Assignment9.2_LincolnBrown

Lincoln Brown

8/7/2021

```
library(farff)
library(ggplot2)
library(coefplot)
library(rcompanion)
library(scales)
thoracic <- readARFF("/media/x/disk/School/DSC520/Datasets/ThoracicSurgery.arff")
```

```
## Parse with reader=readr : /media/x/disk/School/DSC520/Datasets/ThoracicSurgery.arff
```

```
## Loading required package: readr
```

```
##
## Attaching package: 'readr'
```

```
## The following object is masked from 'package:scales':
##
##     col_factor
```

```
## header: 0.006000; preproc: 0.000000; data: 0.021000; postproc: 0.001000; total: 0.028000
```

```
dim(thoracic)
```

```
## [1] 470  17
```

```
colnames(thoracic)
```

```
##  [1] "DGN"     "PRE4"    "PRE5"    "PRE6"    "PRE7"    "PRE8"    "PRE9"
##  [8] "PRE10"   "PRE11"   "PRE14"   "PRE17"   "PRE19"   "PRE25"   "PRE30"
## [15] "PRE32"   "AGE"     "Risk1Yr"
```

Create training and test samples

```
#Total # of rows in data set
n <- nrow(thoracic)

# Set 80% of the rows for training sample
n_train <- round(0.80 * n)

# Create a vector of indices which is an 80% random sample
set.seed(1)
train_indices <- sample(1:n, n_train)

# Subset the data frame to training indices only
train <- thoracic[train_indices,]

#Exclude the training indices for test set
test <- thoracic[-train_indices,]
```

Check the dimensions:

```
paste("train sample size: ", nrow(train))
```

```
## [1] "train sample size:  376"
```

```
paste("test sample size: ", nrow(test))
```

```
## [1] "test sample size:  94"
```

1. Fit a binary logistic regression model to the data set that predicts whether or not the patient survived for one year (the Risk1Y variable) after the surgery. Use the glm() function to perform the logistic regression. See Generalized Linear Models for an example. Include a summary using the summary() function in yor results.

I am using plus notation to create generalized linear model with all of the variables in the data set.

```
thoracic_mdl <- glm(Risk1Yr ~., data=thoracic, family="binomial")
summary(thoracic_mdl)
```

```
##
## Call:
## glm(formula = Risk1Yr ~ ., family = "binomial", data = thoracic)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4929   0.2762   0.4199   0.5439   1.6084
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.604e+01  2.333e+03   0.011 0.991093
## DGNDGN2     -5.557e-01  4.128e-01  -1.346 0.178199
## DGNDGN4     -4.278e-01  4.733e-01  -0.904 0.366122
## DGNDGN6      1.377e+01  1.178e+03   0.012 0.990671
## DGNDGN5     -2.201e+00  6.113e-01  -3.600 0.000318 ***
## DGNDGN8     -3.852e+00  1.550e+00  -2.485 0.012959 *
## DGNDGN1      1.418e+01  2.400e+03   0.006 0.995285
## PRE4         2.272e-01  1.849e-01   1.229 0.219094
## PRE5         3.030e-02  1.786e-02   1.697 0.089715 .
## PRE6PRZ1     1.490e-01  5.783e-01   0.258 0.796647
## PRE6PRZ0    -2.937e-01  7.907e-01  -0.371 0.710303
## PRE7F        7.153e-01  5.556e-01   1.288 0.197884
## PRE8F        1.743e-01  3.892e-01   0.448 0.654188
## PRE9F        1.368e+00  4.868e-01   2.811 0.004942 **
## PRE10F       5.770e-01  4.826e-01   1.196 0.231855
## PRE11F       5.162e-01  3.965e-01   1.302 0.192948
## PRE14OC14   -1.653e+00  6.094e-01  -2.713 0.006675 **
## PRE14OC12   -4.394e-01  3.301e-01  -1.331 0.183177
## PRE14OC13   -1.179e+00  6.165e-01  -1.913 0.055799 .
## PRE17F       9.266e-01  4.445e-01   2.085 0.037092 *
## PRE19F      -1.466e+01  1.654e+03  -0.009 0.992928
## PRE25F      -9.789e-02  1.003e+00  -0.098 0.922273
## PRE30F       1.084e+00  4.990e-01   2.172 0.029840 *
## PRE32F      -1.398e+01  1.645e+03  -0.008 0.993219
## AGE          9.506e-03  1.810e-02   0.525 0.599442
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 395.61  on 469  degrees of freedom
## Residual deviance: 341.19  on 445  degrees of freedom
## AIC: 391.19
##
## Number of Fisher Scoring iterations: 15
```

Create a model of the sampled data

```
train_mdl <- glm(Risk1Yr ~ ., data=train, family="binomial")
summary(train_mdl)
```

```
##
## Call:
## glm(formula = Risk1Yr ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6636   0.2547   0.3969   0.5020   1.3637
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.474e+01  3.393e+03   0.007 0.994183
## DGNDGN2     -7.190e-01  4.850e-01  -1.482 0.138214
## DGNDGN4     -4.038e-01  6.112e-01  -0.661 0.508865
## DGNDGN6      1.369e+01  1.180e+03   0.012 0.990747
## DGNDGN5     -2.194e+00  6.557e-01  -3.345 0.000822 ***
## DGNDGN8     -2.086e+01  2.400e+03  -0.009 0.993065
## DGNDGN1      1.365e+01  2.400e+03   0.006 0.995462
## PRE4         1.656e-01  2.192e-01   0.755 0.450007
## PRE5         2.784e-02  1.823e-02   1.528 0.126623
## PRE6PRZ1    -1.636e-02  7.024e-01  -0.023 0.981424
## PRE6PRZ0    -3.318e-01  9.493e-01  -0.350 0.726683
## PRE7F        7.414e-01  6.080e-01   1.219 0.222681
## PRE8F       -3.703e-01  4.998e-01  -0.741 0.458761
## PRE9F        1.790e+00  5.789e-01   3.092 0.001986 **
## PRE10F       9.275e-01  5.859e-01   1.583 0.113429
## PRE11F       5.227e-01  4.570e-01   1.144 0.252668
## PRE14OC14   -1.192e+00  6.927e-01  -1.720 0.085370 .
## PRE14OC12   -2.157e-01  3.821e-01  -0.564 0.572421
## PRE14OC13   -1.640e+00  6.579e-01  -2.494 0.012647 *
## PRE17F       1.054e+00  5.054e-01   2.084 0.037126 *
## PRE19F      -1.413e+01  2.400e+03  -0.006 0.995300
## PRE25F       1.583e-01  1.027e+00   0.154 0.877482
## PRE30F       1.094e+00  6.060e-01   1.805 0.071034 .
## PRE32F      -1.325e+01  2.400e+03  -0.006 0.995595
## AGE          9.307e-03  2.161e-02   0.431 0.666667
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 305.84  on 375  degrees of freedom
## Residual deviance: 253.98  on 351  degrees of freedom
## AIC: 303.98
##
## Number of Fisher Scoring iterations: 15
```

2. According to the summary, which variables had the greatest effect on the survival rate?

In the thoracic_mdl: DGNDGN5 had the smallest p-value with the next two most significant p-values being the variables PRE9F and PRE14OC14. Other potentially significant p-values were PRE17F and PRE30F.

- DGNDGN5 is the diagnosis and relates to multiple tumors.
- PRE9F is shortness of breath before surgery
- PRE14OC14 is the size of the original tumor

3. To compute the accuracy of your model, use the dataset to predict the outcome variable. The percent of correct predictions is the accuracy of your model. What is the accuracy of your model?

Find predictions on unsplit data set

```
pred <- predict(thoracic_mdl, type="response")
predicted <- round(pred)
conf_matrix_thor <- table(Predicted = predicted, Reference=thoracic$Risk1Yr)
accuracy_thor <- (conf_matrix_thor[1,1] + conf_matrix_thor[2,2]) / nrow(thoracic)

cat("The accuracy of the model without any sampling performed is:", percent(accuracy_thor))
```

```
## The accuracy of the model without any sampling performed is: 84%
```

Find predictions on test data

```
test_pred <- predict(train_mdl, test, type="response")
test_predicted <- round(test_pred)
conf_matrix_test <- table(Predicted = test_predicted, Reference=test$Risk1Yr)
conf_matrix_test
```

```
##          Reference
## Predicted  T  F
##         0  0  3
##         1 17 74
```

```
accuracy_test <- (conf_matrix_test[1,1] + conf_matrix_test[2,2]) / nrow(test)
cat("The accuracy of the model with sampling is:", percent(accuracy_test))
```

```
## The accuracy of the model with sampling is: 79%
```

Part 2

1. Fit a logistic regression model to the binary-classifier-data.csv dataset The dataset (found in binary-classifier-data.csv) contains three variables; label, x, and y. The label variable is either 0 or 1 and is the output we want to predict using the x and y variables.

```
binary <- read.csv("/media/x/disk/School/DSC520/Datasets/binary-classifier-data.csv")
dim(binary)
```

```
## [1] 1498    3
```

Create sample data

```
#Total # of rows in data set
n_bin <- nrow(binary)

# Set 80% of the rows for training sample
n_bin_train <- round(0.80 * n_bin)

# Create a vector of indices which is an 80% random sample
set.seed(1)
bin_train_indices <- sample(1:n_bin, n_bin_train)

# Subset the data frame to training indices only
bin_train <- binary[bin_train_indices,]

#Exclude the training indices for test set
bin_test <- binary[-bin_train_indices,]
```

Check the dimensions:

```
paste("train sample size: ", nrow(bin_train))
```

```
## [1] "train sample size:  1198"
```

```
paste("test sample size: ", nrow(bin_test))
```

```
## [1] "test sample size:  300"
```

Model for unsampled data

```
bin_mdl <- glm(label ~ x + y, data=binary, family=binomial(link="logit"))

summary(bin_mdl)
```

```
##
## Call:
## glm(formula = label ~ x + y, family = binomial(link = "logit"),
##     data = binary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3728  -1.1697  -0.9575   1.1646   1.3989
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.424809   0.117224   3.624  0.00029 ***
```

```
## x           -0.002571   0.001823  -1.411  0.15836
## y           -0.007956   0.001869  -4.257 2.07e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2075.8  on 1497  degrees of freedom
## Residual deviance: 2052.1  on 1495  degrees of freedom
## AIC: 2058.1
##
## Number of Fisher Scoring iterations: 4
```

y has a significant p-value, whereas x does not.

Model for sampled data:

```
bin_train_mdl <- glm(label~., data=bin_train, family="binomial")
summary(bin_train_mdl)
```

```
##
## Call:
## glm(formula = label ~ ., family = "binomial", data = bin_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3667  -1.1648  -0.9606   1.1661   1.3910
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.410894   0.131366   3.128 0.001761 **
## x           -0.002318   0.002057  -1.127 0.259704
## y           -0.007990   0.002089  -3.824 0.000131 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1660.0  on 1197  degrees of freedom
## Residual deviance: 1641.6  on 1195  degrees of freedom
## AIC: 1647.6
##
## Number of Fisher Scoring iterations: 4
```

2. What is the accuracy of the logistic regression classifier?

Find predictions on unsample data:

```
bin_pred <- predict(bin_mdl, type="response")
bin_predicted <- round(bin_pred)
bin_conf_matrix <- table(Predicted = bin_predicted, Reference=binary$label)

accuracy_bin <- (bin_conf_matrix[1,1] + bin_conf_matrix[2,2]) / nrow(binary)

cat("The accuracy of the binary model without any sampling performed is:", percent(accuracy_bin))
```

```
## The accuracy of the binary model without any sampling performed is: 58%
```

Find predictions on sampled data:

```
test_bin_pred <- predict(bin_train_mdl, bin_test, type="response")

test_bin_predicted <- round(test_bin_pred)
test_bin_conf_matrix <- table(Predicted=test_bin_predicted, Reference=bin_test$label)

test_accuracy_bin <- (test_bin_conf_matrix[1,1] + test_bin_conf_matrix[2,2]) / nrow(bin_test)

cat("The accuracy of the bin model with sampling is:", percent(test_accuracy_bin))
```

```
## The accuracy of the bin model with sampling is: 55%
```

3. Keep this assignment handy, as you will be comparing your results from this week to next week.