model_GA_LM.R

atchirc

Mon May 22 16:23:42 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))
```

*

```
**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)
```

```
  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(NPS,list_mrp,discount,SEM,TV))
```

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                   10,499.000 (19,992.170)
## deliverycdays         101,508.000 (158,678.500)      156,675.600* (81,054.250)
## n_saledays             97,749.970 (97,572.260)
## Sponsorship           92,602.130** (35,296.750)      85,066.790** (33,191.760)
## OnlineMarketing             0.022* (0.013)                0.029*** (0.010)
## Other                       0.012 (0.010)                  0.014 (0.010)
## chnglist                  0.0001 (0.0001)
## chngdisc              39,860.640** (15,123.410)      39,916.950*** (14,637.540)
## Constant            1,150,255.000*** (376,624.400) 1,262,486.000*** (327,632.900)
## ----------------------------------------------------------------------------------
## Observations                    53                             53
## R2                             0.557                          0.533
## Adjusted R2                    0.476                          0.483
## Residual Std. Error    1,028,105.000 (df = 44)        1,020,909.000 (df = 47)
## F Statistic             6.905*** (df = 8; 44)         10.729*** (df = 5; 47)
## ==================================================================================
## 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	3.992e+04	1.464e+04	2.727	0.008955	**	1.023096
deliverycdays	1.567e+05	8.105e+04	1.933	0.059277	.	1.063322
OnlineMarketing	2.855e-02	1.028e-02	2.776	0.007868	**	1.922544
Other	1.419e-02	9.570e-03	1.483	0.144733	NA	1.559352
Sponsorship	8.507e+04	3.319e+04	2.563	0.013643	*	1.809277

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

Regularized Linear Model:

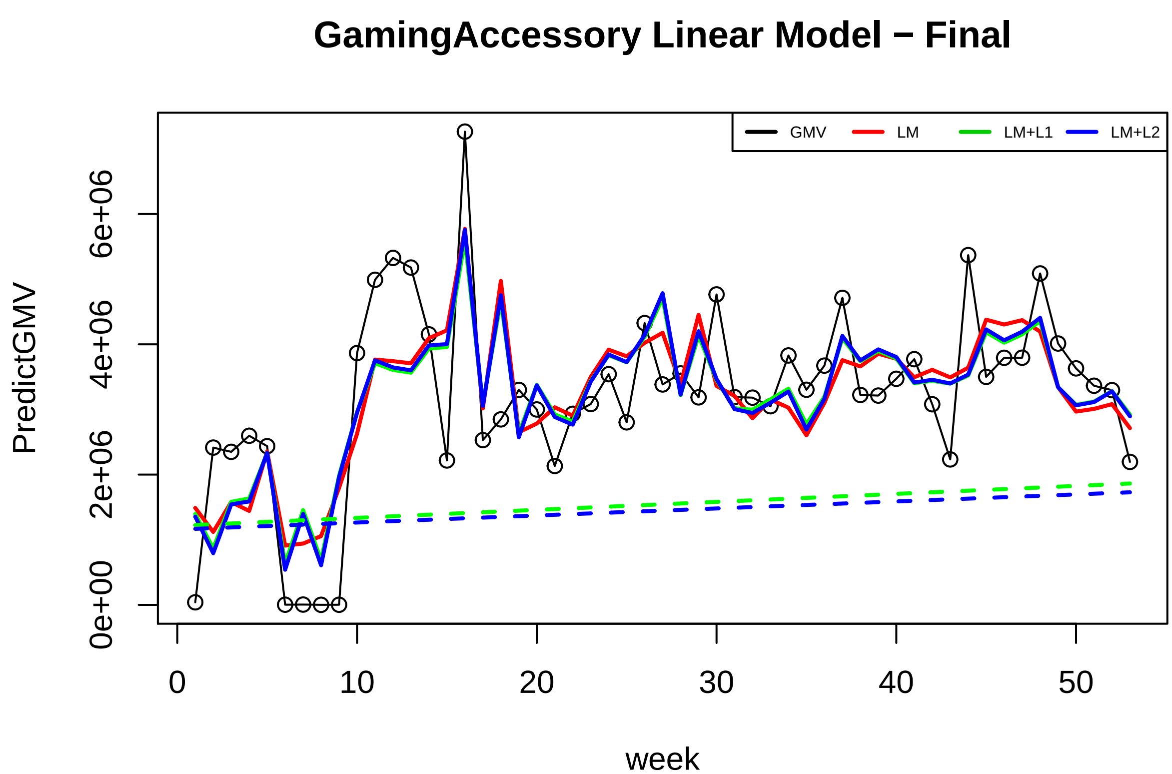
```r
x = as.matrix(subset(model_data, select=-gmv))
y = as.vector(model_data$gmv)

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:

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

```
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) 1.262486e+06 1.213201e+06 1.156598e+06
## 2        chngdisc 3.991695e+04 3.721809e+04 3.957890e+04
## 3        chnglist           NA 6.749217e-05 6.858208e-05
## 4   deliverycdays 1.566756e+05 7.871292e+04 9.727056e+04
## 5      n_saledays           NA 8.940047e+04 9.559254e+04
## 6 OnlineMarketing 2.854816e-02 2.210204e-02 2.247788e-02
## 7           Other 1.419332e-02 1.061425e-02 1.215068e-02
## 8     Sponsorship 8.506679e+04 8.566758e+04 9.183745e+04
## 9            week           NA 1.226571e+04 1.078629e+04
```

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

```
## [1] "Ridge regression R2 : 0.555255369708761"
```

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

```
## [1] "Lasso regression R2 : 0.556613153887992"
```

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

```
## [1] "Multiple R-squared:  0.533,\tAdjusted R-squared:  0.4833 "
## [1] " Linear regression R2 : Multiple R-squared:  0.533,\tAdjusted R-squared:  0.4833 "
```

*

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
```

```
# > print(smry)
# coeff              lm            l1            l2
# 1      (Intercept)  7.829190e+06  5.685142e+06  6.256789e+06
# 2         chngdisc            NA  2.250616e+04  1.691863e+04
# 3         chnglist            NA  2.808944e-05  1.758693e-05
# 4    deliverycdays  2.525241e+05  1.398720e+05  2.145395e+05
# 5         discount  4.207626e+04  2.319559e+04  3.270155e+04
# 6         list_mrp            NA  1.527954e-05 -1.149641e-05
# 7       n_saledays  1.359498e+05  9.466238e+04  1.084861e+05
# 8              NPS -1.571834e-02 -1.035586e-02 -1.195506e-02
# 9  OnlineMarketing            NA  1.038276e-02  8.208942e-03
# 10           Other  1.185966e-02  6.355664e-03  8.722440e-03
# 11             SEM -4.590140e-02 -2.585642e-02 -3.991088e-02
# 12     Sponsorship  1.731888e+05  1.077670e+05  1.491207e+05
# 13              TV            NA  1.415739e+05  1.427371e+05
# 14            week            NA  7.252171e+03  1.590589e+03
#
# > ridge_out@R2
# [1] 0.6335693
#
# > lasso_out@R2
# [1] 0.6466638
```

```
# coeff             lm           l1           l2
# 1     (Intercept) 1.262486e+06 1.213201e+06 1.156598e+06
# 2        chngdisc 3.991695e+04 3.721809e+04 3.957890e+04
# 3        chnglist           NA 6.749217e-05 6.858208e-05
# 4   deliverycdays 1.566756e+05 7.871292e+04 9.727056e+04
# 5      n_saledays           NA 8.940047e+04 9.559254e+04
# 6 OnlineMarketing 2.854816e-02 2.210204e-02 2.247788e-02
# 7           Other 1.419332e-02 1.061425e-02 1.215068e-02
# 8     Sponsorship 8.506679e+04 8.566758e+04 9.183745e+04
# 9            week           NA 1.226571e+04 1.078629e+04
# [1] "Ridge regression R2 : 0.555255369708761"
# [1] "Lasso regression R2 : 0.556613153887992"
# [1] "Multiple R-squared:  0.533,\tAdjusted R-squared:  0.4833 "
# [1] " Linear regression R2 :
#       Multiple R-squared:  0.533,\tAdjusted R-squared:  0.4833 "
# >
```