Vorhersagen und Prognosen – SQL Erweiterungen in der 12c

Detlef E. Schröder
Oracle Deutschland B.V. & Co KG
Düsseldorf

Schlüsselworte
Data Mining, Oracle 12c, SQL, Vorhersage, Anomalie, Kluster, Wahrscheinlichkeit, Einflussfaktoren

Einleitung
Mit der 12c Datenbank hat Oracle auch weider die Möglichkeiten des SQL erweitert. Im Rahmen der Oracle Advanced Analytics Option, der Data Mining Erweitereung der Datenbank, gibt es nun auch die Möglichkeit die Vorhersage direkt im SQL ab zu fragen, ohne vorher ein Modell errechnen zu müssen.

Vorhersagen und Prognosen – SQL Erweiterungen in der 12c

In der Data Mining Option der Datenbank sind acht verschiedene Modelle möglich zu erstellen mit 14 verschiedenen Algorithmen. Diese lassen sich mit dem Oracle Data Miner, der Erweitereung des SQL Developers als UI, bedienen. Hier hat der geneigte Statistiker alle Freiheiten die Algorithmen gezielt mit Parametern zu versehen und die Ausführungen zu steuern. Die hier errechneten Modelle ließen sich dann schon immer im SQL abfragen und für online Mining verwenden.

Nun hat Oracle aber ein weiteres Einsatzfeld geschaffen. Ohne erst Modelle zu erzeugen und sich gedanken über die vielen Aspekte des Data Mining zu machen, kann ich die default Einstellungen verwenden und einfach in meinen normalen Abfragen, vorhersagen einbauen, die dann „on the fly“ berechnet werden.

Auf der Basis des in den verschiedenen Seminaren verwendeten Datenmodells werden im Folgenden diese Möglichkeiten vorgestellt und in ihrer Variabilität beschrieben.
Als erstes wird die Klusterbildung vorgestellt.
Die Kunden der D_Kunde Tabelle sollen anhand der verschiedenen Spalten in 6 Kluster eingeteilt werden und die Kunden des Klusters 6 mit der Wahrscheindlichkeit der Zugehörigkeit zum Kluster 6 mit ausgegeben werden.

```
with vorhersage as
(select kundennr,
       CLUSTER_ID(INTO 6 USING * ) OVER (order by kundennr)
Kluster,
       CLUSTER_PROBABILITY(INTO 6 USING * ) OVER (order by
kundennr) Wahrscheinlichkeit
from dwh.d_kunde)
select * from vorhersage
where Kluster = 6 and Wahrscheinlichkeit > 0.9;
```

Die Beiden Data Mining Funktionen in dieser Abfrage CLUSTER_ID und CLUSTER_PROBABILITY sind von der Syntax her, wie iene analytoische Funktion aufgebaut.

Zuerst wir angegeben, was gerechnet werden soll und in dem OVER () Bereich die Möglichkeiten, ein ORDER BY oder auch ein PARTITIOM BY an zu geben.

Mit dem INTO X wird die Zahl der Kluster angegeben und mit USING, die Spalten, die für die Vorhersage verwendet werden sollen. * bedeutet alle Spalten, die möglich sind.
Eine weitere Spezifikation von Parametern oder Verhaltensweisen der zugrunde liegenden Allgorithmen ist nicht möglich, ergibt aber eine schnelle und einfache Einbindung in den „normalen“ SQL Betrieb.

Mit diesen Funktionen lassen sich auch Materialized Views oder andere Objekte in der DataMart Schicht eines Warehouses einfach mit vielen weiteren Informationen versehen und diese dann dem Anwender anbieten.

Neben der Klusterbildung lassen sich auch Werte Vorhersagen. Dieses können alphanumerische oder numerische Informationen sein, die sich aus den weiteren Spalten prognostizieren lasen.
Als Beispiel aus dem Starschema der DWH Seminare wieder die Kundentabelle und eine View, die die Umsätze aufsummiert, die dann prognostiziert werden können.

```
select kundennr, bildungs_nr,
       PREDICTION ( FOR einkommensgruppe using *) over( partition by
       bildungs_nr order by kundennr) Vorhersage
from dwh.d_kunde;

CREATE OR REPLACE VIEW UMSATZ_PRO_KUNDE AS
  select kundennr, jahr_nummer, sum(umsatz) umsatz, berufsgruppen_nr,
plz
  from dwh.f_umsatz u, dwh.d_kunde k, dwh.D_zeit z
  where u.kunden_id = k.kunden_id
  and   u.zeit_id = z.zeit_id
  group by kundennr, jahr_nummer, berufsgruppen_nr, plz;

select kundennr,
       umsatz,
       prediction ( FOR umsatz using *) over() Vorhersage
from umsatz_pro_kunde;
```

Auch hier ist die Data Mining Funmktion, wie eine analytische Funktion aufgebaut. Die PREDICTION Funktion, wie die PREDICTION_PROBABILITY Funktion, erwarten zuerst die Angabe, was prognostiziert werden soll, hier FOR umsatz oder einkommensgruppe, gefolgt von der Angabe der zu verwendenden Spalten. Die Partitionierungs-option mit over () bietet weider die schon beschreibenen Optionen. Die Einkommensgruppe ist heir ein varchar Feld mit wenigen Ausprägungen, das umsatzfeld eine number Spalte mit unendlichen Möglichkeiten. Es werden also unterschiedliche Prognosemodelle verwendet, die abernicht explizit festgelegt werden müssen, sondern transparent gewählt werden.

Eine besondere prediction Funktion ist die, die Besonderheiten und Auffälligkeiten in den Daten herausfindet. Dann wird die Funktion nicht mit FOR X sondern mit OF ANOMALY angegeben und somit dann die Ausgabe von 0 oder 1 erzeugt. Diese können dann auch wieder mit Wahrscheinlichkeiten ausgegeben werden. Als Beispiel nehmen wir den View von vorher und sehen nach, ob bestimmte Kundenbestellungen auffällig sind.

```
WITH vorhersage as
(select kundennr,
       prediction ( OF ANOMALY using *) over() Vorhersage,
       prediction_probability( OF ANOMALY using * ) over ()
       Wahrscheinlichkeit
from umsatz_pro_kunde)
select * from vorhersage
WHERE Vorhersage = 0 AND Wahrscheinlichkeit > 0.5;
```

Somit ergeben sich auch hier weitere Möglichkeiten, die Datenqualität, hier im Sinne von Korrektheit, zu erweitern. Missbrauch oder Verfahrensfehler können auch wieder sehr einfach dem Endanwender der Daten zur Verfügung gestellt werden, ohne die Data Mining Allgorithmen im Detail verstehen und bedienen zu können oder müssen.

Als vierte und letzte Funktion in der Version 12.1 der Datenbank steht dieErmittlung der relevanten Einflussfaktoren zur Verfügung. Hiermit lassen sich die Spalten ermitteln, die für die Zielgröße den größten Einfluss besitzen.

Dazu lassen sich die möglichen Größen und die Anzahl der zu ermittelden Größen angeben, so wie der konkrete Faktor, der ermittelt wurde.

```
WITH vorhersage as
(select kundennr,
      CAST(FEATURE_SET( INTO 4 USING * ) over () as
ODMR_FEATURE_SET_NUMVD)
      as Einflussgroessen,
      FEATURE_VALUE( INTO 4 USING * ) over () as Wert,
      FEATURE_ID( INTO 4 USING * ) over () as ID
from dwh.d_kunde)
select * from vorhersage
where Wert > 2;
```

Hierbei gilt es zu beachten, das das FEATURE_SET einen besondern Datentyp besitzt und daher mit einem CAST versehen werden muss. Mit INTO X wird die Anzahl der zu suchen Einflussgrössen angegeben und mit USING und OVER wieder die bekannten Parameter. Die Funktioenn FEATURE_VALUE und FEATURE_ID geben die größte Einflussgröße aus.

Dies kann auch wieder für einige Anwender interessant sien, die mit diesen Daten weiter arbeiten und sich dann nicht mehr auf alles konzentrieren müssen, sondern schon die entsprechende Auswahl vor Augen haben, die sich wahrscheinlich lohnt genauer zu analysieren.

Mit diesen vier data mining Funktionen hat Oracle den Anfang gemacht, SQL weiter in die Richtung vorhersage zu entwickeln und die vielen Optionen sehr einfach mittels SQL zur Verfügung zu stellen. Jedes Werkzeug, das SQL spricht kann nun, vorrausgesetzt die Oracle Advanced Analytics Option ist lizensiert, dieses SQL nutzen und seinen Umfang an Analytik und Data Mining erweitern.

Kontaktadresse:
Detlef E Schröder
Oracle Deutschland B.V. & Co KG
Hamborner Straße 51
D-40472 Düsseldorf

Telefon: +49 (040) 8 90 91 – 4 23
E-Mail detlef.e.schroeder@oracle.com
Internet: www.oracledwh.de