Anwendung eines SAS/STAT-Modells ohne SAS/STAT in einem CRM-Projekt

Timm Euler
viadee GmbH
Anton-Bruchausen-Str. 8
48147 Münster
Timm.Euler@viadee.de

Tobias Otte
viadee GmbH
Anton-Bruchausen-Str. 8
48147 Münster
Tobias.Otte@viadee.de

Zusammenfassung

Für eine Analyse der Kundenloyalität mit Data Mining bei einem Finanzdienstleister stand SAS/STAT nur während der Modellbildung zur Verfügung, aber nicht zur Modellanwendung. Daher wurde ein logistisches Regressionsmodell mit SAS/STAT erzeugt und dann mit SAS/BASE ausgelesen und zur Vorhersage eingesetzt. Dieser Beitrag berichtet vom Projekthintergrund, der technischen Umsetzung und der Evaluation.

Schlüsselwörter: Data Mining, Logistische Regression, PROC LOGISTIC, SAS/STAT, CRM, Kundenloyalitätsanalyse, Kündigungsanalyse, Churn

1 Projekthintergrund

Ein Finanzdienstleister möchte seine Kunden anhand einer prognostizierten Wahrscheinlichkeit, dass der Kunde seinen Vertrag mit dem Dienstleister in naher Zukunft kündigt, in verschiedene Segmente (Kundengruppen) einteilen. Die Segmente spiegeln die vermutete Loyalität des Kunden wider: im loyalsten Segment finden sich die Kunden mit der geringsten Kündigungswahrscheinlichkeit und umgekehrt.

Die Verwendung einer binären Klassifikation mittels Data Mining ist ein naheliegender Ansatz zur Lösung dieser Aufgabe. Kundendaten aus der Vergangenheit werden dabei in zwei Klassen aufgeteilt: Kunden, die bereits gekündigt haben, und andere Kunden. Ein Data Mining-Modell wird zur Unterscheidung dieser beiden Klassen auf diesen Daten trainiert und anschließend auf aktuelle Kunden, die noch nicht gekündigt haben, angewendet. Das Modell kann dann vorhersagen, welche der aktuellen Kunden stärker kündigungsgefährdet sind als andere. Der Prozess hat also zwei wesentliche Phasen: Modellbildung und Modellanwendung.

Zur Modellierung von Wahrscheinlichkeiten eignet sich besonders die Methode der logistischen Regression. Im SAS/STAT-Paket findet sich die Prozedur PROC LOGISTIC, mit der ein logistisches Regressionsmodell leicht sowohl gebildet als auch angewandt werden kann, wie weiter unten gezeigt wird.

In diesem Projekt stand jedoch die Lizenz für SAS/STAT in der produktiven Umgebung, in der die Modellanwendung regelmäßig automatisiert stattfinden muss, nicht zur Verfügung. Daher wurde nur die Modellbildung mit PROC LOGISTIC durchgeführt, während einfacher SAS/BASE-Code zur Modellanwendung eingesetzt wurde. Dazu musste das von PROC LOGISTIC erzeugte Modell interpretiert und die Kündigungswahrscheinlichkeit direkt berechnet werden. Dieser Beitrag erläutert das Vorgehen.

2 Standardansatz mit PROC LOGISTIC

2.1 Modellbildung

Für diese klassische Data Mining-Aufgabe liegen die Daten, anhand derer das Modell trainiert wird, in einem einzigen SAS-Dataset vor, das eine Beobachtung (Zeile) für jeden Kunden enthält und die numerische Zielvariable (abhängige Variable, Label, Target) `churn` mit den Werten 0 oder 1 aufweist, wobei der Wert 1 Kunden markiert, die bereits gekündigt haben. Der folgende Prozeduraufruf, der nur möglich ist, wenn SAS/STAT lizensiert wurde, erstellt ein logistisches Regressionsmodell und speichert es im SAS-Dataset `regressionsModell` ab:

```
PROC LOGISTIC DATA     = daten
              OUTMODEL = regressionsModell;
    CLASS feld1 feld2 feld3;
    MODEL churn = feld4 feld5 feld6;
RUN;
```

Dabei sind die Trainingsdaten im SAS-Dataset `daten` enthalten und `feld1` bis `feld3` sind alphanumerische, `feld4` bis `feld6` numerische Variablen, die die Kunden näher beschreiben und die die Basis für die Mustererkennung bilden (sogenannte unabhängige Variablen). Aufgrund des `MODEL`-Statements wird `churn` als abhängige Variable betrachtet und das Regressionsmodell so gebildet, dass es die Abhängigkeit zwischen `churn` und den anderen Variablen möglichst gut abbildet.

2.2 Modellanwendung (Standardansatz)

Um das so gebildete Modell nun auf aktuelle Kunden zur Vorhersage einer Kündigungswahrscheinlichkeit anzuwenden, eignet sich der folgende erneute Aufruf von PROC LOGISTIC:

```
PROC LOGISTIC INMODEL = regressionsModell;
    SCORE DATA = neudaten
          OUT  = vorhersage;
RUN;
```

Dabei wird das Modell auf die Daten aus dem SAS-Dataset `neudaten`, das die Variable `churn` nicht enthalten muss (aber die Variablen `feld1` bis `feld6`), angewandt. Das SAS-Dataset `vorhersage` enthält dann die Variablen `i_churn` sowie `p_0` und `p_1`, wobei `p_0` die vorhergesagte Wahrscheinlichkeit angibt, dass die Zielvariable

`churn` für diesen Kunden den Wert 0 hat, und `p_1` entsprechend die Gegenwahrscheinlichkeit für `churn = 1`. Es gilt also `p_0` = 1 – `p_1`. Der Wert von `i_churn` gibt direkt den vorhergesagten Wert an, der 1 ist, falls `p_1` einen Wert von 0.5 oder höher hat, und sonst 0. Anhand der p-Variablen können Kundensegmente mit hoher bzw. niedriger Kündigungswahrscheinlichkeit direkt gebildet werden.

Wegen der fehlenden Lizenz für SAS/STAT in der Umgebung, in der die Modellanwendung stattfinden musste, konnte im hier beschriebenen Projekt nicht dieser Weg begangen werden. Stattdessen wurde das SAS-Dataset `regressionsModell` mit SAS/BASE-Code ausgelesen und eine Formel zur direkten Berechnung von `p_0` angewandt, wie der nächste Abschnitt zeigt.

3 Modellanwendung mit SAS/BASE

3.1 Formel für die Regressionsanwendung

Bei der logistischen Regression (siehe [1]) wird für jede unabhängige Variable ein Koeffizient ermittelt, der durch ein Optimierungsverfahren während der Modellbildung so gewählt wird, dass die Wahrscheinlichkeit der abhängigen Variablen (hier `churn`), den Wert 0 bzw. 1 zu haben, durch die folgende Formel ausgedrückt wird:

$$p_{0/1} = \frac{exp(k_0 + k_1X_1 + k_2X_2 + \cdots + k_nX_n)}{1 + exp(k_0 + k_1X_1 + k_2\,X_2 + \cdots + k_n\,X_n)}$$

Dabei ist *n* die Zahl der unabhängigen Variablen, *exp* bezeichnet die Exponentialfunktion (*e*-Funktion), k_1 bis k_n sind die in der Regression ermittelten Koeffizienten, X_1 bis X_n sind die unabhängigen Variablen (in unserem Beispiel `feld1` bis `feld6`), und k_0 ist das sogenannte *intercept*, also der Wert, den $p_{0/1}$ annimmt, wenn alle unabhängigen Variablen 0 sind.

Ob man mit dieser Formel die Wahrscheinlichkeit p_0 oder p_1 darstellt, also hier die Wahrscheinlichkeit, dass `churn` den Wert 0 bzw. 1 annimmt, hängt davon ab, auf welchen dieser Werte hin man das Modell trainiert hat. Die SAS-Prozedur PROC LOGISTIC wählt per Default den kleineren der beiden Werte in der abhängigen Variable und trainiert das Modell so, dass dieser vorhergesagt wird (dies lässt sich mit der EVENT-Option des MODEL-Statements ändern), also hier 0.

Demnach können wir die obige Formel anwenden, um für einen aktuellen Kunden, für den die unabhängigen Variablen X_1 bis X_n ja bekannt sind, den Wert p_0 zu berechnen, also die Wahrscheinlichkeit, dass der jeweilige Kunde nicht kündigt. Dazu müssen lediglich die Koeffizienten k_0 bis k_n bekannt sein. Diese können dem im Abschnitt 2.1

erstellten SAS-Dataset `regressionsModell` entnommen werden, wie weiter unten erläutert wird.

3.2 Numerische Kodierung

Damit die Formel wie beschrieben angewandt werden kann, müssen die unabhängigen Variablen natürlich numerisch sein. Hierzu haben wir in der Datenvorverarbeitung alle alphanumerischen Variablen durch die sogenannte Dummy-Kodierung ersetzt, das heißt, für jede Ausprägung der alphanumerischen Variable wurde eine neue, binäre Variable mit den Werten 0 und 1 eingeführt, die nur dann den Wert 1 hat, wenn die ursprüngliche Variable genau die entsprechende Ausprägung hat. Das folgende Beispiel verdeutlicht dies:

Tabelle 1: Dummy-Kodierung einer alphanumerischen Variablen

Ursprünglicher Wert der alphanumerischen Variable *Farbe*	Neue Variable *Farbe_grün*	Neue Variable *Farbe_blau*	Neue Variable *Farbe_rot*
Rot	0	0	1
Blau	0	1	0
Blau	0	1	0
Grün	1	0	0

Diese Kodierung muss auch schon für die Modellbildung erfolgen. Das in Abschnitt 2.1 verwendete CLASS-Statement ist dann nicht mehr notwendig, stattdessen werden alle unabhängigen Variablen im MODEL-Statement aufgelistet.

Tabelle 2: SAS-Dataset mit den Parametern des logistischen Regressionsmodells

TYPE	_NAME_	_CATEGORY_	_NAMEIDX_	_CATIDX_	_MISC_
…	…	…	…	…	…
Z	feld1		0		
Z	feld2		1		
Z	feld3		2		
…	…	…	…	…	…
E	EFFECT	E	0	0	-0.2827
E	EFFECT	E	1	0	0.8118
E	EFFECT	E	2	0	0.1424
…	…	…	…	…	…
E	Intercept	E	0	0	2.2323
…	…	…	…	…	…

3.3 Auslesen der Koeffizienten

Die Tabelle 2 zeigt einen Auszug des SAS-Datasets `regressionsModell`. Es existiert für jede unabhängige Variable ein Eintrag, für den die _TYPE_-Variable den Wert Z hat. Die Variable _NAMEIDX_ nummeriert hierbei die unabhängigen Variablen

durch. Zusätzlich existiert für jede unabhängige Variable ein Eintrag, für den _TYPE_ und _CATEGORY_ beide den Wert E haben und in _NAME_ nicht der Name der Variablen steht, sondern das Schlüsselwort EFFECT. Der Eintrag in _NAMEIDX_ ist aber wieder die Nummerierung, die zu den Einträgen vom Typ Z passt. Aus dem Feld _MISC_ kann bei diesen letzteren Einträgen der Koeffizient k_i gelesen werden. Das *intercept* k_0 hat einen eigenen, leicht zu erkennenden Eintrag.

Da die Koeffizienten bei der Modellanwendung mit den unabhängigen Variablen multipliziert werden müssen, benötigen wir eine direkte Zuordnung der Koeffizienten zu ihren Variablen. Dies lässt sich einfach durch getrenntes Auslesen der Zuordnung von Feldname zu Nummerierung einerseits und der Zuordnung von Nummerierung zu Koeffizient andererseits verwirklichen, wenn danach über die Nummerierung ein Join durchgeführt wird. Das führt zu folgendem SAS-Code:

```
DATA namen (KEEP = feldname nummer koeffizient)
     koeff (KEEP = nummer koeffizient);

    SET regressionsModell;

    feldname    = _NAME_;
    nummer      = _NAMEIDX_;
    koeffizient = _MISC_;

    IF _TYPE_ = 'Z'
    THEN DO;
        koeffizient = .;
        OUTPUT namen;
    END;
    IF _CATEGORY_ = 'E'
    THEN DO;
        IF _NAME_ = 'Intercept'
        THEN nummer = -1;
        OUTPUT koeff;
    END;
RUN;
```

Das *intercept* wird mit der Nummer -1 versehen, anders als die unabhängigen Variablen. Deshalb wird ein Outer Join durchgeführt, um die gewünschte direkte Zuordnung von Variablenname und Koeffizient zu erhalten:

```
PROC SQL;
    CREATE TABLE koeffizienten AS
        SELECT COALESCE(a.feldname, "Intercept") AS feldname,
               SUM(a.koeffizient, b.koeffizient) AS koeffizient,
               nummer
        FROM   namen a
        FULL OUTER JOIN
               koeff b
        ON     a.nummer = b.nummer;
QUIT;
```

Als nächstes werden die Koeffizienten in Makrovariablen eingelesen, damit die Formel aus Abschnitt 3.1 mit ihnen gebildet werden kann:

```
DATA _NULL_;
    SET koeffizienten;
    IF feldname = 'Intercept'
    THEN CALL SYMPUT("intercept", koeffizient);
    ELSE DO;
        CALL SYMPUT(COMPRESS("koeffizient" !! nummer), koeffizient);
        CALL SYMPUT(COMPRESS("feldname"    !! nummer), feldname);
        CALL SYMPUT("anzahl", nummer);
    END;
RUN;
```

Dadurch wird die Makrovariable INTERCEPT mit dem Wert des intercepts belegt, die Makrovariable ANZAHL mit der Zahl der unabhängigen Variablen und die Makrovariablen FELDNAME1, FELDNAME2, … bzw. KOEFFIZIENT1, KOEFFIZIENT2… enthalten die Namen und Koeffizienten der unabhängigen Variablen.

Die eigentliche Modellanwendung beschreibt der nächste Abschnitt.

3.4 Modellanwendung ohne SAS/STAT

Die Makrovariablen aus dem vorigen Abschnitt können nun leicht für eine Makroschleife verwendet werden, die die Summe bildet, die in der Formel in Abschnitt 3.1 zweimal auftaucht. Als Eingabe für diesen letzten Schritt dienen die Daten aktueller Kunden. Die Daten müssen die gleiche Vorverarbeitung durchlaufen wie die für die Modellbildung verwendeten Daten, insbesondere auch die Dummy-Kodierung. Die Zielvariable `churn` muss in diesen Daten nicht enthalten sein. Stattdessen wird die Wahrscheinlichkeit, dass der aktuelle Kunde nicht kündigt, berechnet:

```
DATA vorhersage (DROP = summe);
    SET neudaten;
    summe = SUM(
                  %DO index = 1 %TO &anzahl.;
                       &&feldname&index. *
                               &&koeffizient&index. ,
                  %END;
                  &intercept.
                );
    p_0 = EXP(summe) / (1 + EXP(summe));
RUN;
```

Innerhalb des SUM-Aufrufes erzeugt die Makroschleife für jedes Paar aus Feldname und Koeffizient eine Multiplikation und ein Komma, das die einzelnen Summanden trennt. Nach der Makroschleife, also nach dem letzten Komma, folgt als letzter Summand das *intercept*. Die Variable `p_0` enthält nun die gewünschte Wahrscheinlichkeit; eine direkte Vorhersage (0 oder 1) der Kündigung kann daraus durch Vergleich mit einem Schwellwert, zum Beispiel 0.5, erzeugt werden.

In weiteren Folgeschritten kann nun die Vorhersage ausgewertet werden. Falls es sich um Testdaten handelt, kann die Vorhersage mit dem tatsächlichen Kündigungsverhalten verglichen werden, um die Qualität des Modells zu bestimmen.

4 Ergebnisse

In diesem Projekt ging es um die Vorhersage der Loyalität eines Kunden, die durch seine Wahrscheinlichkeit, nicht zu kündigen, ausgedrückt wurde. Diese Wahrscheinlichkeit führt direkt zu einer Einteilung der Kunden in unterschiedlich loyal eingeschätzte Kundengruppen. Beispielhaft wird hier eine Auswertung von drei Kundengruppen auf echten Daten, die während der Modellbildung nicht verwendet wurden, gezeigt (Abbildung 1). Dabei wurde jeder aktuelle Kunde in eine der drei Gruppen eingeteilt. Für den Endnutzer werden die drei Gruppen durch Farben kodiert: das grüne Segment, in Abbildung 1 die jeweils linke Säule, hat eine niedrige Kündigungswahrscheinlichkeit, das rote (rechte Säule) die höchste. Das gelbe Segment (mittlere Säule) liegt etwa im Durchschnitt.

Die durchschnittliche Kündigungsquote von 1.5% wird im grünen (linken) Segment deutlich unterschritten, im roten (rechten) Segment ist die Kündigungsquote dagegen etwa sechsmal so hoch wie im Durchschnitt. Somit konnten Kundengruppen erkannt werden, die signifikant vom Durchschnitt aller Kunden abweichen. Aus CRM-Sicht ist zudem vorteilhaft, dass nur wenige Kunden in die besonders kündigungsgefährdete Gruppe eingeteilt werden (rechte Säulengruppe, dritte Säule).

Aus technischer Sicht ist natürlich der Vergleich der beiden Ansätze zur Modellanwendung interessant. Hier stellten wir bei Verwendung der gleichen Modellparameter und Testdaten nur wenige Abweichungen bei den Vorhersagen für individuelle Kunden zwischen der Modellanwendung mit PROC LOGISTIC (Abschnitt 2.2) und der selbst programmierten Modellanwendung (Abschnitt 3) fest. Diese Abweichungen mögen auf Rundungsdifferenzen oder auch auf spezielle Berechnungswege innerhalb der Prozedur PROC LOGISTIC zurückzuführen sein, die wir nicht erkannt haben. Aus Anwendersicht ist jedoch die Gesamtqualität der Vorhersagen, gemessen wie in Abbildung 1 oder mit den Standardmaßen Accuracy (Trefferquote), Precision oder Recall, entscheidend. Diese unterschied sich bei den beiden Ansätzen fast gar nicht.

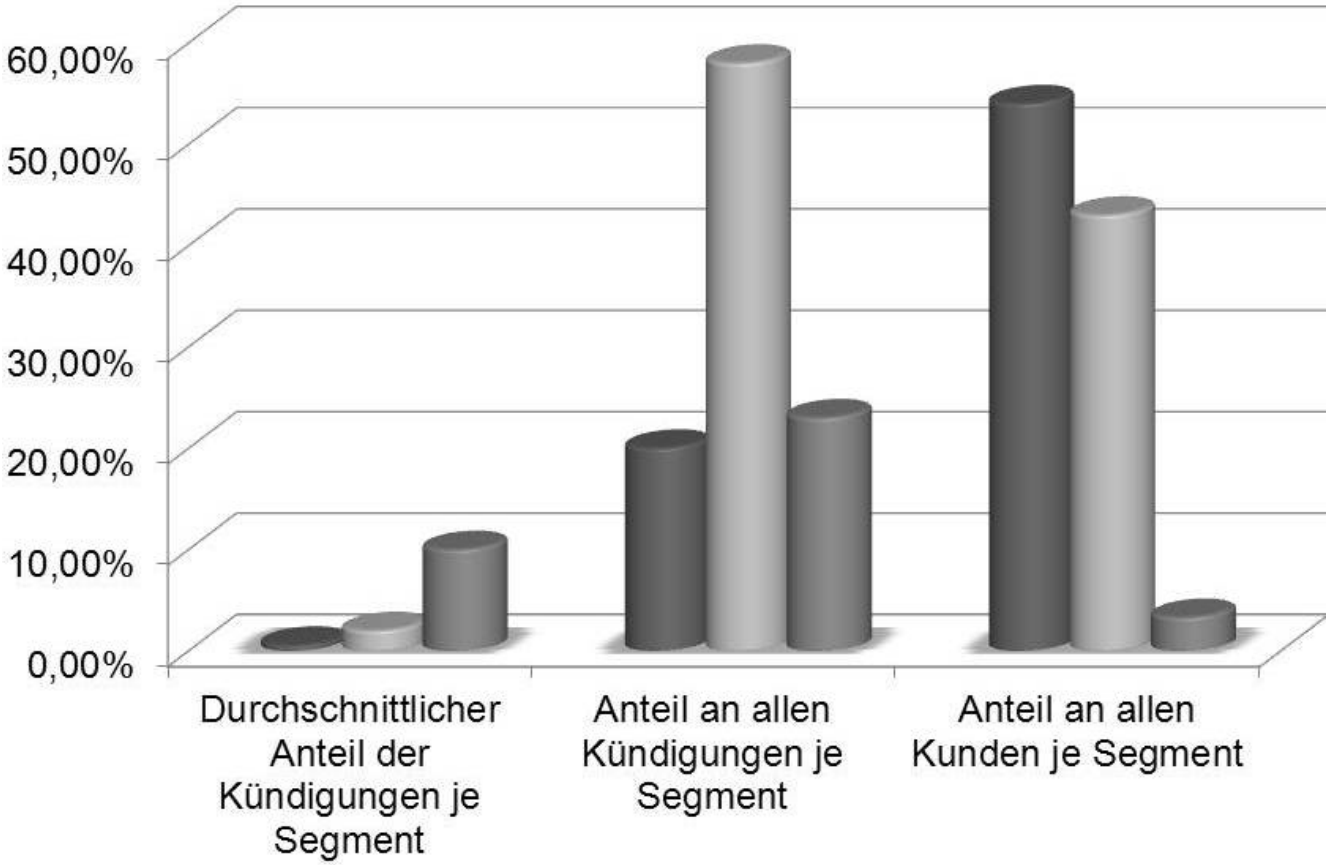

Abbildung 1: Auswertung der vorhergesagten Kundensegmente

Literatur

[1] Hosmer, D. W., Lemeshow, S.: Applied logistic regression. Wiley & Sons, New York 2000.