LogisticRegression_Model.R

rahul

2024-12-03

```
# Load required libraries for Risk Assessment
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
## Warning: package 'tidyr' was built under R version 4.3.3
```

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

```
## Warning: package 'stringr' was built under R version 4.3.3
```

```
## Warning: package 'forcats' was built under R version 4.3.3
```

```
## Warning: package 'lubridate' was built under R version 4.3.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v 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 become errors
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.3
```

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

```r
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.3.3
```

```
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
##
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.3
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.3.3
```

```
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```r
library(ggcorrplot)
```

```
## Warning: package 'ggcorrplot' was built under R version 4.3.3
```

```r
library(readxl)
```

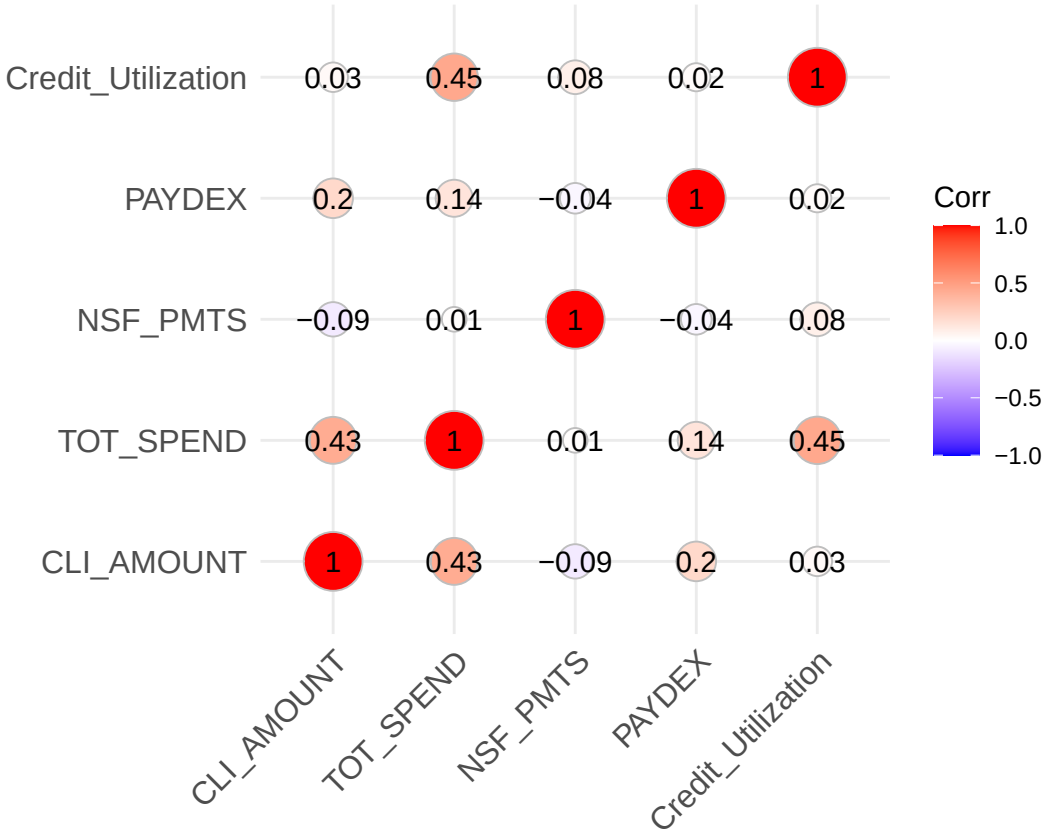
```
## Warning: package 'readxl' was built under R version 4.3.3
```

```r
# Load dataset
data <- read_excel("ACCT_Monitoring_FinalData.xlsx")

# Data Preprocessing
# 1. Handle missing values
data <- data %>% mutate(across(where(is.numeric), ~ifelse(is.na(.), mean(., na.rm = TRUE), .)))

# 2. Calculate derived feature for Credit Utilization
data <- data %>% mutate(Credit_Utilization = TOT_SPEND / CREDIT_LIMIT)
```

```r
# 3. Create a target variable for risk assessment
# High risk is defined as Credit Utilization > 0.8
data <- data %>% mutate(Risk_Level = ifelse(Credit_Utilization > 0.8, 1, 0))

# 4. Select relevant features
selected_features <- c("CLI_AMOUNT", "TOT_SPEND", "NSF_PMTS", "PAYDEX",
                       "Credit_Utilization", "Risk_Level")
data <- data %>% select(all_of(selected_features))

# Ensure target variable is a factor
data$Risk_Level <- as.factor(data$Risk_Level)

# Train-Test Split
set.seed(123)
train_index <- createDataPartition(data$Risk_Level, p = 0.7, list = FALSE)
train_data <- data[train_index, ]
test_data <- data[-train_index, ]

# Correlation:
cor_matrix <- cor(train_data %>% select(where(is.numeric)))
print(cor_matrix)
```

```
##                     CLI_AMOUNT   TOT_SPEND     NSF_PMTS      PAYDEX
## CLI_AMOUNT          1.00000000 0.434303486 -0.092004516  0.20045405
## TOT_SPEND           0.43430349 1.000000000  0.005135212  0.13964078
## NSF_PMTS           -0.09200452 0.005135212  1.000000000 -0.03743348
## PAYDEX              0.20045405 0.139640783 -0.037433485  1.00000000
## Credit_Utilization  0.02645997 0.446951872  0.078267738  0.01983681
##                    Credit_Utilization
## CLI_AMOUNT                  0.02645997
## TOT_SPEND                   0.44695187
## NSF_PMTS                    0.07826774
## PAYDEX                      0.01983681
## Credit_Utilization          1.00000000
```

```r
# Visualize correlations
ggcorrplot(cor_matrix, method = "circle", lab = TRUE)
```




```
# Fit a logistic regression model
model_glm <- glm(Risk_Level ~ ., data = train_data, family = "binomial", control = glm.control(maxit =
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
# Calculate VIF for logistic regression model
vif_results <- vif(model_glm)
print(vif_results)
```

```
##         CLI_AMOUNT          TOT_SPEND          NSF_PMTS            PAYDEX
##           4.723723           4.785640           1.012916           1.051277
## Credit_Utilization
##           1.078458
```

```
# Predict on the test data
pred_glm <- predict(model_glm, newdata = test_data, type = "response")

# Evaluate with ROC and AUC
roc_glm <- roc(test_data$Risk_Level, pred_glm)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

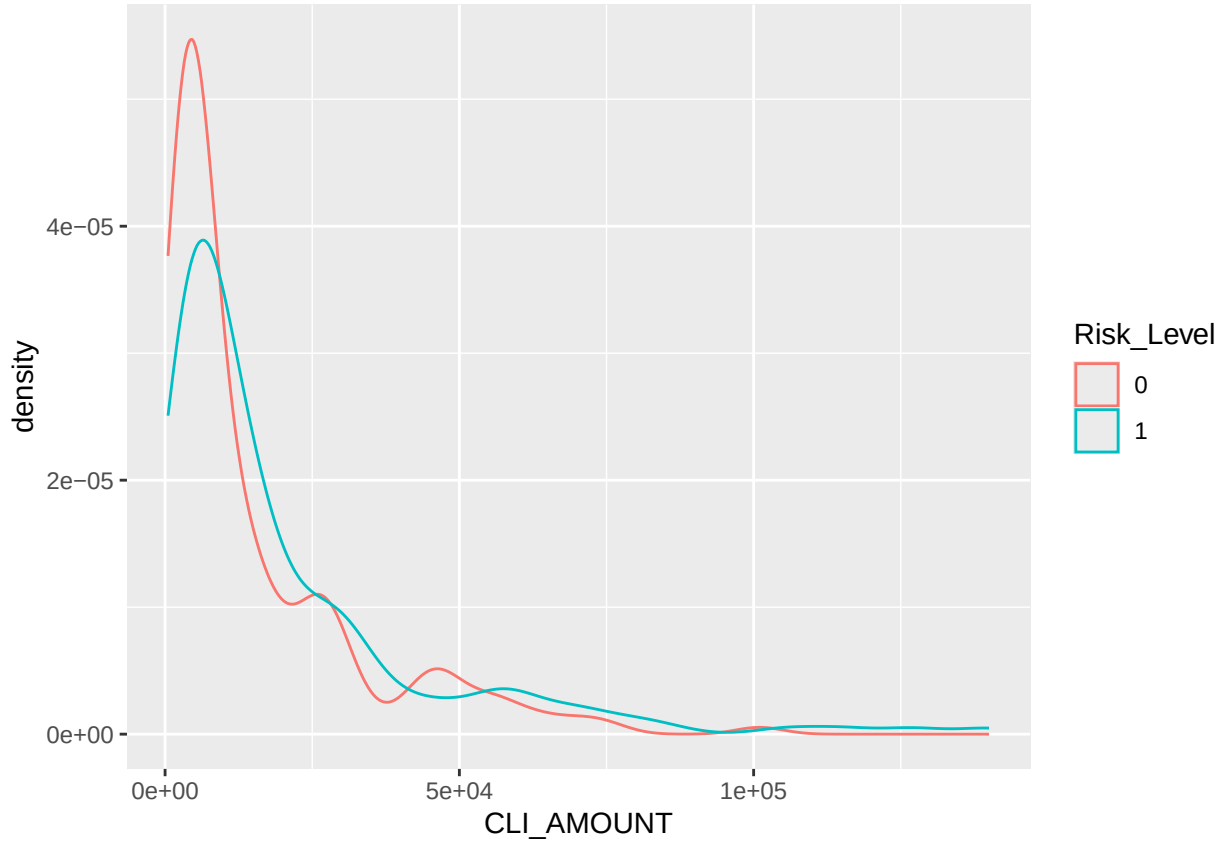
```r
print(paste("AUC for Logistic Regression (GLM): ", auc(roc_glm)))
```

```
## [1] "AUC for Logistic Regression (GLM):  1"
```

```r
# Model Summary
summary(model_glm)
```

```
##
## Call:
## glm(formula = Risk_Level ~ ., family = "binomial", data = train_data,
##     control = glm.control(maxit = 1000))
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -4.422e+02  5.436e+05  -0.001    0.999
## CLI_AMOUNT         -2.027e-04  5.850e+00   0.000    1.000
## TOT_SPEND           1.572e-03  8.168e+00   0.000    1.000
## NSF_PMTS           -5.385e+00  1.107e+06   0.000    1.000
## PAYDEX              2.580e-02  4.050e+03   0.000    1.000
## Credit_Utilization  5.181e+02  5.189e+05   0.001    0.999
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4.8556e+02  on 350  degrees of freedom
## Residual deviance: 1.1976e-09  on 345  degrees of freedom
## AIC: 12
##
## Number of Fisher Scoring iterations: 34
```

```r
ggplot(train_data, aes(x = CLI_AMOUNT, color = Risk_Level)) + geom_density()
```




```
# Predictions
pred_probs <- predict(model_glm, test_data, type = "response")
pred_classes <- ifelse(pred_probs > 0.5, 1, 0)

# Performance Metrics
conf_matrix <- confusionMatrix(factor(pred_classes), test_data$Risk_Level)
print(conf_matrix)
```

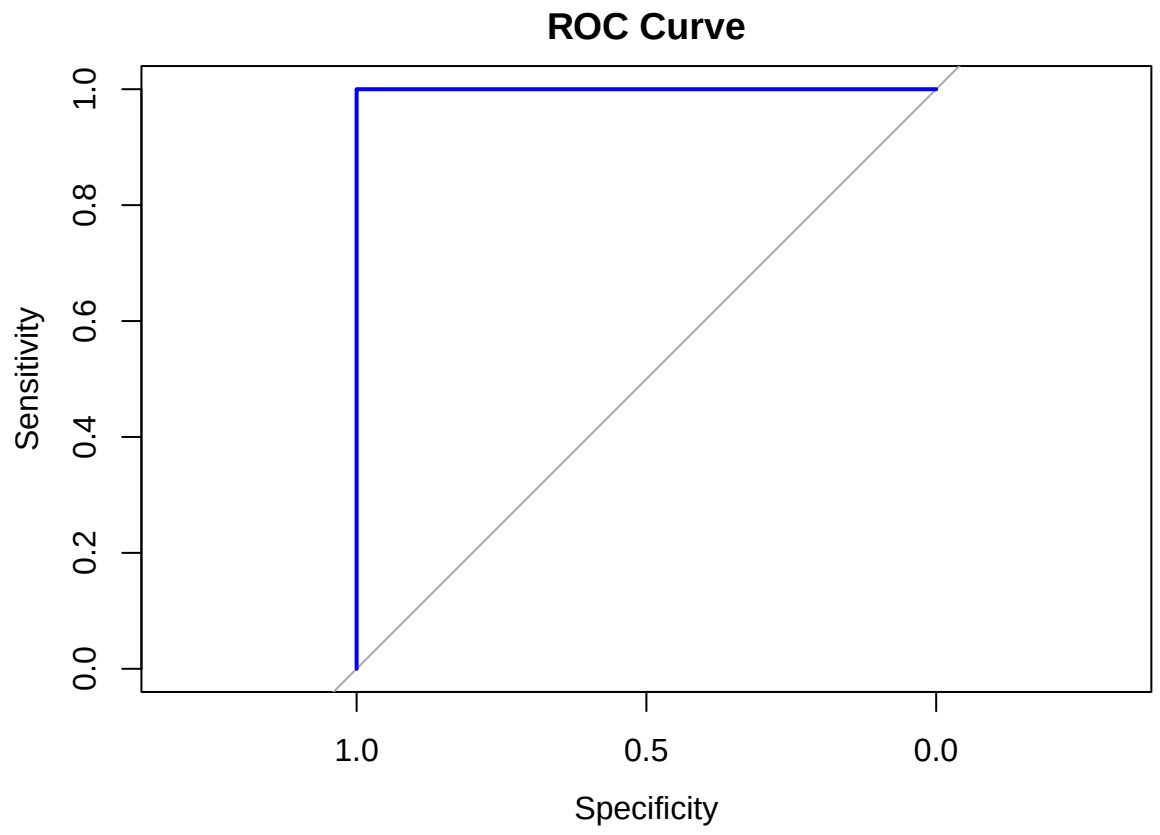
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 79  0
##          1  0 70
##
##                Accuracy : 1
##                  95% CI : (0.9755, 1)
##     No Information Rate : 0.5302
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
```

```
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5302
##          Detection Rate : 0.5302
##    Detection Prevalence : 0.5302
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 0
##
```

```
# AUC-ROC
roc_curve <- roc(as.numeric(test_data$Risk_Level), pred_probs)
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
auc <- auc(roc_curve)
plot(roc_curve, col = "blue", main = "ROC Curve")
```




```
print(paste("AUC:", auc))
```

```
## [1] "AUC: 1"
```