Applying Machine Learning Methods on Loan Default Prediction

Evaluation of data balancing techniques on the performance of diffirent model

Sheldon Rodrigues
School of Information Systems
and Management
Carnegie Mellon University
Pittsburgh, PA
sheldonr@andrew.cmu.edu

Yiru Wei
School of Information Systems
and Management
Carnegie Mellon University
Pittsburgh, PA
yiruw@andrew.cmu.edu

Zhongli Zhao
School of Information Systems
and Management
Carnegie Mellon University
Pittsburgh, PA
zhongliz@andrew.cmu.edu

Abstract – This paper applies machine learning algorithms to construct nonlinear predictions of P2P mortgage loan default. We obtained the dataset from Lending Club with over 800,000 loan data over period 2011-2015. Different machine learning algorithms are applied in the study of loan classification. Evaluation of models are based on F-1 score and ROC curve. Naïve Bayes serves as the base-line model, whereas Random Forest model gives the best prediction result. Since the data does not has a good balance between default and good loans. We steered out study to understand how different data balancing techniques impact the performance of selected models. Applied up-sampling, down-sampling and SMOTE balancing techniques across the three models used and evaluate the performance based on balancing technique. We also explored extreme cases by selecting only 0.1% and 5% of default loans as training set and explore the how the balancing techniques differentiates performance on different model.

Keywords—loan; default; classification; logistic regression; naïve bayes; random forest; artificial neural network; imbalanced data; SMOTE

I. Introduction

Peer-to-peer lending (abbreviated as P2P lending), is the practice of lending money to individuals or businesses through online platform which provides both demand and supply information from borrowers and lenders. [1]. Typically, the rate of return from the investment for lenders is higher as compare to standard saving and investment products offered by banks. On the other hand, borrowers can borrow money at lower interest rate since the online platform requires less service fee as compared to traditional financial institutions. After the 2000s' financial crisis, more people switched from traditional financial institutes to P2P companies for lending and borrowing. As a result, the P2P market faced an increasing investor scrutiny due to the increasing default rate. With this disconnection between lenders and borrowers, the liquidity and the return rate wound both be affected. The disconnection created an inefficient market that the lenders are reluctant to invest due to the default rate and borrowers are having fewer options. The root cause for this problem is the lack of risk evaluation before the loan agreement as well as lack of risk monitoring during the loan period. If there are means to give investors continuous insights on the behavior of lenders, the risk of default payment would be minimized. The lenders would be the direct beneficiary and lender could invest with more confident. More confident investors would positively stimulate the P2P lending market with more liquidity and the "healthy" (low risk) borrower would be benefited as well.

This paper used data publicly available from LendingClub.com. LendingClub provides a risk grading, which is proportional to the loan interest rate. The goal of our machine learning model is to provide a tool for investors to detect loans that are likely to default and prevent losses/increase returns for both investors and LendingClub. [2]. Applying machine learning to loan default predictions showcase a useful application of how to solve real world business problem. We built a model with a conservative investor in mind. The machine learning model will give real time classification of whether a loan will be default. Investors could utilize the model prediction as an indicator and act accordingly to minimize their loss.

Most of the loan dataset consist far more "good" loans (fully paid) than "bad" loans (default and charged off). The data is likely to be imbalanced. In this paper, we first compared the modeling results on unbalanced data versus balanced data. We also researched on how different data balancing techniques such as up-sampling, down-sampling and SMOTE impact the performance of various machine learning models. With the evaluation provided on different sampling strategy, other researchers may find it useful when selecting models and data balancing methods. To further evaluate the differences, we processed the data and skewed the dataset to an extreme level. In this case, the differences between balancing techniques can be amplified. Hope our evaluation could provide valuable insights.

II. Data Understanding and Preparation

A. *Overview*

We obtained the loan data from LendingClub over the period of 2011-2015 which consist of around 870,000 data points with 75 features. The dataset is 378MB in size and comes in .csv format. We will discuss how to better achieve our goal of better understanding by analyzing the data with data dictionary. This allows us to remove certain features based on domain knowledge and get a sense of which columns could be significant in the model.

The input of the dataset is a combination of various features of the loan. The outcome of the dataset is the status of the loan, such as default, charged off, late, fully paid, etc. During the data cleaning, we will perform techniques such as excluding columns that are less relevant to the outcome of the data, removing columns with sizable amount of missing values and imputing null values with median (since it is more robust than mean).

B. Data Description

The model will be used to predict likely default loans and likely fully paid loans based on loan features. The classification can only be trained with labeled loans, which means the training set should have the label of "Default loan" and "Good loan". Thus, we can only study the completed loans. The raw data contains multiple loan status including "fully paid", "charged off", "default", "in grace period", etc. We removed all "issued" and "current" loans from the dataset since those are label-less. We recategorized loan status as the following table.

Table 1: Loan Condition Summary

Loan Status	**Count**	**Class**	**Code**
Fully Paid	207723		
Does not meet the credit policy. Status: Fully Paid.	1988		
Late (16-30 days)	2357		
In Grace Period	6253	Good	0
Subtotal:	**218321**		
Charged off	45248		
Does not meet the credit policy. Status: Charged Off.	761		
Defualt	1219		
Late (31-120 days)	11591	Bad	1
Subtotal:	**58819**		

Among the 75 features, we studied the basic correlations between standard features of a loan for a preliminary understanding, it is not hard to find that higher interest rates normally would lead to higher default rates. As displayed in Table 2, the average interest for an investor is 17.43%, and the average default rate on an investment is 21.22%. If the investor wants to be conservative and choose Grade A loans, the return on investment is 8% with a comparatively loan default rate at 6.82%. If the investor is targeting at higher risk and higher return, the Grade F loan would yield 24% interests, but 2 out of 5 loans are likely to default. [2].

Table 2: Interest and Default Rate per Grade

Grade	**Ang Int**	**# Bad**	**# Total**	**Default Rate**
A	8	2963	43432	6.82%
B	12	11056	80271	13.77%
C	15	17154	72213	23.75%
D	18	13959	46373	30.10%
E	20	8845	23051	38.37%
F	24	3873	9263	41.81%
G	25	969	2537	38.19%
Total	**17.43**	**58819**	**277140**	**21.22%**

C. Data Processing

To be conservative, we are trying to minimize the loss on investment or avoid default loans. The machine learning model will be focused on discriminating against loans with potential for default, on the other hand the model would favor a loan classified as good. For a business stand point, given the availability of the loans (always more loans available than the money to be invested) on LendingClub, investors can afford to incorrectly eliminate good loans as bad. Therefore, in this paper, we will focus on precision, recall, AUC and F1 Scores at the cost of overall accuracy. [2].

LendingClub data required significant cleansing before ingestion. We first examine the features with large amount of null values (80%). We went through the data dictionary to check if any of those columns is critical to the loan condition (given default loan is small in proportion). We removed 23 columns which meet the above two criterions. For the remaining columns null values, we took three different approaches:

1. For features like "emp_title", the most frequent job title only accounts for 0.8% of the entire dataset. We do not observe much variation, so we can delete "emp_title" column.

2. For features like "Id", "URL" which do not provide any indication towards loan classification, we delete those columns.

3. Filled the null values in remaining features with the median value of that feature, which is less affected by the outliers.

After first round screening and filling, we put more emphasis on the meaning of data. For features with majority of non-meaningful "0" or very little variances, such as "acc_now_delinq" and "application_type" which consist of 98% 0-values or same values. We determined those meaningless towards the model and removed such features. For functionally duplicated feature based on data dictionary such as zip code vs state, we chose to keep only one (state in this case).

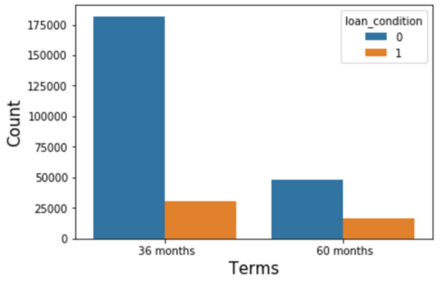
Lastly, we removed all leak variables which directly related to the loan condition. For example, the loans with higher total payment received are clearly less likely to be bad loans. These fields are another form of the representation of our labels.

D. *Exploratory Data Analysis*

We performed a series of EDA to further understand the dataset and interpret the importance of each remaining features towards the classification model. Based on domain knowledge and intuition, we want to explore how features like loan term, home ownership, number of opened credit account and dti performs against loan condition.

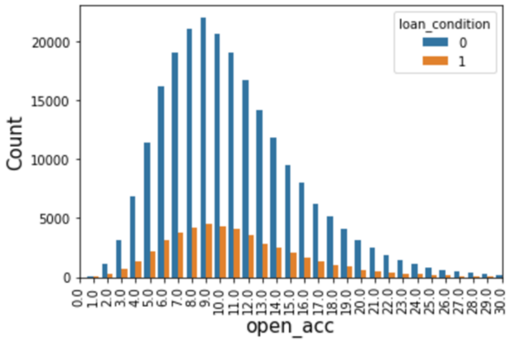
Not surprisingly, loans with a longer term tend to have a higher default rate as shown in Graph 1.

Graph 1: Loan Terms vs Loan Condition



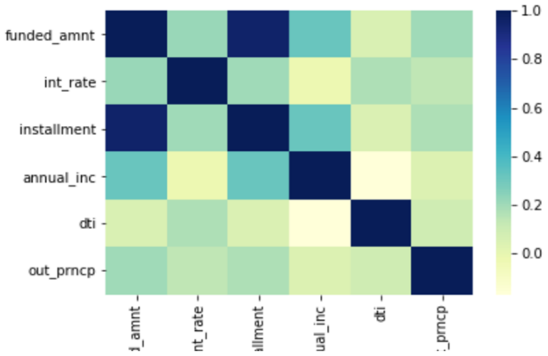
We also assume that number of opened credit account could be an indicator for loan default rate. Surprisingly, the default rate looks not affected by this feature. Graph 3 clearly shows that the distribution of good loan and bad loan against the number of opened account.

Graph 3: # of Opened Account vs Loan Condition



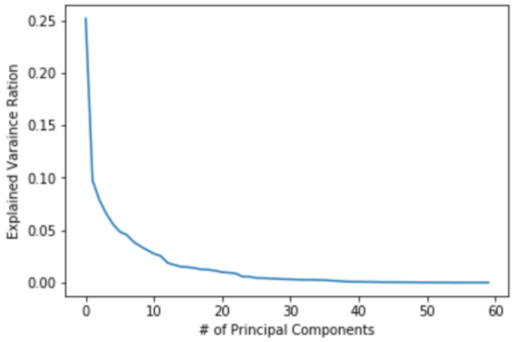
We studied the correlation between features. Based on the correlation heatmap. We removed highly correlated features prior to fit data into models.

Graph 4: Variable Correlation



Lastly, we perform a Principal Component Analysis. As shown in the graph below, we pick n=25 which we feel that is the minimum principal components to represents the most variance explained. We used PCA transformed data with n=25 to fit selected models. For better performance, we normalized all the numerical data in the dataset such as loan_amnt, int_rate, dti, etc. In order for the features to be interpretable for all the selected models, we performed one-hot-encoding over categorial variables. Cross validations are performed for each model to find the best hyper-parameter.

Graph 5: PCA Curve



III. METHODOLOGY

Our initial focus was on comparing and contrasting different machine learning models, in order to see how they perform on predicting the default loans using the loan features input. The models we selected in this step are Naïve Bayes, Logistic Regression and Random Forest. However, since we have an imbalanced dataset, which the ratio of bad and good loan is 1:5, we faced the challenge of handling imbalanced data.

Given the natural of the above-mentioned machine learning models, we decide to use Naïve Bayes model as our baseline model. After the preliminary run with the original imbalanced data, we noticed that Regularized Logistic Regression and Random Forest do achieve a better performance than the baseline model. We decided to take a step further than just evaluate the model performance, but research on how different balancing techniques affect the performance of different machine learning algorithms. We selected the following balancing techniques to perform our analysis:

1. Up-sampling: increase the sample size of the default loan data to match the size of good loans by resampling from the default loan data.
2. Down-sampling: decrease the size of the good loans by randomly sample a subset of the good loans to match the number of default loans.
3. SMOTE: short for Synthetic Minority Over-sampling Technique, which is a generalized up-sampling method. In the feature space of the default loans (which is the minority), this technique would create k nearest neighbors around the existing data point until the minority class matches the majority (good loans) in size.

We performed the following tasks to analysis our research idea in a logical flow:

1. *Clean data as mentioned in previous section*
2. *Performed PCA transformation with 25 principal components*
3. *Split train and test data (80%, 20%)*
4. *Fit the data to three models to evaluate the performance over un-balanced data*
5. *For each model, balance train data using three different approaches, perform cross validation to find the best hyperparameter, compare the model performance across the three balancing techniques on test data*
6. *Use extreme un-balanced dataset to further test the model performance across different balancing methods (see below for a detailed elaboration)*
7. *Perform Mann-Whitney-Wilcoxon test to conclude on which balancing is better for a specific model*

A. *Modeling*

Before applying any balancing techniques, we fine-tuned our original models using 5-folds cross validation to reach the best model on the training set. For Naïve Bayes model, we choose alpha = 0.01 from the range of (0.01 – 1.00) with an increment of 0.1. For Logistic Regression, we choose C = 10 from the list of values (1, 10, 100, 1000, 1000). For Random Forest, we choose max depth of 19, from range (12 – 25) with an increment of 1.

B. *Up-sampling*

The number of bad loan records in out train data is 47K, whereas the number of good loan records is 175K. The ratio is approximately 1:5.

The first approach we tried is to up-sample the minority class of the data, which is bad loan. We used to “resample” function in Python sklearn package to achieve this. The algorithm behind this method is that extra data records of bad loans are randomly selected to match the number of records in majority class. However, downside of this approach is that the records are simply duplicate of existing data points in the minority class.

C. *Down-sampling*

The second approach we used is the opposite of over-sampling: we downsize the majority class to match the number of records in minority class. In this case, one fourth of the good loan records was removed from the train data to match with the bad loans.

This is considered extremely dangerous for several reasons. First, a great number of information is lost due to the big decrease in our entire training data. Second, variance in the good loan data points will also not be captured by the models after down-sampling.

D. *SMOTE*

The last approach we took was to use Synthetic Minority Over-sampling Technique (SMOTE). It is intuitively similar to the robust over-sampling technique we tried. The difference is that the extra data points are randomly chosen from the k nearest neighbors, instead of randomly duplicating the existing minority data points. By doing this, more variance among the minority data is generated, and the final train data is more synthetic. [3].

E. *Compare and Comtrast*

In the context of our dataset and solutions seeking, we assume that SMOTE methodology will best fit our model. The reasons are: our dataset has reasonable size to handle up-sampling, and we want to capture more information on the minority class versus the majority class. Comparisons are shown below.

Method	**Pros**	**Cons**
Up-Sampling	- Good application to work with small dataset.	-Make exact copy of the data points may change the underline structure of the original dataset and cause overfitting towards "noise". - Not practical for very large dataset. Computation nightmare.
Down-Sampling	- When the size of the highly imbalanced dataset is huge, decrease in train data is a necessary evil.	- Increase the chance of overfitting. Especailly true for complex model and small dataset. - May remove potential useful information. - Sample size is reduced.
SMOTE	- The original distribution is not affected and introduced variance to the dataset.	- Assume nearest neighbor has the same label which might not be true. - Not applicable to categorical data.

F. *Extreme Method*

We designed our research into two stages. And we have the consideration about what if the preliminary results do not provide a significant enough difference on performance of different sampling techniques. The second stage would be zoom in and focus on handling un-balanced data and make it more un-balance to an extreme level to better study the effect of sampling techniques.

From the original dataset, we took 10 subsets of the bad loans and combine with all the good loans respectively to make 10 new datasets. The ten subsets of bad loans are in same size and mutually exclusive. We calculated the subsite size to make sure each new dataset has exactly 0.1% of bad loan.

Again, we apply the three balancing techniques. For a given model we compare the performance of three different balancing methods as well as the baseline which is the performance on un-balanced dataset. The reason to generate 10 datasets is to generalize our result, one result might not be promising. We would find an average score and the standard deviation of the performance over the ten datasets.

Lastly, to evaluate on how the performances differ from each method; or is there any of the techniques out perform the others on a specific machine learning model; or is the sampling method truly improved the performance against the baseline, we carry out Man-Whitney-Wilcoxon test to perform a series of hypothesis tests in order to reach a conclusive comparison.

IV. Evaluation

In this paper, we explore on modeling prediction of default loan using Naïve Bayes, Logistic Regression and Random Forest. We referred Naïve Bayes as our baseline model and compared with the other two models. We found out that the Random Forest and Logistic Regression gives better performance compared to Naïve Bayes in terms of F1-scores and AUC. As discussed earlier, we also tried to conclude how different data-balancing techniques will impact our model. The evaluation will focus on the second stage of our research and using un-balanced data performance as the baseline of analyzing the impact of balancing techniques on each model respectively.

A. Model Performance on Un-balanced data

a) Evaluation Metrics: F1-score, AUC score

Without hesitation, we decided to not to use accuracy as an evaluation criterion. The reason is our test data is skewed in the sense that even if our model would classify all the data into one class (the majority class), the accuracy of the model would still be high. Therefore, we looked at F1-score and AUC ROC Curve that better describe the performance according to the precision and recall of the model.

b) Performance

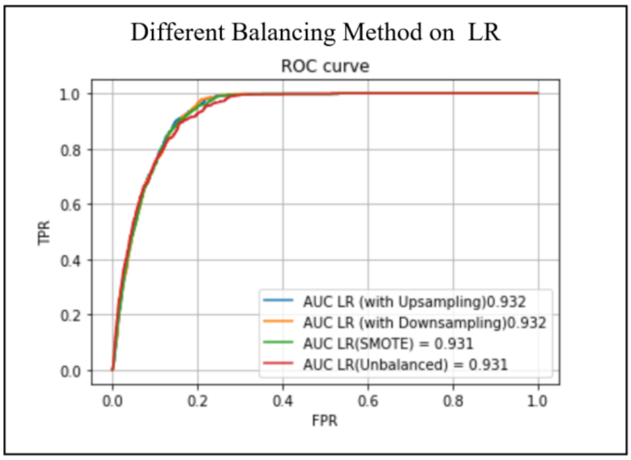
Among all the models, Naïve Bayes has the lowest average f1 and AUC score, which is 0.7 and 0.65. The Logistic Regression gives us 0.79 F1 and 0.9 AUC score, and Random Forest scores 0.84 in F1 and 0.9 in AUC. Since the cost of not detecting the bad loan and falsely report a good loan as a bad loan are very different in our context, we want to look closer at what these metrics mean. More specifically, we agreed that the cost of missing a potentially bad loan is more expensive in the business setting we simulate.

In the Logistic Regression and Random Forest model, AUC are both high with decent F1 score. We suspect it is possible that the high AUC score and low F1 score is a result of the skewed test data. Since we have much less bad loans compared to the good loans in the test data, problems with low recall can occur. In order to achieve a high F1 score, we will need to score high on both recall and precision. Also, the high AUC score can be a result of large false positives instead of large true positives.

B. Impact of Balancing Techniques

In this section, we will discuss the performance of different balancing techniques on a specific model in detail. As discussed in earlier sections, we have 10 sets of training data, which shares the same data with label of “good loan”. The “bad loan” in the 10 datasets are mutually exclusive. For each balancing technique on a specific model, we ran the model 10 times with the 10 datasets to reach a more generalized result. For each run, cross validation is performed to find the best hyperparameter. For up-sampling and SMOTE, we increased the bad loans to match the good loans (163K for each label). For down-sampling, we decreased the good loan to 1860 to match the 0.1% of bad loan.

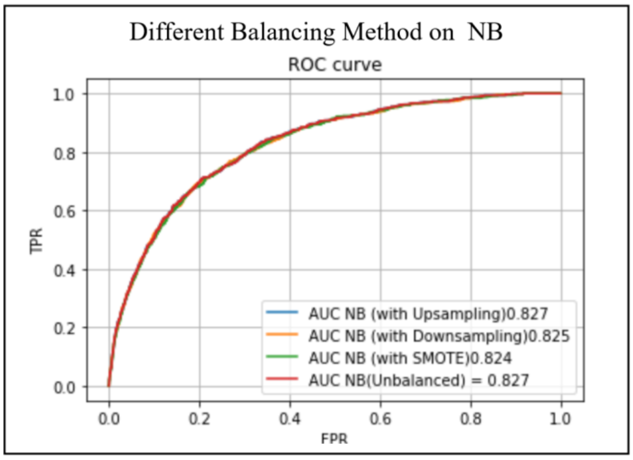
a) The performance of different balancing method on Logistic Regression.



For up-sampling and down-sampling the 5 folds cross validation has the best C as 100. For SMOTE, the best C is 10000. For un-balanced data, the best C is 10.

As shown in the graph, the difference between sampling techniques are not significant based off AUC ROC. Down-sampling and up-sampling share the same score of 0.932. SMOTE and un-balanced data share the same score of 0.931. We will have the comparison of F1 score in later section.

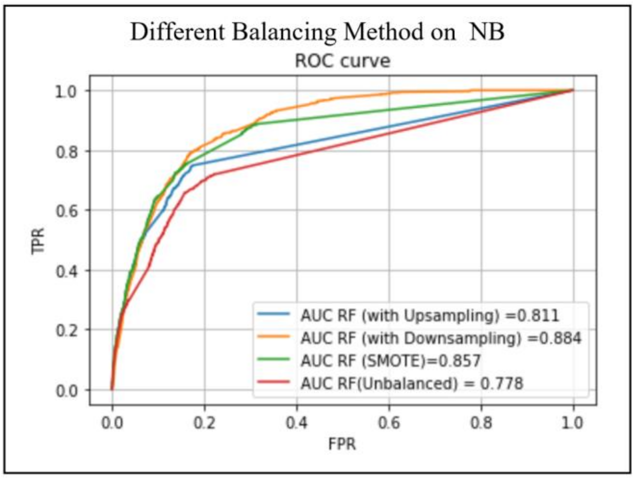
b) The performance of different balancing method on Naïve Bayes.



For all sampling techniques, the 5 folds cross validation has the best alpha as 0.01.

For Naïve Bayes, up-sampling and un-balanced gives very similar result with AUC score of 0.827. And surprisingly, SMOTE provided a worse performance which has a score of 0.824. This might not be an accurate reflection of how each of those sampling techniques preform over the dataset since Naïve Bayes gave a significant worse performance as compare to other two models. We believe that the model itself is too basic which does not have the ability to correctly classify the loan default for this dataset. Though, the difference between different balancing techniques is minor or even negligible on the AUC score.

c) The performance of different balancing method on Random Forest.



For up-sampling and SMOTE, the 5 folds cross validation has the best depth of 24. For down-sampling, the best depth is 14. For un-balanced data, the best depth is 19.

As tree models is comparatively more sensitive to the underline data structure. It is not surprising to see the deviation in AUC ROC. Down-sampling provide the best score of 0.884, followed by SMOTE which has a score of 0.857. Up-sampling comes third with 0.811. And the un-balanced data has the worst performance which has an AUC ROC of 0.778.

C. *Mann-Whitney-Wilcoxon Test on F1-Score*

The natural of AUC ROC can be intuitively understand as, how well the test separate the two label groups being tested into good loan and bad loan category. But in the calculation process of AUC ROC, it involves random selection of data points from the each of the label. Since the involvement of randomness, sampling technique might not affect AUC ROC significantly. So, we performed an in-depth Mann-Whitney-Wilcoxon test to figure our how exactly balancing techniques differ the F1-score. [4].

We gather the F1-scores for the 10 trials and computed the mean F1-score and standard deviation for each sampling technique on each model respectively.

F1 Score				
Model	**Up-Sampling**	**Down-Sampling**	**SMOTE**	**Unbalanced**
Logistic Regression	0.890 ± 0.008	0.983 ± 0.002	0.890 ± 0.018	0.889 ± 0.008
Naive Bayes	0.855 ± 0.022	0.871 ± 0.043	0.863 ± 0.022	0.852 ± 0.022
Random Forest	0.983 ± 0.001	0.983 ± 0.001	0.976 ± 0.002	0.883 ± 0.006

We fit the F1-score of the 10 trials into Wilcoxon test with the null hypothesis of X ≤ Y, and alternate hypothesis of X > Y (Sampling technique X gives better performance than sampling technique Y). If the p-value is less than 0.05, we would reject the null hypothesis. The three tables below are the test result for each machine learning model. X is represented by the left first column of each table. Y is represented by the top second row of each table.

Logistic Regression (F1)				
	Up-Sampling	Down-Sampling	SMOTE	Unbalanced
Up-Sampling		0.99	0.863	0.425
Down-Sampling	0.01		0	0
SMOTE	0.137	1		0.093
Unbalanced	0.575	1	0.907	

Naive Bayes (F1)				
	Up-Sampling	Down-Sampling	SMOTE	Unbalanced
Up-Sampling		0.95	0.94	0.213
Down-Sampling	0.05		0.5	0.02
SMOTE	0.06	0.5		0.023
Unbalanced	0.787	0.98	0.977	

Random Forest (F1)				
	Up-Sampling	Down-Sampling	SMOTE	Unbalanced
Up-Sampling		0.192	0	0
Down-Sampling	0.808		0	0.01
SMOTE	1	1		0.01
Unbalanced	1	0.99	0.99	

The table is diagonal paired. Each pair cross the grey lien add up to 0. The highlighted cells are those statistically significant cells to reject null hypothesis.

D. *Conclusion*

Logistic Regression: Down-sampling shows better performance than the other balancing techniques as well as un-balanced data.

Naïve Bayes: Down-sampling and SMOTE show better performance than un-balanced data. Down-sampling also fits better than up-sampling.

Random Forest: All three techniques show better performance than un-balanced. Down-sampling and up-sampling fit better than SMOTE.

References

[1] https://en.wikipedia.org/wiki/Peer-to-peer_lending

[2] K. Tsai, S. Ramiah, S. Singh "Peer Lending Risk Predictor".

[3] N. Chawla, K. Bowyer, L. Hall, W. Kegelmeyer, "SMOTE: Synthetic Minority Over-Sampling Technique".

[4] www.statstutor.ac.uk/resources/uploaded/mannwhitney.pdf

[5] Evaluation idea of sampling techniques shared by Prof. Leman Akoglu.