# PROJECT 4

stat4squirrels

Xiaoqian Zhang
Xinyi Liu
Shan He
Rachneet Kaur

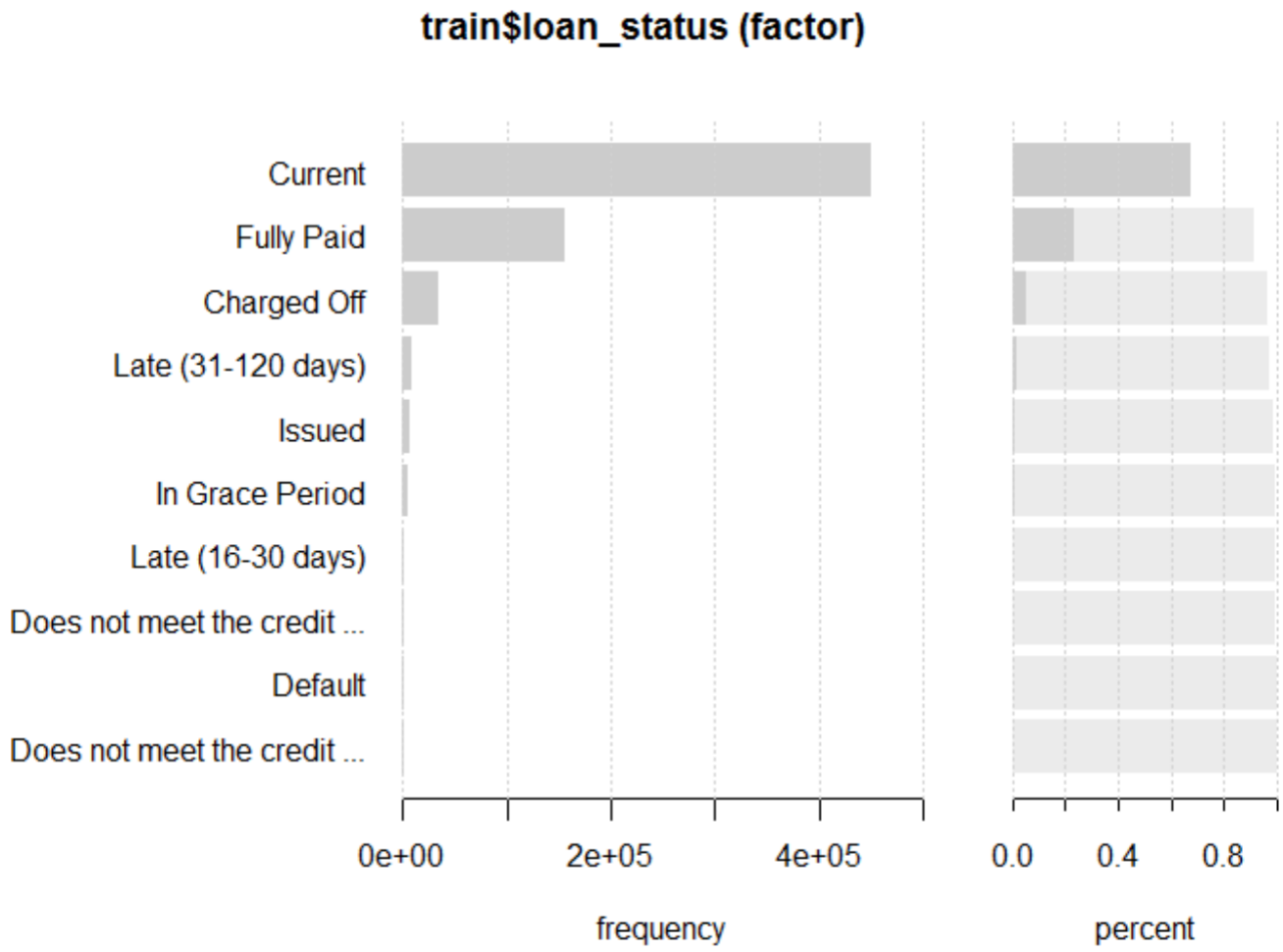
# 1. Introduction

The historical loan data we use is downloaded from Kaggle. It contains 887379 records and 74 explanatory variables. In our data preprocessing procedure, we drop some irrelevant variables, dealing with some unusual values, fill missing values, combine levels of categorical variables and dummy coding. Then we apply various advanced models to predict the loan status. After evaluating their performance, we find that xgboot, random forest and ensemble model are robust and well-performed, so we set them as our three prediction methods.

# 2. Preprocessing

## 2.1 Adding Labels

As is said in the project description, if the value of loan status belongs to "Charged Off", "Default", "Does not meet the credit policy. Status:Charged Off", "Late (16-30 days)" and "Late (31-120 days)", the observation will be labelled as "Default". Meanwhile, the other observations are labelled as "Not default". For convenience, we assign 1 to the "Default", and 0 to the "Not default" observations. The following is the distribution of loan_status.




## 2.2 Drop Irrelevant Variables

After carefully reviewing the dataset, we found that some variables may not helpful for us to solve the classification problems better.

## 2.3 Filling Missing Values

Missing data is a significant issue in the dataset. If we simply exclude variables with missing data from the model, we leave a lot of essential information on the table. Some of the variables have missing data for almost every observation, some have small number of missing values while others have considerable number of missing values. So we decide to delete those variables whose number of missing values are more than 60%. For other categorical variables with missing values, we fill missing values with the most frequent level. For numerical variables, we fill their missing values with the median value.

### 2.4 Data Transformation

For the date-type variables, we keep the year information and transform the data type to numeric. For some factor-type variables, like sub_grade and emp_length, we will transform the data type to numeric in order to reduce the computational complexity.

### 2.5 Reduction of Levels for Categorical Variables

Check the summary of each variables. It can be found that a few categorical variables have too many levels. Combining some levels will help simplify the models and acquire more accurate predictions. Hence, we set the number of levels that we want to keep. Then we could combine levels and classify the other levels into a new group called "other". To keep every categorical variable having the same number of levels, the process for test data will refer to that of train data.

### 2.6 Dummy Coding

When we want to apply some algorithms to our datasets, we need to transform factor variables to dummy variables. There is one simple step to get dummy variables using model.matrix.

## 3. Model Selection

### 3.1 Logistic Regression

Use glm function to fit a logistic regression model to the data. Then, we can acquire the predicted probabilities to default for test data.

### 3.2 Regularization: Lasso and Ridge

To apply regularized methods, we first need to transform to data into matrix and use dummy coding for categorical variables. Then fit ridge and lasso models respectively. Choose lambda.1se through cross-validation.

### 3.3 Random Forest

Based on the number of predictors and memory of R, we use 50 trees to fit a random forest model to the data and get the predicted default probabilities.

### 3.4 Generalized Boosted Regression

Use "Adaboost" method and fit a binary classification model using gbm function in R.

### 3.5 XGBoost

Fit a binary classification model returning probabilities in xgboost. The number of trees used is 400.

### 3.6 Ensemble Model

Since different models enjoy different advantages during prediction, we consider an ensemble model which is a linear combination of different models' prediction results.

# 4. Result

| Computer System | Model | Score | Time |
|---|---|---|---|
| MacBook Air/1.6 GHz Intel Core i5/4 GB 1600 MHz DDR3 | pre-process | --- | 471.563 |
| | logistic | 0.0682 | 67.889 |
| | ridge | 0.0738 | 290.732 |
| | lasso | 0.0695 | 455.487 |
| | random forest | 0.0129 | 809.43 |
| | gbm | 0.0569 | 158.3 |
| | xgboost | 0.0227 | 466.78 |
| | ensemble | 0.0370 | 0.140 |

From the results above, we finally choose xgboost, random forest and ensemble models since they have better performance than other models.