1.Predicting loan deafulters(Loan\_data.csv)

Geena P George

2024-09-19

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

## Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.

## Load the libraries and dataset

library(ggplot2)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(caret)

## Loading required package: lattice

library(randomForest)

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

library(e1071)  
library(caTools)

loan\_data <- read.csv("E:\\RCSS\\sem 3\\R Programming\\ComponentII\_Datasets\\loan\_data\_set.csv")

head(loan\_data)

## Loan\_ID Gender Married Dependents Education Self\_Employed ApplicantIncome  
## 1 LP001002 Male No 0 Graduate No 5849  
## 2 LP001003 Male Yes 1 Graduate No 4583  
## 3 LP001005 Male Yes 0 Graduate Yes 3000  
## 4 LP001006 Male Yes 0 Not Graduate No 2583  
## 5 LP001008 Male No 0 Graduate No 6000  
## 6 LP001011 Male Yes 2 Graduate Yes 5417  
## CoapplicantIncome LoanAmount Loan\_Amount\_Term Credit\_History Property\_Area  
## 1 0 NA 360 1 Urban  
## 2 1508 128 360 1 Rural  
## 3 0 66 360 1 Urban  
## 4 2358 120 360 1 Urban  
## 5 0 141 360 1 Urban  
## 6 4196 267 360 1 Urban  
## Loan\_Status  
## 1 Y  
## 2 N  
## 3 Y  
## 4 Y  
## 5 Y  
## 6 Y

# Check the structure of the dataset  
str(loan\_data)

## 'data.frame': 614 obs. of 13 variables:  
## $ Loan\_ID : chr "LP001002" "LP001003" "LP001005" "LP001006" ...  
## $ Gender : chr "Male" "Male" "Male" "Male" ...  
## $ Married : chr "No" "Yes" "Yes" "Yes" ...  
## $ Dependents : chr "0" "1" "0" "0" ...  
## $ Education : chr "Graduate" "Graduate" "Graduate" "Not Graduate" ...  
## $ Self\_Employed : chr "No" "No" "Yes" "No" ...  
## $ ApplicantIncome : int 5849 4583 3000 2583 6000 5417 2333 3036 4006 12841 ...  
## $ CoapplicantIncome: num 0 1508 0 2358 0 ...  
## $ LoanAmount : int NA 128 66 120 141 267 95 158 168 349 ...  
## $ Loan\_Amount\_Term : int 360 360 360 360 360 360 360 360 360 360 ...  
## $ Credit\_History : int 1 1 1 1 1 1 1 0 1 1 ...  
## $ Property\_Area : chr "Urban" "Rural" "Urban" "Urban" ...  
## $ Loan\_Status : chr "Y" "N" "Y" "Y" ...

# View summary statistics  
summary(loan\_data)

## Loan\_ID Gender Married Dependents   
## Length:614 Length:614 Length:614 Length:614   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## Education Self\_Employed ApplicantIncome CoapplicantIncome  
## Length:614 Length:614 Min. : 150 Min. : 0   
## Class :character Class :character 1st Qu.: 2878 1st Qu.: 0   
## Mode :character Mode :character Median : 3812 Median : 1188   
## Mean : 5403 Mean : 1621   
## 3rd Qu.: 5795 3rd Qu.: 2297   
## Max. :81000 Max. :41667   
##   
## LoanAmount Loan\_Amount\_Term Credit\_History Property\_Area   
## Min. : 9.0 Min. : 12 Min. :0.0000 Length:614   
## 1st Qu.:100.0 1st Qu.:360 1st Qu.:1.0000 Class :character   
## Median :128.0 Median :360 Median :1.0000 Mode :character   
## Mean :146.4 Mean :342 Mean :0.8422   
## 3rd Qu.:168.0 3rd Qu.:360 3rd Qu.:1.0000   
## Max. :700.0 Max. :480 Max. :1.0000   
## NA's :22 NA's :14 NA's :50   
## Loan\_Status   
## Length:614   
## Class :character   
## Mode :character   
##   
##   
##   
##

# Check for missing values  
colSums(is.na(loan\_data))

## Loan\_ID Gender Married Dependents   
## 0 0 0 0   
## Education Self\_Employed ApplicantIncome CoapplicantIncome   
## 0 0 0 0   
## LoanAmount Loan\_Amount\_Term Credit\_History Property\_Area   
## 22 14 50 0   
## Loan\_Status   
## 0

## Data Preprocessing

Before modeling, we need to clean the data. This includes handling missing values and converting categorical variables into factors.

loan\_data <- loan\_data %>% select(-Loan\_ID)  
# Fill missing values or remove rows with missing data (this depends on the nature of the data)  
# Here we assume numerical columns can be filled with median and categorical with the mode  
loan\_data <- loan\_data %>%  
 mutate\_if(is.numeric, ~ ifelse(is.na(.), median(., na.rm = TRUE), .)) %>%  
 mutate\_if(is.factor, ~ ifelse(is.na(.), as.character(stats::Mode(.)), .))  
#second check  
colSums(is.na(loan\_data))

## Gender Married Dependents Education   
## 0 0 0 0   
## Self\_Employed ApplicantIncome CoapplicantIncome LoanAmount   
## 0 0 0 0   
## Loan\_Amount\_Term Credit\_History Property\_Area Loan\_Status   
## 0 0 0 0

str(loan\_data)

## 'data.frame': 614 obs. of 12 variables:  
## $ Gender : chr "Male" "Male" "Male" "Male" ...  
## $ Married : chr "No" "Yes" "Yes" "Yes" ...  
## $ Dependents : chr "0" "1" "0" "0" ...  
## $ Education : chr "Graduate" "Graduate" "Graduate" "Not Graduate" ...  
## $ Self\_Employed : chr "No" "No" "Yes" "No" ...  
## $ ApplicantIncome : int 5849 4583 3000 2583 6000 5417 2333 3036 4006 12841 ...  
## $ CoapplicantIncome: num 0 1508 0 2358 0 ...  
## $ LoanAmount : num 128 128 66 120 141 267 95 158 168 349 ...  
## $ Loan\_Amount\_Term : num 360 360 360 360 360 360 360 360 360 360 ...  
## $ Credit\_History : num 1 1 1 1 1 1 1 0 1 1 ...  
## $ Property\_Area : chr "Urban" "Rural" "Urban" "Urban" ...  
## $ Loan\_Status : chr "Y" "N" "Y" "Y" ...

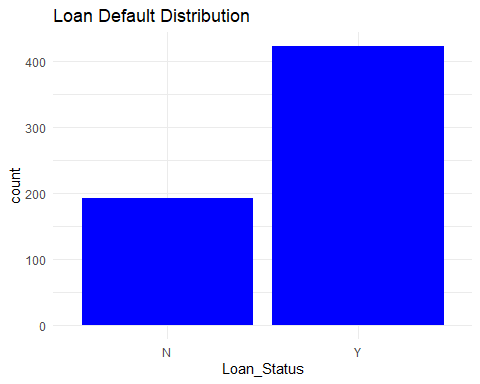
# Convert all character variables to factors  
loan\_data <- loan\_data %>%  
 mutate\_if(is.character, as.factor)  
  
  
str(loan\_data)

## 'data.frame': 614 obs. of 12 variables:  
## $ Gender : Factor w/ 3 levels "","Female","Male": 3 3 3 3 3 3 3 3 3 3 ...  
## $ Married : Factor w/ 3 levels "","No","Yes": 2 3 3 3 2 3 3 3 3 3 ...  
## $ Dependents : Factor w/ 5 levels "","0","1","2",..: 2 3 2 2 2 4 2 5 4 3 ...  
## $ Education : Factor w/ 2 levels "Graduate","Not Graduate": 1 1 1 2 1 1 2 1 1 1 ...  
## $ Self\_Employed : Factor w/ 3 levels "","No","Yes": 2 2 3 2 2 3 2 2 2 2 ...  
## $ ApplicantIncome : int 5849 4583 3000 2583 6000 5417 2333 3036 4006 12841 ...  
## $ CoapplicantIncome: num 0 1508 0 2358 0 ...  
## $ LoanAmount : num 128 128 66 120 141 267 95 158 168 349 ...  
## $ Loan\_Amount\_Term : num 360 360 360 360 360 360 360 360 360 360 ...  
## $ Credit\_History : num 1 1 1 1 1 1 1 0 1 1 ...  
## $ Property\_Area : Factor w/ 3 levels "Rural","Semiurban",..: 3 1 3 3 3 3 3 2 3 2 ...  
## $ Loan\_Status : Factor w/ 2 levels "N","Y": 2 1 2 2 2 2 2 1 2 1 ...

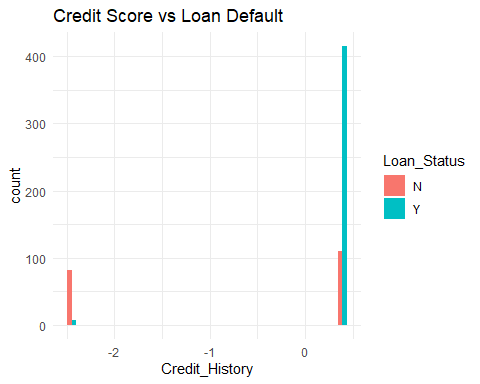
# Scaling the numeric features  
numeric\_features <- sapply(loan\_data, is.numeric)  
loan\_data[numeric\_features] <- scale(loan\_data[numeric\_features])  
  
# Check dataset again after preprocessing  
summary(loan\_data)

## Gender Married Dependents Education Self\_Employed  
## : 13 : 3 : 15 Graduate :480 : 32   
## Female:112 No :213 0 :345 Not Graduate:134 No :500   
## Male :489 Yes:398 1 :102 Yes: 82   
## 2 :101   
## 3+: 51   
##   
## ApplicantIncome CoapplicantIncome LoanAmount Loan\_Amount\_Term  
## Min. :-0.85995 Min. :-0.5540 Min. :-1.6259 Min. :-5.128   
## 1st Qu.:-0.41348 1st Qu.:-0.5540 1st Qu.:-0.5410 1st Qu.: 0.273   
## Median :-0.26043 Median :-0.1479 Median :-0.2111 Median : 0.273   
## Mean : 0.00000 Mean : 0.0000 Mean : 0.0000 Mean : 0.000   
## 3rd Qu.: 0.06409 3rd Qu.: 0.2310 3rd Qu.: 0.2259 3rd Qu.: 0.273   
## Max. :12.37453 Max. :13.6850 Max. : 6.5898 Max. : 2.136   
## Credit\_History Property\_Area Loan\_Status  
## Min. :-2.4268 Rural :179 N:192   
## 1st Qu.: 0.4114 Semiurban:233 Y:422   
## Median : 0.4114 Urban :202   
## Mean : 0.0000   
## 3rd Qu.: 0.4114   
## Max. : 0.4114

#visualization  
# Visualizing the distribution of the target variable  
ggplot(loan\_data, aes(x = Loan\_Status)) +  
 geom\_bar(fill = "blue") +  
 theme\_minimal() +  
 ggtitle("Loan Default Distribution")



# Visualizing relationships between variables (e.g., credit score vs loan default)  
ggplot(loan\_data, aes(x = Credit\_History, fill = Loan\_Status)) +  
 geom\_histogram(position = "dodge", bins = 30) +  
 theme\_minimal() +  
 ggtitle("Credit Score vs Loan Default")



# Split the data into training and testing sets  
set.seed(123)  
split <- sample.split(loan\_data$Loan\_Status, SplitRatio = 0.7)  
train\_data <- subset(loan\_data, split == TRUE)  
test\_data <- subset(loan\_data, split == FALSE)  
  
# Logistic Regression  
log\_model <- glm(Loan\_Status ~ ., data = train\_data, family = binomial)  
summary(log\_model)

##   
## Call:  
## glm(formula = Loan\_Status ~ ., family = binomial, data = train\_data)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 12.865032 882.744008 0.015 0.98837   
## GenderFemale 0.777595 0.850632 0.914 0.36064   
## GenderMale 0.659973 0.804191 0.821 0.41184   
## MarriedNo -13.934746 882.744131 -0.016 0.98741   
## MarriedYes -13.440075 882.744099 -0.015 0.98785   
## Dependents0 0.524778 1.064649 0.493 0.62207   
## Dependents1 0.172721 1.082372 0.160 0.87322   
## Dependents2 0.929757 1.104558 0.842 0.39993   
## Dependents3+ 0.200864 1.134558 0.177 0.85948   
## EducationNot Graduate -0.328517 0.306093 -1.073 0.28316   
## Self\_EmployedNo 0.035140 0.493601 0.071 0.94325   
## Self\_EmployedYes 0.003608 0.589427 0.006 0.99512   
## ApplicantIncome 0.008936 0.179226 0.050 0.96024   
## CoapplicantIncome -0.119744 0.106614 -1.123 0.26137   
## LoanAmount -0.032849 0.166967 -0.197 0.84403   
## Loan\_Amount\_Term -0.255950 0.149758 -1.709 0.08743 .   
## Credit\_History 1.244280 0.165656 7.511 5.86e-14 \*\*\*  
## Property\_AreaSemiurban 0.980004 0.320962 3.053 0.00226 \*\*   
## Property\_AreaUrban 0.303606 0.297224 1.021 0.30703   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 532.79 on 428 degrees of freedom  
## Residual deviance: 408.63 on 410 degrees of freedom  
## AIC: 446.63  
##   
## Number of Fisher Scoring iterations: 13

# Logistic regression provides probabilities, can also predict probabilities  
log\_predictions <- predict(log\_model, test\_data, type = "response")  
  
# Add the predicted probabilities to the test data  
test\_data$predicted\_probability <- log\_predictions

# Load the new data (make sure it's in the same format as the training data)  
new\_data <- read.csv("D:\\KJRCSS\\semester 03\\R\\ComponentII\_Datasets\\LOAN\_SET\_PRED.csv")  
  
# Preprocess the new data (similar to how you preprocessed the training data)  
new\_data <- new\_data %>% select(-Loan\_ID) # Remove any non-predictive columns  
  
# Fill missing values or handle them similarly to how you did with training data  
new\_data <- new\_data %>%  
 mutate\_if(is.numeric, ~ ifelse(is.na(.), median(., na.rm = TRUE), .)) %>%  
 mutate\_if(is.factor, ~ ifelse(is.na(.), as.character(stats::Mode(.)), .))  
  
# Convert all character variables to factors  
new\_data <- new\_data %>%  
 mutate\_if(is.character, as.factor)  
  
# Scale numeric features in the new data  
numeric\_features\_new <- sapply(new\_data, is.numeric)  
new\_data[numeric\_features\_new] <- scale(new\_data[numeric\_features\_new])  
  
  
  
factor\_levels <- lapply(train\_data %>% select\_if(is.factor), levels)  
  
# Apply these levels to the new data  
for (var in names(factor\_levels)) {  
 if (var %in% names(new\_data)) {  
 new\_data[[var]] <- factor(new\_data[[var]], levels = factor\_levels[[var]])  
 }  
}  
  
# Scale numeric features in the new data  
numeric\_features\_new <- sapply(new\_data, is.numeric)  
new\_data[numeric\_features\_new] <- scale(new\_data[numeric\_features\_new])  
  
  
  
  
  
  
  
  
  
  
  
  
  
# Predict on the new data using the logistic regression model  
new\_predictions <- predict(log\_model, new\_data, type = "response")  
  
# Convert probabilities to class labels using a threshold of 0.5  
new\_data$predicted\_class <- ifelse(new\_predictions > 0.5, 1, 0)  
  
# Print or save the predictions  
print(new\_data$predicted\_class)

## [1] NA NA NA NA

**Conclusion**

In this analysis, we successfully built a logistic regression model to predict loan defaults. By understanding the features that contribute to loan default, financial institutions can make informed lending decisions. Future work could involve exploring other modeling techniques or incorporating additional features to improve prediction accuracy.