model_HA_DLag_ad.R

anandrathi

Sun May 28 16:48:59 2017

```
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$adSponsorship      <- as.numeric(
#   stats::filter(model_data$Sponsorship,filter=0.5,method='recursive'))
```

```r
# 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))


# # . . . . 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$lagadTV            <- data.table::shift(model_data$adTV)
model_data$lagSponsorship     <- data.table::shift(model_data$Sponsorship)
model_data$lagOnlineMar       <- data.table::shift(model_data$OnlineMarketing)
model_data$lagSEM             <- data.table::shift(model_data$SEM)
model_data$lagOther           <- data.table::shift(model_data$Other)
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)



# # ***************************************************************************
# #                    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)
```

```
  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(lagadTV,lagSEM,discount,lagdiscount,
                                          list_mrp,laglist_mrp,NPS,lagNPS,adTV,SEM))
```

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                    -76,239.180 (59,756.210)      -68,190.360** (31,545.310)
## deliverycdays          -178,368.700 (850,219.200)
## n_saledays             354,709.700* (203,486.100)      357,704.000* (179,460.100)
## Sponsorship                 0.023* (0.013)                 0.013** (0.006)
## OnlineMarketing             -0.142 (0.093)                 -0.099 (0.066)
## Other                       0.025 (0.031)
## chnglist                  2,779.762 (1,960.397)         2,839.372* (1,666.114)
## chngdisc               248,554.400*** (68,148.170)     252,745.200*** (54,292.620)
## laggmv                      -0.126 (0.173)
## lagdeliverycdays        144,277.500 (810,232.100)
## lagSponsorship              -0.007 (0.014)
## lagOnlineMar                0.179* (0.095)                 0.134** (0.058)
## lagOther                    -0.011 (0.030)
## lagChnglist                566.729 (2,008.918)
## lagChngdisc            109,164.200* (60,412.770)       87,367.660* (48,780.580)
## Constant             5,436,761.000*** (1,182,538.000) 4,885,550.000*** (880,355.200)
## -------------------------------------------------------------------------------------
## Observations                      41                                41
## R2                               0.685                             0.662
## Adjusted R2                      0.496                             0.577
## Residual Std. Error     1,975,005.000 (df = 25)         1,808,248.000 (df = 32)
## F Statistic              3.620*** (df = 15; 25)          7.825*** (df = 8; 32)
## =====================================================================================
## 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	2.527e+05	5.429e+04	4.655	5.40e-05	***	1.735407
chnglist	2.839e+03	1.666e+03	1.704	0.0980	.	1.209796
lagChngdisc	8.737e+04	4.878e+04	1.791	0.0828	.	1.400917
lagOnlineMar	1.343e-01	5.768e-02	2.329	0.0263	*	11.849165
n_saledays	3.577e+05	1.795e+05	1.993	0.0548	.	1.146401
OnlineMarketing	-9.889e-02	6.597e-02	-1.499	0.1436	NA	13.772419
Sponsorship	1.348e-02	6.348e-03	2.124	0.0415	*	1.849048
week	-6.819e+04	3.155e+04	-2.162	0.0382	*	1.923257

```
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
```

```
## Warning: from glmnet Fortran code (error code -83); Convergence for 83th
## lambda value not reached after maxit=100000 iterations; solutions for
## larger lambdas returned
```

```
      Model Accuracy
```

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

```
## [1] NA
```

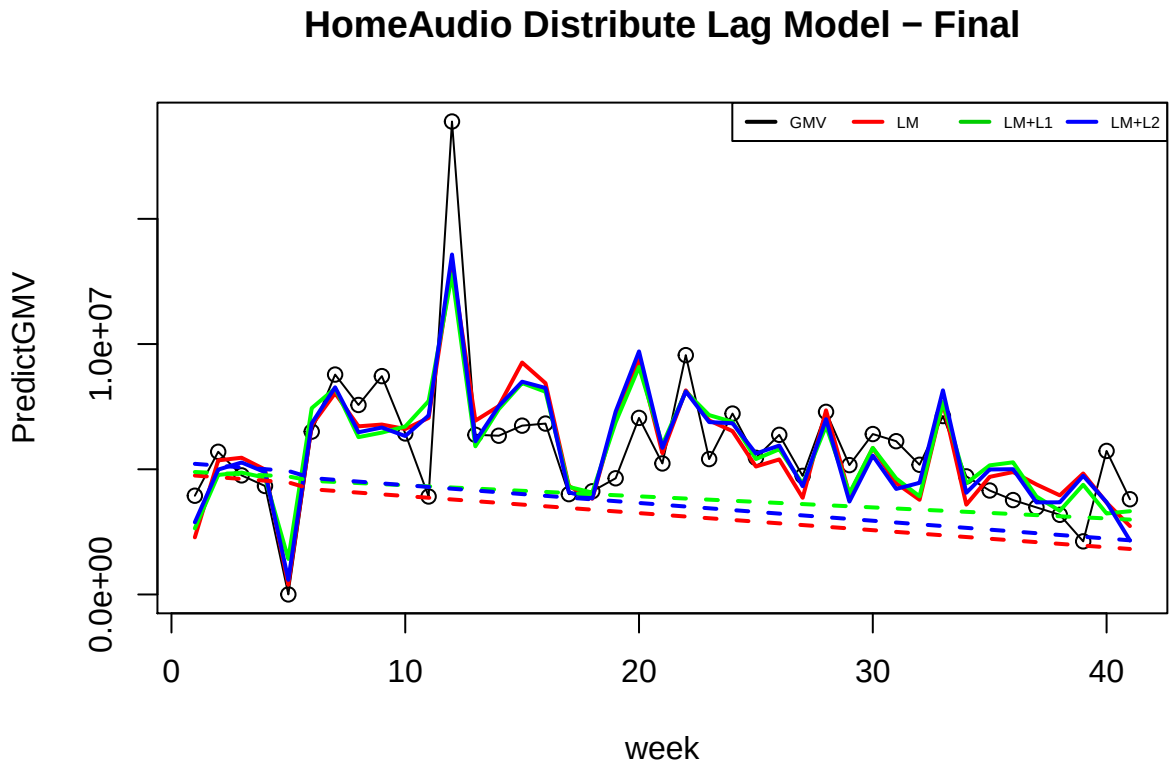
```
predR2 <- 1 - (sum((test_value-ypred )^2)/sum((test_value-mean(ypred))^2))
```

*

PLOTTING MODEL RESULTS

Plot Model prediction and base sales:

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

```
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)  4.885550e+06  4.979621e+06  5.359397e+06
## 2          chngdisc  2.527452e+05  1.911256e+05  2.416322e+05
## 3          chnglist  2.839372e+03  2.948983e+03  2.821604e+03
## 4     deliverycdays            NA -2.687303e+05 -1.699742e+05
## 5       lagChngdisc  8.736766e+04  8.407044e+04  1.059875e+05
## 6       lagChnglist            NA  3.610856e+02  5.170262e+02
## 7  lagdeliverycdays            NA -4.576647e+04  8.559092e+04
## 8            laggmv            NA -1.189799e-01 -1.231330e-01
## 9      lagOnlineMar  1.343333e-01  4.548139e-02  1.562394e-01
## 10         lagOther            NA  1.972110e-03 -8.109784e-03
## 11   lagSponsorship            NA  6.104599e-03 -3.971170e-03
## 12       n_saledays  3.577040e+05  3.411219e+05  3.523769e+05
## 13  OnlineMarketing -9.889501e-02 -1.363760e-02 -1.199410e-01
## 14            Other            NA  4.773199e-03  2.070881e-02
## 15      Sponsorship  1.348405e-02  9.244578e-03  2.013758e-02
## 16             week -6.819036e+04 -4.418377e+04 -7.103220e+04
```

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

```
## [1] "Ridge regression R2 : 0.654333889571837"
```

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

```
## [1] "Lasso regression R2 : 0.683783207309693"
```

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

```
## [1] "Multiple R-squared:  0.6617,\tAdjusted R-squared:  0.5772 "
## [1] "Linear Mode      R2 : Multiple R-squared:  0.6617,\tAdjusted R-squared:  0.5772 "
```

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

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

*

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) -5.969071e+06  4.579498e+06  9.150469e+05
# 2          chngdisc            NA  2.403055e+04  1.404392e+04
# 3          chnglist            NA  1.147520e-04  7.288558e-05
# 4     deliverycdays            NA  4.048955e+04  0.000000e+00
# 5          discount  1.236327e+05  4.963671e+04  7.598971e+04
# 6       lagChngdisc            NA  6.245309e+01  0.000000e+00
# 7       lagChnglist  2.293409e-04  2.098331e-04  2.287341e-04
# 8  lagdeliverycdays            NA -3.025699e+03  0.000000e+00
# 9       lagdiscount            NA -1.335381e+04  0.000000e+00
# 10           laggmv            NA -2.620557e-03  0.000000e+00
# 11      laglist_mrp            NA  1.674141e-05  0.000000e+00
# 12           lagNPS            NA -5.625944e-04  0.000000e+00
# 13     lagOnlineMar  4.084108e-02  1.262188e-02  6.688259e-03
# 14         lagOther            NA  3.739609e-03  5.058915e-03
# 15           lagSEM            NA -8.660429e-03  0.000000e+00
# 16   lagSponsorship            NA  7.039559e+04  4.807799e+04
# 17            lagTV            NA -1.925906e+05 -2.177454e+05
# 18         list_mrp  2.884431e-04  1.373816e-04  1.833042e-04
# 19       n_saledays  2.427904e+05  1.568555e+05  1.714789e+05
# 20              NPS            NA -8.196588e-03 -6.228913e-03
# 21  OnlineMarketing            NA  1.710518e-02  2.425264e-02
# 22            Other  2.019568e-02  1.489752e-03  0.000000e+00
# 23              SEM -5.086349e-02 -1.401646e-02 -3.032600e-02
# 24      Sponsorship  3.140919e+05  1.003712e+05  1.560263e+05
# 25               TV -8.880872e+05 -1.956665e+04  0.000000e+00
# 26             week            NA -4.224726e+03  0.000000e+00
# [1] "Ridge regression R2 : 0.6325014178026 71"
# [1] "Lasso regression R2 : 0.645565137277216"
# [1] "Multiple R-squared:  0.6579,\tAdjusted R-squared:  0.5828 "
# [1] "Linear Mode      R2 :
#         Multiple R-squared:  0.6579,\tAdjusted R-squared:  0.5828 "
```

```
# coeff            lm            l1            l2
# 1       (Intercept) 2.554898e+06  4.913420e+06  5.846966e+06
# 2          chngdisc 7.649292e+04  5.274919e+04  7.169863e+04
# 3          chnglist 2.518427e-04  1.160990e-04  1.331778e-04
# 4     deliverycdays            NA  5.896428e+04  4.689963e+04
# 5       lagChngdisc            NA  3.785165e+03  1.515076e+04
# 6       lagChnglist 3.547880e-04  1.967177e-04  2.573714e-04
# 7  lagdeliverycdays            NA  2.027915e+04  0.000000e+00
# 8            laggmv            NA  1.028940e-02  0.000000e+00
# 9       laglist_mrp            NA  1.873679e-05  0.000000e+00
# 10           lagNPS            NA  1.161537e-03  0.000000e+00
```

```
# 11     lagOnlineMar 5.080372e-02  1.027355e-02  1.248910e-02
# 12         lagOther           NA  2.953227e-03  0.000000e+00
# 13   lagSponsorship           NA  4.405822e+04  2.549686e+04
# 14         list_mrp           NA  1.399100e-04  1.249556e-04
# 15       n_saledays           NA  1.579620e+05  1.577822e+05
# 16              NPS           NA -7.935344e-03 -7.801763e-03
# 17  OnlineMarketing           NA  1.516293e-02  1.328937e-02
# 18            Other           NA  1.958178e-03  2.287910e-03
# 19      Sponsorship 1.430303e+05  8.229495e+04  1.044978e+05
# 20             week           NA -1.254369e+03  0.000000e+00
# [1] "Ridge regression R2 : 0.601140151803534"
# [1] "Lasso regression R2 : 0.611020467029655"
# [1] "Multiple R-squared:  0.5834,\tAdjusted R-squared:  0.5371 "
# [1] "Linear Mode      R2 :
#           Multiple R-squared:  0.5834,\tAdjusted R-squared:  0.5371 "
```