Eindopdracht_deel_3

Mohammed Al Hor

2022-10-30

```
# Libraries
library(dplyr)
library(car)
library(effects)
library(tidyverse)
library(MASS)
library(leaps)
library(sandwich)
library(lmtest)
library(caret)
library(ggplot2)
```

Laden data

```
# setwd("~/Documents/Data-Science-Business-Analytics/Data")
setwd("~/Data-Science-Business-Analytics/Data")
college_statistics <- read.csv("college_statistics.csv", header = TRUE,
strip.white = TRUE, stringsAsFactors = FALSE, na.strings = c("NA", "") )
# Rownames vullen met inhoud van de eerste kolom
rownames(college_statistics) <- college_statistics[,1]
# Verwijder eerste kolom
college_statistics <- college_statistics[,-1]
```

6. Maak een model om de factoren te vinden die bijdragen aan een hoog “slagingssucces”.

6 (a) Definieer een nieuwe variabele die 1 als het slagingspercentage groter is dan 60% en 0 als dat niet zo is.

Graduation rate cannot be higher than 100, therefore we must drop the following observation

```
summary(college_statistics$Grad.Rate)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.00   53.00   65.00   65.46   78.00  118.00
```

We use mutate to create a dummy variable and a case when to set the conditions for this variable.

```
df <- college_statistics %>%
        filter(!Grad.Rate>100) %>%
        mutate(gr_dummy = case_when(Grad.Rate > 60 ~ 1, Grad.Rate <=60 ~ 0))
```

Make this a factor variable

```
df$gr_dummy <-  as.factor(df$gr_dummy)
```

6 (b) Deel de data opnieuw op in een estimation en een test sample.

We set the seed so results can be replicated.

```
set.seed(123)
```

We take the sample for the training data set, we use the same sample size as in previous questions:

```
train_ind <- sample(seq_len(nrow(df)), size=600)

college_statistics_est <- df[train_ind,] # estimation set
college_statistics_test <- df[-train_ind,] # test set
```

6 c) Maak mbv. de estimation data een logit model om de slagingssucces variabele te verklaren. Denk hierbij goed na over transformaties van je variabelen. Bijvoorbeeld heeft het zin om het aantal applicaties, aantal acceptaties, en het aantal enrollments in hetzelfde model op te nemen? Of kunnen sommige van deze variabelen beter als percentages opgenomen worden?

First, we model without any transformations

```
fit1 <- glm(gr_dummy ~ Private + Apps + Accept + Enroll + Top10perc +
Top25perc + F.Undergrad + P.Undergrad + Outstate + Room.Board + Books +
Personal + PhD + Terminal + S.F.Ratio + perc.alumni + Expend ,family =
binomial(link=logit), data = college_statistics_est)

summary(fit1)

##
## Call:
## glm(formula = gr_dummy ~ Private + Apps + Accept + Enroll + Top10perc +
##     Top25perc + F.Undergrad + P.Undergrad + Outstate + Room.Board +
##     Books + Personal + PhD + Terminal + S.F.Ratio + perc.alumni +
##     Expend, family = binomial(link = logit), data =
college_statistics_est)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7202  -0.6983   0.2458   0.6467   2.6117
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.831e+00  1.178e+00  -4.951 7.38e-07 ***
```

```
## PrivateYes   1.312e+00  4.308e-01   3.045  0.00232 **
## Apps         2.082e-04  1.744e-04   1.193  0.23271
## Accept       2.184e-05  3.168e-04   0.069  0.94505
## Enroll       8.903e-04  6.247e-04   1.425  0.15412
## Top10perc   -5.666e-03  2.128e-02  -0.266  0.79004
## Top25perc    3.792e-02  1.534e-02   2.471  0.01347 *
## F.Undergrad -1.794e-04  1.008e-04  -1.781  0.07494 .
## P.Undergrad -1.829e-04  1.276e-04  -1.434  0.15157
## Outstate     1.545e-04  6.050e-05   2.554  0.01064 *
## Room.Board   4.571e-04  1.540e-04   2.969  0.00299 **
## Books       -1.525e-03  7.173e-04  -2.126  0.03351 *
## Personal    -1.414e-04  1.739e-04  -0.813  0.41616
## PhD          1.155e-02  1.262e-02   0.916  0.35989
## Terminal     7.401e-03  1.375e-02   0.538  0.59039
## S.F.Ratio   -1.911e-02  3.650e-02  -0.524  0.60062
## perc.alumni  2.639e-02  1.245e-02   2.119  0.03405 *
## Expend      -1.123e-04  3.942e-05  -2.849  0.00438 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 800.68  on 599  degrees of freedom
## Residual deviance: 514.78  on 582  degrees of freedom
## AIC: 550.78
##
## Number of Fisher Scoring iterations: 6
```

```
vif(fit1)
```

```
##     Private        Apps      Accept      Enroll   Top10perc   Top25perc
##    3.031586   20.257760   31.675250   20.604430    5.660222    5.104667
## F.Undergrad P.Undergrad    Outstate  Room.Board       Books    Personal
##   16.947964    1.941025    2.555122    1.609576    1.144622    1.224818
##         PhD    Terminal   S.F.Ratio perc.alumni      Expend
##    3.159052    3.075968    1.682702    1.372230    2.176333
```

We see that Apps, Accept and enroll have vif values of 20+(this makes sense, there's obviously some multicolineairity at play here). Let's do some transformations on Enroll and Accept to mitigate this.

For both the test and estimation dataframe we calculate the percentage of students that applied and got accepted and the percentage of students enrolled vs. the ones that got accepted.

```
college_statistics_est <- college_statistics_est %>% mutate(acc_rate =
Accept/Apps, enroll_rate = Enroll/Accept) #  accepted and actually enrolled

college_statistics_test <- college_statistics_test %>% mutate(acc_rate =
Accept/Apps, enroll_rate = Enroll/Accept)
```

Let do some modeling on these transformed variables.

```
fit2 <- glm(gr_dummy ~ Private + Apps + acc_rate + enroll_rate + Top10perc +
Top25perc + F.Undergrad + P.Undergrad + Outstate + Room.Board + Books +
Personal + PhD + Terminal + S.F.Ratio + perc.alumni + Expend ,family =
binomial(link=logit), data = college_statistics_est)
summary(fit2)
```

```
##
## Call:
## glm(formula = gr_dummy ~ Private + Apps + acc_rate + enroll_rate +
##     Top10perc + Top25perc + F.Undergrad + P.Undergrad + Outstate +
##     Room.Board + Books + Personal + PhD + Terminal + S.F.Ratio +
##     perc.alumni + Expend, family = binomial(link = logit), data =
college_statistics_est)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6657  -0.6532   0.2452   0.6557   2.5730
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.014e+00  1.667e+00  -3.008  0.00263 **
## PrivateYes   1.320e+00  4.265e-01   3.094  0.00197 **
## Apps         2.454e-04  1.139e-04   2.155  0.03116 *
## acc_rate    -6.182e-01  1.071e+00  -0.577  0.56383
## enroll_rate -1.195e-01  1.048e+00  -0.114  0.90923
## Top10perc    1.312e-04  2.077e-02   0.006  0.99496
## Top25perc    3.216e-02  1.501e-02   2.143  0.03212 *
## F.Undergrad -3.226e-05  7.446e-05  -0.433  0.66487
## P.Undergrad -2.026e-04  1.293e-04  -1.567  0.11723
## Outstate     1.651e-04  6.090e-05   2.711  0.00671 **
## Room.Board   4.076e-04  1.545e-04   2.639  0.00832 **
## Books       -1.539e-03  7.246e-04  -2.124  0.03364 *
## Personal    -1.094e-04  1.730e-04  -0.633  0.52691
## PhD          1.144e-02  1.267e-02   0.903  0.36644
## Terminal     7.693e-03  1.373e-02   0.561  0.57513
## S.F.Ratio   -2.070e-02  3.658e-02  -0.566  0.57159
## perc.alumni  2.898e-02  1.238e-02   2.342  0.01921 *
## Expend      -1.173e-04  3.966e-05  -2.957  0.00311 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 800.68  on 599  degrees of freedom
## Residual deviance: 517.23  on 582  degrees of freedom
## AIC: 553.23
##
## Number of Fisher Scoring iterations: 6
```

```
vif(fit2)

##     Private        Apps    acc_rate enroll_rate   Top10perc   Top25perc
##    2.971695    8.894722    1.533619    1.615540    5.321280    4.910031
## F.Undergrad P.Undergrad    Outstate  Room.Board       Books    Personal
##    9.602901    1.968720    2.558344    1.635465    1.157528    1.220361
##         PhD    Terminal   S.F.Ratio perc.alumni      Expend
##    3.205071    3.082206    1.707379    1.348657    2.165162
```

We observe a significant decrease of the VIF for these variables. Let's move on to the next question and do some variable selection. This model contains too many variables.

6 (d) Gebruik wederom backward selection om het aantal verklarende variabelen te verkleinen.

We use backward selection, the code is as follows.

```
backresults <- stepAIC(fit2, direction = "backward")
```

I will spare you the output of the previous statement, for it is quite long!

We record the best model selected by the backwards method. This line takes the model specification as 'code'

```
backmodel <- backresults$call
backmodel

## glm(formula = gr_dummy ~ Private + Apps + Top25perc + P.Undergrad +
##     Outstate + Room.Board + Books + PhD + perc.alumni + Expend,
##     family = binomial(link = logit), data = college_statistics_est)

# This line evaluates the 'code' of the model
backmodel <- eval(backmodel)
summary(backmodel)

##
## Call:
## glm(formula = gr_dummy ~ Private + Apps + Top25perc + P.Undergrad +
##     Outstate + Room.Board + Books + PhD + perc.alumni + Expend,
##     family = binomial(link = logit), data = college_statistics_est)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6744  -0.6554   0.2599   0.6648   2.5036
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.091e+00  8.195e-01  -7.432 1.07e-13 ***
## PrivateYes   1.350e+00  4.107e-01   3.288 0.001010 **
## Apps         2.120e-04  6.007e-05   3.528 0.000418 ***
## Top25perc    3.316e-02  8.041e-03   4.125 3.71e-05 ***
## P.Undergrad -2.521e-04  1.243e-04  -2.028 0.042580 *
```

```
## Outstate     1.652e-04  5.803e-05   2.846 0.004426 **
## Room.Board   4.646e-04  1.461e-04   3.179 0.001475 **
## Books       -1.589e-03  7.018e-04  -2.264 0.023584 *
## PhD          1.580e-02  8.937e-03   1.768 0.077011 .
## perc.alumni  2.987e-02  1.213e-02   2.462 0.013801 *
## Expend      -1.057e-04  3.390e-05  -3.116 0.001830 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 800.68  on 599  degrees of freedom
## Residual deviance: 519.31  on 589  degrees of freedom
## AIC: 541.31
##
## Number of Fisher Scoring iterations: 5
```

Assign this model to 'fit2'.

```
fit2 <- backmodel

summary(fit2)

##
## Call:
## glm(formula = gr_dummy ~ Private + Apps + Top25perc + P.Undergrad +
##     Outstate + Room.Board + Books + PhD + perc.alumni + Expend,
##     family = binomial(link = logit), data = college_statistics_est)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6744  -0.6554   0.2599   0.6648   2.5036
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.091e+00  8.195e-01  -7.432 1.07e-13 ***
## PrivateYes   1.350e+00  4.107e-01   3.288 0.001010 **
## Apps         2.120e-04  6.007e-05   3.528 0.000418 ***
## Top25perc    3.316e-02  8.041e-03   4.125 3.71e-05 ***
## P.Undergrad -2.521e-04  1.243e-04  -2.028 0.042580 *
## Outstate     1.652e-04  5.803e-05   2.846 0.004426 **
## Room.Board   4.646e-04  1.461e-04   3.179 0.001475 **
## Books       -1.589e-03  7.018e-04  -2.264 0.023584 *
## PhD          1.580e-02  8.937e-03   1.768 0.077011 .
## perc.alumni  2.987e-02  1.213e-02   2.462 0.013801 *
## Expend      -1.057e-04  3.390e-05  -3.116 0.001830 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 800.68  on 599  degrees of freedom
## Residual deviance: 519.31  on 589  degrees of freedom
## AIC: 541.31
##
## Number of Fisher Scoring iterations: 5
```

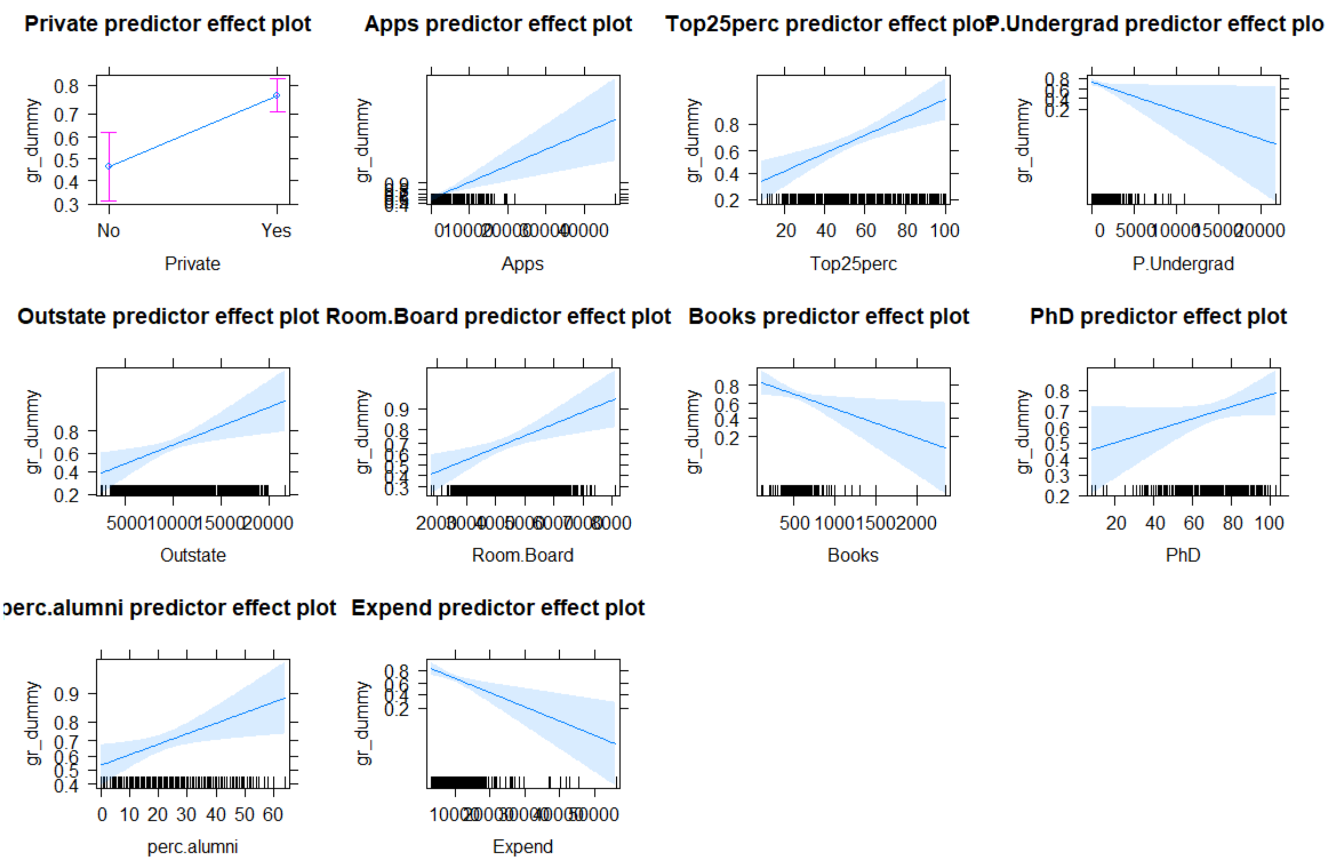
```
vif(fit2)
```

```
##     Private        Apps    Top25perc P.Undergrad    Outstate  Room.Board
##    2.765426    2.477885    1.418699    1.666015    2.368552    1.462017
##       Books         PhD perc.alumni      Expend
##    1.069698    1.604762    1.309924    1.636877
```

None of the vif values are larger than 4 (rule of thumb), thus no multicolinearity.

Lastly to get a better feel for the model and its coefficients we can use the effects package to get ceteris paribus plots

```
plot(predictorEffects(fit2))
```



6 (e) Welke variabelen hebben uiteindelijk een significante invloed?

Private, Apps, Top25perc, P.Undergrad, Outstate, Room.Board, Books, perc.alumni and Expend are significant at the 5 percent level. The variables have a significant effect.

6 (f) Bereken het percentage goed voorspelde scholen zowel voor de estimation sample als voor de test sample (maak eerst voorspellingen voor beide datasets en gebruik daarna bijvoorbeeld de functie confusionMatrix()).

Let's get to the fun stuff and do some predictions on the estimation (training) dataset and the test set we made.

```
college_statistics_test$predict <- predict(fit2, newdata =
college_statistics_test)
college_statistics_est$predict <- predict(fit2, newdata =
college_statistics_est)
```

We need to convert the predictions to 0's and 1's.

```
college_statistics_test <- college_statistics_test %>%
  mutate(predict2 = case_when(predict >= 0.5 ~ 1,predict < 0.5 ~ 0))

college_statistics_est <- college_statistics_est %>%
  mutate(predict2 = case_when(predict >= 0.5 ~ 1,predict < 0.5 ~ 0))
```

Let's take a look at our predictions vs. the actual values using a confusion matrix.

```
confusionMatrix(college_statistics_test$gr_dummy,
as.factor(college_statistics_test$predict2))

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 48 23
##          1 27 78
##
##                Accuracy : 0.7159
##                  95% CI : (0.6432, 0.7812)
##     No Information Rate : 0.5739
##     P-Value [Acc > NIR] : 6.924e-05
##
##                   Kappa : 0.4151
##
##  Mcnemar's Test P-Value : 0.6714
##
##             Sensitivity : 0.6400
##             Specificity : 0.7723
##          Pos Pred Value : 0.6761
```

```
##          Neg Pred Value : 0.7429
##              Prevalence : 0.4261
##          Detection Rate : 0.2727
##    Detection Prevalence : 0.4034
##       Balanced Accuracy : 0.7061
##
##        'Positive' Class : 0
##
```

The accuracy of this model on the test data sits at around 72%. So, to answer the question, the model predicted the correct value (0 or 1) in 72% of the observations.

```
confusionMatrix(college_statistics_est$gr_dummy,
as.factor(college_statistics_est$predict2))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 189  43
##          1  88 280
##
##                Accuracy : 0.7817
##                  95% CI : (0.7464, 0.8141)
##     No Information Rate : 0.5383
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5556
##
##  Mcnemar's Test P-Value : 0.0001209
##
##             Sensitivity : 0.6823
##             Specificity : 0.8669
##          Pos Pred Value : 0.8147
##          Neg Pred Value : 0.7609
##              Prevalence : 0.4617
##          Detection Rate : 0.3150
##    Detection Prevalence : 0.3867
##       Balanced Accuracy : 0.7746
##
##        'Positive' Class : 0
##
```

The accuracy of this model on the estimation data sits at around 78%. So, to answer the question, the model predicted the correct value (0 or 1) in 78% of the observations. This is higher than on the test data, for obvious reasons(we used this data to train our model.)