Predicting Charged Off Loans on Lending Club

Matteo Foglio
mfogli2@uic.edu

Marco Miglionico
mmigli3@uic.edu

Marco Nanni
mnanni2@uic.edu

Riccardo Pressiani
rpress4@uic.edu

Nathan Seitz
nseitz3@uic.edu

Abstract

Lending institutions are always at risk of borrowers defaulting on their loans, and, therefore, are at risk of losing their principal investment and expected gains from interest. So, an important part of the lending process is to evaluate how much risk is assumed by investing in some borrower. The advent of predictive analytics can help companies to reduce this risk by trying to predict the outcome of these loans. From here, a lender can allocate their investment funds in borrowers who default less often which consequently leads to increased profit.

1 Introduction

The process of evaluating a loan application is very important since granting loans to borrowers who do not pay back their debt results in a huge loss for the lender. However, the profit of the lenders corresponds to the interest collected on the debt from loans. So, a lender should minimize the following wrong predictions to help maximize profit: lending to a borrower who will default and not lending to a borrower who will pay off their debt. However, we recognize the most important of the two is to avoid lending to a borrower who will default since it accounts for the greatest loss.

The improvement and the recent growth of the awareness about data science analytics and machine learning techniques for decision support has lead lending institutions to invest in automated processes to evaluate loan applications in order to address two issues: first, to automate the workload in lieu of evaluating every application manually; second, to use machine learning in attempt to detect hidden relationships, correlations, and trends among the features contained in the applications that humans cannot extract.

In order to train robust and accurate classifiers, it is necessary to have historical data about loans that were granted in the past and the related outcome, either the loan was fully paid off or not. Furthermore, it is important to collect a large and heterogeneous set of features of borrowers' loan applications which includes information about the requested loan itself, demographic information about the client, and the client credit history.

A machine learning model to aid in this kind of decision can increase profitability and allows faster processing of loans for the customers and lenders. An example that demonstrates the increasing interest in this field is the recent announcement [1] of Intuit [2]. The company has launched a new product for small business lending which is based on machine learning models developed internally by the company.

2 Problem definition

The goal of this project is to build a classifier able to predict whether a borrower, based on their loan application details, will either fully pay off their loan or not. We will focus only on the application that ended up being fully paid or "charged off", where "charged off" is defined as a loan in which a lender no longer expects the borrower to pay back any of their remaining debt. Simply, we want to predict the outcome of a possible loan before granting it.

3 Dataset

For our project, we used the dataset made available on Kaggle.com [3] by Lending Club [4] to train our models and evaluate them. Lending Club is the world's largest peer-to-peer lending company. It has been operating in the United States since 2007 and, as of June 2017, they have issued more than $28 billion in loans.

Lending Club has collected the data from each loan issued from 2007 to 2015 publicly available. This dataset contains 887,379 entries, each described by 74 features. These features contain details of the loan and the borrower. For example, features exist for the amount loaned, the interest rate, loan status, borrower demographic information, borrower credit credibility, etc. Not all of these features will be useful for our machine learning task. So, feature preprocessing and selection phase will be important.

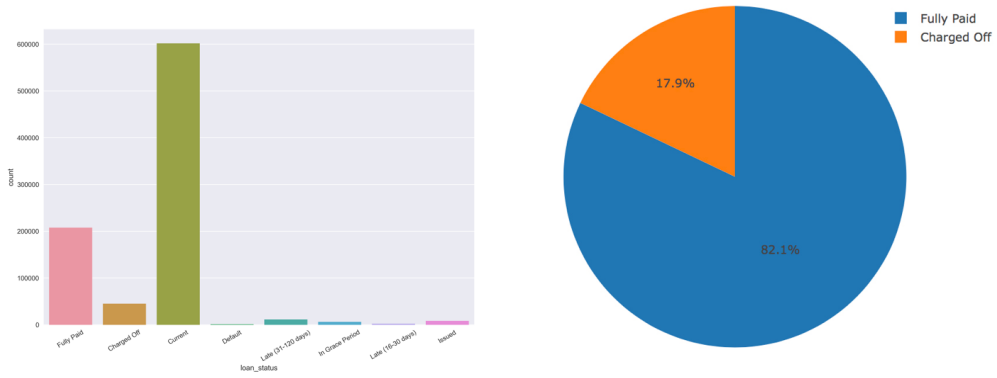
4 Approach

4.1 Feature selection and preprocessing

In this section, the decision that has been made in order to clean and prepare the dataset will be described.

- **Labels** - The dataset there are seven possible loan statuses: "Charged Off", "Current", "Default", "Fully Paid", "In Grace Period", "Late (16-30 days)", and "Late (31-120 days)". As we said before the goal we want to achieve is to predict whether a borrower, based on their loan application details, will either fully pay off their loan or not. So, we are not interested in ongoing loans but the dataset contains many entries that corresponds to ongoing loans. So, we kept only the entries which the label is either "Charged Off" or "Fully Paid" to align the dataset with our goal and discarded all other entries with a different statuses. Note: "Fully Paid" corresponds to a loan that has been fully paid back by the borrower, while "Charged Off" corresponds to a loan where the lender does not expect to receive any more money from the borrower.

  In Figure 1 the distribution of the labels before (1b) and after (1a) the preprocessing is shown. After dropping the entries we were not interested in, we have a dataset strongly unbalances with 17.9% of the loans labeled as "Charged Off" and 82.1% of them labeled as "Fully Paid".



(a) (b)

Figure 1: In the two figures above are shown the accuracy (1b) and F1-Score (1a) results obtained with the four techniques considered.

- **Joint loans** - The Lending Club dataset also includes joint loan applications. Joint applications are the ones applied for by two or more borrowers. After removing the entries with labels we are not interested in, only one joint application is left in the dataset. In addition, there are three features that are only related to joint applications and are empty for all of the individual applications. For this reason, we have decided to drop both the joint application entry and the three related features.
- **Ongoing loans** - As we mentioned before, we are interested only in loans that have come to an end since we want our classifier to focus on the outcome of the loans based on the information that was available at the time of the application. The dataset includes numerous features that represent information of the current status of the loan, which is not known at the time of the application. An example of one of these features: the date of a borrower's last payment. For this reason, we have dropped all of this features.
- **Features transformation** - Some of the features do not add any information to the dataset if considered as standalone features. For this reason, we analyze all of these features and tried to extract new features capturing the essence of the original ones so these new features could be more informative and useful for training the classifier. For example, the dataset include features regarding the date of the first and the last credit lines opened by a borrower. We combined these two features to extract the length of the borrower's credit history by subtracting the two dates.

  Moreover, we have used one-hot-encoding to transform all the categorical features that were described by strings.
- **Remove non informative features** - Some features contained user-entered text by the borrower. For example: the description of the loan, the description of the borrower's job, and the borrower's job title. Since we were not able to extract useful information from these features and many of entries do not have a value for these features, we decided to drop them.

4.2 Techniques

4.2.1 XGBoost

In our analysis, we took into consideration different machine learning models. Since our dataset is heavily unbalanced, instead of focusing on achieving a good accuracy alone, our goal was to also improve F1-score. From this, we can narrow our consideration of different machine learning models to ones appropriate for unbalanced classes. Generally speaking, boosting techniques can deal with the problem of unbalanced datasets. Boosting is an ensemble technique to combine weak learners in order to create a strong learner that can make accurate predictions. Boosting starts out with a weak classifier that is prepared on the training data. A classifier learning algorithm is said to be weak

when small changes in data induce big changes in the classification model. The main advantage of boosting is that, in each iteration, a new classifier focuses on or places more weight on those cases which were incorrectly classified in the last round. Therefore, it prevents the risk of classifying all the points as belonging to the majority class.

Our first choice has been the well know XGBoost algorithm (Extreme Gradient Boosting), that is an advanced and more efficient implementation of Gradient Boosting Algorithm. In the general Gradient Boosting methods many models are trained sequentially. It is a numerical optimization algorithm where each model minimizes the loss function, $y = ax + b + e$, using the Gradient Descent Method.

As we can observe from Figure 2, gradient boosting builds the first learner on the training dataset to predict the samples and calculates the loss (difference between real label and predicted one of the first learner). This loss is then used to build an improved learner in the second stage. At every step, the residual of the loss function is calculated using the Gradient Descent Method and the new residual becomes a target variable for the subsequent iteration. Unlike Gradient Boosting Algorithm which stops splitting a node as soon as it encounters a negative loss, XGBoost splits up to the maximum depth specified, prunes the tree backward and removes splits beyond which there is an only negative loss. Another advantage: it is 10 times faster than the normal Gradient Boosting, as it implements parallel processing, and it is much more flexible, since it is possible to use a custom optimization objective (binary logistic in our specific case).

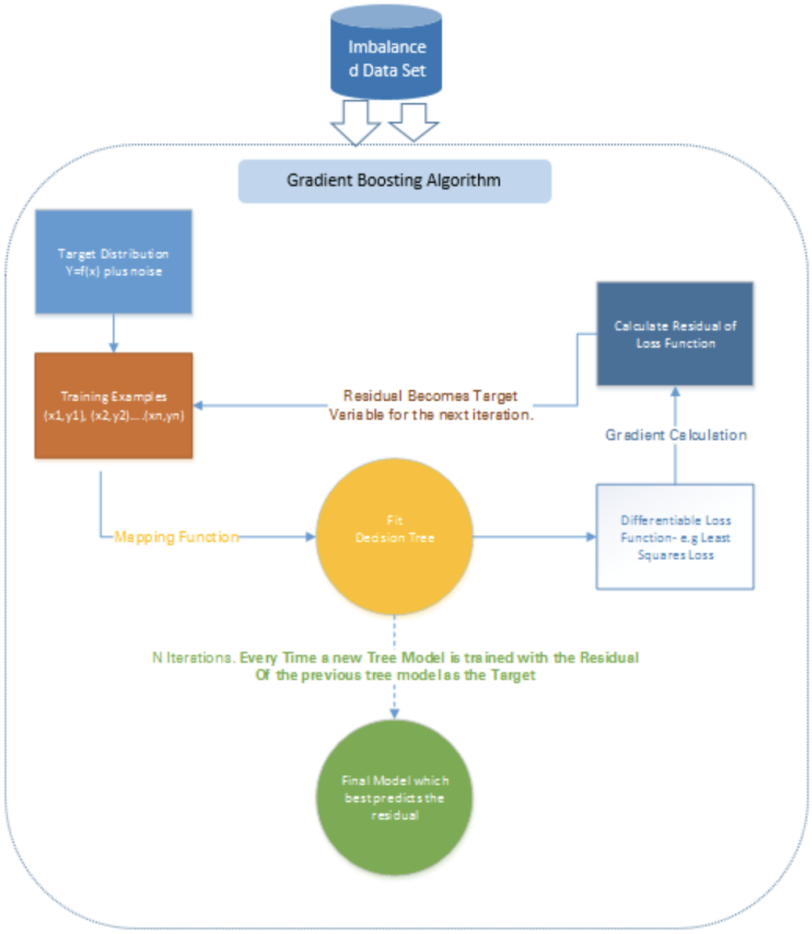


Figure 2: Gradient Boosting pipeline - Source: `https://www.analyticsvidhya.com`

4.2.2 Support Vector Machines

For completeness, we've decided to include classification using Support Vector Machines (SVM). We did not expect this classifier to perform well since it allows for some misclassification to be made in training. The intuition behind this concept is that these misclassifications, under normal circumstances, may just be outliers. Here, we risk an SVM classifying our parts of our minority class as "outliers".

When classes are separable, the decision boundary generated by SVM depends only on the support vectors, which may represent only a small portion of the whole dataset. Given the presence of several features, we had to deal with a multidimensional space which is not easy to plot an analyze visually. Even if we knew we were dealing with a complex dataset, our first attempt was to find a linear classifier, following the Occam's Razor principle. This step gave confirmed the two classes were not linearly separable: linear SVM does not achieve good score on this dataset. Consequently, we tried to find a better decision boundary using different kernels. Our analysis includes the use of linear, polynomial, sigmoid, and Gaussian kernel. They allow to us to achieve better performance with respect to F1-score. However, the improvement in F1-score corresponds to a serious deterioration in accuracy. Moreover, even if F1-score was improved, it still do not overcome XGBoost results.

4.2.3 Logistic Regression

Another attempt was to use Logistic Regression both to solve the classification problem and to understand the linear correlation between features and label. We achieve the best result by balancing the classes and using L1 regularization. Moreover we also tuned the regularization parameters using cross-validation.

5 Results

5.1 Evaluation Metrics

Accuracy is a very common metrics for a binary classifier like the one we are considering, since it takes into account both true positive and true negative with the same weights.

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn} \tag{1}$$

However, since the dataset is very unbalanced, the rate of negative examples (corresponding to the fully paid loans) is about 82% and thus accuracy does not indicate the true performance of the model. For instance, a classifier that will predict always the majority class will achieve an accuracy value of 82%. For this reason we also need to consider other metrics that take into account also false positive and in particular false negative. In fact, the latter corresponds to the case when the classifier predicts that the loan will be paid back but actually this will not happen, meaning that the creditor will lose his money.

The other metric that we will consider is the F1 score, it takes into considers both the precision and the recall of the test to compute the score.

$$Precision = \frac{tp}{tp + fp} \tag{2}$$

$$Recall = \frac{tp}{tp + fn} \tag{3}$$

$$\textit{F1-score} = 2\frac{precision \cdot recall}{precision + recall} \tag{4}$$

	SVM (linear)	SVM (sigmoid)	Logistic Regression with L1	XGBoost
Accuracy	0.8122	0.1788	0.6488	0.6871
Precision	0.0778	0.1788	0.2838	0.3075
Recall	0.0224	1.0000	0.6357	0.5905
F1-Score	0.0347	0.3033	0.3899	0.4044

Table 1: Comparison of accuracy, precision, recall and F1-Score among the techniques used.

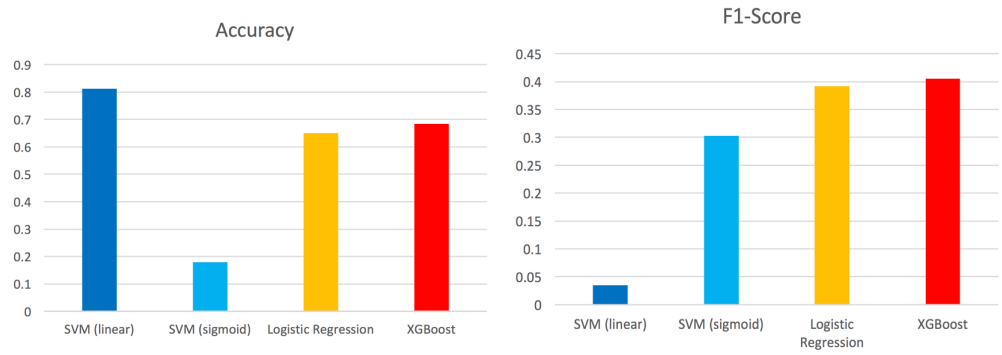
5.2 Techniques Comparison

From the analysis of the previous graph it is possible to understand some important differences between the models. Result for SVM and Logistic Regression are obtained using 10-fold cross-validation whereas XGBoost was evaluated using a 30% bootstrapping.

As expected, SVM with linear kernel have the same accuracy as a dumb classifier that labels all the examples in the dataset as fully paid. The main reason is that the linear kernel cannot separate well the two classes and for this reason create a boundary that classify all the examples with the same label, producing a good result in term of accuracy, since the dataset is very unbalanced. On the hand, the main drawback is that the F1 score will be very low. Differently, a SVM with a sigmoid kernel, will generate a much more complex decision boundary, resulting in an accuracy that is worse than a dumb classifier, but in an F1-score that is better. This is because it tries to effectively discriminate between each data point and not like before when to each of them was given the same label. The logistic regression model reach a good compromise between accuracy and F1 score. In fact, even if the accuracy is still lower than a dumb classifier, for the same reason of the SVM, results are improved thanks to the use of L1 regularization. Moreover, in order to train logistic regression we also balanced the weights associated to the classes. Finally we used cross-validation to tune the regularization parameter.

The XGBoost, as expected, reach the best trade-off in term of both accuracy and F1 score with respect to the previous models. The main reason is that this ensemble method use the loss to iteratively build an improved classifier after each stage and prunes the tree backward to remove splits that have only negative loss. Given the following results, we believe that XGBoost represents the best model to accomplish our task.

Table and Figure 3 summarize the results of our models.



(a) (b)

Figure 3: In the two figures above are shown the accuracy (3a) and F1-Score (3b) results obtained with the four techniques considered.

6 Conclusion

Predicting outcomes of loans from information available at the time of the application can be seen as an anomaly detection problem and, similarly, to all the problems that are included in this category, it is not a trivial task. First of all, one of the major issues that has to be taken into account is the fact that the datasets are strongly unbalanced. For these reasons, techniques that manage to address this problem have to be preferred. As an example all of the approaches that use bagging and boosting techniques can be exploited to train the classifier starting from sampled datasets in which the unbalancing is reduced. The issue just mentioned lead us to the first lesson we have learned during the development of this project: we are convinced a thorough understanding of the problem which we are working with can help us to select some techniques and models rather than others.

The second lesson we learned is analyzing all the features available and accurately prepared the dataset is very important to improve the performance of our machine learning models. With the rapid growth of the Big Data field and storage systems becoming cheaper, a lot of data collected is being stored for future exploration. While, this can be useful because there are many more features that can be exploited to train a machine learning model, there exists noise and useless data too. For this reason we always have to analyze and accurately clean our datasets before starting a machine learning task.

For future work, one possibility is to include in our prediction task also the ongoing loans. This can be interesting to give the lenders an insight about the trend and the status of the current loans so that they can intervene with actions before the loan is definitively declared charged off.

Another interesting direction that can be taken into account is the analysis of bias in our prediction. We should check that our classifier is not biased on sensitive attributes like race, religion, cultural individual aspects, and hometown. Even if these attributes are not explicitly present as features in the dataset, some classifier could unintentionally infer them based on the features available.

References

[1] Frederic Lardinois. Intuit launches QuickBooks Capital, a small business lending service powered by AI. Available at `http://tcrn.ch/2Aqbx36` (December 15, 2017).

[2] Intuit. Available at `https://www.intuit.com` (December 15, 2017).

[3] Kaggle.com - The Home of Data Science & Machine Learning. Available at `https://www.kaggle.com` (December 15, 2017).

[4] LendingClub Corporation. Available at `https://www.lendingclub.com` (December 15, 2017).