ASSIGNMENT 7: S-Mobile - Predicting Customer Churn

Section 81
Trevor Cornell
Dain Hall

Preliminaries

Determine notebook defaults:

Load packages:

Read in the data:

```
load("smobile_churn.Rdata")
smobile <- smobile %>% mutate(churn=factor(churn))
smobile.train <- smobile %>% filter(training==1)
smobile.test <- smobile %>% filter(training==0)
set.seed(1234)
```

Assignment answers

PART 1

1 - Develop Models to Predict

After trying a variaety of models, it seems that logistic regressiondoes perform the best of the group. And if there were any doubt as to choosing one of the other tests, the ability to interpret the logistic regression makes it the clear choice.

```
fm <- as.formula(churn ~ revenue + mou + overage + roam + changem + changer + dropvce +
                   blckvce + unansvce + custcare + threeway + months + uniqsubs + phones +
                   eqpdays + age + agemiss + children + creditaa + refurb + occprof +
                   occcler + occcrft + occstud + occhmkr + occret + occself + travel +
                   retcalls + refer + incmiss + income + mcycle )
lr <- glm(fm, family=binomial, data=smobile.train)
rf <- ranger(fm, data=smobile.train, probability=TRUE, mtry=5, min.node.size=1)
nn3 <- nnet(fm, data=smobile.train, size=3, decay=0.1, maxit=1000)
```

```
# weights:  106
initial  value 2759.156234
iter  10 value 1166.360277
iter  20 value 1161.029865
iter  30 value 1159.087360
iter  40 value 1156.493862
```

```
iter  50 value 1147.740454
iter  60 value 1137.430912
iter  70 value 1135.407785
iter  80 value 1135.114123
iter  90 value 1134.708418
iter 100 value 1133.519288
iter 110 value 1132.265805
iter 120 value 1130.135934
iter 130 value 1122.441767
iter 140 value 1117.997826
iter 150 value 1115.559971
iter 160 value 1113.914622
iter 170 value 1112.958990
iter 180 value 1112.796157
iter 190 value 1112.770603
iter 200 value 1112.769487
final  value 1112.769283
converged
```

```r
nn4 <- nnet(fm, data=smobile.train, size=4, decay=0.1, maxit=1000)
```

```
# weights:  141
initial  value 8839.747872
iter  10 value 1342.177871
iter  20 value 1259.054113
iter  30 value 1210.475201
iter  40 value 1166.119695
iter  50 value 1162.319044
iter  60 value 1156.297554
iter  70 value 1151.910997
iter  80 value 1149.792822
iter  90 value 1146.697629
iter 100 value 1144.489047
iter 110 value 1142.751605
iter 120 value 1140.875243
iter 130 value 1138.760295
iter 140 value 1138.048113
iter 150 value 1137.219898
iter 160 value 1133.749233
iter 170 value 1130.317769
iter 180 value 1127.389109
iter 190 value 1122.664296
iter 200 value 1116.912520
iter 210 value 1113.623394
iter 220 value 1112.476182
iter 230 value 1111.321005
iter 240 value 1109.320023
iter 250 value 1108.548646
iter 260 value 1108.100872
iter 270 value 1106.549498
iter 280 value 1104.277107
iter 290 value 1102.215199
iter 300 value 1101.798035
iter 310 value 1101.620792
```

```
iter 320 value 1101.596029
iter 330 value 1101.545530
iter 340 value 1101.508309
iter 350 value 1101.430279
iter 360 value 1101.323052
iter 370 value 1101.172731
iter 380 value 1101.068408
iter 390 value 1100.540812
iter 400 value 1100.025574
iter 410 value 1099.939455
iter 420 value 1099.907978
iter 430 value 1099.890999
iter 440 value 1099.878690
iter 450 value 1099.858063
iter 460 value 1099.848618
iter 470 value 1099.845426
final  value 1099.844137
converged
```

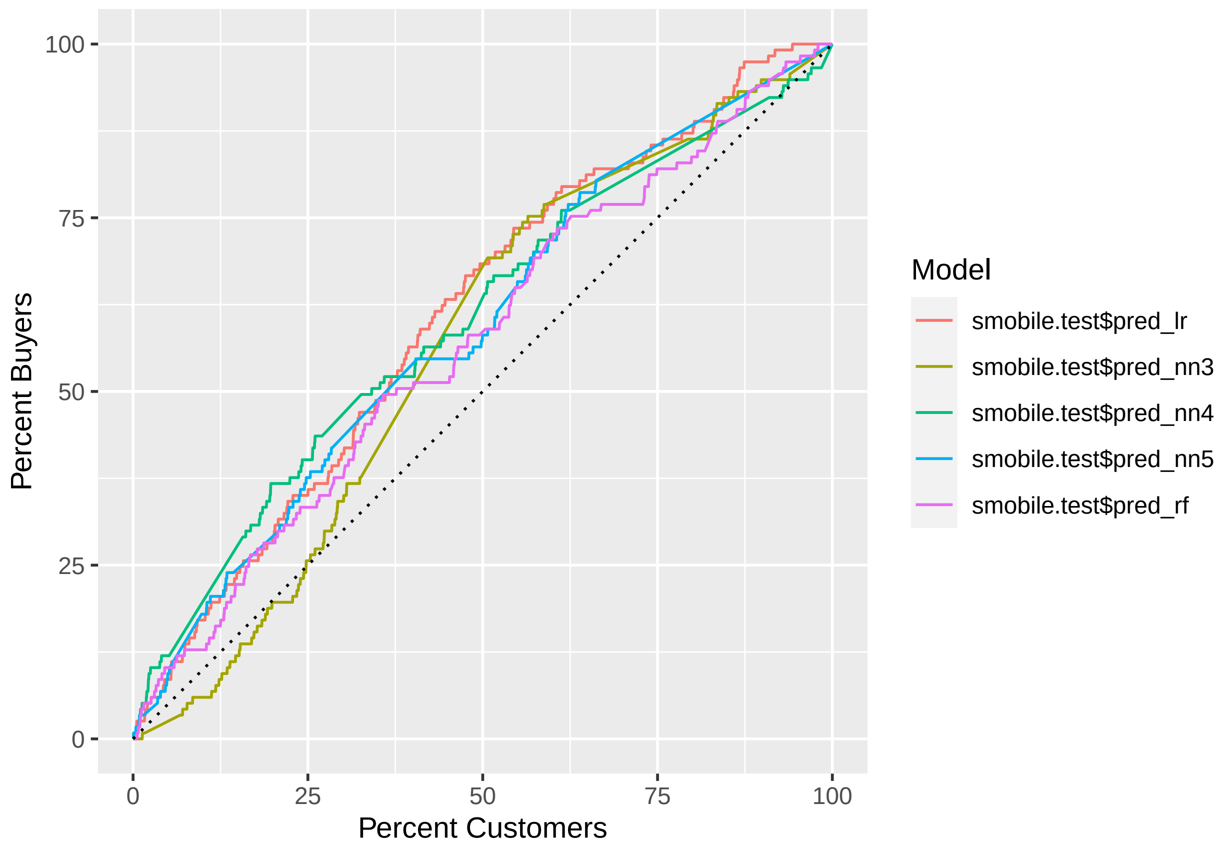
```r
nn5 <- nnet(fm, data=smobile.train, size=5, decay=0.1, maxit=1000)
```

```
# weights:  176
initial  value 3734.111401
iter  10 value 1167.596467
iter  20 value 1163.282544
iter  30 value 1151.949788
iter  40 value 1150.319160
iter  50 value 1148.245413
iter  60 value 1145.020124
iter  70 value 1140.153500
iter  80 value 1136.947045
iter  90 value 1132.617335
iter 100 value 1128.279438
iter 110 value 1122.527697
iter 120 value 1120.377665
iter 130 value 1117.072812
iter 140 value 1112.961819
iter 150 value 1107.138958
iter 160 value 1102.809991
iter 170 value 1099.078356
iter 180 value 1097.277944
iter 190 value 1095.954142
iter 200 value 1095.390337
iter 210 value 1095.062121
iter 220 value 1094.967167
iter 230 value 1094.856017
iter 240 value 1094.823298
iter 250 value 1094.816434
iter 260 value 1094.812699
iter 270 value 1094.792281
iter 280 value 1094.788689
iter 290 value 1094.788249
iter 290 value 1094.788240
iter 290 value 1094.788240
```

```
final  value 1094.788240
converged
```

```r
smobile.test <- smobile.test %>%
  mutate(pred_lr = predict(lr, newdata=smobile.test, type="response"),
         pred_rf = predict(rf, data=smobile.test, type="response")[[1]][,2],
         pred_nn3 = predict(nn3, newdata=smobile.test, type="raw"),
         pred_nn4 = predict(nn4, newdata=smobile.test, type="raw"),
         pred_nn5 = predict(nn5, newdata=smobile.test, type="raw"))

gainsplot(smobile.test$pred_lr,
          smobile.test$pred_rf,
          smobile.test$pred_nn3,
          smobile.test$pred_nn4,
          smobile.test$pred_nn5,
          label.var = smobile.test$churn)
```

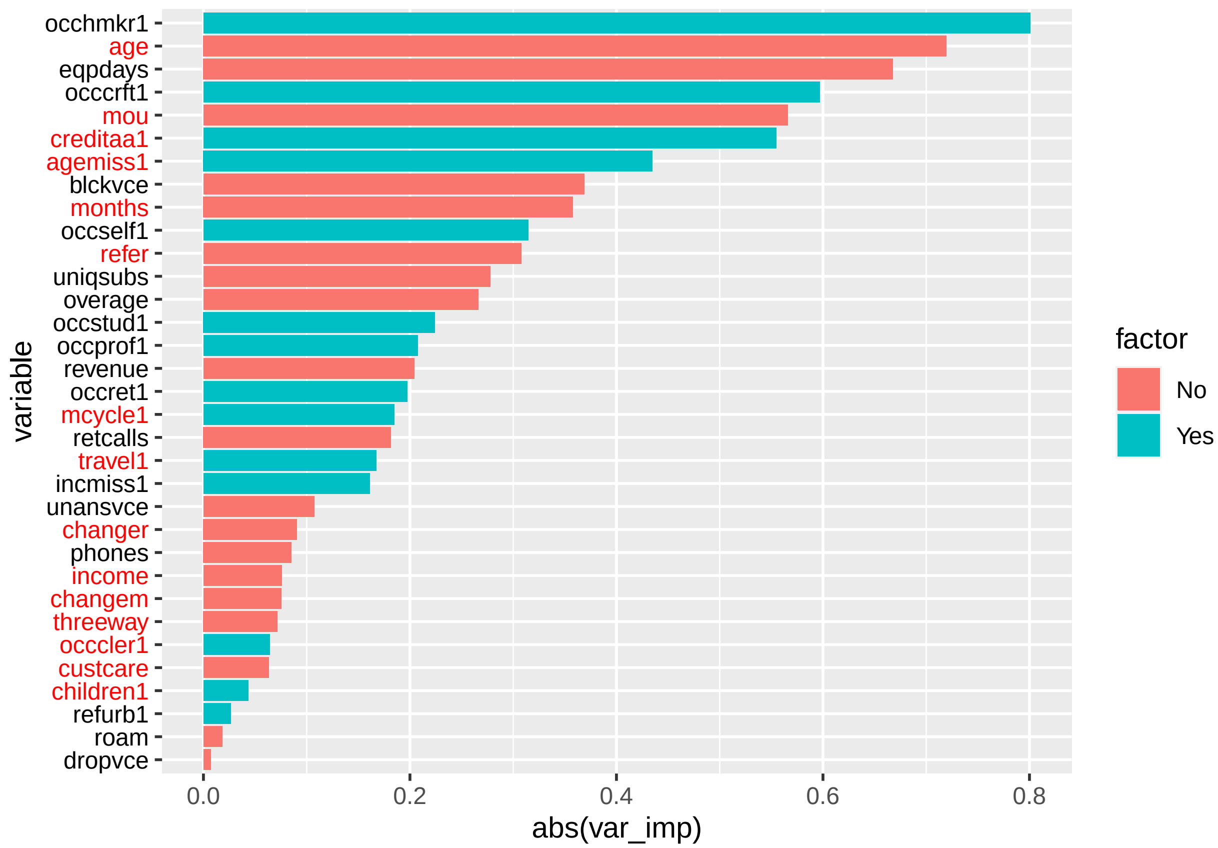



```
# A tibble: 5 x 2
  model                  auc
  <chr>                <dbl>
1 smobile.test$pred_lr  0.614
2 smobile.test$pred_rf  0.574
3 smobile.test$pred_nn3 0.568
4 smobile.test$pred_nn4 0.605
5 smobile.test$pred_nn5 0.595
```

2 - Identify Main Drivers

We can see in the variable importance chart and graph below. Since we are dealing with current customers, we are going to focus on the top non-demopraphic variables to deal with in this case. Looking at these variables that can be acted upon, we are left to consider two variables are positively correlated with churn: Number of days of current equipment and Average number of blocked voice calls. Then on the other hand, we can also consider two variables that have a negative correlation with churn - Mean monthly minutes of use and number of referrals made by subscriber.

```
varimp.logistic(lr) %>% plotimp.logistic()
```




```
# A tibble: 33 x 6
   variable  factor var_imp var_imp_lower var_imp_upper p_value
   <chr>     <chr>    <dbl>         <dbl>         <dbl>   <dbl>
 1 occhmkr1  Yes      0.801       -0.670         2.27     0.286
 2 age       No      -0.720       -1.31         -0.127    0.017
 3 eqpdays   No       0.668        0.289         1.05     0.001
 4 occcrft1  Yes      0.597       -0.0293        1.22     0.062
 5 mou       No      -0.566       -1.05         -0.0813   0.022
 6 creditaa1 Yes     -0.555       -0.999        -0.111    0.014
 7 agemiss1  Yes     -0.435       -1.16          0.287    0.238
 8 blckvce   No       0.369        0.188         0.549    0
 9 months    No      -0.358       -0.775         0.0588   0.092
10 occself1  Yes      0.315       -0.618         1.25     0.508
# ... with 23 more rows
```

3 - Develop Actions/Offers/Incentives

Number of days of current equipment (+) Average number of blocked voice calls (+) Mean monthly minutes of use (-) Number of referrals made by subscriber (-)

Based on the above criteria we have identified as important, we are planning to test out three new actions.

First, we can offer a $200 coupon for customers that have had their device for over a number of years, which we hope will help to re-invest customers that may have otherwise been close to churning, by showing them some of our newer offerings. This should reduce the number of days of current equiptment variable.

Second, we can offer a beta trial of our new telemarketer-blocking service to customers that average a high number of blocked calls. This is a service that we have been working on that will identify and auto-block robo-calls.

Third, we can offer a referral incentive of a $50 credit on your bill if you make at least one referral.

4 - Estimate Impact

We are now going to create three new datasets based off of our Rollout dataset, but with the impact of our test actions incorporated.

For the first version, our research shows that 7% of customers that receive our coupon, will in fact upgrade to a new device. For that reason, those 7% of customers who were previously seen with old equipment, will have their eqpdays reduced to 0. We will try this with 1, 2 and 3 year timelines.

For the second version, we see research supporting the fact that our service will reduce the number of robocalls dramatically, and cuts the number of blocked calls by 80%. We will try this for customers that were blocking 5, 10 and 15 calls.

For the third version, we find that one of every 20 customers on average will refer a new customer after this incentive is offered. This would likely have to be a universal rollout, but we can see the effect of offering it to only those who have not referred customers in the past, versus offering it to all customers.

```
#Create new variables to keep track of treated data
rollout <- rollout%>%
  mutate(eqpdays1y=ifelse(eqpdays>365,1,0),
         eqpdays2y=ifelse(eqpdays>730,1,0),
         eqpdays3y=ifelse(eqpdays>1095,1,0),
         blckvce5=ifelse(blckvce>=5,1,0),
         blckvce10=ifelse(blckvce>=10,1,0),
         blckvce15=ifelse(blckvce>=15,1,0),
         refernew=ifelse(refer==0,1,0))


rm(rollout_coupon1,rollout_coupon2,rollout_coupon3,rollout_block5,rollout_block10,
   rollout_block15,rollout_refernew,rollout_referall)

#Create copies of the rollout data, that can be altered based on actions described above
rollout_coupon1 <- rollout
rollout_coupon2 <- rollout
rollout_coupon3 <- rollout
rollout_block5 <- rollout
rollout_block10 <- rollout
rollout_block15 <- rollout
rollout_refernew <- rollout
rollout_referall <- rollout
```

```
#Create vectors of customers that will be affected by first action
coupon.customers1 <- rollout %>%
  filter(eqpdays>365)%>%
  sample_frac(.07)%>%
  select(customer)
coupon.customers2 <- rollout %>%
  filter(eqpdays>730)%>%
  sample_frac(.07)%>%
  select(customer)
coupon.customers3 <- rollout %>%
  filter(eqpdays>1095)%>%
  sample_frac(.07)%>%
  select(customer)

#Update equipment days in copies of data to reflect first action
rollout_coupon1 <- rollout_coupon1 %>%
  mutate(eqpdays = ifelse(customer %in% coupon.customers1$customer,0,eqpdays))
rollout_coupon2 <- rollout_coupon2 %>%
  mutate(eqpdays = ifelse(customer %in% coupon.customers2$customer,0,eqpdays))
rollout_coupon3 <- rollout_coupon3 %>%
  mutate(eqpdays = ifelse(customer %in% coupon.customers3$customer,0,eqpdays))


#Create vectors of customers that will be affected by second action
block.customers5 <- rollout %>%
  filter(blckvce>=5)%>%
  select(customer)
block.customers10 <- rollout %>%
  filter(blckvce>=10)%>%
  select(customer)
block.customers15 <- rollout %>%
  filter(blckvce>=15)%>%
  select(customer)

#Update blocked calls in copies of data to reflect second action
rollout_block5 <- rollout_block5 %>%
  mutate(blckvce = ifelse(customer %in% block.customers5$customer,blckvce*.2,blckvce))
rollout_block10 <- rollout_block10 %>%
  mutate(blckvce = ifelse(customer %in% block.customers10$customer,blckvce*.2,blckvce))
rollout_block15 <- rollout_block15 %>%
  mutate(blckvce = ifelse(customer %in% block.customers15$customer,blckvce*.2,blckvce))

#Create vectors of customers that will be affected by third action
refer.customersnew <- rollout %>%
  filter(refer==0)%>%
  sample_frac(.05)%>%
  select(customer)
refer.customersall <- rollout %>%
  sample_frac(.05)%>%
  select(customer)

#Update referrals in copies of data to reflect third action
rollout_refernew <- rollout_refernew %>%
```

```r
  mutate(refer = ifelse(customer %in% refer.customersnew$customer,refer+1,refer))
rollout_referall <- rollout_referall %>%
  mutate(refer = ifelse(customer %in% refer.customersall$customer,refer+1,refer))
```

5 - Decide on Targets

Using the methodology above, we can then look at the effects that all of the treatments have on Churn. Below, we look at this both within the groups that are directly being treated, as well as in the Rollout group overall. Based on the overall churn rate of the rollout group, we see that the most promising action is the telemarketer blocking offer to all customers that average atleast 5 blocked calls. This reduces churn by around 20% in that group of customers and by almost 0.2% in the rollout group overall.

```r
#Baseline Churn prediction
churn_baseline <- predict(lr, newdata=rollout, type="response")%>% mean()
#Churn prediction for customers with device for 1 year, before and after treatment
churn_1y <- predict(lr, newdata=rollout %>%
                      filter(eqpdays1y==1), type="response")%>% mean()
churn_1y_treat <- predict(lr, newdata=rollout_coupon1 %>%
                            filter(eqpdays1y==1), type="response")%>% mean()
(churn_1y-churn_1y_treat)/churn_1y
```

```
[1] 0.0379
```

```r
#Churn prediction for customers with device for 2 years, before and after treatment
churn_2y <- predict(lr, newdata=rollout %>%
                      filter(eqpdays2y==1), type="response")%>% mean()
churn_2y_treat <- predict(lr, newdata=rollout_coupon2 %>%
                            filter(eqpdays2y==1), type="response")%>% mean()
(churn_2y-churn_2y_treat)/churn_2y
```

```
[1] 0.0523
```

```r
#Churn prediction for customers with device for 3 years, before and after treatment
churn_3y <- predict(lr, newdata=rollout %>%
                      filter(eqpdays3y==1), type="response")%>% mean()
churn_3y_treat <- predict(lr, newdata=rollout_coupon3 %>%
                            filter(eqpdays3y==1), type="response")%>% mean()
(churn_3y-churn_3y_treat)/churn_3y
```

```
[1] 0.0539
```

```r
#Churn prediction for customers with an average of at least 5 blocked calls, before and after treatment
churn_block5 <- predict(lr, newdata=rollout %>%
                          filter(blckvce5==1), type="response")%>% mean()
churn_block5_treat <- predict(lr, newdata=rollout_block5 %>%
                                filter(blckvce5==1), type="response")%>% mean()
(churn_block5-churn_block5_treat)/churn_block5
```

```
[1] 0.201
```

```r
#Churn prediction for customers with an average of at least 10 blocked calls, before and after treatmen
churn_block10 <- predict(lr, newdata=rollout %>%
                           filter(blckvce10==1), type="response")%>% mean()
churn_block10_treat <- predict(lr, newdata=rollout_block10 %>%
                                 filter(blckvce10==1), type="response")%>% mean()
(churn_block10-churn_block10_treat)/churn_block10
```

```
[1] 0.293
```

```r
#Churn prediction for customers with an average of at least 15 blocked calls, before and after treatmen
churn_block15 <- predict(lr, newdata=rollout %>%
                           filter(blckvce15==1), type="response")%>% mean()
churn_block15_treat <- predict(lr, newdata=rollout_block15 %>%
                                 filter(blckvce15==1), type="response")%>% mean()
(churn_block15-churn_block15_treat)/churn_block15
```

```
[1] 0.364
```

```r
#Churn prediction for customers with no previous referrals, before and after treatment
churn_new <- predict(lr, newdata=rollout %>%
                       filter(refernew==1), type="response")%>% mean()
churn_new_treat <- predict(lr, newdata=rollout_refernew %>%
                             filter(refernew==1), type="response")%>% mean()
(churn_new-churn_new_treat)/churn_new
```

```
[1] 0.0213
```

```r
#Churn prediction for all customers, before and after treatment
churn_all <- predict(lr, newdata=rollout, type="response")%>% mean()
churn_all_treat <- predict(lr, newdata=rollout_referall, type="response")%>% mean()
(churn_all-churn_all_treat)/churn_all
```

```
[1] 0.0212
```

```r
#Effects of the above treatment on the overall Rollout Group
predict(lr, newdata=rollout, type="response")%>% mean()
```

```
[1] 0.0393
```

```r
predict(lr, newdata=rollout_coupon1, type="response")%>% mean()
```

```
[1] 0.0386
```

```r
predict(lr, newdata=rollout_coupon2, type="response")%>% mean()
```

```
[1] 0.0391
```

```r
predict(lr, newdata=rollout_coupon3, type="response")%>% mean()
```

```
[1] 0.0393
```

```r
predict(lr, newdata=rollout_block5, type="response")%>% mean()
```

```
[1] 0.0375
```

```r
predict(lr, newdata=rollout_block10, type="response")%>% mean()
```

```
[1] 0.0378
```

```r
predict(lr, newdata=rollout_block15, type="response")%>% mean()
```

```
[1] 0.0381
```

```r
predict(lr, newdata=rollout_refernew, type="response")%>% mean()
```

```
[1] 0.0385
```

```r
predict(lr, newdata=rollout_referall, type="response")%>% mean()
```

```
[1] 0.0385
```

6 - Evaluate Economics

Below we can see the lifetime value calculations that detail the economic benfit per customer, for each new cost and churn rate. We will make the following assumptions: - begin with the average annual revenue from the rollout customers, - annual revenue groth of 5%, - baseline cost of 50% of revenue, - discount rate of 10%, - since we did mutliple iterations of each action, we will caculate LTV on the version with the greatest effect: - Coupon for 1 year old devices: Churn of .0386, cost of $200 per redeemed coupon, - call blocking for greater than 5 average blocked calls: Churn of .0375, no explicit cost, - referrals for all customers: Churn of .0385, cost of $50 per referral

We can see below that the Call Blocking and Referral Incentives both create customer LTV, by $6.08 and $0.32 respectively, while the Coupon offer reduces LTV by $3.16.

```r
paste("Starting Revenue:",dollar(mean(rollout$revenue)*12))
```

```
[1] "Starting Revenue: $706.23"
```

```r
paste("Cost of Coupon per Customer:",dollar((nrow(coupon.customers1)*200)/8012))
```

```
[1] "Cost of Coupon per Customer: $5.79"
```

```r
paste("Cost of Referral Incentive:",dollar((nrow(refer.customersall)*50)/8012))
```

```
[1] "Cost of Referral Incentive: $2.50"
```

```r
include_graphics("LTV.pdf")
```

	Baseline: Churn of 3.93%					
	0	1	2	3	4	5
Revenue	$ -	$ 706.23	$ 741.54	$ 778.62	$ 817.55	$ 858.43
Cost	$ -	$ 353.12	$ 370.77	$ 389.31	$ 408.77	$ 429.21
Offer Cost	$ -	$ -	$ -	$ -	$ -	$ -
Profit	$ -	$ 353.12	$ 370.77	$ 389.31	$ 408.77	$ 429.21
Prob. Active EOP	100%	96%	92%	89%	85%	82%
Prob. Active WP	100%	98%	94%	90%	87%	84%
Profit	$ -	$ 346.18	$ 349.20	$ 352.25	$ 355.33	$ 358.43
Present Value	$ -	$ 330.07	$ 302.68	$ 277.57	$ 254.54	$ 233.42
						$ 1,398.27

	Coupon Offer: Churn of 3.86%					
	0	1	2	3	4	5
Revenue	$ -	$ 706.23	$ 741.54	$ 778.62	$ 817.55	$ 858.43
Cost	$ -	$ 353.12	$ 370.77	$ 389.31	$ 408.77	$ 429.21
Offer Cost	$ 5.79	$ -	$ -	$ -	$ -	$ -
Profit	$ (5.79)	$ 353.12	$ 370.77	$ 389.31	$ 408.77	$ 429.21
Prob. Active EOP	100%	96%	92%	89%	85%	82%
Prob. Active WP	100%	98%	94%	91%	87%	84%
Profit	$ (5.79)	$ 346.30	$ 349.58	$ 352.89	$ 356.23	$ 359.61
Present Value	$ (5.52)	$ 330.18	$ 303.01	$ 278.07	$ 255.19	$ 234.18
						$ 1,395.12
						$ (3.16)

	Call Blocking: Churn of 3.75%					
	0	1	2	3	4	5
Revenue	$ -	$ 706.23	$ 741.54	$ 778.62	$ 817.55	$ 858.43
Cost	$ -	$ 353.12	$ 370.77	$ 389.31	$ 408.77	$ 429.21
Offer Cost	$ -	$ -	$ -	$ -	$ -	$ -
Profit	$ -	$ 353.12	$ 370.77	$ 389.31	$ 408.77	$ 429.21
Prob. Active EOP	100%	96%	93%	89%	86%	83%
Prob. Active WP	100%	98%	94%	91%	87%	84%
Profit	$ -	$ 346.49	$ 350.18	$ 353.90	$ 357.66	$ 361.46
Present Value	$ -	$ 330.37	$ 303.53	$ 278.87	$ 256.21	$ 235.39
						$ 1,404.36
						$ 6.08

	Referral Incentive: Churn of 3.85%					
	0	1	2	3	4	5
Revenue	$ -	$ 706.23	$ 741.54	$ 778.62	$ 817.55	$ 858.43
Cost	$ -	$ 353.12	$ 370.77	$ 389.31	$ 408.77	$ 429.21
Offer Cost	$ 2.50	$ -	$ -	$ -	$ -	$ -
Profit	$ (2.50)	$ 353.12	$ 370.77	$ 389.31	$ 408.77	$ 429.21
Prob. Active EOP	100%	96%	92%	89%	85%	82%
Prob. Active WP	100%	98%	94%	91%	87%	84%
Profit	$ (2.50)	$ 346.32	$ 349.63	$ 352.98	$ 356.36	$ 359.77
Present Value	$ (2.38)	$ 330.20	$ 303.06	$ 278.14	$ 255.28	$ 234.29
						$ 1,398.59
						$ 0.32