Experiment 9

Student Name: Jatin	**UID: 20BCS5951**
Branch: BE-CSE	**Section/Group:605-B**
Semester: 6th	**Date of Performance: 09/05/2023**
Subject Name: Data Mining Lab	**Subject Code: 20CSP-376**

1. **Aim:** Study of Regression Analysis using R Programming.

2. **Objective:** Linear Regression: It is a commonly used type of predictive analysis. It is a statistical approach for modelling the relationship between a dependent variable and a given set of independent variables.
   There are two types of linear regression.
   - Simple Linear Regression
   - Multiple Linear Regression

Simple Linear regression using R.

It is a statistical method that allows us to summarize and study relationships between two continuous (quantitative) variables. One variable denoted x is regarded as an independent variable and the other one denoted y is regarded as a dependent variable. It is assumed that the two variables are linearly related. Hence, we try to find a linear function that predicts the response value as accurately as possible as a function of the feature or independent variable(x).

3. **Script and Output:**
   The algorithm is as follows:
   - Generates 40 random IQ values with a mean of 30 and a standard deviation of
     2 and assign to "**IQ**" vector.

- Sorts the **IQ** vector in ascending order.
- Creates a vector **result** that contains pass (1) and fail (0) values for the 40 students.
- Combing the **IQ** and **result** vectors using **cbind()** and then converting them to a data frame **df** using **as.data.frame().**
- Opens a PNG file named "**LogisticRegressionGFG.png**" for writing.
- Creates a scatter plot of the **IQ** and **result** variables.
- Fits a logistic regression model to the data.

R Script:

```
# Generate random IQ values with mean = 30 and sd =2
IQ <- rnorm(40, 30, 2)


 # Sorting IQ level in ascending order
IQ <- sort(IQ)


 # Generate vector with pass and fail values of 40 students
result <- c(0, 0, 0, 1, 0, 0, 0, 0, 0, 1,             1, 0,
0, 0, 1, 1, 0, 0, 1, 0,
            0, 0, 1, 0, 0, 1, 1, 0, 1, 1,
1, 1, 1, 0, 1, 1, 1, 1, 0, 1)


 # Data Frame
df <- as.data.frame(cbind(IQ, result))


 # Print data frame print(df)
 # output to be present as PNG file
png(file="LogisticRegressionGFG.png")
```

```
 # Plotting IQ on x-axis and result on y-axis plot(IQ,
result, xlab = "IQ Level",
     ylab = "Probability of Passing")


 # Create a logistic model g = glm(result~IQ,
family=binomial, df)
 # Create a curve based on prediction using the regression
model
curve(predict(g, data.frame(IQ=x), type="resp"), add=TRUE)


 # This Draws a set of points


# Based on fit to the regression model points(IQ,
fitted(g), pch=30)
 # Summary of the regression model summary(g)
 # saving the file dev.off()
```

Output:

```
Console  Terminal  Background Jobs
R 4.3.0 · D:/Course/Semester 6/R/R Lab/
> # Generate random IQ values with mean = 30 and sd =2
> IQ <- rnorm(40, 30, 2)
>
> # Sorting IQ level in ascending order
> IQ <- sort(IQ)
>
> # Generate vector with pass and fail values of 40 students
> result <- c(0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
+             1, 0, 0, 0, 1, 1, 0, 0, 1, 0,
+             0, 0, 1, 0, 0, 1, 1, 0, 1, 1,
+             1, 1, 1, 0, 1, 1, 1, 1, 0, 1)
>
> # Data Frame
> df <- as.data.frame(cbind(IQ, result))
>
> # Print data frame
> print(df)
          IQ result
1  25.60160      0
2  26.22757      0
3  26.78154      0
4  27.37988      1
5  27.79395      0
6  28.05130      0
7  28.07159      0
8  28.67590      0
9  28.75959      0
10 28.75989      1
11 28.83543      1
12 28.87211      0
13 28.94983      0
14 28.98275      0
15 29.00410      1
16 29.05200      1
17 29.38304      0
18 29.42340      0
19 29.95299      1
20 30.16587      0
21 30.41585      0
22 30.52467      0
23 30.65712      1
```

```
Console  Terminal  Background Jobs
R 4.3.0 · D:/Course/Semester 6/R/R Lab/
22 30.52467      0
23 30.65712      1
24 30.65983      0
25 30.72865      0
26 30.94020      1
27 31.07091      1
28 31.13120      0
29 31.30234      1
30 31.37131      1
31 31.47103      1
32 31.62419      1
33 31.66112      1
34 31.88170      0
35 31.96142      1
36 32.04833      1
37 32.14125      1
38 32.26282      1
39 32.70545      0
40 35.71007      1
>
> # output to be present as PNG file
> png(file="LogisticRegressionGFG.png")
>
> # Plotting IQ on x-axis and result on y-axis
> plot(IQ, result, xlab = "IQ Level",
+      ylab = "Probability of Passing")
>
> # Create a logistic model
> g = glm(result~IQ, family=binomial, df)
>
> # Create a curve based on prediction using the regression model
> curve(predict(g, data.frame(IQ=x), type="resp"), add=TRUE)
>
> # This Draws a set of points
> # Based on fit to the regression model
> points(IQ, fitted(g), pch=30)
There were 40 warnings (use warnings() to see them)
>
> # Summary of the regression model
> summary(g)
```

```
>
> # Summary of the regression model
> summary(g)

Call:
glm(formula = result ~ IQ, family = binomial, data = df)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -16.8169     6.7922  -2.476   0.0133 *
IQ            0.5559     0.2252   2.469   0.0135 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 55.352  on 39  degrees of freedom
Residual deviance: 47.104  on 38  degrees of freedom
AIC: 51.104

Number of Fisher Scoring iterations: 4

>
> # saving the file
> dev.off()
null device
          1
>
```