Paper 941-2017

Maximizing Cross-Sell Opportunities with Predictive Analytics for Financial Institutions

Nate Derby, Stakana Analytics, Seattle, WA

ABSTRACT

In the increasingly competitive environment for banks and credit unions, every potential advantage should be pursued. One of these advantages is to market additional products to our existing customers rather than to new customers, since our existing customers already know (and hopefully trust) us, and we have so much data on them. But how can this best be done? How can we market the right products to the right customers at the right time? Predictive analytics can do this by forecasting which customers have the highest chance of purchasing a given financial product.

This paper provides a step-by-step overview of a relatively simple but comprehensive approach to maximize cross-sell opportunities among our customers. We first prepare the data for a statistical analysis. With some basic predictive analytics techniques, we can then identify those customers who have the highest chance of buying a financial product. For each of these customers, we can also gain insight into why they would purchase, thus suggesting the best way to market to them. We then make suggestions to improve the model for better accuracy.

Code snippets will be shown for any version of SAS® but will require the SAS/STAT package. This approach can also be applied to many other organizations and industries.

The `%makeCharts` and `%makeROC` macros in this paper are available at `nderby.org/docs/charts.zip`.

INTRODUCTION: THE CROSS-SELL OPPORTUNITY

Like most financial institutions, suppose a portion of our customers don't have an active checking account.[1] That is, some of them either have *no* checking account, or they have one but rarely use it. Could we get some of these customers to get an active checking account (either by opening a new one or using an existing one) with minimal effort? That is, could some of these customers be nudged with minimal effort? If so, who are our best prospects?

In fact, **cross-selling to existing customers is usually easier and less expensive than gaining new customers** since these customers already know and trust us, and we already knows so much about them. Cross-selling to them might be as simple as sending an email message. The key is making full use of the data we have on our customers.

Overall, **predictive analytics helps us with targeting marketing by allowing us to give the *right* message to the *right* customer at the *right* time**.

[1] Under some agreed-upon definition of "active." For example, it can be a checking account with at least one credit over $250 and three debits over $25 every quarter.

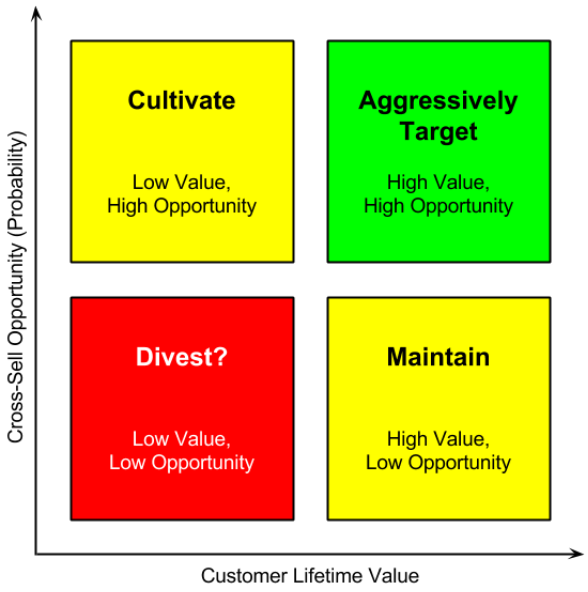


Figure 1: Customer segmentation into quadrants of lifetime value and cross-sell opportunity (i.e., probability).

If we can identify which customers are our best cross-selling opportunities, we can proactively focus our marketing efforts on them. With this in mind, Figure 1 shows four cross-sell marketing quadrants that describe the segmentation of our customers. The horizontal axis shows the *customer lifetime value* (in terms of the aggregate balance of deposits and loans), which may be very different from the present value. For instance, a student typically has a low present value but will have a much higher lifetime value. The vertical axis shows the cross-sell opportunity (i.e., probability). We can divide this region into four quadrants:

- *Divest?*: These are customers with a low value and low cross-sell opportunity. As such, these are customers that we wouldn't really mind losing. If it costs little to maintain them, there's no real reason to divest ourselves of them, but they aren't going to grow our financial institution.
- *Maintain*: These are customers with a high value but low cross-sell opportunity. These are among our best customers, so we should keep them satisfied even if we don't focus on them for cross-sell opportunities.
- *Cultivate*: These are customers with a low value but high cross-sell opportunity. As such, we should focus a little effort on them for cross-sell opportunities and cultivate them to have a higher value. Still, we should target our real efforts on the next quadrant.
- *Aggressively Target*: These are customers with a high value and high cross-sell opportunity, and they are the ones we should focus most of our efforts on. In other words, **these are the customers for whom we can have the most impact**.

But how can we use our data to focus on these high-value, high-opportunity customers?

In this paper, we take the approach of Derby and Keintz (2016) for forecasting customer attrition and apply it to cross-sell opportunities. That paper, in turn, uses ideas from Thomas (2010) and Karp (1998). Customer lifetime value isn't covered in this paper, but Fader (2012) gives a good introduction to the concept and Lu (2003) gives a basic mathematical approach.

Throughout this paper, we'll continue our example of finding our best prospects for getting an active checking account. The terms *cross-sell* and *cross-buy* refer to the same concept, but from the point of view of the bank or of the customer, respectively.

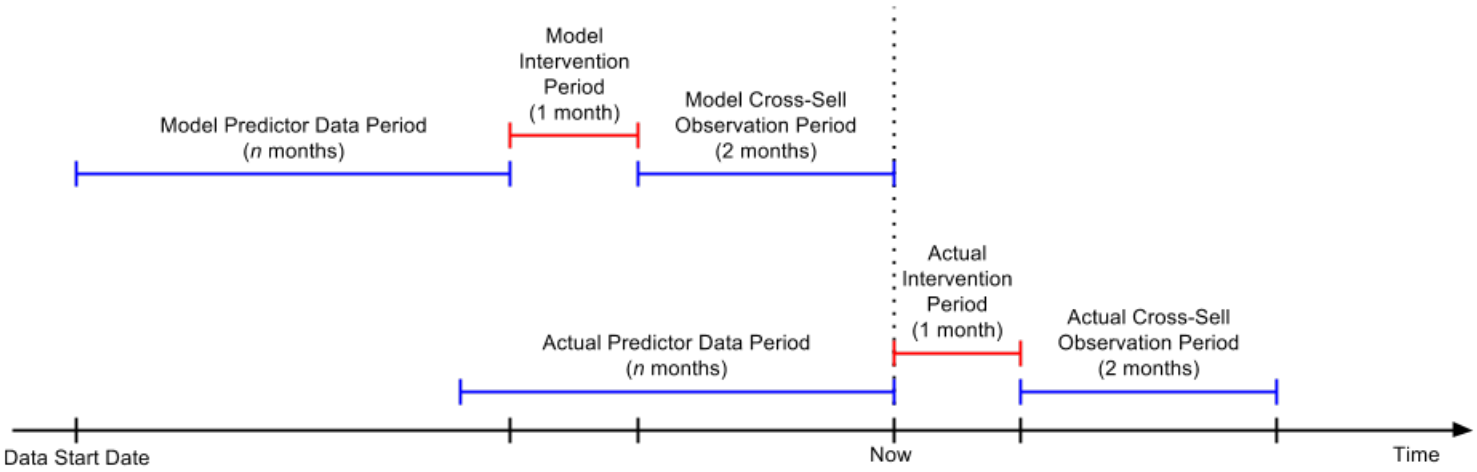


Figure 2: The scheme for duplicating our raw data into a modeling (upper) and a scoring (lower) data set.

DATA PREPARATION

We're building a *statistical model*, an equation that will tell us the probability that a customer will get an active checking account 2-3 months later. Before we can do that, we'll need to prepare the data, which involves three steps applied to raw data from our core processor (customer demographic, account, and transactional data):

- *Duplicating the data* into two data sets that are almost the same: one for building the statistical model, and the other for using that statistical model to give us our forecasts.
- *Building our variables* for both of the above data sets.
- *Partitioning the data* (the first one above, for building the statistical model).

We'll explain the reason behind each step and how we're implementing it. We'll then show results from simulated data inspired by real data from STCU, a credit union in Spokane, WA.

DUPLICATING THE DATA

A statistical model is an equation where the *input variables* are from the known data and the output variable is the unknown quantity we'd like to know. In this case, the input variables X_1, X_2, X_3, ... are attributes about the customer effective as of this point in time and the output variable is that customer's probability of getting an active checking account 2-3 months later:[2]

$$\text{Probability of getting an active checking account in next 2-3 months} = f(X_1, X_2, X_3, ...)$$

where f is some function we don't yet know. Once we have the function f, we'll use the input variables X_1, X_2, X_3 and get our probability of getting the active account. This may sound simple, but to figure out what that function is, we need to use mathematical algorithms on data at a point in time when we can see which customers actually got an active checking account 2-3 months later. In other words,

- To *build* the statistical model, we need to use data as of three months ago, coupled with which customers got an active checking account 2-3 months later (which we know).
- To *use* the statistical model, we need to use data as of now, which will tell us which customers are likely to get an active checking account 2-3 months later (which we don't know).

Since the statistical model requires input and output variables to be defined in the same way (whether we're building or using the statistical model), the time interval for the input variables must be the same length for both creating and using the statistical models. Therefore, from our raw data we'll create two data sets adjusted for the time intervals, as shown in Figure 2:

[2] 2-3 months later gives us a month to intervene and hopefully persuade the customer to get the active checking account.

- The data set for *building* the statistical model will include input variables up to three months in the past, plus cross-sell data for the last two months (i.e., which customers got an active checking account).
- The data set for *using* the statistical model will include only input variables, for a time period moved forward by three months.

For consistency (i.e., some months have 31 days, some have 30, some have 28 or 29), we actually use groups of 4 weeks rather than 1 month, even when we call it a month in Figure 2.

We can efficiently code this in SAS by defining a macro:

```
%MACRO prepareData( dataSet );

  %LOCAL now1 now2 now ... crossSellEndDate;

  PROC SQL NOPRINT;
    SELECT MAX( effectiveDate )
      INTO :now1
      FROM customer_accounts;
    SELECT MIN( tranPostDate ), MAX( tranPostDate )
      INTO :startDate, :now2
      FROM customer_transactions;
  QUIT;

  %LET now = %SYSFUNC( MIN( &now1, &now2 ) );

  %IF &dataSet = modeling %THEN %DO;

      %LET predictorStartDate = &startDate;
        %* starting at the earliest transaction date ;
      %LET predictorEndDate = %EVAL( &now - 84 );
        %* ending three months ago ;
      %LET crossSellStartDate = %EVAL( &now - 56 + 1 );
        %* starting two months ago ;
      %LET crossSellEndDate = &now;
        %* ending now ;

      %END;
    %ELSE %IF &dataSet = scoring %THEN %DO;

      %LET predictorStartDate = %EVAL( &startDate + 84 );
        % starting at the earliest transaction date plus three months ;
      %LET predictorEndDate = &now;
        % ending now ;

      %END;

      [SAS CODE FOR PULLING/PROCESSING THE DATA, USING THE MACRO VARIABLES ABOVE]

%MEND prepareData;
```

We can now create both data sets using the exact same process for each of them with the time periods shifted, as in Figure 2:

```
%prepareData( modeling )
%prepareData( scoring )
```

BUILDING OUR VARIABLES

For both of the data sets described above, we'll build variables that might be predictive of a customer getting an active checkings account. We don't care if these variables are *actually* predictive, as the statistical modeling process will figure that out. But the statistical modeling process is just a mathematical algorithm that doesn't understand human behavior. It needs to know ahead of time which variables to try out. So it's our job to give it those variables to try out, which of course we have to create.

Here are some examples of variables we can try out:

- *Indirect Customer?*: Is the customer an *indirect* customer, who only has a car loan with no other account? These customers often behave very differently from regular ones.
- *Months Being a Customer*: How long has that person been a customer?
- *Number of Checking Accounts*: Does the customer already have one (or more) checking accounts? It might be easier to engage someone with one (or more) inactive checking accounts than someone without an account already set up.
- *Months since Last Account Opened*: When was the last time the customer opened any account? If it's relatively recently, perhaps s/he is more likely to get an active checking account.
- *Mean Monthly Number of Transactions*: How many transactions does the customer have on any account? More transactions would probably lead to an active checking account.
- *Mean Transaction Amount*: What's the total transaction amount a customer typically has in a month? A higher transaction amount could also lead to an active checking account.
- *Transaction Recency*: When was the last transaction (other than automatic transactions like interest)?
- *External Deposit Recency*: When was the last external deposit? A recent one (if there are any) could lead to an active checking account.

Within SAS, we can code these variables into the `%prepareData` macro we previously defined so that we do the exact same process for both time intervals. As shown below, we have to be sure that we confine ourselves to certain transactions type codes (`tranTypCode`).

```
PROC SQL NOPRINT;
  CREATE TABLE predictorData1 AS
    SELECT
        id_customer,
        MAX( ( &predictorEndDate - tranPostDate )/7 ) AS tranRecency
          LABEL='Transaction Recency (Weeks)',
        MEAN( ABS( tranAmt ) ) AS meanTranAmt LABEL='Mean Transaction Amount',
        N( tranAmt )/ MAX( INTCK( 'month', tranPostDate, &now, 'c' ) )
          AS meanNTransPerMonth LABEL='Mean # Transactions per Month'
      FROM customer_transactions
      WHERE
        tranPostDate BETWEEN &predictorStartDate AND &predictorEndDate AND
        UPCASE( tranTypeCode ) IN ( 'CCC', 'CCD', ... 'WTHD' )
      GROUP BY id_customer;
  CREATE TABLE predictorData2 AS
    SELECT
        id_customer,
        MAX( ( &now - tranPostDate )/7 ) AS depRecency
          LABEL='External Deposit Recency (Weeks)'
      FROM customer_transactions
      WHERE
        tranPostDate BETWEEN &predictorStartDate AND &predictorEndDate AND
        UPCASE( tranTypeCode ) = 'XDEP'
      GROUP BY id_customer;
QUIT;
```

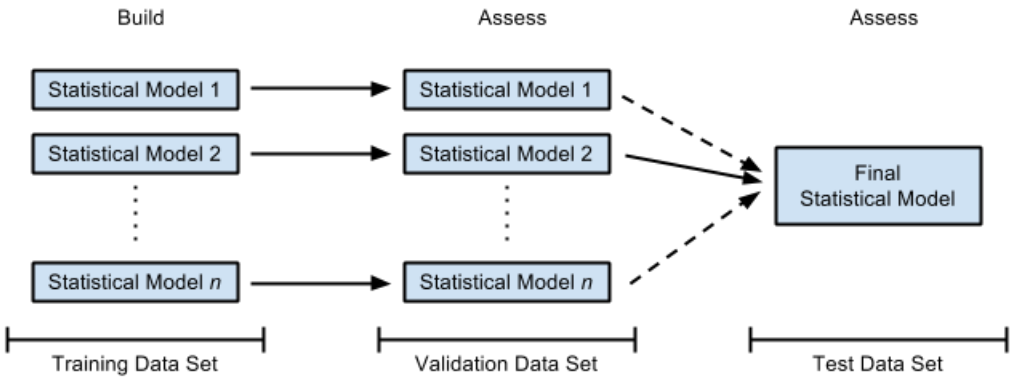


Figure 3: The three data partitions. Only one of the models makes it to the final model (in this case, model 2).

PARTITIONING THE DATA

For the modeling data set in Figure 2, we won't build *one* statistical model for our forecasts. Instead, we'll build several of them, choose the one that gives us the best results, then give an estimate of how accurate those results are. While this process may sound simple, we can't use the same data set for each of these steps, since that could give us biased results (which would be bad). To understand this, think of the data set we use when building a statistical model. This process involves mathematical algorithms that find the equation that best fits the data. If we used the same data set to assess how well that equation fit those data points, then by definition (since the algorithm was *designed* to get the best fit between the equation and the data points) we would get a really good fit. But the whole point of building a statistical model is to predict *data that we haven't seen yet*. **If we've never actually tested how well our statistical model predicts unknown data points, then we won't know how well it forecasts unknown data until we use it.** This could be a recipe for disaster.

There's a much better way to do this. Instead of using the same data set for making the model and then testing it out, we can randomly partition the data points into three distinct sets:

- *The training data set* (60% of the data) is used to build each of the statistical models that we're trying out.
- *The validation data set* (20% of the data) is used to determine how well each of these statistical models actually forecasts cross-sell opportunities. That is, using each of the statistical models we built with the training set, we'll forecast the customers in the validation set who got an active checking account and check their accuracy. **The statistical model that has the best accuracy will be our final statistical model.**
- *The test data set* (20% of the data) is used to determine how well the final model (i.e., the winning model from the validation set) actually forecasts cross-sell opportunities. That is, we'll use the final model to forecast the customers in the test set who got an active checking account and check their accuracy. We do this to double check that everything is OK. If the accuracy is much different than it was for the validation set, it's a sign that something is wrong and we should investigate this further. Otherwise, our final model is all good!

This is illustrated in Figure 3. In SAS, we can do this with the following code at the end of our `%prepareData` macro, using a random uniform distribution with the `RAND` function:

```
DATA trainingData validationData testData;
  SET inputData;
  CALL STREAMINIT( 29 );
  randUni = RAND( 'uniform' );
  IF randUni < .6 THEN OUTPUT trainingData;
    ELSE IF randUni < .8 THEN OUTPUT validationData;
    ELSE OUTPUT testData;
RUN;
```

For our data set of 69,534 customers at the end of September 2016, we get 41,875 customers in our training set (60.22%), 13,807 customers in our validation set (19.86%), and 13,852 customers in our test set (19.92%).

BUILDING THE STATISTICAL MODELS

Building the statistical models is actually easy, as we'll just use logistic regression with different sets of explanatory variables. We can use the following code to implement this with the training set in Figure 3:

```
PROC LOGISTIC DATA=trainingData OUTMODEL=trainingModel1;
  CLASS ageTier( REF='18 and Under' ) / PARAM=ref;
  MODEL crossSell( EVENT='1' ) = ageTier monthsCust nCheck;
  ODS OUTPUT parameterEstimates = parameters_model1;
RUN;
```

A few details about this code:

- The `CLASS` statement establishes the first age tier (for 18 and under) as our reference age tier.
- In the `MODEL` statement,
  - We set the `crossSell`[3] reference level to 1 so that our statistical model predict those customers who are cross-buying, not those who are not doing so.
  - We've listed age tier (`ageTier`), months of being a customer (`monthsCust`), and number of checking accounts (`nCheck`) as our explanatory variables for this particular model.
- The `ODS OUTPUT` statement exports the parameter estimates onto a separate data set.

ASSESSING THE STATISTICAL MODELS

To assess our statistical model as shown in Figure 3, we take the model created from the training set above and apply it to our validation set. We do this in SAS with the `SCORE` statement in `PROC LOGISTIC`:

```
PROC LOGISTIC INMODEL=trainingModel1;
  SCORE DATA=validationData OUT=validationForecasts OUTROC=validationROC;
RUN;
```

The output data sets `validationForecasts` and `validationROC` will be used in our assessments as described in the next few pages. If this is our best model and we want to apply it to our test set in Figure 3, we simply change the `SCORE` statement accordingly:

```
PROC LOGISTIC INMODEL=trainingModel1;
  SCORE DATA=testData OUT=testForecasts OUTROC=testROC;
RUN;
```

Finally, when we're done and want to make forecasts of the entire data set, we change the `SCORE` statement once again:[4]

```
PROC LOGISTIC INMODEL=trainingModel1;
  SCORE DATA=inputData OUT=finalForecasts;
RUN;
```

To compare different models with the validation set, we use *gain charts*, *lift charts*, *K-S charts* and *ROC charts*, as described in the next section (as originally described in Derby (2013)).

[3] The variable `crossSell` is an indicator variable equal to 1 if the customer got an active checking account account 2-3 months in the future and 0 otherwise. The outcome of our model gives the probability that this variable is equal to 1.

[4] The `OUTROC` option won't be needed this time, since we're just making forecasts and won't be assessing them with the ROC curve.

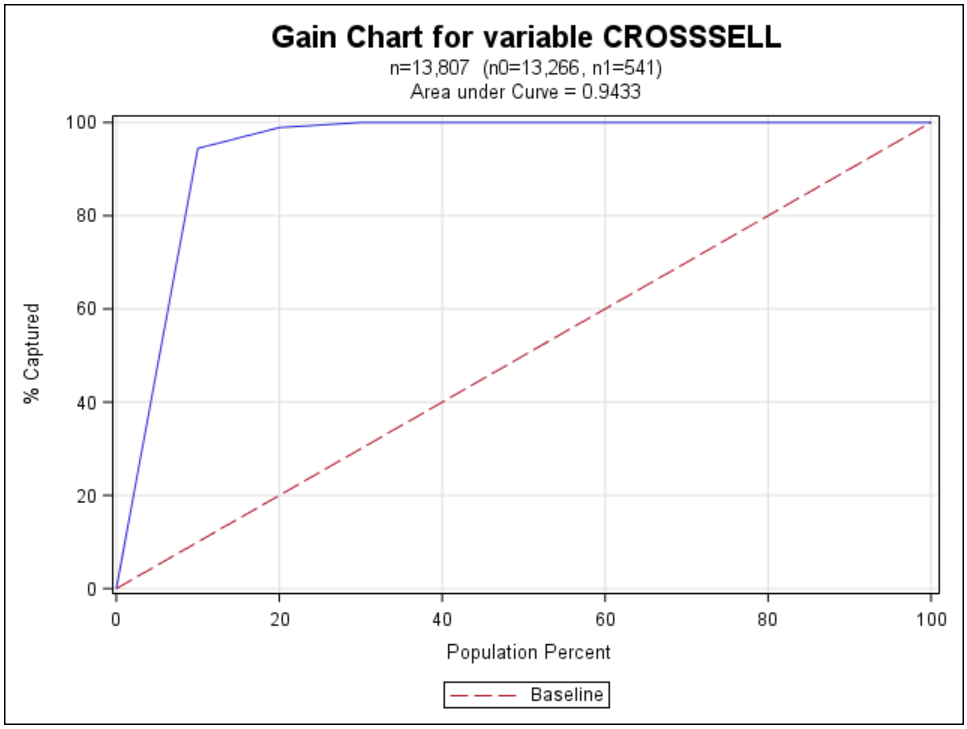


Figure 4: A gain chart.

GAIN CHARTS

A *gain chart* measures the effectiveness of a classification model as the ratio between the results obtained with and without the model. Suppose we ordered our cases by the scores (in our case, the cross-sell probability).

- If we take the top 10% of our model results, what percentage of actual positive values would we get?

In our example,

- If we take the top 10% of our results, what percentage of active checking account customers would we get?

If we then do this for 20%, 30%, etc., and then graph them, we get a gain chart as in Figure 4.[5]

For a baseline comparison, let's now order our cases (i.e., converted active checking account customers) at random. On average, if we take the top 10% of our results, we should expect to capture about 10% of our actual positive values. If we do this for all deciles, we get the straight baseline in Figure 4. If our model is any good, it should certainly be expected to do better than that! As such, the chart for our model (the solid line) should be above the dotted line.

How do we use this chart to assess our model? In general, the better our model, the steeper the solid line in our gain chart. this is commonly reflected in two statistical measurements:

- *The area under the curve*: The better the model, the closer the area under the curve is to 1. In our case (Figure 4), we have 94.33% (which is unusually high mainly because we simulated the data).
- *The 40% measure*: What percentage of our actual targets are captured by our top 40% of predicted values? In our case (Figure 4), we have 100% (which again is unusually high in this case).

In practice, these measures don't mean very much unless we compare them to measures from other models. Indeed, some phenomena are easier to predict than others.

[5]We could do this for *any* percentile, but it's typically just done for deciles.

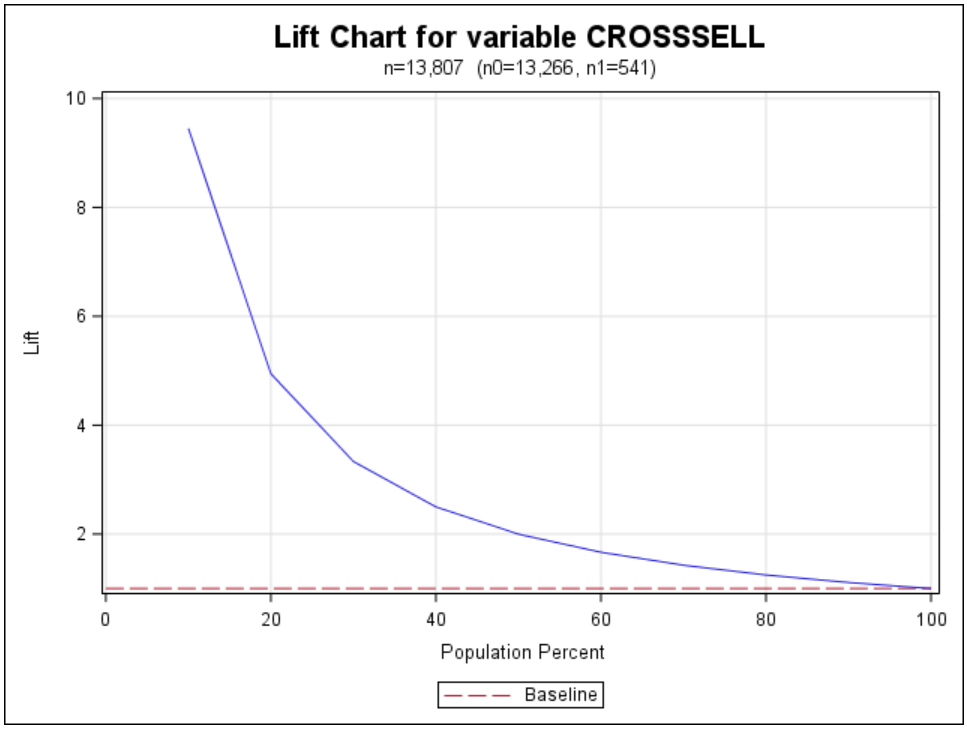


Figure 5: A lift chart.

How did we make the chart in Figure 4? SAS Institute (2011) gives us a macro that makes it relatively easy to create a gain chart. We created our own macro based on that one but creating some more details, as shown in Figure 4:

```
%makeCharts( INDATA=validationForecasts, RESPONSE=crossSell, P=p_1, EVENT=1,
  GROUPS=10, PLOT=gain, PATH=&outroot, FILENAME=Gain Chart );
```

The parameters are the following:

- `INDATA`: The input data set. (Optional: default is `_last_`, the last created data set)
- `RESPONSE`: The response variable.
- `P`: The probability/score variable.
- `EVENT`: Is an event defined when the `RESPONSE` variable is 0 or 1? (Optional: default is `1`)
- `GROUPS`: Do we want to break the data down in groups of ten (at 10% intervals) or twenty (at 5% intervals)? (Optional: default is 20)
- `PLOT`: What graph do we want to plot? Three options (all described in this paper): `gain`, `lift` or `ks`.
- `PATH`: The path of the resulting graph (as a PNG file).
- `FILENAME`: The name of the resulting graph (as a PNG file). (Optional: default is `output`)

LIFT CHART

A *lift chart* simply looks at the ratio of the gain chart results with our model and with the baseline–i.e., the ratio of the solid line to the dotted line. This is shown in Figure 5. Using our same macro from before, we have the following syntax in SAS:

```
%makeCharts( INDATA=validationForecasts, RESPONSE=crossSell, P=p_1, EVENT=1,
  GROUPS=10, PLOT=lift, PATH=&outroot, FILENAME=Lift Chart );
```

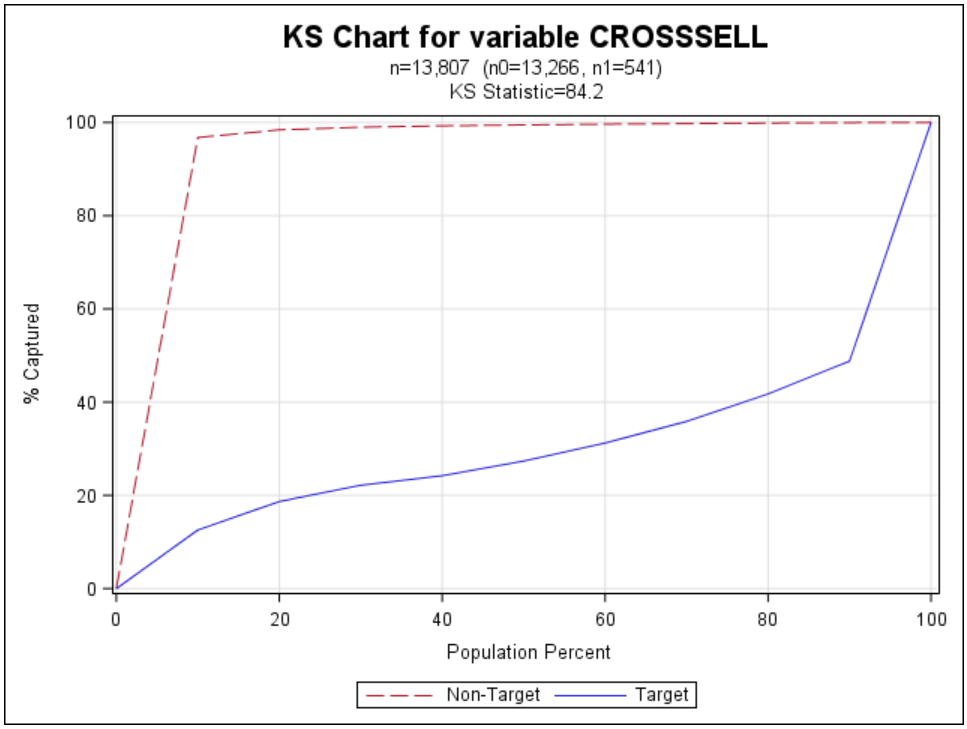


Figure 6: A K-S chart.

K-S CHART

A *K-S Chart* or *Kolmogorov-Smirnov Chart* measures the model performance in a slightly different way:

- If we look at cases with a target probability below 10%, what percentage of actual targets and non-targets would we get?

For our specific situation,

- If we look at customers with a cross-sell probability below 10%, what percentage of actual cross-buying and non-cross-buying customers would we get?

We then look at this for every decile, giving us the graph in Figure 6. We have two lines: For the target (cross-buying customers) and for the non-target (non-cross-buying customers). Depending on how we defined our target (e.g., whether we defined cross-buying as 1 or 0), the target line will be either above or below the non-target line. What's important is their maximal distance away: If we can find a probability cutoff point[6] that maximizes the difference between the targets and non-targets, we should use that one for optimal results. This maximal distance is called the *K-S statistic* or *Kolmogorov-Smirnov statistic*, which is 84.20 for our data (as before, unusually high because we simulated our data). The higher the K-S statistic, the better the model.

To code this in SAS (and generate Figure 6), we use our same macro from before:

```
%makeCharts( INDATA=validationForecasts, RESPONSE=crossSell, P=p_1, EVENT=1,
  GROUPS=10, PLOT=ks, PATH=&outroot, FILENAME=KS Chart );
```

[6]i.e., if the probability cutoff point is *x*, then we predict that a customer will get an active checking account if his/her cross-sell probability is above *x* and predict that he/she won't get an active checking account otherwise. The best cutoff point is *not* always 50%.

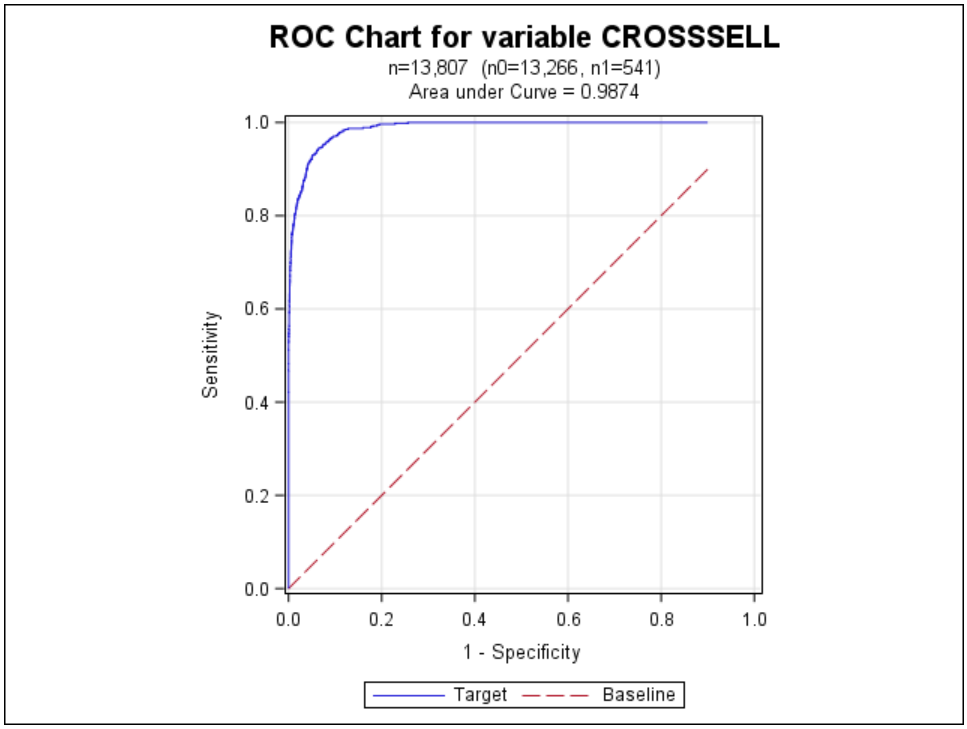


Figure 7: An ROC chart.

ROC CHART

With the K-S chart, we varied the probability cutoff values and looked at the percent of the targets and non-targets captured. With an *ROC chart* (*Receiver Operating Characteristic chart*), we simply look at the different values we would get for different probability cutoff values:[7]

- *Sensitivity*: The proportion of actual positive cases (cross-buying customers) that were correctly identified.
- *1 - Specificity*: The proportion of actual negative cases (non-cross-buying customers) that were incorrectly identified.

The area under the curve (0.9874 here, which is unusually high because we have simulated data) shows how well the model predicts the outcome. The higher it is, the better the model.

SAS makes it very easy to produce ROC charts from `PROC LOGISTIC`, but that's only possible for the training set, as it won't work when using the `INMODEL` option. Therefore, we created a macro similar to the ones before, based in part on SAS Institute (2008):

```
%makeROC( INDATA=validationForecasts, INROC=validationROC, RESPONSE=crossSell,
  P=p_1, PATH=&outroot, FILENAME=ROC Chart );
```

The result is in Figure 7.

OVERALL

Overall, we pick the model that gives us the best results (highest area under the curve for the gain or ROC chart, or the highest K-S statistic). If there's disagreement among the results (i.e., one model has the largest area under the curve for the gain chart but another one has the highest K-S statistic), we can make a judgment call.

[7] More precisely, which values would we get in our *confusion matrix* as described in Derby (2013).

RESULTS

We fit 264 different statistical models, each one with a different set of predictor variables. Using the process described in the previous pages, we chose the model that gave us the best results.

PREDICTIVE MODEL

Here is our predictive model, where the outcome is a number from 0 to 1 telling us the probability of a customer getting an active checking account:

$$\begin{aligned}\text{Probability of a customer getting an active checking account in the next 2-3 months} &= \frac{1}{\left(1+\exp\left(6.8862-\sum_{i=1}^{7}a_iX_i-\sum_{i=1}^{7}\sum_{j=1}^{7}a_ia_jX_iX_j\right)\right)}\\ &= 1/(1+\exp(6.8862-1.9874X_1-3.6251X_2-0.00335X_3+0.00139X_4\\ &\quad -0.00252X_1X_4-1.4724X_5-0.986X_6-1.2144X_7\\ &\quad +4.1597X_1X_5+3.9139X_1X_6+3.2282X_1X_7)).\end{aligned}$$

where $\exp(x) = e^x$ and

- X_1 = 1 if the customer is indirect, 0 otherwise.
- X_2 = # of checking accounts the customer has.
- X_3 = # of months the customer has had at least one account.
- X_4 = # of months since the last account was opened.
- X_5 = 1 if the customer is of age 19-30, 0 otherwise.
- X_6 = 1 if the customer is of age 31-65, 0 otherwise.
- X_7 = 1 if the customer is of age 66 or over, 0 otherwise.

However, having this equation doesn't help us understand or interpret it. We can interpret this model by using *odds*, which is the ratio of the probability of something happening to the probability of that something not happening. To relate odds to probabilities, here's a helpful rule:

- If the odds of something = x times the odds of something else and the probability of something = y times the probability of something else, then these are mathematically true:
  - If $x > 1$, then $y > 1$.
  - If $x < 1$, then $y < 1$.
  - y is always closer to 1 than x.

In other words, **if the odds increase, then the probability also increases. Likewise, if the odds decrease, then the probability also decreases**.

With this rule in mind (relating odds to probabilities), here's the interpretation of our model:[8]

- The odds of active checking for an indirect customer are $e^{1.9874}$ = 7.30 times the odds of active checking for a regular customer (with both customers of age tier 0-18, 0 months since their last accounts, with all other factors equal). Those first two qualifications are needed because we have interaction terms with indirect customer status. So **the odds increase**.
- For every additional checking account that a customer gets, his/her odds of adapting an active checking account multiply by $e^{3.6251}$ = 37.53. So **the odds increase**.
- For every month that someone is a customer, his/her odds of getting an active checking account multiply by $e^{0.00335}$ = 1.0034. So **the odds increase**. This isn't much for a single month, but can add up considerably for multiple months.
- For a regular customer (not indirect), for every month that passes since a customer's last account, his/her odds of getting an active checking account multiply by $e^{-0.00139}$ = 0.999. So **the odds decrease**.
- For an indirect customer, for every month that passes since a customer's last account, his/her odds of getting an active checking account multiply by $e^{-0.00139+0.00252}$ = 1.0011 (the coefficients are added because of the interaction term). So **the odds increase**. It's interesting that **the direction of the odds ratio changes between a regular customer and an indirect customer**.
- The odds of an active checking account for a regular (not indirect) customer aged 19-30 are $e^{1.4724}$ = 4.36 times the odds of an active checking account for a regular customer aged 18 or under, so **the odds increase**.
- The odds of an active checking account for a regular (not indirect) customer aged 31-65 are $e^{0.986}$ = 2.68 times the odds of an active checking account for a regular customer aged 18 or under, so **the odds once again increase**.
- The odds of an active checking account for a regular (not indirect) customer aged 66+ are $e^{1.2144}$ = 3.37 times the odds of an active checking account for a regular customer aged 18 or under, so **the odds once again increase**.
- The odds of an active checking account for an indirect customer aged 19-30 are $e^{1.4724-4.1597}$ = 0.0681 times the odds of an active checking account for an indirect customer aged 18 or under, so **the odds decrease**. Once again, it's interesting that **the direction of the odds ratio changes between a regular customer and an indirect customer**.
- The odds of an active checking account for an indirect customer aged 31-65 are $e^{0.986-3.9139}$ = 0.0535 times the odds of an active checking account for an indirect customer aged 18 or under, so **the odds decrease**. Once again, **the direction of the odds ratio changes between a regular customer and an indirect customer**.
- The odds of an active checking account for an indirect customer aged 66+ is $e^{1.2144-3.2282}$ = 0.1335 times the odds of an active checking account for an indirect customer aged 18 or under, so **the odds decrease**. Once again, **the direction of the odds ratio changes between a regular customer and an indirect customer**.

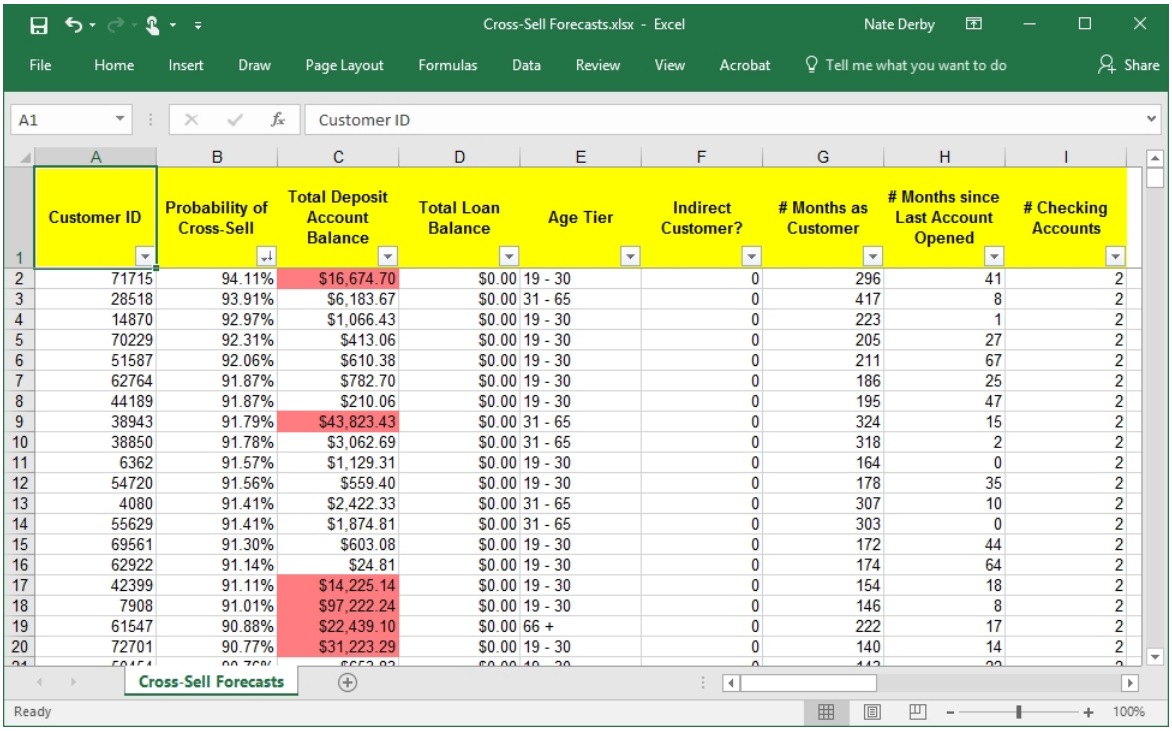
FORECASTS

From our predictive model on page 12, we can assign to each customer a cross-sell probability. We can then order these customers by that probability so that the customers with the highest probability are on the top. Furthermore, we can add the customer's aggregate deposit and loan balances, plus all the explanatory variables from the predictive model. This will give us a measure of customer value[9], plus give us indications why each customer has that high probability.

These results are shown on Figure 8 on the next page.

[8] See Quinn (2001) for a full explanation of how this interpretation comes from our model on page 12.

[9] This isn't as good as the customer lifetime value shown in Figure 1, but it's something we can quickly calculate. We can incorporate a more robust estimate of a customer lifetime value later on.



Cross-Sell Forecasts.xlsx - Excel — Nate Derby

File Home Insert Draw Page Layout Formulas Data Review View Acrobat Tell me what you want to do Share

A1 | Customer ID

	A	B	C	D	E	F	G	H	I
1	Customer ID	Probability of Cross-Sell	Total Deposit Account Balance	Total Loan Balance	Age Tier	Indirect Customer?	# Months as Customer	# Months since Last Account Opened	# Checking Accounts
2	71715	94.11%	$16,674.70	$0.00	19 - 30	0	296	41	2
3	28518	93.91%	$6,183.67	$0.00	31 - 65	0	417	8	2
4	14870	92.97%	$1,066.43	$0.00	19 - 30	0	223	1	2
5	70229	92.31%	$413.06	$0.00	19 - 30	0	205	27	2
6	51587	92.06%	$610.38	$0.00	19 - 30	0	211	67	2
7	62764	91.87%	$782.70	$0.00	19 - 30	0	186	25	2
8	44189	91.87%	$210.06	$0.00	19 - 30	0	195	47	2
9	38943	91.79%	$43,823.43	$0.00	31 - 65	0	324	15	2
10	38850	91.78%	$3,062.69	$0.00	31 - 65	0	318	2	2
11	6362	91.57%	$1,129.31	$0.00	19 - 30	0	164	0	2
12	54720	91.56%	$559.40	$0.00	19 - 30	0	178	35	2
13	4080	91.41%	$2,422.33	$0.00	31 - 65	0	307	10	2
14	55629	91.41%	$1,874.81	$0.00	31 - 65	0	303	0	2
15	69561	91.30%	$603.08	$0.00	19 - 30	0	172	44	2
16	62922	91.14%	$24.81	$0.00	19 - 30	0	174	64	2
17	42399	91.11%	$14,225.14	$0.00	19 - 30	0	154	18	2
18	7908	91.01%	$97,222.24	$0.00	19 - 30	0	146	8	2
19	61547	90.88%	$22,439.10	$0.00	66 +	0	222	17	2
20	72701	90.77%	$31,223.29	$0.00	19 - 30	0	140	14	2

Cross-Sell Forecasts

Ready 100%

Figure 8: Customer cross-sell forecasts from our final model.

In Figure 8, all aggregate balances more than $10,000 are shown in red, highlighting our more valuable customers that we should pay attention to. We then show the probable reasons for that customer's high probabilities, following our interpretations on the previous page:

- *Age Tier*: For regular customers, age tier 19-30 is the one most likely to get an active checking account.
- *Indirect Customer?*: For all age groups other than those under 18 (who wouldn't become indirect customers anyway[10]), indirect customers are less likely to get an active checking account than regular customers.
- *# Months as a Customer*: Customers for more months are more likely to get an active checking account.
- *# Months since Last Account Opened*: Customers with fewer months since their last account are more likely to get an active checking account.
- *# Checking Accounts*: Customers with more checking accounts are more likely to get an active checking account.

Overall, **the forecasts in Figure 8 give us a very convenient way to quickly focus on our high-risk, high-value customers mentioned in Figure 1**.

CAVEATS

These results are based on statistical approximations and standard assumptions from the data and not at all guaranteed. If some external unforeseen circumstance happens (such as, say, a competitor running a major marketing campaign), these results may not hold.

The forecasts listed in Figure 8 are only half of our retention strategy. **Nothing will happen unless these forecasts are paired with an effective intervention strategy** to get our customers to have an active checking account. Finding the optimal strategy for this purpose is one of our goals for the future.

[10] since no one under 18 would quality for a car loan.

CONCLUSIONS AND FURTHER SUGGESTIONS

First of all, our statistical model interpretations on page 13 prove a number of conclusions about marketing strategy. The second one is particularly notable:

- For every additional checking account that a customer gets, his/her odds of adapting an active checking account multiply by $e^{3.6251} = 37.53$ (which is rather large).

This proves the following statement, at least for this example (but possibly not true in general):

> **It's easier to get a customer with an inactive checking account to actively use it than to get a customer without a checking account at all to open one and actively use it.**

This result may sound trivial, but it's actually very important. At some financial institution marketing departments, there's been an ongoing debate about which group of customers is easier to target: Those with inactive checking accounts, or without a checking account at all. There have been arguments for both sides, but this statistical model settles this debate!

Overall, **predictive analytics can be a very powerful tool for customer cross-sell opportunities.** This paper describes a relatively simple yet effective way to predict which customers have a high cross-sell opportunity so that **we can market the *right* product to the *right* customers at the *right* time**.

However, **the techniques in this paper just scratch the surface of how we can increase cross-sell opportunities with predictive analytics**. Thomas (2010) shows how we can profile our customer attrition rates by deciles and then evaluate the biggest overall sources of attrition. Lu (2002) and Su et al. (2009) show how we can better estimate attrition risk with more complex statistical methods. These ideas can all be applied to cross-sell opportunities. Lastly, Fader (2012) introduces the idea of customer value, and Lu (2003) shows how to measure it.

REFERENCES

Derby, N. (2013), Managing and monitoring statistical models, *Proceedings of the 2013 SAS Global Forum*, paper 190-2013.
http://support.sas.com/resources/papers/proceedings13/190-2013.pdf

Derby, N. and Keintz, M. (2016), Reducing credit union member attrition with predictive analytics, *Proceedings of the 2016 SAS Global Forum*.
http://support.sas.com/resources/papers/proceedings16/11882-2016.pdf

Fader, P. (2012), *Customer Centricity: Focus on the Right Customers for Strategic Advantage*, Wharton Digital Press, Philadelphia, PA.

Karp, A. (1998), Using logistic regression to predict customer retention, *Proceedings of the Eleventh Northeast SAS Users Group Conference*.
http://www.lexjansen.com/nesug/nesug98/solu/p095.pdf

Lu, J. (2002), Predicting customer churn in the telecommunications industry – an application of survival analysis modeling using SAS, *Proceedings of the Twenty-Seventh SAS Users Group International Conference*, paper 114-27.
http://www2.sas.com/proceedings/sugi27/p114-27.pdf

Lu, J. (2003), Modeling customer lifetime value using survival analysis – an application in the telecommunications industry, *Proceedings of the Twenty-Eighth SAS Users Group International Conference*, paper 120-28.
http://www2.sas.com/proceedings/sugi28/120-28.pdf

Quinn, K. (2001), Interpreting logistic regression models.
http://nderby.org/docs/QuinnK-LogitInterp.pdf

SAS Institute (2008), Plot ROC curve with labelled points for a binary-response model.
http://support.sas.com/kb/25/018.html

SAS Institute (2011), Gains and lift plots for binary-response models.
http://support.sas.com/kb/41/683.html

Su, J., Cooper, K., Robinson, T. and Jordan, B. (2009), Customer retention predictive modeling in the healthcare insurance industry, *Proceedings of the Seventeenth SouthEast SAS Users Group Conference*, paper AD-007.
http://analytics.ncsu.edu/sesug/2009/AD007.Su.pdf

Thomas, W. (2010), Improving retention by predicting both who and why, *Proceedings of the Twenty-Third Northeast SAS Users Group Conference*.
http://www.lexjansen.com/nesug/nesug10/sa/sa08.pdf

ACKNOWLEDGEMENTS

We thank our good friends at STCU of Spokane, WA for their help and cooperation!

CONTACT INFORMATION

Comments and questions are valued and encouraged, as I depend on partnering financial institutions to do research, so please don't hesitate to contact me:

Nate Derby
Stakana Analytics
815 First Ave., Suite 287
Seattle, WA 98104-1404
nderby@stakana.com
http://nderby.org
http://stakana.com