Mixed Media Modeling & Customer Acquisition

Jethro Perez

10/31/2017

### I. Introduction

In the world of e-commerce [CPG](http://www.investopedia.com/terms/c/cpg.asp) companies must balance year over year growth in revenue while maintaining a positive ratio of lifetime value over customer acquisition cost. The ratio is known as **LTV/CAC** and is one of the most important key performance indicators (KPI) for any company whose business model lies on customers consuming a companies goods on either a one-time or recurring basis.

Maximizing this KPI can be divided into two problems, one centered on classifying customers based on average basket size and targeting customers who purchase the most (i.e. maximizing LTV). The other which is the focus of my project is by minimizing the average cost per acquisition (CPA).

### II. The Problem

In the realm of marketing a [mixed media model](https://en.wikipedia.org/wiki/Marketing_mix_modeling) (MMM) is used by savvy data intensive organizations to minimize CPA's. A MMM assess the impact of each of your marketing partners in driving new customer acquisition. By building a MMM marketers can see which channels drive the highest volume of new customers and conversely which don’t. With this information executives can alter their budgets which normally rely on third party data from advertisers who have different reporting standards and measuring techniques.

Building a model that answers these questions can range from simple to complex depending on the state of your business as well as your marketing plan. For example, building a model for a local restaurant who advertises in newspapers and local ads is rather simple compared to a Fortune 500 company that sells a product in multiple countries and spends billions of dollars on advertising budgets across multiple different partners.

In the case of my company we operate across the United States and spend millions of dollars acquiring customers for multiple different products. We have a baseline set of advertisers which we spend most our yearly budget on while reserving some for testing new partners who may unlock new markets that have yet to be opened.

The marketing team has asked me to create a model that can accurately predict customer acquisition in a given period in time with a predefined budget. If I am able to correctly customer counts with a high degree of accuracy the team would like to know which partners are the most effective in acquiring new customers at the lowest cost in order to drive down CPA's.

### III. Data Wrangling

The data that was gathered for this model came from our third-party advertisers as well as our internal data model which I build and maintain. Acquiring the former was an exercise in generating reports from the respective advertisers reporting platforms or via email. To have substantial sample size *n* given the number of potential predictors *p* I decided on aggregating data on a daily level going back from 2016 through 2017 for four different products. In total I gathered 578 observations and 31 different variables.

The data here has been anonymized and scaled for privacy purposes. Figures stated in my model will be different from this project but the overall process (modeling, results and deck) will be the same when recreating it for company members. I decided to also limit this project model only **Product 1** which is our largest and most important. The findings of this can easily be reproduced to the other three products.

When gathering advertiser data I intentionally omitted customer counts which each partner collects. The reason for this is due to a phenomenon known as [over attribution](https://en.wikipedia.org/wiki/Attribution_(marketing)). This occurs because advertisers have different methodologies when measuring the impact of their platform. For example, a social media company like Twitter can track users who click on advertisements to a subsequent website purchase. That mechanism does not work for a company that sells radio advertisements who may rely on the use of specific promotion codes to measure customer acquisition.

To make matters more complex customers use multiple devices (personal computer, work computer, mobile phone, etc..) and don’t always click on ads but use search engines to engage with the brand. Having multiple touch points results in advertisers stealing attribution from each other due to customers naturally having multiple different touch points prior to purchasing Over attribution results in overstated customer counts compared to actual figures, therefore I needed to gather that data using our internal data model.

Once I had data from each partner I anonymized and joined all the data into a CSV document. The format was intended to have each day as an observation and each column as a different advertisers spend on a product that particular day and the number of actual customers acquired on that date.

I loaded the necessary libraries and read in my document into a dataframe called **data**, I then made some small changes to the column date.

library(tidyverse)  
library(PerformanceAnalytics)  
  
data <- read.csv("capstone\_tidy\_data.csv",stringsAsFactors = F)  
data[1:16] <- sapply(data[names(data[1:16])],parse\_number)

### IV. Exploratory Data Analysis

I began exploring my dataset using some of concepts I adopted after learning to use *dplyr* and *tidyr*. This is where I ran into one of the biggest problems in my entire model. To use data in tidy format each value must have a key to pair to it. In the case of advertisers and the amount spent that requirement is met, however we don’t know how many customers were acquired from that specific advertiser on a given day we only know that in aggregate. To get a 1 for 1 matching for customers by advertiser we need to solve for over attribution. This is done by either building or adopting a multi touch attribution model which is outside the scope of this project.

The inability to have data in this format limited my ability to facet advertisers as a function of amount spent and customers in a scatterplot. But as we all know real world problems will not have complete data that is clean and ready to model.

I overcame this hurdle and began to explore this data through the lens of a time series. My logic here was a belief that CPA is a function of both the mix in media spend as well as some seasonal components. An easy example of this sales for an electronic retail company which sees huge fluctuations in customer sales as the holiday season approaches. My findings suggested that there was no such evidence in any seasonal components but this can be a false assumption, only having 1 ½ years’ worth of data is hard to confirm this theory.

I found no such pattern and using time series modeling in conjunction with regression did not seem to be the path that I wanted to take. Therefore I removed date from the original data set and continued on the path of exploring the data.I adjusted my original CSV to only include product 1 and name the date frame **prod 1**.

prod1 <- select(data,A.prod1,C.prod1,D.prod1,E.prod1,F.prod1,G.prod1,H.prod1,I.prod1,J.prod1,K.prod1,L.all.1,L.prod1,prod1.customers)

I followed this by looking at some basic scatter plots which I did log transformations to make spotting trends easier. There are no categorical variables in this data set so using boxplots or histograms a bit moot. I followed by computing some basic summary statistics including the mean, variance and max/min to see how my data was distributed. Looking at this I removed partners that didn't have an adequete amount of spend during the period in which data was collected. This new data frame is called **prod1\_clean**.

summary(prod1)

## A.prod1 C.prod1 D.prod1 E.prod1   
## Min. : 8400 Min. : 25.83 Min. : 1547 Min. :0   
## 1st Qu.: 45327 1st Qu.: 920.53 1st Qu.:15104 1st Qu.:0   
## Median : 66949 Median : 1270.92 Median :18876 Median :0   
## Mean : 72012 Mean : 1393.95 Mean :19255 Mean :0   
## 3rd Qu.: 95491 3rd Qu.: 1709.35 3rd Qu.:22540 3rd Qu.:0   
## Max. :228909 Max. :15610.84 Max. :52556 Max. :0   
## F.prod1 G.prod1 H.prod1 I.prod1 J.prod1   
## Min. : 0 Min. : 0 Min. :0 Min. : 0 Min. :0   
## 1st Qu.: 0 1st Qu.: 3213 1st Qu.:0 1st Qu.: 14771 1st Qu.:0   
## Median : 0 Median : 5252 Median :0 Median : 40647 Median :0   
## Mean : 1038 Mean : 5365 Mean :0 Mean : 40809 Mean :0   
## 3rd Qu.: 1463 3rd Qu.: 7014 3rd Qu.:0 3rd Qu.: 59400 3rd Qu.:0   
## Max. :10282 Max. :19400 Max. :0 Max. :158525 Max. :0   
## K.prod1 L.all.1 L.prod1 prod1.customers   
## Min. : 0 Min. : 0.00 Min. : 0 Min. : 46.0   
## 1st Qu.: 3133 1st Qu.: 0.00 1st Qu.: 0 1st Qu.: 231.2   
## Median : 5629 Median : 0.00 Median : 0 Median : 299.0   
## Mean : 5403 Mean : 23.32 Mean : 793 Mean : 404.9   
## 3rd Qu.: 7429 3rd Qu.: 39.81 3rd Qu.:1961 3rd Qu.: 588.0   
## Max. :13078 Max. :219.03 Max. :5107 Max. :1197.0

ad\_partner\_sums<- colSums(prod1,na.rm=TRUE)  
ad\_partner\_sums

## A.prod1 C.prod1 D.prod1 E.prod1   
## 41623087.94 805703.50 11129545.00 0.00   
## F.prod1 G.prod1 H.prod1 I.prod1   
## 600174.89 3100776.63 0.00 23587759.73   
## J.prod1 K.prod1 L.all.1 L.prod1   
## 0.00 3123219.75 13478.99 458341.87   
## prod1.customers   
## 234012.00

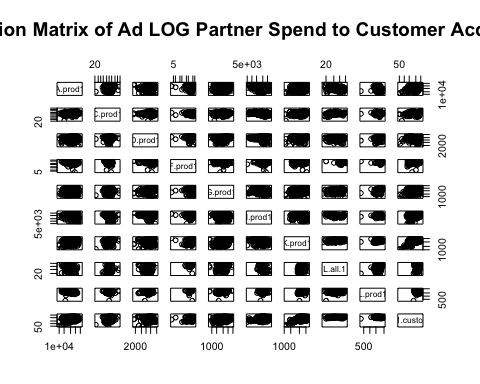
ad\_partner\_means<- colMeans(prod1,na.rm=TRUE)  
ad\_partner\_means

## A.prod1 C.prod1 D.prod1 E.prod1   
## 72012.26287 1393.95069 19255.26817 0.00000   
## F.prod1 G.prod1 H.prod1 I.prod1   
## 1038.36486 5364.66545 0.00000 40809.27289   
## J.prod1 K.prod1 L.all.1 L.prod1   
## 0.00000 5403.49438 23.32005 792.97901   
## prod1.customers   
## 404.86505

prod1\_clean <- select(prod1, -E.prod1, -J.prod1, -H.prod1)  
summary(prod1\_clean)

## A.prod1 C.prod1 D.prod1 F.prod1   
## Min. : 8400 Min. : 25.83 Min. : 1547 Min. : 0   
## 1st Qu.: 45327 1st Qu.: 920.53 1st Qu.:15104 1st Qu.: 0   
## Median : 66949 Median : 1270.92 Median :18876 Median : 0   
## Mean : 72012 Mean : 1393.95 Mean :19255 Mean : 1038   
## 3rd Qu.: 95491 3rd Qu.: 1709.35 3rd Qu.:22540 3rd Qu.: 1463   
## Max. :228909 Max. :15610.84 Max. :52556 Max. :10282   
## G.prod1 I.prod1 K.prod1 L.all.1   
## Min. : 0 Min. : 0 Min. : 0 Min. : 0.00   
## 1st Qu.: 3213 1st Qu.: 14771 1st Qu.: 3133 1st Qu.: 0.00   
## Median : 5252 Median : 40647 Median : 5629 Median : 0.00   
## Mean : 5365 Mean : 40809 Mean : 5403 Mean : 23.32   
## 3rd Qu.: 7014 3rd Qu.: 59400 3rd Qu.: 7429 3rd Qu.: 39.81   
## Max. :19400 Max. :158525 Max. :13078 Max. :219.03   
## L.prod1 prod1.customers   
## Min. : 0 Min. : 46.0   
## 1st Qu.: 0 1st Qu.: 231.2   
## Median : 0 Median : 299.0   
## Mean : 793 Mean : 404.9   
## 3rd Qu.:1961 3rd Qu.: 588.0   
## Max. :5107 Max. :1197.0

Lastly I created a correlation matrix using the pairs and corr\_matrix function to get a better idea of how each variable varied with each other. The linear relationships I saw on each of the predictors variables with the response ignited the idea for the initial algorithm that I would use to apply machine learning to solve the question at hand.



corr\_matrix <- cor(prod1\_clean, method = "pearson", use = "complete.obs")

### IV. Model Selection

Given the structure of my data set I believe multiple linear regression is the best approach and starting point. I began by building a multiple linear regression with my baseline model being one that simply uses all the predictors to measure the response.

I randomly split my data into a training and test set using a 70-30 mix. When using the baseline model I simply regressed all predictor variables on the response to see what the simplest model will yield. The summary of the model yielded a paltry adjusted r-squared value of **0.533** and only 7 out of 9 predictors being statistically significant. The model had an intercept of 204 meaning that if we spent $0 on advertising we would on average acquire 204 customers. The poor r-squared value lead me to the path of refining my model.

smp\_size<- floor(0.70 \* nrow(prod1\_clean))  
set.seed(2017)   
train\_ind<- sample(seq\_len(nrow(prod1\_clean)),size = smp\_size)  
  
train <- prod1\_clean[train\_ind,]  
test <- prod1\_clean[-train\_ind,]  
  
mod1<- lm(prod1.customers~.,data=train)  
summary(mod1)

##   
## Call:  
## lm(formula = prod1.customers ~ ., data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -291.84 -103.37 -34.71 64.89 517.90   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.040e+02 3.155e+01 6.466 2.98e-10 \*\*\*  
## A.prod1 7.888e-04 2.337e-04 3.375 0.000813 \*\*\*  
## C.prod1 1.090e-02 8.216e-03 1.327 0.185301   
## D.prod1 -3.974e-03 1.176e-03 -3.380 0.000797 \*\*\*  
## F.prod1 -1.059e-02 4.638e-03 -2.283 0.022953 \*   
## G.prod1 -1.921e-03 2.935e-03 -0.655 0.513164   
## I.prod1 1.909e-03 2.766e-04 6.900 2.09e-11 \*\*\*  
## K.prod1 1.443e-02 2.814e-03 5.129 4.58e-07 \*\*\*  
## L.all.1 7.278e-01 3.435e-01 2.119 0.034735 \*   
## L.prod1 7.327e-02 1.232e-02 5.947 6.03e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 150.3 on 394 degrees of freedom  
## Multiple R-squared: 0.5431, Adjusted R-squared: 0.5327   
## F-statistic: 52.04 on 9 and 394 DF, p-value: < 2.2e-16

I started removing predictors with large p-values out of the model one by one and noticed that my r-squared value wasn’t moving. This indicated that I need to take a closer look at my data to look for any patterns. I noticed a bit of multicollinearity between a few predictors which violates one of the requirements necessary for OSL to work. Limiting this required adopting another method for variable selection as well as interaction terms.

mod2<- lm(prod1.customers~ A.prod1 + D.prod1 + F.prod1 + I.prod1 + K.prod1 + L.all.1 + L.prod1,data=train)  
summary(mod2)

##   
## Call:  
## lm(formula = prod1.customers ~ A.prod1 + D.prod1 + F.prod1 +   
## I.prod1 + K.prod1 + L.all.1 + L.prod1, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -293.60 -105.10 -33.56 64.99 520.67   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.096e+02 3.009e+01 6.966 1.37e-11 \*\*\*  
## A.prod1 8.310e-04 2.315e-04 3.589 0.000374 \*\*\*  
## D.prod1 -4.096e-03 1.110e-03 -3.688 0.000258 \*\*\*  
## F.prod1 -1.154e-02 4.596e-03 -2.510 0.012463 \*   
## I.prod1 1.915e-03 2.711e-04 7.065 7.27e-12 \*\*\*  
## K.prod1 1.430e-02 2.687e-03 5.321 1.73e-07 \*\*\*  
## L.all.1 7.658e-01 3.427e-01 2.234 0.026012 \*   
## L.prod1 7.219e-02 1.228e-02 5.877 8.85e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 150.4 on 396 degrees of freedom  
## Multiple R-squared: 0.5404, Adjusted R-squared: 0.5323   
## F-statistic: 66.53 on 7 and 396 DF, p-value: < 2.2e-16

mod3<- lm(prod1.customers~ A.prod1 + D.prod1 + F.prod1 + I.prod1 + K.prod1 + L.prod1,data=train)  
summary(mod3)

##   
## Call:  
## lm(formula = prod1.customers ~ A.prod1 + D.prod1 + F.prod1 +   
## I.prod1 + K.prod1 + L.prod1, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -294.35 -104.72 -36.16 63.13 519.12   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.144e+02 3.017e+01 7.108 5.49e-12 \*\*\*  
## A.prod1 7.800e-04 2.316e-04 3.368 0.000830 \*\*\*  
## D.prod1 -4.144e-03 1.116e-03 -3.714 0.000233 \*\*\*  
## F.prod1 -9.370e-03 4.515e-03 -2.075 0.038614 \*   
## I.prod1 1.829e-03 2.697e-04 6.781 4.34e-11 \*\*\*  
## K.prod1 1.452e-02 2.698e-03 5.381 1.27e-07 \*\*\*  
## L.prod1 9.491e-02 6.927e-03 13.701 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 151.1 on 397 degrees of freedom  
## Multiple R-squared: 0.5347, Adjusted R-squared: 0.5276   
## F-statistic: 76.02 on 6 and 397 DF, p-value: < 2.2e-16

I introduced a few interaction terms to see if this would improve the performance of my model and to my delight they did! Adding 3 interaction terms improved my adjusted r-squared up to 0.6403 or a 20% improvement over the baseline. This makes sense intuitively because advertisers and the users who engage in them don’t live in a vacuum. Most people who convert due to advertising are exposed to it a few times to keep it on their mind and may be influenced into buying something because they see ads everywhere they look.

mod4<- lm(prod1.customers~ A.prod1 + D.prod1 + F.prod1 + I.prod1 + K.prod1 + L.all.1 + L.prod1 + L.all.1\*A.prod1,data=train)  
summary(mod4)

##   
## Call:  
## lm(formula = prod1.customers ~ A.prod1 + D.prod1 + F.prod1 +   
## I.prod1 + K.prod1 + L.all.1 + L.prod1 + L.all.1 \* A.prod1,   
## data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -262.68 -96.06 -32.55 53.43 524.28   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.689e+02 2.815e+01 9.554 < 2e-16 \*\*\*  
## A.prod1 1.900e-04 2.221e-04 0.855 0.393   
## D.prod1 -4.464e-03 1.011e-03 -4.413 1.32e-05 \*\*\*  
## F.prod1 -3.900e-03 4.266e-03 -0.914 0.361   
## I.prod1 1.634e-03 2.487e-04 6.571 1.58e-10 \*\*\*  
## K.prod1 1.329e-02 2.448e-03 5.429 9.90e-08 \*\*\*  
## L.all.1 -2.180e+00 4.490e-01 -4.854 1.74e-06 \*\*\*  
## L.prod1 6.600e-02 1.120e-02 5.893 8.11e-09 \*\*\*  
## A.prod1:L.all.1 6.495e-05 7.123e-06 9.118 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 136.8 on 395 degrees of freedom  
## Multiple R-squared: 0.6204, Adjusted R-squared: 0.6127   
## F-statistic: 80.68 on 8 and 395 DF, p-value: < 2.2e-16

mod5<- lm(prod1.customers ~ A.prod1 + L.all.1\*A.prod1 + D.prod1 + I.prod1 + K.prod1 + L.all.1 + L.prod1 + L.prod1\*L.all.1 + I.prod1\*A.prod1, data=train)  
summary(mod5)

##   
## Call:  
## lm(formula = prod1.customers ~ A.prod1 + L.all.1 \* A.prod1 +   
## D.prod1 + I.prod1 + K.prod1 + L.all.1 + L.prod1 + L.prod1 \*   
## L.all.1 + I.prod1 \* A.prod1, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -292.05 -84.35 -33.15 42.12 567.18   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.921e+02 3.156e+01 6.087 2.73e-09 \*\*\*  
## A.prod1 9.470e-04 2.827e-04 3.350 0.000887 \*\*\*  
## L.all.1 2.713e-01 7.798e-01 0.348 0.728057   
## D.prod1 -3.206e-03 9.969e-04 -3.216 0.001409 \*\*   
## I.prod1 3.010e-03 4.734e-04 6.358 5.67e-10 \*\*\*  
## K.prod1 1.147e-02 2.341e-03 4.898 1.42e-06 \*\*\*  
## L.prod1 7.833e-02 1.075e-02 7.289 1.72e-12 \*\*\*  
## A.prod1:L.all.1 5.948e-05 7.361e-06 8.081 7.98e-15 \*\*\*  
## L.all.1:L.prod1 -8.250e-04 1.940e-04 -4.253 2.63e-05 \*\*\*  
## A.prod1:I.prod1 -2.188e-08 5.946e-09 -3.680 0.000266 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 131.9 on 394 degrees of freedom  
## Multiple R-squared: 0.6483, Adjusted R-squared: 0.6403   
## F-statistic: 80.7 on 9 and 394 DF, p-value: < 2.2e-16

Given the large boost I needed to devise a way to search every possible model that can be built using each distinct combination of predictors and interaction terms. I used stepwise selection using the regsubsets command. I created the best 3 models for each combination starting at 1 predictor 28 predictors. I then looked for the most optimal model using *BIC*, *Mallows Cp* and *Adjusted R Squared* to find the optimal number of predictors to use.

library(leaps)  
library(lmtest)  
library(car)  
  
reg\_subset <- regsubsets(prod1.customers ~.^2, data =train, nbest = 3, method = "exhaustive")  
  
subsets(reg\_subset, statistic = "bic")  
subsets(reg\_subset, statistic = "aic", ylim = c(-425, -410), xlim = c(5,10))  
  
subsets(reg\_subset, statistic = "cp")  
subsets(reg\_subset, statistic = "cp", ylim=c(0,50), xlim = c(5,10))  
  
subsets(reg\_subset, statistic = "adjr")  
subsets(reg\_subset, statistic = "adjr", ylim = c(.55,.8), xlim = c(5,10))  
  
reg\_subset\_best <- regsubsets(prod1.customers ~.^2, data =train, nbest = 1, method = "exhaustive")  
  
subsets(reg\_subset\_best, statistic = "bic", ylim = c(-425, -415), xlim = c(0,25))  
subsets(reg\_subset\_best, statistic = "cp", ylim=c(0,50), xlim = c(5,10))  
subsets(reg\_subset\_best, statistic = "adjr2", ylim = c(.60,.8), xlim = c(0,10))  
  
summary(reg\_subset\_best)  
plot(reg\_subset\_best, scale = "adjr2", main = "Adjusted R^2")  
coef(reg\_subset\_best,8)  
  
mod6<- lm(prod1.customers ~ A.prod1 + L.prod1 + A.prod1:D.prod1 + A.prod1:I.prod1 + A.prod1:L.all.1 + I.prod1:K.prod1 + I.prod1:L.prod1 + L.all.1:L.prod1, data = train)  
summary(mod6)

All three statistics showed that 8 is the optimal number of predictors which yielded the following model.

This says that the optimal mix in media occurs when you spend on advertiser A and L independently and spend on pairs of A and D together, A and I together, etc.. If you were to spend $0 you would acquire 186 customers, in real world setting this value would hold for a period of time as the effects of advertising diminish. This latency period can last periods as short as 30 days and as far as 90 days As stated in the Introduction to Statistical Learning book there are a few advertisers that have synergy or work with each other. The result of this work boosted the adjusted r-squared to 0.684 or a 28% improvement from the baseline model.

The presence of multicollinearity made me believe that the last model may still be flawed so I decided to use a ridge regression which performs well with data like this. Ridge regression uses a penalty factor *λ* to adjust each of predictors via a shrinking the beta values based on this penalty. I ran a sequence of over 2,000 models at various lambda values. I found the optimal value and created model. This model unfortunately performed poorly with an adjusted r-squared of 0.537. This makes sense with a basic ridge regression because it simply shrinks the beta values to a very small value but doesn’t remove them.

library(MASS)  
ridge\_data <- scale(prod1\_clean)  
ridge\_data <- as.data.frame(ridge\_data)  
  
lm\_seq<- seq(0,1000,0.01)  
  
## Create Model & Plot  
mod\_rid<- lm.ridge(prod1.customers~., data=ridge\_data, lambda = lm\_seq)  
  
plot(mod\_rid)  
select(mod\_rid)  
  
  
## Using Select our smallest value occurs when lambda is at 15.29  
plot(lm\_seq, mod\_rid$GCV, main="GCV of Ridge Regression", type="l", xlab=expression(lambda), ylab="GCV")  
  
## Create optimal Ridge Model using appropriate lambda value  
  
mod\_rid$lambda[1530]  
coef(mod\_rid)[1530,]  
  
  
## Use GLMNET package with appropriate lambda value  
library(glmnet)  
  
x<- prod1\_clean %>% select(-prod1.customers) %>% data.matrix()  
y<- prod1\_clean$prod1.customers  
  
cv\_fit<- cv.glmnet(x,y, alpha = 0, lambda = lm\_seq)  
plot(cv\_fit)  
  
min\_lambda<- cv\_fit$lambda.min  
min\_lambda  
  
## Using Cross Validation our optimal lambda is 7.67  
## Predict new values using CV lambda value  
fit <- cv\_fit$glmnet.fit  
summary(fit)  
  
y\_pred<- predict(fit, s =min\_lambda, newx = x)  
  
## Compute Summary Stats on This model  
sst <- sum((y - mean(y))^2)  
sse <- sum((y\_pred - y)^2)  
rsq <- 1 - sse / sst  
rsq

After generating these models, I gauged their performance via k fold cross validation for the ridge to simply using the test set those using OSL. This led to the conclusion of using the discovered using the subset selection process.

plot(mod1)  
plot(mod2)  
plot(mod3)  
plot(mod6)  
  
## Test Model 6 on Test Set as well as Training Set  
  
train$prediction <- predict(mod6,train, type= "response")  
head(train)  
  
test$prediction <- predict(mod6, test, type = "response")  
head(test)  
  
## Perform other Summary Statistics (Correlation, RMSE, MAE)  
  
summary(mod6)  
  
# Training Set  
  
train\_corr <- round(cor(train$prediction, train$prod1.customers),2)  
train\_RMSE <- round(sqrt(mean((train$prediction - train$prod1.customers)^2)))  
train\_MAE <- round(mean(abs(train$prediction - train$prod1.customers)),2)  
  
train\_stats <- c(train\_corr,train\_RMSE,train\_MAE)  
train\_stats  
  
# Test Set  
  
test\_corr <- round(cor(test$prediction, test$prod1.customers),2)  
test\_RMSE <- round(sqrt(mean((test$prediction - test$prod1.customers)^2)))  
test\_MAE <- round(mean(abs(test$prediction - test$prod1.customers)),2)  
  
test\_stats <- c(test\_corr,test\_RMSE,test\_MAE)  
test\_stats

This led me to choose *Model 6* as the best of the ones built. It preformed the best amongst the training and test set but had room to improve. Doing so however requires going back to the data collection process.

### V. Summary

As I finished this project I found that there is more complexity to the problem at hand then what was first assumed. I am happy with the model I constructed but I believe with some improvements in the data I can reach one which will perform better.

This leads me to my first recommendation which is to use the presence of the synergies found in my model to better understand which advertisers interact and how many different touch points are there. My model only assumes pairwise connections but there can be interactions with 3 or 4 distinct combinations working together.

If we build a multi-touch attribution model or purchase a service we can understand how many touch points it takes a user to convert. Having this data will give us a deeper understanding of these relationships and their magnitude. Understanding this will assist in variable selection which in turn may produce a better model.

While this is being worked on I recommend testing this model by adjusting spending allocations and spending considerably less on those that don’t appear in this model. Doing so diverts money that would have been wasted on channels that don’t produce any new customers into ones that have the best cost efficiencies. As we have seen in the output of *Model 6* there are a few partners whose interaction is resulting in negative customer acquisition!! One can hypothesize that some of the ad partners that we currently employ have massive overlap in users to their respective platforms, spending money on both could be cathing the same eyes multiple times so spending on both together actually hurts customer acquisition. One could use these findings to run a regional analysis on those platforms to see why this occuring.

My last recommendation would be to further enhance the depth of data that is collected. One factor that was not discussed but impacted the results is the presence of customers who purchased without the influence of advertising. This model assumed that 100% of the variation of customers acquired was due to advertising. That is not the case in most organizations which can use brand recognition to influence purchase behavior. The simplest example here is coca cola which is something that almost everyone remembers but isn’t necessarily because you saw them on Facebook. There are a few current estimates of what percentage of customers are from the Non-Paid channel as it is called. If we collected this and removed it from the equation we would have better model performance.

Overall I am confident that the model presented can be used in practice and iterated on with new data as well as different modeling techniques as we learn more about the impact of advertisers on customer’s decision to purchase. All of this of course is easier with more data!