TABLE OF CONTENT

INTRODUCTION 3
OBJECTIVE 3
DATASET DESCRIPTION 3
TECHNIQUE EMPLOYED FOR OUTLIER TREATMENT 4
DATA CLEANING PROCEDURE 4
ALGORITHMS USED FOR CREATING ML MODELS 10
EVALUATION OF THE MODELS 12

INTRODUCTION

In the ever-evolving landscape of financial services, lending institutions grapple with the daunting challenge of minimizing the risk associated with loan defaults. The ability to accurately predict whether a borrower will default on a loan is not merely advantageous but indispensable for maintaining financial stability and ensuring sustainable growth.

In this context, the integration of machine learning (ML) techniques emerges as a pivotal solution. By leveraging historical data on loan borrowers and a myriad of deterministic factors, ML models offer the promise of predictive analytics that can revolutionize risk management strategies. This report embarks on a journey to harness the power of ML to develop a robust predictive model tailored to the unique needs of lending institutions, ultimately aiming to empower them with the insights needed to make proactive decisions and navigate the complex terrain of financial risk with confidence.

OBJECTIVE

The primary objective of this report is

1. Develop a robust machine learning (ML) model capable of predicting loan defaulters for a banking institution.
2. Utilize past data on loan borrowers to classify whether a new borrower is likely to default on a loan.
3. Explore deterministic factors, and other relevant predictors to enhance predictive accuracy.
4. Investigate various ML techniques, outlier handling methods, and missing value treatment approaches to identify the best-performing model.
5. Provide lending institutions with actionable insights to make informed decisions and mitigate the financial risk associated with loan defaults.

DATASET DESCRIPTION

SN.	Field Name	Description	Category
1	ID	Loan Borrower's ID	Discrete
2	Gender	Gender (Male, Female, Joint, Not Aval.)	Categorical
3	approv_in_adv	Approved in Advance (non-pre or pre)	Categorical
4	loan_type	Loan Type (Type 1, Type 2, or Type 3)	Categorical
5	loan_purpose	Purpose of Loan (p1, p2, p3, or p4)	Categorical
6	Credit_Worthiness	Type of Credit Worthiness (I1 or I2)	Categorical
7	open_credit	Open Credit or not (opc or nopc)	Categorical
8	business_or_commercial	Business loan or commercial loan	Categorical
9	loan_amount	Loan Amount	Continuous
10	rate_of_interest	Interest Rate	Continuous
11	property_value	Property Value	Continuous
12	income	Borrower's Income	Continuous
13	credit_type	Credit Type (CIB, CRIF, EQUI, EXP)	Categorical
14	Credit_Score	Credit Score	Continuous

15	age	Borrower's Age	Categorical
16	LTV	Loan-to-Value Ratio	Continuous
17	Region	Borrower's Region	Categorical
18	Security_Type	Security Type (direct)	Categorical
19	Status	Loan (0: Not approved; 1: Approved)	Categorical

TECHNIQUE EMPLOYED FOR OUTLIER TREATMENT

Quantile-based capping involves identifying extreme values in a numerical variable and replacing them with more reasonable estimates based on quantiles.

1. **Calculate Quartiles and Caps**
    - Quartiles (Q1 and Q3) are calculated to determine the lower and upper bounds of the interquartile range (IQR). These quartiles represent the 25th and 75th percentiles of the data distribution.
    - Caps are then defined as values corresponding to a certain quantile (e.g., 5th and 95th percentiles) to establish the range within which most data points lie.
2. **Determining Tolerance for Outliers**
    - The tolerance for outliers is calculated based on the IQR. A common approach is to use a multiplier (e.g., 1.5 times the IQR) to determine the range within which data points are considered non-outliers.
3. **Cap Outliers**
    - Values beyond the established range (outside of Q1 - 1.5 * IQR and Q3 + 1.5 * IQR) are considered outliers.
    - Outliers are then replaced with the corresponding cap values, ensuring that extreme values are brought within a reasonable range without significantly altering the overall distribution of the data.

DATA CLEANING PROCEDURE

Data Conversion to Factors

- **What** - We converted variables to factors to represent categorical data appropriately in R.
- **Why** - Factors enable R to handle categorical variables efficiently during statistical analyses and modelling.

```
L[,c(2:8, 13,15,17,18,19)] <- lapply(L[,c(2:8, 13,15,17,18,19)], factor)
L <- as.data.frame(L)                # converting to data frame
```

Check for Missing Values

- **What** - We checked for missing values in the dataset using the **map** function to count the number of missing values in each column.
- **Why** - Identifying missing values is crucial as they can affect the accuracy and reliability of analyses and modelling.

```
map(L, ~sum(is.na(.)))
```

Imputing Missing Values

- **What** - We imputed missing values in the dataset using mode for categorical variables and median for numerical variables.
- **Why** - Missing values can introduce bias and hinder the effectiveness of statistical analyses and modelling.

```
L[3] <- impute(L[3], fun = Mode)
L[10] <- impute(L[10], fun = median)
L[11] <- impute(L[11], fun = median)
L[12] <- impute(L[12], fun = Median)
L[15] <- impute(L[15], fun = Mode)
L[16] <- impute(L[16], fun = median)
```

Checking Normality and Outliers

- **What** - We calculated skewness and kurtosis statistics and visualized boxplots to assess normality and identify outliers in selected numerical variables.
- **Why** - Normality and outlier detection are important for ensuring the validity and reliability of statistical analyses and modelling.

```
skewness(L[c(9,10,11,12,14, 16)])
kurtosis(L[c(9,10,11,12,14, 16)])
boxplot(L[c(9,10,11,12,14, 16)])
```

Outlier Treatment

- **What** - We removed outliers from the **income** variable and applied quantile-based capping to other selected numerical variables.
- **Why** - Outliers can distort statistical analyses and modelling outcomes, so it's important to handle them appropriately.

```
#removing outliers in income
outinc <- boxplot(L$income)$out
L<- L[-which(L$income %in% outinc), ]

#treating outliers in other columns
colnames(L[c(9,10,11,14, 16)])
columns_to_process <-
c("loan_amount","rate_of_interest","property_value","Credit_Score","LTV")
 # Get the data for the current column
 x <- L[[col]]

 # Loop through each column in the list and apply the quantile-based capping
for (col in columns_to_process) {

 # Calculating quartiles and caps
 qnt <- quantile(x, probs = c(0.25, 0.75), na.rm = TRUE)
 caps <- quantile(x, probs = c(0.05, 0.95), na.rm = TRUE)

 # Calculate the tolerance for outliers based on IQR
 H <- 1.5 * IQR(x, na.rm = TRUE)
```

```
  # Cap outliers
  x[x < (qnt[1] - H)] <- caps[1]
  x[x > (qnt[2] + H)] <- caps[2]

  L[[col]] <- x              # Assigning the modified data back to the data frame
}
```

EXPLORATORY DATA ANALYSIS

```
summary(L)          #for var info
```

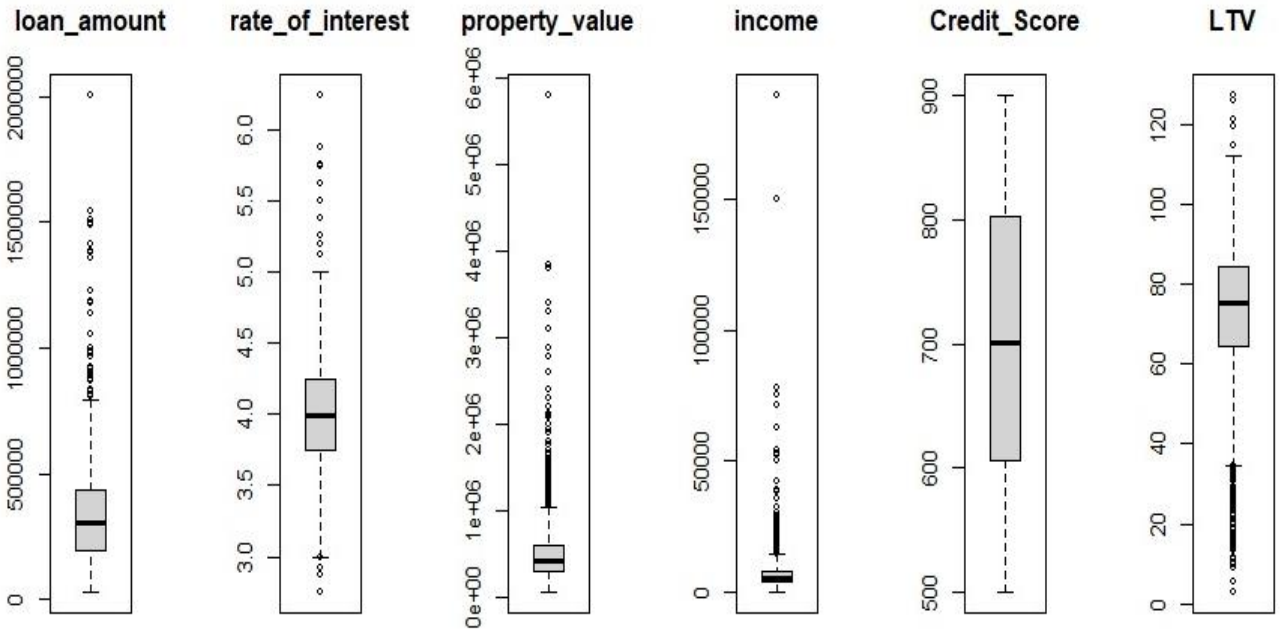
```
  loan_amount     rate_of_interest property_value        income
Min.   : 26500   Min.   :3.000    Min.   :  58000   Min.   :     0
1st Qu.:196500   1st Qu.:3.750    1st Qu.: 298000   1st Qu.:  3780
Median :306500   Median :3.990    Median : 428000   Median :  5640
Mean   :327828   Mean   :4.029    Mean   : 467540   Mean   :  6797
3rd Qu.:436500   3rd Qu.:4.250    3rd Qu.: 598000   3rd Qu.:  7995
Max.   :796500   Max.   :5.000    Max.   :1048000   Max.   :189360

 credit_type  Credit_Score      age           LTV
 CIB :968    Min.   :500.0   <25  : 23   Min.   : 34.58
 CRIF:818    1st Qu.:606.8   >74  :145   1st Qu.: 64.41
 EQUI:305    Median :701.0   25-34:384   Median : 75.15
 EXP :801    Mean   :702.4   35-44:639   Mean   : 73.31
             3rd Qu.:802.0   45-54:690   3rd Qu.: 84.30
             Max.   :900.0   55-64:600   Max.   :112.11
                             65-74:411
      ID                        Gender    approv_in_adv loan_type
Min.   :24890   Female            :579   nopre:2448    type1:2199
1st Qu.:25613   Joint             :790   pre  : 444    type2: 433
Median :26336   Male              :798                 type3: 260
Mean   :26336   Sex Not Available:725
3rd Qu.:27058
Max.   :27781

loan_purpose Credit_Worthiness open_credit business_or_commercial
p1: 626      l1:2783           nopc:2884   b/c  : 433
p2:  54      l2: 109           opc :   8   nob/c:2459
p3:1147
p4:1065
       Region     Security_Type Status
central   : 196   direct:2892   0:2170
North     :1452                 1: 722
North-East:  24
south     :1220
```

- Gender distribution shows that there are different categories, with 'Male' being the most common, followed by 'Female' and 'Joint'.
- The majority of loan approvals are non-preapproved (nopre), and the most common loan type is type1.
- Loan purposes vary across different categories (p1, p2, p3, p4), with p1 being the most common.
- Credit worthiness is mostly categorized as l1.
- Open credit status is predominantly nopc (non-open credit).
- Credit types include CIB, CRIF, EQUI, and EXP.
- Regional distribution shows that loans are dispersed across different regions.
- Security type is predominantly direct.
- Loan status indicates that a significant portion of loans is not approved (0), while others are approved (1).

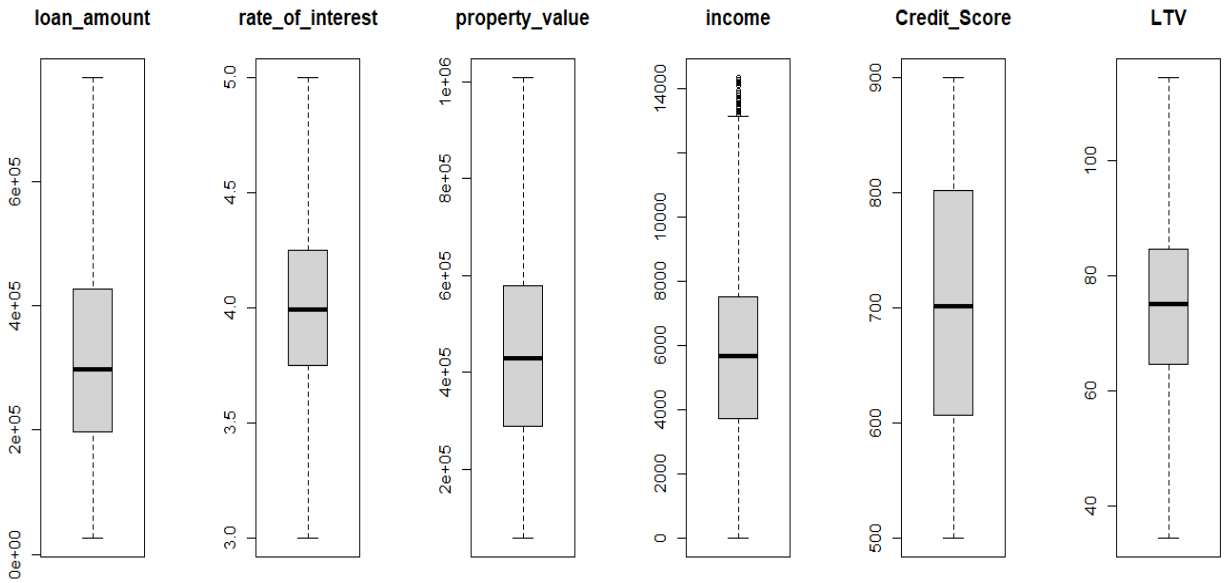
Boxplot

```
#boxplot
x <- L[c(9,10,11,12,14, 16)]
par(mfrow=c(1,6))
for(i in 1:6) {
  boxplot(x[,i], main=names(x)[i])
}
```

Boxplot Before Oulier Treatment



Boxplot after Outlier Treatment

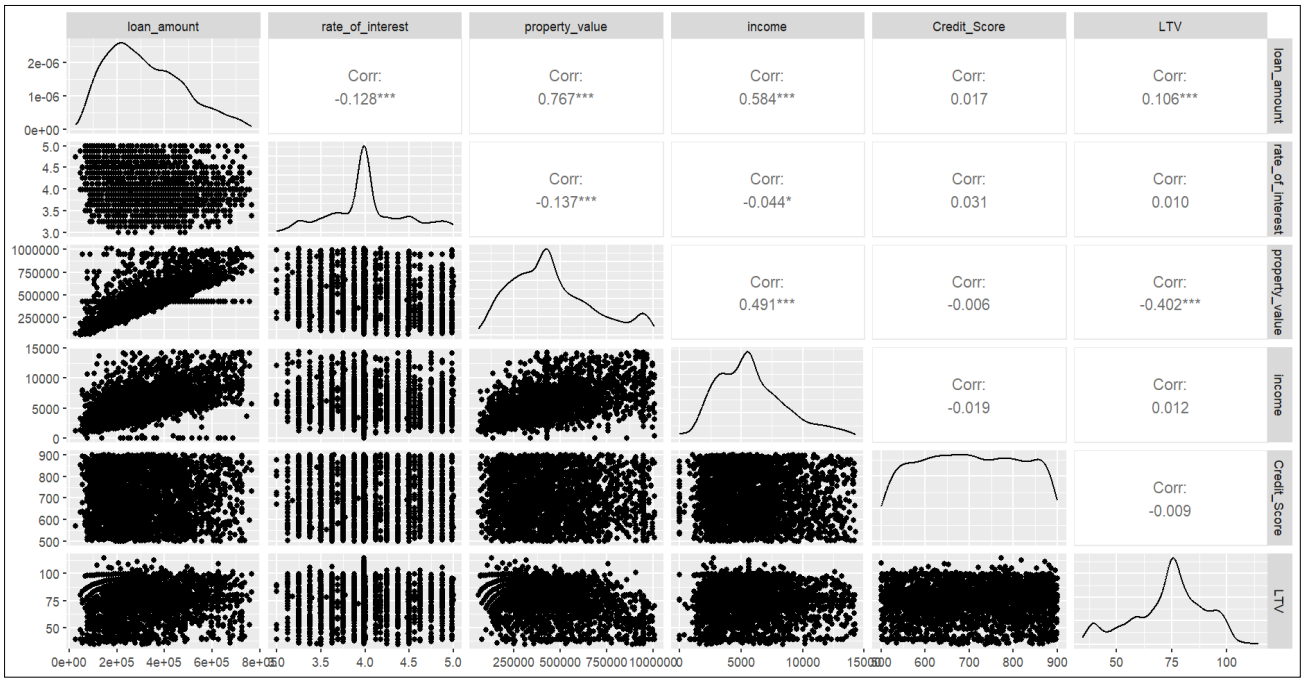


Income has the widest IQR among all metrics, indicating a large spread in incomes. However, the critical point is the presence of 47 outliers, which are data points that fall outside the expected range. This suggests a much more uneven distribution of income compared to the other variables.

Pairplot

The provided pair plot offers valuable insights into the relationships between loan amount, rate of interest, property value, income, and credit score.

```
#pairplot
ggpairs(x)
```



Loan Amount

Loan amount vs. rate of interest - A very weak negative correlation exists (-0.128). There's a slight tendency for interest rates to decrease as loan amounts increase, but it's a negligible effect.

Loan amount vs. property value - A strong positive correlation is evident (0.767). This confirms the positive association observed in the scatter plot. Higher loan amounts are associated with significantly higher property values.

Loan amount vs. income - This reveals a surprisingly moderate negative correlation (-0.584). While the scatter plot suggested a weak positive trend, the correlation value indicates a stronger connection, but in the opposite direction. Borrowers with higher incomes might tend to take out smaller loans, possibly due to larger down payments.

Loan amount vs. credit score - A near-zero correlation exists (0.017). This confirms the lack of any significant relationship between loan amount and credit score as observed in the pairplot.

Loan amount vs. LTV - Another weak negative correlation exists (-0.106). There's a slight tendency for borrowers with higher loan amounts to have lower LTV ratios (meaning a larger down payment), but the effect is minimal.

Property Value

Property value vs. income - A moderate positive correlation is present (0.491). This aligns with the scatter plot, indicating a connection between higher incomes and owning more expensive properties, but the strength is not as strong as the loan amount-property value relationship.

Property value vs. credit score - A negligible correlation exists (-0.006). This confirms the lack of any meaningful association between property value and credit score suggested by the pairplot.

Property value vs. LTV - A moderate negative correlation is observed (-0.402). This indicates a connection between higher property values and lower LTV ratios (meaning a larger down payment). This makes sense as lenders(Bank) might require a larger down payment for more expensive properties.

Rate of Interest

Rate of interest vs. Property Value - A very weak negative correlation (-0.137) exists. There's a slight tendency for properties with higher values to have slightly lower interest rates, but the effect is minimal.

Rate of interest vs. Income - There's near-zero correlation (-0.044) between rate of interest and income confirming the absence of any meaningful relationship.

Rate of interest vs. Credit Score - A weak correlation (0.031), but positive in this case. There might be a slight tendency for borrowers with higher credit scores to receive slightly lower interest rates.

Rate of interest vs. LTV - A negligible correlation exists (0.010). There's no meaningful association between interest rate and LTV.

Income

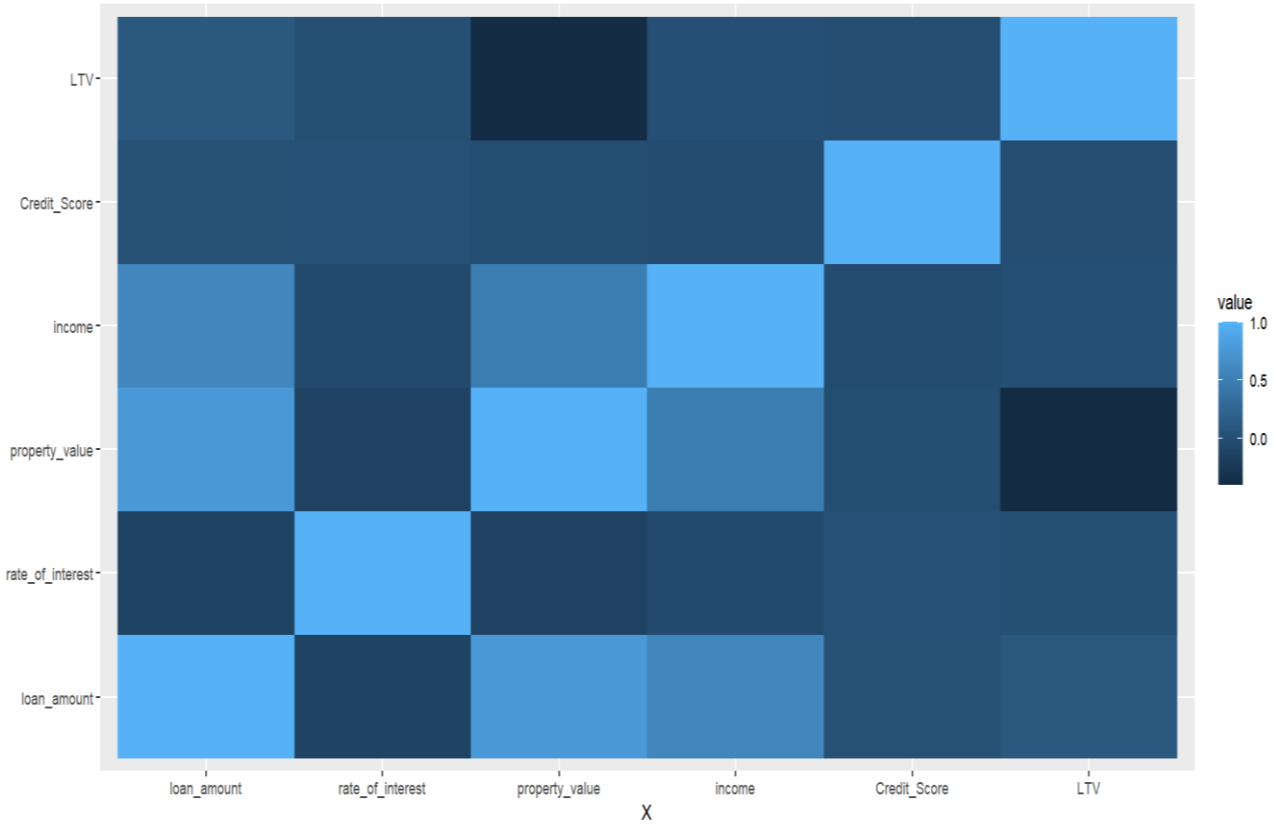
Income vs. credit score - A near-zero correlation exists (-0.019). This confirms the absence of any significant relationship between income and credit score as seen in the pairplot.

Income vs. LTV - A very weak positive correlation exists (0.012). This suggests almost no connection between income and LTV.

Credit Score

Credit Score vs. LTV - Another negligible correlation exists (-0.009). There's no significant relationship between credit score and LTV.

Correlation Heatmap

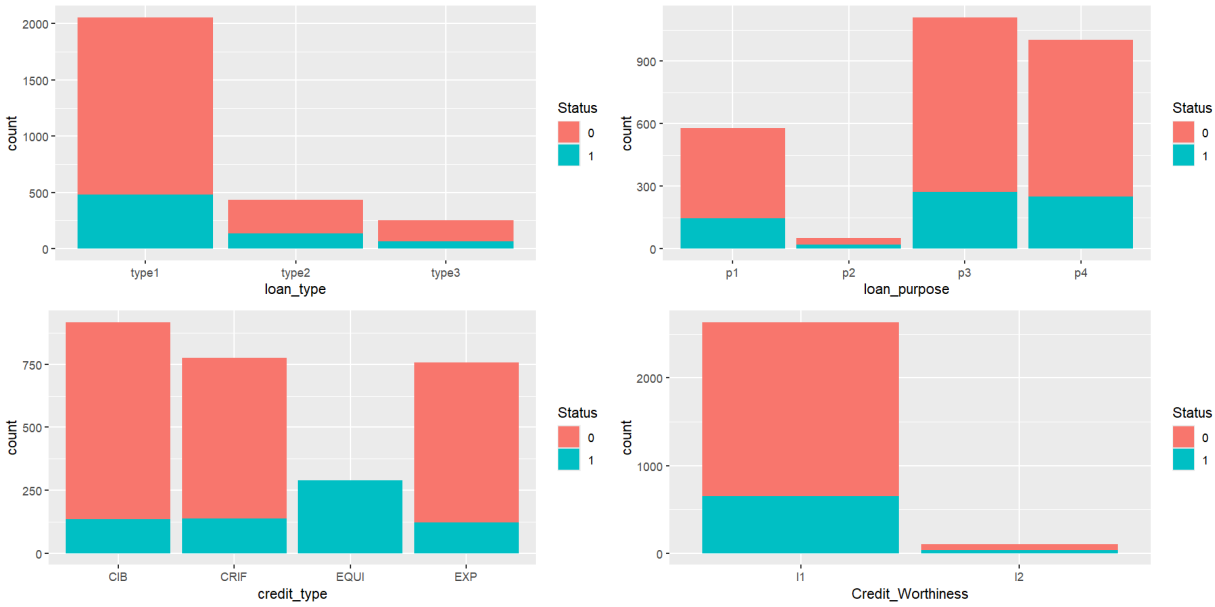


```
#heatmap
x <- L[c(9,10,11,12,14, 16)]
data <- cor(x)
data1 <- melt(data)
ggplot(data1, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  labs(title = "Correlation Heatmap",
       x = "X",
       y = "Y")
```

The darker colour in a correlation heatmap represents a stronger correlation, either positive or negative. the heatmap reveals some expected connections, like the positive correlation between property value and income, and the negative correlation between loan amount and income (due to potentially larger down payments from borrowers with higher incomes). It also highlights some weaker relationships, such as the minimal effect of interest rate on property value or credit score.

Barplot

```
#Barplot
p1 <- ggplot(data = L) +geom_bar(mapping = aes(x = loan_type,fill=Status))
p2 <-ggplot(data = L) +geom_bar(mapping = aes(x = loan_purpose,fill=Status))
p3 <-ggplot(data = L) +geom_bar(mapping = aes(x = credit_type,fill=Status))
p4 <-ggplot(data = L) +geom_bar(mapping = aes(x = Credit_Worthiness,fill=Status))
grid.arrange(p1,p2,p3,p4 ,ncol= 2)
```



Loan Types - Type1 loans are the most common, followed by Type3 and Type2.

Loan Purpose - The majority of loans are for purpose P3, followed by P4 and P1, with P2 being the least frequent.

Credit Worthiness - Borrowers with credit worthiness level L1 constitute the majority, while L2 borrowers are significantly fewer in number.

Credit Types - The most prevalent credit type is CIB, followed by EXP, CRIF, and EQUI.

ALGORITHMS USED FOR CREATING ML MODELS

Note – Security Variable has been deleted from the dataset as it had only one category that is direct.

Model Training

Herein, we segment trains multiple machine learning (ML) models using the dataset for loan status prediction.

```
set.seed(100)

intrain <- createDataPartition(y = L$Status, p = 0.8, list = FALSE)

#subset P to obtain training set
training <- L[intrain, ]
#subset the rest to obtain test dataset
testing <- L[-intrain, ]
```

Naive Bayes

The algorithm calculates the probability of loan status (default or non-default) given predictor variables, assuming independence among predictors like loan amount, credit score, and loan purpose.

```
#Model 1 – Naïve Bayes
Model1 <- train(Status~.,data = training,  method = "naive_bayes")
summary(Model1)

predStatus = predict(Model1,newdata = testing)

confusionMatrix(predStatus,testing$Status, positive = "1")
```

Logistic Regression

Modelling the probability of loan default using a logistic function, considering factors such as loan amount, credit score, and income to estimate the likelihood of default.

```
#Model 2 - Logistic regression

Model2 <- train(data = training, Status~.,  method = "glm", family="binomial")
summary(Model2)

predStatus1= predict(Model2,newdata = testing)
confusionMatrix(predStatus1,testing$Status,positive="1")
```

Decision Tree

The models create a tree-like structure based on loan attributes like loan amount, loan type, and credit worthiness, using either the Gini index (Model 3) or information gain (Model 4) to make optimal splits.

```
#Model 3&4 - Decision Tree
#using gini index
Model3 <- train(data = training, Status~.,  method = "rpart")

#using information gain
Model4 <- train(data = training, Status~.,  method = "rpart", parms = list(split = "information"))

par(mfrow=c(1,2))
rpart.plot(Model3$finalModel)
rpart.plot(Model4$finalModel)
```

```
predStatus3= predict(Model3,newdata = testing)
predStatus4= predict(Model4,newdata = testing)

confusionMatrix(predStatus3,testing$Status,positive="1")
confusionMatrix(predStatus4,testing$Status, positive="1")
```

Random Forest

The model builds an ensemble of decision trees using features such as loan amount, income, and credit score to predict loan default probabilities with improved accuracy and robustness.

```
#Model 5 - Random Forest
Model5 <- train(data = training, Status~.,  method = "rf")
Model5

Model5$finalModel
varImp(Model5)

predStatus5= predict(Model5,newdata = testing)
confusionMatrix(predStatus5,testing$Status, positive="1")
```

EVALUATION OF THE MODELS

Naïve Bayes

Accuracy

The model's accuracy of 85.92% signifies the proportion of correctly classified instances out of the total instances.

Confusion Matrix

True Negative (TN) - 407

False Positive (FP) - 3

False Negative (FN) - 74

True Positive (TP) - 63

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 407  74
         1   3  63

               Accuracy : 0.8592
                 95% CI : (0.8272, 0.8873)
    No Information Rate : 0.7495
    P-Value [Acc > NIR] : 2.325e-10

                  Kappa : 0.5469

 Mcnemar's Test P-Value : 1.496e-15

            Sensitivity : 0.4599
            Specificity : 0.9927
         Pos Pred Value : 0.9545
         Neg Pred Value : 0.8462
             Prevalence : 0.2505
         Detection Rate : 0.1152
   Detection Prevalence : 0.1207
      Balanced Accuracy : 0.7263

       'Positive' Class : 1
```

Sensitivity (True Positive Rate)

Sensitivity measures the model's ability to correctly identify positive instances, in this case, loan defaults. A sensitivity of 45.99% indicates that the model captures only about 46% of actual default cases suggesting that there is a notable proportion of loan defaults that the model fails to detect.

Specificity (True Negative Rate)

Specificity gauges the model's ability to correctly identify negative instances, i.e., non-default cases. The high specificity of 99.27% implies that the model excels in identifying instances where loans are not defaulted, minimizing the false alarm rate.

Positive Predictive Value (Precision)

The positive predictive value (PPV) reflects the proportion of predicted positive cases (loan defaults) that are correctly identified. With a PPV of 95.45%, the model demonstrates a high level of precision in flagging loan defaults among the instances it predicts as positive.

Negative Predictive Value

The negative predictive value measures the proportion of predicted negative cases (non-defaults) that are correctly identified. The model's negative predictive value of 84.62% indicates its effectiveness in recognizing instances where loans are not defaulted among the predicted negative cases.

Insights

The model exhibits high specificity, indicating a low false positive rate, its sensitivity is comparatively lower, indicating a higher false negative rate. This imbalance implies that the model may overlook some actual loan default cases, which could have significant implications for risk assessment in lending scenarios.

Logistic Regression

Accuracy

The model achieves an accuracy of 86.29%, indicating the proportion of correctly classified instances out of the total instances suggesting a reasonably high level of overall accuracy in predicting loan defaults.

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 410  75
         1   0  62

               Accuracy : 0.8629
                 95% CI : (0.8312, 0.8906)
    No Information Rate : 0.7495
    P-Value [Acc > NIR] : 5.334e-11

                  Kappa : 0.5534

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.4526
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.8454
             Prevalence : 0.2505
         Detection Rate : 0.1133
   Detection Prevalence : 0.1133
      Balanced Accuracy : 0.7263

       'Positive' Class : 1
```

Confusion Matrix

True Negative (TN) - 410
False Positive (FP) - 0
False Negative (FN) - 75
True Positive (TP) - 62

Sensitivity (True Positive Rate)

A sensitivity of 45.26%, indicates that the model captures only about 45.2% of actual default cases suggesting that there is a notable proportion of loan defaults that the model fails to detect.

Specificity (True Negative Rate)

The specificity of 100% implies that the model excels in identifying instances where loans are not defaulted, minimizing the false alarm rate.

Positive Predictive Value (Precision)

The positive predictive value (PPV) is 100%, indicating that all predicted loan defaults are correctly identified as such.

Negative Predictive Value

A negative predictive value of 84.54%, the model demonstrates its effectiveness in recognizing instances where loans are not defaulted among the predicted negative cases.

Insights

The Logistic Regression model exhibits high specificity and accuracy, indicating its proficiency in correctly classifying non-default instances. However, similar to the Naive Bayes model, its sensitivity is comparatively lower, suggesting a potential oversight of actual loan default cases.

Decision Tree – Gini

Accuracy The Decision Tree model based on the Gini index achieves an accuracy of 89.58%, indicating its ability to correctly classify instances.

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 353   0
         1  57 137

               Accuracy : 0.8958
                 95% CI : (0.8671, 0.9201)
    No Information Rate : 0.7495
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7562

 Mcnemar's Test P-Value : 1.195e-13

            Sensitivity : 1.0000
            Specificity : 0.8610
         Pos Pred Value : 0.7062
         Neg Pred Value : 1.0000
             Prevalence : 0.2505
         Detection Rate : 0.2505
   Detection Prevalence : 0.3547
      Balanced Accuracy : 0.9305

       'Positive' Class : 1
```

Confusion Matrix

True Negative (TN) - 353
False Positive (FP) - 57
False Negative (FN) - 0
True Positive (TP) - 137

Sensitivity (True Positive Rate)

With a sensitivity of 100%, the model excels in accurately identifying positive instances, particularly loan defaults.

Specificity (True Negative Rate)

The specificity of 86.10% indicates the model's effectiveness in identifying negative instances, i.e., non-default cases resulting in a low false alarm rate.

Positive Predictive Value (Precision)

The positive predictive value (PPV) of 70.62% reflects the precision of the model in identifying loan defaults among the positive predictions. Approximately 71% of instances flagged as loan defaults are correctly identified.

Negative Predictive Value

With a negative predictive value of 100%, the model demonstrates its effectiveness in recognizing instances where loans are not defaulted among the predicted negative cases.

Insights

The Decision Tree model, utilizing the Gini index, demonstrates strong performance in accurately identifying both loan defaults and non-defaults. Its exceptional sensitivity means it captures all actual loan default cases, providing a thorough evaluation of lending risks.

Decision Tree – Information

Accuracy

The Information Decision Tree model achieves an accuracy of 89.58%, indicating its proficiency in correctly classifying instances.

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 353   0
         1  57 137

               Accuracy : 0.8958
                 95% CI : (0.8671, 0.9201)
    No Information Rate : 0.7495
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7562

 Mcnemar's Test P-Value : 1.195e-13

            Sensitivity : 1.0000
            Specificity : 0.8610
         Pos Pred Value : 0.7062
         Neg Pred Value : 1.0000
             Prevalence : 0.2505
         Detection Rate : 0.2505
   Detection Prevalence : 0.3547
      Balanced Accuracy : 0.9305

       'Positive' Class : 1
```

Confusion Matrix

True Negative (TN) - 353
False Positive (FP) - 57
False Negative (FN) - 0
True Positive (TP) - 137

Sensitivity (True Positive Rate)

With a sensitivity of 100%, the model demonstrates its excellence in accurately identifying positive instances, particularly loan defaults.

Specificity (True Negative Rate)

The specificity of 86.10% indicates the model's effectiveness in identifying negative instances, i.e., non-default cases suggesting a low false alarm rate.

Positive Predictive Value (Precision)

The positive predictive value (PPV) of 70.62% reflects the precision of the model in identifying loan defaults among the positive predictions. Approximately 71% of instances flagged as loan defaults are correctly identified.

Negative Predictive Value

With a negative predictive value of 100%, the model demonstrates its effectiveness in recognizing instances where loans are not defaulted among the predicted negative cases.

Insights

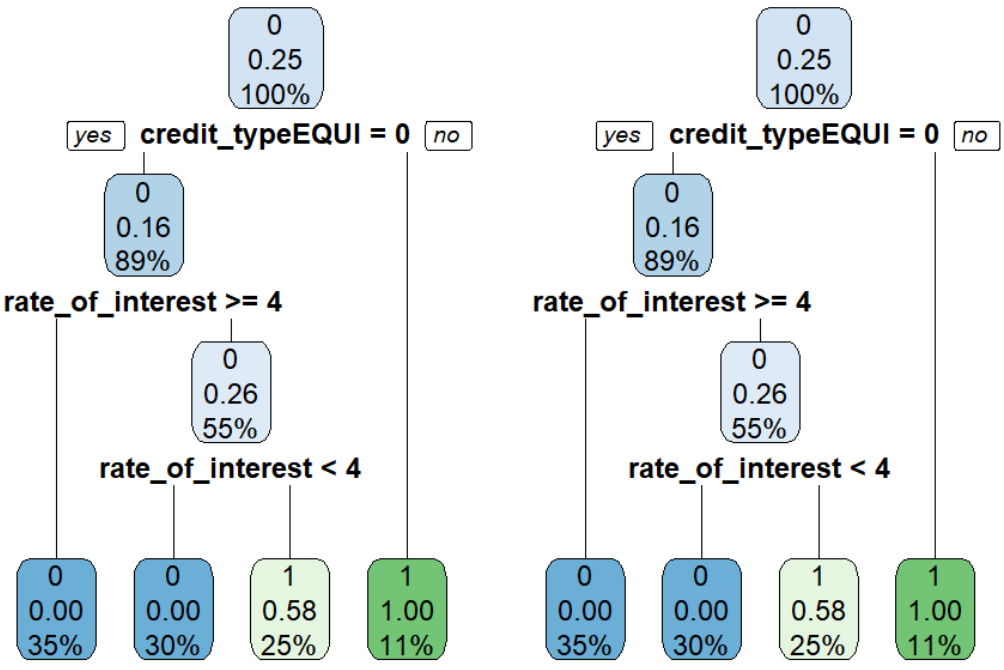
The Information Decision Tree model performs exceptionally well in identifying both loan defaults and non-defaults, as indicated by its high sensitivity and specificity. Its perfect sensitivity means that it captures all instances of actual loan defaults, providing a thorough assessment of lending risk.

Interpretation of the Tree

Both the trees from Gini and Information method showcase the same result.

Both Information Gain and Gini Index criteria potentially lead to very similar decision trees, resulting in the same performance in this case.

They both achieved a perfect sensitivity (recall), correctly classifying all defaulters.



In both decision trees, the "Credit Type" attribute is automatically prioritized first based on its highest information gain and lowest Gini index values. The threshold values are determined automatically by the rpart method.

If "Credit Type" (EQUI) is not equal to 0, the loan is approved. Otherwise, the decision is based on the interest rate: if it's greater than or equal to 4, the loan is labelled as not approved. However, if the interest rate is less than 4, there's a 25% chance of loan approval and a 30% chance of loan disapproval.

Random Forest Model

Accuracy The Random Forest model achieves an accuracy of 92.5%, indicating the proportion of correctly classified instances out of the total instances.

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 385  16
         1  25 121

               Accuracy : 0.925
                 95% CI : (0.8997, 0.9457)
    No Information Rate : 0.7495
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8046

 Mcnemar's Test P-Value : 0.2115

            Sensitivity : 0.8832
            Specificity : 0.9390
         Pos Pred Value : 0.8288
         Neg Pred Value : 0.9601
             Prevalence : 0.2505
         Detection Rate : 0.2212
   Detection Prevalence : 0.2669
      Balanced Accuracy : 0.9111

       'Positive' Class : 1
```

Confusion Matrix

True Negative (TN) - 385
False Positive (FP) - 25
False Negative (FN) - 16
True Positive (TP) - 121

Sensitivity (True Positive Rate)

With a sensitivity of 88.32%, the model demonstrates its proficiency in correctly identifying positive instances, particularly loan defaults.

Specificity (True Negative Rate)

The model achieves a specificity of 93.90%, indicating its ability to accurately identify negative instances, i.e., non-default cases.

Positive Predictive Value (Precision)

The positive predictive value (PPV) stands at 82.88%, implying that approximately 83% of the instances predicted as loan defaults are correctly identified as such.

Negative Predictive Value

With a negative predictive value of 96.01%, the model demonstrates its effectiveness in recognizing instances where loans are not defaulted among the predicted negative cases.

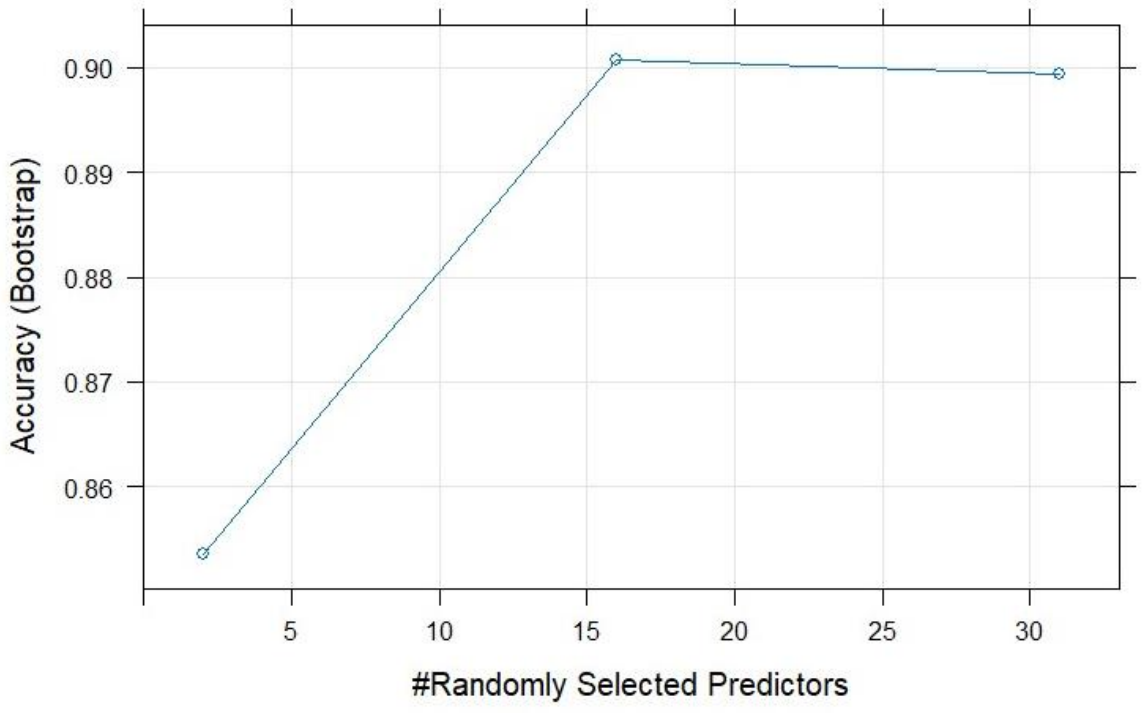
```
rf variable importance

  only 20 most important variables shown (out of 31)

                          Overall
rate_of_interest          100.000
credit_typeEQUI            66.246
LTV                        17.756
property_value             16.700
income                     15.335
ID                         11.377
Credit_Score               11.287
loan_amount                 9.041
age35-44                    1.911
loan_purposep3              1.673
credit_typeCRIF             1.446
GenderJoint                 1.424
GenderMale                  1.419
approv_in_advpre            1.371
GenderSex Not Available     1.343
loan_purposep4              1.323
credit_typeEXP              1.269
age45-54                    1.247
Regionsouth                 1.112
age55-64                    1.109
```

Insights

The Random Forest model demonstrates high sensitivity and specificity, implying effective detection of both loan defaults and non-defaults. This balanced performance suggests robustness in identifying instances across both categories.

Random Forest Plotting

ThE graph illustrates the relationship between the number of randomly selected predictors (mtry) and the resulting accuracy. It's evident that as mtry increases, accuracy generally rises, reaching its peak of 92.5% when mtry equals 16. However, beyond this point, the accuracy begins to decline as mtry increases further.



Final Evalution

Model	Accuracy	Sensitivity (Recall)	Specificity	Balanced Accuracy
Naive Bayes	0.8592	0.4599	0.9927	0.7263
Logistic Regression	0.8629	0.4526	1	0.7263
Gini Decision Tree	0.8958	1	0.861	0.9305
Information Gain Decision Tree	0.8958	1	0.861	0.9305
Random Forest	0.925	0.8759	0.9415	0.9087

Naive Bayes & Logistic Regression, have moderate accuracy, they suffer from a significant drawback - they miss a substantial number of loan defaulters (low sensitivity).

Based on our observations Random Forest model appears to be the best choice for loan default prediction due to its superior accuracy, balanced performance across classes.

- Random Forest achieves the highest accuracy (0.9250) compared to other models, indicating that it correctly classifies a larger portion of loan applications.
- It also demonstrates a good balance between sensitivity (recall) of 0.8759 (identifies most defaulters) and specificity of 0.9415 (correctly classifies most non-defaulters).
- The balanced accuracy (0.9087) combines these metrics, highlighting Random Forest's well-rounded performance across both classes (defaulters and non-defaulters).