model_HA_DLag_ad.R

arman

Sat May 27 13:17:05 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=='HomeAudio',
                     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(
```

```r
  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'))

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

# # . . . . Lag independant variables----
# # Lag weekly avg discount by 1 week
model_data$laggmv       <- data.table::shift(model_data$gmv)
model_data$lagdiscount  <- data.table::shift(model_data$discount)
model_data$lagdeliverycdays <- data.table::shift(model_data$deliverycdays)
model_data$lagTV        <- data.table::shift(model_data$adTV)
model_data$lagSponsorship <- data.table::shift(model_data$adSponsorship)
model_data$lagOnlineMar   <- data.table::shift(model_data$adOnlineMarketing)
model_data$lagSEM         <- data.table::shift(model_data$adSEM)
model_data$lagOther       <- data.table::shift(model_data$adOther)
model_data$lagNPS         <- data.table::shift(model_data$NPS)
model_data$laglist_mrp    <- data.table::shift(model_data$list_mrp)
model_data$lagChnglist    <- data.table::shift(model_data$chnglist)
model_data$lagChngdisc    <- data.table::shift(model_data$chngdisc)
```

*

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,laglist_mrp,
                                           adTV,lagTV,discount,NPS,lagNPS,
                                           lagdiscount,adOnlineMarketing,
                                           laggmv,deliverycdays,lagdeliverycdays,
                                           adSEM,lagChnglist))
```

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                   -75,696.720*** (26,882.860)   -75,576.160*** (22,079.820)
## n_saledays              336,682.600* (180,400.300)    290,411.700* (172,082.800)
## chnglist                 3,326.815** (1,583.111)       3,283.593** (1,539.487)
## chngdisc               217,219.500*** (51,306.430)   228,479.000*** (48,634.130)
## adSponsorship                 0.014* (0.008)               0.006* (0.003)
## adOther                       0.006 (0.019)
## lagSponsorship               -0.007 (0.010)
## lagOnlineMar                  0.017 (0.015)                0.019* (0.011)
## lagSEM                        0.0004 (0.019)
## lagOther                      0.003 (0.019)
## lagChngdisc              86,401.720* (48,137.400)      90,909.710* (46,736.960)
## Constant               4,825,223.000*** (871,313.700) 5,040,480.000*** (801,989.800)
## --------------------------------------------------------------------------------
## Observations                     49                            49
## R2                              0.645                         0.627
## Adjusted R2                     0.540                         0.563
## Residual Std. Error    1,822,386.000 (df = 37)       1,776,226.000 (df = 41)
## F Statistic             6.120*** (df = 11; 37)        9.830*** (df = 7; 41)
## ================================================================================
## 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
adSponsorship	6.071e-03	3.139e-03	1.934	0.06002	.	1.406512

var	Estimate	Std.Error	t-value	Pr(>|t|)	Significance	vif
chngdisc	2.285e+05	4.863e+04	4.698	2.95e-05	***	1.461986
chnglist	3.284e+03	1.539e+03	2.133	0.03897	*	1.155502
lagChngdisc	9.091e+04	4.674e+04	1.945	0.05864	.	1.350122
lagOnlineMar	1.894e-02	1.056e-02	1.795	0.08006	.	1.882258
n_saledays	2.904e+05	1.721e+05	1.688	0.09908	.	1.182246
week	-7.558e+04	2.208e+04	-3.423	0.00142	**	1.622559

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

Regularized Linear Model:

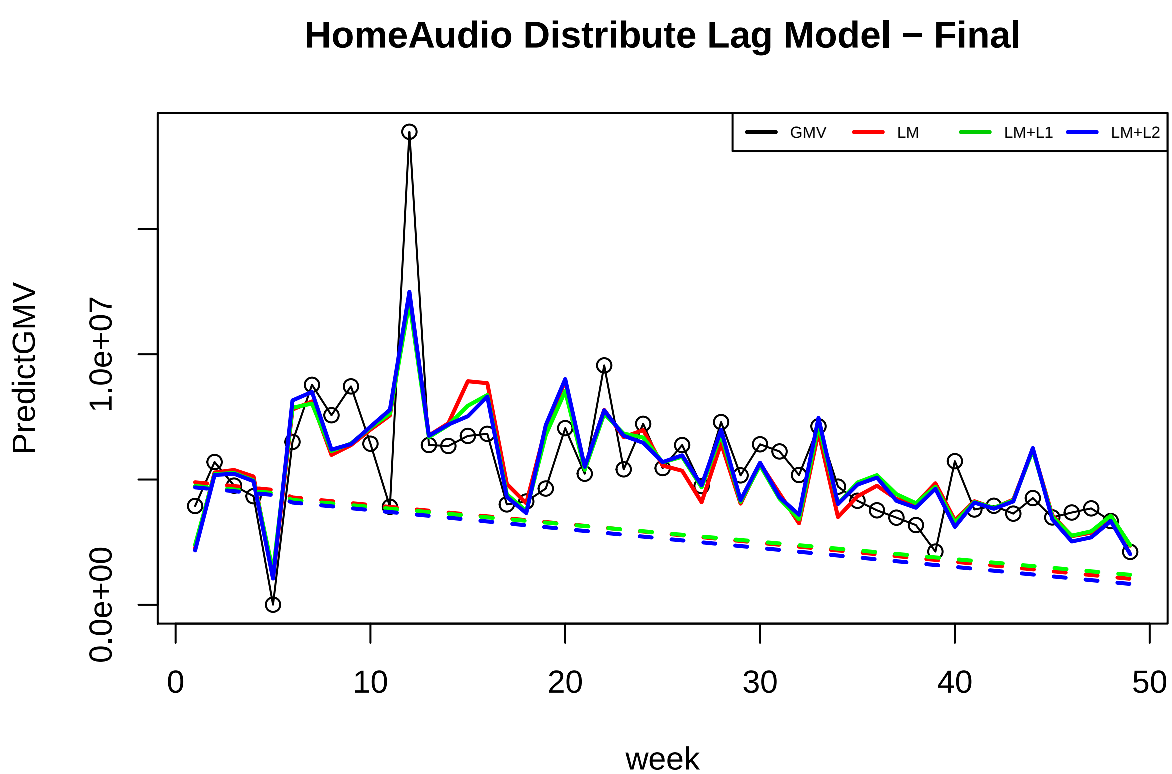
```
x = as.matrix(subset(model_data, select=-gmv))
y = as.vector(model_data$gmv)

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$gmv,main = 'HomeAudio Distribute Lag Model - Final',
     xlab='week',ylab='PredictGMV')
lines(model_data$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:**

```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)  5.040480e+06  4.880196e+06  4.837104e+06
## 2          adOther            NA  4.731087e-03  6.222712e-03
## 3    adSponsorship  6.071219e-03  9.983133e-03  1.370185e-02
## 4         chngdisc  2.284790e+05  2.040410e+05  2.170890e+05
## 5         chnglist  3.283593e+03  3.028442e+03  3.311031e+03
## 6      lagChngdisc  9.090971e+04  7.739293e+04  8.602225e+04
## 7     lagOnlineMar  1.894473e-02  1.367702e-02  1.723475e-02
## 8         lagOther            NA  4.273968e-03  3.014204e-03
## 9           lagSEM            NA -1.115231e-03  0.000000e+00
## 10 lagSponsorship            NA -2.417106e-03 -6.937415e-03
## 11      n_saledays  2.904117e+05  3.246721e+05  3.348200e+05
## 12            week -7.557616e+04 -6.954235e+04 -7.560628e+04
```

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

```
## [1] "Ridge regression R2 : 0.63883048037934"
```

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

```
## [1] "Lasso regression R2 : 0.645282465837913"
```

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

```
## [1] "Multiple R-squared:  0.6266,\tAdjusted R-squared:  0.5629 "
## [1] "Linear Mode      R2 : Multiple R-squared:  0.6266,\tAdjusted R-squared:  0.5629 "
```

*

Significant KPI

Lasso(LM+L2) regression results a simple explainable model with significant KPIs as `Discount Inflation`, `Deliverycday`, `sale days`, `Sponsorship` `week`,`discount`,

```
# Model Optimization

#coeff             lm            l1            l2
#1     (Intercept)  5.040480e+06  4.880196e+06  4.837104e+06
#2         adOther            NA  4.731087e-03  6.222712e-03
#3   adSponsorship  6.071219e-03  9.983133e-03  1.370185e-02
#4        chngdisc  2.284790e+05  2.040410e+05  2.170890e+05
#5        chnglist  3.283593e+03  3.028442e+03  3.311031e+03
#6     lagChngdisc  9.090971e+04  7.739293e+04  8.602225e+04
#7    lagOnlineMar  1.894473e-02  1.367702e-02  1.723475e-02
#8        lagOther            NA  4.273968e-03  3.014204e-03
#9          lagSEM            NA -1.115231e-03  0.000000e+00
#10 lagSponsorship            NA -2.417106e-03 -6.937415e-03
#11     n_saledays  2.904117e+05  3.246721e+05  3.348200e+05
#12           week -7.557616e+04 -6.954235e+04 -7.560628e+04


#[1] "Ridge regression R2 : 0.63883048037934"

#[1] "Lasso regression R2 : 0.645282465837913"

#[1] "Multiple R-squared:  0.6266,\tAdjusted R-squared:  0.5629 "
#[1] "Linear Mode      R2 : Multiple R-squared:  0.6266,\tAdjusted R-squared:  0.5629 "
```