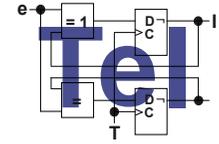


:

Anwendung von Analyseverfahren auf Daten des
Mikroprozessorentwurfs zur Produktivitätsoptimierung

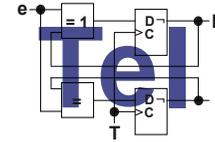
Stephan Radke

Gliederung

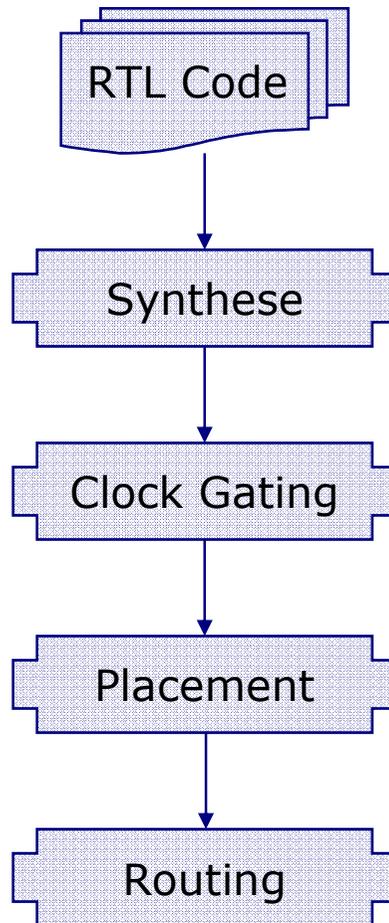


- Motivation
- Prozess der Wissensentdeckung
- Vorstellung von Data-Mining Techniken
- Erste Thesen

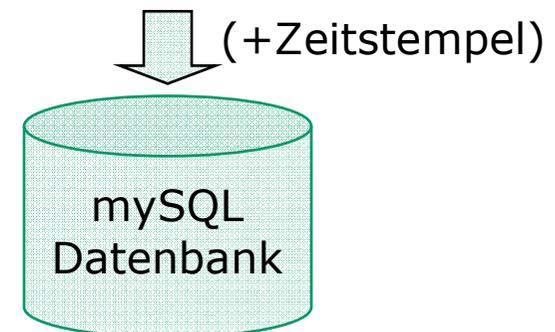
Analysedaten - Schaltungsimplementierung



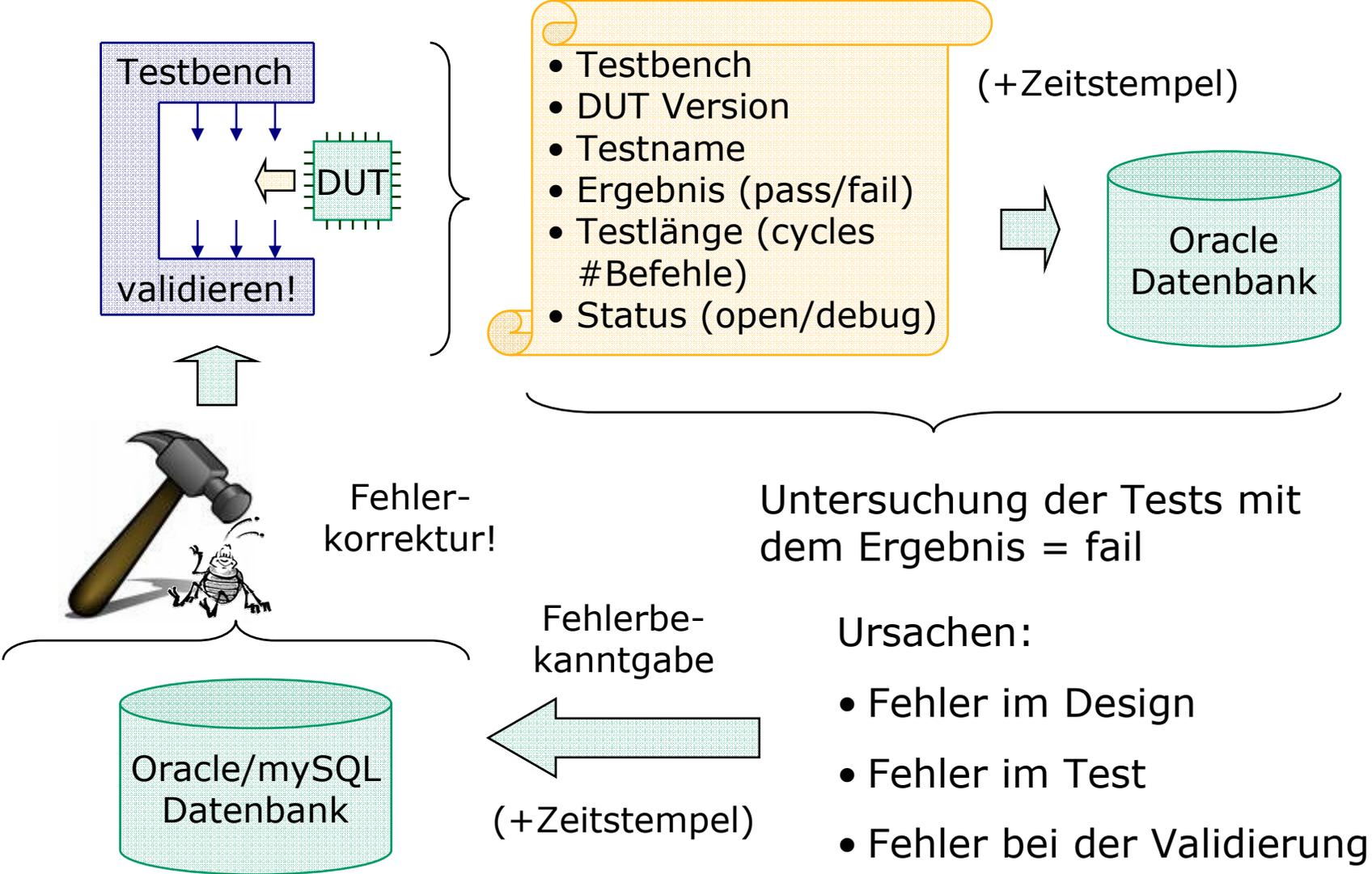
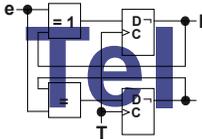
Entwicklungs-
prozess:
(vereinfacht)

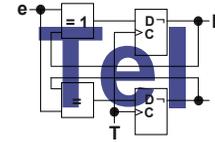


- Blockgröße
- # Std. Zellen
- # Pins
- # Noise Verletzungen
- # Elektromigrationsfehler
- # Spannungseinbrüche
- Layout vs. Schematic
- etc.



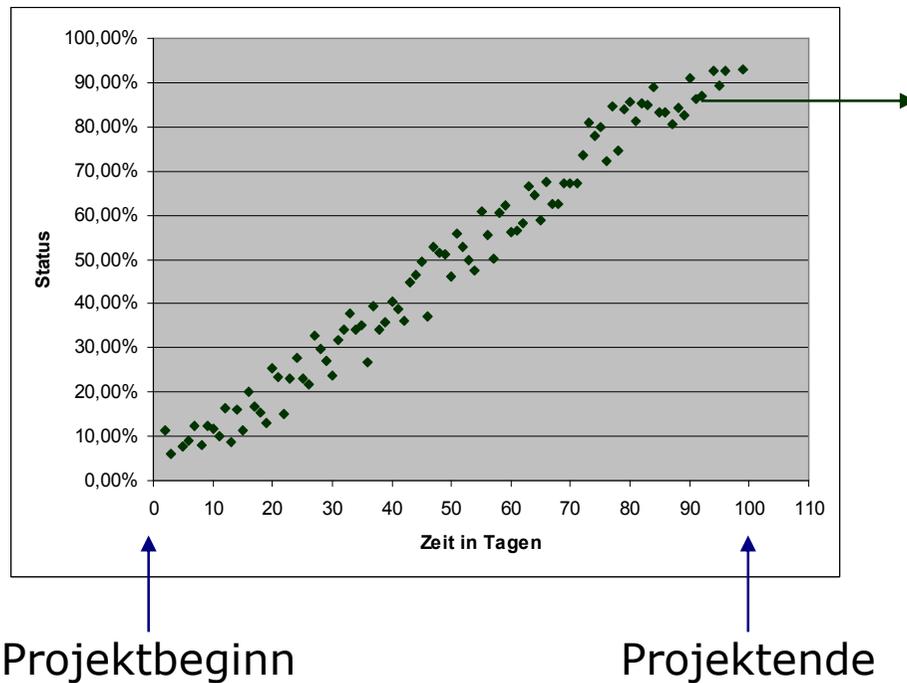
Analysedaten - Verifikation





Motivation

Rekonstruktion des Entwurfsprozesses:

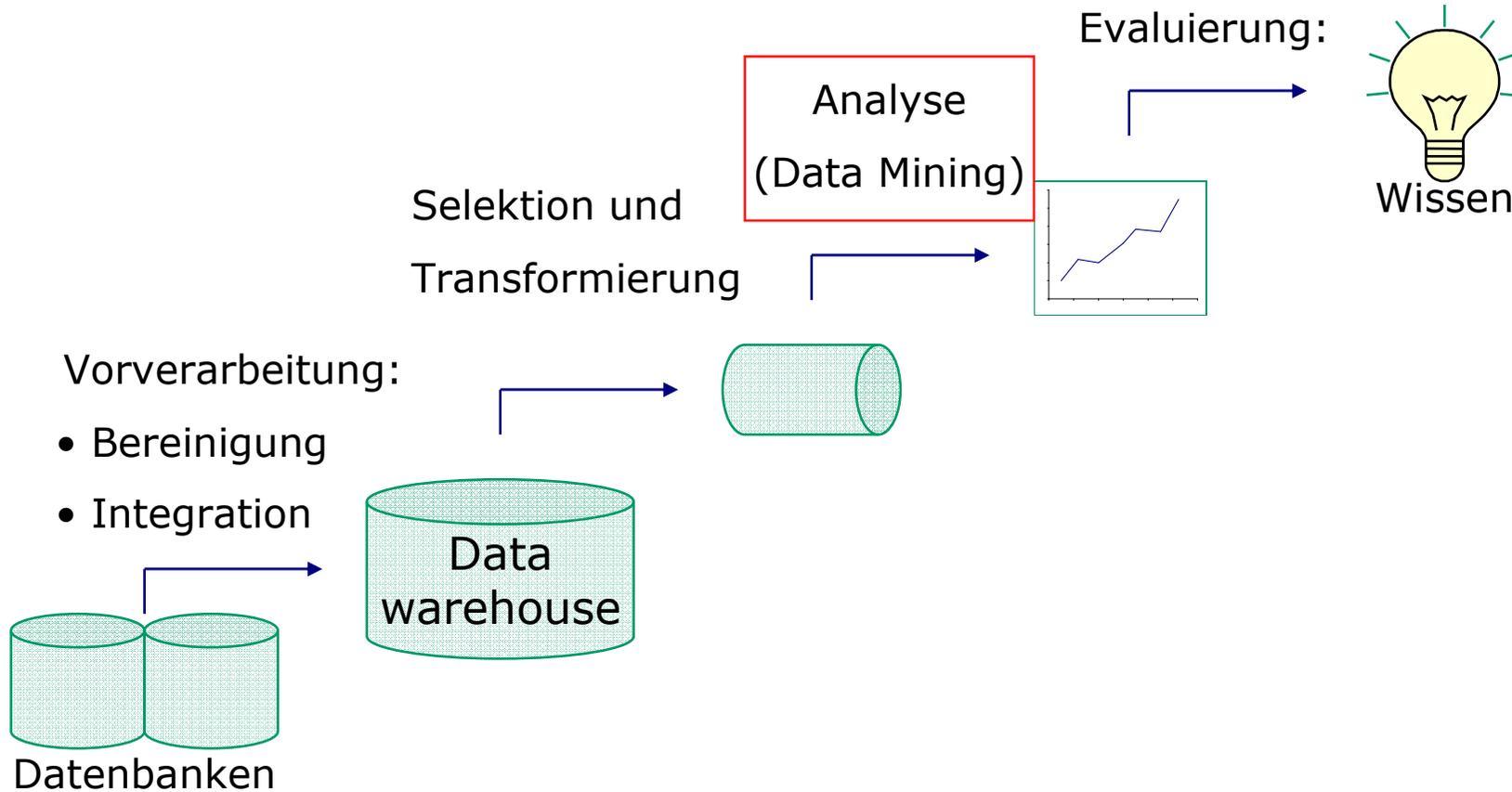
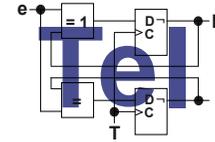


$$f(t, a_1, a_2, \dots, a_n)$$

wobei a_i ($i \in \mathbb{N}, 1 \leq i \leq n$) ein relevantes Attribut einer Relation R (oder mehrerer Relationen $R_a \times R_b \times \dots$) zu einem speziellen Zeitpunkt t ist.

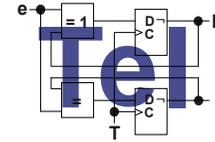
- relevante Attribute bestimmen
- Aufstellen der Funktion f

Prozess Knowledge Discovery in DBs



[vgl. Han, Jiawei / Kamber, Micheline: Data Mining, 2001, S.6]

Data Mining Überblick



Klassifikation /
Prognose

Zuordnung der Datensätze zu Klassen
(Klassen und eine teilweise Zuordnung
der Datensätze muss gegeben sein)

Segmentierung

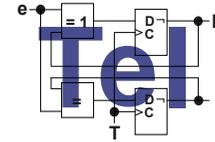
Automatisiertes Finden der Klassen
(Klassen müssen nicht vordefiniert
werden)

Assoziation

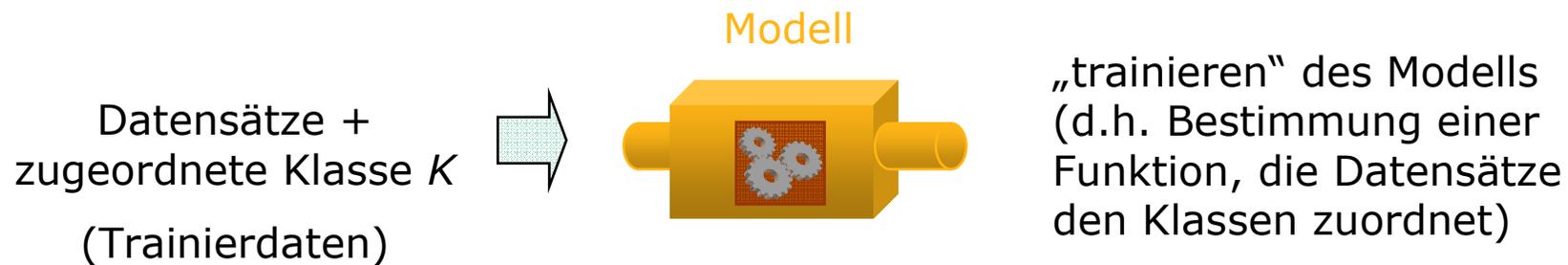
Automatisiertes Finden von
Zusammenhängen innerhalb der
Datensätze bzw. deren Attribute

[vgl. *Dunham, Margaret H.: Data Mining, 2003*]

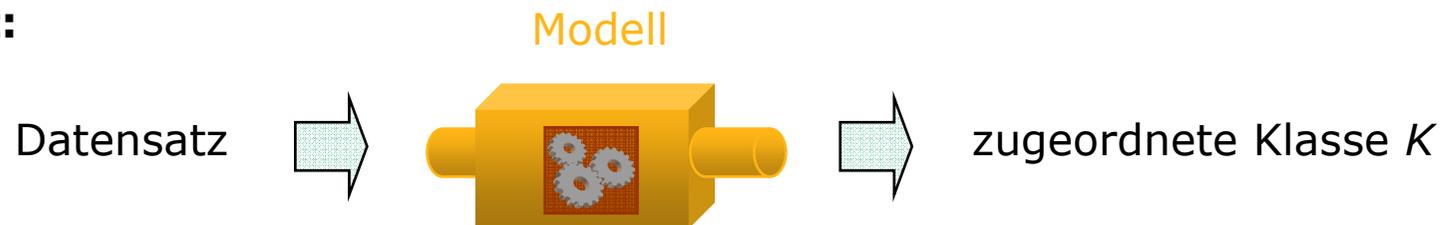
Klassifikation / Prognose Funktionsweise



1. Schritt:



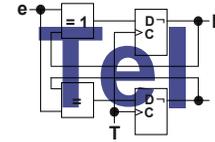
2. Schritt:



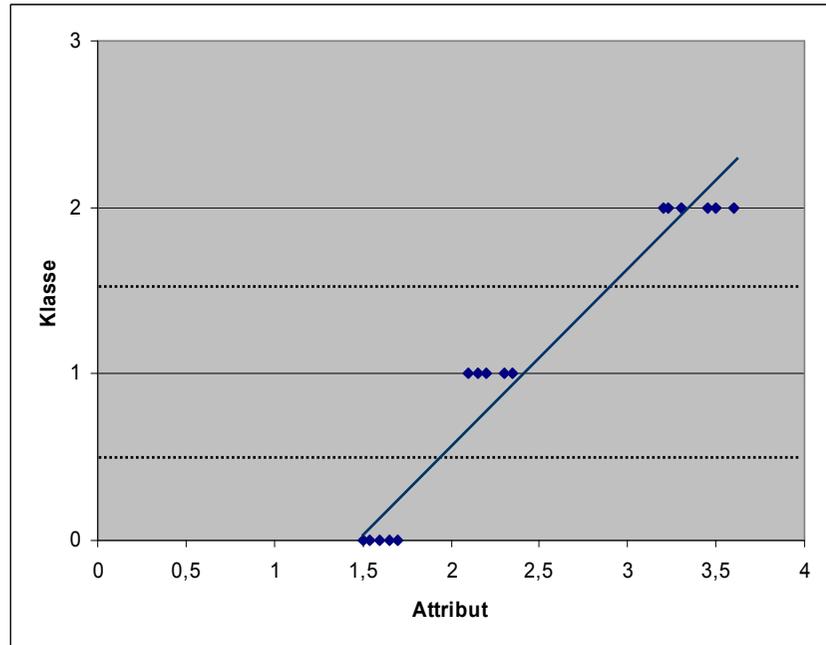
Modell kann fortlaufend „trainiert“ werden (falls die Datensätzen die Klassenzuordnung beinhalten oder besondere Modellalgorithmen verwendet werden)

Klassifikation / Prognose

Statistische Analysen



Regression:



Im Beispiel:

$$y = f(x) = c_0 + c_1x$$

Methode der minimalen Fehlerquadrate:

$$\rightarrow f(x) \approx -1,54 + 1,06x$$

Bestimmung einer Funktion

$$y = f(x)$$

So dass:

für alle (x_i, y_i) der Punktmenge der quadratische Fehler ε^2 in der Gleichung

$$y_i = f(x_i) + \varepsilon$$

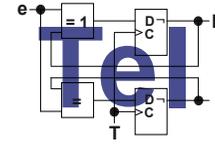
minimal wird.

- Vorteil:
- sehr einfache Berechnungsvorschrift
 - funktioniert mit kontinuierlichen Daten

- Nachteil:
- nur in spez. Fällen anwendbar

Klassifikation / Prognose

Statistische Analysen



Wahrscheinlichkeitsberechnung:

Bestimmung der Wahrscheinlichkeiten für die Zugehörigkeit eines Datensatzes d zu den jeweiligen Klassen K :

$$P(K_i | d), K_i \in K$$

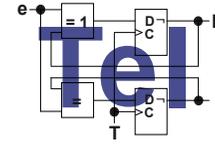
Zuordnung zur Klasse K_i erfolgt für die größte Wahrscheinlichkeit.

Vorteil: - einfaches „trainieren“

Nachteil: - kontinuierliche Daten müssen diskretisiert werden
- Wahrscheinlichkeitsfehler für abhängige Eingangsgrößen

Klassifikation / Prognose

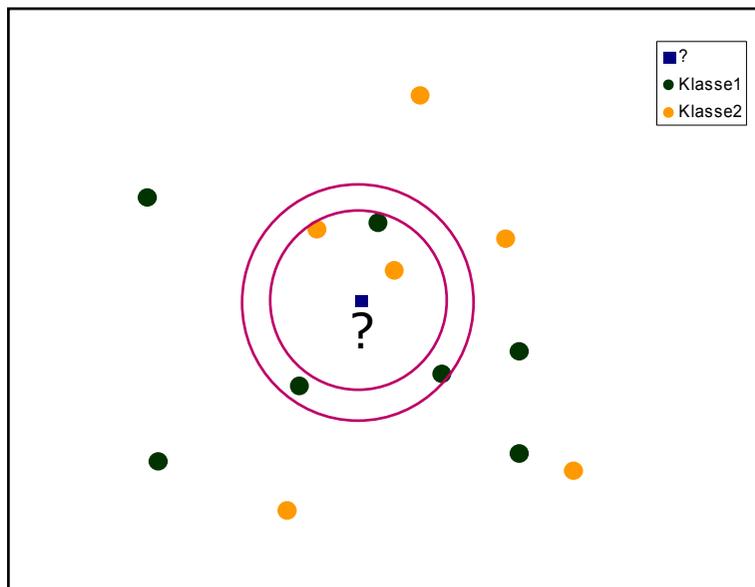
Distanzanalyse



K Nearest Neighbors:

Ermittlung der K-nächsten Nachbarn durch:

- Euclidean Distanz: $dis(d_i, d_j) = \sqrt{\sum_{h=1}^k (d_{ih} - d_{jh})^2}$
- Manhattan Distanz: $dis(d_i, d_j) = \sum_{h=1}^k |d_{ih} - d_{jh}|$



$K = 3, ? \rightarrow \text{Klasse } K_2$

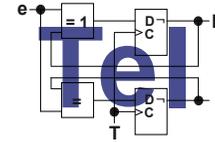
$K = 5, ? \rightarrow \text{Klasse } K_1$

Vorteil: - kontinuierliches Lernen möglich

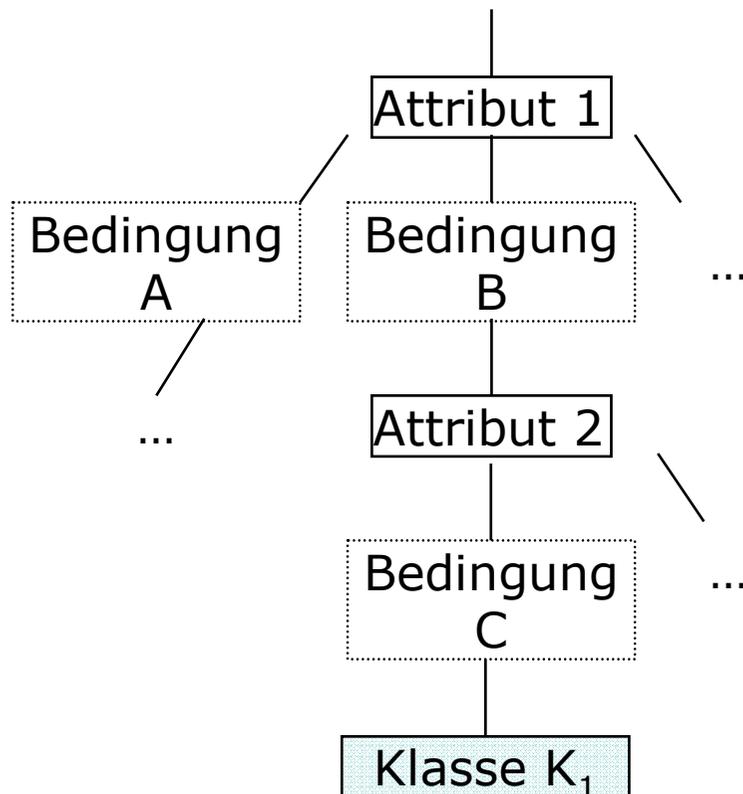
Nachteil: - stark abhängig von K
- hoher Speicherbedarf bei kontinuierlichem Lernen

Klassifikation / Prognose

Entscheidungsbäume



1. Entscheidungsbaum aufbauen
2. Klassifikation erfolgt entlang des Baumes



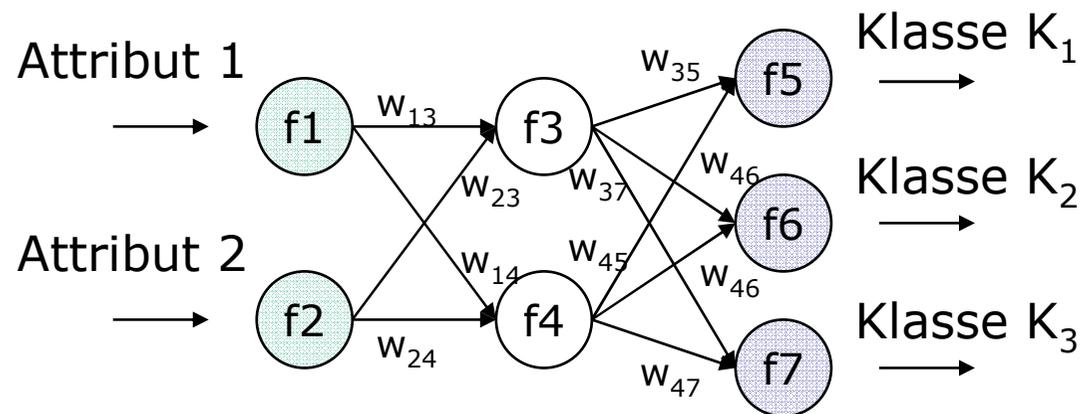
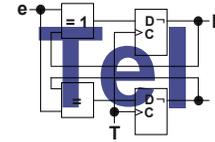
Algorithmen basieren auf Gewinnbeurteilungen vor und nach einer Entscheidung

Vorteil: - einfach überschaubar
(kompakte Form für Entscheidungsalgorithmen)

Nachteil: - sequentielles Betrachten der Attribute

Klassifikation / Prognose

Neuronale Netze



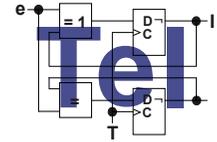
f_i : - lineare Funktion
- Sprungfunktion
- Tangens hyperbolicus
- S-förmige Funktion
- Gaussian Funktion

Vorteil: - fehlertolerant
- verarbeiten verrauschte,
unvollständige, widersprüch-
liche Datensätze

Nachteil: - kein garantierter Lernerfolg
(Training des Netzes
konvergiert nicht)

Evaluation

Wahrscheinlichkeitsmatrix



Zugeordnete Klasse	Reale Klasse		
		K_i	Nicht K_i
K_i		richtig positive (a)	falsch positive (b)
Nicht K_i		falsch negative (c)	richtig negative (d)

Sensitivität: $a / (a+c)$

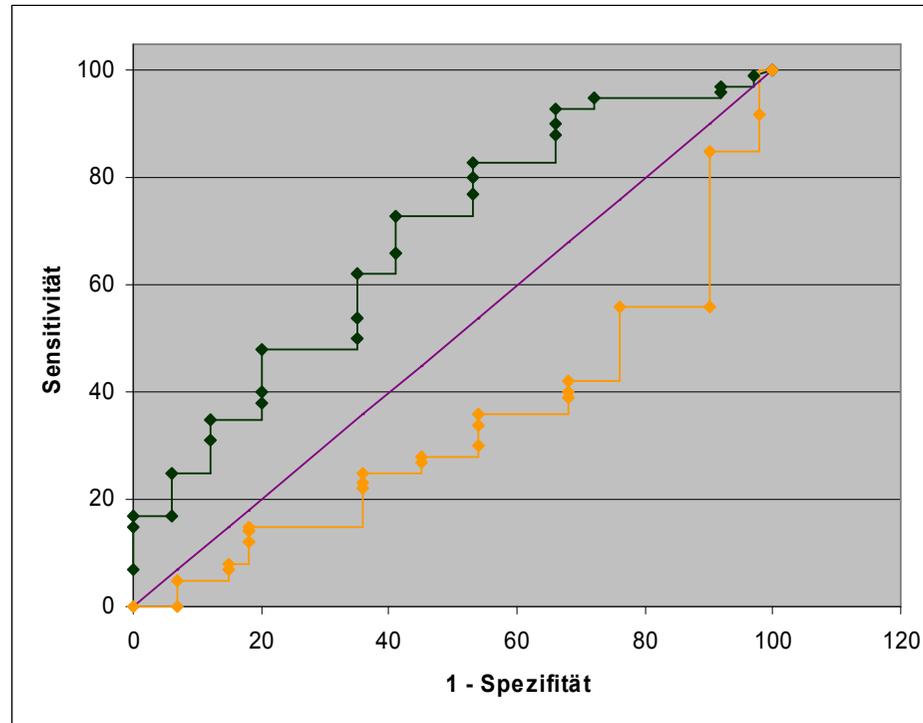
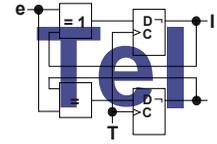
$P(\text{positiv erkannt} \mid \text{tatsächlich positiv})$

Spezifität: $d / (b+d)$

$P(\text{negativ erkannt} \mid \text{tatsächlich negativ})$

Genauigkeit: $(a+d) / (a+b+c+d)$ $P(\text{richtig klassifiziert})$

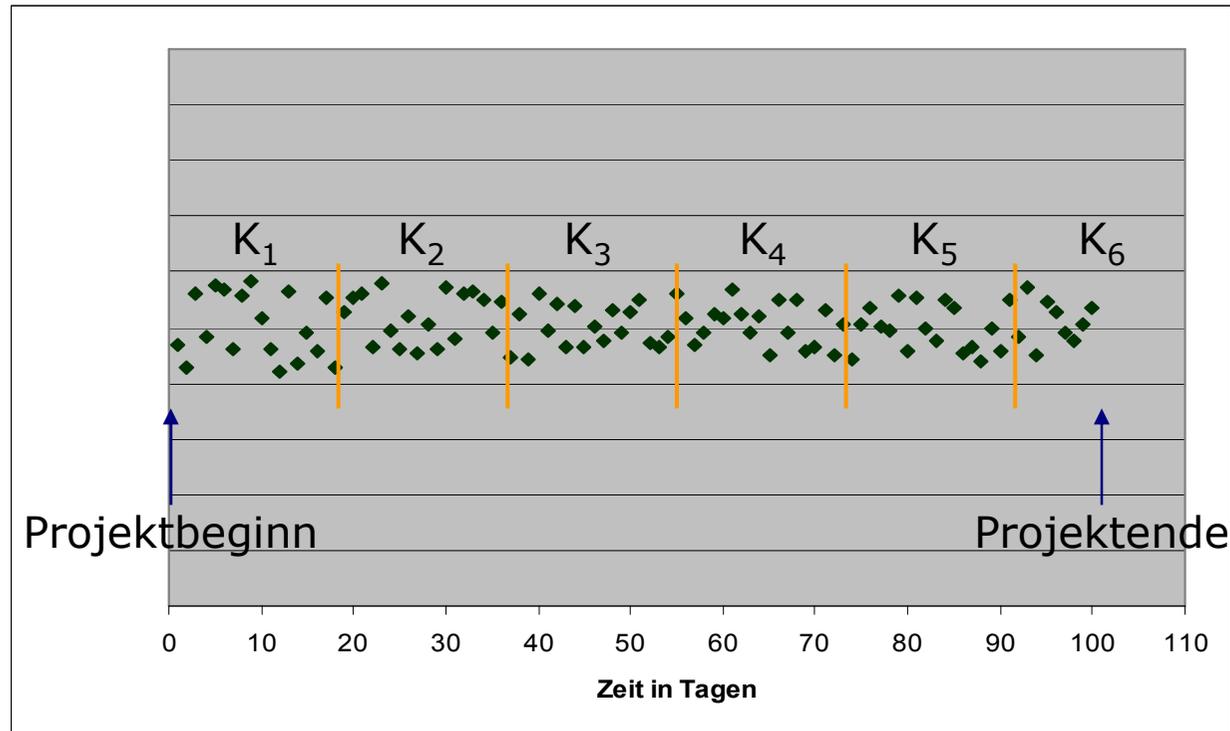
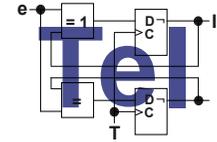
Evaluation Receiver Operating Characteristic



- Fläche unter der Kurve repräsentiert Genauigkeit des Klassifikators
- (blau → guter Klassifikator,
gelb → schlechter Klassifikator)

Erste Thesen

Anwendung der Klassifizierung



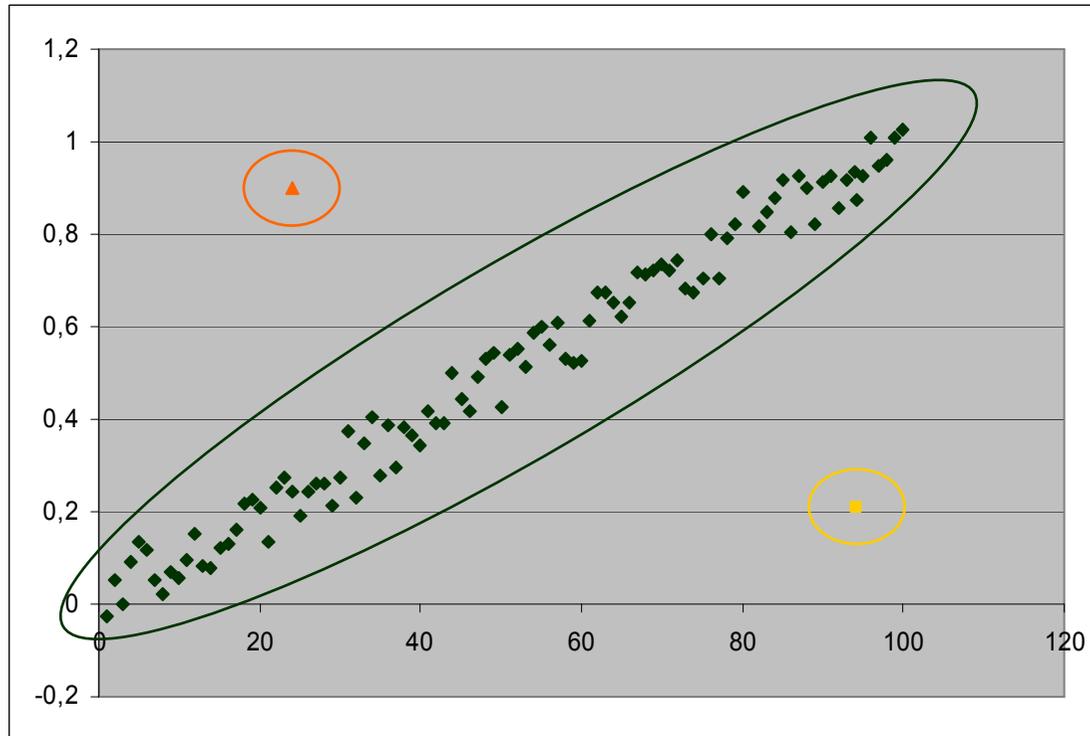
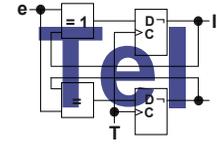
Manuelle Klassenbestimmung in einem Projekt
(Gleichverteilung oder nach Meilensteinen)

└─┬─> „trainieren“ eines Modells

└─┬─> Anwenden des Modells auf
ein anderes Projekt

Erste Thesen

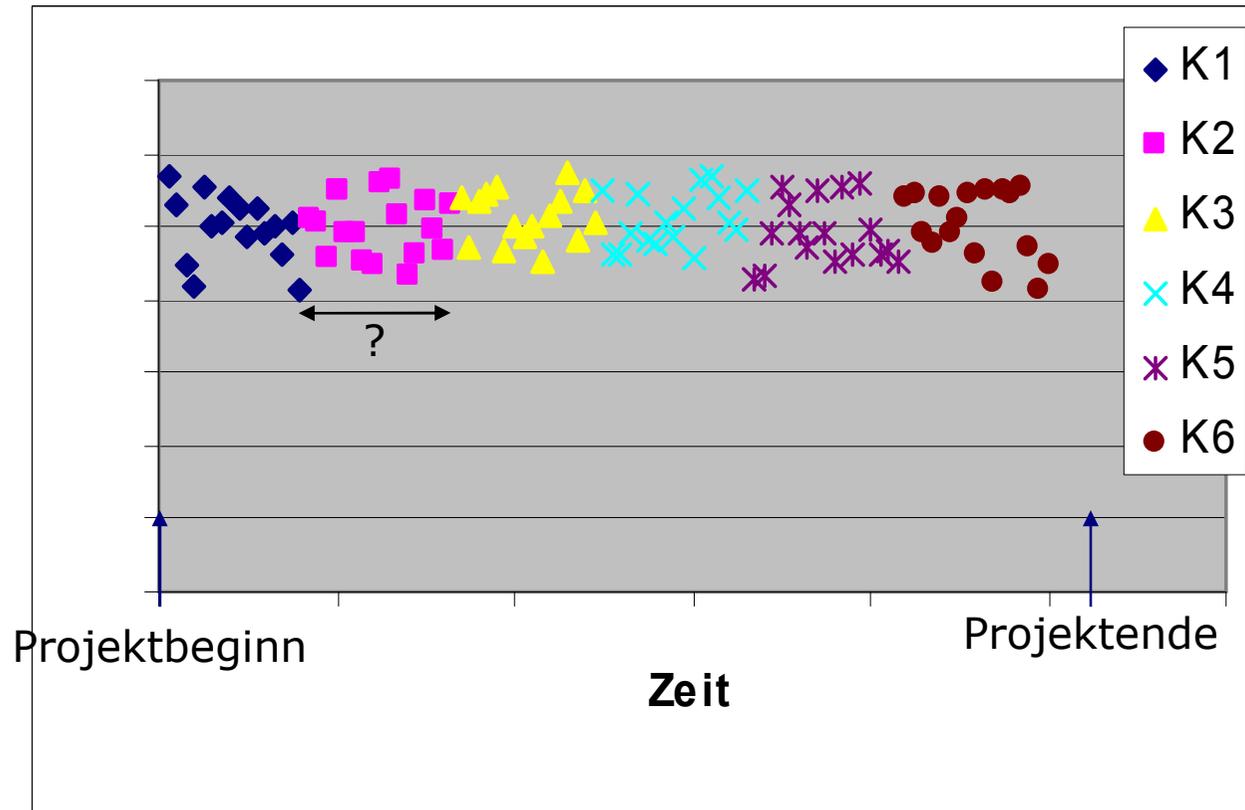
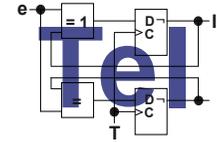
Anwendung der Segmentierung



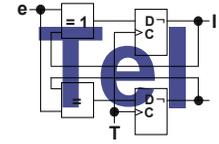
Ausreißer identifizieren

Erste Thesen

Anwendung der Assoziationsregeln



Indikatoren für die Zeitspanne bestimmten



Vielen Dank.