model_GA_MM_ad.R

arman

Sun May 28 15:23:23 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(
```

```r
#   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'))

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

model_data <- subset(model_data,select = -c(TV))

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)


# # ***************************************************************************
# #                   TRAIN and TEST Data  ----
# # ***************************************************************************


test_data <- model_data[c(43:52),-2]
test_value <- model_data[c(43:52),2]

model_data <- model_data[-c(43:52),]
```

*

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,discount,NPS))
```

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                     -0.340 (0.370)
## deliverycdays             0.027 (0.094)
## n_saledays                0.020 (0.059)
## Sponsorship              0.435* (0.244)         0.452** (0.199)
## OnlineMarketing         1.820*** (0.371)        1.916*** (0.295)
## SEM                      -0.436 (0.394)          -0.487 (0.298)
## Other                     0.014 (0.018)
## chnglist                  0.147 (0.092)          0.167* (0.083)
## chngdisc                  0.063 (0.117)
## adTV                     -0.846 (0.514)        -1.074*** (0.299)
## Constant               -16.899* (9.598)       -18.979*** (5.489)
## -------------------------------------------------------------------
## Observations                   43                     43
## R2                            0.879                  0.872
## Adjusted R2                   0.841                  0.855
## Residual Std. Error      0.890 (df = 32)        0.852 (df = 37)
## F Statistic         23.247*** (df = 10; 32) 50.341*** (df = 5; 37)
## ===================================================================
## 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
adTV	-1.07426	0.29867	-3.597	0.000936	***	13.924288
chnglist	0.16689	0.08332	2.003	0.052544	.	1.233650
OnlineMarketing	1.91550	0.29455	6.503	1.31e-07	***	14.880952
SEM	-0.48678	0.29761	-1.636	0.110392	NA	2.375984
Sponsorship	0.45229	0.19947	2.267	0.029297	*	3.935708

```
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
```

Model Accuracy

```
ypred <- predict(step_mdl,new=test_data)
# MSE
mean((ypred-test_value)^2)
```

```
## [1] 0.6642739
```

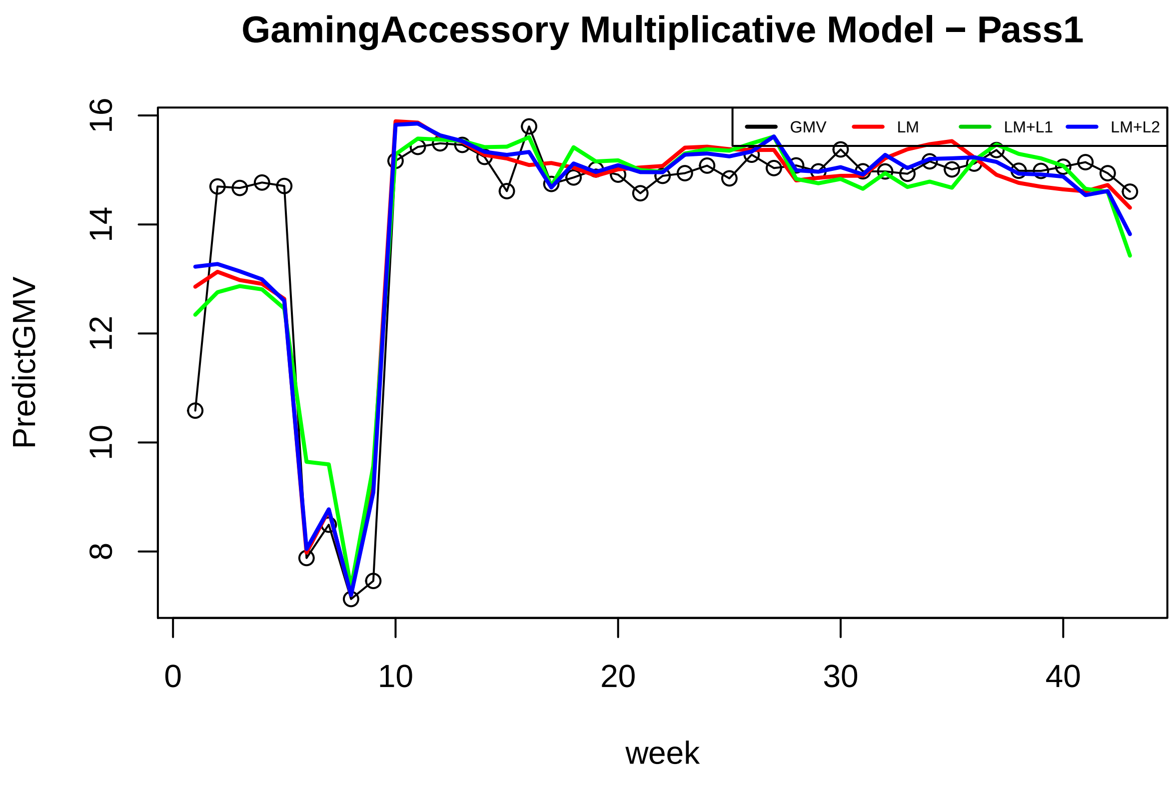
```
predR2 <- 1 - (sum((test_value-ypred )^2)/sum((test_value-mean(ypred))^2))
```

*

PLOTTING MODEL RESULTS

Plot Model prediction and base sales:

```
plot(model_data$gmv,main = 'GamingAccessory Multiplicative Model - Pass1',
     xlab='week',ylab='PredictGMV')
lines(model_data$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) -18.9794787 -1.295234207 -15.63668296
## 2             adTV  -1.0742630  0.205081816  -0.76688207
## 3         chngdisc          NA  0.093430529   0.06349169
## 4         chnglist   0.1668916  0.201700695   0.14693469
## 5    deliverycdays          NA -0.049605510   0.02227332
## 6       n_saledays          NA  0.024762623   0.01983318
## 7  OnlineMarketing   1.9155003  0.784589792   1.76404875
## 8            Other          NA  0.006456108   0.01247057
## 9              SEM  -0.4867797 -0.306071155  -0.43815629
## 10     Sponsorship   0.4522930  0.389085281   0.42252338
## 11            week          NA -0.408253155  -0.36457956
```

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

```
## [1] "Ridge regression R2 : 0.843537727320466"
```

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

```
## [1] "Lasso regression R2 : 0.87889409940987"
```

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

```
## [1] "Multiple R-squared:  0.8718,\tAdjusted R-squared:  0.8545 "
## [1] "Linear Mode      R2 : Multiple R-squared:  0.8718,\tAdjusted R-squared:  0.8545 "
```

```
print(paste0('Predicted        R2 : ',predR2))
```

```
## [1] "Predicted        R2 : -2.01561271519435"
```

*

Significant KPI