model_CA_DLag_ad.R

atchirc

Sun May 28 00:56:48 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=='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'))

# 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                  -17,591.760 (58,196.290)
## deliverycdays         173,662.600 (765,271.200)
## n_saledays            158,755.500 (194,012.200)
## Sponsorship           131,335.800 (136,448.800)    143,484.500*** (47,061.860)
## OnlineMarketing            -0.010 (0.095)
## Other                       0.011 (0.030)
## chnglist                  0.0003** (0.0001)             0.0002** (0.0001)
## chngdisc             116,106.100** (54,745.050)     76,808.200** (30,485.390)
## laggmv                      0.034 (0.184)
## lagdeliverycdays     -380,452.800 (762,903.700)
## lagSponsorship        24,439.470 (133,201.500)
## lagOnlineMar                0.054 (0.098)                0.050*** (0.015)
## lagOther                   -0.003 (0.029)
## lagChnglist              0.0004*** (0.0002)             0.0004*** (0.0001)
## lagChngdisc             45,784.330 (56,122.210)
## Constant             2,684,607.000** (970,482.200) 2,512,585.000*** (618,747.200)
## --------------------------------------------------------------------------------
## Observations                     41                             41
## R2                              0.650                         0.603
## Adjusted R2                     0.440                         0.546
## Residual Std. Error     1,894,530.000 (df = 25)       1,706,209.000 (df = 35)
## F Statistic             3.097*** (df = 15; 25)         10.619*** (df = 5; 35)
## ================================================================================
## 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	7.681e+04	3.049e+04	2.520	0.016470	*	1.109500
chnglist	2.443e-04	1.068e-04	2.288	0.028300	*	1.062137
lagChnglist	3.680e-04	1.100e-04	3.346	0.001966	**	1.123145
lagOnlineMar	5.048e-02	1.547e-02	3.263	0.002467	**	1.204891
Sponsorship	1.435e+05	4.706e+04	3.049	0.004356	**	1.221367

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

Regularized Linear Model:

```r
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 -89); Convergence for 89th
## lambda value not reached after maxit=100000 iterations; solutions for
## larger lambdas returned
```

```
## Warning: from glmnet Fortran code (error code -95); Convergence for 95th
## lambda value not reached after maxit=100000 iterations; solutions for
## larger lambdas returned
```

```
      Model Accuracy
```

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

```
## [1] 1.574904e+12
```

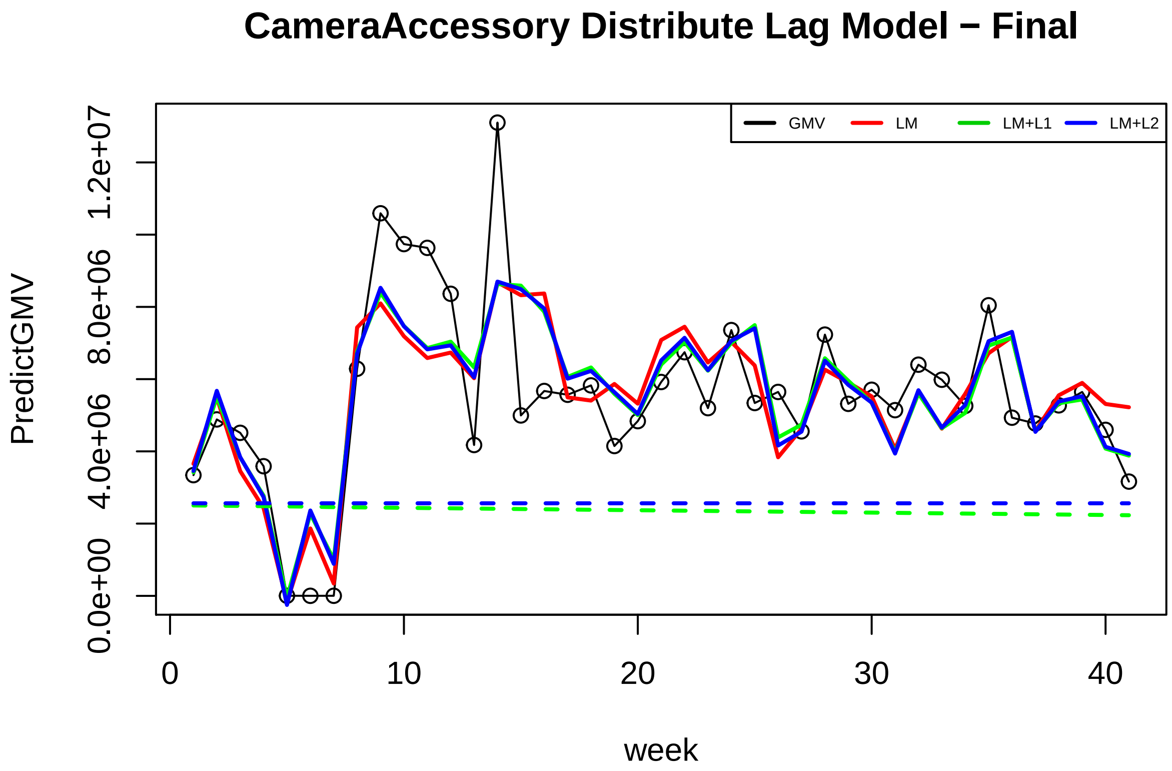
```r
predR2 <- 1 - (sum((test_value-ypred )^2)/sum((test_value-mean(ypred))^2))
```

*

PLOTTING MODEL RESULTS

Plot Model prediction and base sales:

```
plot(model_data$gmv,main = 'CameraAccessory Distribute Lag Model - Final',
     xlab='week',ylab='PredictGMV')
lines(model_data$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) 2.512585e+06  2.515590e+06  2.562214e+06
## 2          chngdisc 7.680819e+04  8.280621e+04  9.293370e+04
## 3          chnglist 2.443237e-04  2.548956e-04  2.718313e-04
## 4     deliverycdays           NA  6.381701e+03  0.000000e+00
## 5       lagChngdisc           NA  1.922148e+04  2.380142e+04
## 6       lagChnglist 3.679852e-04  3.441138e-04  3.773846e-04
## 7  lagdeliverycdays           NA -2.386630e+05 -2.658491e+05
## 8            laggmv           NA  7.612387e-02  5.445028e-02
## 9      lagOnlineMar 5.048111e-02  2.045900e-02  3.675312e-02
## 10         lagOther           NA  2.946872e-03  0.000000e+00
## 11   lagSponsorship           NA  5.517991e+04  4.423518e+04
## 12       n_saledays           NA  1.561630e+05  1.421868e+05
## 13  OnlineMarketing           NA  1.825763e-02  0.000000e+00
## 14            Other           NA  3.165986e-03  3.890816e-03
## 15      Sponsorship 1.434845e+05  9.109450e+04  1.095198e+05
## 16             week           NA -6.589054e+03  0.000000e+00
```

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

```
## [1] "Ridge regression R2 : 0.641505670220015"
```

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

```
## [1] "Lasso regression R2 : 0.644877721747646"
```

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

```
## [1] "Multiple R-squared:  0.6027,\tAdjusted R-squared:  0.5459 "
## [1] "Linear Mode      R2 : Multiple R-squared:  0.6027,\tAdjusted R-squared:  0.5459 "
```

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

```
## [1] "Predicted        R2 : 0.390345215603898"
```

*

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 : 0.632501417802671"
# [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 "
```