Code:

# Load necessary libraries

library(caret)

library(dplyr)

# Read data from the Github link

file\_path <- "C:/Users/MSI/OneDrive/Desktop/ML/oulad-students.csv"

data <- read.csv(file\_path)

# Assuming 'data' is your data frame

subset\_data <- data[, c("id\_student", "date\_registration", "module\_presentation\_length", "studied\_credits", "num\_of\_prev\_attempts", "final\_result")]

# Describe the variables

summary(subset\_data)

# View the first few rows of the data

head(subset\_data)

# Remove rows with missing values

subset\_data <- na.omit(subset\_data)

# Convert categorical variables to factors

subset\_data$final\_result <- as.factor(subset\_data$final\_result)

# Ensure other categorical variables are already factors

# If not, convert them to factors as needed

# Split the data into training and testing sets (80% training, 20% testing)

set.seed(120) # For reproducibility

train\_index <- createDataPartition(subset\_data$final\_result, p = 0.8, list = FALSE)

train\_data <- subset\_data[train\_index, ]

test\_data <- subset\_data[-train\_index, ]

# Train the classification model (logistic regression)

model <- glm(final\_result ~ ., data = train\_data, family = binomial)

# Make predictions on the test data

predictions <- predict(model, newdata = test\_data, type = "response")

# Ensure predicted classes have the same levels as test data

predicted\_classes <- factor(predicted\_classes, levels = levels(test\_data$final\_result))

# Convert predicted classes to factor with the same levels as test\_data$final\_result

predicted\_classes <- factor(predicted\_classes, levels = levels(test\_data$final\_result))

# Evaluate the model

confusionMatrix(predicted\_classes, test\_data$final\_result)

# Adjust the threshold as needed

# Evaluate the model

confusionMatrix(predicted\_classes, test\_data$final\_result)

Output:

> # Load necessary libraries

> library(caret)

> library(dplyr)

> # Read data from the Github link

> file\_path <- "C:/Users/MSI/OneDrive/Desktop/ML/oulad-students.csv"

> data <- read.csv(file\_path)

> # Assuming 'data' is your data frame

> subset\_data <- data[, c("id\_student", "date\_registration", "module\_presentation\_length", "studied\_credits", "num\_of\_prev\_attempts", "final\_result")]

> # Describe the variables

> summary(subset\_data)

id\_student date\_registration module\_presentation\_length studied\_credits

Min. : 3733 Min. :-322.00 Min. :234 Min. : 30.00

1st Qu.: 508573 1st Qu.:-100.00 1st Qu.:241 1st Qu.: 60.00

Median : 590310 Median : -57.00 Median :262 Median : 60.00

Mean : 706688 Mean : -69.41 Mean :256 Mean : 79.76

3rd Qu.: 644453 3rd Qu.: -29.00 3rd Qu.:268 3rd Qu.:120.00

Max. :2716795 Max. : 167.00 Max. :269 Max. :655.00

NA's :45

num\_of\_prev\_attempts final\_result

Min. :0.0000 Length:32593

1st Qu.:0.0000 Class :character

Median :0.0000 Mode :character

Mean :0.1632

3rd Qu.:0.0000

Max. :6.0000

> # View the first few rows of the data

> head(subset\_data)

id\_student date\_registration module\_presentation\_length studied\_credits num\_of\_prev\_attempts

1 11391 -159 268 240 0

2 28400 -53 268 60 0

3 30268 -92 268 60 0

4 31604 -52 268 60 0

5 32885 -176 268 60 0

6 38053 -110 268 60 0

final\_result

1 Pass

2 Pass

3 Withdrawn

4 Pass

5 Pass

6 Pass

> # Remove rows with missing values

> subset\_data <- na.omit(subset\_data)

> # Convert categorical variables to factors

> subset\_data$final\_result <- as.factor(subset\_data$final\_result)

> # Split the data into training and testing sets (80% training, 20% testing)

> set.seed(120) # For reproducibility

> train\_index <- createDataPartition(subset\_data$final\_result, p = 0.8, list = FALSE)

> train\_data <- subset\_data[train\_index, ]

> test\_data <- subset\_data[-train\_index, ]

> # Train the classification model (logistic regression)

> model <- glm(final\_result ~ ., data = train\_data, family = binomial)

> # Make predictions on the test data

> predictions <- predict(model, newdata = test\_data, type = "response")

> # Ensure predicted classes have the same levels as test data

> predicted\_classes <- factor(predicted\_classes, levels = levels(test\_data$final\_result))

> # Convert predicted classes to factor with the same levels as test\_data$final\_result

> predicted\_classes <- factor(predicted\_classes, levels = levels(test\_data$final\_result))

> # Evaluate the model

> confusionMatrix(predicted\_classes, test\_data$final\_result)

Confusion Matrix and Statistics

Reference

Prediction Distinction Fail Pass Withdrawn

Distinction 0 0 0 0

Fail 0 0 0 0

Pass 604 1409 2472 2023

Withdrawn 0 0 0 0

Overall Statistics

Accuracy : 0.3798

95% CI : (0.368, 0.3918)

No Information Rate : 0.3798

P-Value [Acc > NIR] : 0.5047

Kappa : 0

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: Distinction Class: Fail Class: Pass Class: Withdrawn

Sensitivity 0.00000 0.0000 1.0000 0.0000

Specificity 1.00000 1.0000 0.0000 1.0000

Pos Pred Value NaN NaN 0.3798 NaN

Neg Pred Value 0.90719 0.7835 NaN 0.6892

Prevalence 0.09281 0.2165 0.3798 0.3108

Detection Rate 0.00000 0.0000 0.3798 0.0000

Detection Prevalence 0.00000 0.0000 1.0000 0.0000

Balanced Accuracy 0.50000 0.5000 0.5000 0.5000

> # Evaluate the model

> confusionMatrix(predicted\_classes, test\_data$final\_result)

Confusion Matrix and Statistics

Reference

Prediction Distinction Fail Pass Withdrawn

Distinction 0 0 0 0

Fail 0 0 0 0

Pass 604 1409 2472 2023

Withdrawn 0 0 0 0

Overall Statistics

Accuracy : 0.3798

95% CI : (0.368, 0.3918)

No Information Rate : 0.3798

P-Value [Acc > NIR] : 0.5047

Kappa : 0

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: Distinction Class: Fail Class: Pass Class: Withdrawn

Sensitivity 0.00000 0.0000 1.0000 0.0000

Specificity 1.00000 1.0000 0.0000 1.0000

Pos Pred Value NaN NaN 0.3798 NaN

Neg Pred Value 0.90719 0.7835 NaN 0.6892

Prevalence 0.09281 0.2165 0.3798 0.3108

Detection Rate 0.00000 0.0000 0.3798 0.0000

Detection Prevalence 0.00000 0.0000 1.0000 0.0000

Balanced Accuracy 0.50000 0.5000 0.5000 0.5000