model_GA_MM_ad.R

arman

Sat May 27 13:42:00 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'))
model_data$adOnlineMarketing   <- as.numeric(
  stats::filter(model_data$OnlineMarketing,filter=0.5,method='recursive'))
```

```r
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'))

# Prune regular
model_data <- subset(model_data,select = -c(TV,Sponsorship,
                                            OnlineMarketing,
                                            SEM,Other))

#  . . . . Shift KPI to void NAN ----
model_data$chngdisc <- min(model_data$chngdisc)*-1+model_data$chngdisc
model_data$chnglist <- min(model_data$chnglist)*-1+model_data$chnglist
model_data <- log(model_data+0.01)
```

*

    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(list_mrp,discount,NPS))
```

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                    -1.359*** (0.365)      -1.375*** (0.326)
## deliverycdays            -0.014 (0.090)
## n_saledays                0.036 (0.061)
## chnglist                  0.150 (0.098)          0.141 (0.094)
## chngdisc                  0.075 (0.126)
## adTV                     -0.624 (0.391)         -0.621* (0.358)
## adSponsorship            1.032** (0.389)        0.943*** (0.307)
## adOnlineMarketing       1.947*** (0.451)        1.992*** (0.434)
## adSEM                   -0.986** (0.484)        -0.958** (0.404)
## adOther                   0.043 (0.028)          0.041 (0.025)
## Constant                -19.403** (8.773)      -18.722** (8.146)
## ------------------------------------------------------------------
## Observations                  53                      53
## R2                           0.824                   0.819
## Adjusted R2                  0.782                   0.791
## Residual Std. Error     0.953 (df = 42)         0.932 (df = 45)
## F Statistic         19.623*** (df = 10; 42) 29.170*** (df = 7; 45)
## ==================================================================
## 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
adOnlineMarketing	1.99240	0.43372	4.594	3.51e-05	***	21.784662
adOther	0.04125	0.02465	1.673	0.101197	NA	4.296257
adSEM	-0.95752	0.40388	-2.371	0.022092	*	3.584760
adSponsorship	0.94291	0.30713	3.070	0.003621	**	5.809810
adTV	-0.62102	0.35751	-1.737	0.089220	.	17.033153
chnglist	0.14079	0.09396	1.498	0.141019	NA	1.317435

var	Estimate	Std.Error	t-value	Pr(>|t|)	Significance	vif
week	-1.37462	0.32601	-4.216	0.000118	***	5.061968

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

Regularized Linear Model:

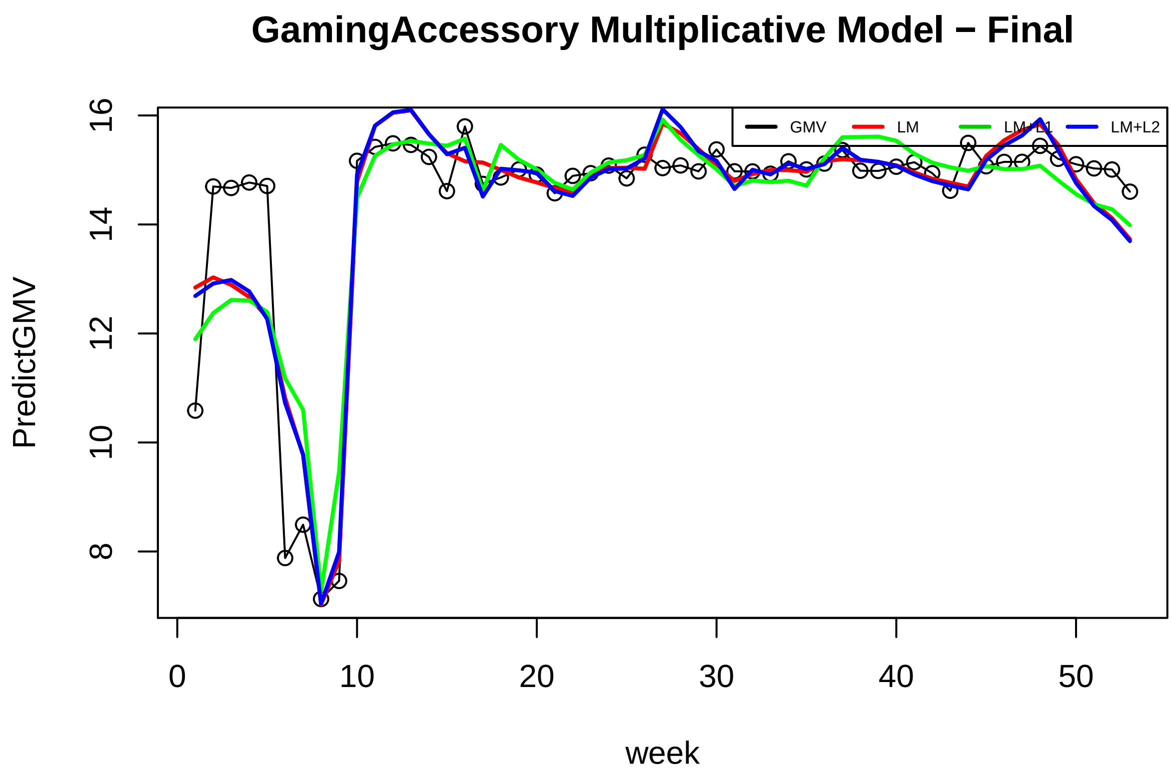
```
x = as.matrix(subset(model_data, select=-gmv))
y = as.vector(model_data$gmv)

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$gmv,main = 'GamingAccessory Multiplicative Model - Final',
     xlab='week',ylab='PredictGMV')
lines(model_data$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) -18.72175671 -5.18701183 -18.78910697
## 2  adOnlineMarketing   1.99239654  0.72962008   1.91345032
## 3            adOther   0.04125158  0.03165738   0.04242279
## 4              adSEM  -0.95752121 -0.32364516  -0.96558644
## 5      adSponsorship   0.94291221  0.65582589   1.01159268
## 6               adTV  -0.62101656  0.19948049  -0.58772491
## 7           chngdisc           NA  0.11157837   0.07573977
## 8           chnglist   0.14079337  0.23155029   0.15013964
## 9       deliverycdays           NA -0.01780922  -0.01096161
## 10         n_saledays           NA  0.01184979   0.03581902
## 11               week  -1.37462304 -0.63212794  -1.34940817
```

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

```
## [1] "Ridge regression R2 : 0.779464503193037"
```

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

```
## [1] "Lasso regression R2 : 0.823654548082647"
```

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

```
## [1] "Multiple R-squared:  0.8194,\tAdjusted R-squared:  0.7913 "
## [1] " Linear regression R2 : Multiple R-squared:  0.8194,\tAdjusted R-squared:  0.7913 "
```

*

Significant KPI

```
#coeff           lm          l1          l2
#1        (Intercept) -18.72175671 -5.18701183 -18.78910697
#2  adOnlineMarketing   1.99239654  0.72962008   1.91345032
#3            adOther   0.04125158  0.03165738   0.04242279
#4              adSEM  -0.95752121 -0.32364516  -0.96558644
#5      adSponsorship   0.94291221  0.65582589   1.01159268
#6               adTV  -0.62101656  0.19948049  -0.58772491
#7           chngdisc           NA  0.11157837   0.07573977
#8           chnglist   0.14079337  0.23155029   0.15013964
#9      deliverycdays           NA -0.01780922  -0.01096161
#10        n_saledays           NA  0.01184979   0.03581902
#11              week  -1.37462304 -0.63212794  -1.34940817

#[1] "Ridge regression R2 : 0.779464503193037"

#[1] "Lasso regression R2 : 0.823654548082647"

#[1] "Multiple R-squared:  0.8194,\tAdjusted R-squared:  0.7913 "

#[1] " Linear regression R2 : Multiple R-squared:  0.8194,\tAdjusted R-squared:  0.7913 "
```