Homework 4
MSBA 400: Statistical Foundations for Data Analytics
UID 106082225, Sarvari Pidaparty

Question: Prediction of Catalogue Orders

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)

part A

Use the R `sample` command to randomly sample 1/2 of the data. The sample command will sample randomly from a list of numbers, e.g. 6, 1, 4, 9, 3 will select 5 from the numbers 1,2,3,4,5,6,7,8,9,10.

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

Hint: see code below (modify)

```r
load("~/Documents/Fall 2022/Statistical Foundations/cat_buy.rda")
count = nrow(cat_buy)
ind.est=sample(1:count, size = count/2)
est_sample = cat_buy[ind.est,]
holdout_sample = cat_buy[-ind.est,]

head(est_sample) #print out head for better understanding
```

```
##       buytabw tabordrs divsords divwords spgtabord moslsdvs moslsdvw moslstab
## 3295        0        2        5        2         1   1.7083   5.4862   1.7083
## 2204        0        4        0        5         1  42.0000   3.9750   3.9750
## 3551        0        6        7        0         5   7.0302  42.0000   9.6912
## 6644        0        9        1        5         4  31.8331   6.1761  17.9698
## 14250       0        3        0        6         1  42.0000   0.3285   4.2707
## 15463       1        8        6        5         6   7.4573   0.3942   5.6505
##       orders
## 3295       9
## 2204      12
## 3551      23
## 6644      15
```

```
## 14250      8
## 15463     17
```

```r
head(holdout_sample)
```

```
##    buytabw tabordrs divsords divwords spgtabord moslsdvs moslsdvw moslstab
## 2        0        6        0        4         5  42.0000  29.5335   5.5519
## 4        0        9        1        9         5  22.0105   4.0736   0.3285
## 5        0        2        0        2         1  42.0000   1.4126  12.4836
## 7        0        2        0        0         0  42.0000  42.0000   8.0158
## 10       0        2        0        2         1  42.0000  11.4652  11.4652
## 13       0        1        1        2         1  22.6675   2.4310  28.2194
##    orders
## 2       9
## 4      14
## 5       7
## 7       6
## 10      5
## 13     18
```

part B

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

Discuss your final specification, do the signs of the coefficients make sense to you?

Should you worry about multi-colinearity in this dataset?

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

```
##
## Call:
## glm(formula = buytabw ~ ., family = "binomial", data = est_sample)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2258  -0.6447  -0.3762  -0.1390   2.9194
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.900157   0.091750  -9.811  < 2e-16 ***
## tabordrs     0.049586   0.013996   3.543 0.000396 ***
## divsords    -0.011929   0.016188  -0.737 0.461199
## divwords     0.093502   0.008089  11.559  < 2e-16 ***
## spgtabord    0.079923   0.019449   4.109 3.97e-05 ***
## moslsdvs    -0.010879   0.002183  -4.983 6.27e-07 ***
## moslsdvw    -0.063383   0.004846 -13.080  < 2e-16 ***
## moslstab    -0.052531   0.004751 -11.056  < 2e-16 ***
## orders      -0.045437   0.005930  -7.663 1.82e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9415.1  on 10312  degrees of freedom
```

```
## Residual deviance: 7498.9  on 10304  degrees of freedom
## AIC: 7516.9
##
## Number of Fisher Scoring iterations: 6
```

The variable `divsords` can be eliminated, as p = 0.845, which is greater than the significance level $\alpha = 0.05$. We fit the model again as below eliminating `divsords`.

```
out_model1 = glm(buytabw ~ tabordrs + divwords + spgtabord + moslsdvs + moslsdvw + moslstab + orders , 
summary(out_model1)
```

```
##
## Call:
## glm(formula = buytabw ~ tabordrs + divwords + spgtabord + moslsdvs +
##     moslsdvw + moslstab + orders, family = "binomial", data = est_sample)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2062  -0.6449  -0.3761  -0.1387   2.9174
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.922263   0.086789 -10.626  < 2e-16 ***
## tabordrs     0.049413   0.013996   3.531 0.000415 ***
## divwords     0.093944   0.008074  11.636  < 2e-16 ***
## spgtabord    0.079601   0.019446   4.093 4.25e-05 ***
## moslsdvs    -0.009973   0.001807  -5.520 3.38e-08 ***
## moslsdvw    -0.063404   0.004845 -13.085  < 2e-16 ***
## moslstab    -0.052662   0.004748 -11.093  < 2e-16 ***
## orders      -0.046730   0.005673  -8.238  < 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: 9415.1  on 10312  degrees of freedom
## Residual deviance: 7499.5  on 10305  degrees of freedom
## AIC: 7515.5
##
## Number of Fisher Scoring iterations: 6
```

The intercept coefficient is the log-odds value of the dependent variable when all independent variables are zero. The other coefficients describe how the log-odds of `buytabw` change when the corresponding variable (X) increases by 1 unit (since all X's here are numeric and not categorical). If you exponentiate the coefficients, you get the odds of `buytabw` $\frac{p}{1-p}$.

The positive coefficients against `tabordrs`, `divwords` and `spgtabord` indicate that an increase in any of these predictors results in an increased probability of orders from the catalogue. The negative coefficients against `moslsdvs`, `moslsdvw`, `moslstab` and `orders` indicate that an increase in any of these predictors results in a decreased probability of orders from the catalogue.

```
cor(est_sample)
```

```
##              buytabw   tabordrs   divsords   divwords  spgtabord   moslsdvs
## buytabw    1.0000000  0.3270656  0.1712556  0.3542028  0.3237393 -0.1548891
## tabordrs   0.3270656  1.0000000  0.4631056  0.6427441  0.8984737 -0.2901190
## divsords   0.1712556  0.4631056  1.0000000  0.4099249  0.4135445 -0.6498960
```

```
## divwords   0.3542028  0.6427441  0.4099249  1.0000000  0.6234318 -0.2513740
## spgtabord  0.3237393  0.8984737  0.4135445  0.6234318  1.0000000 -0.2490347
## moslsdvs  -0.1548891 -0.2901190 -0.6498960 -0.2513740 -0.2490347  1.0000000
## moslsdvw  -0.2487841 -0.2698426 -0.1787926 -0.4641370 -0.2487056  0.1689313
## moslstab  -0.2167844 -0.4648566 -0.1904636 -0.2466165 -0.4033573  0.2070503
## orders     0.2576682  0.7589322  0.5775341  0.7521642  0.6818041 -0.3720008
##             moslsdvw   moslstab     orders
## buytabw   -0.2487841 -0.2167844  0.2576682
## tabordrs  -0.2698426 -0.4648566  0.7589322
## divsords  -0.1787926 -0.1904636  0.5775341
## divwords  -0.4641370 -0.2466165  0.7521642
## spgtabord -0.2487056 -0.4033573  0.6818041
## moslsdvs   0.1689313  0.2070503 -0.3720008
## moslsdvw   1.0000000  0.2143647 -0.3138860
## moslstab   0.2143647  1.0000000 -0.3146979
## orders    -0.3138860 -0.3146979  1.0000000
```

There may be an issue of multicolinearity here as we can see a few values close to 1. For example, `spgtabord` and `tabordrs` have a correlation coefficient of ~0.89 which leads to a multicolinearity issue.

Function VIF can be used to understand this as well:

vif(out_model1)

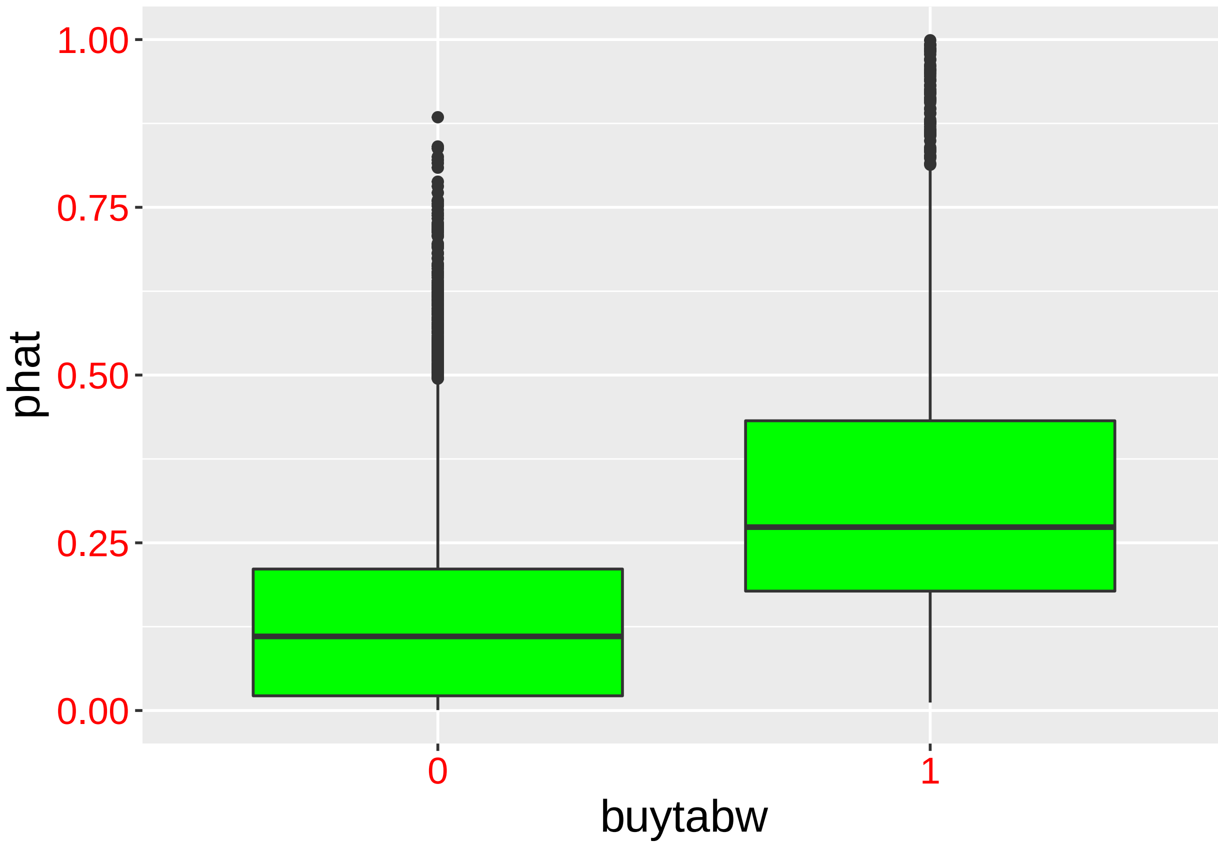
```
tabordrs  divwords spgtabord  moslsdvs  moslsdvw  moslstab    orders
5.474439  2.630030  4.029072  1.125472  1.137376  1.157505  3.809766
```

part C

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

Plot boxplots of the fitted probabilities for each value of `buytabw` for the holdout sample (see code snippets from Chapter 7 for an example)

```
library(ggplot2)
phat = predict(out_model1, holdout_sample, type="response")
qplot(factor(holdout_sample$buytabw), phat, geom="boxplot", fill=I("green"),xlab="buytabw") +
    theme(axis.title=element_text(size=rel(1.5)),
        axis.text=element_text(size=rel(1.25),colour=I("red")))
```



There is an overlap between the two categories, which is not ideal.

Compute a “lift” table as done in Chapter 7 code snippets.

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

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

```
##    decile Mean Response Lift Factor
## 1       1  0.0009689922 0.005503406
## 2       2  0.0000000000 0.000000000
## 3       3  0.0116391853 0.066104932
## 4       4  0.0775193798 0.440272513
## 5       5  0.1503394762 0.853855373
## 6       6  0.1959262852 1.112766357
## 7       7  0.2151162791 1.221756224
## 8       8  0.2434529583 1.382694830
## 9       9  0.3394762367 1.928060520
## 10     10  0.5261627907 2.988349682
```

The lift factor gradually increases with each decile, which is good.