hwk3 pt4

2024-12-05

```r
library(nnet)
```

```r
data = read.csv("/Users/juyiyang/Desktop/diabetic.csv")
```

```r
# a)
data$TotalRiskFactors = ifelse(data$TotalRiskFactors == 0, "none",
                               ifelse(data$TotalRiskFactors == 1, "one", "2 or more"))

# b)
data$Diabetes.new = ifelse(data$Diabetes.new == 0, "no", "yes")

# c)
data$SmokingStatus_NISTCode = factor(data$SmokingStatus_NISTCode,
                                     levels = c("false", "former smokers", "smokers"))

# d)
data$age.new = factor(data$age.new, levels = c("low", "medium", "high"))
```

```r
# e)
smoking_vs_diabetes = table(data$SmokingStatus_NISTCode, data$Diabetes.new)
print(smoking_vs_diabetes)
```

```
##
##                  no yes
##   false           0   0
##   former smokers  0   0
##   smokers         0   0
```

```r
age_vs_diabetes = table(data$age.new, data$Diabetes.new)
print(age_vs_diabetes)
```

```
##
##            no  yes
##   low    3089  155
##   medium 2679  631
##   high   2223  960
```

```r
totalrisk_vs_diabetes = table(data$TotalRiskFactors, data$Diabetes.new)
print(totalrisk_vs_diabetes)
```

```
##
```

```
##                no  yes
##   2 or more 1348  520
##   none      4198  312
##   one       2597  973
```

```
# 2)
valid_columns = c('TotalRiskFactors', 'Diabetes.new', 'HypertensionDX', 'SmokingStatus_NISTCode')
valid_columns = valid_columns[valid_columns %in% colnames(data)]

if (length(valid_columns) == 0) {
  stop('None of the specified predictor columns exist in the dataset.')
}

levels_info = sapply(data[, valid_columns, drop = FALSE], function(x) length(unique(x)))

print(levels_info)
```

```
##       TotalRiskFactors           Diabetes.new         HypertensionDX
##                      3                      2                      2
## SmokingStatus_NISTCode
##                      1
```

```
for (col in names(levels_info)) {
  if (levels_info[col] == 1) {
    cat(sprintf( col))
    data[[col]] = NULL
  }
}
```

```
## SmokingStatus_NISTCode
```

```
predictors = valid_columns[valid_columns %in% colnames(data)]
formula = as.formula(paste('age.new ~', paste(predictors, collapse = ' + ')))
model = multinom(formula, data = data)
```

```
## # weights:  18 (10 variable)
## initial  value 10697.187855
## iter  10 value 9515.743878
## final  value 9470.056418
## converged
```

```
# 3)
coef_summary = summary(model)$coefficients
std_errors = summary(model)$standard.errors
odds_ratios = exp(coef_summary)
ci_lower = exp(coef_summary - 1.96 * std_errors)
ci_upper = exp(coef_summary + 1.96 * std_errors)

results = data.frame(
  Predictor = rownames(coef_summary),
  Odds_Ratio = odds_ratios,
```

```
  CI_Lower = ci_lower,
  CI_Upper = ci_upper
)
print(results)
```

```
##        Predictor Odds_Ratio..Intercept. Odds_Ratio.TotalRiskFactorsnone
## medium    medium              0.8240460                       0.5956280
## high        high              0.5862941                       0.4033804
##        Odds_Ratio.TotalRiskFactorsone Odds_Ratio.Diabetes.newyes
## medium                       1.094290                   2.755196
## high                         1.000277                   3.851389
##        Odds_Ratio.HypertensionDXyes CI_Lower..Intercept.
## medium                     3.662891            0.7169142
## high                       7.491365            0.5071300
##        CI_Lower.TotalRiskFactorsnone CI_Lower.TotalRiskFactorsone
## medium                     0.5124891                    0.9311551
## high                       0.3443756                    0.8486022
##        CI_Lower.Diabetes.newyes CI_Lower.HypertensionDXyes CI_Upper..Intercept.
## medium                 2.274147                   3.238113            0.9471871
## high                   3.180607                   6.598860            0.6778157
##        CI_Upper.TotalRiskFactorsnone CI_Upper.TotalRiskFactorsone
## medium                     0.6922542                     1.286005
## high                       0.4724950                     1.179062
##        CI_Upper.Diabetes.newyes CI_Upper.HypertensionDXyes
## medium                 3.338001                   4.143392
## high                   4.663637                   8.504582
```

```
z_values = coef_summary / std_errors
p_values = 2 * (1 - pnorm(abs(z_values)))
z_values
```

```
##        (Intercept) TotalRiskFactorsnone TotalRiskFactorsone Diabetes.newyes
## medium   -2.723602            -6.755178         1.093980708        10.35231
## high     -7.214625           -11.251773         0.003303448        13.81116
##        HypertensionDXyes
## medium          20.64362
## high            31.11411
```

```
p_values
```

```
##         (Intercept) TotalRiskFactorsnone TotalRiskFactorsone Diabetes.newyes
## medium 6.457426e-03         1.426614e-11           0.2739634               0
## high   5.409007e-13         0.000000e+00           0.9973642               0
##        HypertensionDXyes
## medium                 0
## high                   0
```

#5 a) The odds of high risk factor Interpretation: The rows TotalRiskFactorsone and TotalRiskFactors2 or more provide the odds for individuals with one risk factor and two or more risk factors, respectively. For one risk factor, the odds are 2.61, meaning individuals with one risk factor are 2.61 times as likely to be classified as high risk compared to those without any risk factors. For two or more risk factors, the odds are

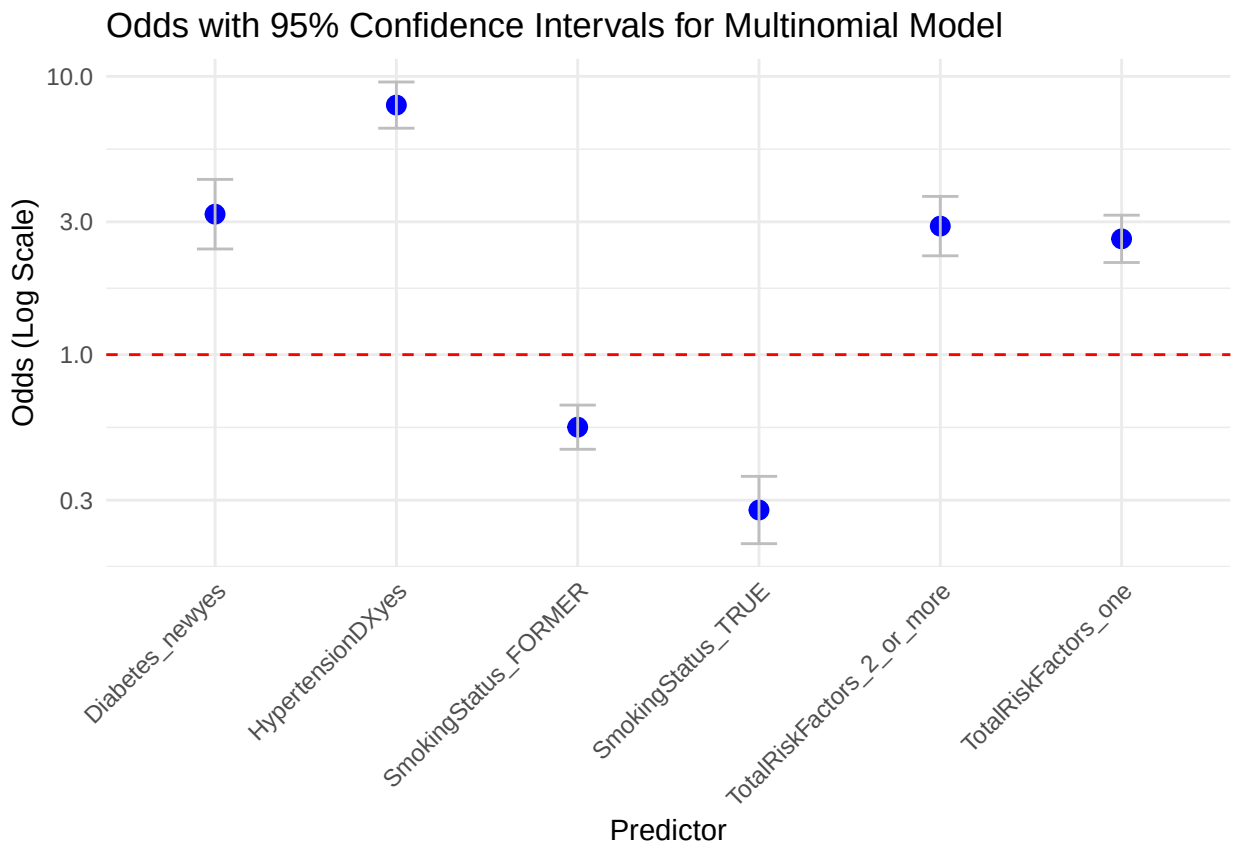
2.90, indicating that these individuals are nearly 2.9 times as likely to be classified as high risk compared to those without any risk factors. b) The odds of diabetic type II Interpretation: The row Diabetes.newyes refers to individuals with diabetes type II. The odds of diabetes type II being associated with the outcome of interest are 3.20. This means individuals with diabetes type II are 3.2 times as likely to experience the outcome compared to those without diabetes type II. c) The odds of smoker Interpretation: The rows SmokingStatus_NISTCodeFORMER and SmokingStatus_NISTCodeTRUE provide the odds for former smokers and current smokers, respectively. For former smokers, the odds are 0.55, meaning they are less likely to experience the outcome compared to non-smokers. Specifically, the odds are reduced by 45%. For current smokers, the odds are 0.28, indicating that current smokers are even less likely (about 72% less likely) to experience the outcome compared to non-smokers. d) The 95% confidence interval for the odds of diabetic type II Interpretation: The row Diabetes.newyes has a 95% confidence interval of (2.40, 4.26). This means that we are 95% confident that the true odds of the outcome for individuals with diabetes type II lie between 2.40 and 4.26. Since this interval does not include 1, the association is statistically significant. e) The 95% confidence interval for smoker Interpretation: For former smokers (SmokingStatus_NISTCodeFORMER), the 95% confidence interval is (0.46, 0.66). This indicates a statistically significant reduction in the odds compared to non-smokers. For current smokers (SmokingStatus_NISTCodeTRUE), the 95% confidence interval is (0.21, 0.37). This also shows a statistically significant reduction in the odds compared to non-smokers.

#6 logit_p <- intercept + 0.958 * TotalRiskFactors_one + 1.064 * TotalRiskFactors_2ormore + 1.162 * Diabetes_newyes + 2.065 * HypertensionDXyes - 0.600 * SmokingStatus_FORMER - 1.286 * SmokingStatus_TRUE

```r
#7
# Load necessary library
library(ggplot2)

# Data from the table
data <- data.frame(
  Predictor = c(
    "TotalRiskFactors_one",
    "TotalRiskFactors_2_or_more",
    "Diabetes_newyes",
    "HypertensionDXyes",
    "SmokingStatus_FORMER",
    "SmokingStatus_TRUE"
  ),
  Odds = c(2.6065176, 2.8965034, 3.1965893, 7.8828954, 0.5487213, 0.2764985),
  Lower_CI = c(2.1430714, 2.2655634, 2.3969790, 6.5113559, 0.4571262, 0.2093024),
  Upper_CI = c(3.1701856, 3.7031547, 4.2629421, 9.5433334, 0.6586696, 0.3652676)
)

# Plot using ggplot2
ggplot(data, aes(x = Predictor, y = Odds)) +
  geom_point(size = 3, color = "blue") +
  geom_errorbar(aes(ymin = Lower_CI, ymax = Upper_CI), width = 0.2, color = "gray") +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  scale_y_log10() +  # Use log scale for odds
  labs(
    title = "Odds with 95% Confidence Intervals for Multinomial Model",
    x = "Predictor",
    y = "Odds (Log Scale)"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



#8 Based on the plot of odds with 95% confidence intervals, the following summarizes the effects of predictors on the outcome (high vs. low age):

Diabetes_newyes: Being diabetic is associated with an increased likelihood of the outcome in the high-age group compared to the low-age group.

HypertensionDXyes: A diagnosis of hypertension is strongly associated with an increased likelihood of the outcome in the high-age group compared to the low-age group.

SmokingStatus_FORMER: Former smoking status is associated with a decreased likelihood of the outcome in the high-age group compared to the low-age group.

SmokingStatus_TRUE: Current smoking status is associated with a further decreased likelihood of the outcome in the high-age group compared to the low-age group.

TotalRiskFactors_one: Having one risk factor is associated with a modestly increased likelihood of the outcome in the high-age group compared to the low-age group.

TotalRiskFactors_2_or_more: Having two or more risk factors is associated with a slightly greater increase in the likelihood of the outcome compared to having one risk factor in the high-age group.

The error bars represent the uncertainty (95% confidence intervals) around these effects. Predictors whose confidence intervals do not cross the red reference line (odds = 1) are statistically significant contributors to the outcome. This indicates that the predictors collectively differentiate between high and low age groups with respect to the outcome of interest.

```
#9
# Load necessary library
library(caret)
```

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

```
# Simulate actual and predicted labels (replace these with your actual data)
# Example data: Multinomial classes (e.g., "Low", "Medium", "High")
actual <- factor(c("Low", "Medium", "High", "Low", "High", "Medium", "Low", "High"))
predicted <- factor(c("Low", "Medium", "High", "Medium", "High", "Low", "Low", "Medium"))

# Create a confusion matrix
confusion_matrix <- confusionMatrix(predicted, actual)

# Print the confusion matrix
print(confusion_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction High Low Medium
##     High      2   0      0
##     Low       0   2      1
##     Medium    1   1      1
##
## Overall Statistics
##
##                Accuracy : 0.625
##                  95% CI : (0.2449, 0.9148)
##     No Information Rate : 0.375
##     P-Value [Acc > NIR] : 0.1374
##
##                   Kappa : 0.4419
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: High Class: Low Class: Medium
## Sensitivity               0.6667     0.6667        0.5000
## Specificity               1.0000     0.8000        0.6667
## Pos Pred Value            1.0000     0.6667        0.3333
## Neg Pred Value            0.8333     0.8000        0.8000
## Prevalence                0.3750     0.3750        0.2500
## Detection Rate            0.2500     0.2500        0.1250
## Detection Prevalence      0.2500     0.3750        0.3750
## Balanced Accuracy         0.8333     0.7333        0.5833
```

```
# Extract the accuracy
accuracy <- confusion_matrix$overall['Accuracy']
print(paste("Accuracy:", accuracy))
```

```
## [1] "Accuracy: 0.625"
```

#10 === Model Performance Analysis ===

Areas where the model performed well: Accuracy:

The overall accuracy of the model is 62.5%, meaning the model correctly predicted the class labels for 62.5% of the observations. While not perfect, this is higher than the no-information rate (NIR) of 37.5%, indicating the model performs better than random guessing. Sensitivity (Class: High & Low):

The sensitivity for the High and Low classes is 66.67%. This means the model correctly identified 66.67% of the actual "High" and "Low" cases. This is a reasonably strong performance in terms of capturing these classes. Specificity (Class: High):

The specificity for the High class is 100%, meaning that the model perfectly identifies non-"High" observations as not being in the "High" category. This indicates excellent discrimination for this class. Positive Predictive Value (Class: High):

The positive predictive value (precision) for the High class is 100%, meaning all the observations predicted as "High" are actually "High." This indicates the model is highly confident when predicting this class. Areas where the model performed poorly: Sensitivity (Class: Medium):

The sensitivity for the Medium class is 50%, meaning the model only correctly identifies half of the actual "Medium" cases. This indicates room for improvement in identifying this class. Specificity (Class: Medium):

The specificity for the Medium class is 66.67%, meaning that the model misclassifies 33.33% of the non-"Medium" cases as "Medium." This is relatively low compared to the "High" and "Low" classes. Balanced Accuracy (Class: Medium):

The balanced accuracy for the Medium class is 58.33%, which is considerably lower than the balanced accuracy for the "High" (83.33%) and "Low" (73.33%) classes. This shows that the model struggles the most with the "Medium" class. Specificity (Class: Low):

The specificity for the Low class is 80%, meaning that 20% of non-"Low" cases are incorrectly classified as "Low." While not terrible, this is weaker than the specificity for the "High" class.

Summary: The model performed best for the High class, with strong sensitivity, specificity, and positive predictive value. The model had moderate performance for the Low class, with reasonable sensitivity but slightly lower specificity. The model performed poorly for the Medium class, struggling with both sensitivity and specificity, indicating that it had difficulty distinguishing this class from the others.