

# Machine Learning: Support Vector Machine

[Stützvektormaschine]

**Präsentation – 26. Januar 2018**

Hauptseminar FG Wirtschaftsinformatik für Industriebetriebe

Benjamin Wörrlein

Tobias Rummelsberger

# Agenda

- Grundlagen
- SVM
- Werkzeuge

1

- Datensatz
- Anwendung

2

- Kritische Würdigung
- Evaluation mit Testdatensatz

3

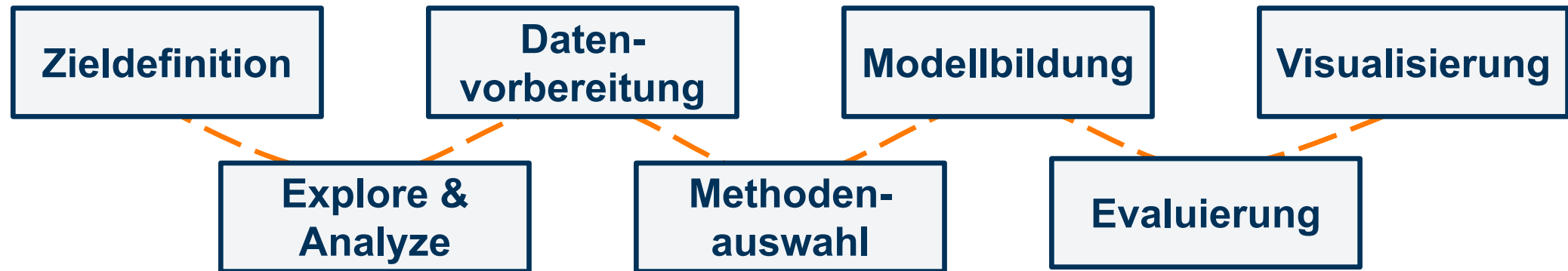
# GRUNDLAGEN

KDD | Machine Learning

# Machine Learning

## Prozess

**Ziel: Erstellung deskriptiver und prädiktiver Modelle auf Basis eines Datensatzes**



- Iterativer Prozess – ggf. wird zu vorherigen Schritten zurückgesprungen
- Kontinuierliche Optimierung des Modells

nach [Ha2009] und [FPS1996]

# SUPPORT VECTOR MACHINE

Begriffe | SVC | SVR

# Begriffe

**Hyperplane:** Der Unterraum (bspw. Ebene), der die Klassen voneinander trennt

**Margin:** Der Abstand zwischen den Stützvektoren zweier Klassen (orthogonal zur Hyperplane)

**Stützvektoren:** Die Datenelemente, die der Hyperplane am nächsten liegen

# Support Vector Classifier

## Hard Margin Classifier

- Keine Datenelemente innerhalb des Randes

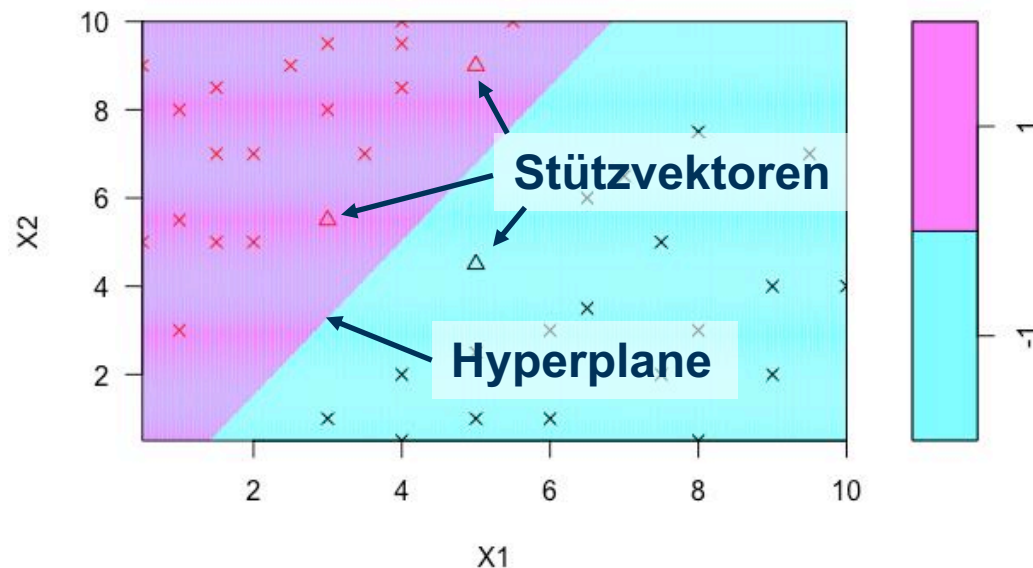


Abbildung 1: Hard-Margin Classifier [eigene Darstellung]

## Soft Margin Classifier

- Mehrere Stützvektoren
- Overfitting unwahrscheinlicher

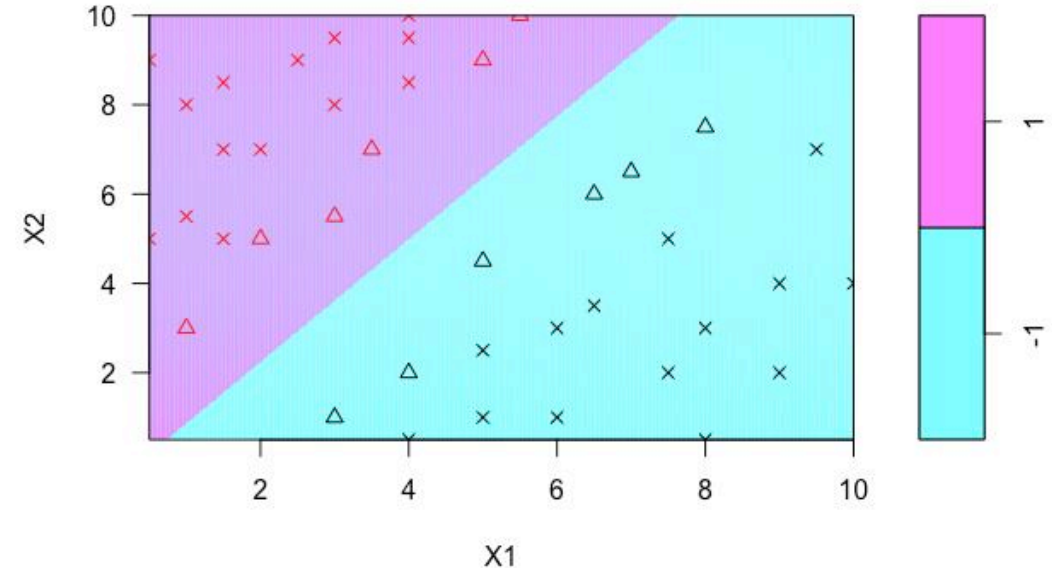


Abbildung 2: Soft-Margin Classifier [eigene Darstellung]

# Support Vector Classifier

**C-Classification:** Einflussnahme auf die Breite des Randes mit dem Kostenbudget

**Nu-Classification:** Einflussnahme auf die relative Anzahl der Stützvektoren mit dem Parameter  $\nu$

**One-Classification:** Neuigkeitsentdeckung und Detektion von Ausreißern



# Support Vector Regression

**Eps-Regression:** Einflussnahme auf die Breite des Randes mit dem Kostenbudget

**Nu-Regression:** Einflussnahme auf die relative Anzahl der Stützvektoren mit dem Parameter  $\nu$

# Multiklassen-Klassifikation

- Multiklassen bestehen aus mehreren binären Klassen
- Testen der Klassen gegeneinander

**One versus all:** Elemente **einer** Klasse werden gegen Elemente **aller** anderen Klassen getestet

**One against One:** Element **einer** Klasse wird jeweils gegen das Element **einer** Klasse getestet

# WERKZEUGE

Programmiersprache | Bibliotheken | Entwicklungsumgebung

# Programmiersprache


## R

- Verwendete Sprache: R
- Orientiert sich an der kommerziellen Sprache S
- Freie Programmiersprache für statistische Berechnungen und Visualisierung
- Verfügbarkeit von SVM-Paketen
  - bspw. e1071



# Bibliotheken

packages

- Für Datenmanipulation, Datenanalyse und Berechnung der SVM
  - binr
  - discretization
  - dplyr
  - modelr
  - e1071  Erstellen und „trainieren“ einer SVM
- Alle Pakete sind unter <https://cran.r-project.org> abrufbar und dokumentiert

# Entwicklungsumgebung

## R Studio

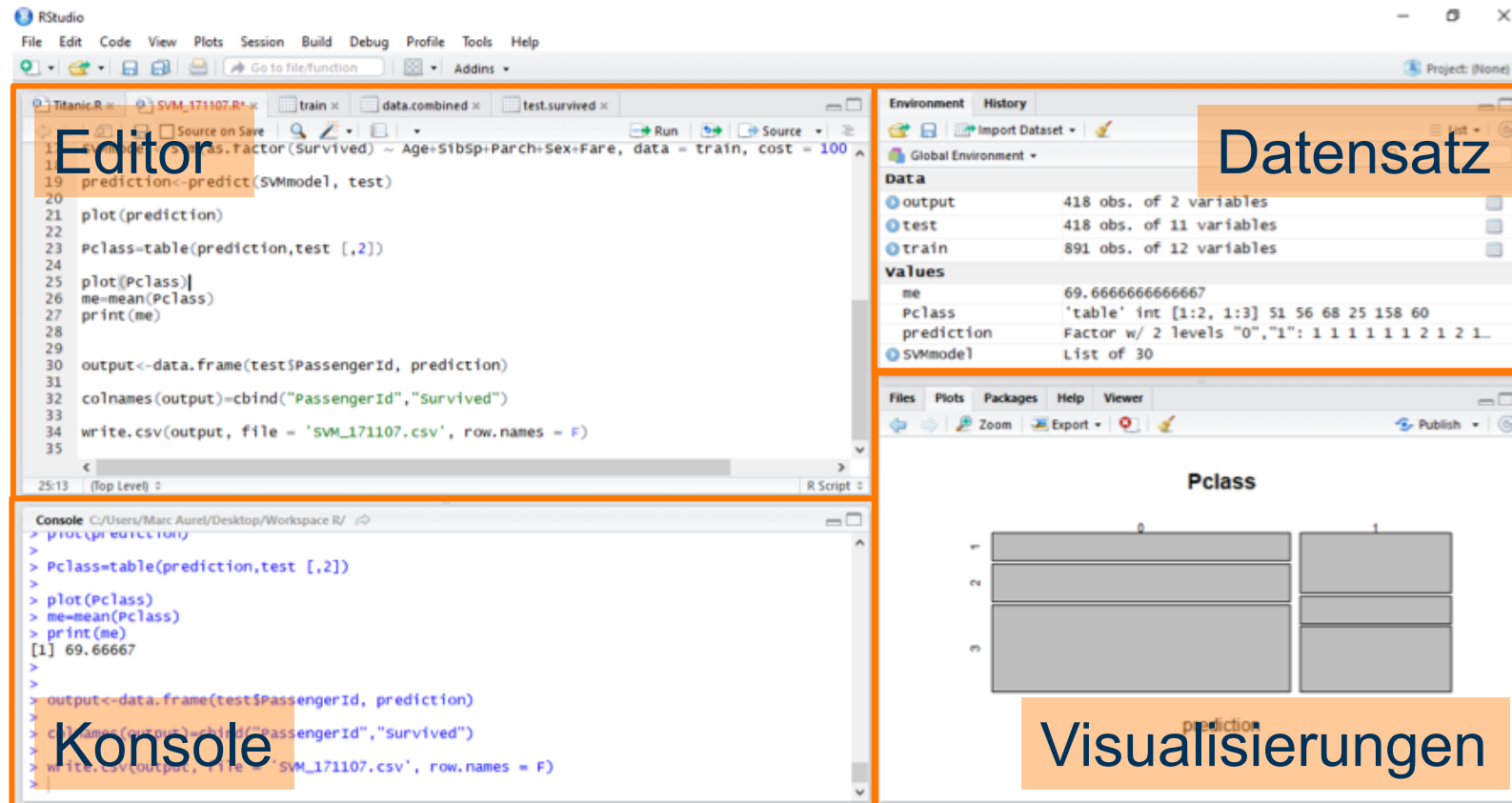
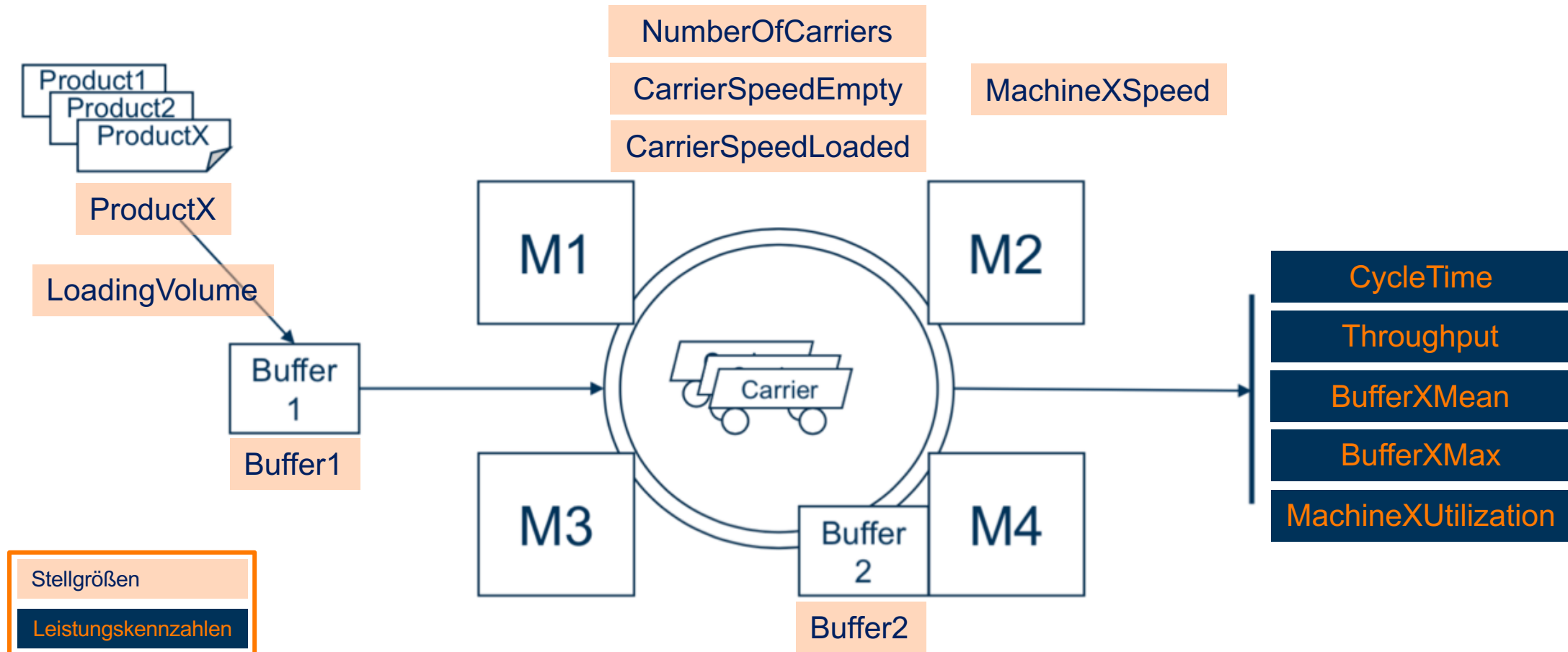


Abbildung : Entwicklungsumgebung R Studio [eigene Darstellung]

# DATENSATZ

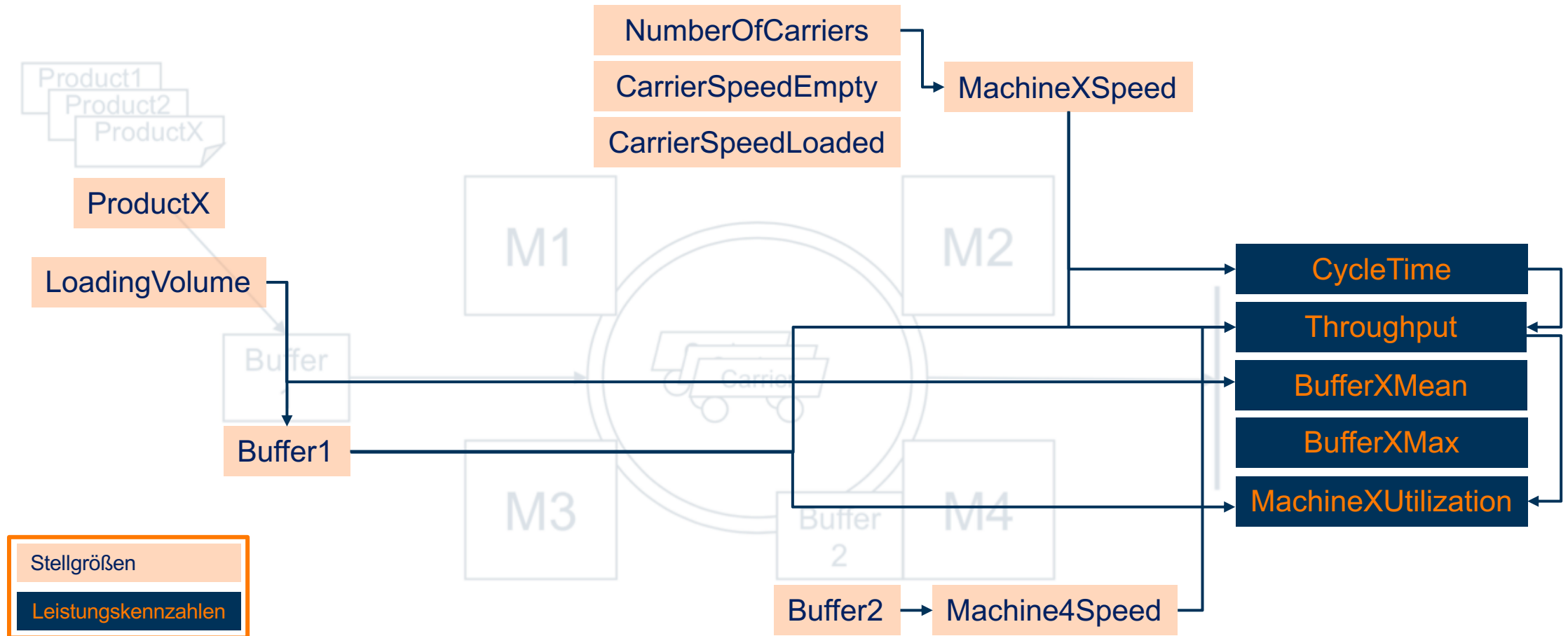
Systemübersicht | EDA

# Systemübersicht





# Systemübersicht



# Exploratory Data Analysis

	Attributname	Beschreibung
Stellgrößen	LoadingVolume	Auftragslast
	Buffer1, Buffer2	Kapazität der Puffer
	NumberOfCarriers	Anzahl Werkstückträger
	CarrierSpeedLoaded	Geschwindigkeit im beladenen Zustand
	CarrierSpeedEmpty	Geschwindigkeit bei Leerfahrt
	ProductX [1-27]	%-Anteil von Produkt X im Auftragsmix
	MachineXSpeed [1-4]	Effizienzfaktor für Maschine X
Leistungskenn- zahlen	Throughput	Gesamtausbringungsmenge
	Buffer1Mean, Buffer2Mean	Durchschnittliche Belegung der Puffer
	Buffer1Max, Buffer2Max	Maximal beobachtete Belegung der Puffer
	CycleTime	Durchschnittliche Durchlaufzeit eines Auftrages
	MachineXUtilization [1-4]	Durchschnittliche Auslastung von Maschine X

# ANWENDUNG

Vorbereitung | Parametrisierung | Modellbildung

# Datenvorbereitung

## Normierung

- Stellgrößen werden normiert, damit keine der Stellgrößen die Modellbildung dominiert
- Min-Max-Normierung in Intervall  $\{0, 1\}$

$$V'_i = \frac{v_i - \min_A}{\max_A - \min_A} (\text{new\_max}_A - \text{new\_min}_A) + \text{new\_min}_A$$

```
# Normieren der Daten
normalize <- function(x){
  return ((x-min(x) / max(x) - min(x)))
}
```

# Datenvorbereitung

## Diskretisierung

- Diskretisierung der Leistungskennzahlen für die Klassifikation
- Einteilen in fünf Klassen mit gleicher Anzahl an einzigartigen Werten (*Equal-Frequency-Binning*)
- Geringerer Rechenaufwand und Bündelung der Informationen

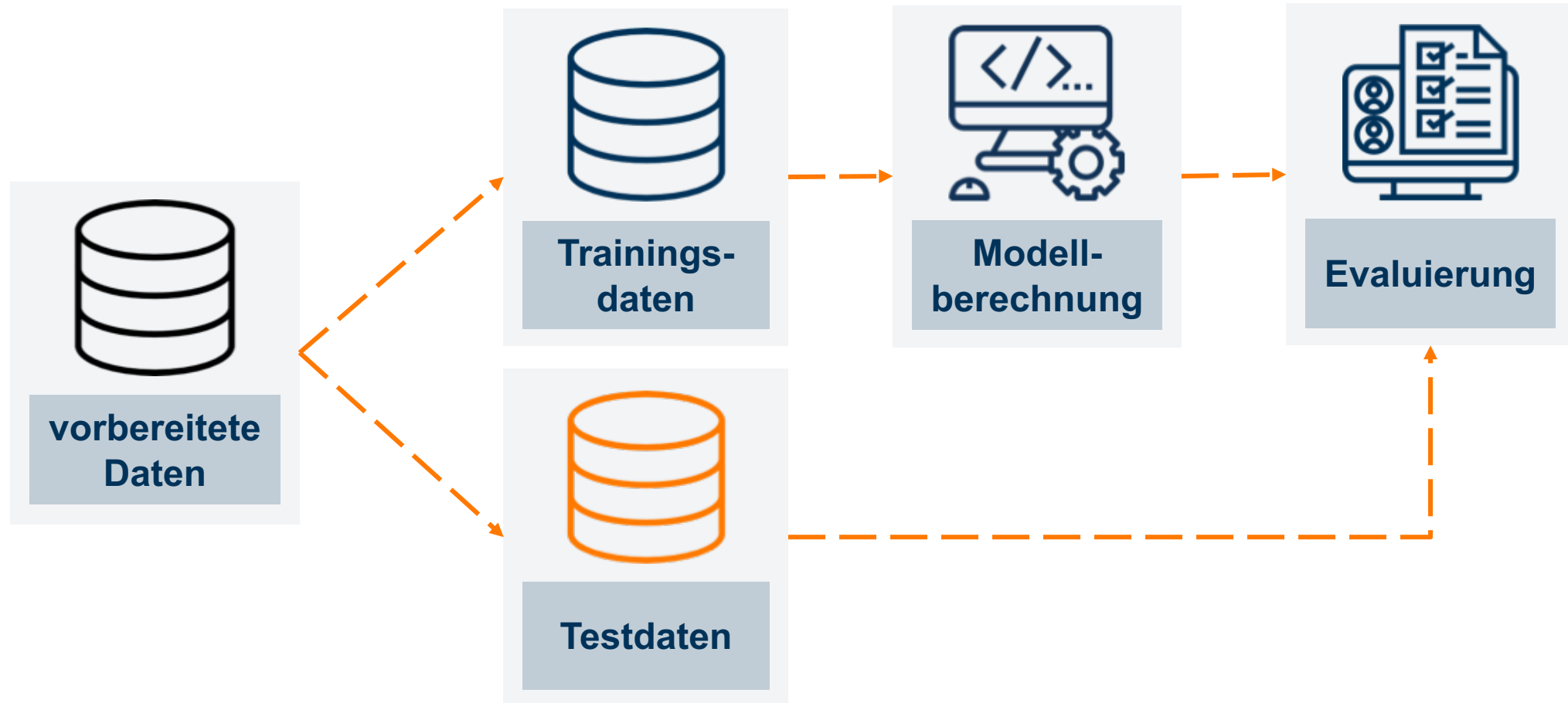
# Datenvorbereitung

## Diskretisierung

```
# Diskretisieren
# Initialisierung eines leeren Dataframes
Leistungskennzahlen_diskretisiert <- data.frame
# Diskretisieren der Leistungskennzahlen in 5 Klassen
for(i in 1:length(Leistungskennzahlen)) {
  Leistungskennzahlen_diskretisiert <- data.frame(cbind(Leistungskennzahlen_diskretisiert,
discretize(Leistungskennzahlen[[i]], 5, 5)))
  names(Leistungskennzahlen_diskretisiert[i]) <- names(Leistungskennzahlen[i])
  Leistungskennzahlen_diskretisiert[[i]] <- factor(Leistungskennzahlen_diskretisiert[[i]])
}
# Namen der Spalten übernehmen
Leistungskennzahlen_diskretisiert <- setNames(Leistungskennzahlen_diskretisiert,
c(names(Leistungskennzahlen)))
# Buffer1Max und Buffer2Max bestanden bereits aus diskreten Werten
Leistungskennzahlen_diskretisiert$Buffer1Max <- factor(Leistungskennzahlen$Buffer1Max)
Leistungskennzahlen_diskretisiert$Buffer2Max <- factor(Leistungskennzahlen$Buffer2Max)
saveRDS(Leistungskennzahlen_diskretisiert, "./Data/Leistungskennzahlen_diskretisiert.rds")
```

# Datenvorbereitung

Trainings- und Testdatensatz



# Modellbildung

## Tuning der Parameter

```
# Listen für Tuning
```

```
cost_list <- c(10^-5, 10^-3, 10^-1, 10^1, 10^3, 10^5, 10^7)
```

```
gamma_list <- c(10^-7, 10^-5, 10^-3, 10^-1, 10^1, 10^3, 10^5)
```

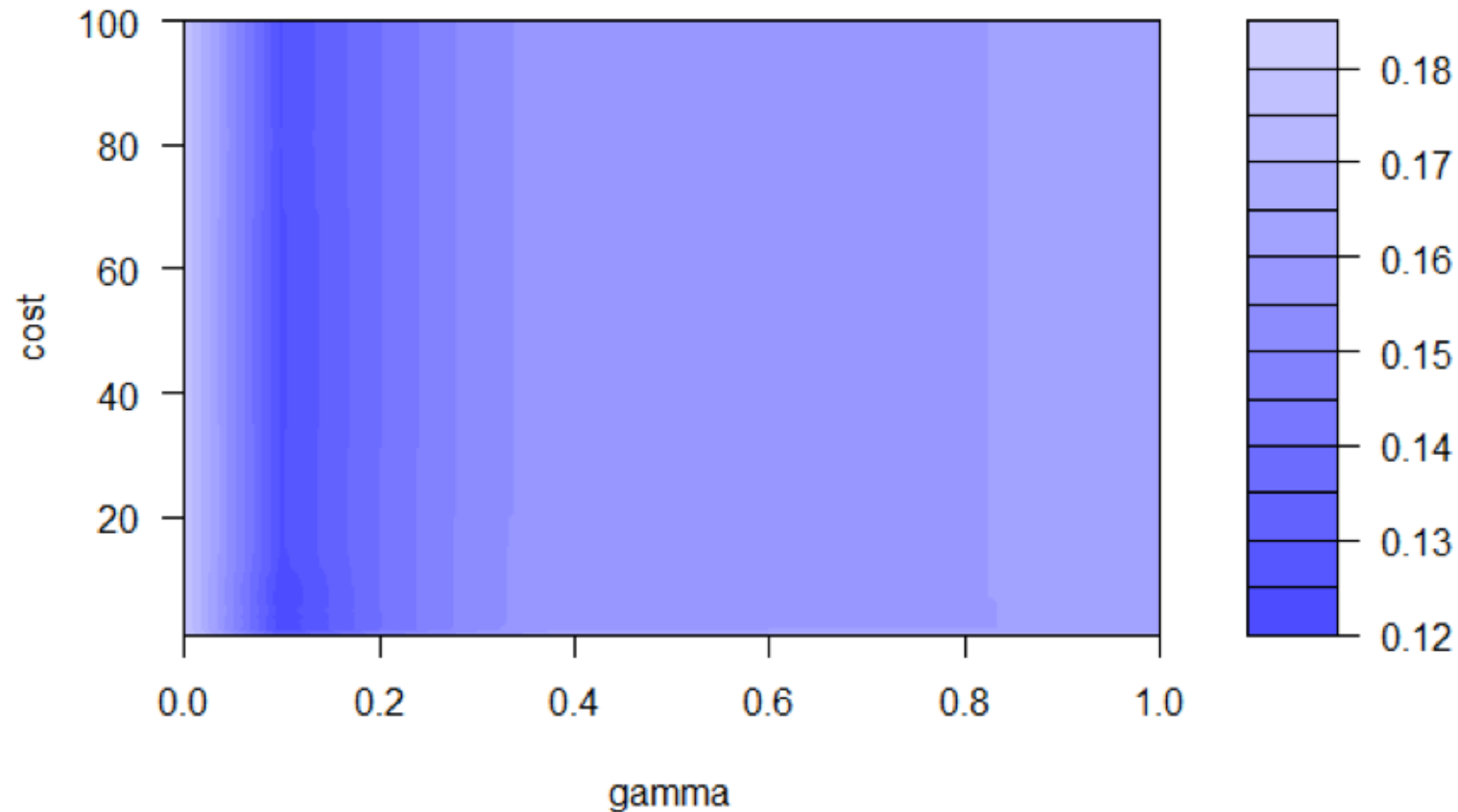
```
# SVR für Throughput
```

```
SVR_Throughput_tune <- tune(svm, Throughput ~ LoadingVolume + Buffer1 + Buffer2 +  
NumberOfCarriers + CarrierSpeedLoaded + CarrierSpeedEmpty + Product1 + Product2 + Product3 +  
Product4 + Product5 + Product6 + Product7 + Product8 + Product9 + Product10 + Product11 +  
Product12 + Product13 + Product14 + Product15 + Product16 + Product17 + Product18 + Product19  
+ Product20 + Product21 + Product22 + Product23 + Product24 + Product25 + Product26 +  
Product27 + Machine1Speed + Machine2Speed + Machine3Speed + Machine4Speed, data = trainset,  
type = "eps-regression", kernel = "radial", ranges = list(gamma = gamma_list, cost =  
cost_list), scale = T)  
print(SVR_Throughput_tune)  
plot(SVR_Throughput_tune)
```



# Modellbildung

## Tuning am Beispiel Throughput



*Abbildung 4: Kreuzvalidierung der SVC für  
Leistungskennzahl Throughput [eigene Darstellung]*

# Modellbildung

## am Beispiel Regression

```
SVR_Throughput <- svm(Throughput ~ LoadingVolume + Buffer1 + Buffer2 + NumberOfCarriers +  
CarrierSpeedLoaded + CarrierSpeedEmpty + Product1 + Product2 + Product3 + Product4 + Product5  
+ Product6 + Product7 + Product8 + Product9 + Product10 + Product11 + Product12 + Product13 +  
Product14 + Product15 + Product16 + Product17 + Product18 + Product19 + Product20 + Product21  
+ Product22 + Product23 + Product24 + Product25 + Product26 + Product27 + Machine1Speed +  
Machine2Speed + Machine3Speed + Machine4Speed , data = trainset)  
SVR_Throughput_validation <- validate.regression(SVR_Throughput, testset, testset$Throughput)  
  
hist(SVR_Throughput_validation$error_norm, breaks = seq(0,  
max(SVR_Throughput_validation$error_norm)+0.05, 0.05), main = "Histogramm der normierten  
Abweichung - Throughput", xlab = "normierte Abweichung", ylab = "Häufigkeit")  
  
saveRDS(SVR_Throughput, "./Model/Regression/SVR_Throughput.rds")
```

# KRITISCHE WÜRDIGUNG

Klassifikation | Regression | Zusammenfassung | kritische Würdigung

# Evaluation

## Klassifikation

Buffer2Max							
		Predicted					
		1	2	3	4	5	6
Actual	1	<b>42.467</b>	277	0	0	0	0
	2	305	<b>44.787</b>	191	23	0	0
	3	5	238	<b>48.190</b>	2.688	44	8
	4	0	34	2.961	<b>43.375</b>	3.074	150
	5	0	0	311	4.975	<b>89.408</b>	9.075
	6	0	0	21	861	10.825	<b>95.707</b>

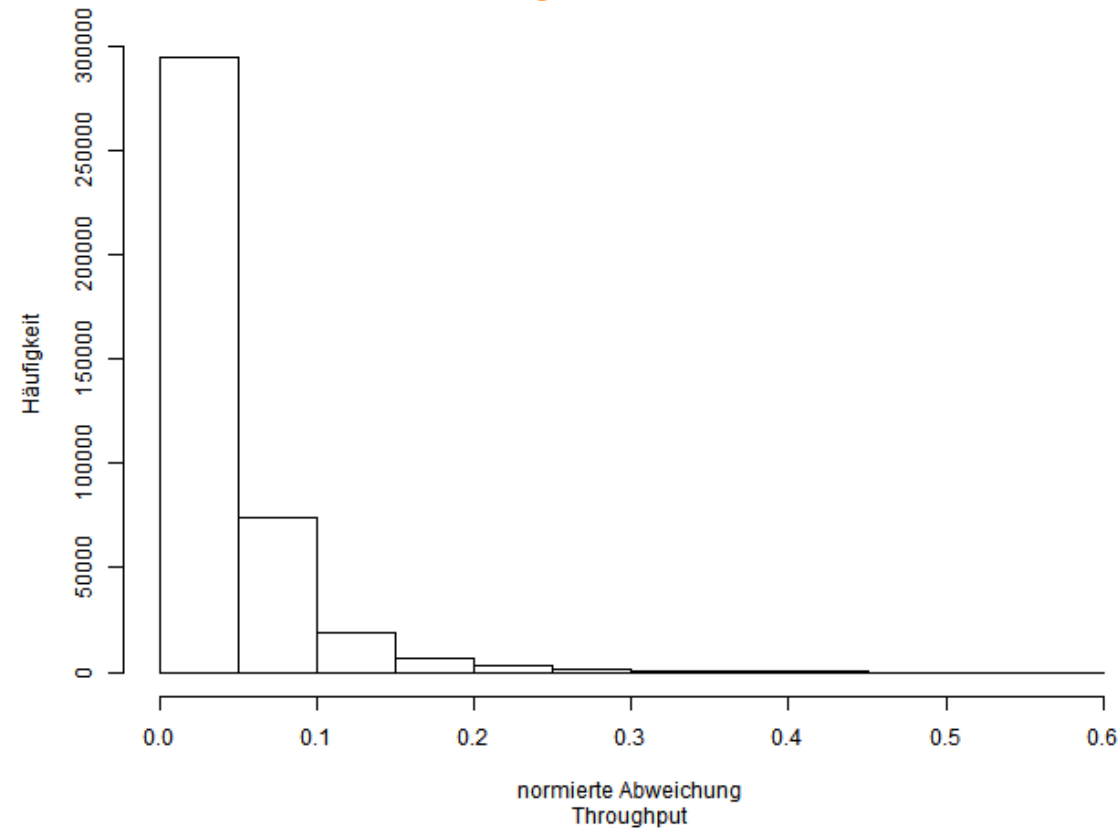
# Evaluation

## Klassifikation

Leistungskennzahl	Genauigkeit	Genauigkeit $\pm 1$ Klasse
Throughput	79,93%	99,32%
CycleTime	77,73%	99,25%
Buffer1Max	75,22%	99,13%
Buffer2Max	86,51%	99,45%
Buffer1Mean	82,65%	---
Buffer2Mean	90,98%	99,64%
Machine1Utilization	80,96%	99,09%
Machine2Utilization	86,38%	99,86%
Machine3Utilization	85,84%	99,75%
Machine4Utilization	87,83%	99,88%

# Evaluation

## Regression



**Abbildung 5: Histogramm der normierten Abweichung der Leistungskennzahl Throughput [eigene Darstellung]**

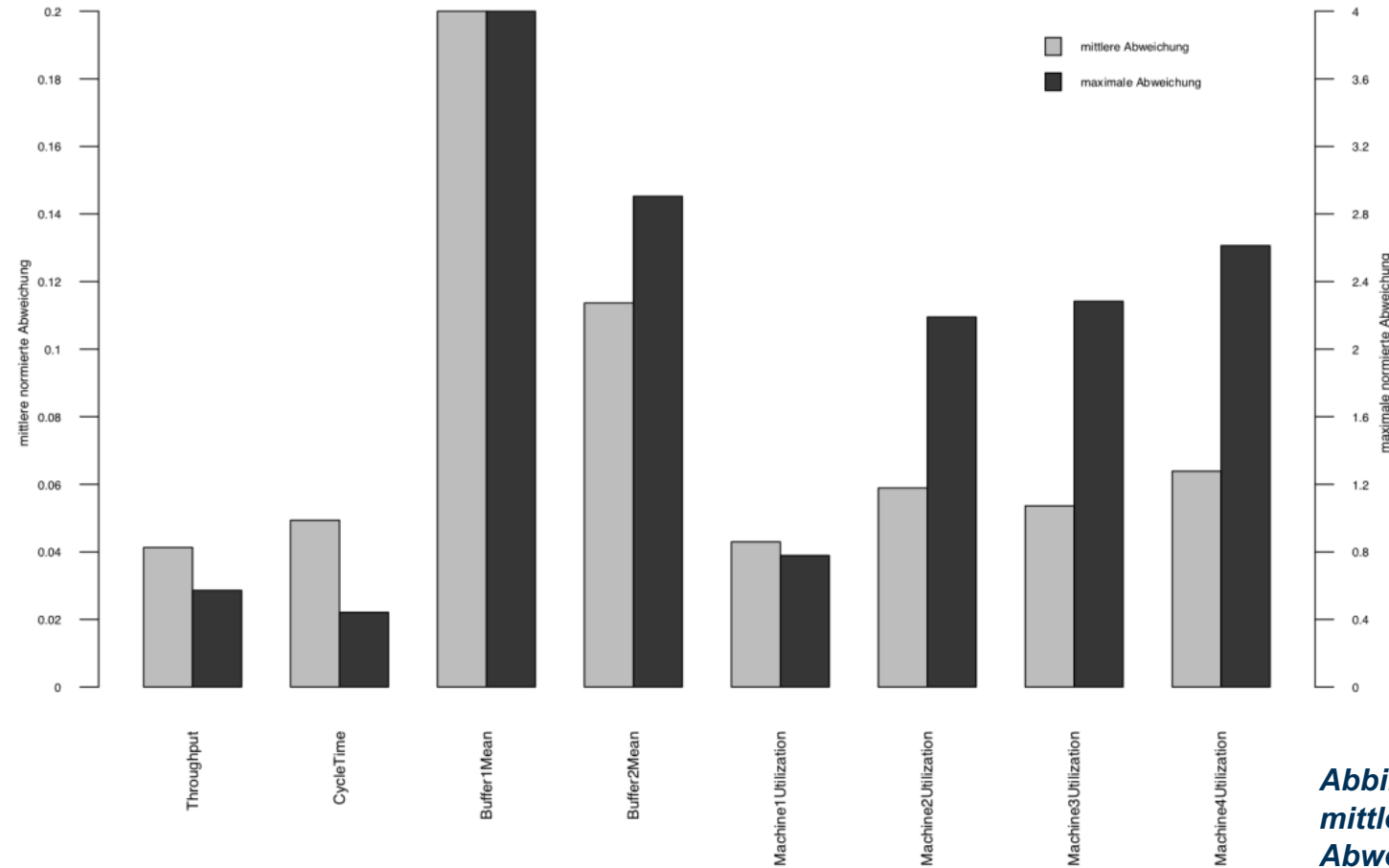
# Evaluation

## Regression

Leistungskennzahl	Mittlere Abweichung	Maximale Abweichung
Throughput	5,58%	121,32%
CycleTime	7,11%	83,78%
Buffer1Max	23,63%	596,64%
Buffer2Mean	16,54%	339,39%
Machine1Utilization	6,41%	117,80%
Machine2Utilization	9,00%	298,19%
Machine3Utilization	7,95%	313,17%
Machine4Utilization	9,66%	390,04%

# Evaluation

## Regression



**Abbildung 6: Säulendiagramm der mittleren der maximalen normierten Abweichung [eigene Darstellung]**

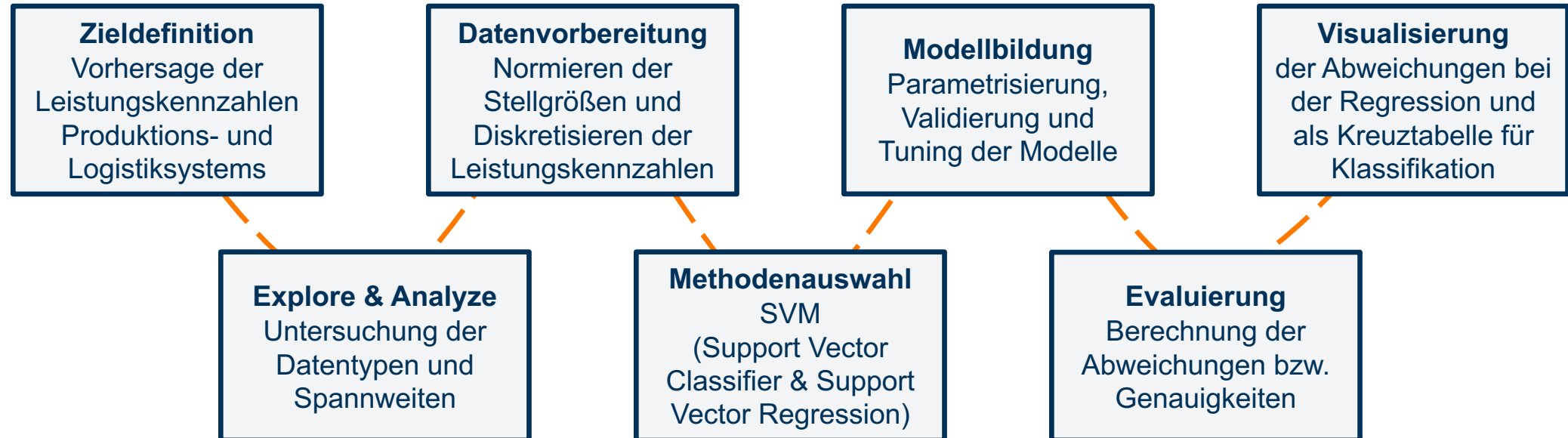


# ZUSAMMENFASSUNG

Zieldefinition | Explore & Analyze | Datenvorbereitung | Methodenauswahl |  
Modellbildung | Evaluierung | Visualisierung

# Machine Learning

**Ziel: Erstellung deskriptiver und prädiktiver Modelle auf Basis eines Datensatzes**



# Kritische Würdigung

- Die Vorverarbeitung ist ein essenzieller Teil des Prozesses
- Klassifikation mit fünf Klassen ist zuverlässiger als die Regression
- Die Parametrierung in Zusammenhang mit dem passenden Kernel muss geschickt ausgewählt werden
- Die Anpassung kann auf einem sehr komplexen Niveau weitergeführt werden
- Bei mehrdimensionalen Anwendungen ist die Visualisierung schwer
- Berechnung kann sehr lange dauern

# EVALUATION MIT TESTDATENSATZ

Anwendung | Auswertung

# Evaluation mit Testdatensatz

- Detaillierte Beschreibung in *liesmich.txt*
- Schritte
  - Einlesen der Daten
  - Vorbereitung der Daten
  - Anwendung der Modelle
  - Berechnung und Ausgabe der Abweichung

# Literatur I

- [ABR1964] Aizerman, M. A.; Braverman, È. M.; Rozonèr, L. I.: *Theoretical foundation of potential functions method in pattern recognition*. In: Avtomat. I Telemekh. Vol. 25, Nr. 6, 1964, S. 917-936
- [Al2003] Al-Laham, A.: *Organisationales Wissensmanagement – eine strategische Perspektive*. 1. Aufl., Vahlen Verlag, München, 2003
- [Al2010] Alpaydin, C.: *Introduction to Machine Learning*. 2. Aufl., MIT Press, Cambridge, MA, 2010
- [Bi2008] Bishop, C.: *Pattern Recognition and Machine Learning*. Information Science and Statistics, Springer, Berlin, 2008
- [BV2008] Bankhofer, U.; Vogel, J.: *Datenanalyse und Statistik*. 1. Aufl., Gabler Verlag, Wiesbaden, 2008
- [CC2011] Chang, C.-C.; Lin, C.-J.: *LIBSVM: a library for support vector machines*. In: ACM Transactions on Intelligent Systems and Technology. Vol. 2, Nr. 3, 2011, S. 27:1-27:27

# Literatur II

- [FPS1996] Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P.: *From Data Mining to Knowledge Discovery in*. In: AI Magazine. Nr. 17, 1996, S. 37-54.
- [Fi1936] Fisher, R. A.: *The use of multiple measurements in taxonomic problems*. In: Annals of Eugenics. Nr. 7, 1936, S. 179-188.
- [Ha2009] Hamel, L.: *Knowledge Discovery with Support Vector Machines*. John Wiley & Sons, New Jersey, 2009
- [HK2006] Han, J.; Kamber, M.: *Data Mining Concepts and Techniques*. 2. Aufl., Elsevier, Morgan Kaufmann, Amsterdam, Boston, San Francisco, CA, 2006
- [Ja+2013] James, G.; Witten, D.; Hastie, T.; Tibshirani, R.: *An Introduction to Statistical Learning with Applications in R*, 1. Aufl., Springer, New York, Heidelberg, Dordrecht, London, 2013
- [Kü1999] Küppers, B.: *Data Mining in der Praxis, ein Ansatz zur Nutzung der Potenziale von Data Mining im betrieblichen Umfeld*. 1. Aufl., Lang, Frankfurt am Main, 1999

# Literatur III

- [Me2017] Meyer, D.: *Support Vector Machines \* The Interface to libsvm in package e1071*. FH Technikum, Wien, 2017
- [Ro1958] Rosenblatt, F.: *The perceptron: A probabilistic model for information storage and organization in the brain*. In: Psychological Review. Vol. 65, Nr. 6, 1958 S. 386-408.
- [VC1995] Vapnik, V. N.; Cortes, C: *Support-Vector Networks*. In: Machine Learning. Nr. 20, 1995, S. 273-297
- [VL1963] Vapnik, V. N.; Lerner, A. Y.: *Pattern Recognition using generalized portraits*. In: Automation and Remote Control. Vol. 24, Nr. 6, 1963, S. 709-715
- [Va1995] Vapnik, V. N.: *The Nature of Statistical Learning Theory*. 2. Aufl., Springer, Berlin, 1995



# Vielen Dank für Ihre Aufmerksamkeit!

Grundlagen | SVM | Werkzeuge | Datensatz |  
Anwendung | Kritische Würdigung | Evaluation