MATH 4322 Final Project Group 9

Introduction

Logistic Regression (Ryan Nguyen, Alan Johnson)

Advantages: Interpretability: Logistic regression coefficients represent the log of the odds ratio, making it easier to interpret the impact of each predictor variable on the probability of the outcome. Efficiency: Logistic regression can handle large datasets efficiently, making it suitable for real-time predictions. Low Variance: It tends to perform well with small datasets and is less prone to overfitting compared to more complex models. Assumption of Independence: Logistic regression doesn't require the predictors to be independent of each other, unlike some other models like Naive Bayes.

Disadvantages: Assumption of Linearity: Logistic regression assumes a linear relationship between the independent variables and the logit of the outcome variable. If this assumption is violated, the model's performance may suffer. Limited Outcome: It's primarily designed for binary classification tasks and may not perform well with multi-class classification without modifications. Sensitivity to Outliers: Logistic regression can be sensitive to outliers, which may disproportionately influence the model's coefficients and predictions. Not Suitable for Complex Relationships: If the relationship between predictors and the log-odds of the outcome is highly non-linear or complex, logistic regression may not capture it effectively.

Model Formula

$$P(\text{cardio} = 1|X) = \frac{\begin{array}{l}\exp(\beta_0 + \beta_1 \times \text{age} + \beta_2 \times \text{height} + \beta_3 \times \text{weight} + \beta_4 \times \text{gender} + \beta_5 \times \text{ap\_hi} \\ + \beta_6 \times \text{ap\_lo} + \beta_7 \times \text{cholesterol} + \beta_8 \times \text{gluc} + \beta_9 \times \text{smoke} + \beta_{10} \times \text{alco} \\ + \beta_{11} \times \text{active})\end{array}}{\begin{array}{l}1 + \exp(\beta_0 + \beta_1 \times \text{age} + \beta_2 \times \text{height} + \beta_3 \times \text{weight} + \beta_4 \times \text{gender} + \beta_5 \times \text{ap\_hi} \\ + \beta_6 \times \text{ap\_lo} + \beta_7 \times \text{cholesterol} + \beta_8 \times \text{gluc} + \beta_9 \times \text{smoke} + \beta_{10} \times \text{alco} \\ + \beta_{11} \times \text{active})\end{array}} \tag{1}$$

Model 1 - Include all predictors

```
library(readr)
cardio_train <- read_delim("cardio_train.csv",
                           delim = ";", escape_double = FALSE, trim_ws = TRUE)
```

```
Rows: 70000 Columns: 13
-- Column specification --------------------------------------------------------
Delimiter: ";"
dbl (13): id, age, gender, height, weight, ap_hi, ap_lo, cholesterol, gluc, ...

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.
```

```
cardio_train$gender = as.factor(cardio_train$gender)
cardio_train$cholesterol = as.factor(cardio_train$cholesterol)
cardio_train$gluc = as.factor(cardio_train$gluc)
cardio_train$smoke = as.factor(cardio_train$smoke)
cardio_train$alco = as.factor(cardio_train$alco)
cardio_train$active = as.factor(cardio_train$active)
cardio_train$cardio = as.factor(cardio_train$cardio)

heart.logistic1 =  glm(cardio  ~  . - id,  family  =  "binomial",
                       data  =  cardio_train)
```

```
Warning: glm.fit: algorithm did not converge
```

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(heart.logistic1)
```

```

Call:
glm(formula = cardio ~ . - id, family = "binomial", data = cardio_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.4904  -0.9635  -0.0980   0.9907   4.6621
```

```
Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)  -8.084e+00  2.213e-01 -36.535  < 2e-16 ***
age           1.485e-04  3.557e-06  41.735  < 2e-16 ***
gender2       1.430e-02  2.107e-02   0.679    0.497
height       -5.626e-03  1.232e-03  -4.567 4.95e-06 ***
weight        1.521e-02  6.607e-04  23.023  < 2e-16 ***
ap_hi         3.951e-02  6.057e-04  65.235  < 2e-16 ***
ap_lo         3.004e-04  6.735e-05   4.460 8.18e-06 ***
cholesterol2  4.222e-01  2.593e-02  16.285  < 2e-16 ***
cholesterol3  1.134e+00  3.444e-02  32.929  < 2e-16 ***
gluc2         3.011e-02  3.438e-02   0.876    0.381
gluc3        -3.387e-01  3.809e-02  -8.894  < 2e-16 ***
smoke1       -1.314e-01  3.320e-02  -3.958 7.57e-05 ***
alco1        -1.695e-01  4.026e-02  -4.211 2.54e-05 ***
active1      -2.101e-01  2.105e-02  -9.981  < 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: 97041  on 69999  degrees of freedom
Residual deviance: 80883  on 69986  degrees of freedom
AIC: 80911

Number of Fisher Scoring iterations: 25
```

Predictors with p-values less than 0.05 are considered significant. For example, age, height, weight, blood pressure, cholesterol levels, smoking status, alcohol consumption, and physical activity are all significant predictors. Gender, glucose level don't appear to be significant predictors as their p-values exceed chosen significance level.

Model 2 - Only include statistically significant predictors

We will use the significant predictors found from heart.logistic1 to to create Model 2. This means all predictors except for id, gender, and glucose will be included.

```r
heart.logistic2 = glm(cardio ~ age+height+weight+ap_hi+ap_lo+cholesterol+smoke+alco+act
                      , family = "binomial",
                      data = cardio_train)

summary(heart.logistic2)
```

```
Call:
glm(formula = cardio ~ age + height + weight + ap_hi + ap_lo +
    cholesterol + smoke + alco + active, family = "binomial",
    data = cardio_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.4904  -0.9639  -0.0992   0.9900   4.6678

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)  -8.124e+00  2.028e-01 -40.062  < 2e-16 ***
age           1.476e-04  3.551e-06  41.550  < 2e-16 ***
height       -5.356e-03  1.103e-03  -4.857 1.19e-06 ***
weight        1.516e-02  6.586e-04  23.023  < 2e-16 ***
ap_hi         3.960e-02  6.047e-04  65.485  < 2e-16 ***
ap_lo         3.028e-04  6.765e-05   4.475 7.63e-06 ***
cholesterol2  4.234e-01  2.497e-02  16.959  < 2e-16 ***
cholesterol3  9.855e-01  2.962e-02  33.275  < 2e-16 ***
smoke1       -1.222e-01  3.205e-02  -3.812 0.000138 ***
alco1        -1.641e-01  4.013e-02  -4.090 4.31e-05 ***
active1      -2.085e-01  2.104e-02  -9.909  < 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: 97041  on 69999  degrees of freedom
Residual deviance: 80984  on 69989  degrees of freedom
AIC: 81006

Number of Fisher Scoring iterations: 8
```

```
step(heart.logistic1)
```

```
Start:  AIC=80910.62
cardio ~ (id + age + gender + height + weight + ap_hi + ap_lo +
    cholesterol + gluc + smoke + alco + active) - id

             Df Deviance   AIC
```

```
- gender       1    80882 80908
<none>               80883 80911
- smoke        1    80900 80926
- alco         1    80902 80928
- ap_lo        1    80902 80928
- height       1    80917 80943
- gluc         2    80960 80984
- active       1    80987 81013
- weight       1    81248 81274
- cholesterol  2    82098 82122
- age          1    82467 82493
- ap_hi        1    87965 87991

Step:  AIC=80907.87
cardio ~ age + height + weight + ap_hi + ap_lo + cholesterol +
    gluc + smoke + alco + active

             Df Deviance   AIC
<none>               80882 80908
- ap_lo        1    80901 80925
- smoke        1    80901 80925
- alco         1    80901 80925
- height       1    80921 80945
- gluc         2    80984 81006
- active       1    80986 81010
- weight       1    81247 81271
- cholesterol  2    82099 82121
- age          1    82466 82490
- ap_hi        1    87982 88006


Call:  glm(formula = cardio ~ age + height + weight + ap_hi + ap_lo +
    cholesterol + gluc + smoke + alco + active, family = "binomial",
    data = cardio_train)

Coefficients:
 (Intercept)           age        height        weight         ap_hi
  -8.1440960     0.0001485    -0.0052551     0.0152121     0.0395279
       ap_lo  cholesterol2  cholesterol3         gluc2         gluc3
   0.0003007     0.4217513     1.1337397     0.0300489    -0.3389234
      smoke1         alco1       active1
  -0.1256165    -0.1681051    -0.2100651
```

```
Degrees of Freedom: 69999 Total (i.e. Null);  69987 Residual
Null Deviance:      97040
Residual Deviance: 80880    AIC: 80910
```

Model 3 - Using predictors from stepwise regression

The step() function was applied on the initial model (heart.logistic1) to perform stepwise regression based on the Akaike Information Criterion (AIC). The stepwise process sequentially evaluates each predictor's contribution to the model and removes predictors that do not significantly improve the model fit, based on AIC.

After the stepwise regression process, the final selected predictors were age, height, weight, ap_hi (systolic blood pressure), ap_lo (diastolic blood pressure), cholesterol, gluc, smoke, alco (alcohol consumption), and active (physical activity). The coefficients for these predictors are provided in the summary output of the final model (heart.logistic3).

Interpretation of Results: The coefficients in the final model represent the log odds of the outcome (cardiovascular disease) associated with each unit change in the predictor, holding other predictors constant. For example, a positive coefficient for a predictor indicates an increase in the log odds of cardiovascular disease with an increase in that predictor, while a negative coefficient indicates a decrease in the log odds. The significance of each predictor is determined by its corresponding p-value, with predictors having p-values less than the chosen significance level (typically 0.05) considered statistically significant.

```r
heart.logistic3 = glm(formula = cardio ~ age + height + weight + ap_hi + ap_lo +
    cholesterol + gluc + smoke + alco + active, family = "binomial",
    data = cardio_train)
summary(heart.logistic3)
```

```
Call:
glm(formula = cardio ~ age + height + weight + ap_hi + ap_lo +
    cholesterol + gluc + smoke + alco + active, family = "binomial",
    data = cardio_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.4904  -0.9638  -0.0979   0.9908   4.6623

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)  -8.144e+00  2.030e-01 -40.123  < 2e-16 ***
```

```
age           1.485e-04  3.556e-06  41.760  < 2e-16 ***
height       -5.255e-03  1.104e-03  -4.760 1.94e-06 ***
weight        1.521e-02  6.607e-04  23.024  < 2e-16 ***
ap_hi         3.953e-02  6.051e-04  65.330  < 2e-16 ***
ap_lo         3.007e-04  6.736e-05   4.464 8.04e-06 ***
cholesterol2  4.218e-01  2.592e-02  16.273  < 2e-16 ***
cholesterol3  1.134e+00  3.444e-02  32.921  < 2e-16 ***
gluc2         3.005e-02  3.438e-02   0.874    0.382
gluc3        -3.389e-01  3.809e-02  -8.898  < 2e-16 ***
smoke1       -1.256e-01  3.209e-02  -3.914 9.07e-05 ***
alco1        -1.681e-01  4.021e-02  -4.181 2.90e-05 ***
active1      -2.101e-01  2.105e-02  -9.980  < 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: 97041  on 69999  degrees of freedom
Residual deviance: 80882  on 69987  degrees of freedom
AIC: 80908

Number of Fisher Scoring iterations: 13
```

Determining Best Model

```r
extract_info  = function(model) {
  deviance <- summary(model)$null.deviance
  residual_deviance <- summary(model)$deviance
  r_squared <- 1 - (residual_deviance / deviance)
  AIC <- AIC(model)
  BIC <- BIC(model)

  return(c(Null_Deviance = deviance,
           Residual_Deviance = residual_deviance,
           R_Squared = r_squared,
           AIC = AIC,
           BIC = BIC))
}

# Extract information from each model
info1 <- extract_info(heart.logistic1)
info2 <- extract_info(heart.logistic2)
```

```r
info3 <- extract_info(heart.logistic3)

# Create a data frame to store the information
model_info <- data.frame(
  Model = c("heart.logistic1", "heart.logistic2", "heart.logistic3"),
  Null_Deviance = c(info1["Null_Deviance"], info2["Null_Deviance"], info3["Null_Deviance"]
  Residual_Deviance = c(info1["Residual_Deviance"], info2["Residual_Deviance"], info3["Res
  R_Squared = c(info1["R_Squared"], info2["R_Squared"], info3["R_Squared"]),
  AIC = c(info1["AIC"], info2["AIC"], info3["AIC"]),
  BIC = c(info1["BIC"], info2["BIC"], info3["BIC"])
)
(model_info)
```

```
            Model Null_Deviance Residual_Deviance R_Squared      AIC      BIC
1 heart.logistic1      97040.58          80882.62 0.1665072 80910.62 81038.81
2 heart.logistic2      97040.58          80983.71 0.1654656 81005.71 81106.43
3 heart.logistic3      97040.58          80881.87 0.1665149 80907.87 81026.91
```

It is evident that Model 3 is the best model out of all the regression models because the AIC and BIC is the lowest. The AIC of Model 3 is 80907.87 while the AIC of Model 1 is 80910.62 and the AIC of Model 2 is 81005.71. We will further test Model 3 (Best Model) on trainign and validation tests.

Final Equation for Logistic Regression Model

$$P(\text{cardio} = 1|X) = \frac{\begin{array}{l}\exp(-8.188 + 0.0001485 \times \text{age} + -0.005255 \times \text{height} + 0.01521 \times \text{weight} \\ + 0.03953 \times \text{ap\_hi} + 0.0003007 \times \text{ap\_lo} + 0.4218 \times \text{cholesterol2} \\ + 1.134 \times \text{cholesterol3} + 0.03005 \times \text{gluc2} + -0.3389 \times \text{gluc3} \\ + -0.1256 \times \text{smoke1} + -0.1681 \times \text{alco1} + -0.2101 \times \text{active1})\end{array}}{\begin{array}{l}1 + \exp(-8.188 + 0.0001485 \times \text{age} + -0.005255 \times \text{height} + 0.01521 \times \text{weight} \\ + 0.03953 \times \text{ap\_hi} + 0.0003007 \times \text{ap\_lo} + 0.4218 \times \text{cholesterol2} \\ + 1.134 \times \text{cholesterol3} + 0.03005 \times \text{gluc2} + -0.3389 \times \text{gluc3} \\ + -0.1256 \times \text{smoke1} + -0.1681 \times \text{alco1} + -0.2101 \times \text{active1})\end{array}} \tag{2}$$

Training/ Validation

```
set.seed(100)
test_errors = numeric(10)
for(i in 1:10){
  # initialize vector to store prediction errors
sample= sample.int(n = nrow(cardio_train), size = floor(0.80*nrow(cardio_train)))
train.heart.logistic = cardio_train[sample,]
test.heart.logistic = cardio_train[-sample,]

train.logistic = glm(formula = cardio ~ age + height + weight + ap_hi + ap_lo +
    cholesterol + gluc + smoke + alco + active, family = "binomial",
    data = cardio_train)

glm.pred = predict.glm(train.logistic, newdata = test.heart.logistic, type = "response")

pred = predict(train.logistic, type = "response", newdata = test.heart.logistic)
val = ifelse(pred <0.5,"0", "1")
tab = table(val, test.heart.logistic$cardio)
test_errors[i] = (tab[2]+tab[3])/(tab[1]+tab[2]+tab[3]+tab[4])
}
(mean_test_error = mean(test_errors))
```

```
[1] 0.2774429
```

The dataset is randomly split into training and validation sets using an 80-20 split. The logistic regression model uses predictors from heart.logistic3 which are age, height, weight, blood pressure (ap_hi and ap_lo), cholesterol level, glucose level, smoking status, alcohol consumption, and physical activity.

The trained model is then used to predict the probability of cardiovascular disease for the validation set using the predict.glm() function.

Predicted probabilities are converted to binary predictions using a threshold of 0.5, where probabilities greater than 0.5 are classified as 1 (indicating presence of cardiovascular disease) and probabilities less than or equal to 0.5 are classified as 0 (indicating absence of cardiovascular disease).

The test error rate is calculated by creating a confusion matrix and dividing the sum of the misclassified observations (false positives and false negatives) by all the observations. Mean Test Error Rate Calculation: The mean error rate is calculated by averaging the test error rates obtained from 10 iterations of the validation process. In each iteration, the logistic regression

model is trained on a randomly selected 80% of the data, and the mean error rate is calculated based on predictions made on the remaining 20% of the data. The test error rate provides an estimate of the model's performance in predicting cardiovascular disease risk.

Mean Error Rate Result: The mean error rate obtained from the validation process is approximately 27.74%. This indicates that, on average, the logistic regression model misclassifies cardiovascular disease status in the validation set for 27.74% of observations.

Results

Based on the logistic regression models trained and validated for predicting cardiovascular disease risk, several important insights can be drawn.

The inclusion of statistically significant predictors, such as age, blood pressure, cholesterol levels, smoking status, alcohol consumption, and physical activity, in the final models underscores their crucial roles in assessing cardiovascular risk. These findings align with established medical knowledge regarding the impact of these factors on heart health. Additionally, the error rate obtained from the validation process indicates that the logistic regression models are performing well in predicting cardiovascular disease status, with an average misclassification rate of only around 27.74%.

Overall, the results suggest that logistic regression modeling, when incorporating significant predictors, can effectively predict cardiovascular disease risk. By leveraging key demographic, clinical, and lifestyle factors, healthcare practitioners can utilize these models to identify individuals at higher risk of cardiovascular disease and tailor preventive strategies and interventions accordingly. These findings contribute to a better understanding of cardiovascular risk assessment and highlight the potential of logistic regression models in supporting proactive heart health management strategies.