Lending Club Portfolio Construction

Sirong Huang, 361150

Rainer Klein, Student number

Rohit Saluja, 723028

Contents

[Executive summary 2](#_Toc509329443)

[Introduction 2](#_Toc509329444)

[Data 2](#_Toc509329445)

[Predictive model and results 8](#_Toc509329446)

[Prescriptive model and results 2](#_Toc509329447)

[Discussion and managerial implications 3](#_Toc509329448)

# Executive summary

Lending Club is the world’s largest online peer to peer lending portfolio, and it publishes almost all of its loan data online. Our objective was to take that data, understand and preprocess it and then find an optimal portfolio that maximizes the return on investment. This involved predicting the probability of default using two ML algorithms namely Logistic Regression and Random Forest, calculating the expected return and finally passing this data through Gurobi, along with portfolio diversification constraints to find the optimal portfolio.

We started with data preprocessing and exploratory data analysis to find the get rid of the missing values in the data and the features that have less predictive significance. Then feature engineering was done to add new categorical variables followed by one-hot encoding of few features. After this, some time was spent on Data Visualization to visually understand the relationship between various features of the dataset. Since, the dataset was heavily imbalanced we tried various methods to fix this and finally settled on the Class Weight method. Logistic Regression and Random Forests were used to predict the probability of default for each loan and then calculate the estimated return. Finally, Gurobi was used to find a portfolio that can maximize the expected return.

One-page description of the problem, your solution methods, and your decision recommendations; preferably one paragraph per each.

# Introduction

**Motivation**

The primary motivation for us as inspiring machine learning engineers is to solve a real time problem using machine learning. Using Machine Learning in the field of finance is really picking up because it is helping in making the investment predictions much more accurate and reducing the risk factor drastically. Each member of our team is quite intrigued by the application of machine learning in Finance and through this project we have learnt a lot about various aspects of applying ML in Finance ranging from dealing with financial data to building an optimized and diverse portfolio to minimize risks and maximize returns.

**Problem Description**

Lending Club is the world’s largest online peer to peer lending portfolio. We are working with lending club’s loan portfolio data to train a Machine Learning model to predict the probability of default for each loan and then eventually find an optimal portfolio to maximize the return on investment.

**Objectives and Expected Results**

* Understanding the lending club data and preprocessing it.
* Using ML models to predict the probability of default for each loan and expected returns for the test dataset.
* Find the optimal investment portfolio that maximizes the expected return and minimize the risk.

# Data

**Exploratory data analysis and preprocessing**

The training and testing datasets consist of 20 predictive features and roughly 100,000 loan records respectively. Various preprocessing steps were conducted to transform the raw data into suitable formats for default probability predictive modeling.

**Target variable**

Firstly, the target variable is defined as such: "grace period", "late", "charged off", and "default" are categorized into the "default" loan category, whereas "current" and "fully paid" are defined as "not in default".

**Missing values**

Both training and testing datasets have missing values in the variables shown in **Table 1**. These missing values were either removed or filled with sensible values chosen in accordance with the meaning of the variables:

* "desc" variable has 100% missing values, so it was simply removed.
* "mths\_since\_last\_record" means the number of months since last loan record. Missing values most likely means it's the first loan for the borrower in the platform. Therefore, the NAs are replaced with zero.
* "mths\_since\_last\_delinq" refers to the number of months since last loan default. Missing value most likely means there is no loan default (within this dataset). Therefore, its NAs are replaced with a number (1000) much larger than the maximum value (195) of this variable, in order to maintain the logical order of the numerical values.
* "emp\_title" and "emp\_length" describes two employment status of the loan applicatns. NA in this case can mean that the borrower is unemployed at the moment. Therefore, their NAs are filled with empty strings and zeros respectively.
* "dti" is a ratio calculated using the borrower’s total monthly debt payments on the total debt obligations. Its NA values are replaced with its mean.
* "zip\_code" contains only 1 missing value. It's replaced with the most common value from the same state.
* "title" is the loan title related to the application. The NAs are replaced with empty string.

A screenshot of a cell phone

Description automatically generated

Table Missing value counts for training and testing dataset

**Feature engineering**

In order to fully capture the explanatory power of the variables, some new variables were generated from existing variables.

"management\_role" is a new binary variable generated from "emp\_title".

There are 38, 367 unique employment titles in the datasets. It's impractical to convert all these titles into one hot encoded binary variables. Very fine-grained job title and industry differences are probably not crucial to the prediction of loan default. Management and senior roles, however, is a logical indicator of higher salary and more job stability. As can be seen from the word cloud in **Figure 1**, management and senior roles constitute a significant portion of the borrowers, around 27%. Therefore, we use "management\_role" to represent the most important information in the variable "emp\_title” and remove the original "emp\_title" variable from future modelling process.

"region" is a new categorical variable generated from "states".

There are 49 states in the datasets. Some states may contain crucial information on the economy and income level of their residences. The "region" variable summarizes and groups the regional difference on a larger geographical scale. The 49 states are grouped 10 categories such as "east", "west", "mid-west", "south west", "north east" etc. based on North American's economical and geographical differences.



Figure Word cloud of borrowers' employment titles

**Categorical variable transformation**

There are 9 categorical variables in the raw datasets, and 2 generated categorical variables.

"emp\_title" and "zip\_code" contains too many unique values, which may cause overfitting even for such large number of records in the training set. Therefore, they are removed for the modeling process.

"earliest\_cr\_line" means the month of the earliest credit record of the borrower. The data type is string, but it's actually a numerical variable in nature. This variable is first transformed into date format, and then the number of days between the first credit record and a recent date (1.4.2019) was calculated and used as the final values.

The rest of the categorical variables contain reasonable amount of unique values and are simply converted into binary variables using one hot encoding technique.

A screenshot of a cell phone

Description automatically generated

Table Unique value count for categorical (non-numerical) variables

**Data visualization**

Most numerical variables have left-skewed long-tail distribution. However, those outliers are very important potential indicator of loan default. Therefore, we decide not to scale or transform those values, in order to preserve their information.

A close up of a map

Description automatically generated

Figure Unique value count for categorical (non-numerical) variables

As can be seen from the correlation heatmap in **Figure 3**, most features are not highly correlated. There are a few exceptions. "loan\_amt" and "installment" were very strongly correlated, as well as "pub\_rec" and "mths\_since\_last\_record". These strong correlations will be taken into consideration in the model discussion section.

The target variable doesn't appear to be strongly related to any variables.

**A screenshot of a cell phone

Description automatically generated**

Figure 3 Correlation between predictive variables

**Final datasets for modeling**

After feature engineering, the original 20 variables are transformed into 90 variables. The resulting shape for training and testing datasets are: (103546, 91) and (96779, 91) respectively.

**Class imbalance and performance measurement**

From the graph below we can see that the target variable is extremely unbalanced, with only around 8.3% default cases. This may cause our classifiers to be overly biased towards "not in default" predictions and harm their capability to learn the default cases properly. In extreme cases, the classifiers may learn to only classify everything as "not in default" which still manages achieve almost 92% accuracy. Therefore, accuracy in this case is not a good measurement of model performance. In model selection, we used ROC AUC score for performance measurement instead.

We attempted to tackle this problem through different methods, such as upsampling the default class, downsampling the not in default class, and setting class weights in classifier settings. Downsampling and setting class weights achieved similar improvements on both ROC AUC and F1 score, and both outperformed upsampling method. Since there are very limited "not in default" cases, up sampling may not be able to generate pseudo samples with good enough quality, and down sampling may reuse the same "not in default" cases too many times that the model simply memorized those samples' labels. Therefore, we decided to use class weights as the final method to deal with the class imbalance issue. The optimal class weights were found through grid search with ROC AUC score and 5-fold cross validation.

A screenshot of a cell phone

Description automatically generated

Figure 3 Target Variable Categorization

# Predictive model and results

We chose Logistic Regression and Random Forests models to predict whether a loan would default or not. The performance of the models was compared using the confusion matrix and accuracy score. Finally, the probability of default for each loan was taken from the models to calculate the Expected Relative Loss and Expected Return. The results of the models have been discussed below.

**Logistic Regression**

The first algorithm we worked with to predict the probability of default was logistic regression. It was observed during the preprocessing phase that the data is heavily imbalanced and therefore we were getting a very high accuracy with our model. To tackle the imbalanced nature of the dataset we decide to use an inbuild method in logistic regression called class\_weight. Using the class\_weight we can incorporate the weights of the class in the objective function of the logistic regression and make it aware of the imbalanced nature of the data. We experimented with different class\_weights and compared the performance of our models using AUC and accuracy score. We chose the Logistic regression model with the best AUC score. Our best logistic regression model gave us an AUC score of 0.67 and an accuracy of 70 %. Given below are the AUC and accuracy scores we got for the various Logistic Regression models.

|  |  |
| --- | --- |
| AUC Score | Accuracy Score |
| 0.633 | 0.89 |
| 0.670 | 0.703 |
| 0.578 | 0.914 |

Table: AUC and Accuracy score comparison for Logistic Regression models with different class\_weights

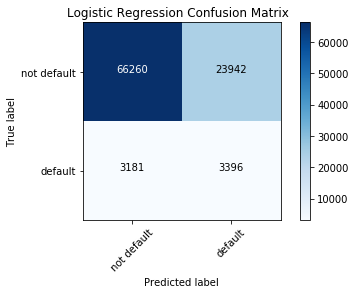


Figure: Random Forest Confusion Matrix

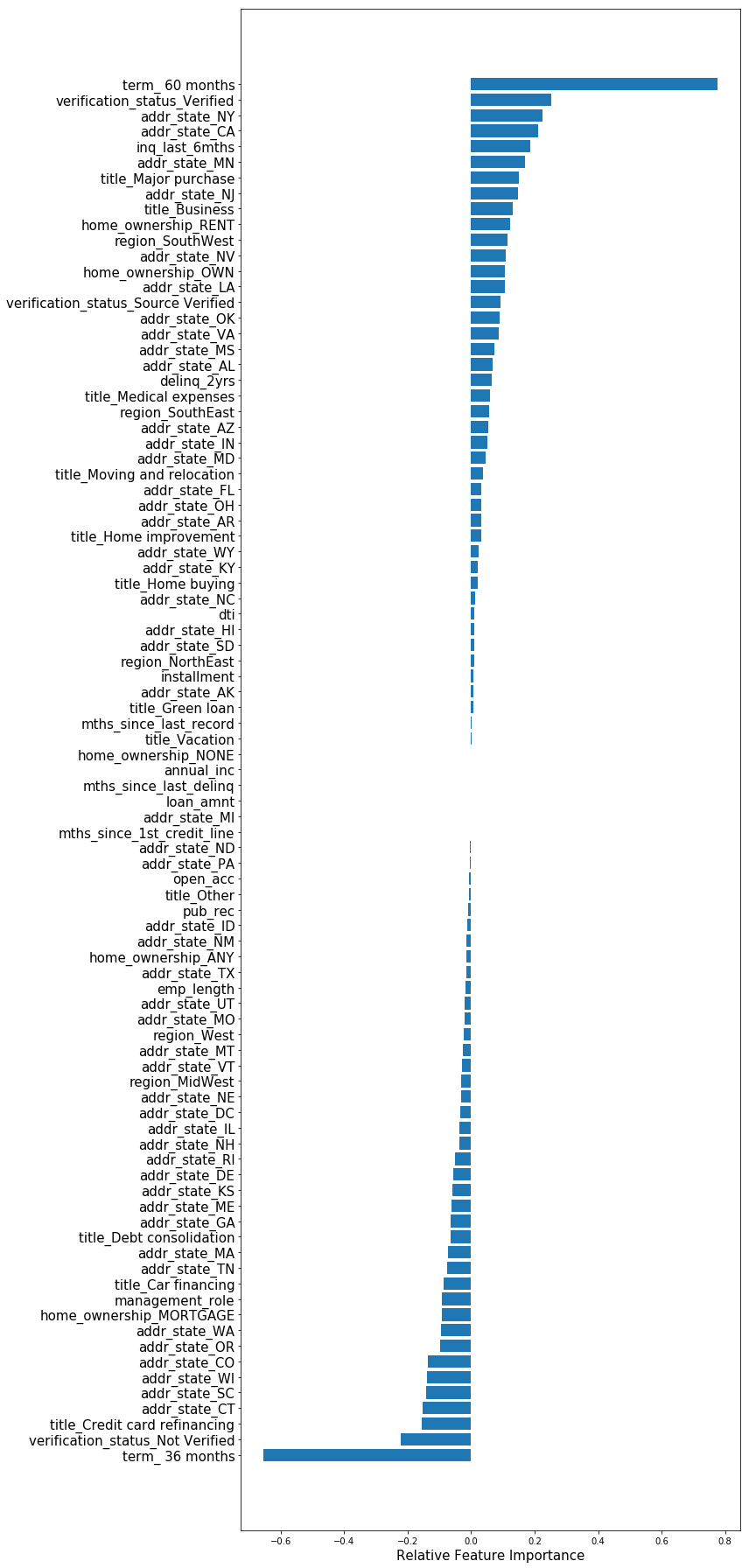


Figure: Relative Feature Importance for Logistic Regression

**Random Forest**

The second model we used was Random Forest. We also built a model with balanced random forest from imblearn library to tackle the imbalanced nature of the dataset but the results we got with the balanced random forest during portfolio optimization were unreasonable and hence we decided to stick with the normal random forest classifier. The AUC score we got with random was 0.55 and an accuracy score of 0.93. Given below is the confusion matrix and the feature importance chart for decision trees.

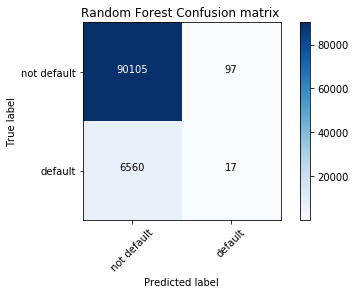


Figure: Random Forest Confusion Matrix

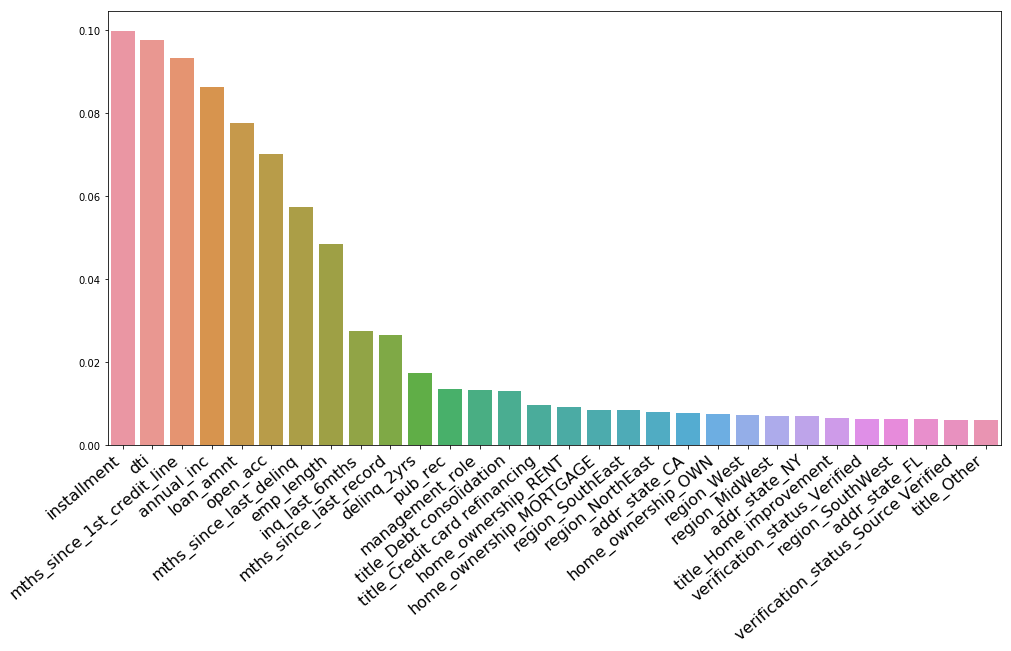


Figure: Feature Importance from Random Forest

# Prescriptive model and results

Formulate and solve the optimization problem. Show and explain your results.

* Sports case: subtasks (e)-(f)
* Finance case: subtasks (4) and Bonus task

# Discussion and managerial implications

Summarize your results, discuss the limitations of your approach, and reflect on the business opportunities created by your solution (if any).

* Sports case: subtask (g) + questions related to the case on slide 40 of lecture 5
* Finance case: subtask (5) + questions related to the case on slide 40 of lecture 5

NB! The length of the main text of your report should not exceed 15 pages. The main text should include all figures and possibly parts of tables that are relevant to be able to follow your solution idea.

Additional figures and tables can be placed in the appendix, the length of which should not exceed 5 pages. The appendix should not contain any text outside Figure or Table captions. Only material that is referred to in the main text should be included. Note that including appendices is by no means necessary.