MATH 4322 Final Project Group 9

Introduction

Logistic Regression (Ryan Nguyen, Alan Johnson)

Paragraph explaining why we are using logistic regression models and the advantages and disadvantages of the model.

Model Formula

$$P(cardio = 1|X) = \frac{e^{(\beta_0+\beta_1 X_1+\beta_2 X_2+...+\beta_n X_n)}}{1 + e^{(\beta_0+\beta_1 X_1+\beta_2 X_2+...+\beta_n X_n)}}$$

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)
```

```r
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
```

```r
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
```

Paragraph explaining which predictors are significant (look at significance table output)

Model 2 - Only include statistically significant predictors

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

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

```
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.logistic2)
```

Model 3 - Using predictors from stepwise regression

```
heart.logistic3 = glm(formula = cardio ~ age + height + weight + ap_hi + ap_lo +cholestero
```

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

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

```
summary(heart.logistic3)
```

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.4904  -0.9635  -0.1015   0.9910   4.6663

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -8.026e+00  2.011e-01 -39.908  < 2e-16 ***
age           1.480e-04  3.550e-06  41.684  < 2e-16 ***
height       -6.002e-03  1.090e-03  -5.507 3.66e-08 ***
```

```
weight        1.516e-02  6.586e-04  23.022  < 2e-16 ***
ap_hi         3.954e-02  6.043e-04  65.434  < 2e-16 ***
ap_lo         3.027e-04  6.753e-05   4.482 7.38e-06 ***
cholesterol2  4.216e-01  2.496e-02  16.894  < 2e-16 ***
cholesterol3  9.850e-01  2.961e-02  33.260  < 2e-16 ***
alco1        -2.143e-01  3.787e-02  -5.660 1.51e-08 ***
active1      -2.101e-01  2.104e-02  -9.989  < 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: 80978  on 69990  degrees of freedom
AIC: 80998

Number of Fisher Scoring iterations: 25
```

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)
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          80978.02 0.1655242 80998.02 81089.58
```

Final Equation for Logistic Regression Model

Insert latex equation here

Training/ Validation

```
set.seed(100)
for(i in 1:10){
  # initialize vector to store prediction errors
test_errors = numeric(10)
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(cardio ~ age + height + weight + ap_hi + ap_lo +cholesterol + alco +

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

# Convert probability to binary
test_predictions_binary = ifelse(glm.pred > 0.5, 1, 0)

# Calculate test prediction error
test_error= mean(test_predictions_binary != test.heart.logistic$cardio)

test_errors[i] = test_error
}
```

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

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

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

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

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

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

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

Warning: glm.fit: algorithm did not converge

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

Warning: glm.fit: algorithm did not converge

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

Warning: glm.fit: algorithm did not converge

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

```
(mean_test_error = mean(test_errors))
```

```
[1] 0.02805
```

Paragraph explaining the the procedure above and the mean error rate Two paragraphs to provide the interpretation of results and your conclusions as it pertains to the original overall question.

Neural Network