PREDICTIVE ANALYSIS ON LENDING CLUB LOAN DATA

STA 545 – Statistical Data Mining - 1

Prepared By:

Aditya Pradeep Patankar

University at Buffalo The State University of New York

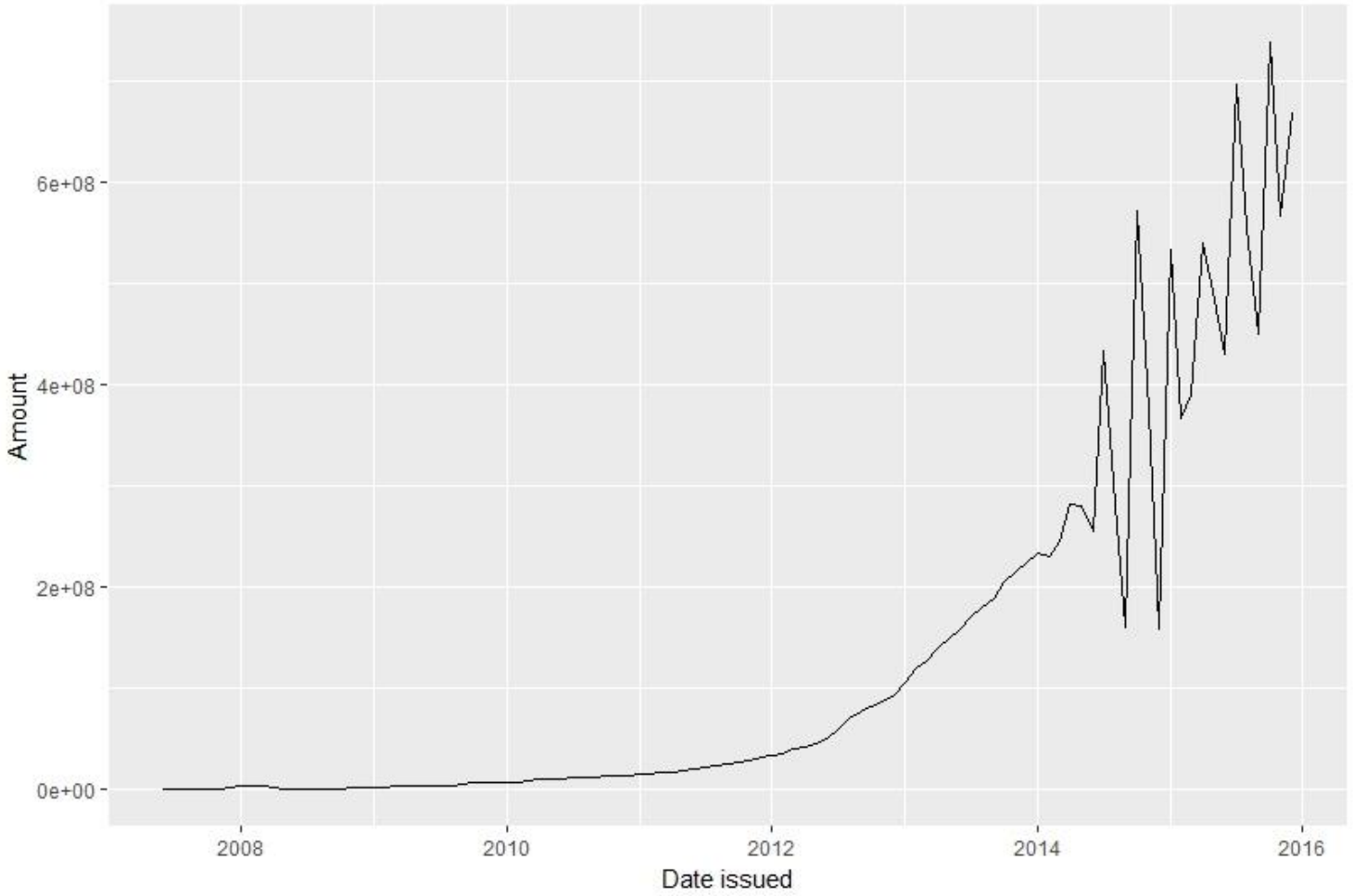
Agenda

1. Introduction
2. Understanding the Dataset
3. Data Cleaning
4. Exploratory Data Analysis
5. Data Modelling
6. Results & Conclusion

Introduction

- **Dataset – Lending club loan data**
- **Lending Club is a US peer-to-peer lending company, headquartered in San Francisco, California founded in 2006.**
- **The motivation of this project is to predict the loan status of loans issued over the span of 8 years (2007 – 2015) based on a variety of variables and parameters**
- **We will be studying 6 machine learning algorithms, namely – Logistic Regression, Linear Discriminant Analysis (LDA), Decision Trees, Random Forest, Support Vector Machine (SVM) and Generalized Boosting method (GBM), followed by comparison**
- **Model will be selected based on accuracy of prediction of the loan status (Default or Fully Paid)**

Loan Amount Growth by Years



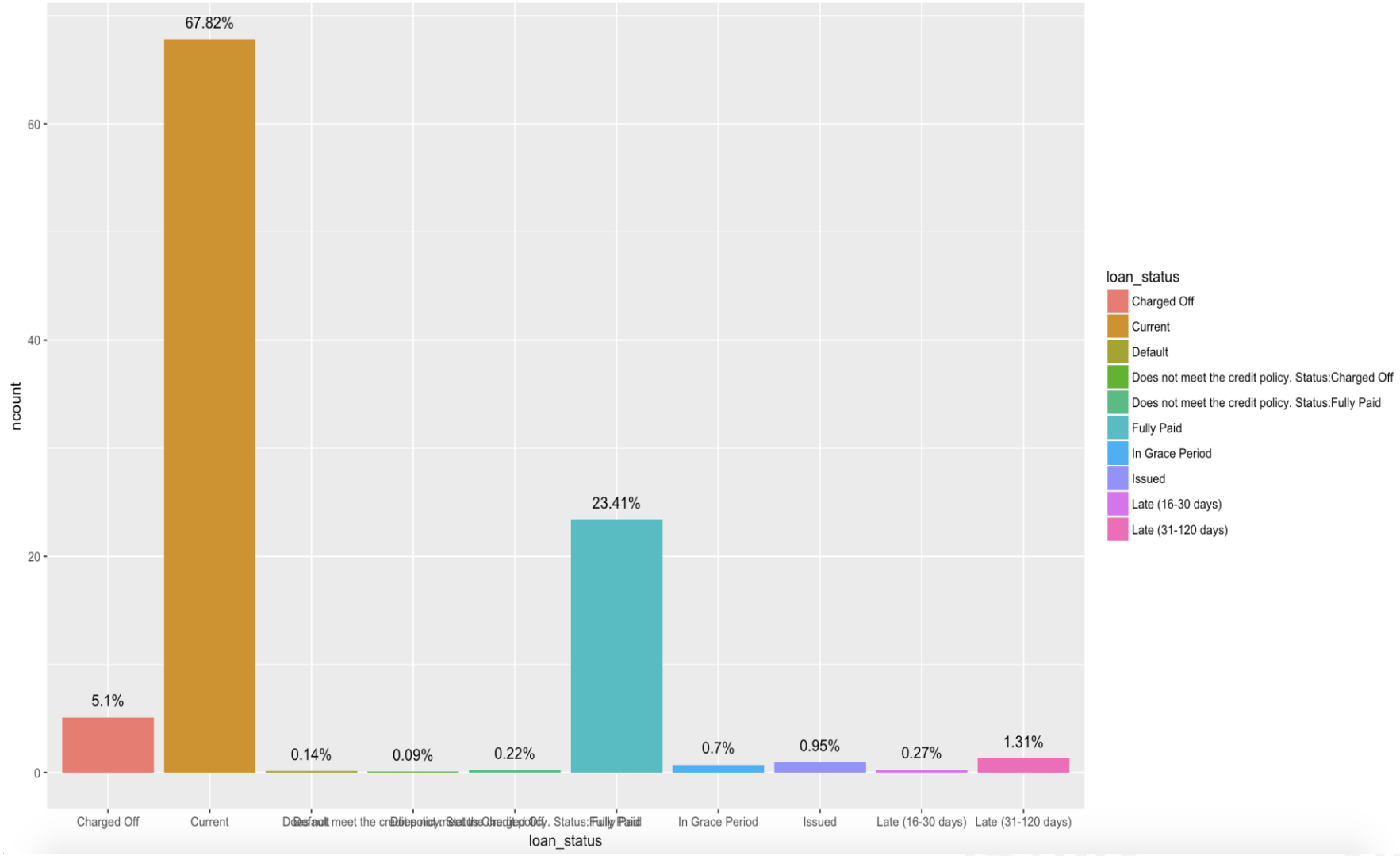
Understanding the Dataset

- **Dataset – Lending club loan data (Source: https://www.kaggle.com/wendykan/lending-club-loan-data)**
- **The files contain all data for loans issued over a span of 8 years (2007 – 2015) including loan status, latest payment information, etc.**
- **Size of the dataset – 890k observations and 74 variables such as credit scores, location of the applicant, number of active credit lines, DTI ratio etc.**

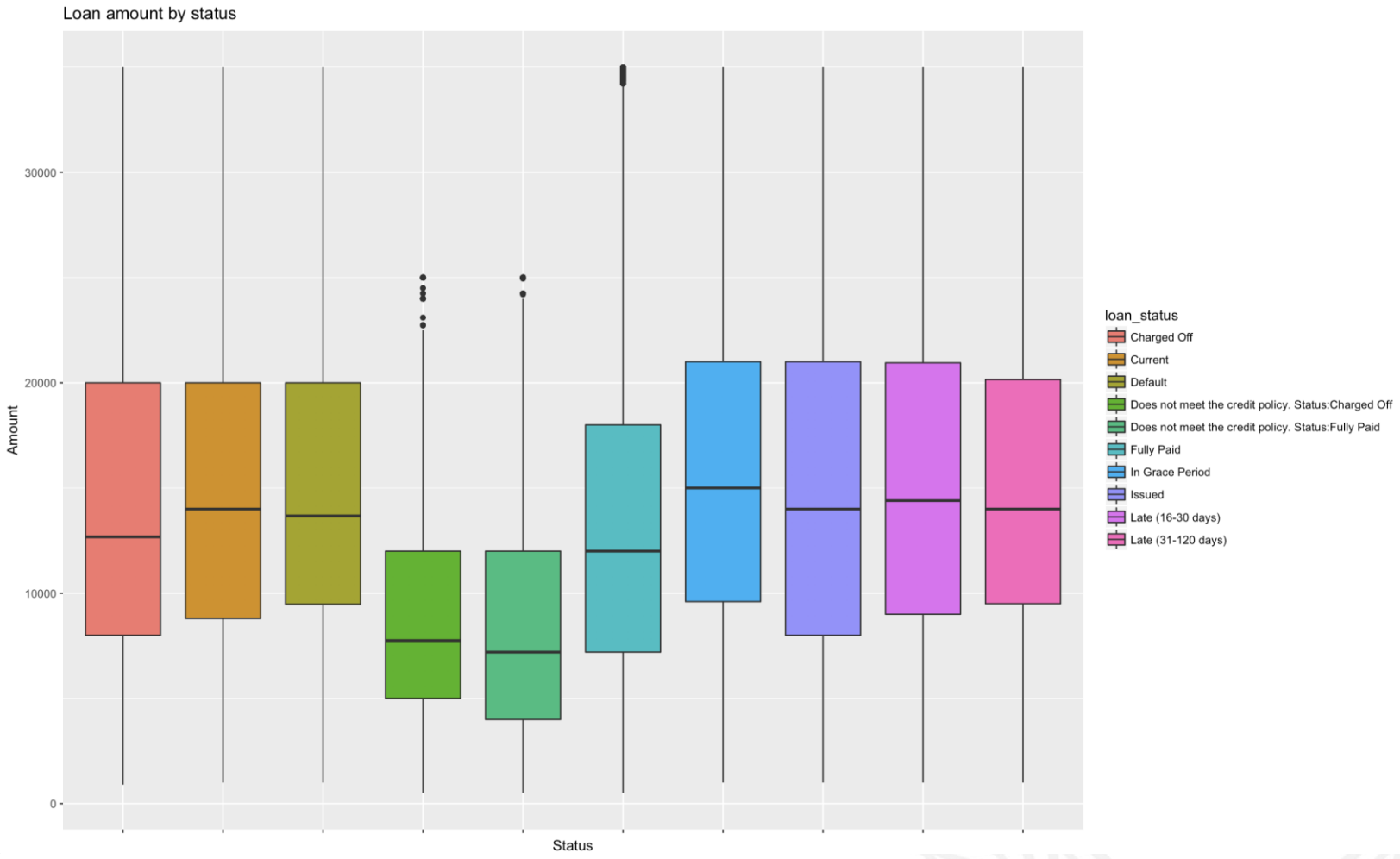
Problem Statement

Predicting the loan status (Default or Fully Paid) using various algorithms and arriving at the best model which has the lowest error rate

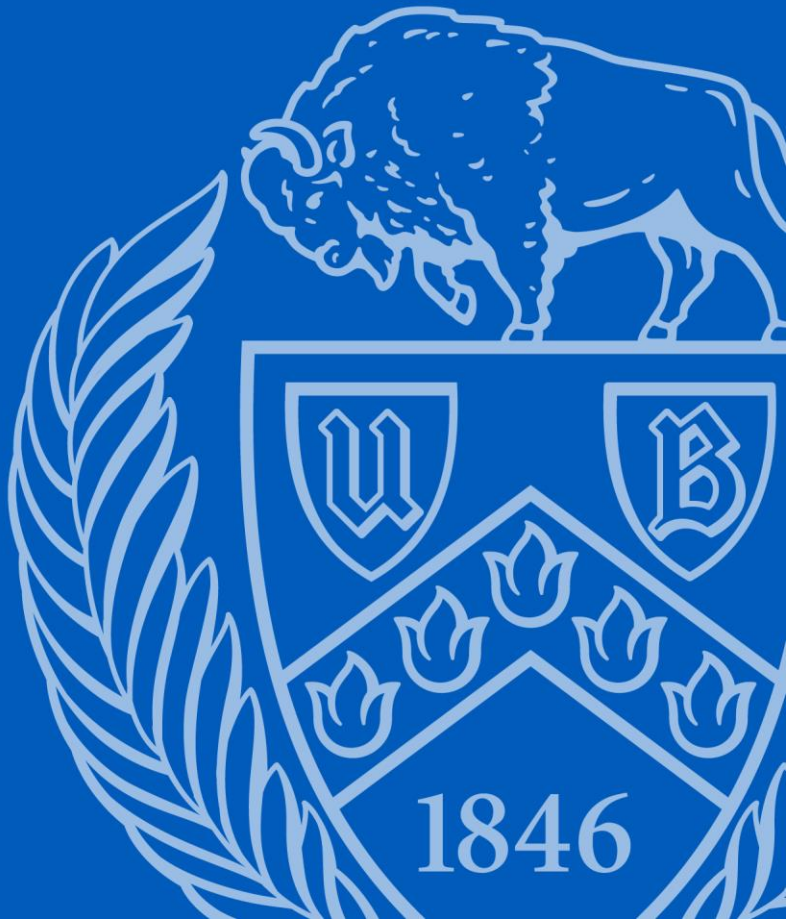
Volume of Loans by Status before Grouping



Boxplots of Amount by Status



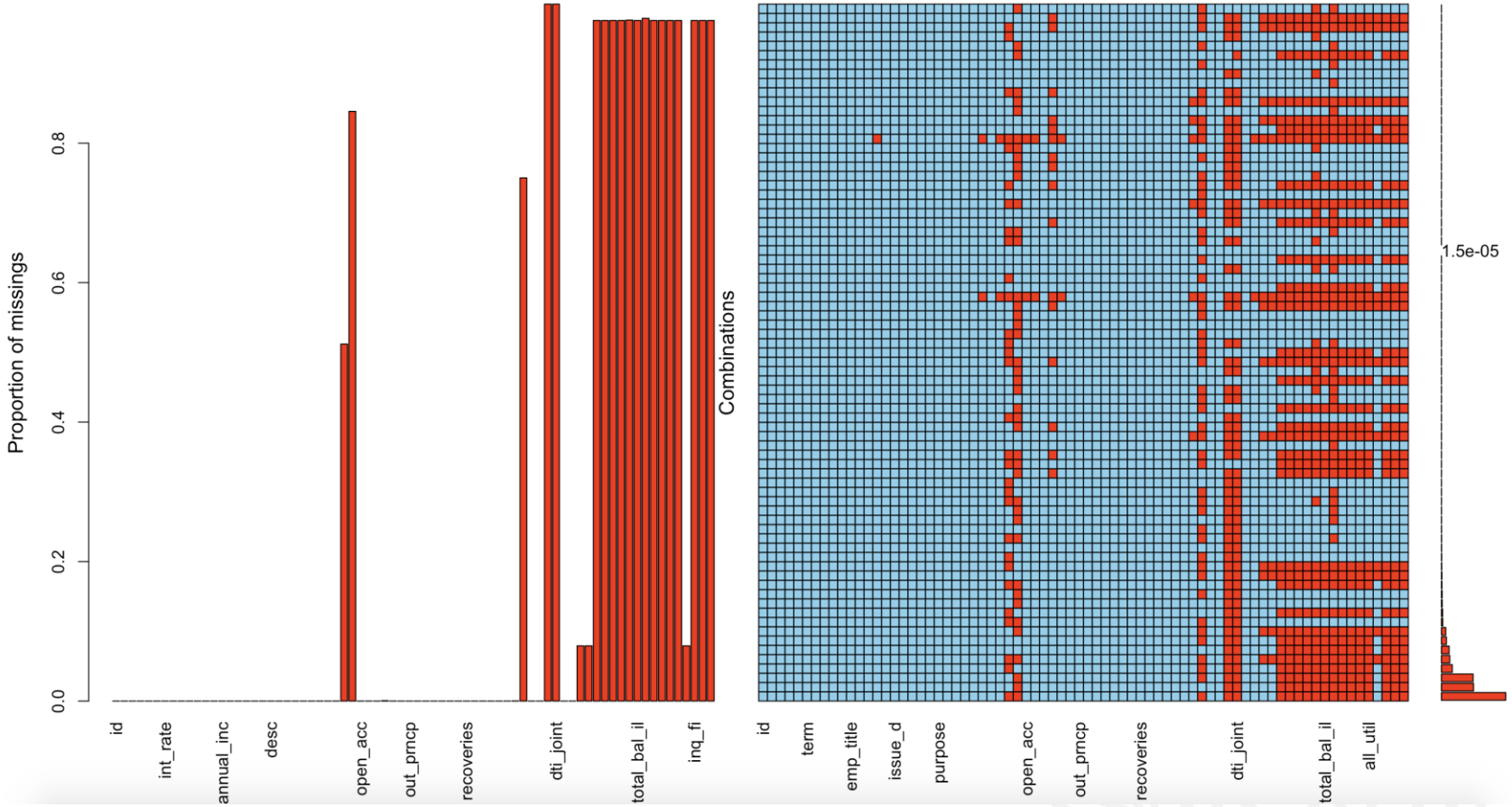
DATA CLEANING



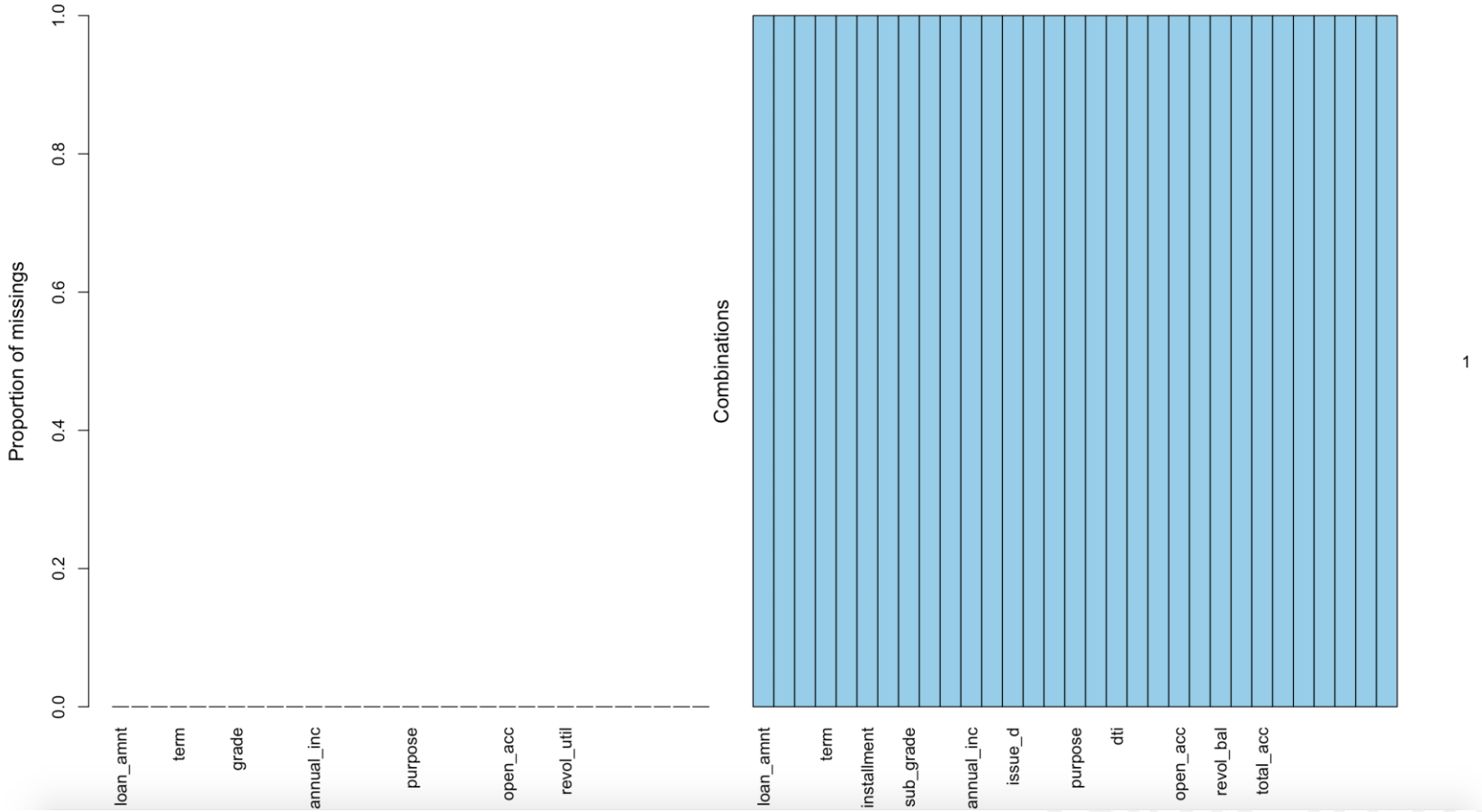
Data Cleaning Procedure

- **Column Reduction: Removal of Columns with >100K Missing Values**
- **Removing Insignificant/Unimportant Columns by Intuition**
- **Data Imputation: Converting all the NA values in Numeric Columns to the mean Values**

Dataset Before Cleaning



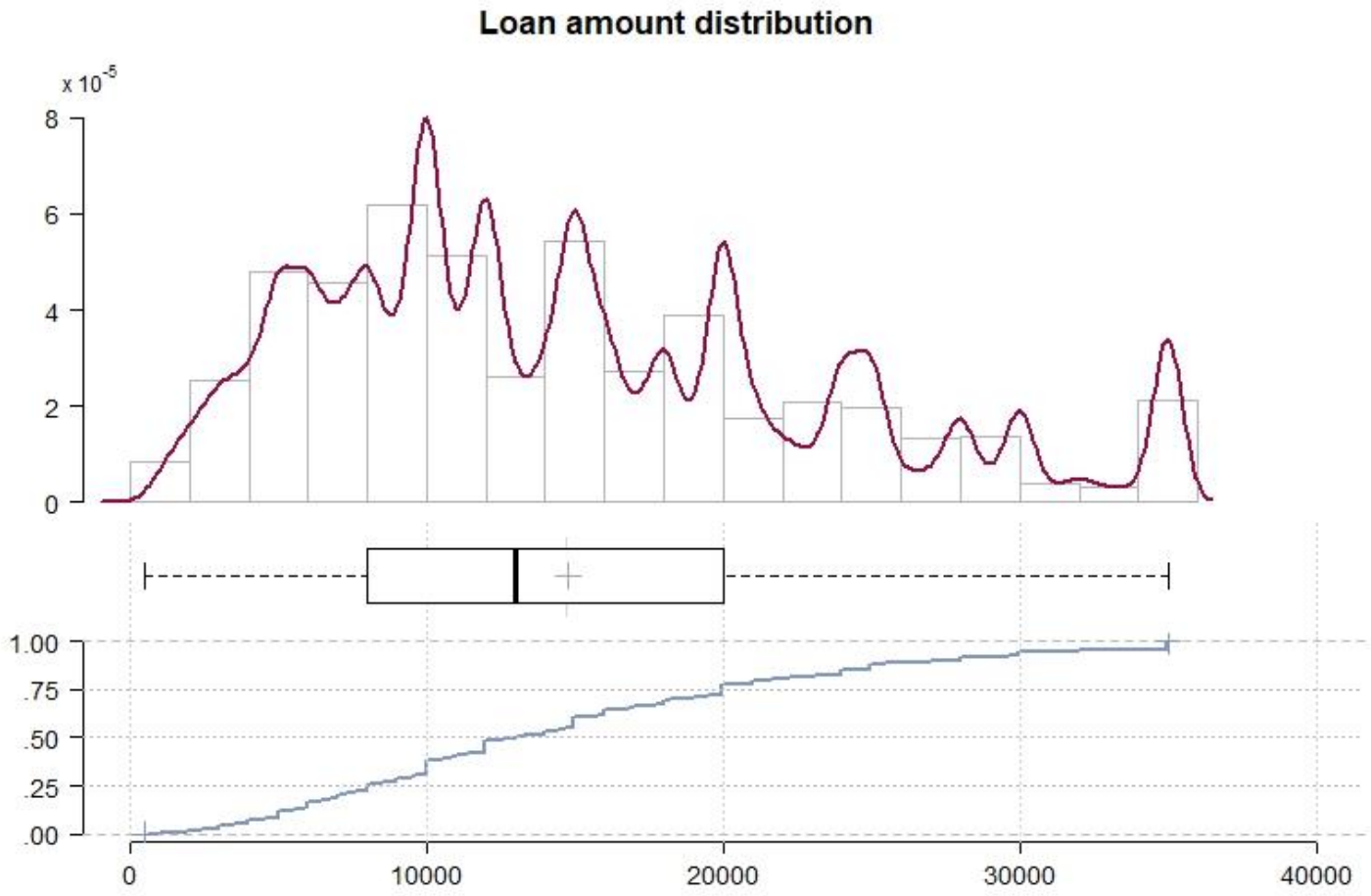
Dataset After Cleaning



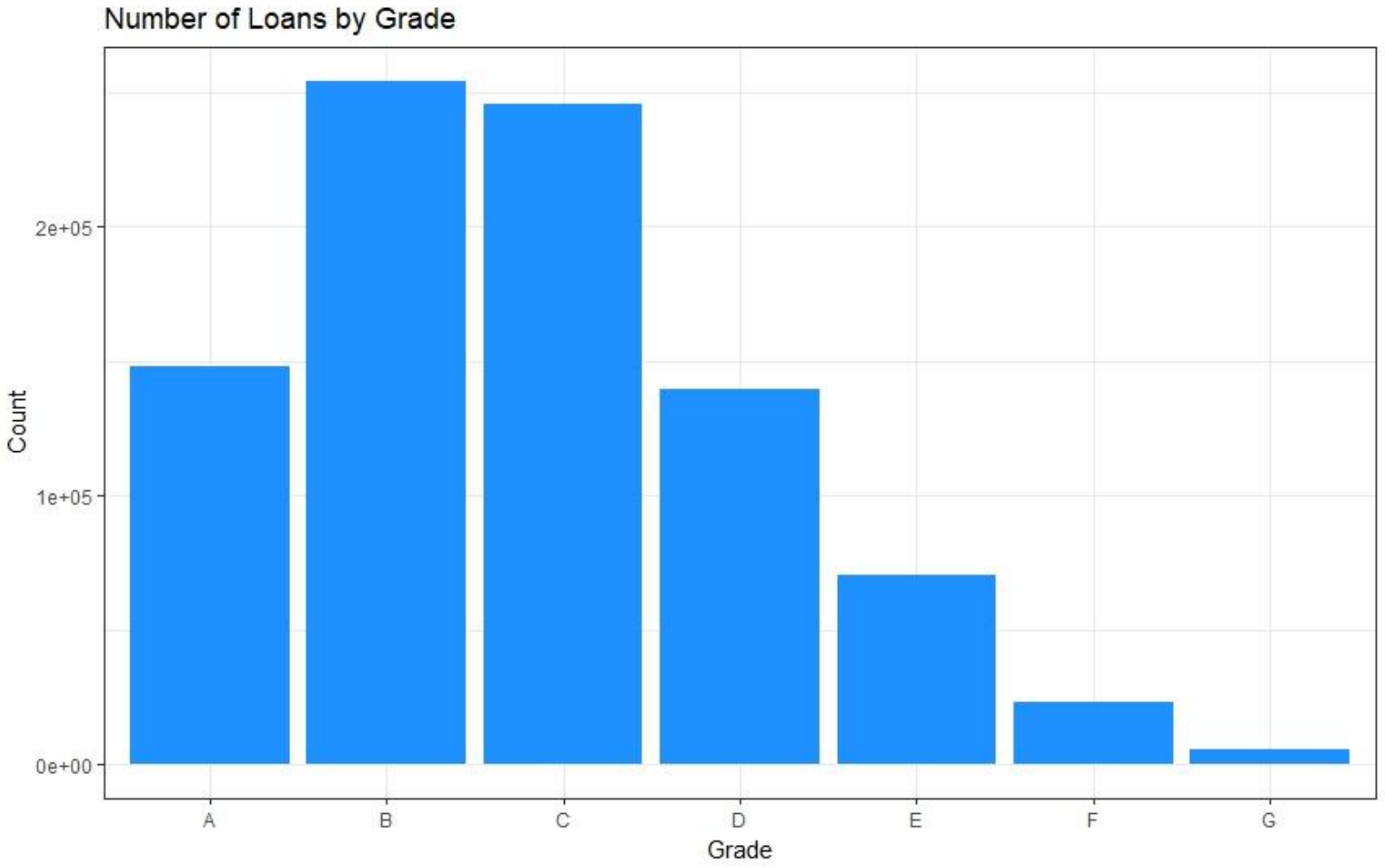
University at Buffalo The State University of New York

EXPLORATORY DATA ANALYSIS

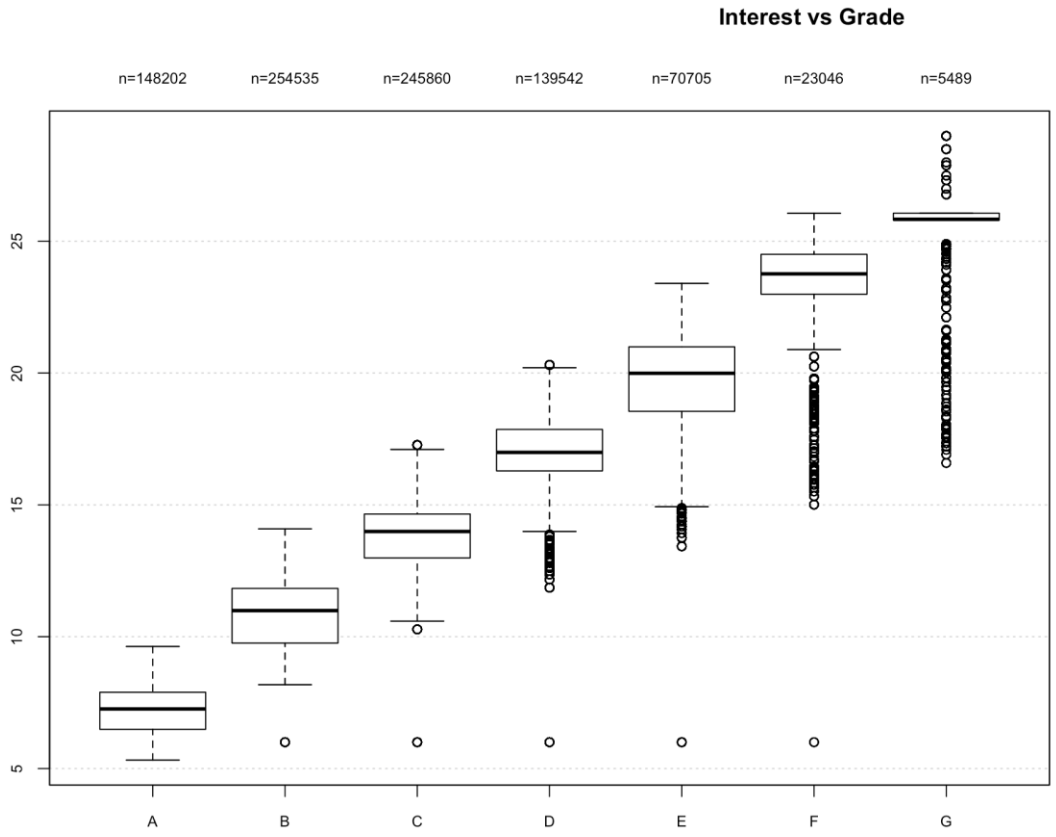
Loan Amount Distribution

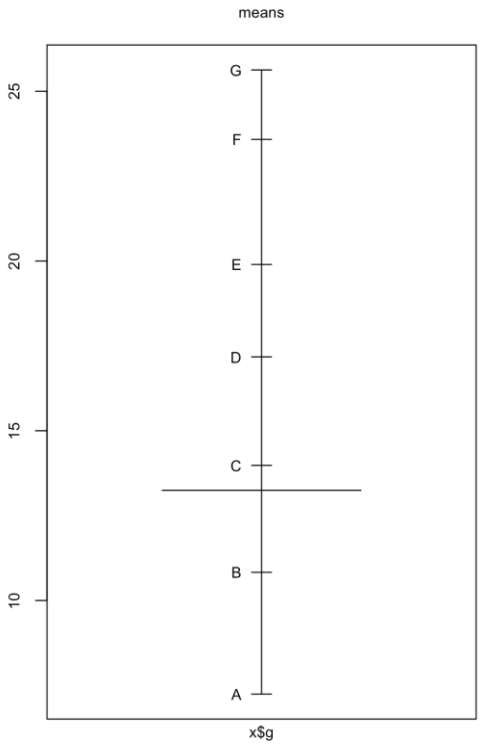


Loan Volume by Grade

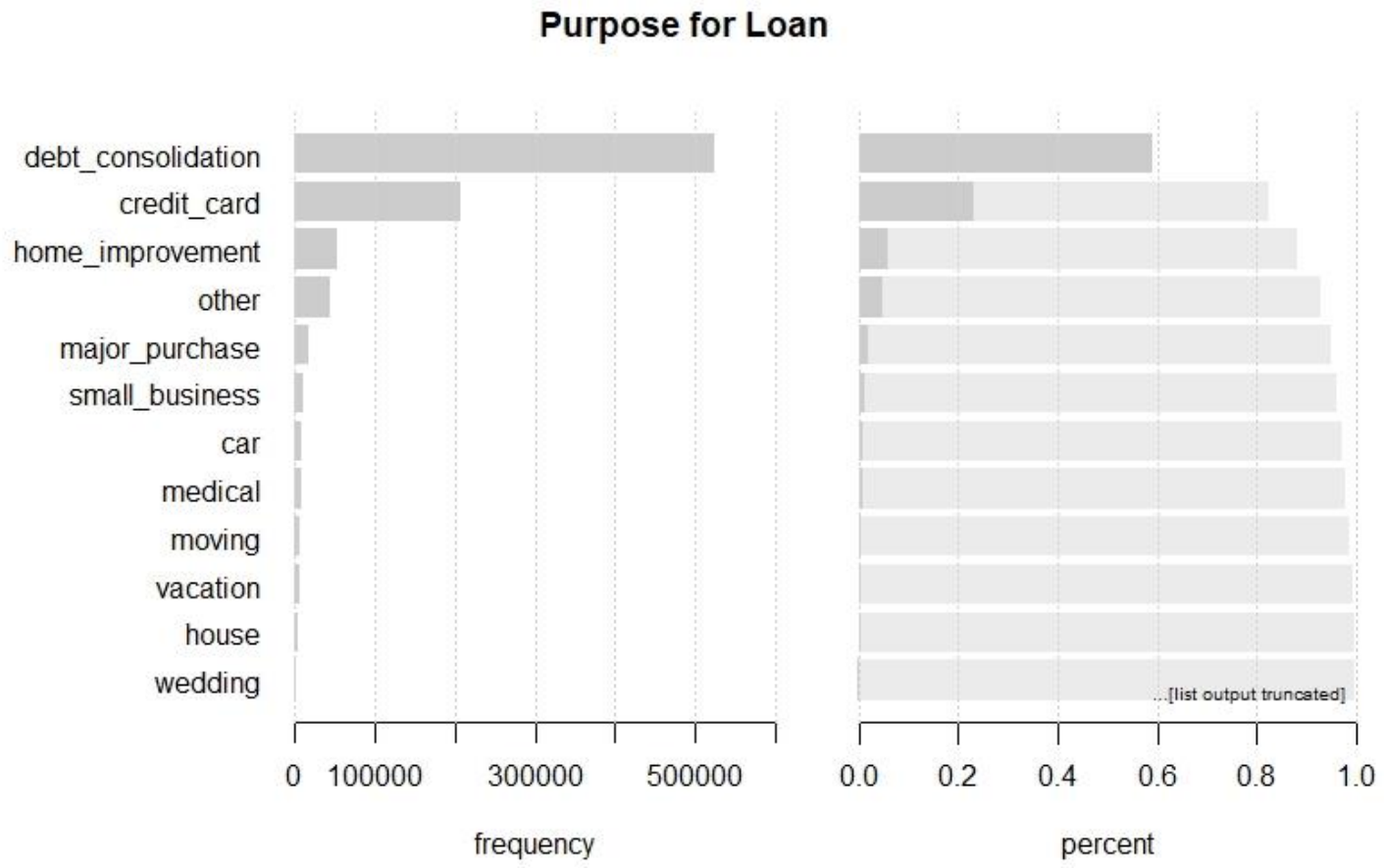


Relation of Interest vs Grade

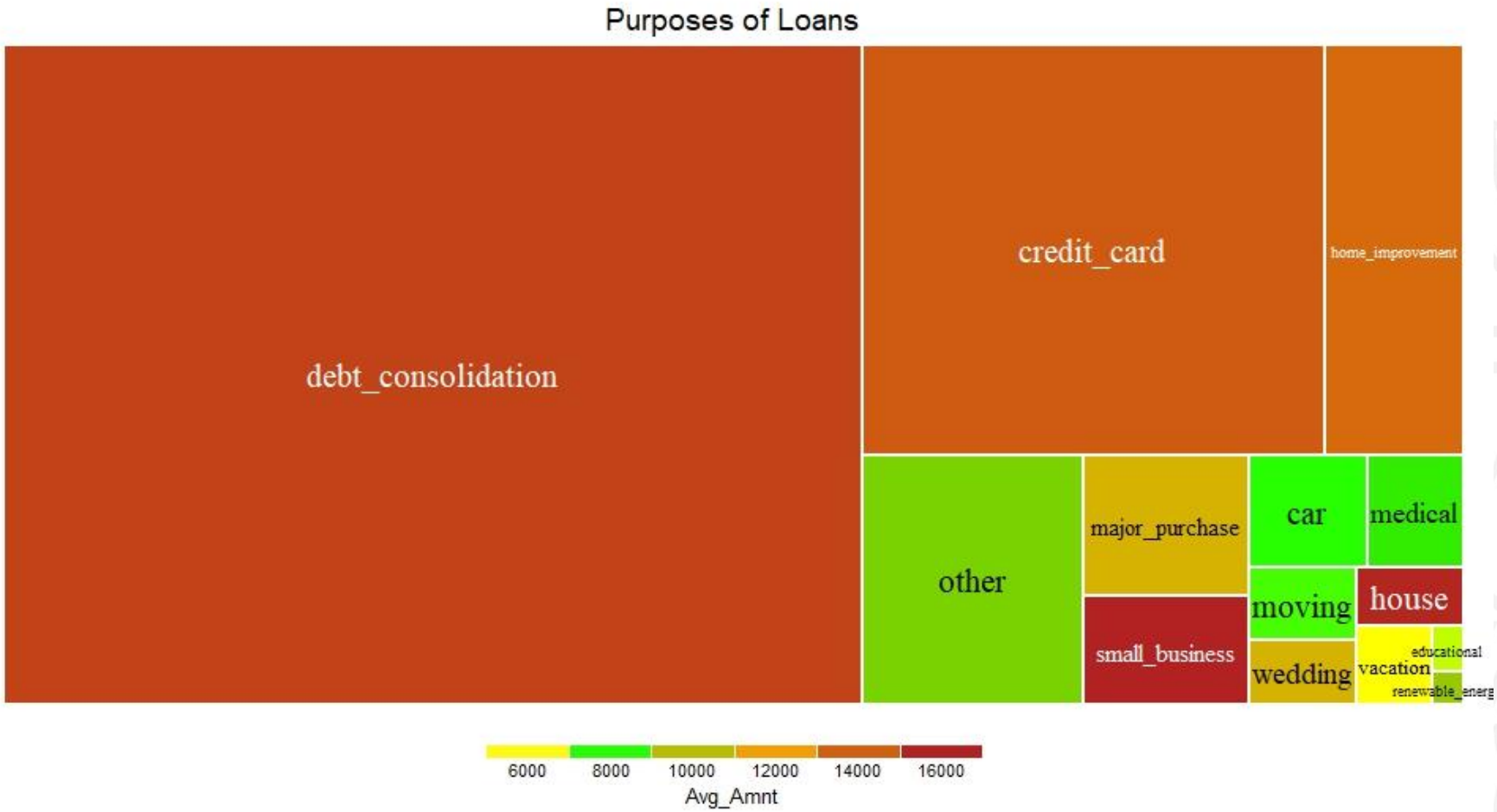




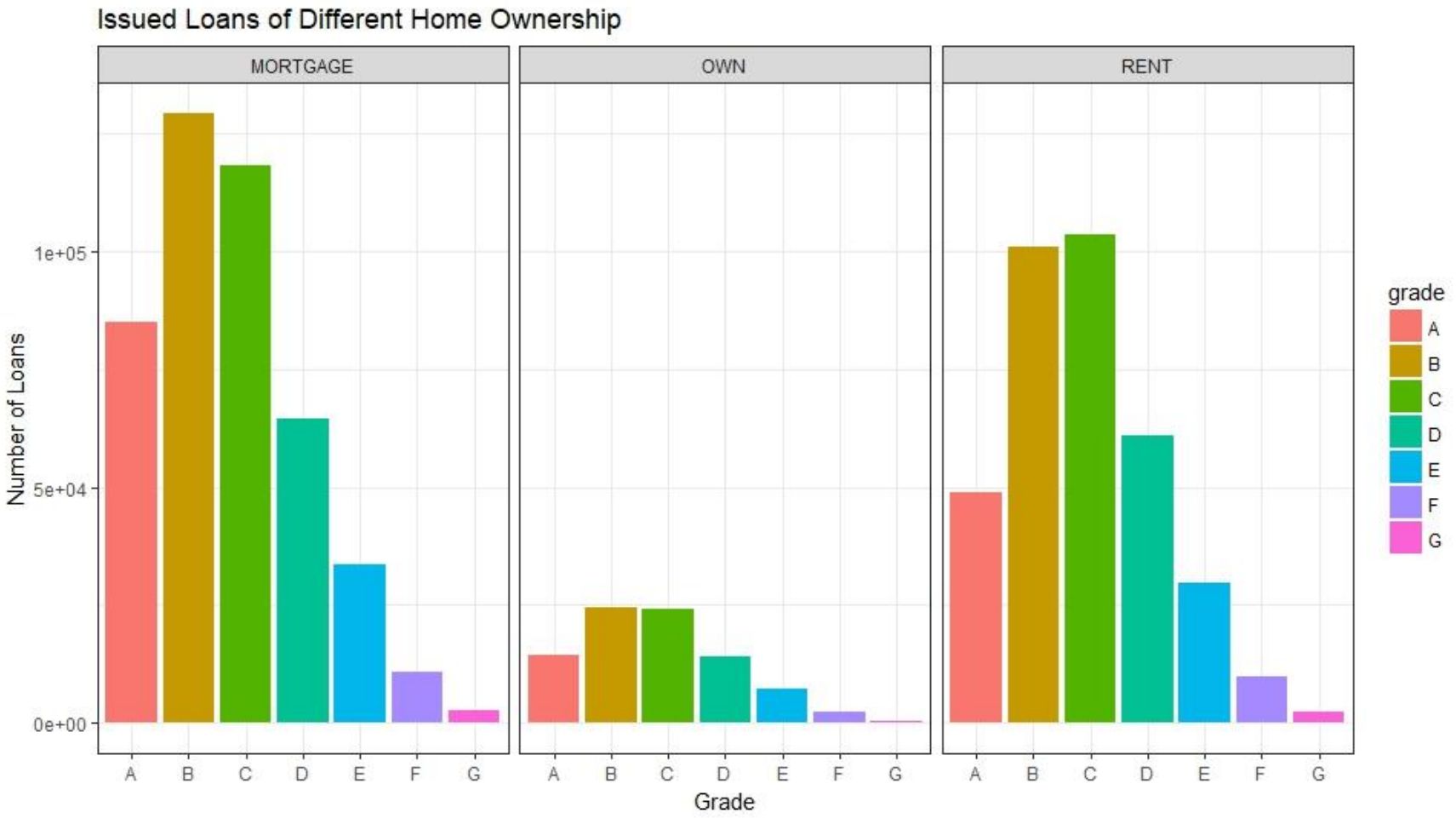
Loan Purpose



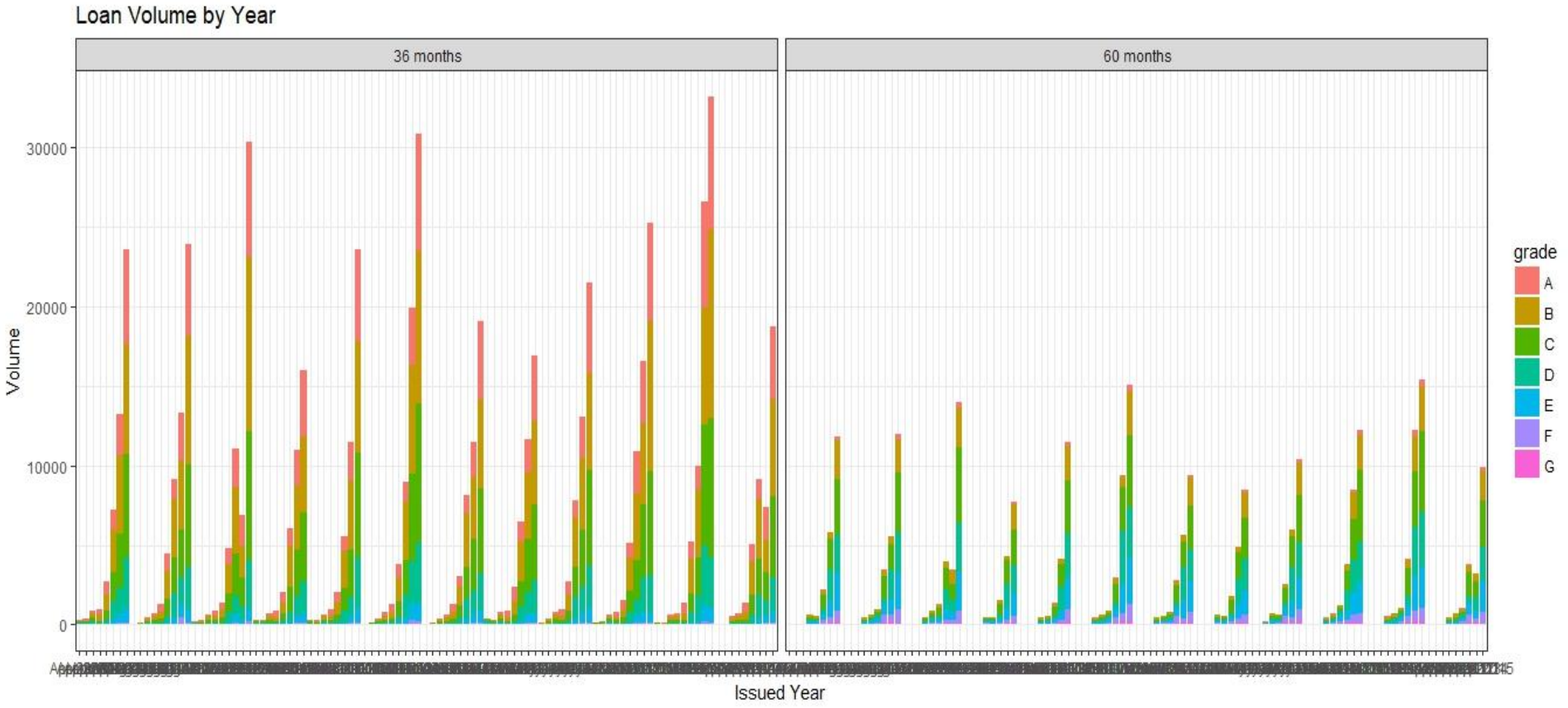
Tree Map showing Loan Purposes



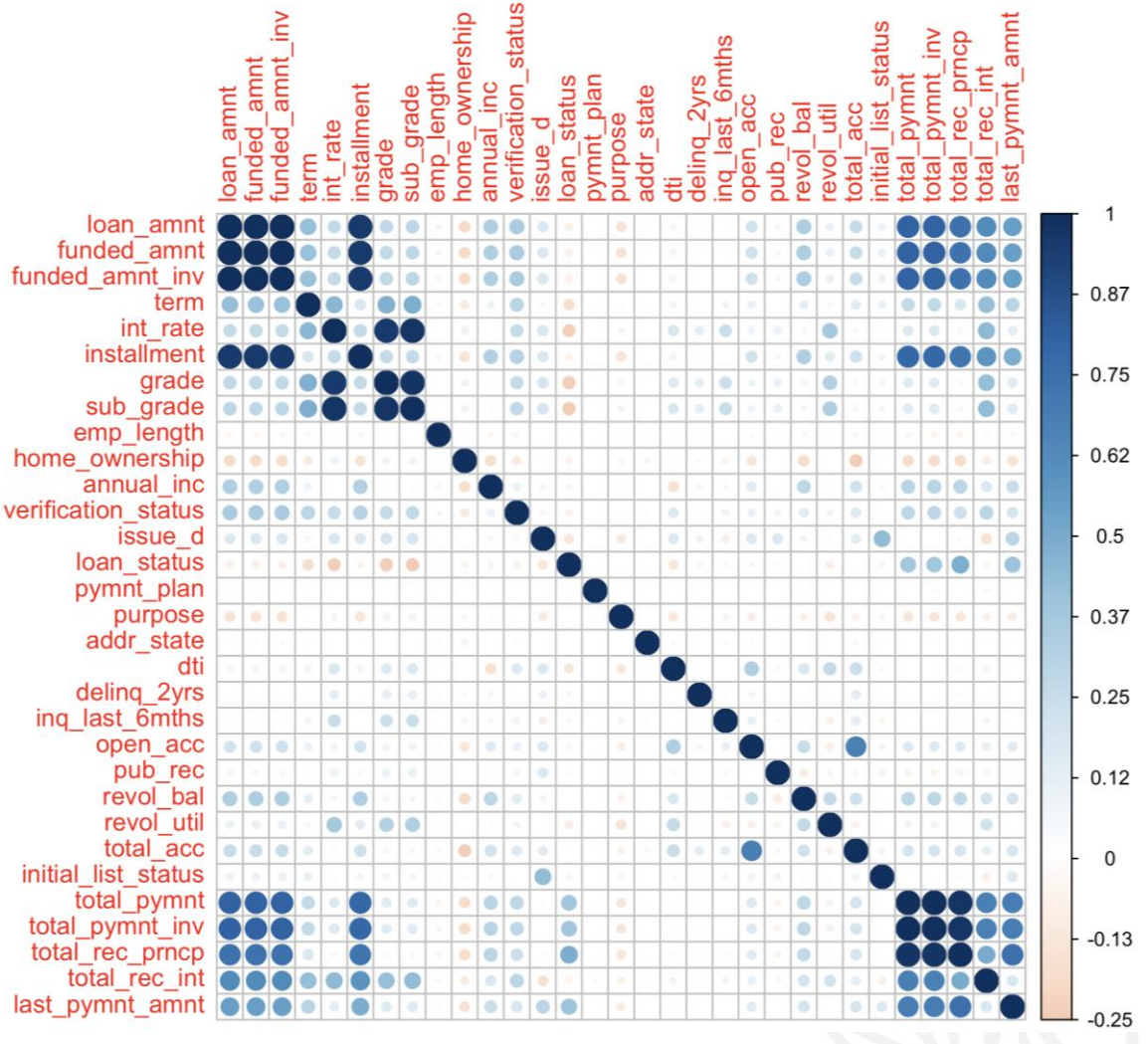
Loans by Home Ownership



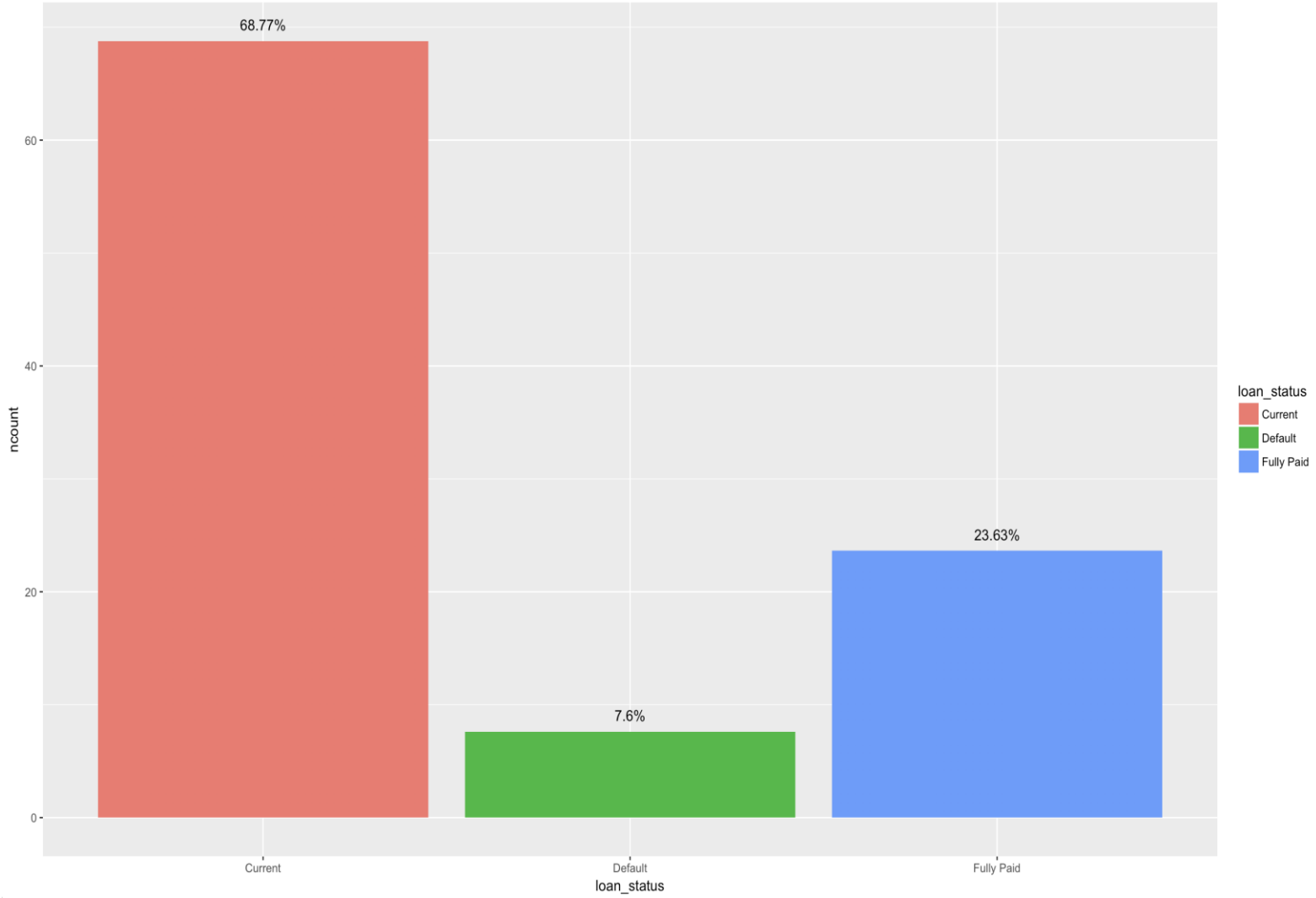
Loan Volume by Term



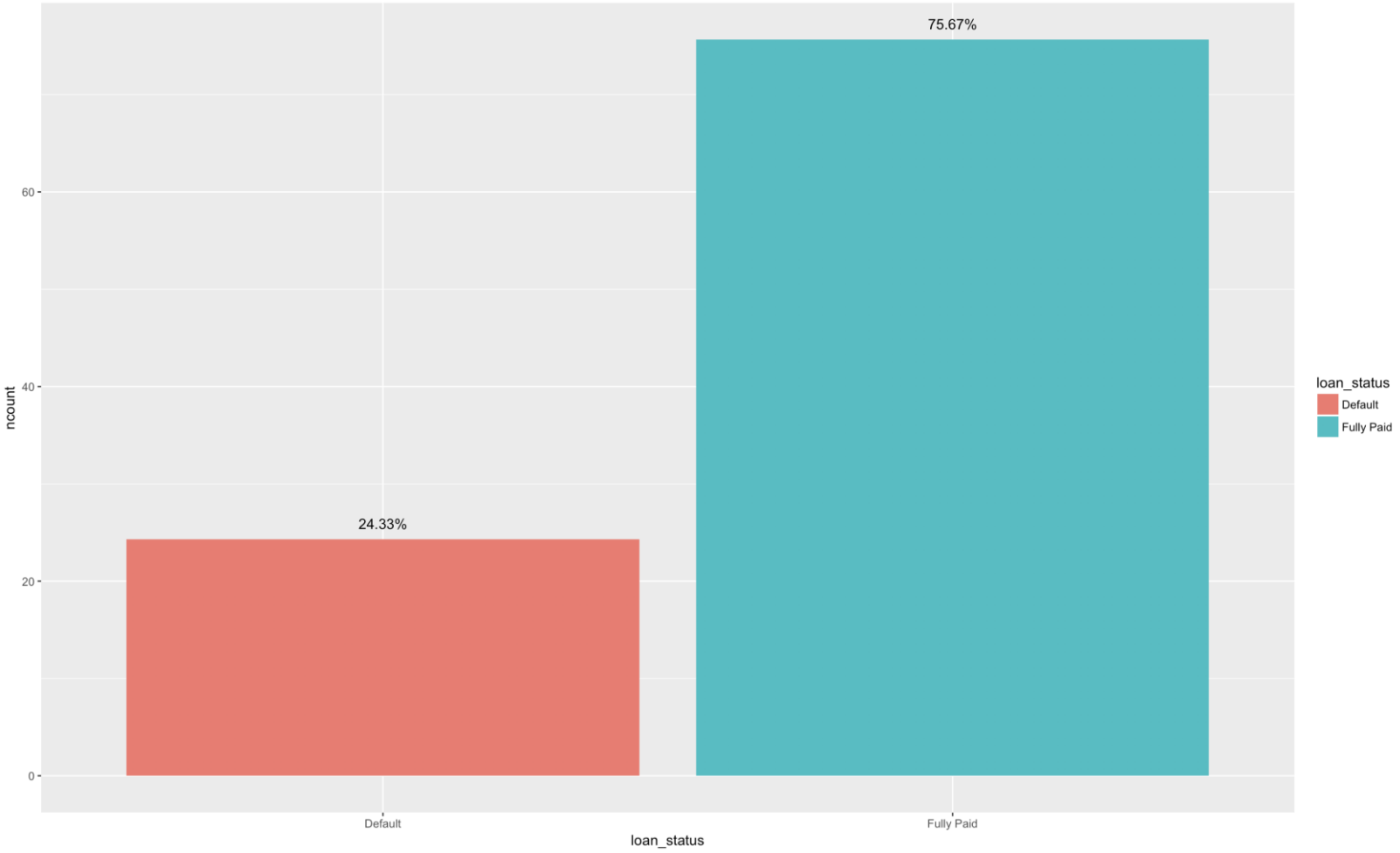
Correlation Plot



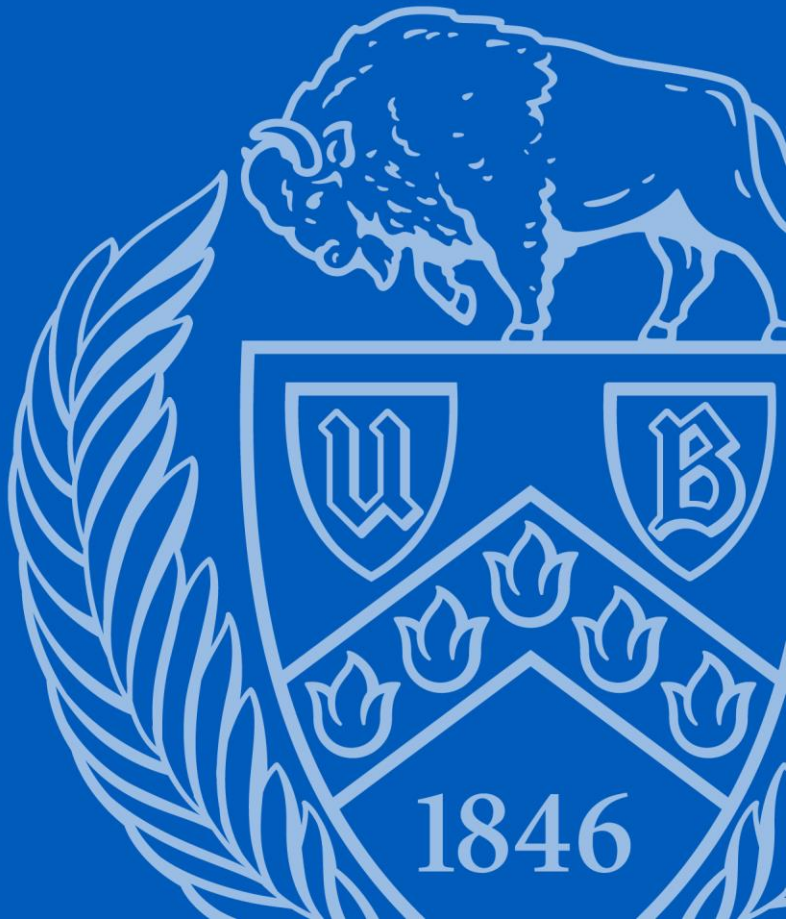
Volume of Loans by Status after Grouping



Volume of Loans by Status after dropping Current



DATA MODELLING



Machine learning algorithms

1. **Logistic Regression**
2. **LDA**
3. **Decision Tree**
4. **Random Forest**
5. **SVM**
6. **Generalized Boosting Method**

Machine learning algorithms

1. Logistic Regression.

- **Logistic regression estimates probabilities using a logistic function.**
- **It is a specialized case of generalized linear model and thus analogous to linear regression**
- **Train error = 8.29 Test error = 8.7**

```
################################
### a. LOGISTIC MODEL
################################
glm_model1 <- glm(loan_status ~ . , data = trainset)
summary(glm_model1)

glm_model2 <- glm(loan_status ~ . -(grade + pymnt_plan + total_pymnt + total_rec_prncp + total_rec_int + total_rec_late_fee +
                                    recoveries + tot_coll_amt), data = trainset)
summary(glm_model2)

glm_model3 <- glm(loan_status ~ . -(grade + pymnt_plan + total_pymnt + total_rec_prncp + total_rec_int + total_rec_late_fee +
                                    recoveries + tot_coll_amt + funded_amnt + il_util), data = trainset)
summary(glm_model3)
```

Machine learning algorithms

2. Linear Discriminant Analysis.

- **LDA assumes the data to be multivariate normal.**
- **LDA attempts to express one dependent variable as a linear combination of other features.**
- **LDA explicitly attempts to model the difference between the classes**
- **Train error = 6.78 & Test Error = 6.99.**

```
#################################
### b. LINEAR DISCRIMINANT ANALYSIS
#################################

lda_fit1 <- lda(loan_status ~.,data = trainset)
summary(lda_fit1)

###Prediction
predictlda <- predict(lda_fit1, newdata = trainset, type = "response")
predictlda <- as.numeric(predictlda$class)
predictldatest <- predict(lda_fit1, newdata = testset)
predictldatest <- as.numeric(predictldatest$class)
```

Machine learning algorithms

3. Decision Tree.

- **Decision trees are useful for visually representing decisions.**
- **A decision tree or a classification tree is a tree in which each internal (non-leaf) node is labeled with an input feature.**
- **Decision trees work well correlated variables.**
- **Train error = 0.83 & Test Error = 1.91.**

```
################################
### d. Decision Tree
################################
model.control <- rpart.control(minsplit = 5, xval = 10, cp = 0)
fitva <- rpart(loan_status~., data = trainset, method = "class", control = model.control)

### Calculating Train and Test Errors ###
pred_singletree_train <- predict (pruned_fitva, newdata = trainset, type ="class")
head(pred_singletree_train)
pred_singletree_train <- as.numeric(pred_singletree_train)
misclass_tree_train <- sum(abs(loan_status_train - pred_singletree_train))/length(pred_singletree_train)
misclass_tree_train
```

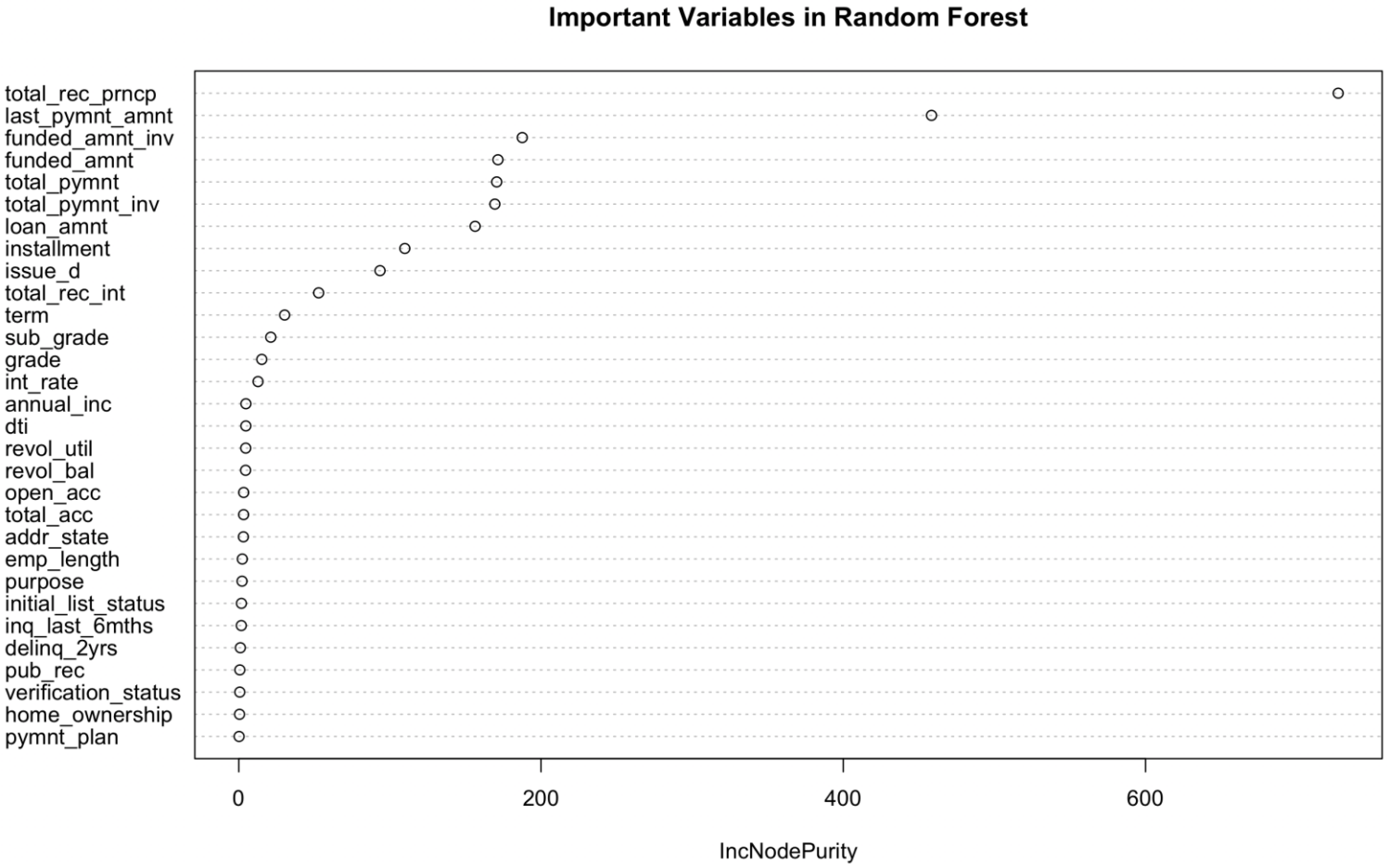
Machine learning algorithms

4. Random Forest

- **Random Forest improve variance by reducing correlation between trees. This is accomplished by random selection of feature- subset for split at each node.**
- **Forest give result competitive with boosting and adaptive bagging, yet do not progressively change the training set.**
- **Parameters = number of trees and number of bagged variables.**
- **Disadvantage : Low interpretability and consumes a lot of memory space.**
- **Train error = 0.51 & Test Error = 1.08**

```
#################################
### e. Random Forest
#################################

rand <- randomForest(loan_status~., data = trainset, n.tree = 5000)
quartz()
varImpPlot(rand, main = "Important Variables in Random Forest")

### Calculating Train and Test Errors ###
ytrue_train <- trainset$loan_status
yhat_train <- predict(rand,newdata = trainset, type = "response")
misclass_rf_train <- sum(abs(ytrue_train - yhat_train))/length(yhat_train)
misclass_rf_train
```

Important parameters in Random Forest



Machine learning algorithms

5. Generalized Boosting Method.

- Gradient boosting method is a technique which produces a prediction model in the form of ensemble of weak prediction models.
- Shrinkage coefficient gives an improvement in the model's generalization ability at the cost of computation time.
- Train error = 2.1 & Test Error = 3.1

```
################################
### f.GBM
################################
boost_train <- trainset;
boost_train$loan_status <-trainset$loan_status
boost_test <- testset;
boost_test$loan_status <- testset$loan_status


boostfit <- gbm((unclass(loan_status)-1)~., data = boost_train, n.trees = 2, shrinkage = .1, interaction.depth = 3, distribution = "adaboost")
boostfit2 <- gbm((unclass(loan_status)-1)~., data = boost_train, n.trees = 2, shrinkage = .6, interaction.depth = 3, distribution = "adaboost")

summary(boostfit)
# For shrinkage = .1
### Calculating Train and Test Errors ###
yhat_boost_train <- predict(boostfit, newdata = boost_train, n.trees = 2, type = "response")
misclass_boost_train <- sum(abs(ytrue_train - yhat_boost_train))/length(yhat_boost_train)
misclass_boost_train
```

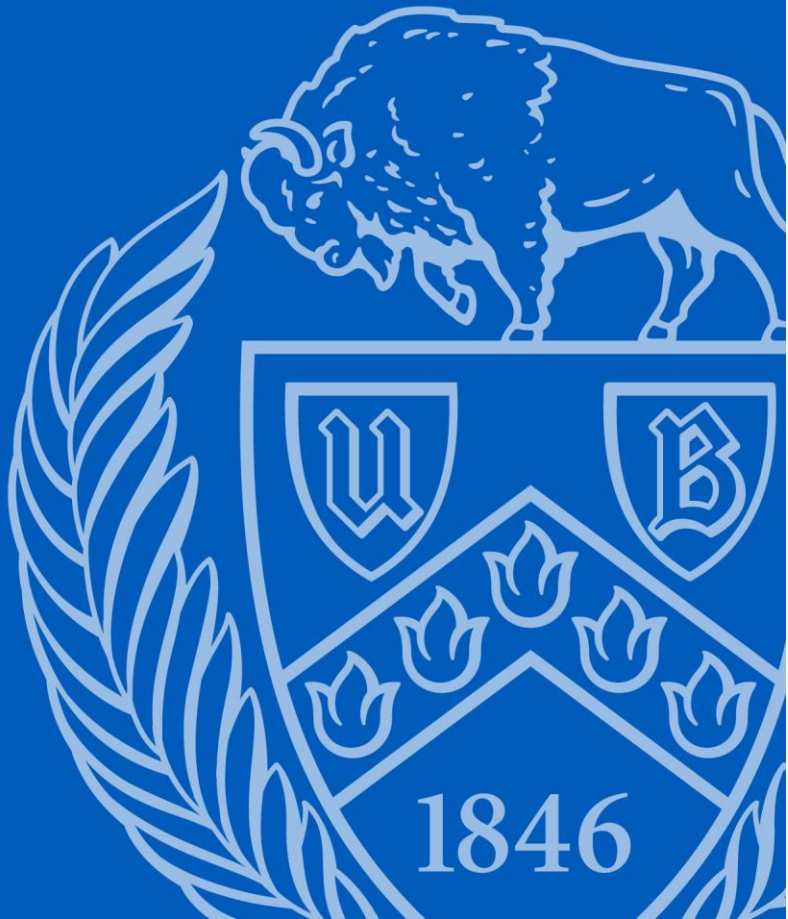
Machine learning algorithms

6. Support Vector Machines

- **SVM is a non- probabilistic binary linear classifier.**
- **SVM needs the input data to be in numeric format and the response variable should contain two factor levels.**
- **Train error = 2.30 & Test Error = 2.94**

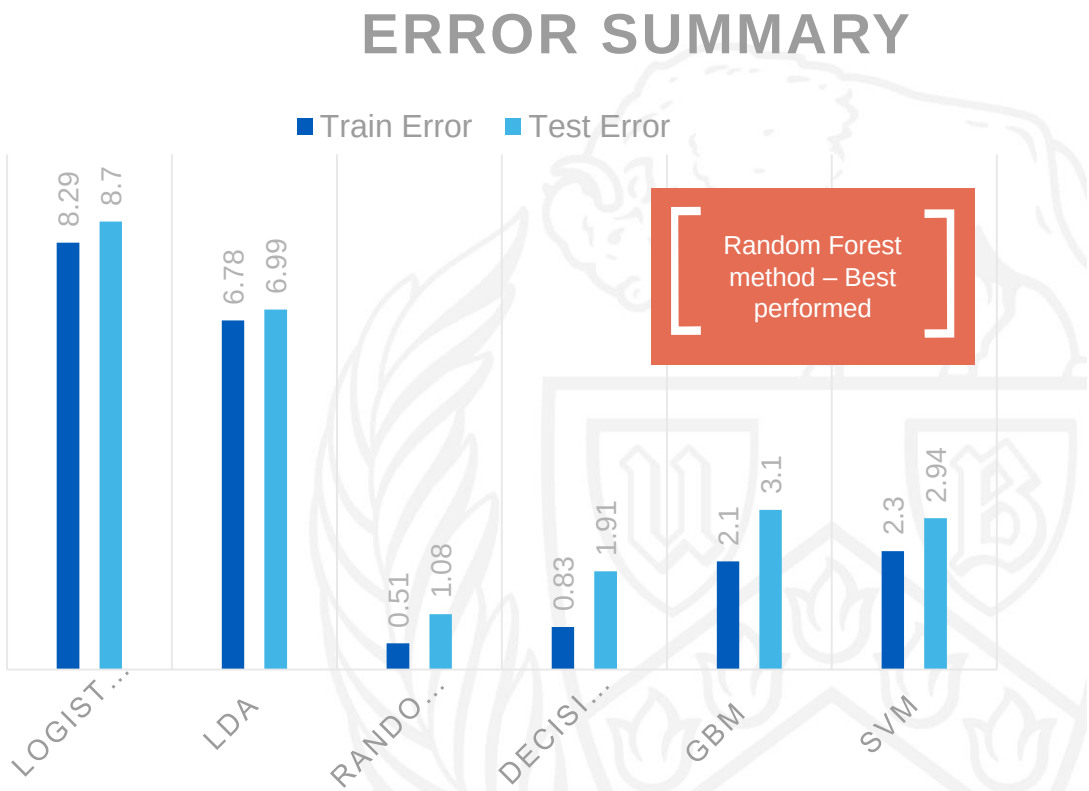
```
################################
### g.SVM
################################
svm_model <- svm(loan_status ~.,data = trainset)
summary(svm_model)

predictsvm <- predict(svm_model, newdata = trainset, type = "response")
predictsvm <- round(predictsvm)
predictsvmtest <- predict(svm_model, newdata = testset,type = "response")
predictsvmtest <- round(predictsvmtest)
```

RESULTS & CONCLUSION

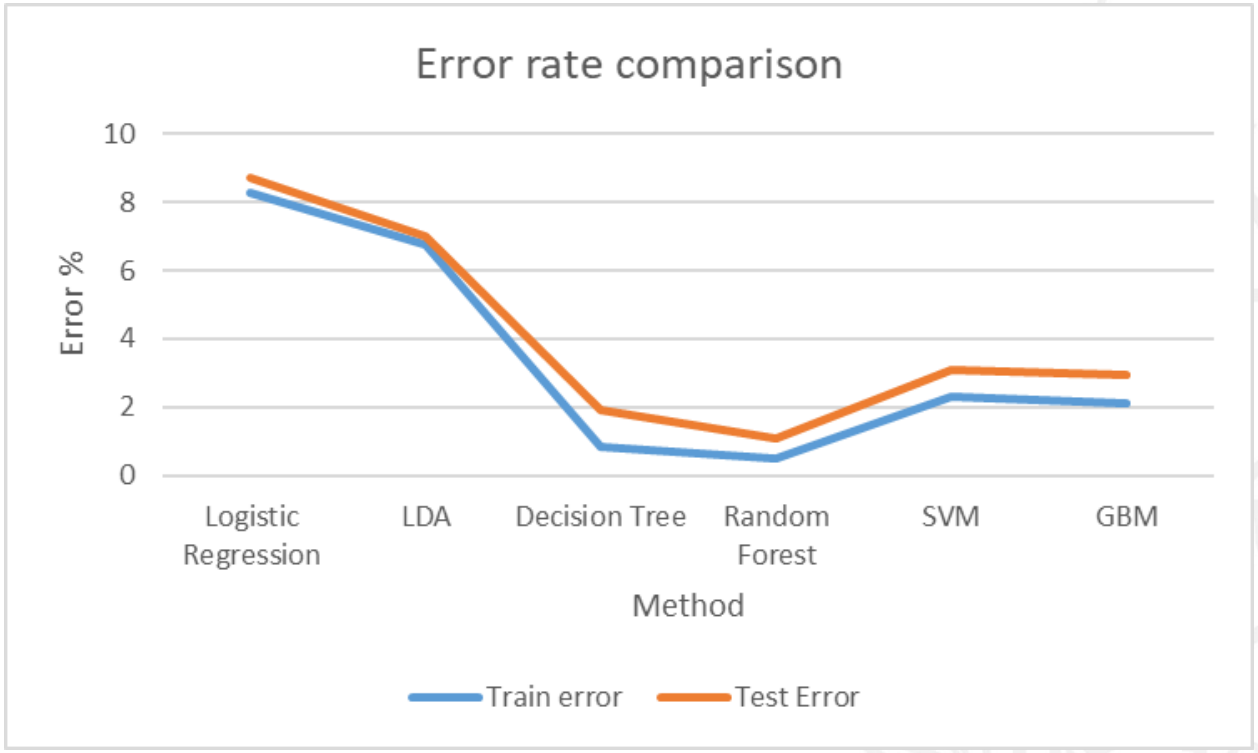


Error rate comparison

- Random forest method provides the best accuracy taking advantage of the imputation.
- Decision tree also provides a competitive performance like Random Forest.
- SVM and GBM have comparable results with some compromise on the accuracy of the model.



Error Rate Comparison



Conclusion

- Like many financial predictions determining loan outcome is also important .
- Low-grade loans are penalized so much that they rather not select anything, and thus prevent us from reaping benefits of higher interest rate.
- Our finding shows a promising loan prediction.

Future Scope

- Due to the data insufficiency/constraints few significant variables were eliminated, which can be included in the model Eg: Zip code,

THANK YOU!!!