model_HA_LM.R

atchirc

Mon May 22 23:37:38 2017

```r
library(MASS)
library(car)
library(DataCombine)   # Pair wise correlation
library(stargazer)
library(dplyr)         # Data aggregation
library(glmnet)
source('../atchircUtils.R')


data    <- read.csv('../../intrim/eleckart.csv')


# KPI selection
# units, product_mrp, list_mrp, COD, Prepaid are factors
# Insig : Affiliates corr OnlineMarketing
# Insig : Radio  corr Other
# Insig : Digitial, ContentMarketing corr SEM
# delivery(b/c)days are corr, lets choose deliverycdays
# will use marketing levers rather TotalInvestment

# Filter significant KPIs
model_data <- subset(data, product_analytic_sub_category=='HomeAudio',
                     select = -c(product_analytic_sub_category,product_mrp,
                                 units,COD,Prepaid,deliverybdays,
                                 TotalInvestment,Affiliates,Radio,Digital,
                                 ContentMarketing,sla,procurement_sla))

model_data_org <- model_data
model_data[,c(8:12)] <- model_data[,c(8:12)]*10000000


# # ***************************************************************************
# #                   FEATURE ENGINEERING -PASS2  ----
# # ***************************************************************************
#
# # . . . . List Price Inflation ----
model_data$chnglist <- c(0,diff(model_data$list_mrp))
#
# # . . . . Discount Inflation ----
model_data$chngdisc <- c(0,diff(model_data$discount))
#
```

*

```
**PROCs:**
```

Linear, Ridge and Lasso Model are wrapped with abstract functions. This would facilitate readable code for model building and Model otpimization. Set Class definitions

```
setOldClass('elnet')
setClass(Class = 'atcglmnet',
         representation (
           R2 = 'numeric',
           mdl = 'elnet',
           pred = 'matrix'
         )
)
```

```
setOldClass('lm')
setClass(Class = 'atclm',
         representation (
           R2 = 'numeric',
           mdl = 'lm',
           pred = 'matrix'
         )
)
```

Finding min lambda from 1000 iterations Function to find Min Lambda using bootstrap method. minlambda identified over 1000 cross validation trails. observed minlambda used for Ridge and Lasso regression.

```
findMinLambda <- function(x,y,alpha,folds) {
  lambda_list <- list()
  for (i in 1:1000) {
    cv.out <- cv.glmnet(as.matrix(x), as.vector(y), alpha=alpha,
                        nfolds=folds)
    lambda_list <- append(lambda_list, cv.out$lambda.min)
  }
  return(min(unlist(lambda_list)))
}
```

Linear Model with Regularization Wrapper function for Ridge and Lasso regression. functions performs Ridge/Lasso regression and returns R2, Model and Predicted values as `atcglmnet` object

```
atcLmReg <- function(x,y,l1l2,folds) {
  # l1l2 = 0 for L1,  1 for L2

  if (l1l2) { # Lasso/L2
    min_lambda <- findMinLambda(x,y,1,folds)
  } else { # Ridge/L1
    min_lambda <- findMinLambda(x,y,0,folds)
  }
  mdl        <- glmnet(x,y,alpha=l1l2,lambda = min_lambda)
```

```r
  pred       <- predict(mdl,s= min_lambda,newx=x)

  # MSE
  mean((pred-y)^2)
  R2 <- 1 - (sum((y-pred )^2)/sum((y-mean(pred))^2))
  return(new('atcglmnet', R2 = R2, mdl=mdl, pred=pred))
}
```

*

MODELING

```r
# Prune KPI as part of model optimization
model_data <- na.omit(model_data)
model_data <- subset(model_data,select=-c(TV,deliverycdays,NPS,
                                          chnglist,OnlineMarketing,
                                          Other,SEM,discount,list_mrp))
```

Linear Model:

```r
mdl      <- lm(gmv~., data=model_data)
step_mdl <- stepAIC(mdl,direction = 'both',trace = FALSE)

stargazer(mdl,step_mdl, align = TRUE, type = 'text',
          title='Linear Regression Results', single.row=TRUE)
```

```
##
## Linear Regression Results
## ===========================================================================================
##                                                  Dependent variable:
##                               -------------------------------------------------------------
##                                                         gmv
##                                           (1)                            (2)
## -------------------------------------------------------------------------------------------
## week                            -29,323.970 (19,573.880)       -29,323.970 (19,573.880)
## n_saledays                    582,266.300*** (184,825.600)   582,266.300*** (184,825.600)
## Sponsorship                   210,550.800*** (53,209.420)    210,550.800*** (53,209.420)
## chngdisc                      133,106.100*** (47,714.060)    133,106.100*** (47,714.060)
## Constant                    4,146,555.000*** (799,146.700) 4,146,555.000*** (799,146.700)
## -------------------------------------------------------------------------------------------
## Observations                               50                             50
## R2                                       0.491                          0.491
## Adjusted R2                              0.446                          0.446
## Residual Std. Error (df = 45)        2,056,667.000                  2,056,667.000
## F Statistic (df = 4; 45)               10.866***                      10.866***
## ===========================================================================================
## Note:                                                     *p<0.1; **p<0.05; ***p<0.01
```

```r
knitr::kable(viewModelSummaryVIF(step_mdl))
```

var	Estimate	Std.Error	t-value	Pr(>|t|)	Significance	vif
chngdisc	133106	47714	2.790	0.007709	**	1.042230
n_saledays	582266	184826	3.150	0.002897	**	1.022261
Sponsorship	210551	53209	3.957	0.000267	***	1.048689
week	-29324	19574	-1.498	0.141085	NA	1.019193

```r
pred_lm <- predict(step_mdl, model_data)
```

Regularized Linear Model:

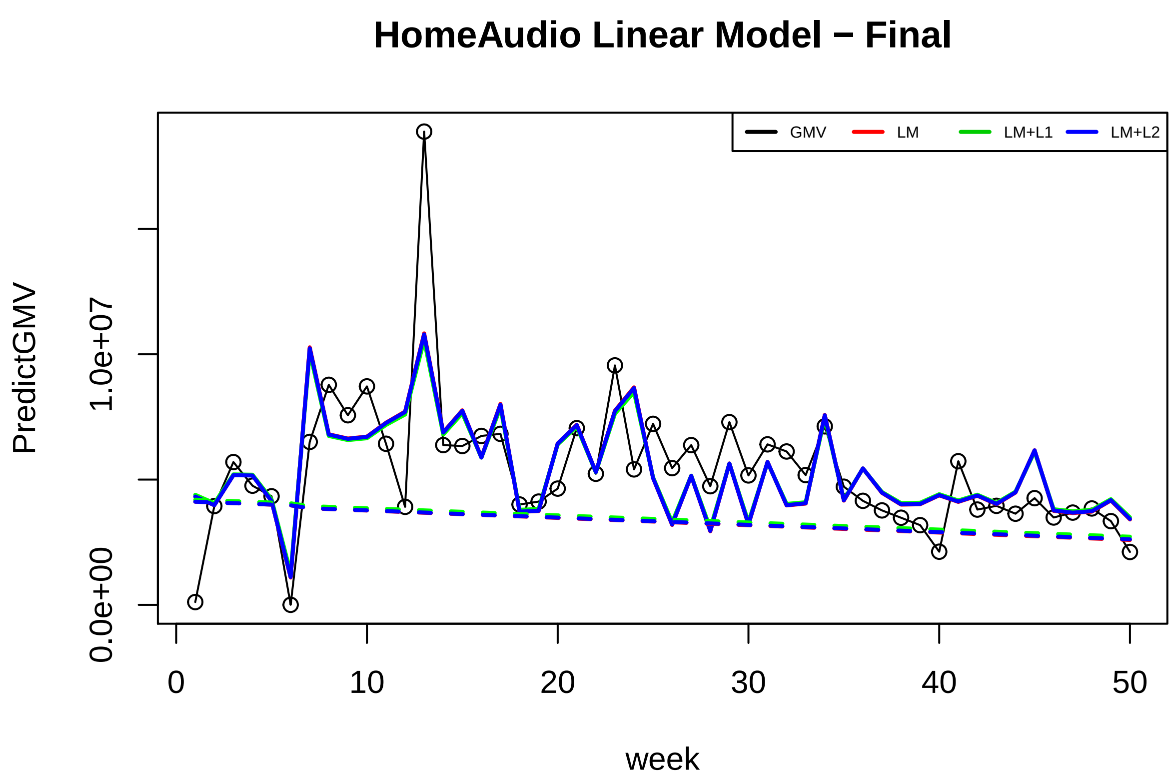
```r
x = as.matrix(subset(model_data, select=-gmv))
y = as.vector(model_data$gmv)

ridge_out <- atcLmReg(x,y,0,3)  # x, y, alpha, nfolds
lasso_out <- atcLmReg(x,y,1,3)  # x, y, alpha, nfolds
```

*

PLOTTING MODEL RESULTS

Plot Model prediction and base sales:

```
plot(model_data$gmv,main = 'HomeAudio Linear Model - Final',
     xlab='week',ylab='PredictGMV')
lines(model_data$gmv)
lines(pred_lm,col='red',lwd=2)
lines(ridge_out@pred,col='green',lwd=2)
lines(lasso_out@pred,col='blue',lwd=2)
lines(step_mdl$coefficients['(Intercept)']+step_mdl$coefficients['week']*model_data$week,
      lty=2,lwd=2,col='red')
lines(ridge_out@mdl$a0+ridge_out@mdl$beta['week',1]*model_data$week,
      lty=2,lwd=2,col='green')
lines(lasso_out@mdl$a0+lasso_out@mdl$beta['week',1]*model_data$week,
      lty=2,lwd=2,col='blue')
legend('topright',inset=0, legend=c('GMV','LM','LM+L1','LM+L2'),horiz = TRUE,
       lwd = 2, col=c(1:4), cex = 0.5)
```



*

*Model Coefficients:**

```r
coeff_lm <- as.data.frame(as.matrix(coef(step_mdl)))
coeff_l1 <- as.data.frame(as.matrix(coef(ridge_out@mdl)))
coeff_l2 <- as.data.frame(as.matrix(coef(lasso_out@mdl)))


lm_df=data.frame('x'=rownames(coeff_lm),'y'=coeff_lm)
colnames(lm_df) = c('coeff','lm')
l1_df=data.frame('x'=rownames(coeff_l1),'y'=coeff_l1)
colnames(l1_df)= c('coeff','l1')
l2_df=data.frame('x'=rownames(coeff_l2),'y'=coeff_l2)
colnames(l2_df) <- c('coeff','l2')

smry <- merge(lm_df,l1_df,all = TRUE)
smry <- merge(smry,l2_df,all=TRUE)

print(smry)
```

```
##         coeff         lm         l1         l2
## 1 (Intercept) 4146554.64 4224907.02 4146289.98
## 2    chngdisc  133106.13  128201.68  132160.32
## 3  n_saledays  582266.29  550913.94  577716.05
## 4 Sponsorship  210550.79  199684.70  209386.58
## 5        week  -29323.97  -28390.65  -28884.37
```

```r
print(paste0('Ridge regression R2 : ',ridge_out@R2))
```

```
## [1] "Ridge regression R2 : 0.490217232426603"
```

```r
print(paste0('Lasso regression R2 : ',lasso_out@R2))
```

```
## [1] "Lasso regression R2 : 0.4912878593603"
```

```r
print(paste0('Linear Mode      R2 : ',getModelR2(step_mdl)))
```

```
## [1] "Multiple R-squared:  0.4913,\tAdjusted R-squared:  0.4461 "
## [1] "Linear Mode      R2 : Multiple R-squared:  0.4913,\tAdjusted R-squared:  0.4461 "
```

*

Significant KPI

Lasso(LM+L1) regression results a simple explainable model with significant KPIs as `Discount Inflation`, `Deliverycday`, `sale days`, `Sponsorship Discount`, `week`, `NPS`

```
# Model Optimization
```

```
# coeff              lm            l1            l2
# 1      (Intercept) -4.205266e+06  3.743013e+06 -2.335133e+06
# 2         chngdisc            NA  3.544890e+04  2.297922e+04
# 3         chnglist            NA  1.274977e-05 -2.125097e-06
# 4    deliverycdays            NA  1.399561e+05  9.078950e+04
# 5         discount  6.485938e+04  6.976909e+03  2.857188e+04
# 6         list_mrp  3.520229e-04  2.898529e-04  3.339852e-04
# 7       n_saledays  2.494251e+05  2.376959e+05  2.589315e+05
# 8              NPS            NA -8.022442e-03  0.000000e+00
# 9  OnlineMarketing  4.147731e-02  2.946905e-02  4.207859e-02
# 10           Other            NA  6.919302e-03  1.216733e-02
# 11             SEM -5.362909e-02 -3.241843e-02 -4.862319e-02
# 12     Sponsorship  2.619984e+05  2.082814e+05  2.920367e+05
# 13              TV            NA -1.952227e+05 -5.558398e+05
# 14            week            NA -6.411466e+03 -1.947268e+03
# [1] "Ridge regression R2 : 0.635910648911486"
# [1] "Lasso regression R2 : 0.648390286764186"
# [1] "Multiple R-squared:  0.6301,\tAdjusted R-squared:  0.5808 "
# [1] "Linear Mode      R2 :
#         Multiple R-squared:  0.6301,\tAdjusted R-squared:  0.5808 "
```

```
# coeff          lm         l1         l2
# 1 (Intercept) 4146554.64 4218388.05 4146289.98
# 2    chngdisc  133106.13  128626.17  132160.32
# 3  n_saledays  582266.29  553555.19  577716.05
# 4 Sponsorship  210550.79  200601.62  209386.58
# 5        week  -29323.97  -28472.55  -28884.37
# [1] "Ridge regression R2 : 0.490395705980193"
# [1] "Lasso regression R2 : 0.4912878593603"
# [1] "Multiple R-squared:  0.4913,\tAdjusted R-squared:  0.4461 "
# [1] "Linear Mode      R2 :
#       Multiple R-squared:  0.4913,\tAdjusted R-squared:  0.4461 "
```