

Featurebasierte Fehlervorhersage mittels Methoden des Machine Learnings

Einführungspräsentation zur Masterthesis

Wintersemester 2019 / 2020

23. Oktober 2019

Stefan Hermann Strüder

1

Einleitung und Motivation

- Softwarefehler als erhebliche Auslöser für finanzielle Schäden und Rufschädigungen

Fehler bei Amazon: Hunderte Waren für einen Penny verkauft

Wegen einer Softwarepanne wurden am Freitag in Großbritannien Hunderte Produkte im Amazon Marketplace für 0,01 Pfund verkauft. Für die Anbieter bedeutet das immense Verluste, einige befürchten ihren Ruin. (Quelle: heise.de, [1])

Technik-Panne

Server-Ausfall legt bundesweit Telekom-Leitungen lahm

(Quelle: tagesspiegel.de, [3])

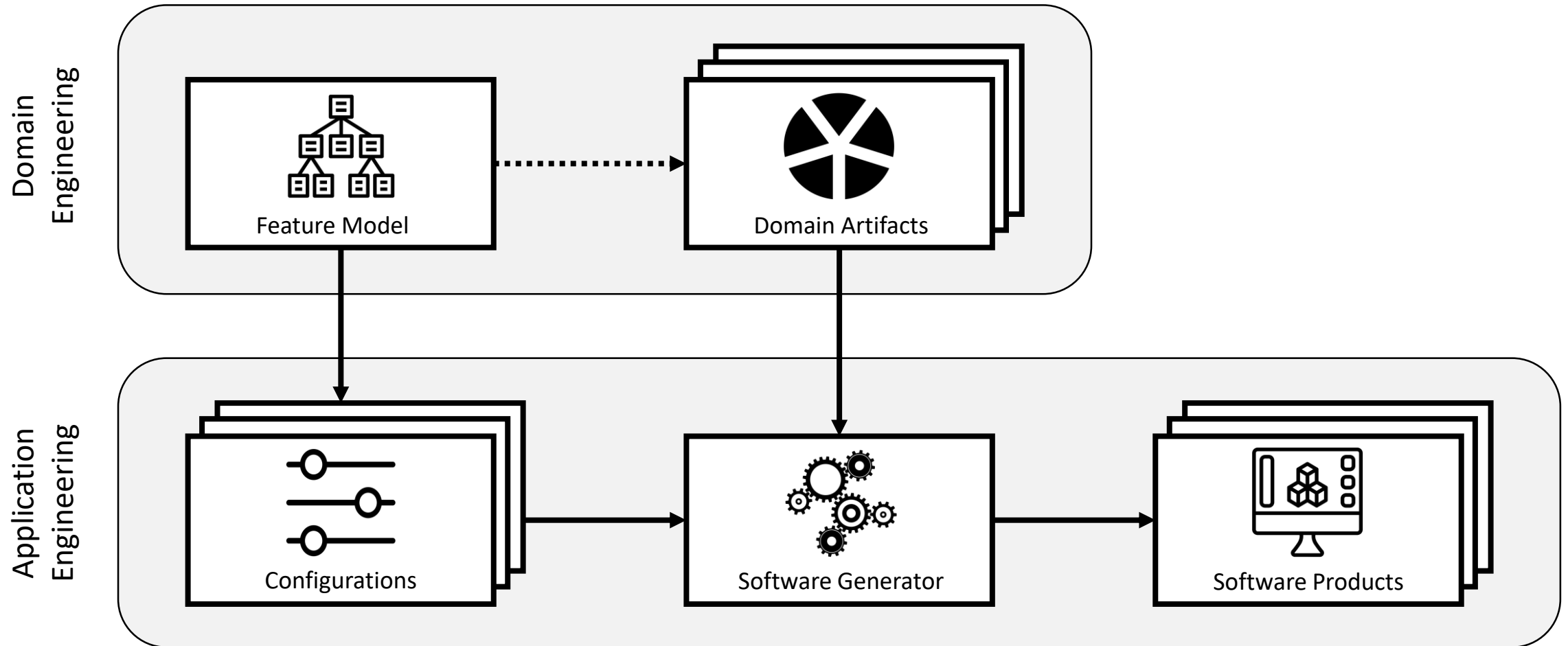
Aktienhandels-Software verzockt 440 Millionen Dollar in 45 Minuten

(Quelle: heise.de, [2])

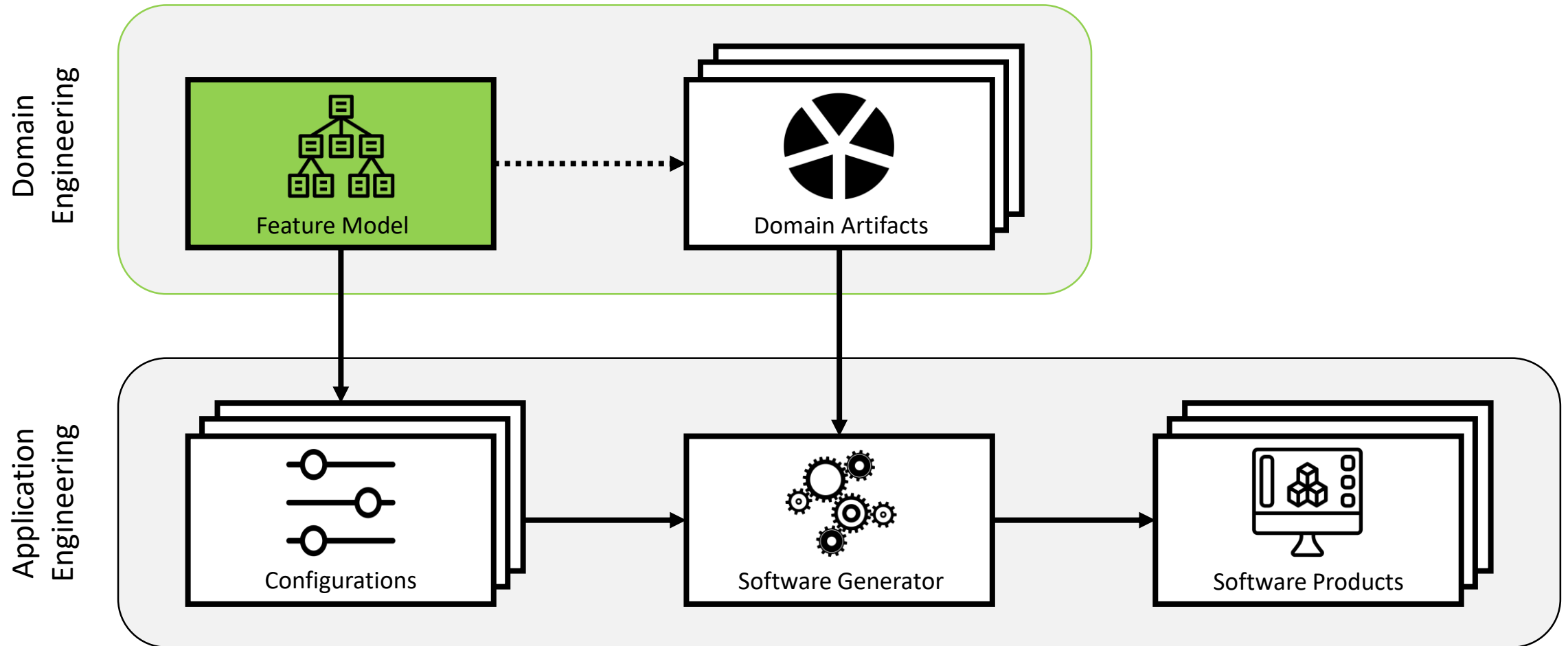
- großes Interesse an Tools zur Erkennung von fehlerhaften Code

- Entwicklung von Techniken zur Fehlererkennung und -vorhersage
 - meist basierend auf Methoden des Machine Learnings
 - Erstellung eines Datensets zur Anlernung von Klassifikatoren
 - Datengrundlage: fehlerfreie und fehlerhafte historische Daten
- große Auswahl an Lernverfahren verfügbar (Son et al., 2019; Challagulla, Bastani, Yen, & Paul, 2008)
 - Entscheidungsbaum-basiert
 - bayessche Verfahren
 - Regression, k-Nearest-Neighbor, künstliche neuronale Netz

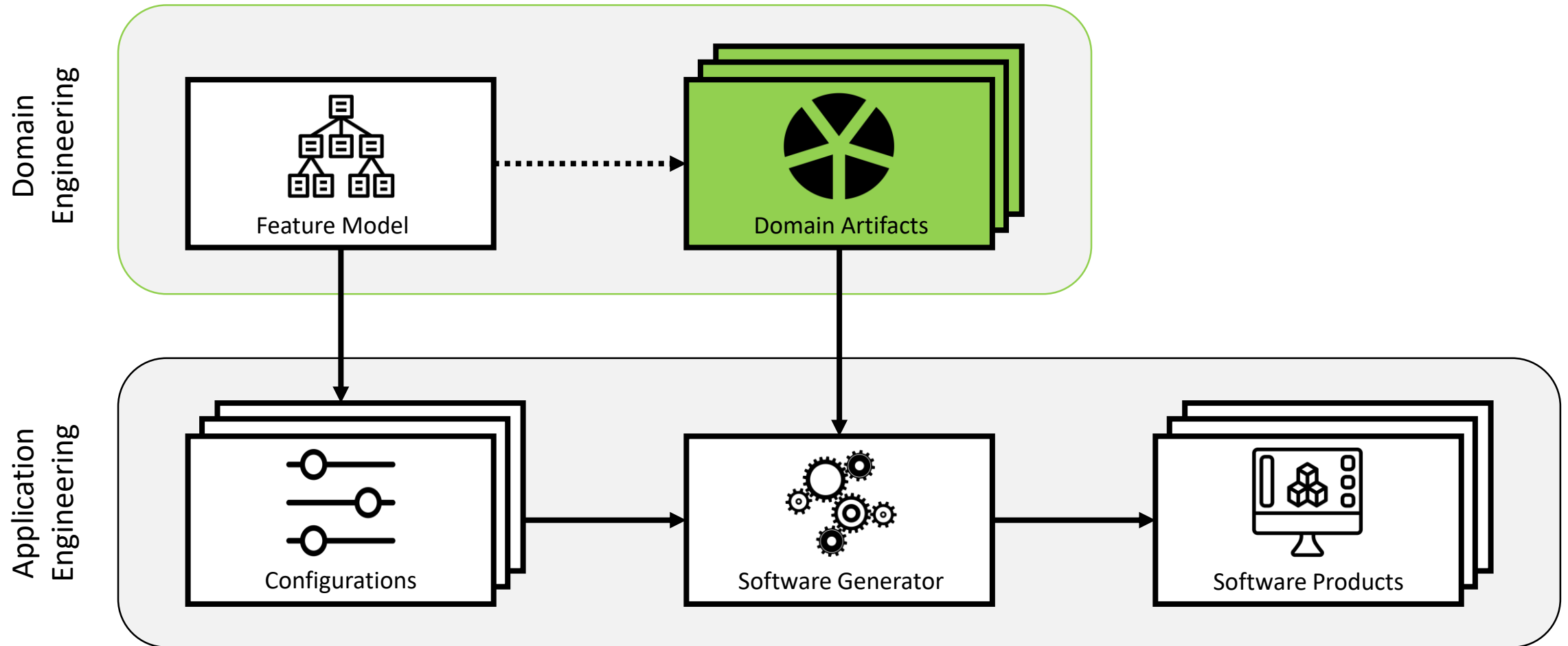
- Software-Features (Apel, Batory, Kästner, & Saake, 2013)
 - funktionale Inkremente eines Softwaresystems
- Software-Produktlinien (Thüm, Apel, Kästner, Schaefer, & Saake, 2014)
 - Menge von ähnlichen Softwareprodukten
 - gemeinsame Menge von Features
 - gemeinsame Codebasis
 - ermöglicht breite Variabilität innerhalb einer Produktlinie



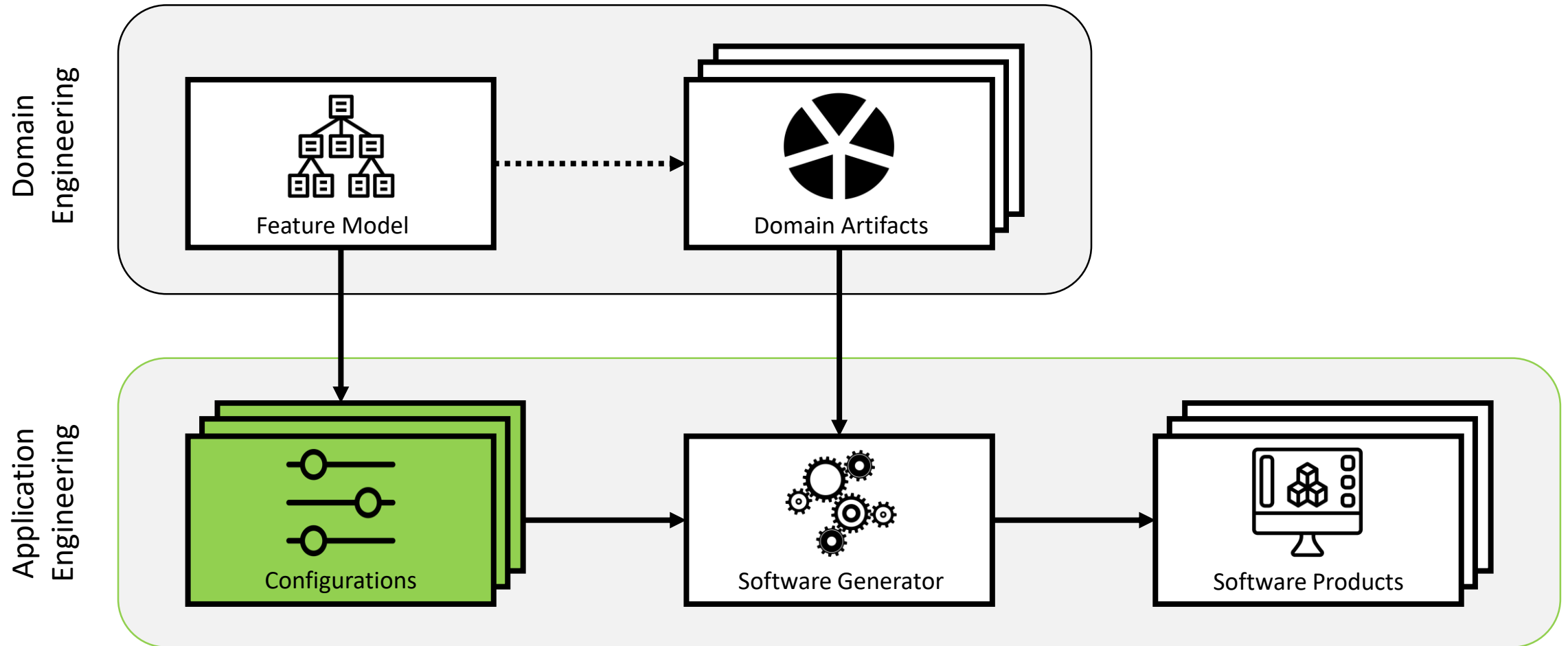
(nach Thüm et al., 2014; Quellen der Icons siehe Quellenübersicht)



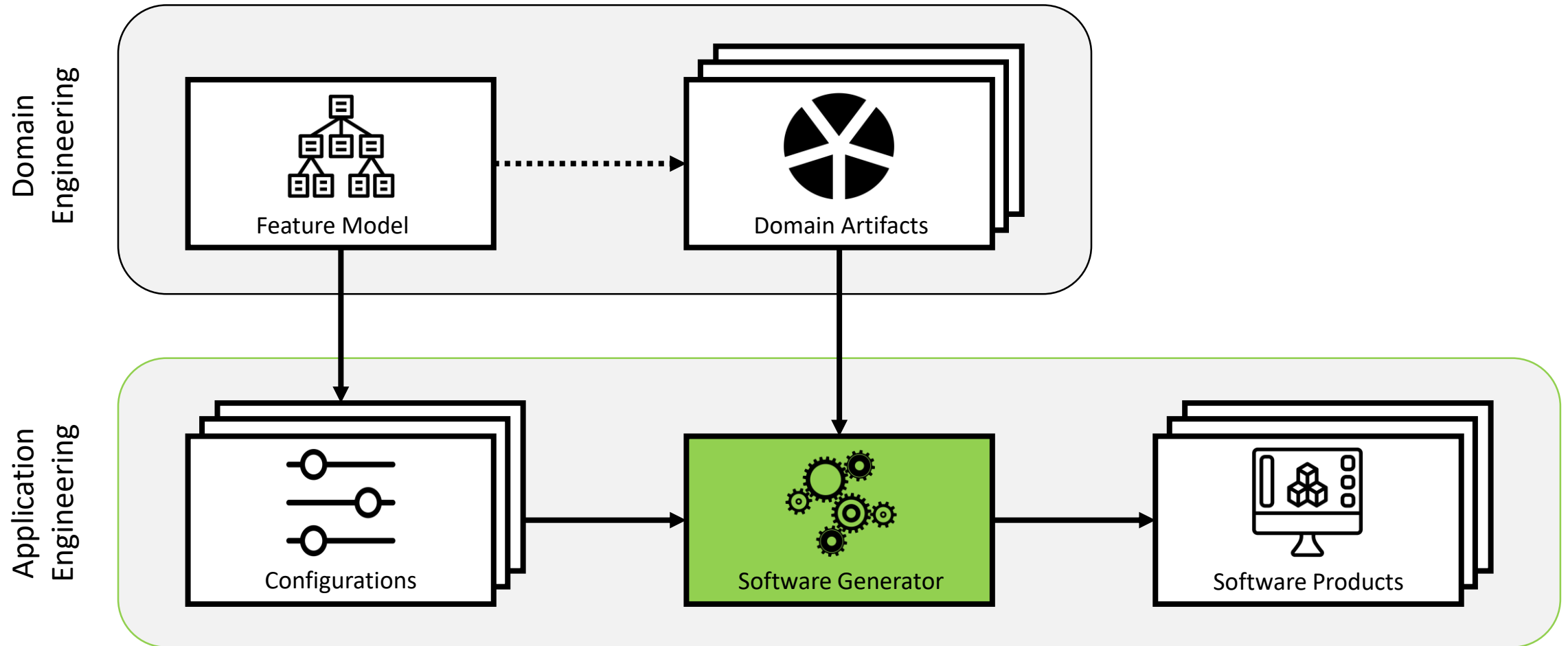
(nach Thüm et al., 2014; Quellen der Icons siehe Quellenübersicht)



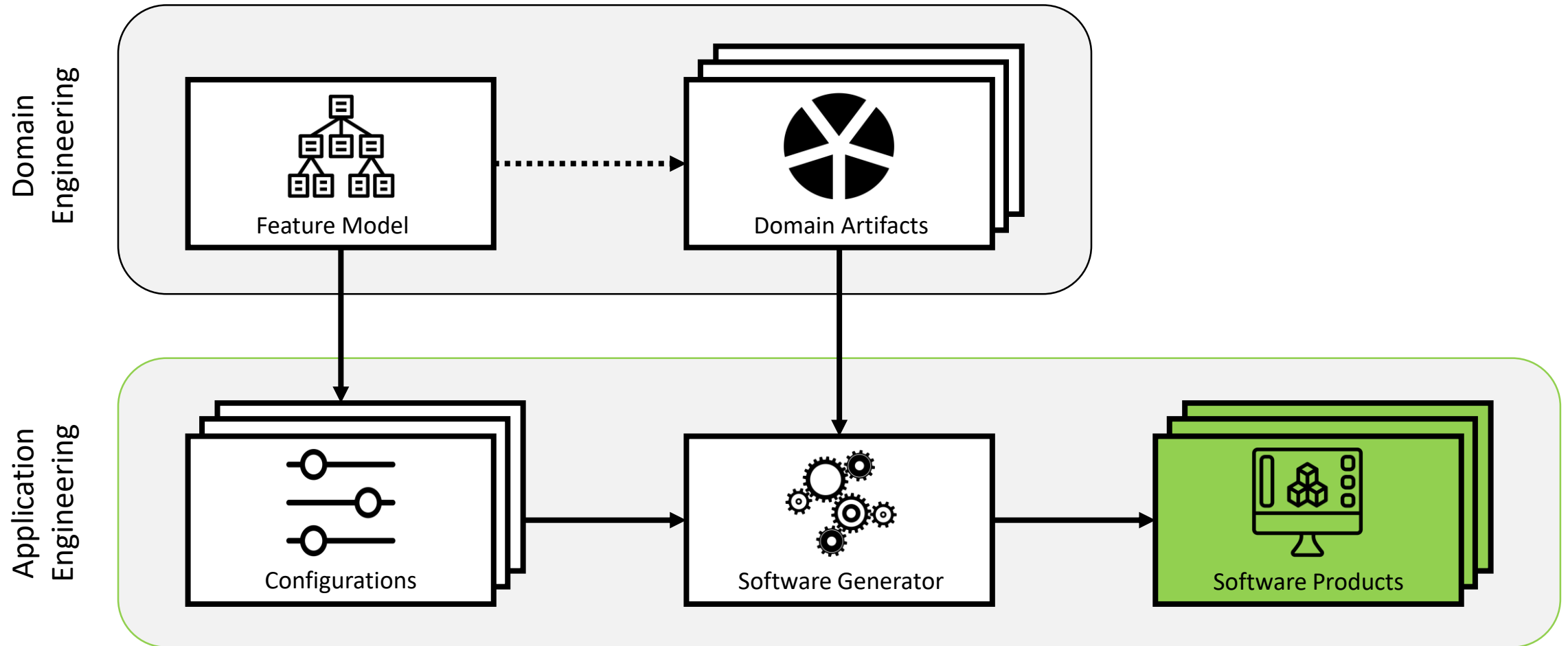
(nach Thüm et al., 2014; Quellen der Icons siehe Quellenübersicht)



(nach Thüm et al., 2014; Quellen der Icons siehe Quellenübersicht)



(nach Thüm et al., 2014; Quellen der Icons siehe Quellenübersicht)



(nach Thüm et al., 2014; Quellen der Icons siehe Quellenübersicht)

- Ziel der Thesis
 - Vorhersagetechnik für Softwarefehler
 - unter Verwendung von Software-Features
 - basierend auf Methoden des Machine Learnings
- chancenreicher Ansatz
 - Fehlervorhersage basierend auf „Vergangenheit“ der Software
 - Eigenschaften von fehleranfälligen Features
 - Fehleranfälligkeit von Features

1. Einleitung und Motivation
2. Hintergrund
 - I. Machine-Learning-Klassifikation
 - II. Fehlervorhersage mittels Machine Learning
3. Ziele und Vorgehensweise
 - I. Zielsetzung
 - II. Methodik
 - III. Arbeitsplanung
4. Verwandte Literatur

2

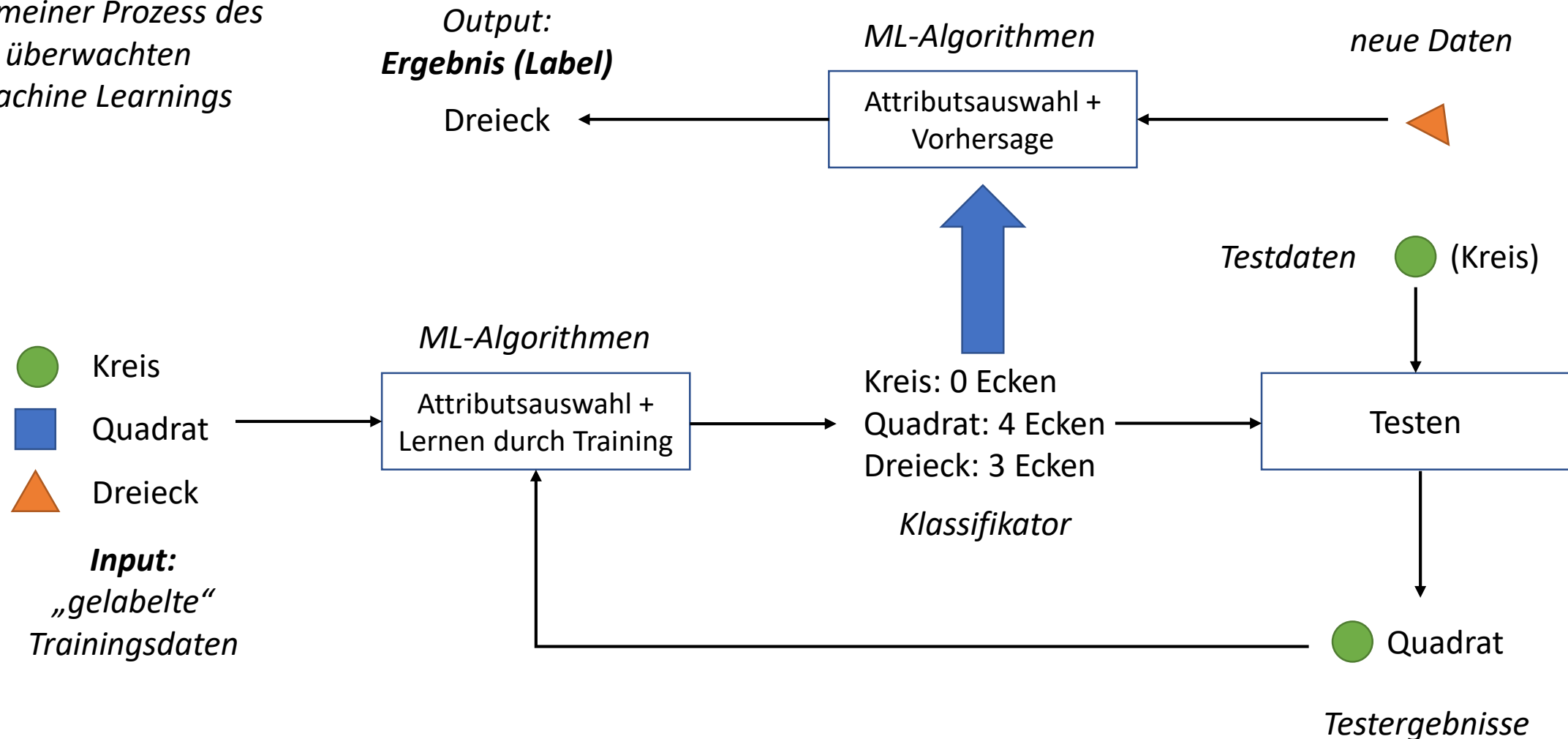
Hintergrund

Machine-Learning-Klassifikation

Fehlervorsage mittels Machine Learning

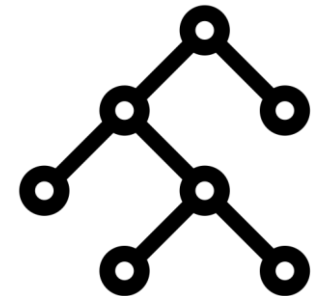
- supervised Machine Learning (überwachtes ML) (Hammouri, Hammad, Alnabhan, & Alsarayrah, 2018)
 - Entwicklung einer Ableitungsfunktion
 - Schlussfolgern aus Ein- und Ausgabedaten innerhalb eines Trainings-Datensets
 - Vorhersage für neue Eingangsdaten
- gängige Algorithmen
 - Naïve Bayes
 - Entscheidungsbäume
 - künstliche neuronale Netze

allgemeiner Prozess des
überwachten
Machine Learnings



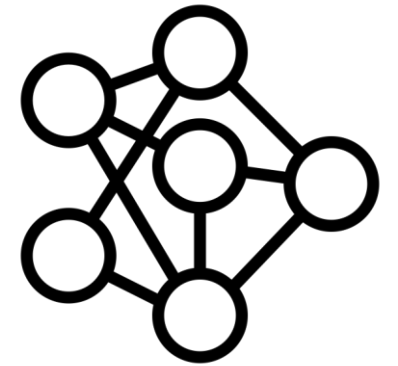
- Naïve Bayes (Hammouri, Hammad, Alnabhan, & Alsarayrah, 2018)
 - probabilistischer Klassifikator
 - basierend auf Bayes Theorem
 - Unabhängigkeit zwischen Attributen
- Entscheidungsbäume (decision trees) (Hammouri, Hammad, Alnabhan, & Alsarayrah, 2018)
 - hierarchisch und prädikativ
 - Attribute der Daten als Verzweigungen
 - Entscheidung als Blattknoten

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$



(Quelle des Icons siehe Quellenübersicht)

- künstliche neuronale Netze (Hammouri, Hammad, Alnabhan, & Alsarayrah, 2018; Jukes, 2017)
 - inspiriert an biologischen neuronalen Netzen
 - nicht-linearer Klassifikator
 - bestehend aus Menge von Verarbeitungseinheiten (Neuronen)
 - parallele Ausführung zur Erarbeitung von Ausgaben
 - Signalübertragung durch Verbindungen
 - Berechnungen basierend auf Summe der Eingaben aller Neuronen



(Quelle des Icons siehe Quellenübersicht)

- Fehlererkennung mittels Machine Learning
 - Challagulla, V. U. B., Bastani, F. B., Yen, I. L., & Paul, R. A. (2008). **Empirical assessment of machine learning based software defect prediction techniques**. International Journal on Artificial Intelligence Tools, 17(2), 389–400. <https://doi.org/10.1142/S0218213008003947>
 - Son, L. H., Pritam, N., Khari, M., Kumar, R., Phuong, P. T. M., & Thong, P. H. (2019). **Empirical study of software defect prediction: A systematic mapping**. Symmetry, 11(2). <https://doi.org/10.3390/sym11020212>
- featurebasierte Fehlererkennung mittels Machine Learning
 - Queiroz, R., Berger, T., & Czarnecki, K. (2016). **Towards predicting feature defects in software product lines**. FOSD 2016 - Proceedings of the 7th International Workshop on Feature-Oriented Software Development, Co-Located with SPLASH 2016, 58–62. <https://doi.org/10.1145/3001867.3001874>

featurebasierter Prozess des überwachten Machine Learnings nach Queiroz et. al.

„gelabeltes“ Datenset basierend auf Releases des Tools BusyBox			
Feature	Release	Metriken	Label
cat	2.1	12,4,8,9,2	fehlerhaft
cat	2.2	14,6,10,6,7	fehlerfrei
echo	2.1	5,2,4,6,4	fehlerfrei
echo	2.2	6,3,5,10,6	fehlerhaft
...
mount	2.1	8,2,4,3,5	fehlerhaft
mount	2.2	15,5,8,4,8	fehlerfrei
...

Trainingsdaten

Testdaten

Attributsauswahl +
Lernen durch Training

Attribute: Prozessmetriken

- COMM
- ADEV
- DDEV
- EXP
- OXP

ML-Algorithmen

NB

Naïve Bayes

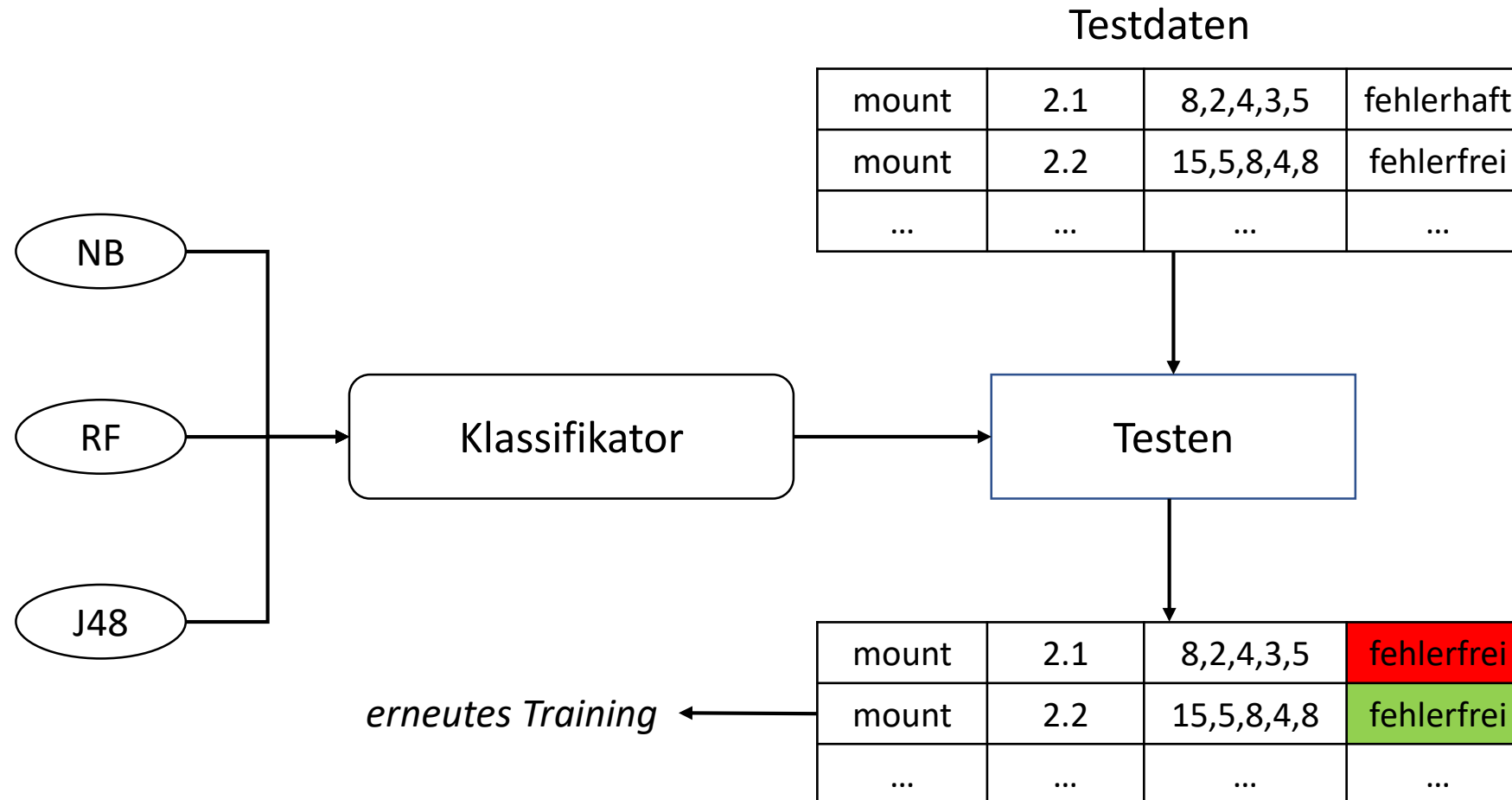
RF

Random Forest

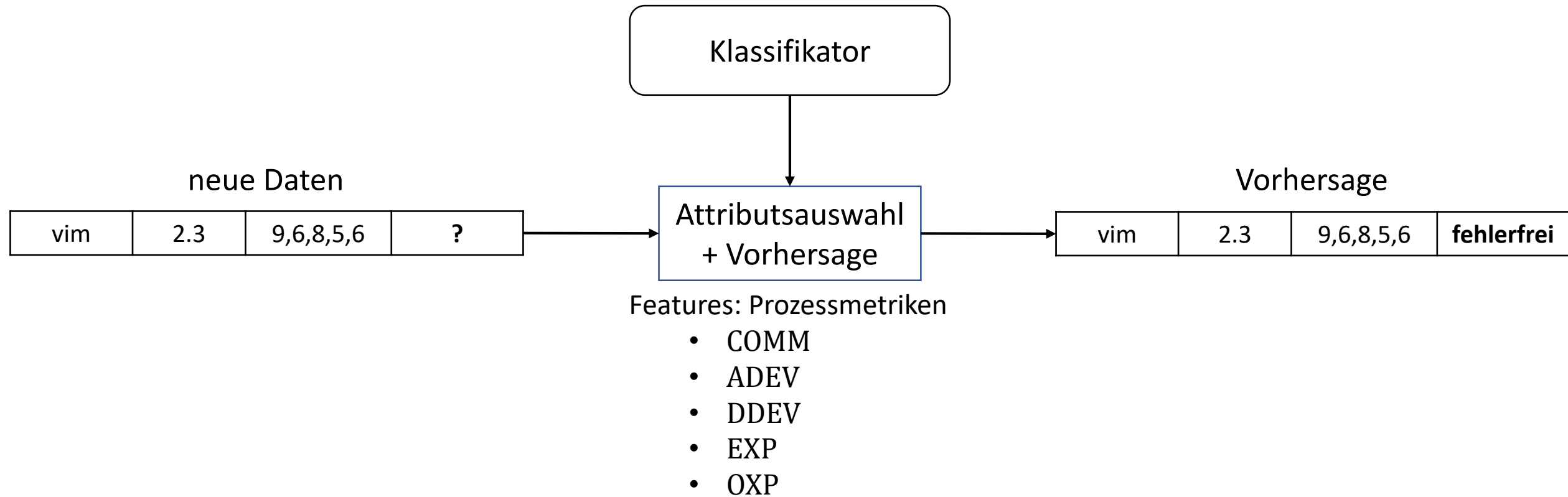
J48

Entscheidungsbaum

featurebasierter Prozess des überwachten Machine Learnings nach Queiroz et. al.



featurebasierter Prozess des überwachten Machine Learnings nach Queiroz et. al.



3

Ziele und Vorgehensweise

Zielsetzung

Methodik

Arbeitsplanung

- übergeordnetes Ziel
 - Entwicklung einer Vorhersagetechnik für Fehler in featurebasierter Software
 - unter Verwendung von Methoden des Machine Learnings
- Datengrundlage: Commits von Versionierungssystemen (Git: GitHub)
 - Commit: zur Verfügungstellung einer aktualisierten Version einer Software
 - fehlerhafte und fehlerfreie Commits zum Anlernen von Klassifikatoren
- drei zu erreichende Forschungsziele mit zugeordneten Forschungsfragen

Forschungsziel 1

Erstellung eines Datensets zum Trainieren von relevanten Machine-Learning-Klassifikatoren.

?

Welche Daten kommen für die Erstellung des Datensets in Frage?

?

Wie weit müssen die Daten vorverarbeitet werden, um sie für das Training nutzbar zu machen?

Forschungsziel 2

Identifikation und Training einer Auswahl von relevanten Machine-Learning-Klassifikatoren basierend auf dem Datenset.

?

Welche Machine-Learning-Klassifikatoren kommen für die gegebene Aufgabe in Frage?

Forschungsziel 3

Evaluierung und Gegenüberstellung der Klassifikatoren sowie Vergleich zu modernen Vorhersagetechniken, die keine Features nutzen.

?

Welche miteinander vergleichbaren Merkmale besitzen die Klassifikatoren?

?

Welche Messwerte können für den Vergleich verwendet werden?

?

Welche Vor- und Nachteile besitzt ein Klassifikator?

?

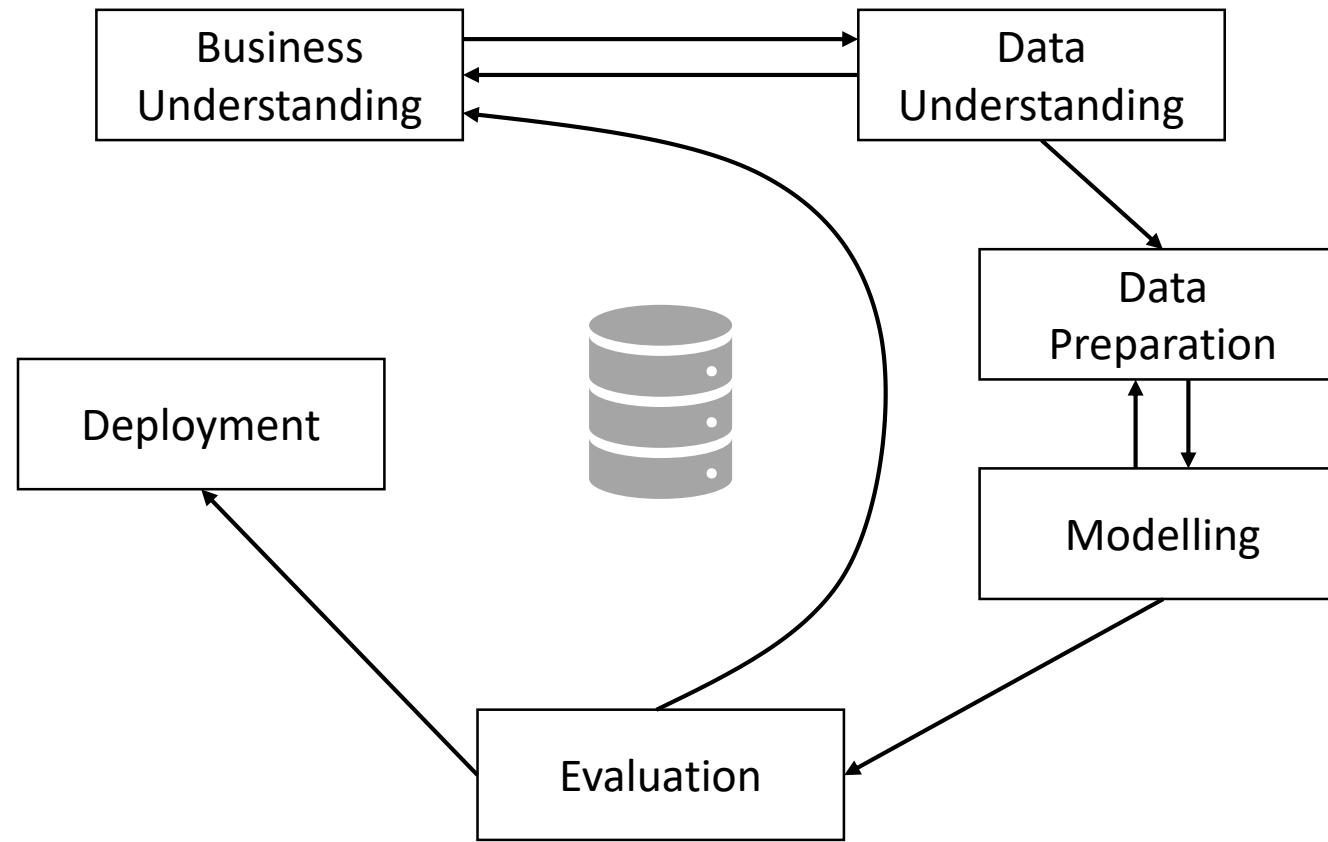
Wie lassen sich die Klassifikatoren mit weiteren Vorhersagetechniken, die keine Features nutzen, vergleichen?

Arbeitsphasen

Es ergeben sich fünf Arbeitsphasen:

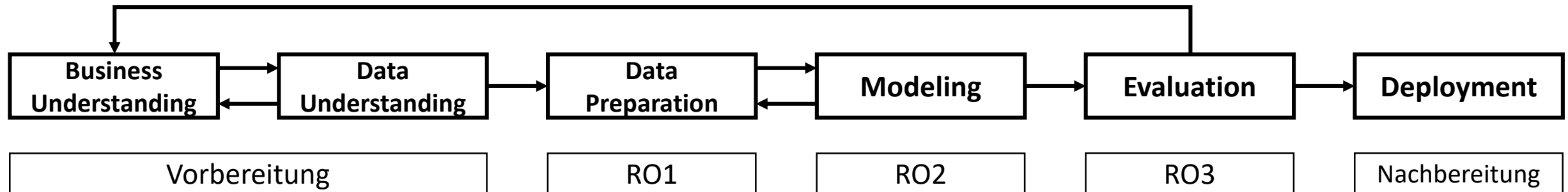
1. Vorbereitung
2. Erarbeitung Forschungsziel 1: *Erstellung des Datensets*
3. Erarbeitung Forschungsziel 2: *Training der ML-Klassifikatoren*
4. Erarbeitung Forschungsziel 3: *Evaluation der Klassifikatoren*
5. Nachbereitung

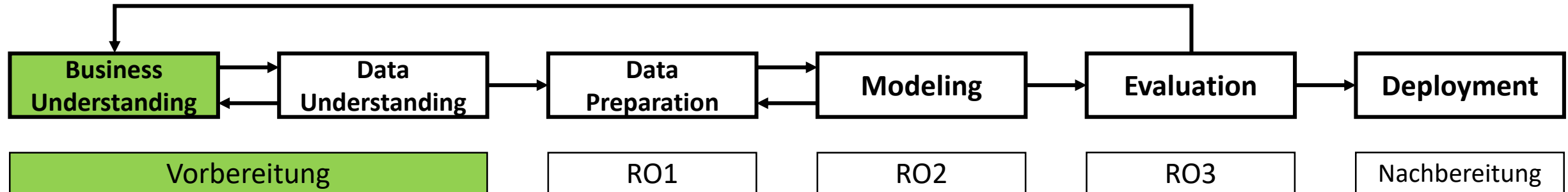
Cross-Industry Standard Process for Data Mining (CRISP-DM) Prozessmodell



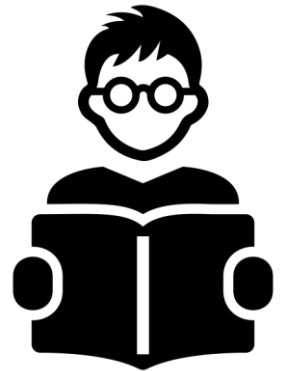
(Chapman et al., 2000)

Zuordnung der Arbeitsphasen

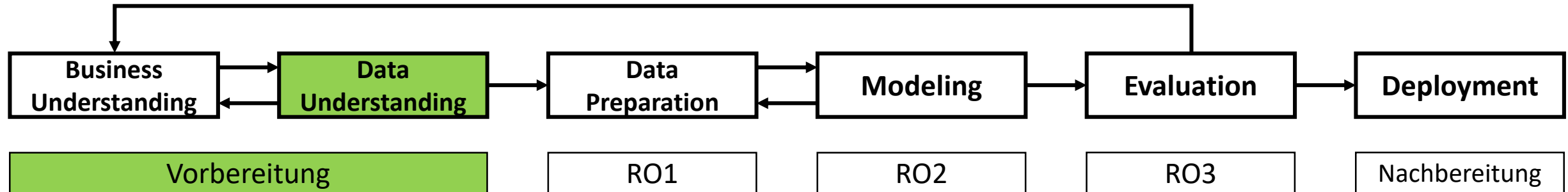




- Business Understanding (Vorbereitung)
 - allgemeine Einarbeitung in das Thema
 - Formulierung der Forschungsziele



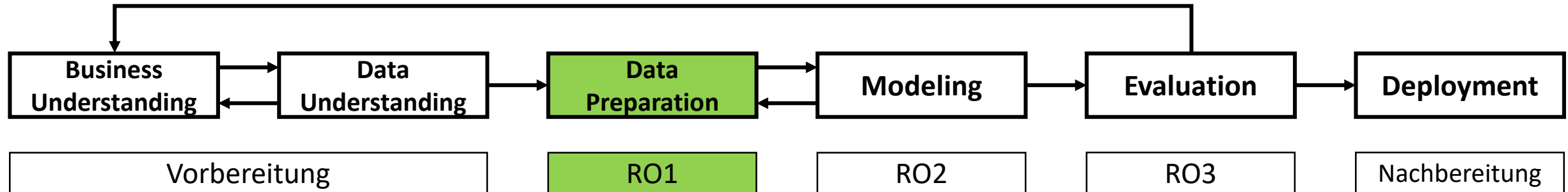
(Quelle des Icons siehe Quellenübersicht)



- Data Understanding (Vorbereitung)
 - Suche + Einsicht von relevanten Daten und vorgefertigten Datensets
 - Schwerpunkt der Suche: GitHub



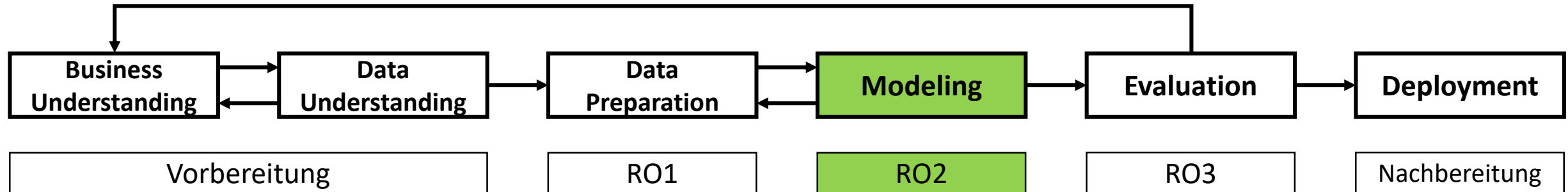
(Quelle des Icons siehe Quellenübersicht)



- Data Preparation (Forschungsziel 1)
 - Prozesse zur Optimierung des Datensets
 - Erstellung des endgültigen Datensets



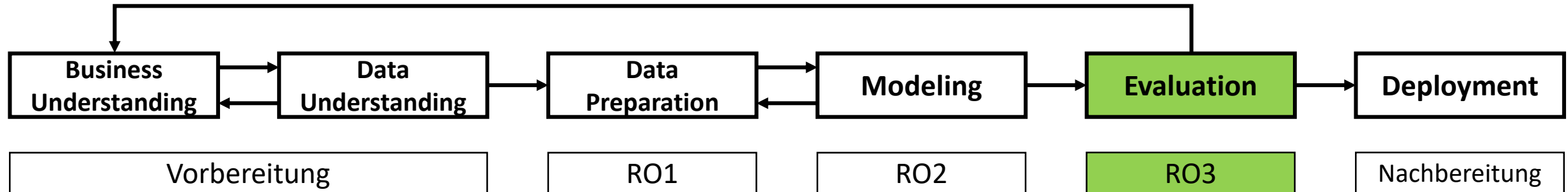
(Quelle des Icons siehe Quellenübersicht)



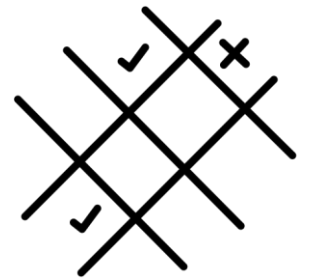
- Modeling (Forschungsziel 2)
 - Anwendung des erstellten Datensets
 - Anlernen der Machine Learning Algorithmen



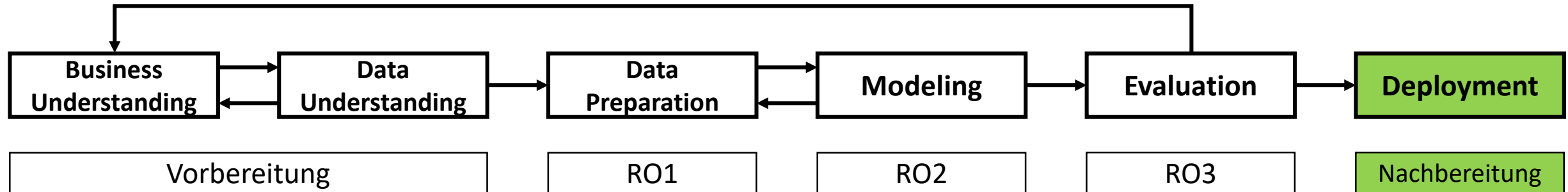
(Quelle des Icons siehe Quellenübersicht)



- Evaluation (Forschungsziel 3)
 - Evaluation der Machine Learning Algorithmen
 - Vergleich der Algorithmen



(Quelle des Icons siehe Quellenübersicht)

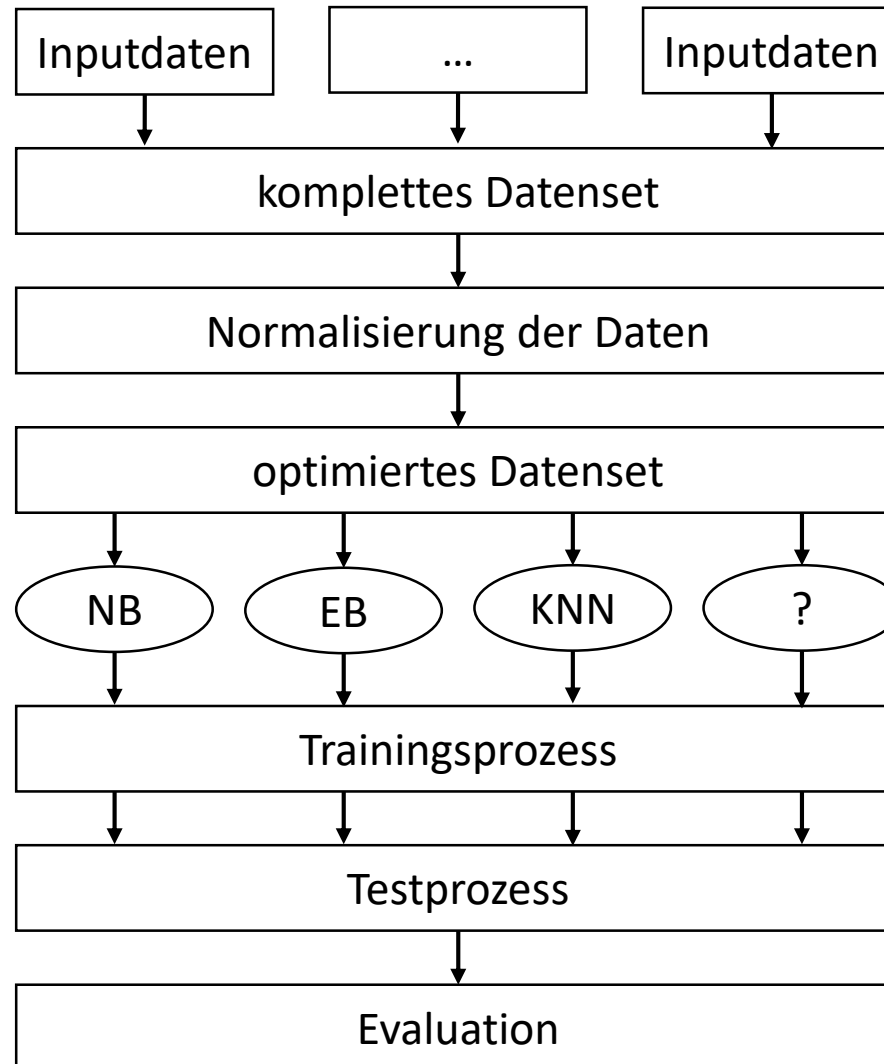


- Deployment (Nachbereitung)
 - Erstellung der Ausarbeitung und Abschlusspräsentation
 - Abhalten des Kolloquiums



(Quelle des Icons siehe Quellenübersicht)

*zur Anwendung kommendes
Machine-Learning-
Verfahren*



Commits einer featurebasierten Software

bestehend aus Trainings- und Testdaten

Preprocessing, Errechnung von Metriken

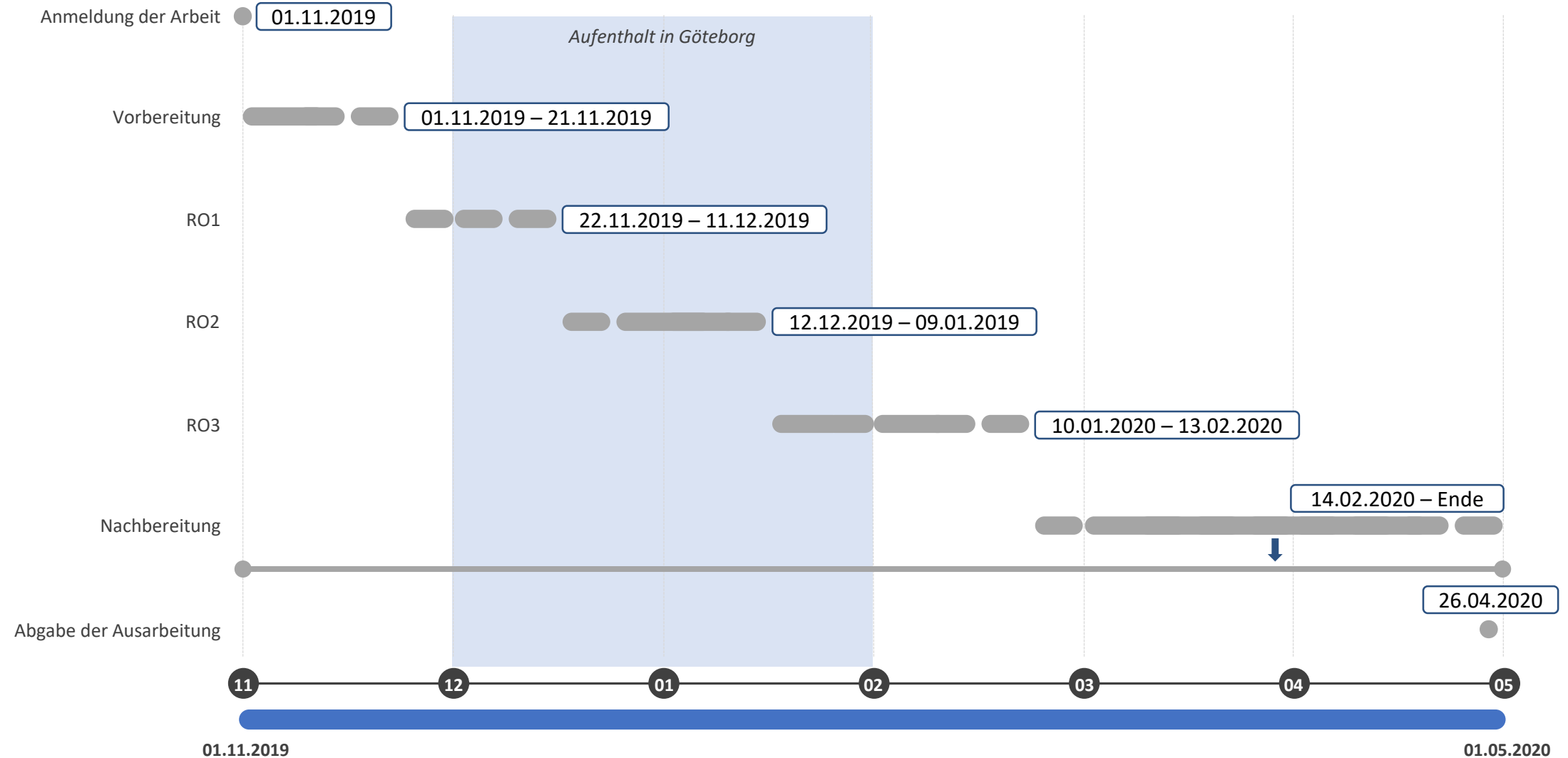
Prozessmetriken als Attribute

Training der Klassifikatoren

Anwendung der Testdaten

Auswahl des performantesten Klassifikators

(nach Ceylan, Kutlubay, & Bener, 2006)



Phase 1: Vorbereitung

Unterziele	Dauer	
<u>Strukturierte Literaturrecherche</u> <ul style="list-style-type: none">• Techniken der featurebasierten Softwareprogrammierung• Techniken zur Fehlererkennung in Software• Klassifikation mittels Machine Learning• Klassifikationsmethoden• Auswahl der Programmiersprache• Tool- & Libraryauswahl• Evaluationsmethoden	2 Wochen	Business Understanding
<u>Recherche zur Bildung eines Datensets</u> <ul style="list-style-type: none">• Merkmale / Aufbau eines Datensets• Suche nach Datenquellen• Suche nach vorgefertigten Datensets• Prüfung der Daten / Datensets auf Eignung• Analyse des Aufbaus der Daten / Datensets	1 Woche	Data Understanding
Total:	3 Wochen	

Phase 2: Forschungsziel 1

Unterziele	Dauer
<u>Finale Datenauswahl</u> <ul style="list-style-type: none">• Festlegung von Kriterien	1 Woche
<u>Datenbereinigung</u> <ul style="list-style-type: none">• „Preprocessing“	1 Woche (G)
<u>Finale Konstruktion des Datensets</u> <ul style="list-style-type: none">• Integration der Daten und des Feature-Aspekts (Metriken)• erneute abschließende Bereinigung sowie Formatierung• Teilung in Training-Set und Test-Set	1 Woche (G)
Total:	3 Wochen

(G) Bearbeitung während des Aufenthalts in Göteborg

Phase 3: Forschungsziel 2

Unterziele	Dauer
Auswahl geeigneter Klassifikatoren	1 Woche (G)
Training der Klassifikatoren	3 Wochen (G)
Total:	4 Wochen

(G) Bearbeitung während des Aufenthalts in Göteborg

Phase 4: Forschungsziel 3

Unterziele	Dauer
<u>Evaluation der einzelnen Klassifikatoren</u> <ul style="list-style-type: none">• Anwendung des Test-Sets• Festlegung der Messwerte• Berechnung der Messwerte	2 Wochen (G)
Vergleich der Klassifikatoren anhand der Messwerte	2 Wochen
Vergleich mit weiteren Vorhersagetechniken, die nicht auf Features setzen	1 Woche
Total:	5 Wochen

(G) Bearbeitung während des Aufenthalts in Göteborg

Phase 5: Nachbereitung

Unterziele	Dauer
<u>Besprechung der vorangegangenen Arbeit mit Betreuer</u> <ul style="list-style-type: none">• Umsetzung möglicher Verbesserungsvorschläge	1 Woche
Erstellung der Ausarbeitung (kontinuierlicher Prozess)	7 Wochen
Erstellung der Abschlusspräsentation	1 Woche
Total:	9 Wochen

4

Verwandte Literatur

■ Einführende Literatur zu Software-Produktlinien

- Apel, S., Batory, D., Kästner, C., & Saake, G. (2013). **Feature-Oriented Software Product Lines**.
<https://doi.org/10.1007/978-3-642-37521-7>
- Pohl, K., Böckle, G., & van der Linden, F. (2005). **Software Product Line Engineering**.
<https://doi.org/10.1007/3-540-28901-1>

■ Einführende Literatur zu featureorientierter Programmierung

- Apel et. al., 2013
- Berger, T., Lettner, D., Rubin, J., Grünbacher, P., Silva, A., Becker, M., ... Czarnecki, K. (2015). **What is a feature?** Proceedings of the 19th International Conference on Software Product Line - SPLC '15, 3(1), 16–25. <https://doi.org/10.1145/2791060.2791108>

■ Beispielhafte Literatur zu nicht-featurebasierter Fehlererkennung

- Zimmermann, T., Premraj, R., & Zeller, A. (2007). **Predicting defects for eclipse**. Proceedings - ICSE 2007 Workshops: Third International Workshop on Predictor Models in Software Engineering, PROMISE'07. <https://doi.org/10.1109/PROMISE.2007.10>

■ Einführende Literatur zu Metriken

- Rahman, F., & Devanbu, P. (2013). **How, and why, process metrics are better**. Proceedings - International Conference on Software Engineering, 432–441. <https://doi.org/10.1109/ICSE.2013.6606589>



Danke für Ihre Aufmerksamkeit.
Zeit für Fragen.

Literaturübersicht

- Apel, S., Batory, D., Kästner, C., & Saake, G. (2013). Feature-Oriented Software Product Lines. <https://doi.org/10.1007/978-3-642-37521-7>
- Challagulla, V. U. B., Bastani, F. B., Yen, I. L., & Paul, R. A. (2008). Empirical assessment of machine learning based software defect prediction techniques. *International Journal on Artificial Intelligence Tools*, 17(2), 389–400. <https://doi.org/10.1142/S0218213008003947>
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0. CRISP-DM Consortium, 76. <https://doi.org/10.1109/ICETET.2008.239>
- Hammouri, A., Hammad, M., Alnabhan, M., & Alsarayrah, F. (2018). Software Bug Prediction using machine learning approach. *International Journal of Advanced Computer Science and Applications*, 9(2), 78–83. <https://doi.org/10.14569/IJACSA.2018.090212>

Literaturübersicht

- Jukes, E. (2017). Encyclopedia of Machine Learning and Data Mining. In C. Sammut & G. I. Webb (Eds.), Reference Reviews (Vol. 32). <https://doi.org/10.1007/978-1-4899-7687-1>
- Son, L. H., Pritam, N., Khari, M., Kumar, R., Phuong, P. T. M., & Thong, P. H. (2019). Empirical study of software defect prediction: A systematic mapping. *Symmetry*, 11(2). <https://doi.org/10.3390/sym11020212>
- Thüm, T., Apel, S., Kästner, C., Schaefer, I., & Saake, G. (2014). A classification and survey of analysis strategies for software product lines. *ACM Computing Surveys*, 47(1). <https://doi.org/10.1145/2580950>
- Ceylan, E., Kutlubay, F. O., & Bener, A. B. (2006). Software defect identification using machine learning techniques. *Proceedings - 32nd Euromicro Conference on Software Engineering and Advanced Applications, SEAA*, 240–246. <https://doi.org/10.1109/EUROMICRO.2006.56>

Onlinequellen

[1] <https://www.heise.de/newsticker/meldung/Fehler-bei-Amazon-Hunderte-Waren-fuer-einen-Penny-verkauft-2490907.html> (heise.de, 15.12.2015)

[2] <https://www.heise.de/newsticker/meldung/Aktienhandels-Software-verzockt-440-Millionen-Dollar-in-45-Minuten-1660142.html> (heise.de, 04.08.2012)

[3] <https://www.tagesspiegel.de/wirtschaft/technik-panne-server-ausfall-legt-bundesweit-telekom-leitungen-lahm/1083126.html> (tagesspiegel.de, 30.10.2007)

[4] <https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/> (siehe Appendix)

[5] <https://rocketloop.de/kuenstliche-neuronale-netze/> (siehe Appendix)

Quellen der verwendeten Icons

- decision tree by Becris from the Noun Project
- Five Parts by Arthur Shlain from the Noun Project
- configuration by joe pictos from the Noun Project
- Gears by Lisa Oregioni from the Noun Project
- Software by Creaticca Creative Agency from the Noun Project
- Random Forest by Knut M. Synstad from the Noun Project
- Neural Network by Becris from the Noun Project

Quellen der verwendeten Icons

- Reading by Arafat Uddin from the Noun Project
- Data Analysis by Brennan Novak from the Noun Project
- Data by fizae from the Noun Project
- Machine Learning by Juicy Fish from the Noun Project
- evaluation by Michael Rojas from the Noun Project
- Writing by Kmg Design from the Noun Project

A

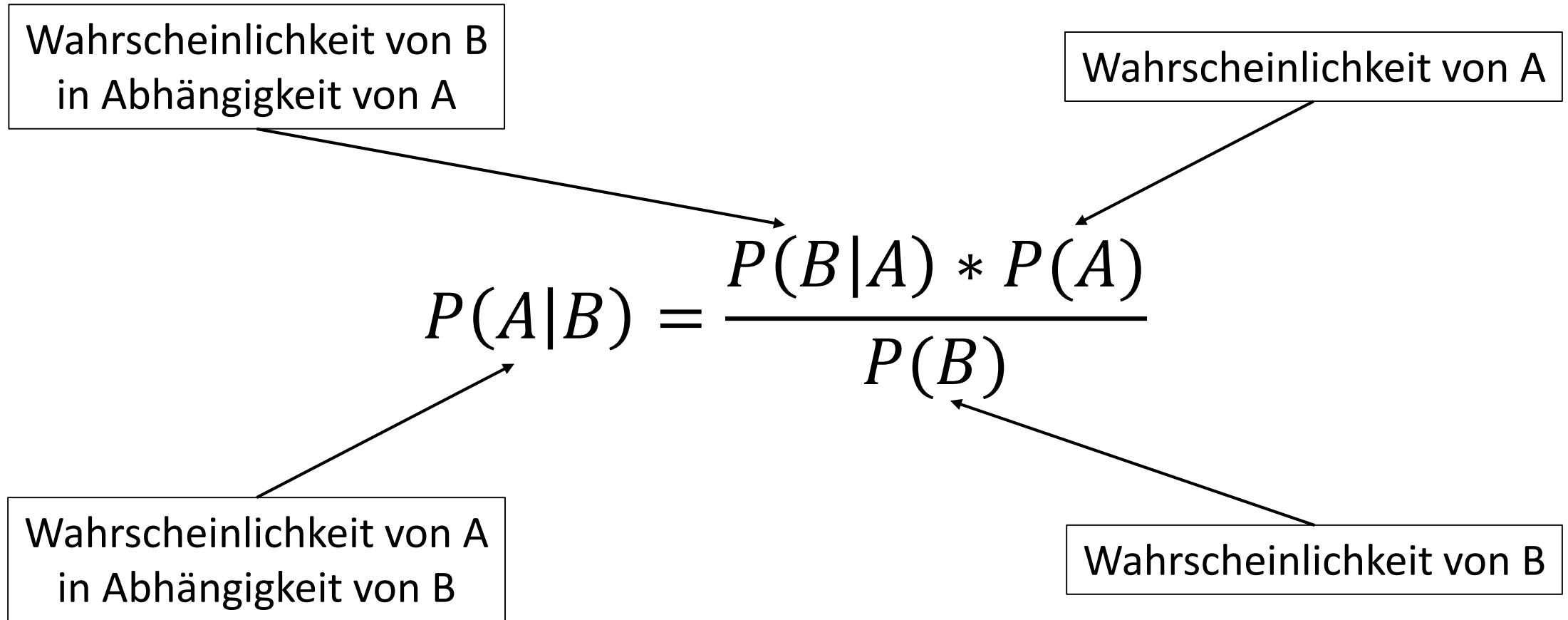
Appendix

Erläuterung Bayes-Theorem + Beispiel

Beispielhafter Entscheidungsbaum

Beispielhafter Aufbau eines künstlichen neuronalen Netzes

Evaluation der Klassifikatoren



Wahrscheinlichkeit von B
in Abhängigkeit von A

Wahrscheinlichkeit von A

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Wahrscheinlichkeit von A
in Abhängigkeit von B

Wahrscheinlichkeit von B

Wetter	Spielen?
Sonnig	Nein
Bewölkt	Ja
Regen	Ja
Sonnig	Ja
Sonnig	Ja
Bewölkt	Ja
Regen	Nein
Regen	Nein
Sonnig	Ja
Regen	Ja
Sonnig	Nein
Bewölkt	Ja
Bewölkt	Ja
Regen	Nein

Häufigkeitsverteilung		
Wetter	Nein	Ja
Bewölkt		4
Regen	3	2
Sonnig	2	3
Gesamt	5	9

Wahrscheinlichkeiten		
Wetter	Nein	Ja
Bewölkt		4
Regen	3	2
Sonnig	2	3
Gesamt	5	9
	= 5/14	= 9/14
	0,36	0,64

[4]

Wird bei sonnigem Wetter gespielt?

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

$$P(Ja|Sonnig) = \frac{P(Sonnig|Ja) * P(Ja)}{P(Sonnig)}$$

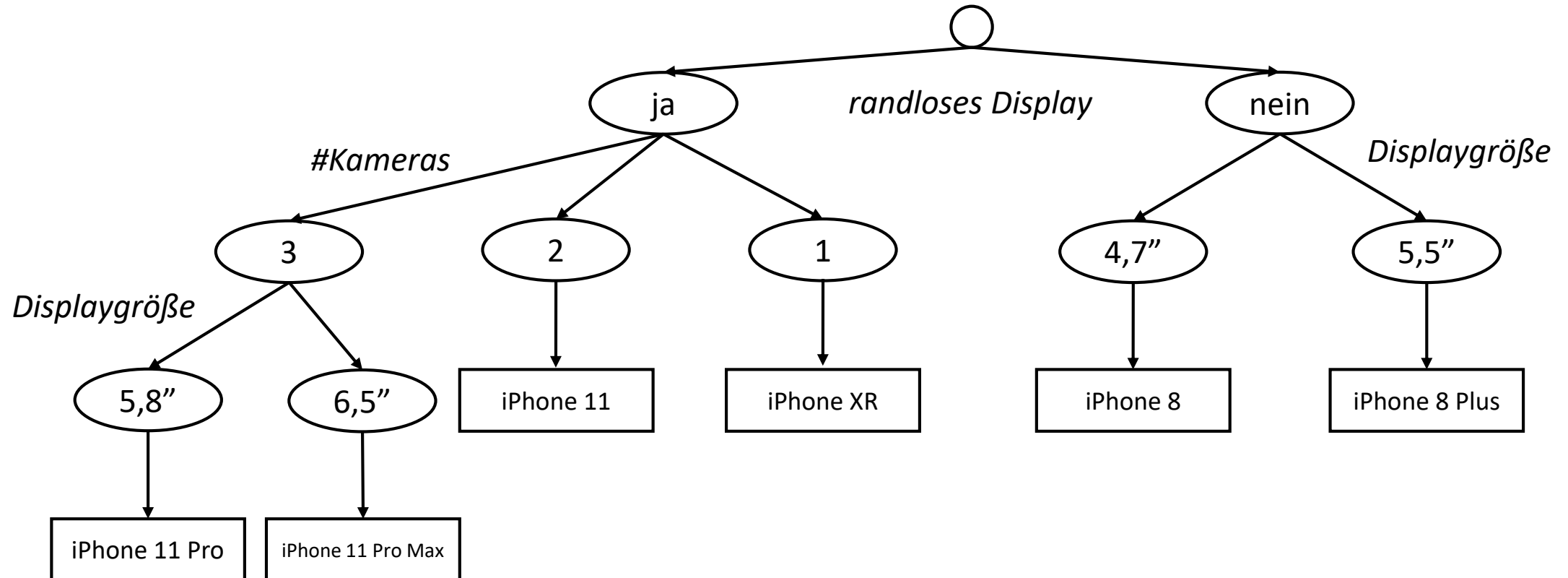
$$P(Ja|Sonnig) = \frac{\frac{3}{9} * \frac{9}{14}}{\frac{5}{14}} = \frac{0,33 * 0,64}{0,36} = \mathbf{0,6}$$

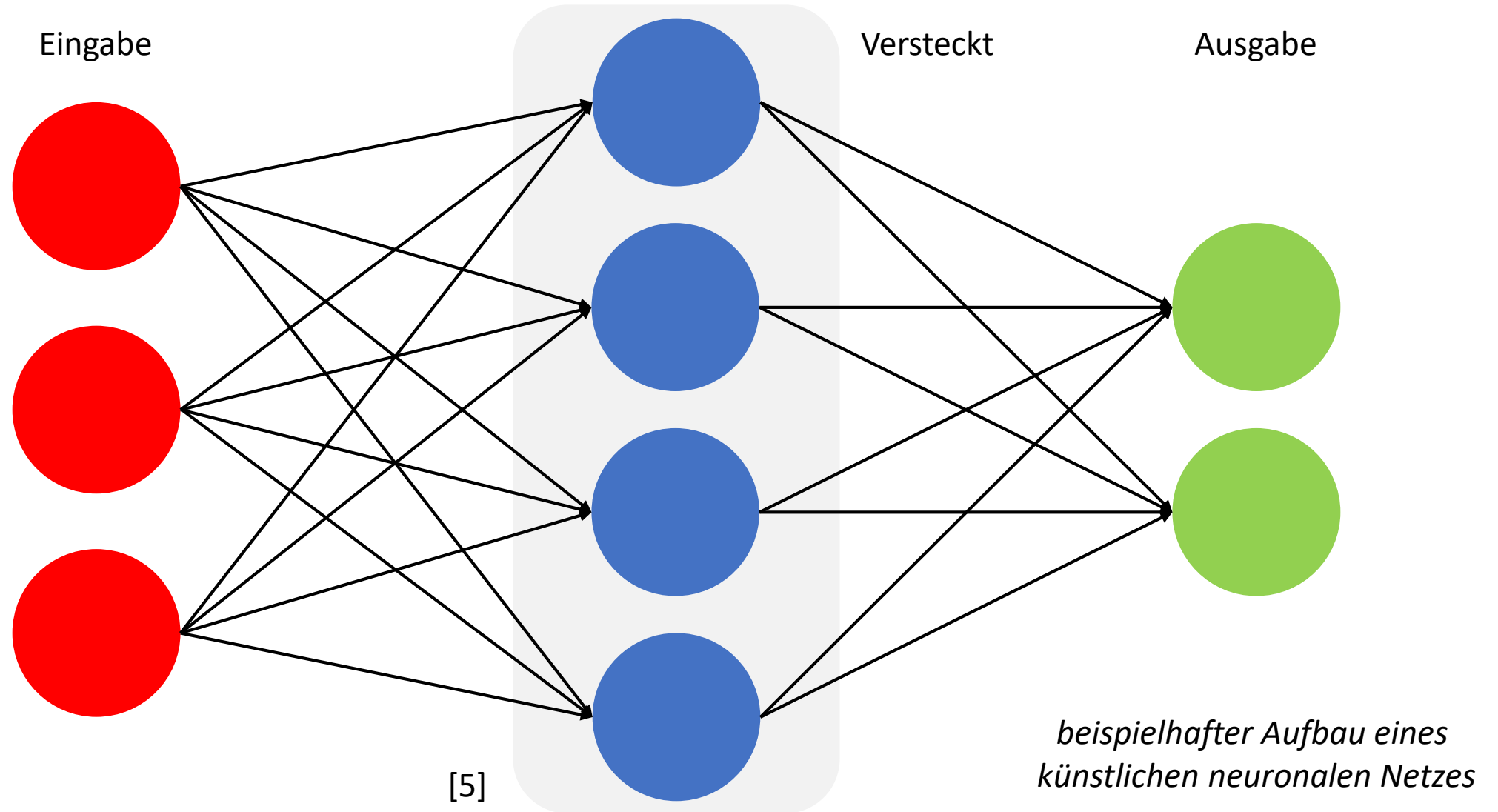
$$P(Nein|Sonnig) = \frac{P(Sonnig|Nein) * P(Nein)}{P(Sonnig)}$$

$$P(Nein|Sonnig) = \frac{\frac{2}{5} * \frac{5}{14}}{\frac{5}{14}} = \frac{0,4 * 0,36}{0,36} = \mathbf{0,4}$$

Es wird gespielt.

Beispielhafter Entscheidungsbaum: aktuelles iPhone-Lineup





<i>Konfusionsmatrix</i>		actual values	
		positive	negative
predicted values	positive	true positive (TP)	false positive (FP)
	negative	false negative (FN)	true negative (TN)

(Jukes, 2017)

- Konfusionsmatrix (Hammouri, Hammad, Alnabhan, & Alsarayrah, 2018; Jukes, 2017)
- Tabelle zur Messung der Performanz von ML Algorithmen
- verschiedene Messwerte berechenbar
 - Treffergenauigkeit: $\frac{TP+TN}{TP+TN+FP+FN}$
 - Genauigkeit: $\frac{TP}{TP+FP}$
 - Trefferquote: $\frac{TP}{TP+FN}$
 - F-Maß: $\frac{2 * Genauigkeit * Trefferquote}{Genauigkeit + Trefferquote}$