model_GA_Kyock_ad.R

arman

Sun May 28 15:17:57 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))

# # . . . . 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)
```

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

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                    1,516.313 (26,627.700)
## deliverycdays          80,377.780 (237,224.100)
## n_saledays             72,994.270 (106,030.900)
## Sponsorship                 0.009* (0.005)               0.008** (0.003)
## OnlineMarketing             0.029 (0.018)                0.028** (0.011)
## Other                       0.016 (0.012)                 0.013 (0.010)
## chnglist                  734.968 (767.013)
## chngdisc               43,984.250** (16,239.840)     40,327.750*** (14,691.030)
## adTV                  -227,473.600 (320,330.400)
## laggmv                      0.123 (0.155)
## Constant             1,189,159.000** (469,001.100) 1,398,962.000*** (347,690.500)
## --------------------------------------------------------------------------------
## Observations                     42                             42
## R2                              0.578                          0.540
## Adjusted R2                     0.441                          0.490
## Residual Std. Error     1,069,187.000 (df = 31)       1,021,579.000 (df = 37)
## F Statistic             4.240*** (df = 10; 31)         10.850*** (df = 4; 37)
## ================================================================================
## 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	4.033e+04	1.469e+04	2.745	0.009281	**	1.020493
OnlineMarketing	2.816e-02	1.140e-02	2.470	0.018229	*	1.912677
Other	1.327e-02	9.606e-03	1.382	0.175406	NA	1.513602
Sponsorship	7.836e-03	3.376e-03	2.321	0.025910	*	1.816885

```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] 1.281779e+12
```

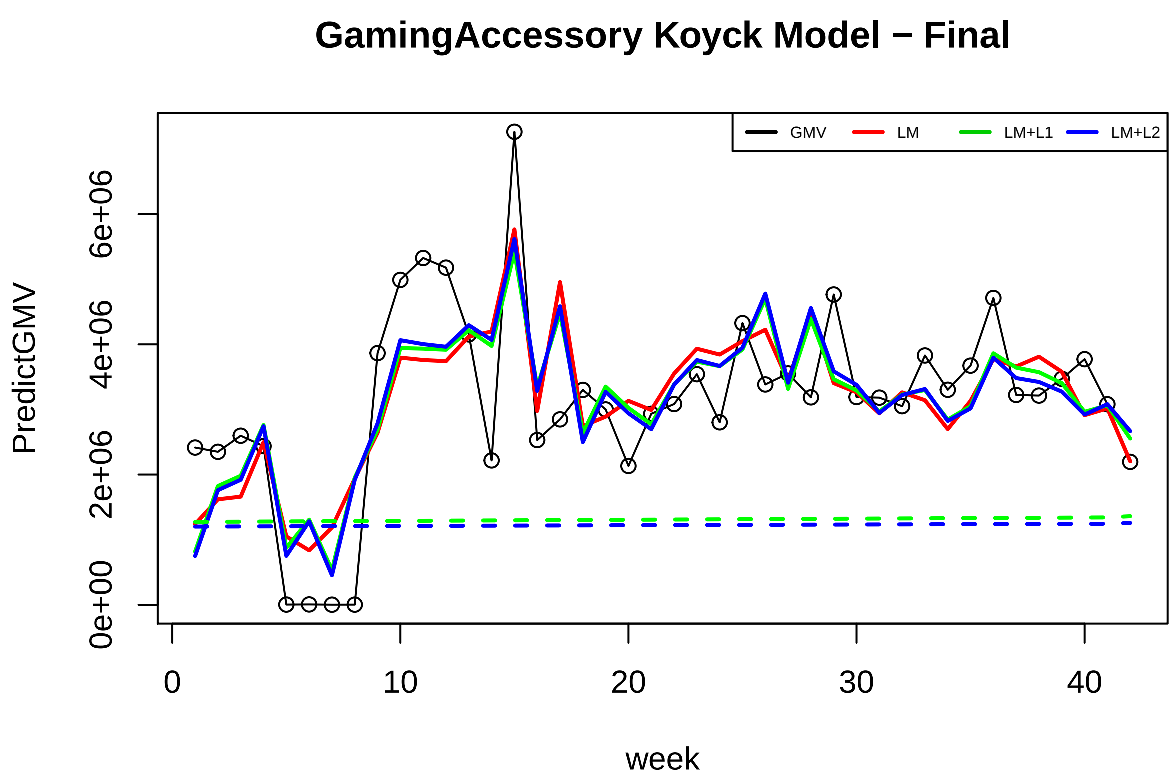
```r
predR2 <- 1 - (sum((test_value-ypred )^2)/sum((test_value-mean(ypred))^2))
```

*

PLOTTING MODEL RESULTS

Plot Model prediction and base sales:

```
plot(model_data$gmv,main = 'GamingAccessory Koyck Model - Final',
     xlab='week',ylab='PredictGMV')
lines(model_data$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) 1.398962e+06  1.269604e+06  1.198674e+06
## 2             adTV           NA -8.474349e+04 -2.130476e+05
## 3         chngdisc 4.032775e+04  4.130034e+04  4.386385e+04
## 4         chnglist           NA  7.311705e+02  7.327198e+02
## 5    deliverycdays           NA  2.407744e+04  7.699762e+04
## 6           laggmv           NA  1.321779e-01  1.230168e-01
## 7       n_saledays           NA  6.694484e+04  7.160612e+04
## 8  OnlineMarketing 2.816190e-02  2.364993e-02  2.892403e-02
## 9            Other 1.327016e-02  1.175369e-02  1.536065e-02
## 10     Sponsorship 7.835575e-03  7.161841e-03  8.385694e-03
## 11            week           NA  1.707781e+03  1.089854e+03
```

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

```
## [1] "Ridge regression R2 : 0.57310366761721"
```

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

```
## [1] "Lasso regression R2 : 0.577621935127604"
```

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

```
## [1] "Multiple R-squared:  0.5398,\tAdjusted R-squared:  0.4901 "
## [1] "Linear Mode      R2 : Multiple R-squared:  0.5398,\tAdjusted R-squared:  0.4901 "
```

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

```
## [1] "Predicted        R2 : -0.0421424387562432"
```

*

Significant KPI