model_CA_LM_ad.R

atchirc

Sun May 28 00:50:43 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=='CameraAccessory',
                     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'))
# 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))

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


# # ***************************************************************************
# #                   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

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

```r
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.

```r
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

```r
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(adTV,discount,SEM,NPS,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                   -14,052.460 (50,757.680)
## deliverycdays         -92,746.160 (403,316.100)
## n_saledays            229,054.200 (196,409.800)
## Sponsorship           172,595.800** (74,233.790)    164,359.600*** (56,875.020)
## OnlineMarketing             0.053 (0.034)                 0.054*** (0.019)
## Other                       0.008 (0.020)
## chnglist                   0.0002 (0.0001)
## chngdisc               44,690.810 (34,609.960)
## Constant            2,193,304.000*** (772,549.800) 2,146,395.000*** (698,532.900)
## ---------------------------------------------------------------------------------
## Observations                      42                             42
## R2                              0.538                          0.461
## Adjusted R2                     0.426                          0.434
## Residual Std. Error    2,002,248.000 (df = 33)        1,988,886.000 (df = 39)
## F Statistic             4.802*** (df = 8; 33)          16.690*** (df = 2; 39)
## =================================================================================
## 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
OnlineMarketing	5.425e-02	1.907e-02	2.845	0.00703	**	1.346539
Sponsorship	1.644e+05	5.688e+04	2.890	0.00627	**	1.346539

```r
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] 1.840667e+12
```

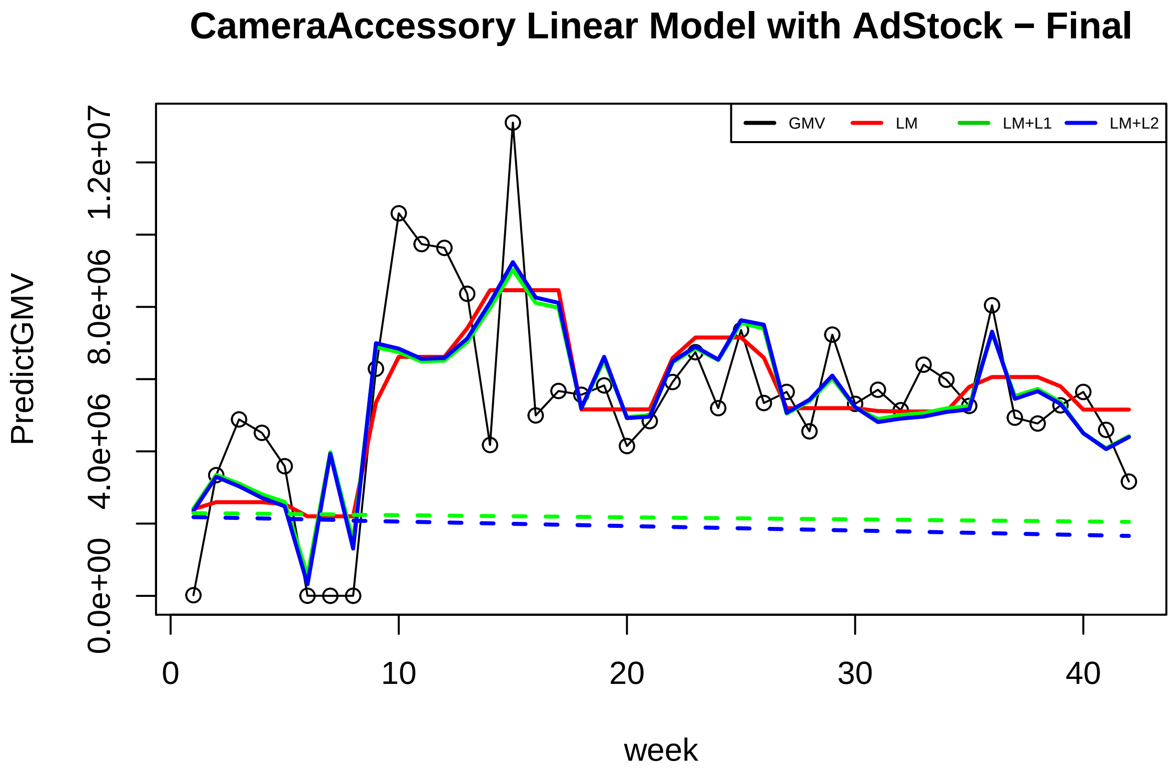
```
predR2 <- 1 - (sum((test_value-ypred )^2)/sum((test_value-mean(ypred))^2))
```

*

PLOTTING MODEL RESULTS

Plot Model prediction and base sales:

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

```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) 2.146395e+06  2.296604e+06  2.193153e+06
## 2        chngdisc           NA  4.186063e+04  4.420058e+04
## 3        chnglist           NA  1.635405e-04  1.731087e-04
## 4   deliverycdays           NA -1.387785e+05 -9.928543e+04
## 5      n_saledays           NA  2.115817e+05  2.267060e+05
## 6 OnlineMarketing 5.425179e-02  4.778630e-02  5.189148e-02
## 7           Other           NA  5.580684e-03  7.095053e-03
## 8     Sponsorship 1.643596e+05  1.657768e+05  1.726051e+05
## 9            week           NA -5.801283e+03 -1.241942e+04
```

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

```
## [1] "Ridge regression R2 : 0.536254118556601"
```

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

```
## [1] "Lasso regression R2 : 0.537901954753684"
```

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

```
## [1] "Multiple R-squared:  0.4612,\tAdjusted R-squared:  0.4336 "
## [1] "Linear Mode      R2 : Multiple R-squared:  0.4612,\tAdjusted R-squared:  0.4336 "
```

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

```
## [1] "Predicted        R2 : 0.353158570966236"
```

*

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) -9.985868e+05  7.343125e+06 -2.414127e+06
# 2  adOnlineMarketing  2.443484e-02  1.094554e-02  2.897134e-02
# 3            adOther            NA  5.072027e-03  1.158766e-02
# 4              adSEM -4.202136e-02 -2.446829e-02 -4.557119e-02
# 5      adSponsorship  1.905706e+05  1.241458e+05  2.222801e+05
# 6               adTV -4.296838e+05 -1.840207e+05 -5.721456e+05
# 7           chngdisc  4.180543e+04  4.653036e+04  4.519779e+04
# 8           chnglist            NA  4.633180e-05  5.844452e-05
# 9      deliverycdays            NA  9.445852e+04  1.027135e+05
# 10          discount            NA -7.942695e+03 -4.903361e+03
# 11          list_mrp  3.527259e-04  2.825139e-04  3.260626e-04
# 12        n_saledays  2.217532e+05  2.069190e+05  2.209012e+05
# 13               NPS            NA -1.258460e-02  3.593073e-03
# 14              week            NA -8.697558e+03 -2.151025e+04
# [1] "Ridge regression R2 : 0.614129507515657"
# [1] "Lasso regression R2 : 0.637433960090558"
# [1] "Multiple R-squared:  0.6227,\tAdjusted R-squared:  0.5626 "
# [1] "Linear Mode      R2 : Multiple R-squared:  0.6227,\tAdjusted R-squared:  0.5626 "
# >
```

```
# 1       (Intercept) 2.407765e+06  2.452181e+06  2.349655e+06
# 2 adOnlineMarketing 2.315286e-02  1.852802e-02  1.867951e-02
# 3           adOther           NA  3.585862e-03  4.526187e-03
# 4     adSponsorship 9.376834e+04  9.827333e+04  1.050286e+05
# 5          chngdisc 5.067210e+04  4.739354e+04  5.039845e+04
# 6          chnglist 2.277615e-04  2.036785e-04  2.190031e-04
# 7     deliverycdays           NA  2.434703e+04  4.042647e+04
# 8        n_saledays           NA  1.903471e+05  2.001274e+05
# 9              week           NA -3.103374e+02 -2.090038e+03
# [1] "Ridge regression R2 : 0.454425529891249"
# [1] "Lasso regression R2 : 0.45573613732237"
# [1] "Multiple R-squared:  0.4395,\tAdjusted R-squared:  0.3918 "
# [1] "Linear Mode      R2 :
#          Multiple R-squared:  0.4395,\tAdjusted R-squared:  0.3918 "
# >
```