

Data Warehousing for Data Mining: A Case Study

C. Olivia Rud, Executive Vice President, DataSquare, LLC

ABSTRACT

Data Mining is gaining popularity as an effective tool for increasing profits in a variety of industries. However, the quality of the information resulting from the data mining exercise is only as good as the underlying data. The importance of accurate, accessible data is paramount. A well designed data warehouse can greatly enhance the effectiveness of the data mining process. This paper will discuss the planning and development of a data warehouse for a credit card bank. While the discussion covers a number of aspects and uses of the data warehouse, a particular focus will be on the critical needs for data access pertaining to targeting model development. The case study will involve developing a Lifetime Value model from a variety of data sources including account history, customer transactions, offer history and demographics. The paper will discuss the importance of some aspects of the physical design and maintenance to the data mining process.

INTRODUCTION

One of the most critical steps in any data mining project is obtaining good data. Good data can mean many things: clean, accurate, predictive, timely, accessible and/or actionable. This is especially true in the development of targeting models. Targeting models are only as good as the data on which they are developed. Since the models are used to select names for promotions, they can have a significant financial impact on a company's bottom line.

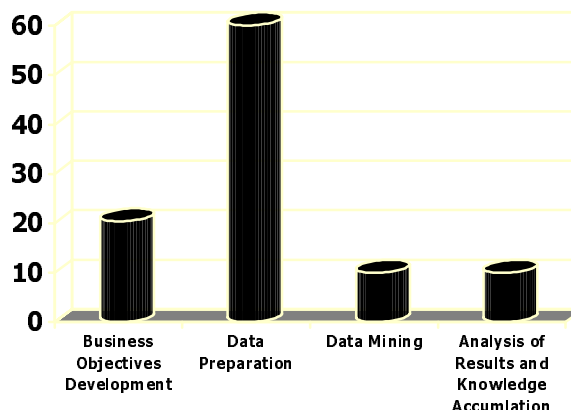
The overall objectives of the data warehouse are to assist the bank in developing a totally data driven approach to marketing, risk and customer relationship management. This would provide opportunities for targeted marketing programs. The analysis capabilities would include:

- Response Modeling and Analysis
- Risk or Approval Modeling and Analysis
- Activation or Usage Modeling and Analysis
- Lifetime Value or Net Present Value Modeling
- Segmentation and Profiling
- Fraud Detection and Analysis
- List and Data Source Analysis
- Sales Management
- Customer Touchpoint Analysis
- Total Customer Profitability Analysis

The case study objectives focus on the development of a targeting model using information and tools available through the data warehouse. Anyone who has worked with target model development knows that data extraction and preparation are often the most time consuming part of model development. Ask a group of analysts how

much of their time is spent preparing data. A majority of them will say over 50%!

WHERE'S THE EFFORT



Over the last 10 years, the bank had amassed huge amounts of information about our customer and prospects. The analysts and modelers knew there was a great amount of untapped value in the data. They just had to figure out a way to gain access to it. The goal was to design a warehouse that could bring together data from disparate sources into one central repository.

THE TABLES

The first challenge was to determine which tables should go into the data warehouse. We had a number of issues:

- Capturing response information
- Storing transactions
- Defining date fields

Response information

Responses begin to arrive about a week after an offer is mailed. Upon arrival, the response is put through a risk screening process. During this time, the prospect is considered 'Pending.' Once the risk screening process is complete, the prospect is either 'Approved' or 'Declined.' The bank considered two different options for storing the information in the data warehouse.

1) The first option was to store the data in one large table. The table would contain information about those approved as well as those declined. Traditionally, across all applications, they saw approval rates hover around 50%. Therefore, whenever analyses was done on either the approved applications (with a risk management focus) or on the declined population (with a marketing as

well as risk management focus), every query needed to go through nearly double the number of records as necessary.

2) The second option was to store the data in three small tables. This accommodated the daily updates and allowed for pending accounts to stay separate as they awaited information from either the applicant or another data source.

With applications coming from e-commerce sources, the importance of the “pending” table increased. This table was examined daily to determine which pending accounts could be approved quickly with the least amount of risk. In today’s competitive market, quick decisions are becoming a competitive edge.

Partitioning the large customer profile table into three separate tables improved the speed of access for each of the three groups of marketing analysts who had responsibility for customer management, reactivation and retention, and activation. The latter group was responsible for both the one-time buyers and the prospect pools.

FILE STRUCTURE ISSUES

Many of the tables presented design challenges. Structural features that provided ease of use for analysts could complicate the data loading process for the IT staff. This was a particular problem when it came to transaction data. This data is received on a monthly basis and consists of a string of transactions for each account for the month. This includes transactions such as balances, purchases, returns and fees. In order to make use of the information at a customer level it needs to be summarized. The question was how to best organize the monthly performance data in the data warehouse. Two choices were considered:

a) Long skinny file: this took the data into the warehouse in much the same form as it arrived. Each month would enter the table as a separate record. Each year has a separate table. The fields represent the following:

Month01 = January – load date
Cust_1 = Customer number 1
VarX = Predictive variable
CDate = Campaign date

The layout is as follows:

```
Month01 Cust_1 VarA VarB VarC VarD VarE CDate C#
Month01 Cust_1 VarA VarB VarC VarD VarE CDate C#
Month03 Cust_1 VarA VarB VarC VarD VarE CDate C#
Month04 Cust_1 VarA VarB VarC VarD VarE CDate C#
|           |           |           |           |
Month12 Cust_1 VarA VarB VarC VarD VarE CDate C#

Month01 Cust_2 VarA VarB VarC VarD VarE CDate C#
Month02 Cust_2 VarA VarB VarC VarD VarE CDate C#
```

```
Month03 Cust_2 VarA VarB VarC VarD VarE CDate C#
Month04 Cust_2 VarA VarB VarC VarD VarE CDate C#
|           |           |           |           |
Month12 Cust_2 VarA VarB VarC VarD VarE CDate C#
```

b) Wide file: this design has a single row per customer. It is much more tedious to update. But in its final form, it is much easier to analyze because the data has already been organized into a single customer record. Each year has a separate table. The layout is as follows:

```
Cust_1 VarA01 VarA02 VarA03 ... VarA12 VarB01
VarB02 VarB03 ... VarB12 VarC01 VarC02 VarC03 ...
VarC12 VarD01 VarD02 VarD03 ... VarD12 VarE01
VarE02 VarE03 ... VarE12 CDate C#
```

```
Cust_2 VarA01 VarA02 VarA03 ... VarA12 VarB01
VarB02 VarB03 ... VarB12 VarC01 VarC02 VarC03 ...
VarC12 VarD01 VarD02 VarD03 ... VarD12 VarE01
VarE02 VarE03 ... VarE12 CDate C#
```

The final decision was to go with the wide file or the single row per customer design. The argument was that the manipulation to the customer level file could be automated thus making the best use of the analyst’s time.

DATE ISSUES

Many analyses are performed using date values. In our previous situation, we saw how transactions are received and updated on a monthly basis. This is useful when comparing values of the same vintage. However, another analyst might need to compare balances at a certain stage in the customer lifestyle. For example, to track customer balance cycles from multiple campaigns a field that denotes the load date is needed.

The first type of analysis was tracking monthly activity by the vintage acquisition campaign. For example, calculating monthly trends of balances aggregated separately for those accounts booked in May 99 and September 99. This required aggregating the data for each campaign by the “load date” which corresponded to the month in which the transaction occurred.

The second analyses focused on determining and evaluating trends in the customer life cycle. Typically, customers who took a balance transfer at the time of acquisition showed balance run-off shortly after the introductory teaser APR rate expired and the account was repriced to a higher rate. These are the dreaded “rate surfers.” Conversely, a significant number of customers, who did not take a balance transfer at the time of acquisition, demonstrated balance build. Over time these customers continued to have higher than average monthly balances. Some demonstrated revolving behavior: paying less than the full balance each month and a willingness to pay interest on the revolving balance. The remainder in this group simply user their credit cards for convenience. Even though they built balances through

debit activity each month, they chose to pay their balances in full and avoid finance charges. These are the “transactors” or convenience users.

The second type of analysis needed to use ‘Months on books’, regardless of the source campaign. This analysis required computation of the account age by looking at both the date the account was open as well as the “load date” of the transaction data. However, if the data mining task is to also understand this behavior in the context of campaign vintage which was mentioned earlier, there is another consideration. Prospects for the “May 99” campaign were solicited in May of 1999. However, many new customers did not use their card until June or July of 1999. There were three main reasons: 1) some wanted to compare their offer to other offers; 2) processing is slower during May and June; and 3) some waited until a specific event (e.g. purchase of a large present at the Christmas holidays) to use their card for the first time.

At this point the data warehouse probably needs to store at least the following date information:

- (a) Date of the campaign
- (b) Date the account first opened
- (c) Date of the first transaction
- (d) Load date for each month of data

The difference between either “b” or “c” above and “d” can be used as the measure used to index account age or month on books.

No single date field is more important than another but multiple date files are problem necessary if vintage as well as customer life-cycle analyses are both to be performed.

DEVELOPING THE MODEL

To develop a Lifetime Value model, we need to extract information from the Customer Information Table for risk indices as well as the Offer History Table for demographic and previous offer information.

Customer Information Table

A Customer Information Table is typically designed with one record per customer. The customer table contains the identifying information that can be linked to other tables such as a transaction table to obtain a current snapshot of a customer’s performance. The following list details the key elements of the Customer Information Table:

Customer ID – a unique numeric or alpha-numeric code that identifies the customer throughout his entire lifecycle. This element is especially critical in the credit card industry where the credit card number may change in the event of a lost or stolen card. But it is essential in any table to effectively link and tract the behavior of and actions taken on an individual customer.

Household ID – a unique numeric or alpha-numeric code that identifies the household of the customer through his or her entire lifecycle. This identifier is useful in some industries where products or services are shared by more than one member of a household.

Account Number – a unique numeric or alpha-numeric code that relates to a particular product or service. One customer can have several account numbers.

Customer Name – the name of a person or a business. It is usually broken down into multiple fields: last name, first name, middle name or initial, salutation.

Address – the street address is typically broken into components such as number, street, suite or apartment number, city, state, zip+4. Some customer tables have a line for a P.O. Box. With population mobility about 10% per year, additional fields that contain former addresses are useful for tracking and matching customers to other files.

Phone Number – current and former numbers for home and work.

Demographics – characteristics such as gender, age, income, etc. may be stored for profiling and modeling.

Products or Services – the list of products and product identification numbers varies by company. An insurance company may list all the policies along with policy numbers. A bank may list all the products across different divisions of the bank including checking, savings, credit cards, investments, loans, and more. If the number of products and product detail is extensive, this information may be stored in a separate table with a customer and household identifier.

Offer Detail – the date, type of offer, creative, source code, pricing, distribution channel (mail, telemarketing, sales rep, e-mail) and any other details of an offer. Most companies look for opportunities to cross-sell or up-sell their current customers. There could be numerous “offer detail” fields in a customer record, each representing an offer for an additional product or service.

Model Scores – response, risk, attrition, profitability scores and/or any other scores that are created or purchased.

Transaction Table

The Transaction Table contains records of customer activity. It is the richest and most predictive information but can be the most difficult to access. Each record represents a single transaction. So there are multiple records for each customer. In order to use this data for modeling, it must be summarized and aggregated to a customer level. The following lists key elements of the Transaction Table:

Customer ID – defined above.

Household ID – defined above.

Transaction Type – The type of credit card transaction such as charge, return, or fee (annual, overlimit, late).

Transaction Date – The date of the transaction

Transaction Amount – The dollar amount of the transaction.

Offer History Table

The Offer History Table contains details about offers made to prospects, customers or both. The most useful format is a unique record for each customer or prospect. Variables created from this table are often the most predictive in response and activation targeting models. It seems logical that if you know someone has received your offer every month for 6 months, they are less likely to respond than someone who is seeing your offer for the first time. As competition intensifies, this type of information is becoming increasingly important.

A Customer Offer History Table contains all cross-sell, up-sell and retention offers. A Prospect Offer History Table contains all acquisition offers as well as any predictive information from outside sources. It is also useful to store several addresses on the Prospect Offer History Table.

With an average amount of solicitation activity, this type of table can become very large. It is important to perform analysis to establish business rules that control the maintenance of this table. Fields like 'date of first offer' is usually correlated with response behavior. The following list details some key elements in an Offer History Table:

Prospect ID/Customer ID – as in the Customer Information Table, this is a unique numeric or alpha-numeric code that identifies the prospect for a specific length of time. This element is especially critical in the credit card industry where the credit card number may change in the event of a lost or stolen card. But it is essential in any table to effectively track the behavior of and actions taken on an individual customer.

Household ID – a unique numeric or alpha-numeric code that identifies the household of the customer through his entire lifecycle. This identifier is useful in some industries where products or services are shared by more than one member of a household.

Prospect Name* – the name of a person or a business. It is usually broken down into multiple fields: last name, first name, middle name or initial, salutation.

Address* – the street address is typically broken into components such as number, street, suite or apartment number, city, state, zip+4. As in the Customer Table, some prospect tables have a line for a P.O. Box. Additional fields that contain former addresses are useful for matching prospects to outside files.

Phone Number – current and former numbers for home and work.

Offer Detail – includes the date, type of offer, creative, source code, pricing, distribution channel (mail, telemarketing, sales rep, email) and any other details of the offer. There could be numerous groups of "offer detail" fields in a prospect or customer record, each representing an offer for an additional product or service.

Offer Summary – date of first offer (for each offer type), best offer (unique to product or service), etc.

Model Scores* – response, risk, attrition, profitability scores and/or any scores other that are created or purchased.

Predictive Data* – includes any demographic, psychographic or behavioral data.

*These elements appear only on a Prospect Offer History Table. The Customer Table would support the Customer Offer History Table with additional data.

DEFINING THE OBJECTIVE

The overall objective is to measure Lifetime Value (LTV) of a customer over a 3-year period. If we can predict which prospects will be profitable, we can target our solicitations only to those prospects and reduce our mail expense. LTV consists of four major components:

- 1) Activation - probability calculated by a model. Individual must respond, be approved by risk and incur a balance.
- 2) Risk – the probability of charge-off is derived from a risk model score. It is converted to an index.
- 3) Expected Account Profit – expected purchase, fee and balance behavior over a 3-year period.
- 4) Marketing Expense - cost of package, mailing & processing (approval, fulfillment).

THE DATA COLLECTION

Names from three campaigns over the last 12 months were extracted from the Offer History Table. All predictive information was included in the extract: demographic and credit variables, risk scores and offer history.

The expected balance behavior was developed using segmentation analysis. An index of expected performance is displayed in a matrix of gender by marital status by age group (see Appendix A).

The marketing expense which includes the mail piece and postage is \$.78.

To predict Lifetime Value, data was pulled from the Offer History Table from three campaigns with a total of 966,856 offers. To reduce the amount of data for analysis and maintain the most powerful information, a sample is created using all of the 'Activation' and 1/25th of the remaining records. This includes non-responders and non-activating responders. We define an ACTIVE as a

customer with a balance at three months. The following code creates the sample dataset:

```
DATA A B;
SET LIB.DATA;
IF 3MON_BAL > 0 THEN OUTPUT A;
ELSE OUTPUT B;

DATA LIB.SAMPDATA;
SET A B (WHERE=(RANUNI(5555) < .04));
SAMP_WGT = 25;
RUN;
```

This code is putting into the sample dataset, all customers who activated and a 1/25th random sample of the balance of accounts. It also creates a weight variable called *SAMP_WGT* with a value of 25.

The following table displays the sample characteristics:

	Campaign	Sample	Weight
Non Resp/Non Active Resp	929,075	37,163	25
Responders/Active	37,781	37,781	1
Total	966,856	74,944	

The non-responders and non-activated responders are grouped together since our target is active responders. This gives us a manageable sample size of 74,944.

MODEL DEVELOPMENT

The first component of the LTV, the probability of activation, is based on a binary outcome, which is easily modeled using logistic regression. Logistic regression uses continuous values to predict the odds of an event happening. The log of the odds is a linear function of the predictors. The equation is similar to the one used in linear regression with the exception of the use of a log transformation to the independent variable. The equation is as follows:

$$\log(p/(1-p)) = B_0 + B_1X_1 + B_2X_2 + \dots + B_nX_n$$

Variable Preparation - Dependent

To define the dependent variable, create the variable *ACTIVATE* defined as follows:

```
IF 3MOBAL > 0 THEN ACTIVATE = 1;
ELSE ACTIVATE = 0;
```

Variable Preparation – Previous Offers

The bank has four product configurations for credit card offers. Each product represents a different intro rate and intro length combination. From our offer history table, we pull four variables for modeling that represent the number

of times each product was mailed in the last 6 months: *NPROD1*, *NPROD2*, *NPROD3*, and *NPROD4*.

Through analysis, the following variables were determined to be the most predictive.

SAM_OFF1 – received the same offer one time in the past 6 months.

DIF_OFF1 – received a different offer one time in the past 6 months.

SAM_OFF2 – received the same offer more than one time in the past 6 months.

DIF_OFF2 – received a different offer more than one time in the past 6 months.

The product being modeled is Product 2. The following code creates the variables for modeling:

```
SAM_OFF1 = (IF NPROD2 = 1);
SAM_OFF2 = (IF NPROD2 > 1);
DIF_OFF1 = (IF SUM(NPROD1, NPROD3, NPROD4) = 1);
DIF_OFF2 = (IF SUM(NPROD1, NPROD3, NPROD4) > 1);
```

If the prospect has never received an offer, then the values for the four named variables will all be 0.

Preparing Credit Variables

Since, logistic regression looks for a linear relationship between the independent variables and the log of the odds of the dependent variable, transformations can be used to make the independent variables more linear. Examples of transformations include the square, cube, square root, cube root, and the log.

Some complex methods have been developed to determine the most suitable transformations. However, with the increased computer speed, a simpler method is as follows: create a list of common/favorite transformations; create new variables using every transformation for each continuous variable; perform a logistic regression using all forms of each continuous variable against the dependent variable. This allows the model to select which form or forms fit best. Occasionally, more than one transformation is significant. After each continuous variable has been processed through this method, select the one or two most significant forms for the final model. The following code demonstrates this technique for the variable *Total Balance* (*TOT_BAL*):

```
PROC LOGISTIC LIB.DATA;
WEIGHT SMP_WGT;
MODEL ACTIVATE = TOT_BAL TOT_B_SQ TOT_B_CU
TOT_B_I TOT_B_LG / SELECTION=STEPWISE;
RUN;
```

The logistic model output (see Appendix D) shows two forms of TOT_BAL to be significant in combination: TOT_BAL TOT_B_SQ. These forms will be introduced into the final model.

Partition Data

The data are partitioned into two datasets, one for model development and one for validation. This is accomplished by randomly splitting the data in half using the following SAS® code:

```
DATA LIB.MODEL LIB.VALID;
  SET LIB.DATA;
  IF RANUNI(0) < .5 THEN OUTPUT LIB.MODEL;
  ELSE OUTPUT LIB.VALID;
RUN;
```

If the model performs well on the model data and not as well on the validation data, the model may be over-fitting the data. This happens when the model memorizes the data and fits the models to unique characteristics of that particular data. A good, robust model will score with comparable performance on both the model and validation datasets.

As a result of the variable preparation, a set of 'candidate' variables has been selected for the final model. The next step is to choose the model options. The backward selection process is favored by some modelers because it evaluates all of the variables in relation to the dependent variable while considering interactions among the independent or predictor variables. It begins by measuring the significance of all the variables and then removing one at a time until only the significant variables remain.

The sample weight must be included in the model code to recreate the original population dynamics. If you eliminate the weight, the model will still produce correct ranking-ordering but the actual estimates for the probability of a 'paid-sale' will be incorrect. Since our LTV model uses actual estimates, we will include the weights.

The following code is used to build the final model.

```
PROC LOGISTIC LIB.MODEL;
  WEIGHT SMP_WGT;
  MODEL ACTIVATE = INQL6MO TOT_BAL TOT_B_SQ
  SAM_OFF1 DIF_OFF1 SAM_OFF2 DIF_OFF2 INCOME
  INC_LOG AGE_FILE NO30DAY TOTCLIM POPDENS
  MAIL_ORD// SELECTION=BACKWARD;
RUN;
```

The resulting model has 7 predictors. (See Appendix C) The parameter estimate is multiplied times the value of the variable to create the final probability. The strength of the predictive power is distributed like a chi-square so we look to that distribution for significance. The higher the chi-square, the lower the probability of the event occurring randomly ($pr > \text{chi-square}$). The strongest predictor is the variable DIFOFF2 which demonstrates the

power of offer history on the behavior of a prospect. Introducing offer history variables into the acquisition modeling process has been single most significant improvement in the last three years.

The following equation shows how the probability is calculated, once the parameter estimates have been calculated:

$$\text{prob} = \frac{\exp(B_0 + B_1X_1 + B_2X_2 + \dots + B_nX_n)}{(1 + \exp(B_0 + B_1X_1 + B_2X_2 + \dots + B_nX_n))}$$

This creates the final score, which can be evaluated using a gains table (see Appendix D). Sorting the dataset by the score and dividing it into 10 groups of equal volume creates the gains table. This is called a Decile Analysis.

The validation dataset is also scored and evaluated in a gains table or Decile Analysis (See Appendix E).

Both of these tables show strong rank ordering. This can be seen by the gradual decrease in predicted and actual probability of 'Activation' from the top decile to the bottom decile. The validation data shows similar results, which indicates a robust model. To get a sense of the 'lift' created by the model, a gains chart is a powerful visual tool (see Appendix D). The Y-axis represents the % of 'Activation' captured by each model. The X-axis represents the % of the total population mailed. Without the model, if you mail 50% of the file, you get 50% of the potential 'Activation'. If you use the model and mail the same percentage, you capture over 97% of the 'Activation'. This means that at 50% of the file, the model provides a 'lift' of 94% $\{(97-50)/50\}$.

Financial Assessment

To get the final LTV we use the formula:
 $LTV = \text{Pr(Paid Sale)} * \text{Risk Index Score} * \text{Expected Account Profit} - \text{Marketing Expense}$

At this point, we apply the risk matrix score and expected account profit value. The financial assessment shows the models ability to select the most profitable customers (See Appendix F). Notice how the risk score index is lower for the most responsive customers. This is common in direct response and demonstrates 'adverse selection'. In other words, the riskier prospects are often the most responsive.

At some point in the process, a decision is made to mail a percent of the file. In this case, you could consider the fact that in decile 7, the LTV becomes negative and limit your selection to deciles 1 through 6. Another decision criteria could be that you need to be above a certain 'hurdle rate' to cover fixed expenses. In this case, you might look at the cumulative LTV to be above a certain amount such as \$30. Decisions are often made considering a combination of criteria.

The final evaluation of your efforts may be measured in a couple of ways. You could determine the goal to mail fewer pieces and capture the same LTV. If we mail the entire file with random selection, we would capture \$13,915,946 in LTV. This has a mail cost of \$754,155. By mailing 5 deciles using the model, we would capture \$14,042,255 in LTV with a mail cost of only \$377,074. In other words, with the model we could capture slightly more LTV and *cut our marketing cost in half!*

Or, we can compare similar mail volumes and increase LTV. With random selection at 50% of the file, we would capture \$6,957,973 in LTV. Modeled, the LTV would climb to \$14,042,255. This is a lift of over 100% $((14042255-6957973)/6957973 = 1.018)$.

CONCLUSION

Successful data mining and predictive modeling depends on quality data that is easily accessible. A well-constructed data warehouse allows for the integration of Offer History which has an excellent predictor of Lifetime Value.

REFERENCES

Cabena, Hadjnia, Stadler, Verhees, Zanasi, *Discovering Data Mining from Concept to Implementation*, Prentice Hall, 1997

Grossman, Randall B. (1999), Building CRM Systems: Lessons Learned the Hard Way, NCDM Proceedings, December 1999

Hosmer, DW., Jr. and Lemeshow, S. (1989), *Applied Logistic Regression*, New York: John Wiley & Sons, Inc.

SAS Institute Inc. (1989) *SAS/Stat User's Guide, Vol.2, Version 6, Fourth Edition*, Cary NC: SAS Institute Inc.

AUTHOR CONTACT

C. Olivia Rud
DataSquare, LLC
350 Theodore Fremd Avenue
Rye, NY 10580

Voice: (610) 918-3801
Fax: (610) 918-3974
Internet: Olivia@datasquare.com

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. © indicates USA registration.

Appendix A – Risk Score Index

	MALE				FEM ALE			
	M	S	D	W	M	S	D	W
< 40	1.22	1.15	1.18	1.10	1.36	1.29	1.21	1.17
40-49	1.12	1.01	1.08	1.02	1.25	1.18	1.13	1.09
50-59	0.98	0.92	0.90	0.85	1.13	1.08	1.10	1.01
60+	0.85	0.74	0.80	0.79	1.03	0.98	0.93	0.88

Appendix B – Variable Selection

The LOGISTIC Procedure

Analysis of Maximum Likelihood Estimates

Variable	DF	Parameter Estimate	Standard Error	Wald Chi-Square	Pr > Chi-Square	Standardized Estimate	Odds Ratio
INTERCPT	1	10.1594	27.1690	0.1398	0.7085	.	.
TOT_BAL	1	-23.2172	0.3284	4.7290	0.0297	-4.287240	0.000
TOT_B_SQ	1	-3.8671	1.7783	5.9005	0.0411	-0.997359	.
TOT_B_CU	1	0.0033	1.3594	0.0057	0.9358	0.851626	.
TOT_B_LG	1	1.9442	0.2658	0.0633	0.8013	0.936637	.
TOT_B_I	1	0.8499	0.7291	1.5507	0.2130	0.672450	.

APPENDIX C – Model Output

The LOGISTIC Procedure

Analysis of Maximum Likelihood Estimates

Variable	DF	Parameter Estimate	Standard Error	Wald Chi-Square	Pr > Chi-Square	Standardized Estimate	Odds Ratio
INTERCPT	1	-2.5744	0.0169	0.1398	0.0001	.	.
INQL6MO	1	-0.0166	0.0059	0.0057	0.0049	-0.030639	0.000
SAMOFF1	1	0.0263	0.0063	5.7290	0.0001	0.043238	1.027
DIFOFF2	1	0.0620	0.0085	7.9005	0.0001	0.081625	1.064
TOT_BAL	1	0.0291	0.0105	0.0633	0.0055	0.038147	1.030
TOT_B_SQ	1	0.0353	0.0081	1.5507	0.0001	0.046115	1.036
INC_LOG	1	-0.2117	0.0057	0.0633	0.0001	-0.263967	0.809
MAIL_ORD	1	0.0634	0.0062	4.5507	0.0001	0.079093	1.065

Association of Predicted Probabilities and Observed Response

Concordant = 57.1%	Somers' D = 0.209
Discordant = 36.2%	Gamma = 0.224
Tied = 6.6%	Tau-a = 0.030
(7977226992 pairs)	c = 0.604

Appendix D – Decile Analysis: Model Data

DECILE	NUMBER OF PROSPECTS	PREDICTED % ACTIVATION	ACTUAL % ACTIVATION	NUMBER OF ACTIVES	CUM ACTUAL % ACTIVATION
1	48,342	11.47%	11.36%	5,492	11.36%
2	48,342	8.46%	8.63%	4,172	9.99%
3	48,342	4.93%	5.03%	2,429	8.34%
4	48,342	2.14%	1.94%	935	6.74%
5	48,342	0.94%	0.95%	459	5.58%
6	48,342	0.25%	0.28%	133	4.70%
7	48,342	0.11%	0.11%	51	4.04%
8	48,342	0.08%	0.08%	39	3.54%
9	48,342	0.00%	0.00%	2	3.15%
10	48,342	0.00%	0.00%	1	2.84%

Appendix E - Decile Analysis: Validation Data

DECILE	NUMBER OF PROSPECTS	PREDICTED % ACTIVATION	ACTUAL % ACTIVATION	NUMBER OF ACTIVES	CUM ACTUAL % ACTIVATION
1	48,342	10.35%	10.12%	4,891	10.12%
2	48,342	8.44%	8.16%	3,945	9.14%
3	48,342	5.32%	5.76%	2,783	8.01%
4	48,342	2.16%	2.38%	1,151	6.60%
5	48,342	1.03%	1.07%	519	5.50%
6	48,342	0.48%	0.56%	269	4.67%
7	48,342	0.31%	0.23%	112	4.04%
8	48,342	0.06%	0.05%	25	3.54%
9	48,342	0.01%	0.01%	5	3.15%
10	48,342	0.00%	0.00%	1	2.83%

Appendix F - Financial Analysis

DECILE	NUMBER OF PROSPECTS	PREDICTED % ACTIVATED	RISK SCORE INDEX	EXPECTED ACCT BEHAVIOR	AVERAGE LTV	CUM AVERAGE LTV	SUM CUM LTV
1	96,685	10.35%	0.94	\$632	\$58.27	\$58.27	\$5,633,985
2	96,686	8.44%	0.99	\$620	\$46.47	\$52.37	\$10,126,713
3	96,686	5.32%	0.98	\$587	\$26.45	\$43.73	\$12,684,175
4	96,685	2.16%	0.96	\$590	\$9.49	\$35.17	\$13,602,084
5	96,686	1.03%	1.01	\$571	\$4.55	\$29.05	\$14,042,255
6	96,685	0.48%	1.00	\$553	\$0.74	\$24.33	\$14,114,007
7	96,686	0.31%	1.03	\$540	(\$0.18)	\$20.83	\$14,096,406
8	96,685	0.06%	0.99	\$497	(\$0.34)	\$18.18	\$14,063,329
9	96,685	0.01%	1.06	\$514	(\$0.76)	\$16.08	\$13,990,047
10	96,686	0.00%	1.10	\$534	(\$0.77)	\$14.39	\$13,915,946