SCA_SS24_Gruppe204_HA3

2024-06-13

```
# Laden der Packages
library(dplyr)
```

```
## Warning: Paket 'dplyr' wurde unter R Version 4.2.3 erstellt
```

```
##
## Attache Paket: 'dplyr'
```

```
## Die folgenden Objekte sind maskiert von 'package:stats':
##
##     filter, lag
```

```
## Die folgenden Objekte sind maskiert von 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## Warning: Paket 'tidyverse' wurde unter R Version 4.2.3 erstellt
```

```
## Warning: Paket 'ggplot2' wurde unter R Version 4.2.3 erstellt
```

```
## Warning: Paket 'tibble' wurde unter R Version 4.2.3 erstellt
```

```
## Warning: Paket 'tidyr' wurde unter R Version 4.2.3 erstellt
```

```
## Warning: Paket 'readr' wurde unter R Version 4.2.3 erstellt
```

```
## Warning: Paket 'purrr' wurde unter R Version 4.2.3 erstellt
```

```
## Warning: Paket 'stringr' wurde unter R Version 4.2.3 erstellt
```

```
## Warning: Paket 'forcats' wurde unter R Version 4.2.3 erstellt
```

```
## Warning: Paket 'lubridate' wurde unter R Version 4.2.3 erstellt
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ forcats   1.0.0     ✓ readr     2.1.4
## ✓ ggplot2   3.5.1     ✓ stringr   1.5.0
## ✓ lubridate 1.9.3     ✓ tibble    3.2.1
## ✓ purrr     1.0.2     ✓ tidyr     1.3.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

```
library(reshape2)
```

```
## Warning: Paket 'reshape2' wurde unter R Version 4.2.3 erstellt
```

```
##
## Attache Paket: 'reshape2'
##
## Das folgende Objekt ist maskiert 'package:tidyr':
##
##     smiths
```

```
library(knitr)
```

```
## Warning: Paket 'knitr' wurde unter R Version 4.2.3 erstellt
```

```
library(ggplot2)
library(corrr)
```

```
## Warning: Paket 'corrr' wurde unter R Version 4.2.3 erstellt
```

```
library(ggcorrplot)
```

```
## Warning: Paket 'ggcorrplot' wurde unter R Version 4.2.3 erstellt
```

** Daten für die Modellierung vorbereiten 1) Laden Sie die Datensätze externals und services. Erstellen Sie aus gesamten Datensatz services jeweils ein Dataframe für Shipping- und Warehousing-Dienstleistungen. Berechnen Sie anschließend für jede durchgeführte Dienstleistung die On-Time-Delivery Status (d.h. 0 oder FALSE, wenn unpünktlich; 1 oder TRUE wenn pünktlich) beziehungsweise die Item Fill Rate (IFR). Stellen Sie anschliessend jeweils die Kennzahlen der durchschnittlichen OTD-Rate und der durchschnittlichen IFR als Kennzahl je Logistikdienstleister aggregiert dar. Geben Sie diese Werte in zwei Tabellen aus. Die Tabellen sollen einen einfachen Vergleich der LDL ermöglichen. Bewertungsrel- evant: Output, Code. Hinweis: Erneut bietet es sich an, eine Variable Periode dem Datensatz hinzu zu fügen, welche aus Jahr und Monat besteht (im Format YYYYMM, z.B. Februar 2019 –> 201902)

```
## Laden der Daten
externals = read.csv2("externals25.csv")
services = read.csv2("output_services_v0025.csv")
services$Periode = paste(services$Year, ifelse(services$Month < 10, paste("0", services$Mont
h, sep=""), services$Month), sep="")


## Erstellung von Dataframes für Shipping- und Warehousing-Dienstleistungen
shipping = subset(services, service == "Shipping")
warehousing = subset(services, service == "Warehousing")


## Erstellung des Attributes OnTimeDeliveryStatus bzw. ItemFIllRate für jede Dienstleistung
shipping$OnTimeDeliveryStatus = with(shipping, ifelse(DaysScheduled >= DaysExecuted, 1, 0))
warehousing$ItemFillRate = with(warehousing, QExecuted/QScheduled)


## Berechnung der durchschnittlichen OTD-Rate und der durchschnittlichen IFR
# OTD-Rate der Shipping-DL
shipping_OTD = shipping %>%
  group_by(vendor) %>%
  summarise(
    avg_OTD_Rate = mean(OnTimeDeliveryStatus, na.rm = TRUE)
  )
# IFR der Warehousing-DL
warehousing_IFR = warehousing %>%
  group_by(vendor) %>%
  summarise(
    avg_IFR = mean(ItemFillRate, na.rm = TRUE)
  )

## Ausgabe der neuen Tabellen zum Vergleich der LDL
shipping_OTD
```

```
## # A tibble: 10 × 2
##    vendor                   avg_OTD_Rate
##    <chr>                           <dbl>
##  1 AHL Express Shipping            0.264
##  2 Bange+Hammer Shipping           0.329
##  3 CPS Shipping                    0.267
##  4 DWL Shipping                    0.335
##  5 EPD Shipping                    0.326
##  6 Flying Mercury Shipping         0.317
##  7 Gifter Shipping                 0.323
##  8 HCX Shipping                    0.311
##  9 IntEx Shipping                  0.336
## 10 JNT Shipping                    0.314
```

```
warehousing_IFR
```

```
## # A tibble: 10 × 2
##    vendor                      avg_IFR
##    <chr>                         <dbl>
##  1 AHL Express Warehousing       0.829
##  2 Bange+Hammer Warehousing      0.844
##  3 CPS Warehousing               0.839
##  4 DWL Warehousing               0.850
##  5 EPD Warehousing               0.810
##  6 Flying Mercury Warehousing    0.837
##  7 Gifter Warehousing            0.825
##  8 HCX Warehousing               0.815
##  9 IntEx Warehousing             0.832
## 10 JNT Warehousing               0.825
```

2. Erzeugen Sie ein neues Dataframe, welches die aggregierte IFR je Warehousing-Logistikdienstleister enthält. Die IFR soll je Warehousing-LDL, Region und Periode (eine Periode = ein Monat eines einzelnen Jahres) aggregiert werden. Nehmen Sie kurz Stellung, wie Sie die Qualität dieser Dienstleistungen allgemein einschätzen. Identi- fizieren Sie danach den insgesamt schlechtesten Warehousing-DL. Geben Sie anschliessend den besten IFR-Wert und die entsprechende Periode aus, den dieser in der Region Japan jemals erreicht hat. Bewertungsrelevant: Kommentar, Output, Code.

```
## Erzeugung eines aggregrieten Dataframe je Warehousing-LDL, Region, Periode
warehousing_aggregated = warehousing %>%
  group_by(vendor, region, Periode) %>%
  summarise(
    avg_IFR = mean(ItemFillRate, na.rm = TRUE)
  )
```

```
## `summarise()` has grouped output by 'vendor', 'region'. You can override using
## the `.groups` argument.
```

```
## Um die Qualität einzuschätzen, ist es notwendig, die allgemine durchschnittliche IFR, die je nur LDL aggregiert ist (bei der obigen Variable warehousing_IFR) zu betrachten. Die IFR, die je LDL, Region und Periode wie bei warehousing_aggregated aggregiert ist, ist nicht dafür nicht vernünftig.
# Ausgabe des schlechtesten Warehousing-DL mit avg_IFR
worst_warehousing_vendor = warehousing_IFR[which.min(warehousing_IFR$avg_IFR),]
cat("Der schlechteste Warehousing-DL ist ", worst_warehousing_vendor$vendor, "mit einem IFR-Wert von ", worst_warehousing_vendor$avg_IFR,"\n")
```

```
## Der schlechteste Warehousing-DL ist  EPD Warehousing mit einem IFR-Wert von  0.8095628
```

```
## Ausgabe des besten IFR-Wertes einer Periode  in Japan
# Filterung der IFR-Werte des schlechtesten WH-DL in Japan
warehousing_aggregated_jp = subset(warehousing_aggregated, region == "Japan")
worst_warehousing_vendor_jp = subset(warehousing_aggregated_jp, vendor == worst_warehousing_vendor$vendor)
# Extraktion der Zeile mit dem besten IFR
best_IFR = worst_warehousing_vendor_jp[which.max(worst_warehousing_vendor_jp$avg_IFR),]
cat("Der beste IFR-Wert in Japan von " , best_IFR$vendor, " ist ", best_IFR$avg_IFR, " in der Periode ", best_IFR$Periode, ".")
```

```
## Der beste IFR-Wert in Japan von  EPD Warehousing  ist  0.8420917  in der Periode  202207 .
```

3. Erzeugen Sie ein neues Dataframe, welches die aggregierte OTD je Shipping-Logistikdienstleister enthält. Die OTD soll je Shipping-LDL, Region und Periode (eine Periode = ein Monat eines einzelnen Jahres) aggregiert wer- den. Nehmen Sie kurz Stellung, wie Sie die Qualität dieser Dienstleistungen allgemein einschätzen. Geben Sie anschliessend den OTD-Wert (und die entsprechende Periode) aus, den der beste Shipping-DL im April 2022 in der Region Shanghai erreicht hat. Bewertungsrelevant: Output, Code.

```
## Erzeugung eines aggregrieten Dataframe je Shipping-LDL, Region, Periode
shipping_aggregated = shipping %>%
  group_by(vendor, region, Periode) %>%
  summarise(
    avg_OTD = mean(OnTimeDeliveryStatus, na.rm = TRUE)
  )
```

```
## `summarise()` has grouped output by 'vendor', 'region'. You can override using
## the `.groups` argument.
```

```
## Um die Qualität einzuschätzen, ist es notwendig, die allgemine durchschnittliche OTD-Rate,
die je nur LDL aggregiert ist (bei der obigen Variable shipping_OTD) zu betrachten. Die OTD-D
ate, die je LDL, Region und Periode wie bei shipping_aggregated aggregiert ist, ist nicht daf
ür nicht vernünftig.


## Ausgabe des besten IFR-Wertes einer Periode  in Japan
# Filterung der IFR-Werte in Japan
shipping_aggregated_sh = subset(shipping_aggregated, region == "Shangh")
shipping_aggregated_sh_april2022 = subset(shipping_aggregated_sh, Periode == "202204")
shipping_aggregated_sh_april2022
```

```
## # A tibble: 10 × 4
## # Groups:   vendor, region [10]
##    vendor                  region Periode avg_OTD
##    <chr>                   <chr>  <chr>     <dbl>
##  1 AHL Express Shipping    Shangh 202204    0.318
##  2 Bange+Hammer Shipping   Shangh 202204    0.417
##  3 CPS Shipping            Shangh 202204    0.227
##  4 DWL Shipping            Shangh 202204    0.16
##  5 EPD Shipping            Shangh 202204    0.389
##  6 Flying Mercury Shipping Shangh 202204    0.4
##  7 Gifter Shipping         Shangh 202204    0.364
##  8 HCX Shipping            Shangh 202204    0.222
##  9 IntEx Shipping          Shangh 202204    0.375
## 10 JNT Shipping            Shangh 202204    0.385
```

```
# Extraktion der Zeile mit der besten OTD-Rate
best_shipping_vendor_april2022_sh = shipping_aggregated_sh_april2022[which.max(shipping_aggre
gated_sh_april2022$avg_OTD),]
best_shipping_vendor_april2022_sh
```

```
## # A tibble: 1 × 4
## # Groups:   vendor, region [1]
##   vendor                region Periode avg_OTD
##   <chr>                 <chr>  <chr>     <dbl>
## 1 Bange+Hammer Shipping Shangh 202204    0.417
```

```
cat("Der beste OTD-Wert ist ", best_shipping_vendor_april2022_sh$avg_OTD, " in der Periode ",
best_shipping_vendor_april2022_sh$Periode, " in Shanghai, bei ", best_shipping_vendor_april20
22_sh$vendor)
```

```
## Der beste OTD-Wert ist  0.4166667  in der Periode  202204  in Shanghai, bei  Bange+Hammer
Shipping
```

4. Wählen Sie den Warehousing-DL “Gifter Warehousing” aus. Vereinigen Sie das eben erzeugte Dataframe (genauer: ein Subset dieses Dataframes bezüglich des gewählten Warehousing-DL) mit den externen Fak- toren der jeweiligen Periode und Region in einem neuen Dataframe. Zeigen Sie davon den Tabellenkopf. Bewertungsrelevant: Output. Hinweis: In der Funktion merge() können mehrere überschneidende Spalten genutzt werden, indem dem “by =”-Parameter ein Vektor der Spalten übergeben wird. Ihnen steht frei, andere Funktionen zu verwenden.

```
# Extraktion der Daten von Gifter Warehousing
gifter = subset(warehousing_aggregated, vendor == "Gifter Warehousing")

# Erzeugung des Attributes Periode in externals
externals$Periode = paste(externals$Year, ifelse(externals$Month < 10, paste("0", externals$M
onth, sep=""), externals$Month), sep="")

# Vereinigung der Dataframes
gifter_externals = merge(gifter, externals, by = c("region", "Periode"), all = TRUE)
head(gifter_externals)
```

```
##   region Periode             vendor   avg_IFR  X Year Month Period
## 1  Japan  201901 Gifter Warehousing 0.8215646  3 2019     1 2019/1
## 2  Japan  201902 Gifter Warehousing 0.8202110  8 2019     2 2019/2
## 3  Japan  201903 Gifter Warehousing 0.8043408 13 2019     3 2019/3
## 4  Japan  201904 Gifter Warehousing 0.8221602 18 2019     4 2019/4
## 5  Japan  201905 Gifter Warehousing 0.8209567 23 2019     5 2019/5
## 6  Japan  201906 Gifter Warehousing 0.8230232 28 2019     6 2019/6
##   Temperature_C Rain_mm Sunshine_h Humidity Congestion InternetStability
## 1        -1.248  39.626     50.327   82.110      38.79              1946
## 2         2.076  36.334     73.275   79.537      53.40              1659
## 3         6.885  38.684    109.297   71.786      45.74              1947
## 4        10.457  39.863    167.497   61.434      50.17              1683
## 5        13.327  55.793    228.877   59.735      52.31              1593
## 6        15.958  65.592    240.627   59.295      41.58              1740
##   PowerGridStability ParkingSpaceAvailability RoadCondition PoliticalStability
## 1               0.20               0.03246347          6.17               2.51
## 2               0.21               0.03073584          5.70               2.75
## 3               0.24               0.02993043          5.96               2.82
## 4               0.30               0.03057238          6.37               2.86
## 5               0.27               0.02905311          6.80               2.95
## 6               0.21               0.03124112          6.67               3.21
##   AvgHealth Criminality AirPollution WaterQuality leisureAndSocialInteractions
## 1     79.77       30.84       246.42         0.98                         6.26
## 2     79.24       30.38       211.20         0.91                         6.43
## 3     78.49       30.14       226.49         0.87                         6.53
## 4     80.01       29.66       211.73         0.96                         6.21
## 5     81.18       28.84       230.60         0.94                         5.88
## 6     82.92       28.50       216.48         0.88                         6.65
##   SkilledLaborAvailability UnskilledLaborAvailability WorkerMotivation Overtime
## 1                    88.00                      64.37             7.43    0.004
## 2                    87.92                      64.46             7.03    0.072
## 3                    87.62                      65.14             6.73    0.173
## 4                    86.65                      65.01             6.45    0.000
## 5                    86.36                      65.50             6.76    0.083
## 6                    85.42                      64.59             7.11    0.051
##    Inflation BusinessConfidence FuelPrice
## 1 0.00101089           101.1787     2.400
## 2 0.00094253           101.0367     2.433
## 3 0.00088346           100.9075     2.387
## 4 0.00091825           100.9459     2.428
## 5 0.00083984           100.9247     2.336
## 6 0.00086169           100.9456     2.376
```

5. Sie möchten sich eine Übersicht zu der Korrelation zwischen den externen Faktoren und der IFR des Warehousing- Dienstleister schaffen. Führen Sie dazu die folgenden Schritte aus:

a. Geben Sie eine unsortierte Tabelle aus, in der die externen Effekte und deren Korrelation zur IFR abgebildet sind.
b. Geben Sie eine Tabelle aus, in der die 5 am stärksten zur IFR korrelierenden externen Effekten und deren Korrelation zur IFR abgebildet sind. Wie bewerten Sie die Korrelation zwischen diesen 5 Faktoren und der IFR?
c. Erstellen Sie ein Korrelations-Plot für diese 5 externen Faktoren. Bewertungsrelevant: Kommentar, Output.

```
## (a) Externe Faktoren und deren Korrelation zur IFR anzeigen
# Berechnung der Korrelationen von den Faktoren zur IFR
correlations <- cor(gifter_externals %>% select_if(is.numeric), use = "complete.obs")
correlations_df <- as.data.frame(as.table(correlations)) %>%
  filter(Var1 == "avg_IFR" & Var2 != "avg_IFR") %>%
  select(Var2, Freq) %>%
  rename(Factor = Var2, Correlation = Freq)
kable(correlations_df, caption = "Externen Faktoren und deren Korrelation zur IFR")
```

Externen Faktoren und deren Korrelation zur IFR

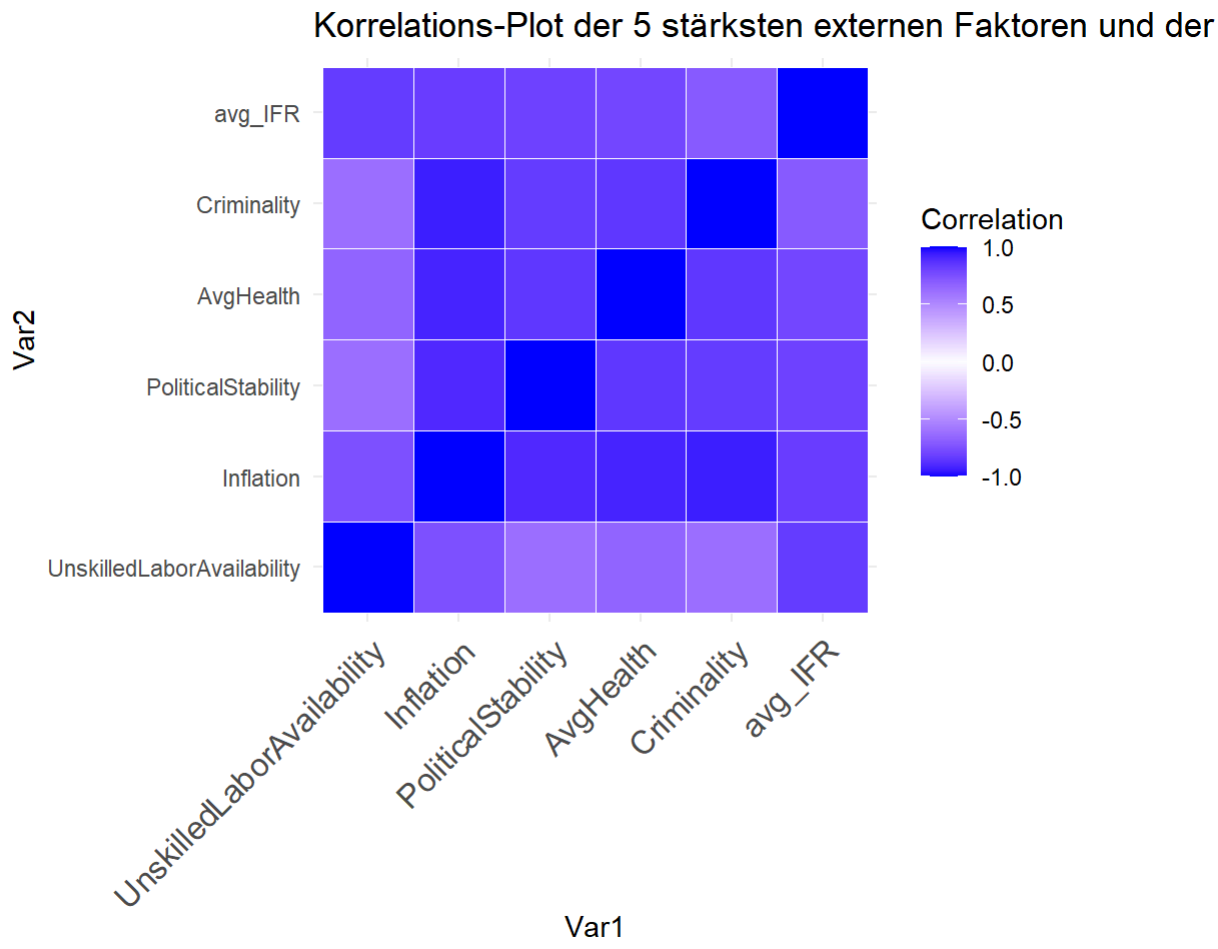
Factor	Correlation
X	0.1476839
Year	0.1485109
Month	-0.0303370
Temperature_C	-0.1937370
Rain_mm	-0.1106732
Sunshine_h	-0.1747338
Humidity	0.1525125
Congestion	-0.0450058
InternetStability	-0.3267066
PowerGridStability	0.5332786
ParkingSpaceAvailability	0.2224489
RoadCondition	-0.6276854
PoliticalStability	0.8105602
AvgHealth	-0.7821141
Criminality	0.7139624
AirPollution	-0.0825646
WaterQuality	-0.3584736
leisureAndSocialInteractions	0.0607541
SkilledLaborAvailability	0.0091716
UnskilledLaborAvailability	-0.8329434
WorkerMotivation	-0.3436794
Overtime	-0.3662529
Inflation	0.8213916
BusinessConfidence	0.4645718
FuelPrice	0.0550331

```
## (b) Top 5 Faktoren mit der höchsten Korrelation auswählen und anzeigen
top5_correlations <- correlations_df %>%
  arrange(desc(abs(Correlation))) %>%
  head(5)
kable(top5_correlations, caption = "Die 5 am stärksten zur IFR korrelierenden externen Effekt
en")
```

Die 5 am stärksten zur IFR korrelierenden externen Effekten

Factor	Correlation
UnskilledLaborAvailability	-0.8329434
Inflation	0.8213916
PoliticalStability	0.8105602
AvgHealth	-0.7821141
Criminality	0.7139624

```
## (c) Korrelationen für die Top 5 Faktoren visualisieren
# Baseline hesaplama (ort. IFR)
# Extraktion der 5 Faktoren
correlation_data <- gifter_externals %>%
  select(all_of(top5_correlations$Factor), avg_IFR)
# Berechnung der Korrelationen zwischen den Faktoren und der IFR
correlation_matrix <- cor(correlation_data, use = "complete.obs")
correlation_melt <- melt(correlation_matrix)
# Visualisierung der Korrelationen
ggplot(data = correlation_melt, aes(Var1, Var2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "blue", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name = "Correlation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 12, hjust = 1)) +
  coord_fixed() +
  labs(title = "Korrelations-Plot der 5 stärksten externen Faktoren und der IFR")
```



6. Sie möchten nun eine Lineare Regression durchführen, um die IFR mit Hilfe der externen Effekte vorherzusagen. Um die Güte Ihrer Modelle vergleichen zu können, benötigen Sie eine geeignete Baseline. Erzeugen Sie eine sin- nvolle Baseline in dem Dataframe zu Ihrem gewählten Warehousing-DL in einer Variable Baseline. Begründen Sie Ihre Wahl. Geben Sie von dem DataFrame den Tabellenkopf aus. Geben Sie Sie nur die Spalten ‘Periode’, ‘Region’, ‘IFR’ und ‘Baseline’ aus. Bewertungsrelevant: Output, Begründung.

```
## Durchschnittliche IFR als Baseline ausgewählt, da sie den zentralen Tendenz der vorhandene
n Daten repräsentiert und einen einfachen Referenzpunkt vor der Erstellung eines komplexen Mo
dells bietet. So kann die durchschnittliche Leistung des erstellten Modells im Vergleich zu d
ieser zentralen Tendenz bewertet werden. Der Durchschnittswert ist oft eine der einfachsten u
nd effektivsten Metriken, um das allgemeine Verhalten der Daten zusammenzufassen.
# Berechnung des historischen Durchschnitts-IFR als Baseline
historical_avg_IFR <- mean(gifter_externals$avg_IFR, na.rm = TRUE)
# Hinzufügen der Baseline-Spalte zum DataFrame
gifter_externals <- gifter_externals %>%
  mutate(Baseline = historical_avg_IFR)
# Ausgabe der relevanten Spalten
gifter_baseline_output <- gifter_externals %>%
  select(Periode, region, avg_IFR, Baseline)
# Ausgabe des Tabellenkopfs
head(gifter_baseline_output)
```

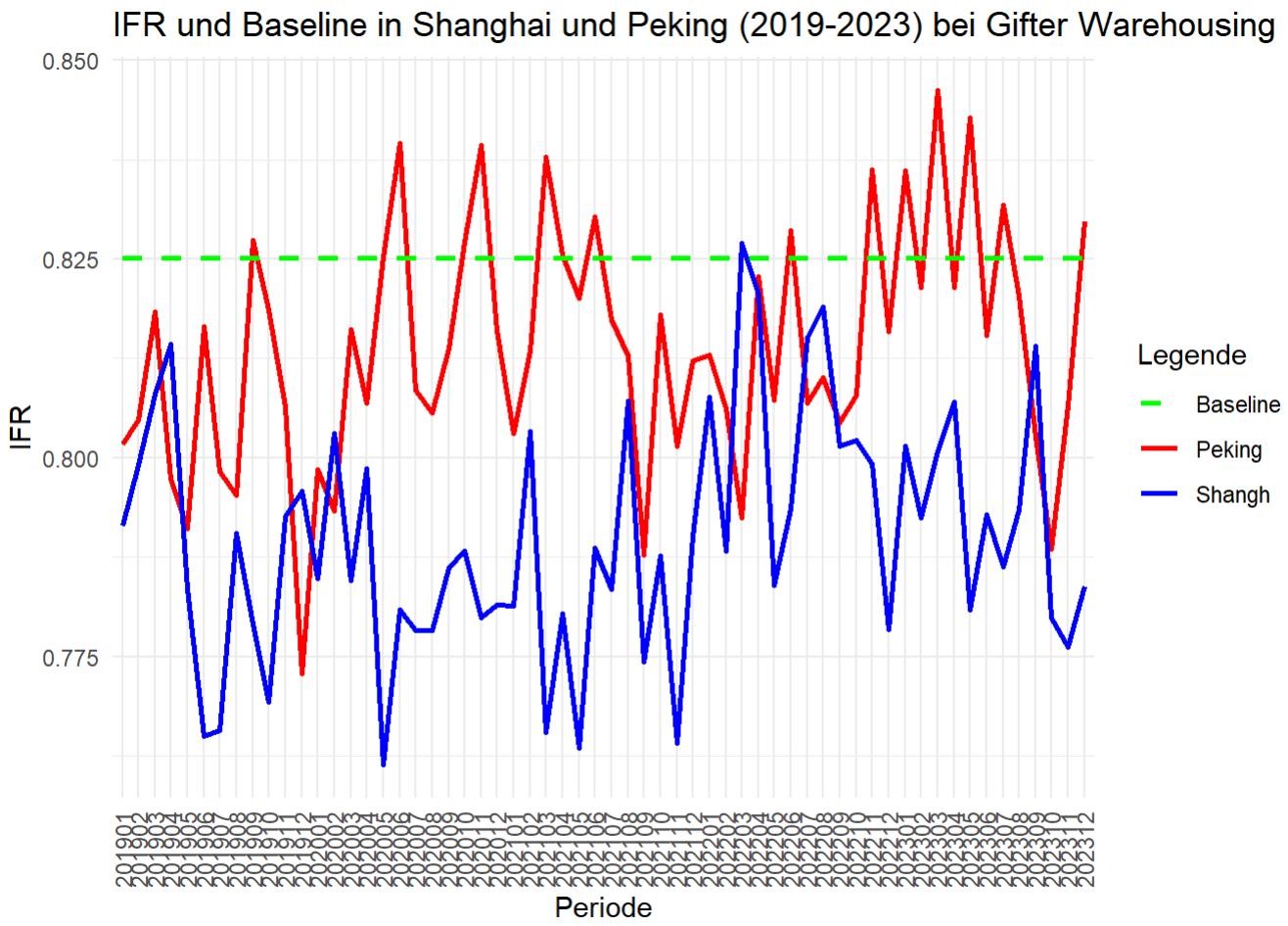
```
##   Periode region   avg_IFR  Baseline
## 1  201901  Japan 0.8215646 0.8250903
## 2  201902  Japan 0.8202110 0.8250903
## 3  201903  Japan 0.8043408 0.8250903
## 4  201904  Japan 0.8221602 0.8250903
## 5  201905  Japan 0.8209567 0.8250903
## 6  201906  Japan 0.8230232 0.8250903
```

7. Visualisieren Sie die Baseline Ihres gewählten LDL für den Zeitraum von 2019 bis 2023 sowie die IFR in der Region Shanghai und die IFR in der Region Peking Bewertungsrelevant: Output.

```
# Filtern der Daten für die Regionen Shanghai und Peking
gifter_shanghai_peking <- gifter_externals %>%
  filter(region %in% c("Shangh", "Peking") & Periode >= 201901 & Periode <= 202312)

# Visualisierung der Baseline und IFR-Werte für Shanghai und Peking
ggplot(gifter_shanghai_peking, aes(x = Periode, group = region)) +
  geom_line(aes(y = avg_IFR, color = region), size = 1) +
  geom_line(aes(y = Baseline, color = "Baseline"), linetype = "dashed", size = 1) +
  labs(title = "IFR und Baseline in Shanghai und Peking (2019-2023) bei Gifter Warehousing",
       x = "Periode",
       y = "IFR",
       color = "Legende") +
  scale_color_manual(values = c("Shangh" = "blue", "Peking" = "red", "Baseline" = "green")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



8. Bewerten Sie die Baseline für Ihren gewählten Warehousing-Logistikdienstleister nach MAE und MAPE und spe- ichern Sie diese in einem Dataframe (z.B. “evaluation”) ab. Dieses Dataframe soll später auch für die Bewertung der Regressionsmodelle genutzt werden. Fügen Sie zudem auch eine Spalte für das Bestimmtheitsmass (R^2) sowie das adjustierte Bestimmtheitsmass (adj. R^2) hinzu, welche im Falle der Baseline 0 sein können. Bewer- tungsrelevant: Output.

```
# Berechnung von MAE und MAPE
baseline_mae = mean(abs(gifter_externals$avg_IFR - gifter_externals$Baseline), na.rm = TRUE)
baseline_mape = mean(abs((gifter_externals$avg_IFR - gifter_externals$Baseline)/gifter_extern
als$avg_IFR)*100, na.rm = TRUE)

# Erstellung des neuen Dataframes zur Speicherung von MAE und MAPE
evaluation = data.frame(
  Model = "Baseline",
  MAE = baseline_mae,
  MAPE = baseline_mape,
  R2 = 0,
  adj_R2 = 0
)
evaluation
```

```
##      Model        MAE     MAPE R2 adj_R2
## 1 Baseline 0.02812247 3.372565  0      0
```

9. Teilen Sie das Dataframe Ihres gewählten Warehousing-Logistikdienstleisters in ein Trainings- (80%) und ein Test- Set (20%) auf. Geben Sie von beiden den Tabellenkopf aus. Setzen Sie vorher den Seed 4141. Bewertungsrele- vant: Code, Output.

```r
# Seed 4141 für Reproduzierbarkeit
set.seed(4141)

# Aufteilung der Daten in Trainings- (80%) und Test-Set (20%)
sample_size <- floor(0.8 * nrow(gifter_externals))
train_indices <- sample(seq_len(nrow(gifter_externals)), size = sample_size)
train_set <- gifter_externals[train_indices, ]
test_set <- gifter_externals[-train_indices, ]

# Ausgabe der Tabellenköpfe
head(train_set)
```

```
##     region Periode             vendor   avg_IFR   X Year Month Period
## 52   Japan  202304 Gifter Warehousing 0.8370769 258 2023     4 2023/4
## 295 Skorea  201907 Gifter Warehousing 0.8917266  34 2019     7 2019/7
## 281 Shangh  202405               <NA>        NA 321 2024     5 2024/5
## 333 Skorea  202209 Gifter Warehousing 0.8810750 224 2022     9 2022/9
## 39   Japan  202203 Gifter Warehousing 0.8371107 193 2022     3 2022/3
## 327 Skorea  202203 Gifter Warehousing 0.8895385 194 2022     3 2022/3
##     Temperature_C Rain_mm Sunshine_h Humidity Congestion InternetStability
## 52         10.215  41.735    168.005   61.424      47.87              1758
## 295        19.573  63.539    283.932   66.910      64.47                64
## 281        10.818  49.595    290.481   62.022      65.73               347
## 333        17.687  52.035    179.996   70.788      51.79                70
## 39          7.520  36.679    124.040   65.490      42.47              1944
## 327        -2.203  40.916    113.034   82.205      45.38                67
##     PowerGridStability ParkingSpaceAvailability RoadCondition
## 52                0.25               0.03030833          5.49
## 295               2.78               0.03301041          2.10
## 281               1.40               0.01766216          6.61
## 333               3.19               0.03267584          2.10
## 39                0.19               0.03333671          6.40
## 327               3.12               0.03275662          2.46
##     PoliticalStability AvgHealth Criminality AirPollution WaterQuality
## 52                1.74     81.77       36.74       220.14         0.97
## 295               5.67     54.95       52.72        53.79         0.69
## 281               1.51     93.32       28.81        66.75         0.90
## 333               6.03     51.33       69.82        59.30         0.69
## 39                2.48     86.47       31.34       204.93         0.98
## 327               6.13     51.71       68.81        58.95         0.73
##     leisureAndSocialInteractions SkilledLaborAvailability
## 52                          7.51                    88.09
## 295                         6.07                    53.92
## 281                         6.46                    51.40
## 333                         7.05                    54.66
## 39                          6.45                    86.95
## 327                         7.13                    56.87
##     UnskilledLaborAvailability WorkerMotivation Overtime  Inflation
## 52                       56.16             5.03    0.006 0.00137639
## 295                      46.36             7.13    0.048 0.01321979
## 281                      74.65             7.68    0.102 0.00143533
## 333                      50.94             5.77    0.150 0.01323376
## 39                       58.51             7.47    0.000 0.00128991
## 327                      49.37             6.19    0.126 0.01313304
##     BusinessConfidence FuelPrice  Baseline
## 52            101.4466     2.877 0.8250903
## 295           101.7995     2.434 0.8250903
## 281            99.5323     2.723 0.8250903
## 333           102.6251     2.875 0.8250903
## 39            101.4138     2.854 0.8250903
## 327           102.3551     2.754 0.8250903
```

```
head(test_set)
```

```
##    region Periode             vendor   avg_IFR   X Year Month  Period
## 1   Japan  201901 Gifter Warehousing 0.8215646   3 2019     1  2019/1
## 15  Japan  202003 Gifter Warehousing 0.8215083  73 2020     3  2020/3
## 24  Japan  202012 Gifter Warehousing 0.8074331 118 2020    12 2020/12
## 27  Japan  202103 Gifter Warehousing 0.7924641 133 2021     3  2021/3
## 29  Japan  202105 Gifter Warehousing 0.8074074 143 2021     5  2021/5
## 30  Japan  202106 Gifter Warehousing 0.8272214 148 2021     6  2021/6
##    Temperature_C Rain_mm Sunshine_h Humidity Congestion InternetStability
## 1         -1.248  39.626     50.327   82.110      38.79              1946
## 15         2.718  45.059    117.070   69.353      54.53              1924
## 24         2.172  56.013     36.750   84.576      38.60              2034
## 27         5.602  42.460    119.255   69.636      52.22              1644
## 29        14.646  43.188    219.853   60.689      54.77              1643
## 30        16.197  68.066    229.851   57.637      46.02              1743
##    PowerGridStability ParkingSpaceAvailability RoadCondition PoliticalStability
## 1                0.20               0.03246347          6.17               2.51
## 15               0.24               0.03116164          6.80               2.49
## 24               0.19               0.02947696          6.17               1.97
## 27               0.22               0.02726092          6.00               1.89
## 29               0.22               0.02958673          5.36               2.08
## 30               0.19               0.02829957          6.20               2.04
##    AvgHealth Criminality AirPollution WaterQuality leisureAndSocialInteractions
## 1      79.77       30.84       246.42         0.98                         6.26
## 15     89.97       30.04       222.73         0.92                         6.38
## 24     91.50       29.35       182.79         0.98                         6.21
## 27     93.95       30.33       231.86         0.94                         6.79
## 29     91.95       27.82       236.93         0.98                         6.89
## 30     91.34       29.44       228.75         0.87                         6.28
##    SkilledLaborAvailability UnskilledLaborAvailability WorkerMotivation
## 1                     88.00                       64.37             7.43
## 15                    84.30                       61.81             6.98
## 24                    87.21                       62.55             8.34
## 27                    88.50                       63.84             8.11
## 29                    89.81                       60.46             8.66
## 30                    89.03                       60.54             8.10
##    Overtime  Inflation BusinessConfidence FuelPrice  Baseline
## 1     0.004 0.00101089           101.1787     2.400 0.8250903
## 15    0.224 0.00119176           100.8039     2.642 0.8250903
## 24    0.109 0.00126889           100.6925     2.600 0.8250903
## 27    0.072 0.00134821           101.0347     2.631 0.8250903
## 29    0.179 0.00136504           101.0379     2.539 0.8250903
## 30    0.067 0.00141953           100.8969     2.521 0.8250903
```

10. Wenden Sie die Forward Selection Variante der Wrapper Methode an (siehe Vorlesung). D.h. erstellen Sie zunächst alle uni-variaten Modelle, bewerten Sie diese Modelle und wählen Sie das Modell mit der besten Bew- ertung aus. Erstellen Sie - basierend auf dem besten Modell der ersten Iteration - alle bi-variaten Modelle (das Modell der vorherigen Wrapper-Iteration wird jeweils um eine Variable erweitert), bewerten Sie diese Modelle und wählen Sie das Modell mit der besten Bewertung aus. Führen Sie dies so lange fort, bis keine Verbesserung mehr erreicht wird. Nutzen Sie zur Modellierung die lineare Regression. Bewerten Sie die Modelle entsprechend nach MAE und MAPE sowie nach regressionsspezifischen Kennzahlen. Nutzen Sie nur die 5 externen Faktoren als Features, die Sie oben als am stärksten korrelierende externe Faktoren identifiziert haben. Kommentieren Sie Ihr Vorgehen zwischen den Iterationen. Bewertungsrelevant: Output, Vorgehen (einschliesslich Kommentare). Hinweis: Tritt eine starke Multikollinearität (“strong multicollinearity”) auf, so können Sie alle Modellierun-

gen mit der entsprechenden Variablen-Kombination unter Bezug auf diesen Hinweis auslassen (siehe Vor- lesungsinhalte zu Korrelation). Hinweis 2: Für das Erstellen der Modelle reicht es aus, zunaechst die Trainings-Daten zu nutzen. Über- prüfen Sie ihr endgültiges Modell jedoch am Ende auf Overfitting, indem Sie die Test-Daten nutzen! Hinweis 3: Sie müssen kein Feature Engineering betreiben. Sie müssen auch nicht die Residuenplots über- prüfen.

```
results = data.frame(Model = character(), MAE = numeric(), MAPE = numeric(), stringsAsFactors
= FALSE)
## Erstellung und Bewertung univariater Modelle
for (e_factor in top5_correlations$Factor) {
  formula = as.formula(paste("avg_IFR~", e_factor))
  model = lm(formula, data = train_set)
  predictions = predict(model, test_set)
  mae = mean(abs(test_set$avg_IFR - predictions), na.rm = TRUE)
  mape = mean(abs((test_set$avg_IFR - predictions)/test_set$avg_IFR)*100, na.rm = TRUE)
  results = rbind(results, data.frame(Model = e_factor, MAE = mae, MAPE = mape))
}
# Ausgabe der MAE und MAPE der Modelle
results
```

```
##                         Model        MAE     MAPE
## 1 UnskilledLaborAvailability 0.01656425 2.036624
## 2                   Inflation 0.01602798 1.985944
## 3          PoliticalStability 0.01520303 1.873114
## 4                   AvgHealth 0.01592079 1.973247
## 5                Criminality 0.01847355 2.287197
```

```
# Von den Ergebnissen stellt man fest, dass der Faktor "PoliticalStability" den kleinsten MAE
und den kleinsten MAPE hat und somit das beste Modell ist.
best_univariate = results[which.min(results$MAE), ]
print(best_univariate)
```

```
##                 Model        MAE     MAPE
## 3 PoliticalStability 0.01520303 1.873114
```

```
bi_results = data.frame(Model = character(), MAE = numeric(), MAPE = numeric(), stringsAsFact
ors = FALSE)
## Erstellung und Bewertung bivariater Modelle
for (e_factor in setdiff(top5_correlations$Factor, best_univariate$Model)) {
  formula = as.formula(paste("avg_IFR~", best_univariate$Model, "+", e_factor))
  model = lm (formula, data = train_set)
  predictions = predict(model, test_set)
  mae = mean(abs(test_set$avg_IFR - predictions), na.rm = TRUE)
  mape = mean(abs((test_set$avg_IFR - predictions)/test_set$avg_IFR)*100, na.rm = TRUE)
  bi_results = rbind(bi_results, data.frame(Model = paste("PoliticalStability + ", e_factor),
MAE = mae, MAPE = mape))
}
# Ausgabe der MAE und MAPE der Modelle

# Von den Ergebnissen stellt man fest, dass "PoliticalStability + UnskilledLaborAvailability"
den kleinsten MAE und kleinsten MAPE hat und somit das beste Modell ist
best_bivariate = bi_results[which.min(bi_results$MAE), ]
best_bivariate
```

```
##                                              Model        MAE     MAPE
## 1 PoliticalStability +  UnskilledLaborAvailability 0.01114345 1.372189
```

```
## Erweiterung des besten bivariaten Modells bis keine Verbesserung mehr erreicht wird
# Funktion zur Iteration der Modellbildung
fw_selection = function(current_model, remaining_factors) {
  improved = TRUE
  while(improved) {
    improved = FALSE
    current_best = current_model
    for(e_factor in setdiff(remaining_factors, unlist(strsplit(current_model, " \\+")))) {
      formula = as.formula(paste("avg_IFR~", current_model, "+", e_factor))
      model = lm(formula, data = train_set)
      predictions = predict(model, test_set)
      mae = mean(abs(test_set$avg_IFR - predictions), na.rm = TRUE)
      mape = mean(abs((test_set$avg_IFR - predictions)/test_set$avg_IFR)*100, na.rm = TRUE)
      if(mae < min(results$MAE)) {
        current_best = paste(current_model, "+", e_factor)
        results = rbind(results, data.frame(Model = current_best, MAE = mae, MAPE = mape))
        improved = TRUE
      }
    }
    current_model = current_best
  }
  return(current_model)
}
# Bestimmung des besten Modells
best_model = fw_selection(best_bivariate$Model, top5_correlations$Factor)
print(paste("Best Model: ", best_model))
```

```
## [1] "Best Model:  PoliticalStability +  UnskilledLaborAvailability + Inflation"
```

```
## Erstellung und Bewertung finales Modells
final_formula = as.formula(paste("avg_IFR~", best_model))
best_model = lm(final_formula, data = train_set)
final_predictions = predict(best_model, test_set)
final_mae = mean(abs(test_set$avg_IFR - final_predictions), na.rm = TRUE)
final_mape = mean(abs((test_set$avg_IFR - final_predictions)/test_set$avg_IFR)*100, na.rm = T
RUE)
print(paste("Best model with einem MAE von ", final_mae, " und MAPE von ", final_mape))
```

```
## [1] "Best model with einem MAE von  0.0110114497845945  und MAPE von  1.3563772602843"
```

```
# Überprüfung auf Overfitting
train_predictions = predict(best_model, train_set)
train_mae = mean(abs(train_set$avg_IFR - train_predictions), na.rm = TRUE)
train_mape = mean(abs((train_set$avg_IFR - train_predictions)/train_set$avg_IFR)*100, na.rm =
TRUE)
print(paste("Overfitting-Prüfung: MAE ist ", train_mae, " und MAPE ist ", train_mape))
```

```
## [1] "Overfitting-Prüfung: MAE ist  0.0116611682893642  und MAPE ist  1.41688223397697"
```

11. Bewerten Sie ihr Modell quantitativ im Vergleich mit der Baseline. Bewertungsrelevant: Output, Kommentar.

```
## Ausgabe der Modellbewertung
evaluation = rbind(evaluation, data.frame(Model = "Best Model", MAE = final_mae, MAPE = final
_mape, R2 = summary(best_model)$r.squared, adj_R2 = summary(best_model)$adj.r.squared))
evaluation
```

```
##        Model        MAE     MAPE        R2    adj_R2
## 1   Baseline 0.02812247 3.372565 0.0000000 0.0000000
## 2 Best Model 0.01101145 1.356377 0.8440053 0.8420053
```

```
## Bewertung:
# Das beste Modell hat einen deutlich niedrigeren MAE (0.0011 im Vergleich zu 0.0281), was be
deutet, dass beim besten Modell die Abweichungen zwischen den tatsächlichen und vorhergesagte
n Werten kleiner ist als bei der Baseline.
# Das beste Modell hat einen viel kleineren MAPE (1.3564 im Vergleich zu 3.3726), was zeigt,
dass das Modell die Vorhersagen prozentual genauer trifft.
# Ein höheres R2 von dem Modell (0.844 vs 0) weist darauf hin, dass es offensichtlich mehr Va
riabilität in den Daten erklärt.
# Das adjustierte R² von dem Modell beträgt 0.8420, was nahe am R2 liegt und zeigt, dass das
Modell auch nach Berücksichtigung der Anzahl der Prädiktoren robust bleibt. Währendessen ist
das adjustierte R2 bei der Baseline auch 0.
```

12. Ihre Chefin kommt auf der Firmenfeier zu Ihnen und schlägt Ihnen eine Wette vor. Sie sagt: “Ich wette mit Ihnen, dass die durchschnittliche IFR des oben betrachteten WH-DL im April 2024 in Japan höher sein wird, als in Shanghai. Sollte dies nicht der Fall sein, gebe ich Ihnen 400 Euro. Habe ich jedoch Recht, müssen Sie mir die 400 Euro geben.” Sollten Sie die Wette eingehen? Bewertungsrelevant: Output, Kommentar. Entscheidung

```
# Extraktion der externen Faktoren des besten Modells
gifter_topfactors = gifter_externals %>% select(region, Periode, avg_IFR, PoliticalStability,
UnskilledLaborAvailability, Inflation)

# Vorhersage der IFR mit dem besten Modell
gifter_topfactors$predicted_IFR <- predict(best_model, newdata = gifter_topfactors)

# Extrakation der Vorhersagen für April 2024
gifter_topfactors_042024 = subset(gifter_topfactors, Periode == "202404")


# Extraktion der Vorhersagen für Japan und Shanghai
gifter_japan_042024 <- gifter_topfactors_042024$predicted_IFR[gifter_topfactors_042024$region
== "Japan"]
gifter_shangh_042024 <- gifter_topfactors_042024$predicted_IFR[gifter_topfactors_042024$regio
n == "Shangh"]
# Ausgabe der IFR im April 2024 in beiden Regionen
print(paste("IFR in Japan im April 2024 wäre ", gifter_japan_042024))
```

```
## [1] "IFR in Japan im April 2024 wäre  0.821751212657736"
```

```
print(paste("IFR in Shanghai im April 2024 wäre ", gifter_shangh_042024))
```

```
## [1] "IFR in Shanghai im April 2024 wäre  0.790581967941824"
```

```
# Im April 2024 scheint die IFR in Japan höher zu sein als in Shanghai. Daher hätte die Chefi
n Recht, weswegen ich die Wette nicht eingehen sollte.
```

13. Ihr Regressionsmodell soll im kommenden Jahr implementiert und langfristig in die Unternehmensprozesse in- tegriert werden. Beschreiben Sie, welche Nutzer und Prozesse davon profitieren könnten und in welcher Form die Lösung bereitgestellt werden könnte. Nehmen Sie ausserdem ausführlich zur Phase der Datenbeschaffung Stellung. Bewertungsrelevant: Kommentar.

Nutzer und Prozesse, die davon profitieren könnten
 # Das Modell kann dabei helfen, die Nachfrage besser vorherzusagen. Damit werden Logistikmanager und -analysten bei der Planung der Lagerbestände und Lieferungen zur Verbesserung der OTD und IFR.
 # Genauere Vorhersagen der IFR können Finanzanalysten und Controller bei der Finanzplanung und Vermeidung unnötiger Kosten.
 # Die Vorhersage der IFR kann die Vertriebsmanager bei der Planung von Verkaufsaktionen und Marketingkampagnen, um zu gewährleisten, dass genügend Produkte verfügbar sind.
 # Ein stabiler Lagerbestand und punktliche Lieferungen führen zu höherer Kundenzufriedenheit und weniger Beschwerden, wovon die Kundenservicemitarbeiter profitieren.

Form der Lösungsbereitstellung
 # Ein interaktives Dashboard, das Echtzeit-Analysen und Vorhersagen anzeigt, wo Nutzer wichtige Kennzahlen wie OTD und IFR sowie externe Faktoren sehen.
 # Eine API, die Vorhersagen bereitstellt, damit Mitarbeiter die Vorhersagen betrachten und damit optimale Entscheidungen treffen können.
 # Regelmäßige Berichte (z.B per E-mail), die an Mitarbeiter gesendet werden, um sie über Vorhersagen und Trends zu informieren.

Datenbeschaffung
 # Identifkation der Datenquellen, einschließlich internen Datenquellen (Lagerbestände, Lieferzeiten, Bestellungen usw.) und externen Datenquellen (politische Stabilität, Wetter usw.)
 # Datenintegration und -bereinigung: Integration der Datenquellen in ein Data Warehouse und Bereinigung der Daten.
 # Datenverarbeitung und -vorbereitung: Transformation der Daten in ein geeignetes Format für die Modellierung und ggf. Feature Engineering
 # Datenaktualisierung und -überwachung: Regelmäßige Aktualisierung der Daten sowie die Überwachung der Datenqualität zur Problemerkennung