**Predicting Loan Defaults: Final Report**

Saloni Jain,

**Introduction**

The financial industry, specifically the lending sector, has always been in the constant pursuit of refining its risk assessment strategies. Loan defaults can have significant financial implications for lending institutions, affecting not just their bottom lines but also their credibility and trustworthiness in the eyes of stakeholders. It is in this context that the loan default dataset becomes a pivotal tool for these institutions, serving as a mirror reflecting borrower behaviours and associated risk factors.

This report meticulously delves into the dataset to uncover patterns, correlations, and predictive markers. Starting with fundamental business questions that resonate with pressing industry concerns, the analysis progressively moves from preliminary exploratory data phases to sophisticated modelling techniques. The goal is to derive meaningful, actionable insights from the data. By harnessing the power of this dataset, we aim to pinpoint and predict the key factors that most influence loan defaults, equipping lending institutions with the knowledge to make more informed decisions moving forward.

**Data Cleaning**

In our exploration of the "Loan\_Default" dataset, we found it rich with over 148,000 entries spread across 34 informative columns, each delving into various aspects of loan information. Within these columns, a mix of data types emerged. Numeric variables like 'loan\_amount,' 'income,' 'Credit\_Score,' and 'LTV' painted a picture of the financial aspects and credit history of applicants. At the same time, categorical variables such as 'loan\_limit,' 'Gender,' and 'Region' provided qualitative insights, touching upon both loan specifics and the demographics of the applicants.

From the summary statistics, it was evident that the dataset carries a wealth of insights. The average loan size stood at a substantial $331,118, with borrowers typically facing an interest rate of around 4.05%. A majority of these loans seem to be designed with a standard duration, showing a 360-month term. Delving into the incomes of applicants, we saw a wide range, averaging out at roughly $6,957. This, combined with the average credit score of about 699.8, gave a glimpse into the creditworthiness of the dataset's population.

However, it's noteworthy that the loan-to-value (LTV) ratio did indicate some data gaps, even though its average was at 72.75. A deeper look also revealed that a significant 24.64% of the loans were labelled as defaults.

A screenshot of a document

Description automatically generatedThe dataset went beyond mere numbers, with variables like 'loan\_limit' and 'Region' shedding light on conditions of loans and geographical nuances. Other columns also informed us about diverse loan applicants' demographics and details such as the application process and property type.

Table 1: Missing values % & Unique values in the dataset.

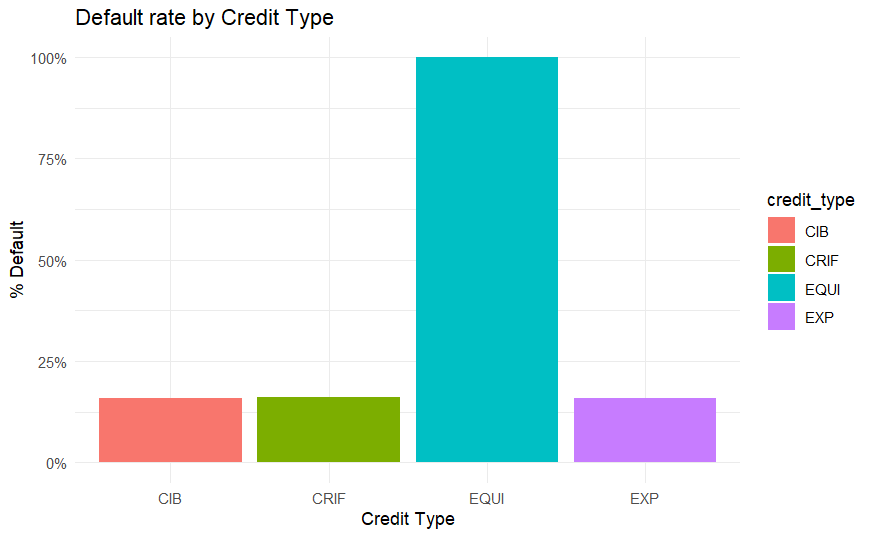
During the cleanup stage, the dataset's integrity was of utmost importance, primarily due to the presence of missing values and potential outliers. An analysis revealed that of the 34 columns presented, 14 displayed data inconsistencies. A specific example is the 'loan\_limit' column, which was missing approximately 2.25% of its data. Given the considerable size of our dataset, these entries were omitted, judging that such a minor missing percentage would not skew our final analysis.

Columns such as 'rate\_of\_interest,' 'Interest\_rate\_spread,' and 'Upfront\_charges' presented substantial data gaps. To address this, we employed median-based imputation, which was determined to be the most suitable given the inherent nature and distribution of the data in these columns. Median-based imputation was chosen because it is less sensitive to outliers and provides a robust estimate, especially for columns with skewed distributions. This ensured that our imputed values did not artificially distort the data, but rather provided a representative and consistent fill for the missing entries. In an effort to maintain dataset clarity, we removed non-essential columns like ‘ID’, ‘year’, ‘construction type’, ‘Security type’, and ‘Secured by’.

A noteworthy observation emerged concerning certain loan applications. Specifically, applications that showcased a property value of less than 1M, yet requested a loan amount surpassing 1M. Such instances were identified as anomalies, grounded in the premise that approvals for such loan configurations are rare. This perspective was bolstered by the striking finding that every single one of these applications had a default rate of 100%, underscoring their removal from our analytical considerations.

**Exploratory Data Analysis**

We began our exploratory analysis by exploring general demographics of Region, Gender, and Age. Comparing the default rate, we saw considerable differences within different classes of these demographics. Northern Region applications had the lowest default rate at 20% while Northeast had the highest at 34%. This meant that of all defaulted loans, 1 in 3 belonged to the North-eastern region. It was also observed that Joint loan applications had considerably lesser defaults than single applications. For the age categories above 74 and below 25, we observed higher default rates (30%) than other age groups. However, the difference was not significant amongst the other age groups (~24%)



Another significant finding was regarding the credit type. It was observed that 99% of applicants with Credit type EQUI defaulted on their loans which can be observed from figure 1, This indicates that mere presence of this Credit type is indicative of a potential defaulting loan.

Figure 1:Bar plot of Default rate by credit type.

Amortisation, which is an accounting technique used to periodically lower the book value of a loan or an intangible asset over a set period of time. Concerning a loan, amortisation focuses on spreading out loan payments over time. Negative amortisation means that even when you pay, the amount you owe will still go up because you are not paying enough to cover the interest. It was found that 41% of the negatively amortised loans defaulted, while nonnegative amortised loans defaulted at half the rate. Therefore, it is also a significant factor in determining potential defaulting loans.

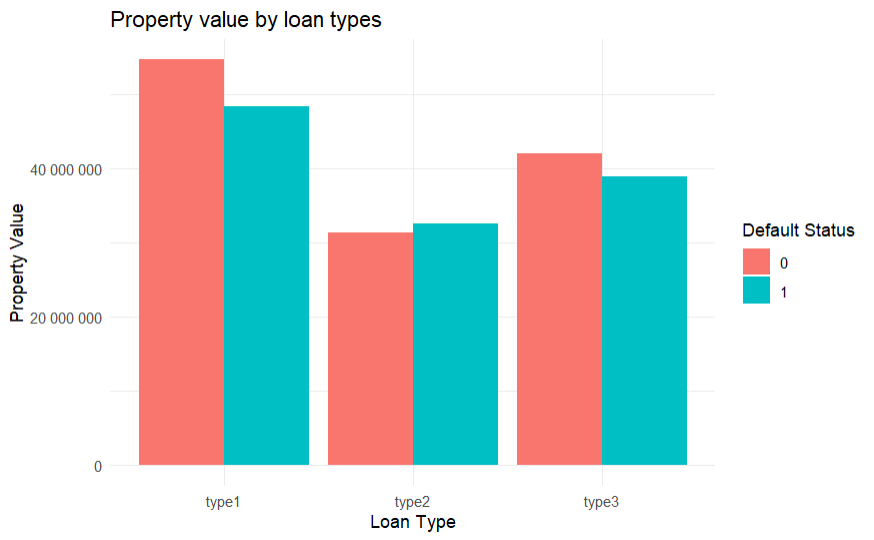
Regarding the product offerings, type1 loans which have the highest property value applicants has the least default rate, while type2 loans are mostly associated with applicants with lesser property values, and 1 in every 3 type2 loan defaults which can be observed from figure 2. The loan amounts associated with these loans follow the same trend. Type2 loan applicant pool applies for lowest amounts of loans but defaults at the highest rates.

Figure 2:Bar plot of Property value by loan types.

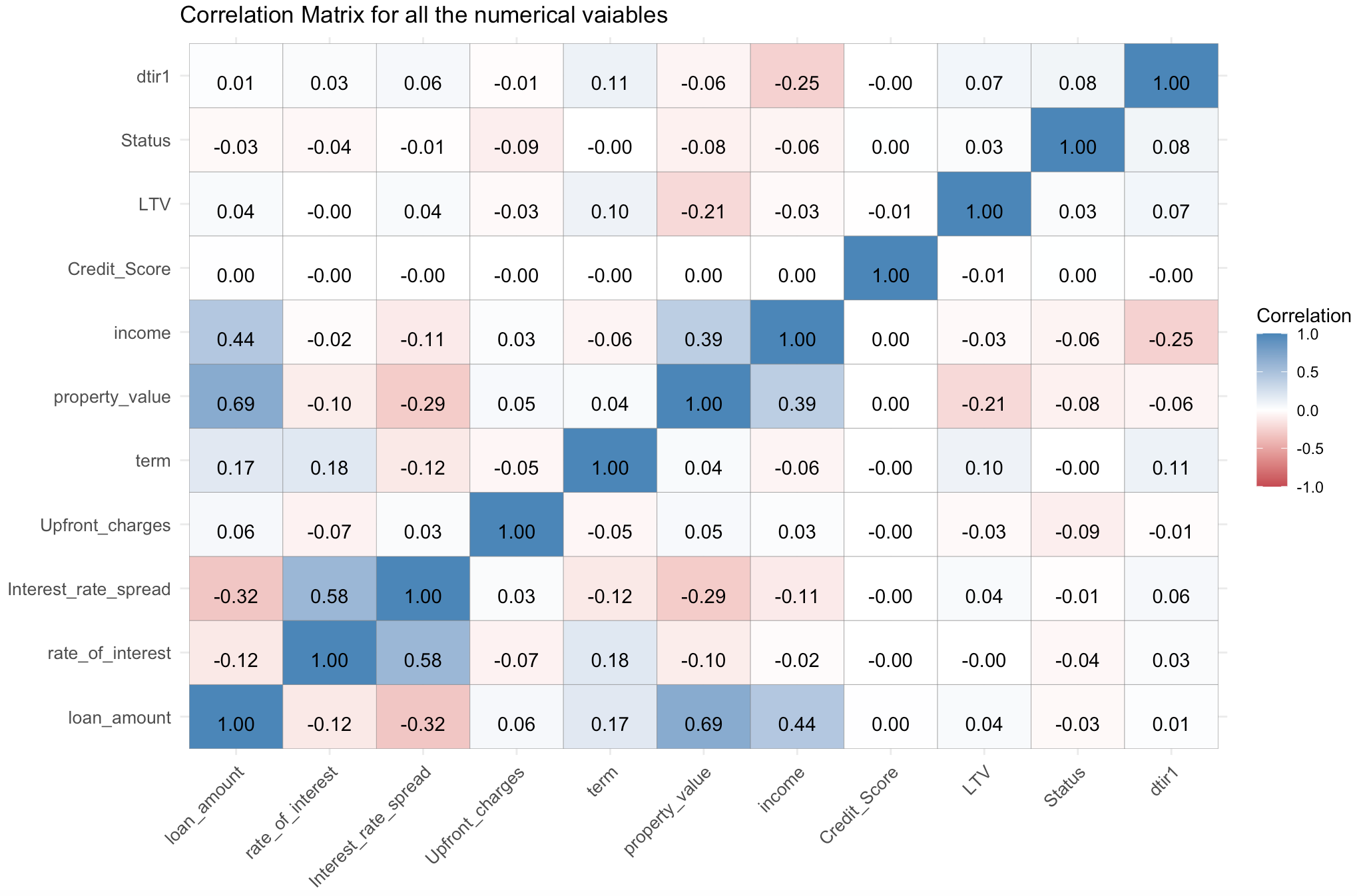
 These were some of the significant categories that can indicate a potential defaulting loan. Moving to the monetary aspects of the loan application, and their impact on loan defaults we can refer to the Correlation matrix from figure 3. We see that loan amount has stronger correlation with property value than income of the applicant. This indicates that property value is a significant driver behind the loan amount. Surprisingly, we saw no correlation between Credit score and the rest of the variables, usually we’d trust the application with high credit score, but the data didn’t show enough evidence to support the hypothesis. The mean credit score for defaulted and defaulted loans was 700.

Figure 3:Correlation matrix for all the numerical variables.

**BaseLine Model Results**

In our pursuit to better understand loan defaults, we turned to a Logistic model. This model offered insights into how different attributes correlate with the likelihood of borrowers defaulting on their loans.

Starting with the basics, our analysis reflected an accuracy rate of 78.81%. Simply put, if we look at any ten borrowers, our model can reliably predict the behaviour of nearly eight. But, to better gauge this model's efficacy, we delved deeper into individual variables.

One of the most prominent findings was the importance of the loan amount. A higher loan amount directly influenced the likelihood of default. Similarly, the rate of interest, a critical element in any loan process, showcased its influence; higher rates corresponded to a higher propensity to default.

The specifics of the loan also painted an insightful picture. Different loan types, be it 'loan\_type\_type1' or 'loan\_type\_type2', had varying associations with default risks. The purpose behind the loan, whether 'loan\_purpose\_p1', 'loan\_purpose\_p2', or 'loan\_purpose\_p3', also made a difference in default probabilities.

Borrower-specific factors were equally telling. Attributes like upfront charges, property value, and an individual's income were pivotal in shaping the default landscape. Surprisingly, even gender had a role to play; 'Gender\_Female', 'Gender\_Joint', and 'Gender\_Male' all showcased unique trends in deault patterns.

Broadening our scope, the demographics provided additional layers of insights. Age categories, for instance, 'age\_25\_34' or 'age\_35\_44', each came with their default propensities. Similarly, regional influences became evident with variations in default tendencies across 'Region\_central' and 'Region\_North'.

**Optimised Model Results**

After our initial baseline model results, we delved further to enhance our understanding and prediction capabilities. Our core objective remained: to predict loan defaults accurately, thereby aiding financial institutions in their lending strategies.

Our detailed analysis of loan defaults utilised several predictive models, as depicted in Table 2. To initiate our optimised modelling phase, we began with the Decision Tree. This model, a staple in predictive analytics, presents data in a tree-like structure, making complex decision-making more tangible.

Based on the significant variables from logistic regression model, we built the subset the features into the decision tree model. In addition to different categories, the tree highlighted the influence of spread of upfront charges and rate of interest on the default status, a pattern that was hard to notice before. We were able to achieve an accuracy of almost 99%. The model was strict on the good loans and classified then a defaults for 1.6% cases.

To improve upon the strictness of Decision Tree, we employed the Random Forest model. The Random Forest, with its ensemble of decision trees, bolstered our predictive capabilities. Interestingly, our model showcased 99.8% accuracy. But, given that near-perfect models can sometimes be misleading in real-world scenarios, we delved deeper. We probed for multicollinearity, ensuring that our variables weren't merely echoing each other, and verified our model using the Variance Inflation Factor (VIF). It was reassuring to find minimal multicollinearity, emphasising the individual strength and importance of our predictors.

Additionally, we implemented a 10-fold cross-validation to ensure our model wasn't overfitting and was primed for real-world applicability. This methodology, particularly with its extensive predictors set, showcased the robustness of the Random Forest model.

Transitioning from the Random Forest, we integrated the XGBoost model. This gradient-boosting algorithm, known for its efficiency and performance, furthered our analytical depth. Feeding significant features extracted from our logistic model, the XGBoost model revealed fascinating results. With a staggering accuracy of 100%, the model reassured us of its precision. Our model could discern non-defaulted loans and potential defaults with perfection, ensuring that financial institutions can make lending decisions with higher confidence.

*Table 2: Full Model Results*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Technique Used** | **Accuracy** | **Precision** | **Recall (Sensitivity)** | **F1 score** | **Specificity** |
| Logistic regression | 0.7812 | 0.7836 | 0.98 | 0.871 | 0.173 |
| Decision Tree | 0.9898 | 1 | 0.9864 | 0.9931 | 1 |
| Random Forest | 0.99986 | 1 | 0.9998 | 0.9999 | 1 |
| XGBoost | 0.997 | 0.9911 | 1 | 0.9955 | 0.9971 |

*Table 3: Confusion matrix for Full Model Result*

|  |  |  |  |
| --- | --- | --- | --- |
| **Confusion matrix** | | | |
| **Logistic Regression** |  | **Reference** |  |
|  |  | 0 | 1 |
|  | **Prediction** 0 | 31865 | 8797 |
|  | 1 | 639 | 1841 |
| **Decision Tree** |  | **Reference** |  |
|  |  | 0 | 1 |
|  | **Prediction** 0 | 32150 | 0 |
|  | 1 | 440 | 10562 |
| **Random Forest** |  | **Reference** |  |
|  |  | 0 | 1 |
|  | **Prediction** 0 | 32519 | 0 |
|  | 1 | 6 | 10658 |
| **XGBoost** |  | **Reference** |  |
|  |  | 0 | 1 |
|  | **Prediction** 0 | 32619 | 0 |
|  | 1 | 0 | 10523 |

The tools and techniques employed in our optimised phase didn't just enhance our predictive capabilities; they provided profound insights. We discerned the strong interplay between credit scores, employment consistency, and loan amounts in predicting defaults. As we progressed, our questions evolved. Initially, we were set on merely predicting defaults, but as the analysis deepened, our focus shifted towards understanding the underlying factors, ensuring our model's real-world readiness, and the nuances of each predictive model.

From the tables 2 & 3, XG boost model performed perfectly, predicting accurately the defaulting and non-defaulting applicants without any misclassification. Random Forest and Decision Tree models was a bit of a strict model that misclassified good loans as defaulting for 0.1% and 1.6% cases. Decision tree model also did well to prevent misclassifying bad loans as good ones, therefore preventing loss making decisions. The logistic model however did a fairly poor job on preventing false negatives and better at false positives.

It is important to minimise false positives (incorrectly predicting a loan as defaulted when it's not) and false negatives (missing actual loan defaults). It is important to strike balance between the two cases since one results in missed opportunities while the other increases the chances of a loss.

**Conclusion**

In our exhaustive exploration of loan default patterns, we have discovered crucial insights that can profoundly impact the lending strategies of financial institutions. Among these, the significance of geographical variations, the intricacies of credit types, and the surprising revelations about age demographics and loan categories stood out.

It's evident that certain regions, like the North-east, require specialized attention due to higher default rates. This can be mitigated by refining our loan scrutiny processes in these areas or through educational outreach programs. Similarly, the high default rates associated with the EQUI credit type and specific loan categories warrant a re-evaluation of our lending criteria and practices.

Surprisingly, our analysis revealed that credit scores, a traditionally valued metric, didn't play as pivotal a role as anticipated. This underscores the need for financial institutions to consider a broader range of factors, such as property value correlation with loan amounts and the implications of negative amortizations. For younger and older borrowers, and those seeking type2 loans, a more tailored approach or additional checks might be warranted.

Furthermore, the powerful predictive capabilities of Decision Tree, Random Forest and XGBoost models advocate for their integration into real-time loan assessment processes. However, in an industry as dynamic as finance, it's essential that our models undergo regular refinements to align with current data and trends.

With this newfound knowledge, we recommend a heightened focus on applicants' credit histories and the introduction of flexible repayment plans tailored to the unique circumstances of individual borrowers. Financial education programs can further empower potential borrowers, instilling a sense of responsibility and awareness about loan commitments. Advanced analytics and machine learning can be harnessed to refine our predictive abilities, ensuring more informed and timely lending decisions.

In the light of these findings, our future strategy should prioritize the continual reassessment of our lending criteria, aligning with both our insights and the ever-shifting economic landscape. By amalgamating these insights with our earlier findings on income stability, credit history, and financial commitments, we can sculpt a more resilient and adaptive lending paradigm. This approach doesn't just minimize our risk, but also champions a culture of informed, responsible borrowing, safeguarding both the financial health of our institution and our valued clientele.

**Appendix A: Preliminary Analysis & Model Code**

**####Credit type default rate visual code########**

ggplot(Loan\_Default, aes(x = credit\_type, y = Status,fill=credit\_type)) +

geom\_bar(stat = "summary", fun = "mean", position = "dodge") +

# Customize the plot labels and title

labs(

x = "Credit Type",

y = "% Default",

title = "Default rate by Credit Type"

) +

scale\_y\_continuous(labels = scales::percent\_format(scale = 100)) +

# Adjust the appearance of the plot

theme\_minimal()

**####### Average Property amount by Loan Type visual code #############**

ggplot(Loan\_Default, aes(x = loan\_type, y = property\_value,fill=factor(Status))) +

geom\_bar(stat = "summary", fun = "mean", position = "dodge") +

# Customize the plot labels and title

labs(

x = "Loan Type",

y = "Property Value",

title = "Property value by Loan Type"

) +

scale\_y\_continuous(labels = scales::number\_format(scale = 1)) +

# Adjust the appearance of the plot

theme\_minimal()

**######Correlation Matrix####################**

# Identify non-numeric columns

non\_numeric\_columns <- names(Loan\_Default)[sapply(Loan\_Default, function(col) !is.numeric(col))]

# Print the non-numeric columns

print(non\_numeric\_columns)

# Exclude non-numeric columns and calculate the correlation matrix

cor\_matrix <- cor(Loan\_Default[, sapply(Loan\_Default, is.numeric)], use = "complete.obs", method = "pearson")

# Print the correlation matrix

print(cor\_matrix)

cor\_data <- melt(cor\_matrix)

# Heatmap with numerical values using the Spectral palette

ggplot(data = cor\_data, aes(x=Var1, y=Var2)) +

geom\_tile(aes(fill=value), color="grey50") +

geom\_text(aes(label=sprintf("%.2f", value)), vjust=1) +

scale\_fill\_gradient2(low=brewer.pal(n=7, name="Spectral")[1],

high=brewer.pal(n=7, name="Spectral")[7],

midpoint=0,

limit=c(-1,1),

space="Lab",

name="Correlation") +

theme\_minimal() +

theme(axis.text.x = element\_text(angle=45, vjust=1, hjust=1, size=10),

axis.text.y = element\_text(size=10)) +

labs(title="Correlation Matrix for all the numerical vaiables", x="", y="")

**###########################Model 1###############################**

set.seed(123)

split\_ratio <- 0.7 # 70% training, 30% testing

sample\_size <- floor(split\_ratio \* nrow(data\_encoded))

train\_indices <- sample(seq\_len(nrow(data\_encoded)), size = sample\_size)

# Create training and testing datasets

train\_data <- data\_encoded[train\_indices, ]

test\_data <- data\_encoded[-train\_indices, ]

# Create the logistic regression model

logistic\_model <- glm(Status~., data = train\_data, family = binomial)

# Summarise the model

summary(logistic\_model)

#prediction logistic model

log\_pred <- predict(logistic\_model, test\_data[, !names(train\_data) %in% "Status"])

accuracy\_log <- mean((log\_pred > 0.5) == (test\_data$Status == 1))

print(paste("Accuracy\_log:", accuracy\_log))

#Binary conversion of logistic output

log\_predicted\_factor <- as.factor(ifelse(log\_pred > 0.5, 1, 0))

test\_data$Status <- as.factor(test\_data$Status)

#confusion matrix

confusion\_matrix\_log <- confusionMatrix(log\_predicted\_factor, test\_data$Status)

confusion\_matrix\_log

#test prediction

log\_pred\_2 <- predict(logistic\_model, test\_data[, !names(train\_data) %in% "Status"])

accuracy\_log\_2 <- mean((log\_pred\_2 > 0.5) == (test\_data$Status == 1))

print(paste("Accuracy\_log2:", accuracy\_log\_2))

log\_predicted\_factor\_2 <- as.factor(ifelse(log\_pred\_2 > 0.5, 1, 0))

test\_data$Status <- as.factor(test\_data$Status)

confusion\_matrix\_log\_2 <- confusionMatrix(log\_predicted\_factor\_2, test\_data$Status)

confusion\_matrix\_log\_2

#roc curve

roc\_obj\_log2 <- roc(as.numeric(test\_data$Status), as.numeric(log\_pred\_2))

# Plot the ROC curve

plot(roc\_obj\_log2, main = "ROC Curve Logistic Regression", col = "blue")

# Add AUC (Area Under the Curve) to the plot

auc2 <- auc(roc\_obj\_log2)

legend("bottomright", legend = paste("AUC =", round(auc2, 4)), col = "blue", lty = 1)

p\_values <- summary(logistic\_model)$coefficients[, "Pr(>|z|)"]

# Filter columns with p-values < 0.05

significant\_features <- names(p\_values[p\_values < 0.05])

significant\_features<- c(significant\_features[-1],"Status")

# Print the significant features

cat("Significant Features (p-value < 0.05):\n")

cat(significant\_features, sep = ", ")

# Create a new model with significant features

filtered\_logistic\_model <- glm(Status ~ ., data = train\_data%>%select(significant\_features), family = binomial)

# Summarise the filtered model

summary(filtered\_logistic\_model)

**################Model 2#########################**

#Decision tree

# Encode categorical variables as factors

Loan\_Default$loan\_limit <- as.factor(Loan\_Default$loan\_limit)

Loan\_Default$Gender <- as.factor(Loan\_Default$Gender)

Loan\_Default$loan\_type <- as.factor(Loan\_Default$loan\_type)

Loan\_Default$Region <- as.factor(Loan\_Default$Region)

# Encode the target variable ('Status') as a binary factor (0 for default, 1 for non-default)

Loan\_Default$Status <- as.factor(1 - Loan\_Default$Status)

# Split the Data into Training and Testing Sets

set.seed(123)

library(caret)

trainIndex <- createDataPartition(Loan\_Default$Status, p = 0.8, list = FALSE)

train\_data <- Loan\_Default[trainIndex, ]

test\_data <- Loan\_Default[-trainIndex, ]

# Build the Decision Tree Model

model <- rpart(Status ~ loan\_amount + income + Credit\_Score + LTV + loan\_limit + Gender + loan\_type + Region, data = train\_data, method = "class")

# Make predictions on the test data

predictions <- predict(model, test\_data, type = "class")

# Calculate accuracy

accuracy <- confusionMatrix(predictions, test\_data$Status)

accuracy

# Tune the Model (Optional)

# Prune the tree to improve performance

pruned\_model <- prune(model, cp = 0.01)

# Visualize the Decision Tree

rpart.plot(pruned\_model)

**################Model 3#########################**

# Random Forest model

set.seed(123) # Setting seed for reproducibility

# Splitting dataset into training and test sets (70-30 split)

sample\_index <- sample(seq\_len(nrow(Loan\_Default)), size = 0.7 \* nrow(Loan\_Default))

train\_data <- Loan\_Default[sample\_index, ]

test\_data <- Loan\_Default[-sample\_index, ]

train\_data$`co-applicant\_credit\_type` <- as.factor(train\_data$`co-applicant\_credit\_type`)

test\_data$`co-applicant\_credit\_type` <- as.factor(test\_data$`co-applicant\_credit\_type`)

names(train\_data)[names(train\_data) == "co-applicant\_credit\_type"] <- "co\_applicant\_type"

names(test\_data)[names(test\_data) == "co-applicant\_credit\_type"] <- "co\_applicant\_type"

train\_data$Status <- as.factor(train\_data$Status)

test\_data$Status <- as.factor(test\_data$Status)

# train the model

rf\_model <- randomForest(Status ~ ., data=train\_data, ntree=10, importance=TRUE)

train\_predictions <- predict(rf\_model, newdata=train\_data)

test\_predictions <- predict(rf\_model, newdata=test\_data)

confusion\_matrix\_train <- table(train\_predictions, train\_data$Status)

confusion\_matrix\_test <- table(test\_predictions, test\_data$Status)

accuracy\_train <- sum(diag(confusion\_matrix\_train)) / sum(confusion\_matrix\_train)

accuracy\_test <- sum(diag(confusion\_matrix\_test)) / sum(confusion\_matrix\_test)

precision\_train <- confusion\_matrix\_train[2,2] / (confusion\_matrix\_train[2,2] + confusion\_matrix\_train[1,2])

precision\_test <- confusion\_matrix\_test[2,2] / (confusion\_matrix\_test[2,2] + confusion\_matrix\_test[1,2])

specificity\_train <- confusion\_matrix\_train[1,1] / (confusion\_matrix\_train[1,1] + confusion\_matrix\_train[1,2])

specificity\_test <- confusion\_matrix\_test[1,1] / (confusion\_matrix\_test[1,1] + confusion\_matrix\_test[1,2])

recall\_train <- confusion\_matrix\_train[2,2] / (confusion\_matrix\_train[2,2] + confusion\_matrix\_train[2,1])

recall\_test <- confusion\_matrix\_test[2,2] / (confusion\_matrix\_test[2,2] + confusion\_matrix\_test[2,1])

confusion\_matrix\_train

confusion\_matrix\_test

cat("Train Metrics:\n")

cat(paste("Accuracy: ", round(accuracy\_train, 2), "\n"))

cat(paste("Precision: ", round(precision\_train, 2), "\n"))

cat(paste("Specificity: ", round(specificity\_train, 2), "\n"))

cat(paste("Recall: ", round(recall\_train, 2), "\n\n"))

cat("Test Metrics:\n")

cat(paste("Accuracy: ", round(accuracy\_test, 2), "\n"))

cat(paste("Precision: ", round(precision\_test, 2), "\n"))

cat(paste("Specificity: ", round(specificity\_test, 2), "\n"))

cat(paste("Recall: ", round(recall\_test, 2), "\n"))

# Calculate F1 score for the test set

f1\_score\_test <- 2 \* (precision\_test \* recall\_test) / (precision\_test + recall\_test)

cat(paste("F1 Score (Test): ", round(f1\_score\_test, 2), "\n"))

# Multicollinearity test

cv\_results <- train(Status ~ ., data=train\_data, method="rf", ntree=10, trControl=trainControl(method="cv", number=5))

print(cv\_results$results)

numeric\_columns <- sapply(train\_data, is.numeric)

correlations <- cor(train\_data[, numeric\_columns & names(train\_data) != "Status"], use="complete.obs")

high\_corr <- which(abs(correlations) > 0.75, arr.ind=TRUE)

print(high\_corr)

#VIF quantifies how much the variance is increased due to multicollinearity.

#A VIF of 1 indicates no correlation, while a VIF greater than 5-10 suggests high multicollinearity.

# Fitting a linear model on one of the numeric predictors as a function of all others

numeric\_predictors <- names(train\_data)[numeric\_columns & names(train\_data) != "Status"]

vif\_model <- lm(as.formula(paste(numeric\_predictors[1], "~", paste(numeric\_predictors[-1], collapse="+"))), data=train\_data)

# Calculate VIF values

vif\_values <- vif(vif\_model)

print(vif\_values)

# Compute correlation matrix for numeric columns

cor\_matrix <- cor(train\_data[, numeric\_columns & names(train\_data) != "Status"], use="complete.obs")

# Identify the upper triangle of the correlation matrix

upper\_triangle <- cor\_matrix[upper.tri(cor\_matrix)]

# Extract pairs with absolute correlation greater than a threshold (e.g., 0.75)

high\_corr\_pairs <- which(abs(upper\_triangle) > 0.75, arr.ind = TRUE)

# Check if high\_corr\_pairs is not empty

if(length(high\_corr\_pairs) > 0) {

for(i in 1:nrow(high\_corr\_pairs)) {

cat(paste0(names(cor\_matrix)[high\_corr\_pairs[i, 1]], " and ",

names(cor\_matrix)[high\_corr\_pairs[i, 2]],

" have a correlation of ",

round(cor\_matrix[high\_corr\_pairs[i, 1], high\_corr\_pairs[i, 2]], 2), "\n"))

}

} else {

cat("No pairs have correlation greater than 0.75\n")

}

# K- fold test

# Define training control

train\_control <- trainControl(method = "cv", number = 5) # 10-fold CV

# Train the model using cross-validation

cv\_results <- train(Status ~ .,

data = train\_data,

method = "rf",

trControl = train\_control)

# Print the results

print(cv\_results$results)

**###############################Model 4##################################**

#XGB Model

xgb\_train<-train\_data%>%select(significant\_features)

xgb\_test<-test\_data%>%select(significant\_features)

xgb\_model <- xgboost(data = as.matrix(xgb\_train[, !names(xgb\_train) %in% "Status"]),

label = xgb\_train$Status,

objective = "binary:logistic",nrounds=4

)

xgb\_predictions <- predict(xgb\_model, as.matrix(xgb\_test[, !names(xgb\_test) %in% "Status"]))

accuracy <- mean((xgb\_predictions > 0.5) == (xgb\_test$Status == 1))

print(paste("Accuracy:", accuracy))

xgb\_predicted\_factor <- as.factor(ifelse(xgb\_predictions > 0.5, 1, 0))

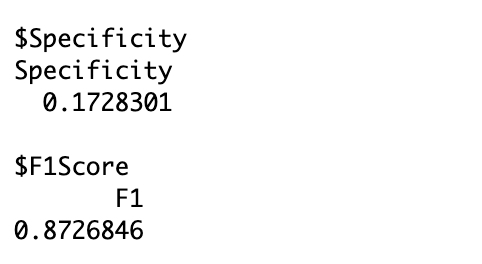
xgb\_test$Status <- as.factor(test\_data$Status)

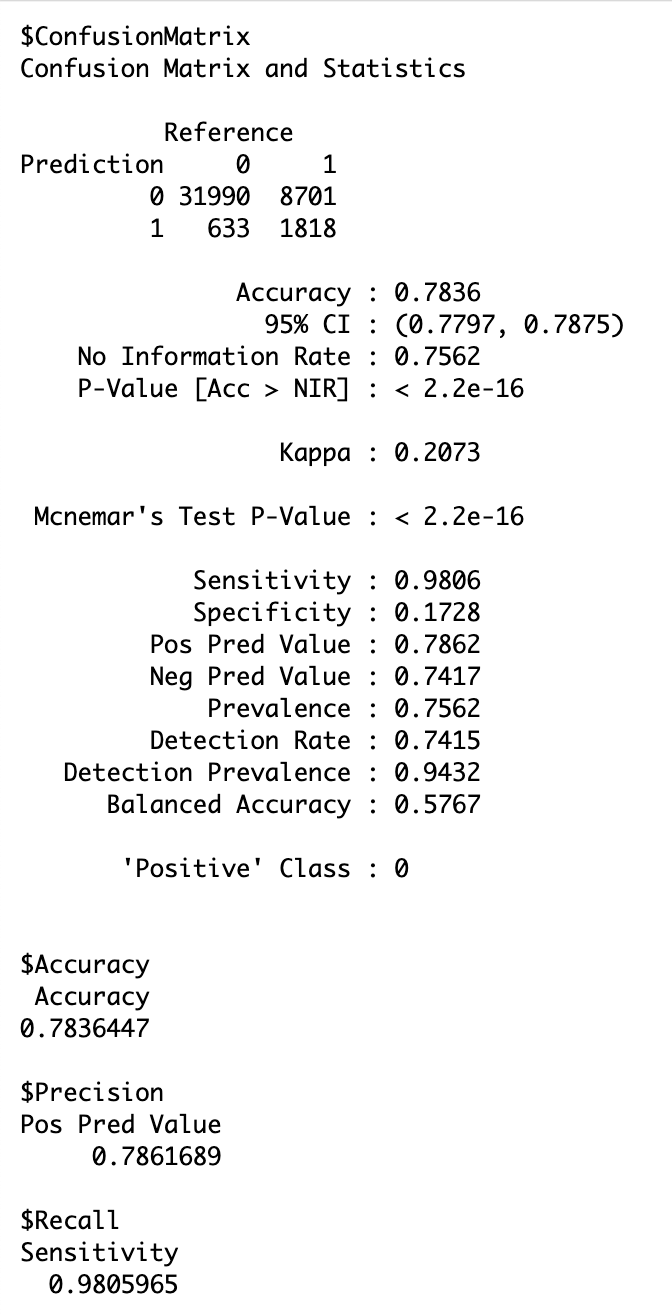
confusion\_matrix <- confusionMatrix(xgb\_predicted\_factor, xgb\_test$Status)

confusion\_matrix

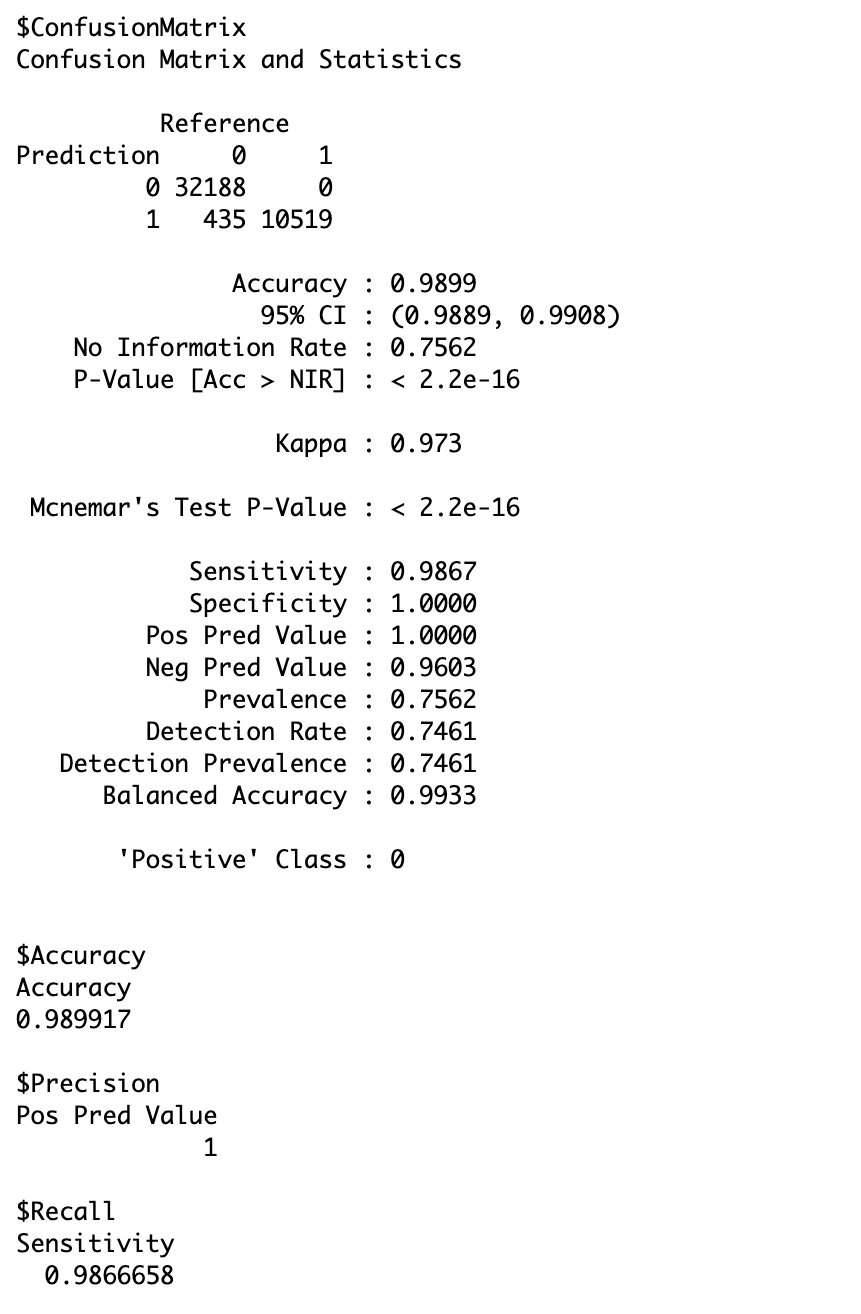
**Appendix B: Full Model Results & Accuracy Statistics**

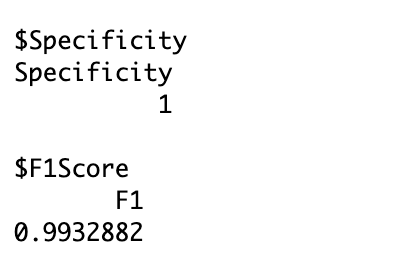
**##########Model 1(Logistic Regression)########**

****

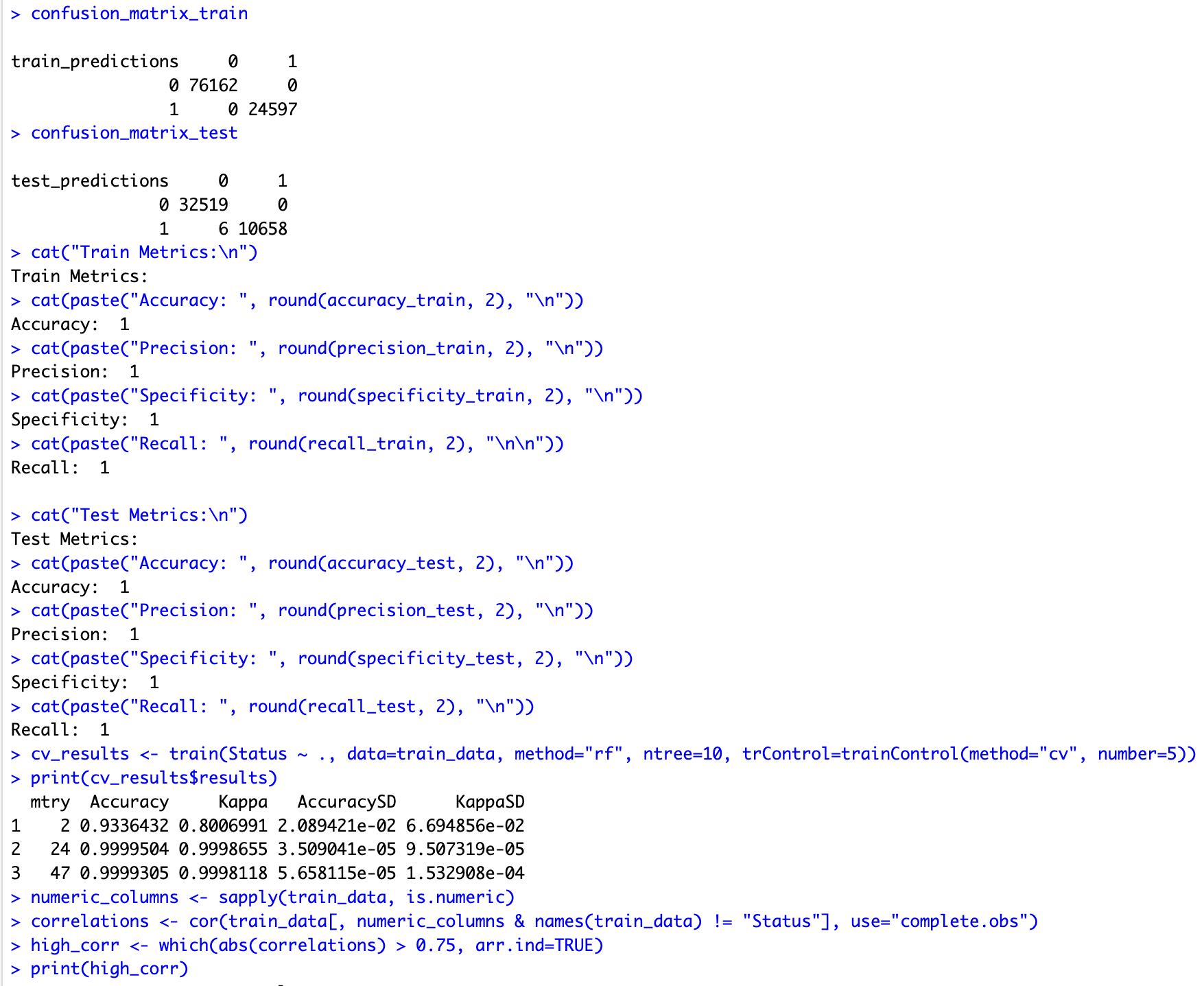


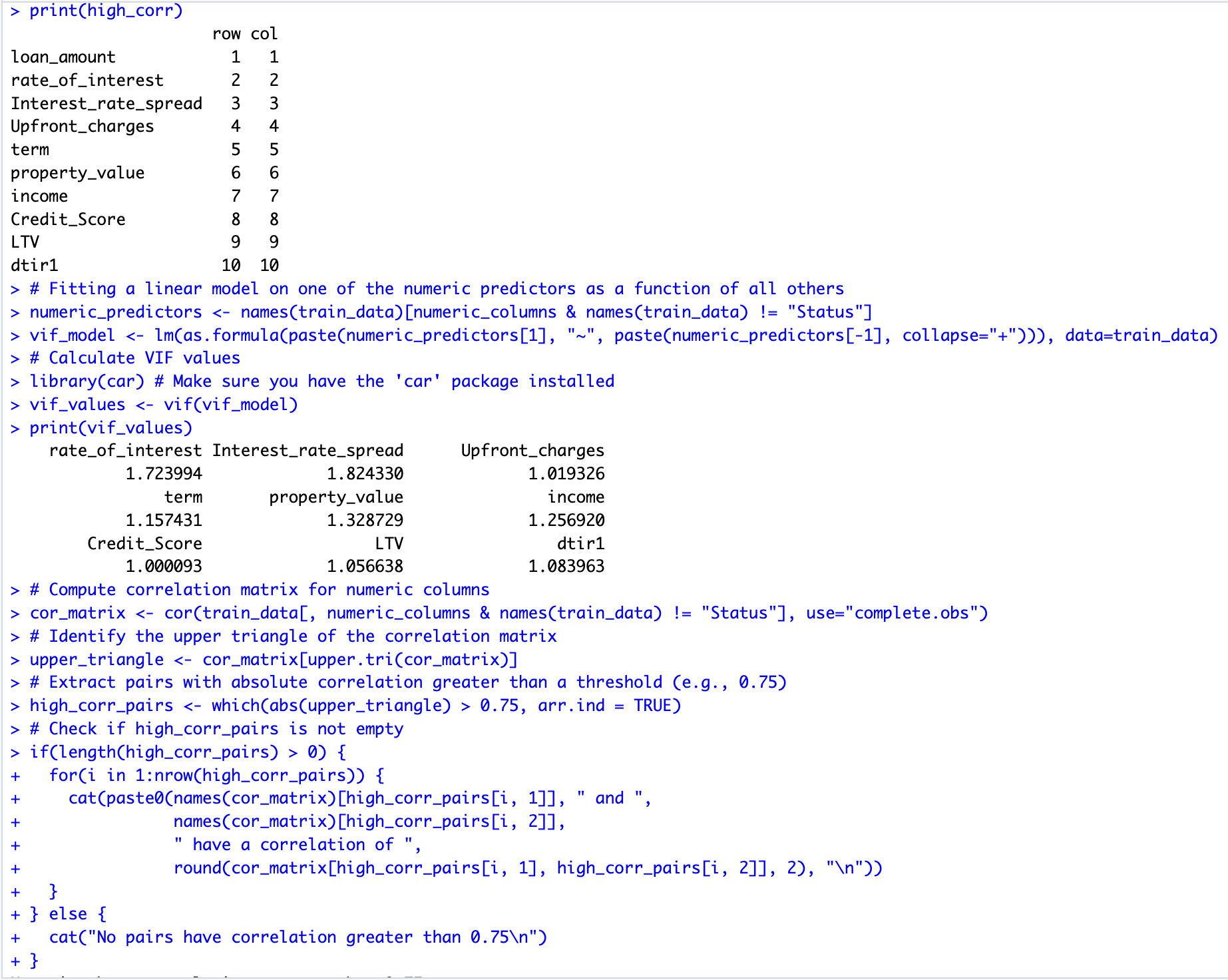
**##########Model 2(Decision Tree)###########**

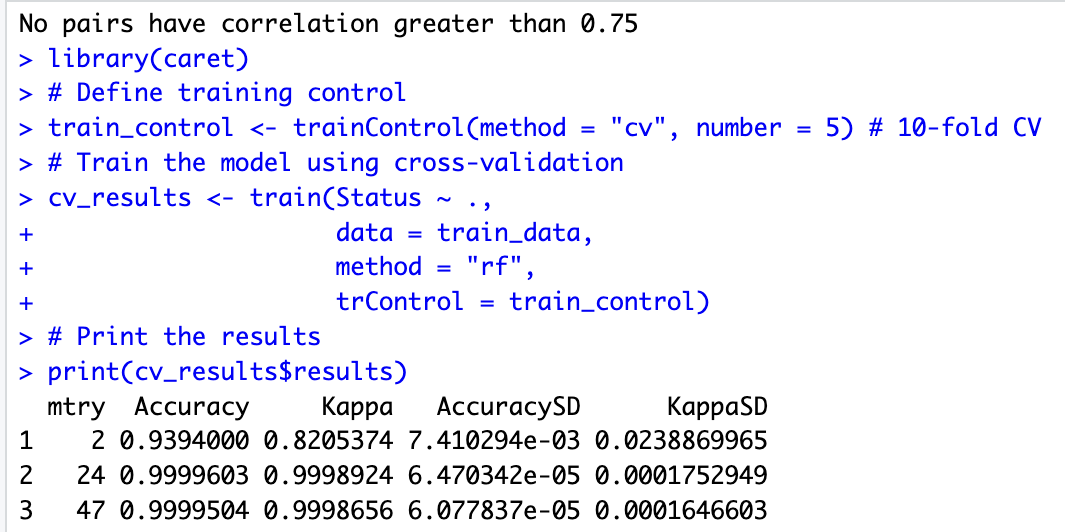


****

**##########Model 3(Random Forest)###########**

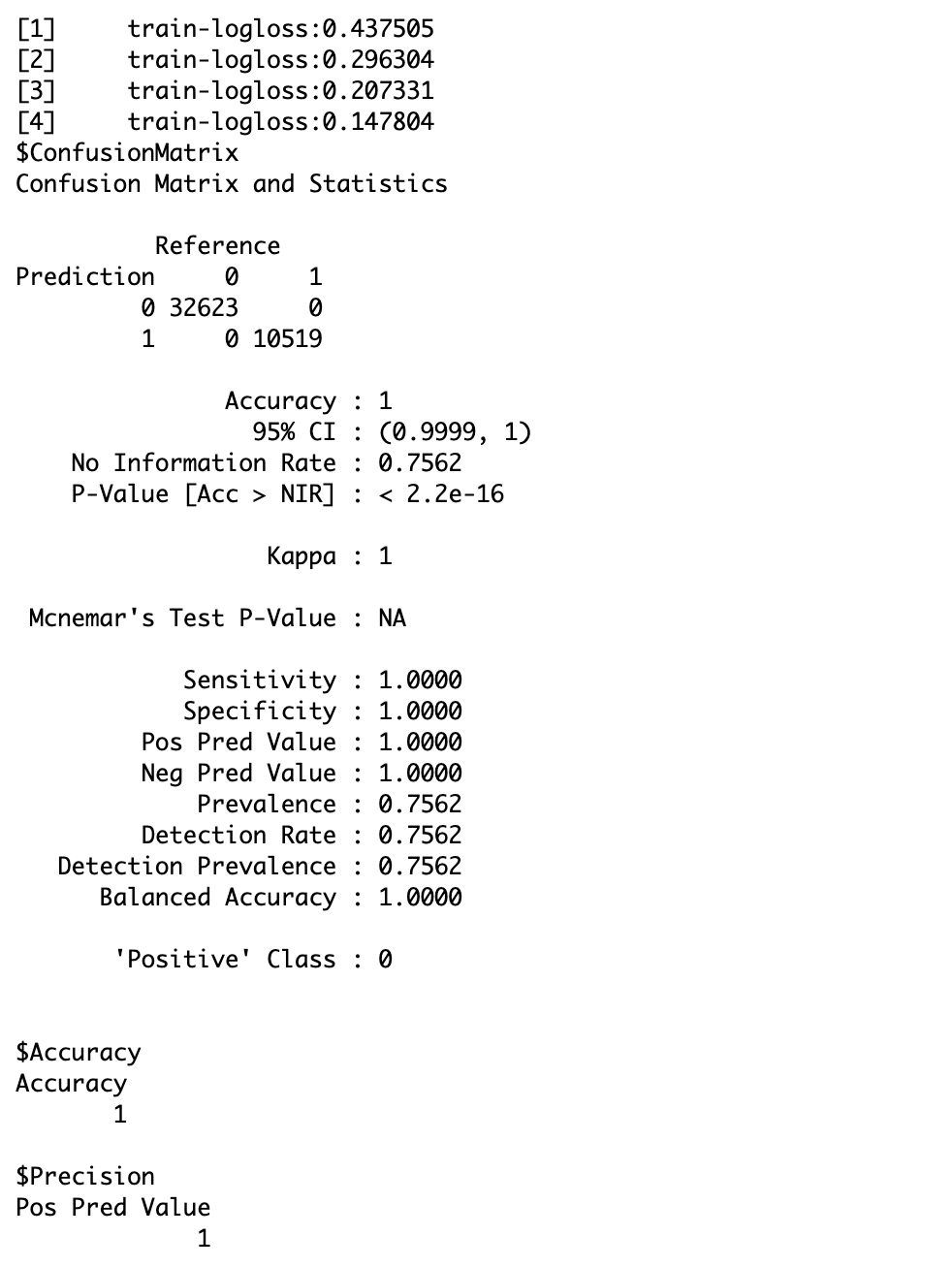
****

****

****

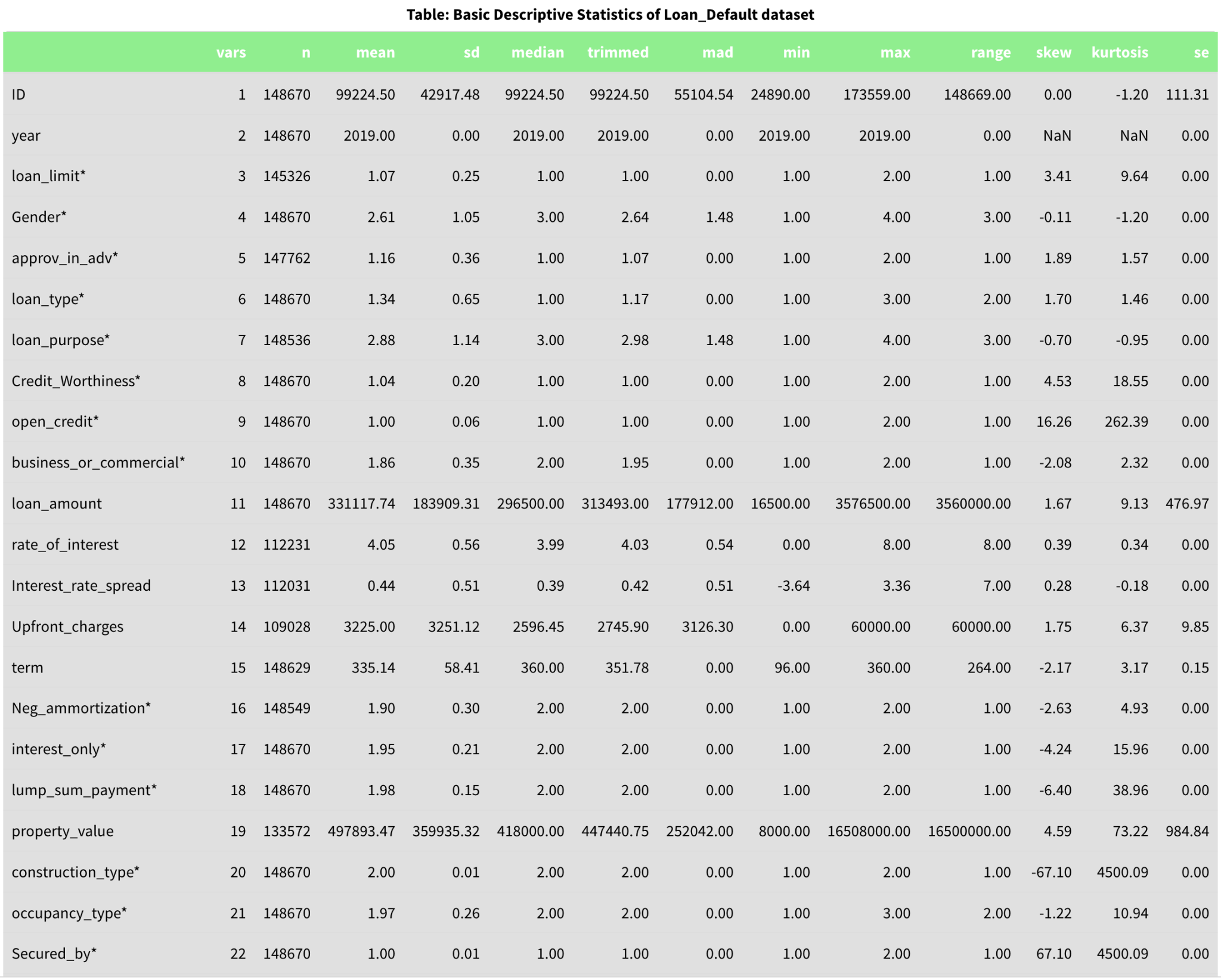
**##################Model 4(XGBoost)##############**

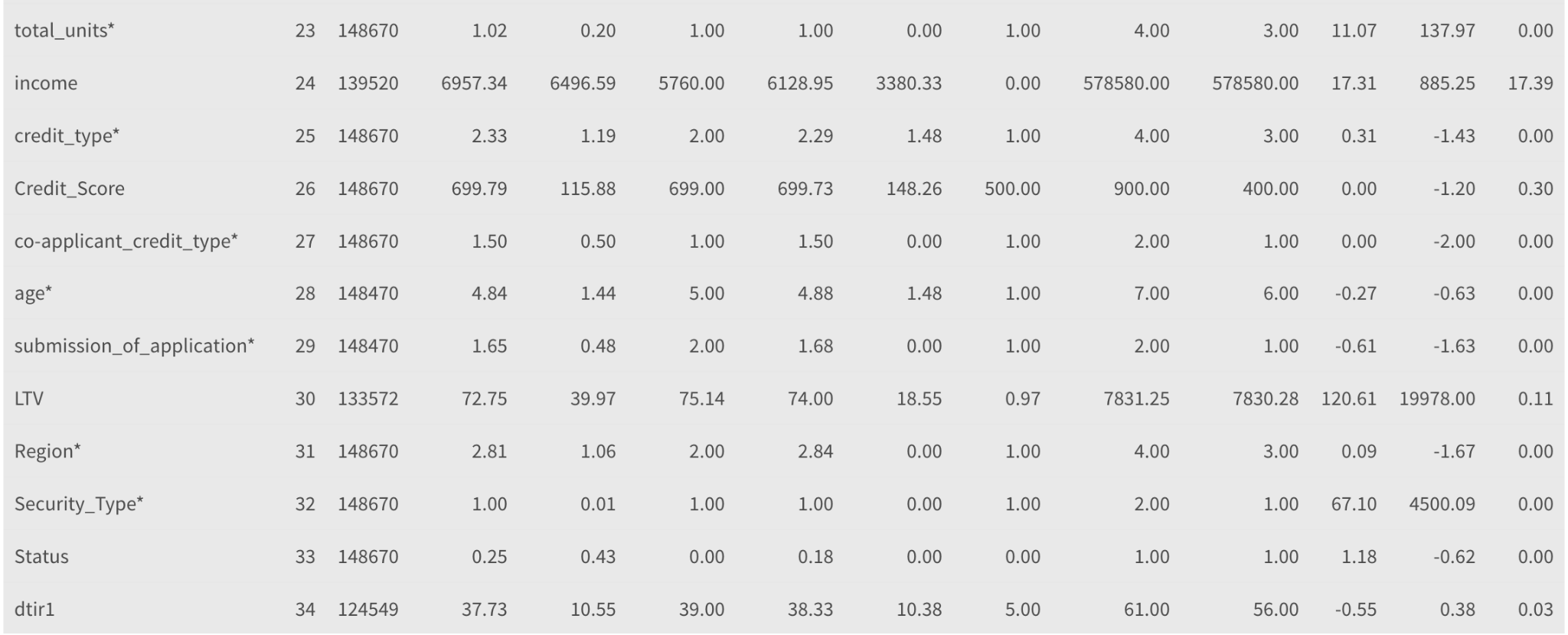
****



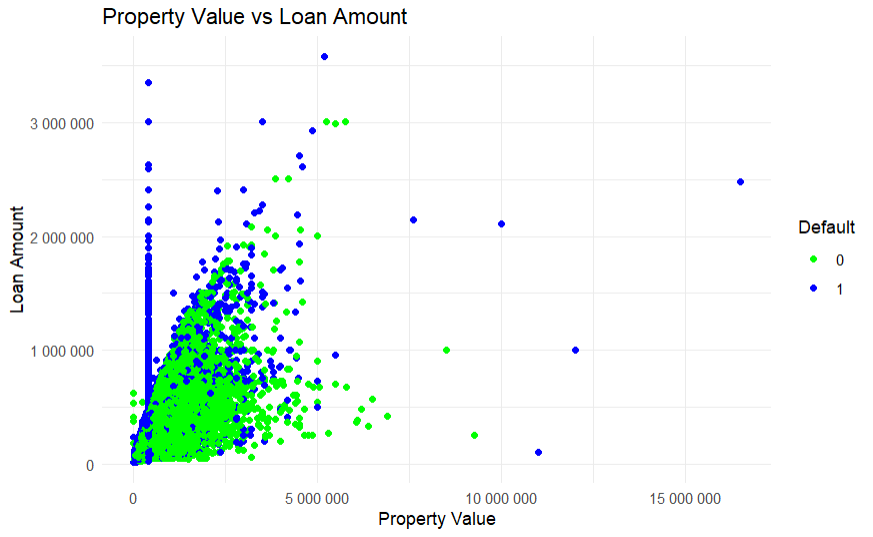
**Appendix C: Visualisation**

***Figure 1: Basic descriptive statistics of loan default dataset***

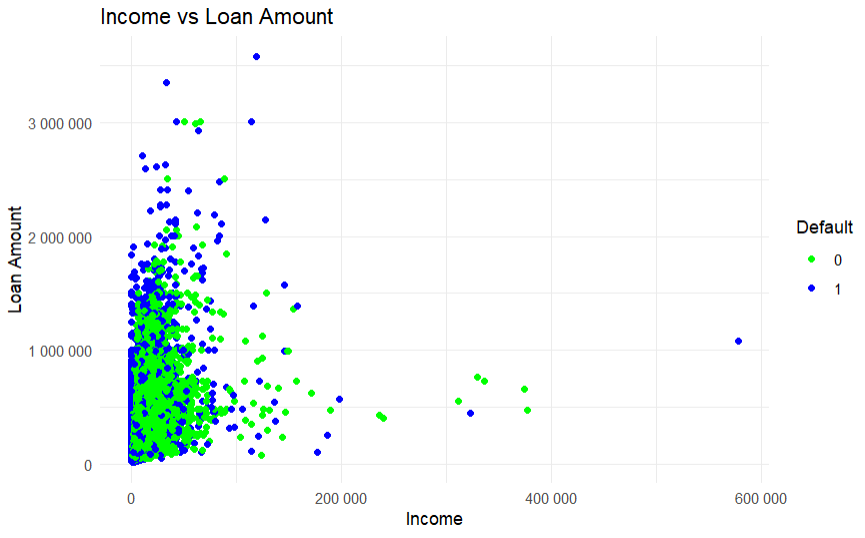
****

****

***Figure 2: Scatter plot of Property value vs Loan Amount***

****

***Figure 3: Scatter plot of Income vs Loan Amount***

****