Build and Evaluate Logistic Regression Model

Dr. Nishat Mohammad

03/02/2024

1. Load data:

Download the data set on student achievementLinks to an external site. in secondary education math education of two Portuguese schools (use the data set Students Math).

```
# load student.mat file
path= "./student_performance/student/student-mat.csv"
student_math <- read.table(path, sep=";",header=TRUE)

any(is.na(student_math))
```

```
## [1] FALSE
```

```
str(student_math)
```

```
## 'data.frame':    395 obs. of  33 variables:
##  $ school    : chr  "GP" "GP" "GP" "GP" ...
##  $ sex       : chr  "F" "F" "F" "F" ...
##  $ age       : int  18 17 15 15 16 16 16 17 15 15 ...
##  $ address   : chr  "U" "U" "U" "U" ...
##  $ famsize   : chr  "GT3" "GT3" "LE3" "GT3" ...
##  $ Pstatus   : chr  "A" "T" "T" "T" ...
##  $ Medu      : int  4 1 1 4 3 4 2 4 3 3 ...
##  $ Fedu      : int  4 1 1 2 3 3 2 4 2 4 ...
##  $ Mjob      : chr  "at_home" "at_home" "at_home" "health" ...
##  $ Fjob      : chr  "teacher" "other" "other" "services" ...
##  $ reason    : chr  "course" "course" "other" "home" ...
##  $ guardian  : chr  "mother" "father" "mother" "mother" ...
##  $ traveltime: int  2 1 1 1 1 1 1 2 1 1 ...
##  $ studytime : int  2 2 2 3 2 2 2 2 2 2 ...
##  $ failures  : int  0 0 3 0 0 0 0 0 0 0 ...
##  $ schoolsup : chr  "yes" "no" "yes" "no" ...
##  $ famsup    : chr  "no" "yes" "no" "yes" ...
##  $ paid      : chr  "no" "no" "yes" "yes" ...
##  $ activities: chr  "no" "no" "no" "yes" ...
##  $ nursery   : chr  "yes" "no" "yes" "yes" ...
##  $ higher    : chr  "yes" "yes" "yes" "yes" ...
##  $ internet  : chr  "no" "yes" "yes" "yes" ...
##  $ romantic  : chr  "no" "no" "no" "yes" ...
##  $ famrel    : int  4 5 4 3 4 5 4 4 4 5 ...
##  $ freetime  : int  3 3 3 2 3 4 4 1 2 5 ...
##  $ goout     : int  4 3 2 2 2 2 4 4 2 1 ...
##  $ Dalc      : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ Walc      : int  1 1 3 1 2 2 1 1 1 1 ...
##  $ health    : int  3 3 3 5 5 5 3 1 1 5 ...
##  $ absences  : int  6 4 10 2 4 10 0 6 0 0 ...
##  $ G1        : int  5 5 7 15 6 15 12 6 16 14 ...
##  $ G2        : int  6 5 8 14 10 15 12 5 18 15 ...
##  $ G3        : int  6 6 10 15 10 15 11 6 19 15 ...
```

2. Pass Fail Column:

Add another column, PF – pass-fail. Mark any student whose final grade is less than 10 as F, otherwise as P and then build a dummy code variable for that new column. Use the new dummy variable column as the response variable with 0 for F and 1 for P.

```r
# Create PF column
student_math$PF <- ifelse(student_math$G3 < 10, "F", "P")

# Encoding
student_math$PF <- ifelse(student_math$PF=="P", 1, 0)
head(student_math$PF)
```

```
## [1] 0 0 1 1 1 1
```

3. Binomial Regression Model:

Build a binomial logistic regression model classifying a student as passing or failing. Eliminate any non-significant variable using an elimination approach of your choice. Use as many features as you like but you must use at least four – choose the ones you believe are most useful.

```r
# Create the binomial regression model
br_model <- glm(PF~age+absences+G1+G2, data=student_math, family = binomial)
summary(br_model)
```

```
##
## Call:
## glm(formula = PF ~ age + absences + G1 + G2, family = binomial,
##     data = student_math)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.71869  -0.05390   0.00726   0.15243   2.26064
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -11.29496    3.57688  -3.158  0.00159 **
## age          -0.35903    0.18201  -1.973  0.04854 *
## absences     -0.02619    0.02421  -1.082  0.27936
## G1            0.27162    0.15267   1.779  0.07522 .
## G2            1.64533    0.26317   6.252 4.05e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 500.50  on 394  degrees of freedom
## Residual deviance: 141.22  on 390  degrees of freedom
## AIC: 151.22
##
## Number of Fisher Scoring iterations: 8
```

This model shows that the absences is least significant, so let us take it off and make a new model without it.

```
# Model without absences
br_model2 <- glm(PF~age+G1+G2, data=student_math, family = binomial)
summary(br_model2)
```

```
##
## Call:
## glm(formula = PF ~ age + G1 + G2, family = binomial, data = student_math)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.70467  -0.05899   0.00732   0.14759   2.33247
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.8744     3.5493  -3.064  0.00219 **
## age          -0.3945     0.1805  -2.185  0.02886 *
## G1            0.2519     0.1523   1.654  0.09808 .
## G2            1.6635     0.2634   6.315  2.7e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 500.5  on 394  degrees of freedom
## Residual deviance: 142.6  on 391  degrees of freedom
## AIC: 150.6
##
## Number of Fisher Scoring iterations: 8
```

Here we can see the G1 has a high p value, so it should be removed.

```
# Take off G1
br_model3 <- glm(PF~age+G2, data=student_math, family = binomial)
summary(br_model3)
```

```
##
## Call:
## glm(formula = PF ~ age + G2, family = binomial, data = student_math)
```

```
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.71828  -0.06404   0.01210   0.14149   2.26721
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -11.7677     3.4759  -3.386  0.00071 ***
## age          -0.3064     0.1669  -1.835  0.06644 .
## G2            1.8490     0.2433   7.601 2.95e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 500.5  on 394  degrees of freedom
## Residual deviance: 145.4  on 392  degrees of freedom
## AIC: 151.4
##
## Number of Fisher Scoring iterations: 8
```

Now the p-value of age is higher than it was in the earlier model. The Akaike Information criteria increased here compared to the second model which has the lowest AIC. we may consider this to be the best model among the three but using the Chi-square test will be wise at this point.

4. Model Equation:

State the regression equation.

```
# Compare all the model 1 and 2.

anova_mod1_2 <- anova(br_model, br_model2, test="Chisq")
anova_mod1_2
```

```
## Analysis of Deviance Table
##
## Model 1: PF ~ age + absences + G1 + G2
## Model 2: PF ~ age + G1 + G2
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       390     141.22
## 2       391     142.60 -1  -1.3824   0.2397
```

```
summary(br_model2)
```

```
##
## Call:
## glm(formula = PF ~ age + G1 + G2, family = binomial, data = student_math)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.70467  -0.05899   0.00732   0.14759   2.33247
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.8744     3.5493  -3.064  0.00219 **
## age          -0.3945     0.1805  -2.185  0.02886 *
## G1            0.2519     0.1523   1.654  0.09808 .
## G2            1.6635     0.2634   6.315  2.7e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 500.5  on 394  degrees of freedom
## Residual deviance: 142.6  on 391  degrees of freedom
## AIC: 150.6
##
## Number of Fisher Scoring iterations: 8
```

The equation for the best model which is the second model is:

Model 2: PF ~ age + G1 + G2.

Looking at the log odds of probability of pass-fail.

$\mathbf{logit(P(PF))} = -10.8744$ -0.3945 × age + (0.2519 * G1) + (1.6635 * G2).

$\mathbf{logit(P(PF))}$ is the log odds of the prob of PF, all the predictor variables stand as multiples of their coefficients form the above summary of the model.

5. Model Accuracy:

```
br_model2_prediction <- predict(br_model2, student_math[c(c("age", "G1", "G2"))], type = 'response')

br_model2_prediction <-  ifelse(br_model2_prediction>0.5, 1, 0)


err <- mean(br_model2_prediction != student_math$PF)
err
```

```
## [1] 0.08101266
```

```
accuracy <- 1 - err
accuracy
```

```
## [1] 0.9189873
```

The accuracy of the second model is 0.9189873.

6. KNN Model:

Predict the variable using kNN (from a package or implementation of your choice) and calculate the accuracy. Compare the accuracy with that of your logistic regression model.

```
# Create train and dummy test sets
student_math$PF <- ifelse(student_math$PF==1, "Pass", "Fail")
tranin_set <- student_math[c("age", "G1", "G2", "PF")]
test <- data.frame(age= c(16,17), G1=c(15,16), G2=c(14,14), PF = c("Pass","Pass"))

# labels
train_labs<- tranin_set$PF
test_labs<- test$PF

# Remove PF form train and test
train<- tranin_set[, 1:3]
test_set <- test[, 1:3]


# knn model
library(class)
knn_model1 <- knn(train = train , test = test_set , cl= tranin_set$PF, k = 3,prob=TRUE)
knn_model1
```

```
## [1] Pass Pass
## attr(,"prob")
## [1] 1 1
## Levels: Fail Pass
```

```
#library(gmodels)
#Xtable_knn_model1 <- CrossTable(x = test_labs,
#                                y = knn_model1,
#                                prop.test= TRUE)

#stats::fisher.test(test_labs, knn_model1)

# Check accuracy based on correct outcomes:
knn_model1
```

```
## [1] Pass Pass
## attr(,"prob")
## [1] 1 1
## Levels: Fail Pass
```

```
test_labs
```

```
## [1] "Pass" "Pass"
```

```
acc <- mean(knn_model1 == test_labs)
acc
```

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

I have created a dummy test model and used the training set I used in Q5 as instructed on Teams Chat. I then just decided to get the mean of the outcomes and since both actual cases were pass and both predicted cases were fail, the accuracy of this model is 0.

7. Difference between the Logistic regression and Knn alogorithms:

What is the difference between the two algorithms? How would you choose which one to use?

Answer:

Based on the accuracy, the Logistic regression model is far more accurate and will be the better over knn model.

The differences between Logistic and Regression algos are clarified below:

	knn algos	Logistic Regression algos
Variables	variables should have complex relationships	variables should have linear relationship
Interpretation	Not easy to interpret nearest neighbors.	easier to interpret
Computing cost	Expensive	Faster so less expensive
Outliers	Sensitive to outliers	Minimal impact from outliers with larger data sets
Scaling	Poor scaling	Scales well with large data sets