AMS580 - Team Project 1 - R

Eshan Shakrani - 112802596 Vishnu Teja Sardee Priyansh Desai
Mustafa Isik

3/2/2023

Loading Necessary Libraries

```
# necessary libraries
if (!requireNamespace("tidyverse")) install.packages("tidyverse")
```

```
## Loading required namespace: tidyverse
```

```
if (!requireNamespace("MASS")) install.packages("MASS")
```

```
## Loading required namespace: MASS
```

```
if (!requireNamespace("magrittr")) install.packages("magrittr")
if (!requireNamespace("caret")) install.packages("caret")
```

```
## Loading required namespace: caret
```

```
if (!requireNamespace("mlbench")) install.packages("mlbench")
```

```
## Loading required namespace: mlbench
```

```
library("tidyverse")
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library("MASS")
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library("magrittr")
```

```
##
## Attaching package: 'magrittr'
##
## The following object is masked from 'package:purrr':
##
##     set_names
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library("caret")
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library("mlbench")
```

Data Cleaning and Manipulation

```
# load the PimaIndiansDiabetes2 dataset and store in variable 'dataset'
data(PimaIndiansDiabetes2)
dataset <- PimaIndiansDiabetes2

# send the data to a csv file to use in Python
if (!file.exists("pimaindiansdiabetes2.csv"))
  write.csv(dataset, file = "pimaindiansdiabetes2.csv")
```

```
# look at the first few observations in the data
head(dataset)
```

```
##   pregnant glucose pressure triceps insulin mass pedigree age diabetes
## 1        6     148       72      35      NA 33.6    0.627  50      pos
## 2        1      85       66      29      NA 26.6    0.351  31      neg
## 3        8     183       64      NA      NA 23.3    0.672  32      pos
```

```
## 4        1      89       66      23      94 28.1    0.167  21      neg
## 5        0     137       40      35     168 43.1    2.288  33      pos
## 6        5     116       74      NA      NA 25.6    0.201  30      neg
```

```
# convert response variable to numeric
# 'neg' --> 0
# 'pos' --> 1
dataset$diabetes <- as.numeric(dataset$diabetes) - 1
```

```
# rows and columns in the data
dim(dataset)
```

```
## [1] 768   9
```

```
# 768 rows, 9 columns

cat("There are", dim(dataset)[1], "observations and", dim(dataset)[2], "features in the dataset.")
```

```
## There are 768 observations and 9 features in the dataset.
```

```
# count the number of missing values in the data
cat("There are a total of", sum(is.na(dataset)), "missing values in the dataset.")
```

```
## There are a total of 652 missing values in the dataset.
```

```
# separate by column
colSums(is.na(dataset))
```

```
## pregnant  glucose pressure  triceps  insulin     mass pedigree      age
##        0        5       35      227      374       11        0        0
## diabetes
##        0
```

```
# remove observations with missing values
dataset <- na.omit(dataset)
```

```
# shape of the data after removing NA values
dim(dataset)
```

```
## [1] 392   9
```

```
# 392 rows, 9 columns

cat("After removing NA values, there are", dim(dataset)[1], "observations left over.")
```

```
## After removing NA values, there are 392 observations left over.
```

```r
head(dataset)
```

```
##    pregnant glucose pressure triceps insulin mass pedigree age diabetes
## 4         1      89       66      23      94 28.1    0.167  21        0
## 5         0     137       40      35     168 43.1    2.288  33        1
## 7         3      78       50      32      88 31.0    0.248  26        1
## 9         2     197       70      45     543 30.5    0.158  53        1
## 14        1     189       60      23     846 30.1    0.398  59        1
## 15        5     166       72      19     175 25.8    0.587  51        1
```

Logistic Regression

```r
# split the data into 80% training and 20% testing
set.seed(123)
training.samples <- dataset$diabetes %>%
  createDataPartition(p = 0.8, list = FALSE)
train.data <- dataset[training.samples, ]
test.data <- dataset[-training.samples, ]
```

```r
# fit the logistic regression model using the training data
model <- glm(diabetes ~ ., data = train.data, family = binomial)
```

```r
summary(model)
```

```
##
## Call:
## glm(formula = diabetes ~ ., family = binomial, data = train.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8800  -0.6865  -0.3863   0.7187   2.2281
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.378740   1.305611  -7.183 6.80e-13 ***
## pregnant     0.033118   0.062612   0.529   0.5968
## glucose      0.035918   0.006363   5.645 1.65e-08 ***
## pressure    -0.002704   0.013176  -0.205   0.8374
## triceps      0.013279   0.018221   0.729   0.4661
## insulin     -0.000333   0.001426  -0.234   0.8154
## mass         0.061365   0.029888   2.053   0.0401 *
## pedigree     1.212056   0.479073   2.530   0.0114 *
## age          0.036957   0.020728   1.783   0.0746 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 402.89  on 313  degrees of freedom
## Residual deviance: 289.30  on 305  degrees of freedom
```

```
## AIC: 307.3
##
## Number of Fisher Scoring iterations: 5
```

```
# use the fitted model to make predictions on the testing data
probs <- model %>% predict(test.data, type = "response")
predicted.classes <- ifelse(probs > 0.5, 1, 0)
```

Results

```
# confusion matrix
confusionMatrix(factor(predicted.classes), factor(test.data$diabetes), positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 52  6
##          1  3 17
##
##                Accuracy : 0.8846
##                  95% CI : (0.7922, 0.9459)
##     No Information Rate : 0.7051
##     P-Value [Acc > NIR] : 0.0001474
##
##                   Kappa : 0.7116
##
##  Mcnemar's Test P-Value : 0.5049851
##
##             Sensitivity : 0.7391
##             Specificity : 0.9455
##          Pos Pred Value : 0.8500
##          Neg Pred Value : 0.8966
##              Prevalence : 0.2949
##          Detection Rate : 0.2179
##    Detection Prevalence : 0.2564
##       Balanced Accuracy : 0.8423
##
##        'Positive' Class : 1
##
```

```
# accuracy
accuracy <- mean(predicted.classes == test.data$diabetes)

cat("The accuracy of the model is", accuracy)
```

```
## The accuracy of the model is 0.8846154
```

```
cm <- table(predicted.classes, test.data$diabetes)
cm
```

```
##
## predicted.classes  0  1
##                 0 52  6
##                 1  3 17
```

```
# true negative - does not have diabetes and was predicted to not have diabetes
tn <- cm[1,1]

# true positive - has diabetes and was predicted to have diabetes
tp <- cm[2,2]

# false negative - has diabetes but was predicted to not have diabetes
fn <- cm[1,2]

# false positive - does not have diabetes but was predicted to have diabetes
fp <- cm[2,1]
```

```
# sensitivity
sens <- tp / (tp + fn)

cat("The probability that a diabetic subject is predicted to have diabetes is", sens)
```

```
## The probability that a diabetic subject is predicted to have diabetes is 0.7391304
```

```
# specificity
spec <- tn / (tn + fp)

cat("The probability that a non-diabetic subject is predicted to not have diabetes is", spec)
```

```
## The probability that a non-diabetic subject is predicted to not have diabetes is 0.9454545
```