Demystifying the workings of Lending Club

Bhatnagar Pujun
353 Serra Mall
Stanford, CA 94305
pujun@cs.stanford.edu

Chow Nick
353 Serra Mall
Stanford, CA 94305
nickchow@stanford.edu

Lai Max
353 Serra Mall
Stanford, CA 94305
maxlai@stanford.edu

Abstract

Lending Club is the world's largest peer-to-peer marketplace connecting borrowers and investors. They claim to transform the banking system by operating at a lower cost than a traditional bank and thereby making credit more affordable and investing more rewarding. Over the last 8 years, the number of loans in the marketplace has increased exponentially, yet little is known about the algorithms that determine if a loan is approved, and if it is, the interest rate a loan is offered at. In this paper we attempt to demystify the inner workings of this marketplace by applying machine learning techniques to Lending Club's publicly available dataset.

Using a basket of supervised learning techniques, we find that we can build highly accurate models, with an F-measure of up to 98%, that predict if an application will be approved. We also find that if a loan is approved, we can determine the interest rate at which the loan will be offered at. We provide an analysis of the performance of different machine learning models applied to our dataset.

With the models generated, we discover that Lending Club has gradually relaxed its application loan approval criteria. We hypothesize that this was due to the company preparing for its initial public offering, which eventually happened in 2014. In addition, we find that certain features, such as if the loan is a credit card refinancing loan, are constantly predictive of whether a loan is approved or denied. Using this newly discovered insight, we suggest some ways to game Lending Club's system to increase an applicant's chances of approval.

Finally, using effective clustering and visualization techniques, we uncover and exhibit structure in this rich dataset, which can be exploited to artificially generate more examples, specially for the years which only a limited number of training examples are available.

1 Introduction

Lending Club, as an online banking platform, is becoming increasingly popular ever since it started in 2007. By applying machine learning techniques, we intend to investigate following questions:

- Using supervised methods, can one predict if a loan application would be approved?
- Given that an application is approved, can we correctly predict the offered interest rate?
- Has the standard of Lending Club approvals changed over the years of 2007-2015, especially after their initial public offering?
- Can we extract a trend of how the significance of various features has changed over the years? Can this information be used to game their online system to increase applications' chance of approval?
- Can we find some structure among this rich dataset which can be used to generate artificial data for our models, especially for the earlier years of Lending Club?

2 Data Pre-Processing

The dataset, available at Lending Club Website, is a comprehensive dataset of all applications for peer-to-peer loans on the Lending Club platform between 2007 and 2015. The data files are csv files which are split by whether the loan is approved or denied. The following is a plot of the Lending Club application statistics each year:

Note that the number of training examples grows exponentially over the years as Lending Club has expanded rapidly. The amount of loan applications grew from 5,000 in 2007 to over 3 million in 2015.

Denied applications contain far fewer features than approved applications. For the approval classification problem, we maximize the available data by combining the features available in both the approved and denied applications. For the interest rate regression problem, we do not have to analyze the denied applications and hence we can use all the features available in the approved applications. Hence, we decide to separate pre-processing for classification and regression.

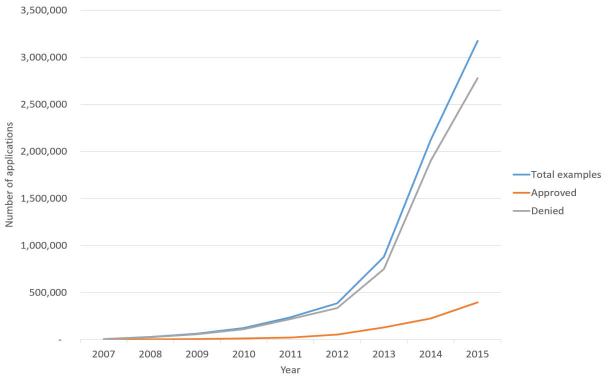


Figure 1: Number of applications received by Lending Club per year

2.1 Classification Task: Loan Approval

To determine if the criteria for approval has changed over the years, we first split up the datasets according to the year the loan was issued. We use R and Python to pre-process the data. All pre-processing scripts used can be found on our Github repository.

Before we start our analysis, we extract the common subset of features from the approved and denied files and combine the two datasets together for each year.

We also notice that in the approved dataset there are only 14 unique values for the *Purpose of loan* column, while in some years of the denied dataset there are over 10,000 unique values. However, we observe that the top 100 unique values for each year in the denied dataset represents over 99% of that year's denied loan applications (with the 2007 data as an exception). Therefore, in hope of cleaning the data, for each year we create a function that maps the top 100 unique values into the 14 unique values in the approved dataset. We delete the last 1% of denied loan applications.

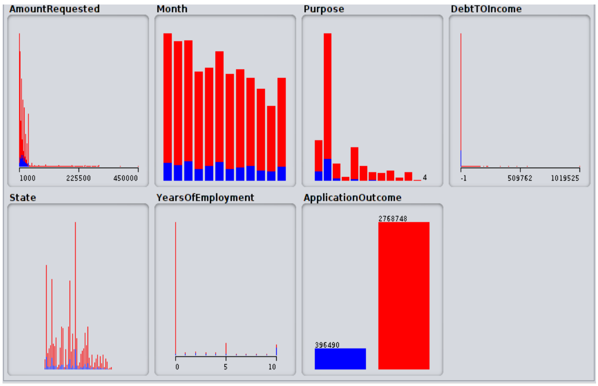


Figure 2: Classification Task: Processed Lending Club Data for 2015

2.2 Regression Task: Interest Rate

Using our general intuition, we carefully select 22 out of 111 features from the approved data, where about half of the features are empty. *Loan amount*, *Interest rate* and *Loan quality* are among some of the selected features. We continue processing the data as highlighted in the previous sub-section for each year.

We notice that many of the features, such as *Income*, have a right skew. Therefore we log-normalize the data with mean 0 and variance 1 to ensure our algorithm treats each feature equally. We run linear regression on this data.

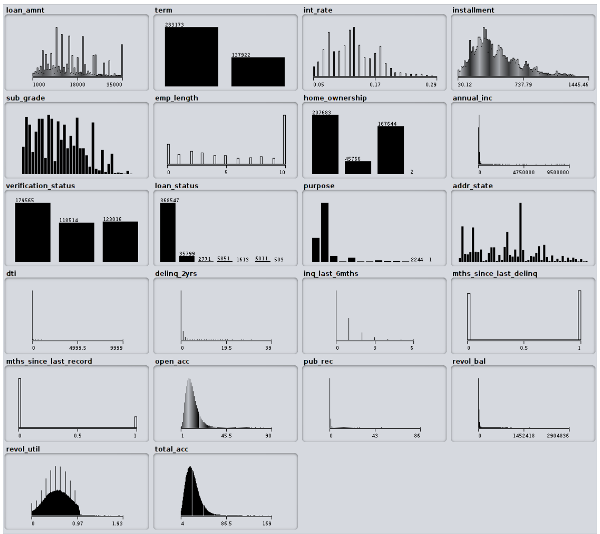


Figure 3: Regression Task: Processed Lending Club Data for 2015

3 Models

3.1 Classification Task

In order to find the model that works best, we apply several machine learning algorithms onto our dataset and compare their performances. For each year of data, we independently run Support Vector Machines [2], Logistic Regression, Boosting [5], Random Forest [1] and Artificial Neural Network [8], for a total of 45 iterations. For fast training, we use Java, Python, and Weka [3], scikitlearn [7]. We split the data into 70% training set and 30% testing set.

3.2 Regression Task

After successfully training a classifier that can predict if a loan will be approved, next step is to predict the interest rate for the approved loans. For accomplishing this task, we apply regression techniques after normalizing data. In order to measure the accuracy of our model, we decide to measure performance by using root mean squared (RMS) error because we want to heavily penalize the model for incorrectly predicting the interest rate.

3.3 Clustering Task

Hoping to find latent structure within the data, we use unsupervised techniques to cluster the data. To investigate this, we remove the *State* and *Purpose of loan* features and cluster the normalized data using K-means [4]. In order to visualize this data, we implement t-SNE [6] using Python. t-Distributed Stochastic Neighbor Embedding (t-SNE) is a (prize-winning) technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets. We use this approach to find if our resulting clusters indicate some latent trend within the data.

4 Analysis

4.1 Classification Results

For each year of our data, we calculate the F-measure associated with the dataset using the confusion matrices. We use this as our primary measure of performance for each of the models because we notice that evaluating model performance based on classification error is potentially misleading. As only a small fraction of the applications (5-15%) are successful in any year, classifiers, such as ADA-Boost for the 2009 data, can get a low classification error just by classifying every loan as denied. Therefore, we use the F-measure, which is a combination of precision and recall, to evaluate our models' performance.

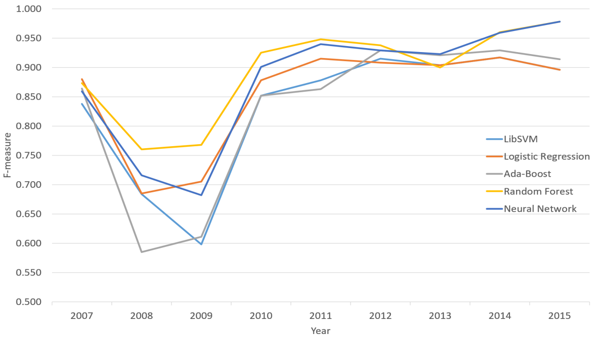
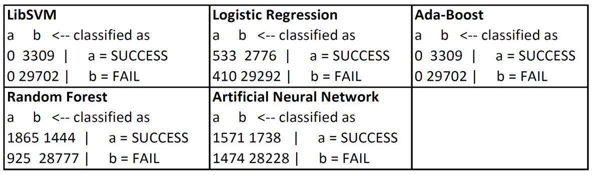


Figure 4: F-measure values by algorithm and year

Figure 4 is a visualization of how the machine learning models compare to each other over the years. ***Overall, we find the Random Forest algorithm generally produces the best predictions, with a F-measure of 98% in 2015***, but Artificial Neural Network performs equally well as the number of training examples increases over the years.

As we run the models on datasets with more examples, we expect the accuracy to trend upwards. We believe the models' accuracy for 2007 in Figure 4 is higher than expected because of a variety of reasons. First, since this was the first year that Lending Club open-sourced their data, their data isn't consistent. During the pre-processing stage, we end up dropping most of the examples, which we suspect ultimately leads to over-simplification and enables us to learn a simpler model that works quite well with the remaining 2007 data. We also hypothesize that in 2007, Lending Club was using a simpler algorithm with limited features, which is easily estimated by our classifiers.



LibSVM a b <-- classified as 0 3309 | a = SUCCESS 0 29702 | b = FAIL	**Logistic Regression** a b <-- classified as 533 2776 | a = SUCCESS 410 29292 | b = FAIL	**Ada-Boost** a b <-- classified as 0 3309 | a = SUCCESS 0 29702 | b = FAIL
Random Forest a b <-- classified as 1865 1444 | a = SUCCESS 925 28777 | b = FAIL	**Artificial Neural Network** a b <-- classified as 1571 1738 | a = SUCCESS 1474 28228 | b = FAIL	

Figure 5: Confusion matrices for year 2009

Figure 5 shows confusion matrices for the classification results. During the analysis, we see that different algorithms classify examples differently and therefore we decide to look at the confusion matrices. In some cases, like ADABoost, the model classifies all examples are positive and doesn't do anything intelligent. We identify these models and make sure to not use this for future testing.

4.2 Has Loan Quality Changed?

We see convincing evidence that Lending Club has gradually relaxed its loan approval standards. Figure 6 shows a plot of *Debt-to-Income (DTI)* for all approved loans in each year between 2007-2015. The graph shows that Lending Club increased the maximum DTI that it will accept on applications gradually from 25% in 2007 to 40% in 2015. Lending Club also relaxed the maximum *Loan Amount* that it will accept from $25,000 in 2007 to $35,000 in 2015, as seen in Figure 7.

A possible explanation for a relaxation over the years is that the management team wanted to generate greater revenue growth and higher profits to prepare for an eventual initial public offering (which happened in 2014). Since a relaxation would result in the approval of more loans, and since Lending Club charges a percentage fee on every loan that is funded on its platform, increasing the maximum *Loan Amount* and *DTI* would result in higher profits and higher valuation of the company.

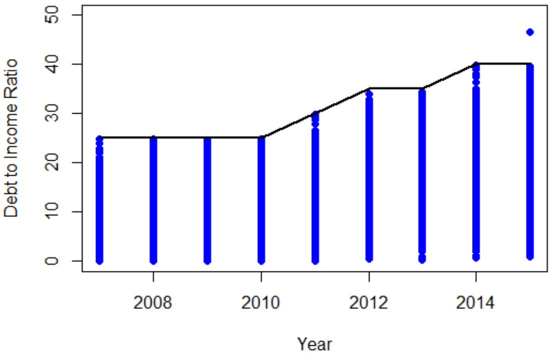


Figure 6: Debt-To-Income of Approved Applications 2007-2015

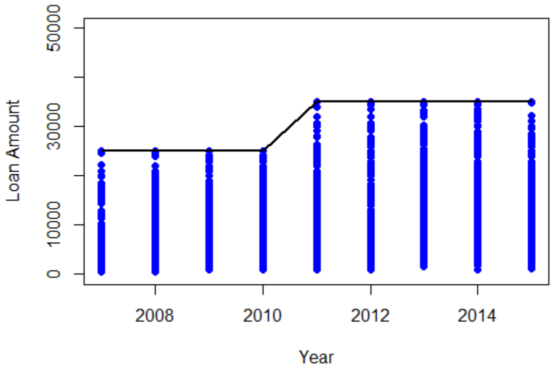


Figure 7: Loan Amounts of Approved Applications 2007-2015

4.3 Has Feature Significance Changed?

We notice that there are certain features that are constantly predictive of whether a loan is approved or denied. In Figure 8, we plot some of the features that are most indicative of approval and denial. This plot shows the ranking of the resulting coefficients of the Logistic Regression. The higher the value, the more predictive the feature is of approval for that year, while the lower the value, the more predictive the feature is of denial for the year.

We notice that educational loans are likely to be denied, whilst credit card consolidation loans are likely to be approved. This can be explained through economic intuition. Education loans are likely to be denied because seekers of these loans, students, are unlikely to have a stable source of income and hence are likely to have a higher chance of defaulting. On the other hand, credit card refinancing applications are likely to be approved because people who want to refinance credit card debt must already have a credit card, which itself requires a stringent credit approval process.

Renewable energy loans tell a particularly interesting story. In 2008 and 2009, renewable energy loans were highly predictive of loan approval. However, this predictiveness disappeared soon after and by 2013 renewable energy loans were actually predictive of loan denial. An explanation for this effect is that in 2008 the Energy Improvement and Extension Act was passed, which provided tax credits to renewable energy initiatives. Therefore the borrowers had, in effect, higher disposable income and hence a higher probability of paying back the loan compared to before. By 2014, many of these tax credits had been phased out, and therefore Lending Club has reversed their algorithm to account for this change.

An immediate takeaway from this analysis is that ***applicants should state that the purpose of the loan is for credit card consolidation to maximize their chances of approval.***

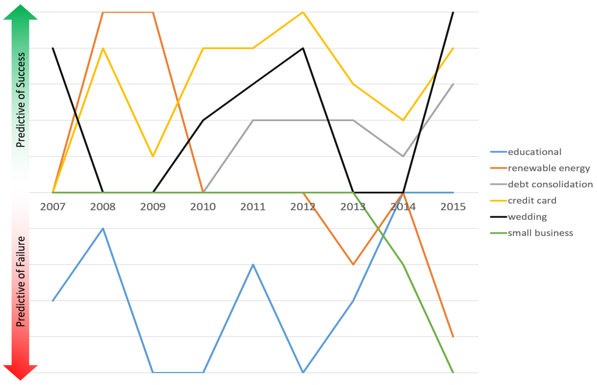


Figure 8: Most significant features by year

4.4 Regression Results

We use regression to predict the interest rate for approved applications. After running linear regression, we find that the RMS errors are very low (less than 0.003 in 2015) because *Loan Grade* is included as a feature. As *Loan Grade* is determined by an algorithm within Lending Club, we decide to not use it and implement different data processing techniques on the remaining data.

We perform PCA and run regression on the transformed data. We hope that by reducing the number of dimensions, we would be able to counter noise present in the data and account for less data, especially for the earlier years, and consequently decrease the generalization error of our interest rate predictions. To determine the number of principle components to include, we run PCA on the normalized data for each year. The results are shown in figure 9. We notice that there is a noticeable 'kink' in the data after 3 principal components, which indicates that most of the variance is captured by the first three eigenvectors. Hence we decide to use $k = 3$ for PCA.

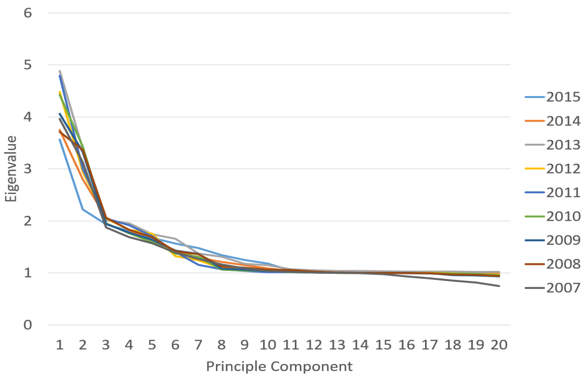


Figure 9: Degree of variance captured by PCA

Figure 10 shows the RMS error of our interest rate predictions using different data processing techniques.

Analyzing the results, we note the following insights:

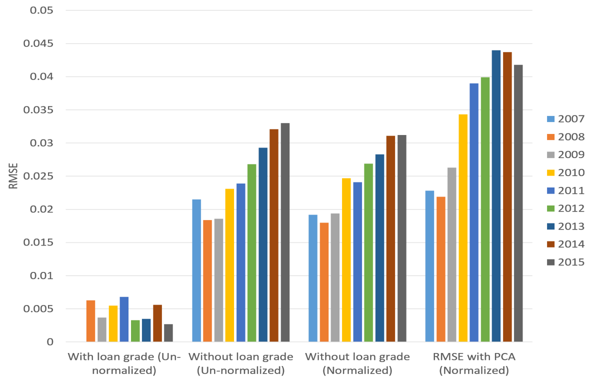


Figure 10: Regression results for Interest Rate.

- ***Loan Grade feature is a near perfect predictor of the interest rate***: This is not surprising, as Lending Club states that they determine the interest rate based on the *Loan Grade* calculated for that loan. We notice that we generally do not have perfect predictions on interest rates for different *Loan Grade*. This is likely because loan rates for different *Loan Grade* will change over time (but the low RMS error shows that interest rates for a specified *Loan Grade* do not vary drastically over a year).
- ***Increased complexity of Lending Club's system over the years***: Even though there are more data samples each passing year, our RMS error has steadily increased. In 2007, our simple linear regression algorithm did a good job of predicting the interest rate of a loan, with a RMS error of 0.02. Using the same type of model, our 2015 RMS error is over 0.03. It also seems that their model now uses more features, some of which may not publicly available.
- ***Low dimensionality of the approved data***: Taking the first 3 (of over 30) principal components generates decent predictions. We realize a RMS error that is about 25% greater than that of taking all principal components. This shows most of the important determinants of interest rate can be represented by the first three principal components.

4.5 Clustering Results

While trying to build models to predict the interest rate and if a loan will be approved, we notice some structure in the data and hypothesize that we may be able to find clusters that are highly indicative of interesting trends. We decide to apply techniques from our unsupervised toolbox to find structures but quickly discover that there isn't any intuitive way of visualizing the results of our experiments. After doing some research, we find t-SNE as one of the ways to visualize our results. We discover the following interesting trends: As seen in figure 11, we generate clear clusters when we remove the *purpose* attribute and try clustering the examples. We observe that the results are sparse and are localized to different parts in the high dimensional space. This proves that our hypothesis about some inherent structure in the data. Also, using the found clusters, we can potentially generate even more examples, which can in turn be used to improve our models performance, especially for the starting years where we have limited data.

5 Acknowledgments

We would like to thank Dr. Duchi and all 229 TAs for guiding us and consistently assisting us throughout the project.

References

[1] Leo Breiman. "Random Forests". In: *Mach. Learn.* 45.1 (Oct. 2001), pp. 5–32. ISSN: 0885-6125. DOI: 10.1023/A:1010933404324. URL: http://dx.doi.org/10.1023/A:1010933404324.

[2] Christopher J. C. Burges. "A Tutorial on Support Vector Machines for Pattern Recognition". In: *Data Min. Knowl. Discov.* 2.2 (June 1998), pp. 121–167. ISSN: 1384-5810. DOI: 10.1023/A:1009715923555. URL: http://dx.doi.org/10.1023/A:1009715923555.

[3] Mark Hall et al. "The WEKA Data Mining Software: An Update". In: *SIGKDD Explor. Newsl.* 11.1 (Nov. 2009), pp. 10–18. ISSN: 1931-0145. DOI: 10.1145/1656274.1656278. URL: http://doi.acm.org/10.1145/1656274.1656278.

[4] Tapas Kanungo et al. "An Efficient k-Means Clustering Algorithm: Analysis and Implementation". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 24.7 (July 2002), pp. 881–892. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2002.1017616. URL: http://dx.doi.org/10.1109/TPAMI.2002.1017616.

[5] Xuchun Li, Lei Wang, and Eric Sung. "AdaBoost with SVM-based Component Classifiers". In: *Eng. Appl. Artif. Intell.* 21.5 (Aug. 2008), pp. 785–795. ISSN: 0952-1976. DOI: 10.1016/j.engappai.2007.07.001. URL: http://dx.doi.org/10.1016/j.engappai.2007.07.001.

[6] Laurens van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE". In: *The Journal of Machine Learning Research* 9.2579-2605 (2008), p. 85.

[7] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[8] O. Postolache et al. "The Multisensor ANN Fusion Method for Accurate Displacement Measurement". In: *Buletinul Institutului Politehnic din Iasi* XLV(IL).Fasc 5A (Nov. 1999), pp. 363–369.

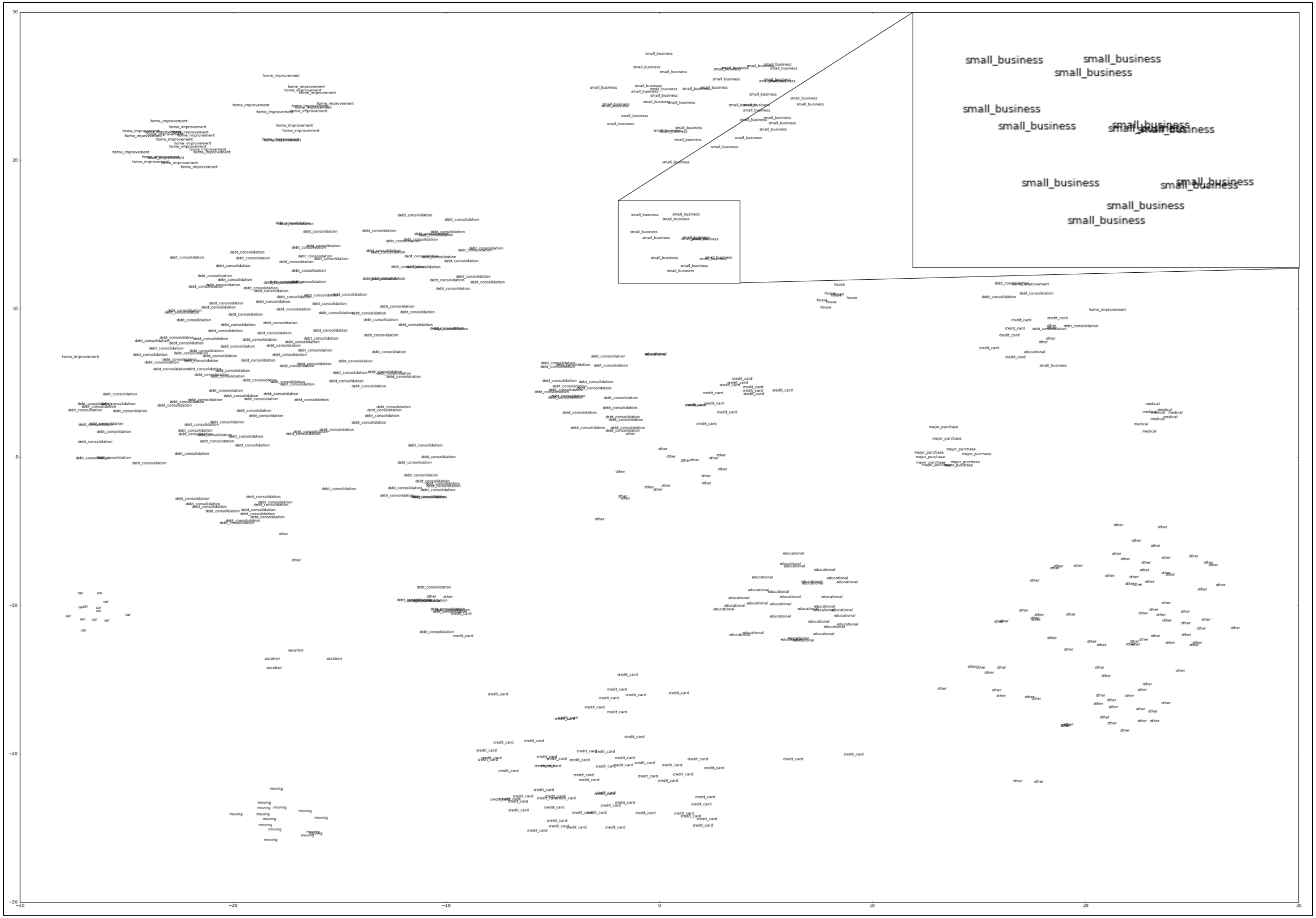


Figure 11: t-SNE visualization of found clusters