model_CA_MM_ad.R

atchirc

Sun May 28 01:01:52 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(
```

```r
#   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))


model_data$chngdisc <- min(model_data$chngdisc)*-1+model_data$chngdisc
model_data$chnglist <- min(model_data$chnglist)*-1+model_data$chnglist
model_data <- log(model_data+0.01)


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

```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(list_mrp,discount,NPS))
```

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                   -1.001** (0.468)       -0.950** (0.438)
## deliverycdays           -0.160 (0.110)        -0.174* (0.101)
## n_saledays              0.027 (0.080)
## chnglist               0.143*** (0.048)       0.148*** (0.046)
## chngdisc                0.318* (0.176)         0.322* (0.174)
## adTV                   1.424*** (0.300)       1.391*** (0.280)
## Constant              12.743*** (2.513)      12.374*** (2.236)
## -----------------------------------------------------------------
## Observations                  42                     42
## R2                          0.774                  0.773
## Adjusted R2                 0.735                  0.742
## Residual Std. Error    1.169 (df = 35)        1.154 (df = 36)
## F Statistic         19.982*** (df = 6; 35) 24.560*** (df = 5; 36)
## =================================================================
## 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
adTV	1.39112	0.28029	4.963	1.68e-05	***	5.684644
chngdisc	0.32187	0.17371	1.853	0.07211	.	1.550127
chnglist	0.14750	0.04636	3.182	0.00301	**	1.198377
deliverycdays	-0.17372	0.10063	-1.726	0.09288	.	1.054744
week	-0.95028	0.43775	-2.171	0.03662	*	4.567297

```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
```

Model Accuracy

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

```
## [1] 6.726945
```

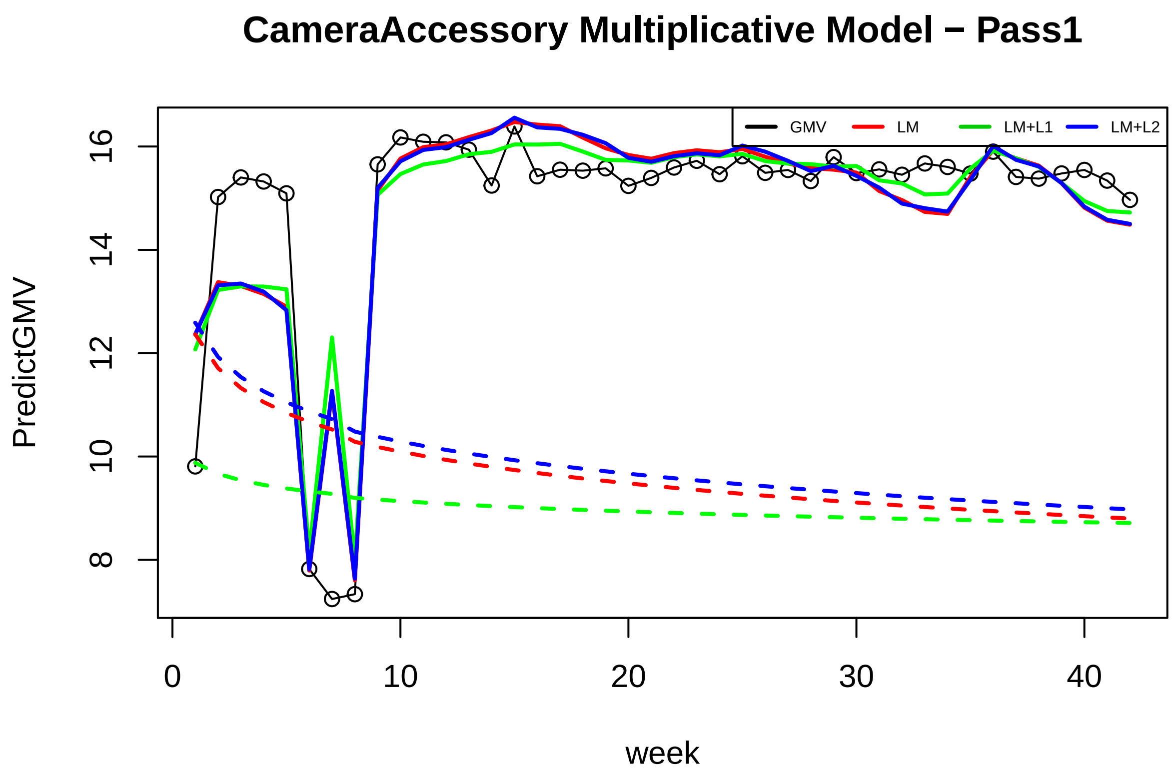
```r
predR2 <- 1 - (sum((test_value-ypred )^2)/sum((test_value-mean(ypred))^2))
```

*

PLOTTING MODEL RESULTS

Plot Model prediction and base sales:

```
plot(model_data$gmv,main = 'CameraAccessory Multiplicative Model - Pass1',
     xlab='week',ylab='PredictGMV')
lines(model_data$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) 12.3738624  9.881922565 12.59962384
## 2          adTV  1.3911160  0.952877418  1.40041619
## 3      chngdisc  0.3218687  0.436468211  0.32202181
## 4      chnglist  0.1475024  0.159638836  0.14396627
## 5 deliverycdays -0.1737175 -0.175403795 -0.16033510
## 6    n_saledays         NA -0.008943843  0.02366783
## 7          week -0.9502828 -0.310606038 -0.96311336
```

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

```
## [1] "Ridge regression R2 : 0.756563795634667"
```

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

```
## [1] "Lasso regression R2 : 0.773986724629475"
```

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

```
## [1] "Multiple R-squared:  0.7733,\tAdjusted R-squared:  0.7418 "
## [1] "Linear Mode      R2 : Multiple R-squared:  0.7733,\tAdjusted R-squared:  0.7418 "
```

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

```
## [1] "Predicted        R2 : -0.0255529708802074"
```

*

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) -291.1095142 -1.156524e+02 -3.247125e+02
# 2         chngdisc    0.2976528  4.529978e-01  3.005713e-01
# 3         chnglist           NA  4.156542e-02 -1.160978e-02
# 4    deliverycdays           NA  5.492330e-02  3.127685e-02
# 5         discount           NA -1.493296e+00 -1.307761e-02
# 6         list_mrp    3.4394110  2.980721e+00  3.721846e+00
# 7       n_saledays           NA  2.042750e-02  6.727983e-03
# 8              NPS   10.0904759  2.763048e+00  1.133940e+01
# 9  OnlineMarketing    1.3481222  4.963282e-01  1.269501e+00
# 10           Other           NA  8.074312e-03  1.067699e-02
# 11             SEM           NA  5.195209e-02  2.492818e-01
# 12     Sponsorship    0.2671538  2.217135e-01  1.930656e-01
# 13              TV   -0.2953724  1.322017e-01 -1.901196e-01
# 14            week           NA  6.922926e-02 -4.158001e-02
# [1] "Ridge regression R2 : 0.907781713555186"
# [1] "Lasso regression R2 : 0.92785868141555"
# [1] "Multiple R-squared:  0.9245,\tAdjusted R-squared:  0.9145 "
# [1] "Linear Mode      R2 :
#       Multiple R-squared:  0.9245,\tAdjusted R-squared:  0.9145 "
```

```
# coeff              lm            l1            l2
# 1      (Intercept) -252.44951515 -2.209457934 -2.079449e+02
# 2         chngdisc    0.26041855  0.335460607  2.106677e-01
# 3         chnglist    0.07072628  0.090937642  5.821151e-02
# 4    deliverycdays   -0.13581758 -0.029507054 -1.140778e-01
# 5       n_saledays           NA  0.021263335  0.000000e+00
# 6              NPS   11.53023534  0.248190242  9.596589e+00
# 7  OnlineMarketing    1.91367470  0.701215112  1.955293e+00
# 8            Other           NA -0.001003973 -2.923578e-03
# 9              SEM           NA -0.204127587 -3.235943e-01
# 10     Sponsorship    0.52639403  0.385357456  5.747898e-01
# 11              TV   -0.65321728  0.105757500 -6.125896e-01
# 12            week           NA -0.035989462 -2.334996e-01
# [1] "Ridge regression R2 : 0.802021944242947"
# [1] "Lasso regression R2 : 0.846611423521393"
# [1] "Multiple R-squared:  0.8419,\tAdjusted R-squared:  0.8167 "
# [1] "Linear Mode      R2 :
#       Multiple R-squared:  0.8419,\tAdjusted R-squared:  0.8167 "
```