# Applied Data Science II - Homework 5

Phileas Dazeley Gaist

08/02/2021

## Libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(nnet)
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-3
```

### 1. ISLR 5.4, Question 5

#### a)

```
library(ISLR2)
```

```
##
## Attaching package: 'ISLR2'
```

```
## The following object is masked from 'package:MASS':
##
##     Boston
```

```
default <- Default
head(default)
```

```
##   default student   balance    income
## 1      No      No  729.5265 44361.625
## 2      No     Yes  817.1804 12106.135
## 3      No      No 1073.5492 31767.139
## 4      No      No  529.2506 35704.494
## 5      No      No  785.6559 38463.496
## 6      No     Yes  919.5885  7491.559
```

```
logit <- glm(default ~ income + balance, data = default, family = binomial(link="logit"))
summary(logit)
```

```
##
## Call:
## glm(formula = default ~ income + balance, family = binomial(link = "logit"),
##     data = default)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4725  -0.1444  -0.0574  -0.0211   3.7245
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.154e+01  4.348e-01 -26.545  < 2e-16 ***
## income       2.081e-05  4.985e-06   4.174 2.99e-05 ***
## balance      5.647e-03  2.274e-04  24.836  < 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: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1579.0  on 9997  degrees of freedom
## AIC: 1585
##
## Number of Fisher Scoring iterations: 8
```

**b)**

```r
set.seed(1)

# split the dataset into training and testing sets
training_samples <- default$default %>%
  createDataPartition(p = 0.8, list = FALSE)
train_data  <- default[training_samples, ]
test_data <- default[-training_samples, ]

# check it worked properly
dim(train_data); dim(test_data)
```

```
## [1] 8001    4
```

```
## [1] 1999    4
```

```r
# Fit a logistic model
logit <- glm(default ~ income + balance, data = train_data, family = binomial(link="logit"))
# summary(logit)
```

```r
# Make predictions for the full model
logit_pred <- predict(logit, newdata=test_data, "response")
logit_predicted_classes <- as.factor(ifelse(logit_pred > 0.5, "Yes", "No"))

# Let's make a table
logit_table <- table(test_data$default, logit_predicted_classes)
caret::confusionMatrix(logit_table)
```

```
## Confusion Matrix and Statistics
##
##      logit_predicted_classes
##         No  Yes
##   No  1930    3
##   Yes   51   15
##
##                Accuracy : 0.973
##                  95% CI : (0.9649, 0.9796)
##     No Information Rate : 0.991
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3479
##
##  Mcnemar's Test P-Value : 1.596e-10
##
##             Sensitivity : 0.9743
##             Specificity : 0.8333
##          Pos Pred Value : 0.9984
##          Neg Pred Value : 0.2273
##              Prevalence : 0.9910
##          Detection Rate : 0.9655
##    Detection Prevalence : 0.9670
##       Balanced Accuracy : 0.9038
##
##        'Positive' Class : No
##
```

### c)

0.3 split:

```r
set.seed(1)

# split the dataset into training and testing sets
training_samples <- default$default %>%
  createDataPartition(p = 0.3, list = FALSE)
train_data  <- default[training_samples, ]
test_data <- default[-training_samples, ]
```

```r
# Fit a logistic model
logit <- glm(default ~ income + balance, data = train_data, family = binomial(link="logit"))
# summary(logit)

# Make predictions for the full model
logit_pred <- predict(logit, newdata=test_data, "response")
logit_predicted_classes <- as.factor(ifelse(logit_pred > 0.5, "Yes", "No"))

# Let's make a table
logit_table <- table(test_data$default, logit_predicted_classes)
caret::confusionMatrix(logit_table)
```

```
## Confusion Matrix and Statistics
##
##      logit_predicted_classes
##         No  Yes
##   No  6737   29
##   Yes  151   82
##
##                Accuracy : 0.9743
##                  95% CI : (0.9703, 0.9779)
##     No Information Rate : 0.9841
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.4653
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9781
##             Specificity : 0.7387
##          Pos Pred Value : 0.9957
##          Neg Pred Value : 0.3519
##              Prevalence : 0.9841
##          Detection Rate : 0.9626
##    Detection Prevalence : 0.9667
##       Balanced Accuracy : 0.8584
##
##        'Positive' Class : No
##
```

0.5 split:

```r
set.seed(1)

# split the dataset into training and testing sets
training_samples <- default$default %>%
```

```r
  createDataPartition(p = 0.5, list = FALSE)
train_data  <- default[training_samples, ]
test_data <- default[-training_samples, ]

# Fit a logistic model
logit <- glm(default ~ income + balance, data = train_data, family = binomial(link="logit"))
# summary(logit)

# Make predictions for the full model
logit_pred <- predict(logit, newdata=test_data, "response")
logit_predicted_classes <- as.factor(ifelse(logit_pred > 0.5, "Yes", "No"))

# Let's make a table
logit_table <- table(test_data$default, logit_predicted_classes)
caret::confusionMatrix(logit_table)
```

```
## Confusion Matrix and Statistics
##
##      logit_predicted_classes
##         No  Yes
##   No  4817   16
##   Yes  126   40
##
##                Accuracy : 0.9716
##                  95% CI : (0.9666, 0.976)
##     No Information Rate : 0.9888
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3495
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9745
##             Specificity : 0.7143
##          Pos Pred Value : 0.9967
##          Neg Pred Value : 0.2410
##              Prevalence : 0.9888
##          Detection Rate : 0.9636
##    Detection Prevalence : 0.9668
##       Balanced Accuracy : 0.8444
##
##        'Positive' Class : No
##
```

0.75 split

```r
set.seed(1)

# split the dataset into training and testing sets
training_samples <- default$default %>%
  createDataPartition(p = 0.75, list = FALSE)
train_data  <- default[training_samples, ]
test_data <- default[-training_samples, ]

# Fit a logistic model
logit <- glm(default ~ income + balance, data = train_data, family = binomial(link="logit"))
# summary(logit)

# Make predictions for the full model
logit_pred <- predict(logit, newdata=test_data, "response")
logit_predicted_classes <- as.factor(ifelse(logit_pred > 0.5, "Yes", "No"))

# Let's make a table
logit_table <- table(test_data$default, logit_predicted_classes)
caret::confusionMatrix(logit_table)
```

```
## Confusion Matrix and Statistics
##
##      logit_predicted_classes
##         No  Yes
##   No  2412    4
##   Yes   58   25
##
##                Accuracy : 0.9752
##                  95% CI : (0.9683, 0.9809)
##     No Information Rate : 0.9884
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.4367
##
##  Mcnemar's Test P-Value : 1.685e-11
##
##             Sensitivity : 0.9765
##             Specificity : 0.8621
##          Pos Pred Value : 0.9983
##          Neg Pred Value : 0.3012
##              Prevalence : 0.9884
##          Detection Rate : 0.9652
##    Detection Prevalence : 0.9668
##       Balanced Accuracy : 0.9193
##
##        'Positive' Class : No
##
```

**d)**

```
set.seed(1)

# split the dataset into training and testing sets
training_samples <- default$default %>%
  createDataPartition(p = 0.75, list = FALSE)
train_data  <- default[training_samples, ]
test_data <- default[-training_samples, ]

# Fit a logistic model
logit <- glm(default ~ income + balance + as.factor(student), data = train_data, family = binor
# summary(logit)

# Make predictions for the full model
logit_pred <- predict(logit, newdata=test_data, "response")
logit_predicted_classes <- as.factor(ifelse(logit_pred > 0.5, "Yes", "No"))

# Let's make a table
logit_table <- table(test_data$default, logit_predicted_classes)
caret::confusionMatrix(logit_table)
```

```
## Confusion Matrix and Statistics
##
##      logit_predicted_classes
##         No  Yes
##   No  2412    4
##   Yes   59   24
##
##                Accuracy : 0.9748
##                  95% CI : (0.9679, 0.9806)
##     No Information Rate : 0.9888
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.4228
##
##  Mcnemar's Test P-Value : 1.022e-11
##
##             Sensitivity : 0.9761
##             Specificity : 0.8571
##          Pos Pred Value : 0.9983
##          Neg Pred Value : 0.2892
##              Prevalence : 0.9888
##          Detection Rate : 0.9652
##    Detection Prevalence : 0.9668
##       Balanced Accuracy : 0.9166
##
```

```
##        'Positive' Class : No
##
```

Adding the dummy variable to the model did not lead to a reduction in test error rate.

## Session Info

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] ISLR2_1.3-1     glmnet_4.1-3    Matrix_1.4-0    caret_6.0-90
##  [5] lattice_0.20-45 nnet_7.3-17     MASS_7.3-55     forcats_0.5.1
##  [9] stringr_1.4.0   dplyr_1.0.7     purrr_0.3.4     readr_2.1.1
## [13] tidyr_1.1.4     tibble_3.1.6    ggplot2_3.3.5   tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-155         fs_1.5.2             lubridate_1.8.0
##  [4] httr_1.4.2           tools_4.1.2          backports_1.4.1
##  [7] utf8_1.2.2           R6_2.5.1             rpart_4.1.16
## [10] DBI_1.1.2            colorspace_2.0-2     withr_2.4.3
## [13] tidyselect_1.1.1     compiler_4.1.2       cli_3.1.1
## [16] rvest_1.0.2          formatR_1.11         xml2_1.3.3
## [19] scales_1.1.1         proxy_0.4-26         digest_0.6.29
## [22] rmarkdown_2.11       pkgconfig_2.0.3      htmltools_0.5.2
## [25] parallelly_1.30.0    dbplyr_2.1.1         fastmap_1.1.0
## [28] rlang_1.0.0          readxl_1.3.1         rstudioapi_0.13
## [31] shape_1.4.6          generics_0.1.1       jsonlite_1.7.3
## [34] ModelMetrics_1.2.2.2 magrittr_2.0.2       Rcpp_1.0.8
## [37] munsell_0.5.0        fansi_1.0.2          lifecycle_1.0.1
## [40] stringi_1.7.6        pROC_1.18.0          yaml_2.2.2
## [43] plyr_1.8.6           recipes_0.1.17       grid_4.1.2
```

```
## [46] parallel_4.1.2      listenv_0.8.0       crayon_1.4.2
## [49] haven_2.4.3         splines_4.1.2       hms_1.1.1
## [52] knitr_1.37          pillar_1.6.5        future.apply_1.8.1
## [55] reshape2_1.4.4      codetools_0.2-18    stats4_4.1.2
## [58] reprex_2.0.1        glue_1.6.1          evaluate_0.14
## [61] data.table_1.14.2   modelr_0.1.8        vctrs_0.3.8
## [64] tzdb_0.2.0          foreach_1.5.1       cellranger_1.1.0
## [67] gtable_0.3.0        future_1.23.0       assertthat_0.2.1
## [70] xfun_0.29           gower_0.2.2         prodlim_2019.11.13
## [73] broom_0.7.12        e1071_1.7-9         class_7.3-20
## [76] survival_3.2-13     timeDate_3043.102   iterators_1.0.13
## [79] lava_1.6.10         globals_0.14.0      ellipsis_0.3.2
## [82] ipred_0.9-12
```