Forecasting Peer-to-Peer Lending Risk

Georgetown University
School of Continuing Studies

Certificate Program in Data Science

September 25, 2016

Team Members:

Archange Giscard Destine
ad1373@georgetown.edu
linkedin.com/in/agdestine

Steven L. Lerner
sll93@georgetown.edu
linkedin.com/in/sllerner

Erblin Mehmetaj
em1109@georgetown.edu
linkedin.com/in/erblinmehmetaj

Hetal Shah
hrs41@georgetown.edu
linkedin.com/in/hetalshah

Abstract

Peer-to-peer lending companies provide online platforms that can quickly pair borrowers seeking a loan with investors willing to fund the loan at an attractive rate. Since these loans are unsecured and companies creating the market generally do not invest their own capital, neither borrowers nor companies assume any risk. Entire credit risk is born by investors. Literature shows that credit risk depends upon borrower characteristics, loan terms and regional macroeconomic factors. To help investors identify unsecured loans likely to be fully paid, a machine learning algorithm was developed to forecast probability of full payment and probability of default. Training and input data consisted of historic loans' data from Lending Club and state level macroeconomic data from government and organizational sources. A logistic regression was shown to provide optimal results, effectively sequestering high risk loans.

Background

Peer-to-peer lending is a relatively new industry, created in 2007, consisting of firms that provide on-line platforms to link borrowers and investors. It has seen rapid growth, that is expected to continue, and is likely to become a major player in consumer financing. PwC has forecasted that the U.S. peer-to-peer lending industry could grow from $5.5 B in loans issued in 2014 to over $150B in loans issued in 2025, a 40% compounded annual growth rate (CAGR).[i] Lending Club, the market leader with 65% market share in 2014, has enjoyed over 100% CAGR over the last five years.[ii] Since its 2007 founding, Lending Club has facilitated over $16B in loans to over 1.3MM borrowers.[iii] Both lending club and the number two player in the market, Prosper, use similar business models.

The growth has been catalyzed by the overall value proposition. From the borrower's point of view, the loans are unsecured, interest rates are often below those charged by credit card companies, and the process can be done easily on-line. From the investor's point of view, they have the potential to earn rates of return (4 to 24+%) appreciably higher than other investment vehicles in this low rate environment and they are able to spread their risk by investing as little as $25 per loan. From the point of view of the firms making these markets, revenue is generated by charging an upfront fee from the borrower, nominally 5%, that is subtracted immediately from loan proceeds, and a service charge of roughly 1% that is subtracted from all payments made to the investor. However, it is important to emphasize that the only party assuming any risk in the transition is the investor and roughly 14% of loans end in default.

Objective

With the above in mind, our objective is to provide a data product that enables investors to avoid loans likely to default. This is achieved by developing a model that predicts probability of default for a potential loan given loan application data and various macroeconomic data.

Although there are several tools today that enable users to segment historical Lending Club data to determine which screens have performed better historically [iv, v, vi, vii], these tools have significant drawbacks. Some are tied to Lending Club loan grades, the basis and consistency of which are unclear. Some are tied to FICO scores which Lending Club has recently stopped providing in their historical loan database. All seem to ignore the importance of macroeconomic factors. The importance of macroeconomic factors, however, is clearly shown in Figures 1[viii] and 2[ix]. Figure 1 shows that unsecured personal loan delinquencies in the second quarter of 2016 varied by roughly a factor of 6, depending upon state. Figure 2 shows how an increase in unemployment from about 4.5 to 10% corresponded to an increase in credit card charge off rates from 4 to 11%.

Fig. 1: Unsecured Personal Loan Delinquencies,2Q16

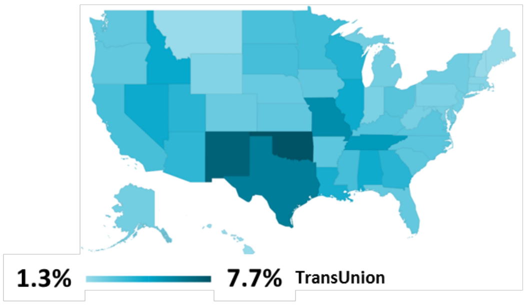
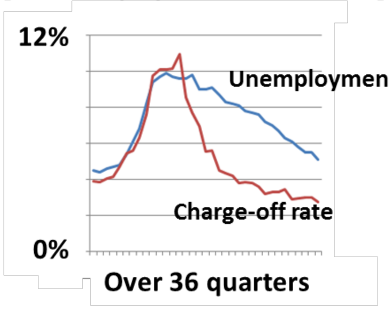


Fig. 2: Unemployment Rate and Charge-Off Rate



Data Selection

For historic loan data we used Lending Club's publically available, historic loan databases which range from 2007 through to 2016. For each completed loan, up to 111 features are provided. One notable change over the years is that FICO scores are no longer provided, making forecasting loan performance much more difficult. Appreciably wrangling required for this initial dataset is discussed below.

A summary of microeconomic data selected for inclusion in the dataset is shown in Table 1.

Table 1.

Measure	State	Federal	Value*	Slope**	Reflection of:
Unemployment	X		X	X	Job loss & replacement difficulty
GDP (per capita)	X		X	X	Overall economic activity
Disposable income	X		X	X	Cost/wage pressure
10-yr to 3-m T-bill spread		X	X		Future economic growth
3-yr T-bill rate		X	X		Short term inflation
Credit card rate (average)		X	X		Alternative borrowing costs
* Value is at loan origination ** Slope is linear least square fit slope for 12 months prior to loan origination					

Monthly unemployment data on the state level[x] was used as a measure of both likelihood of job loss and subsequent difficulty of job replacement. Both the value at time of loan origination and the value of the linear least square fit slope over the 12 months prior to loan origination were included. This later metric was selected based on expert input. Apparently when unemployment is rising and people feel their job may be at risk they take unsecured loans to be able to weather the likely impending economic challenges. However, this also leads to more defaults.

Per capita GDP (gross domestic product) on the state level[xi] was used as a measure of overall economic activity. Both value and slope were included. Since this data is only available on an annual basis, linear interpolation was used to determine monthly values.

Per capital disposable income on the state level[xii] was used to determine pressure on borrowers due to discrepancies between costs of critical items and wage growth. Similar to GDP, both value and slop were used and linear interpolation was required.

The spread between the 10-year and the 3-month U.S. Treasury bill rates[xiii] at the time of loan origination was used as a proxy for economist's view of future economic growth. This data was available on a monthly basis. Similarly, 3-year U.S. Treasury bill rate[xiv] was used as a proxy for short term inflation.

Average credit card interest rates[xv] were also included in the initial feature set. It was speculated that those borrowers who were willing to pay investors an interest rate above this rate were more desperate for credit and hence more likely to default.

Data Wrangling

Data wrangling was extensive, with most activities falling within six buckets: data verification; elimination of irrelevant features; eliminating non-completed loans; formatting; addressing null values, feature outliers and low information features; and merging loan and macroeconomic data and new feature addition.

Data Verification

After importing lending club data into a CSV data frame, it was verified that there were no duplicate or missing loan IDs and that there were no null values for features that would be used later to merge macroeconomic data (e.g., state, term, date). One surprising revelation from this initial overview was the small fraction of loan data that is actually verified by Lending Club. Looking at applicant's income, roughly 31% of incomes are verified while 46% are unverified and the remaining 23% only had income source verified.

Elimination of Irrelevant Features

Although Lending Club provides 111 features in their historic database, all but 29 were quickly eliminated. Since our objective is to provide a tool for potential investors, features that are not available to investors prior to making the investment decision are not relevant and were removed. In addition, several features were redundant; e.g., loan amount, funded amount and invested amount. Other entries were free form or ill defined; e.g., loan description, title, url, employee title, etc. Finally, zip code data was removed because we were not planning on using this level of granularity.

Elimination of Uncompleted Loans

For purposes of machine learning, we needed to restrict our focus to loans that were completed and hence the outcome – paid or defaulted – was known. Therefore, all 36 and 60 month loans issued within the last 36 or 60 months, respectively, were eliminated. Having done this, there were still a large number of loans that were not yet resolved and were still in the categories of 'current' or 'late'. Therefore, 36 month loans in the February to April 2013 timeframe and 60 month loans in the March and April 2011 timeframe also had to be eliminated. Finally, 10 loans still in the '31-120 late' category were assumed to default with principal loss as shown today.

Formatting

For purposes of analysis and machine learning, column feature entries had to undergo routine cleanup; e.g., removing labels such as months, years, %, etc. Also all dates had to be put in a consistent format to enable analysis and merging with macroeconomic data. Finally, ill-defined values for employment length – such as 10+ years, <1 years, etc. – were replaced with specific values based on analysis of employment distribution.

Addressing Null Values, Feature Outliers and Low Information Features

Two features were eliminated since the incidences of null values were more than 50%. Three other features were eliminated since they were always zero with a handful of exceptions (99.96+% zeroes). For other features, where the incidence of null values were extremely small (<0.05%) null values were replaced with mean values. One feature, borrower's income, covered an unexpectedly large range. Since it was felt that this may be a critical feature, a handful of outliers, incomes above $1MM, were eliminated. As none of these loans resulted in default, this would only tend to make any modelling results more conservative.

Merging Microeconomic Data and Feature Additions

Following the above wrangling, macroeconomic data was merged with the loan data frame, with macroeconomic data for each loan selected based on loan state and/or date. Four additional features were later added to this initial feature set; these new features were ratios or combinations of original features that were anticipated to be predictive. A complete list of features, with definitions, is attached as Appendix 1.

The net result of the above was a data frame consisting of 84,795 loans (instances), of which 11,440 resulted in default. The data frame had 36 features, of which 30 were potential modelling attributes, 1 was the target attribute (default or paid), 2 were associated with unpaid principal (in case we later wanted to model impact of default), and 3 were retained for later potential categorization ('grade', 'subgrade' and 'issue_d').

<u>Exploratory Date Analysis</u>

Exploratory data analysis focused on understanding which features are most relevant or most predictive for separating those loans likely to be paid from those loans likely to default. In Figure 3, all loans are plotted versus the feature set, with loans that were paid shown in aqua and loans that ended in default shown in gold. As can be seen in the figure, there is significant overlap and very little separation between the two classes of loans. The implication is that it will be difficult to separate paid and defaulted loans, even given this robust feature set.

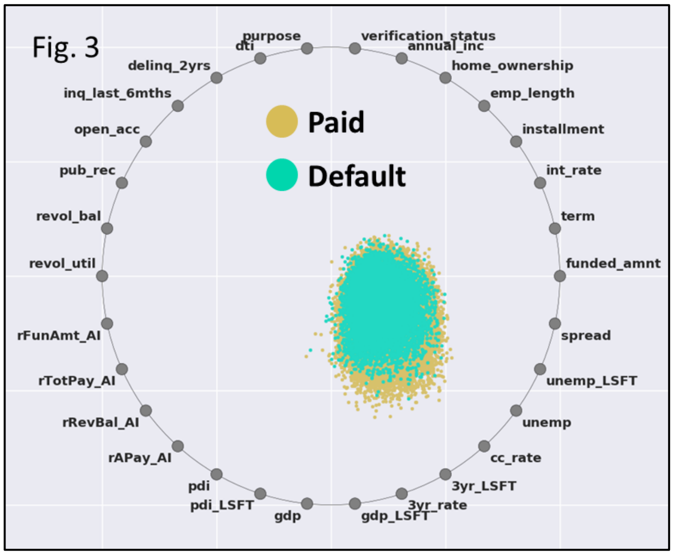
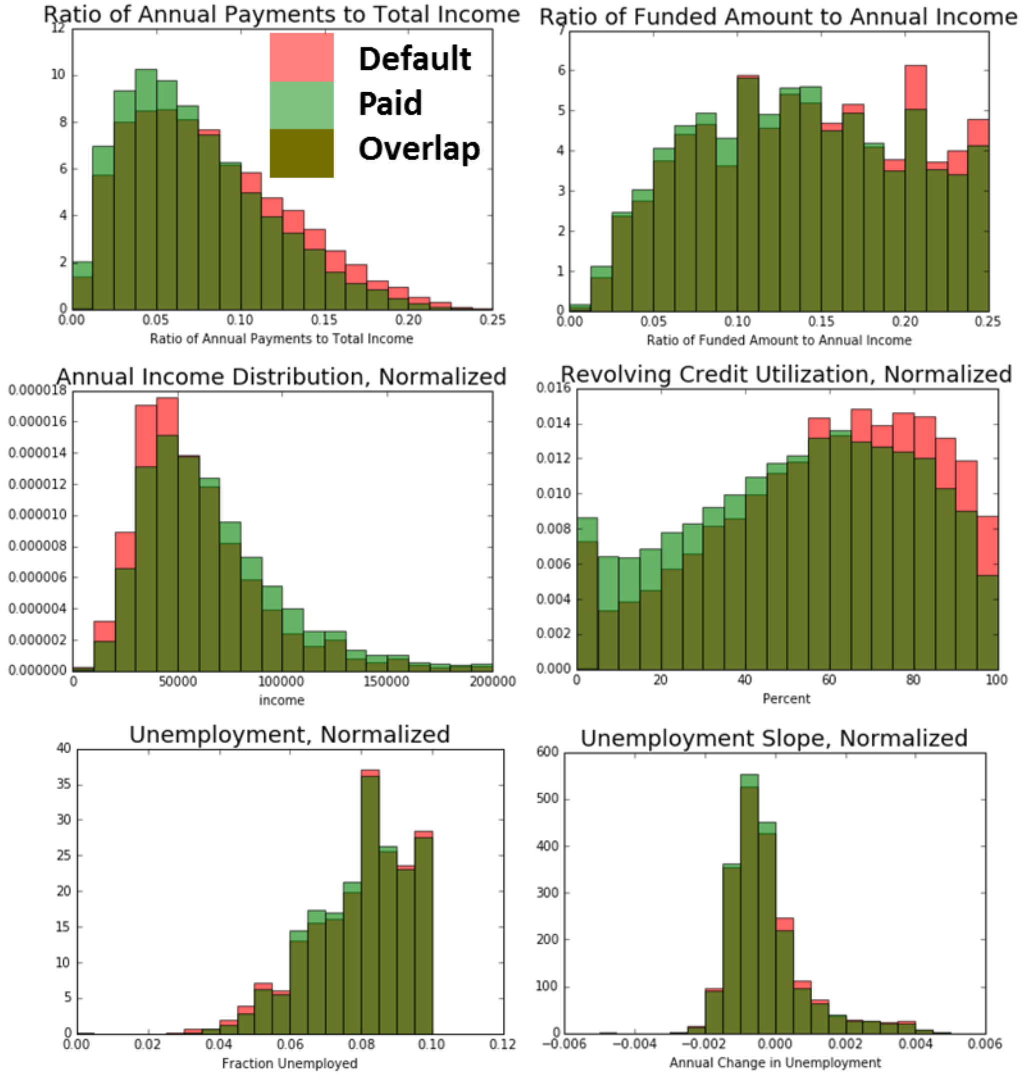


Fig. 3

This was further supported as we explored some of the individual features. In Figures 4(a) through 4(f) are plotted the distribution of key variables for both defaulted and paid loans. For example, in Figure 4(a) the distribution of the ratio of annual payments to annual income is plotted in red for defaulted loan and is plotted in green for paid loans. The areas of overlap are shown as dark olive green. As can be seen there is very little differentiation between the defaulted loan class and the paid loan class. This same theme is reinforced as we look at Figures 4(b) through 4(f) which plot distributions for: (b) the ratio of funded amount to annual income, (c) annual income, (d) revolving credit utilization, (e) state unemployment level, and (f) the slope of the state unemployment level 12 months prior to loan origination.

Figures 4(a) though 4(f)



Initial Model Screening

Since our primary objective is to provide a data product that enables investors to avoid loans likely to default, we set two model selection criteria:

1. A default recall of 90% or higher

Default recall is defined as the total defaults identified by the model divided by the actual total defaults. The model should be able to identify, and hence sequester, at least 90% of defaulted loans.

2. A paid precision of 90% or higher
   Paid precision is defined as paid loans identified correctly divided by the total instances identified as paid. Since we still want to provide investors with investment opportunities, but at lower risk, the model needs to provide a paid precision exceeding 90%.

Using SciKit Learn[xvi], we initially explored logistical regression and several classification algorithms to achieve these outcomes:

Logistic Regression

Logistic regression (David Cox, 1958) can be the appropriate prediction model to use when the dependent variable can assume binary values based on one or more categorical or continuous independent variables; in our case, whether the loan will "default" or be "paid" based on lender, loan and macroeconomic inputs. The output of a logistic regression model is the probability of the dependent variable taking on each of the potential binary outcomes; e.g, the probability of the loan defaulting is 65% and the probability of it being paid is 35%.[xvii,xviii]

Random Forest

The random forest (Breiman, 2001) classifying algorithm works as a large ensemble of decorrelated decision trees which provide a prediction value. It is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve predictive accuracy as well as control over-fitting. The sub-sample size is always the same as the original input sample size, with the default being that samples are drawn with replacement.[xix]

Naïve Bayes

Naive Bayes is a classification algorithm based on Bayes' rule and a set of conditional, independent, naïve assumptions regarding features.[xx] Gaussian and Bernoulli naïve Bayes algorithms were explored; multinomial naïve Bayes could not be considered due to negative values for some features e.g. unemployment slope.

K-Nearest Neighbor Classifier

K-nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance function).[xxi]

Gradient Boosting Classifier

The Gradient Boosting Classifier belongs to the ensemble category. It uses many small/weak classifiers and tries to improve performance using an iterative process to fit the data. After each iteration, the weights of correctly predicted instances are reduced and the weights of misclassified instances are increased. As iterations proceed that performance of the classifier should improve. As you can imagine, implementing this classifier requires deep hyper-parameter tuning, including the number of weak classifiers to use and the maximum number of iterations. The strength of this classifier is that performance shouldn't be affected by bias or variance in the data.[xxii,xxiii]

Voting Classifier

A voting classifier runs several models and weighs each model's forecast based on specified weights. In our case, we ran this model in the "soft" mode which produces an average weighted probability of default, with weights set by the user.

A pipeline was created to run the above models, with the first step being an encoder to transform any categorical values in the data frame to numerical values. All models were initially run with default parameter settings. For this first pass, none of the models met our performance hurdles. Logistic regression and random forest performed best, although the respective classification approaches were markedly different. Figures 5 and 6 depict the relative feature importance for random forest classification and logistic regression, respectively. For random forest, annual income was clearly the most important feature, by a wide margin; for logistic regression, annual income was ranked as relatively unimportant.

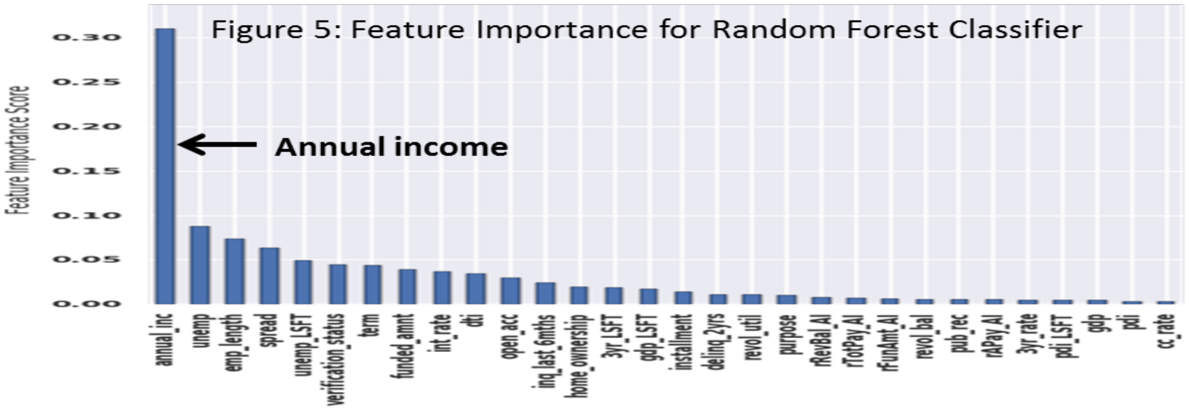
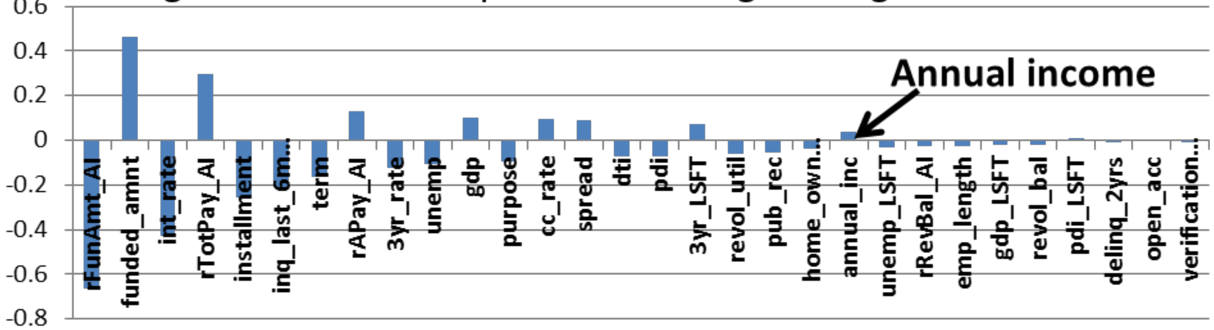


Figure 5: Feature Importance for Random Forest Classifier

Annual income

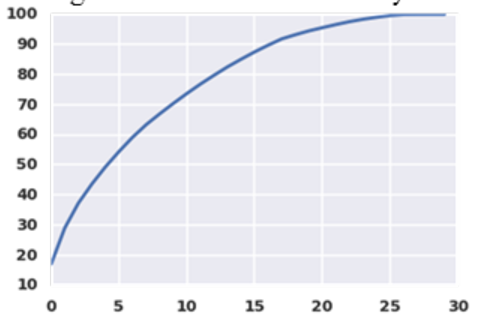
Figure 6: Feature importance for logistic regression


Annual income

For naïve Bayes, test scores were better for Bernoulli compared to Gaussian algorithms. However, both performed appreciably poorer than logistic regression or random forest. K nearest neighbors also did not perform well, even given some initial optimization around K.

In light of the above, several modifications to the data pipeline were made to improve performance. Upfront feature standardization was added. Standardization rescales features so they are centered around 0 with a unit standard deviation. This frequently improves the performance of machine learning algorithms, eliminating the impact of individual feature scale. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) were added for feature reduction for unsupervised models and supervised models, respectively. Figure 7, shows that 99.9% of the variance can be explained by 24 principal features. Hyper-parameter tuning using GridSearch was also added to optimize hyper-parameter selection for each model. In addition, one new feature (total debt to income, including the current loan amount) was added. These steps improved model performance somewhat, but did not get us close to our performance hurdles. A large remaining issue to address, however, was the unbalanced nature of our dataset.

Fig. 7: Results of PCA Analysis



Data Set Balancing

An unbalanced dataset is when one class is represented by a large number of instances while the other class is represented by appreciably fewer instances. The Lending Club dataset is inherently unbalanced. After data wrangling, the data frame consists of roughly 85,000 loans which fall in the distribution shown in Table 2. Only 13.5% of the total loans are classified as 'default' and 86.5% are classified as 'paid'. Standard machine learning methods generally work best when the classes are balanced roughly 50% - 50% and perform poorly when trying to identify the minority class. However, this is exactly what we are trying to do – we are trying to predict when loans will default. We can easily understand why models have difficultly identifying the minority but relevant class. In our case, this class imbalance forces the classifiers to almost always predict that all loans will be 'paid'. This enables the model to achieve an overall precision of over 80%, albeit with a very low (roughly 0%) recall for the defaulted loans (the minority class). Without addressing the dataset imbalance issue, classifiers will generally fail to detect bad loans.

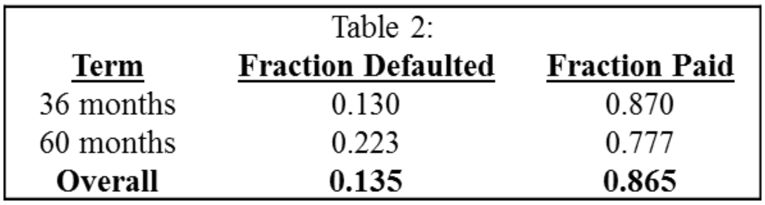


Table 2:

Term	Fraction Defaulted	Fraction Paid
36 months	0.130	0.870
60 months	0.223	0.777
Overall	**0.135**	**0.865**

Two approaches were considered to address the imbalanced class problem:

Under-sampling

Under-sampling, also called down-sizing, involves randomly deleting instances of the majority class until the two classes are in equal proportions. In our case this would involve randomly deleting ‘paid’ loans until the number of ‘paid’ and ‘default’ loans were equal. The drawback of this approach is that much information is lost by deleting ‘paid’ loans.

Over-sampling

Over-sampling involves creating more instances of the minority class until this class contains as many instances as the majority class. Potential oversampling techniques include: (i) duplicating the rare class instances[xxiv] or (ii) synthetically creating more instances of the rare class (e.g., with the SMOTE algorithm) [xxv]. After trying both approaches, the former was used in deference to its simplicity and transparency.

One issue with oversampling is that the model gets to “see” several minority class instances more than once, and hence, unless a rigorous test protocol is put in place, performance metrics could be artificially inflated. With this in mind the below test protocol was adopted.

Test Protocol

Model training and testing followed the below steps:

1. Approximately 1% of the original loan data was randomly extracted. This data set consisted of ‘paid’ and ‘default’ loans in a ratio approximately equal to the original, unbalanced ratio.

2. ‘Default’ loan instances were multiplied roughly six-fold, until the number of ‘default’ instances were roughly equal to the number of ‘paid’ instances.

3. The resulting data frame was randomly divided into two data frames, with one consisting of 80% of the instances and the other consisting of 20% of the instances.

4. The model was trained using the 80% data frame

5. The model was first tested using the 20%, artificially balanced data frame

6. The model was then tested using the 1% data frame, extracted during step 1 above. This is the most meaningful test since this data is unbalanced, and none of the instances have been previously “seen” by the model.

Prior to re-running all models using the above protocol, we decided to also manually optimize the one last remaining lever at our disposal, class weight, for logistic regression and random forest classifiers.

Class weight

According to the AAAI 2000 Workshop on the imbalanced class problem, imposing higher class weights (sometimes called class penalties or misclassification cost) to the minority class can improve classifier performance (for that class)[xxvi]. Class weights for ‘default’ and ‘paid’ initially start out at 0.5 ‘default’ and 0.5 ‘paid’, summing to 1. By increasing the ‘default’ weight, and hence reducing the ‘paid’ weight, the model is forced to pay greater attention to the ‘default’ category and reduce the likelihood that any ‘defaults’ will be missed.

Results

The results of implementing the above strategies are summarized below:

Logistic Regression

Logistic regression performed best when the penalty input argument was set to ‘l2’ and the class weights were set to 0.7 and 0.3 for ‘default’ and ‘paid’ classes, respectively. The model was trained on the 80% dataset with all 30 data features. When the model was run on the 20% test dataset, a ‘default’ recall of 94% and a ‘paid’ precision of 77% were achieved.

Table 3: *Performance of Logistic Regression Model on Test Dataset*

Logistic Regression	Precision	Recall	F1 Score	Support
Paid	0.77	0.20	0.31	14,547
Default	0.52	0.94	0.67	13,568

Most importantly, however, when we ran the model on the unseen dataset, we achieved a ‘default’ recall of 97% and a ‘paid’ precision of 97%, well above our performance targets.

Table 4: *Performance of Logistic Regression Model on Unseen Dataset*

Logistic Regression	Precision	Recall	F1 Score	Support
Paid	0.97	0.18	0.30	734
Default	0.16	0.97	0.20	115

Random Forest

Following PCA for unsupervised dimensionality reduction, the random forest model was trained on 24 out of 30 data features. The model was fine-tuned by choosing a depth of 6 and class weights of 0.6 and 0.4 for ‘default’ and ‘paid’, respectively. When the model was run on the test dataset, a ‘default’ recall of 92% and a ‘paid’ precision of 77% were achieved.

Table 5: *Performance of Random Forest Model on Test Dataset*

Random Forest	Precision	Recall	F1 Score	Support
Paid	0.77	0.25	0.38	14,547
Default	0.53	0.92	0.68	13,568

When the model was run using the unseen dataset, the ‘default’ recall was 95% and ‘paid’ precision was 97%, both above our performance hurdle, but slightly worse than results achieved using logistic regression.

Table 6: *Performance on Random Forest Model on Unseen Dataset*

Random Forest	Precision	Recall	F1 Score	Support
Paid	0.97	0.24	0.39	734
Default	0.16	0.95	0.28	115

Naive Bayes

PCA was used to reduce the number of data features from 30 to 24 and the naive Bayes model was run using Bernoulli probability distributions. We also used a Laplace smoothing of 1 and binarized features and values. Running the model on the test dataset, a 'default' recall of 60% and a 'paid' precision of 61% were achieved.

Table 7: *Performance of Bernoulli Naive Bayes Model on Test Dataset*

Naive Bayes	Precision	Recall	F1 Score	Support
Paid	0.61	0.60	0.60	14,547
Default	0.58	0.60	0.59	13,568

Corresponding performance on the unseen dataset was 'default' recall of 70% and 'paid' precision of 93%.

Table 8: *Performance of Bernoulli Naive Bayes Model on Unseen Dataset*

Naive Bayes	Precision	Recall	F1 Score	Support
Paid	0.93	0.59	0.72	734
Default	0.21	0.70	0.32	115

Gradient Boosting

For the test dataset, the gradient boosting classifier achieved a 'default' recall of 62% and a 'paid' precision of 64%.

Table 9: *Performance of Gradient Boosting Classifier on Test Dataset*

Gradient Boosting	Precision	Recall	F1 Score	Support
Paid	0.64	0.65	0.65	14,547
Default	0.62	0.62	0.62	13,568

When the model was run on the unseen dataset, the corresponding 'default' recall was 63% and the 'paid' precision was 91%.

Table 10: *Performance of Gradient Boosting Classifier on Unseen Dataset*

Gradient Boosting	Precision	Recall	F1 Score	Support
Paid	0.91	0.62	0.31	734
Default	0.21	0.63	0.74	115

Voting Classifier

We initialized four classifiers -- logistic regression, random forest, Gaussian naive Bayes, and K-nearest neighbors -- and used these to initialize a soft-voting voting classifier with weights [2, 1, 1, 1]. When the model was ran using the test dataset, a 'default' recall of 97% and a 'paid' precision of 96% were achieved.

Table 11: *Performance of Voting Classifier on Test Dataset*

Voting Classifier	Precision	Recall	F1 Score	Support
Paid	0.96	0.65	0.77	14,547
Default	0.72	0.97	0.83	13,568

When we run the model on the unseen dataset, however, 'default' recall and 'paid' precision were reduced to 57% and 90%, respectively.

Table 121: *Performance of Voting Classifier on Unseen Dataset*

Voting Classifier	Precision	Recall	F1 Score	Support
Paid	0.90	0.64	0.75	734
Default	0.20	0.57	0.29	115

Conclusion

Both logistic regression and random forest classifier models were successfully constructed that can effectively help investors avoid loans likely to default while identifying loans likely to be fully paid. The logistic regression model required a penalty input of l2 and class weights of 0.7 and 0.3 for 'default' and 'paid' loans, respectively, and provided a 'default' recall of 97% and a 'paid' precision of 97%. The random classifier model required PCA dimensionality reduction, depth of 6, and class weights of 0.6 and 0.4 for 'default' and 'paid' classes, respectively. This provided a 'default' recall of 95% and a 'paid' precision of 97%. These high performance metrics came at the expense of lower 'default' precision; that is some 'paid' loans were sequestered along with the 'default' loans. This was acceptable in light of our objective to reduce investor write-offs. Other models tested -- naive Bayes, gradient boosting, and voting classifier did not perform nearly as well.

Potential Future Work

1. Perform logistic regression on the entire dataset followed by random forest classification on potential 'default' loans in hopes of increasing 'default' precision while maintaining very high 'default' recall and 'paid' precision
2. Identify/create other features that might help in improving the overall model performance
3. Create a tool for aggressive investors to forecast the impact of default (think percent of principal lost)
4. Develop improved approaches to modelling highly imbalanced data

Appendix 1

Table 1	
Feature	**Definition**
funded_amnt	Total funded loan amount
term	Term of the loan: either 36 months or 60 months
int_rate	Interest rate on the loan
installment	The monthly payment owned by the borrower
grade	Lending Club assigned loan grade: A through G
sub_grade	Lending Club assigned loan sub-grade: 1 through 5 (A1, A2, A3, A4, A5, B1, etc.)
emp_length	Employment length in years
home_ownership	Home ownership status provided by the borrower; potential values are: RENT, OWN, MORTGAGE, OTHER.
annual_inc	Annual income provided by the borrower
verification_status	Indicates if income was verified by LC, not verified, or if the income source was verified
issue_d	The month and year the loan was funded
loan_status	Since the data frame contains completed loans, only options are paid off or default (target variable)
purpose	A category provided by the borrower for the loan request.
dti	The borrower’s total monthly debt payments on total debt obligations, excluding mortgage and the requested loan, divided by the borrower’s monthly income
delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years
inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
open_acc	The number of open credit lines in the borrower's credit file
pub_rec	Number of derogatory public records
revol_bal	Total credit revolving balance
revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit
total_rec_prncp	Principal received to date
net_default_amnt	= 'funded_amnt' - 'total_rec_prncp'
Unemp	State level unemployment rate as of the month and year of loan origination
unemp_LSFT	The value of the linear least square fit slope of the state level unemployemt rate over the 12 months prior to loan origination
Pdi	State level, percapita disposable income as of the month and year of loan origination
pdi_LSFT	The value of the linear least square fit slope of the state level percapita disposable income over the 12 months prior to loan origination
Gdp	State level gross domestic product as of the month and year of loan origination
gdp_LSFT	The value of the linear least square fit slope of the state level GDP over the 12 months prior to loan origination
Spread	The difference between the 10-year T-bill interest rate and the 3-month T-bill interest rate as of the month and year of loan origination
3yr_rate	The 3-year T-bill inteest rate as of the month and year of loan origination
3yr_LSFT	The value of the linear least square fit slope of the #-year T-bill interest rate over the 12 months prior to loan origination
cc_rate	The natinoal average credit card interest rate as of the month and year of loan origination
rFunAmt_AI	= 'funded_amnt' / 'annual_inc'

rTotPay_AI	= ('installment * 'term')/'annual_inc'
rRevBal_AI	= 'revol_bal'/'annual_inc'
rAPay_AI	= ('installment' * 12.0)/'annual_inc'
DTI*	Ratio of total debt to income, including the amount of this new loan

* added prior to changing feature weights

References

[i] https://www.pwc.com/us/en/consumer-finance/publications/assets/peer-to-peer-lending.pdf

[ii] https://www.lendingclub.com/info/demand-and-credit-profile.action

[iii] https://www.lendingclub.com/info/statistics.action

[iv] http://www.lendstats.com

[v] http://www.percube.com/lc/

[vi] http://www.interestradar.com/

[vii] http://cs229.stanford.edu/proj2014/Kevin Tsai,Sivagami Ramiah,Sudhanshu Singh,Peer Lending Risk Predictor.pdf

[viii] http://transunioninsights.com/IIR/

[ix] http://www.cardhub.com/edu/historical-credit-card-interest-rates/

[x] http://data.bls.gov/cgi-bin/dsrv

[xi] http://www.bea.gov/iTable/iTable.cfm?reqid=70&step=1&isuri=1&acrdn=2#reqid=70&step=1&isuri=1

[xii]http://www.bea.gov/iTable/iTable.cfm?reqid=70&step=1&isuri=1&acrdn=6#reqid=70&step=29&isuri=1&7022=21&7023=0&7024=non-industry&7001=421&7090=70

[xiii] http://www.federalreserve.gov/datadownload/Build.aspx?rel=H15

[xiv] http://www.federalreserve.gov/datadownload/Build.aspx?rel=H15

[xv] http://www.cardhub.com/edu/historical-credit-card-interest-rates/

[xvi] http://scikit-learn.org/stable/

[xvii] *Cox, DR (1958). "The regression analysis of binary sequences (with discussion)". J Roy Stat Soc B.* ***20****: 215–242.*

[xviii] *Walker, SH; Duncan, DB (1967). "Estimation of the probability of an event as a function of several independent variables".Biometrika.* ***54****: 167–178.*

[xix] *Breiman. (2001). Random Forests Machine Learning. Retrieved from https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf*

[xx] http://scikit-learn.org/stable/modules/naive_bayes.html

[xxi] http://scikit-learn.org/stable/modules/neighbors.html#neighbors

[xxii] http://scikit-learn.org/stable/modules/ensemble.html

[xxiii] https://en.wikipedia.org/wiki/Gradient_boosting

[xxiv] Ling, C. & Li, C. (1998). Data mining for direct marketing problems and solutions. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, New York.

[xxv] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority oversampling technique. Journal of Artificial Intelligence Research, 16, 321–357

[xxvi] CHAWLA, Nitesh V., JAPKOWICZ, Nathalie, et KOTCZ, Aleksander. Editorial: special issue on learning from imbalanced data sets. ACM Sigkdd Explorations Newsletter, 2004, vol. 6, no 1, p. 1-6.
MLA