Blog 2: Multinomial Logistic Regression

Eric Lehmphul

5/22/2022

```
library(tidyverse)
library(skimr)
library(nnet)
library(caret)
```

Introduction

Logistic Regression is typically used for binary outcome variables. What can be done if the outcome variable is not binary? There is multinomial logistic regression for non-ordered categorical variables with 3 or more classes and ordinal logistic regression for ordered categorical variables with 3 or more classes. This blog will run through the creation of a multinomial logistic regression model.

Dataset

A dataset that is provided via the data() function in base R and has more than 2 classes in the target variable in the iris dataset.

```
data(iris)

head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Descriptive Statistics

The dataset contains no missing value. The target class is also balanced as each class level has 50 records.

```
summary_table <- skim_with(numeric = sfl(median = ~ median(., na.rm = TRUE),
                                         min = ~ min(., na.rm = TRUE),
                                         max = ~ max(., na.rm = TRUE),
```

```
                                          hist = NULL, p0 = NULL, p25 = NULL,
                                          p50 = NULL, p75 = NULL, p100 = NULL))

summary_table(iris)
```

Table 1: Data summary

Name	iris
Number of rows	150
Number of columns	5
Column type frequency:	
factor	1
numeric	4
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
Species	0	1	FALSE	3	set: 50, ver: 50, vir: 50

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	median	min	max
Sepal.Length	0	1	5.84	0.83	5.80	4.3	7.9
Sepal.Width	0	1	3.06	0.44	3.00	2.0	4.4
Petal.Length	0	1	3.76	1.77	4.35	1.0	6.9
Petal.Width	0	1	1.20	0.76	1.30	0.1	2.5

Create Multinomial Regression Model

Create train and test set

The data needs to be separated into a train and test set to be able to assess model results. 70% was set to be training and 30% was set to testing.

```
index <- createDataPartition(iris$Species, p = .70, list = FALSE)
train <- iris[index,]
test <- iris[-index,]
```

The distributions are even for both the training and testing set

```
table(train$Species)
```

```
##
##     setosa versicolor  virginica
##         35         35         35
```

```r
table(test$Species)
```

```
##
##     setosa versicolor  virginica
##         15         15         15
```

Set Reference Level for Model

Multinomial Logistic Regression requires that a reference level be defined.

```r
train$Species <- relevel(train$Species, ref = "setosa")
```

Train Model

The package **nnet** has the function multinom() which is used to create multinomial logistic regression.

```r
multinomial.model <- multinom(Species ~ ., data = train)
```

```
## # weights:  18 (10 variable)
## initial  value 115.354290
## iter  10 value 12.311620
## iter  20 value 2.414714
## iter  30 value 1.811380
## iter  40 value 1.549745
## iter  50 value 1.429857
## iter  60 value 1.310272
## iter  70 value 0.771701
## iter  80 value 0.648464
## iter  90 value 0.476405
## iter 100 value 0.447820
## final  value 0.447820
## stopped after 100 iterations
```

```r
summary(multinomial.model)
```

```
## Call:
## multinom(formula = Species ~ ., data = train)
##
## Coefficients:
##            (Intercept) Sepal.Length Sepal.Width Petal.Length Petal.Width
## versicolor    71.66268    -16.93856   -24.31514     38.99913    1.737102
## virginica    -99.05206    -45.60330   -57.32128    108.73189   61.914110
##
## Std. Errors:
##            (Intercept) Sepal.Length Sepal.Width Petal.Length Petal.Width
## versicolor    198.7527     60.23172    54.98849     134.0902    87.09377
## virginica     180.6197    158.58605   106.31834     104.5889    80.26973
##
## Residual Deviance: 0.8956392
## AIC: 20.89564
```

Assess Model Performance

The model performance can be tested in the same manor as a binary logistic regression model. First predictions need to be generated using the model, then a confusion matrix can be created and the performance metrics can be calculated.

```
test$SpeciesPredicted <- predict(multinomial.model, newdata = test)


tab <- table(test$Species, test$SpeciesPredicted)

confusionMatrix(tab)
```

```
## Confusion Matrix and Statistics
##
##
##               setosa versicolor virginica
##   setosa          15          0         0
##   versicolor       0         14         1
##   virginica        0          0        15
##
## Overall Statistics
##
##                Accuracy : 0.9778
##                  95% CI : (0.8823, 0.9994)
##     No Information Rate : 0.3556
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9667
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                 1.0000            1.0000           0.9375
## Specificity                 1.0000            0.9677           1.0000
## Pos Pred Value              1.0000            0.9333           1.0000
## Neg Pred Value              1.0000            1.0000           0.9667
## Prevalence                  0.3333            0.3111           0.3556
## Detection Rate              0.3333            0.3111           0.3333
## Detection Prevalence        0.3333            0.3333           0.3333
## Balanced Accuracy           1.0000            0.9839           0.9688
```