Sp23_Midterm_Exam

Areej Mulla

3/2/2023

Reading the File

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.1.3
```

```
bank_full <- read_delim(paste0(getwd(), "/bank_full.csv"))
```

```
## Rows: 45211 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ";"
## chr (10): job, marital, education, default, housing, loan, contact, month, p...
## dbl  (7): age, balance, day, duration, campaign, pdays, previous
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Cleaning the Data

The data set has no NA values, but rather has "unknown" values.

Addressing the Dependent Variable (y):

```
# convert the dependent variable to 0s & 1s
bank_full$y <- ifelse(bank_full$y == "yes", 1, 0)
```

Addressing the Independent Variables:

```
# treat "education" as a factor from 1 to 4 (has "unknown" values)
bank_full$education <- factor(bank_full$education,
                              level = c("unknown", "primary", "secondary", "tertiary"),
                              exclude = NULL)

bank_full$housing <- ifelse(bank_full$housing == "yes", 1, 0)
```

```
bank_full$loan <- ifelse(bank_full$loan == "yes", 1, 0)
bank_full$default <- ifelse(bank_full$default == "yes", 1, 0)

# treat "contact" as a factor from 1 to 3 (has "unknown" values)
bank_full$contact <- factor(bank_full$contact,
                            level = c("unknown", "cellular", "telephone"),
                            exclude = NULL)

# treat marital as a factor from 1 to 3 (doesn't have "unknown" values)
bank_full$marital <- factor(bank_full$marital,
                            level = c("single", "divorced", "married"),
                            exclude = NULL)

# treat job as a factor from 1 to 12 (has "unknown" values)
bank_full$job <- factor(bank_full$job,
                            level = c("unknown", "unemployed", "student",
                                      "housemaid", "services", "blue-collar",
                                      "retired", "admin.", "self-employed",
                                      "technician", "management", "entrepreneur"),
                            exclude = NULL)

# treat poutcome as a factor from 1 to 4 (has "unknown" values)
bank_full$poutcome <- factor(bank_full$poutcome,
                            level = c("unknown", "failure", "other", "success"),
                            exclude = NULL)

# normalizing "balance" to handle dispersion
bank_full$balance <- (bank_full$balance - mean(bank_full$balance)) / sd(bank_full$balance)
```

Creating a Training Sample to fit the Model & a Test Sample to Test the Model

```
set.seed(100)
bank_full$test_train_indicator <- sample(c("Train", "Test"),
                                nrow(bank_full),
                                replace = T,
                                prob = c(0.8, 0.2))

train <- bank_full[bank_full$test_train_indicator == "Train", ]
test <- bank_full[bank_full$test_train_indicator == "Test", ]
```

Running a Logistic Regression Model on the Training Data

```
glmModel <- glm(y ~ ., data = train[c(-10, -11, -12, -18)] , family = "binomial")
summary(glmModel)
```

```
##
## Call:
## glm(formula = y ~ ., family = "binomial", data = train[c(-10,
##     -11, -12, -18)])
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9835  -0.5099  -0.3880  -0.2572   3.6906
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -2.1919338  0.2441077  -8.979  < 2e-16 ***
## age                 0.0020169  0.0021590   0.934  0.35021
## jobunemployed       0.1033344  0.2358298   0.438  0.66126
## jobstudent          0.5276251  0.2348782   2.246  0.02468 *
## jobhousemaid       -0.3098240  0.2470379  -1.254  0.20979
## jobservices        -0.1058084  0.2269784  -0.466  0.64110
## jobblue-collar     -0.1628168  0.2223572  -0.732  0.46403
## jobretired          0.5096113  0.2270338   2.245  0.02479 *
## jobadmin.           0.0744873  0.2233942   0.333  0.73881
## jobself-employed   -0.1143051  0.2361612  -0.484  0.62838
## jobtechnician      -0.1025033  0.2215319  -0.463  0.64358
## jobmanagement      -0.0607481  0.2217416  -0.274  0.78412
## jobentrepreneur    -0.1761428  0.2423150  -0.727  0.46728
## maritaldivorced    -0.1657315  0.0652650  -2.539  0.01111 *
## maritalmarried     -0.3310499  0.0448057  -7.389 1.48e-13 ***
## educationprimary   -0.2592240  0.1007680  -2.572  0.01010 *
## educationsecondary -0.1180975  0.0889720  -1.327  0.18439
## educationtertiary   0.0700475  0.0937141   0.747  0.45479
## default            -0.1840651  0.1607374  -1.145  0.25216
## balance             0.0710377  0.0144766   4.907 9.25e-07 ***
## housing            -0.5970222  0.0391516 -15.249  < 2e-16 ***
## loan               -0.4917558  0.0583790  -8.424  < 2e-16 ***
## contactcellular     0.9851458  0.0564062  17.465  < 2e-16 ***
## contacttelephone    0.8312768  0.0853955   9.734  < 2e-16 ***
## campaign           -0.1041623  0.0094036 -11.077  < 2e-16 ***
## pdays               0.0004741  0.0003082   1.538  0.12396
## previous            0.0082416  0.0060671   1.358  0.17434
## poutcomefailure    -0.0072819  0.0921834  -0.079  0.93704
## poutcomeother       0.3347953  0.1039682   3.220  0.00128 **
## poutcomesuccess     2.2858738  0.0863331  26.477  < 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: 26111  on 36247  degrees of freedom
## Residual deviance: 22426  on 36218  degrees of freedom
## AIC: 22486
##
## Number of Fisher Scoring iterations: 6
```

```r
# predict the data to identify the accuracy of the training sample
outcome <- predict(glmModel, train[c(-10, -11, -12, -18)], "response")
outcome <- ifelse(outcome > 0.5, 1, 0)

# add the outcome to training sample
train$outcome <- outcome
```

```r
# create a confusion matrix
confusionMatrix <- as.data.frame(table(train$outcome, train$y))
names(confusionMatrix) <- c("prediction", "True_value", "Count")
confusionMatrix
```

```
##   prediction True_value Count
## 1          0          0 31671
## 2          1          0   349
## 3          0          1  3515
## 4          1          1   713
```

```r
# identify the accuracy of the training model
accuracy <- sum(confusionMatrix[confusionMatrix$prediction == confusionMatrix$True_value, "Count"]) / su
accuracy
```

```
## [1] 0.893401
```

Running a Logistic Regression Model on the Test Data

```r
# predict the data to see the accuracy of the test sample
glmModel2 <- glm(y ~ ., data = test[c(-10, -11, -12, -18)] , family = "binomial")
summary(glmModel2)
```

```
##
## Call:
## glm(formula = y ~ ., family = "binomial", data = test[c(-10,
##     -11, -12, -18)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8937  -0.5142  -0.3956  -0.2647   3.0708
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -3.1242227  0.5512487  -5.668 1.45e-08 ***
## age                 0.0038443  0.0042948   0.895   0.3707
## jobunemployed       0.8927840  0.5425887   1.645   0.0999 .
## jobstudent          1.0659939  0.5476933   1.946   0.0516 .
## jobhousemaid        0.5425344  0.5592750   0.970   0.3320
## jobservices         0.6748439  0.5274250   1.280   0.2007
## jobblue-collar      0.7135138  0.5196613   1.373   0.1697
## jobretired          1.3591888  0.5279151   2.575   0.0100 *
## jobadmin.           0.7315287  0.5217795   1.402   0.1609
## jobself-employed    0.5705297  0.5482379   1.041   0.2980
## jobtechnician       0.5800536  0.5184866   1.119   0.2632
## jobmanagement       0.6591348  0.5175784   1.273   0.2028
## jobentrepreneur     0.1904746  0.5655274   0.337   0.7363
## maritaldivorced    -0.0974298  0.1273076  -0.765   0.4441
## maritalmarried     -0.3770166  0.0889189  -4.240 2.24e-05 ***
## educationprimary   -0.0657552  0.2085814  -0.315   0.7526
```

```
## educationsecondary  0.0164231  0.1861873   0.088   0.9297
## educationtertiary   0.2436430  0.1939148   1.256   0.2090
## default            -0.5399865  0.3720990  -1.451   0.1467
## balance             0.0301619  0.0296797   1.016   0.3095
## housing            -0.5150601  0.0775362  -6.643 3.08e-11 ***
## loan               -0.4590508  0.1137076  -4.037 5.41e-05 ***
## contactcellular     1.0127368  0.1114494   9.087  < 2e-16 ***
## contacttelephone    0.8285380  0.1717205   4.825 1.40e-06 ***
## campaign           -0.1187799  0.0196715  -6.038 1.56e-09 ***
## pdays              -0.0004915  0.0005989  -0.821   0.4118
## previous            0.0368374  0.0185080   1.990   0.0466 *
## poutcomefailure     0.1371646  0.1832494   0.749   0.4542
## poutcomeother       0.3495852  0.2152432   1.624   0.1043
## poutcomesuccess     2.2802104  0.1755393  12.990  < 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: 6519.3  on 8962  degrees of freedom
## Residual deviance: 5622.9  on 8933  degrees of freedom
## AIC: 5682.9
##
## Number of Fisher Scoring iterations: 6
```

```
# predict the data to identify the accuracy of the test sample
outcome <- predict(glmModel2, test[c(-10, -11, -12, -18)], "response")
outcome <- ifelse(outcome > 0.5, 1, 0)

# add the outcome to test sample
test$outcome <- outcome

# create a confusion matrix
confusionMatrix <- as.data.frame(table(test$outcome, test$y))
names(confusionMatrix) <- c("prediction", "True_value", "Count")
confusionMatrix
```

```
##   prediction True_value Count
## 1          0          0  7813
## 2          1          0    89
## 3          0          1   878
## 4          1          1   183
```

```
# identify the accuracy of the test model
accuracy <- sum(confusionMatrix[confusionMatrix$prediction == confusionMatrix$True_value, "Count"]) / su
  accuracy
```

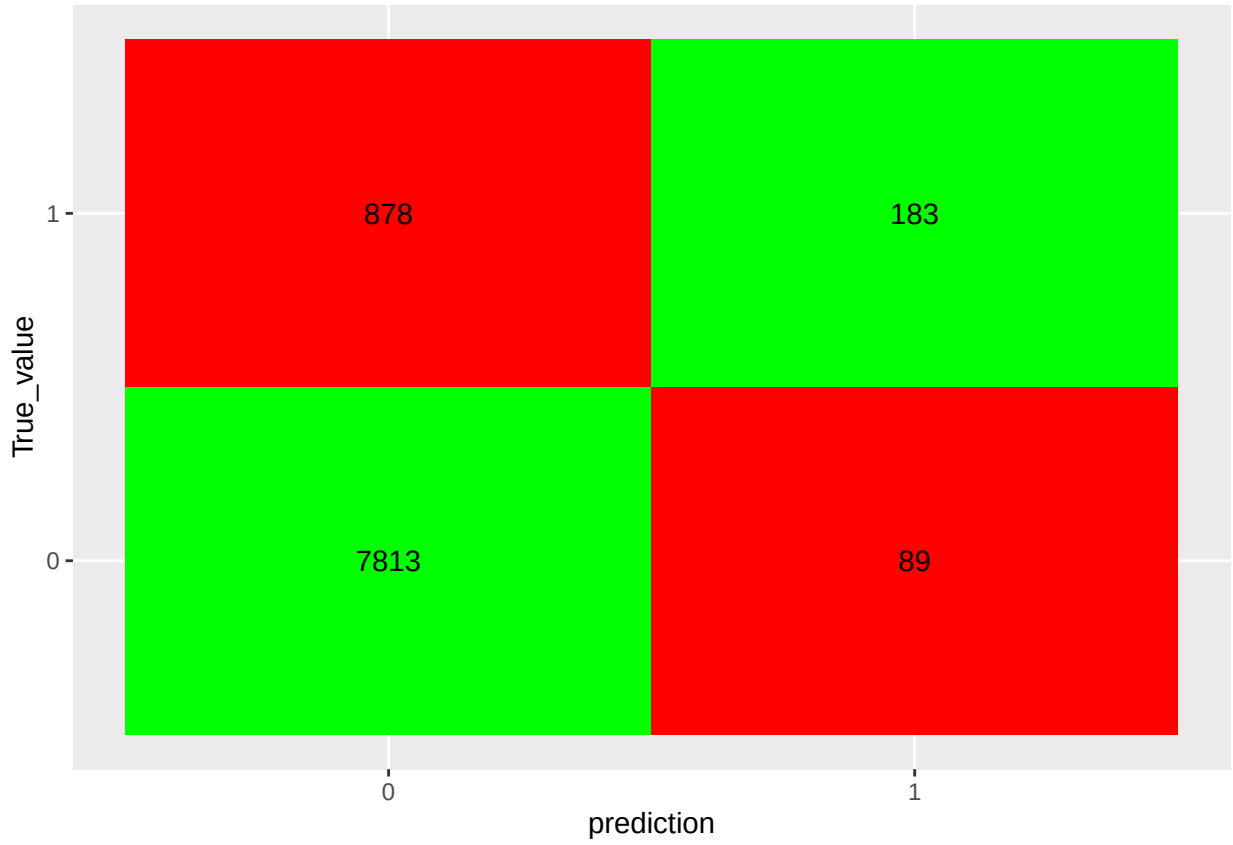
```
## [1] 0.892112
```

Creating a Heat Map as a Tile Plot

```r
# heat..
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```r
ggplot(data = confusionMatrix) +
  geom_tile(aes(x = prediction, y = True_value), fill = c("green", "red", "red", "green")) +
  geom_text(aes(x = prediction, y = True_value, label = Count))
```



Conclusion

- The training sample demonstrates around 89.34% accuracy
- The test sample demonstrates 89.21% accuracy
- The number of correct classifications (32384) > the number of incorrect classifications (3864)

Honors Pledge:

As a student of the Dr. Robert B. Pamplin Jr. School of Business I have read and strive to uphold the University's Code of Academic Integrity and promote ethical behavior. In doing so, I pledge on my honor that I have not given, received, or used any unauthorized materials or assistance on this examination or assignment. I further pledge that I have not engaged in cheating, forgery, or plagiarism and I have cited all appropriate sources.

Student Signature: Areej Mulla