

Data Exploration Project

Predictive Maintenance

WWI23B4 Gruppe 8

Im Rahmen der Prüfung zum Bachelor of Science (B. Sc.)

Daniel Weissenberger - 1500846

Eduardo Stein Mössner - 7561136

Jonas Sigmund - 6499699

25.03.2025

Betreuer: Prof. Dr. Jennifer Schoch

Inhaltsverzeichnis

Abkürzungsverzeichnis	III
Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
1 Einleitung	1
1.1 Zielsetzung	1
1.2 Vorgehensweise	2
2 Business Understanding	4
2.1 Literaturrecherche	4
2.2 Code-Recherche	7
3 Data Understanding	8
3.1 Datensatz	8
3.2 Statistische Analyse der Merkmale	8
3.3 Erkenntnisse der Visualisierungen	8
4 Data Preparation	13
4.1 Datenimport und Typanpassung	13
4.2 Datenaufteilung und Oversampling	14
4.3 Datenstandardisierung und Speicherung	14
5 Modelling	16
5.1 Problemformulierung	16
5.2 Modellauswahl	16
5.3 Hyperparameter-Tuning	17
5.4 Prediction und Inference	17
6 Evaluation	19
6.1 Modellvergleich auf Basis von Metriken	19

6.2	Bias-Variance-Tradeoff	20
6.3	Wahl des besten Modells	21
6.3.1	Scatterplot-Vergleich der Trainings- und Testmetriken	21
6.4	Einfluss des Oversamplings und Diskussion zur Modellperformance	22
6.5	Fazit der Evaluation	24
7	Fazit	25
	Anhang	27
	Hilfsmittelverzeichnis	27
	Literatur	28

Abkürzungsverzeichnis

KI	Künstliche Intelligenz
ML	Maschinelles Lernen
PdM	Predictive Maintenance
IoT	Internet of Things
EDA	explorativen Datenanalyse
CRISP-MLQ	Cross-Industry Standard Process for Machine Learning with Quality Assurance
SMOTE	Synthetic Minority Over-sampling Technique
TWF	Tool Wear Failure
HDF	Heat Dissipation Failure
PWF	Power Failure
OSF	Overstrain Failure
RNF	Random Failures
SVM	Support vector machines
KNN	k-nearest neighbor

Abbildungsverzeichnis

1	Bestandteile des Business Intelligence	4
2	Maschinen-Fehler nach Maschinentyp	10
3	Fehleranteile nach Maschinentyp	10
4	Lufttemperatur und Fehler-Typen	11
5	Prozesstemperatur und Fehler-Typen	11
6	Rotationsgeschwindigkeit und Fehler-Typen	12
7	Drehmoment und Fehler-Typen	12
8	Werkzeugverschleiß und Fehler-Typen	12
9	Korrelationsmatrix Maschinen-Typ	12
10	Train-Test Vergleich der F1-Scores	21
11	Boxplot der Oversampelten Trainingsdaten für Lufttemperatur in Bezug auf die Fehlertypen	23
12	Boxplot der Oversampelten Trainingsdaten für Prozesstemperatur in Be- zug auf die Fehlertypen	23
13	Boxplot der Oversampelten Trainingsdaten für Rotationsgeschwindigkeit in Bezug auf die Fehlertypen	23
14	Boxplot der Oversampelten Trainingsdaten für Drehmoment in Bezug auf die Fehlertypen	23
15	Boxplot der Oversampelten Trainingsdaten für Werkzeugverschleiß in Be- zug auf die Fehlertypen	24

Tabellenverzeichnis

2	Beschreibung der Variablen und deren Eigenschaften	9
3	Lage- und Streuungsmaße der metrischen Variablen	9
4	Vergleich der Modelle hinsichtlich Prediction und Inference	18
5	Performance der Modelle auf den Trainingsdaten	19
6	Performance der Modelle auf den Testdaten	19
7	Modellparameter zur Bewertung der Flexibilität	20

1 Einleitung

Die vierte industrielle Revolution, auch bekannt als Industrie 4.0, beschreibt die zunehmende Digitalisierung und Vernetzung im Fertigungssektor. Ziel dieser Entwicklung ist es, durch den Einsatz moderner Technologien wie Internet of Things (IoT), Big-Data-Analysen und Künstliche Intelligenz (KI) betriebliche Prozesse zu optimieren. Die damit verbundenen Herausforderungen umfassen insbesondere die Reduktion von Kosten, die Steigerung der Effizienz sowie eine verbesserte Ressourcennutzung vgl. Alenizi et al., 2023; Hoffmann, M.A., Lasch, R., 2025.

Ein zentraler Bestandteil dieser technologischen Transformation ist der Bereich des Maschinellen Lernen (ML), insbesondere im Kontext von Predictive Maintenance (PdM). PdM zielt darauf ab, durch die Analyse von Sensordaten potenzielle Maschinenfehler frühzeitig zu erkennen und dadurch ungeplante Ausfallzeiten zu minimieren. Die vorliegende Arbeit beschäftigt sich daher mit der Fragestellung: **Wie genau lassen sich Fehlerursachen einer Maschine mithilfe von Klassifikationsmodellen auf Basis von Sensordaten identifizieren?**

1.1 Zielsetzung

Das Ziel dieser Arbeit ist es, auf Basis eines geeigneten Datensatzes verschiedene Machine-Learning-Modelle zu implementieren, die unterschiedliche Fehlerursachen einer Maschine anhand von Sensordaten klassifizieren können. Um eine fundierte Analyse und optimiertes Modelltraining zu gewährleisten, werden Techniken der explorativen Datenanalyse (EDA) sowie ein umfangreiches Modeling mit unterschiedlichen Machine-Learning-Algorithmen durchgeführt. Die Qualität der entwickelten Modelle wird durch eine umfassende Evaluation bestimmt, die sich insbesondere auf den *F1-Score* stützt.

Im Folgenden wird die Zielsetzung gemäß dem SMART-Prinzip konkretisiert:

- **Spezifisch (S):** Entwicklung eines optimalen Machine-Learning-Modells zur Klassifikation verschiedener Fehlerursachen auf Grundlage von Maschinendaten.
- **Messbar (M):** Bewertung der Modellgüte anhand der Metrik *F1-Score*.

- **Erreichbar (A):** Methodische Vorgehensweise einschließlich umfassender EDA.
- **Relevant (R):** Beitrag zur Entwicklung von Modellen für die PdM und damit zur Reduktion von Maschinenausfällen.
- **Zeitgebunden (T):** Umsetzung innerhalb eines Zeitraums von sechs Wochen mit abschließender Evaluierung in Woche sieben.

1.2 Vorgehensweise

Die Vorgehensweise dieser Arbeit orientiert sich am Cross-Industry Standard Process for Machine Learning with Quality Assurance (CRISP-MLQ), einem etablierten Prozessmodell für die strukturierte Entwicklung von Modellen im Bereich des ML. Arbeitsschritte wie das *Deployment* und *Monitoring* des CRISP-MLQ-Prozesses können in dieser Arbeit vernachlässigt werden, da es sich bei der Arbeit lediglich um eine wissenschaftliche Abhandlung handelt. Die Projektdauer beträgt neun Wochen, beginnend am 04.02.2025 und endet mit der finalen Abgabe am 06.04.2025. Die einzelnen Aufgaben wurden gleichmäßig unter den Gruppenmitgliedern verteilt, um eine effiziente Bearbeitung der jeweiligen Phasen sicherzustellen.

Woche	Phase	Inhalt
1	Business Understanding	<ul style="list-style-type: none">▪ Definition der Forschungsfrage und der Ziele▪ Recherche über bestehende Projekte▪ Durchführung einer Literaturrecherche
2	Data Understanding	<ul style="list-style-type: none">▪ Erstellung eines Daten-Überblicks▪ Explorative Datenanalyse und Visualisierung▪ Bewertung der Datenqualität▪ Ableitung erster Erkenntnisse

3	Data Preparation	<ul style="list-style-type: none">▪ Aufteilung in Trainings- und Testdaten▪ Oversampling der Daten▪ Standardisierung der Daten
4–5	Modeling	<ul style="list-style-type: none">▪ Training verschiedener Modellklassen▪ Hyperparameter-Tuning der Modelle
6	Evaluation	<ul style="list-style-type: none">▪ Vergleich der Modellleistung

Die strukturierte Planung gewährleistet eine systematische und zielgerichtete Bearbeitung des Projekts und stellt zugleich die Relevanz und Korrektheit der Ergebnisse sicher. Diese Vorgehensweise wird unterstützt durch theoretische Grundlagen und eine detaillierte Analyse des verwendeten Datensatzes, welche in den folgenden Kapiteln behandelt werden.

2 Business Understanding

2.1 Literaturrecherche

Der Prozess, Vorhersagen in verschiedenen Bereichen anhand von Daten und deren Potenzial zu treffen, wird „Predictive Analytics“ genannt. Somit fällt unsere gesamte Arbeit unter diesen Begriff. Predictive Analytics wird in ähnlichen Ansätzen bereits seit den 1950er Jahren verwendet. Banken versuchten damals, die Kreditwürdigkeit ihrer Kunden festzustellen, vgl. Markus Seebacher, 2018, S. 1–4.

Predictive Analytics ist jedoch nicht mit Business Intelligence zu verwechseln. Um diese beiden Begriffe voneinander abzugrenzen, betrachten wir eine Abbildung und die Einschätzung von Wayne W. Eckerson (2007). Wayne W. Eckerson (2007) beschreibt Predictive Analytics als Teilbereich von Business Intelligence, in dem Muster und Zusammenhänge in großen Datenmengen identifiziert und darauf aufbauend Vorhersagen generiert werden. Die Abbildung zeigt außerdem, dass Predictive Analytics insbesondere den Bereich „Prediction“ innerhalb von Business Intelligence abdeckt, nicht jedoch Monitoring, Analysis und Reporting. Abbildung 1

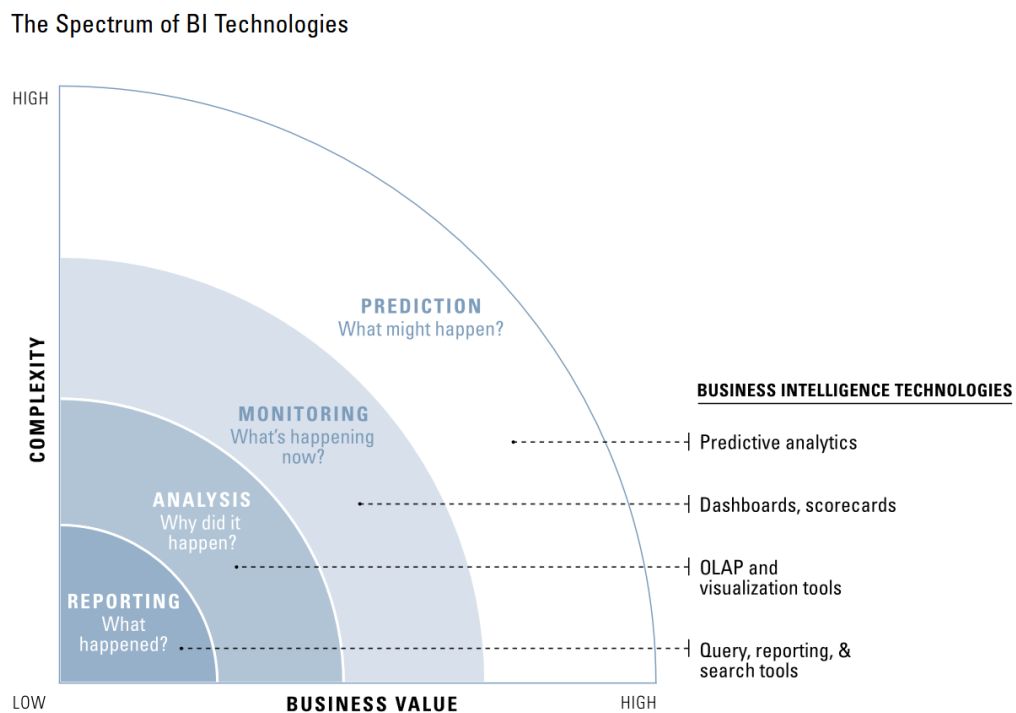


Abb. 1: Bestandteile des Business Intelligence

Ein häufiges Anwendungsziel von Predictive Analytics ist die präventive Fehlerbehebung, um die Ausfallzeiten von Maschinen zu reduzieren. Der entsprechende Fachbegriff hierfür lautet „Predictive Maintenance“ und stellt eine Unterkategorie von Predictive Analytics dar.

Im Zuge der Datenanalyse wird zunächst „Data Mining“ betrieben und anschließend mithilfe von Machine-Learning-Modellen klassifiziert, wann und welche Fehler auftreten. Die Ergebnisse des besten Modells werden anschließend erläutert und interpretiert. Der Prozess des „Data Minings“ wird von Markus Seebacher (2018) im Abschnitt „3.1 Datenabruf und Formatierung“ umfassend beschrieben, auch wenn er diesen nicht explizit als Data Mining bezeichnet.

Er beschreibt die Datenaufbereitung (Data Preprocessing), die Datenbeschreibung (Data Understanding), die Datenkorrelation (Pattern Discovery) sowie die Datenvisualisierung (Data Visualization). Innerhalb der Datenaufbereitung geht er zudem auf die Unterprozesse Datenbereinigung (Data Cleaning), Dateninterpretation (Data Interpretation), Datenreduktion (Data Reduction) und Datentransformation (Data Transformation) ein.

In der Datenverarbeitung werden fehlende Werte, Ausreißer und Duplikate identifiziert und – wenn möglich – bereinigt. Weitere Schritte beinhalten die Zusammenführung verschiedener Datenquellen, das Entfernen irrelevanter Variablen sowie die Normalisierung oder Glättung der Daten zur Verbesserung der Datenqualität. Die Clusterbildung und Kategorisierung der Daten erfolgt in der Phase der Datenbeschreibung. Dafür können unter anderem Verfahren wie K-Means oder neuronale Netze eingesetzt werden. Auf Basis dieser Strukturen lassen sich Zusammenhänge erkennen, die für spätere Vorhersagen verwendet werden können – dieser Schritt wird als Datenkorrelation bezeichnet. In der Datenvisualisierung werden die Analyseergebnisse – wie beispielsweise die Anzahl fehlender Werte – in Diagrammen dargestellt. Diese Darstellungen dienen als Entscheidungsgrundlage für weitere Maßnahmen, vgl. Markus Seebacher, 2018.

Im aktuellen Forschungsstand zur „Predictive Maintenance“ liegt der Schwerpunkt vor allem auf datengetriebenen und hybriden Ansätzen, während regelbasierte Methoden seltener betrachtet werden. Zur Überwachung des Maschinenzustands werden häufig Daten

zu Vibration, Kraft oder Schall verwendet. Diese Informationen stammen entweder aus bereits vorhandenen Sensoren oder werden durch zusätzliche Sensorik erfasst. Beispielsweise untersuchten Daniel Frank Hesser und Bernd Markert (2019) die Nachrüstung einer CNC-Fräsmaschine mit einer Sensorplattform zur Erfassung von Vibrationsdaten für die Verschleißerkennung, vgl. Giuseppe Ferrera, 2021.

Ein weiterer zentraler Forschungsbereich ist die Analyse von Defekten in CNC-Fräsmaschinen, insbesondere bei Fräsen und Kugellagern. „Run-to-Failure“-Experimente werden in mehreren Studien eingesetzt, um Trainingsdaten zu erzeugen, die anschließend mit verschiedenen maschinellen Lernverfahren zur Klassifikation von Verschleiß oder zur Vorhersage der Restlebensdauer verwendet werden, vgl. Giuseppe Ferrera, 2021. Jain A. K. and Chouksey P. and Parlikad A. K. and Lad B. K (2020) zeigen beispielsweise, dass unterschiedliche Sensordatentypen jeweils spezifische Algorithmen erfordern: Naive Bayes für Kraftsensoren, logistische Regression für Vibrationsdaten und Random Forest für akustische Daten. Hybride Ansätze kombinieren mathematische Maschinenmodelle mit Sensordaten, um präzisere Vorhersagen über die verbleibende Lebensdauer zu ermöglichen, vgl. Giuseppe Ferrera, 2021.

Neben der CNC-Fertigung existieren auch Anwendungen in der Halbleiterproduktion, wo etwa Fuzzy-Clustering und Support-Vector Machines zur Fehlererkennung und -klassifikation verwendet werden. Auch alternative Datenquellen wurden untersucht: Koca O. and Kaymakci O. T. and Mercimek M. (2020) nutzten beispielsweise ERP-Daten statt Sensordaten, um mit neuronalen Netzen Maschinenausfälle vorherzusagen, vgl. Giuseppe Ferrera, 2021.

Insgesamt zeigt sich, dass datengetriebene Methoden dominieren. Maschinelles Lernen wird sowohl für Klassifikations- als auch Regressionsprobleme eingesetzt. Überwachtes Lernen wird bevorzugt, sofern gelabelte Daten verfügbar sind, da es präzisere Ergebnisse liefert. Unüberwachte Verfahren kommen primär dann zum Einsatz, wenn keine Ausfalldaten vorliegen, jedoch Prozessdaten genutzt werden können, vgl. Giuseppe Ferrera, 2021.

Für „Predictive Maintenance“ sind Daten essenziell – in der Regel Sensordaten, die

entweder gespeichert oder in Echtzeit verarbeitet werden. Im Zuge der Digitalisierung von Unternehmen und Wertschöpfungsketten wurde die Verfügbarkeit solcher Daten massiv erhöht, vgl. Wolfgang Becker und Felix Schuhknecht, 2020, S. 294. Dies hatte auch Auswirkungen auf Geschäftsmodelle. Einige Unternehmen entwickelten sich hin zu sogenannten „digitalen Geschäftsmodellen“, bei denen ein Großteil der Wertschöpfung auf digitalisierten Prozesselementen basiert, vgl. B. Jahn und M. Pfeiffer, 2014, S. 86. Predictive-Maintenance-Strategien sind ein Beispiel für digital ermöglichte Ansätze, die das Ziel verfolgen, das Verschleißverhalten von Maschinen zu prognostizieren und Ausfälle präventiv zu vermeiden. Predictive Maintenance gilt daher auch als Schlüsselfaktor innerhalb der Industrie 4.0, vgl. S. Zhai und G. Reinhart, 2018, S. 294.

Die Verringerung von störfallbedingten Unterbrechungen führt zu signifikanten Kosteneinsparungen und verbessert die Effizienz innerhalb der Unternehmensprozesse, vgl. Gottmann, 2019, S. 163.

Für Predictive Maintenance können je nach Zielstellung sowohl Regressions- als auch Klassifikationsverfahren eingesetzt werden. Regressionsmodelle eignen sich beispielsweise für Zeitreihenanalysen, mit denen die verbleibende Betriebsdauer eines Bauteils prognostiziert werden kann. Klassifikationsverfahren hingegen sind nützlich, wenn es darum geht, den wahrscheinlich auftretenden Fehler zu identifizieren. Je nach Anwendungsfall muss individuell entschieden werden, welcher Ansatz geeigneter ist.

2.2 Code-Recherche

Da es sich bei dem Datensatz um ein Paradebeispiel für die PdM handelt, existieren bereits zahlreiche Analysen und Machine-Learning-Projekte. Auf Kaggle sind etwa 100 Notebooks verfügbar, von denen rund 25 relevante Ansätze wie Oversampling oder Klassifikation beinhalten. Diese fokussieren sich jedoch meist auf isolierte Teilschritte. Ziel dieser Arbeit war es hingegen, durch Kombination mehrerer Methoden ein leistungsfähiges Gesamtkonzept zu entwickeln. Darüber hinaus existiert ein GitHub-Repository mit Klassifikationsmodellen, welches jedoch keine tiefergehenden Analysen oder wissenschaftliche Einordnung bietet.

3 Data Understanding

3.1 Datensatz

Diese Arbeit verwendet einen synthetisch generierten Datensatz über die Sensordaten von Fräsmaschinen. Er umfasst dabei verschiedene numerische und kategorische Merkmale, die in der Tabelle 2 hinsichtlich ihrer Bedeutung, Einheit, Skalenniveau und Rolle beschrieben werden. Da es sich um einen synthetisch generierten Datensatz handelt, besitzt er eine optimale Datenqualität. So sind keine fehlenden oder fehlerhafte Werte und Duplikate im Datensatz zu finden. Alle 10.000 Datenpunkt haben einen logischen Zusammenhang könnten unter Realbedingungen tatsächlich in der Domäne auftreten. Diese Aspekte wurden durch die Verwendung eines Random-Walk-Prozesses gewährleistet, der mit einer normalverteilten Zufallsvariable rekursiv die Datenpunkte synthetisiert. Der Datensatz kann von dem UC Irvine Machine Learning Repository bezogen werden.

3.2 Statistische Analyse der Merkmale

Zur besseren Einschätzung der Variablen wurden verschiedene Lage- und Streuungsmaße berechnet. Tabelle 3 gibt eine Übersicht über den Mittelwert, die Standardabweichung, das Minimum, das Maximum sowie das 25%- und 75%-Quantil für die metrischen Variablen. Auffällig ist, dass die Drehzahl eine relativ große Streuung aufweist, was potenziell auf eine starke Maschinenbeanspruchung hinweisen könnte. Die Temperaturvariablen liegen im erwarteten Bereich, während die Werte für das Drehmoment (Torque) auf eine moderate bis hohe Belastung hindeuten.

3.3 Erkenntnisse der Visualisierungen

Zur explorativen Datenanalyse wurden verschiedene Visualisierungen erstellt, um die Verteilung der Merkmale und deren Beziehung zu den Fehlerklassen zu untersuchen. Die folgenden Diagramme liefern wertvolle Erkenntnisse über mögliche Zusammenhänge in den Sensordaten.

Variable	Beschreibung	Skalenniveau	Typ
UID	Eindeutige Kennung (1–10.000)	Nominal	-
product ID	Spezifische Seriennummer	Nominal	-
type	Produktvariante (L/M/H)	Nominal	Input
air temperature [K]	Lufttemperatur, normalisiert um 300 K ($\sigma = 2$ K)	Metrisch	Input
process temperature [K]	Berechnet als Lufttemperatur + 10 K ($\sigma = 1$ K)	Metrisch	Input
rotational speed [rpm]	Drehzahl, berechnet aus 2860 W mit normalverteilter Störung	Metrisch	Input
torque [Nm]	Drehmoment, normalverteilt um 40 Nm ($\sigma = 10$ Nm)	Metrisch	Input
tool wear [min]	Werkzeugverschleiß, beeinflusst durch Produktqualität (H: +5 min, M: +3 min, L: +2 min)	Metrisch	Input
machine failure	Maschinenausfall (1 = Ausfall, 0 = kein Ausfall)	Dichotom	Output
Tool Wear Failure (TWF)	Ausfall bei Werkzeugverschleiß von 200–240 min	Dichotom	Output
Heat Dissipation Failure (HDF)	Ausfall, wenn $\Delta T < 8.6$ K und Drehzahl < 1380 rpm	Dichotom	Output
Power Failure (PWF)	Ausfall, wenn Leistung < 3500 W oder > 9000 W	Dichotom	Output
Overstrain Failure (OSF)	Ausfall, wenn Werkzeugverschleiß und Drehmoment Schwellenwert überschreiten	Dichotom	Output
Random Failures (RNF)	Zufälliger Prozessausfall (Wahrscheinlichkeit 0,1%)	Dichotom	Output

Tab. 2: Beschreibung der Variablen und deren Eigenschaften

Variable	Mittelwert	Std.-Abw.	Min	25%-Q	75%-Q	Max
air temperature [K]	300.1	2.0	295.1	298.6	301.5	304.0
process temperature [K]	310.1	1.0	308.0	309.5	310.7	312.0
rotational speed [rpm]	1500.5	300.7	900	1320	1680	2000
torque [Nm]	40.2	9.8	20.0	33.5	45.8	60.0
tool wear [min]	100.3	50.5	10	75	125	200

Tab. 3: Lage- und Streuungsmaße der metrischen Variablen

Balkendiagramm: Zeigt die Verteilung der Maschinen-Fehler nach dem Maschinen Typ. Diesem Diagramm ist zu entnehmen, dass es eine starke Unausgeglichenheit für die Defekte einer Maschine gibt, da nur ca. 3,39% der Maschinen einen Defekt aufwiesen. Dies ist die erste Unausgeglichenheit, die später in der Data-Preparation berücksichtigt werden muss. Abbildung 2

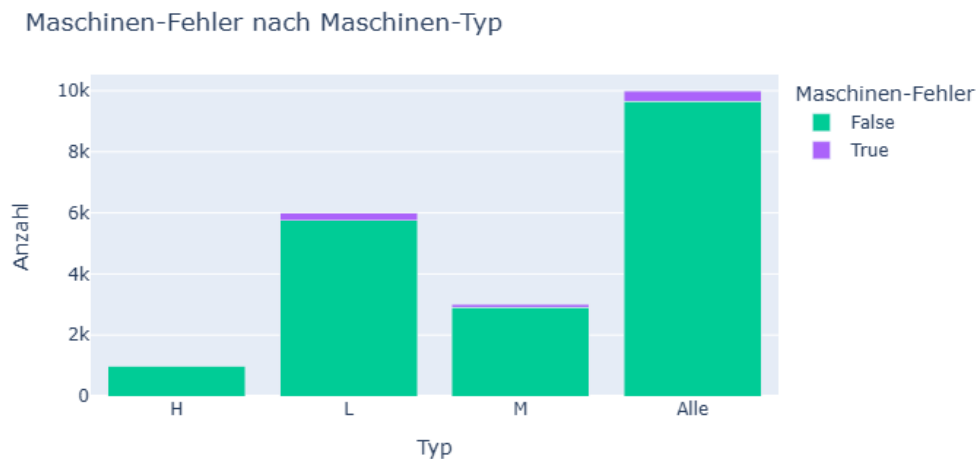


Abb. 2: Verteilung der Maschinenfehler nach Maschinentyp

Kreisdiagramme: Zeigen die Verteilung der individuellen Fehlerklassen je Maschinen-Typ. Dabei wird die zweite Unausgeglichenheit des Datensatzes deutlich, da selbst die individuellen Fehler-Klassen sehr unausgeglichen sind. Auch diese Class-Imbalance muss in der Data-Preparation adressiert werden. Abbildung 3

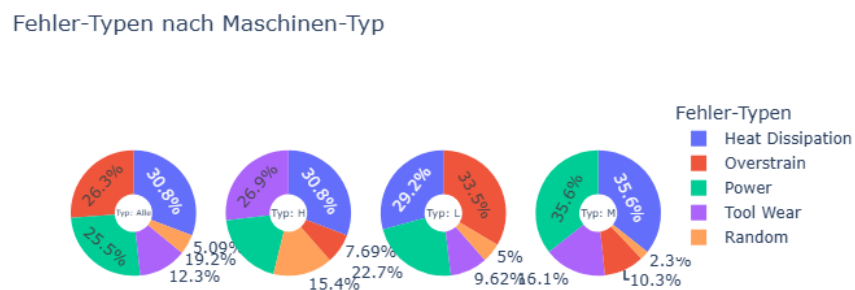


Abb. 3: Anteile der Fehlerarten nach Maschinentypen

Scatter und Boxplots: Zeigen, zunächst die Verteilung der metrischen Variablen ergänzt um die Maschinen-Fehler. Der rechten Seite ist die Verteilung der Fehlerklassen für das jeweilige Merkmal zu entnehmen. Insgesamt zeigt es sich, dass bestimmte Fehlerarten bei spezifischen Sensorwerten häufiger auftreten:

- *HDF* tritt wie erwartet vermehrt bei hohen Temperaturen ($T > 300$ K) auf.
(vgl. Abbildung 4)
- *PWF* tritt hauptsächlich bei hohen Drehzahlen (> 1900 rpm) auf.
(vgl. Abbildung 6)
- *TWF* und *OSF* treten bei starker Abnutzung (tool wear > 172 min) auf.
(vgl. Abbildung 8)

Abbildung 4

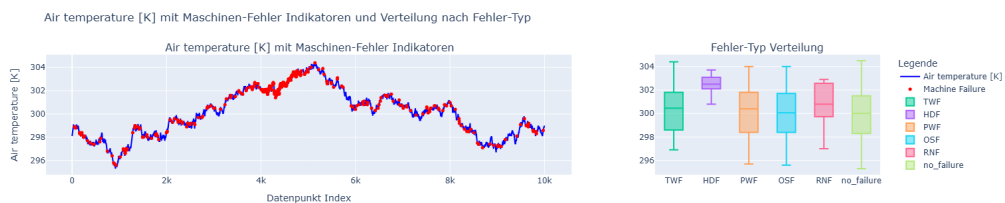


Abb. 4: Lufttemperatur in Bezug auf die Fehlerklassen

Abbildung 5

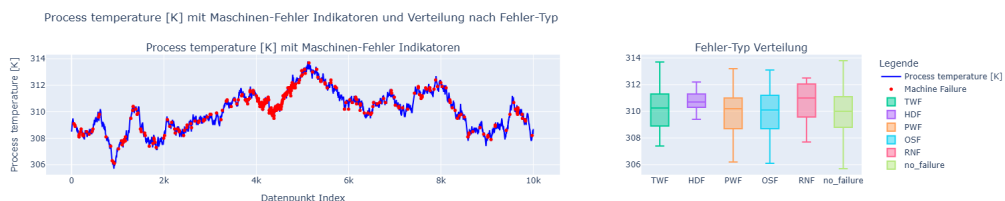


Abb. 5: Prozesstemperatur in Bezug auf die Fehlerklassen

Abbildung 6

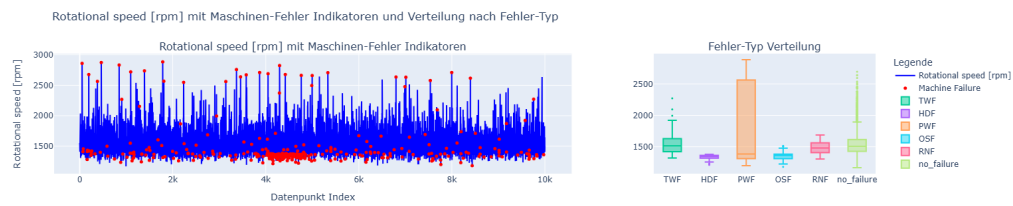


Abb. 6: Rotationsgeschwindigkeit in Bezug auf die Fehlerklassen

Abbildung 7

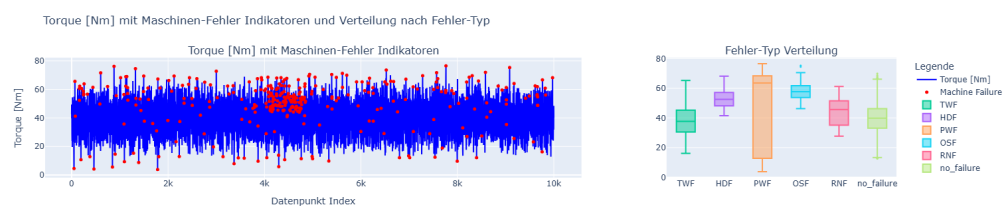


Abb. 7: Drehmoments in Bezug auf die Fehlerklassen

Abbildung 8

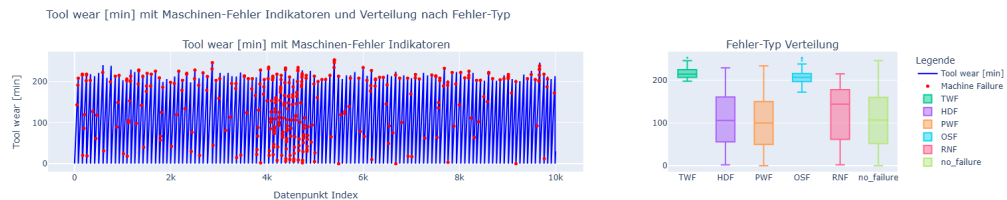


Abb. 8: Werkzeugverschleißes in Bezug auf die Fehlerklassen

Korrelationsmatrix: Die Korrelationsmatrix in Abbildung 9 zeigt, dass *Torque* und *Rotational Speed* mit einem Korrelationskoeffizienten von -0.88 stark negativ korrelieren. Des Weiteren wird deutlich, dass die *Process temperature* und *Air temperature* trivialer Weise einen starken Einfluss aufeinander haben.

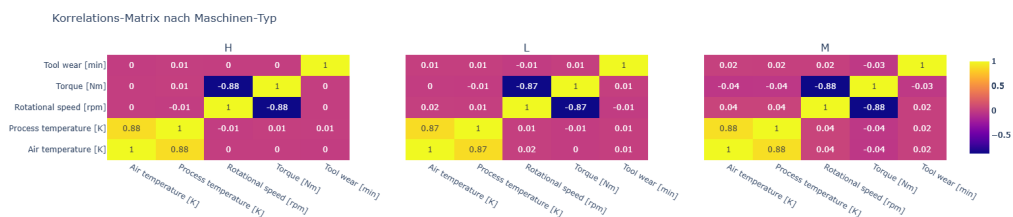


Abb. 9: Korrelationsmatrix zwischen Variablen in Abhängigkeit vom Maschinentyp

4 Data Preparation

Die Datenaufbereitung ist ein essenzieller Schritt für datengetriebene Projekte. Dafür werden die Rohdaten sortiert, gefiltert, abgeleitet und neu strukturiert um eine möglichst passende Form für die anschließenden Schritte zu erhalten. Eine geeignete Datenbasis ist entscheidend um verlässliche Ergebnisse in der Analyse oder dem Modeling zu erzielen.

4.1 Datenimport und Typanpassung

Um die Datentypen der Merkmale gleich richtig zu setzen, werden diese beim Einlesen des Datensatzes manuell der Methode `pandas.read_csv()` übergeben. Dies ist insbesondere notwendig, da die Bibliothek *pandas* die Datentypen nicht selbst optimal erkennt. So werden beispielsweise Merkmale, die als Dichotom mit $x \in \{0; 1\}$ vorliegen als Gleitkommazahl erkannt. Durch die manuelle Typisierung kann nun besser zwischen numerischen Variablen (*float32*), kategorialen Variablen (*category*) und binären Merkmalen (*bool*) Unterschieden werden.

Um das Merkmal *Machine-Type* optimal im Modelling zu verwenden, wurden 3 **Dummy-Variablen** abgeleitet. Zwar würden $n - 1 = 3 - 1 = 2$ **Dummy-Variablen** ausreichen, jedoch ist es für die Übersichtlichkeit und Interpretierbarkeit vorteilhaft, alle 3 Variablen separat anzugeben. Hierbei wurden die ursprünglichen Kategorien *L*, *M* und *H* in die binären Merkmale (*Type_H*, *Type_M*, *Type_L*) transformiert, sodass jede Kategorie durch das *One-Hot-Encoding* als individuelle Spalte repräsentiert wird. Dieser Schritt ist für viele maschinellen Lernverfahren notwendig, da diese mit numerischen Eingangs-Variablen arbeiten und kategoriale Merkmale nicht direkt abbilden können.

Darüber hinaus wurde eine neue Zielvariable *label* generiert, welche die Maschinenfehler in 5 verschiedene Klassen (TWF, HDF, PWF, OSF, RNF) unterteilt. Zusätzlich wurde die Kategorie *no_failure* eingeführt, um den fehlerfreien Zustand der Maschinen zu repräsentieren. Diese Kombination vereinfacht das spätere Training der Modelle ungemein, da eine einfache und direkte kategoriale Ausgabe-Variable vorliegt, die alle möglichen Klassen umfasst.

4.2 Datenaufteilung und Oversampling

Für die Modellierung wurden die Daten in **Trainings- und Testdaten** unterteilt, wobei ein *80:20 Split* verwendet wurde. Dadurch kann ein Großteil der Daten für das Training verwendet werden, während die restlichen Trainingsdaten zur Überprüfung der Generalisierungsfähigkeit des Modells verwendet werden.

Nun können wir uns auch der zuvor erwähnten Klassenunausgeglichenheit widmen. Dafür wird ein *Oversampling* der Trainingsdaten vorgenommen. Die Testdaten werden nicht *Oversampled*, da die Evaluation der Modelle nur auf den originalen Daten vorgenommen werden soll. Deshalb erfolgt das *Oversampling* auch nach der Trennung in Trainings- und Testdaten. Für das *Oversampling* wird auf das Verfahren *SMOTETomek* des Frameworks *imbalanced-learn (imblearn)* zurückgegriffen. *SMOTETomek* kombiniert dabei zwei Verfahren: Zum einen werden durch Synthetic Minority Over-sampling Technique (SMOTE) neue Datenpunkte für Minderheitsklassen synthetisiert, um das Gleichgewicht der Klassen zu verbessern. Dafür wird zunächst ein zufälliger Datenpunkt einer Minderheitsklasse ausgewählt. Anschließend werden die nächsten n Datenpunkte einer Minderheitsklasse ermittelt und anschließend wird zwischen diesen zwei Punkten interpoliert. Des Weiteren werden mit *Tomek Links* redundante Datenpunkte der Mehrheitsklasse entfernt, die nahe an der Entscheidungsgrenze liegen. Durch diese Kombination wurde eine ausgewogene Trainingsbasis von über 40.000 ausgeglichenen Datenpunkten (statt den $10.000 * 80\% = 8.000$ Rohdatenpunkten) geschaffen, die die Modelle befähigt, auch seltene Fehlerklassen gut zu identifizieren.

4.3 Datenstandardisierung und Speicherung

Die metrischen Eingangs-Variablen wurden mithilfe der **StandardScaler**-Klasse normalisiert. Dabei werden die Daten so transformiert, dass sie einen Mittelwert von 0 und eine Standardabweichung von 1 aufweisen und alle Features auf dasselbe Niveau gebracht werden. Dies ist insbesondere für Distanz-basierte Algorithmen wie Support vector machines (SVM) und k-nearest neighbor (KNN) notwendig, da die Berechnung der Distanzen und Entscheidungsgrenzen sonst Merkmale mit größeren Skalenniveaus dis-

kriminieren würde. Abschließend werden die aufbereiteten Daten in zwei separate CSV-Dateien exportiert: *dataset_train_resampled.csv* und *dataset_test.csv*. Diese Schritte der Datenaufbereitung gewährleisteten eine konsistente und qualitativ hochwertige Basis für das folgende Modelltraining, um eine präzise Analyse und aussagekräftige Ergebnisse zu erhalten.

5 Modelling

5.1 Problemformulierung

Bevor mit der Modellierung begonnen werden konnte, musste die Art des maschinellen Lernproblems bestimmt werden. Da die Zielvariable *label* sechs verschiedene Kategorien umfasst (*TWF*, *HDF*, *PWF*, *OSF*, *RNF* und *no_failure*), handelt es sich um ein **mehrklassiges Klassifikationsproblem mit sechs Klassen**. Das Ziel ist nun die Erstellung von Modellen, die Anhand der Eingangs-Variablen (*Type_L*, *Type_M*, *Type_H*, *air temperature [K]*, *process temperature [K]*, *rotational speed [rpm]*, *torque [Nm]*, *tool wear [min]*) eine entsprechende Ausgabe-Klasse ermitteln.

5.2 Modellauswahl

Basierend auf den Anforderungen wurden die folgenden sieben Modelle trainiert:

- **Logistic Regression** – Lineares Modell, das Wahrscheinlichkeiten berechnet, um Klassen optimal voneinander zu trennen.
- **Random Forest** – Ensemble aus mehreren Entscheidungsbäumen, die gemeinsam robustere und genauere Klassifikationen ermöglichen.
- **Support Vector Machine (SVM)** – Algorithmus zur Ermittlung der optimalen Entscheidungsgrenze, die Klassen durch maximierten Abstand am besten trennt.
- **K-Nearest Neighbors (KNN)** – Nicht-parametrischer Algorithmus, der Klassifikationen basierend auf der Ähnlichkeit zu den nächstgelegenen Nachbarn vornimmt.
- **Decision Tree** – Einfacher Baum-basierter Algorithmus, der mittels hierarchischer Entscheidungsregeln Klassen direkt zuordnet.
- **KNN mit optimalem K** – Erweiterte KNN-Variante, bei der die optimale Anzahl der Nachbarn für eine bestmögliche Klassifikation ermittelt wird.
- **Geprunter Decision Tree** – Entscheidungsbaum mit Cost-Complexity-Pruning, um Komplexität zu reduzieren und Overfitting entgegenzuwirken.

5.3 Hyperparameter-Tuning

Zur Optimierung der Vorhersagequalität der eingesetzten Modelle wurde ein systematisches Hyperparameter-Tuning durchgeführt. Dabei wurden zunächst relevante Hyperparameter der eingesetzten Algorithmen identifiziert, deren Variation die Leistung der Modelle wesentlich beeinflussen kann. Für den Random-Forest-Klassifikator wurden insbesondere die Anzahl der Bäume *n_estimators*, die maximale Baumtiefe *max_depth*, sowie die Mindestanzahl der benötigten Stichproben für eine Teilung *min_samples_split* und zur Bildung eines Blattes *min_samples_leaf* berücksichtigt. Zur effizienten Ermittlung der optimalen Parameter wurde eine Grid-Search in Verbindung mit einer k-fachen Kreuzvalidierung (5-fach *Cross-Validation*) durchgeführt. Hierfür wurde ein Parameterraum definiert, der verschiedene realistische Ausprägungen der Hyperparameter umfasst. Neben dem Random-Forest wurde auch ein identisches Vorgehen verwendet, um die beste Anzahl an Nachbarn *k* (*n_neighbours*) für das K-Nearest-Neighbors-Modell zu finden. Im Fall des Entscheidungsbaums wurde zusätzlich ein Pruning-Verfahren durchgeführt, um die optimale Baumstruktur anhand des Komplexitätsparameters *ccp_alpha* zu bestimmen. Dafür wurden zunächst alle relevanten α -Parameter ermittelt und anschließend in einer Schleife zur Modellerstellung verwendet. Die Ermittlung der besten Parameter erfolgte für das gesamte Tuning stets durch den Vergleich der Modelle anhand des *f1-scores*.

5.4 Prediction und Inference

Ein essenzieller Aspekt bei der Modellwahl ist die Unterscheidung der Fragestellung in:

- **Prediction** Wenn das Ziel darin besteht, schnelle und ressourcenschonende Vorhersagen über zukünftige Werte oder Ereignisse auf noch unbekannten Daten anhand der bekannten Mustern vorherzusagen.
- **Inference** Wenn das Ziel darin besteht, Zusammenhänge zwischen den Merkmalen bestehender Daten herauszuarbeiten, ohne dass dabei zwangsläufig eine explizite Vorhersage zukünftiger Ereignisse erfolgt.

Modell	Prediction	Inference	Ergebnis der Vorhersage/Inference
Logistic Regression	Gut	Sehr gut	Berechnung von Wahrscheinlichkeiten für Fehlerklassen durch interpretierbare, lineare Modellierung
Random Forest	Sehr gut	Mittel	Robuste Fehlervorhersage durch Mehrheitsentscheidungen aus zahlreichen Bäumen; Inferenz weniger transparent
Support Vector Machine	Gut	Mittel	Klassifikation der Fehler durch optimale, aber schwer interpretierbare Entscheidungsgrenzen
K-Nearest Neighbors	Mittel	Schlecht	Einfache Fehlervorhersage basierend auf Nachbarschaft; keine direkte interpretierbare Modellstruktur
Decision Tree	Gut	Sehr gut	Einfache, interpretierbare hierarchische Entscheidungsregeln für Fehlerklassifikation
KNN (optimales K)	Gut	Mittel	Verbesserte Vorhersagegenauigkeit durch optimiertes K, jedoch weiterhin begrenzte Interpretierbarkeit
Geprunter Decision Tree	Gut	Sehr gut	Vereinfachte Entscheidungsregeln mit verbesserter Interpretierbarkeit und reduziertem Overfitting

Tab. 4: Vergleich der Modelle hinsichtlich Prediction und Inference

Diese Bewertung unterstützt die spätere Wahl des passenden Modells für den jeweiligen Anwendungsfall. Grundlegend sind für die Fragestellung dieser Arbeit jedoch Modelle zur *Prediction* zu bevorzugen, da zunächst im Rahmen der PdM der Fehler-Status einer Maschine für noch unbekannte Sensordaten ermittelt werden soll.

6 Evaluation

6.1 Modellvergleich auf Basis von Metriken

Der **F1-Score** ist eine geeignete Metrik für die Klassifikation von Maschinenfehlern in der Predictive Maintenance, da er sowohl **Precision** als auch **Recall** berücksichtigt. Dies ist entscheidend, da unerkannte Ausfälle (False Negatives) teure Schäden verursachen können, während Fehllarme (False Positives) die Produktion stören. Der F1-Score hilft, beide Risiken auszubalancieren. So sind die Metriken der Modelle auf den Trainingsdaten der Tabelle 5 und die Metriken der Modelle auf den Testdaten der Tabelle 6 zu entnehmen.

Modell	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.9171	0.9158	0.9171	0.9161
Random Forest	1.0000	1.0000	1.0000	1.0000
Decision Tree	1.0000	1.0000	1.0000	1.0000
Support Vector Machine	0.9737	0.9748	0.9737	0.9733
K-Nearest Neighbors	0.9893	0.9894	0.9893	0.9891
Random Forest (Hyperparameter-Tuning)	1.0000	1.0000	1.0000	1.0000
KNN mit optimalem K	1.0000	1.0000	1.0000	1.0000
Decision Tree mit Pruning	0.9985	0.9985	0.9985	0.9985

Tab. 5: Performance der Modelle auf den Trainingsdaten

Modell	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.7915	0.9734	0.7915	0.8664
Random Forest	0.9625	0.9733	0.9625	0.9676
Decision Tree	0.9505	0.9749	0.9505	0.9621
Support Vector Machine	0.8535	0.9717	0.8535	0.9052
K-Nearest Neighbors	0.9105	0.9638	0.9105	0.9346
Random Forest (Hyperparameter-Tuning)	0.9635	0.9734	0.9635	0.9681
KNN mit optimalem K	0.9385	0.9601	0.9385	0.9490
Decision Tree mit Pruning	0.9495	0.9752	0.9495	0.9618

Tab. 6: Performance der Modelle auf den Testdaten

6.2 Bias-Variance-Tradeoff

Die nachfolgende Tabelle 7 ergänzt nun die verwendeten Flexibilitätsparameter der Modelle. Diese Parameter haben direkten Einfluss auf die Generalisierungsfähigkeit der Modelle und müssen im Rahmen des Bias-Variance-Tradeoffs zur Evaluation berücksichtigt werden. Modelle wie logistische Regression zeigen einen hohen Bias und eine geringe Varianz, während die Random Forests und Entscheidungsbäume ohne Tiefenbeschränkung das Gegenteil aufwiesen. Gezieltes Hyperparameter-Tuning (z.B. Random Forest mit begrenzter Baumdiefe oder Decision Tree Pruning) führt zu einer verbesserten Balance zwischen Bias und Varianz. Dabei wurde das Overfitting verringert und die Generalisierung deutlich erhöht. Insbesondere das optimierte Random-Forest-Modell erreicht die beste Generalisierung und verdeutlichen die Relevanz sorgfältiger Parameterwahl mit Cross-Validation und Grid-Search. Dies wird auch in der Abbildung 10 verdeutlicht.

Modell	Parameter	Erhöhung	Senkung
Logistic Regression	{'C': 1.0}	Bias ↓, Varianz ↑	Bias ↑, Varianz ↓
Random Forest	{'n_estimators': 100, 'max_depth': None}	Bias ↓, Varianz ↓	Bias ↑, Varianz ↓
Decision Tree	{'max_depth': None}	Bias ↓, Varianz ↑	Bias ↑, Varianz ↓
Support Vector Machine	{'C': 1.0}	Bias ↓, Varianz ↑	Bias ↑, Varianz ↓
K-Nearest Neighbors	{'n_neighbors': 5}	Bias ↑, Varianz ↓	Bias ↓, Varianz ↑
RF (Hyperparameter-Tuning)	{'n_estimators': 200, 'max_depth': 20}	Bias ↓, Varianz ↓	Bias ↓, Varianz ↑
KNN mit optimalem K	{'n_neighbors': 10}	Bias ↑, Varianz ↓	Bias ↓, Varianz ↑
Decision Tree mit Pruning	{'ccp_alpha': 3.2e-05}	Bias ↑, Varianz ↓	Bias ↓, Varianz ↑

Tab. 7: Modellparameter zur Bewertung der Flexibilität

6.3 Wahl des besten Modells

Basierend auf dem höchsten **F1-Score** auf den Testdaten wurde das **Random Forest Modell mit Hyperparameter-Tuning** als das beste Modell ausgewählt. Obwohl der **F1-Score** auf den Trainingsdaten 1.0 beträgt, konnte keine Overfitting-Tendenz des Modells festgestellt werden.

Zusätzlich bietet der **Random Forest** auch umfangreiche Möglichkeiten auf die Flexibilität des Modells, durch beispielsweise die Anzahl der Bäume (`n_estimators`) und die Tiefe der Bäume (`max_depth`) einzuwirken.

Zwar besitzt der **Random Forest** eine eher schwer zu interpretierende Modellstruktur und ist deswegen nur bedingt zur Inference geeignet; jedoch kann der **Random Forest** optimal zur Prediction verwendet werden. Das ist insbesondere in der PdM entscheidend, da Überwachungssysteme für Maschinen vor allem auf präzisen Prognosen beruhen.

6.3.1 Scatterplot-Vergleich der Trainings- und Testmetriken

Zur besseren Beurteilung der Generalisierungsfähigkeit der Modelle wurde ein Vergleich der F1-Scores auf Trainings- und Test-Daten mittels eines **Scatterplots** erstellt. Diese Visualisierung zeigt, inwiefern die Leistung eines Modells auf den Trainingsdaten mit der tatsächlichen Performance auf den Testdaten übereinstimmt.

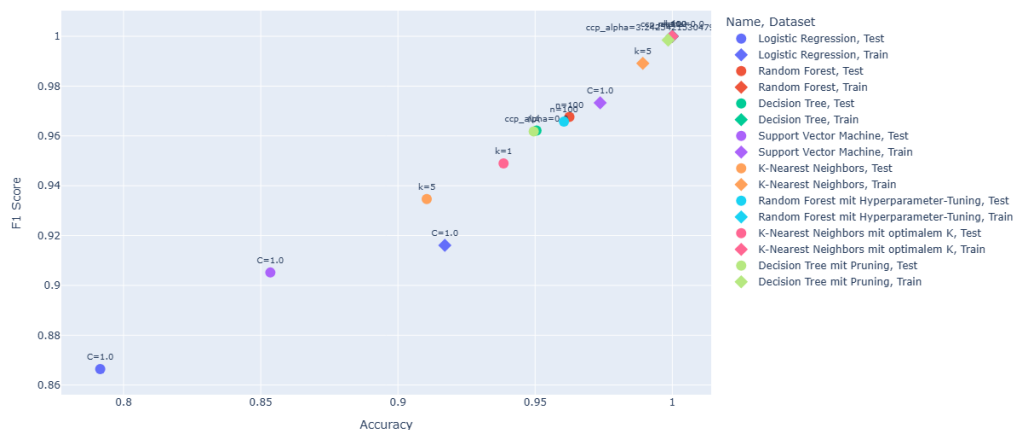


Abb. 10: Scatterplot der F1-Scores für alle Modelle auf Trainings- und Testdaten. Ein hoher Unterschied zwischen beiden Werten kann auf Overfitting hindeuten.

In Abbildung 10 sind die F1-Scores der einzelnen Modelle auf den Trainings- und Testdaten als Punkte dargestellt. Ein Modell mit einer hohen Generalisierung sollte nahe der **diagonalen Referenzlinie** liegen (wo Trainings- und Testwerte gleich wären). Der **Random Forest mit Hyperparameter-Tuning** weist zwar einen hohen Wert auf den Trainingsdaten auf, zeigt aber auch auf den Testdaten eine exzellente Performance. Daher kann ausgeschlossen werden, dass das Modell sich übermäßig an die Trainingsdaten angepasst hat.

Erkenntnisse aus der Visualisierung:

- **Random Forest mit Hyperparameter-Tuning** zeigt eine hohe Konsistenz zwischen Trainings- und Testdaten.
- **Modelle mit großer Differenz (z. B. Decision Tree ohne Pruning)** neigen zu Overfitting.
- **Logistic Regression** zeigt einen niedrigen F1-Scores, was darauf hindeutet, dass diese komplexe Zusammenhänge nur schlecht erfassen kann.

6.4 Einfluss des Oversamplings und Diskussion zur Modellperformance

Da die Modellperformance für einige Modelle überraschend hoch ist, musste überprüft werden, ob das **synthetische Oversampling** einen ungewollten Einfluss auf die Generalisierungsfähigkeit der Modelle ausgeübt haben könnte. Dafür wurden analog zu den originalen Daten auch Diagramme über die Verteilung der oversampelten Daten ertellt. Dadurch ist es einerseits möglich, die konkreten oversampelten Daten zu analysieren, als auch die Verteilung dieser mittels der **Boxplots** zu vergleichen.

Oversampelte Trainingsdaten:

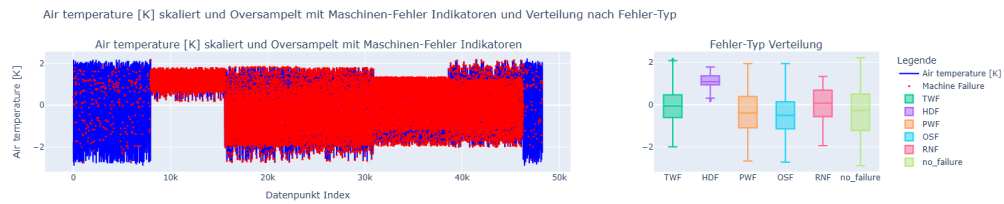


Abb. 11: Boxplot der Oversampelten Trainingsdaten für Lufttemperatur in Bezug auf die Fehlertypen

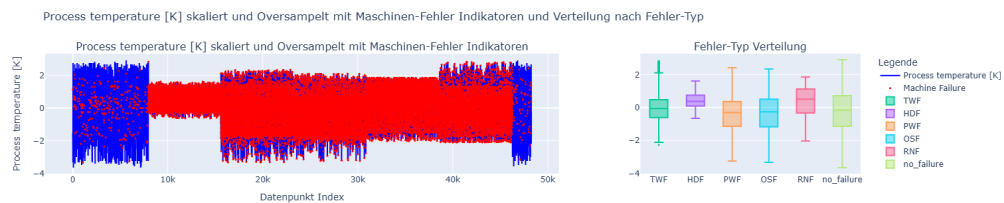


Abb. 12: Boxplot der Oversampelten Trainingsdaten für Prozesstemperatur in Bezug auf die Fehlertypen

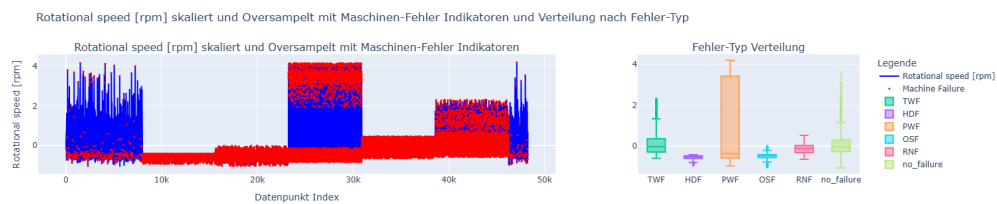


Abb. 13: Boxplot der Oversampelten Trainingsdaten für Rotationsgeschwindigkeit in Bezug auf die Fehlertypen

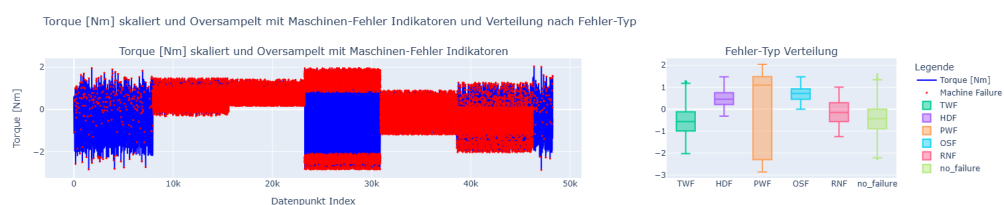


Abb. 14: Boxplot der Oversampelten Trainingsdaten für Drehmoment in Bezug auf die Fehlertypen

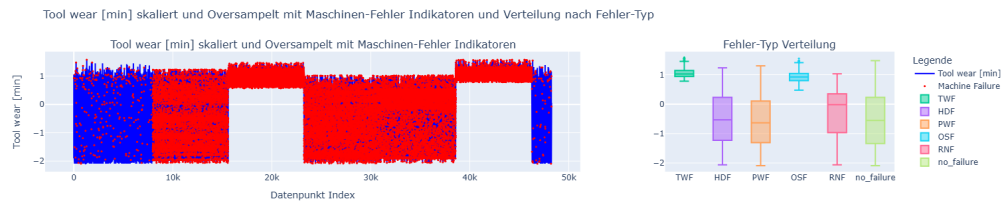


Abb. 15: Boxplot der Oversampelten Trainingsdaten für Werkzeugverschleiß in Bezug auf die Fehlertypen

Die Boxplots zeigen, dass die Verteilungen der oversampelten Trainingsdaten **nahezu identisch** zu den Originaldaten sind (vgl. Abbildung 4, Abbildung 5, Abbildung 6, Abbildung 7, Abbildung 8). Dies deutet darauf hin, dass durch das Oversampling keine signifikante Verzerrung der Daten entstanden ist.

Die Evaluation der Modelle erfolgte zudem ausschließlich auf den originalen (+ standardisierten) Testdaten. Das bedeutet, dass die hohe Modellperformance nicht durch künstliche Datenpunkte beeinflusst wurde, sondern tatsächlich eine gute Generalisierungsfähigkeit widerspiegelt. Dies zeigt, dass das Oversampling eine sinnvolle Maßnahme war, um das Training zu verbessern, ohne die finale Evaluierung zu verfälschen.

6.5 Fazit der Evaluation

- Random Forest mit Hyperparameter-Tuning wurde als bestes Modell gewählt.
- Die Modellperformance wird nicht durch das Oversampling beeinträchtigt.

7 Fazit

In dieser Arbeit wurde untersucht, wie sich Fehlerursachen für Maschinenausfälle mithilfe von Sensordaten durch Klassifikationsmodelle vorhersagen lassen. Hierfür wurden verschiedene Machine-Learning-Modelle trainiert, evaluiert und verglichen. Die besten Ergebnisse wurden mit dem **Random-Forest-Modell mit Hyperparameter-Tuning** erzielt, welcher eine **hohe Vorhersagegenauigkeit** und gleichzeitig eine **stabile Generalisierung** zeigte.

Die wichtigsten Erkenntnisse lassen sich wie folgt zusammenfassen:

- **Datenanalyse und Aufbereitung:** Der verwendete Datensatz war ein **synthetisch generierter Datensatz**, welcher reale Bedingungen aus der Industrie nachbilden sollte. Durch explorative Datenanalyse (EDA) konnten wesentliche Einsichten in die Sensordaten und Fehlerarten gewonnen werden. Zur Verbesserung der Vorhersagequalität wurden eine Standardisierung der metrischen Variablen sowie ein Oversampling der Minderheitsklassen durchgeführt.
- **Modellauswahl und Training:** Es wurden sieben verschiedene Machine-Learning-Modelle trainiert: *Logistic Regression*, *Decision Trees*, *Random Forest*, *Support Vector Machine* sowie *K-Nearest Neighbors*. Die Modelle wurden mittels Grid Search und 5-facher Kreuzvalidierung optimiert, um die optimalen Hyperparameter zu identifizieren.
- **Evaluation und Generalisierung:** Die Evaluierung erfolgte anhand der Metriken F1-Score mit Precision, Recall und Accuracy. Ein Scatterplot-Vergleich zwischen Trainings- und Testmetriken zeigte, dass das Random-Forest-Modell keine Tendenzen zu Overfitting aufwies und eine hohe Generalisierungsfähigkeit besitzt. Die Testmetriken wurden ausschließlich auf den originalen Testdaten berechnet, wodurch sichergestellt wurde, dass die beobachtete Modellperformance nicht auf eine synthetische Verzerrungen zurückzuführen ist.
- **Einfluss des Oversamplings:** Um das Problem der Klassenunausgeglichenheit zu adressieren, wurde SMOTETomek eingesetzt. Ein Vergleich mittels Boxplots bestätigte, dass das Oversampling keinen Einfluss auf die Modellleistung hatte.

- **Einschränkungen und zukünftige Arbeiten:** Zuletzt bleibt offen, ob sich die Modelle in einem realen Produktionsumfeld ebenso leistungsfähig zeigen würden. Eine zusätzliche Evaluierung mit echten industriellen Sensordaten wäre notwendig, um die Generalisierbarkeit weiter zu überprüfen. Zudem wäre es sehr wahrscheinlich erforderlich, ein umfassenderes Data-Cleaning für reale Daten durchzuführen, da Messfehler oder andere Probleme die Datenqualität von realen Maschinendaten beeinträchtigen könnten. Des Weiteren könnten sich zukünftige Arbeiten mit der Kombination individueller Modelle beschäftigen, um eine noch genauere Klassifikation zu erreichen.

Zusammenfassend zeigt diese Arbeit, dass Machine Learning ein effektives Werkzeug zur Früherkennung von Maschinenausfällen darstellt. Durch eine gezielte Kombination aus Datenverständnis, Datenaufbereitung und Modelloptimierung konnten leistungsfähige Klassifikationsmodelle entwickelt werden. Die durchgeführten Tests auf den Original-Testdaten zeigen eine hohe Generalisierungsfähigkeit der Modelle. Zukünftige Forschungen sollten sich mit der Integration der Modelle in eine produktive Plattform sowie mit der Kombination individueller Modelle zur Verbesserung der Klassifikation beschäftigen.

Anhang

- <https://github.com/DataScience-DHBWKA/PredictiveMaintenance>

Hilfsmittelverzeichnis

Name des Hilfsmittels	Verwendung
ChatGPT	Unterstützung bei Textkorrektur, sprachlicher Überarbeitung, Strukturierung der Arbeit sowie bei der Erstellung und Fehlerbehebung von Python-Code.
LeChat	Nutzung zur gezielten Recherche nach ergänzender Fachliteratur und methodischen Grundlagen.
DeepL	Übersetzung englischsprachiger Fachtexte zur inhaltlichen Erschließung wissenschaftlicher Quellen.

Literatur

- Alenizi, F. A., Abbasi, S., Adil Hussein Mohammed & Amir Masoud Rahmani. (2023). The artificial intelligence technologies in Industry 4.0: A taxonomy, approaches, and future directions. *Computers Industrial Engineering*, 185, 109662. <https://doi.org/10.1016/j.cie.2023.109662>
- B. Jahn und M. Pfeiffer. (2014). Die digitale Revolution - Neue Geschäftsmodelle statt (nur) neue Kommunikation. *Marketing Review St. Gallen*, 31. Jg., 79–93. <https://doi.org/10.1365/s11621-014-0323-5>
- Daniel Frank Hesser und Bernd Markert. (2019). Tool wear monitoring of a retrofitted CNC milling machine using artificial neural networks. *Manufacturing Letters*, 1–4. <https://doi.org/10.1016/j.mfglet.2018.11.001>
- Giuseppe Ferrera. (2021, November). *Überblick über maschinelle Lernverfahren für Predictive Maintenance*. <https://d-nb.info/1251237134/34#page=60>
- Gottmann, J. (2019). *Produktionscontrolling: Wertströme und Kosten optimieren*. Springer, Berlin.
- Hoffmann, M.A., Lasch, R. (2025). Unlocking the Potential of Predictive Maintenance for Intelligent Manufacturing: a Case Study On Potentials, Barriers, and Critical Success Factors. *Schmalenbach J Bus Res*. <https://doi.org/https://doi.org/10.1007/s41471-024-00204-3>
- Jain A. K. and Chouksey P. and Parlikad A. K. and Lad B. K. (2020). Distributed Diagnostics, Prognostics and Maintenance Planning: Realizing Industry 4.0. *FAC-PapersOnLine*, 354–359. <https://doi.org/10.1016/j.ifacol.2020.11.057>
- Koca O. and Kaymakci O. T. and Mercimek M. (2020). Advanced Predictive Maintenance with Machine Learning Failure Estimation in Industrial Packaging Robots. *International Conference on Development and Application Systems (DAS). 15th edition, May 21-23, 2020, Suceava, Romania : conference proceedings*, 1–6. <https://doi.org/10.1109/DAS49615.2020.9108913>
- Markus Seebacher. (2018, Mai). *Einsatz von Predictive Maintenance im Bereich von Incident-Management*. <https://epub.technikum-wien.at/obvftwhsmmig/content/titleinfo/9752429/full.pdf>

- S. Zhai und G. Reinhart. (2018). Predictive Maintenance als Wegbereiter für die instandhaltungsgerechte Produktionssteuerung. *Zeitschrift für wirtschaftlichen Fabrikbetrieb (ZWF)*, Jg. 113, Heft 5, 298–301. <https://doi.org/10.3139/104.111912>
- Wayne W. Eckerson. (2007). PREDICTIVE ANALYTICS Extending the Value of Your Data Warehousing Investment. *TDWI BEST PRACTICES REPORT*. http://download.101com.com/pub/tdwi/Files/PA_Report_Q107_F.pdf
- Wolfgang Becker und Felix Schuhknecht. (2020). Instandhaltungscontrolling in der digitalen Welt. *Wissen*, 57–62. https://web.archive.org/web/20220225155502id_/https://www.beck-elibrary.de/10.15358/0935-0381-2020-5-57.pdf