# Marketing Analytics Team Project 4: Bayesian Variable Selection and Customer Scoring

## Executive Summary

We examine the difference between fitting a logistic regression using all given variables and logistic regressions using variables selected using the Bayesian Spike and Slab method. After reducing the variables, we study quadratic and interaction effects. We choose to keep quadratic terms for age and socio-financial orientation. Optimizing for profits using this model yields an optimal target of the top 2.13% in terms of response probabilities. This model yields higher profits than the full-variable model. FServ must also consider ways to reduce their fixed costs. Actual revenues do not justify $100,000 fixed costs at a target population of 30,000. Similarly, expected revenues (calculated as response probabilities multiplied by revenue per response) do not justify $100,000 fixed costs at a target population of 600,000.

## Scoring Model

We fitted a GLM model to the data, filtered coefficients by significance (p<0.05), and ranked coefficients by absolute value. Rather than showing all 95 coefficients, many of which are insignificant or very small, we show the top six after exponentiation in Figure 3.

## Interpretation

We see that marital status has the largest impact on odds of responding: They are 22.83 times more likely to respond than unmarried people. Men are roughly 8.57 times more likely to respond than females. Respondents to previous mailings similarly are 8.22 times more likely to respond. For each additional point of financial orientation, a person is 1.5 times more likely to respond. Likewise, for each additional household member, a person is 1.37 times more likely. Lastly, people who own small businesses are 1.18 times more likely to respond.

## Bayesian Variable Selection

We used the Spike Slab method as implemented in the BoomSpikeSlab package. This variable selection method chooses 13 variables with inclusion probabilities greater than 0. We then fit another logistic regression using only these 13 variables, as opposed to the full 95. Only 8 of them are statistically significant. We present these in

## Transformations and Interactions

We add additional variables to account for quadratic effects and interaction effects. Middle-aged people have more assets and more family members to take care of, so they should be more likely to buy life insurance compared to younger or older people. Thus, age should have a quadratic effect. We also add a quadratic term for socio-financial orientation: People with higher levels of financial orientation may forgo insurance for higher-risk, higher-reward investments. Age squared is significant at the 1% level, and socio-finance squared is significant at the 10% level.

We also added the following interactions effects: age and gender, marital status and gender, marital status and unemployment rate, gender and unemployment status. Only marital status and gender is significant and then only at the 10% level. It is slightly negative, suggesting that married men respond less than unmarried men and married women. This may be due to relationship dynamics such that married women handle finances.

The best model out of these is the one with both quadratic terms with an AICC of 6023.9. This is better than the full-variable logistic regression, whose AICC is 6107.9. Our model with Bayesian selected variables is more parsimonious and has higher explanatory power. We will use this to validate our logistic regression built with Bayesian selected variables.

## Validation

We looked at mailing 0%-5% of all addresses in increments of 0.01%. There is gradual decrease in actual response rate: We do not see a sharp decrease. We ran an optimization assuming a fixed cost of $100,000, a variable cost of $2.50, and a revenue of $20 per response.

For our original GLM model, we should target the top 2.29%. This yields a maximum profit: -$97,695. Our GLM with variables selected using the Spike and Slab method suggests targeting the top 2.13%. This yields a maximum profit of -$97,635. The Bayesian model saves us $60 by targeting 51 fewer people. We compare the response rates of the top percentiles in Figure 3. We cannot achieve a positive profit due to the high fixed cost: Even targeting everyone only yields revenue of $6,840.

## Targeted Addresses

We use the GLM model with quadratic effects and variables selected using Spike and Slab. We will target the top 2.13% as recommended by our optimization over the holdout. The full address list is provided as a csv file. Expected revenue (calculated as response probabilities times $20 revenue per response) is $79,917.08, still lower than the fixed costs of $100,000.

## Recommendations

We recommend that FServ reconsider their fixed costs and target population size. At a target population of 30,000 people, the profit is -$97,635. Similarly, expected revenue with a target population of 600,000 is $79,917.08, still lower than the estimated fixed costs of $100,000. We also recommend that they use the logistic model with the ten linear terms selected by Spike and Slab and quadratic terms for age and socio-financial orientation: This model suggests targeting the top 2.13% of the target population in terms of their response probabilities.

## Appendix

Figure : Full Variable GLM Important Coefficients

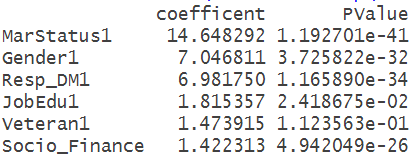


Figure : Bayesian Variables Plus Transformations and Interactions

==========================================================

Dependent variable: Response

Suggested Quadratic Interactions

(1) (2) (3)

----------------------------------------------------------

Age 0.088\*\*\* 0.142\*\* 0.124\*\*

(0.007) (0.058) (0.061)

MailPrefGeneral 0.484\*\*\* 0.486\*\*\* 0.488\*\*\*

(0.049) (0.049) (0.049)

MarStatus1 2.663\*\*\* 2.669\*\*\* 1.649

(0.194) (0.195) (1.898)

Socio\_Finance 0.352\*\*\* 0.183 0.183

(0.026) (0.122) (0.122)

Resp\_DM1 1.849\*\*\* 1.844\*\*\* 1.853\*\*\*

(0.150) (0.150) (0.150)

Gender1 1.921\*\*\* 1.923\*\*\* 0.806

(0.160) (0.160) (1.248)

Affinity\_MailOrder 0.468\*\*\* 0.469\*\*\* 0.471\*\*\*

(0.046) (0.046) (0.046)

HHSize 0.314\*\*\* 0.313\*\*\* 0.314\*\*\*

(0.033) (0.033) (0.033)

ViolentCrime 0.003\*\*\* 0.003\*\*\* 0.003\*\*\*

(0.001) (0.001) (0.001)

SUV1 -0.128 -0.130 -0.133

(0.139) (0.139) (0.140)

UnemployRate 0.108\* 0.109\* 0.078

(0.064) (0.064) (0.223)

MailPrefCollectors -0.057\*\* -0.059\*\* -0.059\*\*

(0.024) (0.024) (0.024)

MailPrefInnovations -0.013 -0.013 -0.013

(0.025) (0.025) (0.025)

I(Age2) -0.0005 -0.0005

(0.001) (0.001)

I(Socio\_Finance2) 0.013 0.014

(0.010) (0.010)

MarStatus1:Gender1 0.435

(0.488)

Age:MarStatus1 0.019

(0.020)

MarStatus1:UnemployRate -0.054

(0.197)

Gender1:UnemployRate 0.099

(0.156)

Constant -26.526\*\*\* -27.570\*\*\* -26.091\*\*\*

(1.100) (1.906) (2.686)

----------------------------------------------------------

Observations 30,000 30,000 30,000

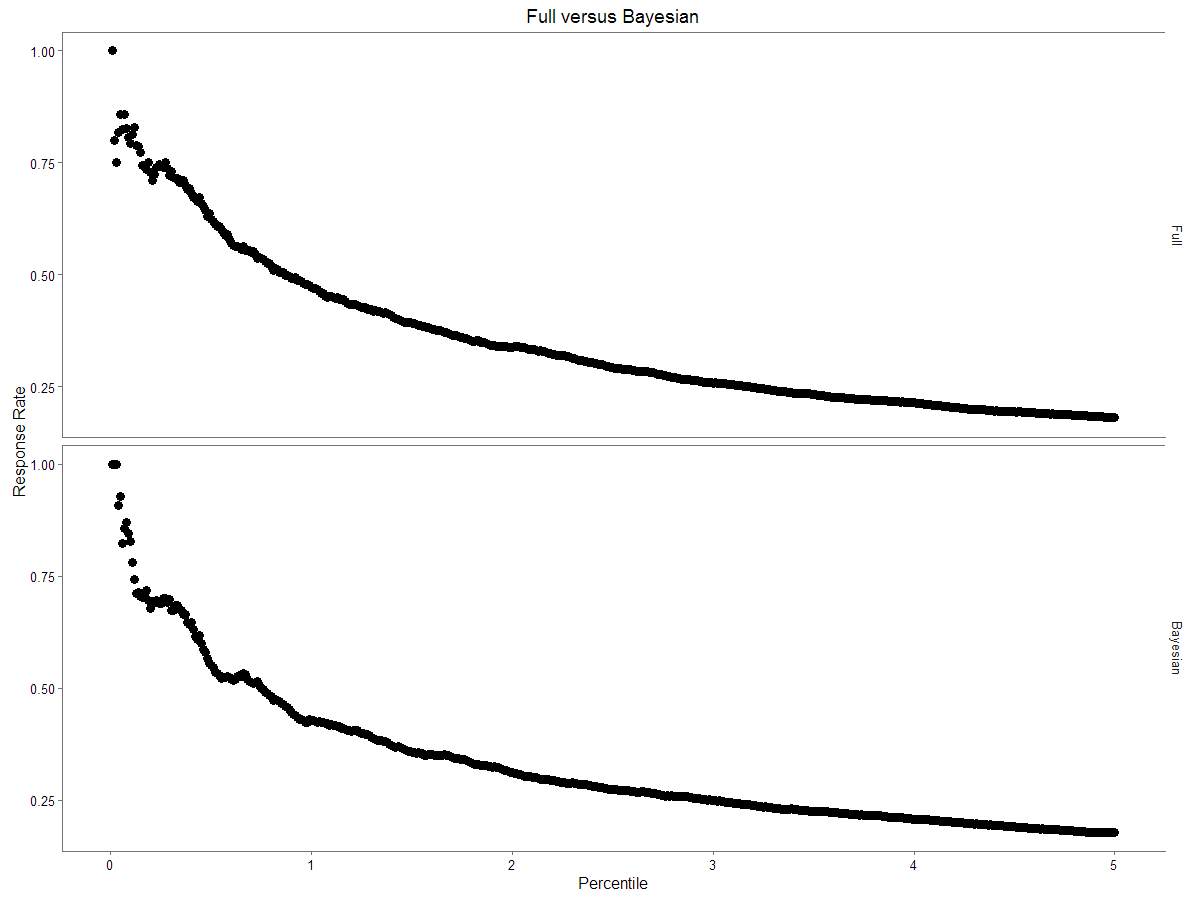
Log Likelihood -997.116 -995.722 -994.723

Akaike Inf. Crit. 2,022.232 2,023.444 2,029.446

==========================================================

Note: \*p<0.1; \*\*p<0.05; \*\*\*p<0.01

Figure : Full versus Bayesian Top Percentiles



**Code**

setwd("C:/Users/Samruddhi Somani/Documents/McCombs/Spring/Marketing/Project 4")

library(BoomSpikeSlab)

library(stargazer)

library(caret)

library(reshape2)

library(ggplot2)

library(ggthemes)

library(Rsolnp)

load('custacquisition\_test.Rdata')

#find categorical variables

dumA = X[, c('Gender', 'Resp\_DM', 'MarStatus', 'USCitizen', 'HomeOwner', 'JobMilitary', 'JobHealthcare', 'JobEdu',

'JobSmallBusiness', 'Veteran', 'RecentlyMovedIn', 'HasComprehensive', 'SUV', 'SportsCar', 'Truck',

'RecentMailOrder')]

dumB = X[,84:96]

dum = cbind(dumA, dumB)

#changing dummies to factors

X\_mod=X

X\_mod[,names(dum)]=lapply(X[,names(dum)],factor)

#programmatic formula creation

nK = length(names(X))

fmla\_comp = rep('', nK) # array of formula components

for (i in 1:nK) { # fill in formular components

fmla\_comp[i] = paste0(names(X)[i])

}

fmla = as.formula(paste("Resp ~ ", paste(fmla\_comp[2:nK], collapse= "+")))

#fitting logit glm

glmfit=glm(fmla,data=X\_mod, family='binomial')

summary(glmfit)

#interpreting

glm\_coefs=summary(glmfit)$coefficients[,c(1,4)]

sig\_coefs = glm\_coefs[order(glm\_coefs[glm\_coefs[,2] < 0.05]),]

order\_coefs = sig\_coefs[order(sig\_coefs[,1], decreasing = TRUE),]

theanswers=cbind.data.frame(coefficent=exp(order\_coefs[1:6,1]),PValue=order\_coefs[1:6,2])

#fitting bayesian model

start=Sys.time()

print(start)

lzp=LogitZellnerPrior(mm,successes=Resp,expected.model.size=20,prior.success.probability=mean(Resp))

ss=logit.spike(fmla,niter=1000,data=X\_mod,initial\_value=glmfit,seed=42,ping=10,nthreads=3,prior=lzp)

end=Sys.time()

timespent=end-start

save.image('oops.Rdata')

print (timespent)

#pulling model names

coef=summary(ss)$coefficients

new=rownames(coef[coef[,5]!=0,])[2:14]

for (n in 1:length(new)){

new[n]=gsub ('[[:digit:]]+', '', new[n])

}

fmla\_new=paste("Resp~",paste(new,collapse="+"))

glmfit2=glm(as.formula(fmla\_new),data=X\_mod,family='binomial')

#quadratic

fmla\_new2=paste0(fmla\_new,"+I(Age^2)+I(Socio\_Finance^2)")

glmfit3=glm(as.formula(fmla\_new2),data=X\_mod, family='binomial')

#interactions

fmla\_new3=paste0(fmla\_new2,"+Gender:MarStatus+Age:MarStatus+UnemployRate:MarStatus+UnemployRate:Gender")

glmfit4=glm(as.formula(fmla\_new3),data=X\_mod,family='binomial')

################

#VALIDATIONS###

################

load("custacquisition\_holdout.Rdata")

#changing dummies to factors

X\_mod=X

X\_mod[,names(dum)]=lapply(X[,names(dum)],factor)

p=predict(glmfit,X\_mod,type="response")

bp=predict(glmfit3,X\_mod,type="response")

#translating to class

p\_f=as.integer(p>0.5)

bp\_f=as.integer(bp>0.5)

tbl=rbind(table(p\_f),table(bp\_f),table(Resp))

rownames(tbl)=c('Frequentist','Bayesian','Actual')

#confusion matrix

confusionMatrix(p\_f, Resp)

confusionMatrix(bp\_f,Resp)

### original glm

ind = order(p, decreasing = TRUE)

prob.rank = p[ind]

df = data.frame(prob = p, response = Resp)

df.order = df[order(df[,1], decreasing = TRUE),]

df.order$rank = 1:dim(df)[1]

head(df.order)

df.order$percentile = df.order$rank / dim(df)[1] \* 100

head(df.order)

rate\_original = rep(0,500)

for (i in 1:500) {

df.select = df.order[df.order$percentile < i/100,]

rate\_original[i] = sum(df.select$Resp)/dim(df.select)[1]

}

#optimization

get\_profit <- function(df, x) {

df.select = df.order[df.order$percentile < x,]

revenue = sum(df.select$Resp \* 20)

cost = dim(df.select)[1] \* 2.5 + 100000

profit = revenue - cost

all = list(revenue = revenue, cost = cost, profit = profit)

return(all)

}

profits = c(NA,500)

revenue = c(NA,500)

cost = c(NA,500)

for (i in 1:500) {

profits[i] = get\_profit(df.order, i/100)$profit

revenue[i] = get\_profit(df.order, i/100)$revenue

cost[i] = get\_profit(df.order, i/100)$cost

}

plot(profits,main="Simple Logit: Profits versus Targeted Percentile",xlab="Percentile x 100",ylab="Profit")

which.max(profits) / 100

plot(rate, type = 'b',main="Response Rate versus Percentile",xlab="Percentile x 100",ylab="Response Rate")

###bayesian glm

ind = order(bp, decreasing = TRUE)

prob.rank = bp[ind]

df = data.frame(prob = bp, response = Resp)

df.order = df[order(df[,1], decreasing = TRUE),]

df.order$rank = 1:dim(df)[1]

head(df.order)

df.order$percentile = df.order$rank / dim(df)[1] \* 100

head(df.order)

rate\_bayesian = rep(0,500)

for (i in 1:500) {

df.select = df.order[df.order$percentile < i/100,]

rate\_bayesian[i] = sum(df.select$Resp)/dim(df.select)[1]

}

plot(rate, type = 'b',main="Response Rate versus Percentile",xlab="Percentile x 100",ylab="Response Rate")

#optimization

get\_profit <- function(df, x) {

df.select = df.order[df.order$percentile < x,]

revenue = sum(df.select$Resp \* 20)

cost = dim(df.select)[1] \* 2.5 + 100000

profit = revenue - cost

all = list(revenue = revenue, cost = cost, profit = profit)

return(all)

}

profits = c(NA,500)

revenue = c(NA,500)

cost = c(NA,500)

for (i in 1:500) {

profits[i] = get\_profit(df.order, i/100)$profit

revenue[i] = get\_profit(df.order, i/100)$revenue

cost[i] = get\_profit(df.order, i/100)$cost

}

plot(profits,main="Bayesian: Profits versus Targeted Percentile",xlab="Percentile x 100",ylab="Profit")

which.max(profits) / 100

#visualization

rate=cbind.data.frame(Full=rate\_original,Bayesian=rate\_bayesian,i=seq(0.01,5,by=0.01))

rate\_melt=melt(rate,id="i")

ggplot(data=rate\_melt)+geom\_point(aes(x=i,y=value),size=3)+theme\_few()+facet\_grid(variable~.)+labs(title="Full versus Bayesian",y="Response Rate",x="Percentile")

#address set

load("custacquisition\_addresslist.Rdata")

#changing dummies to factors

X\_mod=X

X\_mod[,names(dum)]=lapply(X[,names(dum)],factor)

new\_p=predict(glmfit3,X\_mod,type="response")

###bayesian glm

ind = order(new\_p, decreasing = TRUE)

df = data.frame(index = ind, prob = new\_p)

df.order = df[order(df[,2], decreasing = TRUE),]

head(df.order)

df.order$rank = 1:length(new\_p)

head(df.order)

df.order$percentile = df.order$rank / dim(df)[1] \* 100

head(df.order)

selected = df.order[df.order$percentile<2.13,]

dim(selected)

tail(selected)

write.csv(selected, 'address.csv')

sum(selected$prob\*20)

stargazer(glmfit2,glmfit3,glmfit4,type="text",no.space=TRUE,column.labels=c("Suggested","Quadratic","Interactions"))