

CONVERSIONS VORHERSAGEN MIT PYTHON/GOOGLE COLABORATORY

TOBIAS AUBELE & SEBASTIAN SCHMIDT

DER AUTOR



Tobias Aubele ist Professor für E-Commerce, insbesondere Conversion-Optimierung und Usability an der Hochschule Würzburg-Schweinfurt. Darüber hinaus berät er Unternehmen im Bereich Webanalytics & Website-Optimierung.

In den letzten Ausgaben der Website Boosting wurde intensiv auf die umfassenden Analyse-möglichkeiten mit R eingegangen. Neben Kenntnissen in R wird in sehr vielen Stellenanzeigen auf wünschenswerte Erfahrung mit Python verwiesen. Python ist ebenfalls eine Programmiersprache, in der mit wenigen Programmzeilen mächtige Analysen durchgeführt werden können. Google bietet mit Colaboratory eine kostenlose Entwicklungsumgebung in der Cloud an, welche einen unkomplizierten Beginn in Python ermöglicht. Was auf den ersten Blick sehr technisch und aufwendig erscheint, ist im Detail kein Hexenwerk. Dieser Artikel soll für Sie ein Trigger für den Einstieg in die Prognosemöglichkeiten mit Python sein. Erweitern Sie Ihr Skill-Set und nutzen Sie die umfassenden Möglichkeiten von Python in Kombination mit Google und Jupyter Notebook.

Wie immer gilt bei solchen Themen: Einfach hinsetzen und Schritt-für-Schritt nachmachen erhöht das Verständnis enorm und zeigt, wie leicht solche Dinge eigentlich zu machen bzw. zu verstehen sind. Beim bloßen Durchlesen bleibt deutlich weniger hängen und alles wirkt oft viel komplizierter, als es in der Realität ist. Also einfach mal ausprobieren? Nur Mut!

DER AUTOR



Sebastian Schmidt ist Dipl.-Ing. (FH) im Maschinenbau und B. Sc. in E-Commerce. Er verknüpft beide Welten mithilfe physikalischer Simulationen und Datenprognosen in der Optimierung von Windkraftanlagen.

Eine gute und präzise Prognose ist ein großer Wettbewerbsvorteil. Vorhandene Ressourcen können bestmöglich eingesetzt und Kosten bzw. Erträge optimiert werden. Wenn heute schon bekannt wäre, wie nächste Woche die Besucher und Bestellungen eintreffen, könnte die richtige Anzahl an Personal im Kundenservice und Logistikbereich bereitgestellt werden. Keine Wartezeiten im Call-Center, unverzüglicher Versand der Ware und schnelle Webseiten durch entsprechende Ressourcenplanung sind die besten Voraussetzungen für eine gute User-Experience. Doch kein Tag ist wie der andere, die Besucherfrequenzen am Wochenende sind typischerweise unterschiedlich und innerhalb der Tage sind bestimmte Zeiten stärker im Fokus als andere. Eine lineare Regression oder ein gleitender Durchschnitt sind probate Mittel der Prognose, haben jedoch eine entsprechende Streuung in den Daten.

Colaboratory von Google – kostenlos und flexibel

Die Programmiersprache Python wurde Anfang der 90er entwickelt und hat insbesondere in den vergangenen fünf Jahren einen wahren Boom in ihrer Verbreitung erlebt. In den wichtigen Programmiersprachen-Rankings von TIOBE und IEEE ist Python regelmäßig auf den vorderen Plätzen zu finden. Wichtige Gründe dafür sind die Beliebtheit von Python in der Datenanalyse und der Open-Source-Charakter vieler Entwicklungswerkzeuge.

Nutzer von Python haben Zugriff auf eine Vielzahl von Paketen, die mit ihren leistungsfähigen Funktionen in der Datenanalyse und Prognose unterstützen. Auch Facebook, dessen Geschäftsmodell stark auf der Prognose von Nutzerverhalten basiert, hat einige sehr nützliche Pakete für Python entwickelt. Die darin enthaltenen Algorithmen, die Offenheit der Programmiersprache Python und die (kostenlose) Rechenleistung von Google sind in Colaboratory

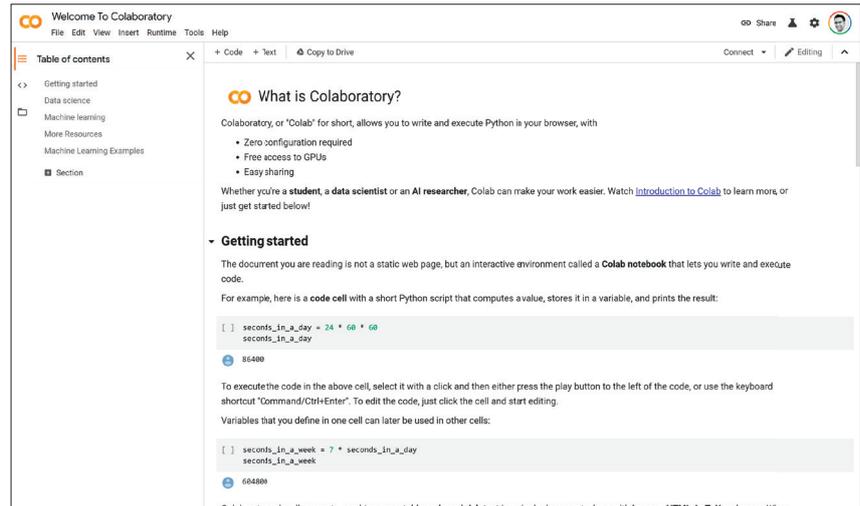


Abb. 1: Entwicklungsumgebung Colaboratory von Google

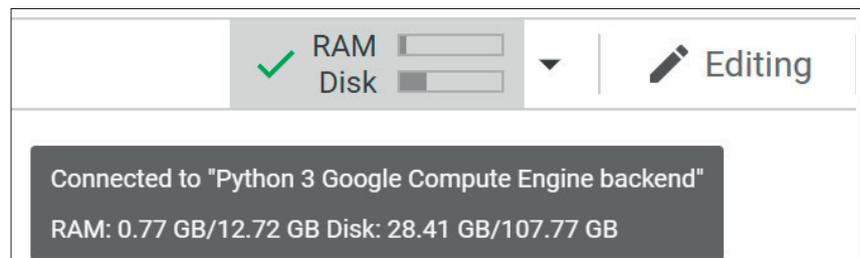


Abb. 2: Bereitgestellte Ressourcen für Analyse und Prognose

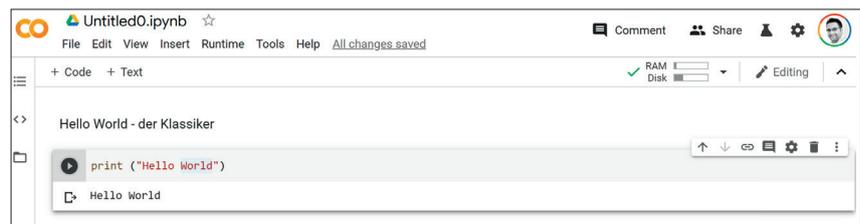


Abb. 3: Das 1. Jupyter-Notebook-Skript

(<https://colab.research.google.com/>) kombinierbar (Abb. 1). Ein wahres Eldorado an Analysemöglichkeiten eröffnet sich dadurch – einzige Voraussetzung ist ein kostenloses Google-Konto.

Nach erfolgreichem Log-in in Colaboratory steht eine interaktive Jupyter-Notebook-Entwicklungsumgebung zur Verfügung, welche komplett in der Cloud läuft. Der Name Jupyter resultiert aus den drei wesentlichen Programmiersprachen **J**ulia, **P**ython und **R**, wobei Jupyter Notebook weitere Programmiersprachen unterstützt. In Summe stellt Google Anwendern kostenfrei und browserbasiert eine Analyse-, Prognose- und Machine-Learning-Lösung bereit, die nach dem Klick auf „Connect“ unmittelbar mit umfassenden Ressourcen zur

Verfügung steht (Abb. 2).

Sobald Colaboratory verbunden wurde, kann mit der Codeeingabe gestartet werden. Am einfachsten wird über den Menüpunkt „File“ -> „New Notebook“ eine neue Umgebung erstellt. Im üblichen Design von Google kann Text und Code eingegeben werden. Über den Befehl „print“ wird bspw. nach Betätigung des Run-Buttons der Inhalt des Print-Befehls ausgegeben. Das heißt, der Code wird direkt ausgeführt und das entsprechend Ergebnis sofort im darunterliegenden Bereich ausgegeben (Abb. 3). In Verbindung mit beschreibendem Text (sogenannte Markdown Cell) kann ein optisch ansprechendes Analysedokument aus Code, Beschreibung und Analysegrafiken

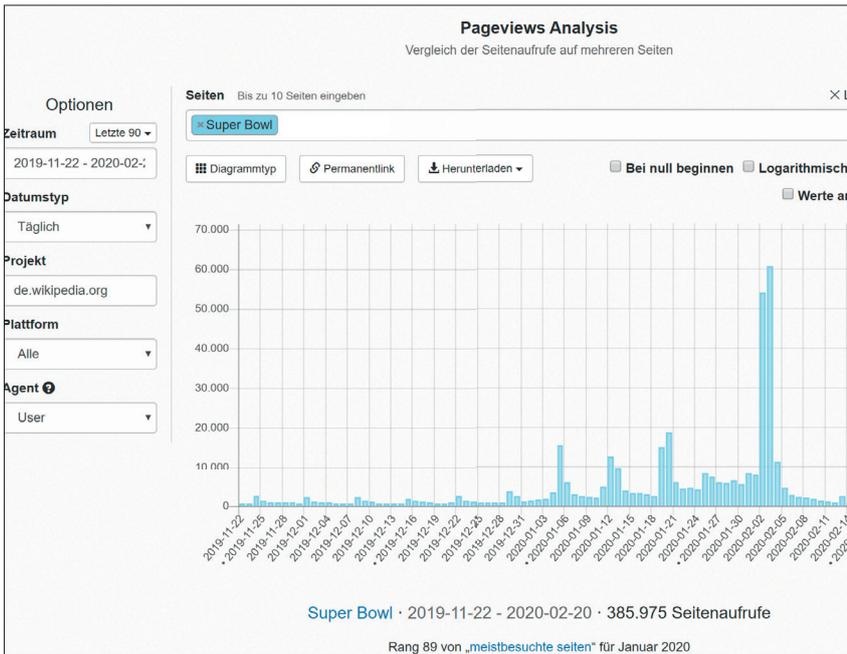


Abb. 4: Wikipedia-Aufrufe der deutschen Seite des Begriffes Super Bowl (tools.wmflabs.org/pageviews/)

erstellt werden. In Google Drive werden die Programmdateien gespeichert und können über die für Drive üblichen Teilungsmöglichkeiten mit anderen Personen ausgetauscht werden. Das heißt, es kann komplett im Team entwickelt werden und die Analysen können mit mehreren Personen geteilt und genutzt werden.

In der Praxis erfolgt die Prognose in der Regel in drei wesentlichen Schritten. Zunächst werden die Daten importiert und aufbereitet. Danach wird aus diesen Daten ein Prognosemodell gebildet, auf Plausibilität geprüft und die eigentliche Prognose erstellt. Im letzten Schritt wird dann das Ergebnis der Prognose visualisiert.

```

+ Code + Text
# Initialisierung von FB Prophet
import pandas as pd
from fbprophet import Prophet

[2] # Import historischer Werte des Spielers Peyton Manning
df = pd.read_csv('https://raw.githubusercontent.com/facebook/prophet/master/examples/example_wp_log_peyton_manning.csv')
df.head()

ds      y
0  2007-12-10  9.590761
1  2007-12-11  8.519590
2  2007-12-12  8.183677
3  2007-12-13  8.072467
4  2007-12-14  7.802572
    
```

Abb. 5: Initialisierung von Facebook Prophet und Import historischer Daten

Facebook Prophet – der Datenprophet

In Colaboratory kann beliebiger Python-Code eingegeben und ausgeführt werden. Der Nutzer profitiert bei Colaboratory (im Vergleich zu einer lokalen Installation von Jupyter Notebook) davon, dass unzählige Python-Pakete vorinstalliert sind. Dass beschleunigt das Arbeiten wesentlich und macht den Einstieg sehr einfach. So kann ohne weiteren Aufwand die Prognoseprozedur Prophet von Facebook genutzt werden, die neben Python auch in R zur Verfügung steht. Facebook Prophet (<https://facebook.github.io/prophet/>) ist ein Verfahren zur Vorhersage von Zeitreihendaten, wobei Trends auf jährlicher, wöchentlicher und täglicher Saisonalität prognostiziert werden können. Selbst Feiertagseffekte werden dabei berücksichtigt. Facebook sagt selbst, dass es am besten mit Zeitreihen funktioniert, die starke saisonale Effekte und mehrere Jahreszeiten mit historischen Daten aufweisen. Prophet ist robust gegenüber fehlenden Daten und Trendverschiebungen und geht in der Regel gut mit Ausreißern um.

```

+ Code + Text

[3] # Anpassung / neue Instanz von Prophet
m = Prophet()
m.fit(df)

INFO:numexpr.utils:NumExpr defaulting to 2 threads.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
<fbprophet.forecaster.Prophet at 0x7f3242247f60>

[4] # Erweiterung des Datensatzes für Prognose der nächsten 365 Tage
future = m.make_future_dataframe(periods=365)
future.tail()

ds
3265  2017-01-15
3266  2017-01-16
3267  2017-01-17
3268  2017-01-18
3269  2017-01-19

# Prognose der logarithmierten Klickdaten mit Intervall (von - bis)
forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()

ds      yhat  yhat_lower  yhat_upper
3265  2017-01-15  8.199274  7.486329  8.893665
3266  2017-01-16  8.524244  7.819909  9.247058
3267  2017-01-17  8.311615  7.614399  9.034306
    
```

Abb. 6: Prognose der zukünftigen Klickdaten mit Angabe eines Intervalls

Produkte und Ereignisse unterliegen starken periodischen Schwankungen.

Ein Dirndl hat eine extrem hohe Nachfrage kurz vor dem Oktoberfest (siehe Google Trends <http://einfach.st/gtrends6>). Über ein Ereignis wie den Super Bowl in Amerika wird ebenfalls kurz vorher bzw. noch am selben Tag intensiv recherchiert (siehe Wikipedia-Aufrufzahlen <https://tools.wmflabs.org/pageviews/>, Abb. 4). Saisonale Abverkäufe von Produkten und Ereignisse sind demnach sehr schwierig zu prognostizieren, weshalb Facebook Prophet hier besonders geeignet ist. Im Folgenden wird Schritt für Schritt eine Prognose von periodisch diffusen Klickdaten durchgeführt. Damit dies bestmöglich am eigenen Rechner mit Programmcode nachvollziehbar ist, wird das Beispiel von Facebook (siehe <http://einfach.st/fbgt2>) genommen.

Die Seitenaufrufe eines Football-Spielers in Wikipedia unterliegen permanenten Schwankungen. In Abhängigkeit von Spieltagen oder Ereignissen wie dem Super Bowl wird die Seite mehr oder weniger oft besucht. Für die Prognose der Zukunft werden Daten aus der Vergangenheit benötigt. Tools wie in Abb. 4 dargestellt oder das R-Paket „wikipediatrend“ liefern historische Seitenaufrufe auf Tagesebene. Damit extreme Ausreißer gemildert werden können, empfiehlt es sich, Daten zu logarithmieren, es findet eine „Stauchung“ der Werte statt (siehe Option „Logarithmische Skala“ in Abb. 4). Liegen diese Daten anschließend bspw. in Form einer Excel- oder CSV-Datei vor, können zukünftige (saisonale) Werte prognostiziert werden.

1. Laden von Facebook Prophet und Import historischer Daten

Im ersten Schritt werden mit Colaboratory zwei wichtige Python-Pakete geladen: Facebook Prophet für die Zeitreihenprognose und pandas zum Laden der Daten in einen sogenannten Dataframe. Vergleichbar mit Bibliotheken in R ist pandas ein Python-Paket, welches optimierte Datenstrukturen für die Ana-

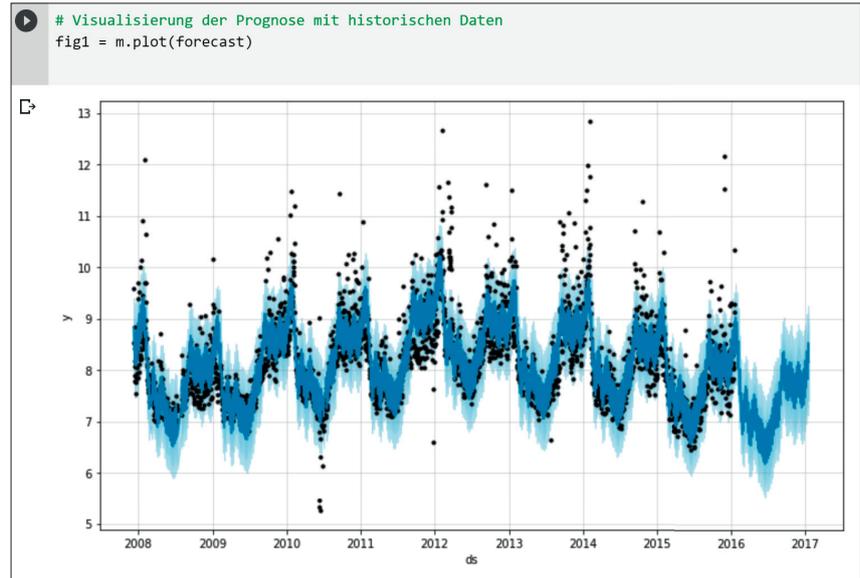


Abb. 7: Historische und zukünftige Klickdaten auf Tagesbasis

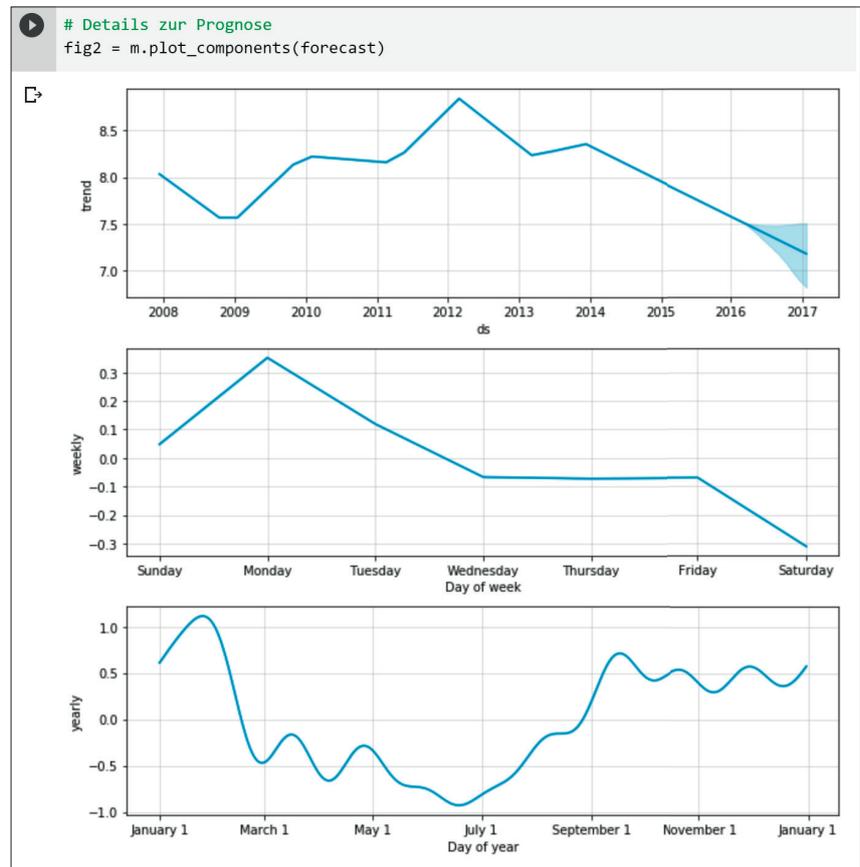


Abb. 8: Details zur Prognose auf Tages- und Monatsebene

lyse ermöglicht. Weiterhin wird über die folgenden Zeilen die CSV-Datei mit Datum und Klickdaten des Wikipediaeintrages des Football-Quarterbacks Peyton Manning in den Dataframe df importiert (siehe Abb. 5). Der Befehl head() zeigt die ersten fünf Datensätze der importierten Datei an.

2. Prognose der Zukunft basierend auf Daten der Vergangenheit

In diesem Schritt zeigt sich die äußerst einfache Erstellung von Prognosemodellen mit Colaboratory und Prophet. Es wird eine neue Instanz von Prophet erstellt und das Modell

Abb. 9: Selektion der Konversionsdaten im Google Analytics Query Explorer

Abb. 10: Auslesen der API Query URI zur Verwendung in Google Colaboratory

```
import pandas as pd
import json # JSON encoder und decoder fuer Python
import requests # Library zum Senden von HTTP requests

from getpass import getpass # Verbergen der Zugriffsvariablen

api_query_uri = getpass('API Query URI:') # Eingabeaufforderung der Google Analytics URI aus dem Query Explorer
r = requests.get(api_query_uri) # Anfrage ausfuehren
data = r.json() # Lesen der Daten im JSON format
df = pd.DataFrame(data['rows']) # Rückgabe der Daten als Pandas data frame
```

Abb. 11: Einlesen der Google-Analytics-Konversionsdaten mit Colaboratory

automatisch an die Gegebenheiten der historischen Datensätze angepasst („Model fit“). So kann mit sehr wenigen Zeilen Programmcode ein umfangreiches Prognosemodell trainiert werden.

Im nächsten Schritt wird in der Methode `Prophet.make_future_dataframe` der Zeitraum der Prognose definiert und der bestehende Datensatz um die entsprechende Anzahl Tage erweitert. Die Methode `Prophet.predict` bestimmt anschließend die Prognosewerte und stellt zusätzlich Unsicherheitsintervalle zur Verfügung (siehe Abb. 6). Das heißt, der spätere tatsächliche Wert wird sich mit sehr hoher Wahrscheinlichkeit (Standard-einstellung von Prophet: 80 % Wahrscheinlichkeit) innerhalb des Intervalls

bewegen. Je weiter die Prognose in die Zukunft geht, desto unsicherer ist die Prognosegüte und desto breiter werden diese Intervalle.

Die prognostizierten Werte sind wieder logarithmierte Ergebnisse und könnten mit der Exponentialfunktion umgekehrt werden. Neben den numerischen Werten empfiehlt sich eine visuelle Darstellung.

3. Visualisierung der Prognose

Die Methode `Prophet.plot` stellt die historischen und prognostizierten Werte in einem Schaubild dar. Für die Vergangenheit wird aufgezeigt, wie gut das Modell die tatsächlichen Werte widerspiegelt, für die Zukunft ergibt sich ein Korridor der zu erwartenden

Seitenaufrufe. Auf den ersten Blick ist erkennbar, wie sich ein langfristiger Trend entwickeln wird. Weitere Details sind über die Methode `Prophet.plot_components` abrufbar (siehe Abb. 8). So kann nicht nur ein globaler Trend aufzeigt, sondern eine Prognose auf Wochentags- und Monatsebene analysiert werden.

Facebook Prophet ist eine sehr gute Möglichkeit, um aus historischen Daten Trends und Prognosen abzuleiten. Ähnlich der Prognose von Seitenaufrufen könnten Telefonanrufe, Traffic nach Kanal und letztlich Konversionen prognostiziert werden – cloudbasiert und kostenlos. Die Daten hierfür können zum Beispiel aus Google Analytics bezogen werden. Hierzu wird über den Google Analytics „Query Explorer“ (ga-dev-tools.appspot.com/query-explorer/) zunächst ein Account und ein Datensatz ausgewählt. Dies können zum Beispiel die Konversionsraten der letzten 30 Tage sein (siehe Abb. 9). Über den Button „Run Query“ kann dann eine API Query URI erzeugt werden (siehe Abb. 10), die dann in Google Colaboratory zur Datenabfrage benutzt wird (siehe Abb. 11).

Insgesamt eröffnet sich damit ein weites Feld in der Prognose von Konversionen über unterschiedliche Zeiträume. Ein Beispiel sind etwa segmentierte Prognosen: Konversionen „Mobile“ mit Browser „Chrome“. So kann gezielt auf Trends in unterschiedlichen Bereichen reagiert werden.

Für Anwender von R steht Prophet ebenfalls zur Verfügung (<http://einfach.st/fbgithub>).

In diesem Sinne: Erweitern Sie Ihr Wissen und erlernen Sie eine Programmiersprache wie Python. Wenige Zeilen Code, die Verwendung leistungsfähiger Algorithmen (wie Prophet) und eine flexible Arbeitsumgebung (wie Google Colaboratory) eröffnen Wettbewerbsvorteile und stärken Ihre Kenntnisse in der Datenanalyse.