Individual Assignment 2a – Logistic Regression

Lucas Hagelstein (559020)

13/11/2020

Dr. Michel van de Velden

Dr. Anastasija Tetereva

Dr. Carlo Cavicchia

Seminar in Data Science for Marketing Analytics

MSc in Data Science and Marketing Analytics

Erasmus School of Economic

1. Introduction:

This report investigates consumer choices between travelling either by car or train. The dataset contains information about travel characteristics of each mode and each individual (income and group size).

The research question is defined as follows:

> Will a consumer choose to travel by train or by car to his or her destination?

2. Method:

Binary Logistic Regression is used to predict a dependent variable with two possible outcomes, it takes values either 1 or 0 by transforming the linear model using the non-linear link function $Pr(\varepsilon > -X'\beta) = \frac{exp\ (X'\beta)}{1 + exp\ (X'\beta)}$, whereas the output is defined as the probability $P(Y = 1)$ that a certain event is true. $X'\beta$ is given by the model equation $\beta_0 + \beta_1 x_1 + \cdots + \beta_J x_J$. For $X'\beta = 0$ is the probability $P(Y = 1) = 0.5$. It converges against 1 for values > 0 and against 0 for values < 0. The inverse of the Logistic Regression is the Logit that can be written as $X'\beta = \ln\left(\frac{P(Y=1)}{1-P(Y=1)}\right)$. The log-odds ratio is represented by each beta coefficient β_k of the model equation and can be interpreted as change in the log-odds in response to a one unit increase in the variable X_k (ceteris paribus). Each $exp(\beta_k)$ is equal to the odds ratio $\frac{P(Y=1)}{1-P(Y=1)}$. It determines the change in the odds that a certain event $P(Y = 1)$ occurs over the alternative event $P(Y = 0)$ given a one unit increase in X_k. Finally, the percentage change in the odds ratio is given by $[exp(\beta_k) - 1] * 100$ (Huang, Zhou, Ding & Zhang, 2011 and Hosmer Jr, Lemeshow & Sturdivant, 2013).

3. Data preparation

Information about terminal waiting time and travel costs are provided for the mode train as well as the alternative travel time and costs for the car. Time is measured in minutes and costs are measured in dollars. Consumer-level information is available regarding the household income (in thousand dollars) and the size of the travel group.

The Variance Inflation Factor (VIF) is used to evaluate multicollinearity in the model. If the VIF exceeds the threshold of 5, high multicollinearity is indicated thus coefficients might be estimated inaccurately. A solution is to either drop one variable or to replace highly correlated variables with a combination of both. Due to the high VIF "time_diff_tc" is created by subtracting car travel time from train travel time. In other words, a positive (negative) value indicates the car (train) to be faster (in minutes). Additionally, the household income per capita of travel groups is considered by the variable "income_pp". Next, Cooks Distance is used to indicate possible outliers. Observations 57, 73 and 91 are suspected outliers but given the very small sample size it is not advisable to drop variables to avoid a selection-bias. Further, there may be subjective reasons such as travel comfort why an individual prefers the car over the train even though its travel time is much longer (Table 1). The final model is specified as

$$X'\beta = \beta_0 + \beta_1 * ttime\_t + \beta_2 * invc\_t + \beta_3 * invc\_c + \beta_4 * income\_pp + \beta_5 * time\_diff\_tc + \varepsilon\ .$$

Table 1 Suspected Outliers

obs	by_car	ttime_t	invc_t	invc_c	income_pp	time_diff_tc
57	1	44	63	10	14.0	-522
73	1	44	90	20	13.5	-508
91	0	99	37	10	3.0	-36

4. Results

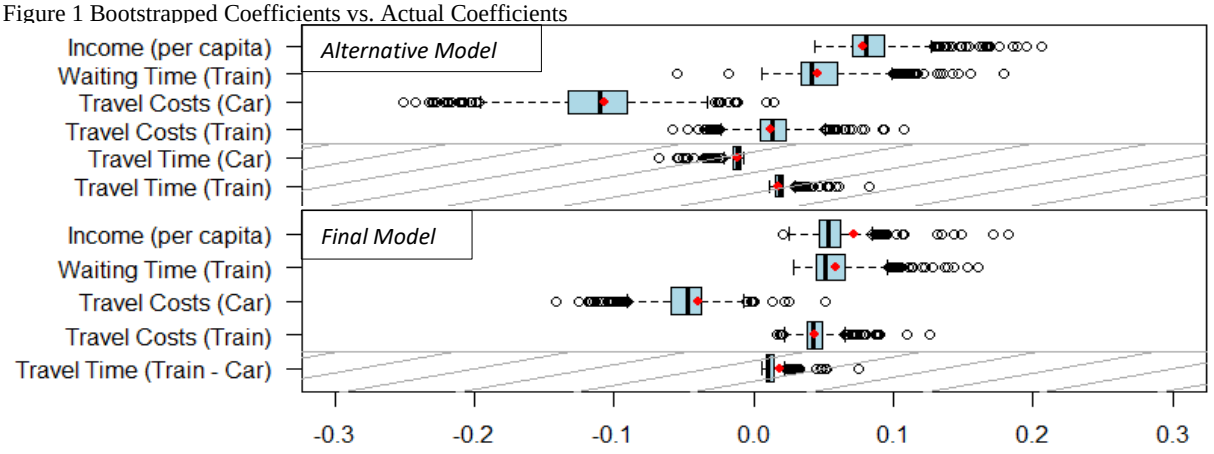
First, the odds ratio of choosing the car over the train given an increase in train terminal waiting time of one minute is equal to 1.06 ($exp(0.0583)$). In terms of percentage change this is equal to 6.00% (p<0.05) ($[1.06 - 1] * 100$). Second, if train travel costs increase by $1, the odds of choosing the car over the train increase by 4.48% (p<0.05). Third, the effect of car travel costs seems to be negative but is not significant. Fourth, an increase of the income per capita by $1000 dollars drives the odds to choose the car up by 7.43% (p<0.05). Finally, if the car is one minute faster than the train the odds of choosing the car over the train will increase by 1.85% (p<0.01). This effect is equal to 41.69% $\left(\left[\frac{exp(20*0.018342)}{exp(0.018342)} - 1\right] * 100\right)$ if consumers are able to save up 20 minutes.

The performance of the fitted model can be assessed with a confusion matrix using the predicted values of the test data. Three predictions (9.6%) are false negatives (Type II-error) that is to predict train but the individual actually prefers the car. 16 predictions are true negatives and 12 predictions, true positives. The accuracy is 90.4% with Cohen's Kappa to be 0.805 which indicates a good fit. A major drawback is the reliance of the accuracy metrics on the 75/25 (train/test) split particularly in small samples. To gain some confidence about the results and to overcome the latter problem a bootstrap is used. In other words, 1000 combinations of train/test splits for the final model and an alternative model are conducted. The latter contains the highly correlated variables seperately instead of "time_diff_tc". Accuracy metrics and coefficient estimates are saved in each iteration to obtain 95% Confidence-Intervals (CI).

5. Conclusion and comments

First, 95% of obtained accuracy values lie within 70.9% and 93.5%. The final model (90.6%) is located below the upper boundary and might overestimate the accuracy but is not significantly different. Second, the alternative model (with separate time variables) performs as good as the final model in terms of accuracy. However, each of the 95% CI of the coefficient estimates is broader. Hence, the alternative model depends stronger on the train/test split. Third, an over-/underestimation of the coefficient estimates can be detected by comparing actual coefficient estimates of the model to the 95% CI's. For example, the effect of travel time and income might be overestimated in the final model (Figure 1). Coming back to the initial research question, if a consumer travels by car or by train rather depends on characteristics based on the train since car travel costs have been found to not have an impact and already a few minutes of saved time (either waiting or driving) increase the probability to travel by car.

Figure 1 Bootstrapped Coefficients vs. Actual Coefficients



6. References

Huang, G. B., Zhou, H., Ding, X., & Zhang, R. (2011). Extreme learning machine for regression and multiclass classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 42(2), 513-529.

Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression (Vol. 398). John Wiley & Sons.

Appendix: Code

```
######
# Individual Report 1 - Logistic Regression
# by Lucas Hagelstein
######
# Set-up
library(car)
library(corrplot)
library(MASS)
library(caret)
######
# Rename Binary Variable
colnames(TransportMode)[1] <- ("by_car")
# Inspect data
summary(TransportMode)
# Check correlation of variables
corrplot(cor(TransportMode[2:8]), method="number")
#> travel time for train and travel time for car are highly correlated (r=0.86)
### Create new variables
## Income per person
TransportMode$income_pp <- TransportMode$hinc / TransportMode$psize
## Combination of highly correlated variables
TransportMode$time_diff_tc <- TransportMode$invt_t - TransportMode$invt_c
### Fit a model with the whole data
m1 <- glm(by_car ~. , data=TransportMode[,c(-7,-8,-10)],family=binomial)
### Diagonstics
## Multicollinearity
vif(m1)
##      ttime_t       invc_t       invt_t       invc_c    invt_c    income_pp
##         1.22         2.03        11.68         2.48      8.80         1.63
#> VIF of invt_c =8.8 and invt_t = 11.68 might be critical!
## Variable time_diff_tc might solve the problem!
# Value is positive (negative) if the car (train) is faster!
m1a <- glm(by_car ~. , data=TransportMode[,c(-4,-6,-7,-8)],family=binomial)
vif(m1a)
##      ttime_t       invc_t       invc_c    income_pp time_diff_tc
##         1.23         1.39         1.28         1.21         1.15
#> VIF < 2 for all variables!
#> Problem solved, continue with m1a
## Autocorrelation and Heteroskedasticity
plot(m1a$residuals, main="Correlation of error term", ylab=("Residuals"))
# Plot residuals
abline(a=0, b=0, col="red")
## No funneling detected (homoscedastic)
#> No Autocorrelation
## Influential values
par(mfrow=c(2,2))
plot(m1a)
# Cooks distance
plot(m1a, which = 4)
#> observation 57, 73, 91 may influence the regression
## Investigate the obs. and compare them to mean values without the outliers
subset <- TransportMode[c(57,73,91),c(1,2,3,5,9,10)]
subset
##    by_car    ttime_t    invc_t    invc_c    income_pp    time_diff_tc
## 57      1         44        63        10         14.0            -522
## 73      1         44        90        20         13.5            -508
## 91      0         99        37        10          3.0             -36

summary(TransportMode[c(-57,-73,-91),c(1,2,3,5,9,10)])
```

```
##         ttime_t    invc_t    invc_c    income_pp    time_diff_tc
## Min.   :    1.0     11.00      2.00         1.33         -498.00
## 1st Qu.:   25.0     25.00      9.50         8.88          -48.50
## Median :   34.0     35.00     14.00        17.50           16.00
## Mean   :   33.5     44.81     18.57        20.11           38.26
## 3rd Qu.:   44.0     66.50     22.00        30.00          132.00
## Max.   :   75.0    101.00     54.00        70.00          327.00
## Evaluate observations:
#> obs. 57: The car is 522 minutes slower,but very cheap compared to the train travel
            costs! CHOICE: CAR -> no outlier
#> obs. 73: Same pattern as in obs. 57! CHOICE: CAR -> no outlier
#> obs. 91: Waiting time of 99 minutes is very high! Also the car is cheaper
            than the train but 36 minutes slower.
            This observation is very irrational but is kept due to the very
            small sample! (Selection-bias)
#>          --> Also might be driven by a very "green-attitude" or no access to
                a car!
#> All suspected outliers are kept!
### Randomly split the data into 75/25 (train/test)
set.seed(777)
sample_size = floor(0.75*nrow(TransportMode))
train = sample(seq_len(nrow(TransportMode)),size = sample_size)
train_data = TransportMode[train,]
test_data =TransportMode[-train,]
### Fit models with the training data and evaluate on test data
## Fit Final_Model with training data
#> This model contains a combination of invt_t and invt_c (=time_diff_tc)
#> Corrects for high VIF of both variables!
m_final <- glm(by_car ~. , data=train_data[,c(-4,-6,-7,-8)],family=binomial)
summary(m_final)
## Coefficients:
##               Estimate    Std. Error    z value Pr(>|z|)
## (Intercept)  -5.476903      1.664977     -3.289 0.001004 **
## ttime_t       0.058313      0.025451      2.291 0.021953 *
## invc_t        0.043835      0.020297      2.160 0.030799 *
## invc_c       -0.040271      0.039494     -1.020 0.307879
## income_pp     0.071661      0.032358      2.215 0.026783 *
## time_diff_tc  0.018342      0.004845      3.785 0.000153 ***
## ---
#predict the test data with m2
predicted <- predict(m_final, test_data,type="response")
test_data$predicted <- predicted
#Create a confusion matrix with the caret-package
confusionMatrix(data = as.factor(as.numeric(predicted>0.5)),
                reference = test_data$by_car)
## show variables that are wrong predicted
test_data$predicted <- as.factor(as.numeric(predicted>0.5))
test_data[test_data$by_car != test_data$predicted,][,c(1,2,3,5,9,10)]
## Fit alternative model
# The alternative model includes both highly correlated variables seperately
m2 <- glm(by_car ~. , data=train_data[,c(-7,-8,-10)],family=binomial)
# predict the test data with the alternative model
predicted <- predict(m2, test_data,type="response")
test_data$predicted <- predicted
# Create a confusion matrix with the caret-package
confusionMatrix(data = as.factor(as.numeric(predicted>0.5)),
                reference = test_data$by_car)
# show variables that are wrong predicted
test_data$predicted <- as.factor(as.numeric(predicted>0.5))
test_data[test_data$by_car != test_data$predicted,][,c(1,2,3,4,5,6,9)]
```

```r
### Bootstrap 1000 combinations of train-test splits to measure average accuracy
##> Try to overcome the reliance on the split by train and test each combination
##> and measuring the accuracy and beta coefficients during this process!
boot_acc1 <- matrix(,0,4)
boot_coef1 <- matrix(,0,7)
boot_acc2 <- matrix(,0,4)
boot_coef2 <- matrix(,0,6)
for (i in 1:1000){
  set.seed(i)
  sample_size = floor(0.75*nrow(TransportMode))
  # randomly split data
  train = sample(seq_len(nrow(TransportMode)),size = sample_size)
  train_sample <- TransportMode[train,]
  test_sample <- TransportMode[-train,]
  boot_model1 <- glm(by_car ~. , data=train_sample[,c(-7,-8,10)],
                     family=binomial)
  boot_model2 <- glm(by_car ~. , data=train_sample[,c(-4,-6,-7,-8)],
                     family=binomial)
  # predict test_sample and create confusion matrix
  boot_pred1 <- predict(boot_model1, test_sample,type="response")
  boot_pred2 <- predict(boot_model2, test_sample,type="response")
  conf1 <- confusionMatrix(data = as.factor(as.numeric(boot_pred1>0.5)),
                           reference = test_sample$by_car)
  conf2 <- confusionMatrix(data = as.factor(as.numeric(boot_pred2>0.5)),
                           reference = test_sample$by_car)
  # store accuracy measures and beta coefficients
  boot_acc1 <- rbind(boot_acc1, conf1$overall[1:4])
  boot_acc2 <- rbind(boot_acc2, conf2$overall[1:4])
  boot_coef1 <- rbind(boot_coef1, boot_model1$coefficients)
  boot_coef2 <- rbind(boot_coef2, boot_model2$coefficients)
}
## Create 95% CI to compare the model coefficients and accuracy measures
boot_acc_CI1 <- apply(boot_acc1, 2, quantile, c(0.025,0.5, 0.975))
boot_coef_CI1 <- apply(boot_coef1, 2, quantile, c(0.025,0.5, 0.975))
boot_acc_CI1
##          Accuracy    Kappa    AccuracyLower    AccuracyUpper
## 2.5%        0.710    0.410            0.520            0.858
## 50%         0.839    0.677            0.663            0.945
## 97.5%       0.935    0.871            0.786            0.992
boot_coef_CI1
##        (Intercept)    ttime_t    invc_t    invt_t    invc_c    invt_c    income_pp
## 2.5%       -10.049      0.020    -0.019     0.013    -0.195    -0.029        0.056
## 50%         -5.583      0.042     0.014     0.018    -0.110    -0.011        0.080
## 97.5%       -4.491      0.108     0.049     0.032    -0.046    -0.008        0.136
boot_acc_CI2 <- apply(boot_acc2, 2, quantile, c(0.025,0.5, 0.975))
boot_coef_CI2 <- apply(boot_coef2, 2, quantile, c(0.025,0.5, 0.975))
boot_acc_CI2
##          Accuracy    Kappa    AccuracyLower    AccuracyUpper
## 2.5%        0.710    0.432            0.520            0.858
## 50%         0.839    0.676            0.663            0.945
## 97.5%       0.935    0.871            0.786            0.992
boot_coef_CI2
##         (Intercept)    ttime_t    invc_t    invc_c    income_pp    time_diff_tc
## 2.5%         -6.773      0.034     0.028    -0.097        0.035           0.007
## 50%          -4.434      0.051     0.043    -0.048        0.054           0.011
## 97.5%        -3.416      0.099     0.065    -0.019        0.089           0.028
## Interpretation
#> Both models perform equal in terms of accuracy
#> Final Model has smaller 95% CI for the beta coefficients
#> Combination of highly correlated variables might improve the model
## Create boxplots of the bootstrapped coefficients
```

```r
#> Check if the actual coefficients of the analysis are over-/ underestimated
par(mfrow=c(2,1),mar=c(2,10,0.5,0.5))
boot_coef1 <- as.data.frame(boot_coef1)
colnames(boot_coef1) <- c("Intercept", "Waiting Time (Train)",
                          "Travel Costs (Train)", "Travel Time (Train)",
                          "Travel Costs (Car)", "Travel Time (Car)",
                          "Income (per capita)")
boxplot(boot_coef1[,c(4,6,3,5,2,7)], horizontal = TRUE, las=1,
        ylim =c(-0.3,0.3),
        main="", col = "lightblue")
points(y = 1:6, x = m1$coefficients[c(4,6,3,5,2,7)],col="red", pch=20,
       cex = 1.2)
rect(-1, 0, 1, 2.5, col="darkgrey", density = 5)
boot_coef2 <- as.data.frame(boot_coef2)
colnames(boot_coef2) <- c("Intercept", "Waiting Time (Train)",
                          "Travel Costs (Train)", "Travel Costs (Car)",
                          "Income (per capita)", "Travel Time (Train - Car)")
boxplot(boot_coef2[,c(6, 3, 4, 2, 5)], horizontal = TRUE, las=1,
        ylim =c(-0.3,0.3), xlab="Coefficient Estimate",
        main="", col = "lightblue")
points(y = 1:5, x = m2$coefficients[c(6, 3, 4, 2, 5)],col="red", pch=20,
       cex = 1.2)
rect(-1, 0, 1, 1.5, col="darkgrey", density = 5)
```