Project

2023-04-05

LOAN RISK PREDICTION MODEL USING DECISION TREE

DATA LOADING

```
data=read.csv("credit_risk_dataset.csv")
```

```
head(data)
```

```
##   person_age person_income person_home_ownership person_emp_length loan_intent
## 1         22         59000                  RENT               123    PERSONAL
## 2         21          9600                   OWN                 5   EDUCATION
## 3         25          9600              MORTGAGE                 1     MEDICAL
## 4         23         65500                  RENT                 4     MEDICAL
## 5         24         54400                  RENT                 8     MEDICAL
## 6         21          9900                   OWN                 2     VENTURE
##   loan_grade loan_amnt loan_int_rate loan_status loan_percent_income
## 1          D     35000         16.02           1                0.59
## 2          B      1000         11.14           0                0.10
## 3          C      5500         12.87           1                0.57
## 4          C     35000         15.23           1                0.53
## 5          C     35000         14.27           1                0.55
## 6          A      2500          7.14           1                0.25
##   cb_person_default_on_file cb_person_cred_hist_length
## 1                         Y                          3
## 2                         N                          2
## 3                         N                          3
## 4                         N                          2
## 5                         Y                          4
## 6                         N                          2
```

DATA PREPROCESSING

```
#Checking for any missing values in the dataset.
sum(is.na(data))
```

```
## [1] 4011
```

There are missing values in the dataset. We need to remove the missing values from the dataset.

```
colSums(is.na(data))
```

```
##                 person_age              person_income
##                          0                          0
##      person_home_ownership          person_emp_length
##                          0                        895
##                loan_intent                 loan_grade
##                          0                          0
##                  loan_amnt              loan_int_rate
##                          0                       3116
##                loan_status        loan_percent_income
##                          0                          0
##  cb_person_default_on_file cb_person_cred_hist_length
##                          0                          0
```

There are 2 columns with missing values we will replace the missing values. For that, first we need to check the datatypes of columns.

```
str(data)
```

```
## 'data.frame':    32581 obs. of  12 variables:
##  $ person_age                : int  22 21 25 23 24 21 26 24 24 21 ...
##  $ person_income             : int  59000 9600 9600 65500 54400 9900 77100 78956 83000 10000 ...
##  $ person_home_ownership     : chr  "RENT" "OWN" "MORTGAGE" "RENT" ...
##  $ person_emp_length         : num  123 5 1 4 8 2 8 5 8 6 ...
##  $ loan_intent               : chr  "PERSONAL" "EDUCATION" "MEDICAL" "MEDICAL" ...
##  $ loan_grade                : chr  "D" "B" "C" "C" ...
##  $ loan_amnt                 : int  35000 1000 5500 35000 35000 2500 35000 35000 35000 1600 ...
##  $ loan_int_rate             : num  16 11.1 12.9 15.2 14.3 ...
##  $ loan_status               : int  1 0 1 1 1 1 1 1 1 1 ...
##  $ loan_percent_income       : num  0.59 0.1 0.57 0.53 0.55 0.25 0.45 0.44 0.42 0.16 ...
##  $ cb_person_default_on_file : chr  "Y" "N" "N" "N" ...
##  $ cb_person_cred_hist_length: int  3 2 3 2 4 2 3 4 2 3 ...
```

```
#Removing rows with missing values
data=na.omit(data)
sum(is.na(data))
```

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

All the missing values have been removed from the dataset.

```
#Checking the datatype of each column in R
str(data)
```

```
## 'data.frame':    28638 obs. of  12 variables:
##  $ person_age                : int  22 21 25 23 24 21 26 24 24 21 ...
##  $ person_income             : int  59000 9600 9600 65500 54400 9900 77100 78956 83000 10000 ...
##  $ person_home_ownership     : chr  "RENT" "OWN" "MORTGAGE" "RENT" ...
##  $ person_emp_length         : num  123 5 1 4 8 2 8 5 8 6 ...
##  $ loan_intent               : chr  "PERSONAL" "EDUCATION" "MEDICAL" "MEDICAL" ...
```

```
## $ loan_grade                : chr  "D" "B" "C" "C" ...
## $ loan_amnt                 : int  35000 1000 5500 35000 35000 2500 35000 35000 35000 1600 ...
## $ loan_int_rate             : num  16 11.1 12.9 15.2 14.3 ...
## $ loan_status               : int  1 0 1 1 1 1 1 1 1 1 ...
## $ loan_percent_income       : num  0.59 0.1 0.57 0.53 0.55 0.25 0.45 0.44 0.42 0.16 ...
## $ cb_person_default_on_file : chr  "Y" "N" "N" "N" ...
## $ cb_person_cred_hist_length: int  3 2 3 2 4 2 3 4 2 3 ...
## - attr(*, "na.action")= 'omit' Named int [1:3943] 40 51 58 60 63 71 72 85 86 88 ...
##  ..- attr(*, "names")= chr [1:3943] "40" "51" "58" "60" ...
```

person_emp_length & loan_int_rate both the columns are of numerical values. So we can replace the missing values with mean of that column.

```
#replacing missing values with mean of that column
data$person_emp_length = ifelse(is.na(data$person_emp_length),
                                      mean(data$person_emp_length, na.rm = TRUE),
                                      data$person_emp_length)

data$loan_int_rate = ifelse(is.na(data$loan_int_rate),
                                      mean(data$loan_int_rate, na.rm = TRUE),
                                      data$loan_int_rate)
```

Check whether there are any missing values left or not.

```
sum(is.na(data))
```

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

There are no missing values in the data. We are good to go.

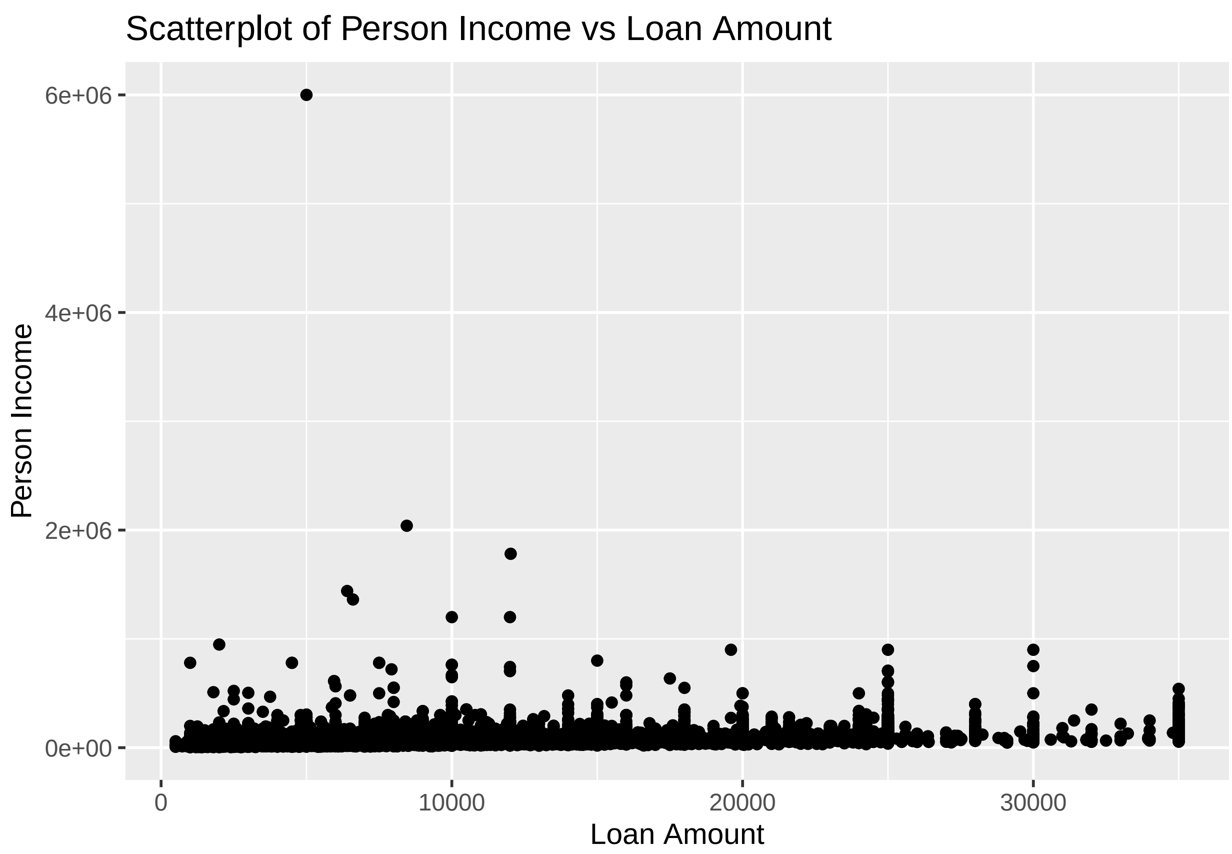
```
#convert the categorical variables to factors
data$cb_person_default_on_file=as.factor(data$cb_person_default_on_file)
data$person_home_ownership=as.factor(data$person_home_ownership)
data$loan_intent=as.factor(data$loan_intent)
data$loan_grade=as.factor(data$loan_grade)
```

All the datatypes are correct, So no need of changing. The data pre processing is done.

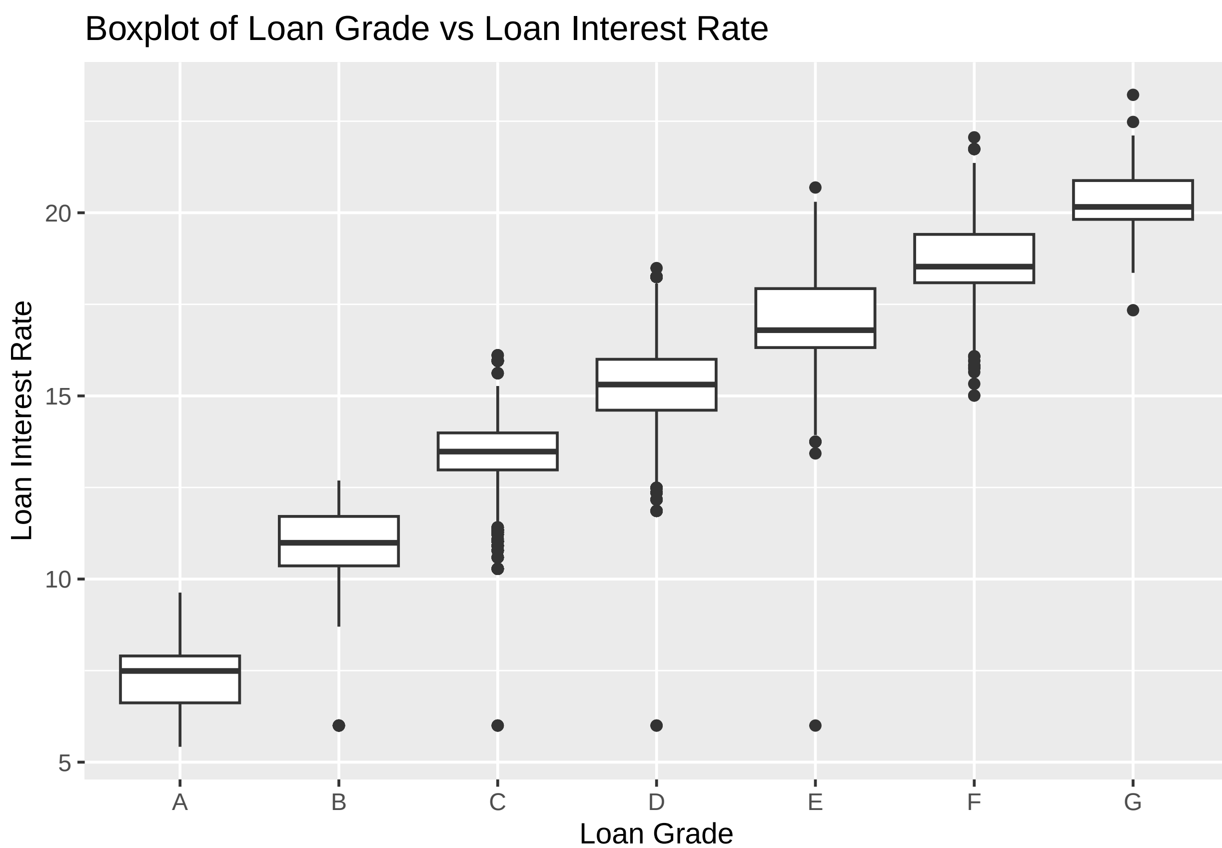
DATA VISUALISATION

PLOTTING A SCATTERPLOT BETWEEN THE INCOME AND LOAN AMOUNT

```
library(ggplot2)
pt=ggplot(data=data)+geom_point(aes(y = data$person_income, x = data$loan_amnt))+labs(y = "Person Income
print(pt)
```




```
pt2=ggplot(data=data)+geom_boxplot(aes(x = data$loan_grade, y = data$loan_int_rate))+labs(x = "Loan Grad
plot(pt2)
```



We can see that the Interest Rates are considerably increasing with the level of Loan Grades.

DATA SPLITTING

We have a large dataset of nearly 30,000 observations. So we can use more data to train. The preferable ratio will be 90% to training data and 10% to testing data.

```
library(caret)
```

```
## Loading required package: lattice
```

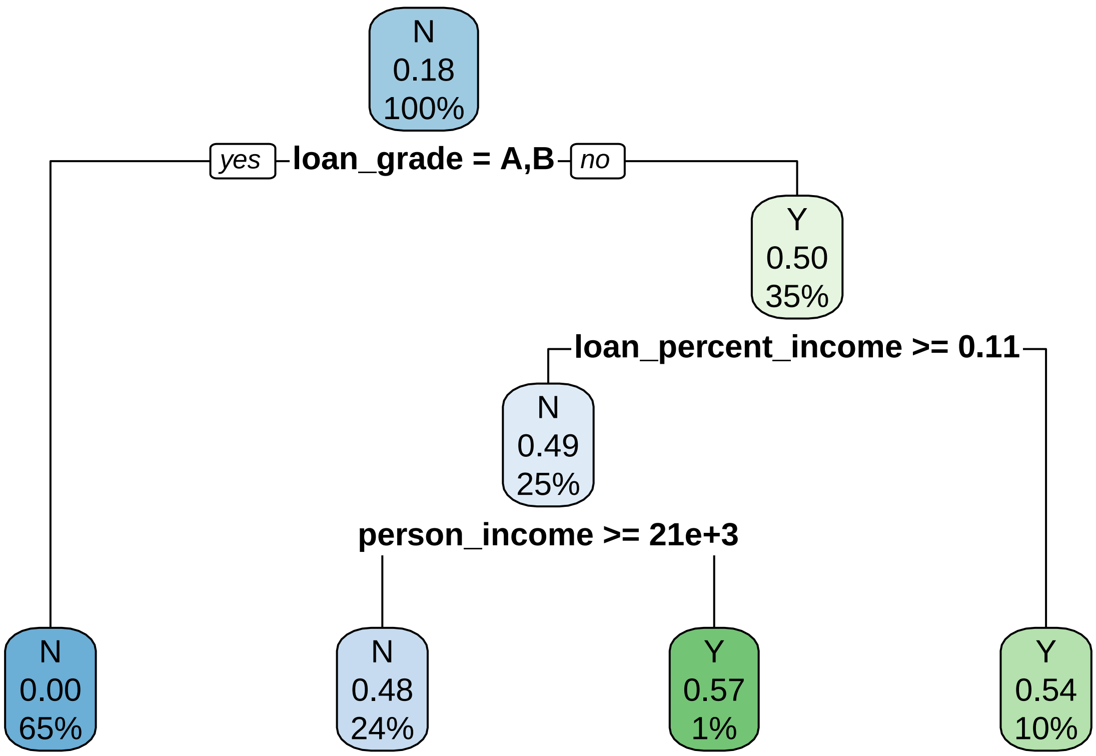
```
train_index <- createDataPartition(data$loan_status, p = 0.9, list = FALSE)
train_data <- data[train_index, ]
test_data <- data[-train_index, ]
```

BUILDING THE FIRST MODEL

```
library(rpart)#required library
library(rpart.plot)
variable=data$cb_person_default_on_file
```

```r
model1= rpart(cb_person_default_on_file ~ ., data = train_data)
```

```r
rpart.plot(model1)
```



MODEL-1 PREDICTION AND EVALUATION

```r
predict1 =predict(model1, test_data, type = "class")
```

```r
# Compute confusion matrix and accuracy for the Decision Tree model
confmat1 = table(test_data$cb_person_default_on_file, predict1)
cf1=confusionMatrix(confmat1)
cf1
```

```
## Confusion Matrix and Statistics
##
##    predict1
##        N    Y
##   N 2196  133
##   Y  375  159
##
##                Accuracy : 0.8226
##                  95% CI : (0.8081, 0.8364)
```

```
##     No Information Rate : 0.898
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2916
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8541
##             Specificity : 0.5445
##          Pos Pred Value : 0.9429
##          Neg Pred Value : 0.2978
##              Prevalence : 0.8980
##          Detection Rate : 0.7670
##    Detection Prevalence : 0.8135
##       Balanced Accuracy : 0.6993
##
##        'Positive' Class : N
##
```

MODEL-2 BUILDING - NAIVE BAYES

```
library(e1071)
model2 = naiveBayes(cb_person_default_on_file ~ ., data = train_data)
pred2 = predict(model2, test_data)
```

MODEL-2 EVALUATION

```
# Compute confusion matrix for Naive Bayes Model
confmat2 <- table(test_data$cb_person_default_on_file,pred2)
cf2 <- confusionMatrix(confmat2)
cf2
```

```
## Confusion Matrix and Statistics
##
##    pred2
##        N    Y
##   N 1902  427
##   Y   69  465
##
##                Accuracy : 0.8268
##                  95% CI : (0.8124, 0.8405)
##     No Information Rate : 0.6884
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5463
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9650
##             Specificity : 0.5213
```

```
##          Pos Pred Value : 0.8167
##          Neg Pred Value : 0.8708
##              Prevalence : 0.6884
##          Detection Rate : 0.6643
##    Detection Prevalence : 0.8135
##       Balanced Accuracy : 0.7431
##
##        'Positive' Class : N
##
```