

# Modell zur Vorhersage von Maschinen- ausfällen

## Ausgangssituation

- Produktionsmaschinen eines Unternehmens
- Wartung in regelmäßigen Zeitabständen oder bei Maschinenausfall
- Nicht optimale Vorgehensweise
- Reaktive Wartungen 3x teurer als präventive Wartungen
- Unnötiger Verbrauch von Personal, Kapital und Ressourcen
- Sammlung von Daten einer Maschine zur Reduzierung der Maschinenausfälle

## Forschungsfrage/ Hypothese

- Vorhersage von Maschinenausfällen mittels maschinellen Lernens?

Hypothese: Gradient Boosting mittels XGBoost ermöglicht eine bessere Vorhersage von außerplanmäßigen Maschinenausfällen als zufälliges Schätzen.

# Gliederung



Datensatz & Balancing der Klassen



Erläuterung Gradient Boosting



Modellevaluation



Nutzen für das Unternehmen



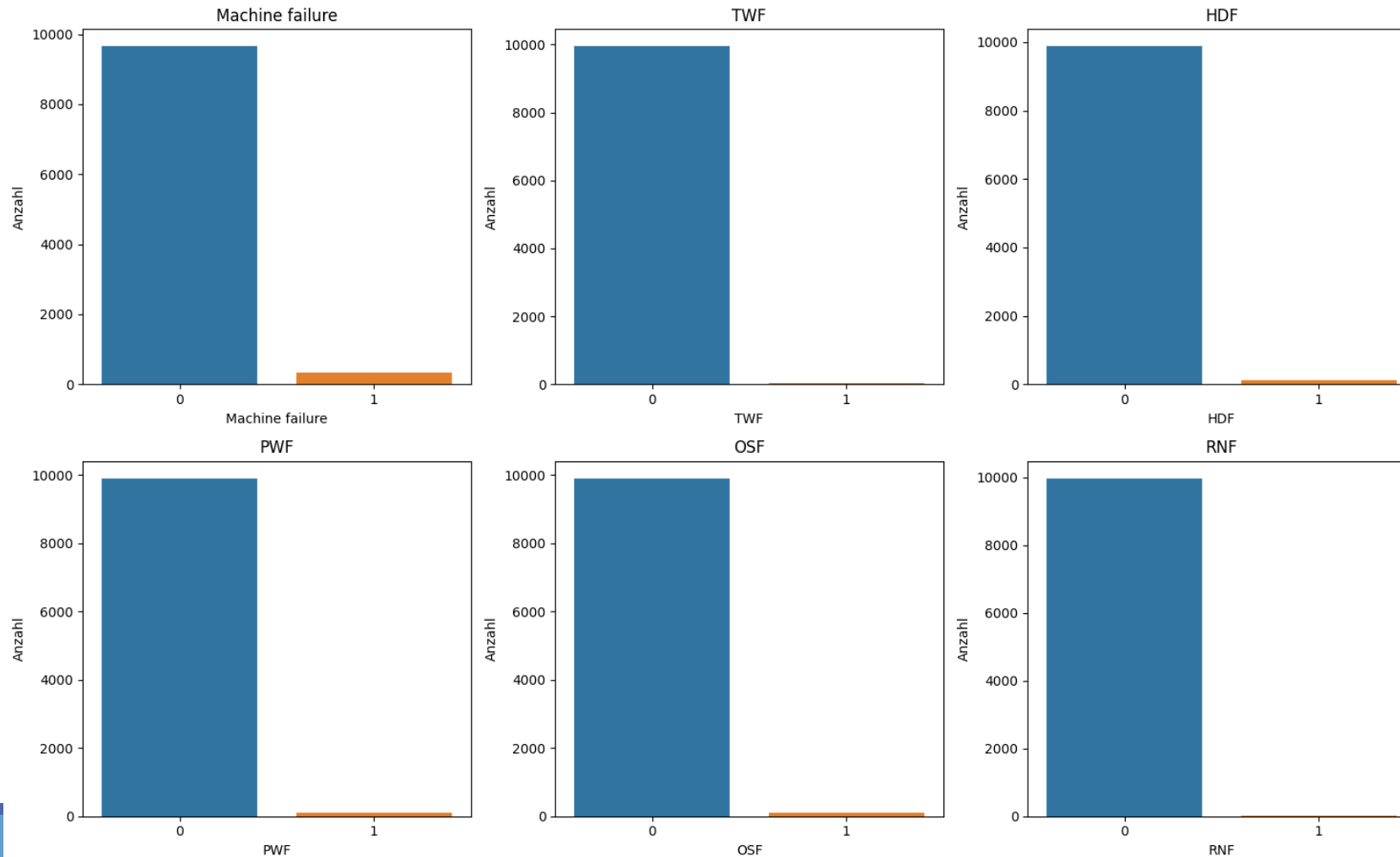
Ausblick

# Datensatz

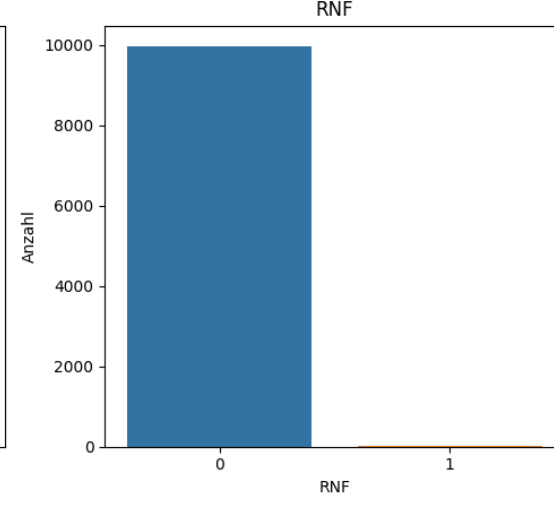
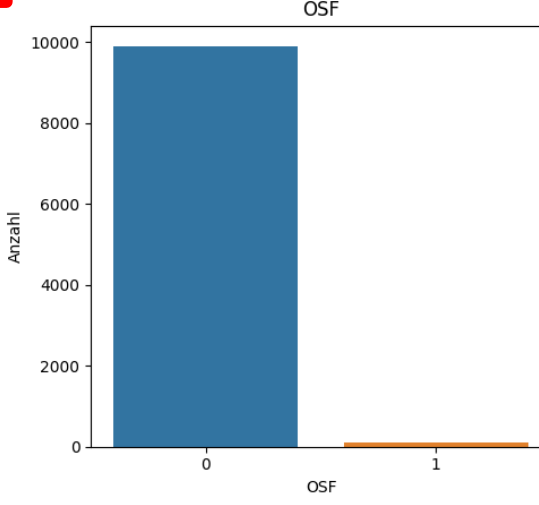
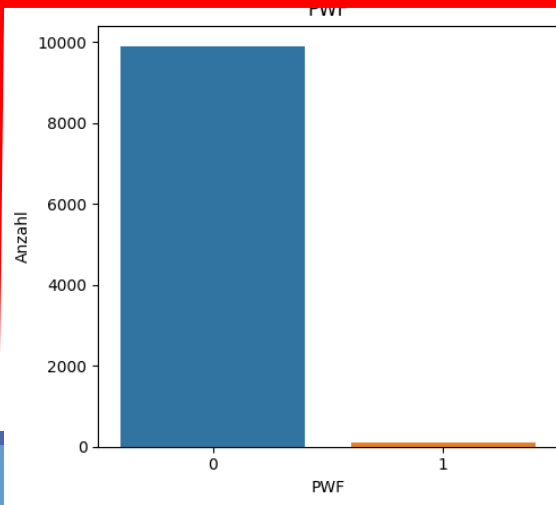
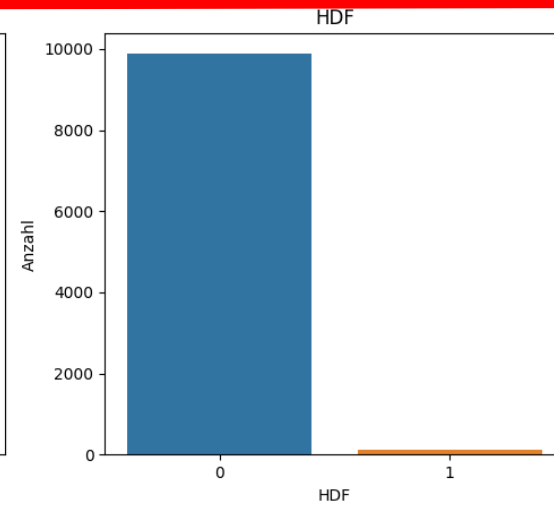
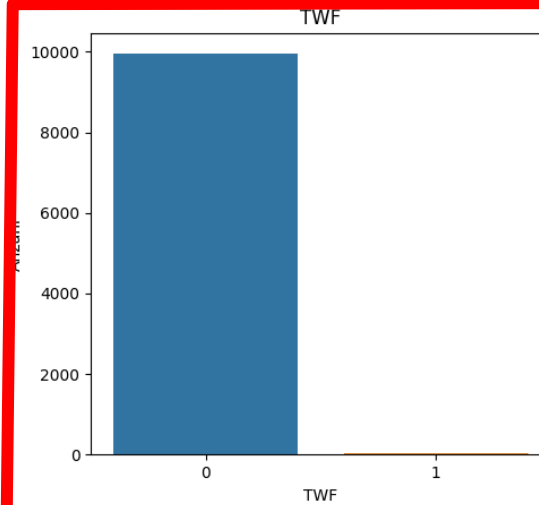
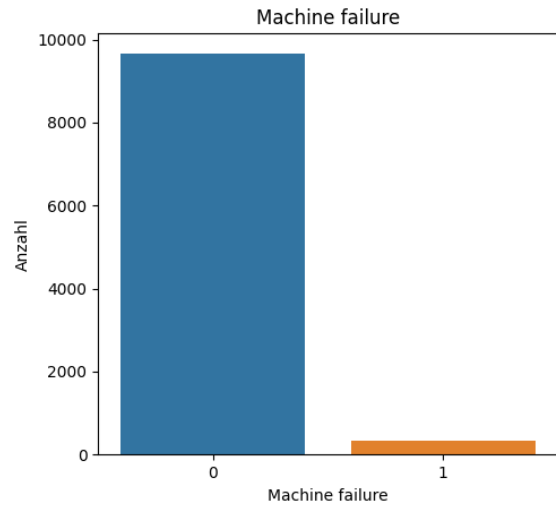
---

- 10.000 Herstellungsprozesse von Produkten
- 14 Attribute:
  - Eindeutige Kennung
  - Produktqualität
  - Dauer der Werkzeugnutzung
  - Luft- & Prozesstemperatur
  - Drehzahl & Drehmoment
  - Maschinenausfall & Fehlerart

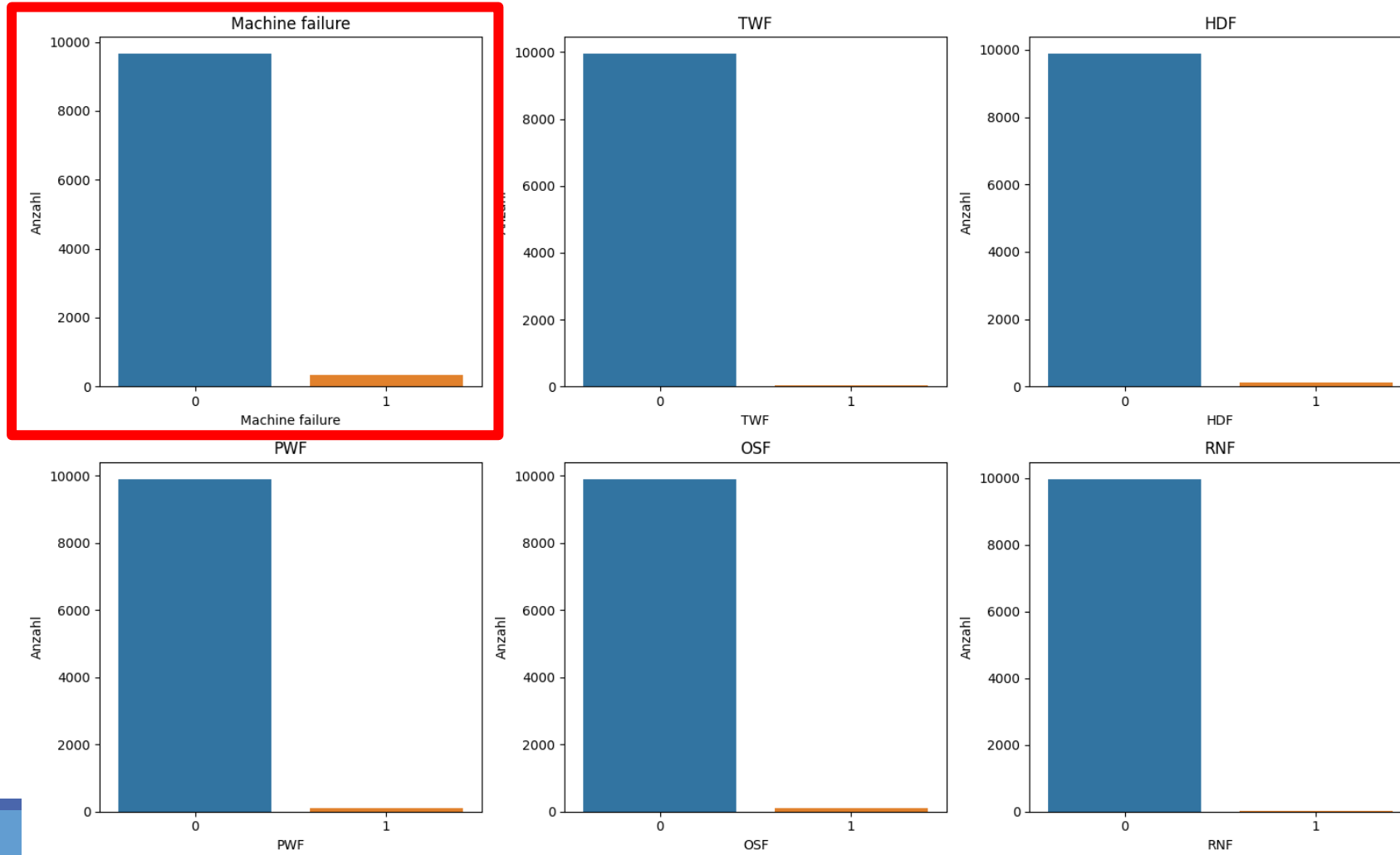
# Balancing der Klassen



# Balancing der Klassen

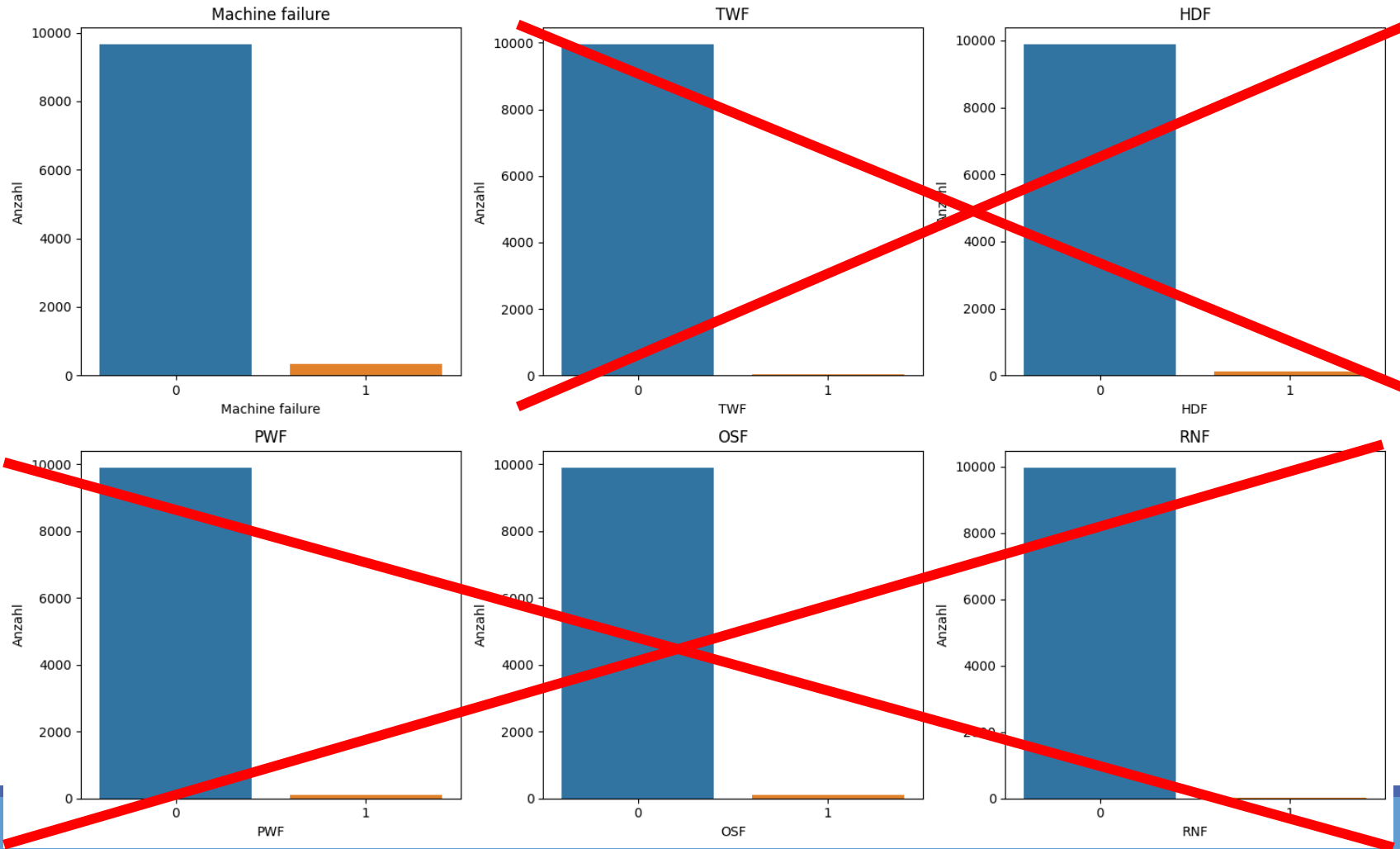


# Balancing der Klassen





# Balancing der Klassen



# Gradient Boosting

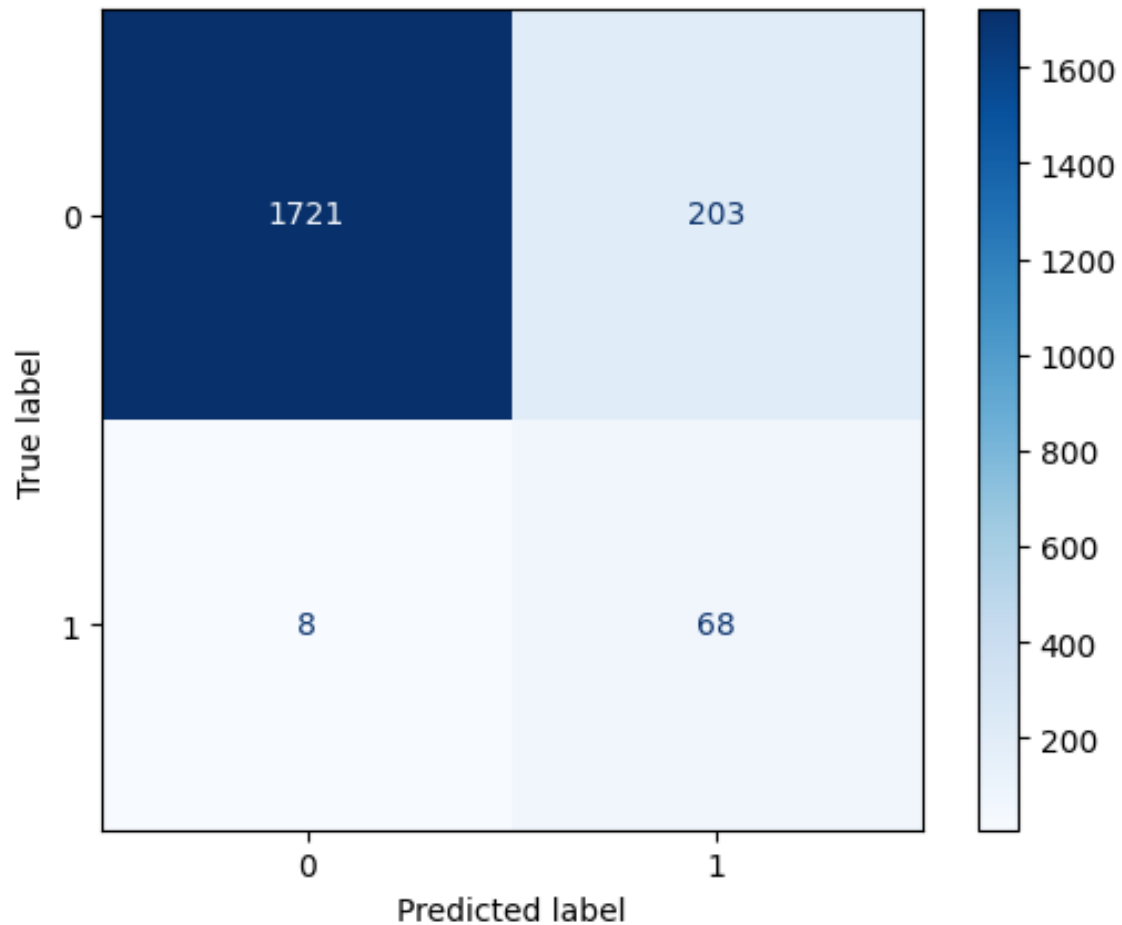
---

- **Ensemble-Methode** kombiniert mehrere schwache Lernmodelle zu einem starken Lernmodell
- **Boosting** ist eine Form des Ensemble-Learnings, bei dem Modelle nacheinander trainiert werden, wobei jedes Modell die Fehler des vorherigen optimiert
- **Gradient Boosting** optimiert nacheinander die Residuenfehler der vorherigen Prädiktoren
- Verwendung von **Entscheidungsbäumen** als Prädiktoren für das Modell
- Entscheidungsbäume zerlegen Datensatz basierend auf Entscheidungskriterien und Grenzwerten (Torque  $\geq 40$  Nm)

# XGBoost

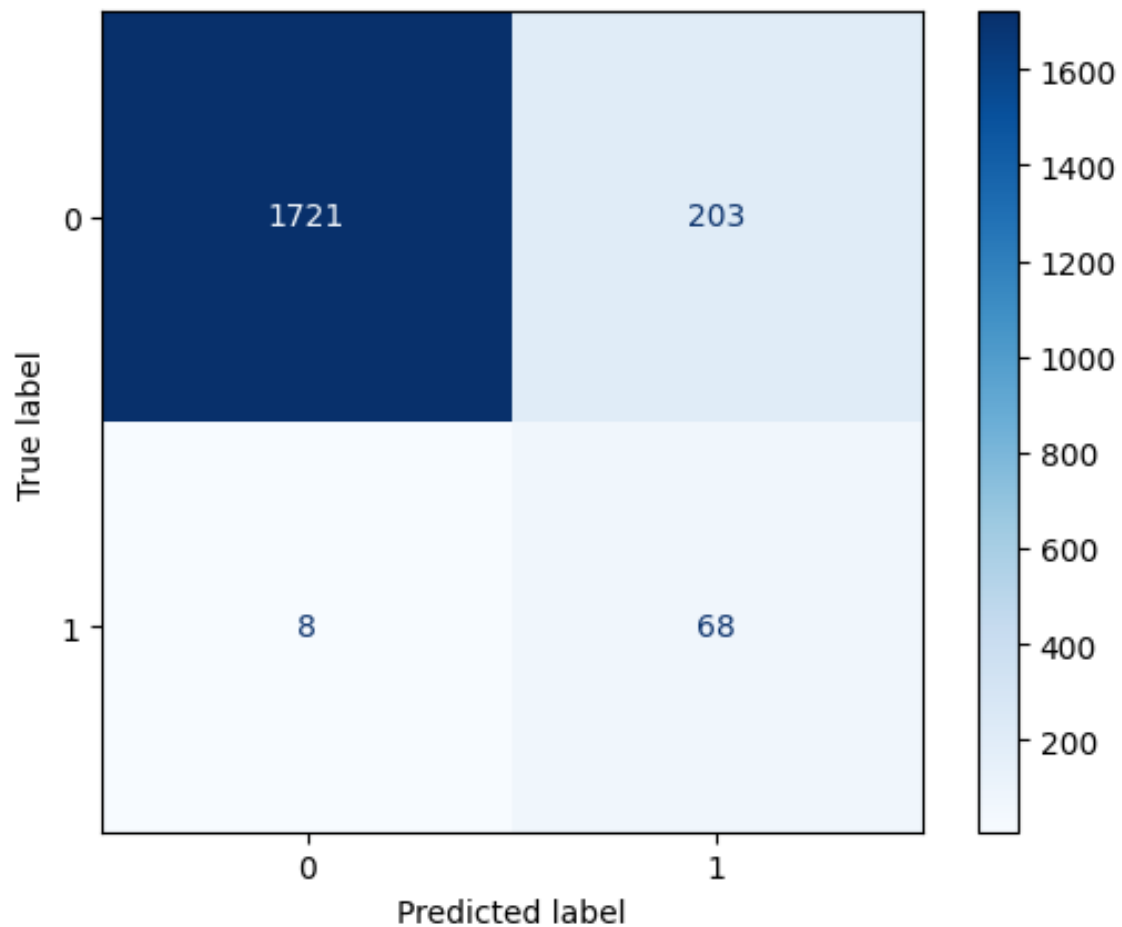
---

- Leistungsstarke Implementierung des Gradient Boostings
- Viele anpassbare Hyperparameter zur Anpassung der Leistung des Modells
- Hyperparameter für Klassenungleichgewicht: „**scale\_pos\_weight**“
- Hyperparameter aus Grid-Search:
  - **learning\_rate**: Schrittweite oder Lernrate bei der Aktualisierung der Modellgewichte nach jeder Iteration
  - **max\_depth**: Maximale Tiefe bzw. Ebenen eines Entscheidungsbaumes
  - **n\_estimators**: Anzahl Bäume im Ensemble
  - **colsample\_bytree**: Anteil der verwendeten Attribute zur Erstellung eines Entscheidungsbaumes
  - **subsample**: Anteil der Trainingsinstanzen zum Trainieren eines Baumes
- Optimale Wertkombination der Hyperparameter wurde mittels Grid-Search bestimmt



# Confusion Matrix

- bildet Anzahl der richtigen und falschen Klassifikationen anhand der vorhergesagten und der tatsächlichen Klasse ab

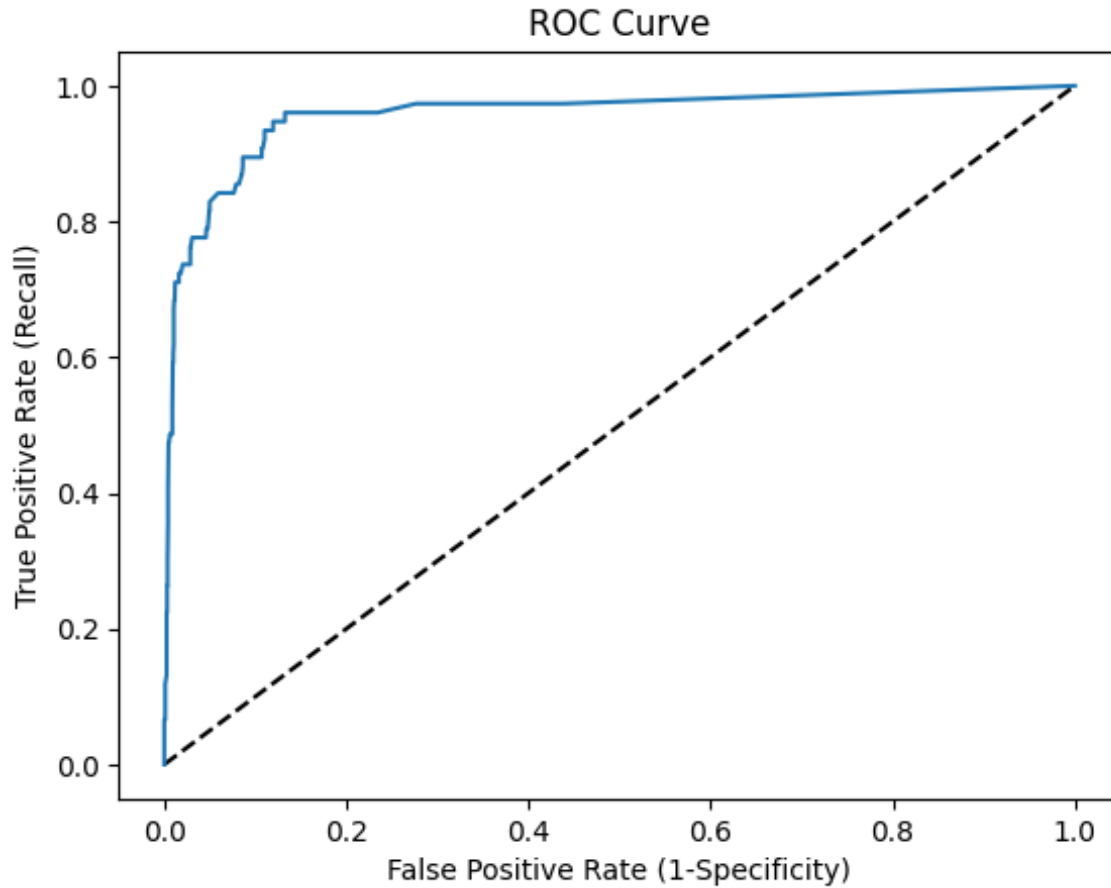


# Confusion Matrix

- 68 Maschinenausfälle korrekt vorhergesagt
- 8 Maschinenausfälle falsch klassifiziert

$$Recall = \frac{68}{68+6} \approx 0,89$$

→ Modell erkennt die meisten tatsächlichen Maschinenausfälle



# ROC & AUC

- Zufälliger Klassifikator  $AUC = 0,5$
- Perfektes Modell  $AUC = 1$
- Erstelltes Modell durchschn.  $AUC = 0,951$

→ Modell wesentlich stärker als zufälliger Klassifikator

# Zusammenfassung

---

- Recall und der AUC-Wert verdeutlichen die Leistungsfähigkeit des Modells
- Hypothese verifiziert

## Nutzen:

- Wahrscheinlichkeit für Maschinenausfall kann mit Modell angegeben werden
- durch Vorhersage können Maschinenausfällen verhindert werden
- Minimierung von reaktiven Wartungen und Ausfallzeiten

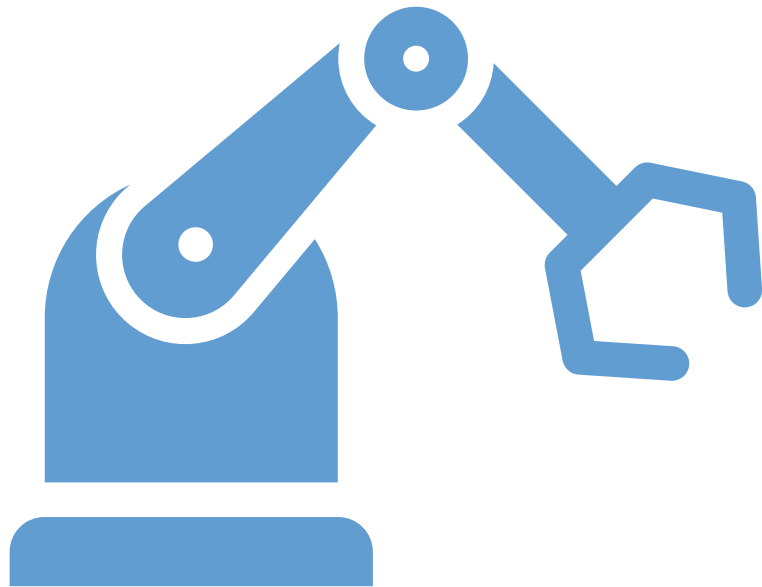
# Ausblick

---

- Weitere Optimierungen zur noch besseren Vorhersage von Maschinenausfällen
- Integration des Modells in die Produktionsumgebung in Echtzeit
- Erweiterung des Modells auf einzelne Fehlerarten







Vielen Dank!