Loan Prediction Using Machine Learning Models in R

Objective

To build a predictive model to determine loan approval using applicant data.

Data Description

- **Features:** Gender, Married, Dependents, Education, Self-Employed, ApplicantIncome, CoapplicantIncome, LoanAmount, LoanAmountTerm, CreditHistory, PropertyArea, LoanStatus.
- **Target Variable:** LoanStatus (Y = Approved, N = Not Approved).

Data Cleaning and Preprocessing

- Loaded and inspected data.
- Removed underscores from column names for consistency.
- Feature Engineering:
  - Created TotalIncome = ApplicantIncome + CoapplicantIncome
- Handling Missing Values:
  - Identified missing values.

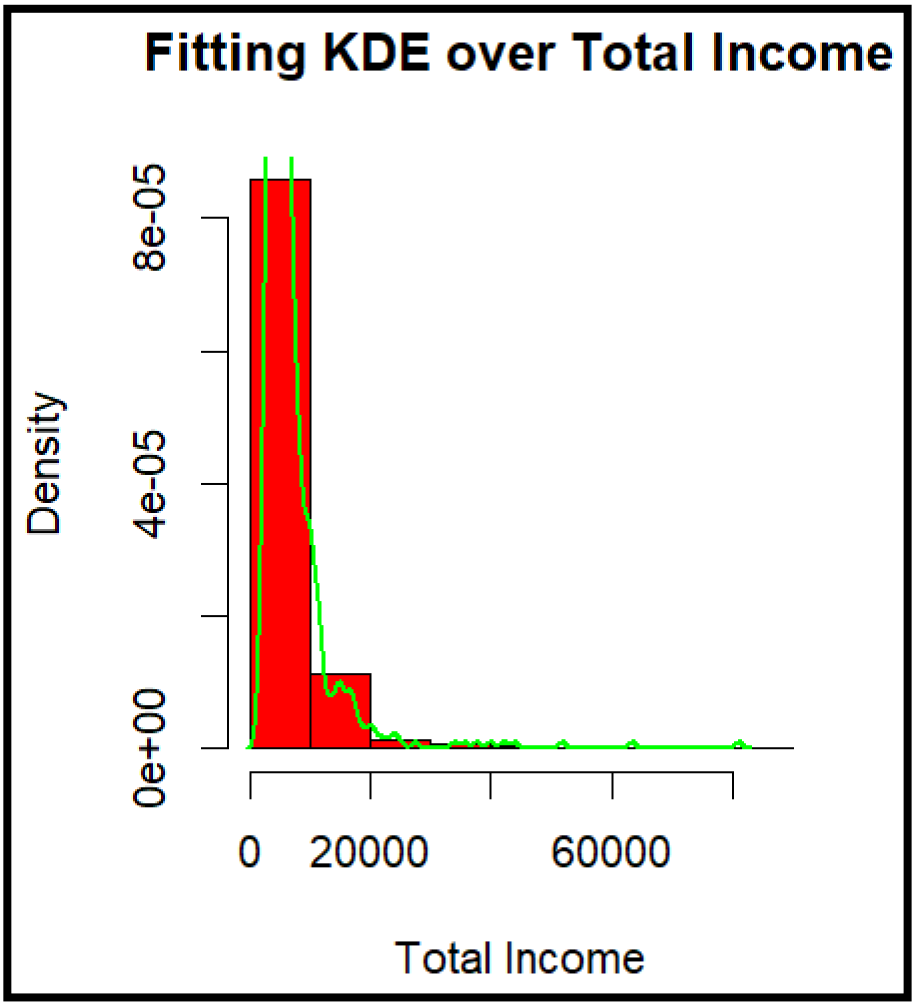
```
> NAvalues[NAvalues> 0]
     Gender      Married   Dependents SelfEmployed
         13            3           15           32
```

```
colSums(is.na(data))
           LoanID            Gender           Married
                0                 0                 0
       Dependents         Education      SelfEmployed
                0                 0                 0
  ApplicantIncome CoapplicantIncome        LoanAmount
                0                 0                22
    LoanAmountTerm     CreditHistory      PropertyArea
               14                50                 0
       LoanStatus       TotalIncome
                0                 0
```

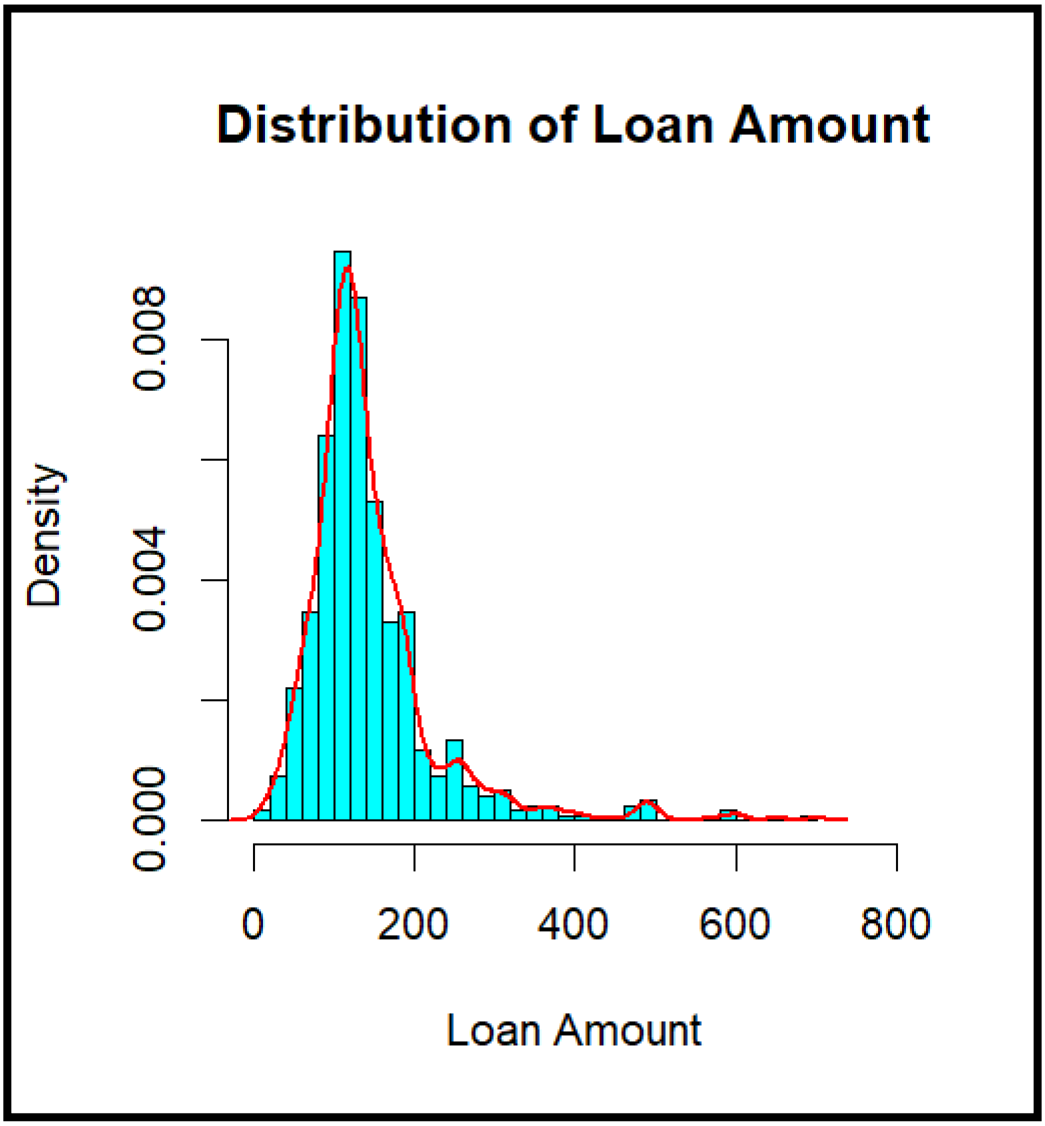
  - Replaced empty strings with NA.
  - Imputation Strategy:
    - Categorical columns: Mode imputation (to avoid bias in tabular, survey-like datasets).
    - LoanAmount: Imputed with mean (post outlier removal using KDE).
    - LoanAmountTerm: Imputed with median (due to skewness).
    - CreditHistory: Imputed with mode.

- Outlier Handling:

```
> #  Making a Histogram
> hist(data$TotalIncome,freq=FALSE,col = "red", main ="Fitting KDE over Total
Income",xlab = "Total Income")
>
> # Add kernel density line
> lines(density(data$TotalIncome),col="green",lwd = 2)
```




```
> loanamount <- na.omit(data$LoanAmount)
> # plot histogram
> hist(loanamount,freq = FALSE,breaks = 30,xlim = c(0, 800),
col = "cyan", main = "Distribution of Loan Amount",xlab = "L
oan Amount")
> # Add density curve
> lines(density(loanamount), col = "red", lwd = 2)
```

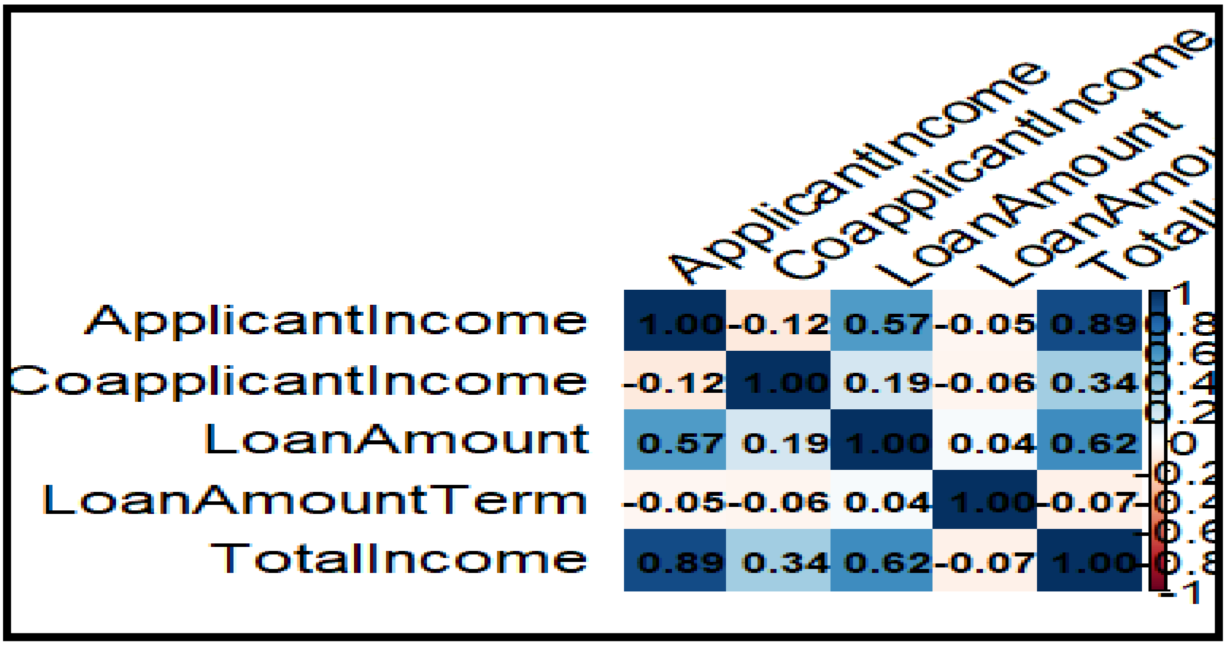


- Plotted KDE and histograms for TotalIncome and LoanAmount.
- Identified right-skewness and trimmed extreme outliers before imputation.

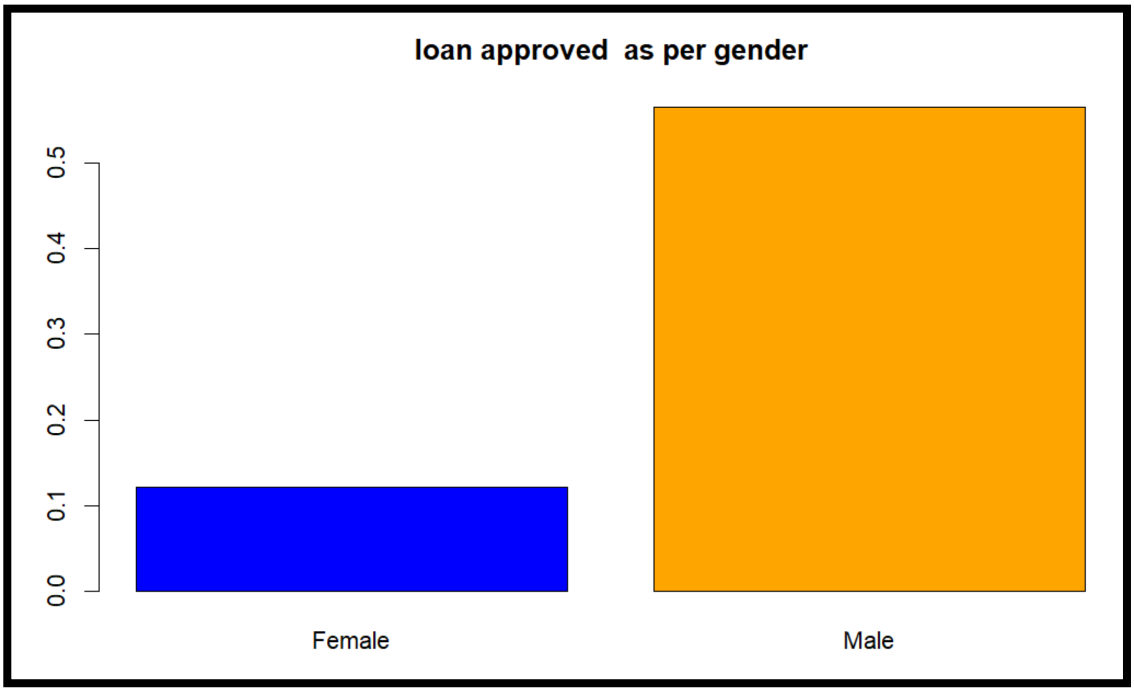
Exploratory Data Analysis & Insights

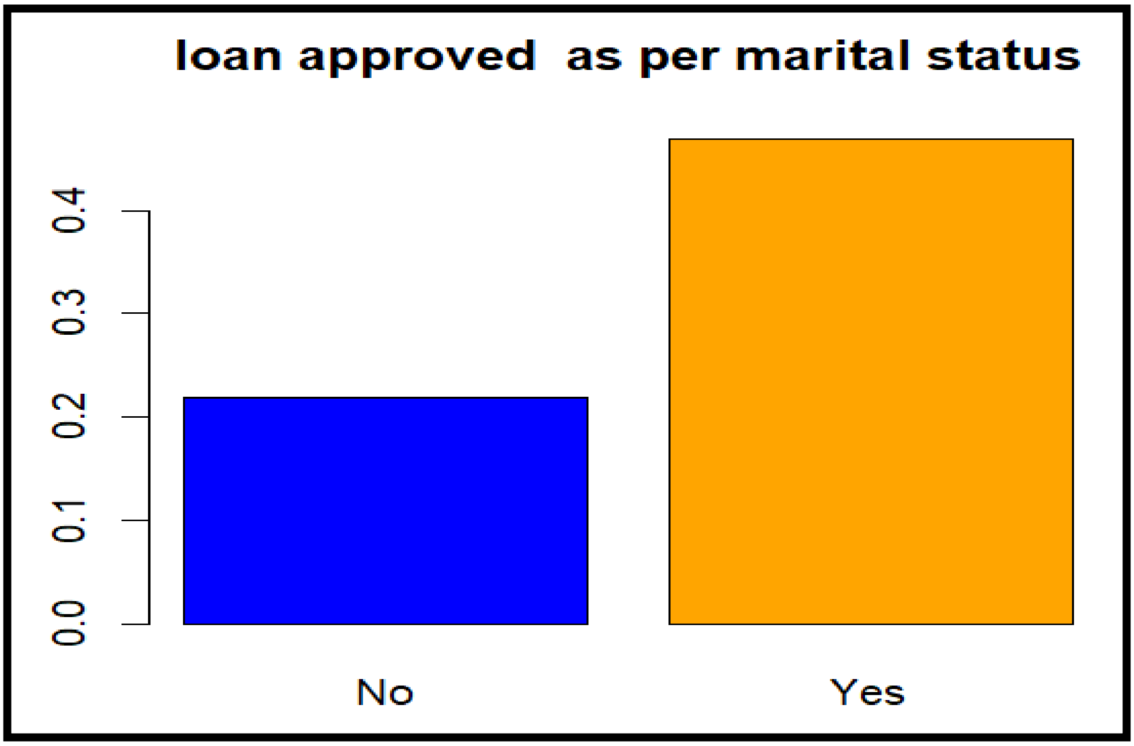
- Correlation Analysis:

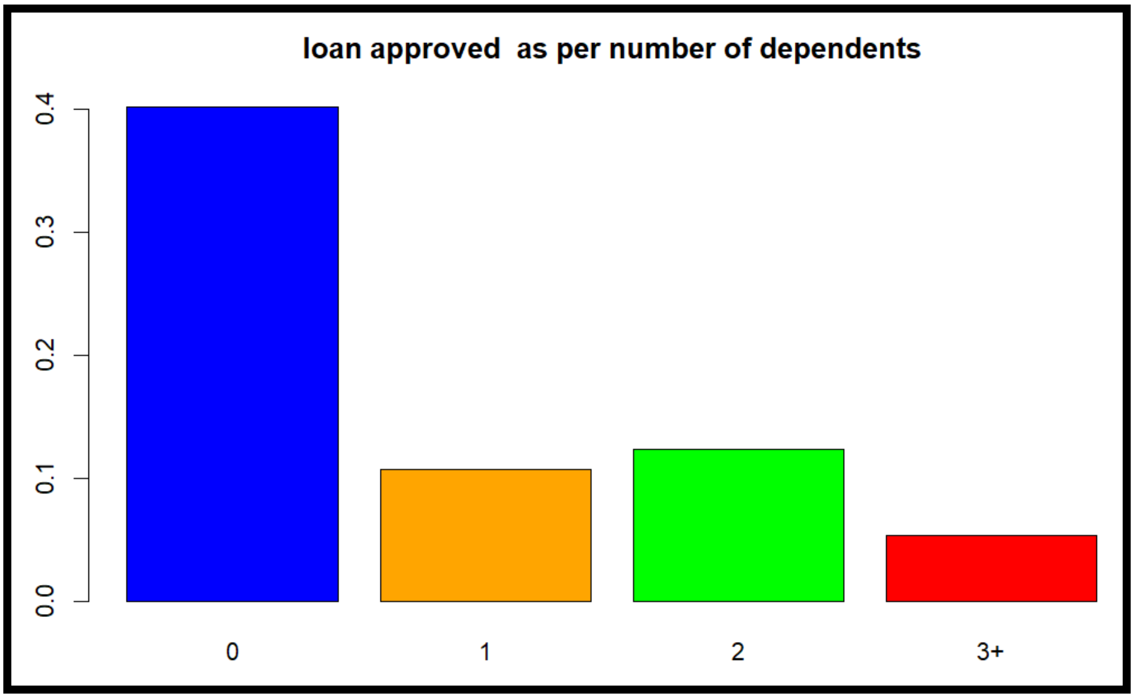
```
> numericdata <- data[sapply(data, is.numeric)]
> cormatrix <- cor(numericdata, use = "complete.obs")
> library(corrplot)
> corrplot(cormatrix, method = "color", addCoef.col = "black",
number.cex = 0.7,
+           tl.col = "black", tl.srt = 45)
```

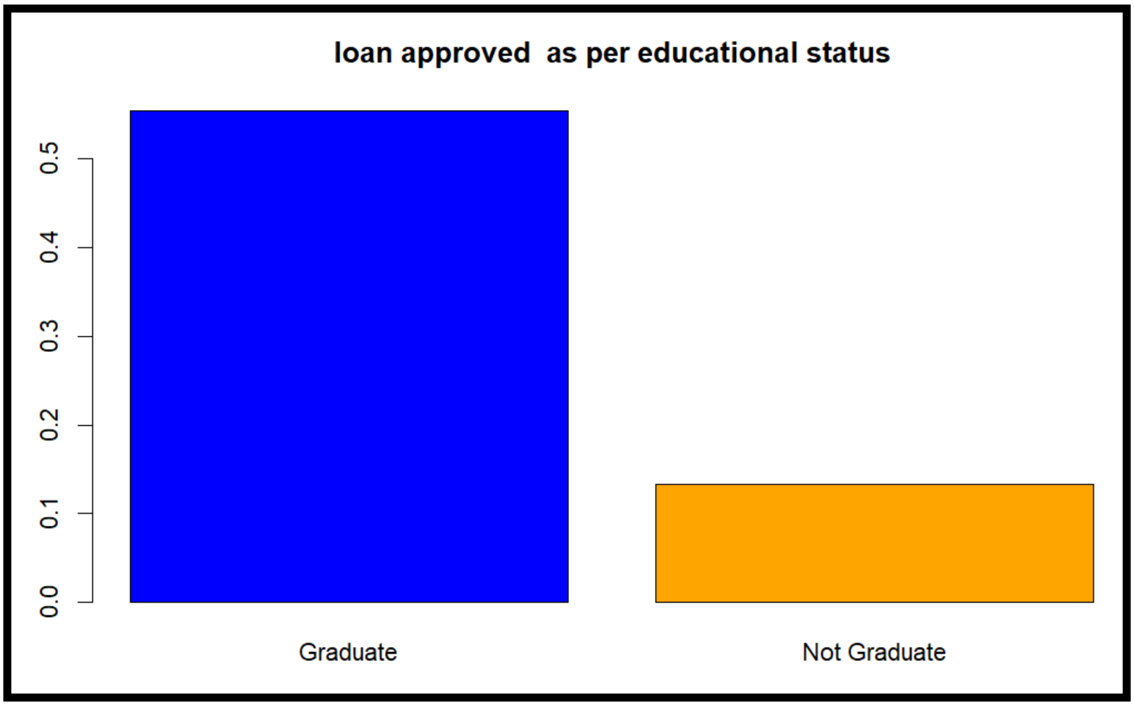


- ◦ Found a positive correlation (~0.62) between LoanAmount and TotalIncome, indicating applicants generally request amounts within their repayment capabilities.
- Approval Patterns Identified: Higher approval rates for:

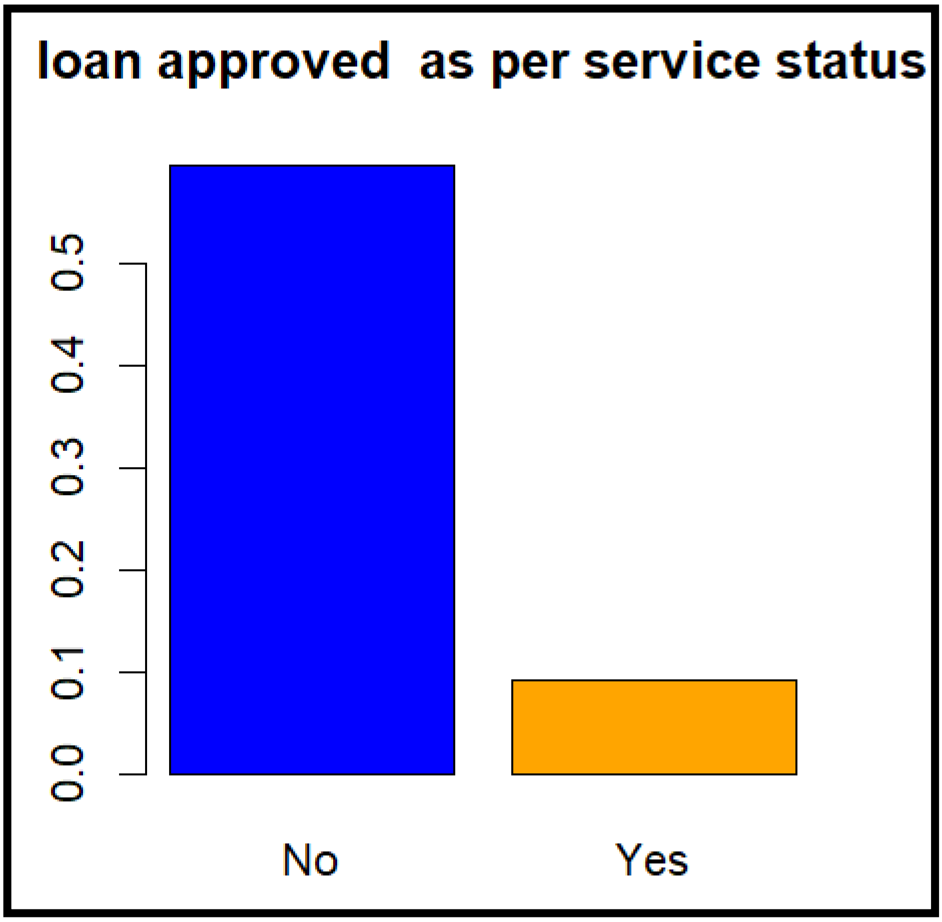



loan approved as per marital status
0.0
0.1
0.2
0.3
0.4
No
Yes

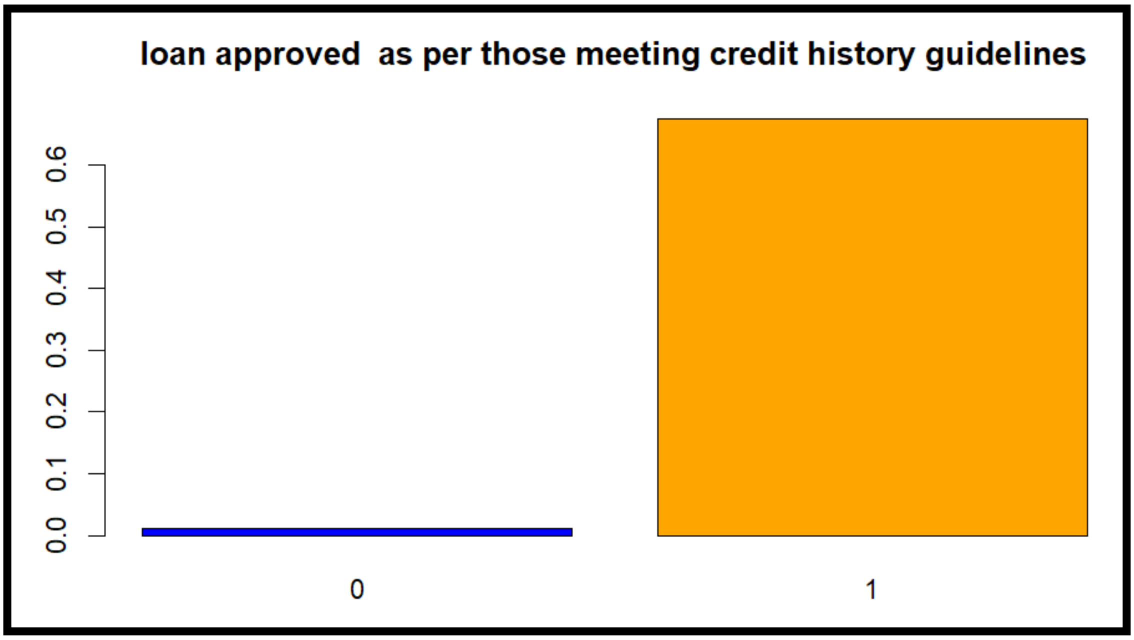

loan approved as per number of dependents
0.0
0.1
0.2
0.3
0.4
0
1
2
3+

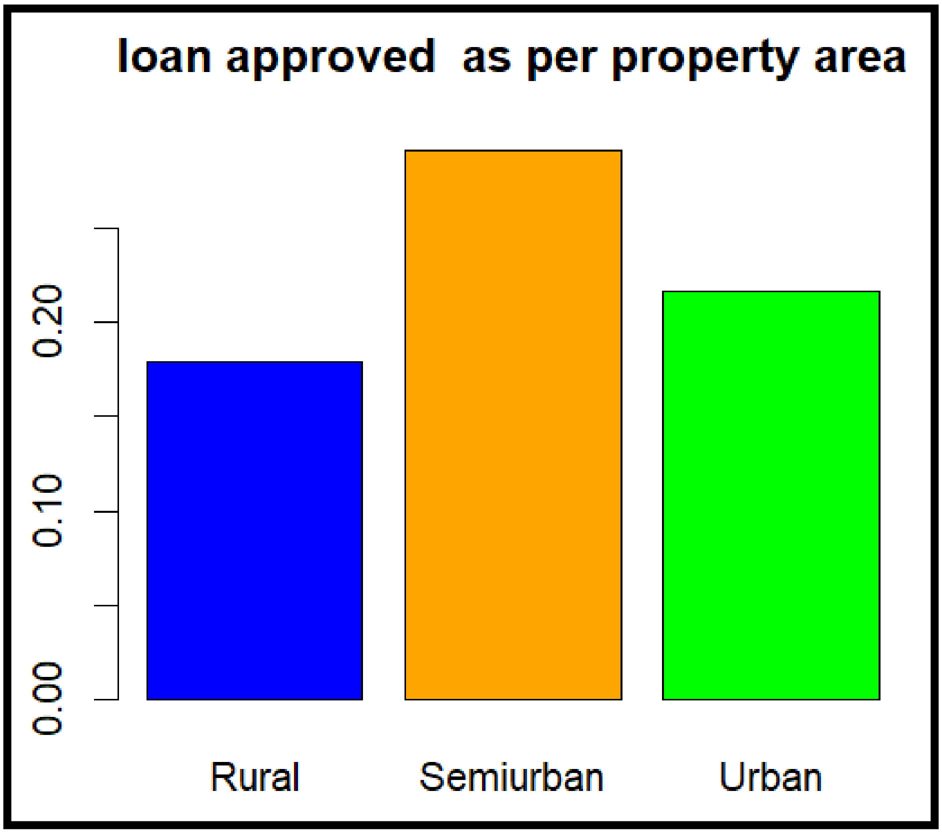


loan approved as per educational status
0.0
0.1
0.2
0.3
0.4
0.5
Graduate
Not Graduate



loan approved as per service status
0.0
0.1
0.2
0.3
0.4
0.5
No
Yes



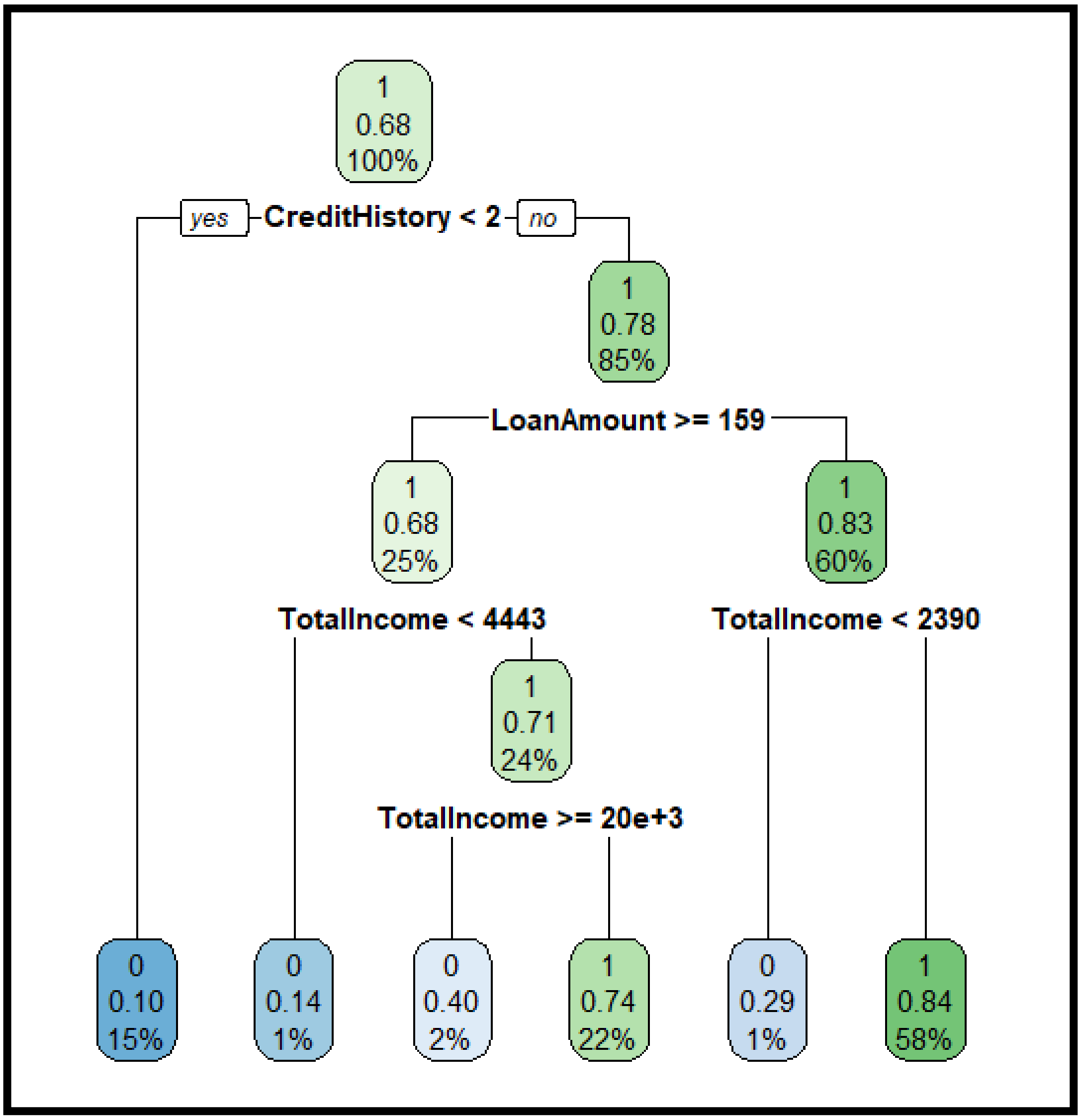


  - Males, Married applicants, Zero dependents, Graduates, Salaried employees, Clean credit history and Semiurban residents
- Feature Selection:
  - Dropped LoadID, ApplicantIncome, and CoapplicantIncome post TotalIncome creation.
- Encoding:
  - Converted categorical variables to numeric using factor() for modelling.

Model Building

- Data Split:
  - Training: 80% (491 samples)

- Testing: 20% (123 samples)
- Models Applied:
    - Decision Tree



        - No scaling required.
        - Accuracy (~0.845)
    - Random Forest
        - Ensemble of 100 trees.
        - Accuracy (~0.854)
    - Logistic Regression
        - Applied after feature scaling.
        - Used probabilistic prediction for binary classification.
        - Accuracy (~0.846)

Testing on New Data

- Applied trained models (Decision Tree, Random Forest, Logistic Regression) on test.csv after replicating preprocessing steps.
- Ensured consistency in pipeline and readiness for deployment.

Results Summary

Metric	Decision Tree	Random Forest	Logistic Regression
Accuracy	~0.73	~0.78	~0.76
Precision	High	Higher	Moderate
Recall	Moderate	Higher	Moderate
F1 Score	Moderate	Higher	Moderate
Type 1 Error Rate	Low	Lower	Low
Type 2 Error Rate	Higher	Lower	Moderate

Random Forest outperformed other models with higher accuracy and balanced precision-recall trade-off.