# Unternehmensbeschreibung

Im Fokus dieser Fallstudie steht das fiktive Unternehmen "StayAwhile Hotels". StayAwhile Hotels ist eine international agierende Hotelkette mit Hauptsitz in Bonn, Deutschland. Mit über 500 Hotels in 50 Ländern bietet StayAwhile Hotels eine Vielzahl von Dienstleistungen an, darunter Unterkünfte, Catering, Veranstaltungsplanung und Reisedienstleistungen.

StayAwhile Hotels hat sich einen Namen gemacht durch seinen Fokus auf Kundenzufriedenheit und Qualitätsservice. Das Unternehmen hat eine starke Marke aufgebaut, die auf den Prinzipien von Gastfreundschaft, Integrität und Exzellenz basiert. Trotz ihres Erfolgs in der Branche sieht sich StayAwhile Hotels mit einer Reihe von Herausforderungen konfrontiert, insbesondere im Zusammenhang mit Stornierungen.

Im Rahmen ihrer digitalen Transformation hat StayAwhile Hotels erkannt, dass Technologie und Datenanalyse Schlüsselkomponenten zur Bewältigung dieser Herausforderungen sind. Das Unternehmen hat daher in die Entwicklung von Machine-Learning-Modellen investiert, um seine Betriebsabläufe zu optimieren und eine datengesteuerte Entscheidungsfindung zu ermöglichen.

# Problem und Fragestellung

Stornierungen sind in der Hotelbranche ein weit verbreitetes Problem, das erhebliche Auswirkungen auf den Betrieb und die Rentabilität eines Unternehmens haben kann. Für StayAwhile Hotels ist das Problem der Stornierungen besonders akut. Unvorhergesehene Stornierungen führen zu unbesetzten Zimmern, die wiederum zu verlorenen Einnahmen führen. Darüber hinaus können sie die Planung und Ressourcenallokation des Unternehmens erheblich beeinträchtigen.

Laut einer Studie von D-EDGE Hospitality Solutions aus dem Jahr 2019 lag die durchschnittliche Stornierungsrate in der europäischen Hotellerie bei etwa 40%. Diese hohe Rate unterstreicht die Dringlichkeit des Problems und die Notwendigkeit, effektive Lösungen zu finden.

Die zentrale Fragestellung dieser Fallstudie ist daher: Wie kann StayAwhile Hotels mithilfe von Machine Learning die Wahrscheinlichkeit von Stornierungen vorhersagen und so ihre Betriebsplanung und Ressourcenallokation verbessern?

Um diese Frage zu beantworten, wurde ein Machine-Learning-Modell entwickelt, das auf historischen Buchungsdaten basiert und in der Lage ist, die Wahrscheinlichkeit zukünftiger Stornierungen vorherzusagen. Durch die Verbesserung der Vorhersagegenauigkeit von Stornierungen kann StayAwhile Hotels seine Betriebsplanung optimieren, seine Ressourcen effizienter zuweisen und letztendlich seine Rentabilität steigern.

# Einordnung in den Bereich Machine Learning

Machine Learning (ML) ist ein Teilbereich der künstlichen Intelligenz, der sich darauf konzentriert, Computern das "Lernen" aus Daten zu ermöglichen, um Muster zu erkennen und Vorhersagen zu treffen. Es hat sich als ein mächtiges Werkzeug in vielen Branchen erwiesen, einschließlich der Hotellerie.

In der Hotellerie kann ML dazu beitragen, eine Vielzahl von Herausforderungen zu bewältigen, von der Verbesserung der Kundenzufriedenheit bis hin zur Optimierung der Betriebsabläufe. Laut einer Studie von McKinsey aus dem Jahr 2020 nutzen bereits 47% der Unternehmen in der Reise- und Logistikbranche ML in irgendeiner Form, und weitere 21% planen, es in den nächsten drei Jahren einzuführen.

In dieser Fallstudie wird ML verwendet, um das Problem der Stornierungen bei StayAwhile Hotels anzugehen. Das entwickelte Modell ist ein Beispiel für überwachtes Lernen, eine Art von ML, bei der ein Modell aus einem Satz von Eingabe- und Ausgabedaten "lernt". In diesem Fall sind die Eingabedaten die historischen Buchungsdaten und die Ausgabedaten sind, ob eine Buchung storniert wurde oder nicht.

Das Modell verwendet verschiedene Merkmale aus den Buchungsdaten, wie z.B. den Buchungszeitpunkt, den Aufenthaltszeitraum und den Buchungstyp, um Muster zu erkennen und Vorhersagen über zukünftige Stornierungen zu treffen. Durch die Verbesserung der Vorhersagegenauigkeit von Stornierungen kann StayAwhile Hotels seine Betriebsplanung optimieren, seine Ressourcen effizienter zuweisen und letztendlich seine Rentabilität steigern.

Zum Angehen des Problems wird sich auf das Framework des CRISP- DM Models bezogen. Hierbei werden Datenanalyse Probleme strukturiert angegangen und können einer Logik folgen. Umso das Businessproblem zu lösen. Das Business Understanding wurde in den ersten beiden Kapiteln behandelt. Zum Data Understanding wurde Jupyter Notebooks hinzugezogen, aus welchem auch Code Snippets mit eingebracht werden.

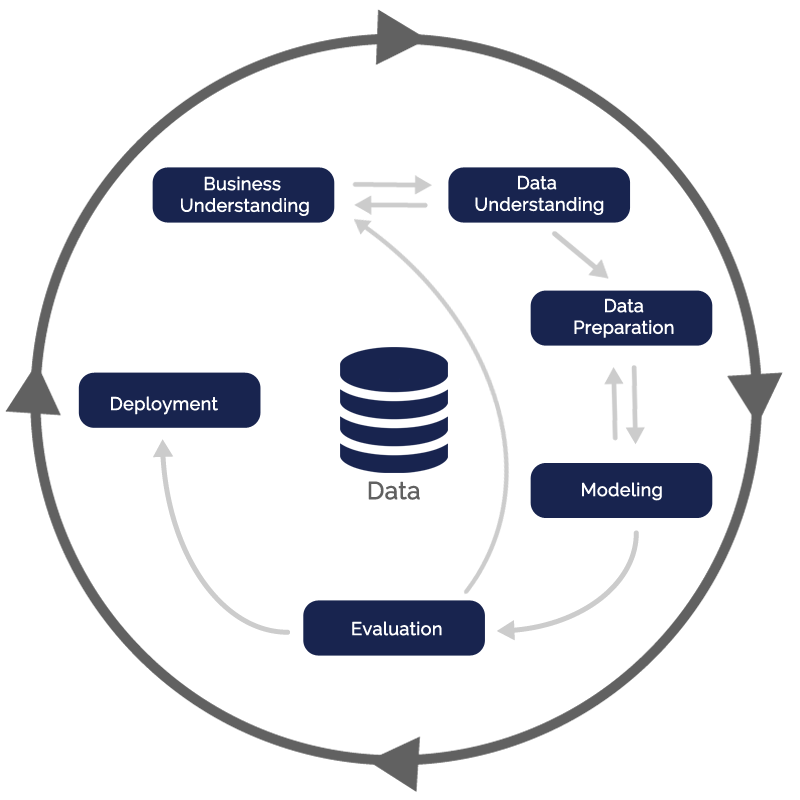


Abbildung 1: Crisp DM Model (https://www.otaris.de/beratung-data-science/)

# Data Understanding

Der vorliegende Abschnitt konzentriert sich auf die Analyse und Interpretation des Datensatzes mit dem Ziel, ein tieferes Verständnis der Daten zu erlangen und wertvolle Erkenntnisse zu gewinnen. Die Daten wurden in einem Jupyter Notebook mit Python analysiert, wobei verschiedene Bibliotheken für Datenanalyse und maschinelles Lernen verwendet wurden.

Der analysierte Datensatz enthält Informationen über verschiedene Aspekte von Hotelreservierungen, darunter die Anzahl der Erwachsenen und Kinder pro Buchung, die Anzahl der Wochenend- und Wochentage pro Buchung, der Typ des gebuchten Mahlzeitplans, ob ein Parkplatz benötigt wird oder nicht, der Typ des reservierten Zimmers und die Zeitspanne zwischen der Reservierung und dem Ankunftsdatum. Jede Reservierung ist durch eine eindeutige Buchungs-ID gekennzeichnet.

Eine erste Untersuchung des Datensatzes ergab, dass es keine Duplikate oder Nullwerte gibt, was auf eine hohe Datenqualität hindeutet. Eine detaillierte statistische Analyse zeigte, dass die meisten Buchungen von ein oder zwei Erwachsenen gemacht werden und dass die meisten Buchungen keine Kinder beinhalten. Darüber hinaus beinhalten viele Buchungen sowohl Wochenendnächte als auch Wochentage, wobei die Anzahl variiert. Interessanterweise benötigen die meisten Buchungen keinen Parkplatz.

Eine Untersuchung der Verteilung der Buchungsstatus ergab, dass etwa 30% der Buchungen storniert werden. Diese Information ist besonders relevant, da sie einen Einblick in das Stornierungsverhalten der Kunden gibt und als Grundlage für weitere Analysen und Vorhersagemodelle dienen kann.

Ein Bild, das Text, Screenshot, Kreis, Diagramm enthält.

Automatisch generierte Beschreibung

Die Verteilung der Zielvariable in einem Datensatz, insbesondere in einem Klassifikationsproblem, hat erhebliche Auswirkungen auf die Modellierung und die Interpretation der Ergebnisse. In diesem Fall ist die Zielvariable der Buchungsstatus, der zwei Klassen hat: "storniert" und "nicht storniert". Die Verteilung dieser Klassen ist etwa 30% zu 70%. Diese Verteilung bedeutet, dass unser Datensatz leicht unausgewogen ist, da die "nicht stornierten" Buchungen etwa 70% der Daten ausmachen, während die "stornierten" Buchungen nur etwa 30% ausmachen. Dies ist jedoch nicht unausgewogen; in der Praxis wird eine Verteilung, die näher an 50/50 liegt, oft als ideal angesehen, aber eine 70/30-Verteilung ist immer noch handhabbar. Die Auswirkungen dieser Verteilung auf die Modellierung können vielfältig sein. Ein Modell könnte dazu neigen, die Mehrheitsklasse ("nicht storniert") zu bevorzugen, da es durch die Vorhersage der Mehrheitsklasse eine höhere Genauigkeit erzielen kann. Dies könnte dazu führen, dass das Modell weniger effektiv bei der Vorhersage der Minderheitsklasse ("storniert") ist.

# Untersuchen der Numerischen Features

Unser Datensatz enthält eine Vielzahl von numerischen Merkmalen, die wichtige Informationen über die Hotelreservierungen liefern. Diese Merkmale umfassen:

- no\_of\_adults: Die Anzahl der Erwachsenen pro Buchung.

- no\_of\_children: Die Anzahl der Kinder pro Buchung.

- no\_of\_weekend\_nights: Die Anzahl der Wochenendnächte pro Buchung.

- no\_of\_week\_nights: Die Anzahl der Wochentage pro Buchung.

- lead\_time: Die Zeitspanne zwischen der Reservierung und dem Ankunftsdatum.

- no\_of\_previous\_cancellations: Die Anzahl der vorherigen Stornierungen des Gastes.

- no\_of\_previous\_bookings\_not\_canceled: Die Anzahl der vorherigen Buchungen des Gastes, die nicht storniert wurden.

- avg\_price\_per\_room: Der durchschnittliche Preis pro Zimmer.

- no\_of\_special\_requests: Die Anzahl der speziellen Anfragen des Gastes.

Darüber hinaus gibt es zwei Merkmale, die numerische Werte annehmen, aber tatsächlich kategoriale Daten darstellen:

- required\_car\_parking\_space: Gibt an, ob ein Parkplatz benötigt wird oder nicht.

- repeated\_guest: Gibt an, ob der Gast wiederholt hat oder nicht.

Obwohl diese Merkmale numerische Werte annehmen (0 und 1), sind sie kategorial, weil sie eine bestimmte Kategorie oder Gruppe darstellen, nicht eine quantitative Messung. Sie sind binär, weil sie nur zwei mögliche Werte haben.

Für alle numerischen Features wurde ein Histogramm erstellt, um diese besser bewerten zu können und eventuelle Auffälligkeiten zu erkennen. Dafür wurde folgender Code verwendet:  
Ein Bild, das Text, Schrift, Software, Screenshot enthält.

Automatisch generierte Beschreibung

Das Histogramm für lead\_time zeigt einen positiven Skew, was darauf hindeutet, dass die meisten Buchungen spontan getätigt werden. Ob hier Ausreiser entfernt werden sollten, wird nächsten Schritt geklärt, wenn ein Vergleich der Werte für storniert und nicht storniert berücksichtigt werden.

Ein Bild, das Text, Screenshot, Reihe, Diagramm enthält.

Automatisch generierte Beschreibung

Das Histogramm für avg\_price\_per\_room zeigt auch einen leichten positiven Skew, dieses ist aber deutlich normalverteilter. Auffällig ist, dass es auch Zimmer umsonst gibt. Hier ist die Frage, ob es fehlerhafte Daten sind oder gegebenenfalls per Gutschein gebuchte Zimmer sind. Ob hier eine Anpassung erfolgen soll, wird auch im nächsten Schritt geklärt. Ein Bild, das Text, Screenshot, Diagramm, Design enthält.

Automatisch generierte Beschreibung

Das Histogramm für `arrival\_month` zeigt, dass die Anzahl der Buchungen je nach Monat variiert. Es gibt einige Monate mit einer höheren Anzahl von Buchungen, was auf die Hochsaison hinweisen könnte, und andere Monate mit einer niedrigeren Anzahl von Buchungen, was auf die Nebensaison hinweisen könnte. Generell ist auffällig, dass sich die Buchungen in den Winter Monaten häufen. Ein Bild, das Text, Diagramm, Reihe, Screenshot enthält.

Automatisch generierte Beschreibung  
  
Es ist zusätzlich noch zu erkennen, dass es Datensätze gibt die eine Anzahl von 0 Erwachsenen Person haben, im weiteren wird untersucht ob diese Erkenntnis besagt, dass die Datensätze falsch sind oder eventuell nur Kinder gebucht haben, das wird im Schritt des Feature Engineerings geklärt. Weitere Verteilungen sind im Notebook zu finden.

# Untersuchung der Numerischen Variablen zu der Zielvariablen

Um die Verteilung in Bezug zur Zielvariablen auszugeben, wurde folgendes Code Snippet benutzt.  
Ein Bild, das Text, Screenshot, Display, Software enthält.

Automatisch generierte Beschreibung

Um die komplette Abbildung zu sehen, muss im Jupyter Notebook die Zelle 19 ausgeführt werden.

Die interessantesten Beobachtungen, sind, dass je kürzer die lead time, desto eher werden die Buchungen nicht storniert, weshalb auch zunächst keine Ausreißer entfernt werden.

Ein Bild, das Text, Diagramm, Screenshot, Reihe enthält.

Automatisch generierte Beschreibung

Zudem fällt beim Preis auf, dass die Verteilung für storniert etwas nach rechts verschoben ist. Inklusive der wichtigsten Erkenntnis, dass Buchungen für 0 € kaum bzw. gar nicht storniert werden. Diese Information ist wichtig und soll dem Modell auch mitgegeben werden, also werden die Daten nicht entfernt.

Ein Bild, das Text, Screenshot, Diagramm enthält.

Automatisch generierte Beschreibung

Um Korrelationen zu finden die dem Modell Probleme bereiten könnten, wird eine Korrelationsmatrix erstellt. Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Es ist zu erkennen dass Korrelationen zwischen repeated\_guest, no\_of\_previous\_cancellations und no\_of\_previous\_bookings\_not\_canceled vorliegt.  
Das macht auch Sinn, da diese alle in Abhängigkeit dazu sind ob schonmal vorher gebucht worden ist. Der avg\_price\_per\_room korreliert zu dem mit der Anzahl der Personen, was auch Sinn ist da größere Räume teurer sein müssten.   
Ansonsten fällt noch auf, wie schon vorher entdeckt, dass die lead\_time negativ mit dem booking\_status korreliert und somit der wichtigste Wert für unser Model ist.

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

# Untersuchung der kategorialen Features

In dem Abschnitt wurden alle Features untersucht welche Kategorial sind. Wichtige Erkenntnisse, die hier aufgefallen sind, sind das Wiederkehrende Gäste, Gäste die einen Parkplatz buchen und geschäftliche Buchungen tendenziell weniger storniert werden.

# Data Preparation

# Ausreiser entfernen

Anfangs sollen keine Ausreiser entfernt werden, dies kann bei Bedarf nach der Evaluation erfolgen. Dazu entschieden wurde sich, da es keine eindeutigen Ausreißer gab und auch die Ausreißer einen Einfluss auf das Buchungsverhalten haben.

# Feature Engineering

Als Feature wurde, das Jahr und der Tag zunächst entfernt. Diese Informationen sind für das Model nicht aussagekräftig. Das Feature Monat wurde behalten, da dieses Saisonalität mit einbehaltet.

Es wurden zwei neue Features erstellt und einige gelöscht. Aus den Features no\_of\_weekend\_nights und no\_of\_week\_nights wurde das Feature total\_stay.  
Und aus no\_of\_adults und no\_of\_children wurde total\_guests.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Total\_guests weißt in diesem Zuge dann auch keine 0 Werte auf, was das Problem mit no\_of\_adults behebt. Denn vermutlich können auch Zimmer für Kinder gebucht werden und nicht nur für Erwachsene.

# Kategoriale Variablen auswertbar machen

Um das Problem der kategorialen Variablen anzugehen, wird One-Hot-Encoding verwendet.

One-Hot-Encoding ist ein weit verbreitetes Verfahren zur Transformation kategorialer Variablen in eine Form, die für maschinelles Lernen und statistische Modelle geeignet ist. Kategoriale Variablen sind typischerweise diskret und repräsentieren verschiedene Klassen oder Kategorien, die keine numerische Beziehung zueinander haben. Daher ist es oft nicht sinnvoll, diese Variablen direkt in numerischer Form zu verwenden, da dies zu irreführenden oder falschen Interpretationen führen kann.

One-Hot-Encoding adressiert dieses Problem, indem es jede Kategorie einer kategorialen Variable in eine separate binäre Variable umwandelt. Jede dieser binären Variablen nimmt den Wert 1 an, wenn die ursprüngliche Variable die entsprechende Kategorie aufweist, und 0 sonst. Auf diese Weise wird jede Kategorie durch eine eindeutige binäre Variable repräsentiert, und es wird keine irreführende numerische Beziehung zwischen den Kategorien impliziert.

One-Hot-Encoding wird für die Variablen: type\_of\_meal\_plan, room\_type\_reserved, market\_segment\_type angewendet.

# 5.4 Trainings und Testdatensätze erstellen und Standardisieren

Der Train- Test Split, ist ein grundlegender Schritt in der Vorbereitung von Daten für maschinelles Lernen und statistische Modellierung. Der Hauptzweck dieser Aufteilung besteht darin, die Fähigkeit des Modells zur Generalisierung auf neue, unbekannte Daten zu bewerten.

Der Trainingsdatensatz wird verwendet, um das Modell zu trainieren, d.h., die Modellparameter so anzupassen, dass sie die Beziehungen in den Trainingsdaten so gut wie möglich abbilden. Der Testdatensatz wird dann verwendet, um die Leistung des trainierten Modells zu bewerten. Da das Modell während des Trainingsprozesses keinen Zugang zu den Testdaten hat, bietet der Testdatensatz eine unvoreingenommene Messung der Fähigkeit des Modells, Vorhersagen für neue Daten zu treffen.

In dem vorliegenden Fall des Hotelbuchungsdatensatzes wird ein Modell trainiert, um den booking\_status auf der Grundlage der anderen Variablen im Datensatz vorherzusagen. Der Trainingsdatensatz wird verwendet, um dieses Modell zu trainieren, und der Testdatensatz wird verwendet, um zu bewerten, wie gut das Modell in der Lage ist, den booking\_status für neue Buchungen vorherzusagen.

Die Standardisierung der Daten ist ein weiterer wichtiger Schritt in der Datenvorbereitung. Viele maschinelle Lernalgorithmen arbeiten besser, wenn die Eingabevariablen auf einer ähnlichen Skala liegen und eine ähnliche Verteilung aufweisen. Die Standardisierung erreicht dies, indem sie jede Variable so transformiert, dass sie einen Mittelwert von 0 und eine Standardabweichung von 1 hat.

Die Standardisierung kann besonders wichtig sein, wenn die Eingabevariablen sehr unterschiedliche Skalen oder Einheiten haben. Im Hotelbuchungsdatensatz haben Variablen wie lead\_time (gemessen in Tagen) und no\_of\_adults (gemessen in der Anzahl der Personen) sehr unterschiedliche Skalen und Einheiten. Die Standardisierung dieser Variablen wird sicherstellen, dass sie auf einer ähnlichen Skala liegen und dass keine Variable aufgrund ihrer Skala oder Einheiten einen unverhältnismäßig großen Einfluss auf das Modell hat.

# Modeling

# Evaluation

# Deployement

# Datenbasierte Services

# Relevante Effekte für StayAwhile

# Datensatz und ML- Modell

Die Daten, welche zur Entwicklung des Modells benutzt, werden kommen aus dem Dataset [Hotel Reservations Dataset](https://www.kaggle.com/datasets/ahsan81/hotel-reservations-classification-dataset), auf Kaggle. Das Dataset ist aus 19 Features und 36.000 Datensätzen aufgebaut. Zu jeder stornierten Buchung kommen 2 nicht stornierte Buchungen. Also ist der Datensatz relativ ausbalanciert.  
  
**Feature 1:** Booking\_ID  
dient nur der Zuordnung zu einem Datensatz und ist nicht relevant für das Modell

**Feature 2-18:** Nominal: Booking\_ID, type\_of\_meal\_plan, required\_car\_parking\_space, room\_type\_reserved, market\_segment\_type, repeated\_guest, booking\_status

Kardinal: no\_of\_adults, no\_of\_children, no\_of\_weekend\_nights, no\_of\_week\_nights, lead\_time, no\_of\_previous\_cancellations, no\_of\_previous\_bookings\_not\_canceled, avg\_price\_per\_room, no\_of\_special\_requests

Ordinal: arrival\_year, arrival\_month, arrival\_date

**Feature 19:** Booking\_status – dieses Feature ist die Zielvariable und wird in binär angegeben.

# Methodik

Für die Vorhersage von Stornierungen im Hotelgewerbe bieten sich verschiedene Algorithmen an, darunter Entscheidungsbäume, Random Forest, XGBoost und K-Nearest Neighbors (KNN).

Entscheidungsbäume ermöglichen die Erstellung von Entscheidungsregeln basierend auf den Merkmalen der Buchungsdaten. Durch die Kombination mehrerer Entscheidungsbäume im Random Forest-Algorithmus können robuste und genaue Vorhersagen erzielt werden. Random Forest berücksichtigt die Vorhersagen aller Bäume und aggregiert sie zu einem kohärenten Ergebnis.

XGBoost ist eine Weiterentwicklung des Gradient Boosting-Algorithmus, der speziell auf Effizienz und Genauigkeit ausgelegt ist. Es verwendet Entscheidungsbäume und optimiert die Modellvorhersagen durch das Minimieren einer Kostenfunktion. XGBoost bietet erweiterte Funktionen wie Regularisierung und paralleles Training, um präzise Vorhersagen zu erzielen.

K-Nearest Neighbors (KNN) basiert auf der Idee, dass ähnliche Beispiele tendenziell die gleiche Klassenzugehörigkeit haben. KNN sucht nach den k nächsten Nachbarn eines Datenpunkts und bestimmt die Mehrheitsklasse dieser Nachbarn als Vorhersage. KNN ist besonders nützlich, um Muster und Ähnlichkeiten in den Buchungsdaten zu erkennen und Vorhersagen zu treffen.

# Ausblick

Die erfolgreiche Implementierung eines prädiktiven Modells zur Vorhersage von Stornierungen im Hotelgewerbe bietet vielfältige Auswirkungen und Möglichkeiten für das Unternehmen. Durch die präzise Vorhersage von Stornierungen kann das Unternehmen seine Betriebsplanung optimieren, die Kapazitätsauslastung verbessern und finanzielle Verluste durch leere Zimmer minimieren.

# Quellen

D-EDGE Hospitality Solutions, "European Hotel Distribution Study: The Rise of Online Travel Agents", Zugriff am 2. Juli 2023, <https://www.d-edge.com/download-european-hotel-distribution-study-the-rise-of-online-travel-agents/>

McKinsey, "The state of AI in 2020", Zugriff am 2. Juli 2023, https://www.mckinsey.com/featured-insights/artificial-intelligence/the-state-of-ai-in-2020