

# Reducing Credit Union Member Attrition with Predictive Analytics

Nate Derby, Stakana Analytics, Seattle, WA  
Mark Keintz, Wharton Research Data Services, Philadelphia, PA

## ABSTRACT

As credit unions market themselves to increase their market share against the big banks, they understandably focus on gaining new members. However, they must also retain (and further engage) their existing members. Otherwise, the new members they gain can easily be offset by existing members who leave. Happily, by using predictive analytics as described in this paper, it can actually be much easier and less expensive to keep (and further cultivate) existing members than to enlist new ones.

This paper provides a step-by-step overview of a relatively simple but comprehensive approach to reduce member attrition. We first prepare the data for a statistical analysis. With some basic predictive analytics techniques, we can then identify those members who have the highest chance of leaving and the highest value. For each of these members, we can also identify why they would leave, thus suggesting the best way to intervene to retain them. We then make suggestions to improve the model for better accuracy. Finally, we provide suggestions to extend this approach to cultivating existing members and thus increasing their lifetime value.

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](http://nderby.org/docs/charts.zip).

## INTRODUCTION: THE PROBLEM WITH MEMBER ATTRITION

Suppose we're the fictitious Fisher Credit Union of Ames, Iowa, which has 70,359 members and \$928MM in assets as of June 30, 2015, with none of those members having only a car loan.<sup>1</sup> Three months before, we had 69,534 members and \$920MM in assets, so we grew their membership by 1.19% and their assets by 0.91%. That might sound good on the surface, but it's really not. Our gain of 825 members is a *net* gain. We actually gained 3541 *new* members but lost 2716 *existing* members. **Our member attrition is dramatically affecting our net membership growth.**

- If we had retained just 20% of those departing members, we would have gained a net 1368 members, for a 1.97% net membership growth.
- If we had retained 25% of those departing members, we would have had gained a net 1504 members, for a 2.16% net membership growth.
- If we had retained 50% of those departing members, we would have had gained a net 2183 members, for an amazing 3.14% net membership growth.

In fact, **retaining existing members is usually easier and less expensive than gaining new members**, since these members already know and trust the credit union, and the credit union already knows so much about them. Keeping them might be as simple as making a friendly phone call. **The key is making full use of the data we have on our members.** Since member attrition can have such a huge effect on membership growth, why not focus some attention to it? Specifically, we can focus on those members with the highest value and the highest risk of closing their accounts.

---

<sup>1</sup>Members with just a car loan typically behave differently from the normal members, so we're ignoring them to keep this analysis simple.

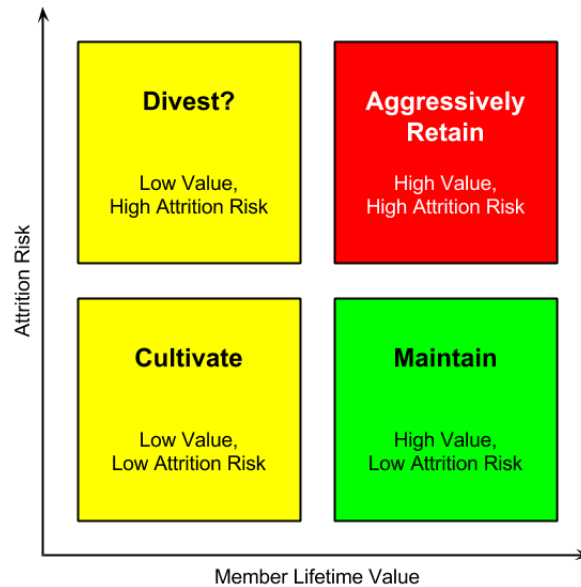


Figure 1: Member segmentation into quadrants of lifetime value and attrition risk.

**If we can identify which members are most likely to leave, we can proactively focus our marketing efforts on retaining them.** With this in mind, Figure 1 shows four attrition marketing quadrants that describe the segmentation of our members. The horizontal axis shows the *member 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 attrition risk. We can divide this region into four quadrants:

- *Maintain*: These are members with a high value and low attrition risk. These are generally our best members, and we should keep them satisfied without focusing much attention on them if possible.
- *Cultivate*: These are members with a low value and low attrition risk. We're not in danger of losing them, but they aren't growing our credit union. We should cultivate these members to have a higher value.
- *Divest?*: These are members with a low value and high attrition risk. For a general business, these are the customers that we wouldn't really mind losing, since they're low value and would require some effort to keep. However, in the tradition of credit unions serving the underserved, profit isn't always the main motivation, and there can be good reasons why we wouldn't want to let these members leave. Still, these members won't grow our credit union. If we're going to keep these members, we'll need to appropriately balance our focus on them with a focus on our high-value members as well!
- *Aggressively Retain*: These are members with a high value and high attrition risk, and **they are the ones we should pay the most attention to**. If we lose these members, it will make a large difference to our asset size (either now or in the future).

### So how can we use our data to focus on these high-value, high-risk members?

Thomas (2010) presents a basic but very useful approach using logistic regression, and Karp (1998) provides details on the logistic regression procedure itself. Lu (2002) and Lu (2003) predict attrition and lifetime value using survival analysis, whereas Su et al. (2009) use statistical clustering. Both of these are more complex methods.

In this paper, we'll follow the example of Thomas (2010), showing a basic but very useful approach that can produce quick results once the code infrastructure is in place.

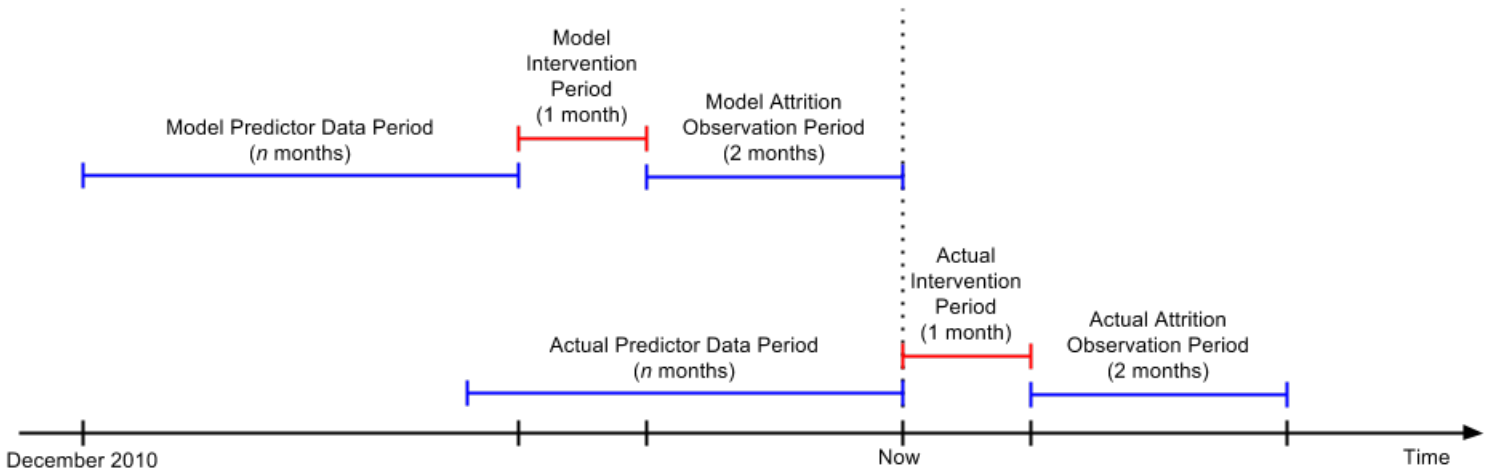


Figure 2: The scheme for duplicating our raw data into a modeling (upper) and a scoring (lower) data set.

## DATA PREPARATION

We're building a *statistical model*, an equation that will tell us the probability that a member will close his/her 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 (member 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 Verity Credit Union of Seattle.

## 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  $X_1, X_2, X_3, \dots$  are attributes about the member effective as of this point in time and the output variable is that member's risk of leaving the credit union 2-3 months later:<sup>2</sup>

$$\text{Probability of leaving 2-3 months later} = f(X_1, X_2, X_3, \dots)$$

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 leaving. 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 members actually left 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 members left 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 members are likely to leave 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:

<sup>2</sup>2-3 months later gives us a month to intervene and hopefully change the outcome so that the member retains his/her account.

- The data set for *building* the statistical model will include input variables up to three months in the past, plus attrition data for the last two months (i.e., which members closed their accounts).
- 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 ... attritionEndDate;

  PROC SQL NOPRINT;
    SELECT MAX( effectiveDate )
      INTO :now1
      FROM member_accounts;
    SELECT MIN( tranPostDate ), MAX( tranPostDate )
      INTO :startDate, :now2
      FROM member_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 attritionStartDate = %EVAL( &now - 56 + 1 );
    %* starting two months ago ;
    %LET attritionEndDate = &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 member closing his/her 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:

- *Transaction Recency*: When was the last transaction (other than automatic transactions like interest)?
- *External Deposit Recency*: When was the last external deposit?
- *Recent Large Transaction*: Was there a recent large transaction? (Above the 90th or 95th percentile of all transactions over the most recent 2 or 4 weeks)
- *Small Number of Transactions*: Was there a recent small number of transactions? (Above the 90th percentile for the monthly number of transactions)
- *Large Number of Transactions*: Was there a recent large number of transactions? (Below the 10th percentile for the monthly number of transactions)
- *A Seasonality Component*: This is a dummy variable to determine if certain members are more likely to close their accounts in the summer, or during the Christmas season.

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

The percentiles can be computed with PROC UNIVARIATE, and all of these data sets can be joined by a simple MERGE statement within a DATA step.

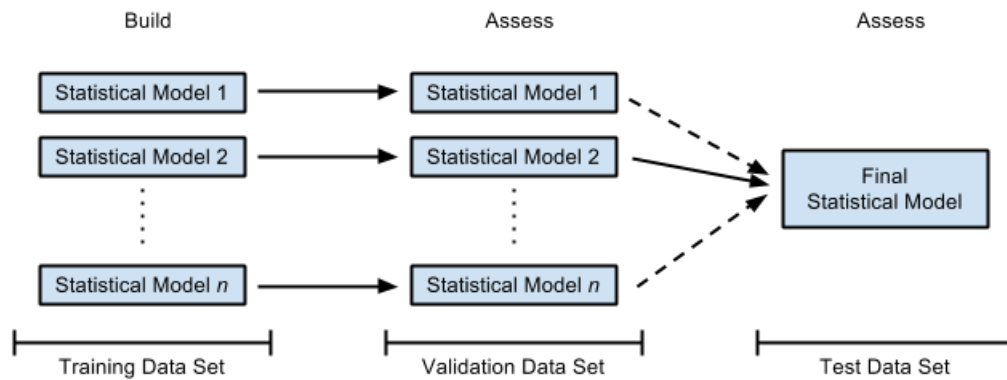


Figure 3: The three data partitions. Only one of the models makes it to the final model (in this case, model 2).

## 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 attrition. That is, using each of the statistical models we built with the training set, we'll forecast the members in the validation set who closed their accounts 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 attrition. That is, we'll use the final model to forecast the members in the test set who closed their accounts 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 members at the end of March 2015, we get 41,875 members in our training set (60.22%), 13,807 members in our validation set (19.86%), and 13,852 members 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=trainingModell;  
  CLASS ageTier( REF='18 and Under' ) / PARAM=ref;  
  MODEL attrition( EVENT='1' ) = depRecency ageTier lom nProducts calls;  
  ODS OUTPUT parameterEstimates = parameters_modell;  
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 `attrition`<sup>3</sup> reference level to 1 so that our statistical model predict those members who are leaving, not those who are staying.
  - We've listed external deposit recency (`depRecency`), age tier (`ageTier`), length of membership (`lom`), number of products (`nProducts`), and number of customer service calls (`calls`) 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=trainingModell;  
  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=trainingModell;  
  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:<sup>4</sup>

```
PROC LOGISTIC INMODEL=trainingModell;  
  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)).

<sup>3</sup>The variable `attrition` is an indicator variable equal to 1 if the member closed his/her 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.

<sup>4</sup>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.

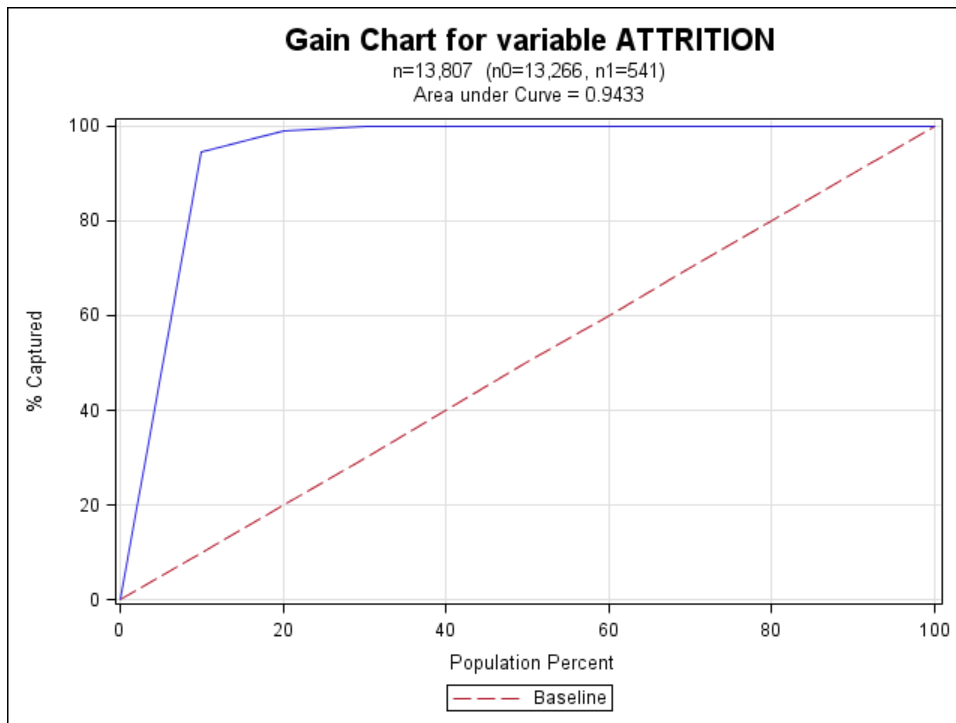


Figure 4: A gain chart.

## GAIN CHARTS

A *gain chart* illustrates the effectiveness of a classification model calculated as the ratio between the results obtained with and without the model. Suppose we ordered our cases by the scores (in our case, the attrition risk).

- 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 model results, what percentage of actual members who left 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.<sup>5</sup>

For a baseline comparison, let's now order our cases (i.e., members who closed their accounts) 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.

<sup>5</sup>We could do this for *any* percentile, but it's typically just done for deciles.



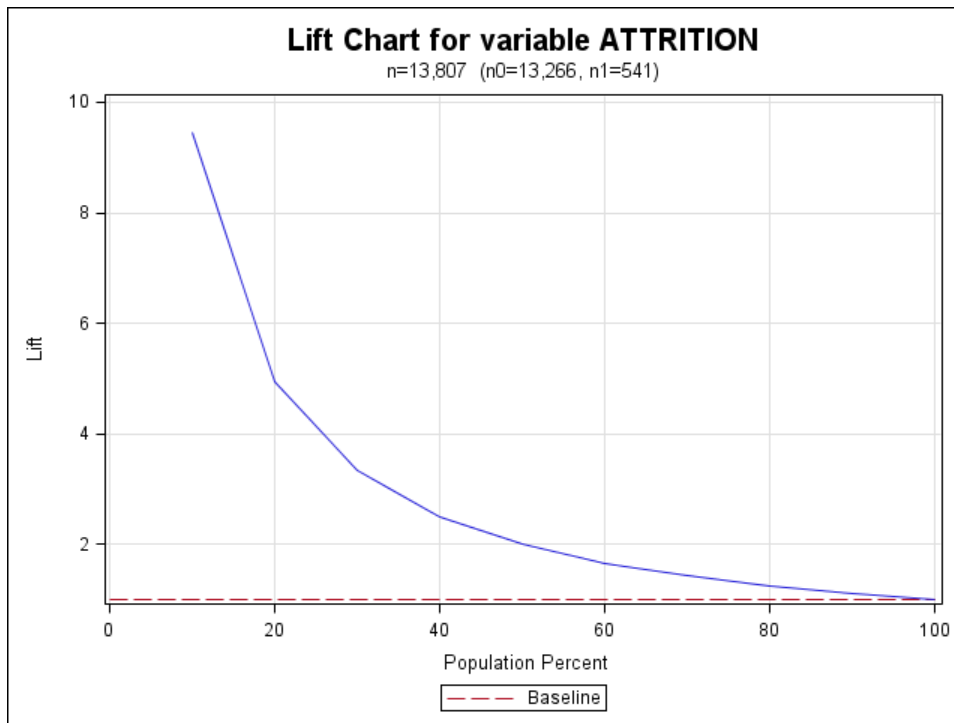


Figure 5: A lift chart.

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=attrition, 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=attrition, P=p_1, EVENT=1,
  GROUPS=10, PLOT=lift, PATH=&outroot, FILENAME=Lift Chart );
```

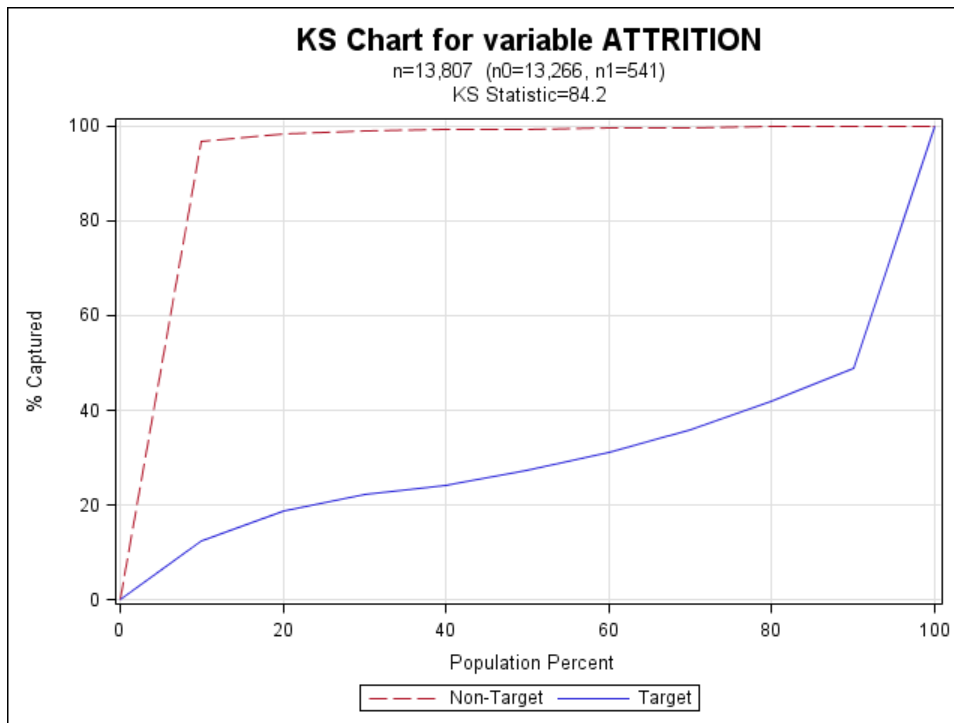


Figure 6: A K-S chart.

## 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 members with an attrition risk below 10%, what percentage of actual members who left and members who stayed 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 (attrition members) and for the non-target (non-attrition members). Depending on how we defined our target (e.g., whether we defined attrition 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<sup>6</sup> 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=attrition, P=p_1, EVENT=1,
  GROUPS=10, PLOT=ks, PATH=&outroot, FILENAME=KS Chart );
```

<sup>6</sup>i.e., if the probability cutoff point is  $x$ , then we predict that a member will close his/her account if his/her attrition risk is above  $x$  and predict that he/she won't close his/her account otherwise. The best cutoff point isn't always 50%!

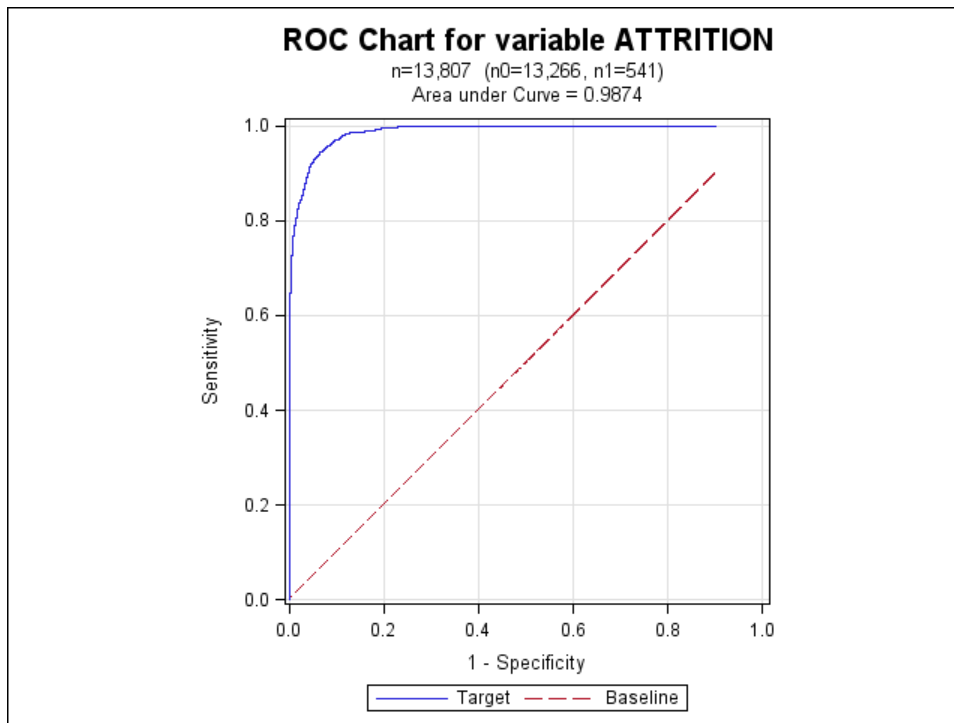


Figure 7: An ROC chart.

## 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:<sup>7</sup>

- *Sensitivity*: The proportion of actual positive cases (attrition members) that were correctly identified.
- *1 - Specificity*: The proportion of actual negative cases (non-attrition members) 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=attrition,
          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 an educated judgment call.

<sup>7</sup>More precisely, which values would we get in our *confusion matrix* as described in Derby (2013).

## RESULTS

We fit 25 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 risk of a member closing his/her account:

Risk of a member  
closing his/her account

$$\begin{aligned} \text{in the next 2-3 months} &= \frac{1}{\left(1 + \exp\left(0.0715 - \sum_{i=1}^{15} a_i X_i\right)\right)} \\ &= 1 / (1 + \exp(0.0715 - 0.00425X_1 - 2.0201X_2 - 3.2913X_3 - 2.9120X_4 - 3.2148X_5 \\ &\quad - 2.7609X_6 - 3.3691X_7 - 2.3624X_8 - 2.6723X_9 - 2.7469X_{10} \\ &\quad - 4.4689X_{11} - 4.1651X_{12} + 0.0938X_{13} + 4.9355X_{14} - 2.9855X_{15})). \end{aligned}$$

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

- $X_1$  = external deposit recency (weeks).
- $X_2$  = 1 if the member is of age 19-24, 0 otherwise.
- $X_3$  = 1 if the member is of age 25-30, 0 otherwise.
- $X_4$  = 1 if the member is of age 31-35, 0 otherwise.
- $X_5$  = 1 if the member is of age 36-40, 0 otherwise.
- $X_6$  = 1 if the member is of age 41-45, 0 otherwise.
- $X_7$  = 1 if the member is of age 46-50, 0 otherwise.
- $X_8$  = 1 if the member is of age 51-55, 0 otherwise.
- $X_9$  = 1 if the member is of age 56-60, 0 otherwise.
- $X_{10}$  = 1 if the member is of age 61-65, 0 otherwise.
- $X_{11}$  = 1 if the member is of age 66-70, 0 otherwise.
- $X_{12}$  = 1 if the member is of age 71 or over, 0 otherwise.
- $X_{13}$  = length of membership (months).
- $X_{14}$  = number of products.
- $X_{15}$  = number of customer service calls in the past month.

However, having this equation doesn't help us interpret it. We can interpret our model using *odds* (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, then the probability of something =  $y$  times the probability of something else, where
  - 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, here's the interpretation of our model:<sup>8</sup>

- For every week that a member goes without an external deposit, his/her odds of attrition multiply by  $e^{0.00425} = 1.004$ . (This number may not sound high, but it's still significant, especially after multiple weeks)
- The odds of a member aged 19-24 leaving is  $e^{2.0201} = 7.5$  times the odds of a member 18 or under leaving.
- The odds of a member aged 25-30 leaving is  $e^{3.2913} = 26.9$  times the odds of a member 18 or under leaving.
- The odds of a member aged 31-35 leaving is  $e^{2.9120} = 18.4$  times the odds of a member 18 or under leaving.
- The odds of a member aged 36-40 leaving is  $e^{3.2148} = 24.9$  times the odds of a member 18 or under leaving.
- The odds of a member aged 41-45 leaving is  $e^{2.7609} = 15.8$  times the odds of a member 18 or under leaving.
- The odds of a member aged 46-50 leaving is  $e^{3.3691} = 29.0$  times the odds of a member 18 or under leaving.
- The odds of a member aged 51-55 leaving is  $e^{2.3624} = 10.6$  times the odds of a member 18 or under leaving.
- The odds of a member aged 56-60 leaving is  $e^{2.6723} = 14.5$  times the odds of a member 18 or under leaving.
- The odds of a member aged 61-65 leaving is  $e^{2.7469} = 15.6$  times the odds of a member 18 or under leaving.
- The odds of a member aged 66-70 leaving is  $e^{4.4689} = 87.3$  times the odds of a member 18 or under leaving. (This is likely due to members dying, which is unavoidable)
- The odds of a member aged 71+ leaving is  $e^{4.1651} = 64.4$  times the odds of a member 18 or under leaving. (This is likely due to members dying, which is unavoidable)
- For every month that a member continues a membership, his/her odds of attrition multiply by  $e^{-0.0938} = 0.91$ . (So it decreases, as we would expect)
- For every product that a member signs up for, his/her odds of attrition multiple by  $e^{-4.9355} = 0.0072$ . (So it sharply decreases, as we would expect)
- For every customer service call that a member makes, his/her odds of attrition multiply by  $e^{2.9855} = 19.8$ . (So it sharply increases, as we would expect)

## FORECASTS

From our predictive model on page 12, we can assign to each member an attrition risk. We can then order these members by that risk so that the members with the highest attrition risk are on the top. Furthermore, we can add the member's aggregate balance, plus all the explanatory variables from the predictive model. This will give us a measure of member value<sup>9</sup>, plus give us indications why each member is at risk.

These results are shown on Figure 8. Here, all aggregate balances more than \$10,000 are shown in red, highlighting our more valuable members that we should pay attention to. We then highlight the probable reasons for that member's high attrition rate in yellow, following our interpretations above:

- *Age Tier*: Age tiers of 25-30, 36-40, and 46-50 have the highest attrition risks, so we highlight them.<sup>10</sup>
- *Length of Membership (Months)*: Members with smaller lengths of membership have a higher attrition risk, so we highlight all lengths of membership of 6 months or less.
- *External Deposit Recency (Weeks)*: Longer external deposit recencies have a higher attrition risk, so we highlight all external deposit recencies of 10 weeks or more.
- *Number of Products*: Members with a fewer number of products have a higher attrition risk, so we highlight all listings of only 1 product.
- *Customer Service Calls (past month)*: Members with more calls have a higher attrition risk, so we highlight 5 or more service calls.

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

<sup>8</sup>See Quinn (2001) for a full explanation of how this interpretation comes from our model on page 12.

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

<sup>10</sup>We're ignoring age tiers of 66-70 and 71+, since those attrition rates are likely due to member death, which is of course unavoidable!

Member ID	Risk of Attrition in 2-3 months	Aggregate Balance	Age Tier	Length of Membership (Months)	External Deposit Recency (Weeks)	Number of Products	Customer Service Calls (past month)
38050	99.99%	\$24,219.38	36 - 40	44	178.01	1	8
37605	99.99%	\$2,868.88	36 - 40	6	14.11	1	7
21035	99.99%	\$21,795.65	61 - 65	14	28.66	1	7
27143	99.99%	\$9,474.69	41 - 45	21	87.23	1	7
8511	99.99%	\$202.41	19 - 24	16	53.89	1	7
1006	99.99%	\$6,473.96	56 - 60	24	20.07	1	7
32628	99.99%	\$13,850.80	66 - 70	22	27.34	1	6
29364	99.99%	\$8,423.36	25 - 30	24	96.49	2	8
70252	99.99%	\$610.98	19 - 24	32	8.39	1	7
69985	99.99%	\$11,398.00	25 - 30	56	197.92	1	7
51167	99.99%	\$14,239.38	41 - 45	17	59.12	1	6
68801	99.99%	\$105,157.41	25 - 30	22	41.49	1	6
29989	99.99%	\$17,112.47	46 - 50	35	140.29	1	6
5638	99.99%	\$17,542.99	36 - 40	64	105.08	1	7
23305	99.99%	\$7,555.09	46 - 50	30	20.18	1	6
47997	99.99%	\$12,247.90	46 - 50	34	91.16	1	6
9277	99.99%	\$47,612.51	71 +	8	32.31	1	5
52856	99.99%	\$11,555.09	31 - 35	31	126.65	1	6
22334	99.99%	\$13,870.93	56 - 60	24	2.84	1	6
25296	99.99%	\$5,404.15	31 - 35	34	138.28	1	6

Figure 8: Member attrition forecasts from our final model. The many members with the 99.99% attrition risk is highly unusual and just a reflection of our simulated data.

### 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 change the decision of our members at high risk of attrition. Finding the optimal strategy for this purpose is one of our goals for the future.

### CONCLUSIONS AND FURTHER SUGGESTIONS

**Predictive analytics can be a very powerful tool for member retention.** This paper describes a relatively simple yet very effective way to predict which members have a high attrition risk so that we can focus on them. However, this needs to be paired with an effective intervention strategy to be effective! Predicting alone won't solve anything.

However, **the techniques in this paper just scratch the surface of how we can reduce member attrition with predictive analytics!** Thomas (2010) shows how we can profile our member 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. Lastly, Lu (2003) shows how we can develop a better measure of lifetime member value.

This approach can also be used to predict other aspects of member behavior, such as when a member will buy a car or a home (and thus need a car loan or a mortgage). As such, **we can also use these techniques to further cultivate our members** by marketing additional products to them at just the right time.

## 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>
- 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 Verity Credit Union of Seattle for their help and cooperation!

## CONTACT INFORMATION

**Comments and questions are valued and encouraged, as we depend on partnering credit unions to do our research.** As such, please don't hesitate to contact us:

Nate Derby  
Stakana Analytics  
815 First Ave., Suite 287  
Seattle, WA 98104-1404  
[nderby@stakana.com](mailto:nderby@stakana.com)  
<http://nderby.org>  
<http://stakana.com>

Mark Keintz  
Wharton Research Data Center, The Wharton School  
3819 Chestnut Street, Suite 300  
Philadelphia, PA 19104  
[mkeintz@wharton.upenn.edu](mailto:mkeintz@wharton.upenn.edu)  
<http://whartonwrds.com>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.