Statistical Learning Lab

Assignment – 2: Logistic Regression

Name: Semanti Ghosh

Roll No.: 22IM10036

Loading the dataset

Code snippet

```
setwd("C:/Study/Semester_6/Statistical_Learning_Lab")
getwd()
data <- read.csv("diabetes.csv")
head(data)
```

Output

```
> setwd("C:/Study/Semester_6/Statistical_Learning_Lab")
> getwd()
[1] "C:/Study/Semester_6/Statistical_Learning_Lab"
> data <- read.csv("diabetes.csv")
> head(data)
  Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI DiabetesPedigreeFunction Age Outcome
1           6     148            72            35       0 33.6                    0.627  50       1
2           1      85            66            29       0 26.6                    0.351  31       0
3           8     183            64             0       0 23.3                    0.672  32       1
4           1      89            66            23      94 28.1                    0.167  21       0
5           0     137            40            35     168 43.1                    2.288  33       1
6           5     116            74             0       0 25.6                    0.201  30       0
> |
```

Preliminary Analysis

Obtaining the correlation between variables

```
cor(data)
str(data)
corr_matrix <- cor(data, use = "complete.obs")
print(corr_matrix)
```

Output

```
> cor(data)
                          Pregnancies    Glucose BloodPressure SkinThickness     Insulin        BMI
Pregnancies                1.00000000 0.12945867    0.14128198   -0.08167177 -0.07353461 0.01768309
Glucose                    0.12945867 1.00000000    0.15258959    0.05732789  0.33135711 0.22107107
BloodPressure              0.14128198 0.15258959    1.00000000    0.20737054  0.08893338 0.28180529
SkinThickness             -0.08167177 0.05732789    0.20737054    1.00000000  0.43678257 0.39257320
Insulin                   -0.07353461 0.33135711    0.08893338    0.43678257  1.00000000 0.19785906
BMI                        0.01768309 0.22107107    0.28180529    0.39257320  0.19785906 1.00000000
DiabetesPedigreeFunction  -0.03352267 0.13733730    0.04126495    0.18392757  0.18507093 0.14064695
Age                        0.54434123 0.26351432    0.23952795   -0.11397026 -0.04216295 0.03624187
Outcome                    0.22189815 0.46658140    0.06506836    0.07475223  0.13054795 0.29269466
                          DiabetesPedigreeFunction         Age    Outcome
Pregnancies                           -0.03352267  0.54434123 0.22189815
Glucose                                0.13733730  0.26351432 0.46658140
BloodPressure                          0.04126495  0.23952795 0.06506836
SkinThickness                          0.18392757 -0.11397026 0.07475223
Insulin                                0.18507093 -0.04216295 0.13054795
BMI                                    0.14064695  0.03624187 0.29269466
DiabetesPedigreeFunction               1.00000000  0.03356131 0.17384407
Age                                    0.03356131  1.00000000 0.23835598
Outcome                                0.17384407  0.23835598 1.00000000
> str(data)
'data.frame':   768 obs. of  9 variables:
 $ Pregnancies             : int  6 1 8 1 0 5 3 10 2 8 ...
 $ Glucose                 : int  148 85 183 89 137 116 78 115 197 125 ...
 $ BloodPressure           : int  72 66 64 66 40 74 50 0 70 96 ...
 $ SkinThickness           : int  35 29 0 23 35 0 32 0 45 0 ...
 $ Insulin                 : int  0 0 0 94 168 0 88 0 543 0 ...
 $ BMI                     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
 $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
 $ Age                     : int  50 31 32 21 33 30 26 29 53 54 ...
 $ Outcome                 : int  1 0 1 0 1 0 1 0 1 1 ...
```

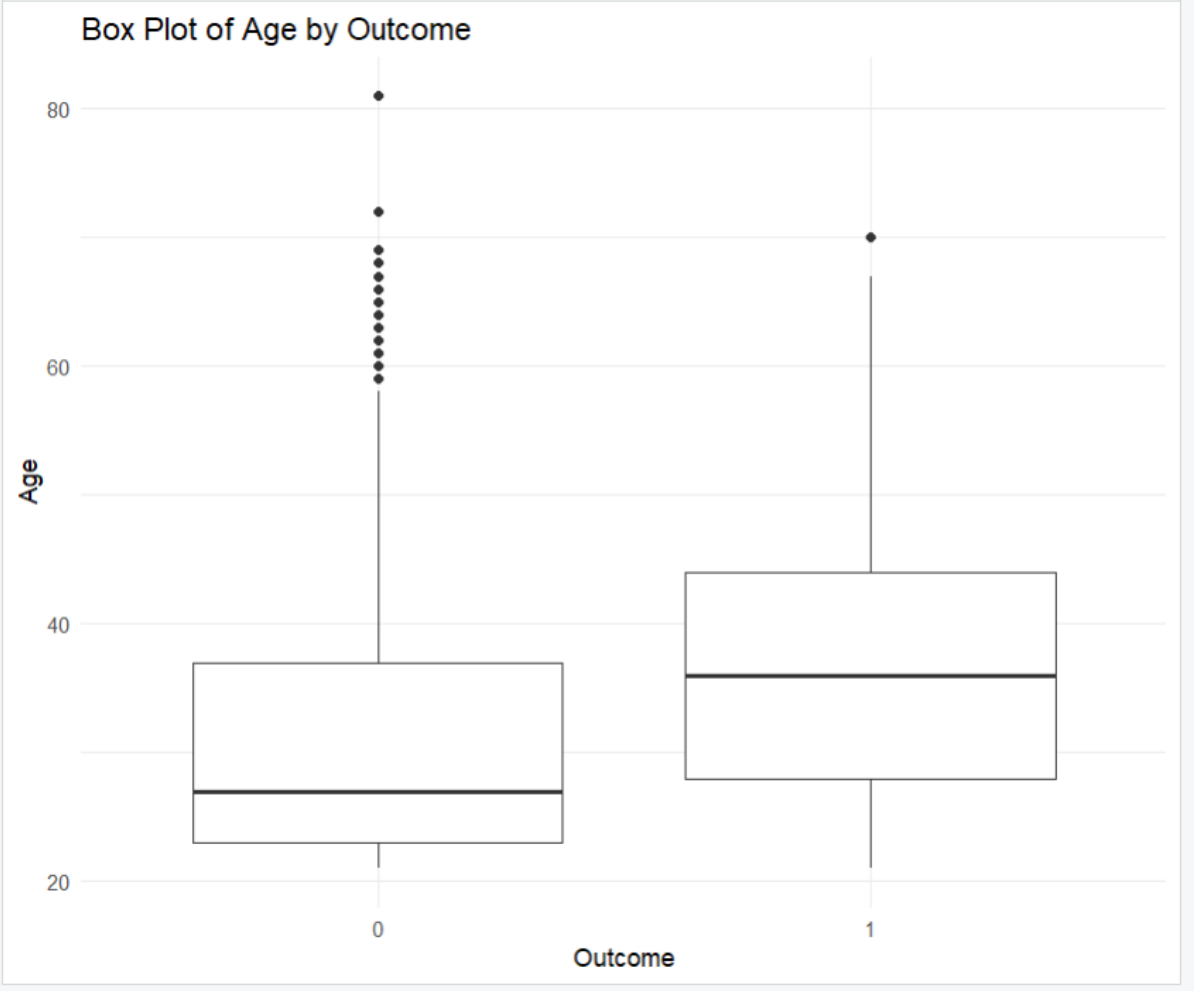
```
> corr_matrix <- cor(data, use = "complete.obs")
> print(corr_matrix)
                          Pregnancies    Glucose BloodPressure SkinThickness     Insulin        BMI
Pregnancies                1.00000000 0.12945867    0.14128198   -0.08167177 -0.07353461 0.01768309
Glucose                    0.12945867 1.00000000    0.15258959    0.05732789  0.33135711 0.22107107
BloodPressure              0.14128198 0.15258959    1.00000000    0.20737054  0.08893338 0.28180529
SkinThickness             -0.08167177 0.05732789    0.20737054    1.00000000  0.43678257 0.39257320
Insulin                   -0.07353461 0.33135711    0.08893338    0.43678257  1.00000000 0.19785906
BMI                        0.01768309 0.22107107    0.28180529    0.39257320  0.19785906 1.00000000
DiabetesPedigreeFunction  -0.03352267 0.13733730    0.04126495    0.18392757  0.18507093 0.14064695
Age                        0.54434123 0.26351432    0.23952795   -0.11397026 -0.04216295 0.03624187
Outcome                    0.22189815 0.46658140    0.06506836    0.07475223  0.13054795 0.29269466
                          DiabetesPedigreeFunction         Age    Outcome
Pregnancies                           -0.03352267  0.54434123 0.22189815
Glucose                                0.13733730  0.26351432 0.46658140
BloodPressure                          0.04126495  0.23952795 0.06506836
SkinThickness                          0.18392757 -0.11397026 0.07475223
Insulin                                0.18507093 -0.04216295 0.13054795
BMI                                    0.14064695  0.03624187 0.29269466
DiabetesPedigreeFunction               1.00000000  0.03356131 0.17384407
Age                                    0.03356131  1.00000000 0.23835598
Outcome                                0.17384407  0.23835598 1.00000000
>
```

Code snippets – obtaining the box plot and scatter plot

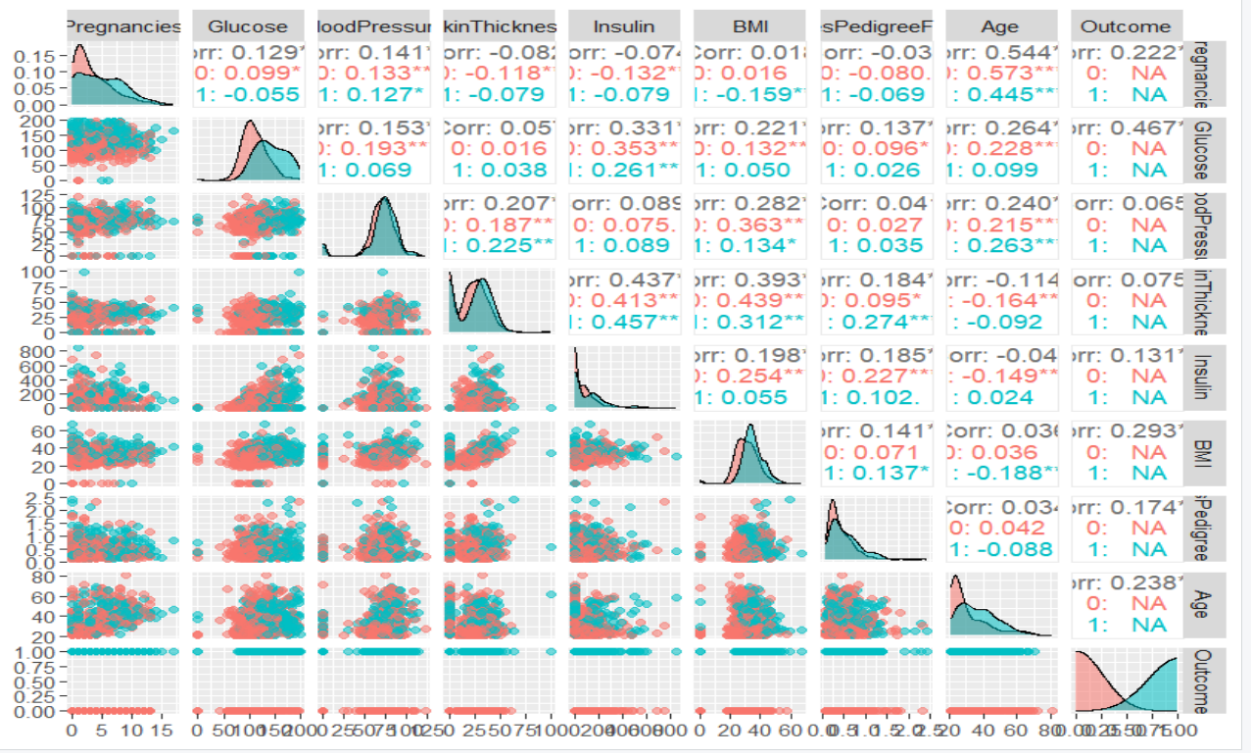
```
install.packages("ggplot2", dependencies=TRUE)
install.packages("GGally")

library(ggplot2)
library(GGally)
for (col in names(data)[-which(names(data) == "Outcome")]) {
  print(ggplot(data, aes(x = as.factor(Outcome), y = get(col))) +
          geom_boxplot() +
          labs(title = paste("Box Plot of", col, "by Outcome"),
               x = "Outcome", y = col) +
          theme_minimal())
}
pairs(data, col = data$Outcome)
ggpairs(data, aes(color = factor(Outcome), alpha = 0.5))
```

Box plot



Scatter plot



Random Sampling of data and fitting a Logistic Regression model

Code snippet

```
set.seed(97)
test_ind <- sample(1:nrow(data), size = 0.2*nrow(data))
test_data <- data[test_ind, ]
train_data <- data[-test_ind, ]

m1 <- glm(Outcome ~ ., data=train_data, family = binomial)
summary(m1)
```

Output

```
> set.seed(97)
> test_ind <- sample(1:nrow(data), size = 0.2*nrow(data))
> test_data <- data[test_ind, ]
> train_data <- data[-test_ind, ]
> m1 <- glm(Outcome ~ ., data=train_data, family = binomial)
> summary(m1)

Call:
glm(formula = Outcome ~ ., family = binomial, data = train_data)

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)             -8.4035146  0.7918792 -10.612  < 2e-16 ***
Pregnancies              0.1236089  0.0352123   3.510 0.000447 ***
Glucose                  0.0332003  0.0041317   8.035 9.33e-16 ***
BloodPressure           -0.0105720  0.0058514  -1.807 0.070800 .
SkinThickness           -0.0003811  0.0078230  -0.049 0.961148
Insulin                 -0.0014745  0.0010610  -1.390 0.164628
BMI                      0.0915629  0.0166884   5.487 4.10e-08 ***
DiabetesPedigreeFunction 1.0703170  0.3308134   3.235 0.001215 **
Age                      0.0136587  0.0102858   1.328 0.184203
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 790.97  on 614  degrees of freedom
Residual deviance: 584.41  on 606  degrees of freedom
AIC: 602.41

Number of Fisher Scoring iterations: 5

>
```

From the summary obtained, the number of pregnancies, the glucose level and the BMI are the most significant parameters (and the intercept is significant too). The Diabetes Pedigree function is also significant, but not as significant as the ones mentioned.

The coefficients of the parameters or the predictors measure the change in log-odds ratio for a unit change in the given parameter. If the coefficient is positive, the log-odds ratio increases with increase in the parameter, and if it is negative, the log-odds ratio decreases.

Confusion Matrix, Accuracy, F1 Score

Code snippet

```
prob <- predict(m1, newdata=test_data, type = "response")
class <- ifelse(prob >= 0.5, 1, 0)

install.packages("caret")
library(caret)
confusion_matrix <- table(Predicted = class, Actual = test_data$Outcome)
print(confusion_matrix)
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
print(accuracy)
precision <- confusion_matrix[2,2] / sum(confusion_matrix[,2])
recall <- confusion_matrix[2,2] / sum(confusion_matrix[2,])
f1_score <- 2 * (precision * recall) / (precision + recall)
print(f1_score)
```

Output

```
> prob <- predict(m1, newdata=test_data, type = "response")
> class <- ifelse(prob >= 0.5, 1, 0)
> library(caret)
> confusion_matrix <- table(Predicted = class, Actual = test_data$Outcome)
> print(confusion_matrix)
         Actual
Predicted  0  1
        0 86 20
        1 10 37
> accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
> print(accuracy)
[1] 0.8039216
> precision <- confusion_matrix[2,2] / sum(confusion_matrix[,2])
> recall <- confusion_matrix[2,2] / sum(confusion_matrix[2,])
> f1_score <- 2 * (precision * recall) / (precision + recall)
> print(f1_score)
[1] 0.7115385
> m2 <- glm(Outcome ~ Pregnancies + Glucose + BMI, data = train_data, family = binomial)
>
```

Fitting and comparing the two models

Code snippets

```
set.seed(97)
test_ind <- sample(1:nrow(data), size = 0.2*nrow(data))
test_data <- data[test_ind, ]
train_data <- data[-test_ind, ]

m1 <- glm(Outcome ~ ., data=train_data, family = binomial)
summary(m1)
coef(m1)
logLik(m1)
deviance(m1)
```

```
m2 <- glm(Outcome ~ Pregnancies + Glucose + BMI, data = train_data, family = binomial)
summary(m2)
logLik(m2)
deviance(m2)
```

Outputs for model 1

```
> m1 <- glm(Outcome ~ ., data=train_data, family = binomial)
> summary(m1)

Call:
glm(formula = Outcome ~ ., family = binomial, data = train_data)

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)             -8.4035146  0.7918792 -10.612  < 2e-16 ***
Pregnancies              0.1236089  0.0352123   3.510 0.000447 ***
Glucose                  0.0332003  0.0041317   8.035 9.33e-16 ***
BloodPressure           -0.0105720  0.0058514  -1.807 0.070800 .
SkinThickness           -0.0003811  0.0078230  -0.049 0.961148
Insulin                 -0.0014745  0.0010610  -1.390 0.164628
BMI                      0.0915629  0.0166884   5.487 4.10e-08 ***
DiabetesPedigreeFunction 1.0703170  0.3308134   3.235 0.001215 **
Age                      0.0136587  0.0102858   1.328 0.184203
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 790.97  on 614  degrees of freedom
Residual deviance: 584.41  on 606  degrees of freedom
AIC: 602.41

Number of Fisher Scoring iterations: 5

> coef(m1)
            (Intercept)             Pregnancies                 Glucose           BloodPressure
          -8.4035145975            0.1236089353            0.0332002864           -0.0105720422
          SkinThickness                 Insulin                     BMI DiabetesPedigreeFunction
          -0.0003810816           -0.0014744572            0.0915628635            1.0703169536
                    Age
           0.0136587090
> logLik(m1)
'log Lik.' -292.2054 (df=9)
> deviance(m1)
[1] 584.4109
```

Outputs for model 2

```
> m2 <- glm(Outcome ~ Pregnancies + Glucose + BMI, data = train_data, family = binomial)
> summary(m2)

Call:
glm(formula = Outcome ~ Pregnancies + Glucose + BMI, family = binomial,
    data = train_data)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.899982   0.698337 -11.313  < 2e-16 ***
Pregnancies  0.133144   0.029739   4.477 7.57e-06 ***
Glucose      0.032201   0.003687   8.734  < 2e-16 ***
BMI          0.082343   0.015038   5.476 4.36e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 790.97  on 614  degrees of freedom
Residual deviance: 602.34  on 611  degrees of freedom
AIC: 610.34

Number of Fisher Scoring iterations: 5

> logLik(m2)
'log Lik.' -301.1718 (df=4)
> deviance(m2)
[1] 602.3437
>
```

Deviance of model 1 = 584.4109

Deviance of model 2 = 602.3437

Model 1 has lower deviance than model 2, which means that model 1 fits the training data better than model. This is expected considering the fact that model 1 is trained taking greater number of parameters into consideration. However, we also try to reduce the number of parameters taken into consideration since too many parameters or too high powers since then the model becomes prone to overfitting. This might be the problem with model 1.