# Loan default prediction using machine learning

## 1. Introduction

In the sector of competitive financial lending, managing the loan defaults effectively has become crucial for maintaining financial stability and also the profitability of company. In this project we aim to use machine learning techniques to predict loan defaults by which financial institution can make decisions for lending loans and can minimize the risk.
We will start by exploring and preprocessing the data followed by performing exploratory data analysis (EDA) to visualize the distribution of loan defaults and analyse correlations. Using different algorithms such as Logistic Regression, Random Forest, Support Vector Machine (SVM), XGBoost, and Neural Network we will develop predictive models to assess the likelihood of loan defaults. The models will be evaluated using some metrics.

In this project we will identify key features which influence loan defaults and develop a risk scoring system based on result predicted by the model. This system will help in assessing the risk level of each loan applicant and formulating targeted risk mitigation strategies. By integrating these AI-driven solutions the financial institution can optimize its lending portfolio and balance risk and also returns. Hence can enhance overall financial stability. The expected outcome is a significant reduction in loan defaults improved risk management and a more profitable and competitive financial institution.

# 2.Data Description and Exploratory Data Analysis

Dataset Overview

The dataset used in this project contains almost 20000 rows of data of loan applicants with 18 variables.

```
# View the structure of the data
str(loan_data)
```

```
## 'data.frame':    20000 obs. of  18 variables:
##  $ LoanID         : chr  "8EGC3UUTY8" "2ZLI6TAHI5" "8WPZH835VS" "HAU91YNH13" ...
##  $ Age            : int  32 64 21 57 35 25 25 58 19 44 ...
##  $ Income         : int  63892 41848 103298 120690 137245 68120 139889 22190 39792 117250 ...
##  $ LoanAmount     : int  66362 177446 111902 30751 176172 147458 195778 79189 191417 110979 ...
##  $ CreditScore    : int  444 693 689 647 585 405 510 320 540 506 ...
##  $ MonthsEmployed: int  90 98 17 46 71 119 86 113 99 84 ...
##  $ NumCreditLines: int  3 3 3 3 1 2 3 3 4 1 ...
##  $ InterestRate   : num  7.45 16.06 23.19 14.86 12.02 ...
##  $ LoanTerm       : int  36 36 24 48 12 48 48 12 36 60 ...
##  $ DTIRatio       : num  0.27 0.46 0.89 0.14 0.4 0.37 0.52 0.12 0.17 0.43 ...
##  $ Education      : chr  "High School" "High School" "Bachelor's" "Master's" ...
##  $ EmploymentType: chr  "Full-time" "Full-time" "Part-time" "Unemployed" ...
##  $ MaritalStatus : chr  "Single" "Single" "Single" "Single" ...
##  $ HasMortgage    : chr  "No" "No" "No" "Yes" ...
##  $ HasDependents : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ LoanPurpose    : chr  "Education" "Other" "Business" "Home" ...
##  $ HasCoSigner    : chr  "No" "No" "No" "Yes" ...
##  $ Default        : int  0 0 1 0 0 1 0 0 0 0 ...
```

```r
# Summary statistics
summary(loan_data)
```

```
##     LoanID               Age              Income          LoanAmount
##  Length:20000       Min.   :18.00   Min.   : 15009   Min.   :  5009
##  Class :character   1st Qu.:31.00   1st Qu.: 48377   1st Qu.: 65308
##  Mode  :character   Median :44.00   Median : 81941   Median :127356
##                     Mean   :43.54   Mean   : 82115   Mean   :127384
##                     3rd Qu.:56.00   3rd Qu.:115763   3rd Qu.:188757
##                     Max.   :69.00   Max.   :149975   Max.   :249992
##   CreditScore    MonthsEmployed   NumCreditLines  InterestRate
##  Min.   :300.0   Min.   :  0.00   Min.   :1.00   Min.   : 2.00
##  1st Qu.:436.0   1st Qu.: 29.00   1st Qu.:2.00   1st Qu.: 7.78
##  Median :573.0   Median : 59.00   Median :3.00   Median :13.43
##  Mean   :573.7   Mean   : 59.38   Mean   :2.51   Mean   :13.49
##  3rd Qu.:712.0   3rd Qu.: 90.00   3rd Qu.:4.00   3rd Qu.:19.28
##  Max.   :849.0   Max.   :119.00   Max.   :4.00   Max.   :25.00
##     LoanTerm        DTIRatio       Education         EmploymentType
##  Min.   :12.00   Min.   :0.1000   Length:20000       Length:20000
##  1st Qu.:24.00   1st Qu.:0.3000   Class :character   Class :character
##  Median :36.00   Median :0.5000   Mode  :character   Mode  :character
##  Mean   :36.14   Mean   :0.5003
##  3rd Qu.:48.00   3rd Qu.:0.7000
##  Max.   :60.00   Max.   :0.9000
##  MaritalStatus      HasMortgage        HasDependents      LoanPurpose
##  Length:20000       Length:20000       Length:20000       Length:20000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  HasCoSigner           Default
##  Length:20000       Min.   :0.0000
##  Class :character   1st Qu.:0.0000
##  Mode  :character   Median :0.0000
##                     Mean   :0.1163
##                     3rd Qu.:0.0000
##                     Max.   :1.0000
```

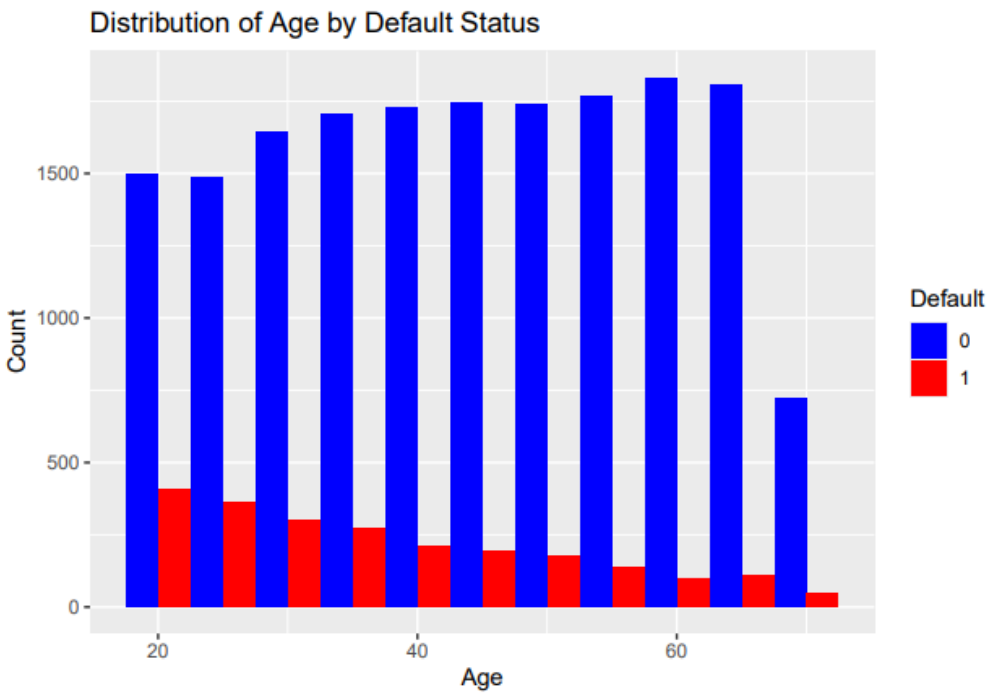
Data preprocessing:

In this step we converted categorical variables such as education, employment type, marital status and loan purpose into factors, which helps for training models properly. Also binary variables like hasmortage, has dependents, has cosigners and default are also converted to factors to ensure they are handled properly during analysis. Also we dropped the variable ID as it do not contribute in prediction of loan default.

Exploratory Data Analysis

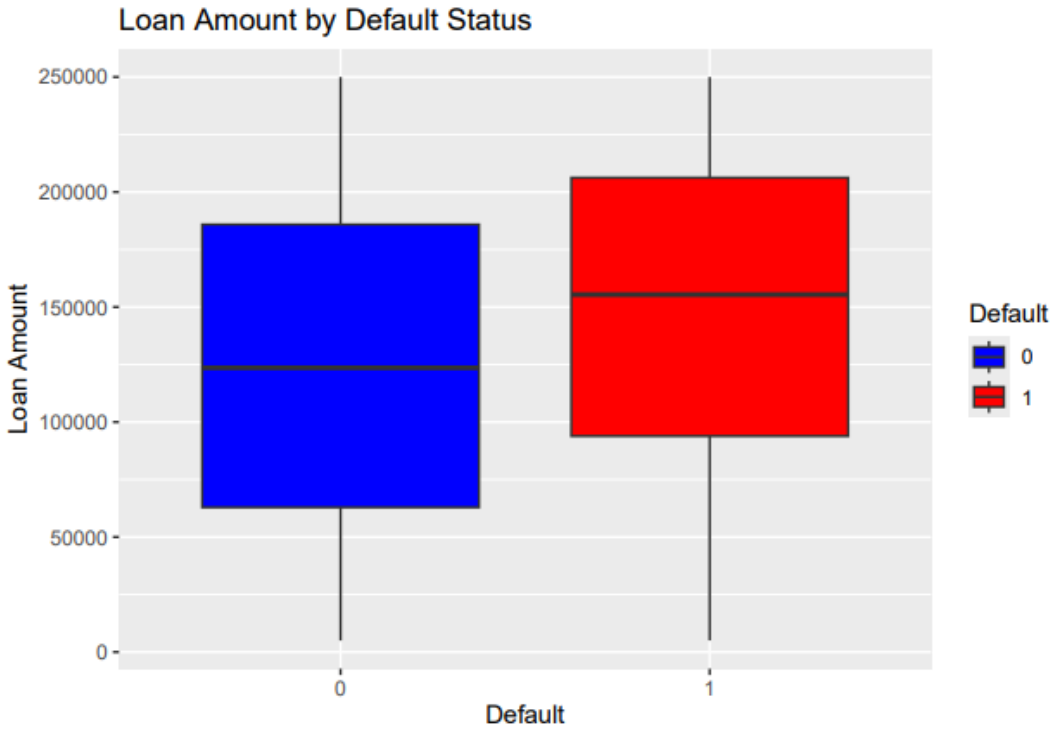
We are exploring the dataset using various plots such as bar, histogram, box and heat maps. By this we can use the appropriate variables for our analysis.

1.1 Age Distribution by Default Status



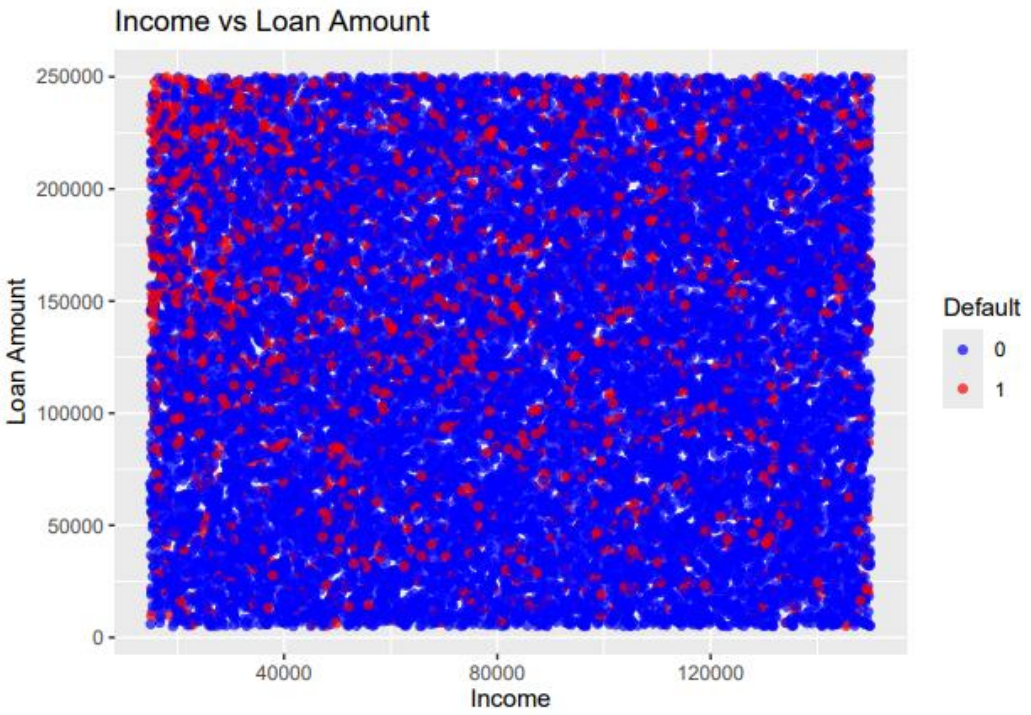

By this above plot we can visualise the distribution of age by default status. By this plot we can see that there are more no of borrowers having age between 25 to 45. Young borrower i.e between age 20 to 40 have a higher defaults, when compare to older borrows. The default rate decreases with age which shows that older borrowers are comparatively less risky borrowers.

2.2 Loan Amount by Default Status



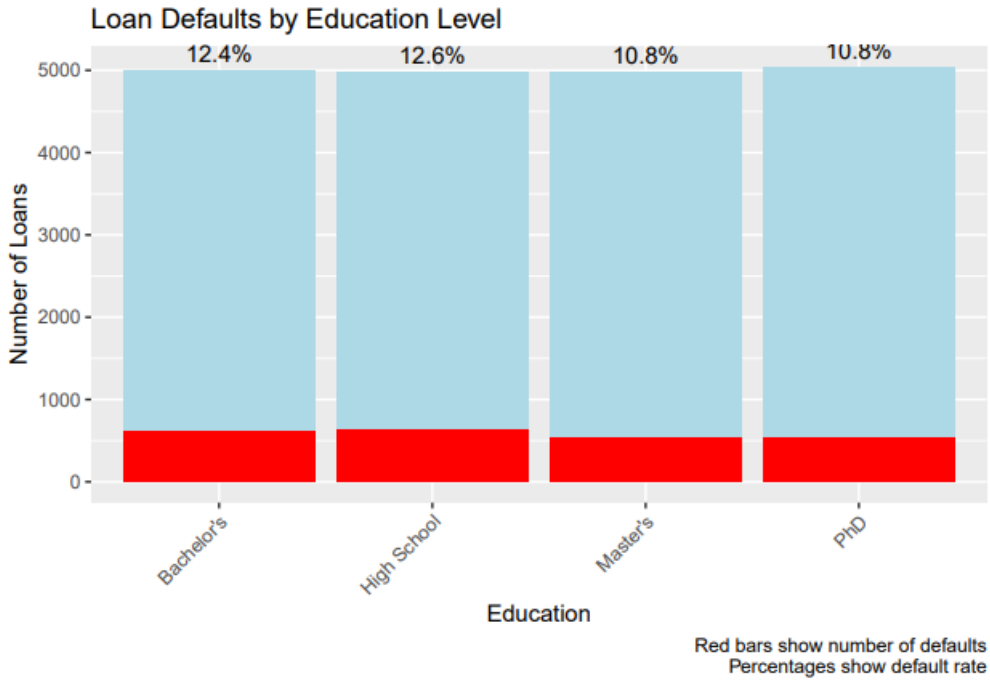

Key observations from above plot is that the loan amount range is from 5000 to 250000.The median of loan amount for defaulted loans is higher than non defaulted loan amount median, which shows that higher the loan amount risk of default also increases. We can also observe that there is overlapping in loan amounts between defaulted and non defaulted loans which indicates that the loan amount cannot be a sole feature for prediction of default status.

2.3 Income vs Loan Amount




Key observation form above scatter plot is that there is a positive correlation between income and loan amount. Default is higher in higher loan amount across various income levels. Also default rate (possibility of defaulting) is much higher when loan amount is higher of a low income borrower.

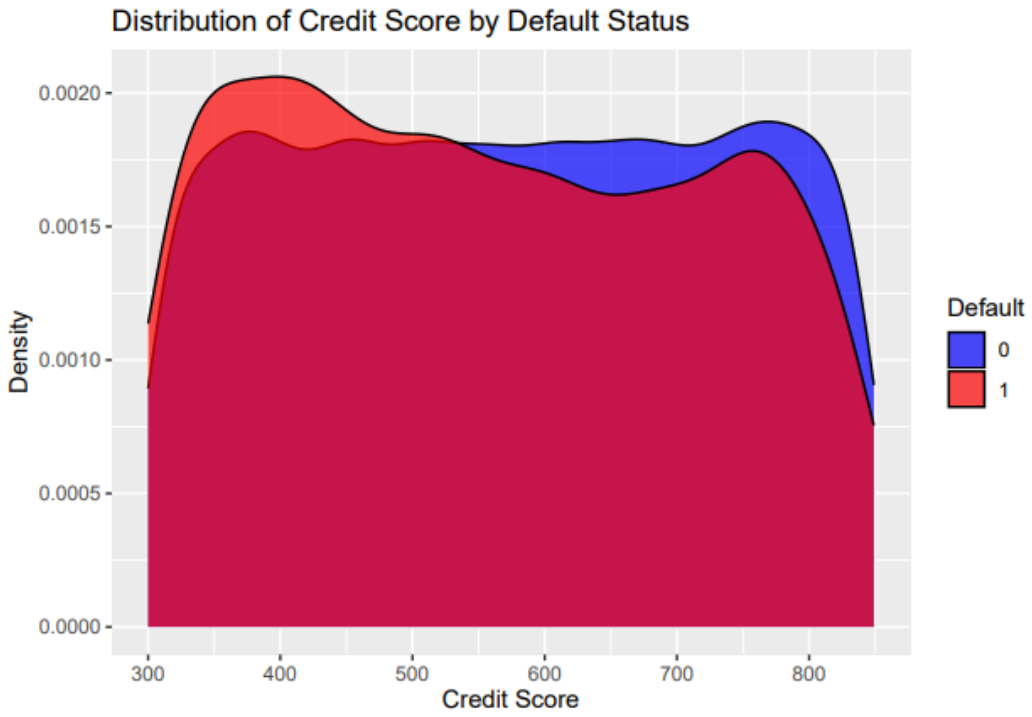
2.4 Loan Defaults by Education Level




Key insights from above bar plot is that PhD holders have the lowest default rate of 10.8% which means that out of 100 PhD holders only 10 are defaulting the loan amount. Master degree holders have default rate of 10.8% and Bachelor's degree holders have 12.4% default rate. High School graduates have the highest default rate of 12.6%.

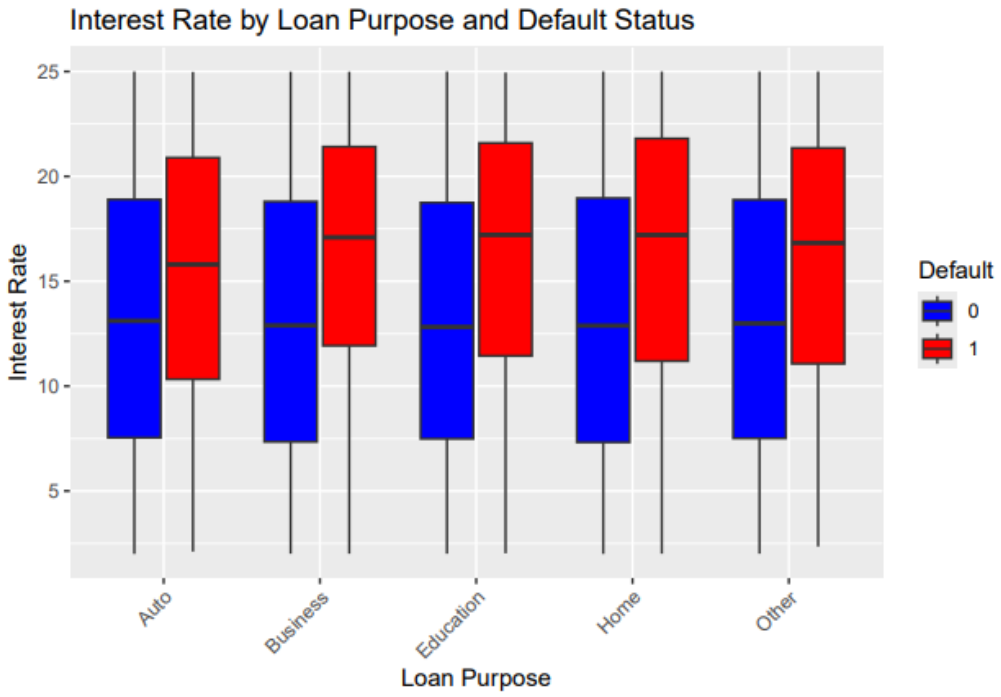
This above analysis shows that higher the education level lower will be the default rate. But the difference in default rates across education is very small which shows that the education level alone may not be a strong predictor of default risk.

2.5 Credit Score Distribution by Default Status



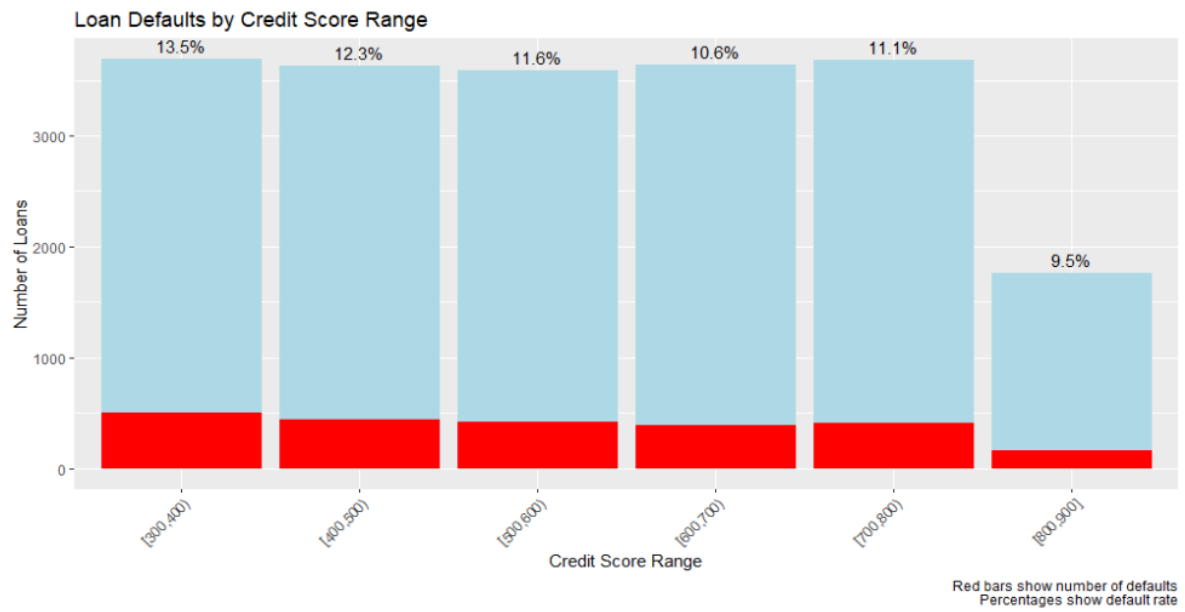

This density plot shows that the credit scores range from 300 to 850. By this plot we can see that lower credit scores have higher density of non defaulted values whereas lower credit scores have higher density of defaulted values, which shows that higher the value of credit score lower will be the risk of default. We also can see that there is overlapping in the range between 500 to 700 which indicates that credit score cannot alone be a significant feature for accurate default prediction.

2.6 Interest Rate by Loan Purpose and Default Status



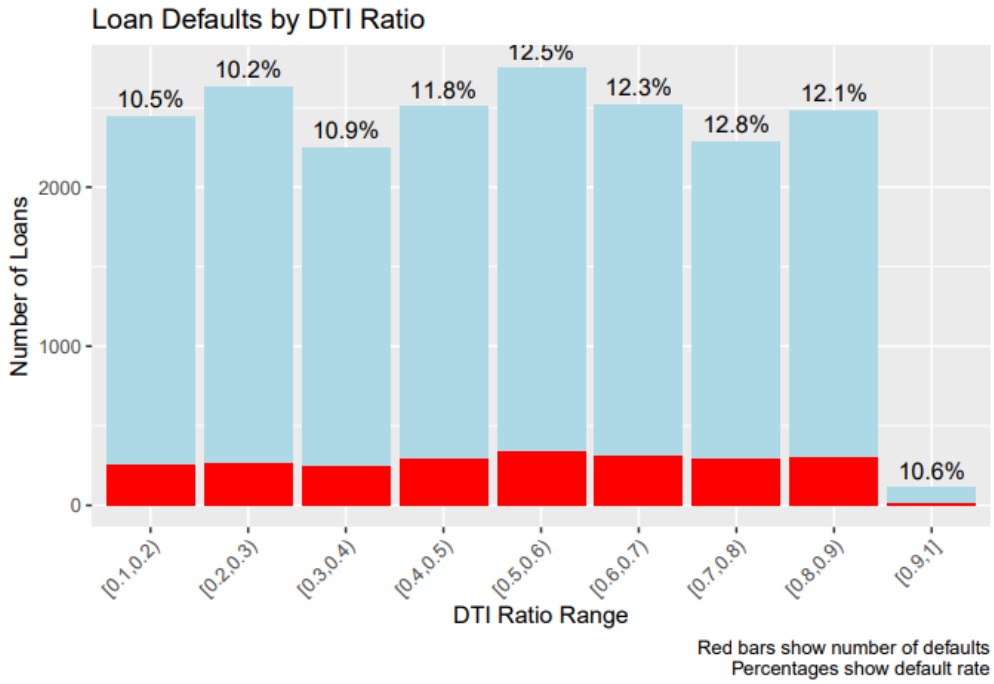

Key observations form above plot is that Business loans have higher interest rate which shows higher risk of default. Home loans have lower average interest rates. Defaulted loans across all purposes have higher interest rates. This shows interest rates and purpose of loan also can be good feature for default prediction.

2.7Loan Defaults by Credit Score Range



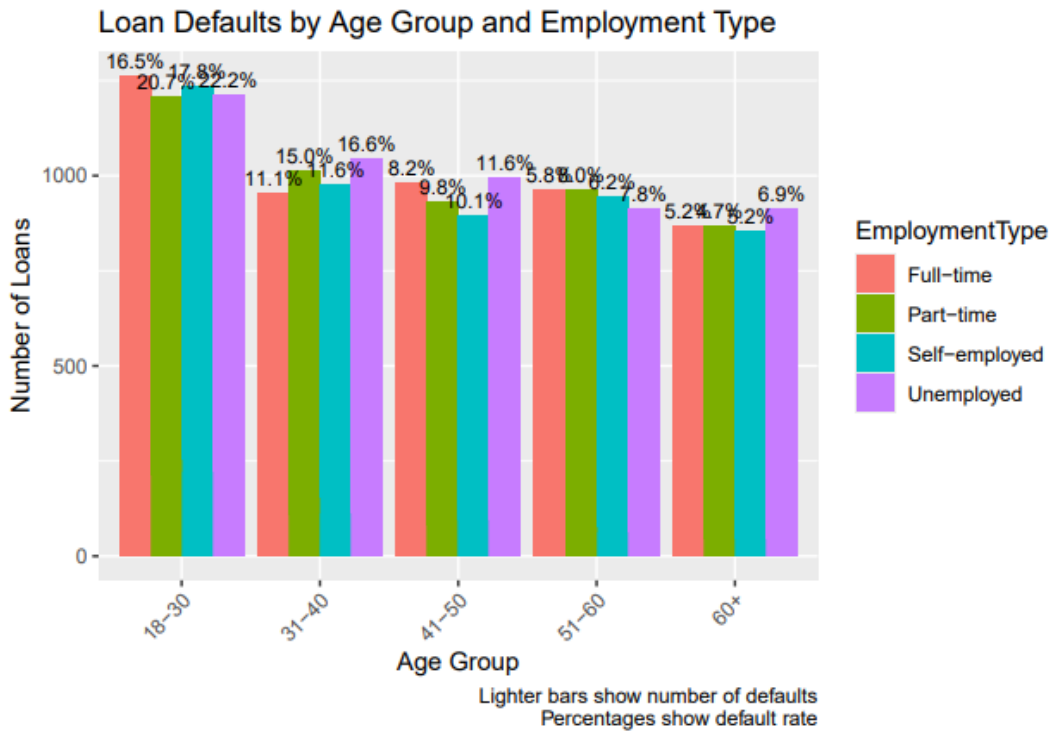

This above graph shows the inverse relationship between credit score and default rates i.e as credit score increases chances of default decreases. The lowest range (300-400) is having highest default rate of 13.4 and 800-900 range is having lowest default rate of 9.5%. These insights suggest implementing stricter criteria for low credit scores.

2.8 Loan defaults by DTI Ratio



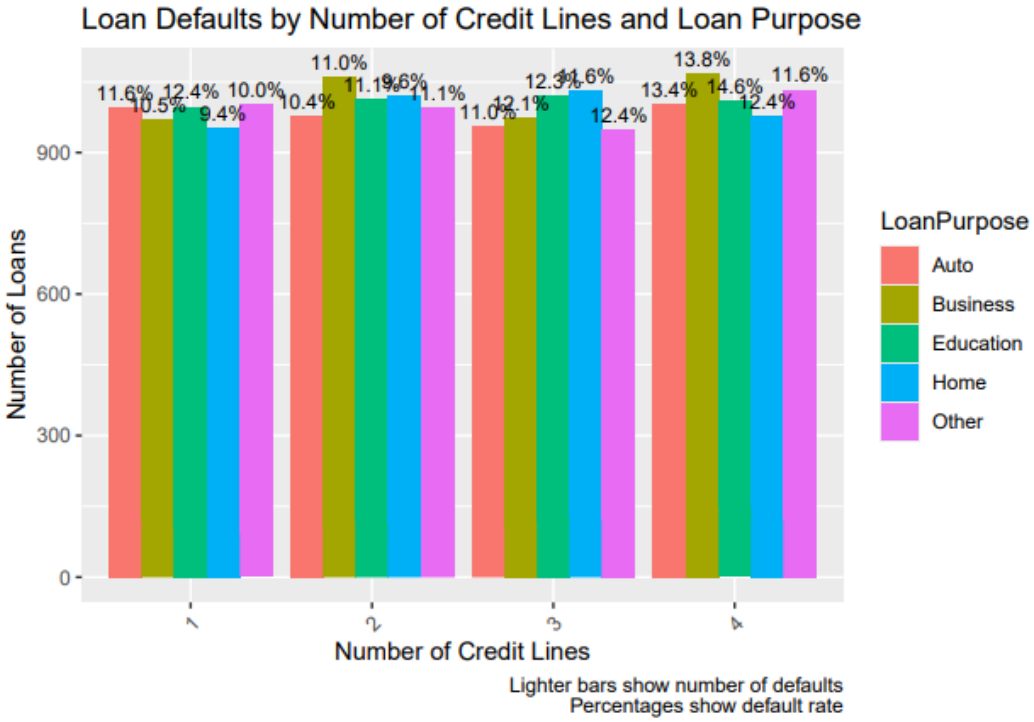

This above bar plot shows the default rate for range of DTI ratios. Each bar height indicate the no of loan and red portion shows the no of defaults. Percentage above each bar represent the default rate for a particular DTI range. The default rate did not increase consistently with higher DTI ratio instead it fluctuated. This suggests that while DTI is a factor in loan defaults, other variables likely play significant roles as well.

2.9 Defaults by Age Group and Employment Type



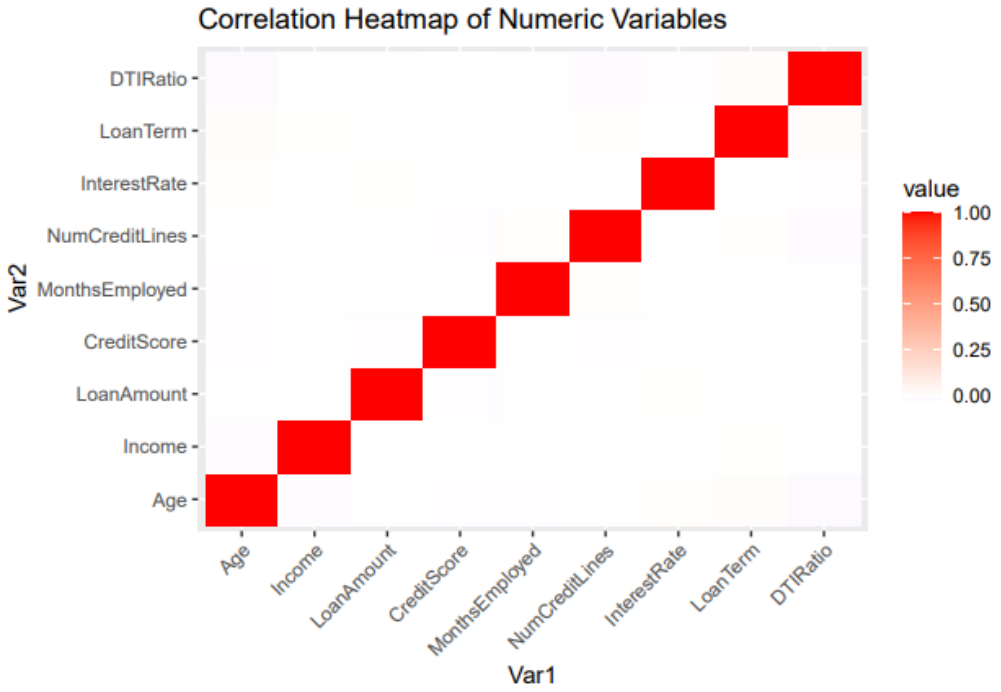

This plot shows the default rates across different age groups and employment types. Findings from this plot is that as age increases chances of default decreases across all employment types. Self employed borrowers have higher default rate in young age groups and unemployed borrowers have highest chance of default across all the age groups.

2.10 Defaults by Number of Credit Lines and Loan Purpose




Stacked bar plot illustrates default rates based on the number of credit lines and loan purpose. Insights from above plot is that as default rate increases with increase in no of credit lines. Business loans have higher default rates particularly with more credit lines. Home loans have lower default rates across different no of credit lines.

2.11 Correlation Heatmap of Numeric Variables




Outcome of corelation analysis:

Income and loan amount have a positive correlation which shows that as income increases loan amount also might increase as well. Credit score and interest rate have a negative correlation which shows that higher credit scores borrowers are given loans on a lower interest rate. Age and no of employed months have a good positive correlation.

In summary correlation analysis shows the complex relation between all features and loan default. Variables like income, loan amount, credit score and interest rate shows non linear relation with default variable. By using models like random forest we can capture non linear patterns.

# 3. Results and Analysis

## 3.1 Model Implementation and Comparison

We implemented five machine learning models namely logistic regression, random forest, support vector machine, XGBoost, neural network. To implement these models we splitted the dataset in to 2 parts 80 percent for training purpose and 20 percent for testing.

Performance Comparison:

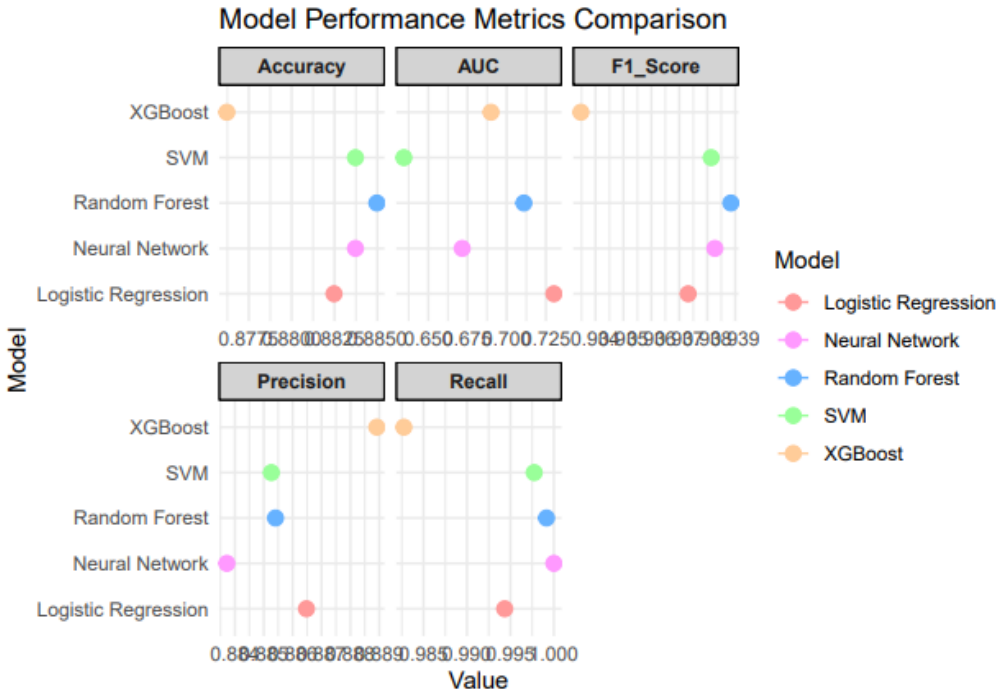
```
# Combine all metrics
all_metrics <- rbind(
  cbind(Model = "Logistic Regression", logit_metrics),
  cbind(Model = "Random Forest", rf_metrics),
  cbind(Model = "SVM", svm_metrics),
  cbind(Model = "XGBoost", xgb_metrics),
  cbind(Model = "Neural Network", nn_metrics)
)

print(all_metrics)
```

```
##                        Model  Accuracy Precision    Recall  F1_Score       AUC
## Accuracy  Logistic Regression 0.8824706 0.8864783 0.9943407 0.9373166 0.7301185
## Accuracy1       Random Forest 0.8849712 0.8854062 0.9991511 0.9388461 0.7109690
## Accuracy2                 SVM 0.8837209 0.8852624 0.9977363 0.9381402 0.6344086
## Accuracy3             XGBoost 0.8762191 0.8889173 0.9827391 0.9334767 0.6899988
## Accuracy4      Neural Network 0.8837209 0.8837209 1.0000000 0.9382716 0.6716180
```

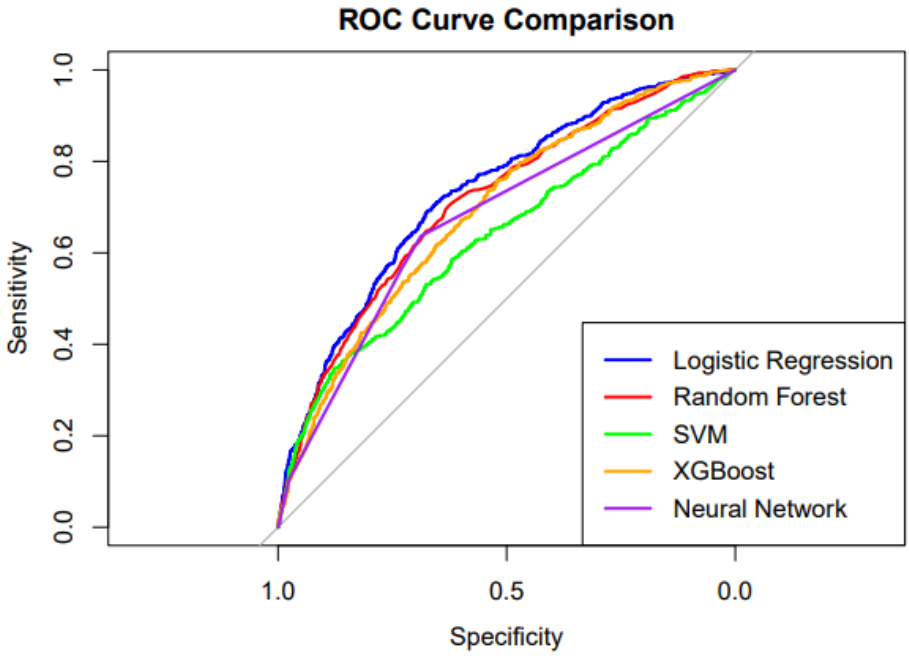
Findings from the above output

All the models have achieved good accuracy higher than 87% and also good precision higher than 88%. Random forest model and neural network model have got the high recall score greater than 99%. Comparing all the models across all the metrices the random forest model have outperformed all the other models,




From above plot we can see that the random forest model has performed good overall and performed better when compared to other models across all metrices.

3.2 ROC Curve Comparison



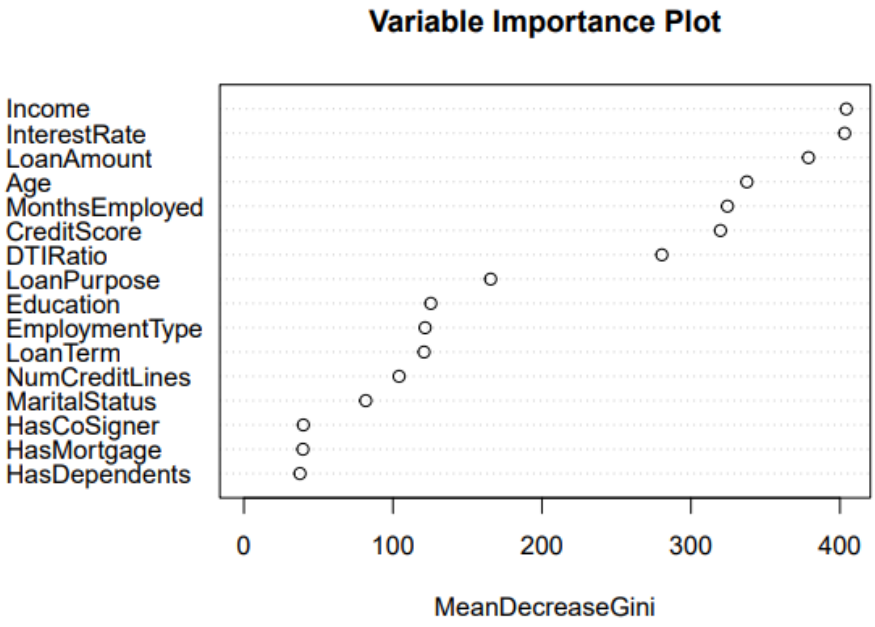
The ROC curves shows that all the models performed better than the random guessing i.e diagonal line. Random forest and logistic regression models show better AUC compared to all other models. The model's performance is quite close at lower false positive rates.

3.3 Model Selection:

Random Forest model is selected for these below reasons:

1. Overall highest accuracy of 88.50%.
2. Good score of precision and recall i.e 88.54% and 99.92% respectively, which shows good balance between both.
3. Highest F1 score of 93.88%.
4. In ROC curve comparison it had a strong AUC performance.
5. It has good ability to handle both numerical and categorical variables.

3.4 Feature Importance Analysis

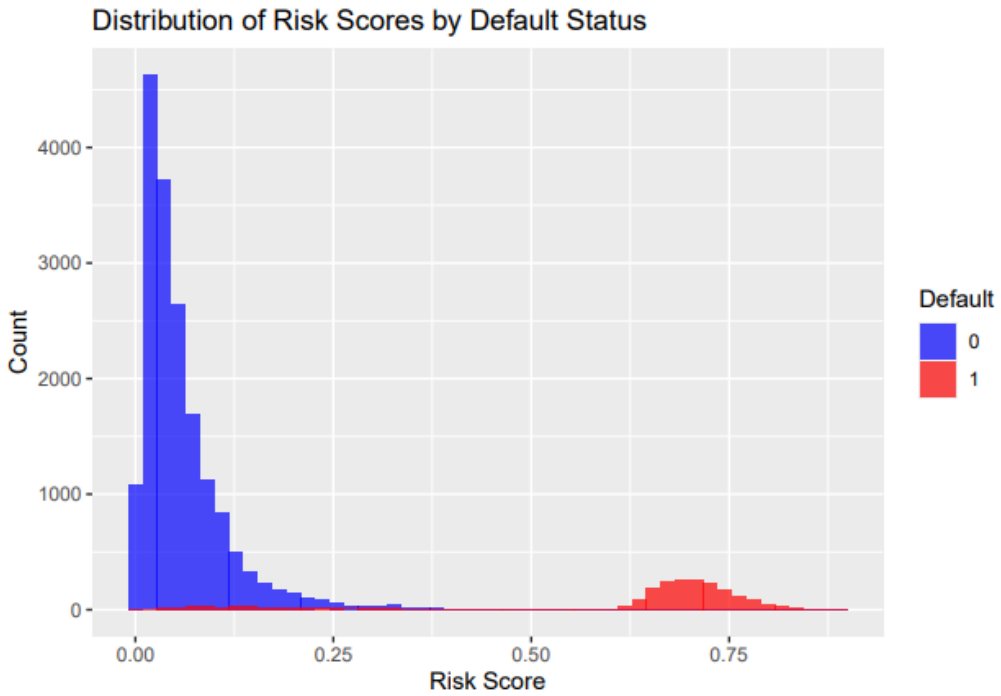


Key insights drawn form the random forest feature importance are:

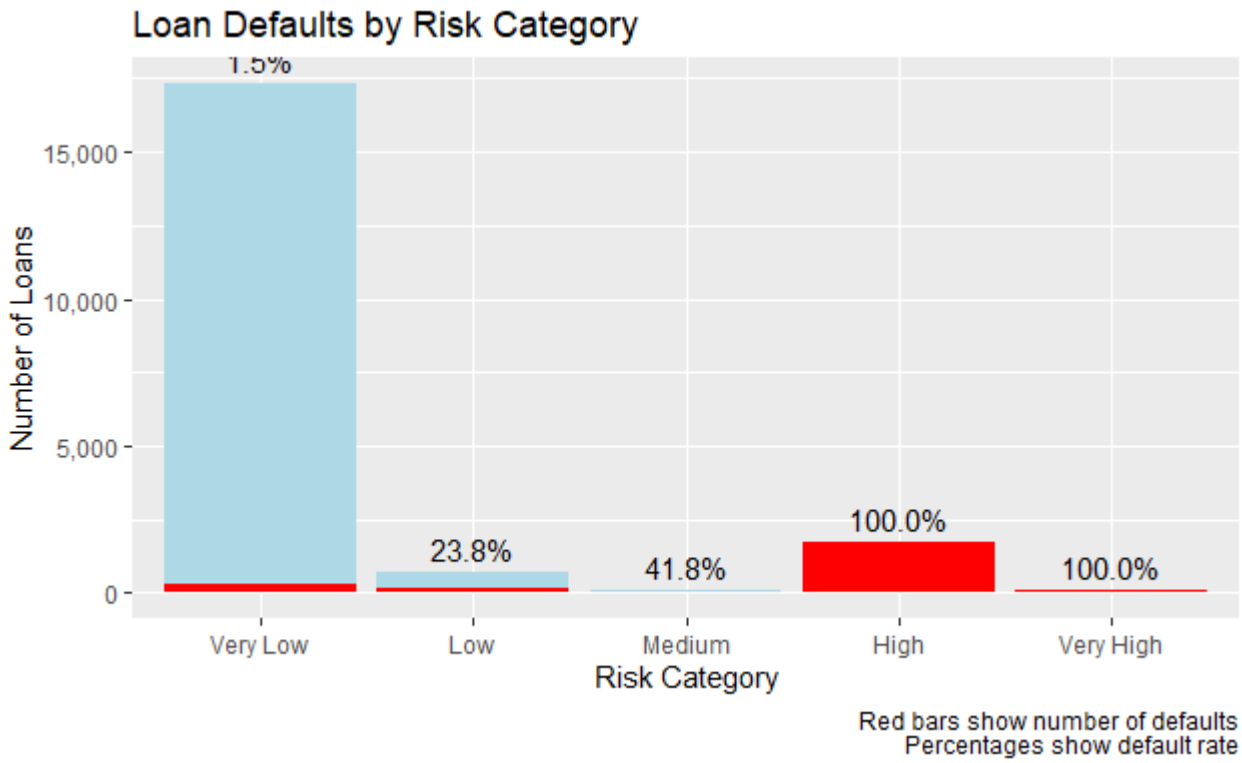
1. Borrower's income is the most important feature which shows that borrower's earnings will have more influence on default possibility.
2. Interest rate is the second most important feature which shows that the loans with higher interest rates have high chance of default.
3. Loan amount is the third important feature which shows that the size of the loan plays important role in default prediction.
4. Age and months employed are the next most significant features , these suggests that older and more stable income borrowers are less likely to default a loan.
5. Credit score is also a important feature as we can see in the above plot.
6. Debt to income ratio, loan purpose, education etc., are the next important features which influence default prediction.

3.5 Risk Scoring System

Using the Random Forest model, we developed a risk scoring system:

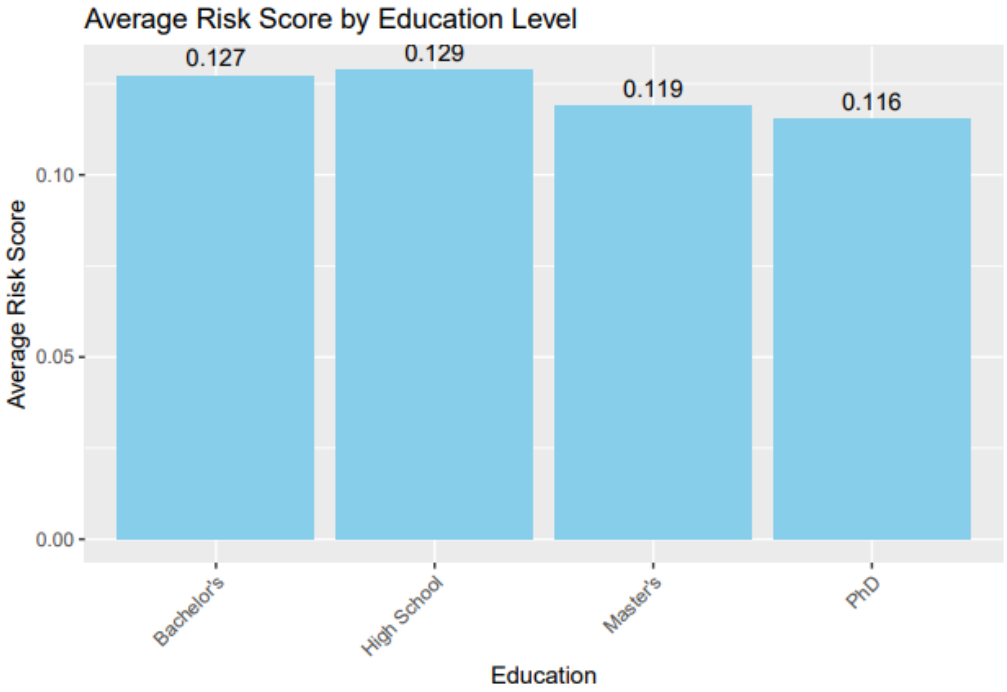



The above histogram plot shows a clear separation between scores of defaulted and non defaulted loans. As we can see in the plot non defaulted loans which is in blue colour is concentrated in range 0.0 to 0.025 which is loan risk range. Non defaulted loans represented in red colour have higher concentration near high risk score range. Hence risk scoring system is successful in separating and predicting high risk and low risk loans.




We have separated every loans different categories namely very low risk, low risk, medium risk, high risk and very high risk categories. As we can see in above histogram plot that very low risk category has lowest default rate of 1.5%, this shows that only 1.5 borrowers are likely to default a loan out of 100 borrowers if our system predicted it into a very low category. Low risk category has default rate of 23.8% , medium risk category has default rate of 41.8%.

High risk and very high risk categories have default rate of 100%  which shows that if our risk scoring system classify a borrower into these 2 category then the borrower is most likely or for sure going to default a loan. This clear progression validates the effectiveness of our risk scoring system.




Average risk scores by education level: This above plot shows that more qualified borrowers are less likely to default a loan. This align with our previous finding that higher education levels are associated with lower default rate.

## 4. Conclusion and discussion

Based on our analysis we can conclude that the random forest model performed better than all other models and also showed good performance in predicting the loan defaults with high accuracy, precision and recall.

Income, interstate and loan amount are the most important factors for the loan default prediction. The risk scoring system can effectively categorise the loans into different risk levels showing difference in default rate across all categories.

We recommend to implement the random forest model and risk scoring system in the loan approval process. Use the model to generate a risk score for each loan application, also establish clear policies for each risk category. Loan lending companies should focus on these key predictors in loan assessment and should take necessary steps such as cross verifying borrower’s income and should consider additional collateral for high risk applicants.

By implementing these recommendations financial institution can potentially reduce default rates and can do risk management. The data driven approach provided by this analysis allows for more informed decision making in the loan approval process ultimately leading to a stronger more resilient loan portfolio.

# Appendix

### Exhibit 1: Structure of data and its description:

Loan default dataset:

1. Loan ID -It is a identifier for each loan borrower.

2. Age- It defines the age of the borrower.

3. Income- It defines the annual income of the borrower.

4. LoanAmount- It denotes the total amount of money borrowed by borrower.

5. CreditScore- It tells the credit score of the borrower.

6. MonthsEmployed- It denotes the number of months the borrower has been employed.

7. NumCreditLines- The number of open credits accounts the borrower has.

8. InterestRate – It is the interest rate applied to the loan.

9. LoanTerm – It is the duration of the loan repayment period in months.

10. DTIRatio - It is is the debt to income ratio of borrower.

11. Education- It is the highest level of education attained by the borrower.

12. Employment Type- It is the type of employment of borrower (e.g., full-time, part-time, self-employed).

13. Marital Status- It shows the marital status of the borrower.

14. HasMortgage-It indicates whether the borrower has an existing mortgage or not.

15. HasDependents-It indicates whether the borrower has dependents or not.

16. LoanPurpose- It is the purpose for which the loan is taken.

17. HasCoSigner- It indicates whether the loan has a co-signer or not.

18.Default – It shows whether borrower defaulted or not (our target variable) 0 means not defaulted and 1 means defaulted.

**Exhibit 2: Complete R Code for the Loan Default Prediction and developing risk scoring system**

```r
# Load required libraries
library(tidyverse)

library(ggplot2)
library(randomForest)

library(e1071)
library(pROC)

library(gridExtra)

library(xgboost)

library(nnet)



# Step 1: Load and explore the data
loan_data <- read.csv("D:/project/Loan_default_updated.csv")

# View the structure of the data
str(loan_data)

# Summary statistics
summary(loan_data)

# Step 2: Data preprocessing
# Convert categorical variables to factors
loan_data$Education <- as.factor(loan_data$Education)
loan_data$EmploymentType <- as.factor(loan_data$EmploymentType)
loan_data$MaritalStatus <- as.factor(loan_data$MaritalStatus)
loan_data$LoanPurpose <- as.factor(loan_data$LoanPurpose)

# Convert binary variables to factors
loan_data$HasMortgage <- as.factor(loan_data$HasMortgage)
loan_data$HasDependents <- as.factor(loan_data$HasDependents)
loan_data$HasCoSigner <- as.factor(loan_data$HasCoSigner)
loan_data$Default <- as.factor(loan_data$Default)

# Step 2: Enhanced Data Visualization

# 1. Distribution of Age
p1 <- ggplot(loan_data, aes(x = Age, fill = Default)) +
  geom_histogram(binwidth = 5, position = "dodge") +
  labs(title = "Distribution of Age by Default Status", x = "Age", y = "Co
unt") +
  scale_fill_manual(values = c("0" = "blue", "1" = "red"))
print(p1)
```

```r
# 2. Loan Amount by Default status
p2 <- ggplot(loan_data, aes(x = Default, y = LoanAmount, fill = Default))
+
  geom_boxplot() +
  labs(title = "Loan Amount by Default Status", x = "Default", y = "Loan A
mount") +
  scale_fill_manual(values = c("0" = "blue", "1" = "red"))
print(p2)



# 3. Income vs LoanAmount, colored by Default status
p3 <- ggplot(loan_data, aes(x = Income, y = LoanAmount, color = Default))
+
  geom_point(alpha = 0.7) +
  labs(title = "Income vs Loan Amount", x = "Income", y = "Loan Amount") +
  scale_color_manual(values = c("0" = "blue", "1" = "red"))
print(p3)



# 4. Default Rate by Education
p4 <- loan_data %>%
  group_by(Education) %>%
  summarise(
    DefaultCount = sum(Default == "1"),
    TotalCount = n(),
    DefaultRate = DefaultCount / TotalCount
  ) %>%
  ggplot(aes(x = Education, y = TotalCount)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  geom_bar(aes(y = DefaultCount), stat = "identity", fill = "red") +
  geom_text(aes(label = sprintf("%.1f%%", DefaultRate*100), y = TotalCount
), vjust = -0.5) +
  labs(
    title = "Loan Defaults by Education Level",
    x = "Education",
    y = "Number of Loans",
    caption = "Red bars show number of defaults\nPercentages show default
rate"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
print(p4)



# 5. Distribution of Credit Score by Default Status
p5 <- ggplot(loan_data, aes(x = CreditScore, fill = Default)) +
  geom_density(alpha = 0.7) +
  labs(title = "Distribution of Credit Score by Default Status", x = "Cred
it Score", y = "Density") +
  scale_fill_manual(values = c("0" = "blue", "1" = "red"))
print(p5)
```

```r
# 6. Interest Rate by Loan Purpose
p6 <- ggplot(loan_data, aes(x = LoanPurpose, y = InterestRate, fill = Default)) +
  geom_boxplot() +
  labs(title = "Interest Rate by Loan Purpose and Default Status", x = "Loan Purpose", y = "Interest Rate") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_manual(values = c("0" = "blue", "1" = "red"))
print(p6)



# 7. Defaults by Credit Score Range
p7 <- loan_data %>%
  mutate(CreditScoreRange = cut(CreditScore, breaks = seq(300, 900, by = 100),include.lowest = TRUE, right = FALSE)) %>%
  group_by(CreditScoreRange) %>%
  summarise(
    DefaultCount = sum(Default == "1"),
    TotalCount = n(),
    DefaultRate = DefaultCount / TotalCount
  ) %>%
  ggplot(aes(x = CreditScoreRange, y = TotalCount)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  geom_bar(aes(y = DefaultCount), stat = "identity", fill = "red") +
  geom_text(aes(label = sprintf("%.1f%%", DefaultRate*100), y = TotalCount), vjust = -0.5) +
  labs(
    title = "Loan Defaults by Credit Score Range",
    x = "Credit Score Range",
    y = "Number of Loans",
    caption = "Red bars show number of defaults\nPercentages show default rate"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
print(p7)



# 8. Defaults by DTI Ratio
p8 <- loan_data %>%
  mutate(DTIRange = cut(DTIRatio, breaks = seq(0, 1, by = 0.1),include.lowest = TRUE, right = FALSE)) %>%
  group_by(DTIRange) %>%
  summarise(
    DefaultCount = sum(Default == "1"),
    TotalCount = n(),
    DefaultRate = DefaultCount / TotalCount
  ) %>%
  ggplot(aes(x = DTIRange, y = TotalCount)) +
  geom_bar(stat = "identity", fill = "lightblue") +
```

```r
  geom_bar(aes(y = DefaultCount), stat = "identity", fill = "red") +
  geom_text(aes(label = sprintf("%.1f%%", DefaultRate*100), y = TotalCount
), vjust = -0.5) +
  labs(
    title = "Loan Defaults by DTI Ratio",
    x = "DTI Ratio Range",
    y = "Number of Loans",
    caption = "Red bars show number of defaults\nPercentages show default
rate"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
print(p8)




# 9. Defaults by Age Group and Employment Type
p9 <- loan_data %>%
  mutate(AgeGroup = cut(Age, breaks = c(0, 30, 40, 50, 60, 100),
                        labels = c("18-30", "31-40", "41-50", "51-60", "60
+"))) %>%
  group_by(AgeGroup, EmploymentType) %>%
  summarise(
    DefaultCount = sum(Default == "1"),
    TotalCount = n(),
    DefaultRate = DefaultCount / TotalCount,
    .groups = 'drop'  # Explicitly drop grouping after summarizing
  ) %>%
  ggplot(aes(x = AgeGroup, y = TotalCount, fill = EmploymentType)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_bar(aes(y = DefaultCount), stat = "identity", position = "dodge", a
lpha = 0.5) +
  geom_text(aes(label = sprintf("%.1f%%", DefaultRate * 100), y = TotalCou
nt),
            position = position_dodge(width = 0.9), vjust = -0.5, size = 3
) +
  labs(
    title = "Loan Defaults by Age Group and Employment Type",
    x = "Age Group",
    y = "Number of Loans",
    caption = "Lighter bars show number of defaults\nPercentages show defa
ult rate"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
print(p9)



# 10. Defaults by Number of Credit Lines and Loan Purpose
p10 <- loan_data %>%
  group_by(NumCreditLines, LoanPurpose) %>%
  summarise(
    DefaultCount = sum(Default == "1"),
    TotalCount = n(),
```

```r
    DefaultRate = DefaultCount / TotalCount,
    .groups = 'drop'  # Explicitly drop grouping after summarizing
  ) %>%
  ggplot(aes(x = as.factor(NumCreditLines), y = TotalCount, fill = LoanPur
pose)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_bar(aes(y = DefaultCount), stat = "identity", position = "dodge", a
lpha = 0.5) +
  geom_text(aes(label = sprintf("%.1f%%", DefaultRate * 100), y = TotalCou
nt),
            position = position_dodge(width = 0.9), vjust = -0.5, size = 3
) +
  labs(
    title = "Loan Defaults by Number of Credit Lines and Loan Purpose",
    x = "Number of Credit Lines",
    y = "Number of Loans",
    caption = "Lighter bars show number of defaults\nPercentages show defa
ult rate"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
print(p10)



# Correlation heatmap for numerical variables
numeric_vars <- loan_data %>% select_if(is.numeric)
cor_matrix <- cor(numeric_vars)
p11 <- ggplot(data = reshape2::melt(cor_matrix)) +
  geom_tile(aes(x = Var1, y = Var2, fill = value)) +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint
= 0) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Correlation Heatmap of Numeric Variables")
print(p11)



# Step 4: Split the data into training and testing sets
set.seed(123)
trainIndex <- createDataPartition(loan_data$Default, p = 0.8, list = FALSE
)
train_data <- loan_data[trainIndex,]
test_data <- loan_data[-trainIndex,]



# Step 5: Define the model formula
model_formula <- Default ~ Age + Income + LoanAmount + CreditScore + Month
sEmployed + NumCreditLines + InterestRate + LoanTerm + DTIRatio + Educatio
n +EmploymentType + MaritalStatus + HasMortgage + HasDependents +LoanPurpo
se + HasCoSigner
```

```r
# Step 6: Train models

# Logistic Regression
logit_model <- glm(model_formula, data = train_data, family = "binomial")

# Random Forest
rf_model <- randomForest(model_formula, data = train_data)

# Support Vector Machine
svm_model <- svm(model_formula, data = train_data, kernel = "radial", prob
ability = TRUE)

# XGBoost
train_matrix <- model.matrix(model_formula, data = train_data)[,-1]
test_matrix <- model.matrix(model_formula, data = test_data)[,-1]
dtrain <- xgb.DMatrix(data = train_matrix, label = as.numeric(train_data$D
efault) - 1)
xgb_model <- xgboost(data = dtrain, nrounds = 100, objective = "binary:log
istic")




# Neural Network
nn_model <- nnet(model_formula, data = train_data, size = 5, maxit = 1000)




# Step 7: Make predictions on test data
logit_pred <- predict(logit_model, newdata = test_data, type = "response")
rf_pred <- predict(rf_model, newdata = test_data, type = "prob")[,2]
svm_pred <- predict(svm_model, newdata = test_data, probability = TRUE)
xgb_pred <- predict(xgb_model, newdata = test_matrix)
nn_pred <- predict(nn_model, newdata = test_data, type = "raw")

# Function to calculate metrics
calculate_metrics <- function(actual, predicted, threshold = 0.5) {
  predicted_class <- factor(ifelse(predicted > threshold, "1", "0"), level
s = levels(actual))
  cm <- confusionMatrix(predicted_class, actual)
  auc <- as.numeric(roc(actual, predicted)$auc)

  data.frame(
    Accuracy = cm$overall["Accuracy"],
    Precision = cm$byClass["Precision"],
    Recall = cm$byClass["Recall"],
    F1_Score = cm$byClass["F1"],
    AUC = auc
  )
}

# Calculate metrics for each model
```

```r
logit_pred_class <- factor(ifelse(logit_pred > 0.5, "1", "0"), levels = le
vels(test_data$Default))
logit_metrics <- calculate_metrics(test_data$Default, logit_pred)

rf_pred_class <- factor(ifelse(rf_pred > 0.5, "1", "0"), levels = levels(t
est_data$Default))
rf_metrics <- calculate_metrics(test_data$Default, rf_pred)

svm_pred_class <- factor(ifelse(attr(svm_pred, "probabilities")[,2] > 0.5,
"1", "0"), levels = levels(test_data$Default))
svm_metrics <- calculate_metrics(test_data$Default, attr(svm_pred, "probab
ilities")[,2])

xgb_pred_class <- factor(ifelse(xgb_pred > 0.5, "1", "0"), levels = levels
(test_data$Default))
xgb_metrics <- calculate_metrics(test_data$Default, xgb_pred)

nn_pred_class <- factor(ifelse(as.vector(nn_pred) > 0.5, "1", "0"), levels
= levels(test_data$Default))
nn_metrics <- calculate_metrics(test_data$Default, as.vector(nn_pred))

# Combine all metrics
all_metrics <- rbind(
  cbind(Model = "Logistic Regression", logit_metrics),
  cbind(Model = "Random Forest", rf_metrics),
  cbind(Model = "SVM", svm_metrics),
  cbind(Model = "XGBoost", xgb_metrics),
  cbind(Model = "Neural Network", nn_metrics)
)

print(all_metrics)



# Step 9: Visualize model performance
metrics_long <- all_metrics %>%
  pivot_longer(cols = -Model, names_to = "Metric", values_to = "Value")

# Create a custom color palette
model_colors <- c("Logistic Regression" = "#FF9999", # Light red
                  "Random Forest" = "#66B2FF",       # Light blue
                  "SVM" = "#99FF99",                 # Light green
                  "XGBoost" = "#FFCC99",             # Light orange
                  "Neural Network" = "#FF99FF")      # Light purple

# Create the plot
ggplot(metrics_long, aes(x = Value, y = Model, color = Model)) +
  geom_point(size = 3) +
  geom_errorbar(aes(xmin = Value, xmax = Value), width = 0.2) +
  facet_wrap(~ Metric, scales = "free_x", ncol = 3) +
  labs(title = "Model Performance Metrics Comparison",
       x = "Value",
       y = "Model") +
  scale_color_manual(values = model_colors) +
  theme_minimal() +
```

```r
  theme(strip.background = element_rect(fill = "lightgray"),
        strip.text = element_text(face = "bold"))




# Step 10: Variable Importance (for Random Forest)
varImpPlot(rf_model, main = "Variable Importance Plot")




# Step 11: ROC Curve Comparison
plot(roc(test_data$Default, logit_pred), col = "blue", main = "ROC Curve C
omparison")

plot(roc(test_data$Default, rf_pred), col = "red", add = TRUE)

plot(roc(test_data$Default, attr(svm_pred, "probabilities")[,2]), col = "g
reen", add = TRUE)

plot(roc(test_data$Default, xgb_pred), col = "orange", add = TRUE)

plot(roc(test_data$Default, as.vector(nn_pred)), col = "purple", add = TRU
E)

legend("bottomright", legend = c("Logistic Regression", "Random Forest", "
SVM", "XGBoost", "Neural Network"),
       col = c("blue", "red", "green", "orange", "purple"), lwd = 2)




# Step 12: Enhanced Risk Scoring System
# Calculate risk scores (probability of default)
risk_scores <- predict(rf_model, newdata = loan_data, type = "prob")[,2]

# Create risk categories
risk_categories <- cut(risk_scores,
                       breaks = c(0, 0.2, 0.4, 0.6, 0.8, 1),
                       labels = c("Very Low", "Low", "Medium", "High", "Ve
ry High"),include.lowest = TRUE, right = FALSE)

# Add risk scores and categories to the original data
loan_data$RiskScore <- risk_scores
loan_data$RiskCategory <- risk_categories

# Distribution of Risk Scores
p12 <- ggplot(loan_data, aes(x = RiskScore, fill = Default)) +
  geom_histogram(bins = 50, position = "identity", alpha = 0.7) +
  labs(title = "Distribution of Risk Scores by Default Status",
       x = "Risk Score", y = "Count") +
  scale_fill_manual(values = c("0" = "blue", "1" = "red"))
print(p12)
```

```r
# Default Rate by Risk Category
p13 <- loan_data %>%
  group_by(RiskCategory) %>%
  summarise(
    DefaultCount = sum(Default == "1"),
    TotalCount = n(),
    DefaultRate = DefaultCount / TotalCount
  ) %>%
  ggplot(aes(x = RiskCategory, y = TotalCount)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  geom_bar(aes(y = DefaultCount), stat = "identity", fill = "red") +
  geom_text(aes(label = sprintf("%.1f%%", DefaultRate*100), y = TotalCount
), vjust = -0.5) +
  scale_y_continuous(labels = scales::comma) +
  labs(
    title = "Loan Defaults by Risk Category",
    x = "Risk Category",
    y = "Number of Loans",
    caption = "Red bars show number of defaults\nPercentages show default
rate"
  )
print(p13)



# Average Risk Score by Categorical Variables
p14 <- loan_data %>%
  group_by(Education) %>%
  summarise(AvgRiskScore = mean(RiskScore)) %>%
  ggplot(aes(x = Education, y = AvgRiskScore)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  geom_text(aes(label = sprintf("%.3f", AvgRiskScore)), vjust = -0.5) +
  labs(title = "Average Risk Score by Education Level", x = "Education", y
= "Average Risk Score") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
print(p14)
```