

# **Prognose des Füllgrads eines Distributionslagers am Beispiel des automatischen Kleinteilelagers der Phoenix Contact GmbH & Co. KG**

**Hausarbeit**

im Studiengang

Data Science & Business Analytics

vorgelegt von

**Sebastian Wölk**

am 17. November 2021

an der Hochschule der Medien Stuttgart

Prüfer/in:

Prof. Dr. Mischa Seiter

---

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis .....</b>	<b>2</b>
<b>1     <b>Einleitung</b>.....</b>	<b>3</b>
1.1    Unternehmensvorstellung.....	3
1.2    Problem- und Zielstellung.....	3
<b>2     <b>Lösungsansätze</b> .....</b>	<b>5</b>
2.1    Definition relevanter Verfahren.....	5
2.1.1   Klassische Zeitreihenanalyse mit dem ARIMA-Modell .....	5
2.1.2   Zeitreihenanalyse mit dem XGBoost .....	6
2.1.3   Monte-Carlo-Simulation zur Fortsetzung der Zeitreihe.....	6
2.2    Umsetzung .....	7
2.2.1   Data Preparation .....	7
2.2.2   Data Exploration.....	7
2.2.3   Feature Engineering.....	9
2.2.4   Zeitreihenprognose mit dem ARIMA-Modell .....	11
2.2.5   Zeitreihenprognose mit dem XGBoost.....	12
2.2.6   Zeitreihenprognose mit der Monte-Carlo-Simulation.....	13
<b>3     <b>Evaluation</b> .....</b>	<b>15</b>
3.1    ARIMA-Modell .....	15
3.2    Monte-Carlo-Simulation.....	15
3.3    Entscheidungsbaummodell „XGBoost“ .....	16
<b>4     <b>Zusammenfassung</b>.....</b>	<b>17</b>
4.1    Iterative Prognose mit bestem Modell.....	17
4.2    Ausblick.....	17
<b>Literaturverzeichnis .....</b>	<b>19</b>

---

# 1 Einleitung

## 1.1 Unternehmensvorstellung

Die Phoenix Contact GmbH & Co. KG ist ein deutsches Unternehmen, das Komponenten, Systeme und Lösungen im Bereich der Elektrotechnik, Elektronik und Automation produziert und weltweit vertreibt. Der Stammsitz ist im lippischen Blomberg. Seit der Gründung im Jahr 1923 hat sich das Unternehmen zu einem internationalen Unternehmen entwickelt.

Zur Phoenix Contact-Gruppe gehören 14 Unternehmen in Deutschland sowie mehr als 55 Vertriebsgesellschaften auf der ganzen Welt. Weltweit wird in elf Ländern produziert, woraus sich ein komplexes Geflecht an globalen Warenströmen ergibt. Das Produktspektrum umfasst Komponenten und Systemlösungen für die Energieversorgung, inklusive Wind und Solar, den Geräte- und Maschinenbau sowie den Schaltschrankbau. Mit ihren Produkten bedient die Phoenix Contact-Gruppe Märkte der Automobilindustrie, der regenerativen Energien und der Infrastruktur.

Im Jahr 2020 erzielte die Phoenix Contact GmbH & Co. KG einen Umsatz von ca. 2,4 Mrd. Euro. Der Umsatz im Jahr 2021 wird voraussichtlich ca. 3 Mrd. Euro betragen.

## 1.2 Problem- und Zielstellung

Die Absicherung eines hohen Lieferbereitschaftsgrads bei Fertigprodukten und eine termingetreue Belieferung der Kunden ist eines der wesentlichen Ziele des im vorangegangenen Kapitel kurz vorgestellten Unternehmens und aller daran beteiligten Organisationseinheiten.

Die Funktionsfähigkeit eines mehrstufig und global organisierten Vertriebs- und Distributionsnetzwerks hängt dabei unmittelbar von der Handlungsfähigkeit aller beteiligten Schnittstellen und Organisationseinheiten und einer Ausbalancierung der vorhandenen Kapazitäten entlang der gesamten Wertschöpfungskette ab.

Dreh- und Angelpunkt nahezu 75% aller Warenströme ist das Distributionslager am Stammsitz im lippischen Blomberg. Ebenso wichtig ist die ausgewogene Abstimmung statischer und dynamischer Kapazitäten innerhalb der Logistikprozesse dieses Distributionslagers, um eine termingetreue und schnelle Belieferung der globalen Kunden sicherzustellen und globale Warenstrombewegungen nicht zu gefährden.

Der Lagerfüllgrad dient hierbei als steuerungsrelevante Kennzahl aus dem Bereich der statischen Logistikkapazitäten, die bei Überschreitung kritischer Grenzwerte, aufgrund hoher Automatisierung im hier betrachteten Distributionslagerort, zu einer Verlangsamung oder gar einem Stillstand der Logistikprozesse, und damit der globalen Warenströme, führen kann.

---

Um eine vorausschauende Überwachung des Lagerfüllgrads und die rechtzeitige Einleitung geeigneter Maßnahmen zur Entlastung der statischen Logistikkapazitäten zu ermöglichen, ist es notwendig, die zu erwartende zukünftige Entwicklung des Lagerfüllgrads bzw. dessen Trendverlauf vorhersagen zu können. Auf diese Weise können entsprechende Gegenmaßnahmen, vor Erreichen der kritischen Grenzwerte der statischen Kapazitäten, eingeleitet werden, um die Handlungsfähigkeit der Supply Chain zu gewährleisten.

Ziel dieser Arbeit ist es den historischen Verlauf des Lagerfüllgrads zu analysieren und ein Prognosemodell zur tagesbasierten Vorhersage des zukünftigen Lagerfüllgrads im Distributionszentrum der Phoenix Contact GmbH & Co. KG zu entwickeln.

Konkret sollen im Rahmen dieser Arbeit insgesamt drei verschiedene Verfahren und Vorgehensweisen zur Lösung des beschriebenen Zeitreihenproblems verprobt werden. Anschließend soll im Rahmen einer kurzen Evaluation, anhand definierter Performanceparameter, das beste Modell zur Extrapolation und Vorhersage des zukünftigen Lagerfüllgradverlaufs ausgewählt werden.

---

## 2 Lösungsansätze

Im Folgenden sollen, mit Bezug zu der oben beschriebenen Problem- und Zielstellung, zunächst einige Lösungsmöglichkeiten aufgezeigt, definiert und anschließend deren Umsetzung angegangen werden.

Ausgangsbasis für die hier vorliegende Arbeit ist ein Datensatz mit einer Zeitreihe zum Lagerfüllgrad vom 01.01.2019 bis zum 31.10.2021.

Datensatz	Füllgrad	File_Date	Anzahl Stellplätze	Freie Plätze	Belegte Plätze
1	86.6	2019-01-02	196636	26349	170287
2	86	2019-01-03	196636	27529	169107
3	84.9	2019-01-04	196636	29692	166944
4	83.92	2019-01-07	196636	31619	165017
5	82.54	2019-01-08	196636	34333	162303

Abbildung 1: Datensatz Lagerfüllgrad

### 2.1 Definition relevanter Verfahren

Zur Lösung des eingangs beschriebenen Zeitreihenproblems kommen verschiedene Verfahren aus dem Bereich des maschinellen Lernens in Betracht.

Die im oben abgebildeten Datensatz enthaltene Variable „Belegte Plätze“, die im Rahmen dieser Arbeit die Zielvariable darstellen wird, ist eine numerische und kontinuierliche Größe. Deshalb kommen insbesondere Verfahren zur Lösung von Regressionsproblemen in Betracht. Da es sich bei dem verwendeten Datensatz um eine Zeitreihe handelt, kommen auch spezialisierte Modelle zur Lösung von Zeitreihenproblemen in Betracht.

Im Folgenden werden die im Rahmen dieser Arbeit behandelten Methoden, um das vorliegende Zeitreihenproblem zu lösen, kurz beschrieben und allgemeingültig definiert.

#### 2.1.1 Klassische Zeitreihenanalyse mit dem ARIMA-Modell

Ein klassisches und weit verbreitetes Modell im Rahmen der Zeitreihenanalyse und -prognose ist das „ARIMA“-Modell. Es ist spezialisiert auf die Lösung von Zeitreihenproblemen. Für ein grundsätzliches Verständnis soll diese Methode kurz allgemeingültig definiert werden.

„Die Abkürzung für ARIMA lautet Auto-Regressive Integrated Moving Average. Beim ARIMA-Modell handelt es sich um eine Abwandlung beziehungsweise Fortführung des ARMA-Modells. Das ARIMA-Modell ist eine leistungsstarke Modellklasse, mit der sich Zeitreihen beschreiben und analysieren lassen. Es besitzt einen autoregressiven Teil (AR-Modell) und einen gleitenden Mittelwertbeitrag (MA-Modell).“<sup>1</sup> Die Komponente „I“ erweitert das klassische ARMA-Modell um einen Ansatz der Trendbeseitigung in der

---

<sup>1</sup> Luber 2021.

---

Zeitreihe. Der dem zugrundeliegende Modellparameter wird als "Grad der Differenzierung" bezeichnet. Damit eignet sich das ARIMA-Modell auch für die Vorhersage von Zeitreihendaten, die mehr oder weniger starke Trendverläufe und Saisonalitäten aufweisen.<sup>2</sup>

Das ARIMA-Modell ist im Kontext von Zeitreihenprognosen ein recht vielversprechender Algorithmus, der im weiteren Verlauf auf die vorliegenden Zeitreihendaten angewendet und verprobt (trainiert und getestet) werden soll.

### **2.1.2 Zeitreihenanalyse mit dem XGBoost**

Entscheidungsbaummodelle, wie beispielsweise der Random Forest oder der XGBoost können genutzt werden, um Klassifikations- und Regressionsprobleme zu lösen.

Ein wesentlicher Vorteil des XGBoost-Modells gegenüber anderen Entscheidungsbaummodellen ist, dass er im Rahmen des Modelltrainings in der Lage ist, unterschiedlich komplexe Bäume zu trainieren. Wie komplex, also wie viele Knoten die im Rahmen des Modelltrainings erstellten Entscheidungsbäume haben werden, ist vom jeweiligen Mehrwert des zusätzlichen Knotens, der dem Baummodell hinzugefügt wird, abhängig. Ein damit wichtiger Hyperparameter des Modells ist die „Learning Rate“, die auch als „Schrumpfungswert“ bekannt ist. Dieser Parameter bzw. diese Funktion des Algorithmus schützt einerseits vor „Overfitting“ – also dem Hinzufügen weiterer Knoten zu einem Baummodell, obwohl diese möglicherweise nur noch einen geringen oder sogar gar keinen Mehrwert für die eigentliche Vorhersage liefern – und andererseits erhöht dieser Mechanismus ebenso die Performance des Modells und optimiert die benötigte Rechenleistung. Alle im Rahmen des Modelltrainings erstellen Entscheidungsbäume, die eine unterschiedliche Tiefe und Komplexität aufweisen können, werden am Ende zu einem Gesamtmodell kombiniert, um eine einzige Vorhersage pro Datensatz abzuleiten.

„Boosted Trees“ gelten im Allgemeinen als sehr performante und vielversprechende Algorithmen zur Lösung von Regressionsproblemen und gewinnen an vielen Stellen in der Data Science Community sogar Wettbewerbe. Außerdem kann das Prinzip der Bildung von Entscheidungsbäumen auch von Business-Anwendern nachvollzogen und nachempfunden werden, was die Akzeptanz eines solchen Modells für den produktiven Einsatz im Betrieb erhöhen kann.

### **2.1.3 Monte-Carlo-Simulation zur Fortsetzung der Zeitreihe**

Die Monte-Carlo-Simulation, auch bekannt als Monte-Carlo-Methode, ist ein mathematisches Verfahren, das zur Abschätzung der möglichen Ergebnisse eines ungewissen Ereignisses verwendet wird.<sup>3</sup>

---

<sup>2</sup> Luber 2021.

<sup>3</sup> IBM Cloud Education 2021.

---

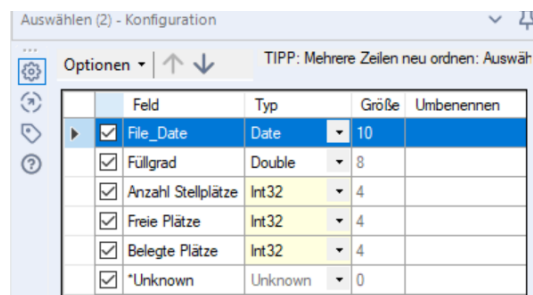
Die Monte-Carlo-Simulation arbeitet mit Wahrscheinlichkeitsverteilungen und Zufallszahlen, um - anders als klassische Prognosemodelle - eine Reihe von möglichen Ergebnissen zu simulieren. Je häufiger diese Berechnungen wiederholt werden, desto genauer kann das Ergebnis werden, da die Anzahl der wahrscheinlichen Ergebnisse bei mehrfacher Wiederholung der Berechnungen zu einer genaueren Festlegung des wahrscheinlichen Wertebereichs der Zielvariable führen.<sup>4</sup>

## 2.2 Umsetzung

Nachdem der betriebswirtschaftliche Kontext, im speziellen die Problem- und Zielstellung, und die relevanten Verfahren, die im Rahmen dieser Arbeit behandelt werden sollen, beschrieben wurden, soll im Folgenden etwas detaillierter auf die konkrete Umsetzung und den entstandenen Machine-Learning-Workflow eingegangen werden.

### 2.2.1 Data Preparation

Im Rahmen der Datenvorbereitung wurden im Rahmen dieser Arbeit ausschließlich einige Datentypen nach dem Importieren der zu untersuchenden Daten bestimmt. Die benötigte Zeitreihe lag bereits in gut strukturierter und aufbereiteter Form vor.



	Feld	Typ	Größe	Umbenennen
<input checked="" type="checkbox"/>	File_Date	Date	10	
<input checked="" type="checkbox"/>	Füllgrad	Double	8	
<input checked="" type="checkbox"/>	Anzahl Stellplätze	Int32	4	
<input checked="" type="checkbox"/>	Freie Plätze	Int32	4	
<input checked="" type="checkbox"/>	Belegte Plätze	Int32	4	
<input checked="" type="checkbox"/>	*Unknown	Unknown	0	

Abbildung 2: Ändern von Datentypen innerhalb der Zeitreihe

### 2.2.2 Data Exploration

Im nächsten Schritt sollen die Zeitreihendaten visuell dargestellt und mithilfe einiger Zeitreihenplots analysiert werden.

Als Zielvariable soll die im Folgenden dargestellte Kennzahl „Belegte Plätze“ dienen.

---

<sup>4</sup> IBM Cloud Education 2021.

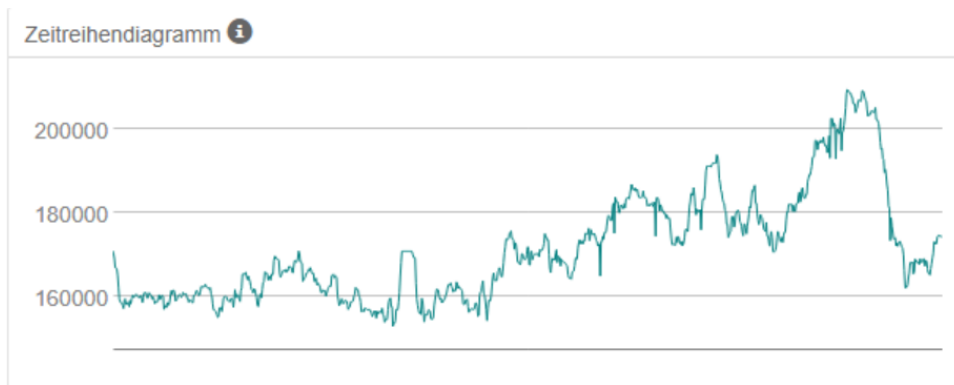


Abbildung 3: Zeitreihendiagramm zum Lagerfüllgrad (Belegte Plätze pro Tag)

Visuell ist im Zeitreihendiagramm zunächst ein grundsätzlich positiver Trend zu erkennen, der allerdings durch starkes Rauschen beeinflusst wird. Außerdem zeichnet sich kein erkennbares saisonales Muster innerhalb der Zeitreihe ab, welches für die Modellierung und Prognose dienlich sein könnte. Dies bestätigen ebenfalls die folgenden beiden Diagramme, die das Thema der Saisonalität in der vorliegenden Zeitreihe näher beschreiben.



Abbildung 4: Saisonale Darstellung eines Zerlegungsdiagramms

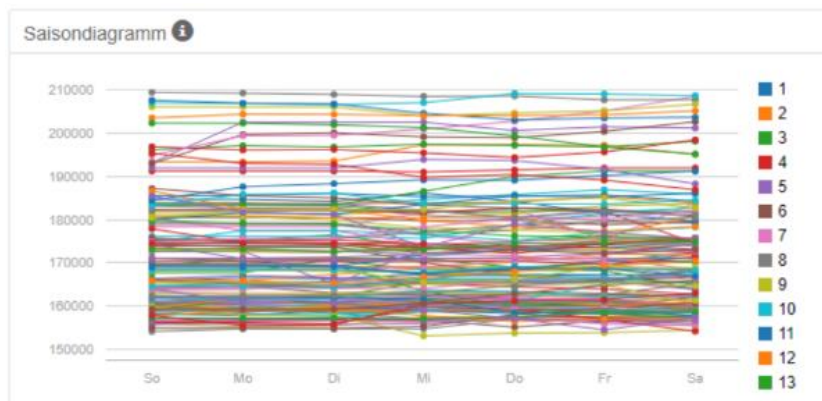


Abbildung 5: Saisonales Diagramm der Zeitreihe

Ein letzter und weiterer Aspekt, der an dieser Stelle genauer betrachtet werden soll, ist der Aspekt der Autokorrelation und partiellen Autokorrelation.



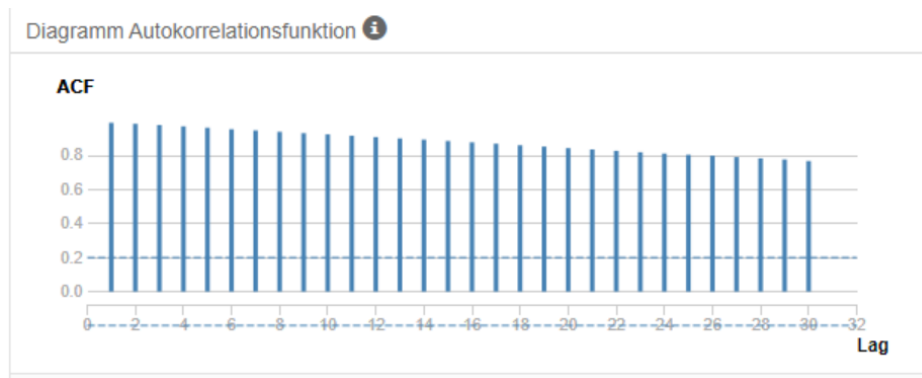


Abbildung 6: Autokorrelationsdiagramm

Im Autokorrelationsdiagramm wird ersichtlich, dass die Zielvariable stark mit ihren zeitverzögerten Merkmalsausprägungen (zu früheren Zeitpunkten in der Messreihe) korreliert. Ein „Lag“ (Verzögerung) der Zielvariable von 30 weist immer noch einen signifikanten Zusammenhang zu ihrem gegenwärtigen Wert auf.

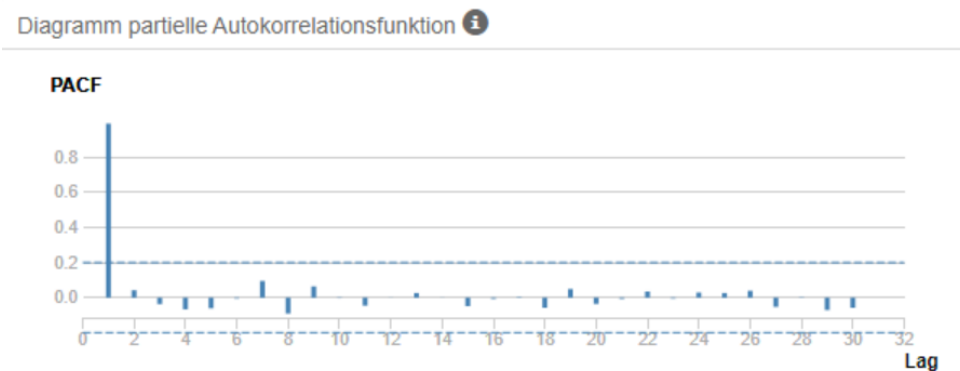


Abbildung 7: Partielle Autokorrelation

Bei der partiellen Autokorrelation wird allerdings auch deutlich, dass mit „Lag 1“ bereits die stärkste Korrelation vorliegt und von „Lag 1“ die größte Erklärungskraft des Zusammenhangs hinsichtlich der gegenwärtigen Ausprägung der Zielvariable ausgeht. Die Hinzunahme weiterer Lags (zeitverzögerter Ausprägungen der Zielvariable) beeinflusst die Beschreibung bzw. Erklärung des vorliegenden Zusammenhangs nicht mehr signifikant.

### 2.2.3 Feature Engineering

Im Rahmen des Feature Engineering für Zeitreihenanalysen und -prognosen kommen diverse Möglichkeiten in Betracht. Es geht im Wesentlichen darum, dem Zeitreihendatensatz weitere selbst berechnete Variablen hinzuzufügen, welche später durch geeignete Modelle als Prädiktoren für die Vorhersage der Zielvariable genutzt werden können.

Im Kontext dieser Arbeit ist das Feature Engineering insbesondere für das Entscheidungsbaummodell „XGBoost“ relevant. „Mit einem effizienten Feature-Engineering-Schritt wird die Vorhersagekraft unseres Modells erhöht.“<sup>5</sup>

„Einige der häufigsten Funktionen sind:

- Datums- und Uhrzeitfunktionen,
- Verzögerungsfunktionen,
- Fensterfunktionen und
- Domänenspezifische Funktionen.“<sup>6</sup>

Im Rahmen dieser Arbeit wurden auf den verwendeten Datensatz primär Verzögerungsfunktionen und Fensterfunktionen in unterschiedlichen Ausprägungen angewendet. Außerdem wurden die aus der Aktienkursanalyse bekannten „Bollinger Bänder“ als neue Features berechnet.

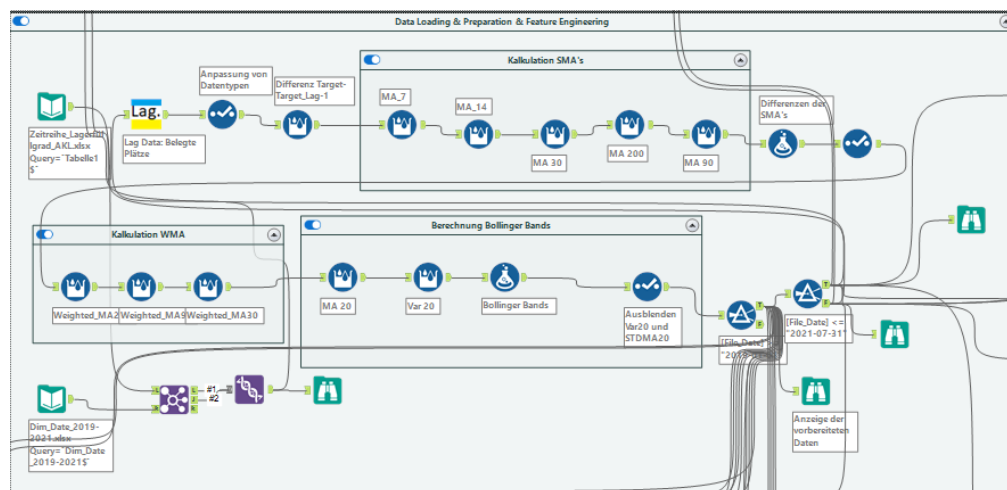


Abbildung 8: Berechnung verschiedener verzögerter und gefensterter Variablen

Zusätzlich wurden auch Datumsfunktionen in die nähere Betrachtung einbezogen. Hierbei wurden Features, wie z.B. der Monat, der jeweilige Tag des Jahres, der Tag der Woche, der Tag des Monats etc. aus dem Datum extrahiert und auf direkte, als auch zeitverzögerte, Korrelation geprüft. Vergeblicherweise konnten sowohl im direkten Zusammenhang zur Zielvariable, als auch in der zeitverzögerten Korrelation, keinerlei signifikanten und für die Modellierung brauchbaren Zusammenhänge identifiziert werden. Dies unterstreicht noch einmal die Aussage der fehlenden Saisonalität in der Zeitreihe.

Insgesamt wurden so im Rahmen des Feature Engineerings 50-60 möglicherweise relevante Features generiert.

<sup>5</sup> Verschiedene Feature-Engineering-Typen in Zeitreihen 2021.

<sup>6</sup> Verschiedene Feature-Engineering-Typen in Zeitreihen 2021.

## 2.2.4 Zeitreihenprognose mit dem ARIMA-Modell

Für die Durchführung einer Prognose mit dem ARIMA-Modell sind grundsätzlich keine speziellen Data Preprocessing-Schritte notwendig, da das Modell in seiner ursprünglichen Ausprägung ein univariates Prognosemodell ist. Zwar bestehen auch Möglichkeiten das ARIMA-Modell als multivariates Modell umzusetzen, was aber in diesem Rahmen zunächst nicht tiefergehend betrachtet werden soll.

Das ARIMA-Modell ist auf die Lösung von Zeitreihenproblemen spezialisiert. Es ist also in der Lage, abhängig von der Parametrisierung des Modells, relevante statistische und stochastische Lagen und Trends in den Daten einer Zeitreihe zu identifizieren und diese in die Zukunft zu extrapolieren.

Da die Parametrisierung des ARIMA-Modells, zur Erzielung von guten Vorhersageergebnissen, relevant ist, wurde ein Hyperparametertuning im Trainings- und Testprozess installiert. Dieses Hyperparametertuning stellt, im Rahmen des Trainings und der Evaluation des Modells, sicher, dass für die relevantesten Parameter des ARIMA-Modells jene Parameter identifiziert werden, die die beste Modellperformance im Rahmen der Prognose der Zielvariable auf den Testdaten, gemessen anhand von RMSE und MAE als ausgewählte Performanceparameter, erreichen. Das Hyperparametertuning erfolgte an dieser Stelle mithilfe der Grid-Search-Methode.

Der Prozess bzw. Workflowbestandteil für das Trainieren und Testen des ARIMA-Modells inkl. Hyperparametertuning soll in der folgenden Abbildung kurz dargestellt werden.

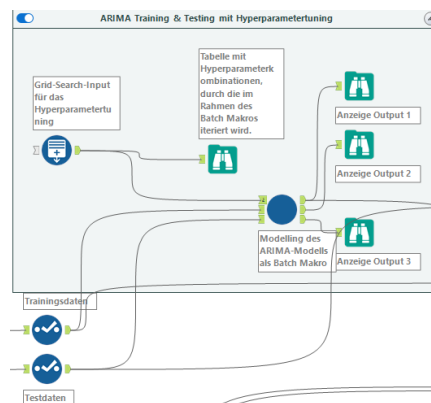


Abbildung 9: Workflow für das Trainieren und Testen des ARIMA-Modells

Die drei Output-Knoten geben letztlich die Ergebnisse des Modelltrainings auf den Trainingsdaten und dessen Evaluation auf den Testdaten aus. Konkret wird eine Übersicht der erzielten Performanceparameter (RMSE, MAE etc.) pro Iterationsdurchlauf der Hyperparameteroptimierung ausgegeben.

## 2.2.5 Zeitreihenprognose mit dem XGBoost

Bei der Verwendung von Entscheidungsbaummodellen, die in der Lage sind Regressionsprobleme zu lösen, sind einige Besonderheiten bei der Lösung von Zeitreihenproblemen zu beachten.

Die Entscheidungsbaummodelle, so auch der hier näher beschriebene XGBoost (Gradient Boosted Tree Model), betrachten, anders als das ARIMA-Modell, jede Beobachtung im Datensatz unabhängig voneinander.

Dies erfordert einige Feature-Engineering-Maßnahmen, die bereits näher im Kapitel „Feature Engineering“ beschrieben wurden, jedoch relativ speziell und wegen des Entscheidungsbaummodells durchgeführt werden mussten. Die im Rahmen des Feature Engineering berechneten Variablen dienen als Prädiktoren für den XGBoost. Bei den hinzugefügten Features handelt es sich i.d.R. um stochastische und statistische Lagemaße, analog zu den Möglichkeiten, die bereits im Feature-Engineering-Abschnitt genauer beschrieben wurden.

Zur Wahrung der Konsistenz der Zeitreihe wird im Rahmen des Modelltrainings, auch beim Entscheidungsbaum, auf Sampling-Methoden, wie eine k-fold crossvalidation, verzichtet.

Wichtige beeinflussbare Parameter beim Erstellen von Entscheidungsbaummodellen, die zu entsprechender Performancesteigerung führen können, sind die Interaktionstiefe, der Schrumpfungparameter und die minimale Anzahl der Beobachtungen pro Baumknoten. Selbstverständlich ist die Palette an Parametern, die eingestellt werden können, noch größer – erfahrungsgemäß führt die Optimierung der eben genannten Parameter allerdings zu den signifikantesten Verbesserungen in der Modellperformance.

An dieser Stelle sei beispielhaft ein Workflow für das Training des XGBoost-Modells aufgezeigt. Dieser beinhaltet ebenso das Hyperparametertuning für die eben erwähnten Parameter. Die Messung der Modellperformance bei unterschiedlichen Hyperparameterkombinationen erfolgt jeweils auf den Testdaten – also dem Modell unbekannte Daten.

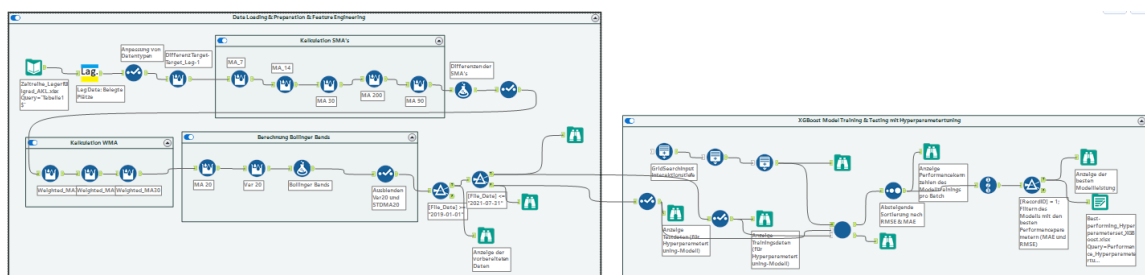


Abbildung 10: Feature Engineering inkl. Modelltraining mit Hyperparametertuning und Ausgabe der Performancemetriken auf den Testdaten

## 2.2.6 Zeitreihenprognose mit der Monte-Carlo-Simulation

Die Vorgehensweise der Monte-Carlo-Simulation als mathematische Methode, die mit Zufallszahlen bzw. Stichprobenziehungen versucht, einen Wertebereich für mögliche Prognoseergebnisse zu generieren, wurde eingangs bereits kurz beschrieben.

Im Rahmen dieser Arbeit wurde ein Monte-Carlo-Simulationsmodell erstellt, welches durch mehrfache Ausführung eine Vielzahl von Stichproben aus der Zielvariable des zugrundeliegenden Datensatzes zieht. Basierend auf der Verteilung der Zielvariable und den in den einzelnen Stichproben gezogenen Werten ergibt sich ein simulierter Prognosewert pro Iteration des Stichprobenmodells. Je höher die Anzahl der Iterationen (ergo: Wiederholung der Stichprobenziehung) desto besser lässt sich ein Wertebereich der möglichen zukünftigen Zielvariable als Prognose definieren.

Um eine Zielvariable mit einer annähernden Normalverteilung für die Simulation zu verwenden, wurde die Variable „PercentChange“ berechnet, die die relative Veränderung der Belegten Plätze zu ihrem Vortageswert berechnet.



Abbildung 11: Variable "PercentChange" als näherungsweise Normalverteilung

Der Alteryx-Workflow zur Umsetzung der Monte-Carlo-Simulation sei an dieser Stelle beispielhaft abgebildet:

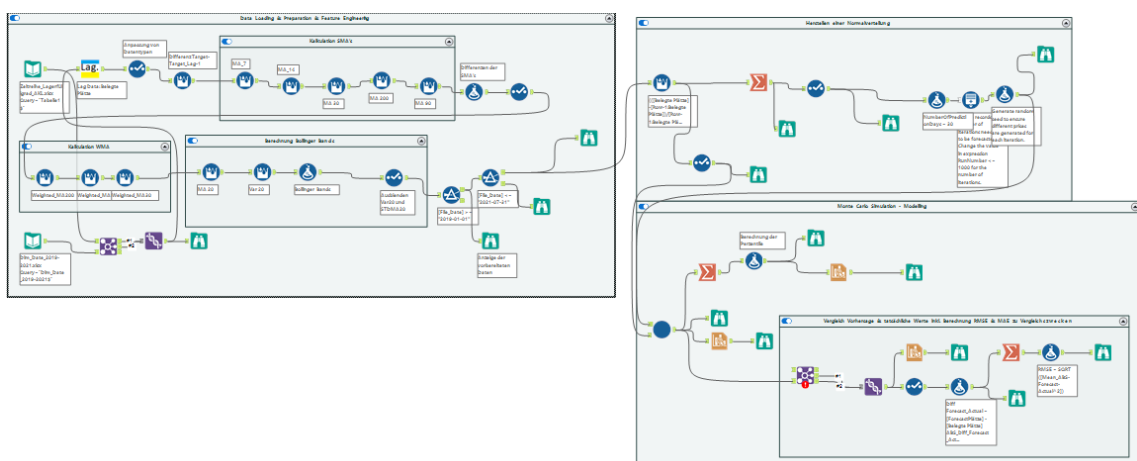


Abbildung 12: Data Preparation, Feature Engineering und Monte-Carlo-Simulation im Workflow

Die Monte-Carlo-Simulation wurde mithilfe eines Batch-Makros umgesetzt. Für die Ausführung des Workflows kann über die in den Workflow integrierten Steuerparameter die Anzahl der Stichprobenziehungen pro Iteration, ein „Random Seed“ und die totale Anzahl der Ausführungen definiert werden. Der „Random Seed“ stellt sicher, dass bei jeder Ausführung der Simulation andere Simulationsergebnisse aus der Stichprobenziehung resultieren.

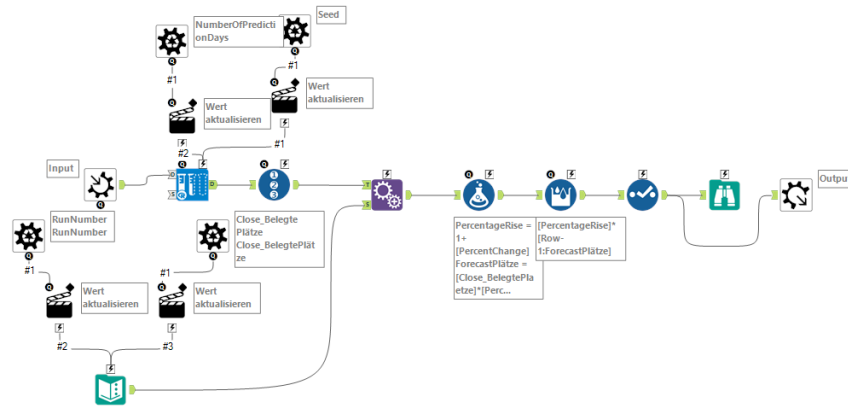


Abbildung 13: Batch-Makro mit Steuerungsparametern im Workflow der Monte-Carlo-Simulation

Im Rahmen der hier beispielhaft aufgebauten Stichprobenziehung (Monte-Carlo-Simulation) sollen zunächst beispielhaft 30 Tage der Zielvariable vorhergesagt werden. Die Stichprobenziehung wird demnach 30 Mal pro Iteration des Batch-Makros ausgeführt. Die Anzahl der Runs (Ausführungen) des Batch-Makros (z.B. 100) verursacht, dass 100 x 30 Vorhersageergebnisse aus der Stichprobenziehung generiert werden. Je höher die Anzahl der Runs des Batch-Makros, desto höher die Wahrscheinlichkeit, dass sich ein engerer Wertebereich für den tatsächlichen Wert der Zielvariable herauskristallisiert und sich auf diese Art und Weise Prognosewerte ableiten lassen.

## 3 Evaluation

### 3.1 ARIMA-Modell

Das ARIMA-Modell ist auf die Lösung von Zeitreihenproblemen spezialisiert.

Über die Parametrisierung des Modells wird insbesondere für nicht-stationäre Zeitreihen die für das ARIMA-Modell relevante Stationarität der Zeitreihe sichergestellt.

Im Folgenden sei eine Übersicht dargestellt, die die Performance des ARIMA-Modells bei optimaler Hyperparameterkombination anhand der Performancemetriken RMSE und MAE ausweist. Der RMSE beträgt 6.914 und der MAE beträgt 6.074.

Datensatz	Model	ME	RMSE	MAE	MPE	MAPE	MASE	d
1	ARIMA	-2452.8765	6914.9652	6079.4437	-1.5384	3.5453	6.5017	0

Abbildung 14: Performance des ARIMA-Modells bei bester Hyperparameterkombination

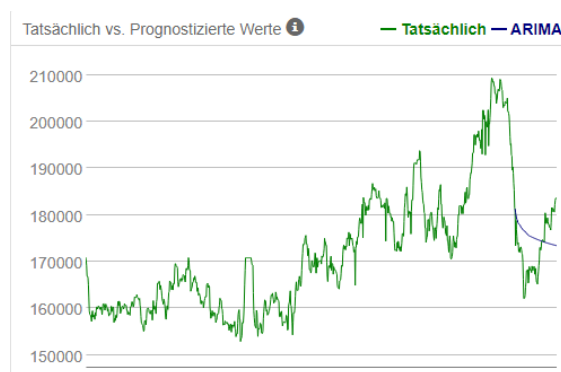


Abbildung 15: Visualisierung der ARIMA-Vorhersage auf Testdaten

### 3.2 Monte-Carlo-Simulation

Die Monte Carlo Simulation ist in diesem Zusammenhang eine gänzlich andere und eine eventuell nicht vollständig mit den im Rahmen dieser Arbeit verwendeten Modellen vergleichbare Methode. Das Ergebnis der Monte-Carlo-Simulation ist eine Vielzahl möglicher (simulierter) Ausprägungen der Zielvariable.

Im Folgenden sei abgebildet, wie sich die Leistung der Monte-Carlo-Simulation darstellt.

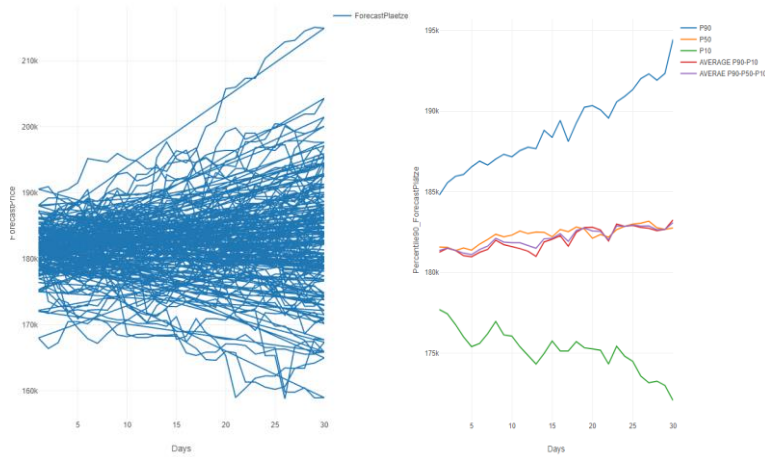


Abbildung 16: Abbildung der Simulationsergebnisse der Monte-Carlo-Simulation

Die im Folgenden dargestellten Performancemetriken RMSE (hier: 13.958) und MAE (hier: 11.557) wurden zu Vergleichszwecken selbst berechnet und beruhen auf dem Mittelwert der Simulationsergebnisse nach einer Vielzahl von Iterationen.

Datensatz	Mean_Forecast-Actual	Mean_ABS_Forecast-Actual (MAE)	Mean_ABS-Forecast-Actual^2	RMSE
1	11371.792	11557.558667	194829950.525333	13958

Abbildung 17: Performanceübersicht der Monte-Carlo-Simulationsergebnisse

### 3.3 Entscheidungsbaummodell „XGBoost“

Beim XGBoost wird im Rahmen des Modelltrainings eine Vielzahl unterschiedlich komplexer Entscheidungsbäume erstellt und zu einem gemeinsamen Modell kombiniert.

Das XGBoost-Modell weist mit einem RMSE von 1.666 und einem MAE von 1.147 die beste Performance auf den Testdaten aus und zeichnet sich als das beste Modell aus.

Datensatz	RecordID	Model	Correlation	RMSE	MAE	MPE	MAPE	Beobachtungen	Interaktionstiefe	Schrumpfungswert
1	1	XGBoost	0.95843	1666.015064	1147.159455	0.204068	0.663364	5	4	0.004

Abbildung 18: Performance des XGBoost bei bester Hyperparameterkombination

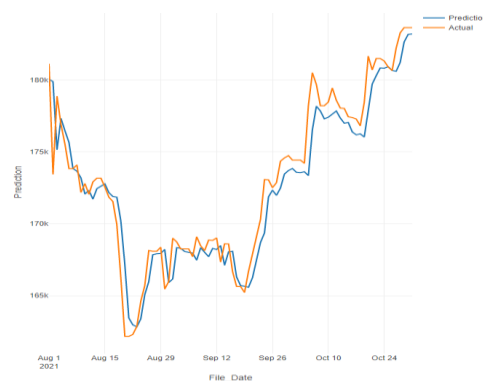


Abbildung 19: Visualisierung der XGBoost-Prognosen auf den Testdaten



## 4 Zusammenfassung

### 4.1 Iterative Prognose mit bestem Modell

Das Modell mit der besten Leistung für den hier dargestellten Use-Case ist unumstrittener Weise das Entscheidungsbaummodell „XGBoost“ mit einem RMSE von 1.666 und einem MAE von 1.147 bei optimaler Hyperparameterkombination.

Für das Entscheidungsbaummodell wurde zu guter Letzt ein iterativer Prognoseprozess (auch bekannt als „Walk-Forward-Prediction“) als weiterer Workflow aufgebaut. Auf diese Weise ist sichergestellt, dass bei jeder Vorhersageiteration alle relevanten Features des Datensatzes (z.B. auch bei der Vorhersage des 30. Tages) vollständig zur Verfügung stehen.

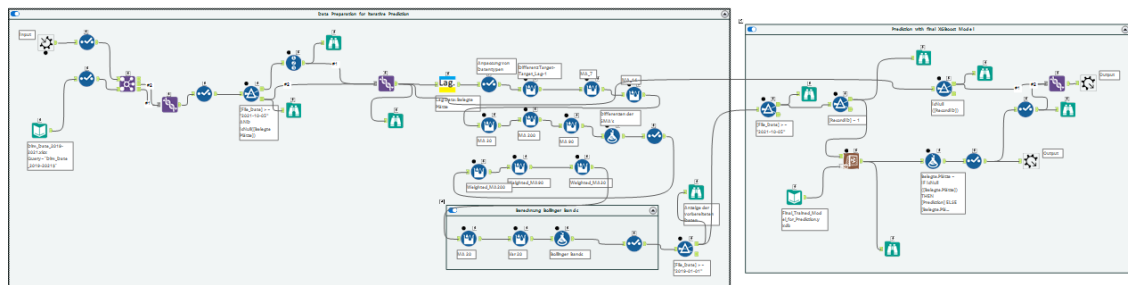


Abbildung 20: Iterativer Prognoseprozess (Walk-Forward-Prediction) für den XGBoost

Im Rahmen des iterativen Prozesses wird zunächst, mithilfe der im Datensatz enthaltenen Features, eine Vorhersage für den Zeitpunkt  $t+1$  vorgenommen. Das daraus resultierende Vorhersageergebnis wird iterativ an den Input des Prozesses zurückgegeben (Loop-Back-Mechanismus), um alle erforderlichen Features für die Vorhersage des Zeitpunkts  $t+2$  neu zu berechnen. Anschließend wird dieser Zeitpunkt vorhergesagt. Der Prozess wiederholt sich bis die eingestellte Iterationshäufigkeit (Anzahl zu prognostizierender Tage) erreicht ist.

### 4.2 Ausblick

Für die Integration des Modells in die betrieblichen Prozesse wird zunächst das Deployment, also die Automatisierung und Operationalisierung der ML-Workflows, angestrebt.

Die erste Phase nach dem Deployment wird eine Form der Validierungsphase darstellen, in der die tatsächliche Performance des Modells und dessen Prognosen engmaschig überwacht werden muss, um ggf. weitere Anpassungen am Modell vornehmen zu können. Die gute Performance (anhand RMSE und MAE) auf den Testdaten muss nicht zwangsläufig auch auf die Realität und die Prognosen zukünftiger Daten zutreffen. So ist es möglich, dass innerhalb der Testdaten ein BIAS vorliegt oder Informationen in den Testdaten enthalten sind, die für die Prognosen in der Zukunft nicht in gleicher Qualität

---

zur Verfügung stehen, was zum aktuellen Zeitpunkt eventuell nicht direkt ersichtlich ist. Die gute Anpassung des XGBoost-Modells an die vorliegenden Testdaten kann ein Warnzeichen für ein solches BIAS sein – muss es jedoch nicht, da der XGBoost üblicherweise durch seine Art und Weise des Modelltrainings und über seine Parametrisierung bereits gut gegen Overfitting robustifiziert wird.

Es ist denkbar und nach einer ersten Validierungsphase angestrebt, das Modell bei Bedarf, um domänenspezifische Features oder weitere Feature-Engineering-Aktivitäten zu erweitern, um die reale Prognosequalität weiter zu verbessern.

Außerdem gilt es, im Rahmen des Deployments, zu definieren, wie häufig und regelmäßig die einzelnen aufgebauten Workflows auszuführen sind. Insbesondere gilt es geeignete Retraining-Zyklen zu definieren, um die Modellperformance auch langfristig auf einem hohen Niveau halten zu können.

---

## Literaturverzeichnis

IBM Cloud Education (2021): Was ist die Monte-Carlo-Simulation? Online verfügbar unter <https://www.ibm.com/de-de/cloud/learn/monte-carlo-simulation>, zuletzt aktualisiert am 17.10.2021, zuletzt geprüft am 17.10.2021.

Luber, Stefan (2021): Was ist das ARIMA-Modell? Online verfügbar unter <https://www.bigdata-insider.de/was-ist-das-arima-modell-a-914956/>, zuletzt aktualisiert am 16.10.2021, zuletzt geprüft am 16.10.2021.

Verschiedene Feature-Engineering-Typen in Zeitreihen (2021), 08.06.2021. Online verfügbar unter <https://ichi.pro/de/verschiedene-feature-engineering-typen-in-zeitreihen-82063455350476>, zuletzt geprüft am 29.10.2021.