model_GA_LM_ad.R

atchirc

Mon May 22 16:25:28 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=='GamingAccessory',
                     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))
#

# # . . . . Ad Stock ----
model_data$adTV               <- as.numeric(
  stats::filter(model_data$TV,filter=0.5,method='recursive'))
model_data$adSponsorship      <- as.numeric(
  stats::filter(model_data$Sponsorship,filter=0.5,method='recursive'))
```

```r
model_data$adOnlineMarketing  <- as.numeric(
  stats::filter(model_data$OnlineMarketing,filter=0.5,method='recursive'))
model_data$adSEM              <- as.numeric(
  stats::filter(model_data$SEM,filter=0.5,method='recursive'))
model_data$adOther            <- as.numeric(
  stats::filter(model_data$Other,filter=0.5,method='recursive'))

model_data <- subset(model_data,select = -c(TV,Sponsorship,
                                            OnlineMarketing,
                                            SEM,Other))
```

*

    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

```
# Prune KPI as part of model optimization
model_data <- na.omit(model_data)
model_data <- subset(model_data,select=-c(list_mrp,adTV,adSEM,NPS))
# dim(model_data)
```

Linear Model:

```
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                   15,218.680 (27,051.370)
## discount               14,942.930 (25,501.670)
## deliverycdays         75,586.340 (177,074.700)
## n_saledays            58,849.440 (102,759.900)
## chnglist                  0.0001 (0.0001)              0.0001 (0.0001)
## chngdisc               33,790.440 (20,721.670)     42,448.330*** (15,412.500)
## adSponsorship         53,480.060** (23,275.050)     30,644.960* (17,620.470)
## adOnlineMarketing          0.008 (0.008)                0.017*** (0.005)
## adOther                    0.007 (0.007)
## Constant             376,946.100 (1,609,897.000) 1,521,518.000*** (333,705.400)
## ------------------------------------------------------------------------------
## Observations                    53                           53
## R2                             0.514                        0.477
## Adjusted R2                    0.412                        0.433
## Residual Std. Error   1,089,259.000 (df = 43)      1,069,379.000 (df = 48)
## F Statistic            5.046*** (df = 9; 43)       10.932*** (df = 4; 48)
## ==============================================================================
## Note:                                             *p<0.1; **p<0.05; ***p<0.01
```

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

var	Estimate	Std.Error	t-value	Pr(>|t|)	Significance	vif
adOnlineMarketing	1.746e-02	4.746e-03	3.679	0.000592	***	1.415692
adSponsorship	3.064e+04	1.762e+04	1.739	0.088415	.	1.417105
chngdisc	4.245e+04	1.541e+04	2.754	0.008287	**	1.033801
chnglist	1.135e-04	7.405e-05	1.532	0.132084	NA	1.040661

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

Regularized Linear Model:

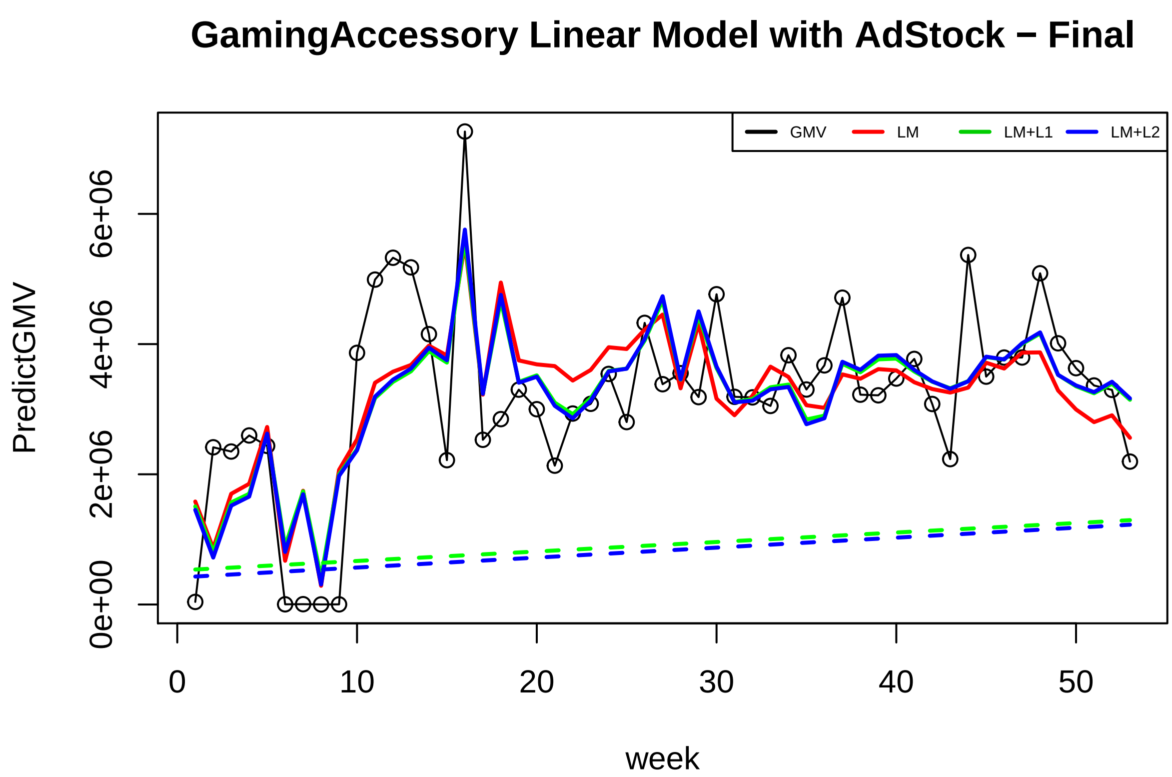
```r
x = as.matrix(subset(model_data, select=-gmv))
y = as.vector(model_data$gmv)

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$gmv,main = 'GamingAccessory Linear Model with AdStock - Final',
     xlab='week',ylab='PredictGMV')
lines(model_data$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:**

```
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) 1.521518e+06 5.227609e+05 4.143994e+05
## 2  adOnlineMarketing 1.746177e-02 9.099769e-03 8.460501e-03
## 3            adOther           NA 6.183418e-03 7.181187e-03
## 4      adSponsorship 3.064495e+04 4.809943e+04 5.305113e+04
## 5           chngdisc 4.244833e+04 3.194748e+04 3.381321e+04
## 6           chnglist 1.134451e-04 9.199252e-05 9.650508e-05
## 7      deliverycdays           NA 6.655708e+04 7.149166e+04
## 8           discount           NA 1.365110e+04 1.441083e+04
## 9          n_saledays           NA 5.432234e+04 5.728148e+04
## 10              week           NA 1.457750e+04 1.530895e+04
```

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

```
## [1] "Ridge regression R2 : 0.512304483802246"
```

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

```
## [1] "Lasso regression R2 : 0.513612042455383"
```

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

```
## [1] "Multiple R-squared:  0.4767,\tAdjusted R-squared:  0.4331 "
## [1] " Linear regression R2 : Multiple R-squared:  0.4767,\tAdjusted R-squared:  0.4331 "
```

*

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) 1.521518e+06 5.227609e+05 4.182997e+05
# 2  adOnlineMarketing 1.746177e-02 9.099769e-03 8.468539e-03
# 3            adOther           NA 6.183418e-03 7.165210e-03
# 4      adSponsorship 3.064495e+04 4.809943e+04 5.300454e+04
# 5           chngdisc 4.244833e+04 3.194748e+04 3.381739e+04
# 6           chnglist 1.134451e-04 9.199252e-05 9.642221e-05
# 7      deliverycdays           NA 6.655708e+04 7.115827e+04
# 8           discount           NA 1.365110e+04 1.435628e+04
# 9         n_saledays           NA 5.432234e+04 5.712563e+04
# 10              week           NA 1.457750e+04 1.530543e+04
# [1] "Ridge regression R2 : 0.512304483802246"
# [1] "Lasso regression R2 : 0.513607064233856"
# [1] "Multiple R-squared:  0.4767,\tAdjusted R-squared:  0.4331 "
# [1] " Linear regression R2 :
#        Multiple R-squared:  0.4767,\tAdjusted R-squared:  0.4331 "
```

```
# > print(smry)
# coeff           lm           l1           l2
# 1        (Intercept) 1.521518e+06 5.114755e+05 4.143994e+05
# 2  adOnlineMarketing 1.746177e-02 9.065892e-03 8.460501e-03
# 3            adOther           NA 6.264452e-03 7.181187e-03
# 4      adSponsorship 3.064495e+04 4.848377e+04 5.305113e+04
# 5           chngdisc 4.244833e+04 3.211148e+04 3.381321e+04
# 6           chnglist 1.134451e-04 9.246335e-05 9.650508e-05
# 7      deliverycdays           NA 6.725724e+04 7.149166e+04
# 8           discount           NA 1.374273e+04 1.441083e+04
# 9         n_saledays           NA 5.462490e+04 5.728148e+04
# 10              week           NA 1.461502e+04 1.530895e+04
#
# > ridge_out@R2
# [1] 0.5125068
#
# > lasso_out@R2
# [1] 0.513612
```