model_HA_LM_ad.R

anandrathi

Sun May 28 16:58:28 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=='HomeAudio',
                     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'))

# model_data <- subset(model_data,select = -c(TV,Sponsorship,
#                                             OnlineMarketing,
#                                             SEM,Other))

model_data <- subset(model_data,select = -c(TV))


# # ************************************************************************
# #                    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(adTV,discount,SEM,NPS,list_mrp))
```

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                    -30,219.000 (53,805.580)
## deliverycdays          -337,760.900 (409,375.800)    -478,655.300* (258,096.700)
## n_saledays              394,963.200* (210,573.500)    443,851.300** (198,614.900)
## Sponsorship                   0.014* (0.008)               0.019*** (0.005)
## OnlineMarketing               0.038 (0.037)
## Other                         0.003 (0.021)
## chnglist                  3,558.962* (1,866.143)       3,774.131** (1,809.999)
## chngdisc                163,776.800*** (51,317.570)    170,566.300*** (49,731.630)
## Constant              3,677,807.000*** (917,328.600) 4,193,061.000*** (606,850.200)
## -----------------------------------------------------------------------------------
## Observations                      42                            42
## R2                               0.576                         0.558
## Adjusted R2                      0.473                         0.497
## Residual Std. Error    2,093,493.000 (df = 33)       2,044,789.000 (df = 36)
## F Statistic             5.598*** (df = 8; 33)         9.107*** (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
chngdisc	1.706e+05	4.973e+04	3.430	0.00153	**	1.138686
chnglist	3.774e+03	1.810e+03	2.085	0.04421	*	1.116807
deliverycdays	-4.787e+05	2.581e+05	-1.855	0.07186	.	1.075986
n_saledays	4.439e+05	1.986e+05	2.235	0.03173	*	1.105204
Sponsorship	1.850e-02	5.380e-03	3.439	0.00149	**	1.070009

```
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] NA
```

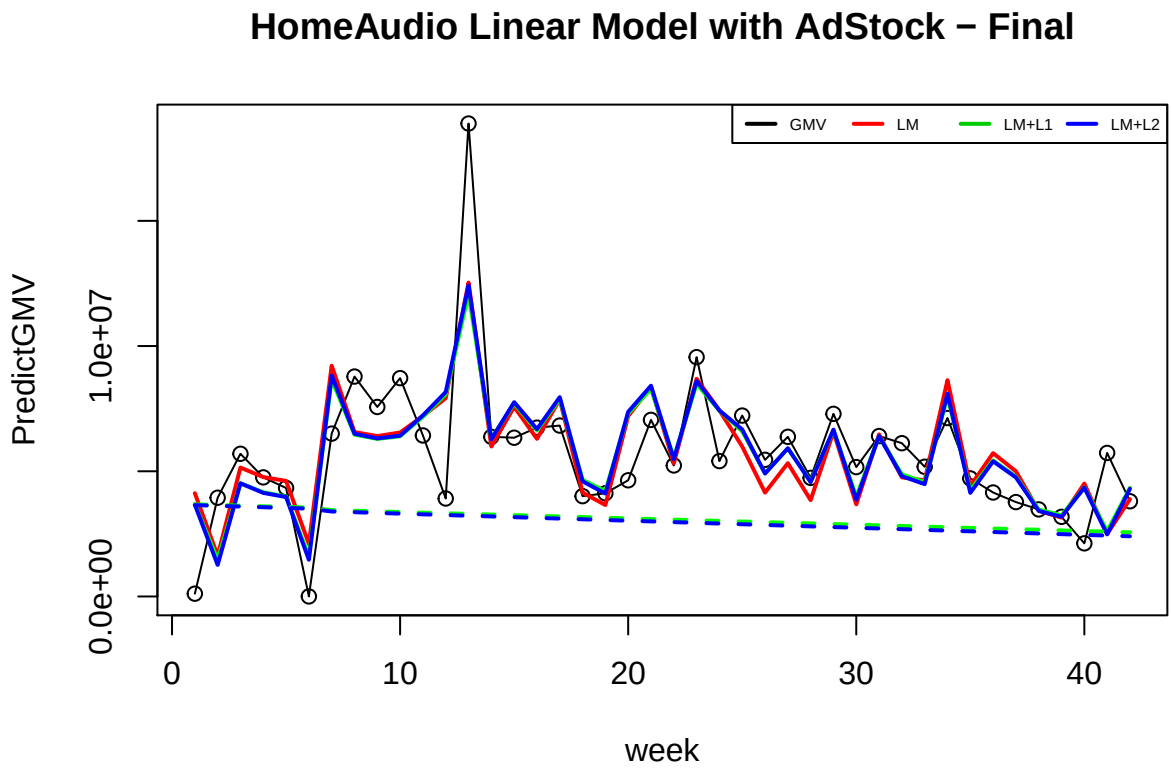
```
predR2 <- 1 - (sum((test_value-ypred )^2)/sum((test_value-mean(ypred))^2))
```

*

PLOTTING MODEL RESULTS

Plot Model prediction and base sales:

```
plot(model_data$gmv,main = 'HomeAudio Linear Model with AdStock - Final',
     xlab='week',ylab='PredictGMV')
lines(model_data$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)  4.193061e+06  3.702033e+06  3.676087e+06
## 2        chngdisc  1.705663e+05  1.538347e+05  1.631007e+05
## 3        chnglist  3.774131e+03  3.273411e+03  3.524668e+03
## 4   deliverycdays -4.786553e+05 -3.507082e+05 -3.462573e+05
## 5      n_saledays  4.438513e+05  3.790258e+05  3.929079e+05
## 6 OnlineMarketing            NA  3.589623e-02  3.696482e-02
## 7           Other            NA  1.929774e-03  2.655230e-03
## 8     Sponsorship  1.850021e-02  1.343365e-02  1.390669e-02
## 9            week            NA -2.533745e+04 -2.827404e+04
```

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

```
## [1] "Ridge regression R2 : 0.574396957409538"
```

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

```
## [1] "Lasso regression R2 : 0.575717332662374"
```

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

```
## [1] "Multiple R-squared:  0.5585,\tAdjusted R-squared:  0.4971 "
## [1] "Linear Mode      R2 : Multiple R-squared:  0.5585,\tAdjusted R-squared:  0.4971 "
```

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

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

*

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) -9.985868e+05  7.343125e+06 -2.414127e+06
# 2  adOnlineMarketing  2.443484e-02  1.094554e-02  2.897134e-02
# 3             adOther            NA  5.072027e-03  1.158766e-02
# 4               adSEM -4.202136e-02 -2.446829e-02 -4.557119e-02
# 5      adSponsorship  1.905706e+05  1.241458e+05  2.222801e+05
# 6               adTV -4.296838e+05 -1.840207e+05 -5.721456e+05
# 7           chngdisc  4.180543e+04  4.653036e+04  4.519779e+04
# 8           chnglist            NA  4.633180e-05  5.844452e-05
# 9      deliverycdays            NA  9.445852e+04  1.027135e+05
# 10          discount            NA -7.942695e+03 -4.903361e+03
# 11          list_mrp  3.527259e-04  2.825139e-04  3.260626e-04
# 12        n_saledays  2.217532e+05  2.069190e+05  2.209012e+05
# 13               NPS            NA -1.258460e-02  3.593073e-03
# 14              week            NA -8.697558e+03 -2.151025e+04
# [1] "Ridge regression R2 : 0.614129507515657"
# [1] "Lasso regression R2 : 0.637433960090558"
# [1] "Multiple R-squared:  0.6227,\tAdjusted R-squared:  0.5626 "
# [1] "Linear Mode      R2 : Multiple R-squared:  0.6227,\tAdjusted R-squared:  0.5626 "
# >
```

```
# 1       (Intercept) 2.407765e+06  2.452181e+06  2.349655e+06
# 2 adOnlineMarketing 2.315286e-02  1.852802e-02  1.867951e-02
# 3           adOther           NA  3.585862e-03  4.526187e-03
# 4     adSponsorship 9.376834e+04  9.827333e+04  1.050286e+05
# 5          chngdisc 5.067210e+04  4.739354e+04  5.039845e+04
# 6          chnglist 2.277615e-04  2.036785e-04  2.190031e-04
# 7     deliverycdays           NA  2.434703e+04  4.042647e+04
# 8         n_saledays           NA  1.903471e+05  2.001274e+05
# 9               week           NA -3.103374e+02 -2.090038e+03
# [1] "Ridge regression R2 : 0.454425529891249"
# [1] "Lasso regression R2 : 0.455736137322237"
# [1] "Multiple R-squared:  0.4395,\tAdjusted R-squared:  0.3918 "
# [1] "Linear Mode      R2 :
#         Multiple R-squared:  0.4395,\tAdjusted R-squared:  0.3918 "
# >
```