Credit risk analysis with machine learning techniques in peer-to-peer lending market

Qionglin Shan & Mikael Nilsson

Abstract

After the sub-prime mortgage crisis of 2007 and global crisis of 2008, credit risk analysis has becomes more important than ever before. This paper conducts the credit risk analysis and compares classification performances among different algorithms (logistic regression, support vector machine, decision tree, multilayer perception, probabilistic neural network, Deep Learning) by using a large peer-to-peer lending dataset composed of a million observations. The findings show that Support vector machine (SVM) provides the most accurate performance, followed by decision tree, logistic regression, multilayer perceptron neural network, probabilistic neural network and deep learning. The main contributions of this paper is the reapplication of machine learning techniques to an alternate dataset composed of significantly larger number of observations with deviating pattern from traditional bank loans. The findings from SVM and Decision tree are consistent with the previous literature. The results from logistic regression and MLP indicate that they are identical based on p2p dataset, which makes a contribution to the debate whether MLP out performs logistic regression. For PNN it is difficult to say if it properly accounts for the data imbalance due to the low performance of the model compared to the others. Deep learning performance is in contrast to previous work as it is the worst performing model comparing with other investigated techniques. This is potentially due to the simple approach to deep learning that this paper adopted and opens up the topic for future research.

Keywords: Credit Risk, Default, Machine Learning, SVM, MLP, Logistic Regression, Decision Tree, PNN, Peer to Peer lending

Stockholm Business School

Master's Degree Thesis 30 HE credits

Subject: Finance

Program: Master's Programme in Banking and Finance 120 HE credits

Spring semester 2018

Supervisor: Desheng Wu

Contents

Introduction

After the sub-prime mortgage crisis of 2007 and global crisis of 2008, credit risk analysis has becomes more important than ever before. Basel committee on banking supervision released Basel II, which required supervised financial institutions to use internal rating to measure their credit risk exposure. Those actions resulted in banks enhancing methods of credit risk analysis. In addition, banks and financial institutions improve their credit scoring system not only due to the policy, but also because any small improvements might lead to a large amount of profits (Hand & Henley, 1997).

Credit risk is the loss to lenders due to borrowers defaulting on their credit obligations. Making decisions about the approval of credit and the interest rate relies on the assessment of credit risk for lenders. There are credit-scoring systems that are designed to enhance lenders' abilities in evaluating creditworthiness of customers in the process of credit risk analysis. The possibility of automating credit approval is one of the most important advantages of credit scoring, because the system can quickly gather the necessary information, evaluate, and determine whether to approve the application, increasing effectiveness. Those credit scores could be used as an indicator of the level of creditworthiness for helping lenders to decide whether to approve for credit, interest rates and repayment terms. Although automatic credit scoring could not take the place of the credit professionals, it speed up the process of decision making at the early stage, where a case should be denied or analysed further (Sakprasat & Sinclair, 2007)

A financial institution has to make a decision whether grant a loan to customer and usually, those customers would be classified as good or bad. A good credit refers to one that is likely to repay financial obligation, while a bad credit means that one may have high probability of default (Yap, Ong, & Husain, 2011). Those decisions are made based on all possible relevant information of applicants such as economic conditions, marital status and intentions. The advance of data storage technology enables to help financial institution to store information and repayment behaviour of borrowers electronically, especially peer-to-peer (p2p) lending showing up.

The emergence in recent years of p2p lending, through companies such as lending club, creates an alternative financing way removed from the traditional gatekeepers of financial institutions such as banks. The main function of p2p lending is to connect borrowers and lenders directly rather than filtering through financial intermediaries. It follows the current trend of decentralized services such as Airbnb and Uber (Caldieraro, Zhang, Cunha Jr, & Shulman, 2018), and the critical role of banks as a delegated monitor is lost in the process. The nature of credit is somewhat divergent from traditional banks, where there is a large institution that filters through large quantities of loans. A feature in the loan structure of p2p lending is the relatively small amount loan, comparing with a bank loan (Guo, Zhou, Chunyu, Liu, & Xiong, 2016). Rather than a centralized agent, debtors and creditors are directly linked through the service. Another feature is that the p2p data is available publicly for research. Traditional banking systems have multiple levels of security and are reluctant to hand out private information. Hence obtaining credit data is a challenge. Therefore, p2p lending will enable us to obtain large quantity of data (Guo, Zhou, Chunyu, Liu, & Xiong, 2016), as well expand the net by which credit risk is analysed. Large datasets, or big data, enables further analysis of credit using machine-learning techniques.

Credit risk assessment is a challenging task in financial analysis, which has been suggested in large amount of academic literatures for dealing with this problem. There are numerous methods to develop credit risk evaluation method, such as traditional statistical method (e.g., logistic regression), non-parametric statistical models (e.g., k-nearest neighbour and decision tree) and neural network method (e.g. multilayer perceptron (MLP), support vector machine (SVM) and probabilistic neural network (PNN)).

For logistic regression, empirical evidence shows that in traditional methods, logistic regression could provide most accurate result in credit scoring (West, 2000). Nowadays, neural network has attracted a lot of attention in credit risk assessment recently, but it remains uncertain of the superiority of neural network comparing to traditional statistical algorithms such as logistic regression. Some evidences support superiority of neural network techniques, presenting that artificial intelligence techniques have improved the results of credit risk analysis when compared with ones provided from classical statistical approaches (Abellan & Castellano, 2017). However, Yap, Ong, & Husain (2011) compare the performances of credit models among logistic regression, decision tree and own credit scorecard model, concluding that there is no model outperform the others (Yap, Ong, & Husain, 2011). Finlay (2012) and Lessman (2015) propose that due to easy implementation and accuracy, LDA and logistic regression remain popular in credit risk analysis (Finlay S. &., 2012) (Lessmann, 2015). So, logistic regression model, as the most accurate model in traditional statistic method, is included in the analysis to compare with other techniques to see its performance.

Considering neural network, Desai et al (1996) clarifies that multilayer perceptron neural network has a better performance than linear discriminant analysis, but only better than logistic regression marginally. Salchenberger et al (1992) compare multilayer perceptron neural network with logistic regression and conclude that multilayer perceptron neural network has better performances than logistic regression model for each examined dataset (Salchenberger, Cinar, & Lash, 1992). Tam and Kiang investigate Texas bank failure prediction and found that the multilayer perceptron is most accurate model, followed by linear discriminant analysis, logistic regression, decision trees and k nearest neighbor. But there is an empirical evidence finding that multilayer perceptron may not be the most accurate neural network model (West, 2000). Moreover, a furthermore study by Desai et al (1997) about credit scoring suggests that logistic regression outperforms the multilayer perceptron neural network at 5% significance level (Desai V. , Conway, Crook, & Overstreet, 1997).

Regarding to Support Vector Machine, in the comparison of different machine learning approaches, Yu, et al., (2010) find that SVM performs best in the single agent case. Burgers (1998) finds that the performance of SVM is either identical or significantly better than other competing techniques in the applications (Burgers, 1998). Moreover, Baesens (2003) finds that SVM achieves the highest classification accuracy rate among the 17 methods tests (Baesens, et al., 2003). Huang , Chen, Hsu, Chen, & Wu, (2004) compare with other three techniques (neural networks, genetic programming and decision tree), suggesting that SVM approach could provide similar classification accuracy, even though there are relatively few input variables (Huang, Chen, & Wang, 2007). Hens and Tiwari (2012) suggest that SVM method is competitive in terms of accuracy rate and computational time (Hens & Tiwari, 2012). However, Lee et al (2006) indicates that CART and MARS provide better performance in credit scoring accuracy than support vector machine (Lee, Chiu, Chou, & Lu, 2006).

As for decision tree model, Davis et al (1992) conduct a research related to credit card scoring and conclude that a comparable level of accuracy is obtained for decision tree model and multilayer perceptron neural network (Davis, Edelman, & Gammerman, 1992). En empirical evidence finds that no model outperform the other when logistic regression model, decision tree and credit scorecard mode are compared (Tap & Ong, 2011). A research by Zhao, Xu, Kang, Kabir, Liu, & Wasinger (2015) suggests that the decision tree model performed a little better than back propagation, but both techniques could achieve high accuracy rates. Galindo and Tamayo (2000) use a mortgage loan dataset to make a comparison among different techniques. It indicates that decision tree model could have best performance for default (Galindo & Tamayo, 2000).

Alternatively there is the probabilistic neural network. Research related to financial distress of Chinese public companies, probabilistic neural network has good performance for classification. Research working on credit scoring in small-business lending compares the accuracies of models extracted by different machine learning techniques. Probabilistic neural network, as one of the neural network algorithms produce the highest hit rate and the lowest type I error meaning lowest value of false positive (Bensic, Sarlija, & Zekic-Susac, 2005). In addition, Hájek (2011) suggests that it is possible to deal with the problem of imbalanced dataset by using PNN, because of the ability of PNN to classify the classes correctly in the research (Hájek, 2011). There is limited research related to probabilistic neural network in credit risk field.

Considering deep learning, the application of deep learning in credit analysis is limited to Credit Default Swap (Luo, Desheng, & Dexiang, 2017), and Japanese credit card application (Yu, Yang, & Tang, 2016) by using an ensemble DBN with extreme machine learning. In both papers deep learning methods show positive performance compared to other machine learning techniques.

Although many researches are conducted on logistic regression, support vector machine, decision tree and multilayer perception, there are some contrary research results. Based on a large and new p2p lending data, classification performances of those algorithms are investigated to make a contribution on related research literature. In addition, since there are limited literature related to probabilistic neural network and deep learning, those two algorithms are conducted to make a contribution in credit risk field. Therefore, the research question is to compare classification performances among different algorithms (logistic regression, support vector machine, decision tree, multilayer perception, probabilistic neural network, Deep Learning) by using a recent and large p2p lending dataset.

This paper contributes to the literature in peer-to-peer marketplace. There are a few researches exploring credit problem based on peer-to-peer lending. This research differs from previous literature in various aspects. Firstly, p2p lending data is obtained for extending risk analysis research. As for the dataset used for credit risk or classification, many studies used datasets from the UCI Machine learning repository, which contains German, Austrian, and Taiwan data (Yeh, 2009)(Khashman, 2010)(Khashman,2011). But the time period of German and Austrian dataset are around 1990. Regarding to the Taiwan dataset, there is no specific time cited in data description either. Far old dataset might not that meaningful in nowadays credit risk analysis. However, finding credit risk dataset in public is difficult and asking banking for credit data is not likely due to the privacy. Thanks to the peer-to-peer online lending platform showing up, it provides a transparent public dataset with numerous samples. P2P lending data contains huge amount of information of applicants, like annual income, employment title and

length etc, except ID number, which provide us a large and recent data source for credit scoring. The second contribution is that p2p dataset contains numerous data from applicants. Wu, Liang, & Yang, (2008) suffers from small dataset problem and use 10-fold cross validation to deal with small sample (Wu, Liang, & Yang, 2008). The instances of dataset are around 1048574, which provide a large enough dataset for conducting the research. The third contribution is probabilistic neural network (PNN) and Deep Learning techniques are included in classification analysis, which are seldom used in previous literature. Due to its fast learning capability and efficiency as well as the possibility to solve imbalanced data problem, the probabilistic neural network is selected and compared among other techniques to contribute to literature on using PNN to analyse large p2p dataset for classification. Deep learning, as a popular algorithm, also apply into analysis of credit assessment. The forth contribution is related to researches in p2p market place, since the literature on p2p is limited. In the paper by Caldieraro et al (2018) they sought to address information asymmetry in p2p lending with the loss of the traditional role of financial intermediaries. It also addressed the increased ability to attain large-scale datasets, which is an area where our paper seeks to contribute.

This paper attempts to conduct big data analysis. The first step is to pre-process data, in order to deal with high-dimension data. This is done through factor analysis of mixed data (FAMD) (Pagès, 2004). WEKA, MATLAB, and KNIME software are used to conduct different algorithms. WEKA (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. KNIME is similar with WEKA, which can conduct lots of machine-learning techniques that WEKA lacks of, such as PNN. In results part, AFER (accuracy rate) and ROC are the main measurements for classification performance. AFHR refers to accuracy rate. Sharda and Delen (2006) uses the average percent hit ratio (APHR) as the performance metric in prediction, which is calculated by using number of correctly classified instances over total number of instances (Sharda & Delen, 2006).

This paper is to compare classification performances among different algorithms (logistic regression, support vector machine, decision tree, multilayer perception, probabilistic neural network, Deep Learning) by using a recent and large p2p lending dataset. In conclusion, Support vector machine (SVM) provides the most accurate performance for the classification, following by decision tree, logistic regression, multilayer perceptron neural network, probabilistic neural network and deep learning. The performance of the multilayer perceptron (MLP) is identical with the performance of logistic regression. Deep Learning algorithm that was used in this paper performed poorly in comparison to all other techniques and so did PNN. Both techniques were especially bad at classifying the default category.

This paper will continue to conduct a literature review in section 2. Previous literatures on credit risk are presented, and followed by a thorough and critical discussion about the algorithms used for classification. The research design is in section 3. The paper is thereafter continued in section 4, methodology, explaining different techniques for credit risk analysis. In section 5, based on data analysis, the results will be discussed. In section 6, the main findings are presented in a conclusion, as well as a brief discussion of limitations and recommendations for further research.

Literature review

Credit risk

Credit risk and default probability is one of the most prominent areas of financial research with a long history. Credit risk is defined as the loss to creditors due to debtors defaulting on their credit obligations. Making decisions about the approval of credit and the interest rate relies on the assessment of credit risk for lenders. The ability to predict default and isolating the driven factors is a widely studied topic within finance with a long history.

Although automatic credit scoring could not take place of the credit professionals, it can speed up the process of decision making at the early stage where a case should be denied or analysed further (Sakprasat & Sinclair, 2007). The principal drawback is the artificial nature of credit rating. Credit rating agencies do not accurately present the true nature of the risk associated with certain factors (Bolton, Freixas, & Shapiro, 2012). Credit rating agencies tend to generally inflate the value of the client due to market pressures. The various conflicts of interest between credit rating agencies and the clients they score generates a dubious relationship between the two, hence reducing the value to the investors using the credit score to assess investment opportunities. Various countermeasures documented in Bolton et al (2012) suggest foresight of the dilemma imposed by credit scoring system conducted by credit rating agencies such as Moody's. A previous paper done by (Yu & Zhu, 2015) uses a variable 'grade' that is the credit rating done by lending club agency, as a independent variable to build the classifier for predicting default risk. It might be a problematic research design, because the classifier is built based on artificial rating done by agency, which is not real data from applicants. Credit scores and rating are artificial, which are based on implicit assumptions by the agents. Hence, they are subjected to various forces that may under or overvalue creditworthiness (Bolton, Freixas, & Shapiro, 2012). Credit rating from different agencies varies, which might not appropriate to be considered as independent variable for building a classifier. Biased results could exist.

Alternatively, a market-based approach would involve using credit default swaps (CDS) spreads to infer credit risk from clients. Since CDS is like an insurance upon the event of a default, the price reflects the market perception of a default event. Implied default probability of default from CDS offers a view of investors risk neutral assertions to credit events (Pan & Singleton, 2008), and often contrasted to credit scoring systems in financial research (Pan & Singleton, 2008; Longstaff, Mithal, & Neis, 2005). Both the default and non – default component of corporate spreads can be extrapolated from CDS (Longstaff, Mithal, & Neis, 2005), implying that CDS are more closely connected to evaluations by market forces. In the paper by Luo et al (2017) conducted a machine learning approach to classify credit ratings that were derived from the CDS data.

The availability of big data, and recent advances in computer processing power enables a novel approach to the problem involving machine-learning application to credit risk analysis. Contrary to the traditional approaches of credit analysis that rely on statistical regression techniques and discriminant analysis of established variables, machine learning enables an algorithm to analyse the dataset and produce a procedure that will predict the class an observation belongs to. Conducting these kinds of analytical techniques requires large datasets (Kaastra & Boyd, 1996; Renault, 2017). Data from lending club peer-to-peer loans dataset provides a necessary dataset to address this feature.

Logistic regression

There are two popular statistical tools to conduct credit risk evaluation: linear discrimination analysis and logistic regression. Linear discrimination analysis, as a simple parametric statistical model, was one of the first credit scoring models. However, it had been criticized because of the categorical nature of the credit data, which is not normally distributed. Moreover, the covariance matrices of the good and bad credit classes are unequal. More advanced statistical model shows up to solve some deficiencies of the LDA mode (West, 2000). A logistic regression model by Henley was used for credit scoring applications (Henley, 1995). Logistic regression could produce probability of a dichotomous outcome when potential predictor variables are inserted.

Empirical evidence shows that in traditional methods, logistic regression could provide most accurate result in credit scoring (West, 2000). Nowadays, neural network has attracted a lot of attention in credit risk assessment recently, but it remains uncertain of the superiority of neural network comparing to traditional statistical algorithms such as logistic regression. Some evidences support superiority of neural network techniques, presenting that artificial intelligence techniques have improved the results of credit risk analysis when compared with ones provided from classical statistical approaches (Abellan & Castellano, 2017). Baesens, Setiono, Mues, & Vanthienen (2003) says that comparing with the logistic regression classifier and the popular C 4.5 algorithm, neural network rule extraction techniques, such as neurorule and trepan, provide a very good classification accuracy (Baesens, Setiono, Mues, & Vanthienen, 2003).

However, Yap, Ong, & Husain (2011) compares the performances of credit models among logistic regression, decision tree and own credit scorecard model, concluding that there is no model outperform the others (Yap, Ong, & Husain, 2011). Finlay et al. (2012) proposed that Due to easy implementation and accuracy, LDA and logistic regression remain popular in credit risk analysis. Lessman et al.(2015) hold the similar opinion that logistic regression is used widely in practice because of its simplicity and balanced error distribution.

Artificial neural network

There is an increasing attention towards machine learning when evaluating credit risk. In an ever changing environment with increased complexities, non- linearity becomes a more integrated part of a world comprised of big data (Khandani , Kim, & Lo, 2010). Neural network refers to mathematical representations of the human brain' functions. There are different types of neural network, supervised or unsupervised learning. Using machine learning algorithms to peer through the data allows for pattern recognition rather than simple relying on internal models, where a common approach is to use a logit or probit regression (Kruppa, Schwarz, Arminger, & Ziegler, 2013). Compared with traditional models, Random Forest models outperformed traditional approaches (Kruppa, Schwarz, Arminger, & Ziegler, 2013), even when comparing with nearest k neighbour. Kruppa, et al., (2013) suggest the possibilities of alternative machine learning approaches, such as Neural Networks, as potentially promising compared to the standard statistical approaches, which is consistent with several empirical evidences. For example, Min and Lee (2008) also suggests that neural network could be considered as an accurate tool for credit analysis among others.

Due to its universal approximation property, neural network has attracted a lot of attention. Although it lacks explanation capability, neural network can provide a high predictive

accuracy. A literature from Baesens, Setiono, Mues, & Vanthienen (2003) named neural network decisions by explanatory rules, trying to help credit-risk manager to explain why this particular applicant should be divide into good or bad. By applying neural network rule extraction and decision tables, they believe an advanced and user-friendly decision-support system could be build for credit-risk evaluation (Baesens, Setiono, Mues, & Vanthienen, 2003).

Multilayer Perceptron Network

A common type of artificial neural network is the multilayer perceptron (MLP).
An MLP usually have an input layer, one or more hidden layers, and an output layer, which are all have several neurons. Every neuron starts from inputs and produces an output value, and then transmitted to the neurons in the subsequent layer in a feedforward manner. (Baesens, Setiono, Mues, & Vanthienen, 2003). MLP is a feedforward neural network where the inputs are transmitted through the net without feedback (Baesens, et al., 2003).

The decision about the number of hidden layers determines the complexity or depth of the neural net. A MLP model with more than 2 hidden layers is categorized as a deep neural network (Hamori, et al., 2018). There is evidence to suggest that 1 hidden layer network outperforms 2 hidden layers in credit risk evaluation. Following the previous work with MLP and credit scoring, there is a tendency towards one hidden layer (Zhao , et al., 2015; Luo, et al., 2017). The number of hidden nodes in the layer on the other hand is chosen based on the complexity of the problem. In the work done by Zhao et al (2015), they have an experiment with a variety of hidden nodes from 6 to 39, training 34 MLP models effectively. Albeit too few nodes in the hidden layer would be inadequate to solve a complex problem such as credit scoring or credit risk classification. The number of nodes scale and the computational difficulty of the problem make it expensive in terms of speed and power required. They found that the adequate number of nodes amount is 9. Although the increasing accuracy with additional neurons might be negligible, the increase in computational demands is not. Common transfer functions are sigmoid function and the hyperbolic tangent (Baesens, et al., 2003), where we follow the standard sigmoid function when applying a transfer function to the perceptron (Kaastra & Boyd, 1996; Luo, et al., 2017). Standard practice in credit analysis with neural network is using a shallow network with one hidden layer (Luo, Desheng, & Dexiang, 2017).

Desai et al (1996) use datasets from three credit unions to conduct credit-scoring analysis by using multilayer perceptron neural network, linear discriminant analysis, logistic regression and a mixture of experts neural network. It clarifies that multilayer perceptron neural network has a better performance than linear discriminant analysis, but only better than logistic regression marginally (Desai, Conway, & Overstreet, A comparison of neural networks and linear scoring models in the credit union environment , 1996). However, a furthermore study by Desai et al (1997) about credit scoring suggest that logistic regression outperforms the multilayer perceptron neural network at 5% significance level (Desai V. , Conway, Crook, & Overstreet, 1997). However, Salchenberger et al (1992) compared multilayer perceptron neural network with logistic regression using a dataset of 3420 financial health of saving and loans. They conclude that multilayer perceptron neural network has better performances than logistic regression model for each examined dataset (Salchenberger, Cinar, & Lash, 1992).

Tam and Kiang investigated Texas bank failure prediction from 1985 to 1987 by using a multilayer perceptron neural network model and other four techniques. The finding is that the

multilayer perceptron is most accurate model, followed by linear discriminant analysis, logistic regression, decision trees and k nearest neighbour. But there is an empirical evidence, which finds that multilayer perceptron may not be the most accurate neural network model. Mixture-of-experts and radical basis function neural network models are preferred for credit scoring applications (West, 2000).

SVM

Support Vector Machine (SVM) is a popular machine learning techniques for classification, which has three advantages. The first is fewer assumptions about the distribution or the continuity of the input variables. The second is the ability to perform nonlinear mapping. The third is the attempt to learn the separating hyper-plane when solving the maximization problem. Thank to these characteristics, SVM is a popular tool in machine learning (Yu, Wuyi, Shouyang, & Lai, 2010). Furthermore, an important feature of SVM is to put original training set into high dimensional feature space, since in high-dimension space, a linear discriminant function could substitute for non-linear separated features.

In the comparison of different machine learning approaches, Yu, et al., 2010 found that SVM performs best in the single agent case, with the argument given that it avoids being trapped at a local minima, which often occurs with Fuzzy Neural Networks for instance. The second prominent result from Yu et al (2010) is that multi-agent models do have a higher accuracy than single agent models.
Burgers (1998) find that the performance of SVM is either identical or significantly better than other competing techniques in the applications (Burgers, 1998). Considering credit scoring problem, Baesens (2003) used eight credit scoring datasets to apply various classification algorithms. It finds that SVM achieves the highest classification accuracy rate among the 17 methods tests (Baesens, et al., 2003).

Moreover, Huang , Chen, Hsu, Chen, & Wu, (2004) compares the performance in credit risk evaluation between SVM method and back propagation neural networks, concluding that SVM achieves only slight improvement over back propagation neural networks (Huang z. , Chen, Hsu, Chen, & Wu, 2004). In later research, they compare with other three techniques (neural networks, genetic programming and decision tree, suggesting that SVM approach could provide similar classification accuracy, even though there are relatively few input variables (Huang, Chen, & Wang, 2007).

Support vector machine is applied for credit rating with good performance. Hens and Tiwari (2012) refined a SVM model by reducing features and took a sample for creating credit-scoring mode. Results suggest that SVM method is competitive in terms of accuracy rate and computational time (Hens & Tiwari, 2012). Yao & Lu (2011) uses neighbourhood rough set for input feature selection and constructs a hybrid SVM-based credit scoring models for credit scoring. It finds that the best credit scoring performance could be achieved by using neighbourhood rough set and SVM based hybrid classifier. This hybrid classifier outperforms linear discriminant analysis, logistic regression and neural networks (Yao & Lu, 2011).

However, Lee etal.2006 use CART and MARS to investigate the effectiveness of credit risk evaluation. The results indicate that CART and MARS provide better performance in credit scoring accuracy than traditional logistic regression, and support vector machine (Lee, Chiu, Chou, & Lu, 2006).

Decision trees

Decision tree, also called as classification trees, is a fast and easy type of classifier to understand and interpret. An important feature of this technique is that data has few variations, which could be used to learn, produce important differences in the model (Tsymbal, Pechenizkiy, & Cunninghan, 2005). A decision tree method divide a large amount of observations into several smaller homogeneous group based on a set of rules and particular target variable (Yap, Ong, & Husain, 2011).

Davis et al (1992) apply decision tree and a multilayer perceptron neural network to conduct a research related to credit card scoring. Based on a single data partition and neural network, it concludes that a comparable level of accuracy is obtained for decision tree model and multilayer perceptron neural network (Davis, Edelman, & Gammerman, 1992). En empirical evidence find that no model outperform the other when logistic regression model, decision tree and credit scorecard mode are compared. The classification error rates of logistic regression model, decision tree and credit scorecard model are 28.8%, 28.1% and 27.9% respectively (Tap & Ong, 2011).

A research by Zhao, Xu, Kang, Kabir, Liu, & Wasinger (2015) is conducted on the accuracy of several models by analysing German, Australian, and Japanese credit dataset. It suggests that the decision tree model performed a little better than back propagation, but both techniques could achieve high accuracy rates. Galindo and Tamayo (2000) use a mortgage loan dataset to make a comparison among decision tree model, neural network, k-nearest neighbour and probit algorithms. In the findings, it indicates that decision tree model could have best performance for default, with an average 8.31% error rate (Galindo & Tamayo, 2000).

Decision tree is another good alternative method. Wang, Ma, Huang,&Xu (2012) developed a dual strategy ensemble tree rely on bagging and random subspace, which reduced the effect of noise data and redundant attributes to obtain relatively higher classification accuracy (Wang, Ma, Huang, & Xu, 2012).

Bagging scheme, as a well-known procedure, creates ensembles of classifiers for inaccurate and unstable base classifier. Those ensembles of classifiers perform well. Decision trees provide an excellent model for the Bagging ensemble scheme, which is applied for solving scoring problem (Abellan & Castellano, 2017).

Probabilistic neural network

The probabilistic neural network (PNN) is developed to be used for classification problems. PNN follows Bayesian classifier to provide a general solution for classification, which taken relative likelihood of events and priori information into consideration for prediction. The training set is used for estimating distribution functions that computing the probability of an instance being part of class. The class for a given instance is determined by combining learned patterns and a priori probability of each class.

In a research related to financial distress of Chinese public companies, probabilistic neural network has good performance for classification. Even although multivariate normality of the data is not required, probabilistic neural network can produce good prediction. In the results,

probabilistic neural network can predict company distress with short-term accuracy 87.5%, and medium-term accuracy 81.3% (Wu, Liang, & Yang, 2008). A research working on credit scoring in small-business lending compares the accuracies of models extracted by different machine learning techniques. Probabilistic neural network, as one of neural network algorithms produce the highest hit rate and the lowest type I error meaning lowest value of false positive (Bensic, Sarlija, & Zekic-Susac, 2005). The ability of neural network is investigated by Abdou, Pointon and Elmasry (2008). Several machine learning techniques, including probabilistic neural network, are compared for evaluating credit risk in Egyptian banks. It finds that neural network provide better accuracy rate than other techniques (Abdou, Pointon, & Elmasry, 2008).

Due to its fast learning capability and efficiency, the probabilistic neural network is selected. In addition, a research uses highly skewed distribution of municipal credit ratings, which is an imbalanced dataset. It suggests that it is possible to deal with the problem of imbalanced dataset by using PNN, because of the ability of PNN to classify the classes correctly in the research (Hájek, 2011).

Deep Learning

Deep learning involves creating neural networks that are longer than shallow networks. In the classic example MLP consist of 3 layers. In a deep net the number of hidden layers increase from 1 to multiple. Research by Zhao et al (2015) showed that 6 hidden layers present the best performance. This indicates that there is potential in applying deep networks to financial data. Deep Belief Networks (DBN) is one of the techniques in deep learning. In the paper by Hinton et al (2006) they propose a greedy algorithm to solve Deep Belief Networks (DBN), by stacking layers of Restricted Boltzman Machines (RBM) upon each other; showing that one can train a deep belief net on layer at a time. This approach to machine learning moves the neural network into deep networks with the increased hidden layers (Hamori, Kawai, Kume, Murakami, & Watanabe, 2018), and becomes more capable of computing complex tasks. The application of DBN upon credit risk and credit scoring is quite limited (Yu, Yang, & Tang, A novel multistage deep belief network based extreme learning machine ensemble learning paradigm for credit risk assessment, 2016; Luo, Desheng, & Dexiang, 2017). Despite the potential of deep learning, or the outperformance of ensemble techniques in terms of accuracy compared to single models (Yu, Yang, & Tang, 2016) rather few attempts have been made in advancing deep learning algorithms to classify credit risk or default. The move toward more complex neural networks onto the financial market is further demonstrated by Dixon et al (2017) by presenting an MLP model with 2 hidden layers, applying a deep neural network to forecast FX futures and commodities. However, the application of deep learning in credit analysis is limited to Credit Default Swap (Luo, Desheng, & Dexiang, 2017), and Japanese credit card application (Yu, Yang, & Tang, 2016) using an ensemble DBN with extreme machine learning. In both papers deep learning methods show positive performance compared to other machine learning techniques. This result fails to account for other deep learning techniques, or other datasets that might be larger. The limited works done on deep learning applications in finance presents a gap in the literature along with the application of big data upon credit risk

Considering credit assessment, there are two types of scoring model applied extensively: application scoring and behavioural scoring. The task of application scoring is to divide credit applicants into ‘good’ and ‘bad’ risk groups. Generally, financial and demographic information about applicants are used for modelling. As for behavioural scoring, it deals with

existing customers. In this case, information about previous loan payment is used. Whether customer's payment pattern considered is the main difference between behavioural scoring and application scoring (Khashman, Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes, 2010). Lim and Sohn (2007) proposed a behavioural scoring model that dynamically accommodates the changes of borrowers' characteristics after loan are made, which is more likely to minimize the loss because of bad creditors than static models (application scoring). The model set separate classifiers for each cluster at various time points. It can help lenders to protect themselves from bad creditors in a timely manner (Lim & Sohn, 2007). In this paper, behavioural scoring model is not included.

Research Design

This section will explore and address the approach of preparing the dataset from lending club. A description of the data structure is presented firstly. Then in order to deal with high-dimension data, first factor analysis of mixed data (FAMD) is used for selecting the variables. It followed by a correlation analysis for avoiding multi-collinearity. Thereafter k nearest neighbour approach is used to deal with missing data, followed by a brief overview of the remaining dataset.

Data

Data is obtained from lending club, which is a database contains lots of peer-to-peer online lending data. The dataset consists of approved loans from 2015 – 2017, which contains 1 299 083 observations with 101 variables. The empty columns and obvious repeat variables are eliminated to avoid multi collinearity. There are a combination of categorical variables and numerical variables, where 39 variables are categorical and 62 are numeric. After processing the data and checking for omitted values, the dataset was reduced to 86285 observations if omitting observations with missing entries. In this case, the dataset lost 93,36% of its data by excluding observations with missing values. It indicates that the missing values are supposed to be assigned. In order to solve the problem about missing value, a k nearest neighbour (KNN) approach is used. The standard 5 nearest neighbours are applied, because of the mix of categorical and numerical data.

Missing values

To input the missing values in the dataset involves the use of a built-in function in R for k – nearest neighbors. Due to size restriction and memory limit of computer, the data need to be split into 53 even segments of rows with 24 511 observations each. Traditional techniques involve either using a regression analysis to determine input value or using randomize functions. The advantage of KNN is the ability to apply to both types of data rather than cobble together different techniques. In a comparison among techniques for imputing missing data, there is an evidence that imputing the mean is the least reliable technique; and others while perhaps outperforming KNN does not show consistency of results when size of datasets varied. Therefore, KNN is selected in part due to the consistency, as well as adhering to

practical limitations. KNN imputes the values based on the average of the k nearest data points, and the default value of 5 is used in this paper.

Before computing the missing value with KNN, the composition of missing values in the variables is examined, because if there are too many missing values in one variable, KNN becomes unreliable as a method of imputation. The table with missing value is shown in table 1. It finds that the variable with the highest missing value is “total_bal_il” , which has 39984 missing values and constitute 3,077% of the total data. Since there is not a substantial number of the missing value originating from a single variable, KNN is a reliable method of imputation. The missing values per variable is illustrated in Table 1.

Table 1

Variable	Number of Missing values
loan_amnt	0
num_bc_sats	102
pct_tl_nvr_dlq	102
open_acc	102
int_rate	0
annual_inc	100
bc_util	15110
revol_bal	102
out_prncp	102
total_rec_int	102
total_rec_prncp	102
last_pymnt_amnt	102
total_bc_limit	102
dti	681
total_bal_il	39984
default	100

Variable Selection

The second problem is the large amount of variables contained in the dataset. To reduce the high dimensional data into a more compact form, conducting a factor analysis with mixed data is an effective way. Sharma (1996) suggest that a huge dataset containing lots of variables might cause several problems in any statistical analysis (Sharma, 1996). Factor analysis of mixed data (FAMD) is a method to select principal component, which is suitable for a dataset with both quantitative and qualitative variables. FAMD could be regarded as a combination between principal component analysis (PCA) for quantitative variables and multiple correspondence analysis (MCA) for qualitative variables (Pagès, 2004). The initial step is to prune the data into a more manageable size, which in turn makes the imputation of missing values more efficient. This involves a process to reduce the dimension of the dataset. It transforms a high dimension dataset into a lower dimension dataset by reducing the number of variables, which makes the data more manageable for machine learning methods, and emphasizes critical variables in the data. Due to the mix of categorical and numerical variables, principal component analysis is not appropriate, since it only works for the

numerical data. Therefore, FDMA method is used for the categorical variables, which breaks the data down into quantitative and qualitative variables. Based on the percentage of contribution to the variance of the data, the variables that contribute above average are selected.

Based on the results from FAMD, 22 variables are selected from the high dimension dataset. The selection process is based on the order of parentage contribution to the variance of the data above a certain threshold in the most prominent dimensions. Observing the scree plot (Appendix Figure 1), one can observe that the first two dimensions explain more than the other dimensions. Thereafter it flattens out. There is an examination of the percentage contribution of the variables to the variance of the data. The threshold in R was 1% as a cut-off point. In Appendix figure 2 and 3, the variable contributions to the first and second dimension are shown respectively. The low contribution percentage is an effect of the number of variables in the dataset. In addition, higher dimensions were cross-referenced briefly, and two variables were included on assumption are interest rate and loan amount, which could make loans more expensive and difficult to pay off. This method of selecting variables is not bound by previous literature, but rather attempts to break down the data as is. With the limited access of research on p2p lending, this approach enables us to select variables based on the data structure rather than theoretical framework. Additionally, the size of the data made this approach more direct. Although this paper did include other variables, such as loan amount and interest rate from the intuition, it makes this paper less pure in its approach. However, since machine-learning classification is the purpose of this paper, a data-oriented approach to variable selection was deemed appropriate.

After variable selection, in order to avoid multi collinearity, a test for correlation among those variables is conducted, shown in the correlation matrix in Appendix table 1. Only one variable should be used if there is a high correlation between two variables. Certain variables that followed other inversely or in an exact patter were removed before as they replicated another variable making it apparent without testing. Based on the correlation matrix, it shows that there is high correlation between Loan-amnt and instalment, which is 0.937, therefore, instalment is deleted, which is a similar result with (Yu & Zhu, 2015). In addition, total payment has a very high correlation with total_rec_prncp 0.97, and with loan_amnt 0.78. So, based on those two circumstances, total payment variable was deleted. Furthermore, the correlation between Revol_util and bc_util with 0.86. In addition, num_op_rev_tl is highly correlated to open acc, with a correlation of 0.84. So num_op_rev_tl and revol_util were removed from the variable list, after checking their whole correlation with other variables. The final result is 16 variables left for analysis, where 15 are input variables and 1 is the output variable.

The final display of the selected variables, and their definitions can be examined in the following Table 2.

Table 2

Variable	Definition; as stated by Lending Club
loan_amnt	The listed amount of the loan applied for by the borrower.
num_bc_sats	Number of satisfactory bankcard accounts.
pct_tl_nvr_dlq	Percent of trades never delinquent.
open_acc	The numbers of open credit lines in the borrower's credit file.
int_rate	Interest Rate on the loan.
annual_inc	The self-reported annual income provided by the borrower during registration.
bc_util	Ratio of total current balance to high credit/credit limit for all bankcard accounts.
revol_bal	Total credit revolving balance.
out_prncp	Remaining outstanding principal for total amount funded.
total_rec_int	Interest received to date.
total_rec_prncp	Principal received to date.
last_pymnt_amnt	Last total payment amount received.
total_bc_limit	Total bankcard high credit/credit limit.
Dti	A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, divided by the borrower's self-reported monthly income.
total_bal_il	Total current balance of all installment accounts.
Default	Current status of the loan involving current, default and fully paid.

Data Exploration

Loan status is considered as Y variable or the predicted class. There are 7 levels of this factor at the beginning, ranging from current to default. Due to the distribution of the various levels, it created a large imbalance problem. To work around this, 4 levels are combined into one category called default. This category is composed of charged off, defaulted and various late payment categories. After contracting the levels in loan status, it is composed of three classes, which are fully paid, current and default. Current means that applicants are still paying their loan and their payments are on time. The fully paid category means that the loan is paid off, and the default category is the combined category of multiple delinquency categories.

The dataset suffers from imbalance-data problem from observing the figure 1 that there is one dominant class "current" that composes 70% of loans and "fully paid" makes up 20%. Examining the loan status variable, it shows that the majority of the loans fall in the category of current, which means that they are paying off their loan and are on time with payments. Thereafter there is a sizable chunk of loans that are fully paid. The rest of the loans fall into various forms of late payments, a minority have defaulted and in some instances that the creditor has unilaterally deemed the debtor in default. To analyse default class and classify properly requires some more substantive data points on the desired class. To address this problem, this paper will combine the various late payments and other non-performing loan categories into a single "default" category. The adjustment is in part to solve the imbalance problem. The second adjustment is that late payments may be seen as more than an indicator of tardiness, but rather a difficulty to commit to payment obligation, which indicates default risk.

The lending club uses its own rating system based on p2p dataset and there is a clear pattern of different types of loans that they engage in, shown in figure 2. Looking at the composition of graded loans, it mainly falls within A to C grade. There is a notable decline in the loans at grade D and lower. The distribution is skewed to the left and may also in part explain the low number of defaulted loans.

Figure 1. Default Distribution

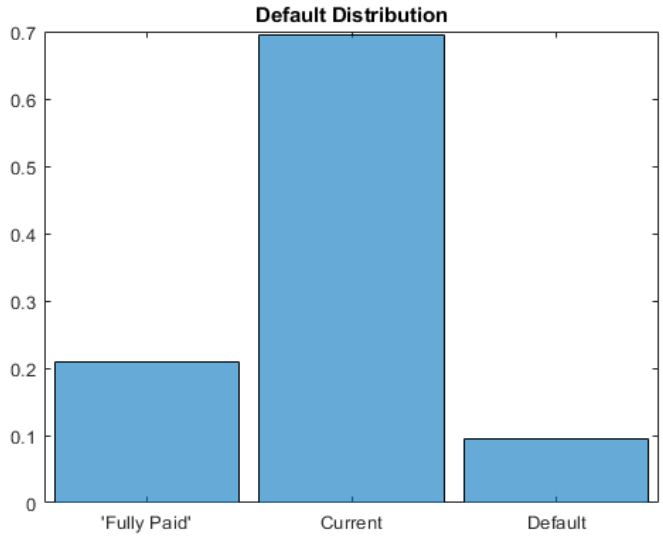
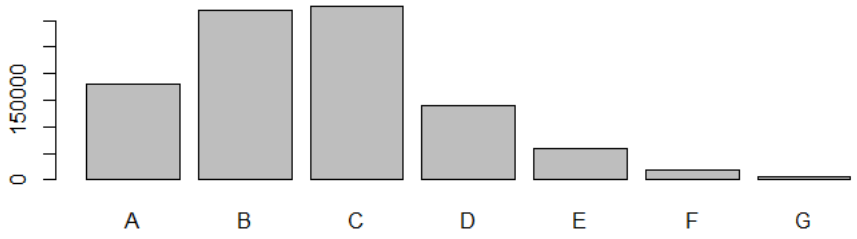


Figure 2 rating class from Lending Club based on p2p dataset



Methodology

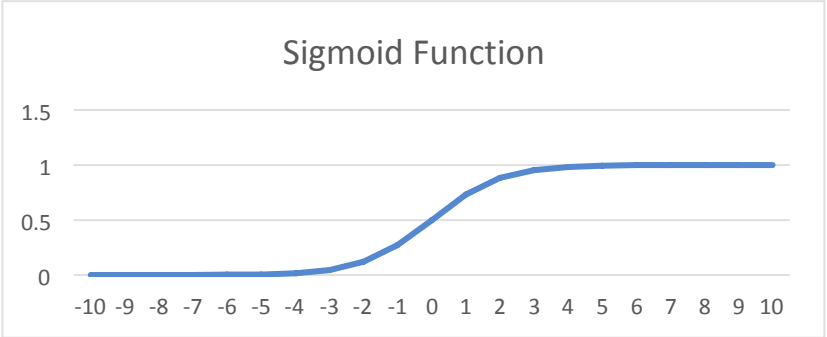
In methodology part, the methods of logistic regression, support vector machine, decision tree, multilayer perception, probabilistic neural network, Deep Learning are presented. Model specification and its application are elaborated. In addition, how those techniques applied in WEKA software is explained.

Logistic Regression

The logistic regression is often used when the dependent variable is binary, as the result falls within the range of 0 to 1. The setup for the logistic regression is the logistic function. The values of Y take either 0 or 1 depending on if it defaults or not (Luo, Desheng, & Dexiang, 2017).

$$L(x) = \frac{1}{1 + e^{-x}}$$

It is also referred to the sigmoid function in the machine learning literature and is a common activation function in neural networks.



Since this paper deals with multiclass output variable, the logistic regression requires adaption to manage categorical variables. So the multinomial logistic regression is used, which has the same basic setup with logistic regression (Luo, Wu, & Wu, A deep learning approach for credit scoring using dedault swaps, 2017). Therefore, the dependent variable is categorical instead of binary.
Assuming there is a training data containing n instances $\{x_{i,}, y_i\}$, where $x_{i,} \in R^m$ and $y_i \in \{1, \dots K\}$, the probability of the i-th instance belong to j-th class with the exception of the last class is

$$P(Y = j) = \frac{e^{\beta_j' X_i}}{1 + \sum_{k=1}^{K} e^{\beta_k' X_i}}$$

Decision Trees

Decision trees operate by splitting data into branches with leafs at the end using a split criteria. There are multiple decision tree algorithms used to create classification rules. One of the more common is CART (Classification and Regression Trees). CART uses the Gini Index (GI) splitting criteria when partitioning the tree branches. This splitting criterion can be stated as the following optimization problem (Xu, Saric , & Kouhpanejade, 2014), where the aim is to minimize the Gini Index.

$$\mathrm{GI_m} = \sum_{c=1}^{C} \hat{p}(y = c | x \in \mathrm{R_m})\, \hat{p}(y \neq c | x \in R_m)$$

The Bernoulli distribution gives the following probability

$$\hat{p}(y = c | x \in \mathrm{R_m}) = 1 - \hat{p}(y \neq c | x \in \mathrm{R_m})$$

Cross validation is used to ensure the model fits the data as accurately as possible, since each cross validation creates a possible new decision tree. To compare the different decision trees, a cost function is considered. For a given tree T with $|T|$ regions we have

$$C_\alpha(T) = \sum_{m=1}^{|T|} 1\{y_i \neq \hat{c}(x_i)\} + \alpha |T|$$

If $\alpha = 0$ the max tree will minimize the cost function. Importantly if, $\alpha > 0$ then there will be a unique pruned tree that will solve the minimization problem. Pruning the shortest branch by minimizing the cost function generates a sequence of trees $\{T_\alpha^b, \alpha = 0, \dots\}$. So for each α apply T_α^b in the b test set to get the error rate TE_α^b. Then average error rate is calculated.

$$TE_\alpha = \frac{1}{B} \sum_{b=1}^{B} TE_\alpha^b$$

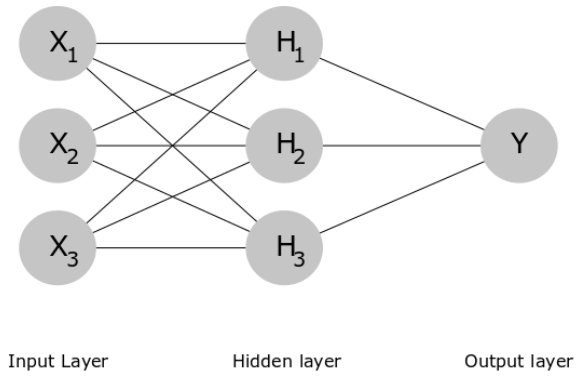
It is required to solve the value of α^* for minimizing the average error rate

$$\alpha^* = \arg\min_{\alpha} TE_\alpha$$

The step are to build the max tree on all the data, prune the tree using a sequence of $\alpha = 0$ ... and build a sequence of trees T_α . Finally, the max tree is pruned to minimize the cost complexity function $\alpha = \alpha^*$. A CART decision tree is generated.

Multilayer Perceptron

MLP is a model that is composed of an input layer and output layer with any number of hidden layers in between the input and output layers. The perceptron, which is the building block of the MLP as well as other neural networks, can vary in number within the layers. In the input layer, the number of perceptron nodes is the same as the number of explanatory variables. For the output layer will consist of 1 node.



Let's assume an MLP model with **m** perceptron nodes in **h** hidden layers and n perceptron nodes in **v** visible layer. For each layer, it has to be an activation function $f(x)$ in order to determine the output. The perceptrons in the hidden layer receive information from the perceptrons occupying the visible input layer, which in turn transfer it to the output layer. This paper will follow the traditional approach and using a sigmoid function, a special case of the logistic function, as the activation function for the perceptron in the hidden layer (Luo, Desheng, & Dexiang, A deep learning approach for credit scoring using credit default swap, 2017).

$$f(x) = \frac{1}{1 + e^{-x}}$$

This implies that the perceptron $\mathbf{h_j}$ receives information from $\mathbf{v_i}$ that are weighted $\mathbf{w_{ij}}$ given that i = 1, 2, 3…, n and j = 1, 2, 3…, m. Therefore the net input to the perceptron $\mathbf{h_j}$ is given by

$$n_j = \sum_{i=1}^{n} \boldsymbol{v_i}\, w_{ij} + b_j \ , \qquad j = \ 1, 2 \dots, m$$

Where b_j is the bias value for the node. Applying the sigmoid activation function will yield.

$$a_j = f(n_j) = f\left(\sum_{i=1}^{n} \boldsymbol{v_i}\, w_{ij} + b_j\right)$$

The last step is transferring information from the hidden layer to the output layer, following the same procedure as previous step. The output value Y for Neuron k will be

$$Y_k = \sum_{j=1}^{m} f\left(\sum_{i=1}^{n} \boldsymbol{v_i}\, w_{ij} + b_j\right) w_{jk} + b_k$$

The various parameters in the MLP is determined by an iterative process using a backpropagation learning algorithm.

Probabilistic Neural Network

A Probabilistic Neural Network (PNN) is four-layer neural network. PNN assumes a Gaussian, or normal distribution, with each class estimated using a Parzen Window (Gaganis , Pasiouras , & Doumpos , 2007).

$$\widehat{f_h}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

Where h is the smoothing parameter, and K is Gaussian kernel function. The kernel can be described as follows.

$$K(X) = \frac{1}{2\pi^{\frac{p}{2}}\sigma^p} \frac{1}{m} \sum_{i=1}^{m} \exp\left[-\frac{(X - X_i)^t (X - X_i)}{2\sigma^2}\right]$$

Where X is training sample, σ is the smoothing parameter and m is the number of training samples.

The first layer in the PNN is the input layer like any other neural network. However, rather than a series of hidden layers like MLP, the second layer is a pattern layer. Each sample in the training set has a corresponding unit in this layer where the kernel function is computed. The third layer is the summation layer where each category has a corresponding unit that sums up the kernel function computed in the previous layer. The fourth and final layer are the output layer. In our case, we will have 15 input nodes, 209 715 pattern nodes, 3 summation nodes and 1 output node.

Support Vector Machine

Support vector machine (SVM) is a classification technique that solves linearly separable problems, such that a median line can be drawn between two sets of points. The median separating the categories is as far away from the nearest point from both. Two parallel lines to the median are draw, and there is one tangential to a point in the first category, and the other tangential to the second. It will create straight between the two categories, which SVM seeks to maximize. In the cases where the problem is not linearly separable, a kernel transformation

is used to produce a dimension. Solving SVM involves solving the following Lagrange function (Boser, Guyon , & Vapnik, 1992)

$$L = \frac{1}{2}\left|\left|\bar{w}\right|\right|^2 - \sum \alpha_i [y_i D(\boldsymbol{x}_i) - 1]$$

$$st\ \alpha \geq 0$$

Where D(x) is the decision function of which category it belongs to.

$$D(\boldsymbol{x}_i) = \boldsymbol{w} * \boldsymbol{\varphi}(\boldsymbol{x}) + b$$

Solving partial differential equation of with respect to **w** and b of the Lagrange function L and inserting the back into L we get the following.

$$L = \sum_i \alpha_i - \frac{1}{2}\sum_j \sum_i \alpha_i \alpha_j y_i y_j \varphi(\mathbf{x}_\mathrm{i}) \cdot \varphi(\mathbf{x}_\mathrm{j})$$

The last dot product can be expressed as a kernel function K

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \varphi(\mathbf{x_i}) \cdot \varphi(\mathbf{x}_\mathrm{j})$$

A common kernel function that will transform the subspace to make the problem separable is (Luo, Desheng, & Dexiang, 2017).

$$e^{-\frac{||x_j - x_i||}{\sigma}}$$

Deep Learning

The application of deep neural networks in contrast to shallow neural network is in the application of hidden layers between input and output layers. Shallow networks tend to have one hidden layer. But by increasing the hidden layers, or deepening the network, the application tends towards deep learning. There are various techniques to construct deep nets, ranging from deep belief networks to recurrent neural networks. This paper takes a simple approach to deep learning by using a WEKA package called DL4jMLPClassifier, which allows stacking various forms of neural layers. Using stacked dense layers with an output layer, a simple deep learning network is created to use. Applying larger and more complex network will be outside the scope of this paper and beyond our computational capacity.

Software Application

Regarding to software application, Standard machine learning techniques, such as logistic regression, decision tree and MLP, are conducted in WEKA standard classification library. WEKA DI4jMLPClassifier is used to conduct deep learning. For the SVM WEKA was too slow to produce results therefore we shifted the work to Matlab. KNIME is similar with WEKA, which can conduct lots of machine-learning techniques that WEKA lacks of, such as Probabilistic Neural Network.

Data mining process need to build a classifier for a dataset. A classifier's job is to predict the class given by an instance. It is not difficult to predict the class of the instance. But the aim is to estimate a new instance, the one that haven't show up before. In WEKA software, using training data option could mislead the results, because the model is built based on all training data, which means that it predicts the data that is used to build classifier at the beginning. It should go beyond to see in the training data and predict outcome for new, unseen data. So, the dataset is split into two parts: the training data and the test data. Training data is used for building the model that shows how instances should be classified. As for test data, it is used to evaluate the model. The percentage split is 80%, which means that training data accounts for 80% of whole dataset, the rest 20% as test data. The reason why training data have higher percentage is that the algorithm need more data for training to see most patterns in the data, so that it could performs quite well on the remaining test data. Different split produce slight different results. Therefore, model based on training data is applied to each test instances, and the results show predicted classes for instances. WEKA compares the predicted classes with the real class defined for the instances and calculate the percentage of Correctly Classified Instances.

10-fold cross-validation is widely used for classification in WEKA. Cross-validation refers to a systematic way of doing repeated holdout, which can reduce the variance of the estimate. Repeated holdout is using the holdout method with different random-number seeds each time. Holdout method involves splitting the data into two sets, one for training the model and one for testing it. This method is best used when the dataset is large, as it requires substantial data to work with. Alternatively, if the dataset is small or medium sized cross validation is used instead to avoid over-fitting. Over-fitting is a common problem that plagues all machine-learning methods. It happens when a classifier fits the training data too tightly but has a lousy performance on independent test sets. Over-fitting is one of the important reasons why evaluation cannot be only based on the training set. 10-fold cross-validation means that one dataset is divided independently 10 separate times into a training set and a test data. With cross-validation, dataset is cut into 10 pieces. 9 of the pieces are used for training and rest for testing. With same division, another 9 pieces is used for training. After process 10 times, a different segment is used for testing each time. The result is obtained by taking the average of the 10 results. That is the whole process of 10-fold cross validation. Since the dataset is large enough with around 1 million observations, it makes holdout validation a viable candidate to avoid over-fitting. Using the split percentage of 80/20, there are 838 859 for training set and 209 715 for test set. In addition, because of error messages in readability of CSV files, some rows of the data are cut to be readable for WEKA. So the number of observations is reduced from 1 299 083 to 1 048 574, essentially losing 250 509 observations.

Results

In this paper, the accuracies of different algorithms performances are compared. The algorithms are logistic regression (LP), the multilayer perceptron (MLP), and decision tree (DT), support vector machine (SVM), Probabilistic Neural Network (PNN), and Deep Learning. In results part, AFER (accuracy rate) and ROC are the main measurements for classification performance.

Sharda and Delen (2006) uses the average percent hit ratio (APHR) as the performance metric in prediction, which is calculated by using number of correctly classified instances over total number of instances (Sharda & Delen, 2006). A larger number of APHR indicates better prediction performance. APHR is a common measurement to evaluate classification performance (Yu & Zhu, 2015). APHR has same definition with the accuracy rate that both are used wisely in previous literature (Wu, Liang, & Yang, 2008) (YU & ZHU, 2015). Accuracy refers to how many instances are classified correctly and provide measurements of model performances. As shown in table 3, Support vector machine has the highest accuracy performance, which is 97%, comparing among logistic regression (93.71%), decision tree (95.28%), the multilayer perceptron (93.62%), PNN (92,96%), and deep learning (90%). Quick observation reveals that the simple deep learning model in WEKA performs worse than the other models, with SVM being the most accurate.

ROC refers to a receiver operating characteristic curve, which is created by plotting the true positive rate against the false positive rate at different threshold settings. The ROC indicates the classification performance of a model. The performance of a classification model is better with larger ROC. The true positive rate means the probability of detection, while the false positive rate shows the probability of false alarm. The ROC also can be seen as a plot of the power as a function of type I error, indicating the sensitivity as a function of probability of false alarm. The ROC can hence be seen as the ability to distinguish positive from negative classification. The ROC and accuracy of each model are examined, it shows that they tend to follow similar patterns that the higher accuracy rate is, the higher ROC value shows. There is one notable exception with decision trees, where accuracy rate is lower than SVM but has a higher ROC than SVM. It indicates that decision tree has a powerful ability to distinguish True Positive and True Negative. There is a trade-off between Specificity and Sensitivity. ROC is a measure to take this into account. Additionally, it is worth noticing that deep learning underperforms in this context as well as PNN.

Table 3

Model	ROC	Accuracy
Logistic Regression	0,853	93,7358%
MLP	0848	93,6228%
Decision Tree	0,889	95,6388%
SVM	0,87	97%
PNN	0,673	92,957%
Deep Learning	0,621	90,0713%

Confusion Matrix

Besides the standard accuracy rate and ROC analysis, the classification algorithms could be represented by confusion matrix. If for instance an applicant status is classified as good, but is bad, it would cause a loss. This will allow for the identification of false positives and false negatives, also referred to as type 1 and type 2 error. Except accuracy performance, it is vital to know how the models perform respect to different categories, which is represented in a series of confusion matrix

In decision tree model shown in table 4,taking fully paid as an example, in the confusion matrix, the sum of total fully paid instances is 43793+59+11=43864. But in the model, 59 instances are classified incorrectly as current class, and 11 instances are classified incorrect as

default class instead of fully paid class correctly. So for fully paid class, decision tree model produces accuracy 0.9984. This model provides relatively good performance for fully paid class. Similar explanation applies for classification of current class. What's more, it is necessary to put more attention on classification of default class. 46 instances are classified as fully paid and 8110 instances are in current class, while they belong to the default class actually. Only 63% data of default class is classified correctly. It indicates that an applicant belongs to default class, while it classified into fully paid or current class. If this classification is used for granting a loan, it would cause a potential loss for the financial institution.

In logistic regression model shown in table 5,it shows that fully paid class are classified into correct class with 100% percent accuracy. Regarding to current class, the accuracy rate is high as well, around 99.8%. However, logistic regression model has poor performance on classification for default class. Only 35% of default instances from testing dataset are classified correctly, while 64.5% of default instances from testing dataset are misclassified into current. In this case, the overall accuracy rate of logistic regression is relatively low.

In multilayer perceptron model shown in table 6,it also gives good performances for fully paid classification as well as current classification. But it fails to classify default instances well. Only 35% of default instances from testing dataset are classified correctly, while 64.5% of default instances from testing dataset are misclassified into current, which has worse performance than logistic regression.

In support vector machine model shown in table 7,it provides excellent performances for fully -paid class and current class. Considering default class, the percentage of correctly classified instances increases significantly to 68.6%, comparing among 35.4% (LR), 34.1% (MLP), and 59% (DT).

Shown in table 8, Deep learning model provides high accuracy rates for classification in terms of fully paid class and current class, which are 99.587% and 99.573% respectively. Considering default class, it gives worse classification that 99.67% of default instances are classified into current class instead of default class. This would be a main reason why the accuracy rate of deep learning model is relatively low, around 90.1%.

Regarding to PNN shown in table 9,the classification performance is relatively poor than other algorithms for fully paid class. There is only 86% instances classified correctly comparing 100% corrected classified obtained from LR, MLP and SVM. For current class, the classification performance is similar with other algorithms. Considering default class, the performance is really bad. 78% of instances with default class are classified as current. It fails to classify whether the applicant is default or not.

Based on the results from confusion Matrix, most of classification models gives good performances for fully paid classification as well as current classification, while it fails to classify correctly for default class, especially PNN and deep learning. The best classification performance for default class is given by SVM, around 68%.

Table 4

Decision Tree Confusion Matrix

Actual	Classified		
	Fully Paid	Current	Default
Fully Paid	43793	59	11
Current	184	144126	1497
Default	46	8110	11889

Decision Tree Confusion Matrix as Percentage

Actual	Classified			Total
	Fully Paid	Current	Default	
Fully Paid	0.9984	0.00135	0.00025	1
Current	0.0013	0.98847	0.01027	1
Default	0.0023	0.40459	0.59312	1

Table 5

Logistic Regression Confusion Matrix

Actual	Classified		
	Fully Paid	Current	Default
Fully Paid	43864	0	0
Current	54	145569	184
Default	11	12940	7094

Logistic Regression Confusion Matrix as Percentage

Actual	Classified			Total
	Fully Paid	Current	Default	
Fully Paid	1	0	0	1
Current	0.00037	0.99837	0.00126	1
Default	0.00055	0.64555	0.35390	1

Table 6

MLP Confusion Matrix

Actual	Classified		
	Fully Paid	Current	Default
Fully Paid	43864	0	0
Current	71	145647	89
Default	18	13196	6831

MLP Confusion Matrix as Percentage

Actual	Classified			Total
	Fully Paid	Current	Default	
Fully Paid	1	0	0	1
Current	0.00049	0.99890	0.00061	1
Default	0.00090	0.65832	0.34078	1

Table 7

SVM Confusion Matrix

Actual	Classified		
	Fully Paid	Current	Default
Fully Paid	43864	0	0
Current	95	145755	0
Default	28	6250	13712

SVM Confusion Matrix as Percentage

Actual	Classified			Total
	Fully Paid	Current	Default	
Fully Paid	1	0	0	1
Current	0.00065	0.99935	0	1
Default	0.00140	0.31266	0.68594	1

Table 8

Deep learning Confusion Matrix

Actual	Classified		
	Fully Paid	Current	Default
Fully Paid	43682	179	2
Current	310	145185	312
Default	40	19979	26

Deep learning Confusion Matrix as Percentage

Actual	Classified			Total
	Fully Paid	Current	Default	
Fully Paid	0.99587	0.00408	0.00005	1
Current	0.00213	0.99573	0.00214	1
Default	0.00200	0.99671	0.00130	1

Table 9

PNN Confusion Matrix

Actual	Classified		
	Fully Paid	Current	Default
Fully Paid	37915	5948	0
Current	1448	144225	134
Default	4111	15709	225

PNN Confusion Matrix as Percentage

Actual	Classified			Total
	Fully Paid	Current	Default	
Fully Paid	0.86439	0.13561	0	1
Current	0.00993	0.98915	0.00092	1
Default	0.20509	0.78369	0.01122	1

Positive Predicted Vale and Negative Predicted Value

The confusion matrix is used to calculate the Predicted Values of the classifiers. This is the percentage of true prediction compared to the total calls. It is a measure of precision, as it measures the true call divided by the total calls for both positive and negative cases. The formal calculation can be demonstrated in the following equations.

$$PPV = \frac{TP}{TP + FP}$$

$$NPV = \frac{TN}{TN + FN}$$

This dataset is a multiclass problem with 3 classifications, which are default, current, and fully paid rather than a binomial case. Therefore, separate calculations need to be done for each category. This is also done for the sensitivity and specificity calculations in the next section. The default category is emphasized as it is of interest in this paper. Upon examination of the tables 10 below, certain patterns start to emerge. The first NPV is greater than PPV for Current and Fully Paid classes and a reversed pattern for the default class. This implies that false positives are less common than false negatives when compared to total prediction. A possible dilemma is that false negatives in default predictions are more serious than false positive from the perspective of an investor or a regulator. If a class is predicted as not default and it thereafter defaults, there is a considerable loss for the investor that put money in the investment.

Table 10

Current		
Model	PPV	NPV
Logistic Regression	0,918	0,995
MLP	0,917	0,997
Decision Tree	0,946	0,971
SVM	0,959	0,998
PNN	0,946	0,971
Deep Learning	0,878	0,986
Fully Paid		
Model	PPV	NPV
Logistic Regression	0,999	1,000
MLP	0,998	1,000
Decision Tree	0,995	1,000
SVM	0,997	1,000
PNN	0,995	1,000
Deep Learning	0,992	0,999
Default		
Model	PPV	NPV
Logistic Regression	0,975	0,936
MLP	0,987	0,935
Decision Tree	0,887	0,958
SVM	1,000	0,968
PNN	0,887	0,958
Deep Learning	0,076	0,904

True Positive Rate vs True Negative Rate

Using the results of the confusion matrix, the true positive rate (TPR) and the true negative rate (TNR) are computed. TNR is called specificity, and TPR is called sensitivity. TNR and TPR are calculated as follows

$$TPR = \frac{TP}{TP + FN}$$

$$TNR = \frac{TN}{TN + FP}$$

Where TP is true positive, FP is false positive, TN is true negative, and FN is false negative. The values will be computed in the tables 11. Observing the pattern, it is quite clear that TNR is higher than TPR in all cases except the default category. It infers that there is difficulty to classify true non-events. However the results are quite staggering when one looks at the PPV. The best performance is SVM with a sensitivity of 68%, implying that in the default category, the classification algorithm is bad at avoiding false negatives. High specificity and low sensitivity provides a difficulty regarding the effectiveness of credit risk assessments. Despite the good accuracy, the models should act rather conservatively with regard to risk assessment, instead of sacrificing sensitivity to risk for specificity to ensure that a positive prediction is accurate.

Table 11

Current		
Model	TPR	TNR
Logistic Regression	0,998368	0,797525
MLP	0,998903	0,793519
Decision Tree	0,988471	0,872176
SVM	0,999349	0,90212
PNN	0,989150041	0,70657391
Deep learning	0,995734	0,685007
Fully Paid		
Model	TPR	TNR
Logistic Regression	1	0,999608
MLP	1	0,999463
Decision Tree	0,998404	0,998613
SVM	1	0,999258
PNN	0,86439596	0,96648216
Deep Learning	0,995874	0,99789
Default		
Model	PPV	NPV
Logistic Regression	0,975	0,936
MLP	0,987	0,935
Decision Tree	0,887	0,958
SVM	1,000	0,968
PNN	0,887	0,958
Deep Learning	0,076	0,904

Computational Analysis

The computational cost is also a problem when neural networks models are used in financial applications. There are three layers (input, hidden, and output) for simplest multilayer perceptron neural network, and neural network model for credit evaluation usually use two hidden layers. The problem is that there are more layers added, leading to higher computational cost, as well as longer processing time (Khashman,2011).

Table 12

Model	Computation Time in Seconds to Build Model	Computation Time in Seconds to Test Model
Logistic Regression	317.7s	0.42s
MLP	1898.85s	0.62s
Decision Tree	451.11s	0.31s
SVM	8489,3s	2,13s
PNN	43 200s	8s
Deep Learning	14 000s	2,32s

From a computational perspective, logistic regression and decision trees are the fastest to compute, whereas SVM and MLP take a long time to train and test. From an efficiency argument and examining the accuracy of the models briefly, one could say that decision trees produce stable and reliable result with low intensity on the hardware. This may become important if datasets become larger. Both MLP model and Logistic regression model demonstrate similar accuracy and ROC result when examining default. There is however a significant difference between the two models, which is the computational difficulty as it take about 6 times longer to train the MLP model. Weka performance of decision tree is comparable to logistic regression classifier in terms of speed with about 100 seconds longer computation time to train the model, however the application seems to go faster with testing it. Support Vector Machine was conducted using MATLAB classify learner, due to the computational difficulty of the SVM in WEKA. It implies practical limitations with using SVM as a classifying method as it takes around 2,35 hours to complete. While it does pertain, SVM has a higher accuracy compared to decision trees and logistic regression. PNN and deep learning techniques require much longer time than other four techniques, showing a high computational cost, while it didn't provide better performances.

Discussion

P2P lending data is used for analysing credit assessment, conducted by different algorithms. The performances of different models are discussed. Comparing among the logistic regression (LP), the multilayer perceptron (MLP), decision tree (DT), support vector machine (SVM), probabilistic neural network (PNN), and deep learning, it finds that support vector machine (SVM) provides the most accurate performance for the classification, which has accuracy rate 97%. This finding is consistent with the results from Burgers (1998) and Baesens (2003). Burgers (1998) find that the performance of SVM is either identical or significantly better than other competing techniques in the applications (Burgers, 1998). Moreover, Baesens (2003) finds that SVM achieves the highest classification accuracy rate among the 17 methods tests (Baesens, et al., 2003). Decision tree model also produces good accuracy rate around

95.6%, which is consistency with the findings from Zhao, Xu, Kang, Kabir, Liu, & Wasinger (2015) and Galindo and Tamayo (2000). It suggests that the decision tree model performed a little better than back propagation, but both techniques could achieve high accuracy rates. Galindo and Tamayo (2000) indicated that decision tree model could have best performance for default when SVM is not included for comparison (Galindo & Tamayo, 2000). Decision trees tend to outperform logistic regression, results that are in concordance to literature (Luo, Desheng, & Dexiang, 2017), both when looking at the ROC curve and accuracy.

As for logistic regression and the multilayer perceptron neural network, the performances are relatively poor comparing with SVM and Decision tree. The accuracies are 93.7% for logistic regression, 93.6% for the multilayer perceptron (MLP). Logistic regression, as the most accurate model in traditional method, fails to compete with SVM and decision tree. What's more, the performance of the multilayer perceptron (MLP) is identical with the performance of logistic regression, which makes a contribution on the debate whether MLP has better performance than logistic regression. In previous debate, Salchenberger et al (1992) concluded that multilayer perceptron neural network has better performances than logistic regression model for each examined dataset (Salchenberger, Cinar, & Lash, 1992). However, a furthermore investigated by Desai et al (1997) suggested that logistic regression outperforms the multilayer perceptron neural network at 5% significance level (Desai V. , Conway, Crook, & Overstreet, 1997). In p2p dataset analysis, no model outperforms each other, which indicates that MLP and decision tree have similar performance based on p2p dataset.

Contrary to other findings (Luo, Desheng, & Dexiang, 2017; Yu, Yang, & Tang, 2016) this paper does not see improvements by constructing a deep net to classify default. Despite training time for computation, it failed to outperform any model. It was however conducted in a shallow manner using a built in function in WEKA, but the accuracy is well over 90% with a difficulty to classify default class. The performance the deep learning method as presented in WEKA underperforms greatly in comparison to other models, in particularly classifying the default category. Although it is inconsistent with other findings regarding deep nets, it is worth addressing that the problem may lie in the software application of deep net and could be improved using alternative techniques. It is however consistent with the literature that simply deepening the network does not necessitate and increase the performance.

PNN have relatively poor accuracy comparing with other algorithms, but it still provides good accuracy rate around 92.9%. The result from PNN is a little different with a research working on credit scoring in small-business lending, showing that probabilistic neural network, as one of neural network algorithms produce the highest hit rate and the lowest type I error meaning lowest value of false positive (Bensic, Sarlija, & Zekic-Susac, 2005). The different results might be due to the data size, since Bensic, Sarlija, & Zekic-Susac (2005) uses small-business lending, a large p2p dataset used in this paper.

A striking feature is the prevalence of false negatives in predicating defaults. It may present a problem as false negatives present a higher hazard potential than false positives. It is unclear if the possible cause is the imbalance dataset though or this is a multiclass problem rather than the traditional binomial classification problem of default. This presents a possible question for future evaluation.

A shortcoming of the paper is the relatively imbalanced data. Since the predicted class is composed of 70% as current, with only 10% as default and 20% as fully paid. It could have an

adverse effect upon classification accuracy in this paper. Furthermore, analysis is more tenable by collecting the late payments into a combined default category, but at the expense of imposing some bias upon the data. For instance, there is a difference between 30 days late payment and 60 days. This collection of categories loses any discernable differences between those late payments. However, we argue that due to the increased probability of default with incursion of late payments, the collection of those into a default category is justified, because they are still red flags and are subject to possible difficulties. Access to a more balanced dataset, or implementation of smote methods to address this imbalance problem is a potential for further research into p2p lending and default classification.

Conclusion

This paper is to compare different techniques (logistic regression, support vector machine, decision tree, multilayer perception, probabilistic neural network (PNN)), Deep Learning) to analyse credit assessment by using recent p2p lending dataset. In conclusion, Support vector machine (SVM) provides the most accurate performance for the classification, followed by decision tree, logistic regression, multilayer perceptron neural network, probabilistic neural network and deep learning. The performance of the multilayer perceptron (MLP) is identical with the performance of logistic regression. PNN was the second worst with major difficulty classifying both fully paid and default. In addition, deep learning algorithm also performed poorly despite the long training time.

The main contributions of this paper is the reapplication of machine learning techniques to an alternate dataset composed of significantly larger number of observations with deviating pattern from traditional bank loans. The findings from SVM and Decision tree are consistent with the previous literature. The results from logistic regression and MLP indicate that they are identical based on p2p dataset, which makes a contribution to the debate whether MLP out performs logistic regression. For PNN it is difficult to say if it properly accounts for the data imbalance due to the low performance of the model compared to the others. Deep learning performance is in contrast to previous work as it is the worst performing model comparing with other investigated techniques. This is potentially due to the simple approach to deep learning that this paper adopted and opens up the topic for future research. In terms of classification accuracy all models demonstrated high accuracy with all falling in the 90% accuracy in predictions. This is a clear difference to other literature that demonstrates lower accuracies around 87% (Luo, Desheng, & Dexiang 2017). A possible explanation is the size of the dataset increasing accuracy, since large p2p lending data is used. The utilizing larger datasets to classify default would increase performance of established techniques. Although this paper identifies a distinct dilemma by using these techniques, it is the tendency to prioritize accurate positive claims,meaning high specificity but low sensitivity. Upon examination, the reduced attention to classify non-default events may present possible dangers for investors as well as regulators. When these safe events are misclassified, there are consequences for respective parties.

The limitation of this paper is to comprehensibly address deep learning and is a potential for the future research. Since little work has been done in the progression of deep learning in

credit risk classification this makes it available for future research. There are variable techniques that are possible to implement such as deep belief networks, recurrent neural networks, and convolutional neural networks.

There are four implications for future research and practice. Firstly, behavioural scoring model could be included for analysing p2p lending data, since it is more likely to minimize the loss due to bad creditors than static models (application scoring). The model set separate classifiers for each cluster at various time points. It can help lenders to protect themselves from bad creditors in a timely manner (Lim & Sohn, 2007). Secondly, future research into p2p lending classification could improve upon the accuracy and the reliability of the models by addressing the data imbalance in the dataset using smote, or alternative methods. The third implication for further research into p2p lending could address the variables driving credit risk in the p2p lending market, competed to traditional financial intermediaries. This paper relied on a data-oriented approach to variable selection; however future research could establish variables through credit risk literature. The fourth subject to address is a serious consideration of sensitivity and specificity trade-off. This paper shows an inclination towards failing to identify false negatives. A contemplation between the advantages of type 1 error against type 2 error is possible in future research rather than pure accuracy, since confidence in a non-default probability falsely classified poses a threat to investors.

Bibliography

Abdou, H., Pointon, J., & Elmasry, A. (2008). Neural nets versus conventional techniques in credit scoring in Egyptian banking. *Expert Systems with Applications, 35*(3), 1275-1292.

Abellan, J., & Castellano, J. G. (2017). A comparative study on base classifiers in ensemble methods for credit scoring. *Expert Systems With Applications*, 1-10.

Altman, E. I. (1968). Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankrupcy. *The journal of Finance*, 589-609.

Baesens, B., Gestel, T. V., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, 627–635.

Baesens, B., Setiono, R., Mues, C., & Vanthienen, J. (2003). Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation. *operations research adn the management science*, 312-329.

Beck, J., & Shultz, E. (1986). The use of relative operating characteristic (ROC) curves in test performance evaluation. *Archives of Pathology & Laboratory Medicine*, 13-20.

Bensic, M., Sarlija, N., & Zekic-Susac, M. (2005). Modelling Small - Business Credit Scoring by Using Logistic Regression, Neural Networks and Decision Trees. *Intelligent Systems in Accounting, Finance and Management*, 133–150.

Bolton, P., Freixas, X., & Shapiro, J. (2012). The Credit Ratings Game. *The Journal of Finance , 67*(1), 85 - 111.

Boser, B. E., Guyon , I. M., & Vapnik, V. N. (1992). A Training Algorithm for Optimal Margin Classiers. Pittsburgh: COLT '92 Proceedings of the fifth annual workshop on Computational learning theory Pages 144-152 .

Brown, I., & Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, 3446-3453.

Burgers, C. J. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 121–167.

Caldieraro, F., Zhang, J. Z., Cunha Jr, M., & Shulman, J. D. (2018). Strategic Information Transmission in Peer-to-Peer Lending Markets. *Journal of Marketing, 82*, 42 - 63.

Davis, R., Edelman, D., & Gammerman, A. (1992). Machine learning algorithms for credit-card applications . *IMA Journal of Mathematics Applied in Business and Industry* , 43-51.

Desai, V., Conway, D., Crook, J., & Overstreet, G. (1997). Credit scoring models in credit union environment using neural network and generic algorithms. . *IMA Journal of Mathematics Applied in Business & Industry* , 323–346 .

Desai, V., Conway, J., & Overstreet, G. (1996). A comparison of neural networks and linear scoring models in the credit union environment . *European Journal of Operational Research* , 24-37.

Eggermont, J., Kok, J., & Kosters, W. (2004). Genetic programming for data classification: Partitioning the search space. *In Proceedings of the 2004 symposium on applied computing*, 1001–1005.

Finlay, S. &. (2012). Instance sampling in credit scoring: An empirical study of sample size and balancing. *International Journal of Forecasting*, 224-238.

Finlay, S. (2011). multiple classfier architectures adn their application to credit risk assessment. *European Journal of Operational Research*, 368-378.

Gaganis , C., Pasiouras , F., & Doumpos , M. (2007). Probabilistic neural networks for the identification of qualified audit opinions . *Expert Systems with Applications* , 114–124 .

Galindo, J., & Tamayo, P. (2000). Credit risk assessment using statistical and machine learning: basic methodology and risk modelling applications . *Computational Economics* , 107–143 .

Guo, Y., Zhou, W., Chunyu, L., Liu, C., & Xiong, H. (2016). Instance-based credit risk assessment for investment decisions in P2P lending. *European Journal of Operational Research, 249*, 417–426.

Hájek, P. (2011). Municipal credit rating modelling by neural networks . *Decision Support Systems* , 108–118 .

Hamori, S., Kawai, M., Kume, T., Murakami, Y., & Watanabe, C. (2018). Ensemble Learning or Deep Learning? Application to Default Risk Analysis. *Journal of Risk and Financial Management, 11*(1).

Hand, D., & Henley, W. (1997). Statistical classification methods in consumer credit scoring : a review. *Journal of then royal statistical Society*, 523-541.

Hastie, T., Tibshirani, R., & Friedman, J. (2008). Unsupervised Learning. *The Elements of Statistical Learning*, 485-585.

Henley, W. (1995). statistical aspects of credit scoring. *The open university Miltion keynes* .

Hens, A., & Tiwari, M. (2012). Computational time reduction for credit scoring: An integrated approach based on support vector machine and stratified sampling . *method. Expert Systems with Applications*, 6774–6781.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets . *Neural Computation, 18*(7), 1527 - 1554 .

Huang, C.-L., Chen, M.-C., & Wang, C.-J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert System with Applications*, 847-856.

Huang, z., Chen, h., Hsu, c.-j., Chen, w.-h., & Wu, s. (2004). Credit rating analysis with support vector machines and neural networks: A market comparative study. *decision support system*, 543-558.

Kaastra, I., & Boyd, M. (1996). Designing a neural network for forecasting financial and economic time series . *Neurocomputing , 10*, 215 - 236.

Khandani , A. E., Kim, A. J., & Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*, 2767 - 2787.

Khashman, A. (2010). Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes. *Expert System with Application*, 6233-6239.

Khashman, A. (2011). Credit risk evaluation using neural networks: Emotional versus conventional models . *Applied Soft Computing* , 5477–5484 .

Khashman, A. (2011). Credit risk evaluation using neural networks: Emotional versus conventional models . *Applied Soft Computing* , 5477–5484.

Kruppa, J., Schwarz, A., Arminger, G., & Ziegler, A. (2013). Consumer credit risk: Individual probability estimates using machine learning. *Expert Systems with Applications*, 5125 - 5131.

Käfer, B. (2018). Peer-to-Peer Lending – A (Financial Stability) Risk Perspective. *Review of Economics, 69*(1), 1–25.

Lee, T.-S., Chiu, C.-C., Chou, Y.-C., & Lu, C.-J. (2006). Mining the customer credit using classification and regression tree and multivariate adaptive regression splines. *Computational Statistics & Data Analysis*, 1113-1130.

Lessmann, S. B. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 2473–2480.

Lim, M. K., & Sohn, S.-Y. (2007). Cluster-based dynamic scoring model. *Expert System with Applications*, 427-431.

Longstaff, F. A., Mithal, S., & Neis, E. (2005). Corporate Yield Spreads: Default Risk or Liquidity? New Evidence from the Credit Default . *The Journal of Finance, 60*(5), 2213 -2253.

Luo, C., Desheng, W., & Dexiang, W. (2017). A deep learning approach for credit scoring using credit default swap. *Engineering Applications of Artificial Intelligence, 65*, 465 - 470.

Luo, C., Wu, D., & Wu, D. (2017). A deep learning approach for credit scoring using dedault swaps. *Engineering Applications of Artificial Intelligence*, 465-470.

Pagès, J. (2004). Analyse Factorielle De Données Mixtes: Principe Et Exemple D'application. *Laboratoire de mathématiques appliquées*.

Pan, J., & Singleton, K. J. (2008). Default and Recovery Implicit in the Term Structure of Sovereign CDS Spread. *The Journal of Finance, 63*(5), 2345 - 2384.

Renault, T. (2017). Intraday online investor sentiment and return patterns in the U.S. stock market. *Journal of Banking and Finance, 84*, 25 - 40.

Sakprasat, S., & Sinclair, M. C. (2007). classification rule mining for automatic credit approval using genetic programming]. *evolutionary computation*.

Salchenberger, L., Cinar, E., & Lash, N. (1992). Neural networks:a new tool for predicting thrift failures. *Decision Sciences*.

Sharda, R., & Delen, D. (2006). Predicting box-office success of motion pictures with neural networks . *Expert Systems with Applications , 30*, 243-254.

Sharma, S. (1996). *Applied Multivariate Techniques.* New York: John Wiely & Sons,Ins.

Shumway, T. (2001). Forecasting Bankruptcy More Accurately: A Simple Hazard Model. *The Journal of Business*, 101-124.

Tap, B. W., & Ong, H. S. (2011). Using data mining to improve assessment of credit worthiness via credit scoring models. *Expert Systems with Applications*, 13274-13283.

Tsymbal, A., Pechenizkiy, M., & Cunninghan, P. (2005). Diversity in search strategies for en semble feature selection. *information fusion*, 83-98.

Wang, G., Ma, J., Huang, L., & Xu, K. (2012). Two credit scoring models based on dual strategy ensemble trees . *Knowledge-Based Systems*, 61–68.

West, D. (2000). Neural network credit scoring model. *Computers & operation research*, 1131-1152.

Wu, D., Liang, l., & Yang, Z. (2008). Analyzing the financial distress of Chinese public companies using probabilistic neural networks and multivariate discriminate analysis . *Socio-Economic Planning Sciences* , 206–220 .

Xiao, J., Xie, L., He, C., & Jiang, X. (2012). Dynamic classifier ensemble model for customer classification with imbalanced class distribution. *Expert Systems with Applications*, 3668-3675.

Xu, X., Saric , Z., & Kouhpanejade, A. (2014). Freeway Incident Frequency Analysis Based on CART Method. *Promet - Traffic & Transportation*, 191 - 199.

Yao, P., & Lu, Y. (2011). Neighborhood rough set and SVM based hybrid credit scoring classifier. *Expert Systems with Applications, 38*(9), 11300-11304.

Yap, B., Ong, S., & Husain, N. (2011). Using data mining to improve assessment of credit worthiness via credit scoring models. *Expert Systems with Application*.

Yeh, I.-C. &.-h. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 2473–2480 .

Yu, J., & Zhu, Y. (2015). *A data-driven approach to predict default risk of loan for online Peer-to-Peer(P2P) lending.* Hangzhou: 2015 Fifth International Conference on Communication Systems and Network Technologies .

YU, J., & ZHU, Y. (2015). A data-driven approach to predict default risk of loan for online Peer-to-Peer(P2P) lending. *Fifth International Conference on Communication Systems and Network Technologies* .

Yu, L., Wuyi, Y., Shouyang, W., & Lai, K. K. (2010). Support vector machine based multiagent ensemble learning for credit risk evaluation. *Expert Systems with Applications*, 1351-1360.

Yu, L., Yang, Z., & Tang, L. (2016). A novel multistage deep belief network based extreme learning machine ensemble learning paradigm for credit risk assessment. *Flexible Services and Manufacturing Journal, 28*(4), 576 - 592.

Zhao , Z., Xu, S., Kang, B. H., Kabir, M. M., & Liu, Y. (2015). Investigation and improvement of multi-layer perceptron neural. *Expert Systems with Applications, 42*, 3508–3516.

Zhao, Z., Xu, S., Kang, B. H., Kabir, J., Liu, Y., & Wasinger, R. (2015). Investigation and improvement of multi-layer perceptron neural networks for credit scoring . *Expert Systems with Applications* , 3508-3516.

Appendix

Table 1 Correlation Matrix

	loan_amnt	num_bc_sats	pct_tl_nvr_dlq	open_acc	int_rate	annual_inc	bc_util	revol_bal	revol_util	installment	out_prncp	total_pymnt	total_rec_int	total_rec_prncp	last_pymnt_amnt	total_bc_limit	dti	total_bal_il	num_op_rev_tl
loan_amnt	1.000	0.221	0.084	0.182	0.127	0.298	0.062	0.318	0.114	0.947	0.635	0.781	0.614	0.487	0.268	0.368	0.039	0.146	0.164
num_bc_sats	0.221	1.000	0.150	0.638	-0.053	0.134	-0.176	0.298	-0.120	0.211	0.123	0.141	0.107	0.129	0.062	0.626	0.075	0.041	0.756
pct_tl_nvr_dlq	0.084	0.150	1.000	0.107	-0.060	-0.004	-0.041	0.108	-0.040	0.067	0.041	0.055	0.019	0.057	0.040	0.217	0.066	0.003	0.140
open_acc	0.182	0.638	0.107	1.000	-0.007	0.138	-0.106	0.235	-0.145	0.170	0.096	0.125	0.121	0.106	0.059	0.391	0.185	0.323	0.837
int_rate	0.127	-0.053	-0.060	-0.007	1.000	-0.082	0.234	-0.034	0.220	0.162	0.110	0.050	0.392	-0.057	0.060	-0.217	0.138	0.036	-0.012
annual_inc	0.298	0.134	-0.004	0.138	-0.082	1.000	0.011	0.287	0.047	0.284	0.176	0.190	0.131	0.178	0.095	0.268	-0.122	0.192	0.087
bc_util	0.062	-0.176	-0.041	-0.106	0.234	0.011	1.000	0.178	0.862	0.080	0.034	0.063	0.179	0.021	-0.019	-0.243	0.130	0.013	-0.162
revol_bal	0.318	0.298	0.108	0.235	-0.034	0.287	0.178	1.000	0.251	0.300	0.187	0.210	0.182	0.187	0.084	0.482	0.099	0.084	0.245
revol_util	0.114	-0.120	-0.040	-0.145	0.220	0.047	**0.862**	0.251	1.000	0.129	0.069	0.092	0.202	0.047	-0.006	-0.166	0.125	0.022	-0.211
installment	**0.937**	0.211	0.067	0.170	0.162	0.284	0.080	0.300	0.129	1.000	0.552	0.597	0.567	0.514	0.269	0.332	0.042	0.135	0.160
out_prncp	0.635	0.123	0.041	0.096	0.110	0.176	0.034	0.187	0.069	0.552	1.000	-0.144	0.334	-0.245	-0.324	0.231	0.052	0.117	0.081
total_pymnt	0.585	0.141	0.055	0.125	0.050	0.190	0.063	0.210	0.092	0.597	-0.144	1.000	0.570	0.972	0.716	0.214	-0.006	0.058	0.117
total_rec_int	0.614	0.107	0.019	0.121	0.392	0.131	0.179	0.182	0.202	0.567	0.334	0.570	1.000	0.370	0.072	0.100	0.071	0.073	0.105
total_rec_prncp	0.487	0.129	0.057	0.106	-0.057	0.178	0.021	0.187	0.047	0.514	-0.245	**0.972**	0.370	1.000	0.791	0.214	-0.027	0.045	0.102
last_pymnt_amnt	0.268	0.062	0.040	0.059	0.060	0.095	-0.019	0.084	-0.006	0.269	-0.324	0.716	0.072	0.791	1.000	0.107	-0.018	0.039	0.053
total_bc_limit	0.368	0.626	0.217	0.391	-0.217	0.268	-0.243	0.482	-0.166	0.332	0.231	0.214	0.100	0.214	0.107	1.000	0.032	0.085	0.444
dti	0.039	0.075	0.066	0.185	0.138	-0.122	0.130	0.099	0.125	0.042	0.052	-0.006	0.071	-0.027	-0.018	0.032	1.000	0.157	0.122
total_bal_il	0.146	0.041	0.003	0.323	0.036	0.192	0.013	0.084	0.022	0.135	0.117	0.058	0.073	0.045	0.039	0.085	0.157	1.000	0.045
num_op_rev_tl	0.164	**0.756**	0.140	**0.837**	-0.012	0.087	-0.162	0.245	-0.211	0.160	0.081	0.117	0.105	0.102	0.053	0.444	0.122	0.045	1.000

Figure 1

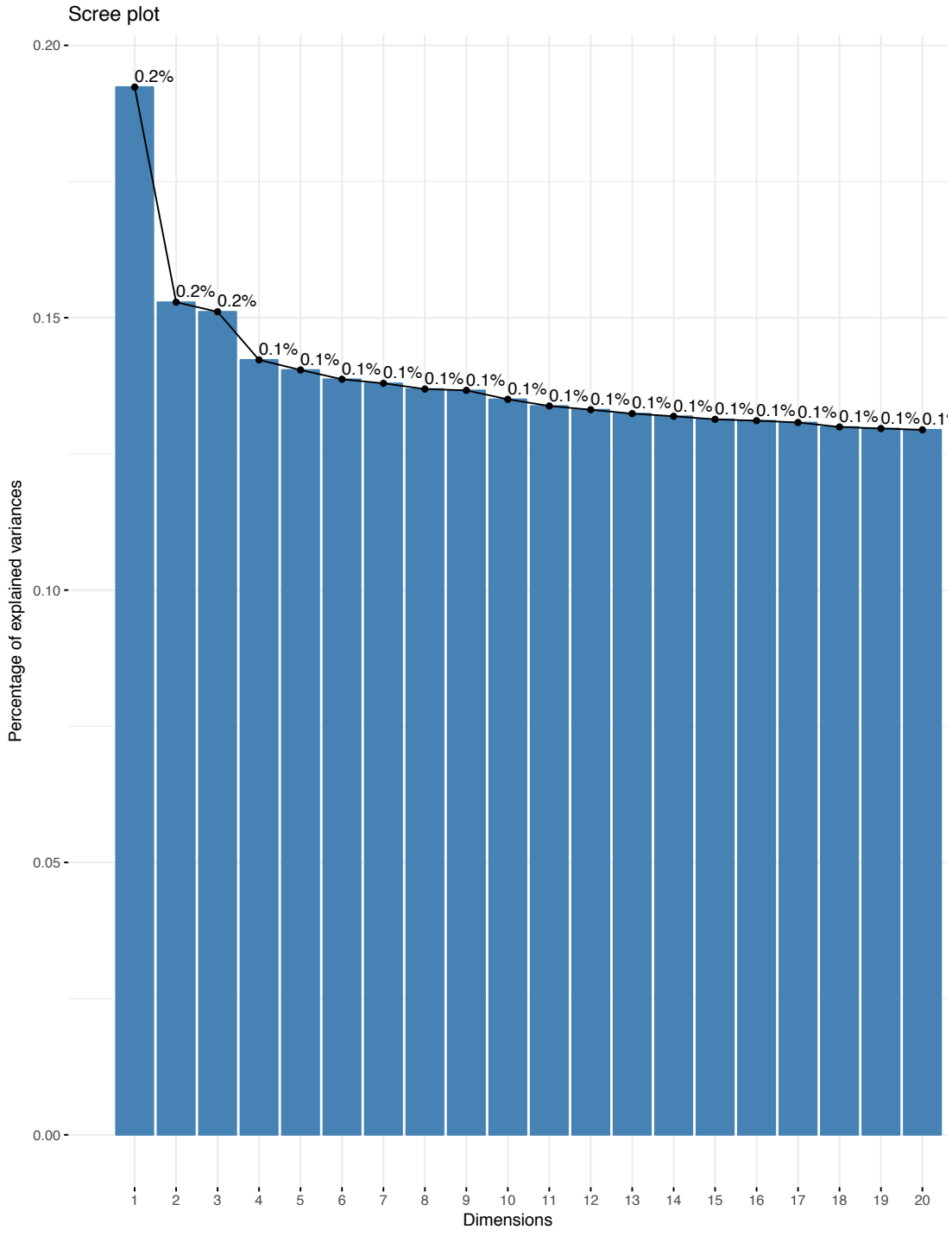

Scree plot
Percentage of explained variances
Dimensions
0.20
0.15
0.10
0.05
0.00
0.2%
0.2%
0.2%
0.1%
0.1%
0.1%
0.1%
0.1%
0.1%
0.1%
0.1%
0.1%
0.1%
0.1%
0.1%
0.1%
0.1%
0.1%
0.1%
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

Figure 2

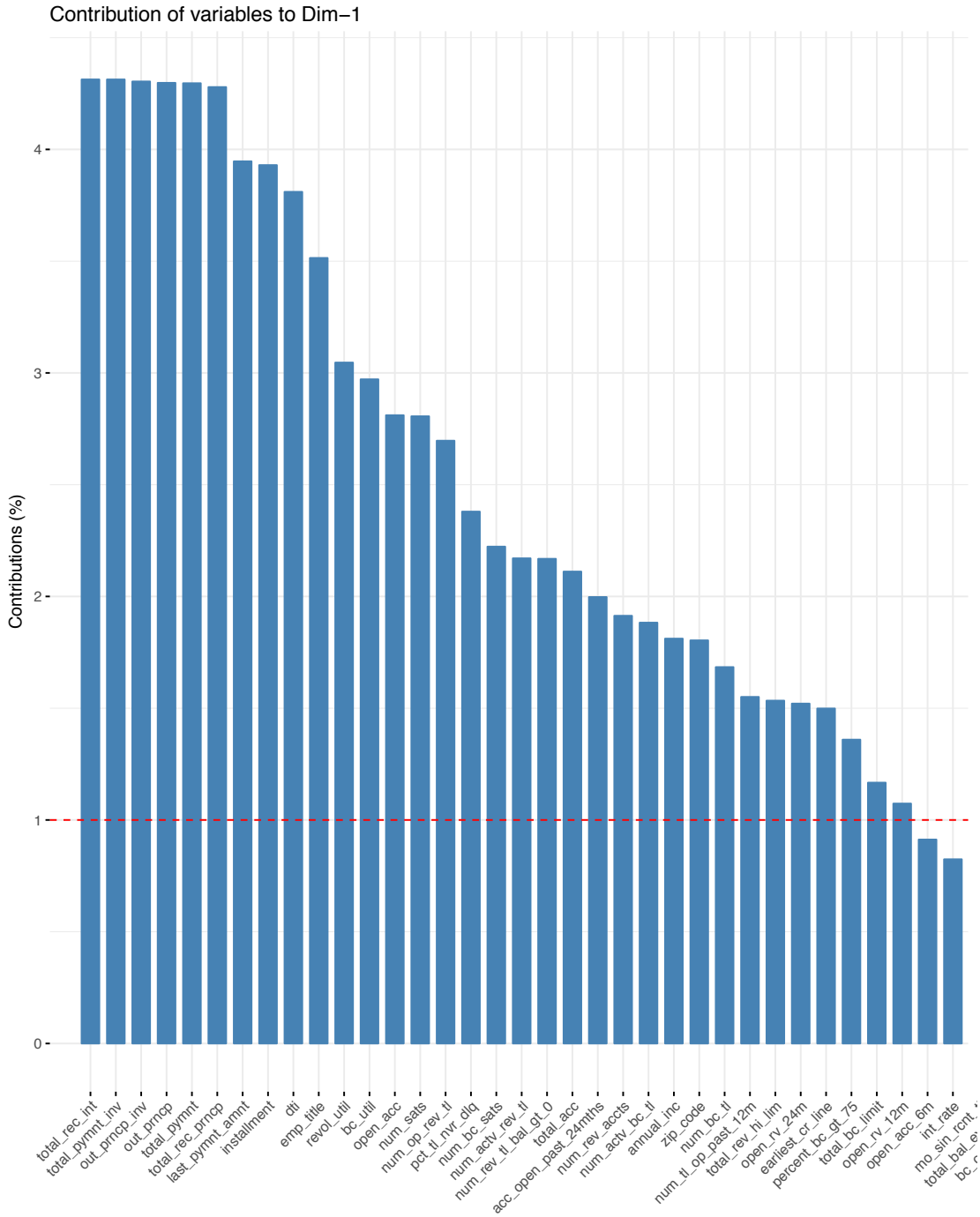
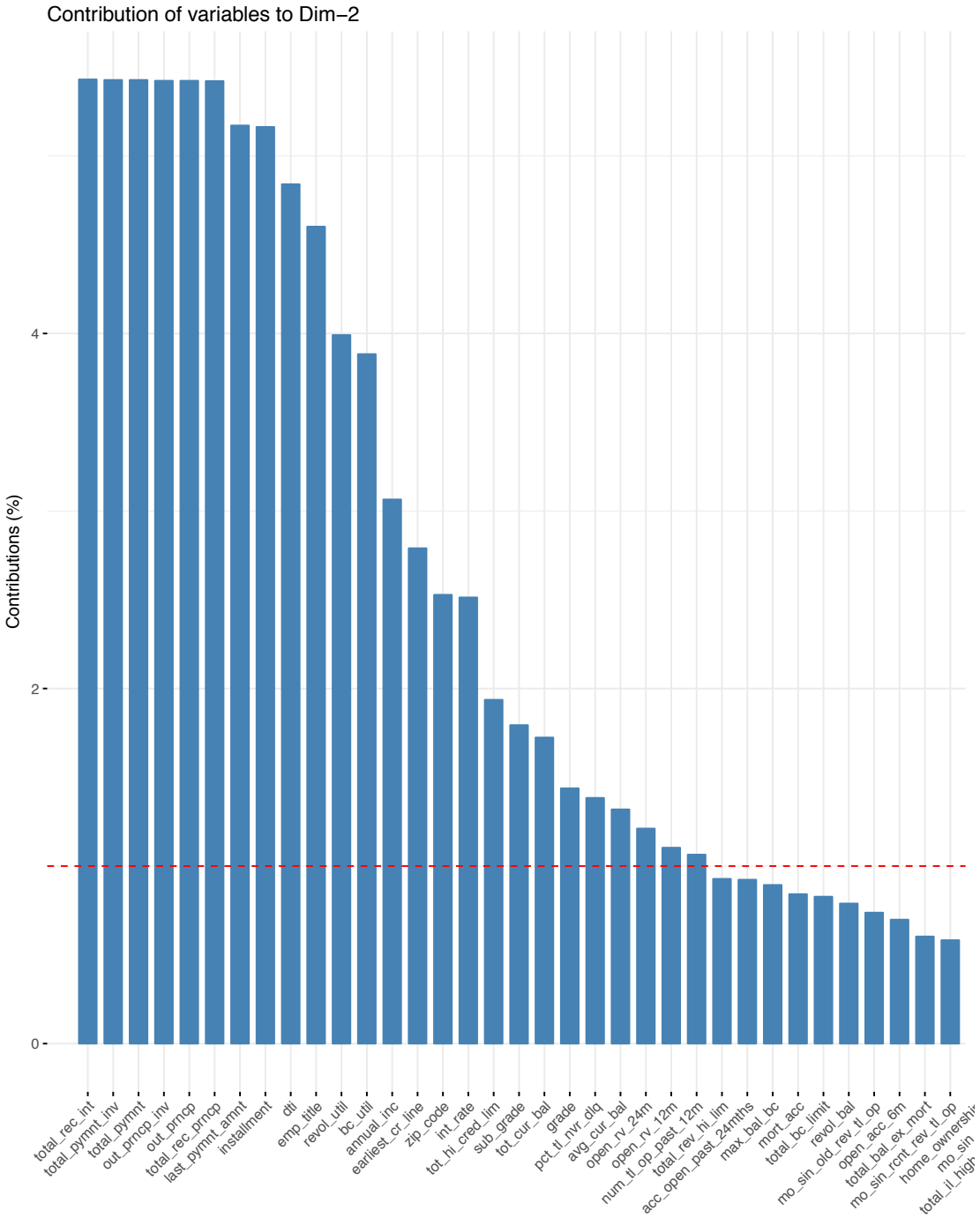

Contribution of variables to Dim−1
Contributions (%)
0
1
2
3
4
total_rec_int
total_pymnt_inv
out_prncp_inv
out_prncp
total_pymnt
total_rec_prncp
last_pymnt_amnt
installment
dti
emp_title
revol_util
bc_util
open_acc
num_sats
num_op_rev_tl
pct_tl_nvr_dlq
num_bc_sats
num_actv_rev_tl
num_rev_tl_bal_gt_0
total_acc
acc_open_past_24mths
num_rev_accts
num_actv_bc_tl
annual_inc
zip_code
num_bc_tl
num_tl_op_past_12m
total_rev_hi_lim
open_rv_24m
earliest_cr_line
percent_bc_gt_75
total_bc_limit
open_rv_12m
open_acc_6m
int_rate
mo_sin_rcnt_
total_bal_e
bc_

Figure 3


Contribution of variables to Dim−2
Contributions (%)
4
2
0
total_rec_int
total_pymnt_inv
total_pymnt
out_prncp_inv
out_prncp
total_rec_prncp
last_pymnt_amnt
installment
dti
emp_title
revol_util
bc_util
annual_inc
earliest_cr_line
zip_code
int_rate
tot_hi_cred_lim
sub_grade
tot_cur_bal
grade
pct_tl_nvr_dlq
avg_cur_bal
open_rv_24m
open_rv_12m
num_tl_op_past_12m
total_rev_hi_lim
acc_open_past_24mths
max_bal_bc
mort_acc
total_bc_limit
revol_bal
mo_sin_old_rev_tl_op
open_acc_6m
total_bal_ex_mort
mo_sin_rcnt_rev_tl_op
home_ownershi
mo_sin
total_il_high

Stockholm University
SE-106 91 Stockholm
Tel: 08 – 16 20 00
www.sbs.su.se