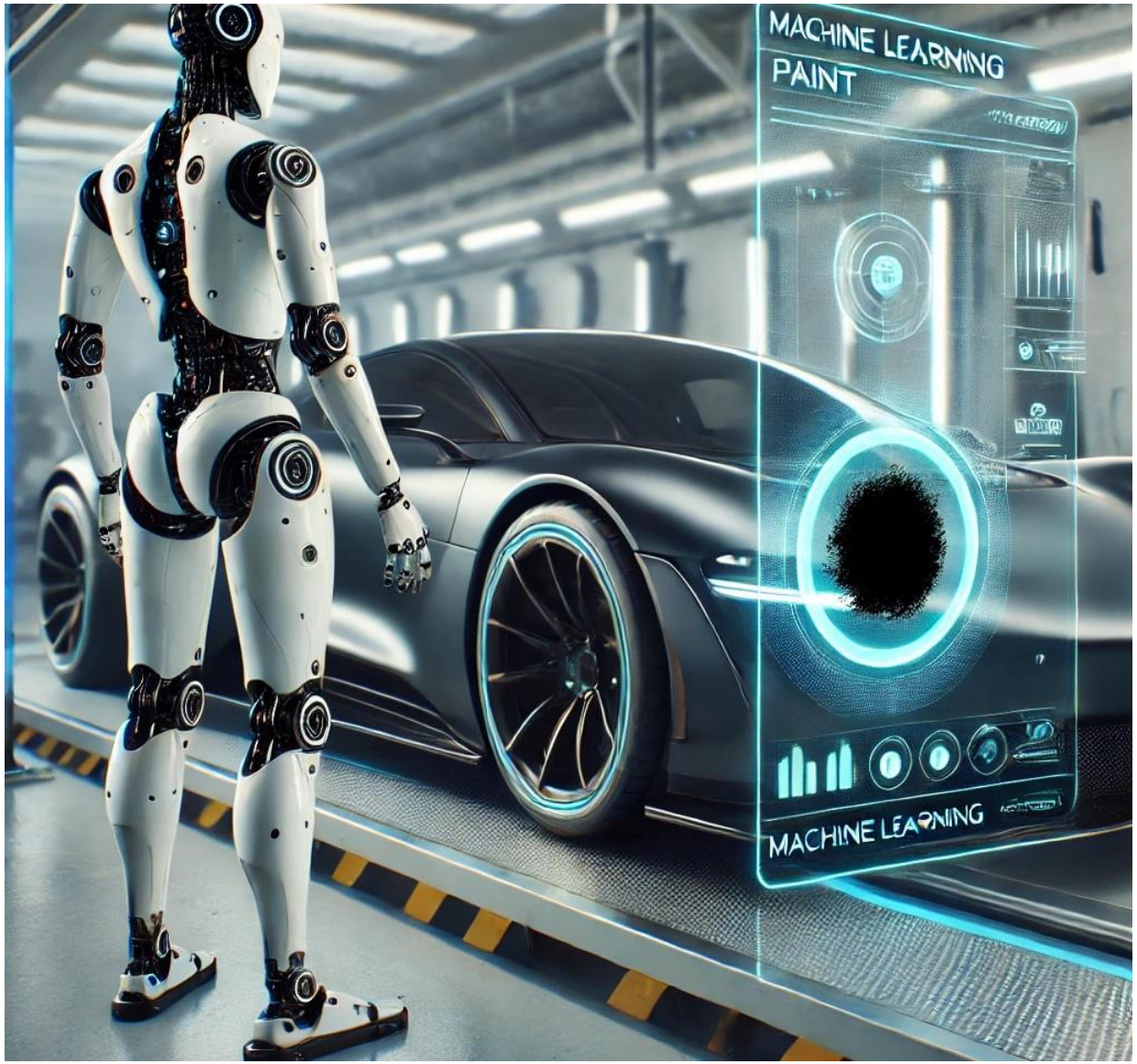
Applied Machine Learning Seminararbeit



Datensatz Lackdefekte

Gründe für den Einsatz von ML

- Wir verwenden ML, um die Schwere von Produktionsdefekten vorherzusagen
- ML-Algorithmen können komplexe Muster und Beziehungen in Daten erkennen, die für Menschen schwer zu durchschauen sind
- ML-Modelle können den Prozess der Datenanalyse und Vorhersage automatisieren. Sobald das Modell trainiert ist, kann es schnell und effizient Vorhersagen treffen
- Lernen und Besserwerden

Die Aufgabe, die Daten und mögliche Performance Metriken (Definition Mitchell)

- **Task:** Vorhersagen der Schwere von Produktionsdefekten
- **Experience:** Der verwendete Datensatz enthält Merkmale wie Versuchsnummer, Kennung des Defekts, Defektkategorie, Status, Maximale und minimale z-Werte, Volumenangaben, Länge, Farbe der Beschichtung, Schwere des Defekts
- **Performance Measure:** Als Performance-Metriken werden die Genauigkeit (Accuracy), die Präzision (Precision) und die Wiederabrufquote (Recall) verwendet, um die Effektivität des Modells in der Identifikation tatsächlicher Abwanderungsfälle zu bewerten.

ML Kategorie Zuordnung

- **Da der Datensatz gelabelt ist, fällt die vorliegende Aufgabe in die Kategorie des Supervised Learning und Klassifikation. Diese Art des Lernens ermöglicht es dem Modell, aus den vorhandenen gelabelten Daten zu lernen und Vorhersagen für neue, nicht gelabelte Daten zu treffen.**
- Gelabelte Daten: Der Datensatz ist gelabelt. Es gibt eine abhängige Variable, defect_severity, die Aufschluss über den Schweregrad des Defektes gibt
- Klassifikation: Wir behandeln die Variable defect_severity als kategoriale Variable, daher handelt es sich um eine Klassifikationsaufgabe. Es ist auch möglich, die Variable ordinal zu skalieren und eine Regression zu nutzen.
- Lernprozess: Das Modell lernt aus diesen gelabelten Daten, indem es Zusammenhänge zwischen den Features (wie max_z, min_z, Volumen usw.) und den Labels erkennt.- Vorhersagen für neue Daten: Nachdem das Modell trainiert wurde, kann es genutzt werden, um die Labels für neue, nicht gelabelte Daten vorherzusagen.

Explorative Datenanalyse

Der Datensatz besteht aus 11 Spalten und 2693 Reihen

	trial	defect_id	defect_category	status	max_z	min_z	lower_volume	upper_volume	length	paint_color	defect_severity
0	1	20	category_1	0	5.72	0.50	0.000387	0.002803	1.070	B	small
1	1	142	category_1	0	12.21	1.11	0.002033	0.006628	1.730	B	small
2	1	98	category_1	0	4.72	0.28	0.000125	0.003138	0.961	B	small
3	1	13	category_4	0	2.48	21.62	0.039500	0.009190	2.430	B	irreparable
4	1	57	category_4	0	0.88	5.30	0.007397	0.000757	1.010	B	medium
5	1	40	category_1	0	6.56	0.59	0.000524	0.005026	1.210	B	small
6	1	136	category_1	0	10.20	0.45	0.000597	0.007903	1.530	B	small

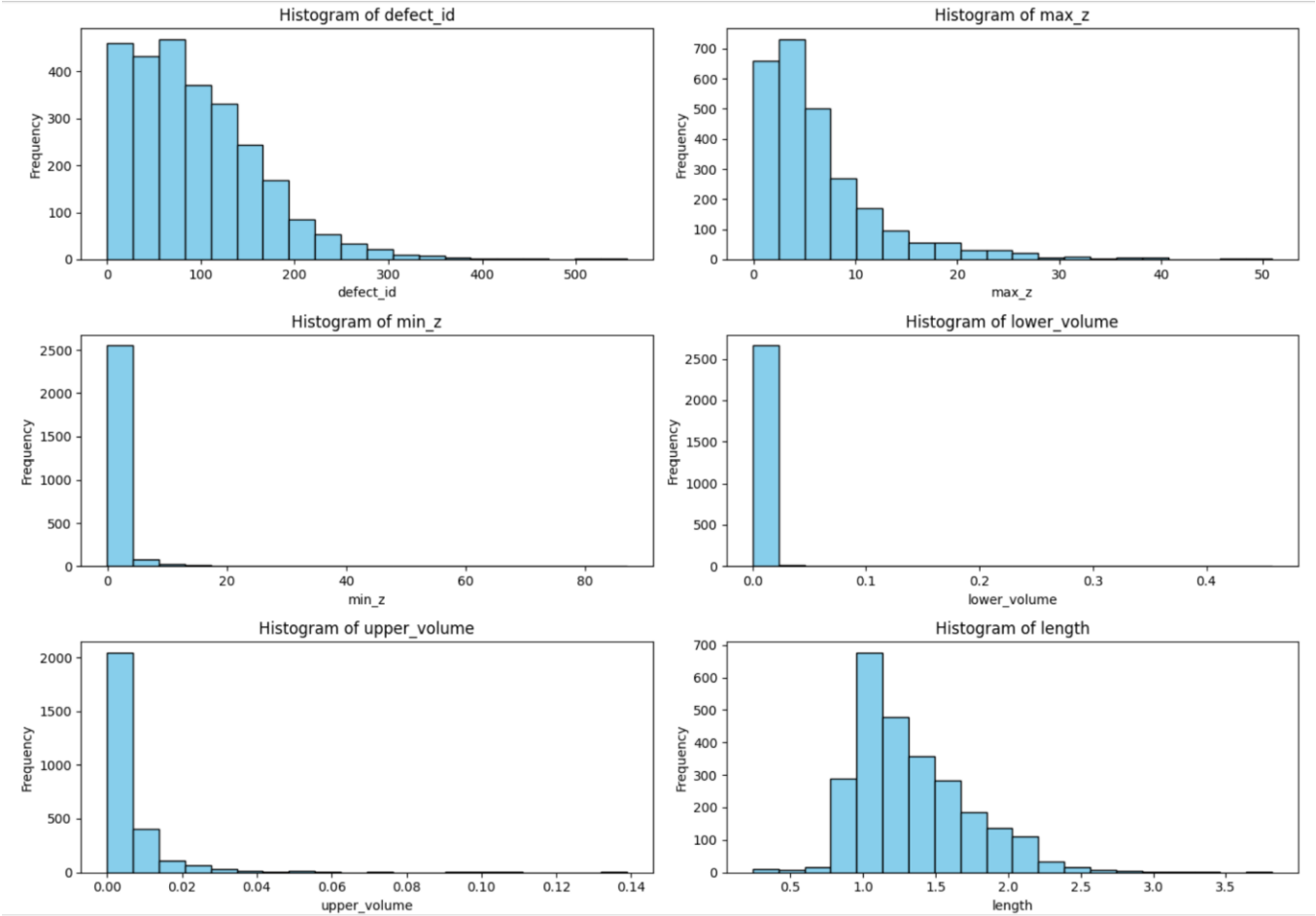
Zusammenfassende Statistiken

	trial	defect_id	status	max_z	min_z	lower_volume	upper_volume	length
count	2693.0	2693.000000	2693.0	2650.000000	2683.000000	2683.000000	2683.000000	2621.000000
mean	1.0	94.607130	0.0	6.457649	1.427600	0.002506	0.005853	1.338877
std	0.0	69.721219	0.0	6.304988	3.453644	0.015254	0.008645	0.397659
min	1.0	0.000000	0.0	-0.080000	-0.050000	0.000000	-0.000000	0.239000
25%	1.0	40.000000	0.0	2.480000	0.470000	0.000333	0.001536	1.050000
50%	1.0	83.000000	0.0	4.725000	0.780000	0.000713	0.003385	1.240000
75%	1.0	134.000000	0.0	8.322500	1.270000	0.001629	0.006627	1.570000
max	1.0	554.000000	0.0	50.910000	86.930000	0.458000	0.138800	3.820000

- **count:**
- Anzahl der Datenpunkte
- **mean:**
- Durchschnittswert
- **std:**
- Standardabweichung (Maß für die Streuung der Daten)
- **min:**
- Minimalwert
- **25% (1. Quartil):**
- Wert, unter dem 25% der Daten liegen
- **50% (Median):**
- Wert, unter dem 50% der Daten liegen
- **75% (3. Quartil):**
- Wert, unter dem 75% der Daten liegen
- **max:**
- Maximalwert

Deskripive Statistik

- defect_id: Die meisten Defekte haben niedrige IDs, und die Verteilung ist rechtsschief, was auf eine abnehmende Häufigkeit höherer IDs hinweist.
- max_z: Die Werte konzentrieren sich im niedrigen Bereich, und die Verteilung ist rechtsschief, was auf wenige hohe Werte hinweist.
- min_z: Die Werte liegen hauptsächlich nahe Null, was eine starke Konzentration im unteren Bereich zeigt. Die Verteilung ist ebenfalls rechtsschief.
- lower_volume: Die Werte sind überwiegend niedrig, und die Verteilung ist rechtsschief, was auf eine Häufung kleiner Werte hinweist.
- upper_volume: Die meisten Werte sind nahe Null konzentriert, und die Verteilung ist rechtsschief, mit wenigen höheren Werten.
- length: Die Werte konzentrieren sich um 1, und die Verteilung ist leicht rechtsschief.
- Variablen wie trial und status wurden nicht visualisiert, da ihre Werte konstant sind und keine nützlichen Informationen liefern.



•**min_z und lower_volume (0.89)**:

•Diese beiden Variablen weisen eine sehr starke positive Korrelation auf. Ein Anstieg von min_z ist mit einem entsprechenden Anstieg des lower_volume verbunden. Dies deutet auf eine starke Abhängigkeit zwischen diesen Merkmalen hin.

•**max_z und upper_volume (0.85)**:

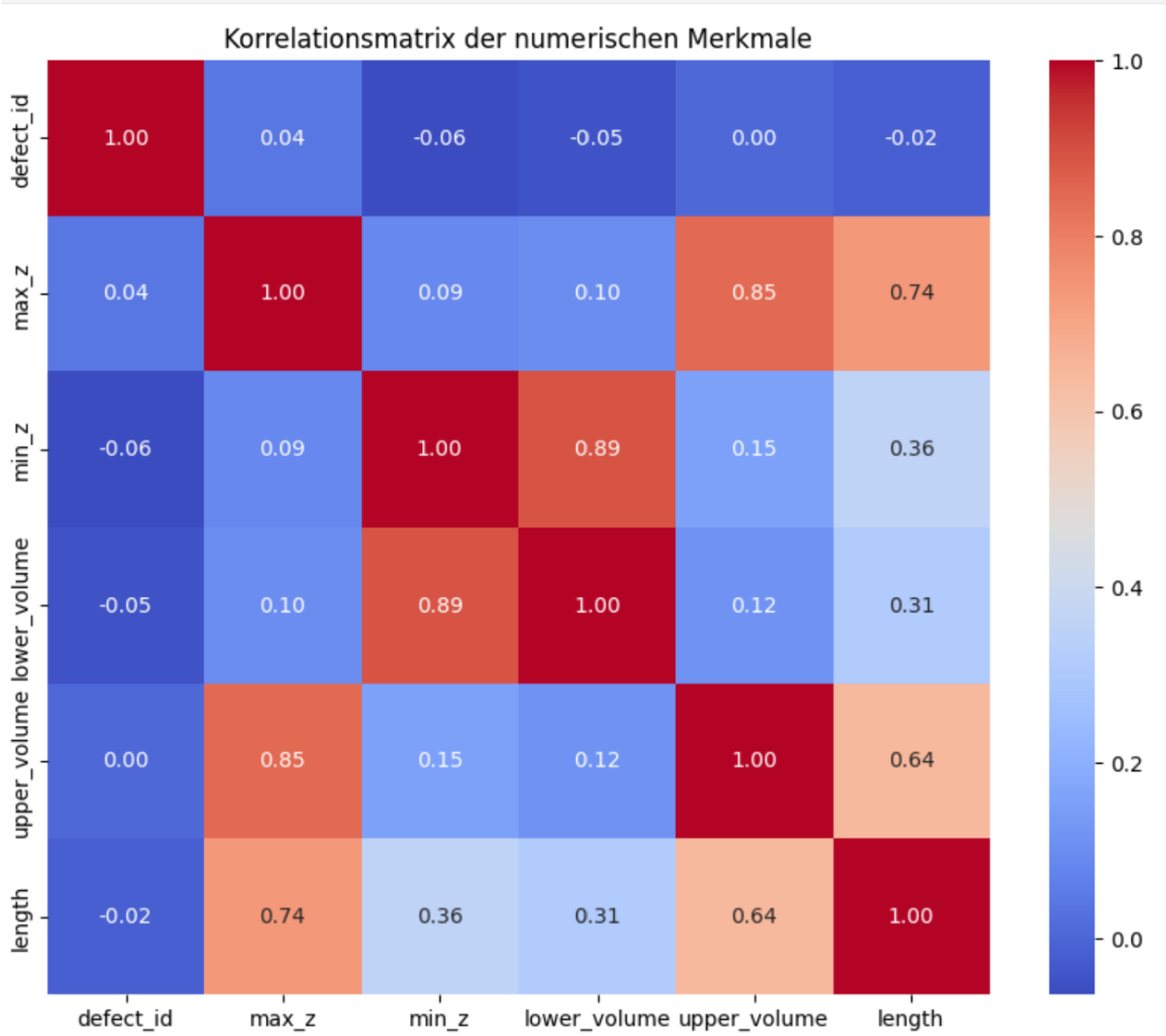
•Auch hier gibt es eine starke positive Korrelation. Höhere Werte von max_z sind oft mit einem höheren upper_volume verbunden, was auf eine ähnliche Abhängigkeit wie bei min_z und lower_volume hinweist.

•**max_z und length (0.74)**:

•Es gibt eine signifikante positive Korrelation zwischen max_z und length. Dies zeigt, dass längere Objekte tendenziell höhere maximale Z-Werte aufweisen, was darauf hindeutet, dass Länge und maximale Höhe proportional zueinander sind.

•**upper_volume und length (0.64)**:

•Diese Korrelation ist ebenfalls stark positiv. Objekte mit einem größeren oberen Volumen sind tendenziell länger, was auf einen Zusammenhang zwischen oberen Volumen und Länge hinweist.



Umgang mit fehlenden Daten

```
Fehlende Daten pro Spalte:
trial               0
defect_id           0
defect_category     0
status              0
max_z              43
min_z              10
lower_volume       10
upper_volume       10
length             72
paint_color         0
defect_severity     0
dtype: int64
```

```
Fehlende Daten pro Spalte nach Imputation:
trial              0
defect_id          0
defect_category    0
status             0
max_z              0
min_z              0
lower_volume       0
upper_volume       0
length             0
paint_color        0
defect_severity    0
dtype: int64
```

Datenimputation: Ersetzt fehlende Werte in einem Datensatz durch geschätzte Werte basierend auf den vorhandenen Daten.

```
Eine Übersicht, wie oft der Wert 'Unknown' in den Spalten auftritt:
trial              0
defect_id          0
defect_category    0
status             0
max_z              0
min_z              0
lower_volume       0
upper_volume       0
length             0
paint_color        0
defect_severity    0
dtype: int64
```

Umgang mit Ausreißern

trial und status:

Beide Variablen zeigen konstante Werte ohne Varianz, daher bieten die Boxplots keine weiteren Informationen.

defect_id:

Der Boxplot zeigt eine breite Verteilung der Defekt-IDs mit einigen Ausreißern. Der Großteil der Werte liegt im unteren Bereich, während die Ausreißer nach oben hin sichtbar sind.

max_z:

Die Werte von max_z zeigen eine signifikante Streuung mit mehreren Ausreißern. Die meisten Werte sind relativ niedrig, aber es gibt einige extrem hohe Werte.

min_z:

Der Boxplot von min_z zeigt eine starke Konzentration der Werte nahe null, mit einigen Ausreißern, die weit höher liegen.

lower_volume:

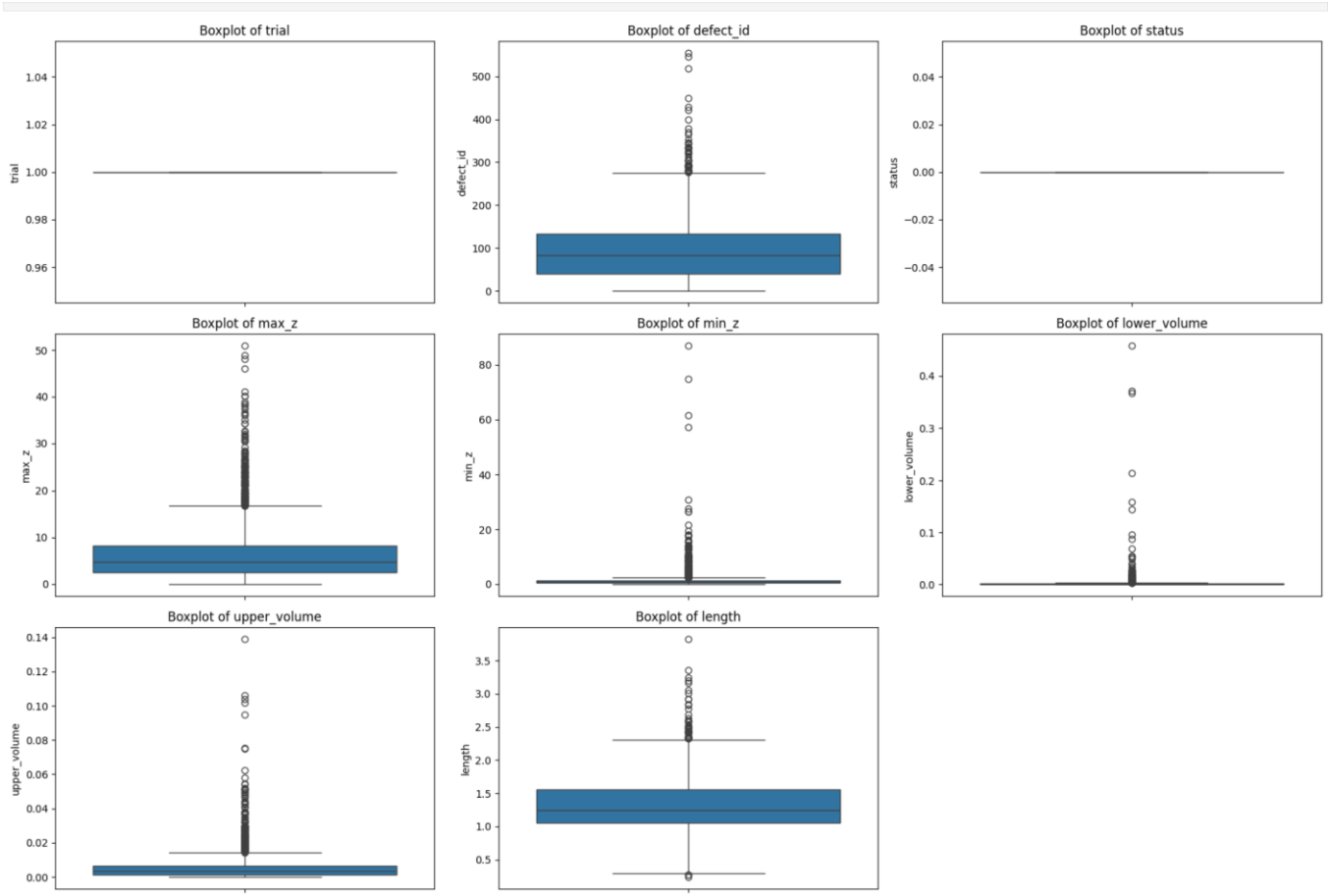
Die meisten Werte von lower_volume liegen nahe null, aber es gibt einige Ausreißer, die deutlich höher sind.

upper_volume:

Ähnlich wie bei lower_volume liegen die meisten Werte von upper_volume nahe null, mit einigen höheren Ausreißern.

length:

Der Boxplot zeigt, dass die Werte von length überwiegend zwischen 0.5 und 1.5 liegen, mit einigen Ausreißern, die bis zu 3.5 reichen.



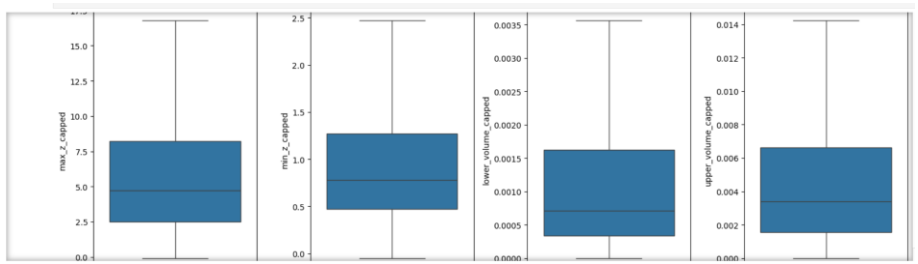
Ausreißer entfernen mit Capping

Capping:

Extremwerte, die unterhalb der unteren Grenze oder oberhalb der oberen Grenze lagen, wurden auf diese Grenzwerte begrenzt. Dies reduziert den Einfluss extremer Werte auf die Datenanalyse.

Beibehaltung bestimmter Ausreißer:

- defect_id und length: Hier wurden Ausreißer bewusst beibehalten. Gründe dafür sind:
  - Ausreißer können wichtige Informationen über seltene, aber bedeutende Ereignisse oder Anomalien enthalten, z.B. seltene, kritische Defekte oder besonders große/kleine Objekte.
  - Domänenspezifisches Wissen kann darauf hinweisen, dass bestimmte extreme Werte realistisch und wichtig sind.
  - Das Entfernen von Ausreißern kann zu einem Verlust wichtiger Variabilität führen, wodurch die Datenintegrität beeinträchtigt werden könnte.



Datenfehler suchen

Es wurden keine Datenfehler gefunden

Überprüfung und Bereinigung von Duplikaten

```
# Vor dem Entfernen auf Duplikate prüfen
duplicates_before = data.duplicated().sum()

# Duplikate entfernen
data_cleaned = data.drop_duplicates()

# Nach dem Entfernen auf Duplikate prüfen
duplicates_after = data_cleaned.duplicated().sum()

duplicates_before, duplicates_after
```

```
(32, 0)
```

Daten standardisieren und transformieren

- **Kategorische Merkmale kodiert**:
  - paint_color, defect_category, defect_severity mit OneHotEncoder kodiert.
- **Numerische Merkmale skaliert und transformiert**:
  - Merkmale: max_z, min_z, lower_volume, upper_volume, length.
  - **StandardScaler**: Skaliert Daten auf Mittelwert 0, Standardabweichung 1.
  - **PowerTransformer (Yeo-Johnson)**: Normalisiert Datenverteilung und reduziert Schiefe.
- **Pipeline erstellt und angewendet**:
- Kombiniert Kategorisierung und Skalierung in einem Schritt.
- Nicht spezifizierte Spalten bleiben unverändert.
- Neuer DataFrame mit transformierten Werten erstellt.

Feature Selection und Extraktion

•**Ziel**: Reduzierung der Anzahl der Features durch Entfernen hoch korrelierter Merkmale.
•**Vorgehen**:
1.Korrelation berechnen:
    •Ermittlung der Korrelationen zwischen den numerischen Merkmalen.
    •Auswahl der Korrelationen, die über einem Schwellenwert liegen (hier: 0.5).
2.Identifikation hoch korrelierter Paare:
    •Festlegen eines Schwellenwertes für hohe Korrelation (0.8).
    •Finden von Merkmalspaaren, deren Korrelation diesen Wert überschreitet.
3.Merkmale entfernen:
    •Auswahl der zu entfernenden Merkmale aus den korrelierten Paaren.
    •Entfernen dieser Merkmale aus dem Datensatz.
4.Ergebnis:
    •Reduzierter Datensatz mit weniger redundanten Features, was die Modellleistung verbessern kann.

Zusammenführung, Modifikation und Aufteilung der Zielvariablen

Kombiniert die Werte von defect_severity_not repairable mit defect_severity_irreparable und löscht die Spalte defect_severity_not repairable.

Ziel- und Eingabemerkmale definieren:

Zielvariablen: defect_severity_irreparable, defect_severity_medium, defect_severity_small.

Eingabemerkmale: Alle anderen Spalten außer den Zielvariablen.

Trainings- und Testdatensätze erstellen:

Aufteilen der Daten in Trainings- (80%) und Testdatensätze (20%) mit train_test_split.

Ergebnis überprüfen:

Größen der Trainings- und Testdatensätze:

Trainingsdaten: 2154 Einträge, 11 Spalten.

Testdaten: 539 Einträge, 11 Spalten.

Warum das alles ?

- **Trainingsdaten:** Zum Trainieren des Modells
- **Validierungsdaten:** Zur Hyperparameter Optimierung und Bewertung des Modells während dem Training
- **Testdaten:** Zur Endgültigen Bewertung des Modell. Zeigt, wie gut da Modell auf neuen, ungesehenen Daten funktioniert.
- **Warum wird gesplittet?**
- Um Überanpassungen zu verhindern
- Ein Modell, dass nur auf Trainingsdaten getestet wird, könnte sich die Daten merken und nicht gut auf neue Daten generalisieren

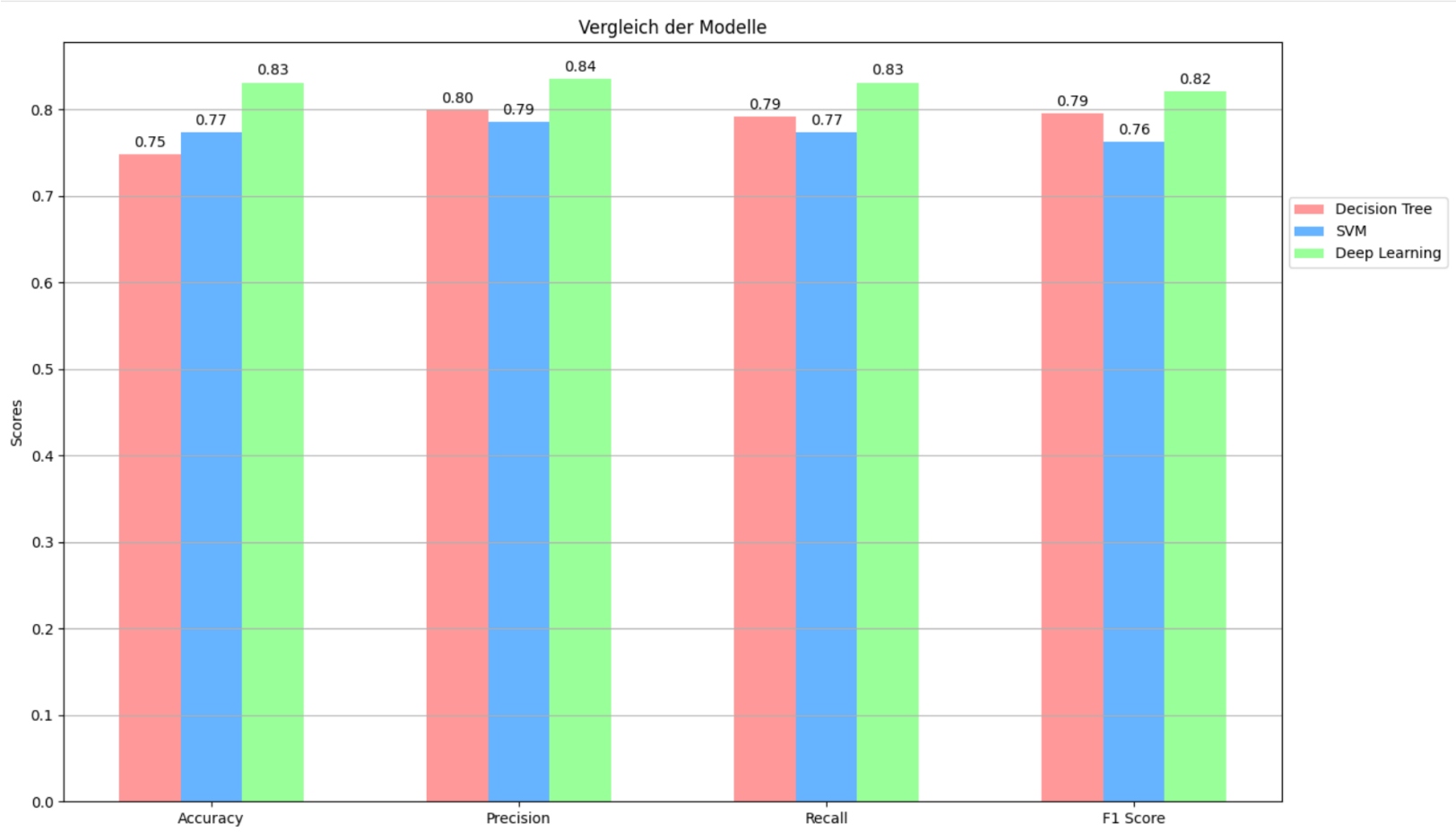
Übersicht über Modelle, Loss-Funktionen und Bewertungsmetriken

Modell	Beschreibung	Loss-Funktion	Aktivierungsfunktion	Metriken
Decision Tree	Ein Modell, das Entscheidungen basierend auf den Eingabevariablen trifft, indem es eine Baumstruktur verwendet. Interpretierbar und leicht zu visualisieren.	Hinge Loss	-	Accuracy, Precision, Recall, F1
Support Vector Machine (SVM)	Ein Modell, das eine Hyperplane findet, die die Datenpunkte der verschiedenen Klassen am besten trennt. Besonders effektiv bei hochdimensionalen Daten.	Hinge Loss	-	Accuracy, Precision, Recall, F1
Deep Learning	Ein Modell, das künstliche neuronale Netzwerke mit vielen Schichten verwendet, um komplexe Muster in großen Datenmengen zu lernen.	Binary Cross-Entropy Loss	ReLU (versteckte Schichten)	Accuracy, Precision, Recall, F1
			Sigmoid (Ausgabeschicht)	

Bewertungsmetriken

- **Accuracy:** Wie viele Vorhersagen sind insgesamt richtig.
- **Precision:** Wie viele der als positiv vorhergesagten Fälle tatsächlich positiv sind.
- **Recall:** Wie viele der tatsächlich positiven Fälle richtig erkannt wurden.
- **F1-Score:** Ein Maß für die Balance zwischen Precision und Recall.

Vergleich der Modelle nach den Training



Was ist Over- und Underfitting?

- **Overfitting:** wenn ein Modell zu gut auf den Trainingsdaten performt, aber schlecht auf neuen, ungesehenen Daten
- **Underfitting:** wenn ein Modell weder auf Trainingsdaten noch auf neuen Daten gut performt

Überprüfen auf Over- und Underfitting

Trainingsgenauigkeit (rote Kurve):

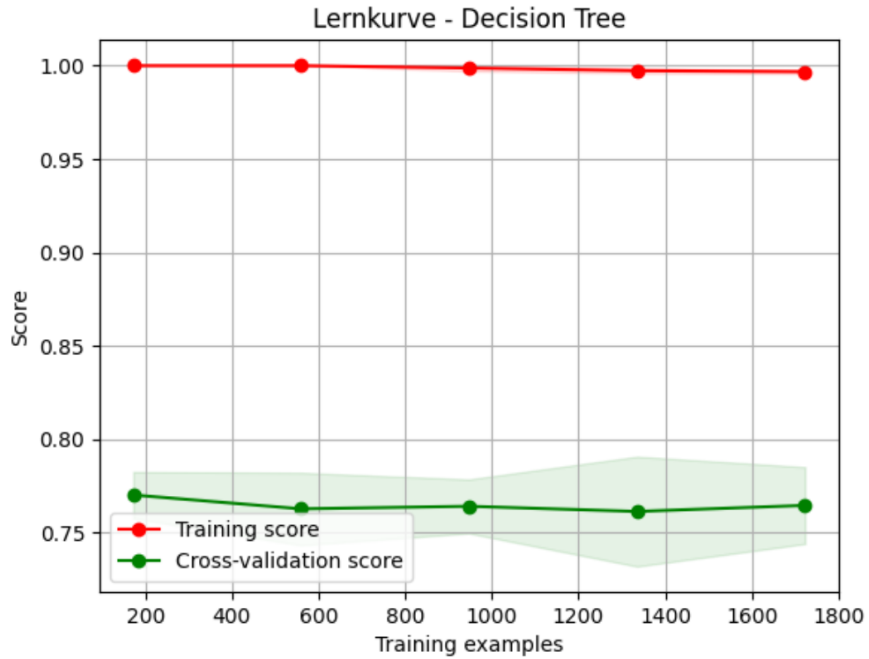
- Bleibt konstant bei 1.0.
- Perfekte Klassifikation der Trainingsdaten.
- Hinweis auf Overfitting, da das Modell die Trainingsdaten vollständig gelernt hat.

Kreuzvalidierungsgenauigkeit (grüne Kurve):

- Bleibt bei etwa 0.75 konstant.
- Moderate Leistung auf Validierungsdaten.
- Großer Abstand zur Trainingsgenauigkeit deutet auf Schwierigkeiten bei der Generalisierung hin.

Interpretation:

- Das Modell zeigt Overfitting: Es lernt die Trainingsdaten perfekt, performt aber mäßig auf neuen Daten.



Überprüfen auf Over- und Underfitting

Trainingsgenauigkeit (rote Kurve):

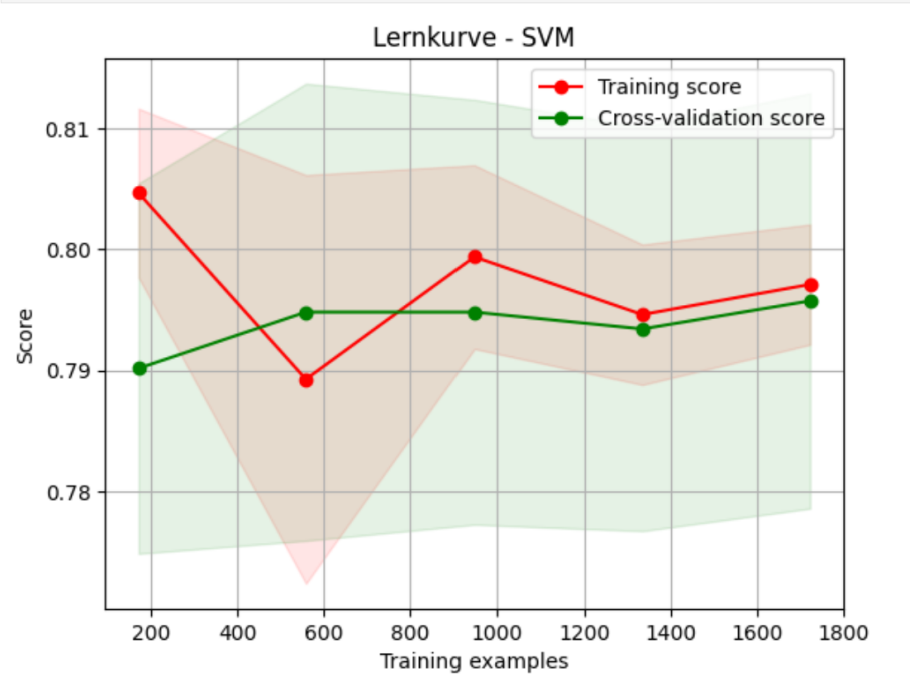
- Schwankt leicht, bleibt aber relativ stabil um 0.80.
- Deutet darauf hin, dass das Modell verschiedene Muster in den Trainingsdaten erfasst.

Kreuzvalidierungsgenauigkeit (grüne Kurve):

- Bleibt konstant zwischen 0.79 und 0.80.
- Zeigt, dass das Modell gut generalisiert und keine Anzeichen von Overfitting oder Underfitting aufweist.

Interpretation:

- Kleiner Abstand zwischen Trainings- und Validierungsgenauigkeit.
- Überlappung der Fehlerbänder deutet auf konsistente Leistung auf Trainings- und Validierungsdaten hin.
- Keine Anzeichen von Overfitting oder Underfitting.
- Gute Balance zwischen Bias und Varianz, was auf die Zuverlässigkeit des Modells hinweist.



Überprüfen auf Over- und Underfitting

Modellgenauigkeit (linkes Diagramm):

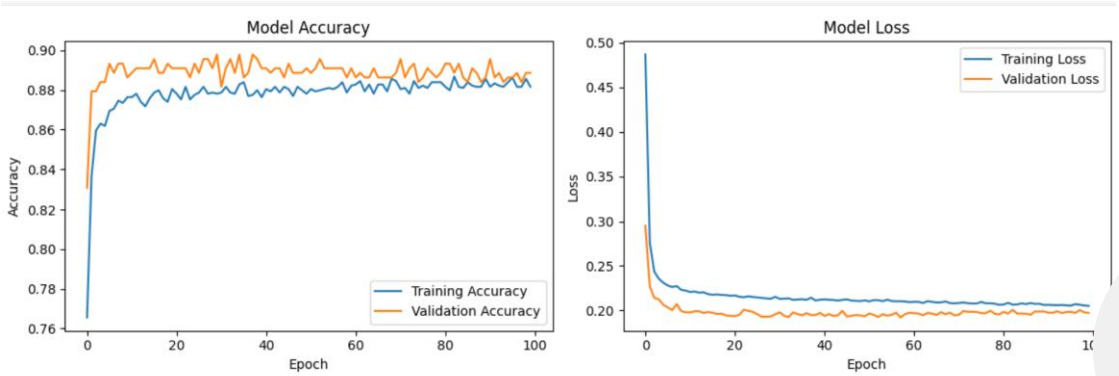
- **Trainingsgenauigkeit** (blaue Linie): Steigt schnell an und stabilisiert sich nach etwa 10 Epochen.
- **Validierungsgenauigkeit** (orange Linie): Steigt ebenfalls schnell an und bleibt meist höher als die Trainingsgenauigkeit.
- Die enge Übereinstimmung zwischen Trainings- und Validierungsgenauigkeit zeigt, dass das Modell gut generalisiert, ohne Overfitting oder Underfitting.

Modellverlust (rechtes Diagramm):

- **Trainingsverlust** (blaue Linie): Sinkt schnell zu Beginn und erreicht ein Plateau nach etwa 10 Epochen.
- **Validierungsverlust** (orange Linie): Sinkt ebenfalls schnell und bleibt konstant niedriger als der Trainingsverlust.
- Ein niedrigerer Validierungsverlust im Vergleich zum Trainingsverlust deutet auf eine gute Generalisierung des Modells hin.

Interpretation:

- Das Modell zeigt keine Anzeichen von Overfitting, da die Validierungsmetriken nicht signifikant schlechter als die Trainingsmetriken sind.
- Die Konsistenz zwischen Trainings- und Validierungsmetriken spricht für die Zuverlässigkeit und Robustheit des Modells.
- Das Modell hat die zugrunde liegenden Muster in den Daten erfolgreich gelernt und kann diese Erkenntnisse auf neue Daten anwenden.



Ensemble Modell

- Kombination von Decision Tree, Support Vector Machine (SVM) und Deep Learning Modell.
- Ziel: Verbesserung der Gesamtleistung durch Nutzung der Stärken verschiedener Modelle.
- **Vorgehen:**
    - Einzelne Vorhersagen der drei Modelle wurden kombiniert, um die Mehrheitsklasse für jede Dimension zu bestimmen.

Interpretation:

- Das Ensemble-Modell zeigt eine gute Gesamtleistung in Bezug auf Genauigkeit, Präzision, Recall und F1-Score.
- Die enge Übereinstimmung zwischen den Metriken zeigt, dass das Modell gut generalisiert und sowohl auf Trainings- als auch auf Validierungsdaten konsistent performt.
- Durch die Kombination verschiedener Modelle nutzt das Ensemble-Modell die individuellen Stärken jedes Ansatzes und gleicht deren Schwächen aus.
- Das Modell zeigt keine signifikanten Schwächen und liefert konsistente Vorhersagen, was es zu einer robusten Wahl für die gegebene Vorhersageaufgabe macht.

Evaluation Results - Ensemble Model:

Metric	Value
Accuracy	0.80705
Precision	0.831297
Recall	0.808905
F1-Score	0.807968

Hyperparameter Optimierung

Verwendete Hyperparameter:
Decision Tree:
•max_depth
•min_samples_split
•min_samples_leaf
•**SVM:**
•svc_c
•svc_kernel: linear (Optionen: linear, poly, rbf)
•svc_gamma
•**Deep Learning:**
•num_layers
•num_units
•dropout_rate

Maßnahmen gegen Over- und Underfitting

Decision Tree:

- **Overfitting verhindern:**
  - **Begrenzung der Baumtiefe (max_depth)**: Verhindert, dass der Baum zu tief wird und sich zu stark an die Trainingsdaten anpasst.
  - **Erhöhung der minimalen Samples für Knotenaufteilung (min_samples_split)**: Reduziert die Wahrscheinlichkeit, dass der Baum kleine, spezifische Muster lernt.
  - **Erhöhung der minimalen Samples für Blattknoten (min_samples_leaf)**: Stellt sicher, dass jeder Blattknoten genügend Datenpunkte hat, um verlässliche Vorhersagen zu treffen.

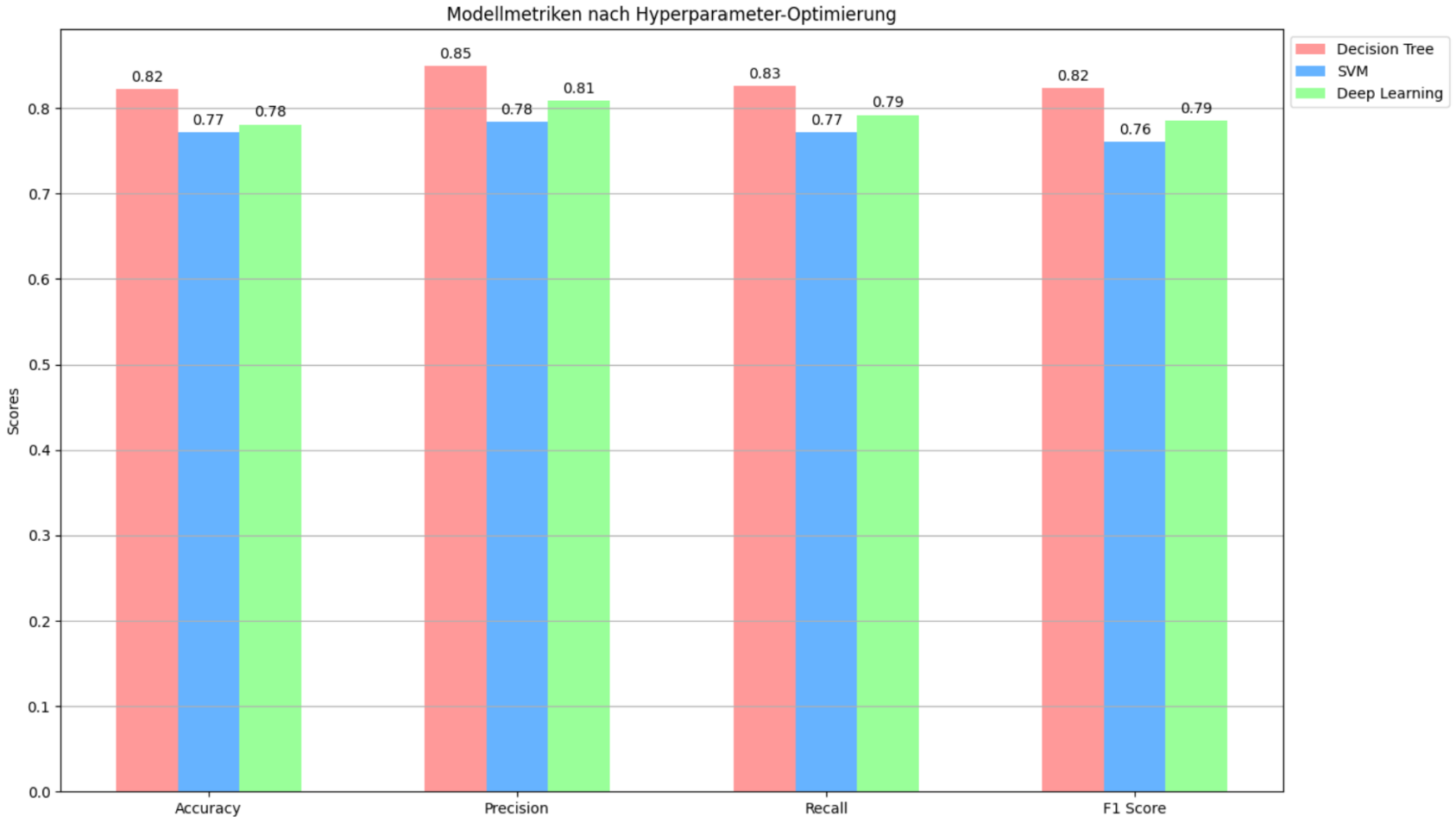
SVM:

- **Overfitting verhindern:**
  - **Regularisierung mit svc_c**: Kontrolliert die Komplexität des Modells, indem es bestraft wird, wenn es sich zu sehr an die Trainingsdaten anpasst.
  - **Auswahl eines passenden Kernels (svc_kernel)**: Bestimmt die Art der Entscheidungsgleichung und hilft dabei, das Modell auf die Struktur der Daten abzustimmen.
  - **Feineinstellung des Gamma-Werts (svc_gamma)**: Beeinflusst die Reichweite des Einflusses eines einzelnen Trainingsbeispiels, um Überanpassung zu vermeiden.

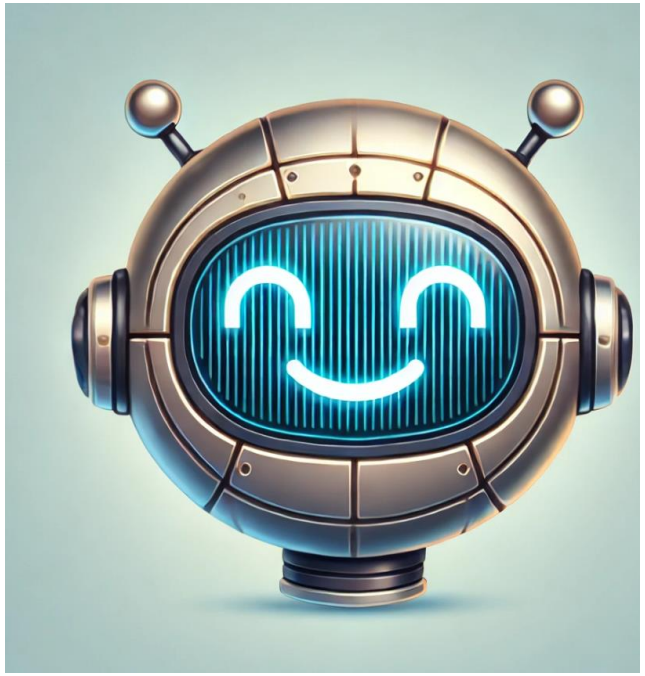
Deep Learning:

- **Overfitting verhindern:**
  - **Verwendung von Dropout-Layern (dropout_rate)**: Deaktiviert zufällig Neuronen während des Trainings, um zu verhindern, dass das Modell zu stark an spezifische Merkmale der Trainingsdaten angepasst wird.
  - **Einsatz von Early Stopping (EarlyStopping)**: Beendet das Training frühzeitig, wenn sich die Leistung auf den Validierungsdaten nicht mehr verbessert, um Überanpassung zu vermeiden.
  - **Aufteilung der Daten in Trainings- und Validierungsdaten (validation_split)**: Überwacht die Modellleistung auf einem separaten Validierungssatz während des Trainings, um Überanpassung zu erkennen und zu verhindern.

Vergleich der Modelle nach der Optimierung



Vielen Dank für Ihre Aufmerksamkeit!



Quellen

- **ChatGPT, OpenAI:**
- OpenAI ChatGPT
- **Verwendete Skripte aus dem Kurs "Applied Machine Learning":**
  - Applied_ML_2_2, Seiten 12, 15, 23, 34, 36, 37, 45
  - Applied_ML_2_1, Seiten 36, 41, 45, 46, 49, 50
  - Applied_ML_1, Seiten 21, 23, 38, 42, 47, 51, 54, 55
  - Applied_ML_7_Model_Evaluation, Seiten 28, 29, 30
  - Applied_ML_8_Model_selection, Seite 21
- **DALL-E, OpenAI:**
- OpenAI DALL-E