Predictive Analytics in der Praxis -
Zeitreihenanalyse mit Oracle R Enterprise

Marco Nätlitz
areto consulting gmbh
Köln

Schlüsselworte
Predictive Analytics, Time Series Analysis, Oracle R Enterprise, Oracle Advanced Analytics

Predictive Analytics – der Blick in die Zukunft

Das Thema Predictive Analytics (PA) ist zurzeit in aller Munde und schafft es immer häufiger auf die Titelseiten der Fachliteratur. Die letzte Finanzkrise, Rückrufaktionen bei Automobilherstellern oder das Winterchaos bei der Bahn sind nur ein paar Beispiele für Ereignisse, die die betroffenen Unternehmen überrascht und in Gefahr gebracht haben. Aus diesem Grund sind heutige Unternehmen verstärkt an Zukunftsprognose und Datenanalysen interessiert, die solche Gefahren erkennen und abwehren.

PA ermöglicht diesen Blick in die Zukunft durch vielfältige Methoden und Ansätze. Genauso vielfältig wie die Bandbreite an Verfahren sind allerdings auch die Definitionen von PA, die Mertens und Rässler (2012) umfassend gegenüberstellen und PA abschließend wie folgt definieren[1]: „Predictive Analytics bezeichnet (…) eine Form der Aufbereitung und Auswertung von Daten zur zukunftsorientierten Entscheidungsunterstützung auf allen Unternehmensebenen. Mithilfe von Prognosewerten wird das Data Mining erweitert, um Informationen über die Zukunft zur Entscheidungsfindung zur Verfügung zu stellen." Im Folgenden wird eine beispielhafte und praxisnahe Anwendung einer Zeitreihenanalyse, einem Teilgebiet der PA, sowohl mit R als auch mit Oracle R Enterprise (ORE), dargestellt.

Definition von Zeitreihen

Eines der zahlreichen Verfahren der PA ist die Zeitreihenanalyse. Gegenstand dieser Analysen sind sogenannte Zeitreihen, die eine Folge von zeitlich geordneten Beobachtungswerten beschreiben.[2] In der Regel wird angenommen, dass diese Beobachtungen diskret und äquidistant sind, also in gleichen zeitlichen Abständen (z.B. täglich, wöchentlich oder monatlich) beobachtet werden können. Als Beispiele für Zeitreihen sind Börsenkurse, Temperaturverläufe aber auch Umsatz- und Absatzzahlen zu nennen. Wie viele Beobachtungswerte die Zeitreihe enthält ist Definitionssache und natürlich abhängig von der Datenlage. Beispielsweise liegen für den DAX ausführliche und historische Informationen vor, wohingegen der Umsatz eines drei Monate jungen Unternehmens aufgrund fehlender historischer Daten nicht hinreichend analysiert werden kann.

Die Zerlegung von Zeitreihen

Besonders bei ökonomischen Zeitreihen findet man langfristige Veränderungen, die Mustern gleichen. So ist z.B. der Stromverbrauch im Winter größer als im Sommer, ebenso steigt der Absatz im Einzelhandel zur Weihnachtszeit. Zur Beschreibung solcher Muster wurde das Komponentenmodell entwickelt, das diesen Gegebenheiten Rechnung trägt. Es zerlegt eine Zeitreihe in folgende Bestandteile oder Komponenten:[3]

---

[1] Vgl. (Mertens und Rässler 2012), S. 522.
[2] Vgl. zu diesem Absatz (Stier 2013), S. 1ff.
[3] Vgl. (Schlittgen und Streitberg 2001), S. 9f.

- **Trend** – eine langfristige systematische Veränderung des mittleren Niveaus der Zeitreihe.
- **Saison** – eine regelmäßige Schwankung, die sich relativ unverändert z.B. jährlich, monatlich oder in bestimmten zeitlichen Intervallen wiederholt.
- **Rest** – der nicht zu erklärende Teil der Zeitreihe, der z.B. durch Störungen oder durch Zufall zustande kommt.

Als Beispiel soll nun die Zeitreihe des Umsatzes eines Einzelhändlers dienen, die in Abb. 1 dargestellt wird. Der oberste Graph stellt die tatsächliche Zeitreihe dar. In den drei Graphen darunter werden die einzelnen Komponenten Trend, Saison und Rest dargestellt. Stellt man sich eine Zeitreihe als Addition vor, so stellt jeder Punkt des obersten Graphs die Summe aus den drei darunterliegenden Graphen dar.

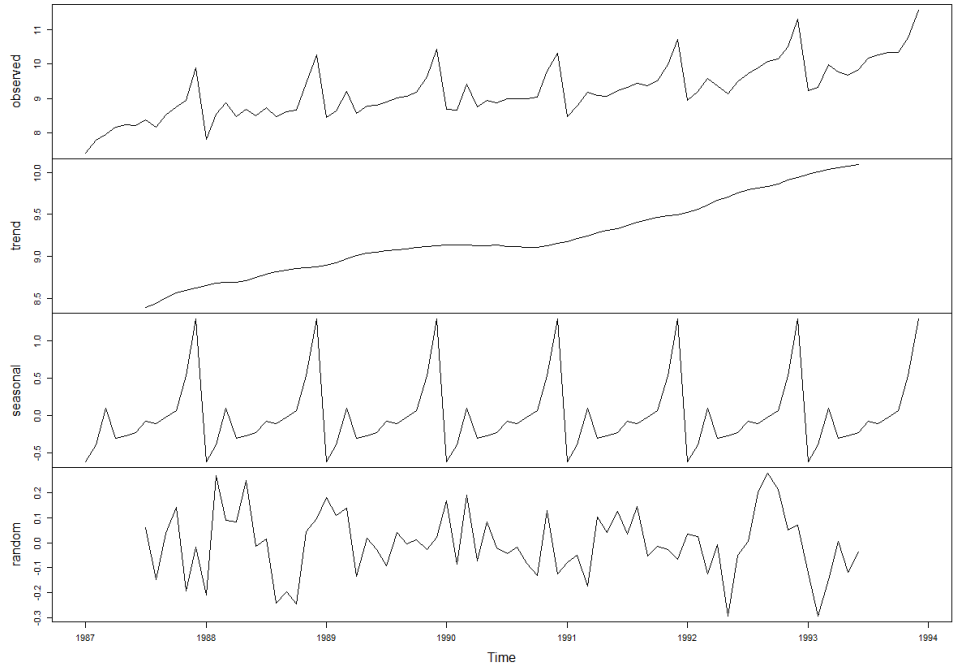

Abb. 1: Dekomposition des Umsatzes eines Einzelhändlers[4]

Anhand der Zerlegung einer Zeitreihe lassen sich die Eigenschaften der Zeitreihe ablesen: ein nahezu linear steigender Trend (zweiter Graph von oben) und eine Saisonkomponente (dritter Graph von oben) mit Peaks im März und im Dezember sind Abb. 1 zu entnehmen. Der Anteil der Zeitreihe, der nicht zur Trend- oder Saisonkomponente zugeordnet werden kann, findet sich im Rest wieder. Je kleiner die Rest-Komponente einer Zerlegung ist, desto besser lässt sich die Zeitreihe beschreiben. Enthält die Rest-Komponente hingegen Muster, so ist die Trend- bzw. Saisonkomponente nicht adäquat berechnet worden.

Zeitreihenanalyse mit R

Die Statistikumgebung R[5] biete zahlreiche Möglichkeiten zur Zeitreihenanalyse.[6] Bevor auf eine umfangreiche Analyse mit ORE eingegangen wird, soll zunächst kurz vorgestellt werden, wie eine Zeitreihenanalyse mit den Bordmitteln von R angegangen werden kann. Als Beispieldatensatz dienen die Umsatzzahlen des Einzelhändlers, die bereits in Abb. 1 dargestellt und beschrieben wurden.

[4] Daten via http://robjhyndman.com/tsdldata/data/fancy.dat.
[5] Siehe https://www.r-project.org/.
[6] Hier ist besonders das R-Paket „forecast“ (https://cran.r-project.org/web/packages/forecast/index.html) und „fpp“ (https://cran.r-project.org/web/packages/fpp/index.html) zu nennen.

Mit folgendem R-Skript wird die Zerlegung in die Komponenten der Zeitreihe aus Abb. 1 erzeugt:

```
fancy = scan("http://robjhyndman.com/tsdldata/data/fancy.dat")
fancyts = ts(fancy, frequency=12, start=c(1987,1)) # Zeitreihenobjekt
fancylog = log(fancy) # Normalisierung durch Logarithmierung
fancylogts = ts(fancylog, frequency=12, start=c(1987,1))
dec_fancy = decompose(fancylogts) # Zerlegung der Komponenten
plot(dec_fancy) # Plot der Dekomposition
plot.ts(fancylogts, xlab="Time", ylab="Log(Sales $)") # Plot der norm. Zeitreihe
```

Die Umsatzzahlen des Einzelhändlers unterliegen starken Schwankungen, sodass der Umsatz zur Weihnachtszeit signifikant höher ist als in den anderen Monaten des Geschäftsjahres. Damit diese Schwankungen keinen zu starken Einfluss auf ein statistisches Modell haben, müssen diese normalisiert werden. Zur Normalisierung wurde in diesem Beispiel der Logarithmus gewählt.[7]

Zur Prognose des nächsten Geschäftsjahres wird das Verfahren nach Holt/Winters gewählt.[8] Das Verfahren beruht auf der Tatsache, dass künftige Beobachtungen eher der jüngeren Vergangenheit gleichen. So kann durch drei Parameter der Einfluss der einzelnen Komponenten einer Zeitreihe auf die Prognose beeinflusst werden. Die rekursive Definition der Funktion sorgt dafür, dass das Gewicht der Glättungsparameter entlang der Zeitreihe in Richtung Vergangenheit abnimmt. So hat beispielsweise der Trend zum Anfang der Zeitreihe eine kleinere Bedeutung für die Prognose als der Trend zum Ende der Zeitreihe. Das Verfahren nach Holt/Winters eignet sich besonders für die Prognose von Zeitreihen mit einer Trend- und Saisonkomponente, weshalb es ein adäquates Prognoseverfahren die Zeitreihe zum Umsatz des Einzelhändlers darstellt. Die mathematischen Details des Verfahrens nach Holt/Winters würde den Umfang dieses Manuskriptes überschreiten, weshalb an dieser Stelle auf die anschaulichen Erklärungen von Hyndman verwiesen wird.

In R wird das Verfahren nach Holt/Winters mit der Funktion „HoltWinters“ implementiert. Die optimale Belegung der Glättungsparameter für die einzelnen Komponenten wählt die Funktion dabei selbst. Mit Hilfe der Funktion können Zeitreihen auf simple Art und Weise prognostiziert werden, was folgender R-Code veranschaulicht:

```
fancylogts_train = ts(fancylog, frequency=12, start=c(1987,1), end=c(1992,12))
fit1 = HoltWinters(fancylogts) # Analyse nach Holt/Winters
fc1 = forecast.HoltWinters(fit1, h=12) # Prognose, 12 Monate
plot.forecast(fc1) # Plot mit Legende…
```

Die Zeitreihe enthält Daten von Januar 1987 bis Dezember 1993. Für die weitere Analyse wurde das letzte Jahr der Zeitreihe abgeschnitten und als Verifikation für die Prognose herangezogen. Mit Hilfe des Verfahrens nach Holt/Winters wurde anschließend das abgeschnittene Jahr 1993 prognostiziert. Abb. 2 stellt das Ergebnis der Prognose als blaue Linie dar. Die rote Linie im Graph stellt die originalen Zahlen (Normalisiert durch Logarithmieren) dar. Dem Graph in Abb. 2 ist zu entnehmen, dass die Schwankungen der Umsatzzahlen adäquat mit dem gewählten Verfahren vorherzusagen sind und sich die Prognose einer hohen Genauigkeit erfreut:[9]

---

[7] Vgl. (Wikipedia 2015).
[8] Vgl. zu diesem Abschnitt (Hyndman und Athanasopoulos 2015), Kapitel 7.
[9] Coghlan beschreibt ausführlich, wie die Güte einer Prognose einer Zeitreihe statistisch beurteilt werden kann. Vgl. (Coghlan 2015).

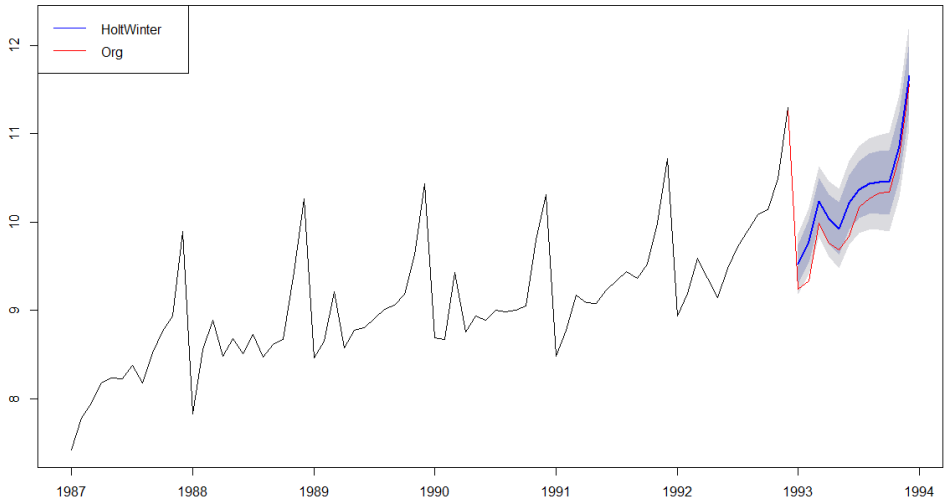

Abb. 2: Prognose des Umsatzes nach Holt/Winter

Vorstellung von Oracle R Enterprise (ORE)

Die bisherigen Ausführungen haben die Zeitreihenanalyse mit einer klassischen R-Instanz beschrieben, die in der Regel auf einem Client-Computer installiert ist. Gerade im Enterprise-Kontext reicht die Leistung solcher Client-Computer nicht aus, weshalb auf leistungsfähigere Server umgestiegen werden muss. Darüber hinaus steigt die Anzahl der Daten und die Komplexität der Berechnungen im Zeitalter von Big Data stetig, weshalb eine performante Datenbank oder ein Data Warehouse als unumgänglich für heutige Unternehmen gilt. Besonders in einem Data Warehouse liegen Daten sehr häufig mit einer zeitlichen Dimension vor, sodass die Zeitreihenanalyse oder -Prognose eine hohe Bedeutung beigemessen wird.

Oracle adressiert oben geschildertes Problem mit seiner Lösung Oracle R Enterprise (ORE) als Bestandteil der Oracle Advanced Analytics Option der Oracle Database (Enterprise Edition).[10] Architektonisch erweitert Oracle die Datenbank so, dass performante statistische Analyse integriert ausgeführt werden können. Als Client dient dabei eine R-Instanz, die selbst keine Berechnungen durchführt, sondern lediglich die Berechnungsergebnisse darstellt bzw. die Kommandos an die Datenbank koordiniert. Abb. 3 stellt den Architektonischen Aufbau von ORE dar.

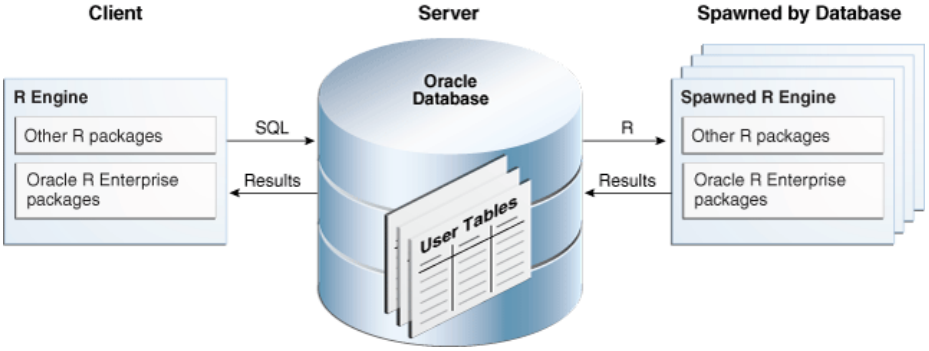

Abb. 3: Architektur von Oracle R Enterprise

Technisch besteht ORE aus drei Komponenten, die in Verbindung mit Abb. 3 im Folgenden erläutert werden:

[10] Vgl. zu diesem Abschnitt (Oracle 2016).

- Zunächst ist der ORE Transparency Layer (TL) zu nennen. Der TL ist eine Sammlung von R-Paketen, die alle grundlegenden Datentypen von R auf äquivalente Datentypen der Datenbank mappen. Durch den TL arbeitet der Benutzer, obwohl er R-Code auf seinem Laptop eintippt, direkt auf der Datenbank. Die TL ermöglicht so die Arbeit mit großen Datenmengen und komplexen Berechnungen.
- Weiter enthält ORE eine Statistics Engine (SE). Die SE ist eine Sammlung von statistischen Funktionen, die als R-Paket in der R-Instanz geladen werden und mit den Datentypen aus dem TL zusammenarbeiten. Das Besondere der SE ist, dass die statistischen Funktionen direkt in der Datenbank ausgeführt werden und ebenfalls keine Ressourcen des Client-Computers belasten. So sind beispielsweise lineare Modelle direkt auf der Datenbank verfügbar.
- Zuletzt bietet ORE die Funktion der Embedded R. Die Datenbank übernimmt die Kontrolle der Verarbeitung der R-Skripte, die der Benutzer mit TL und SE an die Datenbank überträgt. ORE startet dafür nach Bedarf R-Instanzen, sodass z.B. auch parallel Ausführung komplexer Aufgaben möglich wird. Durch Embedded R ist kein Datentransfer aus der Datenbank erforderlich, da die R-Skripte direkt die Tabellen, Views oder andere Objekte der Datenbank als Ressource verwenden.

Zeitreihenanalyse mit Oracle R Enterprise (ORE)
Nach der Vorstellung der Funktionsweise von ORE soll nun die bereits durchgeführte Zeitreihenanalyse mit ORE gezeigt werden. Zur Realisierung einer Prognose der Umsatzzahlen des Einzelhändlers (analog zu den bisherigen Beispielen) wurden die Geschäftszahlen in eine Tabelle („TS_FANCY) auf der Datenbank abgelegt. Diese Tabelle enthält zwei Spalten: in der ersten Spalte ein Datum und der zweiten den Umsatz zum jeweiligen Datum. ORE bietet die Möglichkeit R-Skripte auf ganze Tabellen der Datenbank mit der Funktion „ore.tableApply“ anzuwenden:

```
var_ore1 <- ore.tableApply(TS_FANCY[,c("DTS","SALES")], ore.connect = TRUE,
function(dat) {
  library(forecast)
  salests = ts(dat$SALES, start=c(1987,1), frequency=12) # Zeitreihenobjekt
  hw_fit1 = HoltWinters(salests) # Prognosemodell
  for_fit1 = predict(hw_fit1, n.ahead=12) # Berechnung der Zukunftswerte
  as.data.frame(for_fit1)
});
```

Der oben dargestellt R-Code erzeugt zunächst ein Zeitreihenobjekt und wendet im Anschluss das Verfahren nach Holt/Winters auf dieses Zeitreihenobjekt an. Im Anschluss werden die nächsten 12 Monate der Zeitreihe mit dem Prognosemodell nach Holt/Winters berechnet. Eine Rückgabe in ORE bedeutet konkret die Materialisierung der R-Objekte in der Datenbank, die mit der Referenz „var_ore1“ im weiteren Verlauf auf Client-Seite abgerufen werden können. Soll nun auf Client-Seite die Berechnung in R fortgeführt werden, so können die mit „ore.tableApply“ generierten Berechnungsergebnisse wie folgt zum Client übertragen werden:

```
lcl_var_ore1 = ore.pull(var_ore1) # Hole die Rückgabe als data.frame
summary(lcl_var_ore1[[1]])
```

Integration von Oracle R Enterprise (ORE)
Neben der Ad-hoc-Analyse in R können die Methoden und Verfahren der Predictive Analytics direkt in die Datenbank integriert werden. ORE stellt beispielsweise die PL/SQL-Funktionen

„rqScriptCreate“ zum Speichern und „rqTableEval“ zum Aufruf bereit. Die Speicherung des R-Skriptes zur Umsatzprognose wird wie folgt realisiert:

```
begin
   sys.rqScriptCreate('salesForecastFunc', 'function(dat) {
         library(forecast)
         salests = ts(dat$SALES, start=c(1987,1), frequency=12) # Zeitreihenobjekt
         hw_fit1 = HoltWinters(salests) # Prognosemodell
         for_fit1 = predict(hw_fit1, n.ahead=12) # Berechnung der Zukunftswerte
         as.data.frame(for_fit1)
   }');
end;
/
```

Nun ist das R-Skript in der Datenbank gespeichert und kann mit SQL aufgerufen werden, sodass auch auf diesem Wege der TL und die SE von ORE Verwendung findet:

```
select * from table(
   rqTableEval(
    cursor(select dts, sales from ts_fancy),
    null,
    null,
    'salesForecastFunc')
);
```

Die Rückgabe dieser Query enthält 12 Zeilen und beschreibt die prognostizierten Umsätze der kommenden 12 Monate des beispielhaften Einzelhändlers. Weiter kann diese Query als View in der Datenbank hinterlegt werden und so im Rahmen von Reporting- oder ETL-Prozessen integriert werden.

Fazit

Dieses Dokument zeigt in Ausschnitten die theoretischen Grundlagen der Zeitreihenanalyse und wie diese sowohl mit R als auch mit ORE durchgeführt werden kann. Es wurde gezeigt, dass ORE eine elegante Möglichkeit für statistische Berechnungen auf großen Datenmengen darstellt, die die verfügbaren Ressourcen optimal ausnutzt. Darüber hinaus können R-Skripte in der Datenbank gespeichert und weiterverarbeitet werden, sodass diese entlang der Wertschöpfungskette in BI-Projekten Verwendung finden können.

Neben der Zeitreihenanalyse deckt ORE die gesamte Bandbreite der Verfahren der Predictive Analytics ab, weshalb ORE ein fundiertes Werkzeug für Datenanalysen und dessen Prognosen darstellt.

Literaturverzeichnis

Coghlan, Avril. „Using R for Time Series Analysis.“ 2015. https://a-little-book-of-r-for-time-series.readthedocs.org/en/latest/src/timeseries.html#holt-winters-exponential-smoothing (Zugriff am 22. 09 2015).

Eckstein, Peter P. *Repetitorium Statistik: Deskriptive Statistik - Stochastik - Induktive Statistik.* 4. Springer, 2013.

Hyndman, Rob J., und George Athanasopoulos. „Forecasting: Principles and Practice.“ otexts.org, 01. 09 2015.

Kreiss, Jens-Peter, und Georg Neuhaus. *Einführung in die Zeitreihenanalyse: Statistik und ihre Anwendungen.* Berlin: Springer, 2006.

Mertens, Peter, und Susanne Rässler. *Prognoserechnung.* Bd. 7. Berlin: Springer, 2012.

Oracle. *Overview of Oracle R Enterprise.* 16. 09 2016. http://docs.oracle.com/cd/E36939_01/doc.13/e36761/intro.htm#OREUG109.

Schlittgen, Rainer, und Bernd H.J. Streitberg. *Zeitreihenanalyse: Lehr- und Handbücher der Statistik.* 9. Oldenbourg Verlag, 2001.

Schneider, Matti, und Sebastian Mentemeier. *Zeitreihenanalyse mit R.* 2010. http://wwwmath.uni-muenster.de/statistik/lehre/SS10/BlockprakZeit/Zeitreihenanalyse.pdf.

Stier, Winfried. *Methoden der Zeitreihenanalyse.* Berlin: Springer, 2013.

Wikipedia. *Data transformation (statistics).* 16. 09 2015. https://en.wikipedia.org/wiki/Data_transformation_%28statistics%29.

Kontaktadresse:
Marco Nätlitz
areto consulting gmbh
Carlswerk, Gebäude „Labor 1.7“
Schanzenstraße 6-20
51063 Köln

Telefon: +49 221 66 95 75-0
Telefax: +49 221 66 95 75-99
Email: Marco.Naetlitz@areto-consulting.de