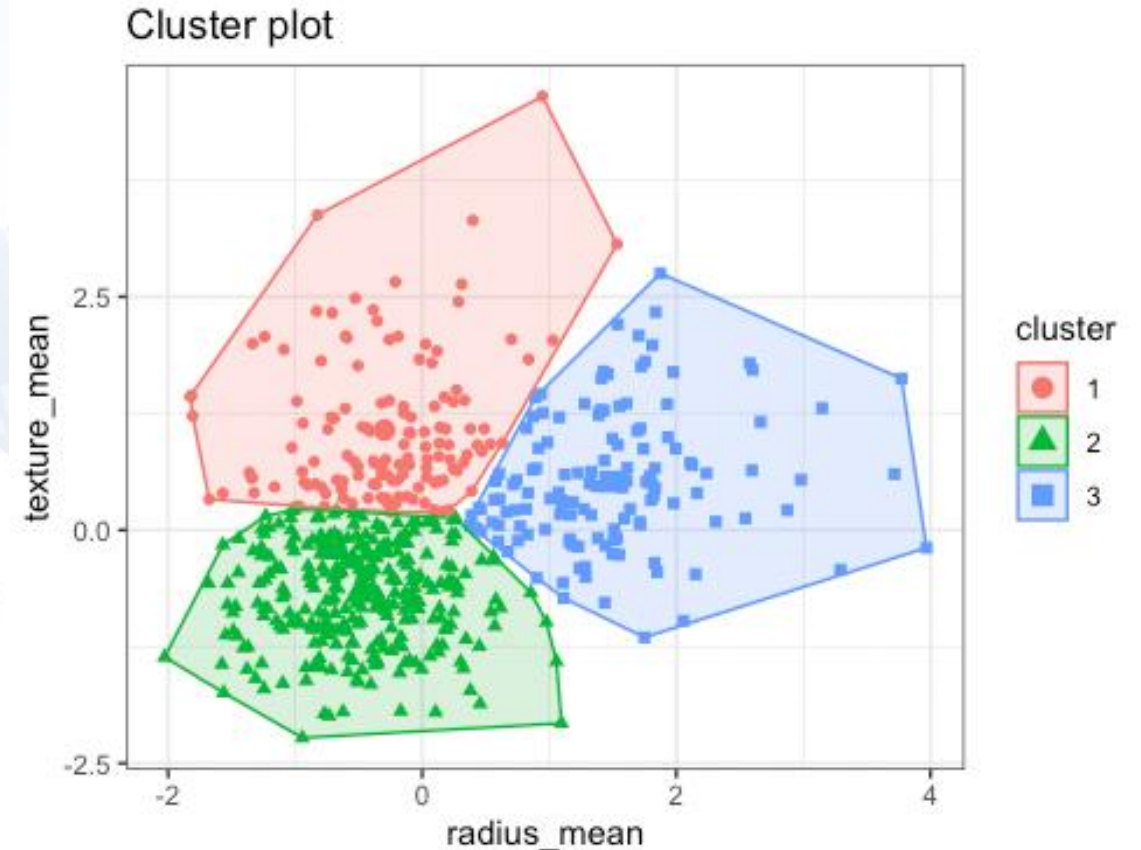
## 6- Clusteranalyse

Beispielmenge in Gruppen (Cluster) aufteilen.  
Elementen innerhalb einer Klasse weisen eine große Ähnlichkeit

Die Ähnlichkeit wird durch eine vorzugebende  
Distanzfunktion auf den Beispieldaten berechnet  
Bekannteste Modell:

K-means (Anzahl der Cluster vorgegeben)  
es wird mit beliebigen K Punkten als Clusterzentren  
gestartet und allen Beispielen ihrem ähnlichsten  
Clusterzentrum zugeordnet

Prozess wird oft wiederholt und alle Punkte erneuert  
zuordnen



# Wichtigste Algorithmen des Maschinellen Lernens

## 6- Clusteranalyse

Lernaufgabe: Clustering

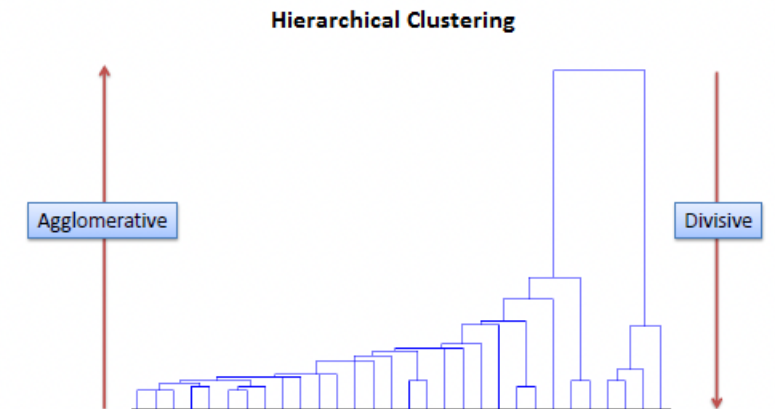
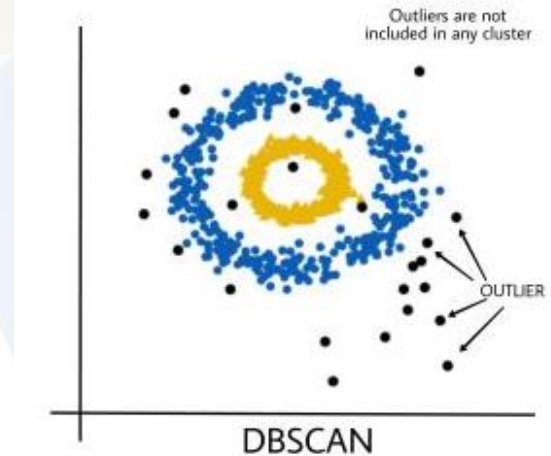
Lernstil: unüberwacht, analogistischer Ansatz

Es gibt Zahlreiche Verfahren, die unterschiedlich die Gruppen bestimmen:

Anzahl der Gruppen vorgegeben werden: K-means

Anzahl der Gruppe Willkürlich: Hierarchische Verfahren (Cluster nach vorgegebenen Kriteriums nach und nach berechnet)

Dr Houssam Jedidi - Provadis



# Wichtigste Algorithmen des Maschinellen Lernens

## 7- Stützvektormaschine (SVM)

Lernaufgabe: Klassifikation  
Lernstil: überwacht, analogistischer Ansatz

Lernt eine entscheidungsebene in dem Raum der Eingabedaten- die genau den maximalen Abstand zu den nächstliegenden Datenpunkten hat

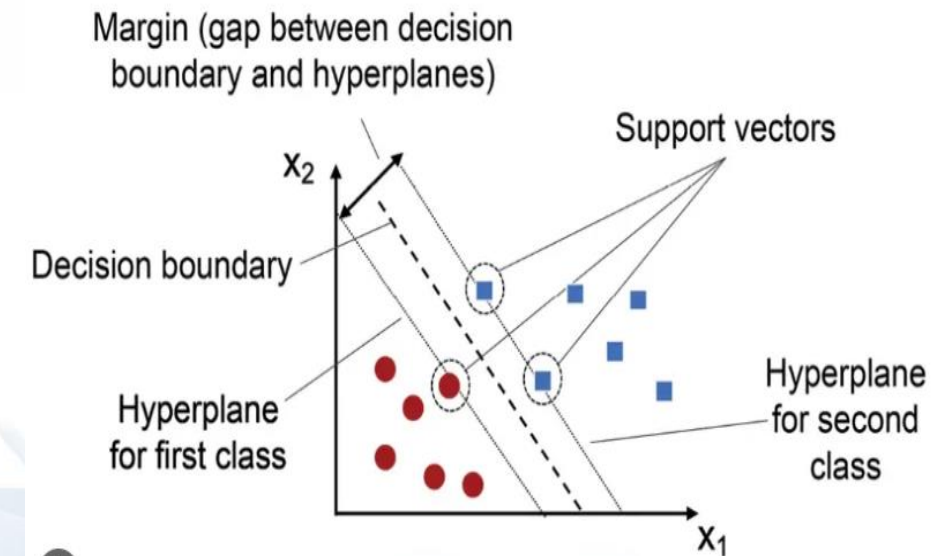
Für Mehrklassenprobleme oder hochkomplexe Strukturen können mehrere SVM-Modelle (z. B. OvO, OvR oder hierarchische SVMs) verwendet werden.

Per „Kernel-trick“- (Bsp. RBF) lassen sich auch nicht lineare Entscheidungsebenen recheneffizient lernen, in dem die Ebene in einem Implizierten höher-dimensionalen Raum bestimmt wird

Multi-Class SVM für mehr als 2 Klassen

SVR für Regression

Transductive SVM für Semi-überwachtes Lernen



# Wichtigste Algorithmen des Maschinellen Lernens

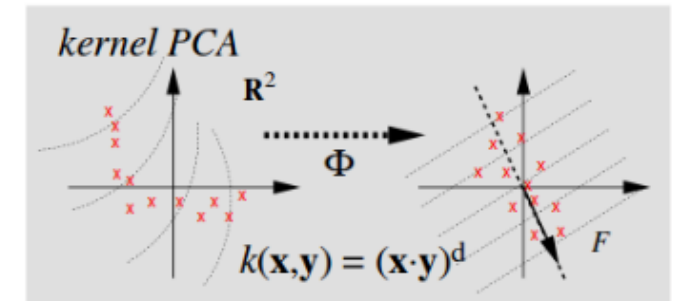
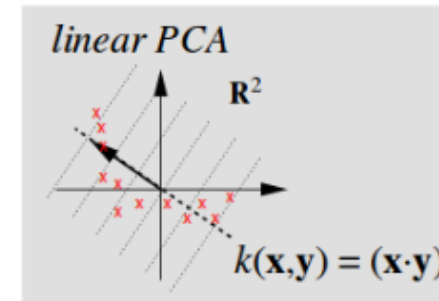
## 8- Kernel Principal Component Analysis PCA

Lernaufgabe: Dimensionsreduktion, Strukturerkennung  
Lernstil: unüberwacht, statistische Methode

Idee dahinter:

Variablen mit hoher Korrelation in wenigen (linear)  
unkorrelierten Variablen konvertieren

Per „Kernel Trick“ wird dieses Verfahren auch nichtlinear





# Wichtigste Algorithmen des Maschinellen Lernens

## 9- Künstliche Neuronale Netze

### Modellaufbau:

Funktionsweise aus menschlichem Gehirn abstrahiert

Künstliche Neuronen mit Aktivierungsfunktion

Input: Werte als Signale über mehrere Schichten ins Ausgabe umwandeln

### Lernprozess:

Die Stärke der Verbindungen zw. den Knoten benachbarter Schichten

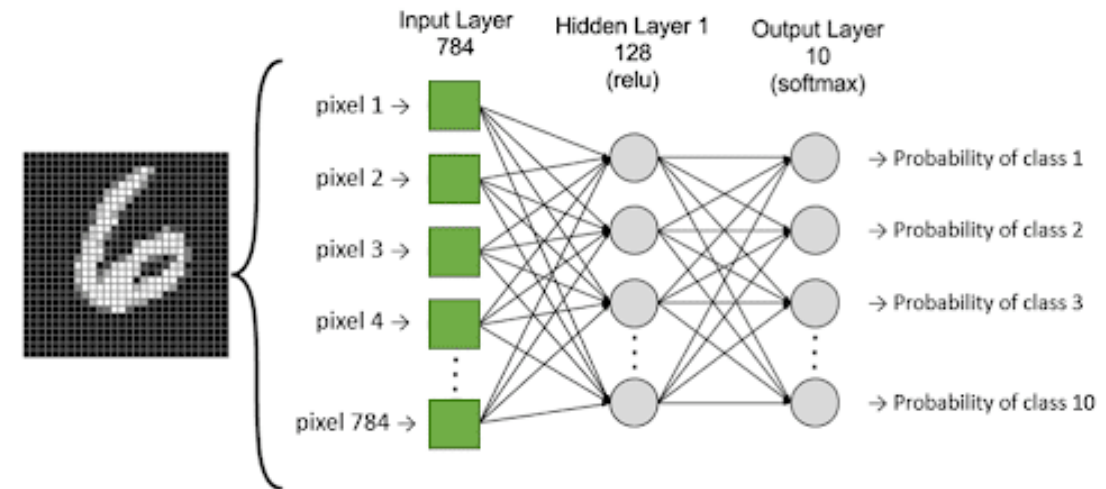
Fehler zw. berechneter und richtiger Ausgabe zurückgerechnet werden

Neue Eingaben werden durch das Netz propagiert in dem sie mit den Gewichten multipliziert und vor jedem Knoten aufsummiert and die jeweilige Aktivierungsfunktion übergeben werden

Die Aktivierungsfunktionen sind nicht linear → Gesamt gelernte Funktion komplizierte nicht lineare Funktionen approximieren kann

Lernaufgabe: Klassifikation oder Regression

Lernstil: überwacht, Konnektionistischer Ansatz





# Wichtigste Algorithmen des Maschinellen Lernens

## 10- Bayessche Modelle

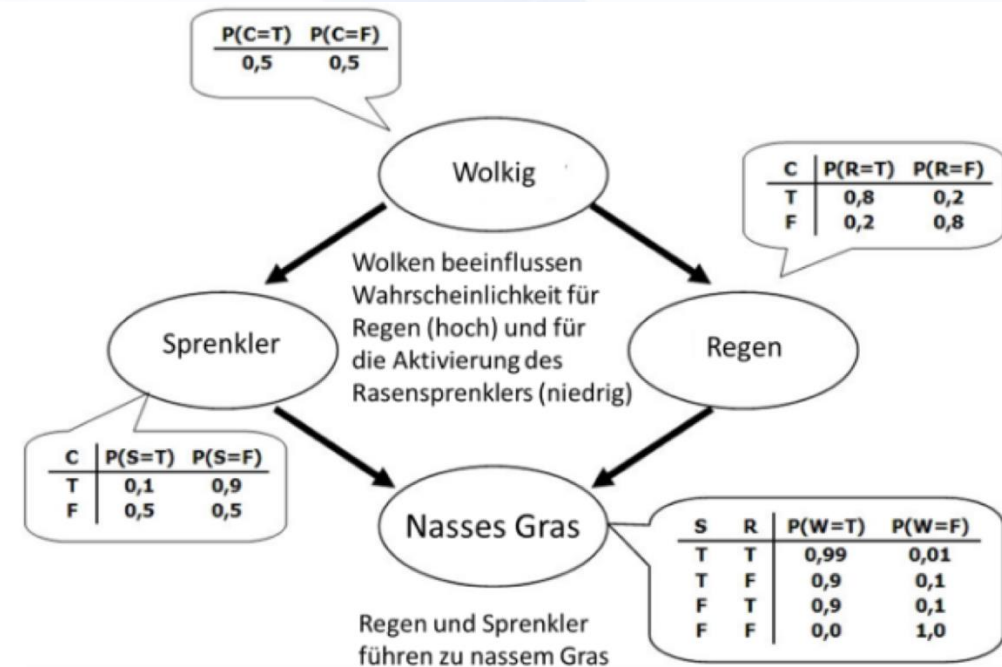
Bayessches Netz besteht aus Knoten für Zufallsvariablen und Pfeilen für die Bestimmung der Abhängigkeit

In jedem Knoten sind Wahrscheinlichkeitstabellen: Wahrscheinlichkeit der unbeobachteten bedingten Variablen aus Wahrscheinlichkeiten von beobachteten Größen bestimmt werden

Wahrscheinlichkeitstabellen hängen von Parametern, die aus den Daten gelernt werden

Lernaufgabe: Klassifikation

Lernstil: überwacht, Bayesscher Ansatz



# Wichtigste Algorithmen des Maschinellen Lernens

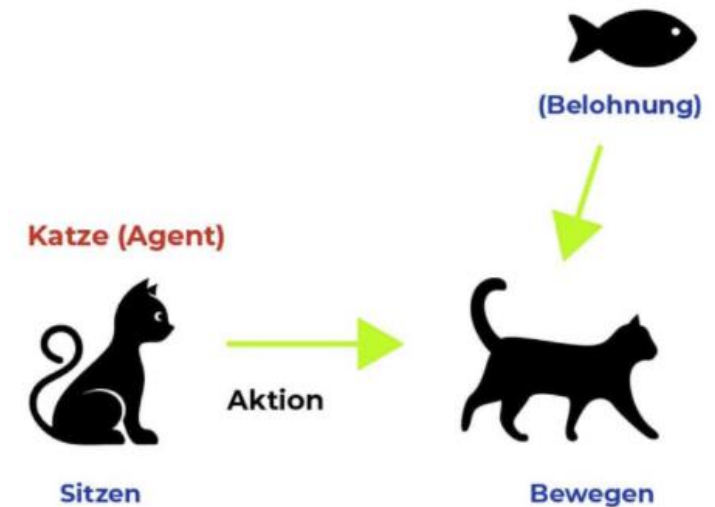
## 11- Sequentielle Entscheidungen

Besteht aus einer Entscheidungsfunktion, die einem Agenten in einem Zustand eine Aktion zuweist

Der Agent Wandert von einem Zustand zum nächsten und trifft sequentiell die jeweils aussichtsreichste Entscheidung

Q-lernen: jedem Paar von Zustand und Aktion wird einem Zahlenwert zugewiesen.

Im Einsatz wird, in jedem Zustand, die Aktion mit dem höchsten zugewiesenen Wert gewählt





# Wichtige Themen & Grundlagen

## Genauigkeit:

Definition: Die Genauigkeit ist der Anteil der korrekt vorhergesagten Instanzen im Verhältnis zur Gesamtanzahl der Instanzen.

Beispiel: Angenommen, Sie haben ein Modell zur Erkennung von Spam-E-Mails. Von 100 E-Mails wurden 80 korrekt als Spam erkannt und 10 wurden fälschlicherweise als Spam klassifiziert. Die Genauigkeit beträgt dann  $(80 + 0) / 100 = 80\%$ .

## Verlustfunktionen:

Definition: Verlustfunktionen messen den Fehler zwischen den vorhergesagten Werten des Modells und den tatsächlichen Werten.

Beispiel: Bei einem Regressionsproblem könnte die mittlere quadratische Abweichung (Mean Squared Error, MSE) als Verlustfunktion verwendet werden. Je niedriger der MSE, desto besser die Leistung des Modells.

Angenommen, das Modell sagt für eine Instanz einen Wert von 5 vorher, während der tatsächliche Wert 7 ist. Der quadratische Fehler wäre  $(7 - 5)^2 = 4$ . Wenn dies für mehrere Instanzen gemacht wird, wird der Durchschnitt als MSE berechnet.

### Überanpassung & Unteranpassung:

Definition: Überanpassung tritt auf, wenn ein Modell zu gut auf die Trainingsdaten passt, aber nicht gut auf neuen, nicht gesehenen Daten generalisiert. Unteranpassung tritt auf, wenn ein Modell zu einfach ist und nicht in der Lage ist, die Muster in den Daten zu erfassen.

Beispiel: Ein Modell, das alle Trainingsdaten perfekt passt, könnte überangepasst sein und schlecht auf neuen Daten abschneiden. Ein zu einfaches Modell könnte unterangepasst sein und sowohl auf den Trainingsdaten als auch auf neuen Daten schlecht abschneiden.



### Receiver Operating Characteristics (ROC):

Definition: Der ROC ist ein Diagramm, das die Leistung eines binären Klassifikators darstellt, während der AUC den Bereich unter der ROC-Kurve misst.

Beispiel: In einem medizinischen Diagnosemodell zeigt die ROC-Kurve, wie gut das Modell zwischen kranken und gesunden Patienten unterscheiden kann. Eine AUC von 1,0 bedeutet perfekte Vorhersagen, während 0,5 bedeutet, dass das Modell nicht besser ist als zufälliges Raten.

**Genauigkeit:** Hohe Genauigkeit ist gut, aber sie könnte täuschen, wenn die Klassen im Datensatz unausgeglichen sind.

**Verlustfunktionen:** Geringe Verluste sind wünschenswert und zeigen an, dass das Modell die Daten gut vorhersagt.

**Überanpassung/Unteranpassung:** Ein ausgewogenes Modell ist das Ziel, um sowohl auf den Trainingsdaten als auch auf neuen Daten gut abzuschneiden.

**ROC und AUC:** Höhere AUC-Werte zeigen eine bessere Leistung des Modells an.

# Praktische Anwendung

Bilderkennung – NLP –  
Empfehlungssysteme – Gesichterkennung  
– Autonome Fahrzeuge



### Bilderkennung:

Bilderkennung basiert auf neuronalen Netzwerken, die durch viele Bilder trainiert werden. Diese Netzwerke lernen, Muster und Merkmale in Bildern zu erkennen. In der Anwendung werden diese Modelle auf neue Bilder angewendet, um Gesichter zu identifizieren und Emotionen zu analysieren.

### Natürliche Sprachverarbeitung (NLP):

NLP-Modelle verwenden große Mengen an Textdaten, um menschenähnliche Sprachverarbeitung zu erlernen. Bei der Anwendung analysieren sie Kundenbewertungen automatisch, identifizieren Schlüsselwörter und erkennen die Stimmung, um Unternehmen Einblicke in die Kundenmeinung zu geben.

### Empfehlungssysteme:

Empfehlungssysteme nutzen Algorithmen wie kollaboratives Filtern. Sie analysieren das Verhalten ähnlicher Benutzer, um Vorhersagen darüber zu treffen, welche Produkte oder Inhalte einem Benutzer gefallen könnten. Diese Vorhersagen werden dann für personalisierte Empfehlungen verwendet.

### Gesichtserkennung:

Gesichtserkennungsalgorithmen extrahieren Merkmale wie Augen, Nase und Mund aus Bildern. Diese Merkmale werden dann verwendet, um Gesichter zu identifizieren. In Smart Cities werden Kameras mit diesen Algorithmen ausgestattet, um öffentliche Bereiche zu überwachen.





# Ethik und Fairness in Maschinellen Lernen

Dr Houssam Jedidi - Provadis





## Definition:

Ethik und Fairness im maschinellen Lernen beziehen sich auf die Verantwortung und Gerechtigkeit bei der Entwicklung und Anwendung von ML-Modellen. Es geht darum, sicherzustellen, dass Algorithmen transparent, diskriminierungsfrei und für alle Nutzer fair sind

## Bias in Modellen:

Modelle können durch Voreingenommenheit beeinflusst werden, abhängig von den Daten, mit denen sie trainiert werden. Hier liegt der Fokus darauf sicherzustellen, dass Modelle nicht diskriminierend sind und gleiche Behandlung für unterschiedliche Gruppen gewährleisten



## Datenschutz (DSGVO):

Datenschutz im maschinellen Lernen zielt darauf ab, persönliche Daten zu schützen. Modelle sollen so entwickelt werden, dass sie sensible Informationen angemessen behandeln und Datenschutzrichtlinien respektieren.

## Verantwortungsbewusste Entwicklung:

Entwickler sollten ethische Grundsätze in den gesamten Lebenszyklus von ML-Systemen integrieren. Dies beinhaltet die sorgfältige Datensammlung, transparente Modellentwicklung, umfassende Tests und eine kontinuierliche Überwachung, um sicherzustellen, dass die Technologie positiv und verantwortungsbewusst genutzt wird.



# Zukünftige Entwicklungen

### Erweiterte KI-Modelle:

Die rasante Evolution von KI-Modellen, insbesondere mit beeindruckenden Beispielen wie GPT-4o, deutet auf eine Ära hin, in der diese Modelle in der Lage sind, nicht nur komplexe Muster zu verstehen, sondern auch kontextbezogene Aufgaben zu meistern. Dies ermöglicht KI-Anwendungen, die über herkömmliche Anwendungen hinausgehen und eine tiefgreifende Interaktion mit menschenähnlichem Verständnis bieten.

Dr Houssam Jedidi - Provadis



### AI der nächsten Generation- Autonomous AI

Autonomous AI: Selbstlernende Systeme, die ohne menschliche Eingriffe eigenständig agieren.

Beispiel: Fortschritte in der autonomen Mobilität durch Unternehmen wie Waymo oder Tesla werden durch autonome KI-Systeme vorangetrieben, die in Echtzeit Entscheidungen treffen können (Waymo AI, 2024).

Perspektive:

Die Entwicklung von autonomen Entscheidungsprozessen für Industrie- und Dienstleistungssektoren könnte eine massive Umwälzung traditioneller Geschäftsmodelle bedeuten.

Dr Houssam Jedidi - Provadis

### Quantum Computing

- **Forschungsfokus:**

- **Quantum Computing** wird die Berechnungskapazitäten exponentiell erweitern und traditionelle Algorithmen revolutionieren.
- von Bits zu Qubits | von Seriell zu Parallel (Dank Überlagerung)

- **Perspektive:**

- **Optimierung von Geschäftsprozessen:** Quantencomputer könnten komplexe Berechnungen in den Bereichen Logistik, Finanzwesen und Gesundheitswesen deutlich beschleunigen.
- **Beispiel:** Volkswagen testet Quantenalgorithmen zur Optimierung von Verkehrsflüssen in Städten und spart dadurch Zeit und Kosten ~10% (**Volkswagen Research, 2023**).
- **Google Quantum AI:** In der Forschung zur Krebsbehandlung könnte ein Quantencomputer verwendet werden, um zu analysieren, wie verschiedene Medikamente auf bestimmte Tumorzellen wirken. Dies kö

### Quantum Computing

- **Forschungsfokus:**
  - **Quantum Computing** wird die Berechnungskapazitäten exponentiell erweitern und traditionelle Algorithmen revolutionieren.
  - von Bits zu Qubits | von Seriell zu Parallel (Dank Überlagerung)
- **Perspektive:**
  - **Optimierung von Geschäftsprozessen:** Quantencomputer könnten komplexe Berechnungen in den Bereichen Logistik, Finanzwesen und Gesundheitswesen deutlich beschleunigen.
  - **Beispiel:** Volkswagen testet Quantenalgorithmen zur Optimierung von Verkehrsflüssen in Städten und spart dadurch Zeit und Kosten ~10% (**Volkswagen Research, 2023**).
  - **Google Quantum AI:** In der Forschung zur Krebsbehandlung könnte ein Quantencomputer verwendet werden, um zu analysieren, wie verschiedene Medikamente auf bestimmte Tumorzellen wirken. Dies kö

### Erklärbarkeit und Ethik:

Die steigende Bedeutung der Erklärbarkeit von KI-Entscheidungen ist von essenzieller Bedeutung.

Zukünftige Modelle sollen nicht nur akkurate Vorhersagen liefern, sondern auch den Grund für ihre Entscheidungen verständlich machen.

Dies fördert Vertrauen und ermöglicht eine ethischere Nutzung von KI in verschiedenen Anwendungsbereichen, insbesondere wenn es um sensible Daten und kritische Entscheidungen geht.

Dr Houssam Jedidi - Provadis



## 1. EU-KI-Verordnung (AI Act):

1. Regelt KI in Hochrisikobereichen (Gesundheit, Justiz).
2. Strenge Vorgaben für **Transparenz** und **Zertifizierung**. (European Commission, 2024)

## 2. World Economic Forum:

1. Globale Richtlinien für **KI-Ethik** (Datenverantwortung, Nicht-Diskriminierung).
2. Unternehmen passen KI-Strategien an WEF-Standards an. ( WEF, 2023)

## 3. USA - Algorithmic Accountability Act:

1. Verpflichtet Unternehmen zu **KI-Audits & IP-Rechte** (Bias, Diskriminierung).
2. Beispiel: **Personalrekrutierung** – Algorithmen müssen diskriminierungsfrei sein. (U.S. Congress, 2023)

### • Wirkung auf Unternehmen:

- Integration ethischer Standards zur Sicherstellung von **Rechtssicherheit** und **Vertrauen**.
- Ethisch verantwortungsvolle KI schafft Wettbewerbsvorteile.

# Danke

Dr Houssam Jedidi - Provadis

STUDENTS BY PROVADIS  
**THINKING** ↑  
**INDUSTRY** NEW