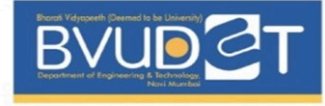




Bharati Vidyapeeth
Deemed to be University



Department of Engineering and Technology

Plot no. KC-1, Sector 3, Kharghar, Navi Mumbai-410210

Subject: Computing Lab - III | Project Based Learning (PBL) (3rd YEAR CSE-AIML 2023-2024)

Roll No: 11

Name: Kamran Khan

Class: CSE-AIML

Batch: B1

PRN: 2143110133

Date of Experiment: __ / __ / 2024

Marks (Out of 25):

Date of Submission: __ / __ / 2024

Aim:

Theory:

1. Required Libraries

I have imported the necessary libraries for data manipulation (tidyverse), data visualization (ggplot2), machine learning model training and evaluation (caret), and handling imbalanced datasets (DMwR2).

```
library(tidyverse)
```

```
## — Attaching core tidyverse packages — tidyverse 2.0.0 —
## ✓ dplyr      1.1.4      ✓ readr      2.1.5
## ✓ forcats    1.0.0      ✓ stringr    1.5.1
## ✓ ggplot2     3.5.0      ✓ tibble     3.2.1
## ✓ lubridate  1.9.3      ✓ tidyr      1.3.1
## ✓ purrr      1.0.2
## — Conflicts — tidyverse_conflicts() —
## X dplyr::filter() masks stats::filter()
## X dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

```
library(caret)
```

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

```
library(smotefamily)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

2. Load the Dataset

I have loaded the “diabetes.csv” dataset into the R environment for further analysis.

```
cd <- read.csv('diabetes.csv')
```

3. Exploratory Data Analysis

In this section, I perform exploratory data analysis to understand the structure and characteristics of the dataset. We check the first few rows, the data types of the variables, and the summary statistics.

```
head(cd)
```

```
##   Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI
## 1           6    148           72           35      0 33.6
## 2           1     85           66           29      0 26.6
## 3           8    183           64            0      0 23.3
## 4           1     89           66           23     94 28.1
## 5           0    137           40           35    168 43.1
## 6           5    116           74            0      0 25.6
##   DiabetesPedigreeFunction Age Outcome
## 1                   0.627  50      1
## 2                   0.351  31      0
## 3                   0.672  32      1
## 4                   0.167  21      0
## 5                   2.288  33      1
## 6                   0.201  30      0
```

```
str(cd)
```

```
## '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 ...
```

```
summary(cd)
```

```
## Pregnancies      Glucose      BloodPressure      SkinThickness
## Min.   : 0.000    Min.   : 0.0    Min.   : 0.00    Min.   : 0.00
## 1st Qu.: 1.000    1st Qu.: 99.0    1st Qu.: 62.00    1st Qu.: 0.00
## Median : 3.000    Median :117.0    Median : 72.00    Median :23.00
## Mean   : 3.845    Mean   :120.9    Mean   : 69.11    Mean   :20.54
## 3rd Qu.: 6.000    3rd Qu.:140.2    3rd Qu.: 80.00    3rd Qu.:32.00
## Max.   :17.000    Max.   :199.0    Max.   :122.00    Max.   :99.00
## Insulin      BMI      DiabetesPedigreeFunction      Age
## Min.   : 0.0    Min.   : 0.00    Min.   :0.0780    Min.   :21.00
## 1st Qu.: 0.0    1st Qu.:27.30    1st Qu.:0.2437    1st Qu.:24.00
## Median : 30.5    Median :32.00    Median :0.3725    Median :29.00
## Mean   : 79.8    Mean   :31.99    Mean   :0.4719    Mean   :33.24
## 3rd Qu.:127.2    3rd Qu.:36.60    3rd Qu.:0.6262    3rd Qu.:41.00
## Max.   :846.0    Max.   :67.10    Max.   :2.4200    Max.   :81.00
## Outcome
## Min.   :0.000
## 1st Qu.:0.000
## Median :0.000
## Mean   :0.349
## 3rd Qu.:1.000
## Max.   :1.000
```

4. Check for missing values

I check if there are any missing values in the dataset, which is an important step before proceeding with further analysis and modeling.

```
sum(is.na(cd))
```

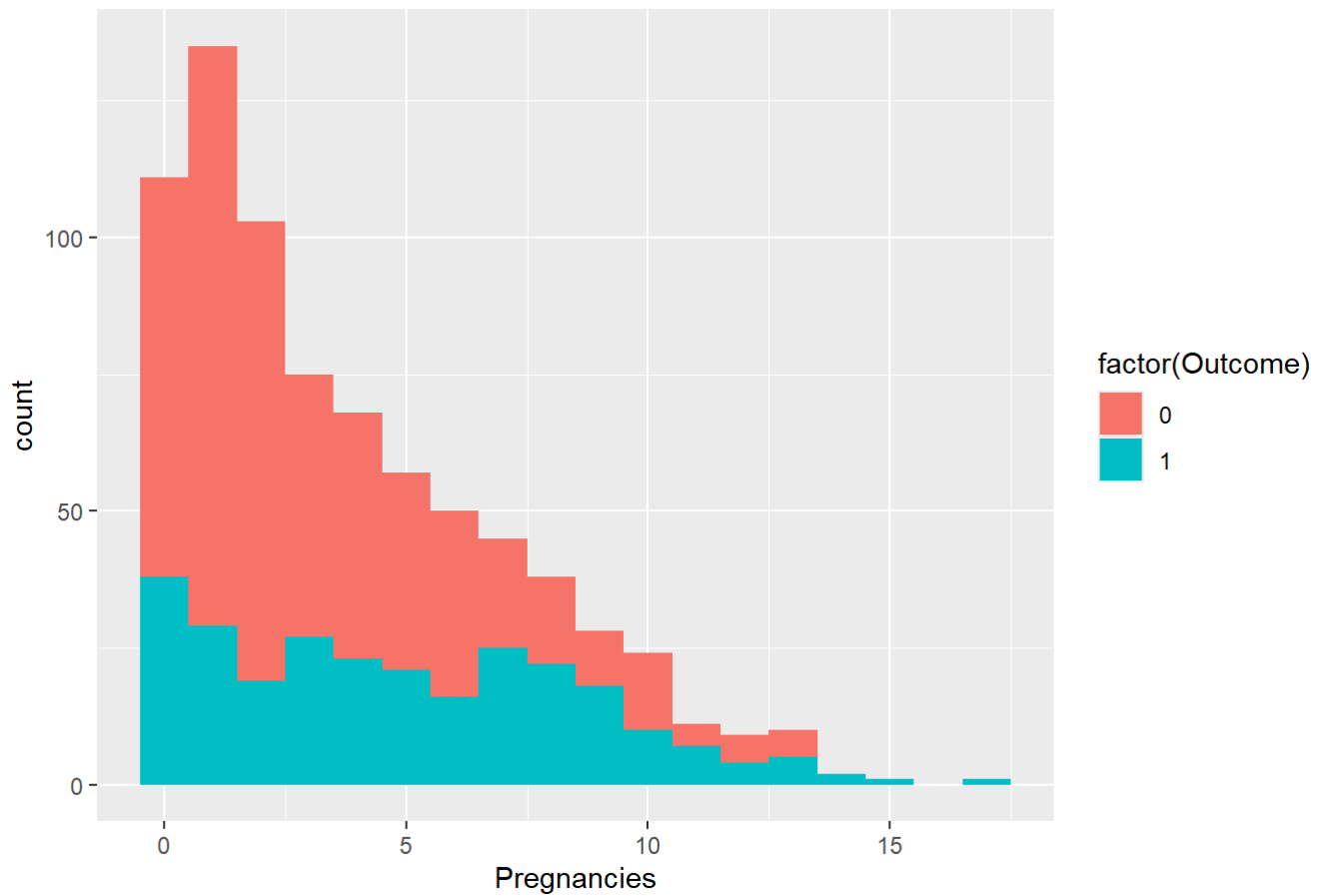
```
## [1] 0
```

5. Data Visualization

I create various histograms to visualize the distribution of different features in the dataset, such as Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, and Age. This helps us gain a better understanding of the data and identify any potential patterns or outliers.

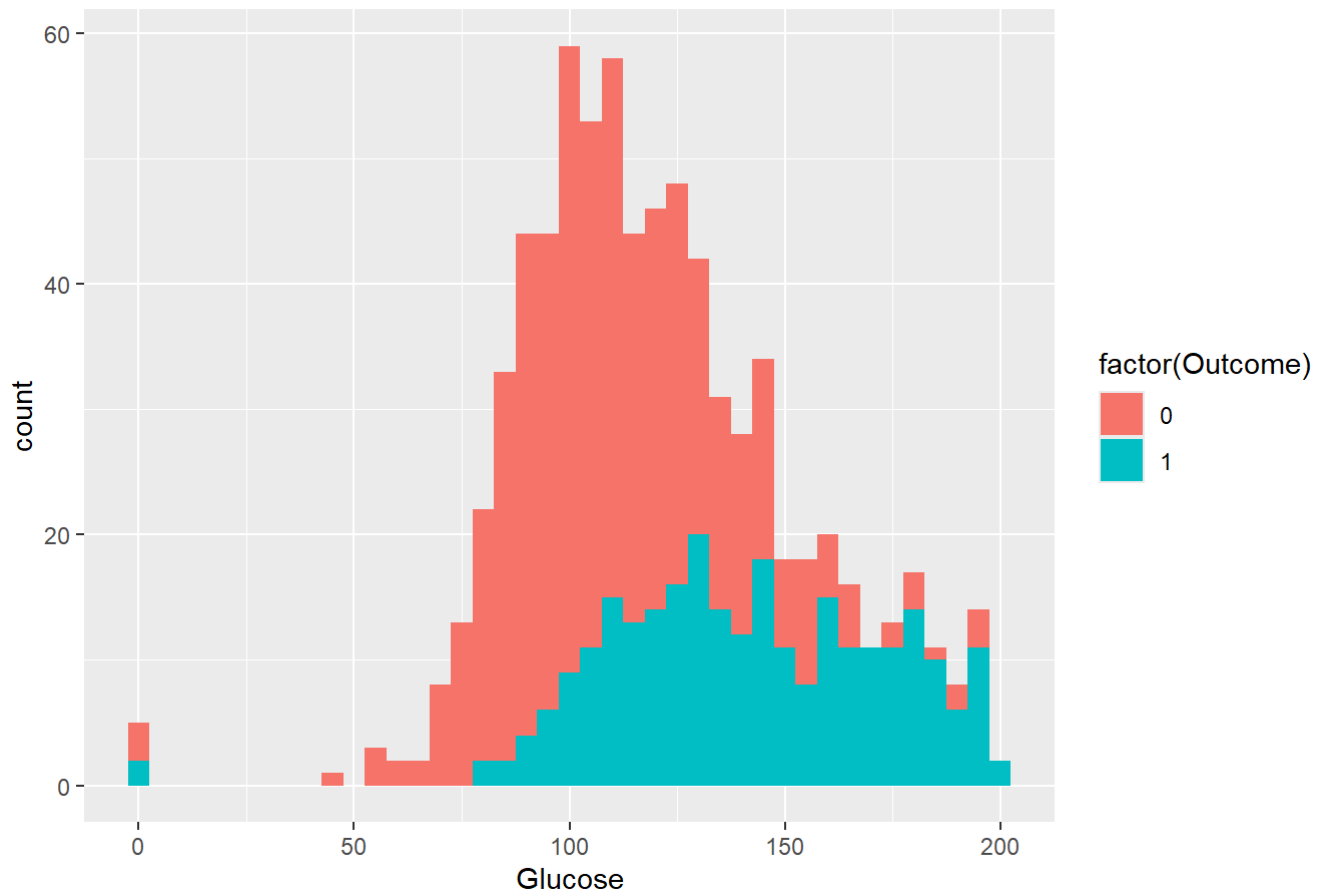
```
# Explore the data visually
ggplot(cd, aes(x = Pregnancies, fill = factor(Outcome))) +
  geom_histogram(binwidth = 1) +
  labs(title = "Histogram of Pregnancies")
```

Histogram of Pregnancies



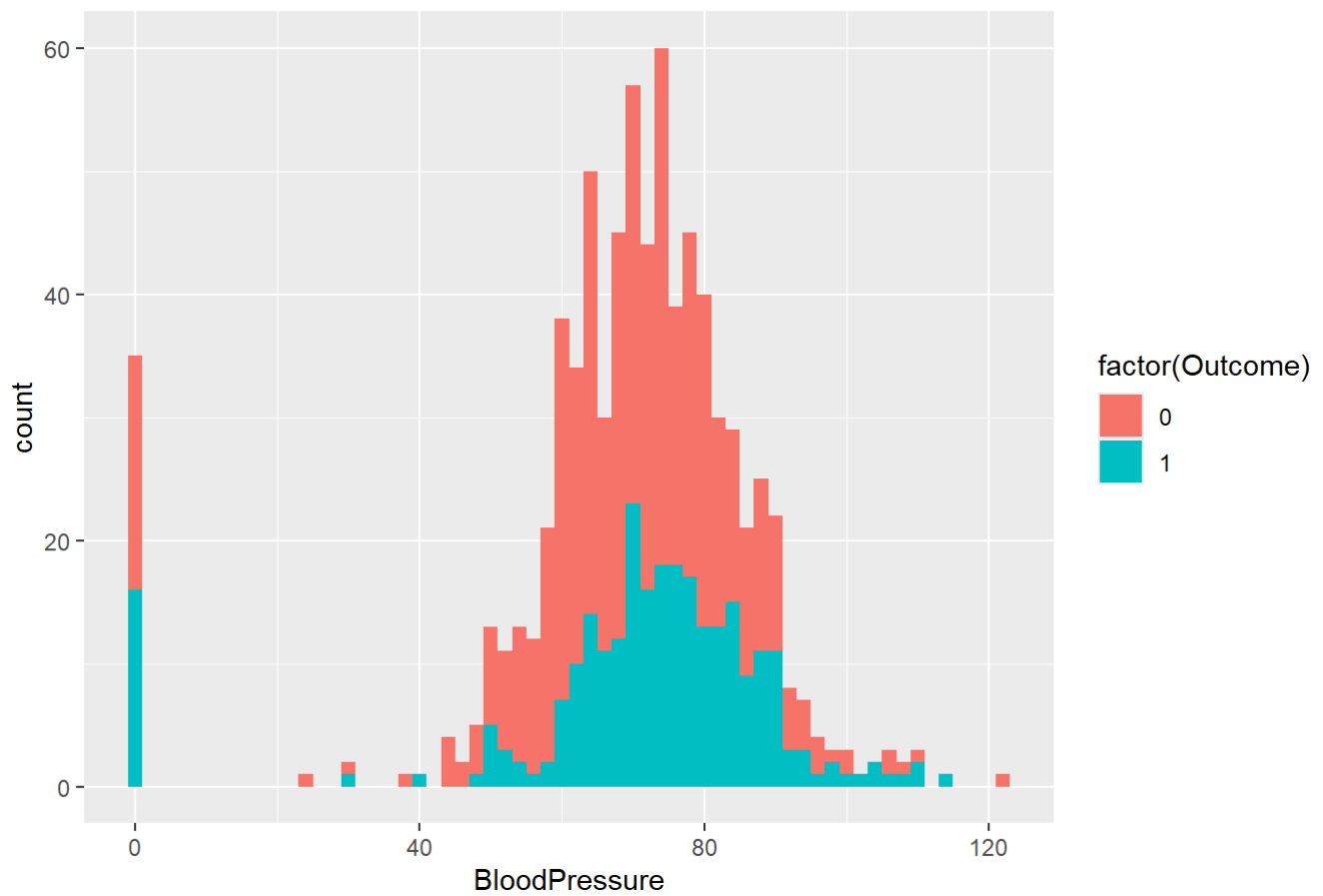
```
ggplot(cd, aes(x = Glucose, fill = factor(Outcome))) +  
  geom_histogram(binwidth = 5) +  
  labs(title = "Histogram of Glucose")
```

Histogram of Glucose



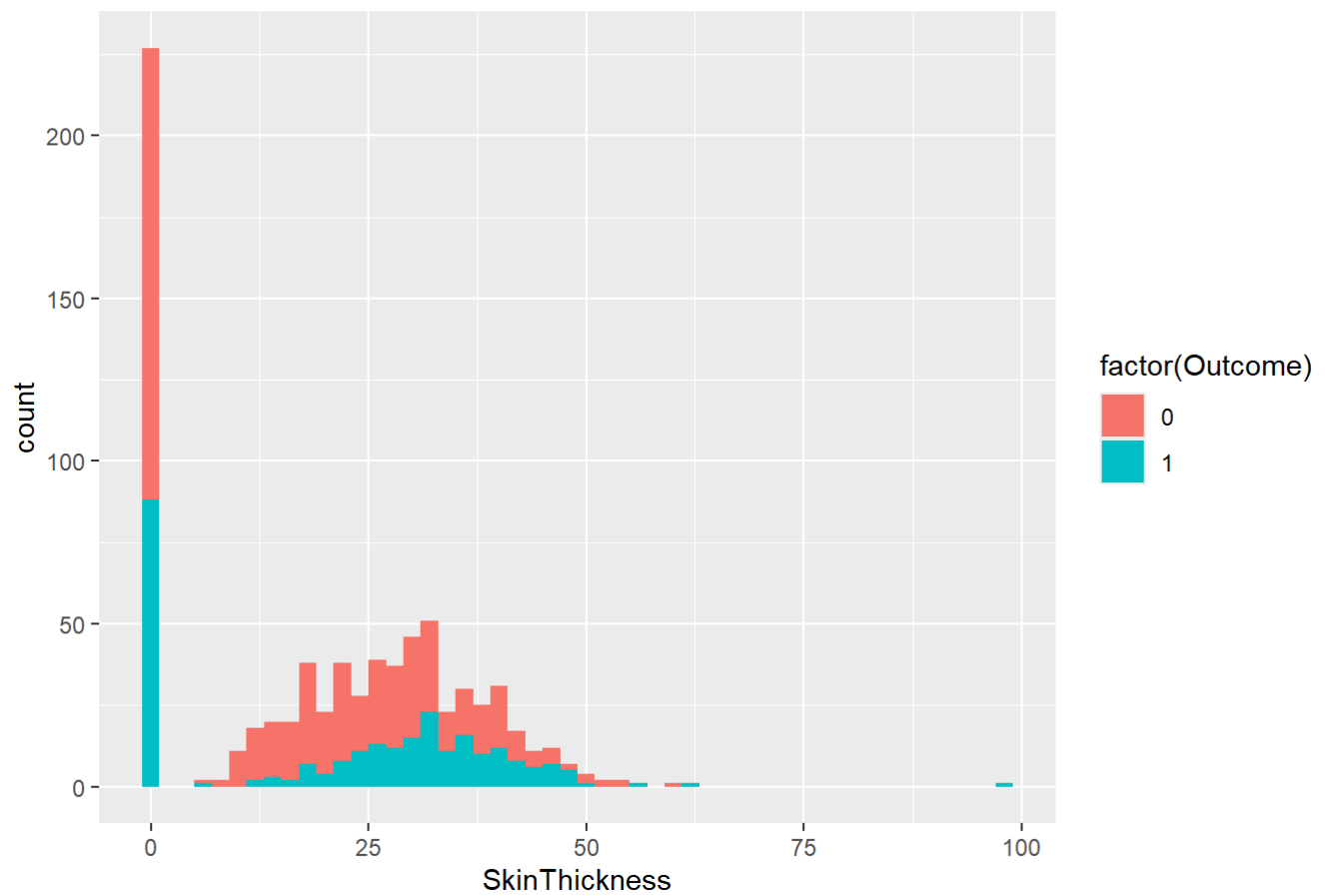
```
ggplot(cd, aes(x = BloodPressure, fill = factor(Outcome))) +  
  geom_histogram(binwidth = 2) +  
  labs(title = "Histogram of Blood Pressure")
```

Histogram of Blood Pressure



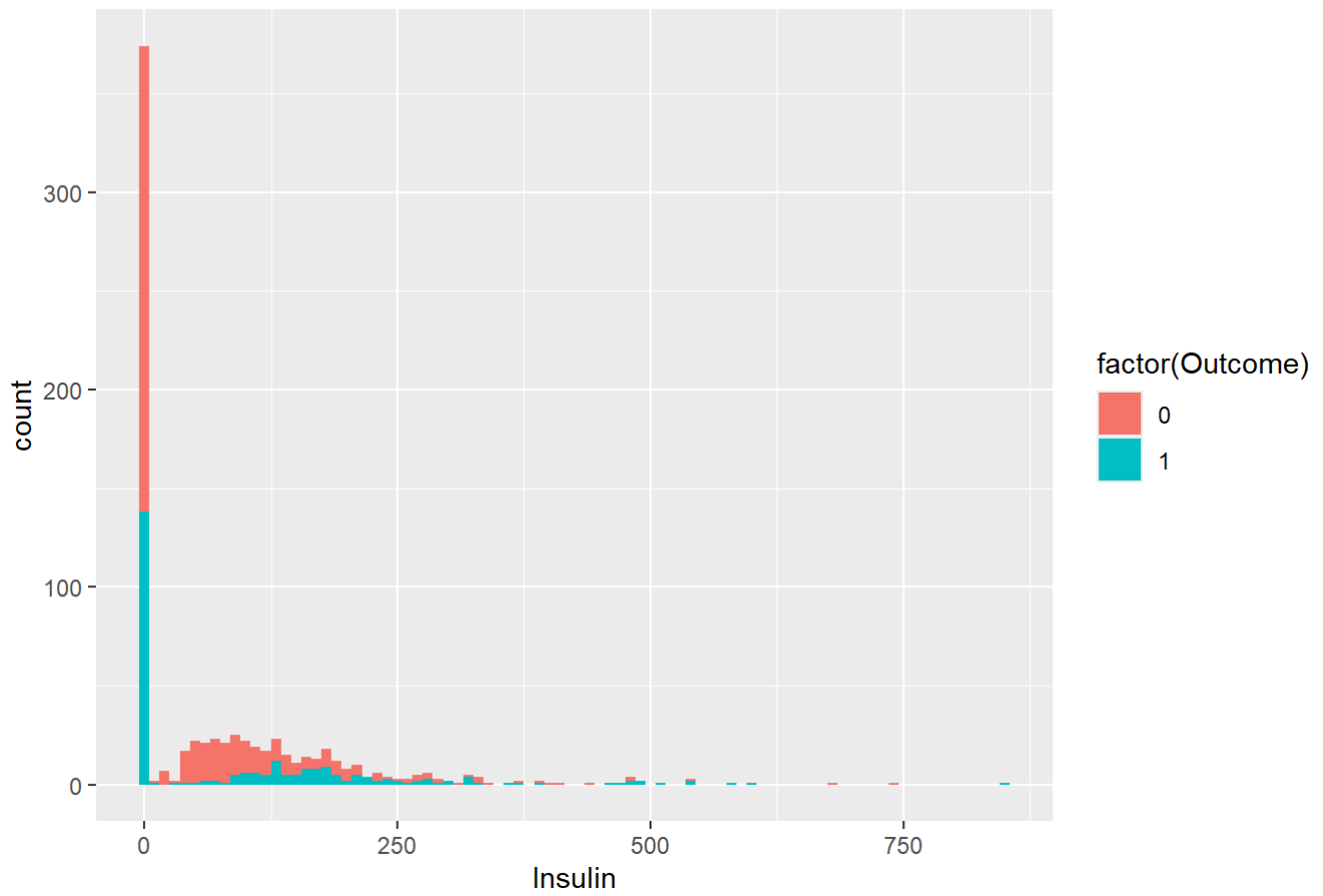
```
ggplot(cd, aes(x = SkinThickness, fill = factor(Outcome))) +  
  geom_histogram(binwidth = 2) +  
  labs(title = "Histogram of Skin Thickness")
```

Histogram of Skin Thickness



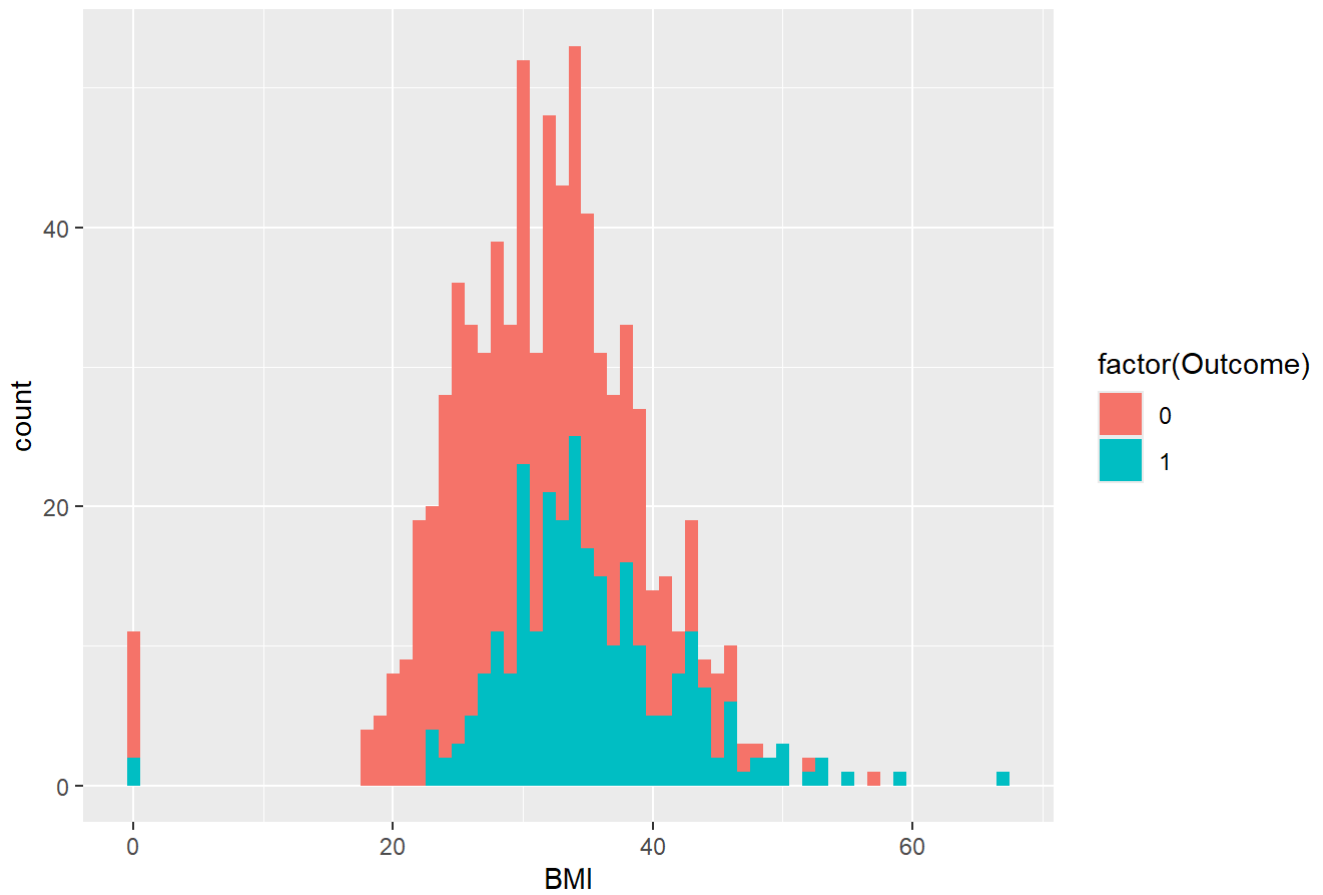
```
ggplot(cd, aes(x = Insulin, fill = factor(Outcome))) +  
  geom_histogram(binwidth = 10) +  
  labs(title = "Histogram of Insulin")
```

Histogram of Insulin



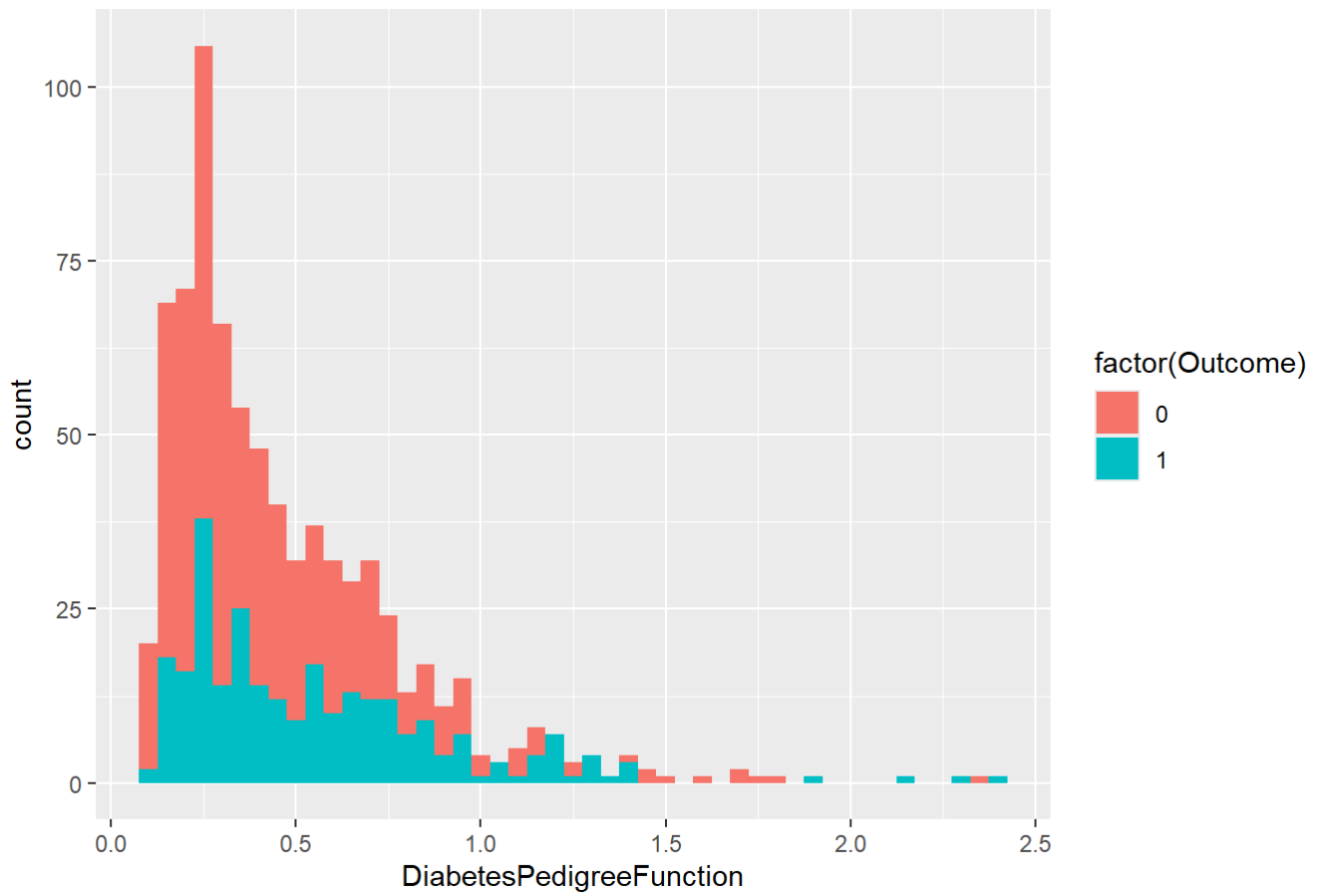
```
ggplot(cd, aes(x = BMI, fill = factor(Outcome))) +  
  geom_histogram(binwidth = 1) +  
  labs(title = "Histogram of BMI")
```


Histogram of BMI



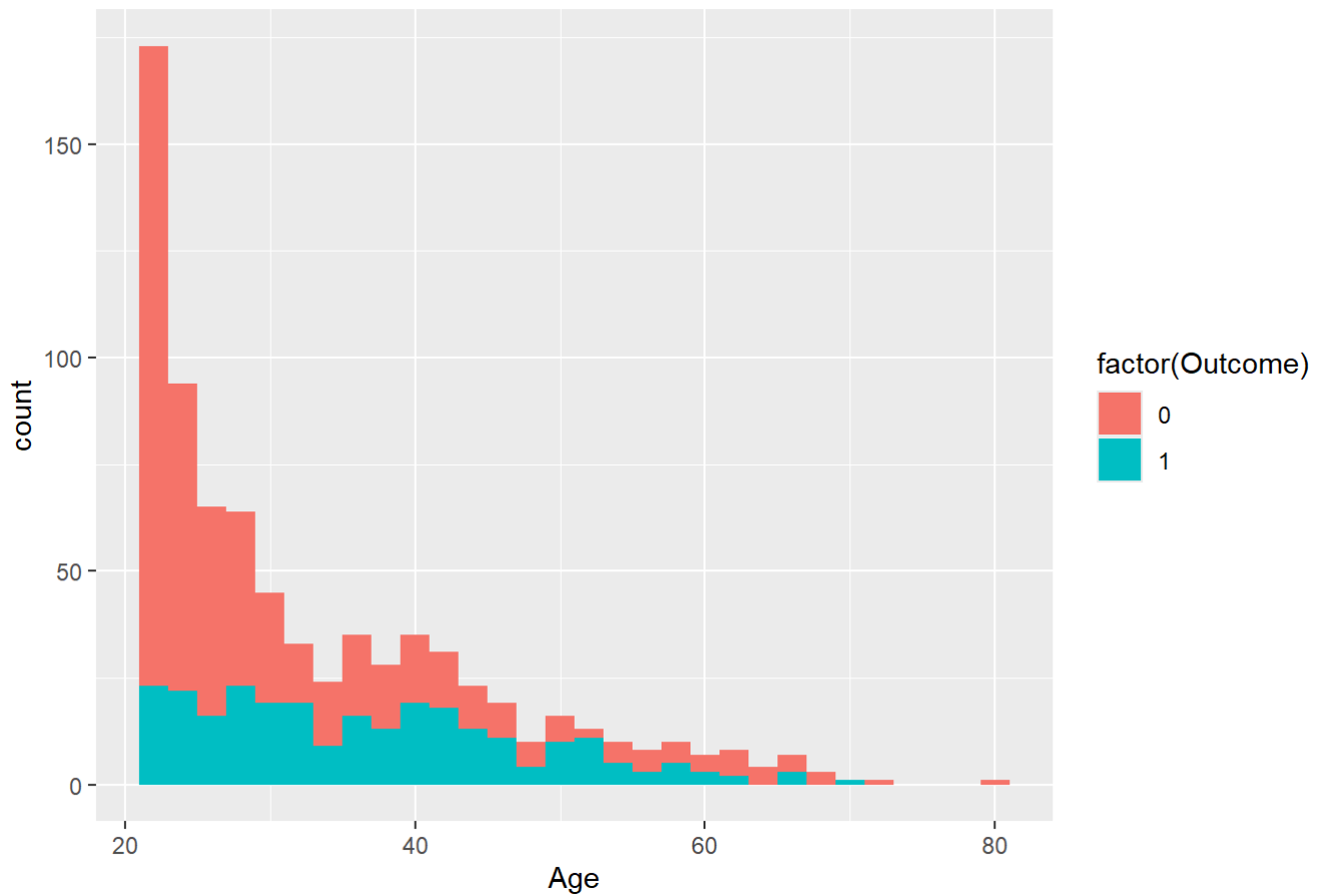
```
ggplot(cd, aes(x = DiabetesPedigreeFunction, fill = factor(Outcome))) +  
  geom_histogram(binwidth = 0.05) +  
  labs(title = "Histogram of Diabetes Pedigree Function")
```

Histogram of Diabetes Pedigree Function



```
ggplot(cd, aes(x = Age, fill = factor(Outcome))) +  
  geom_histogram(binwidth = 2) +  
  labs(title = "Histogram of Age")
```

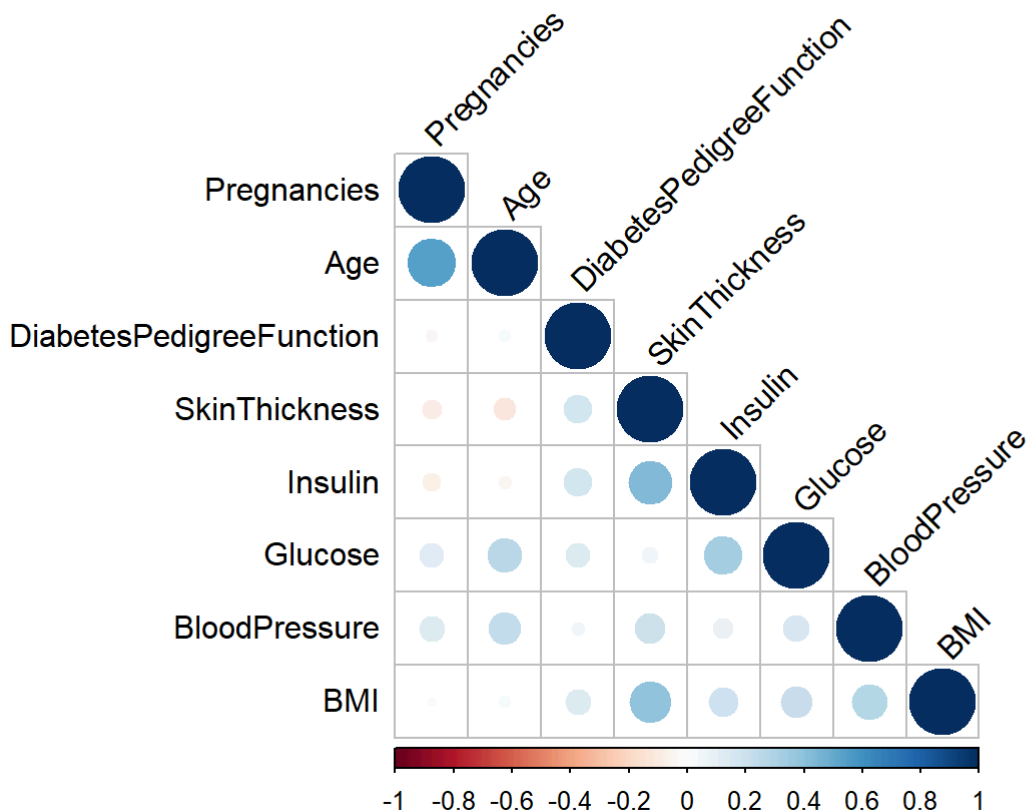
Histogram of Age



6. Correlation matrix

The correlation matrix analysis is performed to understand the relationships between the features in the dataset. This information can be used to identify potentially redundant or highly correlated features, which can then inform the feature selection or engineering process.

```
cor_matrix <- cor(cd[, -9])
corrplot(cor_matrix, method = "circle", order = "hclust",
         type = "lower", tl.col = "black", tl.srt = 45)
```



7. Feature engineering

In this step, I select the relevant features (Glucose, BMI, and Age) that will be used as input to the Logistic Regression model.

```
X <- cd[, c("Glucose", "BMI", "Age")]
y <- cd$Outcome
```

8. Handle imbalanced dataset

The diabetes dataset is likely to be imbalanced, as the number of people with diabetes and without diabetes may not be equal. We use the SMOTE (Synthetic Minority Over-sampling Technique) method to oversample the minority class (people with diabetes) to address the imbalance.

```
table(y)
```

```
## y
##   0   1
## 500 268
```

```
over_sampler <- SMOTE(X, y)
```

```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```

```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```

```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```

```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```

```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```

```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```

```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```

```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```

```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```

```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```

```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```

```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```

```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```

```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```

```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```

```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```

```
## Warning in matrix(unlist(P set[i, ]), sum dup, ncD, byrow = TRUE): non-empt
```

[illegible]

[illegible]

```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P set[i, ]), sum dup, ncD, byrow = TRUE): non-empt
```


[illegible]

```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty  
## data for zero-extent matrix
```



```
## Warning in matrix(unlist(P set[i, ]), sum dup, ncD, byrow = TRUE): non-empt
```


[illegible]

[illegible]

[illegible]

[illegible]


```
## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty
## data for zero-extent matrix

## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty
## data for zero-extent matrix

## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty
## data for zero-extent matrix

## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty
## data for zero-extent matrix

## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty
## data for zero-extent matrix

## Warning in matrix(unlist(P_set[i, ]), sum_dup, ncD, byrow = TRUE): non-empty
## data for zero-extent matrix
```

```
X_over <- over_sampler$data[, 1:3]
y_over <- over_sampler$data[, 4]
```

9. Split the data into training and testing sets

I split the dataset into training and testing sets, with 70% of the data used for training the model and 30% for evaluating its performance.

```
set.seed(2)
train_index <- createDataPartition(y_over, p = 0.7, list = FALSE)
X_train <- X_over[train_index, ]
y_train <- y_over[train_index]
X_test <- X_over[-train_index, ]
y_test <- y_over[-train_index]
```

8. Logistic Regression

I train a Logistic Regression model using the training data. Logistic Regression is a suitable algorithm for this binary classification problem, as it can predict the probability of a person having diabetes or not.

```
logit_model <- train(x = X_train, y = as.factor(y_train), method = "glm", family = "binomial",
  trControl = trainControl(method = "none"))
```

9. Evaluate the model

Finally, we evaluate the performance of the Logistic Regression model on the testing data using various metrics such as accuracy, precision, recall, and F1-score. These metrics provide a comprehensive assessment of the model's performance and help us understand its strengths and weaknesses.

```
y_pred_lr <- predict(logit_model, newdata = X_test)
confusion_matrix <- confusionMatrix(y_pred_lr, as.factor(y_test))

accuracy <- confusion_matrix$overall['Accuracy']
precision <- confusion_matrix$byClass['Pos Pred Value']
recall <- confusion_matrix$byClass['Sensitivity']
f1_score <- 2 * precision * recall / (precision + recall)

print(paste("Accuracy:", accuracy))
```

```
## [1] "Accuracy: 0.782608695652174"
```

```
print(paste("Precision:", precision))
```

```
## [1] "Precision: 0.808641975308642"
```

```
print(paste("Recall:", recall))
```

```
## [1] "Recall: 0.873333333333333"
```

```
print(paste("F1-score:", f1_score))
```

```
## [1] "F1-score: 0.83974358974359"
```

Conclusion:

Signature of Lab Incharge
(Prof. Supriya Khaitan)