Prediction of Catalogue Orders

MSBA 400: Statistical Foundations for Data Analytics
Professor Rossi
Author: Sai Akhil Siruguppa

Introduction

The dataset `cat_buy.rda` contains data on the response of customers to the mailing of spring catalogues. The variable `buytabw` is `1` if there is an order from this spring catalogue and `0` if not. This is the dependent or response variable (literally was there a "response" to or order from the direct mailing).

This spring catalogue was called a "tabloid" in the industry. The catalogue featured women's clothing and shoes. The independent variables represent information gathered from the internal `house file` of the past order activity of these 20,617 customers who received this catalogue.

In direct marketing, the predictor variables are typically of the "RFM" type: 1. Recency 2. Frequency and 3. Monetary value. This data set has both information on the volume of past orders as well as the recency of these orders.

The variables are: * tabordrs (total orders from past tabloids)
* divsords (total orders of shoes in past)
* divwords (total orders of women's clothes in past)
* spgtabord (total orders from past spring cats)
* moslsdvs (mos since last shoe order)
* moslsdvw (mos since last women's clothes order)
* moslstab (mos since last tabloid order)
* orders (total orders)

Split Data into training and validation

Using `sample` we select row numbers and then use these row numbers to divide the data into two parts. One part for estimation and one part for validation.

```r
load(file = "cat_buy.rda")
set.seed(1)
sample_size = nrow(cat_buy)/2
ind.est=sample(nrow(cat_buy),size=sample_size)
est_sample = cat_buy[ind.est,]
holdout_sample = cat_buy[-ind.est,]
```

Modeling

Fit a logistic regression model using the estimation sample produced in part A. Eliminate insignificant variables.

```r
out=glm(buytabw~.,family="binomial",data=est_sample)
summary(out)
```

```
##
## Call:
## glm(formula = buytabw ~ ., family = "binomial", data = est_sample)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1835  -0.6370  -0.3769  -0.1311   3.0633
##
```

```
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -1.083313   0.093559 -11.579  < 2e-16 ***
## tabordrs     0.067046   0.013916   4.818 1.45e-06 ***
## divsords     0.023015   0.015977   1.441 0.149721    
## divwords     0.107208   0.008277  12.953  < 2e-16 ***
## spgtabord    0.047141   0.019019   2.479 0.013189 *  
## moslsdvs    -0.007613   0.002198  -3.464 0.000533 ***
## moslsdvw    -0.067523   0.005235 -12.899  < 2e-16 ***
## moslstab    -0.046271   0.004523 -10.231  < 2e-16 ***
## orders      -0.049407   0.005851  -8.444  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 9468.7  on 10312  degrees of freedom
## Residual deviance: 7481.2  on 10304  degrees of freedom
## AIC: 7499.2
## 
## Number of Fisher Scoring iterations: 6
```

We can see that the variable `divsords` has a high p-value indicating that we can accept the null hypothesis that the estimate of the coefficient of `divsords` is zero.

```
# let's refit with only the factors that have significant coefficients
out_2=glm(buytabw~tabordrs + divwords + spgtabord + moslsdvs +
                moslsdvw + moslstab + orders,
                data=est_sample,family="binomial")
summary(out_2)
```

```
## 
## Call:
## glm(formula = buytabw ~ tabordrs + divwords + spgtabord + moslsdvs + 
##     moslsdvw + moslstab + orders, family = "binomial", data = est_sample)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.1748  -0.6372  -0.3761  -0.1311   3.0630  
## 
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -1.038283   0.087950 -11.805  < 2e-16 ***
## tabordrs     0.066833   0.013899   4.809 1.52e-06 ***
## divwords     0.106777   0.008259  12.928  < 2e-16 ***
## spgtabord    0.048072   0.018983   2.532   0.0113 *  
## moslsdvs    -0.009403   0.001809  -5.197 2.03e-07 ***
## moslsdvw    -0.067400   0.005233 -12.881  < 2e-16 ***
## moslstab    -0.046102   0.004523 -10.194  < 2e-16 ***
## orders      -0.047026   0.005594  -8.407  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
```

```
##     Null deviance: 9468.7  on 10312  degrees of freedom
## Residual deviance: 7483.2  on 10305  degrees of freedom
## AIC: 7499.2
##
## Number of Fisher Scoring iterations: 6
```

After the removal of `divsords`, the AIC has remained the same and thus we can justify the removal of the variable from our model.

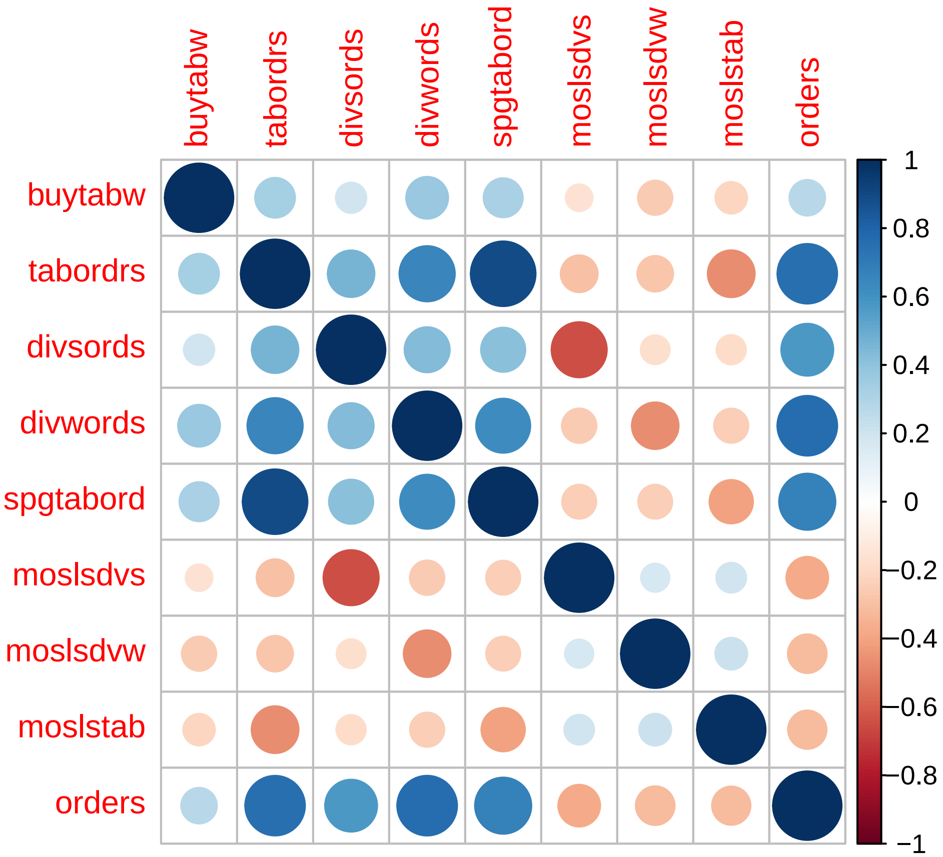
Secondly, we can see that `tabordrs`, `divwords` and `spgtbord` have positive coefficients. These variables indicate a positive purchase behavior in the past and thus it makes sense that they correspond to increased likelihood to purchases in the future.

We can also see that `mosldvs`, `moslsdvw` and `moststab` have negative coefficients. These indicate that if it has been a longer duration since a customer's purchase, the likelihood for the next purchase decreases or is negatively correlated which is intuitive as well.

Additionally, In case of `orders` we can see a negative coefficient. This is counter intuitive since, if a person has more purchases in the past, we would expect more purchases from the same user.

We can calculate the correlation matrix as well as the VIF to check for multicollinearity in the given dataset.

```
corrplot::corrplot(cor(est_sample))
```



```
library("car")
```

```
## Loading required package: carData
```

```
vif(out_2)
```

```
##   tabordrs   divwords  spgtabord   moslsdvs   moslsdvw   moslstab     orders
##  5.457637  2.584978  4.033273  1.125631  1.143236  1.151136  3.636826
```

We can see that `tabordrs` has a high correlation with `spgtabord`, `divwords` and `orders`. Further we can see that it has a relatively high VIF of `5.45`. We can thus retrain the model by dropping the `tabordrs` variable and check the `AIC` to compare the performance of the model.

```
# let's refit with only the factors that have significant coefficients
out_3=glm(buytabw~divwords + spgtabord + moslsdvs +
                moslsdvw + moslstab + orders,
                data=est_sample,family="binomial")
summary(out_3)
```

```
##
## Call:
## glm(formula = buytabw ~ divwords + spgtabord + moslsdvs + moslsdvw +
##     moslstab + orders, family = "binomial", data = est_sample)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2500  -0.6401  -0.3804  -0.1308   3.0128
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.978018   0.086693 -11.281  < 2e-16 ***
## divwords     0.104104   0.008180  12.727  < 2e-16 ***
## spgtabord    0.117519   0.012422   9.460  < 2e-16 ***
## moslsdvs    -0.009431   0.001805  -5.224 1.75e-07 ***
## moslsdvw    -0.067457   0.005218 -12.928  < 2e-16 ***
## moslstab    -0.050984   0.004492 -11.350  < 2e-16 ***
## orders      -0.035936   0.004972  -7.228 4.90e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9468.7  on 10312  degrees of freedom
## Residual deviance: 7506.4  on 10306  degrees of freedom
## AIC: 7520.4
##
## Number of Fisher Scoring iterations: 6
```

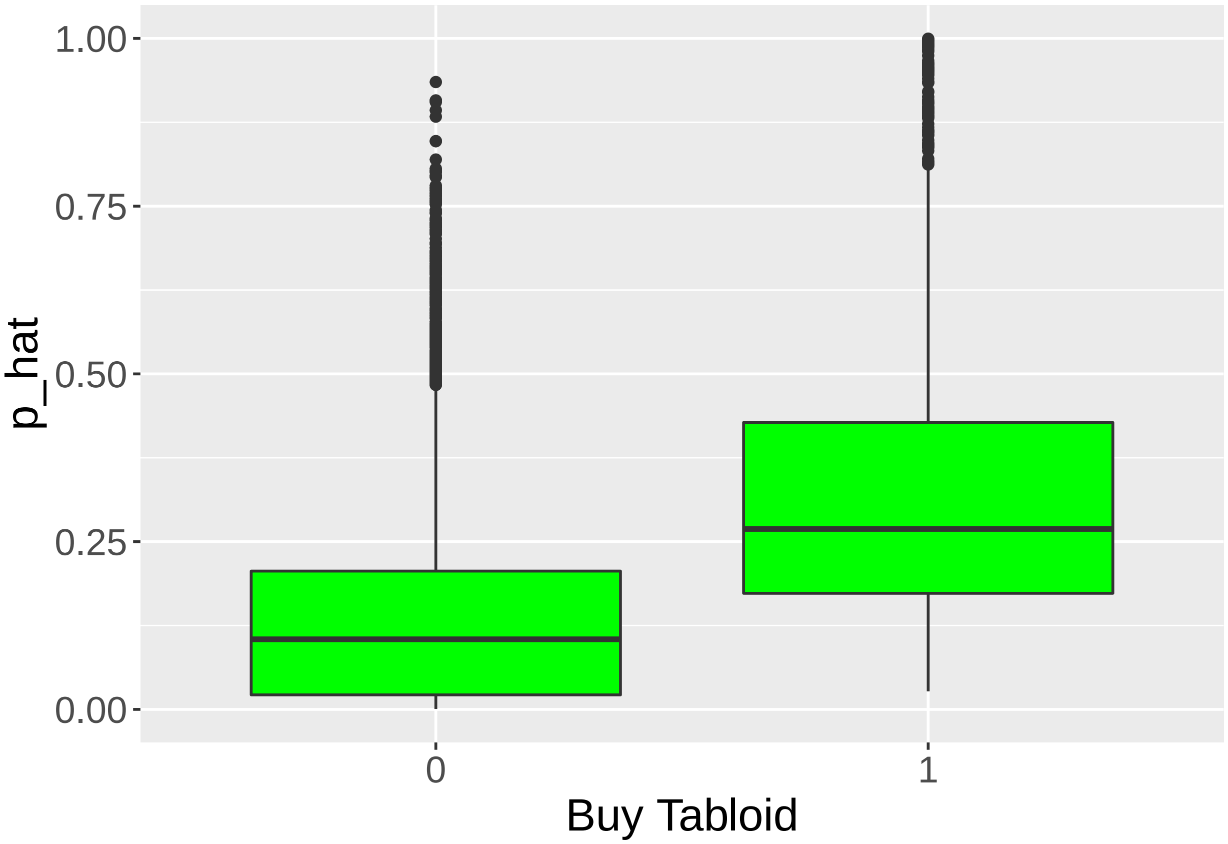
We can see that the `AIC` increased to 7520.4 in the model without the independent variable `tabordrs` indicating a deterioration in the performance of the model. Hence we can see that multicollinearity is not always a cause of concern and in this case we do not need to drop `tabordrs` variable from our model. Additionally, the `VIF` for all the variables is ≤ 10 and thus we can see that there is no substantial cause of concern for multicollinearity in the given dataset.

Prediction using best-fit model

Use the best-fit from part B to predict using the holdout sample.

```
p_hat = predict(out_2, holdout_sample, type='response')
```

```
library(ggplot2)
qplot(factor(holdout_sample$buytabw),
      p_hat, geom = "boxplot", fill=I("green"),
      xlab="Buy Tabloid") +
  theme(axis.title = element_text(size=rel(1.5)),
        axis.text = element_text(size=rel(1.25)))
```



We can see from the box plots that there is a substantial degree of separation between the `p_hat` value for different classes thus indicating a good model performance.

Computing a "lift" table

```
# create deciles
deciles = cut(p_hat, breaks = quantile(p_hat, probs = c (seq(from = 0, to = 1, by = 0.1))),
              include.lowest = TRUE)
deciles = as.numeric(deciles)

# construct data frame and calculate lift table
df = data.frame(deciles = deciles, phat = p_hat, default = holdout_sample$buytabw)
lift = aggregate(df, by = list(deciles), FUN="mean", data=df)
lift = lift[,c(2,4)]
lift[,3]=lift[,2]/mean(holdout_sample$buytabw)
names(lift) = c("decile", "Mean Response", "Lift Factor")
lift
```

```
##    decile Mean Response Lift Factor
## 1       1   0.000000000   0.0000000
## 2       2   0.000000000   0.0000000
## 3       3   0.006789525   0.0389256
## 4       4   0.070736434   0.4055451
## 5       5   0.158098933   0.9064104
## 6       6   0.188166828   1.0787953
## 7       7   0.236434109   1.3555205
## 8       8   0.237633366   1.3623961
## 9       9   0.336566440   1.9295977
## 10     10   0.509689922   2.9221467
```

From the above lift table, we can see that the lift factors increase monotonically as we move up the deciles which indicates a good model performance