model_GA_DLag_ad.R

arman

Sat May 27 13:37:01 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$adOnlineMarketing  <- as.numeric(
  stats::filter(model_data$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)
```

```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

```r
# Prune KPI as part of model optimization
model_data <- na.omit(model_data)
model_data <- subset(model_data,select=-c(lagdiscount,laggmv,lagTV,NPS,lagNPS,
                                          laglist_mrp,lagSEM,discount,
                                          lagSponsorship))
```

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                     -4,791.077 (34,917.040)
## list_mrp                   544.761 (1,850.700)
## deliverycdays           -23,870.940 (396,320.700)
## n_saledays              96,613.750 (100,656.500)
## chnglist                  1,082.816 (1,500.666)           1,440.178* (776.648)
## chngdisc                47,279.040** (17,422.210)     49,598.940*** (15,648.050)
## adTV                   -314,859.200 (267,239.500)     -341,898.000 (203,990.600)
## adSponsorship              0.010** (0.004)                 0.010*** (0.003)
## adOnlineMarketing           0.024 (0.024)                  0.017*** (0.006)
## adSEM                      -0.019* (0.011)                 -0.020** (0.009)
## adOther                     0.005 (0.011)
## lagdeliverycdays        82,727.600 (431,906.400)
## lagOnlineMar               -0.008 (0.025)
## lagOther                    0.006 (0.011)                    0.009 (0.006)
## lagChnglist               1,107.281 (1,033.729)          1,302.734 (795.678)
## lagChngdisc              21,323.940 (17,365.940)       22,884.230 (15,785.610)
## Constant               1,247,889.000 (1,715,755.000) 1,755,445.000*** (318,961.600)
## ----------------------------------------------------------------------------------
## Observations                       52                              52
## R2                                0.621                           0.605
## Adjusted R2                       0.448                           0.521
## Residual Std. Error      1,012,288.000 (df = 35)        943,312.000 (df = 42)
## F Statistic              3.585*** (df = 16; 35)         7.151*** (df = 9; 42)
## ==================================================================================
## 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
adOnlineMarketing	1.689e-02	5.923e-03	2.852	0.00672	**	2.626946
adSEM	-1.951e-02	8.802e-03	-2.216	0.03215	*	3.933446
adSponsorship	9.913e-03	3.174e-03	3.123	0.00324	**	5.720724
adTV	-3.419e+05	2.040e+05	-1.676	0.10116	NA	2.837744
chngdisc	4.960e+04	1.565e+04	3.170	0.00285	**	1.377963
chnglist	1.440e+03	7.766e+02	1.854	0.07072	.	1.477880
lagChngdisc	2.288e+04	1.579e+04	1.450	0.15457	NA	1.400041
lagChnglist	1.303e+03	7.957e+02	1.637	0.10905	NA	1.551150
lagOther	8.787e-03	6.227e-03	1.411	0.16558	NA	2.126652

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

Regularized Linear Model:

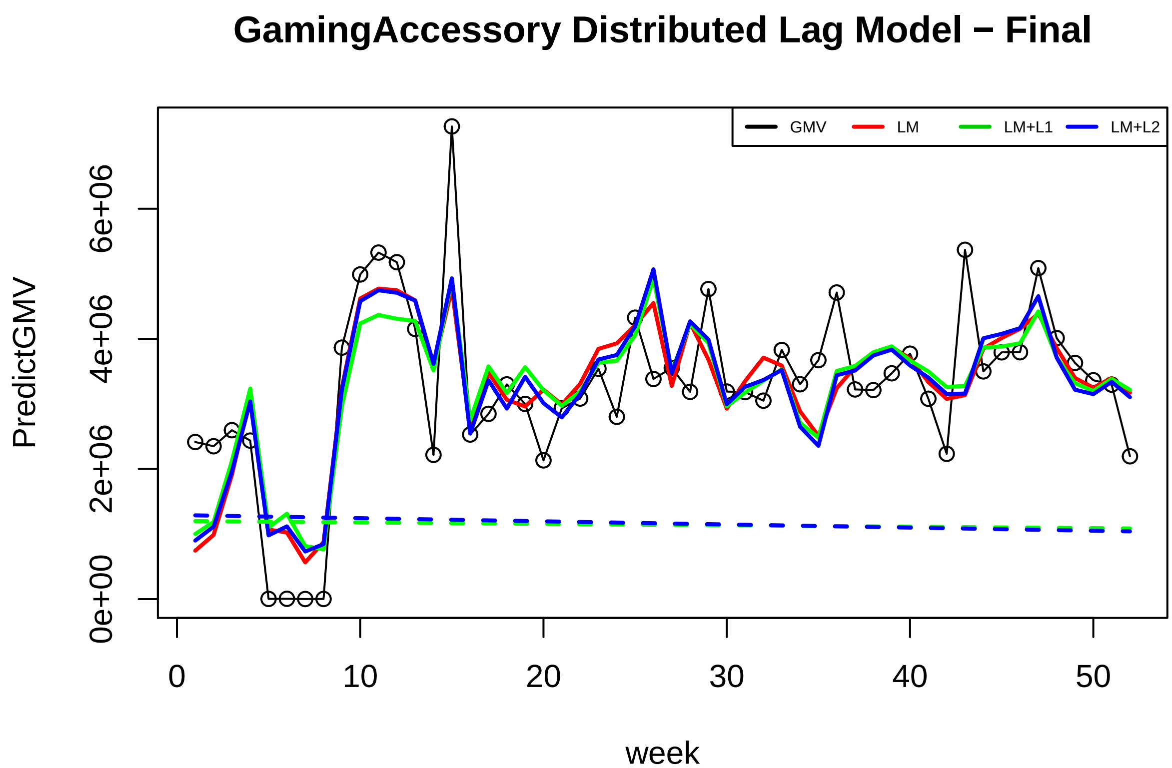
```r
x = as.matrix(subset(model_data, select=-gmv))
y = as.vector(model_data$gmv)

ridge_out <- atcLmReg(x,y,0,3)  # x, y, alpha, nfolds
lasso_out <- atcLmReg(x,y,1,3)  # x, y, alpha, nfolds
```

*

PLOTTING MODEL RESULTS

Plot Model prediction and base sales:

```
plot(model_data$gmv,main = 'GamingAccessory Distributed Lag Model - Final',
     xlab='week',ylab='PredictGMV')
lines(model_data$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)  1.755445e+06  1.202900e+06  1.296864e+06
## 2  adOnlineMarketing  1.689058e-02  1.292184e-02  2.242359e-02
## 3            adOther            NA  4.540615e-03  5.007159e-03
## 4              adSEM -1.950668e-02 -1.156897e-02 -1.930645e-02
## 5      adSponsorship  9.913086e-03  7.044008e-03  9.843532e-03
## 6               adTV -3.418980e+05 -1.844815e+05 -3.142599e+05
## 7           chngdisc  4.959894e+04  4.664159e+04  4.725652e+04
## 8           chnglist  1.440178e+03  1.128130e+03  1.092644e+03
## 9      deliverycdays            NA -2.952921e+03  0.000000e+00
## 10       lagChngdisc  2.288423e+04  2.176313e+04  2.138957e+04
## 11       lagChnglist  1.302734e+03  1.254372e+03  1.134502e+03
## 12  lagdeliverycdays            NA  5.154316e+04  5.768857e+04
## 13      lagOnlineMar            NA  1.200895e-03 -6.133268e-03
## 14          lagOther  8.786703e-03  3.271855e-03  5.483514e-03
## 15          list_mrp            NA  6.850935e+02  4.962336e+02
## 16         n_saledays            NA  8.417336e+04  9.471128e+04
## 17              week            NA -2.200588e+03 -4.824647e+03
```

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

```
## [1] "Ridge regression R2 : 0.605301169149325"
```

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

```
## [1] "Lasso regression R2 : 0.620900439092102"
```

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

```
## [1] "Multiple R-squared:  0.6051,\tAdjusted R-squared:  0.5205 "
## [1] "Linear Mode      R2 : Multiple R-squared:  0.6051,\tAdjusted R-squared:  0.5205 "
```

*

Significant KPI

```
#coeff             lm            l1            l2
#1        (Intercept)  1.755445e+06  1.202900e+06  1.293840e+06
#2  adOnlineMarketing  1.689058e-02  1.292184e-02  1.564770e-02
#3            adOther            NA  4.540615e-03  5.142134e-03
#4              adSEM -1.950668e-02 -1.156897e-02 -1.816109e-02
#5      adSponsorship  9.913086e-03  7.044008e-03  9.553842e-03
#6               adTV -3.418980e+05 -1.844815e+05 -2.988337e+05
#7           chngdisc  4.959894e+04  4.664159e+04  4.709677e+04
#8           chnglist  1.440178e+03  1.128130e+03  1.090774e+03
#9      deliverycdays            NA -2.952921e+03  0.000000e+00
#10       lagChngdisc  2.288423e+04  2.176313e+04  2.133594e+04
#11       lagChnglist  1.302734e+03  1.254372e+03  1.156870e+03
#12  lagdeliverycdays            NA  5.154316e+04  2.867923e+04
#13      lagOnlineMar            NA  1.200895e-03 -9.161772e-05
#14          lagOther  8.786703e-03  3.271855e-03  4.269528e-03
#15          list_mrp            NA  6.850935e+02  5.137078e+02
#16        n_saledays            NA  8.417336e+04  8.566850e+04
#17              week            NA -2.200588e+03 -1.920790e+03

#[1] "Ridge regression R2 : 0.605301169149325"

#[1] "Lasso regression R2 : 0.617712166177703"

#[1] "Multiple R-squared:  0.6051,\tAdjusted R-squared:  0.5205 "
#[1] "Linear Mode      R2 : Multiple R-squared:  0.6051,\tAdjusted R-squared:  0.5205 "
```