Paper 3218-2015

A mathematical model for optimizing product mix and customer lifetime value

Atul Thatte

Dr. Goutam Chakraborty

ABSTRACT

Companies that offer subscription based services (such as telecom and fitness clubs) must evaluate the tradeoff between non-contract customers, who yield lower value due to lower lifetime, and customers who commit to a longer term contract in return for lower price. The objective, of course, is to maximize the Customer Lifetime Value (CLV). This tradeoff must be evaluated not only at the time of customer acquisition, but throughout the customer's tenure, particularly for fixed-term contract customers whose contract becomes due for renewal.

In this paper, we present a mathematical model that maximizes the revenue against this tradeoff with customer lifetime value. We use revenue as a proxy for CLV, to keep things simple. The model can be easily extended to take into account cash flow discounting, account for activity based costing, etc. The model is presented in the context of a cohort of new customers, some of whom are non-contract customers and others are on contracts. The model optimizes the number of non-contract customers to be switched to contracts as well as the number of contract renewals that should be pursued, at various term lengths and other factors.

We estimate customer life using a discrete-time survival model, with time varying covariates related to contract expiration and product changes. Thereafter, an optimization model is used to find the optimal trade-off between revenue and customer lifetime value.

INTRODUCTION

This paper is organized as follows. In section 1, we define key terms used. In section 2, we describe the specific problems we want to solve using a mathematical model presented in later sections. In particular, we describe why a mathematical model is required in the first place. Section 3 presents the solution approach, assumptions, data, exploratory analyses, and two sub-models in the overall solution – a survival model that estimates customer lifetime in response to retention treatment and an optimization model that computes the best tradeoff between revenue and customer life. Section 4 presents results. Finally, section 5 concludes the paper with potential extensions and applications of the model.

KEY TERMS

Customer churn: an event marking the loss of a customer to competition

Discrete time survival model: a statistical model that estimates the probability that a customer does not churn beyond t months of tenure

Survival: probability that a customer does not churn (survives) beyond time t

Hazard: the churn rate at time t, conditional upon survival up to time t-1

Decision variables: variables for which optimal values are to be determined. These optimal values maximize or minimize an objective function (of these variables) given certain constraints expressed as functions of the decision variables.

Linear Optimization model: a model that find the optimal values for certain decision variables in order to maximize or minimize a linear objective function, subject to linear constraints

Mixed-Integer Linear Optimization model: a model that find the optimal values for certain decision variables, some or all of which must have integral values, in order to maximize or minimize a linear objective function, subject to linear constraints

Non-linear Optimization model: a model that find the optimal values for certain decision variables in order to maximize or minimize a non-linear objective function, subject to non-linear constraints

Feasible solution: one of a set of possibly infinite solutions to an optimization model

Optimal Solution: one or more feasible solutions that maximize or minimize an objective function

Non-contract customers(abbreviated MTM):customers who have very short or no contracts/commitments

Contract customers: customers who desire greater value by signing long term contracts

THE PROBLEM

We discuss the problem being solved in this paper in the context of a hypothetical fitness club that offers memberships on a non-contract and fixed-term contract basis. Contracts are offered for either one or two years, after which customers have a choice to either renew the contract, switch to a non-contract schedule or churn. Non-contract customers, of course, can switch to a contract at any time. The average prices, per month, for non-contract, 1-year contract and 2-year contract customers are $55, $50 and $45 respectively. Customers could signup via inbound (call center and online) and outbound channels. As part of the firm's retention efforts, price deltas (which are discounts on the original price) are offered to non-contract and contract customers (at the time their contract expires), in order to incent them to extend their tenure with the firm. The objective is to maximize revenue over the lifetime of the customer by optimizing the tradeoff between price deltas and customer lifetime.

THE SOLUTION

The solution involves efficiently evaluating combinations of product type, channel, original prices, price deltas and customer life, to arrive at the optimal revenue. The revenue is computed by multiplying the difference between original price and the price delta with the average customer life for customers in each combination of product type and channel. Obviously, doing this manually is impractical because of the computational intensity, among other reasons.

OVERALL APPROACH

In order to optimize the tradeoff between the price deltas and customer life, we first estimate the impact of various price deltas on customer survival using survival modeling techniques. The mean customer life is estimated by the following covariates: product type, channel, original price and price deltas. It is computed by calculating the area under the survival curve for each combination of these variables.

The estimated mean customer life is then combined with the available data to determine the optimal revenue. The optimal revenue is determined by a mixed integer linear optimization model, which optimally allocates customers across various price deltas, in order to maximize the revenue.

ASSUMPTIONS

As mentioned above, the optimization model allocates customers to various price deltas. It is assumed that customers will accept the optimal price deltas/discounts offered to them. This may not be very restrictive because the number of customers who accept offers can be influenced via various marketing tactics. The optimization model helps by segmenting beneficial price deltas. Furthermore, the optimization model can be easily extended to incorporate data on the probability that a price delta would be accepted.

DATA

The data used for the models in this paper has been simulated to reflect how customers might react to various retention treatments and how their lifetime might be affected as a result of such treatments. In particular, the following factors, that affect customer lifetime, are simulated:

Buyer's remorse: the customer may sign up for service, regret it soon thereafter and may take advantage of the "satisfaction guarantee" period, which many companies offer, to get out of a contract.

Channel of acquisition: while customers acquired via e-commerce websites may have a low Cost-per-Acquisition (CPA), they very often tend to be behave differently and may have different lifetimes than customers acquired through other channels. As such, the channel of acquisition needs to be considered as a variable for the survival model. In the optimization model, we consider channels to be either inbound or outbound, with no further sub-classification.

Contract term length: the lack of a contract, or the term length, if a contract exists, is another factor that will likely affect customer lifetime.

Retention treatments and incentive value: The price discounts offered to non-contract customers in lieu of a contract commitment would, of course, impact customers' lifetime. Similarly, around the time when a customer's contract term is about to expire, a price discount offered to renew the contract, to the same or a different term length product, will also impact the customer's lifetime.

A DISCRETE TIME SURVIVAL MODEL TO ESTIMATE CUSTOMER LIFE

The purpose of the survival model is to estimate the probability that a customer will survive (continue to be a customer) beyond t months. In response to a renewal or product switch, at a certain price point, the survival probability will be different. When plotted, survival curves show the probability of survival on the vertical axis and the elapsed customer life (in months) on the horizontal axis. As mentioned before, the average life is the area under the survival curves. Hence, "higher" the survival curve, greater will be the average life.

Some results from the survival model are shown below. The following survival curves show the improvement in survival if a 12 month term contract is renewed, as opposed to not:

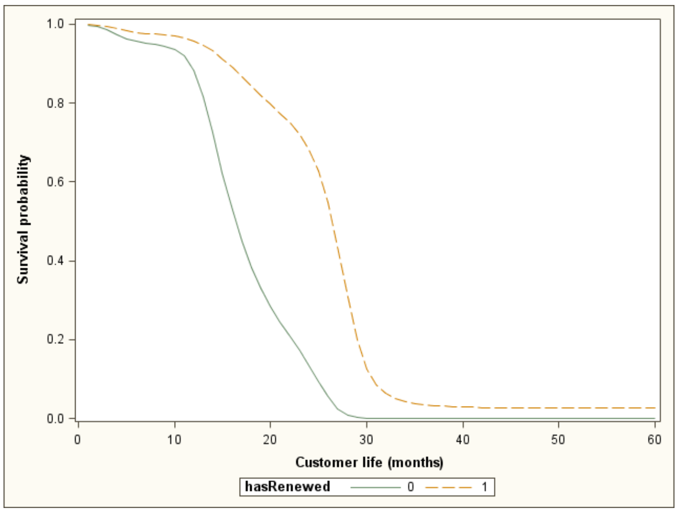

Figure 1. Customer survival: renewal vs. no renewal

Here, the curve associated with "hasRenewed=1" shows the customer survival if a 12 month contract customer renews the contract. As one would expect, the area under the curve for customers who renew is greater than that for customers who do not.

Similarly, switching a non-contract customer to a contract improves average life, as shown below:

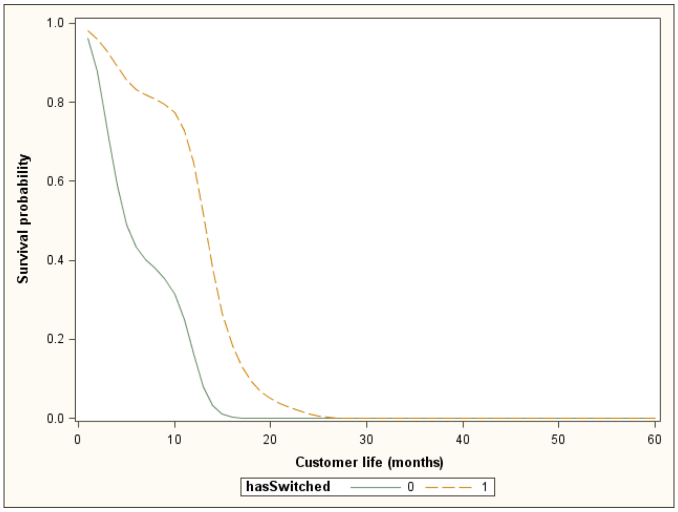

Figure 2. Customer survival: switch to contract vs. no switch

Here, the curve associated with “hasSwitched=1” shows the customer survival for non-contract customers who have switched to contracts.

Now that we can estimate the average life from survival curves, we will estimate the relationship between the renewal/switch discount and the improvement in average life. We do this using linear regression models, one for each combination of product type, original price and channel.

For various (non-zero) renewal and switch discounts, the improvement in average lifetime, for each combination of product type, original price and channel, is shown below:

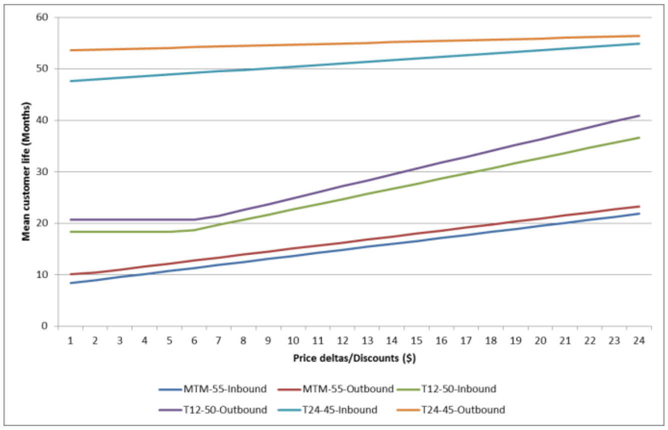

Figure 3. Average Customer Life vs. Price Deltas

The price deltas are on the horizontal axis and the mean lifetime is on the vertical axis. Each series represents one combination of product type, original price and channel. We have used only the average original price for each product type.

The corresponding regression equations are given below:

Product Type	Original price		Channel	Regression equations
MTM	$	55	Inbound	avgLife = 7.81053 + 0.58388*price_delta
MTM	$	55	Outbound	avgLife = 9.23125 + 0.58473*price_delta
12-Month Contract	$	50	Inbound	avgLife = 12.70491 + 0.99809*price_delta
12-Month Contract	$	50	Outbound	avgLife = 13.40528 + 1.14641*price_delta
24-Month Contract	$	45	Inbound	avgLife = 47.27727 + 0.31571*price_delta
24-Month Contract	$	45	Outbound	avgLife = 53.47193 + 0.12090*price_delta

Table 1. Regression equations to estimate average life from price delta

These regression equations quantify the increase in customer loyalty as the discount increases. The average life values are stored in a dataset named "est_avglife_final2", which will be read into the SAS/OR® model discussed in a later section.

A MATHEMATICAL MODEL TO OPTIMIZE CLV

In this section we'll create an optimization model to allocate customers across price deltas in order to maximize revenue. We use the following notation for product types and channels. From this point onwards, the non-contract, 12-month contract and 24-month contract product types are abbreviated to non-contract, T12 and T24 respectively. Also, the inbound and outbound channels are identified by the integers 0 and 1 respectively. Furthermore, we assume that non-contract, T12 and T24 product types have $55, $50 and $45 as their original prices respectively. The variance in prices for each of these product types is ignored for the sake of simplicity. However, the optimization model can be easily extended to incorporate multiple original prices per product type.

We will assume the following additional business constraints:

1. Lower and upper bound on product mix: There must always be at least 5% and at most 40% non-contract customers. The optimization model must prevent over-allocation to a particular product type.
2. Minimum revenue requirement: Discounting cannot cause the revenue to fall below $30.

The following structural constraints must also be observed:

1. Total allocation: The optimal allocation across price deltas must add up to the total number of customers we started with
2. The decision variable values must be integral and non-negative

The conceptual model formulation is as follows:

$$\text{Max revenue} = \sum_{\substack{pt \in \text{prod type} \\ c \in \text{channel} \\ pr \in \text{orig price} \\ pd \in \text{price delta}}} (\text{allocation}[pt,c,pr,pd] \times (pr-pd) \times \text{CustomerLifetime}[pt,c,pr,pd])$$

Subject to:

Constraint minProdMix: $$\sum_{\substack{c \in \text{channel} \\ pr \in \text{orig price}}} \text{allocation}[\text{"MTM"},c,pr,0] \geq 0.05 \times \sum_{\substack{pt \in \text{prod type} \\ c \in \text{channel} \\ pr \in \text{orig price} \\ pd \in \text{price delta}}} \text{allocation}[pt,c,pr,pd]$$

Constraint maxProdMix: $$\sum_{\substack{c \in \text{channel} \\ pr \in \text{orig price}}} \text{allocation}[\text{"MTM"},c,pr,0] \leq 0.40 \times \sum_{\substack{pt \in \text{prod type} \\ c \in \text{channel} \\ pr \in \text{orig price} \\ pd \in \text{price delta}}} \text{allocation}[pt,c,pr,pd]$$

Constraint minRev: $$\sum_{\substack{pt \in \text{prod type} \\ c \in \text{channel} \\ pr \in \text{orig price} \\ pd \in \text{price delta}}} \text{allocation}[pt,c,pr,pd] = 0 \ \forall \ (pr-pd) < 30$$

Constraint totalAlloc $$\left\{ \begin{matrix} pt \in \text{prod type} \\ c \in \text{channel} \\ pr \in \text{orig price} \end{matrix} \right\} : \sum_{pd \in \text{price delta}} \text{allocation}[pt,c,pr,pd] = total\_customer\_count$$

Constraint IntegralNonNegativity: $$\text{allocation}[pt,c,pr,pd] \text{ integer} \geq 0 \ \forall \begin{matrix} pt \in \text{prod type} \\ c \in \text{channel} \\ pr \in \text{orig price} \\ pd \in \text{price delta} \end{matrix}$$

THE SAS/OR® MODEL

Sets and parameters:

The ptype denotes the product types MTM, T12 and T24. The channel set has two members: 0 for inbound and 1 for outbound. Price deltas can be integers between 0 and 24 (inclusive). Also, as mentioned before, we assume that fixed original prices for each product. This is encoded in a matrix in the p_orig_price parameter.

```
set<str> ptype=/MTM T12 T24/;
set<num> channel=/0 1/;
set<num> price_delta = 0..24;
set<num> orig_price = /55 50 45/;
/*Assume only one original price per product type*/
num p_orig_price{ptype, orig_price} = [
                                              1       0       0
                                              0       1       0
                                              0       0       1
                                    ];
```

We optimize the revenue for 600 customers, 100 within each combination of the product type, channel and original price:

```
num p_cust_count{ptype, channel, orig_price} init 100;
```

Next, we read in the average life values from the "est_avglife_final2" dataset into the "p_avgLife" parameter.

```
num p_avgLife{ptype, channel, orig_price, price_delta} init 0;
read data d.est_avglife_final2 into [ptype channel orig_price price_delta] p_avgLife=avgLife;
```

Decision variables:

We index the alloc decision variable by the product type, channel, original price and price delta. The fourth constraint (integrality and non-negativity) is addressed here.

```
        var alloc{ptype, channel, orig_price, price_delta} integer >= 0;
```

Constraints:

The first three constraints described above are coded as follows:

```
/*Constraint #1: MTM cannot be more than 40% of total allocation but must be at least 5%*/
con min prodMix:
        sum{c in channel, pr in orig_price}alloc["MTM", c, pr, 0]
                >=
        0.05* sum{pt in ptype,c in channel,pr in orig_price,pd in price_delta}alloc[pt,c,pr,pd];
con max prodMix:
        sum{c in channel, pr in orig_price}alloc["MTM", c, pr, 0]
                <=
        0.40* sum{pt in ptype,c in channel,pr in orig_price,pd in price_delta}alloc[pt,c,pr,pd];
/* Constraint #2: Revenue cannot drop below $30*/
con minRev:sum{pt in ptype,c in channel,pr in orig_price,pd in price_delta:pr-pd<30}
                        alloc[pt,c,pr,pd]=0;
/* Constraint #3: Allocation must add up to the start counts. Also, p_orig_price control which
product type can have which price*/
con totCount{pt in ptype, c in channel, pr in orig_price}:
        sum{pd in price_delta} alloc[pt, c, pr, pd]
                =
        p_cust_count[pt, c, pr]*p_orig_price[pt, pr];
```

Finally the objective is:

```
/*Maximize revenue. The parameter p_avgLife contains average life estimates*/
max rev=sum{pt in ptype, c in channel, pr in orig_price, pd in price_delta}alloc[pt, c, pr,
pd]*(pr-pd)*p_avgLife[pt, c, pr, pd];
```

THE RESULTS

If we did not have the average life estimates from the survival model, and the optimization model to allocate customers to price deltas, we would make $634,522 by taking no action at all. In other words, if we did nothing and kept customers at their original prices, the revenue we would get from the pre-optimization scenario would be $634,522. However, the optimization model maximizes the value and the post-optimization scenario yields revenue of $839,103. That's $204,581 more than the pre-optimization version! The comparison with and without the model is given below:

Revenue Post-Optimization

				Price deltas							
Count	**ptype**	**channel**	**orig_price**	**$ -**	**$ 1**	**$ 19**	**$ 20**	**$ 21**	**$ 22**	**$ 23**	**$ 24**
100	No contract	Inbound	$ 55	$ -	$ -	$ -	$ -	**$ 68,245**	$ -	$ -	$ -
100	No contract	Outbound	$ 55	**$ 16,719**	$ -	$ -	**$ 51,268**	$ -	$ -	$ -	$ -
100	Contract - 1 yr	Inbound	$ 50	$ -	$ -	**$ 120,973**	$ -	$ -	$ -	$ -	$ -
100	Contract - 1 yr	Outbound	$ 50	$ -	$ -	**$ 131,880**	$ -	$ -	$ -	$ -	$ -
100	Contract - 2 yrs	Inbound	$ 45	$ -	**$ 211,809**	$ -	$ -	$ -	$ -	$ -	$ -
100	Contract - 2 yrs	Outbound	$ 45	$ -	**$ 238,208**	$ -	$ -	$ -	$ -	$ -	$ -
			Total	**$ 16,719**	**$ 450,018**	**$ 252,853**	**$ 51,268**	**$ 68,245**	$ -	$ -	$ -
										Total	**$ 839,103**

Revenue Pre-Optimization

				Price deltas							
Count	**ptype**	**channel**	**orig_price**	**$ -**	**$ 1**	**$ 19**	**$ 20**	**$ 21**	**$ 22**	**$ 23**	**$ 24**
100	No contract	Inbound	$ 55	**$ 45,587**	$ -	$ -	$ -	$ -	$ -	$ -	$ -
100	No contract	Outbound	$ 55	**$ 55,731**	$ -	$ -	$ -	$ -	$ -	$ -	$ -
100	Contract - 1 yr	Inbound	$ 50	**$ 91,492**	$ -	$ -	$ -	$ -	$ -	$ -	$ -
100	Contract - 1 yr	Outbound	$ 50	**$ 103,322**	$ -	$ -	$ -	$ -	$ -	$ -	$ -
100	Contract - 2 yrs	Inbound	$ 45	**$ 154,850**	$ -	$ -	$ -	$ -	$ -	$ -	$ -
100	Contract - 2 yrs	Outbound	$ 45	**$ 183,539**	$ -	$ -	$ -	$ -	$ -	$ -	$ -
										Total	**$ 634,522**

Table 2. Comparison of pre and post optimization revenue

For the MTM product, the allocation to zero price-delta indicates that those customers remained on the MTM product and did not sign a contract. For T12 and T24, an allocation into a non-zero price delta takes into account the fact that the original price was the revenue during the contract and the price delta took effect after 12 months of tenure for T12 and 24 months for T24. After the contract expired the revenue per month was the difference between the original price and the price delta.

As is evident, the post-optimization allocation identifies price deltas that should be offered to customers having various product types, channels of acquisition and original prices. This optimal allocation can now be used to create targeted offers (in conjunction with propensity models for retention offer acceptance) to maximize the revenue.

OTHER APPLICATIONS AND EXTENSIONS

This paper discussed revenue maximization. This model could be extended to take into account costs and other business metrics. Also, predictive models to estimate propensity to accept retention offers could be developed and used to fine tune the retention strategy.

REFERENCES

Allison, P.D., Survival Analysis Using the SAS System: A Practical Guide, 2nd edition, SAS Institute, Cary, NC, SAS Institute, Inc., 2010.

H. Paul Williams, Model Building in Mathematical Programming, 5th edition, Wiley, 2013

APPENDIX

SAS CODE: SURVIVAL MODEL

```
/*Sample training and validation*/
data d.customer_train d.customer_val;
       set d.customers2;
       if ranuni(0) < 0.5 then output d.customer_train;
       else output d.customer_val;
run;
%let path=C:\SAS GF 2015 Paper\data;
%let knots=4 8 13 18 25 29 33;
%let nk=7;

data expanded_train;
       set d.customer_train;
       g=0;
       do t=1 to tenure;
              array k{&nk} _temporary_ (&knots);
              array csb{&nk};
              do i=1 to &nk;
                 csb[i]=(t>k[i])*(t-k[i])**3-t**3+3*k[i]*t**2-3*k[i]**2*t;
              end;

              if t=tenure then g=churned;

              swapFlag=0;
              swapDelta=0;
              if t>=swapMonth then do;
                     swapFlag=1;
                     swapDelta=swapValue;
              end;

              renewFlag=0;
              renewDelta=0;
              if t>=renewMonth then do;
                     renewFlag=1;
                     renewDelta=renewValue;
              end;

              rolloverFlag=0;
              if t>=rolloverMonth then do;
                     rolloverFlag=1;
              end;

              output;
       end;
       drop i;
       keep custid
                     g
                     t
                     csb:
                     swapFlag
                     swapDelta
                     renewFlag
                     renewDelta
                     rolloverFlag
                     orig_price
                     ptype
                     orig_channel
                     hasRenewed
                     hasSwapped
                     swapValue
```

```
                    swapMonth
                    renewValue
                    renewmonth
                    tenure
                    ;
run;


ods output parameterestimates=pe;
proc logistic data=expanded_train(drop=renewValue renewFlag swapValue swapFlag
rolloverFLag renewMonth swapMonth);
       class ptype orig_channel / param=ref;
       model g(ref='0')=t csb1-csb&nk ptype orig_channel /*dum:*/ swap: renew:
/*rollover:*/ orig_price /*tenure*/
             / link=glogit selection=none;
run;


proc sql;
select substr(variable,4) into: csbs separated by ','
from pe
where response='1' and substr(variable,1,3)='csb';
quit;

data _null_;
call symput('ncsb',1+length("&csbs")-length(compress("&csbs",',')));
run;

data _null_;
file "&path\score1.txt";
if _n_=1 then do;
   array k{&nk} _temporary_ (&knots);
   do i=1 to &nk;
      if i in (&csbs) then put 'csb' i '=(t>' k[i] ')*(t-' k[i]
      ' )**3-t**3+3*' k[i] '*t**2-3*' k[i] '**2*t;';
   end;
end;
length variable $20 plus $4;
set pe    end=last;
by response;
if variable='Intercept' then variable='eta1=1';
plus='+';
if last.response then plus=';';
if classval0='' then put variable '*' estimate best16. plus;
else do;
       put '(' variable '=' classval0 $quote15. ')*' estimate best16. plus;
/*   if type=2 then put '(' variable '=' classval0 $quote10. ')*'
   estimate best16. plus;
   else put '(' variable '=' classval0 ')*' estimate best16. plus;
*/
end;
if last then do;
   put 'shf1=exp(eta1)/(1+exp(eta1));';
   put 'hf=shf1;';
end;
run;
/*
*/
/*Validation*/

data expanded_val;
       set d.customer_val;
       g=0;
```

```
	do t=1 to 60;
		array k{&nk} _temporary_ (&knots);
		array csb{&nk};
		do i=1 to &nk;
		   csb[i]=(t>k[i])*(t-k[i])**3-t**3+3*k[i]*t**2-3*k[i]**2*t;
		end;

		if t=tenure then g=churned;

		swapFlag=0;
		swapDelta=0;
		if t>=swapMonth then do;
			swapFlag=1;
			swapDelta=swapValue;
		end;

		renewFlag=0;
		renewDelta=0;
		if t>=renewMonth then do;
			renewFlag=1;
			renewDelta=renewValue;
		end;

		rolloverFlag=0;
		if t>=rolloverMonth then do;
			rolloverFlag=1;
		end;

		output;
	end;
	drop i;
	keep custid
			g
			t
			csb:
			swapFlag
			swapDelta
			renewFlag
			renewDelta
			rolloverFlag
			orig_price
			ptype
			orig_channel
			hasRenewed
			hasSwapped
			swapValue
			swapMonth
			renewValue
			renewmonth
			tenure
			;
run;

data scored_val;
	set expanded_val;
	%include "&path\score1.txt";
run;

proc sort data=scored_val;
	by custid t;
run;

data scored_val_surv;
```

```
        set scored_val;
        retain surv;
        by custid;
        if first.custid then do;
                surv_lag = 1;
        end;
        else do;
                surv_lag = surv;
        end;
        One_minus_hf=1-hf;
        surv = surv_lag*(1-hf);

        *drop surv_lag;
run;

proc sql;
create table scored_avglife as
select custid, tenure, hasRenewed, hasSwapped, orig_channel, orig_price, ptype,
max(renewDelta) as renewDelta, max(renewFlag) as renewFlag, max(renewMonth) as
renewMonth, max(renewValue) as renewValue, max(rolloverFlag) as rolloverFlag,
max(swapDelta) as swapDelta, max(swapFlag) as swapFlag, max(swapMonth) as swapMonth,
max(swapValue) as swapValue, count(*) as tcount, sum(surv) as avgLife
from scored_val_surv
group by 1,2,3,4,5,6,7
;
```

SAS CODE: OPTIMIZATION MODEL PARAMETERS

```
/*Compute average life as area under the survival curve*/
proc sql;
        create table avgLife_val as
        select custid, sum(surv) as avgCustLife
        from scored_val_surv
        group by 1
;

proc sql;
create table d.customer_val2 as
select a.custid, a.ptype, a.orig_price, a.orig_channel, a.swapValue, a.renewValue,
b.avgCustLife
from d.customer_val a
inner join avgLife_val b
        on a.custid=b.custid
;

/*Model average life as a function of price delta*/
proc sql;
create table mtm_avglife as
select ptype,
        orig_price,
        case when orig_channel="Outbound" then 1 else 0 end as channel,
        swapValue as price_delta,
        avg(avgCustLife) as avgLife
from customer_val2
where ptype="MTM" and orig_price=55
group by 1,2,3,4
order by 1,2,3,4
;
proc reg data=mtm_avglife(where=(price_delta gt 0 and channel eq 0));
        model avgLife=price_delta;
run;/*avgLife = 7.81053 + 0.58388*price_delta;*/
proc reg data=mtm_avglife(where=(price_delta gt 0 and channel eq 1));
        model avgLife=price_delta;
```

```
run;/*avgLife = 9.23125 + 0.58473*price_delta;*/

proc sql;
create table t12_avglife as
select ptype,
       orig_price,
       case when orig_channel="Outbound" then 1 else 0 end as channel,
       renewValue as price_delta,
       avg(avgCustLife) as avgLife
from customer_val2
where ptype="T12" and orig_price=50
group by 1,2,3,4
order by 1,2,3,4
;
proc reg data=t12_avglife(where=(price_delta gt 0 and channel eq 0));
       model avgLife=price_delta;
run;/*avgLife = 12.70491 + 0.99809*price_delta;*/
proc reg data=t12_avglife(where=(price_delta gt 0 and channel eq 1));
       model avgLife=price_delta;
run;/*avgLife = 13.40528 + 1.14641*price_delta;*/

proc sql;
create table t24_avglife as
select ptype,
       orig_price,
       case when orig_channel="Outbound" then 1 else 0 end as channel,
       renewValue as price_delta,
       avg(avgCustLife) as avgLife
from customer_val2
where ptype="T24" and orig_price=45 /*and renewValue <= 10000*/
group by 1,2,3,4
order by 1,2,3,4
;
proc reg data=t24_avglife(where=(price_delta gt 0 and channel eq 0));
       model avgLife=price_delta;
run;/*avgLife = 47.27727 + 0.31571*price_delta;*/
proc reg data=t24_avglife(where=(price_delta gt 0 and channel eq 1));
       model avgLife=price_delta;
run;/*avgLife = 53.47193 + 0.12090*price_delta;*/

data d.est_avglife;
       ptype="MTM";
       orig_price=55;
       channel=0;
       do price_delta=1 to 24;
              avgLife = 7.81053 + 0.58388*price_delta;
              output;
       end;
       channel=1;
       do price_delta=1 to 24;
              avgLife = 9.23125 + 0.58473*price_delta;
              output;
       end;
       ptype="T12";
       orig_price=50;
       channel=0;
       do price_delta=1 to 24;
              avgLife = 12.70491 + 0.99809*price_delta;
              output;
       end;
       channel=1;
       do price_delta=1 to 24;
              avgLife = 13.40528 + 1.14641*price_delta;
```

```
			output;
		end;
		ptype="T24";
		orig_price=45;
		channel=0;
		do price_delta=1 to 24;
			avgLife = 47.27727 + 0.31571*price_delta;
			output;
		end;
		channel=1;
		do price_delta=1 to 24;
			avgLife = 53.47193 + 0.12090*price_delta;
			output;
		end;
run;
proc sql;
create table est_avglife2_1 as
select ptype,
		orig_price,
		case when orig_channel="Outbound" then 1 else 0 end as channel,
		renewValue as price_delta,
		avg(avgCustLife) as avgLife
from customer_val2 a
where ptype="MTM" and orig_price=55 and price_delta=0
group by 1,2,3,4
;

proc sql;
create table est_avglife2_2 as
select ptype,
		orig_price,
		case when orig_channel="Outbound" then 1 else 0 end as channel,
		renewValue as price_delta,
		avg(avgCustLife) as avgLife
from customer_val2
where ptype="T12" and orig_price=50 and price_delta=0
group by 1,2,3,4
;
proc sql;
create table est_avglife2_3 as
select ptype,
		orig_price,
		case when orig_channel="Outbound" then 1 else 0 end as channel,
		renewValue as price_delta,
		avg(avgCustLife) as avgLife
from customer_val2
where ptype="T24" and orig_price=45 and price_delta=0
group by 1,2,3,4
;
data d.est_avglife_zero;
		set
			est_avglife2_1
			est_avglife2_2
			est_avglife2_3;
run;

data d.est_avglife_final;
		set d.est_avglife
			d.est_avglife_zero;
run;
proc sort data=d.est_avglife_final;
		by ptype orig_price channel price_delta;
run;
```

```
proc sql;
create table d.est_avglife_final2 as
select a.ptype,
             a.orig_price,
             a.channel,
             a.price_delta,
             case when a.avgLife <= b.avgLife then b.avgLife else a.avgLife end as
avgLife
from d.est_avglife_final a
inner join est_avglife_zero b
       on a.ptype = b.ptype
       and a.orig_price=b.orig_price
       and a.channel=b.channel
;

proc sql;
select ptype, channel, sum(orig_price*avgLife) as noOptRev
from d.est_avglife_zero
group by 1,2
;
```

SAS CODE: OPTIMIZATION MODEL

```
proc optmodel;
       set<str> ptype=/MTM T12 T24/;
       set<num> channel=/0 1/; /*0 => Inbound => Outbound*/
       set<num> price_delta = 0..24;
       set<num> orig_price = /55 50 45/;
       /*Assume only one original price per product type*/
       num p_orig_price{ptype, orig_price} = [
                                                     1     0     0
                                                     0     1     0
                                                     0     0     1
                                              ];

       num p_cust_count{ptype, channel, orig_price} init 100;

       num p_avgLife{ptype, channel, orig_price, price_delta} init 0;
       read data d.est_avglife_final2 into [ptype channel orig_price price_delta]
p_avgLife=avgLife;

       var alloc{ptype, channel, orig_price, price_delta} integer >= 0;

       /*Constraint #1: MTM cannot be more than 40% of total allocation but must be at
least 5%*/
       con min_prodMix:
              sum{c in channel, pr in orig_price}alloc["MTM", c, pr, 0]
                     >=
              0.05* sum{pt in ptype, c in channel, pr in orig_price, pd in
price_delta}alloc[pt, c, pr, pd];
       con max_prodMix:
              sum{c in channel, pr in orig_price}alloc["MTM", c, pr, 0]
                     <=
              0.40* sum{pt in ptype, c in channel, pr in orig_price, pd in
price_delta}alloc[pt, c, pr, pd];
       /*Constraint #2: Revenue cannot drop below $30*/
       con minRev:
              sum{pt in ptype, c in channel, pr in orig_price, pd in price_delta: pr-pd
< 30}alloc[pt, c, pr, pd]=0;
       /*Constraint #3: Allocation must add up to the start counts*/
       con totAlloc{pt in ptype, c in channel, pr in orig_price}:
```

```
            sum{pd in price_delta} alloc[pt, c, pr, pd]
                   =
            p_cust_count[pt, c, pr]*p_orig_price[pt, pr];
      /*Maximize revenue. Penalize discounting outbound*/
      max rev=sum{pt in ptype, c in channel, pr in orig_price, pd in 
price_delta}alloc[pt, c, pr, pd]*(pr-pd)*p_avgLife[pt, c, pr, pd];
solve;
      create data d.optalloc from [ptype channel orig_price price_delta] = {pt in 
ptype, c in channel, pr in orig_price, pd in price_delta}alloc = alloc[pt, c, pr, pd];
quit;
```

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Atul Thatte

Email: atul.thatte@gmail.com

Atul Thatte is a Senior Manager of Advanced Transaction and Consumption Analytics at TXU Energy, the largest retail electricity provider in Texas. At TXU Energy, Atul leads a team of seasoned analytics professionals to shape TXU's analytics strategy and provide a competitive advantage through measurable applications of mathematical models. In addition to creating business value through advanced analytics, his professional interests include teaching practical applications of advanced analytics. As an adjunct faculty member (Marketing) at Oklahoma State University, he immensely enjoys teaching marketing optimization, which applies advanced mathematical modeling techniques to solve problems in the marketing domain.

Atul's academic qualifications include a Master's degree in Operations Research (Southern Methodist University), a Bachelor's degree in Mathematics (Bombay University) and a Graduate Certificate in Data Mining (Oklahoma State University).

Dr. Goutam Chakraborty, Oklahoma State University, Stillwater OK

Email: goutam.chakraborty@okstate.edu

Dr. Goutam Chakraborty is Ralph A. and Peggy A. Brenneman professor of marketing and founder of SAS® and OSU data mining certificate and SAS® and OSU business analytics certificate at Oklahoma State University. He has published in many journals such as Journal of Interactive Marketing, Journal of Advertising Research, Journal of Advertising, Journal of Business Research, etc. He has chaired the national conference for direct marketing educators for 2004 and 2005 and co-chaired M2007 data mining conference. He has over 25 years of experience in using SAS® for data analysis. He is also a Business Knowledge Series instructor for SAS®.