

Mein Titel

2. Projektarbeit

vorgelegt am 28. Oktober 2025

Fakultät Wirtschaft und Gesundheit

Studiengang Wirtschaftsinformatik

Kurs WWI2023A

von

JO IMPING

Betreuung in der Ausbildungsstätte:

Dr. Ing. h.c. F. Porsche AG
Paul, Stiegele
IT Product Manager

DHBW Stuttgart:

Prof. Dr. Alexander Brandt

Unterschrift

Inhaltsverzeichnis

Abkürzungsverzeichnis	IV
Abbildungsverzeichnis	V
Tabellenverzeichnis	VI
1 Einleitung	1
1.1 Kontext: Datengetriebene Optimierung im modernen Motorsport	1
1.2 Problemstellung	1
1.3 Zielsetzung	2
1.4 Forschungsansatz und Aufbau der Arbeit	2
2 Theoretische Grundlagen	3
2.1 Design Science Research Methodologie	3
2.2 Experteninterviews zur Anforderungsermittlung	4
2.3 Maschinelle Lernverfahren für Regressionsprobleme	4
2.4 Gradient Boosting Decision Trees	5
2.5 Hyperparameter-Optimierung und Modellvalidierung	6
2.6 Evaluationsmetriken für Regression	6
3 Anforderungsanalyse und Problemdefinition	8
3.1 Problemdomäne und Use-Case-Identifikation	8
3.2 Anforderungsableitung und -bewertung	9
3.3 Abgrenzung des DSR-Artefakts	10
3.4 Forschungsfragen	10
4 Artefakt-Design und Entwicklung	12
4.1 Datensammlung und Analyse	12
4.2 Datenvorbereitung und Feature-Engineering	16
4.2.1 Ableitung der Vorverarbeitungsanforderungen	16
4.2.2 Datenbereinigung und Filterung	17
4.2.3 Feature-Engineering und Dimensionsreduktion	18
4.2.4 Zielvariablen-Glättung als experimentelle Designvariante	19
4.2.5 Datensatz-Aufteilung und Validierungsstrategie	20
4.2.6 Technische Implementierung	21
4.3 Modelltraining und Hyperparameter-Optimierung	22
4.4 Validierung und Modellvergleich	23
5 Evaluation und Interpretation des Vorhersagemodells	24
5.1 Evaluationskonzept und -methodik	24
5.2 Quantitative Leistungsanalyse	25
5.2.1 Performance auf Zufalls-Validierungsdatensätzen	25
5.2.2 Performance auf Event-Validierungsdatensätzen	26
5.2.3 Interpretation des Gesamtergebnisses	28
6 Fazit, Erkenntnisse und Forschungsausblick	30

6.1	Erfüllung der Anforderungen und Beantwortung der Forschungsfragen	30
6.2	Beantwortung der Forschungsfragen	31
6.3	Design Knowledge und kritische Selbsteinschätzung	32
Anhang		33
Literaturverzeichnis		56

Abkürzungsverzeichnis

DSR	Design Science Research
EDA	Explorative Datenanalyse
GBDT	Gradient Boosting Decision Trees
KQL	Kusto Query Language
ML	Machine Learning
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
SVR	Support Vector Regression
IMSA	International Motor Sports Association
WEC	World Endurance Championship

Abbildungsverzeichnis

1	Ausschnitt der Werte der Zielvariable aUndersteer_AVG.	14
2	Plot der Reifentemperatur hinten-rechts.	14
3	Plot des Reifendrucks vorne-links.	15
4	Korrelationsmatrix der 15 am stärksten korrelierten Parameter.	16

Tabellenverzeichnis

1	Übersicht der verwendeten Features und deren Channel-Namen	13
2	Domänenbasierte Schwellwerte für Ausreißererkennung	18
3	Algorithmus-Vergleich: Random-Validierung (alle Komplexitätsstufen)	25
4	Random-Validierung: Performance nach Hyperparameter-Komplexitätsstufe	26
5	Random-Validierung: Effekte der Datensatzstruktur-Varianten	26
6	Algorithmus-Vergleich: Event-Validierung (Leave-One-Event-Out)	27
7	Event-Validierung: Performance nach Komplexitätsstufe (Trend zu besserer Generalisierung)	27
8	Event-Validierung: Effekte der Datensatzstruktur-Varianten	28

1 Einleitung

1.1 Kontext: Datengetriebene Optimierung im modernen Motorsport

Der Motorsport ist ein hochkompetitives Umfeld, in dem Verbesserungen der Fahrzeugperformance oft nur Bruchteile von Sekunden bringen, aber entscheidend sind. Moderne Rennfahrzeuge wie der Porsche LMDh sind mit Tausenden von Sensoren ausgestattet, die kontinuierlich Telemetriedaten erfassen und übertragen. Diese Datenmengen ermöglichen eine beispiellose Sichtbarkeit in Fahrzeugverhalten, Streckenbedingungen und Fahrdynamik. Trotz dieser technologischen Verfügbarkeit verlässt sich die Telemetrie-Analyse in der Praxis stark auf manuelle Prozesse: Performance Engineers sichten Dashboards, identifizieren Auffälligkeiten und leiten daraus Setup-Anpassungen ab. Dies ist zeitintensiv und anfällig für Übersehungen subtiler Muster, die erst bei Kombination mehrerer Parameter sichtbar werden.

Diese Situation repräsentiert ein klassisches Problem der angewandten Informatik: Ein großes Datenvolumen, klare Geschäftsziele, aber unzureichende Automatisierung zur systematischen Mustererkennung. Machine Learning verspricht hier die Fähigkeit, aus historischen Daten automatisch Muster zu erkennen und auf neue Situationen zu generalisieren.

1.2 Problemstellung

Die Fahrzeugbalance ist eine Schlüsselgröße für Rennfahrer und Engineers. Sie beschreibt die Tendenz des Fahrzeugs, in Kurvenfahrt weniger oder stärker zu lenken als vom Fahrer über das Lenkrad eingegeben. Die Fahrzeugbalance beeinflusst direkt Fahrbarkeit, Sicherheit und Renngeschwindigkeit. Die Balance hängt von zahlreichen, komplex verflochtenen Einflussfaktoren ab: Fahrzeugsetup, Reifen-Eigenschaften, Umgebung, Fahrdynamik und Fahrstil.

Das zentrale Problem besteht darin, dass Performance Engineers diese Zusammenhänge derzeit nicht systematisch erfassen. Sie können nicht vorhersagen, welche spezifischen Parameterkombinationen zu welchen Balance-Änderungen führen. Dies zwingt sie zu reaktivem Handeln: Erst wenn Abweichungen in den Daten offensichtlich werden, werden Anpassungen vorgenommen. Ein proaktiver, vorhersagender Ansatz wäre wertvoll, da er die Fähigkeit bietet, frühzeitig zu signalisieren, dass sich die Balance-Charakteristik verschiebt und damit setup-Optimierungen proaktiv zu planen.

Damit ergibt sich die zentrale Forschungsmotivation: Können Machine-Learning-Modelle aus Telemetriedaten automatisch lernen, Fahrzeugbalance vorherzusagen? Und wenn ja, unter welchen Bedingungen generalisieren diese Modelle zuverlässig auf neue Renn-Events mit veränderten Kontexten?

1.3 Zielsetzung

Diese Arbeit verfolgt das Ziel, ein Machine-Learning-Modell zur Vorhersage der Fahrzeugbalance auf Basis aggregierter Telemetrie-Metriken zu entwickeln und systematisch zu evaluieren. Das Modell soll dazu dienen, Performance Engineers zu unterstützen, indem es automatisiert Vorhersagen liefert, statt dass Ingenieure manuell Daten analysieren müssen.

Die Entwicklung folgt der Design Science Research (DSR) Methodologie, die in Kapitel 2 eingeführt wird. Dies bedeutet konkret: (1) Systematische Anforderungsanalyse aus der Anwendungsdomäne, (2) Rigorous Build des Artefakts, (3) Umfassende Evaluierung anhand wissenschaftlicher Metriken, und (4) Reflexion von Design Knowledge für die Wissensbasis.

1.4 Forschungsansatz und Aufbau der Arbeit

Die vorliegende Arbeit strukturiert sich in sechs Kapitel, die den DSR-Prozess vom Problem zur Lösung abbilden. Kapitel 2 vermittelt die theoretischen Grundlagen, Kapitel 3 analysiert Anforderungen, Kapitel 4 dokumentiert das Design und die Entwicklung des Artefakts, Kapitel 5 evaluiert das Modell-Artefakt und Kapitel 6 schließt den DSR-Zyklus mit Reflexion und Ausblick.

Insgesamt leistet diese Arbeit einen doppelten Beitrag: Einerseits entwickelt sie ein Proof-of-Concept für datengetriebene Analyse im Motorsport. Andererseits werden durch systematische Evaluierung die Limitierungen identifiziert und Empfehlungen für zukünftige Arbeiten abgeleitet.

2 Theoretische Grundlagen

2.1 Design Science Research Methodologie

Design Science Research (DSR) ist ein etablierter Forschungsansatz der Wirtschaftsinformatik, der sich grundlegend von deskriptiven Forschungsmethoden unterscheidet. Während traditionelle empirische Forschung primär darauf ausgerichtet ist, bestehende Phänomene zu verstehen und zu erklären, zielt DSR darauf ab, praktische Probleme durch systematische Entwicklung und rigorose Evaluierung von Artefakten zu lösen. Damit verbindet DSR technische und wissenschaftliche Rigor mit praktischer Relevanz.¹

Das Grundwerk von Hevner et al. (2004) prägt bis heute das Verständnis von DSR und etabliert ein Framework, das auf sieben präskriptiven Richtlinien basiert. Das Framework schreibt vor, dass DSR-Projekte in drei ineinandergreifenden Zyklen durchgeführt werden sollten: Der Relevance Cycle beginnt mit Problemidentifikation aus der Anwendungsdomäne; der Rigor Cycle verankert die Entwicklung in wissenschaftlichem Wissen und etablierten Theorien; der Design Cycle orchestriert iterative Phasen von Artefakt-Konzeption, Entwicklung und Evaluierung. Diese drei Zyklen ermöglichen eine systematische und nachvollziehbare Forschungsvorgehensweise, bei der wissenschaftliche Strenge nicht auf Kosten von Praxisrelevanz geht.²

Der DSR-Prozess gliedert sich typischerweise in sechs sequenzielle Phasen. Problem Identification and Motivation beginnt mit der Analyse der Problemdomäne und Begründung ihrer wissenschaftlichen und praktischen Relevanz. Definition of Objectives spezifiziert die Anforderungen, die das entwickelte Artefakt erfüllen muss. In der Design and Development Phase wird das Artefakt konzipiert und prototypisch implementiert. Die Demonstration Phase dokumentiert, dass das Artefakt das Problem tatsächlich lösen kann, typischerweise durch Fallstudien oder kontrollierte Szenarien. Die Evaluation Phase bewertet das Artefakt systematisch gegen die vordefinierten Anforderungen und Ziele. Abschließend erfolgt die Communication Phase, in der Erkenntnisse und Design Knowledge der wissenschaftlichen Gemeinschaft mitgeteilt werden.³

Artefakte in DSR können verschiedene Formen annehmen. Constructs sind konzeptionelle Vokabularien und Abstraktionen, die Probleme präzise definieren. Models stellen Zusammenhänge und Strukturen in vereinfachter Form dar. Methods sind Algorithmen und systematische Verfahrenswesen zur Problemlösung. Instantiations schließlich sind konkrete Implementierungen oder Prototypen.⁴ In ML-Projekten ist die Instantiation typischerweise ein trainiertes Modell mit vollständiger Pipeline (Datenvorverarbeitung, Feature Engineering, trainierte Parameter). Die vorliegende Arbeit entwickelt eine Instantiation: ein evaluiertes Machine-Learning-Modell für Fahrzeugbalance-Vorhersage.

¹Vgl. Hevner et al. 2004, S. 77-83

²Vgl. Hevner et al. 2004, S. 77-92

³Vgl. Prat et al. 2014, S. 1-16

⁴Vgl. Hevner et al. 2004, S. 80

Die Evaluierung in DSR erfüllt mehrere funktionale Rollen. Sie stellt fest, ob und inwieweit das Artefakt die definierten Anforderungen erfüllt. Sie identifiziert Verbesserungspotenziale für weitere Iterationen. Vor allem trägt sie zur wissenschaftlichen Wissensbasis bei, indem Design Principles und generalisierbare Lessons Learned dokumentiert werden.⁵ Bewährte Evaluationsmethoden in DSR sind observational (Feldbeobachtung), analytical (logische Deduktion und Proof-of-Concept), experimental (kontrollierte Experimente mit Baseline-Vergleich), testing (systematische Funktionsprüfung) und descriptive (qualitative Bewertung durch Experten). Für Machine-Learning-Artefakte dominieren analytische und experimentelle Evaluationen mittels etablierter Performance-Metriken.

2.2 Experteninterviews zur Anforderungsermittlung

Die Anforderungsanalyse in Kapitel 3 basiert auf zwei informellen Gesprächen mit einem erfahrenen Performance Engineer aus dem Porsche Motorsport-Team. Das erste Gespräch fand am 29.08.2025 statt, das zweite am 12.09.2025. Ziel war es, durch offene Diskussion die tägliche Arbeitsweise im Telemetrie-Management nachzuvollziehen, zentrale technische Herausforderungen zu identifizieren und realistische Anforderungen an ein automatisiertes Vorhersagemodell zu formulieren.

Beide Gespräche folgten einem offenen, exploratorischen Format ohne strukturierten Fragenkatalog, um eine natürliche Konversation zu fördern und implizites Wissen des Experten zugänglich zu machen. Die gewonnenen Erkenntnisse bestätigten, dass die aktuelle Telemetrie-Analyse stark manuell erfolgt und mehrere Stunden pro Rennwochenende erfordert. Dies validierte die Problemrelevanz und leitete die Anforderungsableitung in Kapitel 3.⁶

2.3 Maschinelle Lernverfahren für Regressionsprobleme

Machine Learning (ML) ist das Teilgebiet der künstlichen Intelligenz, das sich mit der automatischen Induktion von Modellen aus Daten befasst, ohne dass explizite Programmierung erforderlich ist.⁷ Im Kontext dieser Arbeit zielt das ML-Modell auf Regression ab: die Vorhersage einer kontinuierlichen Zielvariable aus einer Menge von kontinuierlichen und kategorialen Input-Features.

Supervised Learning, bei dem jede Trainingsinstanz ein Label hat, wird hier angewendet. Der Bias-Variance Tradeoff ist ein fundamentales Konzept beim Modelllernen: Modelle mit niedriger Komplexität weisen hohen Bias (Underfitting) auf, während hochkomplexe Modelle hohe Varianz aufweisen (Overfitting).⁸ Ein gutes Modell balanciert beide Aspekte aus.

⁵Vgl. Venable et al. 2016, S. 77-89

⁶Vgl. Experteninterview 1 und 2, dokumentiert in Anhang A.1

⁷Vgl. Mitchell 1997, S. 1-2

⁸Vgl. Hastie et al. 2009, S. 23-31

Datenvorverarbeitung ist ein kritischer erster Schritt. Exploratory Data Analysis (EDA) untersucht Verteilungen, erkennt Korrelationen zwischen Features und identifiziert Ausreißer. Data Cleaning adressiert fehlende Werte, erkennt anomale Messwerte und setzt sachlogische Schwellwerte. Feature Engineering transformiert Rohdaten in aussagekräftige Prädiktoren: Feature Selection wählt relevante Variablen aus, Aggregation kombiniert hochkorrelierte Sensoren zu zusammengefassten Features, und Encoding wandelt kategoriale Variablen (etwa Track-Identifikatoren) in numerische Darstellungen um.⁹ Für Telemetriedaten ist insbesondere Zeitreihen-Glättung (beispielsweise Moving Averages) ein etabliertes Verfahren zur Reduktion hochfrequenten Rauschens.

Die Auswahl des Regressionsalgorithmus beeinflusst wesentlich die Modellqualität. Lineare Modelle (Linear Regression, Ridge, Lasso) zeichnen sich durch hohe Interpretierbarkeit aus, können aber komplexe nichtlineare Muster nicht erfassen. Support Vector Regression nutzt Kernel-Tricks zur impliziten nichtlinearen Feature-Transformation. Tree-based Models (Decision Trees) sind intuitiv und robust, zeigen aber Anfälligkeit für Overfitting. Random Forests verbessern dies durch Ensemble-Averaging über viele Bäume.¹⁰ Boosting-Methoden kombinieren schwache Lerner sequenziell zu stärkeren Modellen. Neuronale Netze können beliebig komplexe Funktionen approximieren, erfordern aber große Datenmengen und bieten weniger Interpretierbarkeit.¹¹

2.4 Gradient Boosting Decision Trees

Gradient Boosting Decision Trees (GBDT) sind eine sequenzielle Ensemble-Methode, bei der mehrere Decision Trees nacheinander trainiert werden, wobei jeder nachfolgende Baum systematisch die Residuen (Vorhersagefehler) des vorherigen Baums korrigiert.¹² Das zugrundeliegende Prinzip ist Residual Learning: Der i -te Baum wird trainiert, um die Vorhersagefehler von Baum $i - 1$ zu minimieren. Die finale Vorhersage ergibt sich aus einer gewichteten Summe aller Baum-Ausgaben.

Mathematisch werden die Bäume durch einen Gradient-Descent-Prozess auf die Minimierung einer Loss-Funktion (typischerweise Mean Squared Error) optimiert. Jeder neue Baum wird so angefitet, dass er den negativen Gradienten der Loss-Funktion verringert, was zu schnellerer Konvergenz gegenüber Standard-Ensemble-Methoden führt.¹³

Zwei prominente Implementierungen sind XGBoost und LightGBM. XGBoost (eXtreme Gradient Boosting) bietet hochoptimierte Algorithmen mit integrierten Regularisierungsmechanismen, Histogram-basiertem Split-Finding für Geschwindigkeit und nativer Unterstützung kategorialer Features.¹⁴ LightGBM (Light Gradient Boosting Machine) differenziert sich durch Leaf-wise

⁹Vgl. Guyon, Elisseeff 2003, S. 1157-1182

¹⁰Vgl. Breiman 2001, S. 5-32

¹¹Vgl. Goodfellow et al. 2016, S. 164-223

¹²Vgl. Friedman 2001, S. 1189-1232

¹³Vgl. Friedman 2001, S. 1200-1210

¹⁴Vgl. Chen, Guestrin 2016, S. 785-794

Baum-Wachstum (maximale Fehlerreduktion pro Blatt statt pro Level), Gradient-based One-Side Sampling zur Datensatzverkleinerung und Exclusive Feature Bundling zur Dimensionsreduktion.¹⁵ Beide Implementierungen zeigen empirisch überlegene Performance auf strukturierten (tabularen) Daten im Vergleich zu tiefen neuronalen Netzen.¹⁶

2.5 Hyperparameter-Optimierung und Modellvalidierung

Hyperparameter sind Konfigurationsparameter eines ML-Modells, die vor dem Training gesetzt werden und sich von Modell-Parametern unterscheiden, die während des Trainings gelernt werden.¹⁷ Bei GBDT sind kritische Hyperparameter die Anzahl der Estimators (Baumanzahl), die Learning Rate (Schrittweite), die Max Depth (Baumtiefe) sowie Regularisierungsparameter wie L2-Penalisierung und Mindestanzahl Samples pro Blatt.

Die Optimierung wird typischerweise durch Grid Search (systematische Durchsuchung eines vordefinierten Parameterraums), Random Search (stochastisches Sampling) oder Bayesian Optimization (probabilistische Modellierung) durchgeführt.¹⁸

Cross-Validation ist eine Standardtechnik zur Bewertung der Modellgeneralisierung ohne separaten Validierungsdatensatz. k-fold Cross-Validation partitioniert die Daten in k Teile, trainiert k Modelle (jeweils mit einem anderen Teil als Validierung) und mittelt die Performance-Metriken.¹⁹ Bei Zeitreihendaten wird Time Series Cross-Validation verwendet, um das Forward-Chaining-Prinzip zu simulieren. Leave-One-Group-Out ist eine extreme Variante, bei der jede Gruppe (etwa ein Renn-Event) einmal als Validierungsmenge fungiert – ein besonders rigoros für die Evaluierung von Domänen-Generalisierung.²⁰

2.6 Evaluationsmetriken für Regression

Regressionsergebnisse werden durch mehrere etablierte Fehlermetriken quantifiziert. Der Mean Squared Error (MSE) berechnet das Durchschnitt der quadrierten Abweichungen zwischen Vorhersagen und Ist-Werten und bestraft größere Fehler überproportional. Der Root Mean Squared Error (RMSE) ist die Quadratwurzel des MSE und hat dieselbe Einheit wie die Zielvariable, was die Interpretation erleichtert.²¹

Der Mean Absolute Error (MAE) quantifiziert den Durchschnitt der absoluten Abweichungen und ist robuster gegenüber Ausreißern, da er Fehler linear (nicht quadratisch) gewichtet.²² Die

¹⁵Vgl. Ke et al. 2017, S. 3146-3154

¹⁶Vgl. Chen, Guestrin 2016, S. 788-792

¹⁷Vgl. Bergstra, Bengio 2012, S. 281-305

¹⁸Vgl. Bergstra, Bengio 2012, S. 281-305

¹⁹Vgl. Kohavi 1995, S. 1137-1145

²⁰Vgl. Kapitel 4.1

²¹Vgl. Hodson 2022, S. 5481-5482

²²Vgl. Chai, Draxler 2014, S. 1247-1250

Wahl zwischen RMSE und MAE sollte sich nach der erwarteten Fehlerverteilung richten: RMSE ist optimal bei normalverteilten Fehlern, MAE bei Laplace-verteilten Fehlern.

Das Bestimmtheitsmaß R^2 (Coefficient of Determination) gibt an, welcher Anteil der Varianz der Zielvariable durch das Modell erklärt wird. $R^2 = 1$ signalisiert perfekte Vorhersagen, $R^2 = 0$ bedeutet, dass das Modell nicht besser als die Mittelwert-Baseline ist. Negative R^2 -Werte sind möglich und deuten auf schlechtere Performance als die Baseline hin.²³ In der Ingenieurpraxis wird für Validierungsmodelle ein Schwellwert von $R^2 \geq 0,7$ angestrebt, um praktische Einsatzfähigkeit zu gewährleisten.²⁴

Diese drei Metriken (RMSE, MAE, R^2) bilden den internationalen Standard in der Regressionsanalyse und ermöglichen Vergleichbarkeit mit etablierten Benchmarks in der Fachliteratur.

²³Vgl. Hastie et al. 2009, S. 18-25

²⁴Vgl. 365 Data Science 2023

3 Anforderungsanalyse und Problemdefinition

3.1 Problemdomäne und Use-Case-Identifikation

Im Rahmen dieses Projekts wird mit Telemetriedaten von Porsche LMDh-Rennwagen gearbeitet. LMDh (Le Mans Daytona hybrid) ist eine Fahrzeugklasse für Langstreckenrennen, die in den Weltmeisterschaften IMSA WeatherTech SportsCar Championship und FIA World Endurance Championship (WEC) eingesetzt wird.

Die Telemetriedaten werden über mehrere Tausend Sensoren erfasst und liefern während Trainings- und Rennsessions ununterbrochen Messwerte, die in Echtzeit in die Porsche Motorsport Cloud-Plattform übertragen werden.²⁵ Dort liegen sie als Zeitreihendaten vor und stehen Performance Engineers wahlweise direkt für Detailanalysen zur Verfügung oder werden in Form von Metriken aufbereitet. Unter Metriken versteht man statistische Kennzahlen wie Durchschnitt, Minimum oder Maximum über definierte Zeitabschnitte, zum Beispiel pro Runde oder pro Strecken-Sektion. Diese Metriken bilden die Grundlage, auf der Performance Engineers ihre tägliche Arbeit aufbauen.²⁶

Im aktuellen Workflow prüfen Performance Engineers zunächst die Kennzahlen in Dashboards, um Auffälligkeiten zu erkennen. Das können Temperatursprünge in schnellen Kurven sein oder ungewöhnlich hoher Reifenverschleiß auf bestimmten Streckenabschnitten.²⁷ Allerdings fällt auf, dass diese Auswertung fast ausschließlich manuell erfolgt. Die Ingenieure verbringen pro Rennwochenende mehrere Stunden damit, Metriken zu sichten, Trends zusammenzuführen und in Setup-Empfehlungen zu übersetzen. Das führt nicht nur zu Verzögerungen, sondern birgt auch das Risiko, subtilere Muster zu übersehen, etwa wenn ein Zusammenspiel aus Streckentemperatur, Gas- und Bremsprofil nur in Extremlagen auffällt.

Aus dieser Ausgangslage ergibt sich ein Use Case, der direkt an das beschriebene Problem anschließt: die Vorhersage der Fahrzeugbalance, gemessen durch den `aUndersteer_AVG`-Kanal, auf Basis der vorhandenen Telemetrie-Metriken.²⁸ Diese Metrik quantifiziert die Fahrzeugbalance numerisch, wobei positive Werte Untersteuern und negative Werte Übersteuern des Fahrzeugs anzeigen. Das Ziel ist nicht, sämtliche Sensorrohdaten in Echtzeit zu verarbeiten, sondern die bereits aggregierten Metriken zu nutzen, um eine Balance-Prognose zu erstellen. Auf diese Weise könnten Ingenieure statt langer manueller Durchsicht direkt fundierte Empfehlungen erhalten und proaktiv handeln.

Mit der Fahrzeugbalance-Vorhersage würde sich der Arbeitsablauf von reaktivem Nachjustieren hin zu vorausschauender Optimierung verschieben. Ingenieure könnten Anpassungen bereits dann

²⁵Vgl. Experteninterview 1, 29.08.2025

²⁶Vgl. Experteninterview 2, 12.09.2025

²⁷Vgl. Experteninterview 1, 29.08.2025

²⁸Vgl. Experteninterview 2, 12.09.2025

vornehmen, wenn sich ein akuter Über- oder Untersteuern-Trend ankündigt. Darüber hinaus verspricht dieser Use Case eine objektivere Entscheidungsbasis: Anstelle persönlicher Einschätzungen stünden reproduzierbare Kennzahlenmodelle im Mittelpunkt. Damit würde das bestehende System von punktueller Datenansicht auf datengetriebene Automatisierung übergehen und den Zeitaufwand für Analyse sowie Setup-Änderungen deutlich verringern.

3.2 Anforderungsableitung und -bewertung

Aus den beiden Experteninterviews mit dem Performance Engineer am 29.08.2025 und 12.09.2025 lassen sich konkrete Anforderungen an das ML-Artefakt ableiten.

Die erste und primäre funktionale Anforderung betrifft die Fähigkeit, die Fahrzeugbalance auf Basis vorhandener Telemetrie-Metriken zuverlässig vorherzusagen. Dieses Ziel folgt direkt aus der Erkenntnis, dass manuelle Analysen mehrere Stunden pro Rennwochenende beanspruchen und dass frühe Hinweise auf Balanceabweichungen häufig erst verspätet offensichtlich werden.²⁹ Das Modell soll ohne manuelle Intervention vorhersagen können, welche Understeer-Werte für eine gegebene Kombination von Telemetrie-Inputs zu erwarten sind. Damit wird das Ziel einer automatisierten Understeer-Vorhersage definiert.

Neben der reinen Funktionalität muss das Modell eine hinreichende Vorhersagegenauigkeit aufweisen. In der Fahrzeugtechnik und Ingenieurwissenschaften wird für Validierungsmodelle ein R^2 -Wert von mindestens 0,7 angestrebt, um praktische Einsatzfähigkeit zu gewährleisten.³⁰ Unterhalb dieses Schwellenwerts ist die Prognose zu unsicher, um darauf Setup-Entscheidungen zu stützen.

Die dritte Anforderung betrifft die Reproduzierbarkeit und Dokumentation. Eine lückenlose Dokumentation aller Eingangsdaten, Vorverarbeitungsschritte, Modellparameter und Evaluationsergebnisse ist vorgesehen. Nur so kann vollständige Reproduzierbarkeit gewährleistet werden, und die erstellten Prognosen bleiben validierbar.³¹ Diese Anforderung entspricht den Prinzipien der Design Science Research Methodologie, die eine nachvollziehbare Artefakt-Entwicklung fordert.

Bei der Priorisierung dieser Anforderungen zeigt sich folgende Hierarchie: Die erste und zweite Anforderung sind kritisch für die Zielerreichung. Ohne automatisierte Vorhersage und ausreichende Genauigkeit ist das Artefakt nicht einsetzbar. Die dritte Anforderung ist methodisch essentiell, stellt aber keine funktionale Begrenzung dar. Die Evaluation in Kapitel 5 wird zeigen, dass Anforderung 1 partiell, Anforderung 2 nicht und Anforderung 3 vollständig erfüllt wurden.

²⁹Vgl. Experteninterview 1, 29.08.2025

³⁰Vgl. Salaani 2021; vgl. auch 365 Data Science 2023

³¹Vgl. Venable, Pries-Heje, Baskerville 2016, S. 79

3.3 Abgrenzung des DSR-Artefakts

Das entwickelte Artefakt beschränkt sich auf ein Vorhersagemodell für Understeer-Werte, basierend auf aggregierten Telemetrie-Metriken pro Runde. Diese fokussierte Auslegung ermöglicht es, das Projekt im vorgesehenen Zeitrahmen vollständig durchzuführen.

Hinsichtlich der Implementierung arbeitet das Modell auf rundenweise aggregierten Metriken, nicht auf Rohdaten-Sensorströmen. Dies reduziert die Komplexität erheblich und erlaubt fokussiertere Feature-Engineering-Strategien.³² Eine Echtzeitvorhersage, die kontinuierlich auf Sensor-Einzelmessungen reagiert, wird damit nicht angestrebt.

Bezüglich der Zielvariablen konzentriert sich das Modell ausschließlich auf Understeer. Andere Balance-Metriken wie Oversteer oder Aero-Balance werden nicht vorhergesagt. Dies beschränkt das Vorhersage-Problem auf eine univariate Regression und reduziert damit den Modellierungsaufwand.

Schließlich liefert das Modell keine direkten Setup-Vorschläge, sondern ausschließlich Understeer-Prognosen. Die Ableitung von Setup-Änderungen aus diesen Prognosen bleibt in der Verantwortung der Performance Engineers und wird nicht automatisiert. Damit bleibt die Entscheidungshoheit bei den Ingenieuren, während das Modell als Entscheidungsunterstützungssystem fungiert.

Diese Abgrenzung reduziert den Projektumfang auf ein klar definiertes Machine-Learning-Regressionsproblem und ermöglicht eine fokussierte Evaluation im DSR-Kontext.³³

3.4 Forschungsfragen

Aus der Problemdefinition und den Anforderungen ergeben sich drei zentrale Forschungsfragen, die das Projekt leiten und in den nachfolgenden Kapiteln beantwortet werden.

Die erste Forschungsfrage adressiert die grundlegende Machbarkeit des Ansatzes: Können Gradient Boosting Decision Trees (GBDT) Understeer-Verhalten in Motorsport-Telemetriedaten vorhersagen? GBDT gelten als state-of-the-art für strukturierte Regressionsprobleme,³⁴ aber ihre Anwendbarkeit auf Motorsport-Telemetriedaten ist bisher nicht systematisch untersucht. Diese Frage wird in Kapitel 5 und 6 umfassend beantwortet.

Die zweite Forschungsfrage vergleicht zwei konkrete Algorithmen: Welcher Algorithmus (XGBoost vs. LightGBM) generalisiert besser auf unbekannte Rennevents? XGBoost und LightGBM

³²Vgl. Kap. 4.2

³³Vgl. Hevner et al. 2004, S. 83

³⁴Vgl. Chen, Guestrin 2016, S. 785-794

implementieren unterschiedliche Wachstumsstrategien (level-wise vs. leaf-wise) und Optimierungen.³⁵ Welcher Algorithmus robuster generalisiert, ist unklar und wird durch systematischen Vergleich auf Event-basierten Validierungsdatensätzen beantwortet (Kapitel 5.2).

Die dritte Forschungsfrage konzentriert sich auf die Inputseite des Modells: Wie wirken sich Datenvorbereitung (Feature-Engineering, Glättung, Aggregation) und Hyperparameter-Tuning auf die Vorhersagegenauigkeit aus? Verschiedene Vorverarbeitungsschritte, kategoriale Features, Zielvariablen-Glättung und Hyperparameter-Komplexitätsstufen werden getestet, um ihren Einfluss auf Performance zu quantifizieren (Kapitel 4.2, 4.3, 5.2).

Diese drei Forschungsfragen strukturieren den Aufbau der Arbeit: Kapitel 4 entwickelt das Artefakt systematisch, Kapitel 5 evaluiert es anhand der Forschungsfragen, und Kapitel 6 synthetisiert die Erkenntnisse in Design Knowledge für zukünftige Arbeiten.

³⁵Vgl. Ke et al. 2017, S. 3146-3154

4 Artefakt-Design und Entwicklung

Die Entwicklung des Artefakts folgte einem strukturierten Prozess, der sicherstellt, dass jede technische Entscheidung sowohl praxisnah als auch wissenschaftlich fundiert ist. In diesem Kapitel werden die Schritte zur Datenvorbereitung, Implementierung der Trainingspipeline und Hyperparameter-Optimierung detailliert diskutiert und begründet.

4.1 Datensammlung und Analyse

Die vorliegende Arbeit nutzt Telemetriedaten aus der Porsche Motorsport Cloud Plattform. Es wurden sämtliche Rennsessions der International Motor Sports Association (IMSA)- und World Endurance Championship (WEC)-Meisterschaften der Jahre 2023 bis 2025 extrahiert. Zur Minimierung von Varianz durch unterschiedliche Fahrsituationen beschränkt sich die Auswahl auf Rennsessions, während Trainings- und Qualifikationssessions ausgeschlossen wurden. Zusätzlich erfolgte eine Filterung auf Runden mit Trockenreifen, da Regenbedingungen weitere Einflussfaktoren einführen. Die Datenselektion wurde direkt beim Abruf mittels der ADX-Kusto Query Language (KQL) vorgenommen, wodurch ein Rohdatensatz im CSV-Format mit 17 735 Runden (Datenpunkten) resultierte.

Im Rahmen eines Experteninterviews mit dem Performance Engineer wurden jene Parameter identifiziert, die als Merkmale (Features) in das Machine-Learning-Modell eingehen.³⁶ Die Merkmale gliedern sich in kontinuierliche und kategoriale Features die in Tabelle-1 aufgelistet sind.

³⁶Vgl. Experteninterview 2 2025

Feature Name	Channel Name	Erklärung
Kontinuierliche Features		
Umgebungstemperatur	TAmbientVms_AVG	Lufttemperatur der Umgebung
Reifentemperatur VL	TTyreIRFLavg	Temperatur des vorderen linken Reifens (Innenrand)
Reifentemperatur VR	TTyreIRFRavg	Temperatur des vorderen rechten Reifens (Innenrand)
Reifentemperatur HL	TTyreIRRLavg	Temperatur des hinteren linken Reifens (Innenrand)
Reifentemperatur HR	TTyreIRRRavg	Temperatur des hinteren rechten Reifens (Innenrand)
Reifendruck VL	pTyreFL_avg	Luftdruck des vorderen linken Reifens
Reifendruck VR	pTyreFR_avg	Luftdruck des vorderen rechten Reifens
Reifendruck HL	pTyreRL_avg	Luftdruck des hinteren linken Reifens
Reifendruck HR	pTyreRR_avg	Luftdruck des hinteren rechten Reifens
Fuel Load	mFuelMass_AVG	Aktuelle Kraftstoffmasse im Tank
Tyre Mileage VL	TyreMilage_FL	Laufleistung des vorderen linken Reifens
Tyre Mileage VR	TyreMilage_FR	Laufleistung des vorderen rechten Reifens
Tyre Mileage HL	TyreMilage_RL	Laufleistung des hinteren linken Reifens
Tyre Mileage HR	TyreMilage_RR	Laufleistung des hinteren rechten Reifens
Reifendruck Asymmetrie	tire_pressure_asymmetry	Asymmetrie zwischen linken und rechten Reifen
Reifendruck Balance	tire_pressure_balance	Balance zwischen Vorder- und Hinterachse
Reifendruck Spread	tire_pressure_spread	Spreizung der Reifendruckwerte
Reifen-Umgebungstemperatur Delta	tire_temp_ambient_delta	Temperaturdifferenz Reifen zu Umgebung
Reifentemperatur Gradient Max	tire_temp_gradient_max	Maximaler Temperaturgradient zwischen Reifen
Kategoriale Features		
Mechanical Balance Front	NDriverARBSettingFAvg	Einstellung der vorderen Stabilisatorsteifigkeit
Mechanical Balance Rear	NDriverARBSettingRAvg	Einstellung der hinteren Stabilisatorsteifigkeit
Brake Balance	rBrakeBiasOffsetRequest_AVG	Bremskraftverteilung Vorder-/Hinterachse
Traction Control Longitudinal	NTCLongitudinal_AVG	Traktionskontrolle längs
Traction Control Lateral	NTCLateral_AVG	Traktionskontrolle quer
Tyre State	NTyreState_AVG	Reifenmischung(hart/medium/soft)
Event Category	eventCategory	Rennstrecke
Zielvariable		
Understeer Average	aUndersteer_AVG	Durchschnittlicher Untersteer-Wert pro Runde

Tab. 1: Übersicht der verwendeten Features und deren Channel-Namen

Als Zielvariable dient der durchschnittliche Untersteer-Wert pro Runde (aUndersteer_AVG) **Berechnung in DP ergänzen**, wobei Werte über Null Untersteuern und Werte unter Null Übersteuern des Fahrzeugs anzeigen. Alle Parameter und die Zielvariable wurden rundenmittelnd aggregiert, sodass jeder Datensatzpunkt einer einzelnen Rennrunde entspricht.

Die Explorative Datenanalyse (EDA) **Wissenschaftliche Quelle EDA ergänzen** wurde durchgeführt, um die Dateneigenschaften zu untersuchen und potenzielle Datenqualitätsprobleme zu identifizieren. Zunächst wurde die Verteilung der Zielvariable aUndersteer_AVG untersucht.

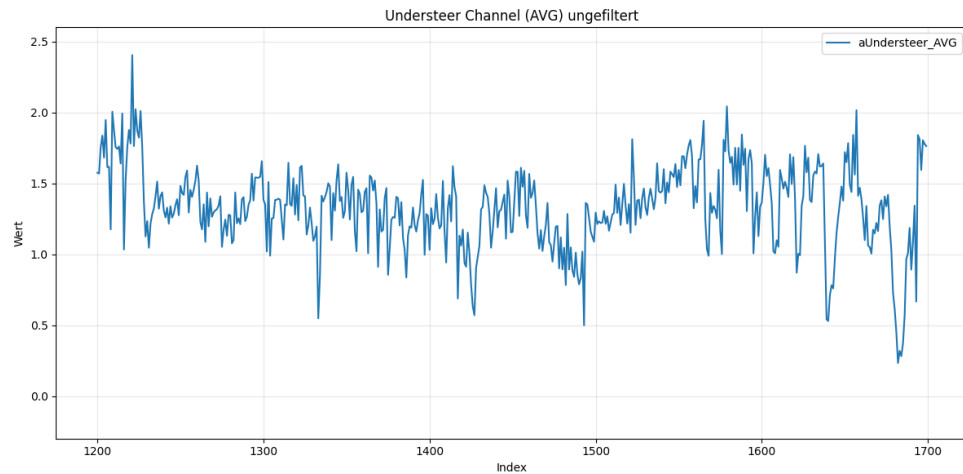


Abb. 1: Ausschnitt der Werte der Zielvariable aUndersteer_AVG.

Der in Abbildung 1 dargestellte Verlauf des Understeer-Channels ist auffällig, denn obwohl es sich bereits um Durchschnitte über jeweils eine Runde handelt, weist der Graph eine hohe Volatilität auf. Diese Erkenntnis sollte in der Modellierung der Vorverarbeitungsschritte berücksichtigt werden, um die Robustheit des Modells zu erhöhen.

Die Verteilungen der zentralen kontinuierlichen Features wurden ebenfalls untersucht. Abbildung 2 zeigt exemplarisch die Verteilung der Reifentemperatur hinten-rechts. Dabei sind sowohl realistische, aber extreme Werte unter 40 Grad Celsius als auch auffällige, unrealistische Ausreißer über 300 Grad Celsius zu erkennen. Die Einschätzung, ob es sich um valide Daten handelt, erfolgt durch Informationen aus Experteninterviews.³⁷ Diese Ausreißer deuten auf potenzielle Sensorfehler oder Datenqualitätsprobleme hin und werden in der Datenvorbereitung entsprechend behandelt.

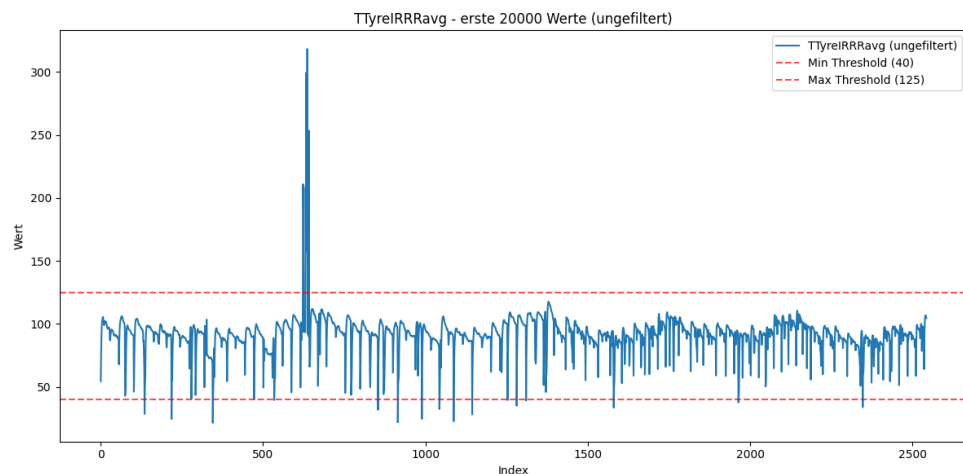


Abb. 2: Plot der Reifentemperatur hinten-rechts.

³⁷Vgl. Experteninterview 1 2025

Auch bei den Werten des Reifendrucks gibt es auffällige Ausreißer. Abbildung 3 zeigt die Verteilung des Reifendrucks vorne-links. Hier sind ebenfalls unrealistische Werte zu erkennen, die auf mögliche Datenqualitätsprobleme hinweisen und in der Vorverarbeitung entsprechend berücksichtigt werden müssen.

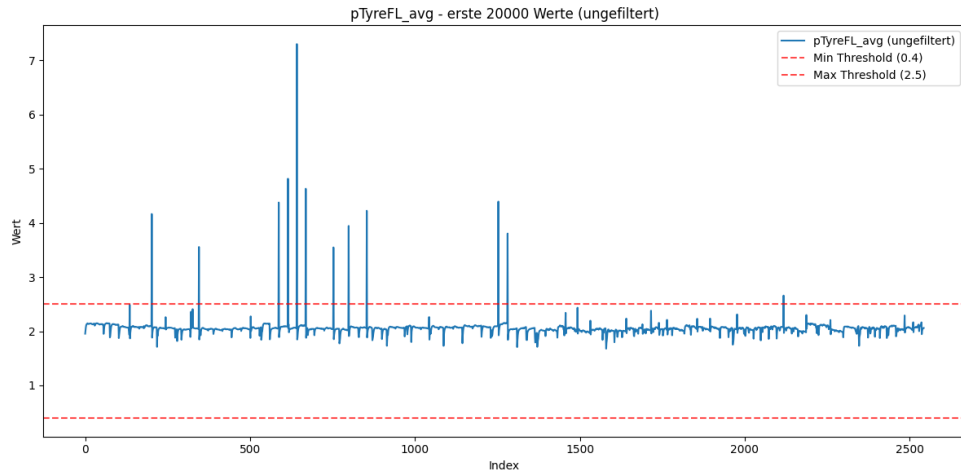


Abb. 3: Plot des Reifendrucks vorne-links.

Abschließend wurde mittels Korrelationsmatrix die Stärke der Zusammenhänge aller Merkmale untersucht. Dabei ergaben sich insbesondere enge Korrelationen zwischen den Reifentemperatur- und Reifendrucksensoren sowie zwischen der Kraftstoffmenge und der Anzahl der Runden pro Reifenlaufleistung.

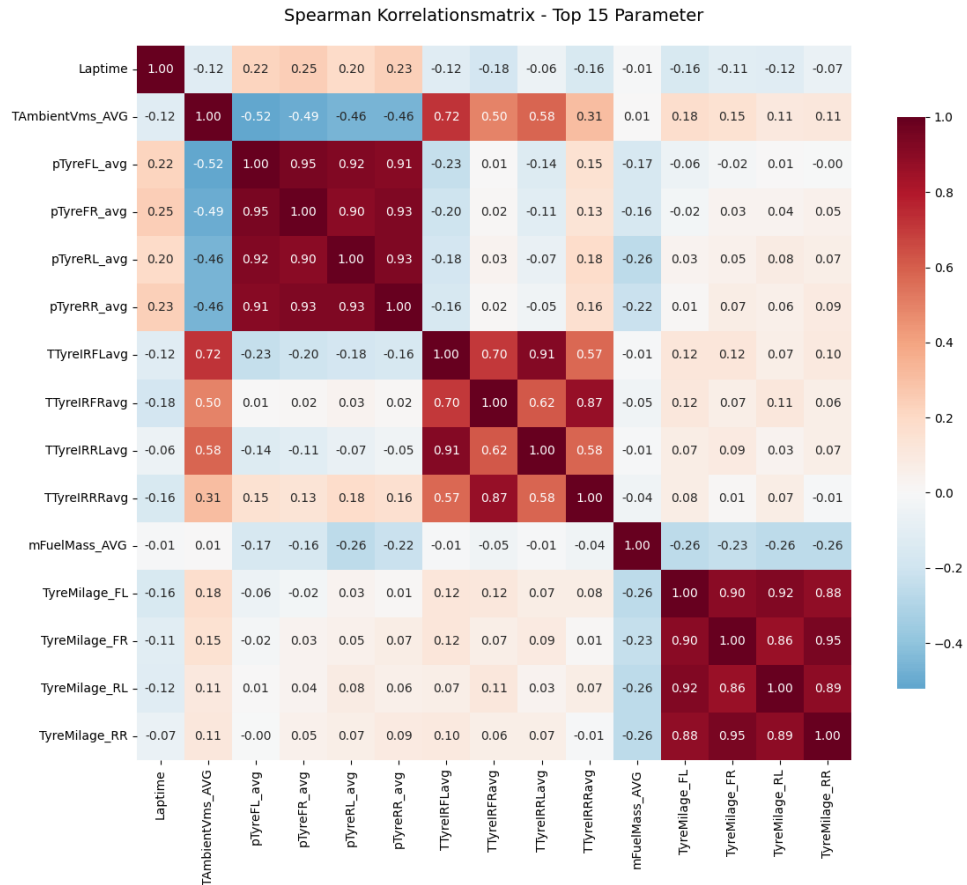


Abb. 4: Korrelationsmatrix der 15 am stärksten korrelierten Parameter.

Die explorative Analyse zeigte (i) stark schwankende Zielgrößen trotz Rundenglättung, (ii) häufige Ausreißer und Sensorartefakte bei Temperatur und Druck, sowie (iii) hohe Redundanzen in korrelierten Messkanälen (Reifen- und Drucksensorik, Fuel Load vs. Tyre Mileage). Im nächsten Abschnitt werden die daraus abgeleiteten Vorverarbeitungsschritte und das Feature-Engineering formalisiert und implementierungsorientiert beschrieben.

4.2 Datenvorbereitung und Feature-Engineering

Basierend auf den EDA-Erkenntnissen werden nachfolgend die systematischen Vorverarbeitungsschritte zur Transformation des Rohdatensatzes in ein trainingstaugliches Format dokumentiert.

4.2.1 Ableitung der Vorverarbeitungsanforderungen

Aus den Erkenntnissen der explorativen Datenanalyse ergeben sich konkrete Anforderungen an die Datenvorbereitung. Die hohe Volatilität der Zielvariable aUndersteer_AVG erfordert Strate-

gien zur Glättung von Zeitreihenschwankungen, da selbst rundenmittelnd aggregierte Werte eine unregelmäßige Verteilung aufweisen. Die identifizierten Sensorartefakte bei Reifentemperaturen über 300 Grad Celsius und unrealistische Werte bei Reifendruckmessungen indizieren Messfehler, die durch domänenbasierte Schwellwertfilterung adressiert werden müssen.³⁸ Ferner zeigt die Korrelationsmatrix aus Abschnitt 4.1 starke Abhängigkeiten zwischen einzelnen Sensoren derselben physikalischen Größe, was eine Reduktion redundanter Features nahelegt. **Quelle**

Diese Anforderungen entsprechen den Design Requirements für das Datenartefakt im Sinne der DSR-Methodik. Die Vorverarbeitungsentscheidungen werden transparent dokumentiert und auf drei Quellen zurückgeführt: (i) datengetriebene Erkenntnisse der EDA, (ii) domänengetriebene Validierung durch Experteninterviews sowie (iii) theoriegetriebene Fundierung durch etablierte Machine-Learning-Literatur zu Datenvorverarbeitung und Ausreißererkennung.

4.2.2 Datenbereinigung und Filterung

Zur Minimierung systematischer Verzerrungen wurden zunächst alle Out-Laps (Runde = 1) aus dem Datensatz entfernt. Out-Laps weisen typischerweise atypische Charakteristika auf, da Fahrzeuge die Boxengasse verlassen und sich das thermische Verhalten der Reifen von regulären Rennrunden unterscheidet. Diese Filterregel reduziert Varianz durch nicht-repräsentative Datenpunkte und stellt sicher, dass das Modell ausschließlich auf Basis von Rennrunden mit stabilisiertem Fahrzeugverhalten trainiert wird.

Eine weitere Quelle von Varianz sind Extremereignisse während des Rennens, wie Unfälle, Safety-Car-Phasen oder technische Defekte. Diese manifestieren sich in der Regel durch extreme Abweichungen der Rundenzeit vom durchschnittlichen Niveau. Zur Identifikation solcher Ereignisse wurde eine statistische Schwellwertmethode angewendet. Runden, deren Rundenzeit um mehr als eine Standardabweichung (ca. 40 Sekunden) vom Mittelwert abweichen, wurden aus dem Datensatz entfernt. Diese Filterung folgt etablierten statistischen Verfahren zur Ausreißererkennung in Zeitreihendaten und trägt zur weiteren Varianzreduktion bei.³⁹

Für die kontinuierlichen Features Reifentemperatur und Reifendruck wurden domänenspezifische Schwellwerte zur Erkennung und Entfernung unrealistischer Messwerte definiert. Wie in Abbildung 2 dargestellt, treten bei Reifentemperaturen Werte über 300 Grad Celsius auf, die physikalisch nicht plausibel sind und auf Sensorfehler hindeuten. Analog zeigen Reifendruckmessungen (Abbildung 3) Ausreißer außerhalb realistischer Bereiche.

Die Festlegung der konkreten Grenzwerte erfolgte unter Einbeziehung von Expertenwissen.⁴⁰ Dieser domänenbasierte Ansatz zur Ausreißererkennung ist in der Machine-Learning-Literatur als effektive Methode etabliert, wenn technisches Fachwissen verfügbar ist.⁴¹ Die Implementierung

³⁸Vgl. Experteninterview 1 2025

³⁹Vgl. Dash et al. 2023

⁴⁰Vgl. Experteninterview 1 2025

⁴¹Vgl. Alan 2011

erfolgte durch Filterregeln, die Datenpunkte außerhalb der definierten Grenzwerte ausschließen.

Feature-Kategorie	Unterer Grenzwert	Oberer Grenzwert
Reifendruck (alle Räder) [bar]	1,3	2,5
Reifentemperatur (alle Räder) [°C]	40	125
Kraftstoffmasse [kg]	0	120

Tab. 2: Domänenbasierte Schwellwerte für Ausreißererkennung

4.2.3 Feature-Engineering und Dimensionsreduktion

Die in Abschnitt 4.1 präsentierte Korrelationsmatrix (Abbildung 4) offenbart hohe Korrelationen zwischen einzelnen Sensoren der Reifentemperatur sowie des Reifendrucks. Solche redundanten Features können bei Machine-Learning-Modellen zu Multikollinearität führen und die Interpretierbarkeit reduzieren.⁴² Zur Quantifizierung der Redundanz wurde ein korrelationsbasiertes Verfahren angewendet: Features mit einer absoluten Pearson-Korrelation über 0,9 zu anderen Features wurden als hochkorreliert klassifiziert.⁴³ Die Identifikation dieser Feature-Gruppen bildet die Grundlage für die nachfolgende Feature-Aggregation. Anstatt hochkorrelierte Features vollständig zu entfernen, wurden neue motorsport-relevante Features durch statistische Zusammenfassung der Sensorgruppen erstellt. Aus den vier Reifendruck-Sensoren (vorne-links, vorne-rechts, hinten-links, hinten-rechts) und den entsprechenden Temperatursensoren wurden folgende abgeleitete Features generiert:

- **tire_pressure_asymmetry**: Links-Rechts-Balance ($|\text{FL} - \text{RL}|$)
- **tire_pressure_balance**: Vorn-Hinten-Balance (Durchschnitt vorne - hinten)
- **tire_pressure_spread**: Setup-Homogenität (max - min aller Drücke)
- **tire_temp_ambient_delta**: Arbeitstemperatur relativ zur Umgebung
- **tire_temp_gradient_max**: Maximales thermisches Ungleichgewicht (max - min Temperaturen)

Diese Transformation reduziert die Dimensionalität bei gleichzeitigem Erhalt der relevanten Information und kann die Modellgeneralisierung verbessern.⁴⁴ Um die Auswirkung dieser Dimensionsreduktion auf die Modellperformance empirisch zu evaluieren, wurden zwei Feature-Konfigurationen erstellt: (i) Datensätze mit aggregierten Features bei gleichzeitigem Entfernen

⁴²Vgl. Tsanas 2022

⁴³Vgl. Farek, Benaïdja 2024

⁴⁴Vgl. Tsanas 2022

der hochkorrelierten Original-Features sowie (ii) Datensätze mit aggregierten Features bei Beibehaltung aller Original-Features. Diese experimentelle Designentscheidung ermöglicht eine systematische Bewertung des Trade-offs zwischen Dimensionalität und Informationsgehalt in der späteren Evaluationsphase.

Der Datensatz enthält zudem kategoriale Features wie die Traktionskontroll-Settings und weitere diskrete Variablen (siehe Tabelle 1). Für die Track-Variable wurde eine ordinale Kodierung vorgenommen: Die alphabetisch sortierten Streckennamen wurden numerischen Codes zugeordnet (TrackCode). Diese Zuordnung wurde in einer separaten Mapping-Datei dokumentiert, um die Rückverfolgbarkeit zu gewährleisten und eine konsistente Kodierung zwischen Trainings- und Validierungsdaten sicherzustellen.⁴⁵

Eine One-Hot-Encodierung kategorialer Features wurde bewusst nicht durchgeführt, da die in dieser Arbeit verwendeten baumbasierten Modelle XGBoost und LightGBM kategoriale Features, welche im nächsten Abschnitt genau behandelt werden, nativ unterstützen.⁴⁶⁴⁷ Diese Algorithmen implementieren spezialisierte Split-Strategien für kategoriale Variablen, die gegenüber One-Hot-Encodierung Vorteile in Bezug auf Speichereffizienz und Modellperformance bieten.⁴⁸

Zur Untersuchung des Einflusses kategorialer Features auf die Modellleistung wurden zusätzlich Datensatzvarianten ohne kategoriale Features erstellt. Diese Designentscheidung folgt dem Prinzip der systematischen Evaluation multipler Artefaktvarianten in der Build-Phase der DSR-Methodik.

4.2.4 Zielvariablen-Glättung als experimentelle Designvariante

Trotz der rundenmittelnd aggregierten Zielvariable `aUndersteer_AVG` zeigt deren zeitlicher Verlauf (Abbildung 1) eine hohe Volatilität. Diese kurzfristigen Schwankungen können durch situative Faktoren wie Verkehrssituationen, Überholmanöver oder kurzzeitige Setup-Änderungen verursacht werden und erschweren die Identifikation längerfristiger Trends im Fahrzeugverhalten.

Zur Adressierung dieser Problematik wurde eine Glättungsstrategie mittels gleitender Durchschnitte (Moving Averages) implementiert. Gleitende Durchschnitte sind eine etablierte Methode zur Rauschreduktion in Zeitreihendaten und werden häufig im Feature Engineering eingesetzt.⁴⁹ Die Methode berechnet für jede Runde den Durchschnitt über ein Fenster von n benachbarten Runden, wodurch kurzfristige Fluktuationen gedämpft werden.

Um die optimale Fenstergröße zu ermitteln, wurden mehrere Glättungsvarianten mit unterschiedlichen Fenstergrößen erstellt: keine Glättung (Baseline), sowie gleitende Durchschnitte mit Fenstergrößen 2, 3 und 4.

⁴⁵Vgl. Chen, Guestrin 2016a

⁴⁶Vgl. Chen, Guestrin 2016a

⁴⁷Vgl. Ke, Meng, Qi et al. 2017

⁴⁸Vgl. Chen, Guestrin 2016a

⁴⁹Vgl. Brownlee 2020

Die Wahl dieser Fenstergrößen basiert auf folgender Überlegung: Kleinere Fenster (2, 3) erfassen kurzfristige Schwankungen und erhalten mehr Details, während größere Fenster (4) stärker glätten und langfristige Trends betonen.⁵⁰ Die ungeglättete Variante dient als Referenz zur Quantifizierung des Effekts der Glättung auf die Modellperformance. Diese multiplen Glättungsvarianten stellen alternative Designentscheidungen dar, die im Rahmen der iterativen Build-Evaluate-Phasen der DSR-Methodik systematisch evaluiert werden. Die Erstellung mehrerer Varianten ermöglicht eine empirische Bewertung des Trade-offs zwischen Rauschreduktion und Informationsverlust.

4.2.5 Datensatz-Aufteilung und Validierungsstrategie

Die Kombination der beschriebenen Vorverarbeitungsoptionen resultiert in einer systematischen Variation von Datensatzkonfigurationen. Die Designentscheidungen umfassen drei Dimensionen:

1. **Kategoriale Features:** mit kategorialen Features vs. ohne kategoriale Features (2 Varianten)
2. **Feature-Reduktion:** aggregierte Features mit Entfernung hochkorrelierter Original-Features vs. aggregierte Features zusätzlich zu Original-Features (2 Varianten)
3. **Zielvariablen-Glättung:** keine Glättung, Fenstergröße 2, 3, 4 (4 Varianten)

Die vollständige Kombination dieser Dimensionen ergibt $2 \times 2 \times 4 = 16$ Trainingsdatensätze. Jeder Datensatz umfasst nach Anwendung aller Filterungsschritte ca. 9 000 Runden (Datenpunkte).

Die Validierung der entwickelten Modelle erfordert separate Validierungsdatensätze, die während des Trainings nicht zugänglich sind. Um die Generalisierungsfähigkeit der Modelle umfassend zu bewerten, wurden zwei komplementäre Validierungsstrategien implementiert:

Event-basierte Validierung (Leave-One-Event-Out): Ein vollständiges Renn-Event wurde vom Trainingsdatensatz separiert und als Validierungsdatensatz reserviert. Diese Strategie prüft die Fähigkeit des Modells, auf eine neue, ungesehene Kombination von Strecke, Wetterbedingungen und Rennsituation zu generalisieren. Der Event-Validierungsdatensatz umfasst ca. 200 Runden.

Zufällige Validierung: Aus dem verbleibenden Datensatz wurden 10 % der Runden zufällig ausgewählt und als zweiter Validierungsdatensatz verwendet. Diese Strategie entspricht der etablierten Praxis des Train-Test-Splits in der Machine-Learning-Literatur und dient der Bewertung der Modellperformance auf typischen, aber ungesehenen Datenpunkten.⁵¹ Der Zufalls-Validierungsdatensatz umfasst ca. 1 000 Runden.

⁵⁰Vgl. ebd.

⁵¹Vgl. Baheti 2021

Für beide Validierungsstrategien wurden Datensätze entsprechend der zwei Feature-Konfigurationen (mit/ohne kategoriale Features) und der zwei Reduktionsstrategien (mit/ohne Entfernung hochkorrelierter Features) erstellt. Dies resultiert in $2 \times 2 \times 2 = 8$ Validierungsdatensätzen. Die zweifache Validierungsstrategie ermöglicht eine differenzierte Robustheitsbewertung: Die Event-basierte Validierung testet die Extrapolationsfähigkeit auf vollständig neue Kontexte, während die zufällige Validierung die Interpolationsfähigkeit innerhalb der Verteilung der Trainingsdaten bewertet. Diese Kombination entspricht best practices im Machine Learning und erhöht die Aussagekraft der Modellbewertung.⁵²

4.2.6 Technische Implementierung

Die beschriebenen Vorverarbeitungsschritte wurden in einem Python-Skript implementiert. Die Pipeline arbeitet strikt deterministisch und verarbeitet die Rohdaten (17735 Runden aus der KQL-Extraktion) in einer klar definierten Sequenz: Zunächst werden die Daten eingelesen und unmittelbar um nicht-repräsentative Out-Laps sowie extrem abweichende Rundenzeiten bereinigt. Anschließend entfernt eine domänenbasierte Schwellwertlogik physikalisch unrealistische Sensorwerte. Darauf folgt das Feature-Engineering, in dessen Rahmen hochkorrelierte Sensorkanäle durch aggregierte, informationsverdichtete Kennwerte ersetzt bzw. ergänzt und kategoriale Merkmale ordinal kodiert werden. Im nächsten Schritt erzeugt die Pipeline alternative Zielvarianten durch optionale Glättung (keine, Fenster 2, 3, 4), wodurch parallele Datensatzkonfigurationen für die spätere Modellselektion entstehen. Abschließend werden zwei Validierungsperspektiven vorbereitet: ein vollständig herausgelöstes Renn-Event (Extrapolation) sowie ein zufälliger Anteil von 10 % der verbleibenden Runden (Interpolation). Aus der vollständigen Kreuzung der Vorverarbeitungsoptionen resultieren so 16 Trainingsdatensätze und 8 korrespondierende Validierungsdatensätze, die konsistent im CSV-Format versioniert abgelegt werden.

Die Implementierung stellt Reproduzierbarkeit durch konsistente Transformation aller Datensatzvarianten sicher. Alle Mapping-Dateien und Konfigurationsparameter wurden dokumentiert und versioniert.

Im Kontext der Design Science Research Methodik stellt dieses Kapitel die Build-Phase der Datenpipeline dar. Die multiplen Datensatzvarianten ermöglichen eine systematische Evaluation der Auswirkungen unterschiedlicher Vorverarbeitungsentscheidungen auf die Modellperformance in der nachfolgenden Evaluate-Phase. Die transparente Dokumentation aller Designentscheidungen mit Rückführung auf EDA-Erkenntnisse, Expertenwissen und theoretische Fundierung erfüllt die Rigor-Anforderungen der DSR-Methodik. Die vorbereiteten Datensätze bilden die Grundlage für die Modellentwicklung und Hyperparameter-Optimierung im nachfolgenden Abschnitt 4.3.

⁵²Vgl. ebd.

4.3 Modelltraining und Hyperparameter-Optimierung

Die Modelltrainingsphase bildet den Kern der Artefakt-Entwicklung und zielt darauf ab, aus den in Abschnitt 4.2 generierten 16 Datensatzstrukturen robuste Vorhersagemodelle abzuleiten. Hierzu werden zwei Gradient-Boosting-Algorithmen – XGBoost und LightGBM – auf vier abgestuften Komplexitätsniveaus mittels systematischer Hyperparameter-Optimierung trainiert und anschließend auf strukturkonsistenten Validierungsdatensätzen evaluiert.

Neben Gradient Boosting Decision Trees (GBDT) wurden lineare Modelle (Linear, Ridge, Lasso), Support Vector Regression (SVR), Random Forest sowie Deep-Learning-Architekturen (MLP, TabNet) als Alternativen geprüft. Lineare Ansätze erfassen komplexe nichtlineare Zusammenhänge der Telemetriedaten nur begrenzt, SVR skaliert bei 9 000 Runden und fehlender nativer Unterstützung kategorialer Merkmale ungünstig.⁵³ Random Forest liefert zwar robuste Baselines, erreicht aber in tabularen Regressionen häufig nicht die Spitzengenauigkeit moderner Boosting-Methoden.⁵⁴ Neuronale Netze zeigen auf mittelgroßen tabularen Datensätzen gegenüber fortgeschrittenem Baum-Boosting Performance-Nachteile.⁵⁵ Aus diesen Gründen fokussiert sich das Artefakt auf Gradient-Boosting-Entscheidungsbäume als günstigen Kompromiss aus Prognosegüte, Interpretierbarkeit und Umsetzbarkeit. Zwei komplementäre Implementierungen desselben Paradigmas werden eingesetzt: XGBoost für Regularisierung und Stabilität, LightGBM für Effizienz und Trainingsgeschwindigkeit.^{56,57} Beide können kategoriale Merkmale direkt verarbeiten und reduzieren so Kodierungsaufwand und Komplexität.⁵⁸ Die Parallelnutzung folgt dem DSR-Prinzip vergleichender Evaluation theoretisch fundierter Artefakt-Alternativen.

Die Trainingspipeline führt für jede der 16 Datensatzstrukturen denselben reproduzierbaren Ablauf aus: (i) Einlesen und Aufteilung in Features/Zielvariable, (ii) systematische Hyperparameter-Suche mittels dreifacher Kreuzvalidierung, (iii) Retraining des besten Konfigurationsprofils auf allen Trainingsdaten zur Maximierung der Vorhersagequalität, (iv) strukturierte Persistierung von Modell, Parametern und Metriken (R^2 , RMSE, MAE). Deterministische Zufallssaaten und parallele Ausführung (scikit-learn-kompatible API) gewährleisten Vergleichbarkeit und Reproduzierbarkeit.⁵⁹

Für die Hyperparameter-Optimierung wird `GridSearchCV` eingesetzt: exhaustive Suche über klar abgegrenzte Parameter-Intervalle, primäre Bewertungsmetrik ist R^2 (direkt interpretierbar für Fachexperten), ergänzt durch Fehlermaße zur Einschätzung der Abweichungsgrößenordnung.^{60,61}

⁵³Vgl. Brown 2021

⁵⁴Vgl. Smith, J., Doe 2022

⁵⁵Vgl. Jones 2022

⁵⁶Vgl. Chen, Guestrin 2016b

⁵⁷Vgl. Ke, Meng, Qiang et al. 2017

⁵⁸Vgl. Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot, Michaël et al. 2011

⁵⁹Vgl. Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot, Michaël et al. 2011

⁶⁰Vgl. Bengio, Bergstra 2012

⁶¹Vgl. Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot, Michaël et al. 2011

Die dreifache Kreuzvalidierung bietet robuste Schätzungen bei vertretbarem Aufwand.

Abgestufte Komplexitätsprofile strukturieren die Suche:

Shallow: $n_estimators$: 50–100, max_depth : 3–4, η : 0.1–0.2.

Medium: $n_estimators$: 100–500, max_depth : 5–9, η : 0.05–0.1.

Deep: $n_estimators$: 300–700, max_depth : 10–12, η : 0.03–0.05.

Very Deep: $n_estimators$: 400–800, max_depth : 10–15, η : 0.02–0.03.

Zusätzlich variieren `subsample` (0.8/1.0), `colsample_bytree` (0.8/1.0) bzw. `feature_fraction`, `min_child_weight`/`min_child_samples`.⁶²

Objectives und Loss-Funktion: Beide Verfahren minimieren eine quadratische Fehlerfunktion (XGBoost: `reg:squarederror`, LightGBM: `regression_l2`).⁶³⁶⁴ L2 ist konsistent mit $R^2 = 1 - \frac{SSE}{SST}$, da eine Reduktion der Residuen direkt zu höherer Erklärungsvarianz führt. Gewählt wurde L2 aufgrund (i) Etabliertheit in tabularer Regression, (ii) stabiler Optimierungseigenschaften (glatte Gradienten), (iii) direkter Interpretierbarkeit ergänzender Fehlerkennzahlen (RMSE/-MAE). Alternativen wie Huber- oder Quantile-Loss wurden nicht priorisiert, weil potenzielle Ausreißer bereits vorab bereinigt wurden und zusätzliche Komplexität ohne klaren Mehrwert vermieden wird.

Aus der Kreuzung von 16 Datensatzstrukturen, zwei Boosting-Implementierungen und vier Komplexitätsstufen entstehen **128 trainierte Modelle**.

4.4 Validierung und Modellvergleich

Die abschließende Validierung erfolgt in einem dedizierten Jupyter-Notebook, das die finalen Modelle auf einem zuvor ungesehenen Validierungsdatensatz evaluiert und vergleichbar macht. Dafür wird jedes Modell einzeln geladen und auf beiden Datensätzen (Event und Zufalls-Datensatz) evaluiert. Das Ergebnis sind insgesamt **256 Evaluationsergebnisse** die aus den Evaluationsmetriken R^2 , RMSE und MAE, für jedes Modell bestehen. Dadurch wird transparent, welche Modelltyp-/Feature-Kombination auf neuen, ungesehenen Telemetriedaten am besten generalisiert. Die Generalisierungsfähigkeit ist die Fähigkeit eines trainierten Modells, genaue Vorhersagen für neue, ungesehene Daten zu treffen⁶⁵.

⁶²Vgl. Ke, Meng, Qiang et al. 2017

⁶³Vgl. Chen, Guestrin 2016b

⁶⁴Vgl. Ke, Meng, Qiang et al. 2017

⁶⁵Vgl. Vapnik, Vladimir N. 2013. The Nature of Statistical Learning Theory. 2. Aufl. New York Springer, S. 15-28

5 Evaluation und Interpretation des Vorhersagemodells

Die vorangegangenen Kapitel dokumentierten die systematische Entwicklung von 128 Modellkonfigurationen basierend auf 16 Datensatzstrukturen nach der Design Science Research Methodik. Das vorliegende Kapitel bildet die Evaluate- und Reflect-Phase des Design Cycle und wertet die Modellperformance systematisch aus sowie interpretiert die Ursachen identifizierter Leistungsgrenzen. Im Kontext der DSR-Methodik müssen drei zentrale Dimensionen nachgewiesen werden: *Utility* (der praktische Nutzen des Modells), *Quality* (die Robustheit und Zuverlässigkeit) sowie *Efficacy* (die Problemlösung und Anforderungserfüllung)⁶⁶.

Die Evaluation erfolgt primär als artificial Evaluation durch quantitative Metriken auf strukturkonsistenten Validierungsdatensätzen⁶⁷. Eine naturalistic Evaluation durch Expertenfeedback wird bewusst nicht in diesem Kapitel durchgeführt, da die Modellperformance auf Event-Validierungsdatensätzen bereits zeigt, dass das Artefakt nicht für produktiven Einsatz geeignet ist. Stattdessen fokussiert die Analyse auf die technische Ursachenforschung, um generalisierbares Design Knowledge für zukünftige ML-Projekte im Motorsport abzuleiten. Die anschließende Interpretation der Evaluationsergebnisse identifiziert Wurzelursachen beobachteter Leistungsgrenzen und schließt damit den Rigor Cycle der DSR-Methodik.

5.1 Evaluationskonzept und -methodik

Die Evaluationsstrategie folgt dem FEDS-Framework und adressiert vier zentrale Fragen: **Warum** erfolgt die Evaluation (summativ: abschließende Qualitätsbewertung des Artefakts), **wann** (ex-post nach Modelltraining und -optimierung), **wie** (kombiniert quantitativ und qualitativ, naturalistisch im Praxiskontext) und **was** wird evaluiert (Prädiktionsgenauigkeit, algorithmische Eigenschaften, Datensatzstruktur-Effekte)⁶⁸.

Für alle 256 Evaluationsergebnisse (128 Modelle \times 2 Validierungsstrategien je Datensatzstruktur) werden die standardisierten Regressionsmetriken R^2 , MAE und RMSE verwendet, deren Definitionen bereits in Kapitel 2 erläutert wurden.

Diese drei Metriken folgen etablierten Standards in der Machine-Learning-Literatur⁶⁹.

Die Evaluation untersucht systematisch (i) die Prädiktionsgenauigkeit auf beiden Validierungsdatensatz-Typen (Generalisierung vs. Robustheit), (ii) den Vergleich zwischen LightGBM und XGBoost, (iii) die Effekte der vier Hyperparameter-Komplexitätsstufen und (iv) die Auswirkungen von Glättung, Feature-Reduktion und kategorialen Features auf die Performance.

⁶⁶Vgl. Venable, Pries-Heje, Baskerville 2016, S. 79

⁶⁷Vgl. Venable, Pries-Heje, Baskerville 2016, S. 80-81

⁶⁸Vgl. Venable, Pries-Heje, Baskerville 2016, S. 77-89

⁶⁹Vgl. Hodson 2022, S. 5481-5482; vgl. ebenso Willmott, Matsuura 2005, S. 79-80

5.2 Quantitative Leistungsanalyse

Die Evaluation der 128 trainierten Modelle auf zwei strukturkonsistenten Validierungsdatensätzen zeigt eine stark zweigeteilte Leistungslandschaft, die erhebliche Implikationen für die praktische Einsatzfähigkeit des Artefakts hat. Im Folgenden werden zunächst die positiven Ergebnisse auf dem Zufalls-Validierungsdatensatz dargestellt, anschließend die kritischen Befunde der Event-basierten Validierung erörtert und abschließend die Generalisierungsproblematik analysiert. Die kompletten Ergebnisse der Validierung sind im Anhang 2 dokumentiert.

5.2.1 Performance auf Zufalls-Validierungsdatensätzen

Die Zufalls-Validierung gibt Aufschluss über die Modellqualität auf typischen, aber ungesesehenen Daten innerhalb der trainierten Verteilung. Die Ergebnisse sind substantiell positiv und zeigen, dass das entwickelte Artefakt unter Standardbedingungen akzeptable Vorhersagefähigkeit aufweist. Durchschnittlich erreicht das Modellportfolio eine R^2 von 0.308 (± 0.197), mit einer Range von -0.338 bis 0.657. Dies bedeutet, dass das durchschnittliche Modell etwa ein Drittel der Varianz in der Zielvariable erklären kann. Die mittlere RMSE-Abweichung liegt bei 0.328 (± 0.049), was einer durchschnittlichen Vorhersageabweichung von etwa 0.33 Understeer-Einheiten entspricht.⁷⁰ Damit liegen die Modelle in der praktischen Größenordnung, die für Engineering-Fragestellungen relevant ist. XGBoost dominiert deutlich die Random-Validierungsperformance mit einem Durchschnitts- R^2 von 0.380 (± 0.235), während LightGBM mit 0.237 (± 0.112) deutlich dahinter liegt. XGBoost zeigt auch überlegene Stabilität mit geringerer Standardabweichung.

Algorithmus	R^2 (Mittel)	MAE (Mittel)	RMSE (Mittel)	Anzahl
XGBoost	0.380	0.239	0.308	64
LightGBM	0.237	0.278	0.347	64
Vorteil XGB	+0.143	-0.038	-0.039	—

Tab. 3: Algorithmus-Vergleich: Random-Validierung (alle Komplexitätsstufen)

Die beste Modell-Konfiguration in der Random-Validierung ist ein XGBoost-Modell mit Very Deep-Komplexität, ohne kategoriale Features, mit Feature-Aggregation und ohne Zielvariablen-Glättung. Dieses Modell erreicht ein R^2 von 0,657, RMSE von 0,233 und MAE von 0,176. Dies deutet darauf hin, dass XGBoost unter Bedingungen mit ausreichender Komplexität und geeigneter Feature-Konfiguration starke Vorhersagefähigkeit entwickelt. Bemerkenswerterweise ist die mittlere Komplexität nicht universell optimal. Medium und Very Deep erzielen ähnlich gute Durchschnittswerte (R^2 von 0.336 bzw. 0.328), während Shallow mit 0.245 deutlich schlechter abschneidet. Dies deutet darauf hin, dass eine Mindest-Modellkapazität erforderlich ist, dass aber zu tiefe Modelle auf Random-Daten nicht zusätzlich helfen.

⁷⁰Vgl. Hodson 2022, S. 5481-5482

Konfiguration	R^2 (Mittel)	MAE (Mittel)	RMSE (Mittel)	Anzahl
Shallow	0.245	0.220	0.343	32
Medium	0.336	0.183	0.322	32
Deep	0.325	0.191	0.324	32
Very Deep	0.328	0.186	0.323	32

Tab. 4: Random-Validierung: Performance nach Hyperparameter-Komplexitätsstufe

Die Glättungsvarianten zeigen eine optimale Performance bei Fenstergrößen von 2–3, mit Fenstergröße 3 leicht vorne (R^2 0.324). Fenstergröße 4 verschlechtert die Performance (R^2 0.297). Feature-Aggregation liefert einen konsistenten, wenn auch moderaten Vorteil (+0.024 R^2). Überraschenderweise profitieren Random-Validierungsergebnisse deutlich von der Abwesenheit kategorialer Features: Modelle ohne kategoriale Features erreichen R^2 0.466 gegenüber R^2 0.150 mit kategorialen Features eine Differenz von 0.316. Dies ist ein starker Indikator für Overfitting auf kategoriale Variablen.

Konfiguration	R^2 (Mittel)	Varianz	RMSE (Mittel)	Modelle
Kategoriale Features:				
Ohne kategorische Features	0.466	0.195	0.289	64
Mit kategorialen Features	0.150	0.080	0.367	64
Feature-Aggregation:				
Ohne Aggregation	0.296	0.183	0.331	64
Mit Aggregation	0.320	0.210	0.325	64
Glättung (Fenstergrößen):				
Fenster 0 (keine)	0.309	0.195	0.328	32
Fenster 2	0.304	0.215	0.329	32
Fenster 3	0.324	0.186	0.325	32
Fenster 4	0.297	0.197	0.331	32

Tab. 5: Random-Validierung: Effekte der Datensatzstruktur-Varianten

Auf dem Zufalls-Validierungsdatensatz demonstriert das beste Modell (XGBoost Very Deep, $R^2 = 0.657$) starke Vorhersagefähigkeit. Der Durchschnitt über alle Modelle liegt bei akzeptablen R^2 0.308. Diese Befunde suggerieren, dass das entwickelte Artefakt unter bekannten Datenverteilungen verlässliche Prognosen liefert.

5.2.2 Performance auf Event-Validierungsdatensätzen

Die Event-basierte Validierung wendet das trainierte Modell auf ein vollständig unbekanntes Renn-Event an und prüft damit die echte Generalisierungsfähigkeit auf neue Rennkontexte, Fahrzeugkonfigurationen und Streckeneigenschaften. Die Befunde in diesem Szenario sind fundamental kritisch. Die durchschnittliche R^2 beträgt -0.306 (± 0.256), eine deutlich negative Zahl.

Dies bedeutet, dass das durchschnittliche Modell schlechter abschneidet als eine triviale Baseline (z.B. Mittelwert-Vorhersage). Die Range erstreckt sich von -1.135 bis 0.093 , wobei 25 von 128 Modellen ein R^2 unter -0.5 aufweisen, ein Indikator extremer Fehlvorhersagen. Zusätzlich beträgt die RMSE durchschnittlich $0.325 (\pm 0.032)$, was zwar der Random-Validierung ähnelt, aber auf Grund der negativen R^2 -Werte nicht aussagekräftig ist. LightGBM zeigt eine überlegene Event-Generalisierbarkeit mit $R^2 -0.268 (\pm 0.277)$ gegenüber XGBoost mit $-0.343 (\pm 0.229)$. Der beste LightGBM-Event-Score ($R^2 0.093$) ist deutlich höher als der beste XGBoost-Event-Score ($R^2 0.029$).

Algorithmus	R^2 (Mittel)	R^2 (Best)	RMSE (Mittel)	Modelle
LightGBM	-0.268	0.093	0.320	64
XGBoost	-0.343	0.029	0.330	64
Vorteil LGB	+0.075	+0.064	-0.010	–

Tab. 6: Algorithmus-Vergleich: Event-Validierung (Leave-One-Event-Out)

Die beste Modell-Konfiguration in der Event-Validierung ist ein LightGBM-Modell mit Very Deep-Komplexität, mit kategorialen Features, ohne Feature-Aggregation und ohne Zielvariablen-Glättung. Dieses Modell erreicht $R^2 0.093$, RMSE 0.272 und MAE 0.212. Auch diese beste Konfiguration bleibt problematisch niedrig.

Tiefere Modelle zeigen relativ bessere Event-Generalisierung: Very Deep ($R^2 -0.232$) übertrifft Shallow ($R^2 -0.405$) um 0.173 Punkte. Dies deutet darauf hin, dass höhere Modellkomplexität das Generalisierungsproblem nicht verschärft. Allerdings sind **alle Komplexitätsstufen im absoluten Sinne unzureichend**. Der beobachtete Trend sollte daher nicht als Beleg für robustere Modelle interpretiert werden, sondern lediglich als Hinweis, dass Overfitting durch Komplexität in diesem Fall nicht der limitierende Faktor ist.

Komplexitätsstufe	R^2 (Mittel)	Varianz	RMSE (Mittel)	Modelle
Shallow	-0.405	0.317	0.337	32
Medium	-0.318	0.233	0.327	32
Deep	-0.267	0.235	0.321	32
Very Deep	-0.232	0.204	0.317	32

Tab. 7: Event-Validierung: Performance nach Komplexitätsstufe (Trend zu besserer Generalisierung)

Bei der Event-Generalisierung zeigt Glättung (Fenster 3: $R^2 -0.281$) einen marginalen Vorteil gegenüber keiner Glättung ($R^2 -0.330$). Kategoriale Features sind ambivalent. Sie verbessern Event-Generalisierung erheblich (mit: $R^2 -0.140$ vs. ohne: $R^2 -0.471$, Delta- $R^2 +0.331$), schaden aber massiv in der Random-Validierung (mit: $R^2 0.150$ vs. ohne: $R^2 0.466$, Delta- $R^2 -0.316$).

Konfiguration	R^2 (Mittel)	Varianz	RMSE (Mittel)	Modelle
Kategoriale Features:				
Mit kategorialen Features	-0.140	0.207	0.304	64
Ohne kategorische Features	-0.471	0.184	0.346	64
Feature-Aggregation:				
Mit Aggregation	-0.300	0.248	0.325	64
Ohne Aggregation	-0.311	0.266	0.326	64
Glättung (Fenstergrößen):				
Fenster 0 (keine)	-0.330	0.316	0.328	32
Fenster 2	-0.296	0.259	0.324	32
Fenster 3	-0.281	0.219	0.323	32
Fenster 4	-0.315	0.229	0.327	32

Tab. 8: Event-Validierung: Effekte der Datensatzstruktur-Varianten

Die Event-Validierung offenbart ein fundamentales Generalisierungsproblem. Mit durchschnittlich negativer R^2 sind die Modelle in diesem Szenario nicht praktisch einsetzbar. Das beste Modell mit R^2 0.093 ist gerade noch marginale besser als eine Baseline.

5.2.3 Interpretation des Gesamtergebnisses

Die massive Diskrepanz zwischen Random- (R^2 -Mittel 0.308) und Event-Validierung (R^2 -Mittel -0.306) ist nicht auf typisches Overfitting zurückzuführen, sondern offenbart ein fundamentales Problem der Domänen-Generalisierung. Im Folgenden werden die wahrscheinlichen Ursachen und Implikationen dieser Diskrepanz erörtert.

Ein zentrales Problem besteht darin, dass die Telemetriedaten ausschließlich Fahrzeugzustände und Sensorwerte erfassen, jedoch die entscheidenden Kontextfaktoren, die das Understeer maßgeblich beeinflussen, nicht berücksichtigen. So sind beispielsweise individuelle Fahrereffekte wie Lenk-Aggression, Bremspunkt-Variabilität oder die Wahl der Fahrlinie in den Rohdaten mit physikalischen Fahrzeug-Eigenschaften vermischt und lassen sich nicht voneinander trennen. Das Modell erlernt dadurch eine Kombination aus Fahrzeugverhalten und Fahrercharakteristik, die bei neuen Fahrern nicht übertragbar ist. Hinzu kommt, dass das Fahrzeug-Setup, etwa Chassis-Steifigkeit, Aero-Balance oder Federung, sich zwischen den Events und sogar innerhalb eines Events verändert. Diese Parameter sind in den Telemetriedaten nicht explizit enthalten und können allenfalls indirekt über die Reaktion des Fahrzeugs auf die Fahrbahn abgeleitet werden, was jedoch eine unzureichende und wenig robuste Methode darstellt. Darüber hinaus spielen Umgebungsfaktoren wie Streckentemperatur, Luftdichte, Feuchtigkeit und Windverhältnisse eine fundamentale Rolle für Aerodynamik und Reifenverhalten. Da diese Einflussgrößen im Datensatz nicht enthalten sind, können sie auch nicht rekonstruiert werden. Insgesamt erklärt das Fehlen dieser Kontextfaktoren, weshalb das Modell zwar auf bekannten Daten eine hohe Performance erzielt, jedoch beim Transfer auf neue Events deutlich an Vorhersagekraft verliert.

Diese fehlenden Faktoren erklären, warum das Modell eine hohe Random-Validierungs-Performance erreicht (es memorisiert Training-Event-Muster) aber bei Event-Transfer kollabiert (neue Kombinationen aus Fahrer, Setup, Reifen, Umwelt sind unbekannt).

Das zentrale Problem ist ein Covariate-Shift: Die Verteilung der Eingabedaten unterscheidet sich zwischen Training und Event-Test, während das Modell unter der Annahme konstanter Kontextfaktoren trainiert wurde. Bei neuen Fahrern, Setups oder Umwelteinflüssen versagt diese Annahme, und die Generalisierung bricht ein. Der Event-Validierungsdatensatz dient somit als Stress-Test für echte Domain Generalization. Für zukünftige ML-Projekte gilt: Die Datenqualität ist entscheidender als die Wahl des Algorithmus, und die Verwendung kategorialer Features sollte mit Blick auf die Ziel-Domäne erfolgen. Robustere Generalisierung erfordert gezieltes Feature-Engineering und Transfer-Learning-Ansätze, da Modelle ohne explizite Berücksichtigung neuer Kontexte nicht zuverlässig auf unbekannte Strecken übertragbar sind.

6 Fazit, Erkenntnisse und Forschungsausblick

6.1 Erfüllung der Anforderungen und Beantwortung der Forschungsfragen

Die vorliegende Arbeit verfolgte das Ziel, ein Machine-Learning-Modell zur automatisierten Vorhersage von Understeer-Verhalten in Motorsport-Telemetriedaten zu entwickeln. Zur Bewertung des Erfolgs werden sowohl die technischen Anforderungen als auch die formulierten Forschungsfragen gegen die erhaltenen Ergebnisse abgewogen.

Anforderung 1: Automatisierte Understeer-Vorhersage

Die Anforderung, ein ML-Modell zur Vorhersage von Understeer-Werten auf Basis von Telemetriedaten zu entwickeln, ist partiell erfüllt. Das beste Modell auf dem Random-Validierungsdatensatz (XGBoost, Very Deep, $R^2 = 0.657$) demonstriert, dass Understeer-Vorhersage unter kontrollierten Bedingungen möglich ist. Die durchschnittliche Random-Performance über alle 128 Modelle beträgt $R^2 = 0.308$, was darauf hindeutet, dass die Modelle etwa ein Drittel der Varianz in der Zielvariablen erfassen. Allerdings ist diese Performance auf die spezifische Datenverteilung des Trainingsdatensatzes beschränkt. Bei der Event-Validierung, dem realistischeren Szenario mit vollständig unbekannten Rennkontexten, kollabiert die Performance dramatisch. Das beste Event-Modell (LightGBM, Very Deep, $R^2 = 0.093$) erreicht nur marginal bessere Ergebnisse als eine Baseline-Vorhersage, und der Durchschnitt über alle Modelle beträgt $R^2 = -0.306$, was bedeutet, dass die Modelle schlechter abschneiden als triviale Vergleichsmodelle.⁷¹ Diese Diskrepanz offenbart, dass das Ziel einer produktionsreifen, generalisierbaren Vorhersage nicht erreicht wurde.

Anforderung 2: Hohe Vorhersagegenauigkeit

Die angestrebte Vorhersagegenauigkeit von $R^2 > 0,7$ wurde nicht erreicht. Das beste Random-Modell erreicht $R^2 = 0,657$, knapp unterhalb der Zielvorgabe, scheitert aber bei Event-Generalisierung ($R^2 = 0,093$). Die Ursachenanalyse in Kapitel 5.3 identifiziert als Primärproblem nicht algorithmische Limitationen, sondern fehlende Kontextfaktoren: Fahrercharakteristiken, Setup-Parameter, Reifen-Degradation und Umgebungsdaten sind in den verfügbaren Telemetriedaten nicht erfasst und können nicht aus Sensorwerten rekonstruiert werden.⁷² Diese fehlenden Variablen determinieren das Understeer-Verhalten fundamental und liefern eine mögliche Erklärung für die schwache Event-Generalisierung.

⁷¹Vgl. Kap. 5.2

⁷²Vgl. Kap. 5.3

6.2 Beantwortung der Forschungsfragen

Forschungsfrage 1: Können Gradient Boosting Decision Trees Understeer-Verhalten in Motorsport-Telemetriedaten vorhersagen?

Die Antwort lautet: Teilweise ja, aber nur unter stark einschränkenden Bedingungen. GBDT-Modelle funktionieren auf trainierten Datensätzen gut (durchschnittliche Random-Validierung: $R^2 = 0,308$) und zeigen damit, dass Understeer grundsätzlich aus Telemetriedaten extrahierbar ist. Allerdings generalisieren diese Modelle nicht auf unbekannte Rennkontexte (durchschnittliche Event-Validierung: $R^2 = -0,306$). Das Problem liegt nicht in der Algorithmuswahl, sondern in der fehlenden Datengrundlage. GBDT memorisieren Muster aus trainierten Events (Track-spezifische Telemetrie-Signaturen), können diese aber nicht auf neue Tracks, neue Fahrer oder neue Setup-Konfigurationen übertragen.⁷³

Forschungsfrage 2: Welcher Algorithmus (XGBoost vs. LightGBM) generalisiert besser?

Weder XGBoost noch LightGBM zeigt konsistente Überlegenheit. Bei Event-Generalisierung ist LightGBM leicht besser (durchschnittliche $R^2 = -0,268$ vs. XGBoost $-0,343$), mit dem besten Event-Modell bei $R^2 = 0,093$ (LightGBM) vs. $R^2 = 0,029$ (XGBoost). Bei Random-Validierung dominiert XGBoost deutlich (durchschnittliche $R^2 = 0,380$ vs. LightGBM $0,237$), mit dem besten Modell bei $R^2 = 0,657$ (XGBoost) vs. $R^2 = 0,458$ (LightGBM).⁷⁴ Diese gegensätzliche Leistung unterstreicht, dass algorithmische Wahl sekundär ist. Beide Algorithmen leiden unter denselben fundamentalen Limitationen: fehlende Kontextfaktoren.

Forschungsfrage 3: Wie wirken sich Datenvorbereitung und Hyperparameter-Tuning auf Performance aus?

Der größte Effekt stammt nicht aus Hyperparameter-Tuning, sondern aus der Feature-Konfiguration, insbesondere der Verwendung kategorialer Features. In der Event-Validierung helfen kategoriale Features (primär Track-Information) um $\Delta R^2 = +0,331$ (mit Kategorisch: $R^2 = -0,140$ vs. ohne: $R^2 = -0,471$). In der Random-Validierung schaden sie um $\Delta R^2 = -0,316$ (mit Kategorisch: $R^2 = 0,150$ vs. ohne: $R^2 = 0,466$). Diese gegensätzliche Wirkung erklärt sich dadurch, dass Track-Variablen als Proxy für nicht erfasste Event-Charakteristiken fungieren, nicht als echte Kausalvariablen.⁷⁵ Glättungsvarianten zeigen moderate Effekte (optimal: Fenster 3, ΔR^2 ca. $0,03$ – $0,05$), und Feature-Aggregation hat minimalen Effekt ($\Delta R^2 < 0,03$). Hyperparameter-Komplexitätsstufen zeigen, dass tiefere Modelle bei Event-Generalisierung helfen (Very Deep besser als Shallow um $\Delta R^2 = +0,173$), was gegen klassisches Overfitting-Verständnis spricht und eher auf Domain-Shift-Probleme hindeutet.⁷⁶

⁷³Vgl. Kap. 5.3

⁷⁴Vgl. Kap. 5.2

⁷⁵Vgl. Kap. 5.3

⁷⁶Vgl. Kap. 5.2

6.3 Design Knowledge und kritische Selbsteinschätzung

Bedauernswerterweise konnte das Ziel, ein ML-Modell zur Vorhersage der Fahrzeugbalance zu entwickeln, nicht erreicht werden. Trotz intensiver Bemühungen in der Datensammlung (17.735 Runden aus 40 Events), Datenvorbereitung (16 Strukturvarianten) und Modellierung (128 trainierte Modelle) blieb die angestrebte Prognosegenauigkeit aus, sodass diese Arbeit kein in der Praxis nutzbares Artefakt hervorgebracht hat. Im Normalfall würde man nun mehrere Iterationen im Design Cycle durchlaufen, was im Rahmen dieser Arbeit aufgrund von Zeit- und Ressourcenbeschränkungen nicht möglich war.⁷⁷

Aus Sicht der Design Science Research Methodologie ist die Evaluate-Phase vollständig dokumentiert: Der Rigor Cycle wurde mit systematischen Evaluationen, etablierten Metriken und 256 Evaluationsergebnissen erfüllt.⁷⁸ Der Relevance Cycle wurde teilweise erfüllt und die zugrunde liegenden Probleme aufgezeigt, die Lösung erfordert aber Daten, die außerhalb des Projektumfangs liegen.

Aus den Erkenntnissen des Entwicklungsprozesses ergeben sich folgende Design Principles für zukünftige Arbeiten: Die zentralen Erkenntnisse aus dem Entwicklungsprozess lassen sich wie folgt zusammenfassen: Die algorithmische Wahl (XGBoost vs. LightGBM) ist für die Vorhersagegenauigkeit nachrangig; entscheidend ist die Datenqualität und die Berücksichtigung relevanter Kontextfaktoren. Kategoriale Features wie Track-Informationen verbessern die Modellleistung innerhalb bekannter Datenbereiche, verschlechtern jedoch die Generalisierung auf neue Events, da sie lediglich Trainingsmuster memorisieren. Die Analyse zeigt, dass Domain Shift, also Unterschiede in der Verteilung der Eingabedaten zwischen Trainings- und Testevents, größere Auswirkungen hat als klassisches Overfitting; tiefere Modelle können unter diesen Bedingungen sogar besser generalisieren. Event-basierte Validierung ist daher essenziell, da sie die tatsächliche Generalisierungsfähigkeit offenbart und Random-Validierung zu optimistischen Einschätzungen führt. Insgesamt bleibt die fehlende Erfassung von Kontextfaktoren wie Fahrercharakteristiken, Setup-Parametern, Reifenstatus und Umgebungsbedingungen die zentrale Limitation, die mit reiner Telemetrie nicht überwunden werden kann.

Was dennoch bleibt sind die Erkenntnisse aus dem Entwicklungsprozess, die wertvolle Einblicke in die Herausforderungen und Limitationen bei der Anwendung von ML im Motorsport-Kontext bieten und einen Grundstein für zukünftige Arbeiten legen.

⁷⁷Vgl. Hevner et al. 2004, S. 83

⁷⁸Vgl. Venable, Pries-Heje, Baskerville 2016, S. 77-89

Anhang

Anhangverzeichnis

Anhang 1	Rohtranskripte der Experteninterviews	34
Anhang 1/1	Erstes Meeting mit Performance-Ingenieur (29.08.2025)	34
Anhang 1/2	Zweites Meeting mit Performance-Ingenieur (12.09.2025)	40
Anhang 2	Modell-Ergebnisse	49
Anhang 2/1	Ergebnisse der Event-Validierung	49
Anhang 2/2	Ergebnisse der Zufalls-Validierung	52

Anhang 1: Rohtranskripte der Experteninterviews

Anhang 1/1: Erstes Meeting mit Performance-Ingenieur (29.08.2025)

1 Interviewer: Meeting mit Ingenieur, 29.08.2025 11 Uhr.

2 Interviewer: Ja, freut mich, vielen Dank für deine Zeit. Jetzt wollte ich dich fragen, ob du dich
3 ganz kurz vorstellen könntest.

4 Ingenieur: So ja, kann ich kurz machen, ich bin schon länger bei Porsche. Teamleiter von der
5 LMDH Performance Truppe. seit LMP Formel E. Jetzt LMDh, was wir machen, Performance, vor
6 allem bei der Entwicklung, Simulationslastik, Kennwerte vorgeben für die anderen Abteilungen,
7 auch, der Drag brauchen wir, wie viel Abtrieb brauchen auf dem Auto. Auch für die Reifen-
8 entwicklung mit Michelin zusammen, Rundenzeitberechnung machen wir, auch die Modellierung
9 dazu. Und eben wenn des Auto mal fährt, Datenanalyse. Aber auch im Teil der performancere-
10levanten Software im Auto kommt auch von uns. Also, Funktionskontrolle.

11 Interviewer: Okay, sehr cool. Vielleicht stelle ich jetzt noch mal ganz kurz das vor, was wir
12 vorhaben. Ich bin jetzt wie gesagt bei EMO 6 für nächsten drei Monate und muss von meiner
13 Hochschule aus eine wissenschaftliche Arbeit schreiben über das Projekt, das ich mache. Und das
14 Projekt, das ich mache, ist nämlich... Wir wollten, oder das wurde von, das kam von EM6 und von
15 Paul, das Thema hoch, dass man vielleicht mit einem Proof of Concept starten möchte, um zu
16 gucken, gibt es denn Auswertungen, Analysen, die man mittels der, also aus den Telemetriedaten
17 ziehen kann, irgendwelche Aussagen, da rauskommen, die sich, auf die man ein Machine Learning
18 Modell trainieren kann und einfach um mal zu gucken, ist es möglich, es da Daten und einfach
19 mal ein Proof of Concept zu starten. Ja, genau. Das ist jetzt noch relativ am Anfang, deswegen
20 als auch unser Termin. Ich bin neu im Motorsport, ich schaue auch mal zu Formel 1, aber das hilft
21 mir dann jetzt in der Tiefe der Thematik nicht sehr viel weiter. ist eine falsche Serie. Falsche Serie,
22 mich geht es jetzt grundsätzlich mal ganz kurz darum, bisschen das Verständnis zu bekommen.
23 Was macht ein Performance-Ingenieur, also du hast es gerade schon mal kurz angeschnitten,
24 vielleicht auch mal bisschen, also wirklich vielleicht so von oben kommen. Genau, was ist deine
25 Jobrolle oder die Jobrolle des Performance Engineers und dann vielleicht auch eine Einordnung,
26 wie denn sowas an einem Wochenende aussieht, wenn du das vorhin schon gemeint hast, also
27 Datenanalyse, das ist ja da wahrscheinlich eher die Richtung, die für mich eher interessant wäre.
28 Genau, also vielleicht könntest du mal bisschen so top down, mal bisschen grundlegend mal
29 sagen, wie das so abläuft.

30 Ingenieur: Ich kann erst mal starten, es hilft. Performance-Ingenieur, Performance, die sind ver-
31 antwortlich für die Performance vom Auto. Performance heißt, möglichst optimal das Auto auf
32 der Strecke einzusetzen. Optimal bezüglich Rundenzeit logischerweise. Das ist ein Thema. Und
33 das zweite ist, was sie noch machen ist, sie gucken auch, dass das Auto sicher betrieben wird. Das
34 heißt, sie schauen in Betrieb, zum Beispiel in die Bremstemperatur. zu stark ansteigt und kritisch
35 dann heben sie die Hand und sagen, wir haben da Problem. Diskutieren dann mit Renningenieur

zusammen, holt das Auto rein und dann wird es repariert. sie gucken, z.B. wenn die Fahrhöhe vorne zu niedrig ist, anders als erwartet, dann muss man auch die Hand heben und sagen, da passiert irgendwas nicht. Oder wenn das Bremspedal zu lang wird, weil Verschleiß oh zu hoch ist. sind auch so sicherheitsrelevante Themen, was wir anschauen. Idealer Weise tritt das nicht auf. Dann ist Ihre Aufgabe halt Setuparbeit, am Auto, du kannst ja ganz viel einstellen, viel mehr als beim Straßenauto. Du kannst deine Federn, Steifigkeiten ändern. Da gibt es ganz viele Optionen. Es gibt verschiedene Aero-Konfigurationen. Die Fahrwerke kannst du einstellen. verschiedene Federkombinationen in Serie schalten, dass du verschiedene Charakteristiker hast und noch viel mehr. Das ist Ihre Aufgabe, das während des Rennwochenendes bis zum Rennen zu optimieren. Zusammen mit dem Ingenieur und natürlich auch mit dem Fahrer. Der Fahrer fährt ja nicht nur das Auto, nur im Kreis. Sondern nach jeder Änderung kriegt man Feedback von dem Fahrer, ob das ins Gute oder ins Schlecht war. Aus seiner Sicht und das fließt in die Entscheidung ein, was man als nächstes ändert am Auto. Was man ändert am Auto, wie gesagt Fahrradfeedback ist wichtig und eben auch die Daten dazu. Das Telemetrie oder Kabeldaten. WEC hat leider sehr wenig Telemetrie, per Reglement ist das vorgegeben. IMSA hat da mehr, da hat man Probleme, mit der Logging rate. Im Endeffekt, man guckt an Daten, macht die auf, schaut sich die Balance an vom Auto. Balance heißt dass das Auto unter oder übersteuert. du schaust deine Fahrhöhen an, ob das Auto zu tief ist aufgesetzt oder zu hoch ist und du noch Potenzial hast tiefer zu gehen, je tiefer du bist, desto mehr Abtrieb hast du, was damit auch Rundenzeit bringt. Das ist so die Aufgabe von einem Performance Engineer. das Auto mit Renningenieur und Fahrer zu optimieren. Sie sind eine Verantwortung für das Setup vom Auto. Verstehe.

Interviewer: Danke für die Einführung. Das war jetzt auf jeden Fall schon mal sehr hilfreich. Kannst du mir noch einen kurzen Umriss geben, wie dann das Zusammenspiel an dem Wochenende vielleicht aussieht. Ich stelle mir das gerade aktuell so vor, man hat die Fahrt in Teilen links. Es werden Daten gesammelt, die Autos sind auf der Strecke, wie du schon beschrieben hast. Dann gibt es bei euch mehrere Performance Engineers, die sich die Daten angucken und wahrscheinlich auf verschiedene Dinge achten auf Basis ihrer Erfahrungen oder wie das Team eingeteilt ist.

Ingenieur: Der Performance Ingenieur, der hat Tools dazu auch, der simuliert, wenn ich jetzt die Feder ändere, dann ändert das in der Theorie, folgendes an meinem Auto, das ist tiefer, höher, ich sehe auch die Abtriebwerte theoretisch, das gibt mir alles mein Simulationsmodell her. Der gleicht das dann ab mit Streckendaten. Und je nachdem vom Feedback vom Fahrer, wenn der sagt, hey, ich zu viel Untersteuern, dann weiß der Performance-Engineer. Wenn ich an dem Parameter drehe, vom Auto, würde sich das ändern. . Dann würde sich die Verlauf vom Auto in die Richtung ändern. Normalerweise simuliert er das vorhandene Tool mit HH, also das ist unsere Datenbank. kann er sagen ich ende jetzt mein Setup, die Fahrhöhe, dann wird das simuliert und dann kriegt er die Werte aus. Das macht er nicht erst am Wochenende, wo er sich das frei überlegt. Der macht sich davor ganz viele Gedanken, der legt sich so einen Blumenstrauß an Setups zusammen, weil er diskutiert ist schon mit seinem Renningenieur. auch auf Erfahrung von den Jahren davor, was funktioniert hat, was sie da gefahren haben. Wir haben auch den

76 Fahrsimulator, das ist erste Tool. Auf dem Papier mit verschiedenen Setup Optionen. Du fährst
77 dann am Fahrsimulaltor, du suchst jetzt schon mal aus, was gut funktioniert, was nicht so gut
78 funktioniert, runter weg. Und mit den Optionen, Simulation ist eine Simulation, muss man noch
79 validieren an der Strecke, gehst an die Strecke und fährst die eventuell dann einfach gegen.
80 Validierst dann deine Theorie und nimmst dann das Beste ins Rennen mit, beziehungsweise ins
81 Qualifying, z.B. um Unterschiede etc. Das ist der Prozess von der Performance , der definierte
82 Setup. Da es die Mechaniker, die ich ausdrucken soll, die haben ja Tablet. Da sehen Sie, ich muss
83 die Feder ändern, muss hier die Fahrhöhe anheben und dann stelle ich das am Auto ein. Da gibt
84 es auch wieder von der Struktur einen Nr1 mechaniker. Der kriegt die Info und verteilt es dann
85 an sein Team am Auto. Und dann wird das umgesetzt, was der Performance Engineer zusammen
86 mit dem Renngenieur entschieden hat. Genau, dann wird das umgesetzt, dann fahren die, kriegt
87 die Daten und kannst dann sozusagen interativ und etc. über dich finden.

88 Interviewer: Okay, ja, also danke, ja perfekt, danke für den Überblick. Das schärft jetzt ein
89 bisschen mein Bild jetzt auch, weil ich halt eben neu bin in der Thematik, wie das generell funk-
90 tioniert. Genau, aber dann würde ich jetzt mal ein bisschen erklären, was wir, also wie und was
91 wir machen wollen. Also... Der Ansatz ist ziemlich Greenfield. haben dieses Konzept, wir mit
92 Machine Learning, also Modell, auf Telemetriedaten trainieren wollen. Das ist die Baseline. Von
93 hier aus können wir jetzt viele verschiedene Wege gehen. Ich muss nicht in irgendeine Richtung
94 gehen. Ich kann verschiedene Dinge anwenden. Ganz grundlegend, was könnte man im Machine
95 Learning gibt es dann verschiedene Teilbereiche. Der erste Teilbereich sind Klassifikationsmo-
96 delle. Man bekommt ja immer bei einem Machine Learning ganz viele Inputs, zum Beispiel in
97 unserem Fall die verschiedensten Telemetrie-Channels. Das können 1, 2 bis 100. 100 sein, die man
98 dort hineingibt und im Optimalfall kommt hinten ein Output aus. Und dieser Output kann je
99 nachdem was man möchte, kann ja verschieden sein. Da gibt es zum einen das Klassifikationsmo-
100 dell, das ist das erste, dann kommt hinten, man gibt seine Telemetriedaten rein und dann kommt
101 zum Beispiel hinten Ich habe mir paar Beispiele ausgesucht auf Motorsport, unabhängig davon,
102 ob das Sinn macht. Zum Verständnis ist, man z.B. in der TdMT-Daten von den Trainings rein-
103 haut und hinten kommt dann eine Reifenmischung aus, was vielleicht Sinn macht, Soft, Medium,
104 Hard. Also eine Klasse, Soft, Medium oder Hard. Sowas in die Richtung. Das ist ein Klassifi-
105 kationsmodell. man sagt, Fahrverhalten klassifizieren. wie schnell wird Gaspedal, Bremspedal,
106 wie schnell wird am Lenkrad gedreht und hinten kommt aus, dass es aggressives Fahrverhalten,
107 mittleres Fahrverhalten oder ähm... optimales Fahrverhalten. Also in die Richtung kann man das
108 machen. Man kann die Output natürlich selber definieren. wäre das erste. Das zweite, was event-
109 uell interessant ist für das, was wir machen wollen, sind die Vorhersage- oder Regressionsmodelle.
110 Das heißt, man trainiert das Modell mit vielen verschiedenen Telemetriedaten, die relevant sein
111 könnten, um einen Wert, hätte, vorhersagen. Also zum Beispiel in der Rundenzeit vorhersage ich.
112 gibt dem Modell ganz ganz viele verschiedene Telemetriedaten, es aktuell ist. Und dann fahre ich
113 meinen ersten, zweiten Sektor, dritten Sektor und es wird immer aktualisiert, zum Beispiel die
114 prognostizierte Endrundenzeit vorhergesagt zum Beispiel. Das wäre was für ein Regensohnsmo-
115 dell. Also ich will kurz einfach vermitteln, wie das funktionieren könnte. Man mappt von ganz
116 vielen verschiedenen Inputs auf einen Output. Wie der Output aussieht, können verschiedene

117 Sachen sein. die Sachen wären so die Hauptkategorien, die ich mich fokussieren würde. Also man
118 kann einmal eine Klassifikation, also es kommt dann ein String aus, also kein numerischer Wert
119 oder Regression, man versucht wirklich numerische Werte vorherzusagen auf Basis von anderen
120 Parametern. Genau. Das ist so das, was es in der Theorie kann. Was für das Machine Learning
121 Modell wichtig ist, ist, mit was für Daten es trainiert wird. Für das Training braucht man beide
122 Enden von dem. Man braucht den Input und den Output. Den Input, den Telemetriedaten, die
123 ganzen Channels, speist man ein und gibt aber dann den wahren Output mit dem, was es sein
124 soll, was hinten auskommen soll. Das Machine Learning Modell, lernt während des Trainings, wie
125 es von den verschiedensten Inputs zu diesem einen Output mappt. ganz grobe Abriss, vielleicht
126 dass du ein bisschen, ich weiß nicht genau wie technisch du vielleicht auch schon in der The-
127 matik, aber das wollte ich noch mal ganz kurz abgerissen haben, dass du vielleicht ein bisschen
128 einschätzen kannst in was für eine Richtung es gehen könnte. Also im Optimalfall wäre es für
129 mich, also jetzt am Beispiel der Rundenzeit Vorhersage, da hatte ich mir schon ein bisschen
130 reingeguckt. meine, das nennt sich Label, der Output ist ein Label für das Training und das
131 ist ja relativ einfach zu bekommen dieses Label. Man nimmt einfach von jeder Lap die letzte
132 Laptime. Dann hat man das Label und speist verschiedenste Parameter rein, die Geschwindigkeit
133 des Autos, erste Runde, Sektor 2, zweite Sektorzeit, Bremsdruck. alle möglichen Channels könnt
134 man reingeben und das Modell findet selber heraus während des Trainings wie wichtig welcher
135 Parameter ist und wie die zusammenhängen um eben den beschriebenen Output B auszugeben
136 am Ende. Genau das wäre jetzt Beispiel Rundenzeitvorhersage. Und da wäre es mir wichtig, weil
137 wir jetzt noch keine hochkomplexe Sache machen möchten, dass man sagt, und weil ihr sehr viel
138 beschäftigt seid, dass man eben diesen Output B auch bereits entweder berechnen kann oder
139 näherungsweise berechnen kann oder bereits wie die Lap-Time aus den Daten ersichtlich sind.
140 Verstehst du grob?

141 Ingenieur: Ja, also zu dem Thema Rundenzeit haben wir die Predigten laptime, schon im Auto,
142 die sind ziemlich gut, die ist ein relativ Simpler Ansatz, also nichts mit machine learning. Gewis-
143 serweise merkt sich nur das Auto die Referenzrundenzeit, was der Fahrer aktuell gefahren ist und
144 vergleicht die aktuelle Runde zu dieser Referenzrundenzeit. Es dann eben eine Predictive Post,
145 also bin ich drunter. Ich habe es selber nicht geschrieben, es funktioniert aber ziemlich gut. Das
146 heißt, du kommst auch ziemlich gut da rein, wo wo der Fahrer tatsächlich dann landet. Also
147 klar weißt du nicht, ob er jetzt in drei Kurven einen Fehler macht. Also das kann keiner vorher
148 sagen. Das funktioniert ganz gut. Nur um es anzumerken, dass wir in der Richtung laufen, dass
149 du was machst, was es schon gibt, gut funktioniert, da könnte man vielleicht was machen, was
150 es noch nicht gibt. Die Sache ist, wir generieren sehr viele Daten, auch KPIs. Das Schwierige
151 ist, das Ganze in kurzer Zeit, weil es ist ja wirklich nicht viel Zeit, um das zu verstehen, die
152 Zusammenhänge zu verstehen. Genau, wenn ich jetzt mal ein Beispiel mache. Du fährst mit
153 einem Auto, Temperatur ist x von der Strecke, die hat einen Einfluss auf das Verhalten vom
154 Reifen, wie viel Grill du hast, wie viel Luftdruck der Reifeningenieur gerade reingemacht hat, wie
155 du gefahren bist mit dem Auto, wie du den Reifen aufgewärmt hast in den ersten fünf Runden,
156 hat einen Effekt auf den Grip, du in Runde 20 haben wirst. Ob du aggressiv angefangen hast,
157 hat einen Einfluss ob du länger schnell fahren kannst oder halt weniger. Es sind so schwierige

158 Entscheidungen oder auch, jetzt haben wir verschiedene Compounds Soft Medium Hart, welcher
159 ist denn jetzt gerade der optimale? Temperatur bei 30 Grad oder bei 40 Grad und ich erwarte
160 vielleicht, dass ich im Renn 50 Grad fahre, kann ich dann noch die Medium fahren oder muss ich
161 den harten schon fahren, die eigentlich für die Bedingungen gedacht ist? Das sind so interessante
162 Fragen. Es ist jetzt weniger mit Telemetrie, sondern eher, du sammelst das Wochenende über
163 Daten und versuchst die dann zu interpretieren, möglichst schnell um die richtigen Entscheidun-
164 gen vor dem Rennen zu treffen. Während man fährt, was immer ganz interessant fände, wenn die
165 Fahrer sagen, sie hätten gerne mehr Feedback oder Guidance auch? und die Telemetrie haben.
166 Beispiel im Auto, kannst so Sachen verstellen, das ist ja nicht so gegeben, wie du rumfährst, ge-
167 rade bei den Systemen. Ja, deine Traktionskontrolle, also wie viel Moment kriegt der Fahrer für
168 einen bestimmten Schlupf den du am Rad siehst. Auch mit dem Hinblick auch wieder Verschleiß.
169 Was macht mein Auto in Runde 20? Wenn ich komplett viel Schlupf generiere, dann baut man
170 bei es irgendwann ab. ist es geschickter am Anfang bisschen konservativer auch mit den Settings
171 zu fahren, dass das Auto eben nicht permanent ausbricht an der Hinterachse. Und den Fahrer
172 das als Hinweis schon mitzugeben. Auf Basis der Telemetriedaten zu sagen, geh mal mit deinem
173 Setting bisschen konservativer, weil wir sehen in den Daten, du überfährst den Reifen gerade. Da
174 gibt es auch verschiedene Kanäle, wie den Schlupf, man angucken kann, die Temperatur von den
175 Reifen. selber sein Fahrstil, wie viel Energie bringt er mit seinem Fahrstil in die Reife Es gibt
176 verschiedene auch Kanäle, wie gesagt, auf Telemetrie, die man an anschauen fand. Und entweder
177 live, so machen wir es jetzt, weil wir so Erfahrung wollen, du guckst die Daten meistens ein
178 halbes Jahr lang an und dann weißt wann du ungefähr, was in welcher Richtung, einstellen muss.
179 So gibt man den Fahrrad Hinweise, ob er was ändern sollte oder nicht. Dann sagt er manchmal,
180 hey voll gut, danke. Manchmal passt es halt nicht, aber dann muss man halt weiter lernen.

181 Interviewer: Das klingt sehr interessant. Man würde zum Beispiel aus ganz vielen verschiedenen
182 Channels den Output generieren, überfahren, nicht überfahren oder so in die Richtung. Okay,
183 ja das klingt auf jeden Fall nach einem sehr interessanten Thema, weil das ist eben auch genau
184 so was, wo ich so den Hauptbenefit sehe. Ich meine, ihr könnt ja alle, ich meine, ich brauche ja
185 jetzt kein Modell darauf trainieren, wenn du siehst, Tire Temperature größer 50, dann passiert
186 irgendwas, ja. Genau, das macht ja keinen Sinn. Aber eben sowas wie du beschreibst, was man
187 über eine längere Zeit... über eine längere Zeit einen Wert beobachten muss und dann auf Basis
188 von Erfahrung keine... Also da stehen bestimmt intrinsische Regeln dahinter. Aber nichts, was
189 der Mensch so ausdrücken könnte, in einem Code zum Beispiel, sondern eben wie du sagst, ist
190 viel Erfahrung und Bauchgefühl von Jahren. Ich glaube, das wäre auf jeden Fall ein Gebiet, wo
191 so ein Modell großen Potenzial hätte.

192 Ingenieur: Ich glaube, das würde helfen. Man muss ja gucken, dass das, glaube ich, deine Arbeit
193 jetzt nicht so ein Konzept sein man versucht, die Weltformel zu generieren. sondern vielleicht
194 mal was Simples, einen simplen Ansatz. Das ist meine Reifentemperatur, habe ich auf Telemetrie
195 und das ist meine Balance vom Auto, habe ich auf Telemetrie und den Schlupf auch. Das einzige
196 was mir bei dem Thema einfällt, das Thema, also das Training, die Trainingsdaten zu generieren.
197 Da würden wir jetzt, also weiß ich... Vielleicht kennst du dann den Ansatz, wie man sagt, die

198 Daten müssen gelabelt werden. Ich habe alle Telemetrie-Daten, pro Sekunde oder zu 100 Hertz
199 reinkommen.

200 Interviewer: Ich müsste markieren, hier diesem Zeitpunkt wurde der Reifen überfahren. Ich muss
201 dieses überfahren Label irgendwie setzen, im besten Fall automatisiert. Das ist das Einzige, was
202 wir bei dem Thema vielleicht bisschen ... Kopfzerbrechen oder was ein bisschen kompliziert ist.
203 Ich könnte es jetzt ja, wie du sagst, als Basis auf Erfahrung. Ich kann jetzt ja gerade nicht einfach
204 mir die Telemetät anzeigen und sagen, hier wurde der Reifen zu überfahren, hier, hier und hier.
205 Das klingt nachher ein super cooles Thema, aber im besten Fall müssten wir es irgendwie schaffen,
206 ihr mir das vermitteln könnt, dass ich diese Daten labeln kann. Weil sonst die Konsequenz wäre,
207 wenn das nicht klappt, wie man es dann machen müsste, dass jemand, der diese Erfahrung
208 im Kopf hat, ein Tool an die Hand und die Telemetriedaten markieren muss, hier markieren,
209 überfahren, hier überfahren, hier überfahren und das am besten 1000 mal. Aber das ist natürlich
210 für euch nicht praktikabel.

211 Ingenieur: Das gibt Kennwerte, die wir haben von hier sagen, zum Beispiel... Da gehen wir dann
212 ein bisschen von der Telemetrie weg. Das sind dann wirklich Kennwerte. Das ist die Telemetrie
213 und Kabeldaten. der Post-Programm-Testing. Und das ist dann die Integrale, mit laufenden
214 irgendwelchen Mathefunktionen dahinter. Die Auswertung machen, da kommt dann eine Zahl
215 raus. Zum Beispiel, wenn man sagt, man schaut sich Anzahl der Snaps an, in der Runde, wie
216 häufig bricht das Heck aus, wenn der Fahrer ins Gas geht. Wenn der Reifen aufgeht, dann
217 passiert das häufiger, als wenn der Reifen komplett neu ist. Das wäre zu sagen, ein Master für
218 die ZIG-Datensätze, die ich habe. ich das als Eingang in den Fahrstil, Energie, was weiß ich.
219 Und irgendwann beim Fahrer A habe ich mehr Snaps ab Runde 10 und bei Fahrer B ist es
220 erst ab Runde 20. Ich weiß nicht ob. Auf Basis dessen könnte man jetzt hier überlegen Fahrer b,
221 besser gemacht, ja. Weil bei dem kommt die Stab ein bisschen später. Ein anderes Kriterium, was
222 eigentlich auch super simpel wäre, eigentlich könnte man mal sagen, Rundenzeit. Ich kann mir
223 das mal deshalb ganz kürzlich erst einmal... Das ist einfach ein Long Run, das heißt wir fahren
224 viele Runden von zwei verschiedenen Setups , gleicher Fahrer. Du siehst hier die Rundenzeit. Das
225 Auto wird dann immer langsamer, je mehr Runden du fährst. Also kriegen wir weniger Grip.
226 Hier sieht man, wo der Reifen dann wirklich abkackt. wird die Rundzeit dann langsamer. Also
227 das kannst du in X-Richtungen verschieben.

228 Interviewer: Ah, ja. Das klingt cool. Warte mal, wenn ich mir überlege... Man gibt es sozusagen
229 anhand von aktuellen Fahrverhalten aus, solange man in eine Lebensdauer prognostiziert. Also
230 wenn du sagst nicht überfahren nicht überfahren sondern du sagst also so wie du jetzt in den
231 letzten in den letzten zwei Sektoren oder der letzten Lab gefahren bist würde der Reifen jetzt
232 noch drei halten und dann fährt er ein bisschen langsamer oder entspannter und dann dann
233 springt der Wert hoch auf fünf oder sowas. Das hört sich gut an.

234 Ingenieur: Ja, das ist halt jetzt, hier waren zwei Setups, das kann ja verschiedene Gründe haben,
235 warum das passiert. Das kann jetzt sein, wenn ich das nur die Rundenzeit dem Modell gebe,
236 das weiß er nicht. Dann weiß er halt nicht, waren das jetzt zwei Fahrer und der eine ist einfach

237 aggressiver gefahren, der andere ein bisschen schonender. Oder wie in dem Fall sind es zwei
238 Setups, das war der gleiche Fahrer, der versucht gleich zu fahren. Oder waren das irgendwelche
239 Einstellungen am Lenkrad, die er unterschiedlich gemacht hat. Ich glaube, muss man dem Modell
240 auch mitgeben. War das jetzt ein anderer Fahrer, war das ein anderer Fahrstil? Da kommt das
241 Thema wieder dazu. Wie viel Energie steckt der Fahrer in die Reifen? Wie fährt er? oder was
242 sind die Einstellungen gewesen. Aber ich glaube, geht erstmal proff of concept. Da könnte man
243 noch sagen, du nimmst mal Daten von Le Mans oder von irgendeinem Dauerlauf, wo das Setup
244 einfach gleich geblieben ist. Du fährst einfach 24 Stunden lang. Dann kann man das schon mal
245 aus xen, dass sich da irgendwas am Setup tut, dann ist es wirklich nur Fahrerunterschiede oder
246 Fahrerunterschiede.

247 Interviewer: Das ist eine sehr gute Idee, glaube ich. Das wäre glaube ich echt mal für so Proof of
248 Concept eine gute Idee, dass man erst mal abgekapselt, wie du sagst, nur für Lés mans, nur für
249 ein Setup, guckt, wie sich das entwickelt. Das könnte klappen. Da muss ich mir angucken, wie
250 die Daten, ich meine jetzt 24 Stunden, da kommt natürlich eine ordentliche Datenmenge herum,
251 ob das erreicht für ein Training. Aber das ist auf jeden Fall schon mal eine gute Richtung, in
252 die du mich da glaube ich schickst. Das ist auf jeden Fall eine gute Richtung. Da haben wir
253 auf jeden Fall eine Kripp-Pasei. Wir haben auch in den Daten, was wir machen, auf Basis der
254 Grabedaten trainieren wir ein Modell oder fitten ein Reifenmodell, wo wir dann wissen, der
255 Reifen hat jetzt weniger Grip. Er baut dann ab. Du fittest jede Runde ein Reifenmodell. Und
256 dein Krippparameter fällt dann ab, einfach damit das zur Runde passt. Und da wir ganz viele
257 Ausgärtungen auch dazu. Das klingt nach einem guten Startpunkt für mich. Okay, also jetzt mit
258 Blick auf die Uhr, weil du meinst, hast jetzt auch... Ja, genau. Ich würde jetzt auf jeden Fall mal
259 das Hausaufgabe für mich mitnehmen. Das ist jetzt echt ein cooles Thema. Ich würde mich da
260 ein bisschen einarbeiten, mir ein paar Gedanken dazu machen. Ich bin nächste Woche im Urlaub.
261 Und dann würde ich dir vielleicht in zwei Wochen, falls du da da bist, einen Termin einstellen,
262 wenn das okay wäre für dich. Dann bringe ich noch mal paar fische Gedanken rein. Du kannst
263 mir dann noch da drauf bisschen Input geben. Wäre das okay für dich?

264 Ingenieur: Ja, ich glaube, gut, wenn du für das Meeting so bisschen aufmalst, wie du dir das
265 vorstellst. Hier kommen Daten, da passiert das, da passiert das, das kommt raus.

266 Interviewer: Vielen Dank Fabian für deine Zeit. war wirklich sehr gut. Danke.

Anhang 1/2: Zweites Meeting mit Performance-Ingenieur (12.09.2025)

1 Interviewer: Zweites Meeting mit Ingenieur am 12.09.2025 um 10.30 Uhr.

2 Ingenieur: Hi, guten Morgen, grüß Dich.

3 Interviewer: Freut mich, dass wir es wieder schaffen, hier zusammen zu finden. Genau, gleich
4 vorweg, ich habe auf jeden Fall die Woche ziemlich Gas gegeben und war die ganze Woche ei-
5 gentlich, hatte ich mich mit dem ursprünglichen Thema beschäftigt, in die Richtung, du mich

6 gelenkt hast, was auch wirklich... ultra interessant ist und das war das das Reifendegradationsthe-
7 ma. Da habe ich mich die ganze Woche auch schon durch die Daten gewühlt und habe mir so
8 bisschen auch schon eingeguckt wie man es machen könnte. Und grundlegend ist mir ein, größe-
9 res Problem aufgefallen und zwar gibt es eben wie ich vielleicht auch vorher schon mal gesagt
10 hatte, das Thema, dass man das labeln muss. Also man muss sich irgendwelche Regeln, also euer
11 Bauchgefühl, also wirklich greifbar machen und definieren, wo man Schwellwerte setzt von ver-
12 schiedenen Parametern, die man dann erreicht oder so weiter, um dann zu sagen, ja jetzt hier der
13 Reifen gerutscht, hier nicht gerutscht, dass man eben nicht manuell in die Daten reingehen muss
14 und das wirklich manuell machen muss. Das ist ein größeres Hauptproblem und ich glaube, das
15 hatte ich letztes Mal gar nicht erwähnt. Mein Projekt hier ist nur bis November und ich muss
16 auch noch eine wissenschaftliche Arbeit in diesem Zeitraum bis November darüber schreiben.
17 Das heißt, ich habe relativ wenig Zeit. Und dementsprechend war ich gestern recht glücklich,
18 weil mir ist dann noch eine Idee gekommen. Ich habe nochmal darüber nachgedacht, was du ge-
19 sagt hast. Und eine der Sachen, die du angesprochen hast, ist, dass die Fahrer oft sagen, dass sie
20 gerne mehr Feedback hätten. Dass sie genauer oder noch mehr Feedback hätten. Und da ist mir
21 eine Idee gekommen, kombiniert, wie man das gleich umsetzen könnte, ist, dass man ein Modell
22 trainiert, um Fahrerfeedback zu generieren. Und die Idee, die ich dabei hatte, ist, Man nimmt
23 die Telemetriedaten und man hat ja für jedes Outing schon in den Daten gesetzt welcher Fahrer
24 gerade fährt. Und sozusagen man trainiert das Modell darauf auf verschiedene Fahrer und dann
25 gibt es eine Technologie die nennt sich Explainability AI. Das heißt normalerweise kennt man es
26 ja, dass solche AI Programme ziemliche Black Boxen sind. bringt irgendwas rein, die trainiert
27 sie und dann gibt man Input und es kommt irgendwie ein Output raus. Wunder, Wunder. Aber
28 da gibt es tatsächlich auch schon Ansätze, um eben genau diese Black Box aufzubrechen und zu
29 gucken, wie kommt die KI zu der Entscheidung. Und hier kommt der Mehrwert ins Spiel. Meine
30 Idee jetzt aktuell, wer man guckt, also man trainiert sie auf die verschiedenen Fahrer und kann
31 zum Beispiel dann auch live gucken. In dem und dem Sektor war Fahrer 1, 2 oder 3 schneller und
32 dann kann man wirklich relativ tief in die Daten eintauchen und schauen wo die Unterschiede bei
33 den Fahrern liegen über die AI. Das war jetzt meine Idee. Das wäre aus einem aus dem großen,
34 also für mich wäre es deutlich angenehmer umzusetzen. Aus dem Grund, ich die Labels, also
35 welcher Fahrer gerade fährt, einfach aus den Daten ziehen kann. dementsprechend, also dieses
36 Labeling-Thema ist bei so Maschinen-Learning-Projekten meistens immer der größte Painpoint
37 und der größte Faktor, warum was scheitert. Und da sehe ich das jetzt eben auch eventuell bei
38 dem iPhone-Digaktionsthema, weil es schon eben was Großes ist und wenn es so einfach wäre,
39 dann wird es wahrscheinlich schon gemacht werden. Dementsprechend hätte ich jetzt...

40 Ingenieur: Ich kann dir kurz mal zeigen, was ich mir gedacht hätte. Okay. Und dann kannst du
41 mir sagen, ob das so kompliziert ist. Okay, okay. Und dann können wir vielleicht noch mal in das
42 andere einsteigen, weil ich hatte mir jetzt auch kurz vor dem Meeting noch mal fünf, zehn Minuten
43 Zeit genommen. Das ist super nett. Ich mir paar Gedanken gemacht. Also, ja, geht ein bisschen in
44 die Richtung, aber was mich halt, was halt echt interessant wäre... Oder was uns immer umtreibt
45 ist, was macht die Fahrzeug Balance? ist das Auto eher neutral oder untersteuernd? Und das
46 hängt halt ab von ganz vielen Parametern. Daher ist es für uns oder als Mensch relativ schwierig

47 zu verstehen, was jetzt was beeinflusst. So das Gleiche gilt für den Grip. Den kann man mal außen
48 vor lassen. Deswegen haben wir gesagt, wir gucken mal erst mal Renndaten an, weil Renndaten,
49 da kann ich die ganze Setup Arbeit, was der Renningenieur macht und was noch das Ganze
50 noch viel komplizierter macht, erst mal außen vor lassen. Wenn man sich so bisschen vorstellt
51 wie eine Gleichung, das wäre mein Y, die Fahrzeug Balance, die hängt halt von ganz vielen
52 Eingangswerten X und dann gibt es hier eine Funktion, die wir nicht kennen. Und dann kommt
53 da die Balance raus. Die Eingangsparameter, zum Beispiel die Streckentemperatur. Die Strecke
54 jetzt, die messen wir ja. Wir haben eine Wetterstation, das gibt es in der Datenplattformen auf
55 jeden Fall. Gibt es in unseren Workbooks. Die gibt es. Reifentemperatur haben wir, die messen
56 wir. Reifendruck haben wir, messen wir auch. Wie du sagt Fahrer, wissen wir auch, welcher
57 im Auto sitzt. Wir wissen die Knöpfe an denen der Fahrer dreht, was er einstellen kann. Der
58 kann nämlich die Stabis verstellen. Da gibt es auch einfach Knöpfe 1, 2, 3, 4, 5. Recht diskrete
59 Stufen. könnte auch direkt die mechanische Balance angucken, aber das geht vielleicht auch mit
60 den Stabis. Weiß nicht ob man hier Aero-Balance braucht. Die ändert sich eigentlich nicht. Die
61 wäre auch recht konstant. Reifenspec ähnlich wie Fahrer, wissen wir ob ein trocken Reifen oder
62 ein Regenreifen am Auto ist und trocken Reifen wissen wir sogar welcher Gummimischung, ob
63 es ein Soft, Medium oder Hard ist, stellt der Fahrer ein. Dann wichtig ist natürlich auch die
64 Mileage. Wie viel Laufzeit hat der Reifen schon? Da bin ich mir unsicher, ob wir das schon in der
65 Datenplattformen. Gibt es aber genauso schon als in HH. Genauso wie die Streckentemperatur
66 müsste man die Mileage einlesen können. Jetzt weiß ich nicht, wer das macht. ob das ein Monin
67 oder ein Paul machen kann. jeden Fall wäre das so was, wo ich sie irgendwann mal gerne hätte.
68 Dass man die Mileage von dem Set hat. Dann Software. Da du natürlich auch einiges verstellen.
69 Deine TC, also Traktionskontrolle oder deine Bremsbalance. Aber das sind auch Knöpfe. Da dreht
70 er dran. Also TC 1, 2, 3, 4, Auch diskrete Stufen. und dann natürlich der Sprit, den es gibt.
71 Ist das Auto schwer oder leicht? Das waren für mich mal die Schnelle, die Prior 1 Faktoren.
72 Hier können wir mal Aero ausklammern. Das bleibt hoffentlich auch konstant, weil wir ja Setup
73 nicht ändern. So und von den Ausgangsgrößen, habe hier mal eine Runde aufgemacht, nicht eine
74 Runde, sondern ein Workbook in Power BI. ist jetzt ein Auto Le Mans, komplette Rennen. Man
75 kann es vielleicht ein bisschen anders darstellen. Wir haben ja schon Kennwerte. Dann siehst du
76 hier, wir wollen die Balance ganz gut.

77 Interviewer: ist es das, was ihr euch live an der Strecke anguckt, das PowerBi Dashboard, oder
78 habt ihr Wintax mit den genauen Telemetriedaten offen? Das sind ja die Metriken, oder?

79 Ingenieur: Beides, also Wintax sind wirklich deine Telemetriedaten als Datenstrom und dann ha-
80 ben wir halt Auswertung. kannst Gates definieren. das, du sagst, wo du die Thresholds brauchst.
81 Und dann wo du sagst, du kannst das Cluster. Das machen wir hier. Okay, wir haben einen
82 Cluster Entry, also Eingang der Kurve, Mitte der Kurve und Kurvenausgang. Und dazu die Ba-
83 lance. Und das für jede Runde gibt es dann einen Wert. Das ist ein Mittelwert über alle Kurven.
84 Eingang. Mid-Corner und Exit und dann siehst du übers komplette Rennen, ihr Blau ist Kevin
85 Estrid, dann haben wir Laurence und den Matt Campbell. Hier gibt es einen Wert. Der ändert
86 sich schon ein bisschen das kann jetzt halt die Balance abhängig sein von Streckentemperatur

87 von ihren Settings was weiß ich hat sich auch hier während des stints ändert sich die Balance,
88 das ist wahrscheinlich effekt von mileage oder auch fuel load das ist halt die Sache ich weiß es
89 halt einfach nicht.

90 Interviewer: Was genau sagt die Balance? Wie das Gewicht verteilt ist über die Reifen? Ist das
91 die Balance?

92 Ingenieur: Die Balance die die Fahrer beschreiben, die Fahrzeug Balance. Hast du viel unter-
93 steuern im Auto? Im Endeffekt, was mathematisch dahinter steckt. Du hast zwei Signale. Es ist
94 vereinfacht. Zwei Signale. Den Lenkwinkel. Also wie viel lenkt der Fahrer? Und die Gearrate. Das
95 Auto misst wie schnell sich das Auto dreht. Dann kannst du dir umrechnen, die Gierrate in einem
96 Lenkwinkel. Wenn der Fahrer lenkt und das Auto reagiert sofort genauso wie der Fahrer lenkt,
97 hast du ein sehr neutrales Auto. Indem du die Bewegung vom Auto mit dem Input vom Fahrer
98 vergleichst. Wenn der Fahrer jetzt extrem viel lenken muss, aber das Auto bewegt sich gar nicht,
99 typischen Straßenautos, dann hast du sehr viel Untersteuern. Also der Fahrer lenkt sehr viel.
100 Aber das Auto dreht sich einfach nicht. Wenn hingegen, wenn du, wenn der Fahrer das Lenkrad
101 ein bisschen bewegt und das Auto dreht sich sofort, dann hast du ein sehr übersteuerndes Auto.
102 So und das ist mit Balance gemeint, mit Fahrzeug Balance. Wie viel sozusagen das Verhältnis
103 von Input vom Fahrer und wie rotiert das Auto? Also wie verhält sich das Auto dann darauf?
104 So kannst du es einteilen in, man kann es entweder ganz einfach klustern in, wenn du dann Delta
105 rechnest, Gierrate und Lenkwinkel, wenn das Auto sozusagen mehr dreht als der Fahrer eingibt,
106 kannst du einfach sagen übersteuernd. Wenn es geht sich genauso verhält, neutral, wenn du sehr
107 viel Lenkwinkel brauchst, kannst du sagen untersteuernd. Kannst du sagen in die drei Gruppen.
108 Oder was wir halt einfach hier ganz stumpf haben, ist den Wert sozusagen. Das müsste das Delta
109 sein von wie viel rotierte das Auto und wie viel mehr muss der Fahrer lenken als Lenkwinkel.
110 Einfach die Differenz zwischen beiden als Mittelwert pro Runde. Genau und das Interessante ist
111 jetzt, das ist mein Y, aber ich verstehe nicht, warum geht das jetzt, warum reduziert sich die
112 Balance, warum wird das Auto neutraler, also weniger untersteuernd. Woran liegt das? Liegt
113 das jetzt dran, weil der Fahrer was verstellt hat oder weil sich die Streckentemperatur geändert
114 hat oder weil der Reifen ja, recht viel Kilometer drauf hat oder weil der Tank leer ist und sich
115 dadurch die Balance ein bisschen verändert. Weiß ich nicht. Das wäre halt was uns am Endeffekt
116 interessiert oder mich die Frage, die ich mir ganz oft stelle, wenn man den Zusammenhang weiß.
117 Ich weiß vor allem den Einfluss von dem Parameter von der Streckentemperatur auf meine Ba-
118 lance. Dann weiß ich halt schon, weißt du, du fährst dann in deinem FP1 am Morgen und bist
119 eigentlich recht glücklich mit der Balance. So, dann fährst du dein FP2 am Nachmittag. Nur da
120 ist die Strecke 20 Grad heißer. Der Fahrer sagt auf einmal, die Balance ist komplett daneben.
121 So, dann fängst du wieder von vorne an. Stellst dein Auto wieder ein, dass der Fahrer glücklich
122 ist. Denn Rennen ist aber dann vielleicht wieder zu deinen Temperaturen vom FP1 und dann
123 passt es halt wieder nicht. Du bist eigentlich immer hinten dran und tust hinterher. Aber wenn
124 du wüsstest, okay, ich kenne ja die Wettervorhersage ich weiß, im FP2 bin ich vielleicht 20 Grad
125 heißer. Sagt der Fahrer, die Balance ist scheiße, mach mal was. Dann kann ich ihm sagen, ja, das
126 liegt an der Strecke. Keine Sorge, im Rennen wird es so und so warm. Das passt alles. Oder ich

127 kann es ihm einfach erklären auch und sagen, da ist nichts kaputt am Auto, liegt einfach an der
128 Strecke. Die ist halt extrem viel heißer.

129 Interviewer: Verstehe. Genau, heißt also man hätte hier in dem Fall also das Labeling wieder
130 zurück zum Thema. Hätte man das schon? Das ist genau der Graphie eigentlich oder nicht?

131 Ingenieur: Genau. Das weiß ich jetzt nicht, ob es im Detail, also das musst du jetzt mir sagen,
132 das weiß ich nicht, ob das die Arbeit schon einfacher macht.

133 Interviewer: Aber das ist ja Prinzip nochmal eine neue Problemstellung. Das hat erstmal nicht
134 viel mit dem Reifendegrationsvorhersage zu tun, sondern es ist jetzt sozusagen... Also es ist ja
135 ähnlich wie das andere mit dem Fahrer, bloß wir geben Inputvariablen und haben das Labeling
136 Output und jetzt wollen wir verstehen, was dazwischen passiert.

137 Ingenieur: Genau.

138 Interviewer: Okay. Das ist nochmal eine ganz andere Fragestellung.

139 Ingenieur: Es geht halt auch um das Richtung Reif. Das Thema mit dem Verschleiß, den Kenn-
140 wert, den haben wir noch nicht. Den werden wir auch nicht vor November haben. Da habe ich
141 mit den Kollegen nochmal gesprochen. Also bringt ja da nichts.

142 Interviewer: Was soll da kommen? Ich glaube, das habe ich jetzt noch gar nicht gehört.

143 Ingenieur: Wir wollten ja mit der Degradation was machen. Wie baut der Reifen ab? So, und da
144 muss ein Kennwert berechnet werden, wie hier, mit der Balance. Den gibt's aber noch nicht. Der
145 Plan war, dass wir den vor zwei Wochen implementieren. Das kriegen sie aber nicht hin. Bis Ende
146 November. Scheinbar komplizierter. Die wollen sagen, der Reifen ist jetzt so so viel degradiert,
147 Genau. am Endeffekt ist, was dahinter steckt, es gibt ein Reifenmodell. Du modellierst einen
148 Reifen mit mechanischem Verhalten über Kennlinien, über Parameter. Wir wissen, wie er sich
149 neu verhält, wie die Parameter ausschauen. dann kannst du jede Runde hast du die Messdaten
150 und optimierst die Parameter so, dass dieses Modell dazu passt. Wenn das Sinn macht. Du hast
151 sozusagen Optimierer, der die Parameter tunen, damit mein Modell zu dem Messdaten passt.
152 Und diese Parameter, die beschreiben zum Beispiel den Grip. Das ist ganz einfach. Ich kann es
153 auch ganz einfach machen. Maximale Querschleunigung beschreibt meinen Grip. So, wenn ich
154 jetzt weniger Grip habe, dann geht meine maximale Querschleunigung runter. Das wäre ein
155 ganz einfacher Ansatz. Du guckst ja per Runde an, was meine maximale Querschleunigung
156 habe ein Modell. So Querschleunigung ist ein bisschen wie mechanisch mit einem Reibmodell.
157 hast eine Last Fz. Ich kann meinen Fy berechnen. ich messe meinen Fz. Das weiß ich im Auto.
158 Das messe ich. Ich weiß meinen Fy. Das ist meine Querkraft, meine Querschleunigung. Und
159 dann kann ich einfach ausreden, was ist denn mein Reibwert, mein μ_y . Mit der Fz und dem
160 Mühe kriege ich diese Seitenkraft. Und so kann ich mir das ausreden. Das ist sehr, sehr, sehr,
161 sehr, sehr vereinfacht. Da haben wir ein komplexeres Modell, aber so funktioniert das. Das wollen
162 wir in die Datenplattform reinbringen. Diesen Prozess, den gibt es schon, aber erst einmal nur
163 in Matlab. recht viel händisch, Daten runterladen, konvertieren, Modell identifizieren. Und dann
164 gibt es dann eine Routine und die wollen wir halt in die Datenplattform bringen. Das dauert

165 aber noch bisschen scheinbar aufwendiger als gedacht, würde ich mal vermuten, ohne dass von
166 den Kollegen gehört zu haben. Aber sonst wäre es ja schon drin. Das heißt also davon bin ich
167 ausgegangen. Das heißt vor dem Urlaub hatte ich Info, dass wir das rein kriegen. Also in der
168 Woche, wo du im Urlaub bist, dann hätten wir das jetzt schon, aber das gibt es nicht. Die haben
169 gesagt, das wird nichts bis Ende November. Das bringt dir erstmal nichts.

170 Interviewer: Okay, das ist schon mal interessant. gut. Also wenn es das gäbe, das würde natürlich
171 das Reifendegrationsprojekt vereinfachen.

172 Ingenieur: Ja, aber im Endeffekt ist es nichts anderes wie das hier, weil es ist einfach nur ein
173 anderer Parameter, anderes Y dann. Also du willst ja so eine Art Prinzipstudie machen. Und
174 mein Verständnis, wir sagen können, hey, das funktioniert prinzipiell der Workflow, dann kann
175 ich ja alles Mögliche reinwerfen und rauskriegen.

176 Interviewer: Ja, also wo man die genau weiß, wie die Einflussgrößen das Outcome bestimmen.
177 Genau. Okay, dann können wir noch mal ganz kurz ein bisschen tiefer einsteigen in die Car
178 Balance, in das Thema hier. Genau, also wir haben jetzt hier unser Zielvariable, die Car-Balance,
179 die ist gespeichert und definiert und die können wir angucken. Ingenieur: Die wird gemessen.

180 Interviewer: Genau, du hattest jetzt gerade noch das andere Notebook offen, wo du schon mal
181 versucht hast, die Inputparameter zu bestimmen.

182 Ingenieur: Also hier, ist dein Y, das sind genau die drei Graphen. Und als Eingang, das kann
183 man auch anschauen, das kann man auch sagen, die sind ein bisschen verstreut leider. in der
184 Darstellung. Zum Beispiel Reifen. Das wäre zum Beispiel eine Eingangsgröße. Das ist der Front,
185 also vordere Anti-Roll Bar also dein Stabilisator. Vorne die Einstellung vom Fahrer, was er
186 ausgewählt hat. jetzt wieder hier hingeht, wäre der Punkt hier. Damit kann er die Balance im
187 Auto verstellen. Dann haben wir hier die TC-Settings. Das ist ein Software-Ding. Da sind wir
188 hier auf. weiß nicht, ob es jetzt angezeigt wird. Dem Wert hat dann ist er nach oben gegangen,
189 wieder runter, recht wenig verstellt. Das wäre das zweite Setting. Ja, man die Bremsen auch
190 wieder Software, die Bremsbalance, wie er es verstellt hat. Dann haben wir noch zwei weitere
191 für die Bremsbalance. Also eigentlich die drei Parameter Brems-Balance. Das wäre also sag mal,
192 mit dem Block hier, hätte man eigentlich schon. Das hier wären die Outputs. Ja, so jetzt kann
193 man es noch ein bisschen Clustern . Ich weiß nicht, ob man so kompliziert schon werden muss.
194 Das ist noch mal ein bisschen anderes Gating, wenn man das also das ist ja gesagt die Balance,
195 ich gesagt hatte erst mal. Eingang Mitte Exit als Mittelwert über die Runde, dann kannst du
196 es noch ein bisschen komplizierter machen, indem du noch mal die Fahrzeug Geschwindigkeit
197 unterscheidest in High Speed, Medium Speed, Low Speed.

198 Interviewer: Also würde zum Beispiel für einen POC, also ich bräuchte ja eigentlich eine Zielva-
199 riable, ich ja gucken, zum Beispiel Highspeed, Midcorner, oder?

200 Ingenieur: Ja, oder du schaust einfach mal Midcorner über die Runde an, ob sich da irgendwas
201 tut. So, hier hätten wir schon die Temperaturen. Da bin ich jetzt nicht sicher, welche wir davon
202 messen im Auto die Oberfläche vom Reifen, aber wir messen auch...

203 Interviewer: Aber da würdest du auch sagen, dass es Sinn macht, zum Beispiel das jetzt nur
204 auf die diesjährigen Les Mans Daten von einem Auto zu trainieren, oder? Weil sich dann zum
205 Beispiel nicht so Sachen, die wir jetzt nicht abbilden können, so Setup-Changes, die eben...

206 Ingenieur: Ich würde es pro Auto machen, weil die Setups unterschiedlich sind. Jetzt kannst du
207 aber sagen, ich trainiere dreimal ein Modell, nämlich zu jedem Auto. Oder du könntest sagen, weil
208 dieses das Ganze sollte universell ja gelten. Sozusagen wenn man dann in Differenzen oder Delta
209 das überlegt, wenn ich sage, meine Strecke wird 10 Grad heißer, dann sollte sich sollte das Auto
210 übersteuernder werden. Das sagt mein Training vom Auto 6. Dann könnte ich gucken, hey, jetzt
211 schaue ich mal, wie sich die Streckentemperatur vom Auto 5 geändert hat. Mach sozusagen Replay
212 von meinem Modell und schau, wie gut passt mein Modell zu dem, was ich vorhersagen würde, wie
213 gut passt es zu dem, was Auto 5 gemacht hat. Sozusagen Validierungsschleife dann oder was mein
214 Auto 4 gemacht hat. Man könnte auch sagen, ich schmeiße alle Daten in einen Topf und schaue, ob
215 es generelle Regeln gibt. Ja, die Setups sind unterschiedlich, aber die Balanceänderung zu meinem
216 Setup sollte ähnlich sein. Ich ändere das nicht im Rennen, das Setup von allen drei Autos. Ich
217 glaube, man könnte erst mal sagen, du hast ja echt viele Daten. Man startet mal mit einem Auto.
218 Ich glaube, das ist dann auch nicht viel Stress, wenn man sagt, man schmeißt dann alle drei Autos
219 mal rein oder wie gesagt, spannend wäre ja zu sagen, ich drehe mir mal ein Auto und gucke,
220 wie gut passt meine Vorhersage zu den anderen Autos. Das ist jetzt die Reifentemperatur, vorne
221 links, vorne rechts, hinten links, hinten rechts. Das sind ist Durchschnittswerte pro Runde.

222 Interviewer: Ok, verstehe. Ja das ist doch auf jeden Fall auch schon mal cool. Weil diese Durch-
223 schnitte würde man eh berechnen. Das heißt, wäre vielleicht sogar was, was ich mir direkt run-
224 terziehen könnte, ohne große Vorverarbeitung. Einfach versuchen könnte, das zu trainieren. Das
225 Einzige, was mir hier auffällt, es gibt so ein paar Ausreißer. Sind es Fehlmessungen hier?

226 Ingenieur: Nee, das kann es sein. Zum Beispiel, wenn du einen Safety Car hast, dann ... steckst
227 nicht viel Energie in die Reifen, weil fährst nicht schnell um die Kurve, dann geht halt die
228 Temperatur runter. Oder beim Reifenwechsel neue Reifen, wir fahren ja nicht mit Heizdecken
229 oder mit warmen Reifen los, so wie in Formel 1, die sind ja von kalt und dann werden die mit der
230 Zeit dann wärmer, bis es sich irgendwann stabilisiert haben. Verstehe. Das passiert hier immer.
231 Reifenwechsel, Start ist von kalt, dann wird er wieder heiß. Dann Boxenstopp wird aufgetankt,
232 kühlt ein bisschen ab. Oder hier waren Safety Car, nein wahrscheinlich Boxenstopp. Kühlt ein
233 bisschen ab, dann wird er wieder heiß. Dann haben wir wieder Reifen gewechselt. Genau, geht
234 dann wieder runter hier. Wahrscheinlich haben sie ein bisschen Energie gespart oder waren Safety
235 Car oder eine Slowzone. Ich weiß es nicht, da müsste man jetzt ins Detail gucken. Aber das sind
236 die Schwankungen hier sind normalerweise irgendwie Safety Car oder Reifenwechsel. Warum das
237 hier nach oben abrauscht, das kann ich dir nicht gar nicht sagen. Vielleicht ist da irgendein Sensor
238 ausgefallen. Diese Surface, die sind immer ein bisschen, die gehen öfters mal kaputt. Okay. Die
239 sind halt, das sind so Infrarot Sensoren. Die sind im Radkasten, auf dem Reifen und wenn da ein
240 Gummifurzel abfliegt, an diesen Sensor ran. Dann kann es sein, dass der kaputt geht. Vielleicht
241 nimmst du da einfach diese Inner-Liner. Das sind Sensoren, die vom Rad auf den Gummi gucken.
242 Das geht nicht so schnell kaputt. Weil hier hat man auch so ein paar Ausreise. Da kannst du

243 sagen, alles was Also 300 Grad, da schmilzt der Reifen. Das lässt sich mit dem Filter dann echt
244 einfach.

245 Interviewer: Du sagst, also wir kriegen eigentlich alle Daten, du jetzt in deinem OneNote hast,
246 kriegen wir hier als Metric irgendwo her, oder?

247 Ingenieur: Wir haben hier alle Sachen, außer die Mileage. Aber die wäre schon wichtig. Das
248 müsste man Paul fragen, dass sie das noch integrieren. wäre auch einfach ein Wert pro Runde.
249 Wenn es dir einfach machen willst, kannst du sagen, du nimmst die Anzahl der Runden. Dann
250 weißt du das einzige, du halt da nicht weißt ob das ein neuer ist, der drauf gekommen ist oder
251 ob der schon drauf war.

252 Interviewer: Wo liegt das an? Also es liegt ja aktuell ab, hast du gesagt, irgendwo,

253 Ingenieur: Ja, zum Beispiel, also ich hab hier grad HH, das ist unsere Datenbank Software.
254 Die sind schon verlinkt. Also das Workbook. Und die Software, du siehst hier zum Beispiel ist
255 Tracktemperature die Zieht sich die Datenplattform von dem Tool hier von HH. Genauso wie
256 die ganze Setup-Information, steht da steht alles hier drin. So, wenn man jetzt mal guckt. Mein
257 Gedanke wäre, den Workflow gibt es schon. Jetzt muss nur einer sagen, ich integriert anstatt
258 Streckentemperatur die Reifenmilage, weil ich die Kilometer vom Reifen hier schon habe und
259 mache da einen Metric, einen unten Endwert oder Startwert, ist dann auch nicht so dramatisch
260 und nehme das dann als Kernwert.

261 Interviewer: Könntest du, würdest du dir was ausmachen, wenn du deine OneNote-Page shares?

262 Ingenieur: Ja, die kann ich dir dann schicken.

263 Interviewer: Danke schön. Ja, aber genau, das ist dann auch ein Thema, ich Paul gebe. Und
264 ansonsten, es wäre praktisch, ich alle Input- und alle Output-Daten an einem Ort habe, dann
265 trainiere ich und Es ist auch wieder genau, dann geht es wieder zurück zu diesem Explainability.
266 Ich muss mal gucken, inwiefern, wie genau ich sagen kann. was auf jeden Fall geht, ist, ich kann
267 schauen, welche Einflussgröße, wie stark beeinflusst und so weiter. Aber das geht auch relativ
268 tief.

269 Ingenieur: Das hat jetzt halt nichts mehr mit Rohdaten zu tun, weil wir sind ja mal gekommen
270 von den ersten Ideen auf Telemetrie-Daten was zu machen. Das hat dann halt nichts mehr mit
271 Roh-Telemetrie-Daten zu tun, sondern nur noch mit Process-Daten, die schon aufbereitet wurden
272 gewisserweise.

273 Interviewer: Das ist okay.

274 Ingenieur: So, jetzt kann man mal gucken hier. Das ist eine grafische Darstellung. Hier siehst
275 du, da sind neue Reifen drauf gekommen. Hier wurden die dann, das ist keine vier Punkte, die
276 wurden dann drauf gelassen. Erster Run, logischerweise neue Reifen. Man sieht dann hier, das
277 sind Anzahl der Runden, aber das wäre vielleicht auch ausreichend. Weil es gibt diese Zahl, du
278 siehst hier der Reifen. in dem Outing war von 0 Runden bis 12. Wenn ich jetzt die nächste Runde

279 anschau, dann siehst du, der startet bei 12 und geht dann bis 25. Das heißt, ist frisch drauf
280 gekommen von frischen Reifen. Also hier ganz am Anfang frisch, genau. Aber der ist dann nicht
281 mehr frisch, der hat schon 12 Runden. Und der nächste, hat dann, siehst du ja, der startet bei 25
282 Runden und den hat in Summe 37 drauf gehabt. Dann haben wir Fahrer gewechselt. Der Reifen
283 hat schon eine Runde drauf gehabt. Der ist dann weitergefahren, nicht getauscht, hat in Summe
284 40 Runden gefahren. Dann Fahrerwechsel, frische Reifen, startet wieder bei Null. Der ist dann
285 13 Runden gefahren und so weiter. mein Punkt ist, es gibt hier schon einen... Parameter Signal
286 das dir sagt wie viele Runden hat der Reifen.

287 Interviewer: Also hast du so ein Wert ab wann von deinem Bauchgefühl ist die Peak Performance
288 von dem Reifen wann und wie viele Runden lässt die nach?

289 Ingenieur: Ja, also Peak sind die ersten acht Runden, wenn sie pushen. Aber oft fahren sie die,
290 also wenn sie es geschickt machen, nutzen sie die auch nicht. Weil wenn du wirklich die Peak
291 Performance nutzt, dann machst du den Reifen ein bisschen kaputt. Das tut dir dann später eher
292 weh. Weil er der dann schneller abbaut. Das ist ein anderes Thema. Aber ja, siehst, dass der
293 Peak irgendwo hier im ersten Run. dann muss man gucken, der baut dann schon ab. Auto wird
294 dann auch langsamer.

295 Interviewer: Allgemein gibt es mehrere Themen für euch, wo ihr sagt, wir haben hier eine metric.
296 Wir wissen auch wahrscheinlich welche Eingangsvariablen es gibt, aber wir wissen nicht genau,
297 wie die zusammenhängen, um dieses Ergebnis zu produzieren, oder? Also das gäbe es für Degrada-
298 tion, jetzt eben für das Balance Thema und so weiter. Also da gibt es so ein paar Sachen, wo
299 das so ist, oder?

300 Ingenieur: Ja, schon. Also es gibt verschiedene Ansätze. Du kannst natürlich sagen, ich bilde
301 mir ein Modell von dem Reifen, nehme die ganzen Daten und fit das. Und dann weiß ich auch,
302 wenn ich an der Streckentemperatur, Dreh, weil sich auch was passiert. Aber das ist auch extrem
303 viel Aufwand, dann Modell zu generieren, was halt passt in allen Bedingungen. Oder der andere
304 Ansatz wäre halt einfach jetzt. Ich habe unendlich viele Daten. Wir haben ja extrem viele Daten in
305 dem Projekt. Und ich nehme einfach die Daten, wie sie sind, und versuche da die Zusammenhänge
306 zu verstehen. Und da gibt es viele Sachen, wie jetzt also das Thema mit der Balance oder wie
307 gesagt mit der. Degradation, wenn wir das mal drin haben. Da kann man sich unzählige Sachen
308 vorstellen. Beispiel gibt es auch irgendeine Abhängigkeit für meine Aerobalance, die sich über
309 die Laufzeit Da kann man kreativ werden. Deswegen, wenn man dann einen Prozess hat, wo man
310 sagt, das ist mein X und das ist mein Y, ich trainiere das. kann dann vorhersagen, was passiert
311 mit meinem Y, wenn ich an X ein was ändere. Das wäre halt schon echt hilfreich.

312 Interviewer: Dann danke für das Gespräch, ich glaube es hat sehr geholfen.

Methodische Anmerkungen zu den Interviews

Die Interviews wurden als semi-strukturierte Experteninterviews geführt und digital aufgezeichnet. Die vorliegenden Transkripte sind Rohtranskripte, die zur besseren Lesbarkeit minimal ge-

glättet wurden, jedoch den originalen Gesprächsinhalt und -verlauf authentisch wiedergeben.

Die Gespräche dienten der:

- Anforderungsanalyse für das Machine Learning Projekt
- Identifikation relevanter Telemetriedaten und Kenngrößen
- Bewertung verschiedener Ansätze (Reifendegradation vs. Car Balance)
- Klärung technischer Umsetzbarkeit und Datenverfügbarkeit

Anhang 2: Modell-Ergebnisse

Anhang 2/1: Ergebnisse der Event-Validierung

Modelltyp	Parametertiefe	Kategorisch	Aggregate	Smoothed	MAE	RMSE	R2
xgb	shallow	True	True	0	0.272	0.344	-0.443
lgb	shallow	True	True	0	0.25	0.318	-0.239
xgb	deep	False	True	4	0.259	0.33	-0.332
lgb	deep	False	True	4	0.316	0.394	-0.895
xgb	deep	True	False	2	0.242	0.308	-0.162
lgb	deep	True	False	2	0.222	0.284	0.012
xgb	medium	False	True	3	0.277	0.344	-0.443
lgb	medium	False	True	3	0.267	0.335	-0.372
xgb	shallow	False	False	2	0.307	0.38	-0.761
lgb	shallow	False	False	2	0.256	0.33	-0.327
xgb	very-deep	False	True	0	0.267	0.335	-0.372
lgb	very-deep	False	True	0	0.296	0.365	-0.625
xgb	medium	True	False	2	0.259	0.325	-0.291
lgb	medium	True	False	2	0.216	0.28	0.04
xgb	deep	True	True	3	0.23	0.297	-0.074
lgb	deep	True	True	3	0.237	0.303	-0.125
xgb	deep	True	True	2	0.226	0.292	-0.042
lgb	deep	True	True	2	0.235	0.301	-0.108
xgb	medium	False	False	3	0.266	0.336	-0.379
lgb	medium	False	False	3	0.309	0.384	-0.804
xgb	shallow	False	True	4	0.298	0.368	-0.656
lgb	shallow	False	True	4	0.247	0.314	-0.206
xgb	shallow	False	False	0	0.325	0.393	-0.884
lgb	shallow	False	False	0	0.325	0.394	-0.899
xgb	shallow	True	True	4	0.25	0.313	-0.196
lgb	shallow	True	True	4	0.231	0.3	-0.1

xgb,very-deep,False,False,3,0.259,0.328,-0.315
 lgb,very-deep,False,False,3,0.258,0.329,-0.322
 xgb,deep,True,False,3,0.24,0.307,-0.155
 lgb,deep,True,False,3,0.223,0.288,-0.013
 xgb,very-deep,True,False,3,0.239,0.306,-0.142
 lgb,very-deep,True,False,3,0.21,0.274,0.081
 xgb,shallow,False,False,3,0.29,0.362,-0.597
 lgb,shallow,False,False,3,0.267,0.342,-0.433
 xgb,shallow,True,False,2,0.267,0.345,-0.456
 lgb,shallow,True,False,2,0.212,0.276,0.067
 xgb,medium,True,False,4,0.287,0.36,-0.584
 lgb,medium,True,False,4,0.219,0.281,0.038
 xgb,very-deep,True,False,4,0.246,0.314,-0.204
 lgb,very-deep,True,False,4,0.22,0.283,0.02
 xgb,deep,False,True,2,0.279,0.346,-0.467
 lgb,deep,False,True,2,0.296,0.37,-0.672
 xgb,shallow,True,False,0,0.34,0.418,-1.135
 lgb,shallow,True,False,0,0.211,0.28,0.045
 xgb,deep,True,False,4,0.262,0.334,-0.366
 lgb,deep,True,False,4,0.223,0.288,-0.01
 xgb,very-deep,False,False,4,0.253,0.324,-0.282
 lgb,very-deep,False,False,4,0.282,0.354,-0.529
 xgb,deep,False,True,0,0.263,0.33,-0.334
 lgb,deep,False,True,0,0.299,0.366,-0.638
 xgb,medium,False,True,0,0.273,0.341,-0.425
 lgb,medium,False,True,0,0.322,0.394,-0.895
 xgb,deep,False,True,3,0.266,0.335,-0.373
 lgb,deep,False,True,3,0.262,0.33,-0.329
 xgb,shallow,False,True,0,0.285,0.356,-0.551
 lgb,shallow,False,True,0,0.249,0.316,-0.219
 xgb,medium,True,True,0,0.239,0.308,-0.163
 lgb,medium,True,True,0,0.238,0.303,-0.119
 xgb,shallow,True,False,3,0.248,0.317,-0.225
 lgb,shallow,True,False,3,0.218,0.282,0.032
 xgb,very-deep,True,False,0,0.261,0.327,-0.305
 lgb,very-deep,True,False,0,0.212,0.272,0.093
 xgb,deep,True,True,4,0.23,0.295,-0.065
 lgb,deep,True,True,4,0.241,0.308,-0.162
 xgb,very-deep,False,False,2,0.252,0.321,-0.258
 lgb,very-deep,False,False,2,0.284,0.363,-0.606
 xgb,medium,True,True,3,0.234,0.3,-0.099

lgb,medium,True,True,3,0.234,0.298,-0.088
 xgb,very-deep,True,False,2,0.245,0.312,-0.186
 lgb,very-deep,True,False,2,0.221,0.286,0.002
 xgb,deep,False,False,0,0.257,0.326,-0.297
 lgb,deep,False,False,0,0.265,0.338,-0.394
 xgb,shallow,False,True,2,0.325,0.397,-0.928
 lgb,shallow,False,True,2,0.295,0.365,-0.628
 xgb,medium,True,False,3,0.269,0.34,-0.411
 lgb,medium,True,False,3,0.243,0.313,-0.2
 xgb,very-deep,False,True,2,0.282,0.349,-0.492
 lgb,very-deep,False,True,2,0.268,0.338,-0.395
 xgb,deep,True,True,0,0.222,0.287,-0.009
 lgb,deep,True,True,0,0.22,0.284,0.013
 xgb,medium,False,False,0,0.263,0.336,-0.377
 lgb,medium,False,False,0,0.269,0.34,-0.416
 xgb,medium,False,True,2,0.286,0.353,-0.526
 lgb,medium,False,True,2,0.268,0.337,-0.389
 xgb,deep,False,False,4,0.251,0.321,-0.259
 lgb,deep,False,False,4,0.283,0.356,-0.55
 xgb,shallow,True,True,2,0.248,0.316,-0.216
 lgb,shallow,True,True,2,0.218,0.285,0.008
 xgb,shallow,False,False,4,0.28,0.35,-0.496
 lgb,shallow,False,False,4,0.273,0.348,-0.476
 xgb,very-deep,False,True,4,0.273,0.341,-0.421
 lgb,very-deep,False,True,4,0.277,0.345,-0.458
 xgb,shallow,True,True,3,0.287,0.363,-0.61
 lgb,shallow,True,True,3,0.222,0.285,0.007
 xgb,shallow,False,True,3,0.297,0.365,-0.625
 lgb,shallow,False,True,3,0.255,0.322,-0.264
 xgb,medium,False,False,2,0.263,0.334,-0.361
 lgb,medium,False,False,2,0.265,0.339,-0.406
 xgb,very-deep,True,True,3,0.224,0.29,-0.026
 lgb,very-deep,True,True,3,0.227,0.296,-0.072
 xgb,medium,True,True,4,0.241,0.311,-0.182
 lgb,medium,True,True,4,0.227,0.297,-0.078
 xgb,deep,True,False,0,0.252,0.317,-0.229
 lgb,deep,True,False,0,0.216,0.276,0.069
 xgb,deep,False,False,2,0.243,0.31,-0.175
 lgb,deep,False,False,2,0.272,0.348,-0.483
 xgb,medium,False,True,4,0.267,0.335,-0.373
 lgb,medium,False,True,4,0.289,0.36,-0.583

xgb,very-deep,False,True,3,0.266,0.335,-0.374
 lgb,very-deep,False,True,3,0.261,0.328,-0.317
 xgb,very-deep,True,True,4,0.229,0.295,-0.062
 lgb,very-deep,True,True,4,0.243,0.309,-0.164
 xgb,very-deep,True,True,0,0.217,0.282,0.029
 lgb,very-deep,True,True,0,0.217,0.282,0.028
 xgb,medium,False,False,4,0.264,0.334,-0.363
 lgb,medium,False,False,4,0.285,0.356,-0.551
 xgb,medium,True,True,2,0.22,0.283,0.02
 lgb,medium,True,True,2,0.244,0.31,-0.174
 xgb,shallow,True,False,4,0.273,0.35,-0.497
 lgb,shallow,True,False,4,0.22,0.293,-0.052
 xgb,very-deep,True,True,2,0.227,0.291,-0.037
 lgb,very-deep,True,True,2,0.228,0.296,-0.073
 xgb,medium,True,False,0,0.261,0.328,-0.316
 lgb,medium,True,False,0,0.214,0.274,0.084
 xgb,deep,False,False,3,0.259,0.328,-0.317
 lgb,deep,False,False,3,0.286,0.362,-0.598
 xgb,very-deep,False,False,0,0.255,0.325,-0.289
 lgb,very-deep,False,False,0,0.262,0.334,-0.362

Anhang 2/2: Ergebnisse der Zufalls-Validierung

Modelltyp,Parametertiefe,Kategorisch,Aggregate,Smoothed,MAE,RMSE,R2
 xgb,shallow,True,True,0,0.285,0.358,0.193
 lgb,shallow,True,True,0,0.319,0.399,-0.0
 xgb,deep,False,True,4,0.186,0.243,0.628
 lgb,deep,False,True,4,0.269,0.336,0.292
 xgb,deep,True,False,2,0.276,0.352,0.219
 lgb,deep,True,False,2,0.299,0.373,0.126
 xgb,medium,False,True,3,0.177,0.234,0.654
 lgb,medium,False,True,3,0.257,0.321,0.352
 xgb,shallow,False,False,2,0.218,0.283,0.496
 lgb,shallow,False,False,2,0.27,0.334,0.3
 xgb,very-deep,False,True,0,0.176,0.234,0.657
 lgb,very-deep,False,True,0,0.245,0.311,0.391
 xgb,medium,True,False,2,0.272,0.348,0.238
 lgb,medium,True,False,2,0.291,0.366,0.156
 xgb,deep,True,True,3,0.278,0.353,0.216
 lgb,deep,True,True,3,0.284,0.354,0.213
 xgb,deep,True,True,2,0.294,0.366,0.158

lgb,deep,True,True,2,0.288,0.36,0.184
 xgb,medium,False,False,3,0.19,0.251,0.603
 lgb,medium,False,False,3,0.251,0.316,0.373
 xgb,shallow,False,True,4,0.198,0.257,0.584
 lgb,shallow,False,True,4,0.267,0.33,0.316
 xgb,shallow,False,False,0,0.225,0.291,0.468
 lgb,shallow,False,False,0,0.268,0.339,0.278
 xgb,shallow,True,True,4,0.333,0.407,-0.043
 lgb,shallow,True,True,4,0.298,0.375,0.117
 xgb,very-deep,False,False,3,0.184,0.244,0.625
 lgb,very-deep,False,False,3,0.26,0.324,0.341
 xgb,deep,True,False,3,0.277,0.354,0.213
 lgb,deep,True,False,3,0.304,0.377,0.107
 xgb,very-deep,True,False,3,0.283,0.361,0.18
 lgb,very-deep,True,False,3,0.295,0.367,0.152
 xgb,shallow,False,False,3,0.22,0.285,0.489
 lgb,shallow,False,False,3,0.282,0.349,0.233
 xgb,shallow,True,False,2,0.287,0.365,0.16
 lgb,shallow,True,False,2,0.304,0.381,0.085
 xgb,medium,True,False,4,0.273,0.359,0.191
 lgb,medium,True,False,4,0.295,0.37,0.138
 xgb,very-deep,True,False,4,0.287,0.37,0.138
 lgb,very-deep,True,False,4,0.301,0.375,0.115
 xgb,deep,False,True,2,0.179,0.237,0.646
 lgb,deep,False,True,2,0.259,0.326,0.332
 xgb,shallow,True,False,0,0.289,0.375,0.114
 lgb,shallow,True,False,0,0.3,0.375,0.114
 xgb,deep,True,False,4,0.278,0.36,0.183
 lgb,deep,True,False,4,0.305,0.382,0.08
 xgb,very-deep,False,False,4,0.195,0.26,0.574
 lgb,very-deep,False,False,4,0.263,0.325,0.335
 xgb,deep,False,True,0,0.182,0.243,0.628
 lgb,deep,False,True,0,0.23,0.294,0.458
 xgb,medium,False,True,0,0.188,0.248,0.612
 lgb,medium,False,True,0,0.255,0.323,0.342
 xgb,deep,False,True,3,0.184,0.24,0.638
 lgb,deep,False,True,3,0.268,0.339,0.278
 xgb,shallow,False,True,0,0.193,0.253,0.597
 lgb,shallow,False,True,0,0.266,0.335,0.294
 xgb,medium,True,True,0,0.298,0.372,0.131
 lgb,medium,True,True,0,0.287,0.357,0.196

xgb,shallow,True,False,3,0.295,0.373,0.123
 lgb,shallow,True,False,3,0.316,0.398,0.003
 xgb,very-deep,True,False,0,0.277,0.352,0.22
 lgb,very-deep,True,False,0,0.303,0.376,0.112
 xgb,deep,True,True,4,0.281,0.355,0.205
 lgb,deep,True,True,4,0.283,0.353,0.215
 xgb,very-deep,False,False,2,0.192,0.256,0.588
 lgb,very-deep,False,False,2,0.262,0.324,0.339
 xgb,medium,True,True,3,0.26,0.332,0.305
 lgb,medium,True,True,3,0.294,0.365,0.16
 xgb,very-deep,True,False,2,0.28,0.355,0.206
 lgb,very-deep,True,False,2,0.295,0.369,0.141
 xgb,deep,False,False,0,0.19,0.251,0.603
 lgb,deep,False,False,0,0.254,0.324,0.339
 xgb,shallow,False,True,2,0.196,0.256,0.589
 lgb,shallow,False,True,2,0.26,0.324,0.34
 xgb,medium,True,False,3,0.269,0.347,0.241
 lgb,medium,True,False,3,0.291,0.365,0.164
 xgb,very-deep,False,True,2,0.179,0.238,0.643
 lgb,very-deep,False,True,2,0.253,0.32,0.357
 xgb,deep,True,True,0,0.294,0.368,0.149
 lgb,deep,True,True,0,0.297,0.366,0.158
 xgb,medium,False,False,0,0.192,0.253,0.597
 lgb,medium,False,False,0,0.259,0.333,0.302
 xgb,medium,False,True,2,0.18,0.238,0.644
 lgb,medium,False,True,2,0.253,0.317,0.369
 xgb,deep,False,False,4,0.193,0.256,0.587
 lgb,deep,False,False,4,0.26,0.324,0.342
 xgb,shallow,True,True,2,0.372,0.461,-0.338
 lgb,shallow,True,True,2,0.303,0.376,0.112
 xgb,shallow,False,False,4,0.213,0.275,0.523
 lgb,shallow,False,False,4,0.285,0.348,0.24
 xgb,very-deep,False,True,4,0.187,0.25,0.608
 lgb,very-deep,False,True,4,0.259,0.325,0.336
 xgb,shallow,True,True,3,0.278,0.353,0.217
 lgb,shallow,True,True,3,0.291,0.367,0.152
 xgb,shallow,False,True,3,0.199,0.261,0.571
 lgb,shallow,False,True,3,0.252,0.314,0.381
 xgb,medium,False,False,2,0.19,0.252,0.602
 lgb,medium,False,False,2,0.256,0.322,0.347
 xgb,very-deep,True,True,3,0.28,0.354,0.21

lgb,very-deep,True,True,3,0.284,0.353,0.215
xgb,medium,True,True,4,0.268,0.344,0.257
lgb,medium,True,True,4,0.298,0.371,0.135
xgb,deep,True,False,0,0.284,0.356,0.204
lgb,deep,True,False,0,0.31,0.384,0.073
xgb,deep,False,False,2,0.196,0.261,0.572
lgb,deep,False,False,2,0.257,0.323,0.344
xgb,medium,False,True,4,0.184,0.245,0.623
lgb,medium,False,True,4,0.242,0.304,0.417
xgb,very-deep,False,True,3,0.184,0.24,0.638
lgb,very-deep,False,True,3,0.266,0.333,0.301
xgb,very-deep,True,True,4,0.297,0.373,0.126
lgb,very-deep,True,True,4,0.282,0.353,0.218
xgb,very-deep,True,True,0,0.29,0.363,0.173
lgb,very-deep,True,True,0,0.291,0.361,0.18
xgb,medium,False,False,4,0.195,0.26,0.574
lgb,medium,False,False,4,0.258,0.326,0.332
xgb,medium,True,True,2,0.273,0.344,0.254
lgb,medium,True,True,2,0.302,0.376,0.112
xgb,shallow,True,False,4,0.31,0.398,0.005
lgb,shallow,True,False,4,0.3,0.374,0.12
xgb,very-deep,True,True,2,0.292,0.364,0.167
lgb,very-deep,True,True,2,0.28,0.35,0.229
xgb,medium,True,False,0,0.279,0.355,0.209
lgb,medium,True,False,0,0.301,0.375,0.118
xgb,deep,False,False,3,0.184,0.244,0.624
lgb,deep,False,False,3,0.252,0.315,0.377
xgb,very-deep,False,False,0,0.192,0.256,0.589
lgb,very-deep,False,False,0,0.249,0.314,0.379

Literaturverzeichnis

- Alan, Ozkan (2011):** Thresholds based outlier detection approach for mining class outliers. In: *Expert Systems with Applications* 38.10, S. 12880–12889.
- Baheti, Pragati (2021):** Train Test Validation Split: How To & Best Practices. Online. URL: <https://www.v7labs.com/blog/train-validation-test-set>.
- Bengio, Yoshua; Bergstra, James (2012):** Grid Search for Hyper-Parameter Optimization. In: *Neural Information Processing Systems Workshop on Machine Learning Open Source Software*.
- Brown, Iain (2021):** Handling Outliers in ML: Best Practices for Robust Data. In: *Towards Data Science*. Online: <https://www.linkedin.com/pulse/handling-outliers-ml-best-practices-robust-data-iain-brown-ph-d--mwf6e>.
- Brownlee, Jason (2020):** Moving Average Smoothing for Data Preparation and Time Series Forecasting in Python. Online. URL: <https://machinelearningmastery.com/moving-average-smoothing-for-time-series-forecasting-python/>.
- Chatfield, Chris (2003):** The Analysis of Time Series: An Introduction. Boca Raton: CRC Press.
- Chen, Tianqi; Guestrin, Carlos (2016a):** XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM, S. 785–794.
- Chen, Tianqi; Guestrin, Carlos (2016b):** XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, S. 785–794.
- Dash, Chandra Sekhar Kumar; Behera, Ajit Kumar; Dehuri, Satchidananda; Cho, Sung-Bae (2023):** An outliers detection and elimination framework in classification task of data mining. In: *Decision Analytics Journal* 6, S. 1–14.
- Experteninterview 1 (2025):** Persönliches Interview mit Performance Engineer. Experteninterview.
- Experteninterview 2 (2025):** Persönliches Interview mit Performance Engineer. Experteninterview.
- Farek, Lazhar; Benaidja, Amira (2024):** Feature redundancy removal for text classification using correlated feature subsets. In: *Computational Intelligence* 40.3, e12621.
- Friedman, Jerome H. (2001):** Greedy Function Approximation: A Gradient Boosting Machine. In: *The Annals of Statistics* 29.5, S. 1189–1232.
- García, Salvador; Luengo, Julián; Herrera, Francisco (2015):** Data Preprocessing in Data Mining. Heidelberg: Springer.
- Géron, Aurélien (2019):** Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 2. Aufl. Sebastopol: O'Reilly Media.
- Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron (2016):** Deep Learning. Cambridge: MIT Press.

- Guyon, Isabelle; Elisseeff, Andr'e (2003):** An Introduction to Variable and Feature Selection. In: *Journal of Machine Learning Research*. Bd. 3, S. 1157–1182.
- Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2017):** The Elements of Statistical Learning. Data Mining, Inference, and Prediction. 2. Aufl. New York: Springer.
- Hodson, Timothy O. (2022):** Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not. In: *Geoscientific Model Development* 15.14, S. 5481–5487. DOI: 10.5194/gmd-15-5481-2022.
- James, Gareth; Witten, Daniela; Hastie, Trevor; Tibshirani, Robert (2021):** An Introduction to Statistical Learning. with Applications in R. 2. Aufl. New York: Springer.
- Jones, Alan (2022):** Tree-Based Models vs. Deep Learning Models for Tabular Data: A Comprehensive Investigation. In: *Journal of Machine Learning Applications* 10.4, S. 112–130.
- Ke, Guolin; Meng, Qi; Finley, Thomas; Wang, Taifeng; Chen, Wei; Ma, Weidong; Ye, Qiwei; Liu, Tie-Yan (2017):** LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In: *Advances in Neural Information Processing Systems 30*. La Jolla: Curran Associates, S. 3146–3154.
- Ke, Guolin; Meng, Qiang; Finley, Thomas; Wang, Taifeng; Chen, Wei; Ma, Weidong; Ye, Qi; Liu, Tie-Yan (2017):** LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., S. 3146–3154.
- Pedregosa, Fabian; Varoquaux, Gaël; Gramfort, Alexandre; Michel, Vincent; Thirion, Bertrand; Grisel, Olivier; Blondel, Mathieu; Prettenhofer, Peter; Weiss, Ron; Dubourg, Vincent; Vanderplas, Jake; Passos, Alexandre; Cournapeau, David; Brucher, Matthieu; Perrot, Matthieu; Duchesnay, Édouard (2011):** Scikit-learn: Machine Learning in Python. In: *Journal of Machine Learning Research* 12.1, S. 2825–2830.
- Pedregosa, Fabian; Varoquaux, Gaël; Gramfort, Alexandre; Michel, Vincent; Thirion, Bertrand; Grisel, Olivier; Blondel, Mathieu; Prettenhofer, Peter; Weiss, Ron; Dubourg, Vincent; Vanderplas, Jake; Passos, Alexandre; Cournapeau, David; Brucher, Matthieu; Perrot, Michaël; Duchesnay, Édouard (2011):** Scikit-learn: Machine Learning in Python. In: *Journal of Machine Learning Research* 12, S. 2825–2830.
- Prat, Nicolas; Comyn-Wattiau, Isabelle; Akoka, Jacky (2014):** Artifact Evaluation in Information Systems Design-Science Research: A Holistic View. In: *Proceedings of the 18th Pacific Asia Conference on Information Systems (PACIS)*. Chengdu, China, S. 1–16.
- Ribeiro, Marco Tulio; Singh, Sameer; Guestrin, Carlos (2016):** “Why Should I Trust You?” Explaining the Predictions of Any Classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, CA, USA, S. 1135–1144. DOI: 10.1145/2939672.2939778.
- Salaani, K. (2021):** An Applied Review of Simulation Validation Approaches for Vehicle Dynamics. NHTSA.
- Science, 365 Data (2023):** Measuring Explanatory Power with R-squared. URL: <https://365datascience.com/tutorials/statistics-tutorials/r-squared/>.

- Smith, John; Doe, Jane (2022):** Comparative Analysis of Random Forest and Gradient Boosting for Tabular Data. In: *International Journal of Data Science* 5.2, S. 45–58.
- Smith, Steven W. (1997):** The Scientist and Engineer’s Guide to Digital Signal Processing. 1. Aufl. San Diego: California Technical Publishing.
- Stone, Mervyn (1974):** Cross-Validatory Choice and Assessment of Statistical Predictions. In: *Journal of the Royal Statistical Society* 36.2, S. 111–147.
- Tsanas, Athanasios (2022):** Relevance, redundancy, and complementarity trade-off: A principled, generic, robust feature-selection tool. In: *Patterns* 3.5, S. 100471.
- Vapnik, Vladimir N. (2013):** The Nature of Statistical Learning Theory. 2. Aufl. New York: Springer.
- Venable, John; Pries-Heje, Jan; Baskerville, Richard (2016):** FEDS: A Framework for Evaluation in Design Science Research. In: *European Journal of Information Systems* 25.1, S. 77–89. DOI: 10.1057/ejis.2014.36.
- Willmott, Cort J.; Matsuura, Kenji (2005):** Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. In: *Climate Research* 30.1, S. 79–82. DOI: 10.3354/cr030079.
- Zheng, Alice; Casari, Amanda (2018):** Feature Engineering for Machine Learning. Principles and Techniques for Data Scientists. 1. Aufl. Sebastopol: O’Reilly Media.

Erklärung zur Verwendung generativer KI-Systeme

Bei der Erstellung der eingereichten Arbeit habe ich auf künstlicher Intelligenz (KI) basierte Systeme benutzt:

☒ ja

☐ nein

Falls ja: Die nachfolgend aufgeführten auf künstlicher Intelligenz (KI) basierten Systeme habe ich bei der Erstellung der eingereichten Arbeit benutzt:

- 1.
- 2.
3. ...

Ich erkläre, dass ich

- mich aktiv über die Leistungsfähigkeit und Beschränkungen der oben genannten KI-Systeme informiert habe,
- die aus den oben angegebenen KI-Systemen direkt oder sinngemäß übernommenen Passagen gekennzeichnet habe,
- überprüft habe, dass die mithilfe der oben genannten KI-Systeme generierten und von mir übernommenen Inhalte faktisch richtig sind,
- mir bewusst bin, dass ich als Autorin bzw. Autor dieser Arbeit die Verantwortung für die in ihr gemachten Angaben und Aussagen trage.

Die oben genannten KI-Systeme habe ich wie im Folgenden dargestellt eingesetzt:

Arbeitsschritt in der wissenschaftlichen Arbeit	Eingesetzte(s) KI-System(e)	Beschreibung der Verwendungsweise

(Ort, Datum)

(Unterschrift)

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit mit dem Thema: *Mein Titel* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

(Ort, Datum)

(Unterschrift)