model_CA_Kyock_ad.R

atchirc

Sun May 28 00:45:19 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'))
```

```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 GMV ----
# # Lag weekly avg discount by 1 week
model_data$laggmv <- data.table::shift(model_data$gmv)


# # ****************************************************************************
# #                    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

```
# Prune KPI as part of model optimization
model_data <- na.omit(model_data)
model_data <- subset(model_data,select=-c(adTV,SEM,list_mrp,NPS,discount))
```

Linear Model:

```
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                  -24,386.570 (52,036.050)
## deliverycdays         -3,702.380 (408,420.000)
## n_saledays            254,115.000 (196,741.500)
## Sponsorship           143,193.100* (76,370.210)    163,995.400*** (56,409.050)
## OnlineMarketing             0.039 (0.036)               0.048** (0.020)
## Other                       0.009 (0.020)
## chnglist                   0.0002 (0.0001)
## chngdisc               50,439.010 (34,542.720)
## laggmv                      0.179 (0.172)
## Constant           2,166,773.000** (930,056.100) 2,464,873.000*** (735,880.700)
## --------------------------------------------------------------------------------
## Observations                  41                           41
## R2                           0.527                        0.423
## Adjusted R2                  0.389                        0.393
## Residual Std. Error    1,978,944.000 (df = 31)      1,972,566.000 (df = 38)
## F Statistic             3.832*** (df = 9; 31)        13.955*** (df = 2; 38)
## ================================================================================
## Note:                                              *p<0.1; **p<0.05; ***p<0.01
```

```
knitr::kable(viewModelSummaryVIF(step_mdl))
```

var	Estimate	Std.Error	t-value	Pr(>|t|)	Significance	vif
OnlineMarketing	4.764e-02	1.960e-02	2.431	0.01990	*	1.312825
Sponsorship	1.640e+05	5.641e+04	2.907	0.00606	**	1.312825

```
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
```

Model Accuracy

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

```
## [1] 1.76e+12
```

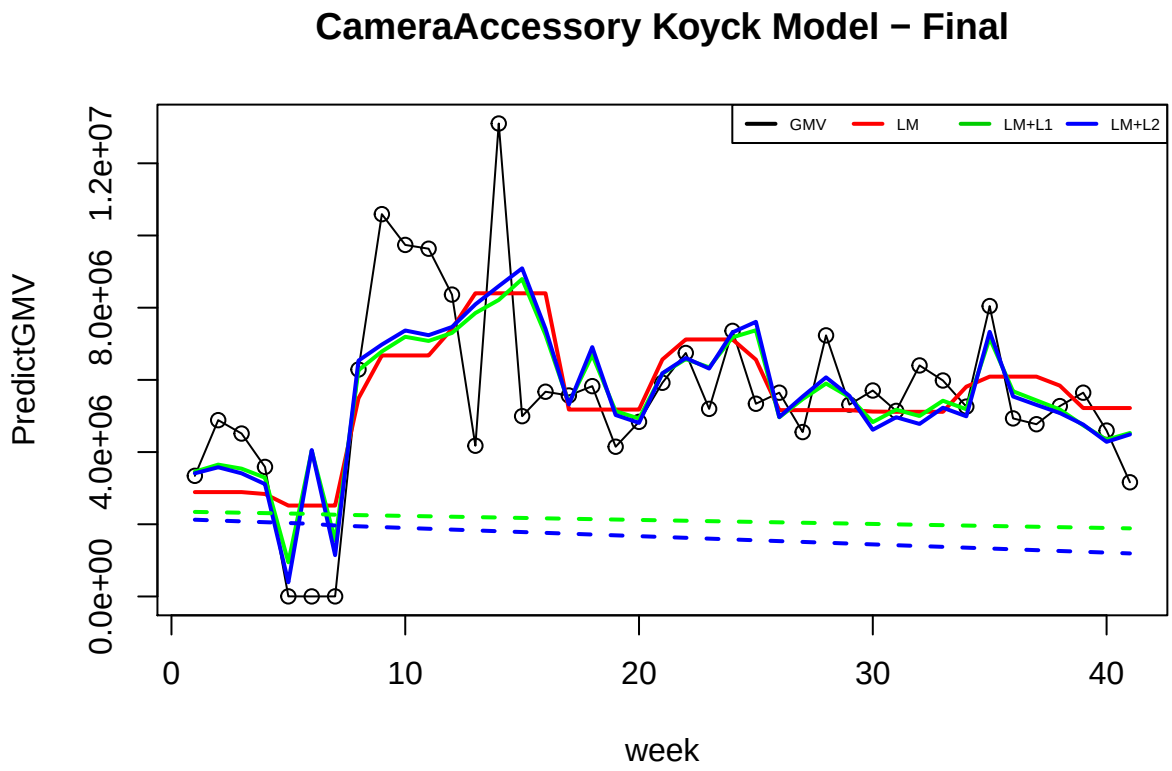
```
predR2 <- 1 - (sum((test_value-ypred )^2)/sum((test_value-mean(ypred))^2))
```

*

PLOTTING MODEL RESULTS

Plot Model prediction and base sales:

```
plot(model_data$gmv,main = 'CameraAccessory Koyck Model - Final',
     xlab='week',ylab='PredictGMV')
lines(model_data$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) 2.464873e+06  2.364804e+06  2.171367e+06
## 2         chngdisc           NA  4.276413e+04  4.982827e+04
## 3         chnglist           NA  1.888756e-04  2.205304e-04
## 4    deliverycdays           NA -8.659115e+04 -1.010557e+04
## 5           laggmv           NA  1.667684e-01  1.773628e-01
## 6       n_saledays           NA  2.092822e+05  2.510645e+05
## 7  OnlineMarketing 4.763977e-02  3.414722e-02  3.885919e-02
## 8            Other           NA  4.649841e-03  8.380687e-03
## 9      Sponsorship 1.639954e+05  1.287132e+05  1.433017e+05
## 10            week           NA -1.112606e+04 -2.277504e+04
```

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

```
## [1] "Ridge regression R2 : 0.519861123432712"
```

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

```
## [1] "Lasso regression R2 : 0.526577185576059"
```

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

```
## [1] "Multiple R-squared:  0.4235,\tAdjusted R-squared:  0.3931 "
## [1] "Linear Mode      R2 : Multiple R-squared:  0.4235,\tAdjusted R-squared:  0.3931 "
```

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

```
## [1] "Predicted        R2 : 0.363092140730299"
```

*

`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) -4.298317e+06  5.794557e+06  2.441952e+06
# 2         chngdisc            NA  2.540553e+04  1.292035e+04
# 3         chnglist            NA  1.401253e-05  0.000000e+00
# 4    deliverycdays            NA  1.634303e+05  1.131055e+05
# 5         discount  7.349317e+04  2.748224e+04  4.521718e+04
# 6           laggmv            NA -1.719029e-02 -3.629628e-02
# 7         list_mrp  3.394976e-04  2.577053e-04  2.832874e-04
# 8       n_saledays  2.476512e+05  2.283399e+05  2.439161e+05
# 9              NPS            NA -1.209985e-02 -8.154427e-03
# 10 OnlineMarketing  3.826100e-02  2.476685e-02  3.161093e-02
# 11           Other            NA  7.390478e-03  1.108746e-02
# 12             SEM -5.215457e-02 -3.435368e-02 -4.995371e-02
# 13     Sponsorship  2.577525e+05  2.008037e+05  2.753698e+05
# 14              TV            NA -1.929945e+05 -4.687682e+05
# 15            week            NA -1.500513e+04 -1.048237e+04
# [1] "Ridge regression R2 : 0.610734183034274"
# [1] "Lasso regression R2 : 0.623163910472765"
# [1] "Multiple R-squared:  0.6006,\tAdjusted R-squared:  0.5461 "
# [1] "Linear Mode      R2 : Multiple R-squared:  0.6006,\tAdjusted R-squared:  0.5461 "
```

```
# coeff            lm            l1            l2
# 1      (Intercept) -6.796520e+05  4.902846e+06  1.579565e+06
# 2         chngdisc  4.705754e+04  4.466822e+04  4.797107e+04
# 3         chnglist            NA  2.532189e-05 -1.008472e-05
# 4    deliverycdays            NA  1.829073e+05  2.898561e+05
# 5           laggmv            NA  3.852542e-03 -2.355309e-02
# 6         list_mrp  3.364383e-04  2.764882e-04  3.425742e-04
# 7       n_saledays            NA  2.220402e+05  2.514192e+05
# 8              NPS            NA -9.020384e-03 -4.332059e-03
# 9  OnlineMarketing  3.879036e-02  1.877191e-02  2.490978e-02
# 10           Other            NA  7.327572e-03  1.300734e-02
# 11     Sponsorship  1.272081e+05  1.300052e+05  1.561283e+05
# 12            week            NA -6.177148e+03 -1.397800e+04
# [1] "Ridge regression R2 : 0.567908635526672"
# [1] "Lasso regression R2 : 0.572846210681951"
# [1] "Multiple R-squared:  0.5319,\tAdjusted R-squared:  0.4912 "
# [1] "Linear Mode      R2 :
#         Multiple R-squared:  0.5319,\tAdjusted R-squared:  0.4912 "
```