Predicting Retailer Revenue

Arvind Koul, Ethel Zhang, Johnny Chiu, Saurabh Tripathi, Varun Gupta, Yuqing Liu

1/21/2018

# Outline

* Set up
* Exploratory Data Analysis
* Feature Engineer
* Model Building
* Model selection

### Set up

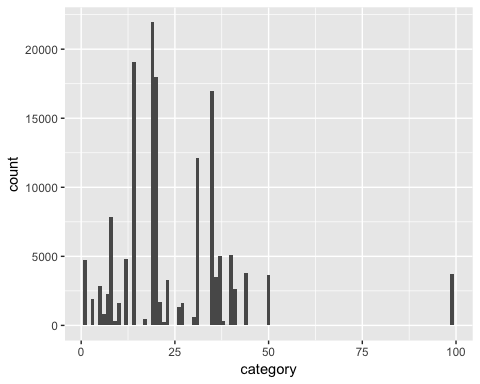
# Set working directory and source function  
pwd = getwd()  
source(paste(pwd,'/function.R',sep=''))  
  
# Read Data  
booktrain\_path = paste(getwd(),'/../data/booktrain.csv',sep='')  
booktrain = read.csv(booktrain\_path)  
orders\_path = paste(getwd(),'/../data/orders.csv',sep='')  
orders = read.csv(orders\_path)  
  
# Only keep the rows from orders that can be matched to booktrain, i.e., the rows with logtarg.  
all= merge(orders, booktrain, by='id',all.x= TRUE)  
  
# Create Train and Test Dataset  
train\_original = all %>% filter(!is.na(logtarg)) # note that train data contains only 8224 customer id instead of 8311, since there are 87 ids in booktrain that can not be matched to the records in orders.  
train\_original = data\_manipulate(train\_original)  
  
test\_original = all %>% filter(is.na(logtarg))  
test\_original = data\_manipulate(test\_original)

### Exploratory Data Analysis

### check NA  
colSums(is.na(train\_original)) # do not have any NA

## id orddate ordnum category qty price   
## 0 0 0 0 0 0   
## logtarg total\_price   
## 0 0

### check the distribution for categorical columns  
ggplot(data=train\_original, aes(x=category)) + geom\_histogram(binwidth=1)



### check outlier for numerical columns  
# > qty  
# check\_outlier(train\_original, 'qty', 10) # I think all the outliers are correct but extreme value. Decide to keep it for now.  
# check\_outlier(train\_original, 'price', 10) # I think all the outliers are correct but extreme value. Decide to keep it for now.  
  
# > price  
# there are 3.25% of rows with the value 0 in column 'price', and most of them are from category 99 (61%).  
  
table(train\_original[train\_original$price==0,]$category)/dim(train\_original[train\_original$price==0,])[1]

##   
## 1 5 7 8 14   
## 0.0214141414 0.0185858586 0.0131313131 0.0004040404 0.1993939394   
## 17 19 20 23 31   
## 0.0012121212 0.0195959596 0.0002020202 0.0002020202 0.0016161616   
## 35 36 37 41 44   
## 0.0046464646 0.0036363636 0.0539393939 0.0004040404 0.0476767677   
## 99   
## 0.6139393939

### Feature Engineering

train = feature\_engineer(train\_original)  
test = feature\_engineer(test\_original, isTrain=FALSE)

Here are the list

|  |  |
| --- | --- |
| **Feature** | Description |
| **days\_recent\_purchase** | days since last purchase |
| **days\_first\_purchase** | days since first purchase |
| **order\_count** | total number of orders |
| **avg\_qty** | average quantity per order |
| **avg\_ord\_value** | average order price |
| **slope** | build a linear regression for each id using date as predictor and daily total quantity as response and use the slope as the trend |
| **coe\_va** | Coefficient of variation using the qty over year. The value is calculated using the variance divide by mean |
| **cat\_count** | number of categories purcahsed per user |
| **entrophy** | how diverse the amount of purchase are from all categories. The value is calculated using entropy function |
| **total\_money** | total money spent per user |
| **avg\_cats** | Average number of categories per order |
| **distinct\_orddate** | count distinct order date per user |
| **aug\_after\_orders** | ?? |
| **ordperyr** | order per year |

### Model Building

**Fit a simple regression model for benchmark**

fit\_multiple = lm(logtarg~.-id, data = train)  
summary(fit\_multiple) # Residual standard error: 0.6114 | Adjusted R-squared: 0.01354

##   
## Call:  
## lm(formula = logtarg ~ . - id, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.5368 -0.1531 -0.1084 -0.0587 5.5538   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.098674347 0.034298166 2.877 0.00403 \*\*   
## days\_recent\_purchase -0.000013324 0.000020562 -0.648 0.51701   
## days\_first\_purchase -0.000073064 0.000018163 -4.023 0.000058 \*\*\*  
## total\_duration NA NA NA NA   
## order\_count 0.026120545 0.032816499 0.796 0.42608   
## avg\_qty 0.009459409 0.005505497 1.718 0.08580 .   
## avg\_ord\_value 0.000247379 0.000233580 1.059 0.28960   
## slope 0.187550185 0.061513015 3.049 0.00230 \*\*   
## coe\_va 0.088771495 0.033497866 2.650 0.00806 \*\*   
## cat\_count 0.009657506 0.004805571 2.010 0.04450 \*   
## entrophy -0.017184729 0.027293220 -0.630 0.52895   
## total\_money -0.000017855 0.000008175 -2.184 0.02898 \*   
## avg\_cats -0.004656328 0.014044949 -0.332 0.74025   
## distinct\_orddate -0.017108734 0.032834036 -0.521 0.60233   
## aug\_after\_orders -0.003189100 0.001033152 -3.087 0.00203 \*\*   
## ordperyr -0.000338384 0.003153378 -0.107 0.91455   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.6113 on 8209 degrees of freedom  
## Multiple R-squared: 0.01556, Adjusted R-squared: 0.01388   
## F-statistic: 9.267 on 14 and 8209 DF, p-value: < 0.00000000000000022

**Stepwise Linear Regression**

# \*\*Backward\*\*  
fit\_stepback <- stepAIC(fit\_multiple,direction = c("backward"),trace=FALSE)  
  
# \*\*Forward\*\*  
fit\_zero <- lm(logtarg ~ 1, data = train)  
fit\_stepforw <- stepAIC(fit\_zero,direction = c("forward"), scope=list(upper=fit\_multiple,lower=fit\_zero),trace=FALSE)

summary(fit\_stepback) # Residual standard error: 0.6112 | Adjusted R-squared: 0.01422

##   
## Call:  
## lm(formula = logtarg ~ days\_first\_purchase + order\_count + avg\_qty +   
## slope + coe\_va + cat\_count + total\_money + aug\_after\_orders,   
## data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.5569 -0.1530 -0.1067 -0.0580 5.5938   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 0.085736218 0.017555772 4.884 0.00000106106911846  
## days\_first\_purchase -0.000079962 0.000010117 -7.904 0.00000000000000305  
## order\_count 0.009831125 0.001947985 5.047 0.00000045887000725  
## avg\_qty 0.009167911 0.004020912 2.280 0.02263  
## slope 0.187822320 0.060347451 3.112 0.00186  
## coe\_va 0.091779546 0.028884492 3.177 0.00149  
## cat\_count 0.006801919 0.002803086 2.427 0.01526  
## total\_money -0.000015717 0.000007491 -2.098 0.03592  
## aug\_after\_orders -0.002918825 0.000954236 -3.059 0.00223  
##   
## (Intercept) \*\*\*  
## days\_first\_purchase \*\*\*  
## order\_count \*\*\*  
## avg\_qty \*   
## slope \*\*   
## coe\_va \*\*   
## cat\_count \*   
## total\_money \*   
## aug\_after\_orders \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.6112 on 8215 degrees of freedom  
## Multiple R-squared: 0.0152, Adjusted R-squared: 0.01424   
## F-statistic: 15.85 on 8 and 8215 DF, p-value: < 0.00000000000000022

summary(fit\_stepforw) # Residual standard error: 0.6112 | Adjusted R-squared: 0.01418

##   
## Call:  
## lm(formula = logtarg ~ days\_recent\_purchase + cat\_count + total\_duration +   
## order\_count + aug\_after\_orders + slope + coe\_va + avg\_ord\_value,   
## data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.8045 -0.1529 -0.1099 -0.0600 5.6117   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.10416289 0.01628863 6.395 0.00000000016951 \*\*\*  
## days\_recent\_purchase -0.00008978 0.00001298 -6.916 0.00000000000498 \*\*\*  
## cat\_count 0.00838132 0.00266445 3.146 0.00166 \*\*   
## total\_duration -0.00007644 0.00001332 -5.738 0.00000000991081 \*\*\*  
## order\_count 0.00781010 0.00185698 4.206 0.00002629161570 \*\*\*  
## aug\_after\_orders -0.00262454 0.00092964 -2.823 0.00477 \*\*   
## slope 0.16997665 0.05863347 2.899 0.00375 \*\*   
## coe\_va 0.08108321 0.02921947 2.775 0.00553 \*\*   
## avg\_ord\_value 0.00033295 0.00021220 1.569 0.11668   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.6114 on 8215 degrees of freedom  
## Multiple R-squared: 0.01479, Adjusted R-squared: 0.01383   
## F-statistic: 15.41 on 8 and 8215 DF, p-value: < 0.00000000000000022

**Lasso Linear Regression**

y=train$logtarg  
x=model.matrix(logtarg~.-id,train)  
param\_lasso = my\_cv\_glmnet(y,x,1)$small.lambda.betas  
param\_lasso = param\_lasso[param\_lasso!=0]  
lasso\_feature = c('logtarg','days\_recent\_purchase', 'days\_first\_purchase', 'order\_count', 'avg\_qty','coe\_va')  
fit\_lasso = lm(logtarg~., data = train[lasso\_feature])  
summary(fit\_lasso) # Residual standard error: 0.6113 | Adjusted R-squared: 0.01386

##   
## Call:  
## lm(formula = logtarg ~ ., data = train[lasso\_feature])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.8510 -0.1500 -0.1121 -0.0653 5.7901   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.11413191 0.01663943 6.859 0.00000000000743 \*\*\*  
## days\_recent\_purchase -0.00002712 0.00001658 -1.635 0.10199   
## days\_first\_purchase -0.00005968 0.00001267 -4.710 0.00000251230717 \*\*\*  
## order\_count 0.00714228 0.00138001 5.176 0.00000023264367 \*\*\*  
## avg\_qty 0.00607723 0.00287794 2.112 0.03475 \*   
## coe\_va 0.08836352 0.02842542 3.109 0.00189 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.612 on 8218 degrees of freedom  
## Multiple R-squared: 0.01222, Adjusted R-squared: 0.01162   
## F-statistic: 20.34 on 5 and 8218 DF, p-value: < 0.00000000000000022

### Model selection

Choose the model according to Root Mean Square Error(RMSE) and Adjusted R-squared

|  |  |  |
| --- | --- | --- |
| Model | RMSE | adj\_R\_square |
| Stepwise Regression Backward | 0.6112286 | 0.0142376 |
| Multiple Linear Regression | 0.6113396 | 0.0138793 |
| Stepwise Regression Forward | 0.6113557 | 0.0138276 |
| Lasso Regression | 0.6120392 | 0.0116212 |

In our final model, the significant features are

|  |  |
| --- | --- |
| **Feature** | Description |
| **days\_first\_purchase** | days since first purchase |
| **order\_count** | total number of orders |
| **avg\_qty** | average quantity per order |
| **slope** | build a linear regression for each id using date as predictor and daily total quantity as response and use the slope as the trend |
| **coe\_va** | Coefficient of variation using the qty over year. The value is calculated using the variance divide by mean |
| **cat\_count** | number of categories purcahsed per user |
| **total\_money** | total money spent per user |
| **aug\_after\_orders** | ?? |

### Make prediction

predict\_reg = predict(fit\_multiple, newdata=test)

## Warning in predict.lm(fit\_multiple, newdata = test): prediction from a  
## rank-deficient fit may be misleading

### Generating submitting file

test$logtarg = predict\_reg  
file\_version1 = test %>% dplyr::select(id, logtarg)  
file\_version1$logtarg = ifelse(file\_version1$logtarg<0, 0, file\_version1$logtarg)  
colnames(file\_version1) = c('id','yhat')  
# write.csv(file\_version1, file=paste('./submission/model\_',format(Sys.time(), "%b%d\_%H%M%S"),'.csv',sep=''),row.names = FALSE)