Predicting Defaults of Personal Loans On Lending Club

Boya Yan, Mengmei Chen, Yuze Liu

May 17, 2016

Contents

1 Introduction

Over the past decade, the emergence and spread of peer-to-peer lending platforms have transformed the traditional loan business. Whereas in the past most households and businesses had to go through a long painful process and ask banks for loans when they need some extra money, now they have the alternative to apply their loans online through peer-to-peer lending platforms. On the other hand, investors can browse through the information borrowers provide when they apply for their loans and decide for themselves who to lend to and what amount they want to lend.

Lending Club is the worlds largest peer-to-peer lending platform today. It attracts borrowers by offering lower interest rates than traditional banks do, and it makes its profits by earning origination fees from borrowers and service fees from investors. However, like banks, it has the problem of information asymmetry, where they just cannot get the same amount of information on borrowers than the borrowers have for themselves. Unlike banks, almost all of its loans are uncollateralized, which adds another layer of risks to its business. Therefore, it is especially crucial for Lending Club to have a good screening mechanism to determine who is too risky to lend to. Our goal in this project is to use machine learning algorithms to predict defaults of personal loans for Lending Club, so that it will not only help Lending Club achieve a higher accuracy in detecting bad loans, but also increase the return of investments for its investors and decrease the cost of lending over time.

2 Problem Definition and Algorithms

2.1 Task

Our task is to use machine learning algorithms to predict defaults of personal loans for Lending Club based on the information on the borrowers.

2.2 Algorithms

We will learn several binary classifiers with Support Vector Machine, Logistic Regression, Decision Tree and Random Forest. We will use the libraries provided by Scikit learn and perform a grid search to find the best parameters for each classifier.

3 Data

3.1 Basic Structure

We obtained our data from Lending Club's website: `https://www.lendingclub.com/info/download-data.action` where it lists detailed personal information such as the borrowers ID, the amount of the loan, the employment length, and the credit score range. Finally, it tells us whether a loan was 'fully paid' or 'charged off'. We have around 220,000 data in total. However, the data is very imbalanced, with 87.3% fully paid and 12.7% charged off.

3.2 Preprocessing

3.2.1 Numerical/categorical Features

There are around 100 features in total, but we didn't use all of them.

First of all, since we have plenty of data, we discarded entries that contain "N/A" and empty fields because we think using real, valid data provides better results than filling in those fields with mean/mode of all data as an estimation.

Then, we removed some of the features that are empty for most or all of the data and features that are irrelevant to our prediction or very hard to make use like:

- Borrowers ID
- Next payment date
- Issue date of the loan
- Employment Title

Among the rest, we selected 22 useful numerical/categorical features like

- Employment length
- Home ownership status: Rent, Own, Mortgage, Other
- State: NY, NJ...
- Annual income
- Debt-to-income ratio

Finally, we transformed categorical features into boolean features and normalized all feature values to a range of [0,1]. After this, we got a feature vector with length 86.

3.2.2 Text Feature

We observed that every borrower could choose to write a short description on why they needed the loan and we processed the description into a numerical feature. First, we removed things like stop words and special characters. Then we kept the stem of words using stemming package. At first we did some natural language processing on the descriptions and selected 19 words to be binary features, but later we realized that doing this may not be very useful because most of the descriptions don't contain these 19 words. Hence, we decide to preprocess the descriptions like what we did in spam classification in homework 1. We made a vocabulary list from descriptions of all inputs and transformed them to features; we also think the length of descriptions is important so we added it to the features. Then we learned a probability for the description with logistic regression and appended the probability to the 86 non-text features.

Sample:

- before processing - Borrower added on 04/04/12 > I am seeking funds to remodel both of our bathrooms as well as with the kitchen.<br><br> Borrower added on 04/07/12 > My monthly expenses are around $1900.00 monthly. I have been with my current employer for almost 4 years. I understand the true meaning and purpose of a loan that is why I take pride in repaying a loan.<br>
- after processing - seek fund remodel bathroom well kitchen month expens around dollarnumber month current employ almost numbernumber year understand true mean purpos loan tak prid repay loan

3.2.3 Data Balancing & Splitting

We have mentioned before that our data is very imbalanced with 87.3% 'fully paid' and 12.7% 'charged off'. In order to improve the performances of our chosen models such as SVM and to gain more information from 'charged off' data, what we first did was balancing the data. Then we selected 15000 entries from the balancing data and split them into training, validation and testing set.

- Total: 15000
- Training Set: 9600
- Validation Set: 2400
- Test Set: 3000

4 Methodology

4.1 Evaluation Metrics

		True Value	
		True	False
Predicted Value	True	**TP**	**FP**
	False	**FN**	**TN**

- Accuracy: 1 - error rate
- Sensitivity: $\frac{TP}{TP + FN}$
- Specificity: $\frac{TN}{TN + FP}$

We evaluated our models based on these three metrics. Besides accuracy, we also looked at *sensitivity*, which is out of all the people who actually paid back their loans, how many of them we predict to pay back; and *specificity*, which is out of all the people who defaulted on their loans, how many of them we predict to default. Finally, we used ROC(receiver operating characteristic curve). The model with a bigger AUC(area under curve) is better.

4.2 Baseline

We also set a baseline accuracy and baseline AUC for our models. If our models achieve better performances than the baseline, we would consider our models effective.

4.2.1 Random Guessing

As our data is well-balanced, under random guessing, we will get around 50% accuracy and a straight ROC curve.

4.2.2 Lending Club Accuracy

However, we wanted to be more ambitious so we aimed to outperform Lending Club. Lending Club assigns each of its loan a grade ranging from A to F. To find out its accuracy and AUC, we can set different thresholds (e.g if the threshold is C, data with grade above C will be predicted 1 and data under C will be predicted -1).

Table 1: Accuracy for Different Threshold

Threshold	Score
A	0.5827
B	0.6109
C	0.5907
D	0.5541
E	0.5185
F	0.5

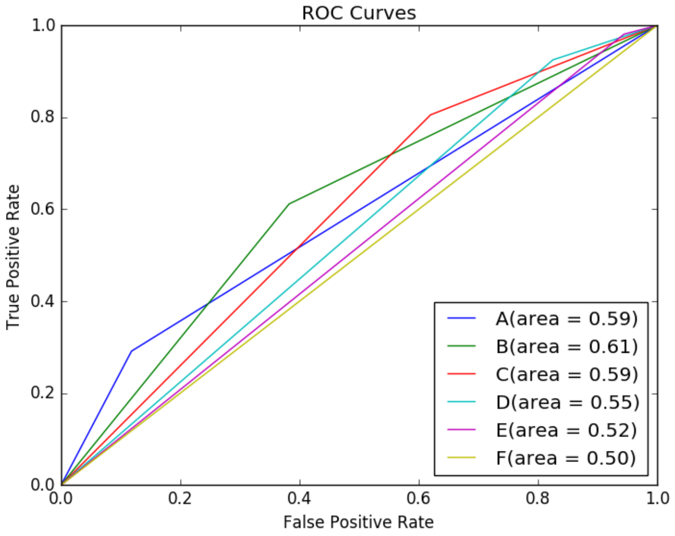


Figure 1: ROC Curve for Different Threshold)

We can see that we get the highest accuracy and the greatest AUC if we set the threshold at grade B, which is the best Lending Club currently can do. Thus, we set our baseline accuracy to be 61.1% and our baseline AUC to be 0.61.

5 Results

5.1 Different Algorithms

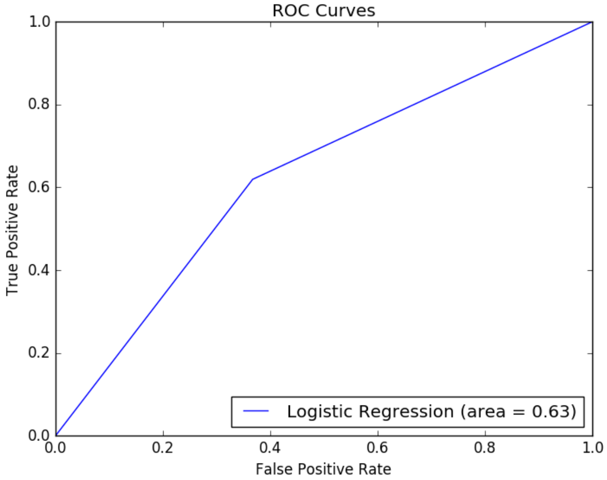
We tried different machine learning algorithms using sklearn and tuned the parameters with the validation set. We show the best result for each classifier in the followings.

5.1.1 Logistic Regression

According to what we learned in class, the objective is to minimize $l(w)$ where

$$l(w) = ln \prod P(y^i | x^j, w) \tag{1}$$

And by using Logistic Regression, we get an accuracy of 62.7% and an AUC of 0.63.



5.1.2 Support Vector Machine

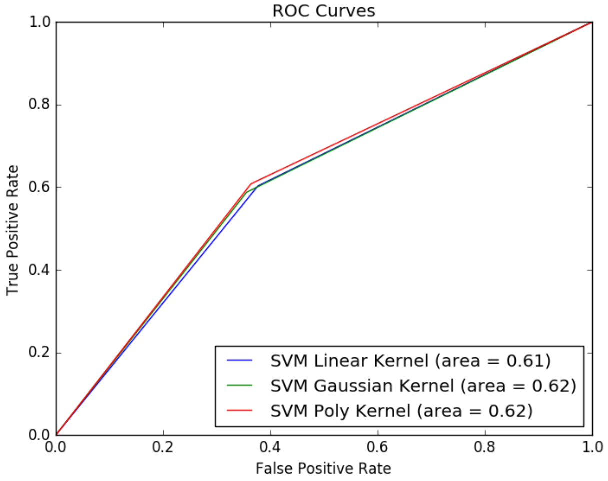
The objective of our SVM is to minimize:

$$L(w, b, \alpha) = \frac{1}{2}||\omega||^2 - \sum \alpha_i [y^{(i)}(\omega^T x^{(i)+b}) - 1] \tag{2}$$

Also, there exists different kernels like ‘poly’, ‘rbf’, and ‘sigmond’. By trying different kernels of SVM, we got the following results,

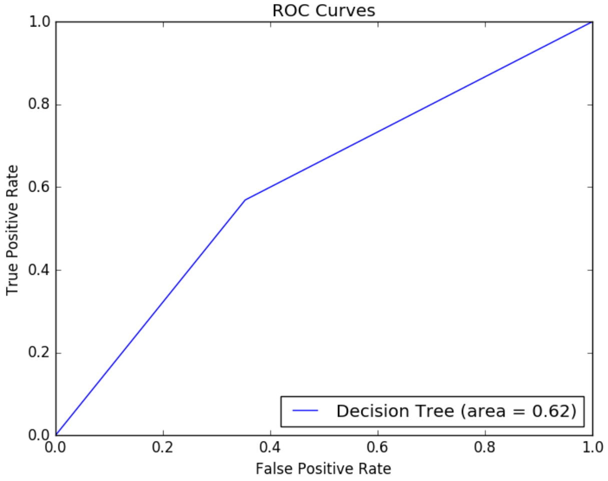
Table 2: Kernels vs Accuracy

Poly	RBF	Sigmond
61.6%	62.4%	61.3%



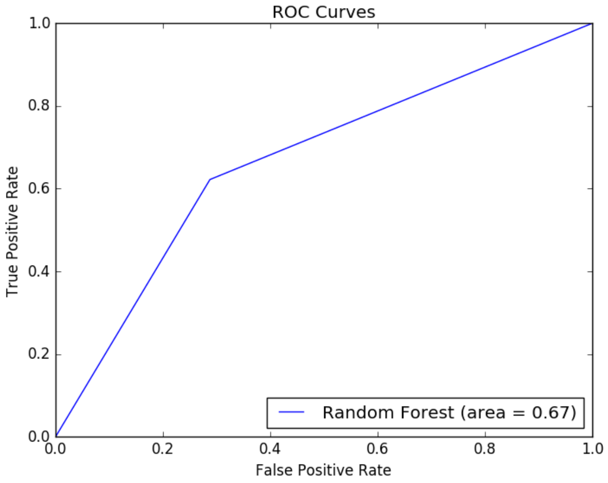
5.1.3 Decision Tree

Decision Tree is built by recursively splitting on the next best attribute.
By using the Decision Tree model, we got an accuracy of 62.13% and an AUC of 0.62.



5.1.4 Random Forest

Random Forest is an ensemble learning method and is constructed by a bag of decision trees.
By using Random Forest, we got an accuracy around 66.9% and AUC at 0.67.

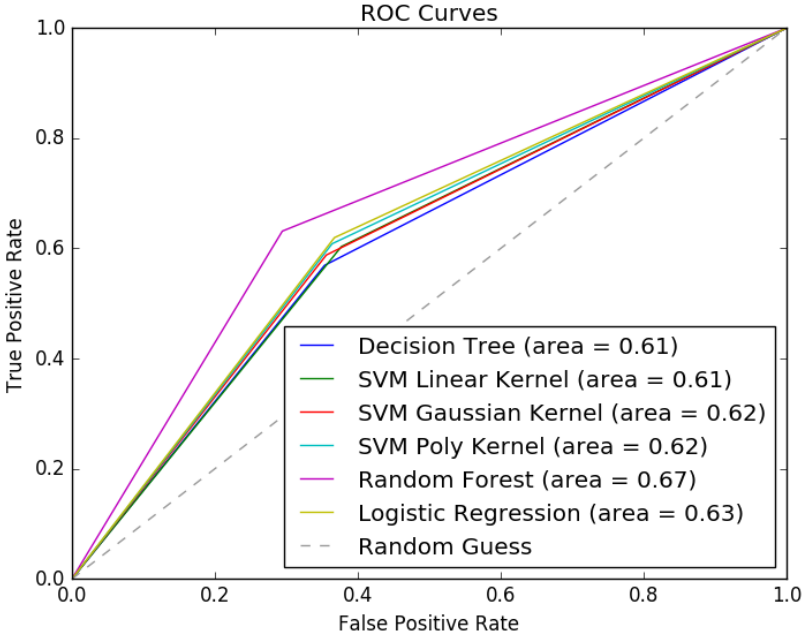


5.2 Comparison

The table and the graph below summarize the performances of all algorithms. Comparing different algorithms, we found out that Random Forest is the best of all. This is probably because its bagging method and its random choice of subsets of features make it more robust against overfitting.

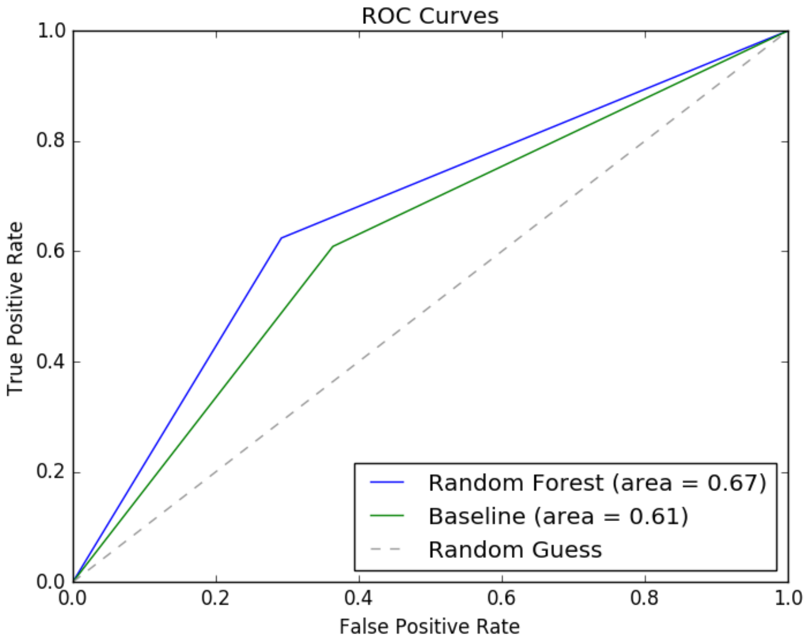
Table 3: Algorithms vs Accuracy

Logistic	SVM	Decision	Random Forest
62.7%	62%-63%	62.13%	66.9%



5.3 Final Evaluation on Test Set

Using Random Forest on our test set, we obtained an accuracy of **67.5%** and an AUC of **0.67**. As Lending Club has an accuracy of only 61.1% and an AUC of 0.61, we can see that our strategy is actually more effective.



.

6 Discussion

Some of our hypothesis weren't supported. For example, we thought the length of a borrower's description might reflect the seriousness of his will on paying back the money, but adding the field to our feature vectors didn't improve our result at all. We also thought that some categories of reasons for borrowing money might have a correlation with the chance of paying back. See Section 6.3 for the weights of words that our Logistic Regression classifier discovered.

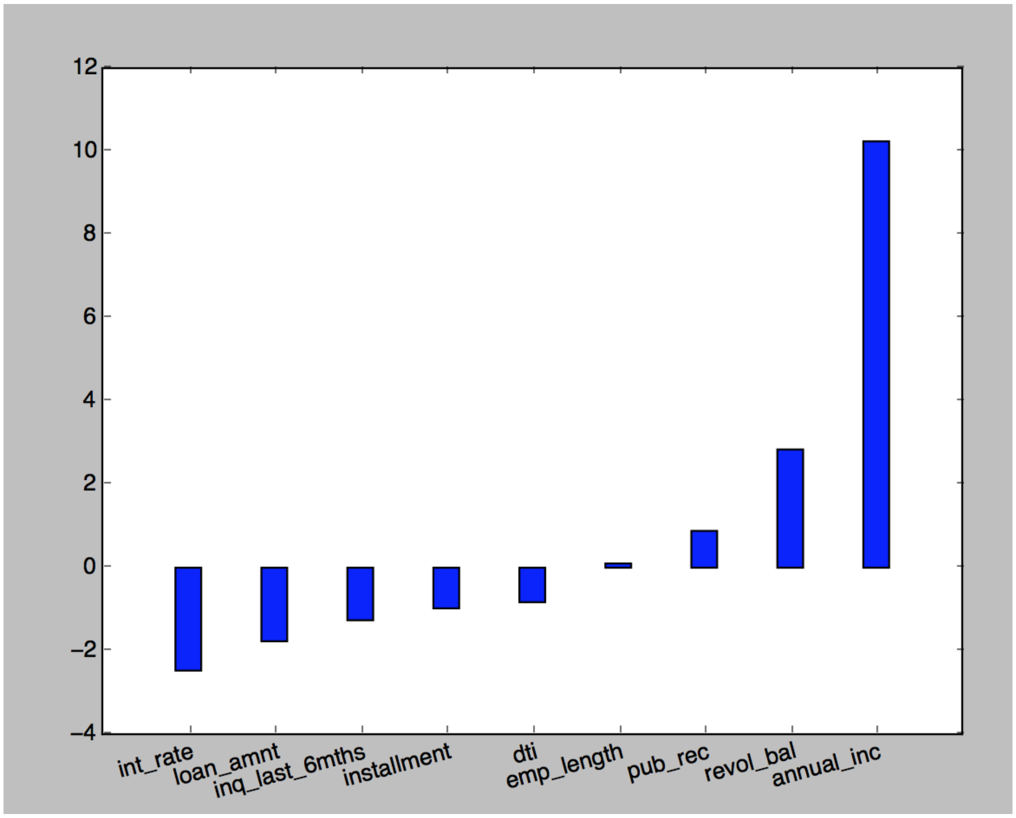
6.1 Feature Weight

We selected some features with the most positive and the most negative weights and made a plot. Here are what the labels represent:

- int_rate: Interest Rate on the loan
- loan_amnt: The listed amount of the loan applied for by the borrower
- inq_last_6mths: The number of inquiries in past 6 months
- installment: The monthly payment owed by the borrower if the loan originates

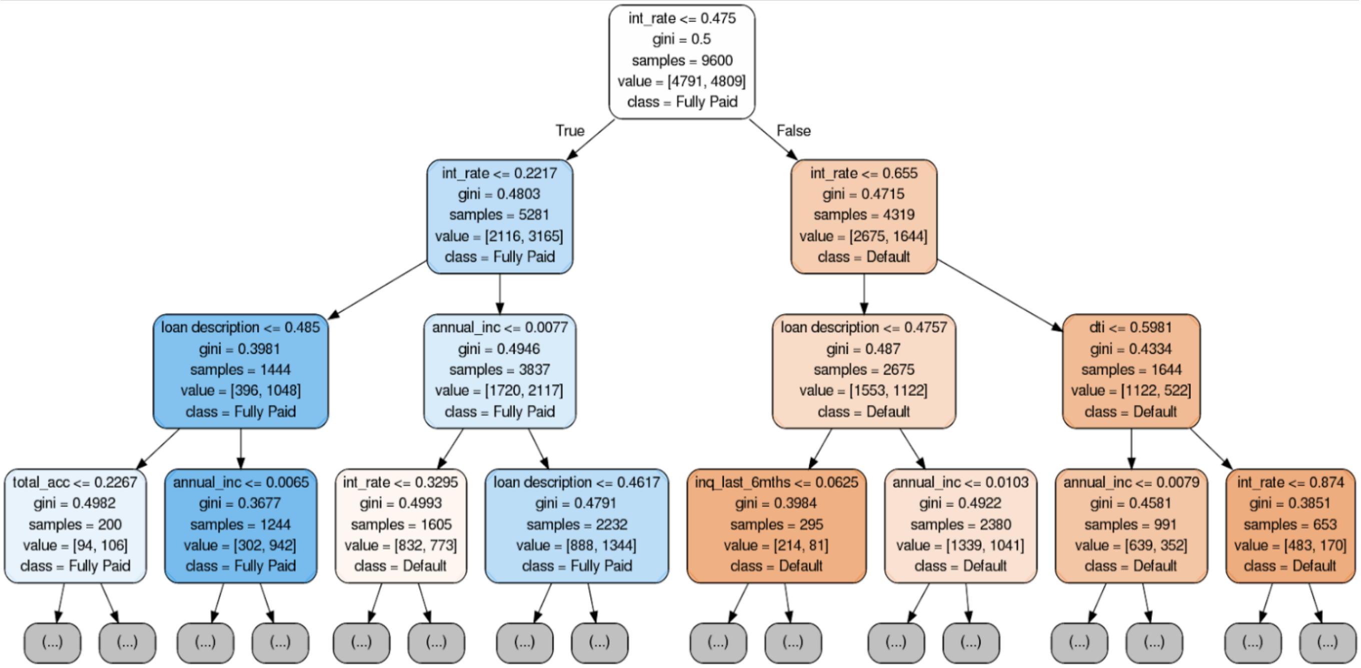
- dti: A ratio calculated using the borrowers total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrowers self-reported monthly income
- emp_length: Employment length in years
- pub_rec: Number of derogatory public records
- revol_bal: Total credit revolving balance
- annual_inc: The self-reported annual income provided by the borrower during registration

Most of these weights make sense, for example, employment length and annual income are positively correlated with paying back, and debt to income ratio is negatively correlated with paying back. However, there exist relationships that are against our common sense. For example, the number of bad public records and the total credit revolving balance appear to be positively related to paying back.



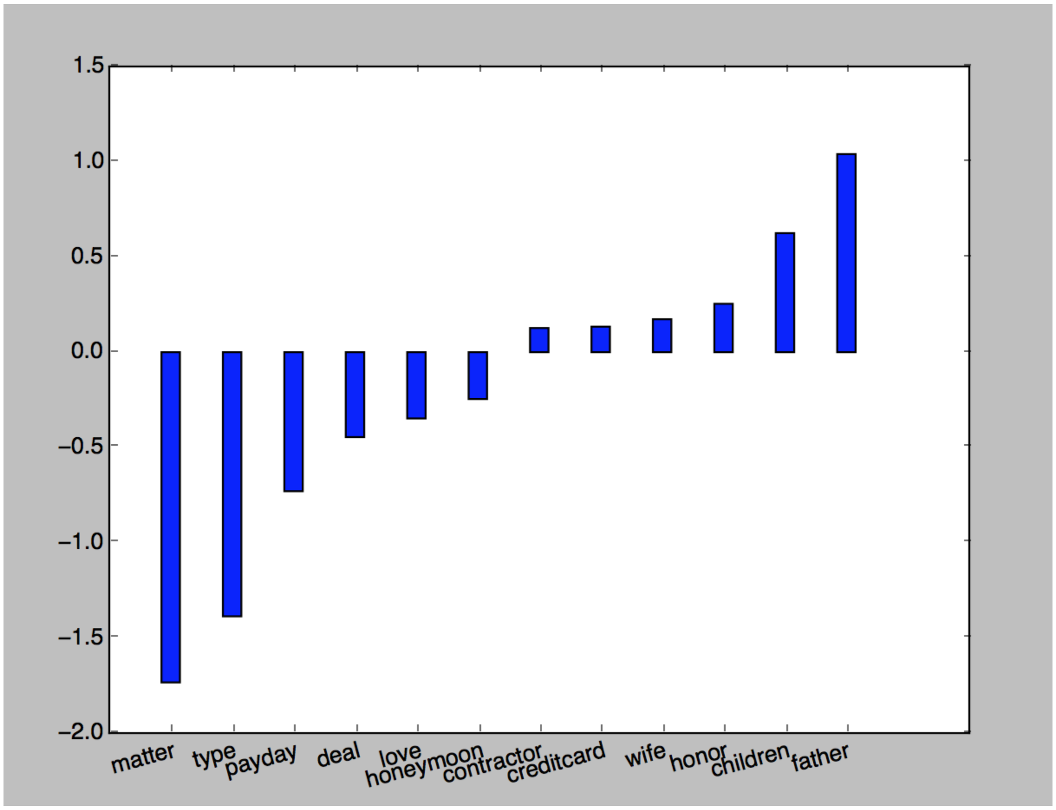
6.2 Decision Tree

Here is a visualization of our decision tree. It shows the first three layers of the tree. We can see that interest rate is one of the most important features. Specifically, when interest rate is higher than 0.475% (normalized), loans are more likely to default. Annual income is another crucial factor. When annual income is higher than 0.0077 (normalized), loans are more likely to be fully paid.



6.3 Weight of Words in Description Field

We also made a plot using the weights of some selected words in the description field. The whole vocabulary contains around 5000 words, but only about 100 of them have a non-zero weight. Notice that among the words that are related to relationships, ‘wife’, ‘children’ and ‘father’ have positive weights, while ‘love’ and ‘honeymoon’ have negative weights.



7 Conclusion and Future Work

In conclusion, our model is more effective in classifying personal loans than Lending Club's. It can potentially help Lending Club detect more bad loans and guide their investors away from such risky loans. To further improve our model in the future, we can also add some macroeconomics factors such as unemployment rate by state of residence. To make even better use of the loan descriptions, we can also do some topic modeling exploration on the text.

References

[1] Sontag, David. ”Problem Set 1: Perceptron Algorithm.” (n.d.): n. pag. Introduction to Machine Learning, Spring 2016. David Sontag. Web. http://cs.nyu.edu/ dsontag/courses/ml16/assignments/ps1.pdf.

[2] Sontag, David. ”Introducion to Bayesian methods” (n.d.): n. pag. David Sontag. Web. http://cs.nyu.edu/ dsontag/courses/ml16/slides/lecture8.pdf.

[3] Sontag, David. ”L1 Regulariza & Intro to Learning Theory Lecture 17.” (n.d.): n. pag. David Sontag. Web.http://cs.nyu.edu/ dsontag/courses/ml16/slides/lecture17.pdf.

[4] Sontag, David. ”Support Vector Machine.” (n.d.): n. pag. David Sontag. Web. http://cs.nyu.edu/ dsontag/courses/ml16/slides/lecture4.pdf.

[5] Sontag, David. ”Decision Trees.” (n.d.): n. pag. David Sontag. Web. http://cs.nyu.edu/ dsontag/courses/ml16/slides/lecture11.pdf.

[6] Ng, Andrew. CS229 Lecture Notes. N.p.: n.p., n.d. Web. ¡http://cs229.stanford.edu/notes/cs229-notes3.pdf¿.