



Predictive Maintenance

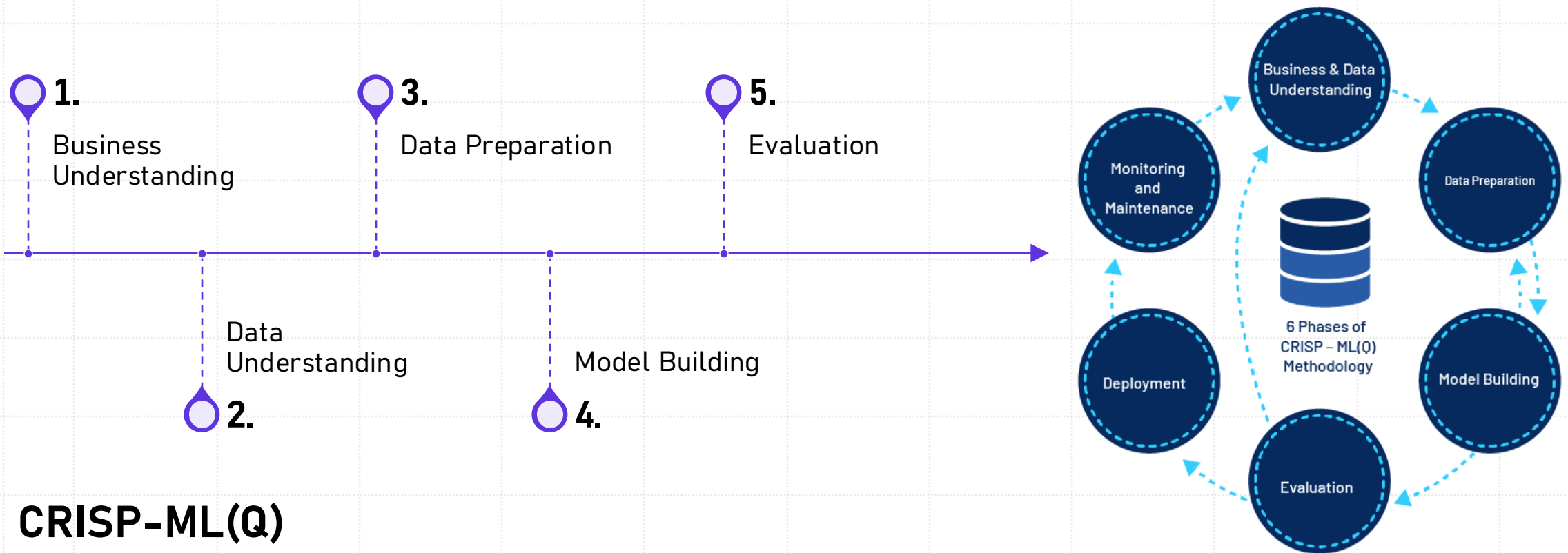
Daniel Weissenberger

Eduardo Stein-Mössner

Jonas Sigmund

DHBW | Gruppe 8 | 25.03.2025

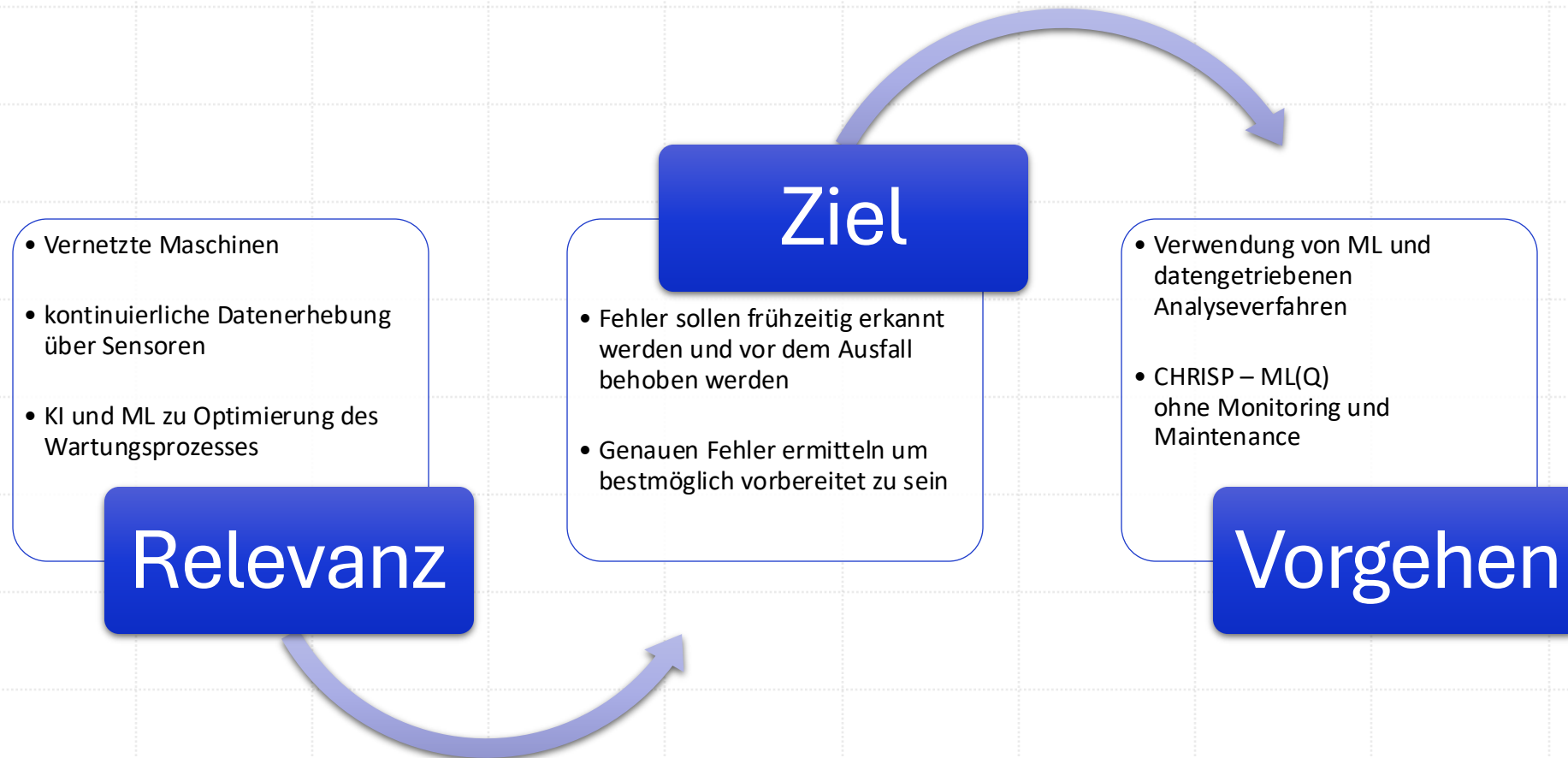
Gliederung & Methode



1. Business Understanding



1. Predictive Maintenance





1. Fragestellung

Wie genau lassen sich die Fehlerursachen einer Maschine durch Sensordaten mithilfe von Klassifikationsmodellen identifizieren (gemessen anhand von F1-Score mit Precision und Recall)?

- **S:** Mehrere Machine-Learning-Modelle sollen entwickelt werden, das verschiedene Fehlerursachen basierend auf Sensordaten klassifizieren.
- **M:** Die Modellperformance wird anhand von F1-Score, Precision und Recall gemessen.
- **A:** Durch Explorative Datenanalyse und ausführliches Modeling machbar.
- **R:** Durchführbar innerhalb des geplanten Zeitrahmens auf der Grundlage eines geeigneten Datensatzes und der bisher erworbenen Kompetenzen.
- **T:** Umsetzung innerhalb von 6 Wochen mit Evaluierung in Woche 7.

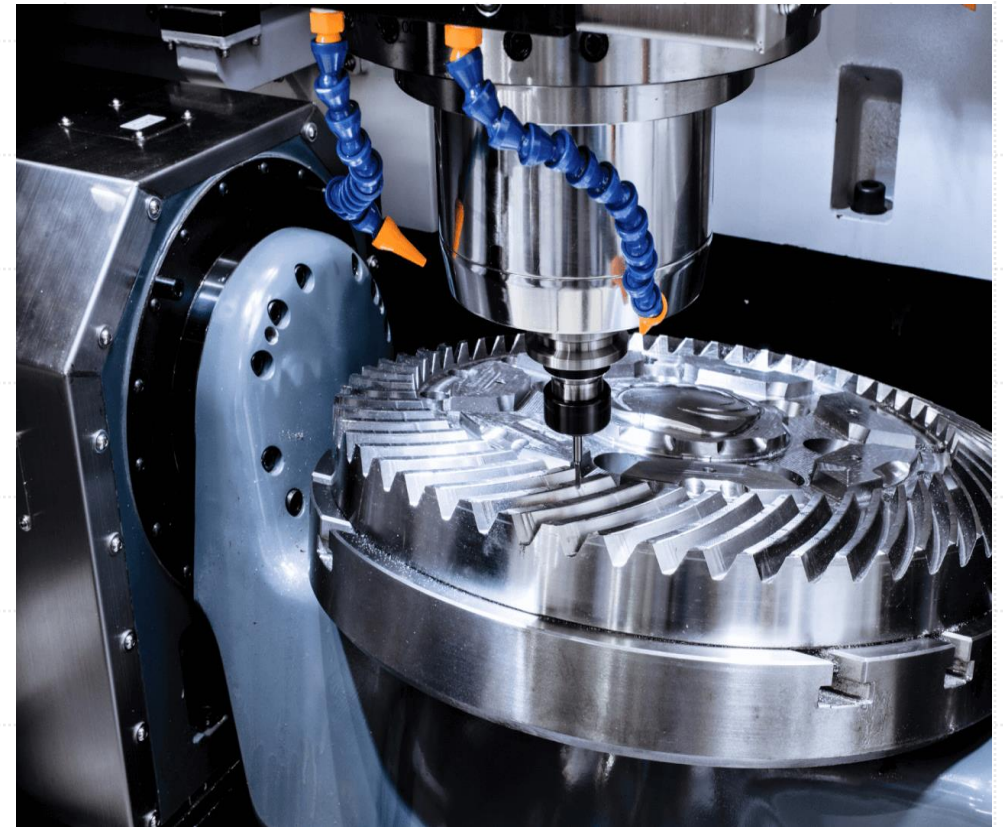
2. Data Understanding



2. Datensatz

- Synthetischer Datensatz (Random Walk Process)
- Sensordaten einer Fräsmaschine
- Predictive Maintenance
- Bildet Fehler und Fehlerursache ab
- 10.000 Zeilen x 14 Spalten

Input	Output
<ul style="list-style-type: none">▪ Product-Id▪ Type▪ Air temperature▪ Process temperature▪ Torque▪ Tool wear	<ul style="list-style-type: none">▪ tool wear failure▪ heat dissipation failure▪ power failure▪ overstrain failure▪ random failures



2. Datenverteilung

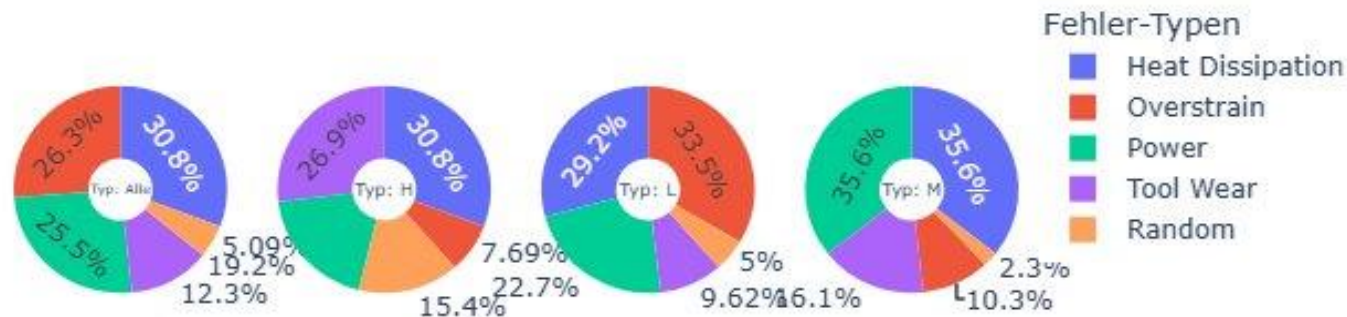


Nur **3,39%** der Maschinen sind überhaupt defekt

1. Class-Imbalance

2. Datenverteilung

Fehler-Typen nach Maschinen-Typ



Overall → Heat Dissipation

Type: H → Tool Wear

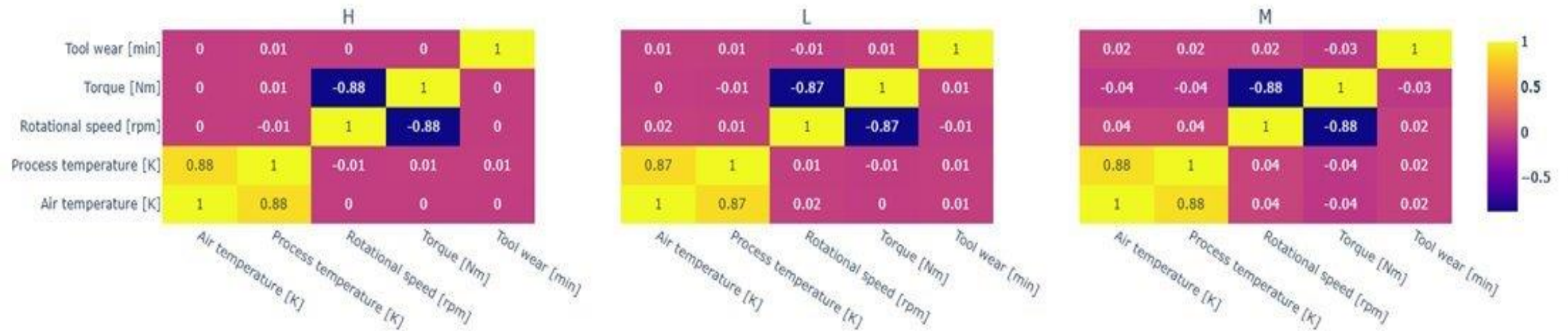
Type: M → Power

Type: L → Overstrain

2. Class-Imbalance

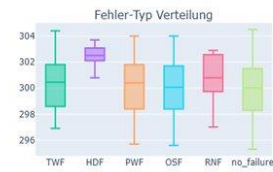
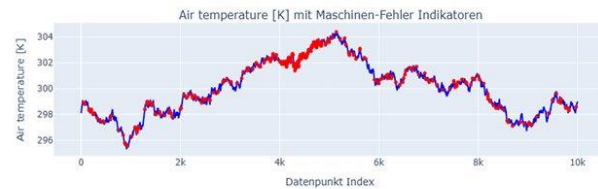
2. Datenverteilung

Korrelations-Matrix nach Maschinen-Typ

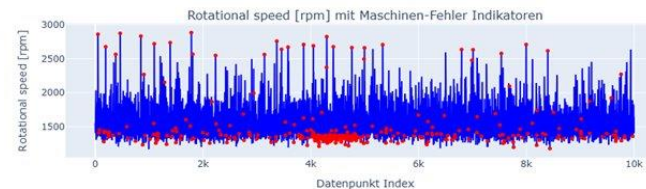


2. Datenverteilung

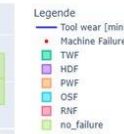
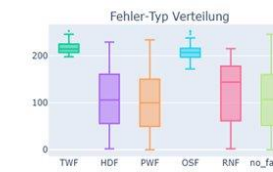
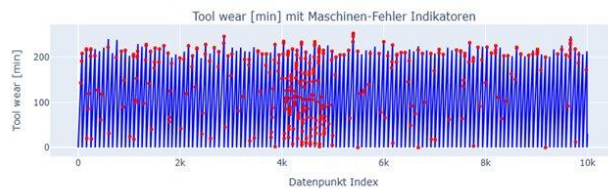
Air temperature [K] mit Maschinen-Fehler Indikatoren und Verteilung nach Fehler-Typ



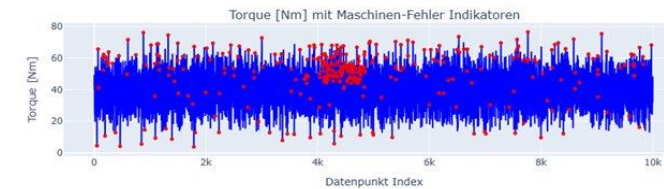
Rotational speed [rpm] mit Maschinen-Fehler Indikatoren und Verteilung nach Fehler-Typ



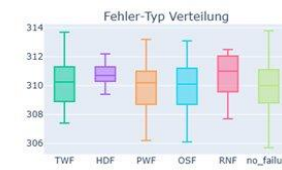
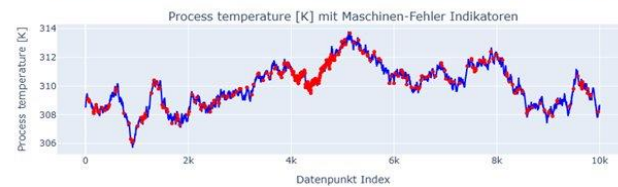
Tool wear [min] mit Maschinen-Fehler Indikatoren und Verteilung nach Fehler-Typ



Torque [Nm] mit Maschinen-Fehler Indikatoren und Verteilung nach Fehler-Typ



Process temperature [K] mit Maschinen-Fehler Indikatoren und Verteilung nach Fehler-Typ



2. Erkenntnisse

Fehlerursache

Bestimmte Fehler treten verstärkt bei spezifischen Sensordatenwerten auf

Korrelationen

Negative Korrelation zwischen Drehmoment und Drehzahl (-0.88)

Positive Korrelation zwischen Lufttemperatur und Prozesstemperatur (0.88)

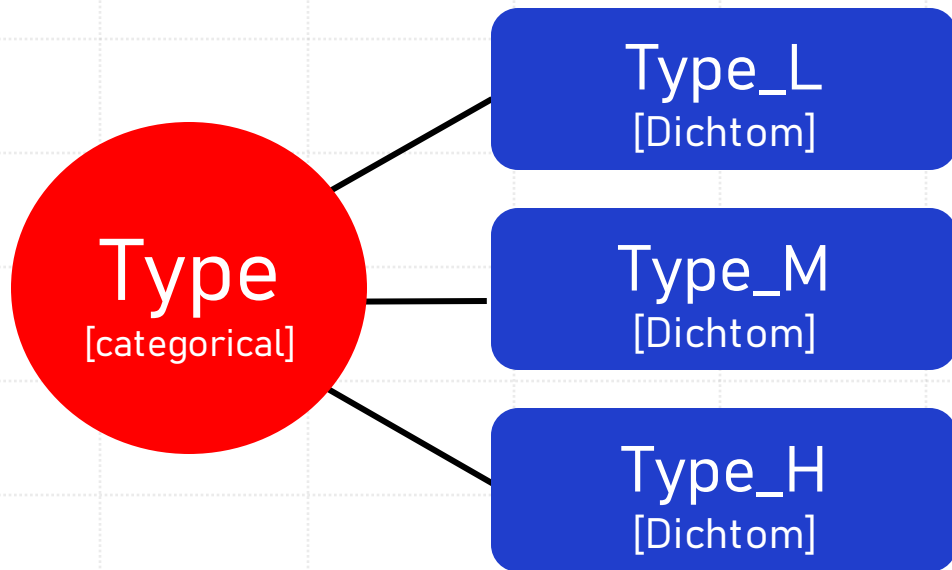
Klassenverteilung

Unausgeglichene Klassenverteilung
→ Sampling-Strategien notwendig

3. Data Preperation

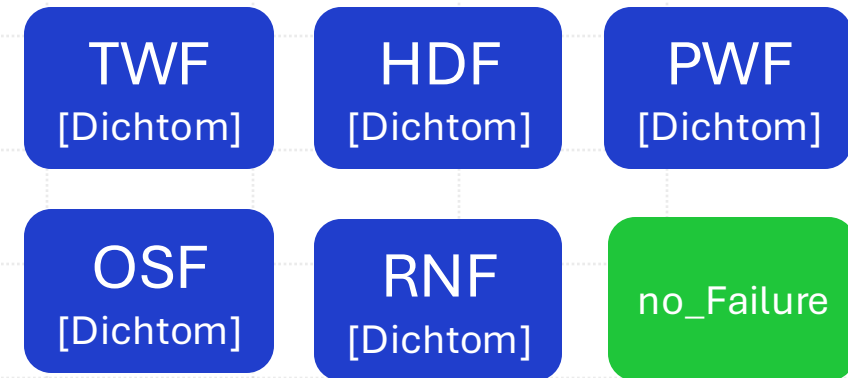


3. Kategorische Merkmale



One-Hot-Encoding

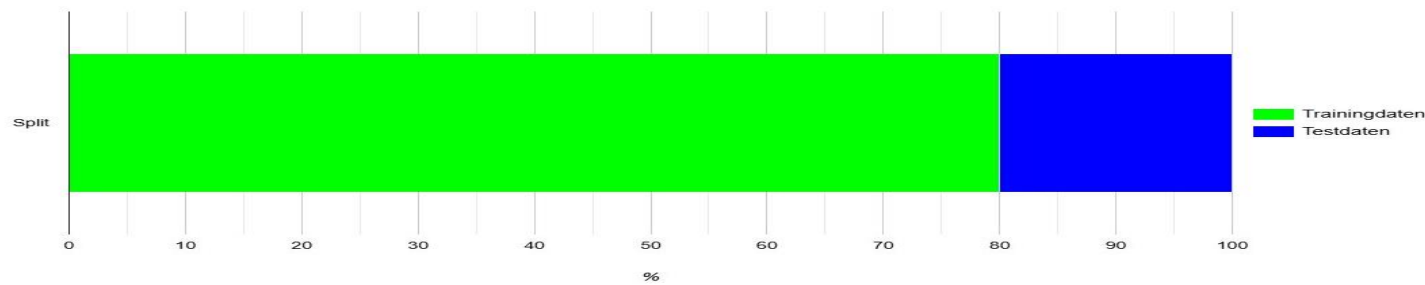
Dummy-Variablen für den Maschinen-Typ einführen (3 für Übersicht und Interpretierbarkeit)



Label erstellen

Kategorisches Label aus den Dummy-Variablen der Fehlertypen ableiten. (5 + 1)

3. Datenaufteilung

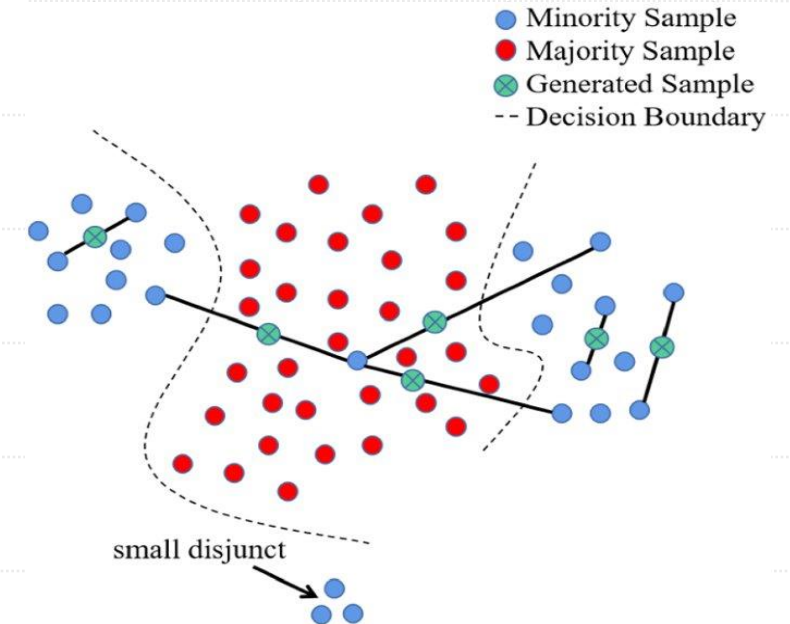


Traning-Test-Split

80:20 → möglichst viele Daten fürs Training verwenden.

Oversampling

- Nur Trainingsdaten ■ werden oversampled
- SMOTETomek von imbalanced-learn (imblearn)
- SMOTE: Synthetic Minority Over-sampling Technique
- Tomek-Links: reduziert Datenpunkte der Mehrheitsklasse
- 46.000 statt 8.000 Datenpunkte mit ausgeglichenen Klassen



3. Standardisierung

Notwendig um die Diskriminierung von Merkmalen mit großem Skalenbereich zu vermeiden.

Standard-Skalierung, um die relative Verteilung beizubehalten, die Interpretierbarkeit zu verbessern und dennoch nicht so anfällig gegenüber Ausreißern zu sein.

Gleiche Parameter für Trainings- und Testdaten

Zuletzt werden diese Trainings- und Testdaten in zwei separaten CSV-Dateien gespeichert. (fixed Random State)

$$z = \frac{x - \mu}{\sigma}$$

μ = Mean

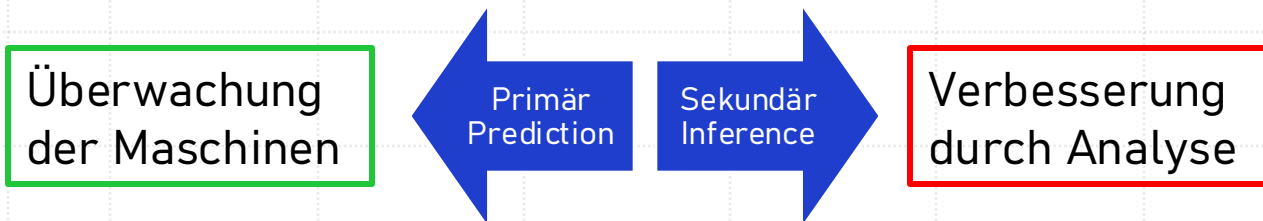
σ = Standard Deviation

4. Modelling



4. Zielsetzung

Mehrklassige Klassifikation über die Zielvariable "label" mit **5 + 1 Klassen**. (→ Supervized Learning)



Modell	Prediction	Inference
Logistic Regression	Gut	Sehr gut
Random Forest	Sehr gut	Mittel
Support Vector Machine	Gut	Mittel
K-Nearest Neighbors	Mittel	Schlecht
Decision Tree	Gut	Sehr gut
KNN (optimales K)	Gut	Mittel
Geprunter Decision Tree	Gut	Sehr gut

Vergleichsmetrik: **F1-Score**
$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Da sowohl **Precision** als auch **Recall** berücksichtigt werden.

Denn unerkannte Ausfälle (**FN**) verursachen hohe Reparaturkosten.
Und Fehllarme (**FP**) bringen die Produktion ins Stocken.

4. Hyperparametertuning

Random Forest

- n_estimators
- depth
- min_samples_split
- min_samples_leaf

K-Nearest-Neighbors

- k Nachbarn

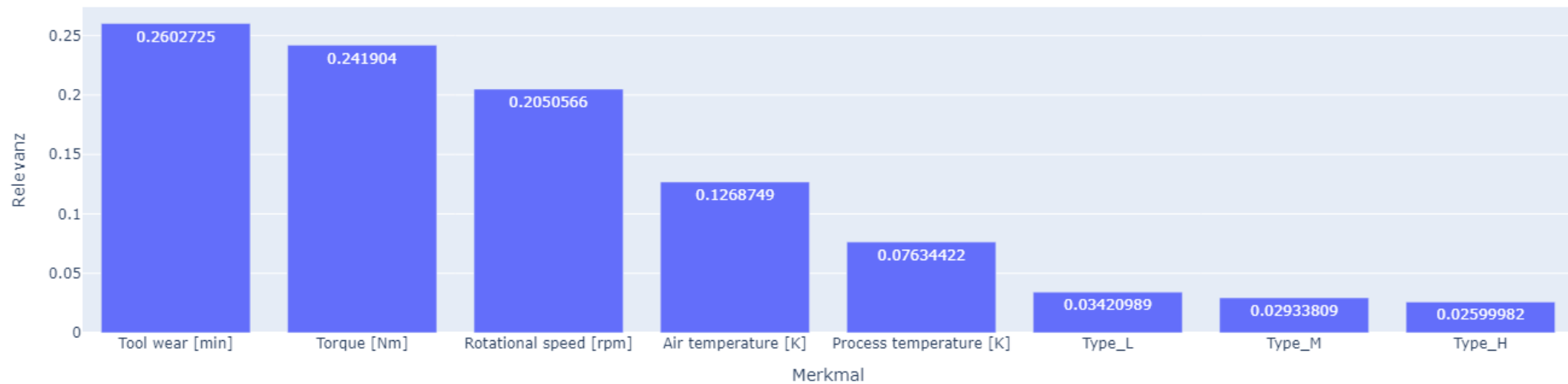
Descicion Tree

- ccp_alpha

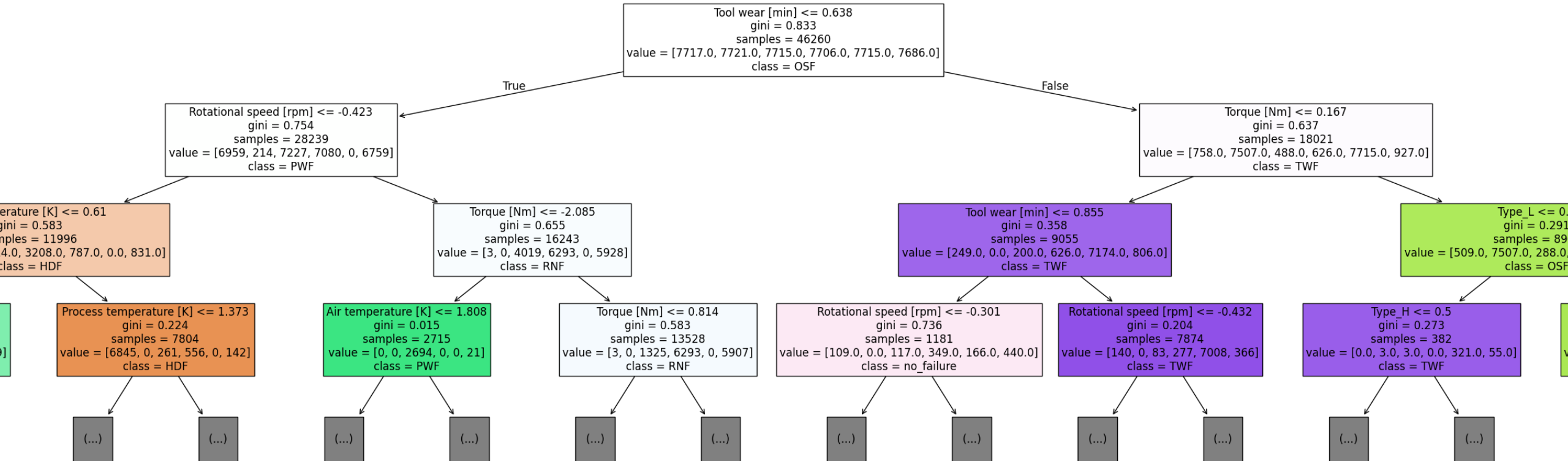
Für den RF und KNN Algorithmus wurde auf eine Grid Search und 5-fach Kreuzvalidierung zurückgegriffen.

4. Analyse

Feature-Import des RandomForest



4. Analyse



5. Evaluation



5. Modellperformance

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



5. Modellperformance

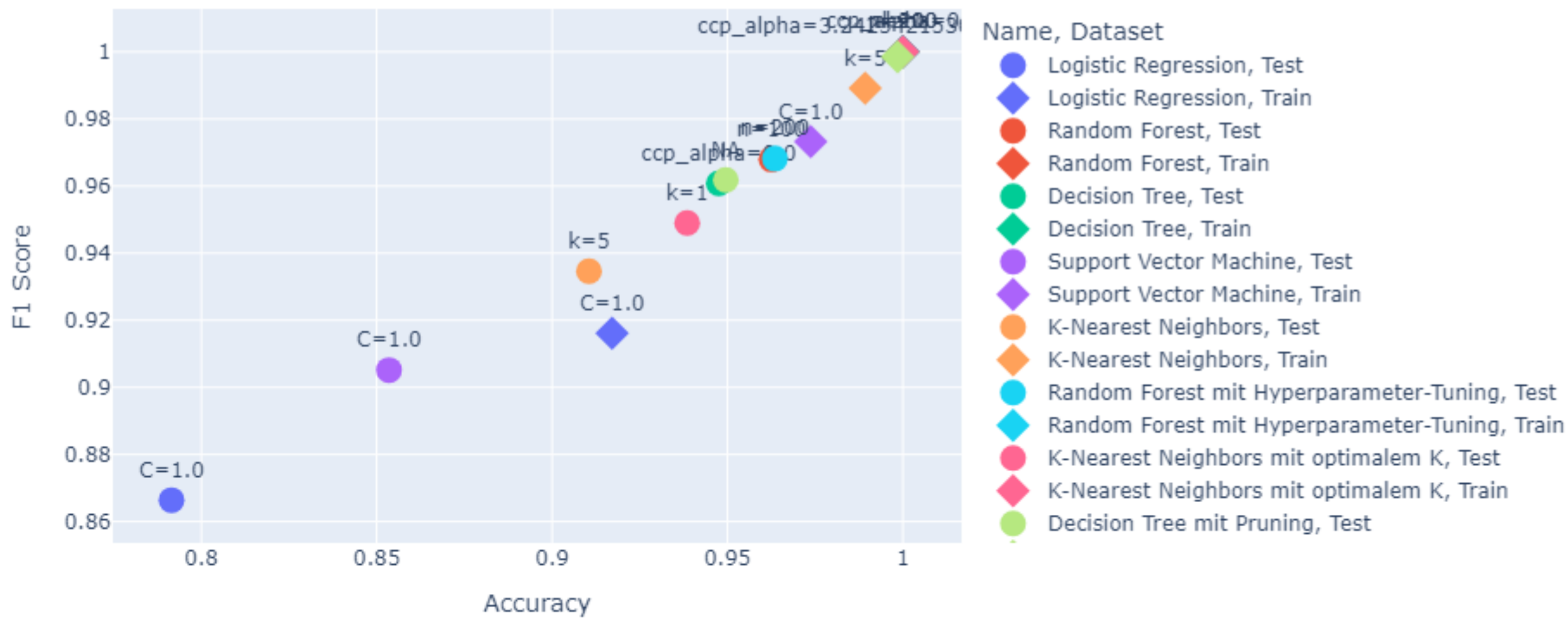
- Evaluierung anhand von Accuracy, Precision, Recall und F1-Score
- Modelle mit starker Abweichung zwischen Trainings- und Test-Metriken neigen zu Overfitting
- Random Forest mit Hyperparameter-Tuning zeigte die beste Performance
- Random Forest mit Hyperparameter-Tuning generalisiert gut

5. Modellflexibilität

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


Tab. 7: Modellparameter zur Bewertung der Flexibilität

- Modelle mit mehr Parametern sind flexibler, können aber Overfitting verursachen
- Entscheidungsbaum-basierte Modelle lassen sich gut regulieren (Pruning)
- Random Forest mit Hyperparameter-Tuning hat eine hohe Flexibilität durch viele Bäume (n_estimators, max_depth)

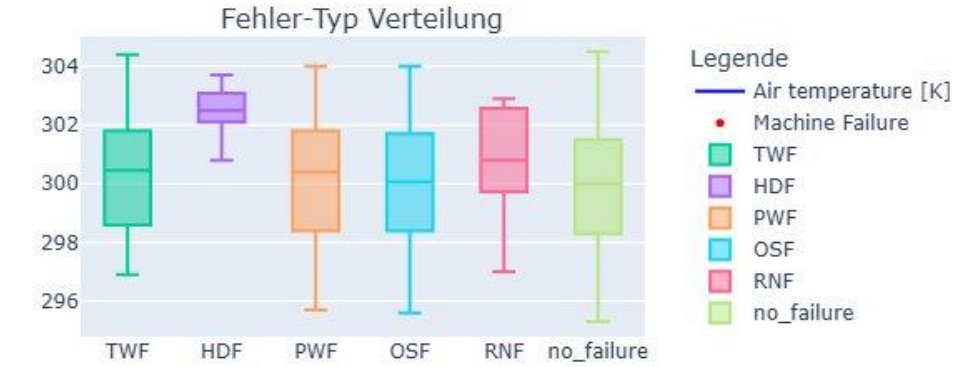
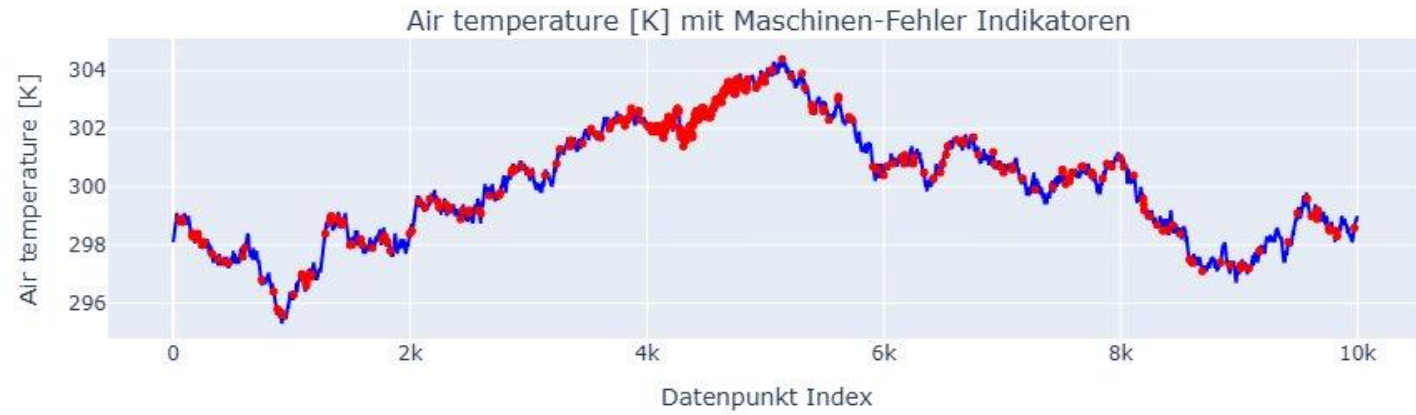




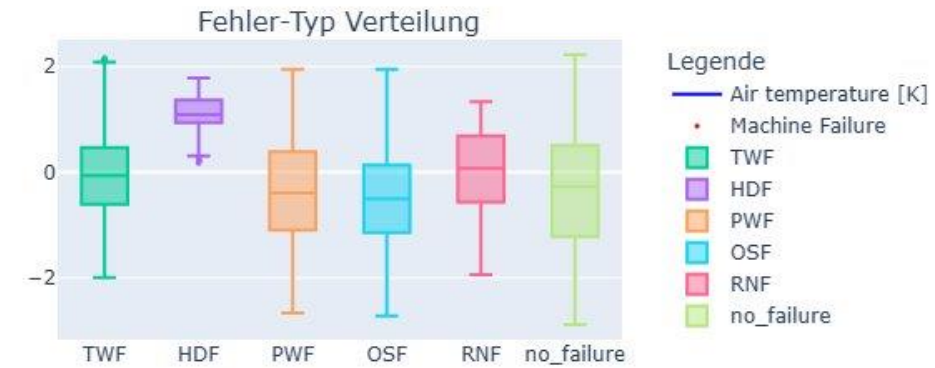
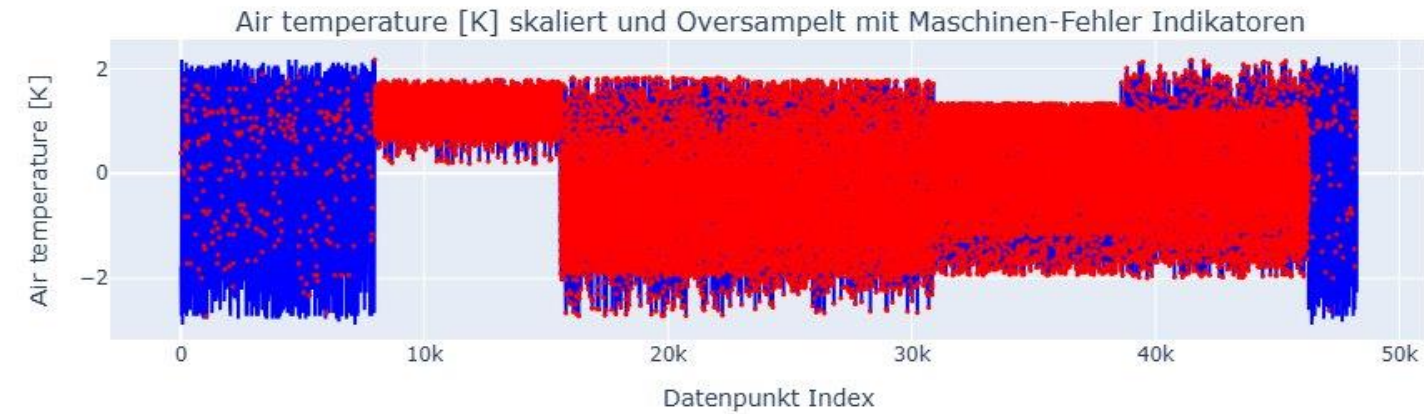
5. Einfluss des Oversamplings

- 
- Vergleich der Boxplots von Original- und Oversampelten Daten
 - Keine signifikante Veränderung in der Verteilung der Merkmale
 - Oversampling hat keinen negativen Einfluss auf die Modellleistung

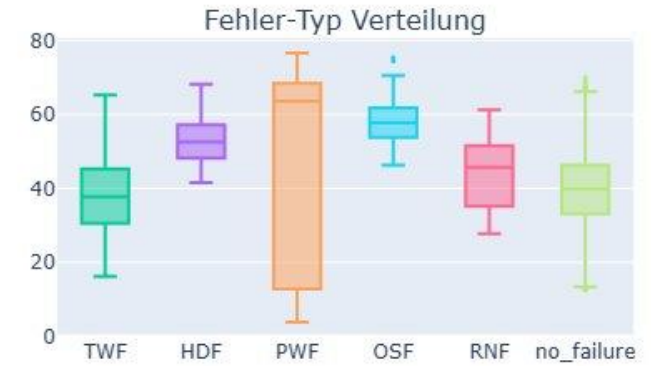
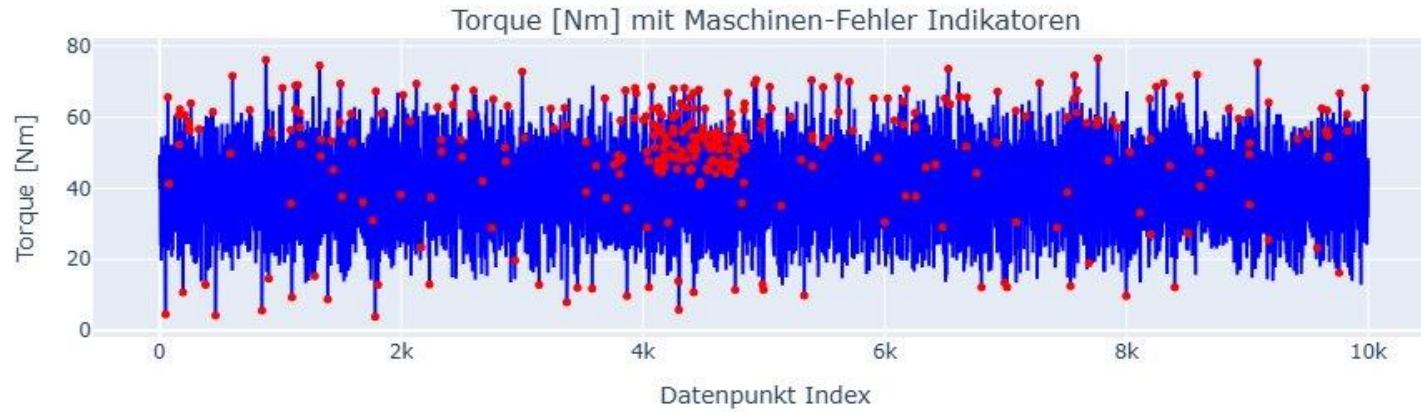
Air temperature [K] mit Maschinen-Fehler Indikatoren und Verteilung nach Fehler-Typ



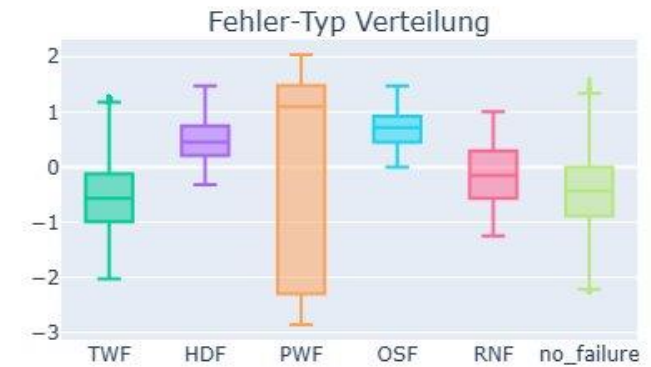
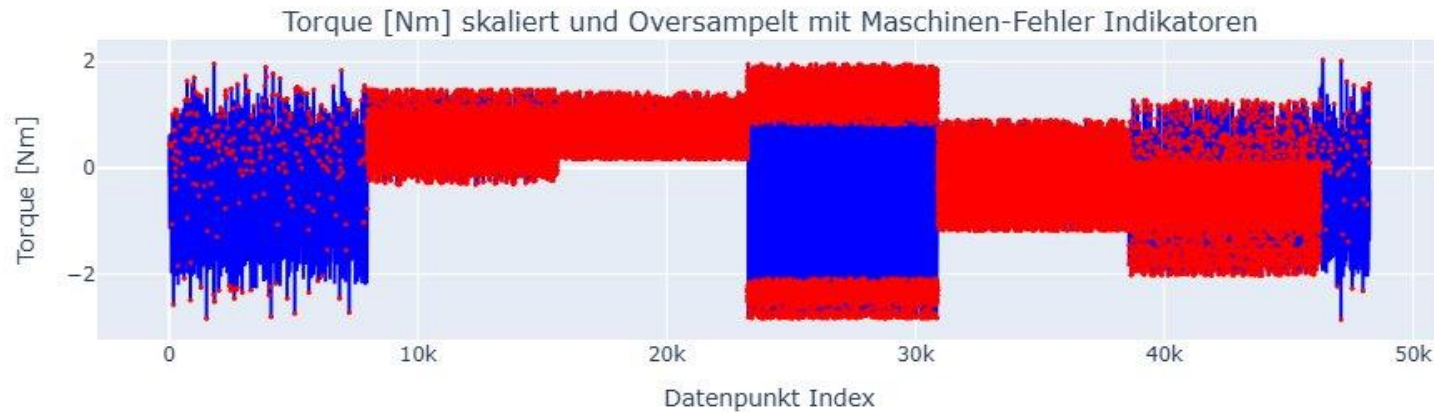
Air temperature [K] skaliert und Oversampelt mit Maschinen-Fehler Indikatoren und Verteilung nach Fehler-Typ



Torque [Nm] mit Maschinen-Fehler Indikatoren und Verteilung nach Fehler-Typ



Torque [Nm] skaliert und Oversampelt mit Maschinen-Fehler Indikatoren und Verteilung nach Fehler-Typ



6. Fazit



6. Modellwahl

- Random Forest mit Hyperparameter-Tuning wurde als bestes Modell gewählt
- Trotz F1-Score von 1.0 auf Trainingsdaten zeigt es keine Overfitting-Tendenzen
- Gute Generalisierung auf Testdaten mit F1-Score von 0.9681

6. Ausblick

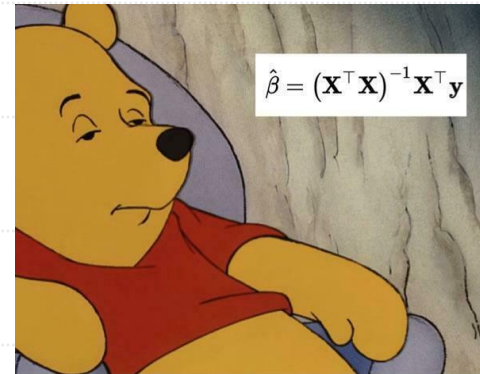
Der Datensatz ist synthetisch generiert → reale Produktionsdaten könnten abweichen

Integration in ein produktives Überwachungssystem auf Echtzeit Basis

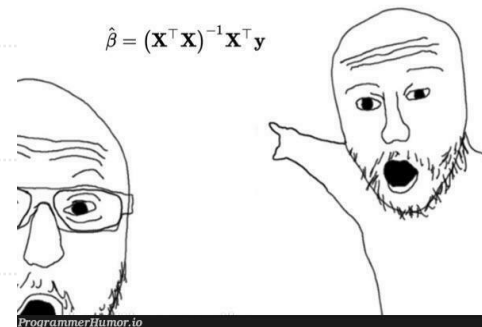
Kombination verschiedener Modellen für individuelle Fehler

Vielen Dank für Ihre
Aufmerksamkeit 🙄

Gibt es noch Fragen ?



Employers
when you tell
them your app
uses linear
regression



Employers
when you tell
them your app
uses “machine
learning and
A.I.”



Predictive Maintenance

Daniel Weissenberger

Eduardo Stein-Mössner

Jonas Sigmund

DHBW | Gruppe 8 | 25.03.2025