Logistic Regression Analysis

Andrew Wang

12/04/2021

Question: Prediction of Catalogue Orders

part A

The code below allows us to randomly sample 1/2 of the data. In my sample() command, I entered the number of rows for the first argument, then set the size equal to about half of the rows in order to divide the data into two parts for estimation and validation.

```
library(DataAnalytics)
data('cat_buy')

ind.est=sample(1:20627,size=10313)
est_sample = cat_buy[ind.est,]
holdout_sample = cat_buy[-ind.est,]
```

part B

```
cat_buy_logit <- glm(buytabw ~ tabordrs+divwords+spgtabord+moslsdvs+moslsdvw+moslstab+orders,
                     data=est_sample, family=binomial)
summary(cat_buy_logit)
```

```
##
## Call:
## glm(formula = buytabw ~ tabordrs + divwords + spgtabord + moslsdvs +
##     moslsdvw + moslstab + orders, family = binomial, data = est_sample)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2047  -0.6448  -0.3704  -0.1159   3.1022
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.887457   0.087829 -10.104  < 2e-16 ***
## tabordrs     0.052891   0.013580   3.895 9.83e-05 ***
## divwords     0.096240   0.008183  11.761  < 2e-16 ***
## spgtabord    0.065277   0.018913   3.451 0.000558 ***
## moslsdvs    -0.009414   0.001820  -5.172 2.31e-07 ***
## moslsdvw    -0.073643   0.005486 -13.424  < 2e-16 ***
## moslstab    -0.054512   0.004873 -11.187  < 2e-16 ***
## orders      -0.045454   0.005579  -8.147 3.72e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9446.7  on 10312  degrees of freedom
## Residual deviance: 7456.1  on 10305  degrees of freedom
## AIC: 7472.1
##
## Number of Fisher Scoring iterations: 7
```

In the logistic regression model above, I chose to fit the model against all variables except for "divsords" as that variable had a higher P value when compared to the 95% significance level which told me it was an insignificant variable to consider. The signs of the coefficients make sense to me as well, since the variables related to ordering items relevant to the spring catalogue are positive while variables related to recency of purchase are all negative signifying that the longer someone hasn't ordered something, the less likely they are to order something from the spring catalogue which makes sense to me.

We do not need to worry about multi-collinearity because even if we have highly correlated x variables for our model, it does not violate our ability to make predictions for the output. We do not need to delete any X variables simply because they are highly correlated with one another, as we can still estimate multiple regression from it.

part C

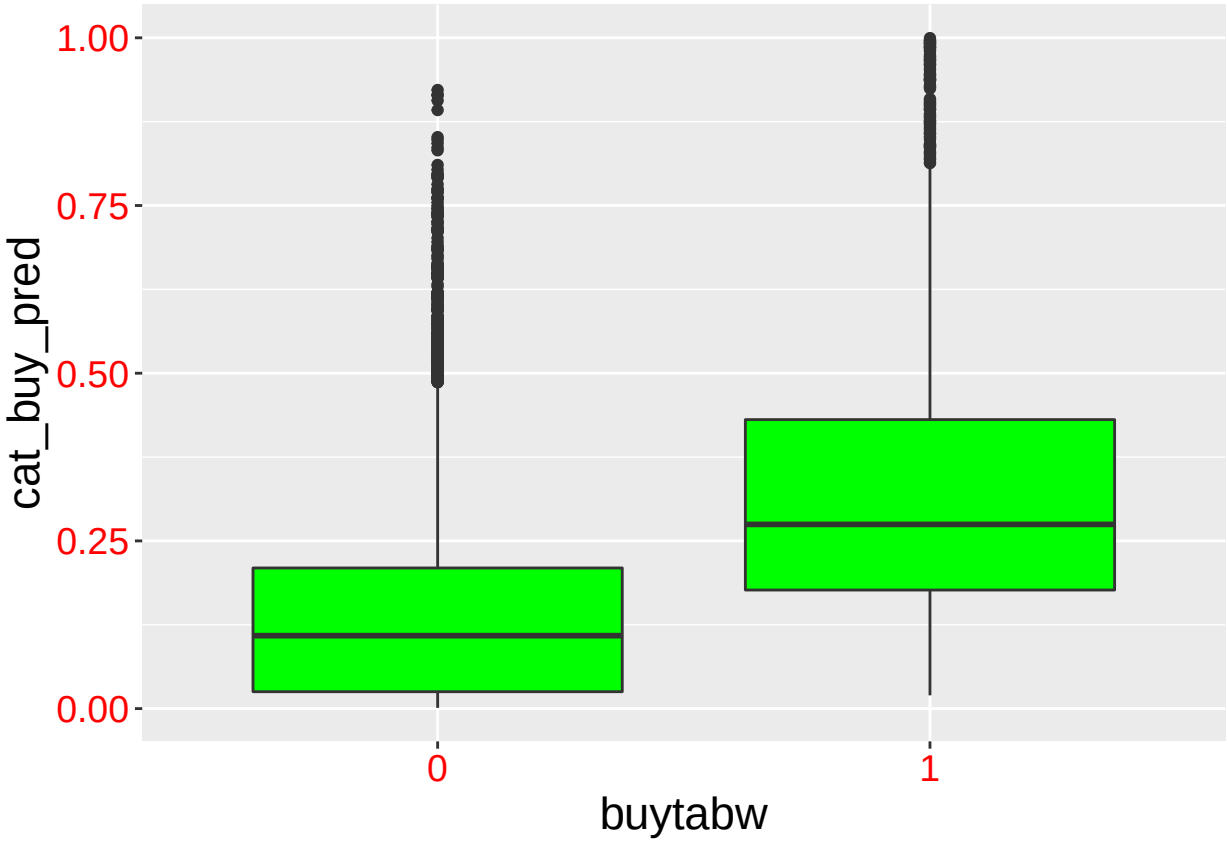
Below, we use the best-fit from part B to predict using the holdout sample.

```
cat_buy_bestfit <- glm(buytabw ~ tabordrs+divwords+spgtabord+moslsdvs+moslsdvw+moslstab+orders,
                       data=holdout_sample, family=binomial)
cat_buy_pred = predict(cat_buy_bestfit, type="response")

library(ggplot2)
```

Then, we plot boxplots of the fitted probabilities for each value of `buytabw` for the holdout sample.

```
qplot(factor(holdout_sample$buytabw),cat_buy_pred,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")))
```



Now, we compute the lift table to evaluate the ability of our model to make predictions.

```
deciles=cut(cat_buy_pred,breaks=quantile(cat_buy_pred,probs=c(seq(from=0,to=1,by=.1))),include.lowest=TR
deciles=as.numeric(deciles)
df=data.frame(deciles=deciles,cat_buy_pred=cat_buy_pred,buytabw=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   0.000000000  0.00000000
## 2       2   0.001939864  0.01107849
## 3       3   0.011639185  0.06647096
## 4       4   0.076550388  0.43717647
## 5       5   0.152279340  0.86966175
## 6       6   0.196896217  1.12446710
## 7       7   0.208333333  1.18978405
## 8       8   0.260911736  1.49005739
## 9       9   0.331716780  1.89442241
## 10     10   0.510658915  2.91635440
```