An Improved Partially Stacking Blend Model for Credit Risk Assessment of Internet Loan

Master's Thesis

for acquiring the degree Master of Science (M.Sc.)

in Master Program in Economics and Management Science

Submitted by

PengCheng Liu

Student Number: 561220

Submitted to:

Prof. Dr. Stefan Lessmann

Prof. Dr. Max Klimm

Humboldt University of Berlin

School of Business and Economics

Chair of Information Systems

Berlin, September 2017

List of abbreviations

BBM	the single Best Base Model
BM	Base Model
CV	Cross Validation
DBM	the base model differs from BBM to the largest scale
DBM_Sub	DBM that takes only part of samples
IPSBM	Improved Partially Stacking Blend Model
LHS	Left Hand Side
LR	Logistic Regression
PCC	Pearson's Correlation Coefficient
RF	Random Forest
RHS	Right Hand Side
XGB	XGBoost

List of Figures

List of Tables

Abstract

In the field of credit scoring, imbalanced data sets frequently occur as the number of defaulting loans in a portfolio is usually much lower than the number of observations that do not default. Under the scheme of misclassification costs, the state-of-the-art ensemble methods used for credit risk assessment, i.e. Random Forest and XGBoost etc., tend to misclassify plenty of 'good' instances as 'bad' when these instances are close to the decision boundary. This type of misclassification has an impact on the Internet banking industry because the negative impact of mouth of word from the disappointed rejected users may be substantially enlarged by the Internet.

Stacking is a widely-used algorithm to address this problem. However, empirical studies found that the performance of direct stacking is insignificant and unstable in the context of severe data imbalance. In early 2017 the 'Partially stacking blend based user credit assessment model' was proposed by a team from Sun Yat-sen University and it has been claimed that their model could mitigate the negative impacts of the weak base estimators in stacking. However, their research gives little details, and there is no further published work to justify their study yet.

To close the research gaps, this paper firstly provides instructions for practitioners to carry out this model. In addition, this research describes an empirical study of the predicted behaviour of this model based on four real-world and up-to-date Internet loan data sets. The experimental evaluation shows that the model has only slight potential to improve the prediction accuracy. Lastly this study proposes a practical approach to tuning the critical parameter of this model.

Keywords: stacking, XGBoost, credit risk assessment, imbalanced data

1 Introduction

Internet finance, a newborn industry, or a new direction of traditional finance has received considerable attention from all over the world (Zhang 2015). Third Party Payment, Crowdfunding, P2P Lending, Microloan and Consumer Finance etc. compose the main product matrix of Internet finance (Liu 2015). In contrast to conventional banking sector, it is relatively challenging for the Internet financial companies to strike a balance in growing rapidly and sustainably at the same time: Internet financial Start-ups are under considerable pressure to occupy the market in a short time; nonetheless, an slight increase in default rate may cause severe loss or even bankruptcy (Wang et al. 2015). In this context a high criteria for credit scoring is favored, nonetheless this is at the cost of slower growth.

This dilemma challenges the current state-of-the-art credit risk assessment algorithms. Empirical investigations (Malekipirbazari & Aksakalli 2015; Zheng zb et al. 2017) reveal that, in the context of imbalanced data, decision-tree based ensemble methods such as Random Forest (RF) and XGBoost (XGB) are superior in identifying best of the best and worst of worst borrowers. However, they exhibit a decrease in relative performance for tricky instances in between. In other words, such classifiers tend to misclassify a large body of ‘good’ instances as ‘bad’. In spite of a high overall accuracy, companies may suffer from the negative impact of the electronic word of mouth (Barrutia & Echebarria 2005) from the disappointed customers, adverse selection of customers (Dell’Ariccia et al. 1999), growth stagnation or even bankruptcy (Wang et al. 2015).

In this context there is a strong demand for a mechanism that is able to improve the predictive accuracy of the tricky instances which are difficult for decision trees based ensemble methods to deal with, the marginal effect of such progress would be great to the Internet financial agencies. One direction to explore is heterogeneous ensemble, which exploits the diversity of skillful predictions from different models (Duan et al. 2007) by compensating each other so as to achieve an improved overall performance. Indeed, the success of heterogeneous ensemble has been demonstrated in many studies. However, given that a multi-model contains information from all participating models including the less skilful ones, the question remains as to why, and under what conditions, a multi-model

can outperform the best participating single model (Weigel et al. 2008). In practice, it is argued by empirical works (Zheng zb et al. 2017) that a direct multi-model ensemble may perform worse than the best participating individual classifier. In this sense, it makes sense to provide more transparency or human interventions in heterogeneous ensemble.

In 2017 a successful trial in this field is achieved by the team of Zheng et al. from Sun Yat-sen University: their study proposed a well-direct stacking model that could correct the errors made by XGB. Thanks to this new model, they claimed to improve their online AUC in a machine learning challenge for credit risk modeling by 0.003[1].

To be more specific, their model consists of the following steps: firstly, a number of base models are fitted. Secondly, the tricky samples that are supposed to be misclassified by XGB are forwarded to be fitted by another model at Level Two; hence, these samples obtain more accurate evaluations. Eventually, the predictions of samples fitted by both XGB and the different model are blended as the final prediction. In this way this novel model combines diverse models that take good care of different samples, consequently the overall performance is supposed to be superior to the ordinary stacking.

In spite of the excellent idea, until now the model has not yet received much attention: no evidence in this research has been formally discussed, nor have any further empirical studies been released. In this context, the current paper is proposed to close the research gaps. Put differently, the objective of this paper is to: firstly, elaborate the original model by providing more details for Zheng et al.'s article so that practitioners could put it into practice. Secondly, to evaluate the performance of this model by an empirical investigation into four real-world up-to-date and large-scale Internet loan data sets. Finally, to improve the model usability and performance by proposing a new parameter tuning approach.

The remainder of this paper is organized as follows. Firstly, some background knowledge on ensemble methods and their applications in credit scoring of Internet banking are given. Secondly, the instructions to implement this model are presented. Thirdly an empirical study provides evidence on the performance of this model. Finally, some conclusions are drawn.

[1] http://bbs.pkbigdata.com/static/417_detail.html

2 Ensemble Methods in Credit Scoring

This chapter briefly introduces the background of ensemble methods and how these ensemble methods are being employed in credit scoring of Internet banking.

2.1 Ensemble Methods

The objective of ensemble learning is to combine the predictions of several base estimators built with a given learning algorithm, so as to improve generalizability and robustness over a single estimator (Pedregosa et al. 2011). The mechanism behind this idea is that each learning algorithm uses different methods to represent the knowledge and different learning biases so that the hypothesis space will be explored from different perspectives with the aim of generating a pool of diverse classifiers. Thus, when their predictions are merged, the resultant model is expected to be more accurate than each individual member (Dietterich 2000).

Generation of ensembles can be categorized into two types: 1) homogeneous, if the base learning model is built from the same learning algorithm, and 2) heterogeneous, where different learning algorithms are combined to generate the base learning models (Rooney et al. 2004). For example, the outputs of decision tree classifiers could be combined with the outputs of support vector machines to create a heterogeneous ensemble (Gilpin & Dunlavy 2008). Bagging and Boosting are the most representative examples of algorithms for generating homogeneous ensembles of classifiers. Similarly, Stacking has become a commonly used technique for generating ensembles of heterogeneous classifiers since Wolpert presented his study entitled *Stacked Generalization* in 1992 (Ting & Witten 1997).

As the representative examples of homogeneous ensembles, Bagging has received remarkable attention from the scientific community. In accordance with Bühlmann (2011), the Bagging procedure turns out to be a variance reduction scheme, at least for some base procedures. In many cases, Bagging methods constitute a fairly simple way to improve with respect to a single model, without making it necessary to adapt the underlying base algorithm. As they provide a way to reduce overfitting, Bagging methods work best with strong and complex models. With respect to Breiman (2001), RF is a typical

implementation of Bagging. It is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the data set and use averaging to improve the predictive accuracy and curb overfitting.

In contrast to Bagging, Boosting methods are primarily reducing the model bias of the base procedure (Bühlmann 2011). As an important example, Gradient Boosted Decision Trees (GBDT) is a generalization of Boosting to arbitrary differentiable loss functions. According to Friedman (2001), the term "Gradient Boosting" is proposed in the paper *Greedy Function Approximation: A Gradient Boosting Machine (Friedman 2001)*. It is an accurate and effective off-the-shelf procedure. The main idea of GBDT is that, as an ensemble method, GBDT combines a great number of tree models to make a much more accurate and efficient ensemble model. By continuously iterating, each round of iteration generates a new tree and cases that were not correctly classifies in the last round would be assigned higher weight in this round; hence, the overall performance could improve through iteration. Tree boosting has been shown to give state-of-the-art results on many standard classification benchmarks (Li 2012).

Similar to GBDT, AdaBoost belongs to the Boosting family as well. The core principle of AdaBoost is to fit a sequence of weak learners on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction (Hastie et al. 2003).

Afterwards, XGB is an implementation of GBDT in particular designed for speed and performance (Chen & Guestrin 2016). XGB is short for "Extreme Gradient Boosting". It has been widely used by data scientists to achieve state-of-the-art results on many machine learning challenges: for example it is the champion solution of Higgs Boson Machine Learning Challenge (Chen & He 2015). Within two years XGB became the state-of-the-art solutions on machine learning challenges (Zhou et al. 2017).

The latest critical development of GBDT tools is LightGBM (Meng et al. 2016) from Microsoft[2]. It uses histogram based algorithms to accelerate training process and reduce memory consumption and also incorporates advanced network communication algorithms

[2] https://github.com/Microsoft/LightGBM

to optimize parallel learning (Ranka & Singh 1998; Jin & Agrawal 2003). Same as XGB, LightGBM has already been widely examined in machine learning competitions though it has been available for only a few months. According to (Liu et al. 2017), in recent 4 Kaggle competitions, at least 3 champion teams have employed LightGBM in their solutions, which indicates its potential widespread use in real-world applications .

In addition to homogeneous ensemble learning, the heterogeneous ensemble method Stacking (stacked generalization) is a scheme for minimizing the generalization error rate of one or more generalizers. In particular, Stacking works by deducing the biases of the generalizers with respect to a provided learning set (Wolpert 1992). This deduction proceeds by generalizing in a second level model fitting whose inputs are, for example, the prediction outputs of the original generalizers when taught with part of the learning set and trying to guess the rest of it, and whose output is, for example, the correct guess. Another heterogeneous ensemble method is Blending. It is a word introduced by the Netflix winners on Kaggle (Jahrer et al. 2010). Blending is rather similar to Stacking, however, instead of creating out-of-fold predictions for the train set, a small holdout set of the train set is created in Blending. The stacker model then trains on this holdout set only.

2.2 Credit Risk Assessment of Internet Banking

In recent years Internet banking business has grown rapidly all over the world, particularly in China and other emerging economies (Zhang 2015). With the expansion of targeting market, it is possible that riskier customers, who lack historical credit records, such as students and peasantries in remote parts of China, are eligible to obtain credit without any required collateral (Li & Wang 2017). These individuals hardly had any access to financing from formal lenders before the internet finance revolution (Li Zhou & Takeuchi 2010).

In spite of new opportunities, the high risks associated with agricultural lending (Maurer 2014), micro-credit (Francis et al. 2017), consumer credit (Duong et al. 2017) and other relative risky niche markets substantially challenge the current risk control system. To be more specific, the old-fashioned system is fragile in the face of the tremendously increasing loan requests, consummate professional frauds, rapidly changing external environment such as stock market volatility as well as the slow model iteration etc. In particular, in the section of credit risk evaluation, the conventional modeling tools

encounter much more sophisticated data nowadays: the data set has features consisting of only limited and indirect information but having considerably complex nonlinear relationships. It is also possible that the data is unstructured, high dimensional, sparse, incomplete and even seriously biased. In particular the high dimensionality poses significant statistical challenges and renders many traditional classification algorithms impractical to use (Pappu & Pardalos 2014).

Realizing that the major potential pitfall is credit risk (Wei 2015) and the shortness of out-of-date algorithms in toolkits, Internet finance companies start to implement a variety of innovations to cope with credit risk. For instance, implemented technologies include Big Data (Yang et al. 2017), Facial Recognition (Leong et al. 2017), Voice Recognition, Id-Mapping, Location Based Risk Management and Natural Language Processing etc. Take Big Data as an example. Alibaba Group and Jingdong Group have started their own consumer credit business with their data advantages to expand their financial business, and strive to gain a foothold in the important strategy of consumer finance (Zhao 2017).

In the task of credit risk assessment or credit scoring, machine learning algorithms have been explored and proven to be of great practical value. Especially ensemble learning has been one of the most active areas of recent research in machine learning. In particular, in recent years ensemble methods have gradually become one of the most effective and popular tools for credit risk assessment with a large body of empirical investigations revealing that ensemble classifiers can substantially improve credit risk analysis (Wang et al. 2011; Yu et al. 2008; Bequé & Lessmann 2017).

Particularly RF (Breiman 2001) used to be considered as the best off-the-shelf classifier for credit risk evaluation because it is easy to use and has high accuracy (Lessmann et al. 2015). In the year of 2017 XGB is becoming the standard method in credit scoring of Internet banking industry in China[3]. For instance, CA Fintech uses XGB to help his clients identify true creditworthiness of a potential loan borrower[4], and Alibaba Cloud deployed XGB in its Big Data processing platform MaxCompute[5] to facilitate Big Data Risk Controlling.

[3] https://zhuanlan.zhihu.com/p/25862872
[4] https://www.cafintech.com/category/pro-mode.html?t=mb
[5] https://www.alibabacloud.com/product/maxcompute

3 An Improved Partially Stacking Blend Model

Based on the work of Zheng et al. (2017) this chapter proposes an improved ensemble model that lets another classifier fix the errors made by XGB and finally blend their predictions together so as to increase the overall predictive power. The chapter is organised this way. Firstly, the phenomenon that XGB tends to misclassify certain samples in imbalanced data set and the reasons for this phenomenon are discussed. Secondly, an improved ensemble model is proposed to address the errors made by XGB. Finally, the model is explained in detail.

3.1 Misclassifications of XGBoost

XGB makes its name owing to its advantages of fast speed and high accuracy (Ren et al. 2017). Nevertheless this does not necessarily mean that it could keep the power in constantly dealing with all the observations in a data set. In accordance with the study (Zheng zb et al. 2017), XGB performs great in dealing with samples that have default probabilities close to 0 or 1. In contrast, it exhibits a decrease in relative performance in differentiating tricky samples whose predicted values are close to the decision boundary.

The errors that XGB tends to make are measured by its prediction instability on every single observation, or in other words the high volatility in the default-probability based ranking of each observation. This is because a classifier is considered to be accurate when its classification error is lower than that obtained when the classes are randomly assigned (Sesmero et al. 2015). After values of all the hyper parameters in addition to the rounds of iteration are fixed, a good classifier gives accurate, consistent and robust prediction results independently on the change of iteration rounds as long as the iteration rounds are large enough for the model to converge. By contrast, a bad classifier will give random results when the value of iteration rounds varies, because it is unable to classify these samples properly. This randomness reveals the inability of this classifier in classifying this type of samples.

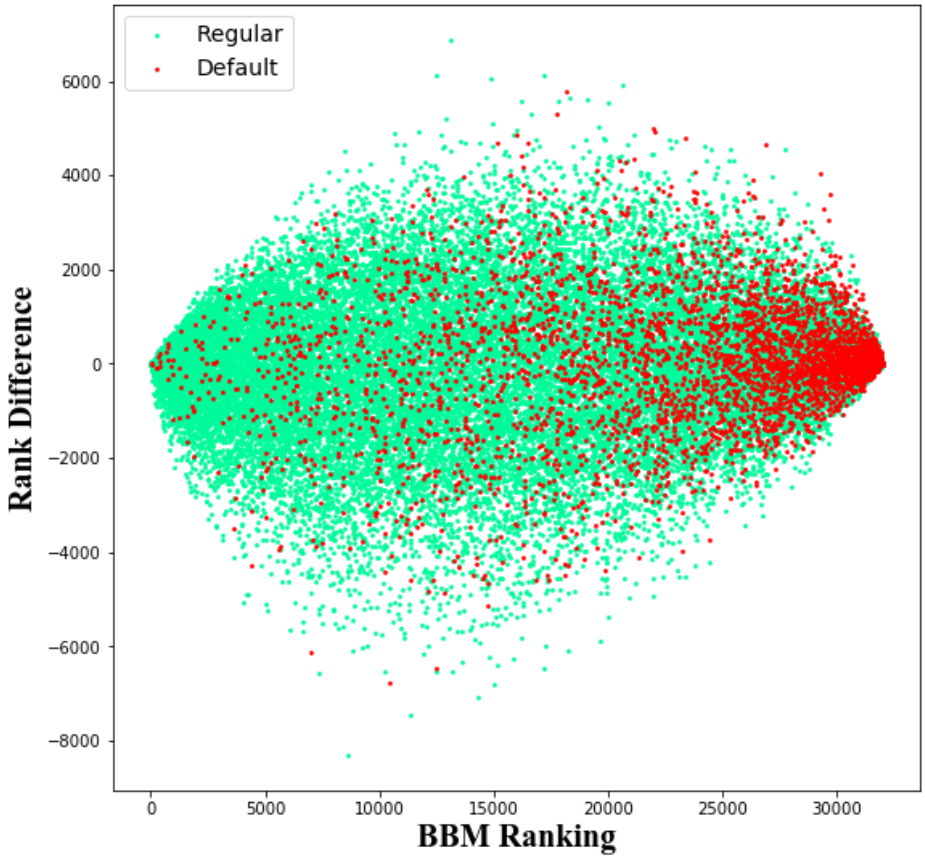

Figure 3.1: Random Estimations on the Tricky Instances

Figure 3.1 visualizes this argument. Based on the data from QianHai Credit Reference (See Chapter 4.1), two XGB models are employed to fit the data with 32,000 instances. Fixing all the values of hyper parameters except for rounds of iteration, XGB_2000 iterates for 2,000 rounds while XGB_1000 for 1,000 rounds, this difference triggers great change in the prediction results of the models. More specifically, the prediction results are presented in the format of rankings of each instance's default-possibility. For example, one user is ranked No.1 because this user has the probability to default at 0 while another one is ranked No.32000 because this user is considered to default with 99.9% possibility. Put differently, the predicted probability is converted into ranking in this scenario, 0 means Ranking No.1 while 1 indicates Ranking No.30000 for instance. In short, the closer the probability to zero, the less risky to default, and the closer the dot to the Left Hand Side (LHS) in the chart. By contrast, the closer the predicted possibility to 1, the more likely to default, then the closer the dot to the Right Hand Side (RHS) of the diagram.

As can be seen in the chart, the X axis is the ranking for all 32,000 observations by XGB_2000, which is the single best base model (BBM). In this sense a large body of the

default cases (red) gather together at the end close to 32,000. This indicates that XGB_2000 is classifying them correctly. For the type who paid back on time (green), XGB_2000 has done a good job as well since many of green points locate close to zero.

Meanwhile, the Y axis indicates the difference between each observation's ranking in XGB_2000 and that in XGB_1000 (Rank_2000 - Rank_1000). To note, the range of the difference is around (-6000,6000). According to the egg-wise shape, one can argue that XGB is accurate in extreme instances that are close to either 0 or 1 and especially give robust evaluation on these samples, since the difference (values on Y) is much smaller than the values of samples locate in the middle of the line. These samples are those nearby the decision boundary and XGB clearly classifies them randomly. This implies that XGB is naive to these tough samples.

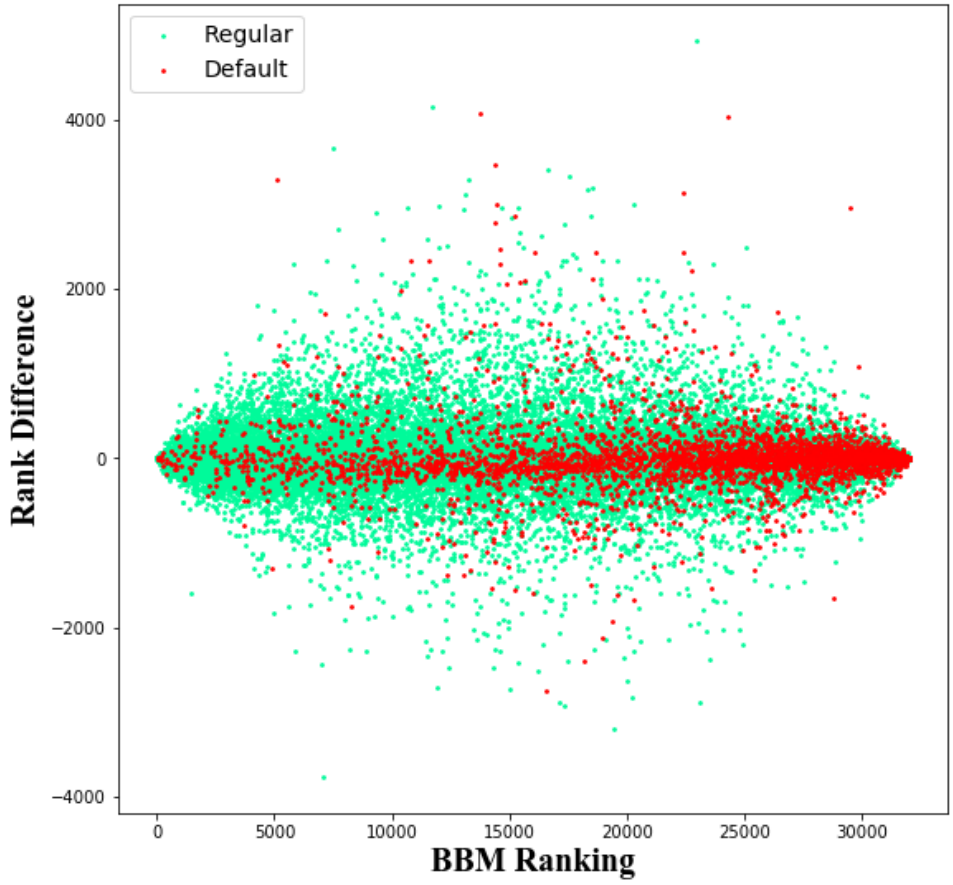

Figure 3.2: Relative Constant Estimations on the Tricky Instances

In contrast to XGB, the shape of Logistic Regression (LR) on the same data is much flatter (Figure 3.2). In comparison to (-6000,6000) of XGB, it is important to note that LR has a

main range of volatility from -2000 to 2000 only. Hence LR gives much more robust evaluations on all the observations in the entire data set than XGB. Even though LR (AUC 0.73) performs in overall poorly than XGB (AUC 0.75).

In a nutshell, the two plots vividly demonstrate the fact that, on the one hand XGB is outstanding in detecting the extreme cases, which makes XGB the superior individual classifier in many machine learning tasks. On the other hand XGB is weak in the tough region. The following paragraphs strives to explain the reasons for this phenomenon.

It is important to note that, this phenomenon is particularly obvious in seriously biased data, which means that the number of regular observations substantially exceeds the number of instances that default. In this scenario it is unavoidable for a large number of 'good' to be distributed around 'bad' at the end on the RHS of the ranking line. For example, data used in Figure 3.1 and Figure 3.2 has default/normal ratio 13:87, there are consequently a mass of green points at the 'bad' end. In this case, it is challenging for the classifier to correctly classify these instances.

To overcome this obstacle, classifiers tend to learn detailed rules and unavoidably capture noise simultaneously. This causes the problem of overfitting. In order to curb overfitting, some techniques such as boosting, regularization, pruning and constraint of tree depth are employed by the advanced ensemble classifiers such as XGB. Thanks to these techniques, the problem of overfitting could be significantly mitigated. However, this is at the cost of misclassifying the tricky samples that are close to the decision boundary as 'bad'. For example, many of the samples with high volatility in the chart above are good samples (green). This is mainly because of the setting of misclassification costs that bad cases in general have much higher misclassification cost than normal cases.

To obtain an ensemble of classifiers that outperforms all its members, the base learners must be both accurate and diverse. In order to compensate XGB the other classifier should be diverse from it in the sense that they make errors at different instances. Linear model, for instance LR, is a great alternative. Owing to the more straightforward and consistent rules of classification, linear models are able to perform better on complicated samples than sophisticated algorithms, though they may perform poorly in overall level.

In a nutshell, owing to the anti-overfitting techniques and misclassification costs, XGB tends to misclassify the good cases as bad when the samples are tricky to deal with. In this sense, it is not difficult to acknowledge that XGB performs poorly in the region where regular samples and default samples are twisting together. By contrast, simpler models, LR for example, could give them more accurate estimation due to their simpleness.

3.2 Fixing Misclassifications of XGBoost

Concerning the errors that XGB tends to make as well as the reliable evaluations on these misclassified samples from LR, it is not difficult to propose an ensemble solution that coordinates these models to increase the overall generalizability. The Improved Stacking Blend Model (IPSBM) takes into account the differences of classifiers and assigns them different samples rather than blending all samples to multi-models together.

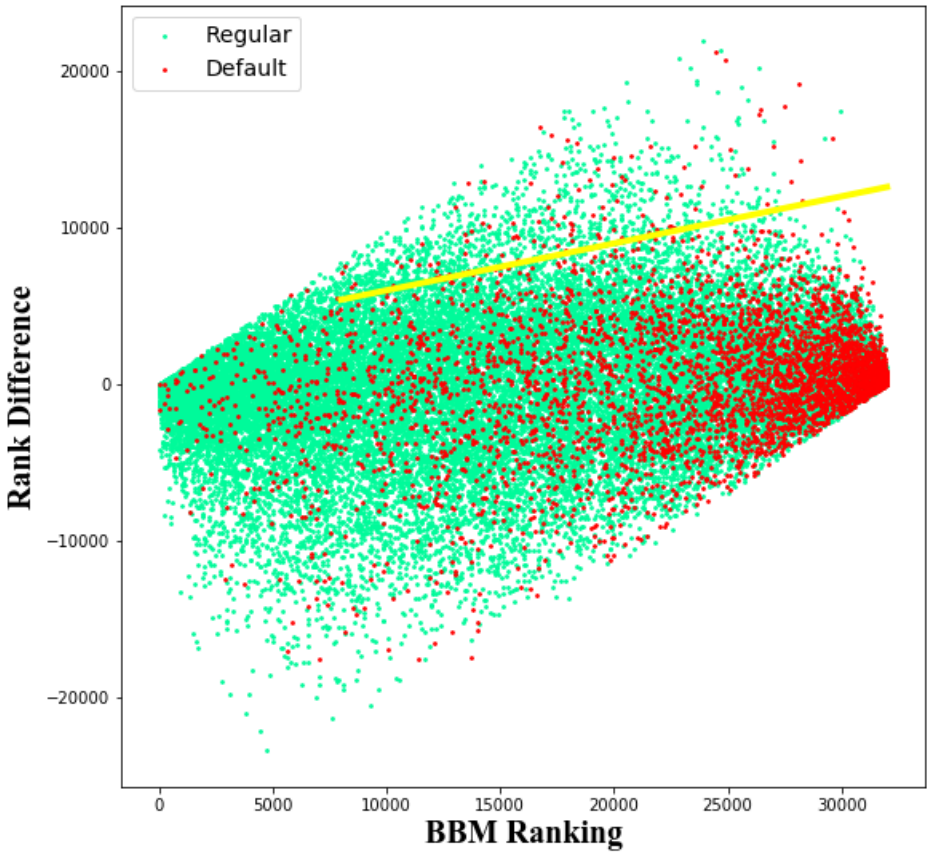

Figure 3.3: The Ranking Difference between BBM and DBM

The main procedure of this model is as follows. Firstly, the data is trained by a variety of Base Models (BMs) in Level One and the single best base model, which is named here as BBM, is determined. X axis in Figure 3.3 illustrates the prediction results of BBM. Secondly, the BM differs from BBM to the largest scale is selected and named here as DBM. Thirdly, the differences of the prediction results (probabilities converted to rankings) of the two models are calculated, which is the Y axis in the chart.

Subsequently, samples above the yellow line are trained in the Level Two by DBM_Sub, which is the DBM that takes only part of samples. Based on the new ranking from DBM_Sub, the Top L instances in the subsamples that are considered to be 'good', their predicted probabilities will be set to 0 manually. Finally, the results of BBM in Level One and the results of modified instances from DBM_Sub in Level Two are blended together.

3.3 Detailed Model Setting

In the previous section the main idea of IPSBM is introduced. In this part instructions to implement this model are elaborated in detail. As shown in Figure 3.4, the two-level model primarily consists of three steps. The following subsections provide more details on how each step is carried out.

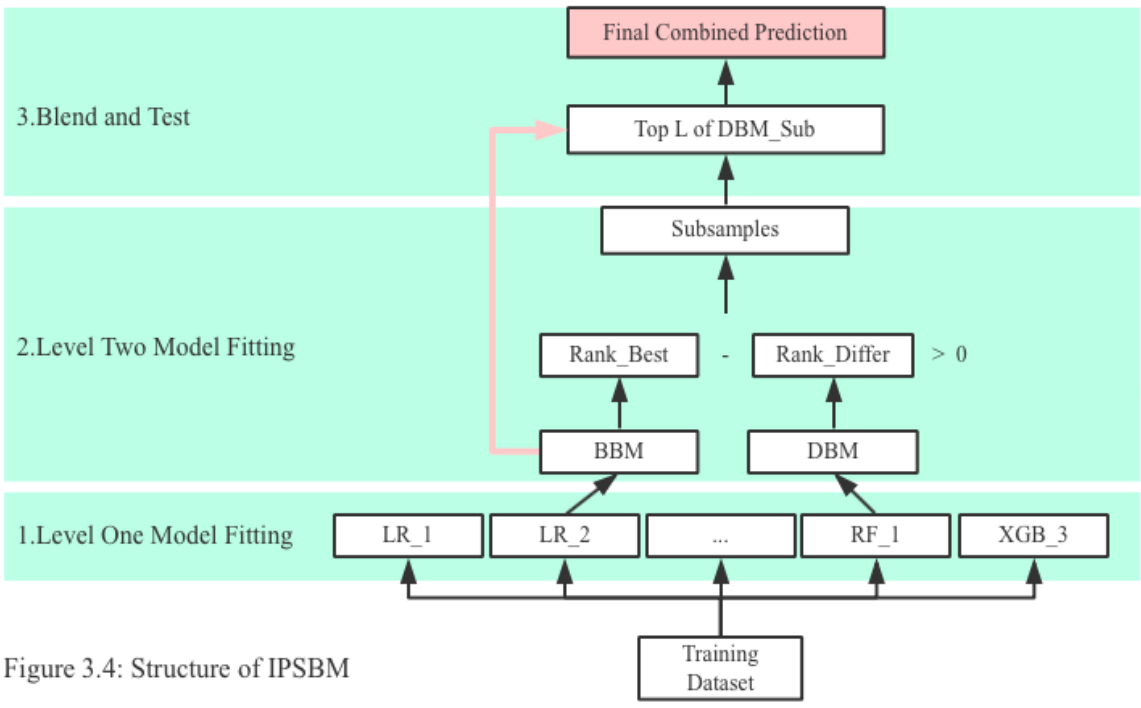

Figure 3.4: Structure of IPSBM

3.3.1 Level One Model Fitting

As can be seen from the diagram below, the main task of Step One is to train each BM and measure its performance by average AUC. In particular, as shown by the two arrows on the LHS in the chart, the horizontal arrow indicates that each layer represents a Cross Validation (CV) procedure where a BM such as LR and XGB trains the entire data set and calculates the average AUC of accumulated scores in each CV. By contrast, the vertical arrow tells that such CV procedure will be gone through by all the K BMs.

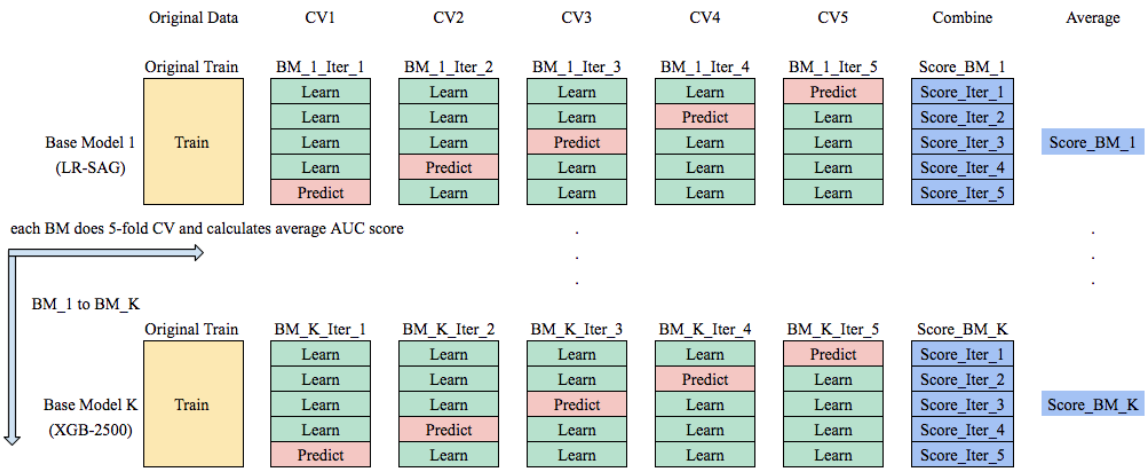

Figure 3.5: Train Base Models---Overall

To note, the reason for choosing AUC as the unique evaluation metric is that AUC shows the ability of the classifier to rank the positive instances relative to the negative instances (Fawcett 2006). This is in line with the scheme of credit rating which represents the rank-orderings of creditworthiness (Cantor & Packer 1995) or in other words default risk in accordance with Moody's[6]. In this sense, AUC is an appropriate metric for this research. Other metrics, for instance Kolmogorov-Smirnov (KS) test, are in favor of this scenario as well (Bradley 2013), however this paper focuses on AUC solely.

3.3.2 Level Two Model Fitting

Based on the average AUC of all BMs in Level One, the model fitting in Level Two completes the following tasks. Firstly, the BBM and DBM are determined. Specifically, the BM that achieves the highest average AUC is selected as BBM. Take the empirical research of QianHai data for example. Table 3.1 indicates that an XGB model turns out to

[6] https://www.moodys.com/sites/products/ProductAttachments/Moody%27s%20Rating%20System.pdf

be the BBM owing to its outstanding performance in CV. Note that the best performing classifier or value of parameter in each table of this research is highlighted by bold face. Furthermore, under the criteria of Pearson's Correlation Coefficient (PCC) (Pearson 1909), DBM is defined as the model which has the lowest PCC value with BBM. In the example below, the BM of LR-SAG becomes the most different base model from the single best base model.

Table 3.1: Choosing BBM and DBM

BM_Id	BM	AUC	PCC with BBM
xgb2000	XGB, 2000 iterations	**0.751**	1
xgb1000_6	XGB, 1000 iterations, higher mis-cost, lower eta, deeper	0.747	0.918
xgb2000_6	XGB, 2000 iterations, higher mis-cost, lower eta, deeper	0.746	0.920
lr_sag_1000	LR-SAG, 1000 iterations	0.740	0.920
lr_sag_1500	LR-SAG, 1500 iterations	0.739	0.778
gbdt100	GBDT, 100 estimators	0.739	0.785
xgb1000_2	XGB, 1000 iterations, higher mis-cost	0.739	0.791
xgb1000_4	XGB, 1000 iterations, higher mis-cost, lower eta	0.739	0.792
lr_sag	LR-SAG	0.738	**0.746**
lr_liblinear	LR-Liblinear	0.737	0.771
lr_newton	LR-Newton	0.737	0.792
ada100	Adaboost with RF as base estimator, 100 estimators	0.736	0.806
ada50	Adaboost with RF as base estimator, 50 estimators	0.734	0.815
gbdt50	GBDT, 50 estimators	0.730	0.805
ada20	Adaboost with RF as base estimator, 20 estimators	0.729	0.773
rf500	RF, 500 estimators	0.724	0.754
rf200	RF, 200 estimators	0.723	0.754
rf100	RF, 100 estimators	0.721	0.766
lr_lbfgs	LR-LBFGS	0.719	0.770
xgb2500_2	XGB, 2500 iterations, higher mis-cost	0.705	0.766

Secondly, the input for Level Two model fitting is chosen. Based on the ranking difference of each instance, which is the difference between an ascending sort of the prediction from BBM and that from DBM (Rank_Best - Rank_Differ), samples that own a relatively large

positive value in the ranking divergence and distribute close to the ‘bad’ end on the ranking line are selected into the sub data set for Level Two model training.

As the critical criteria in this research, ranking difference reveals the fact that which samples are likely misclassified by BBM. More specifically, in reality, most of these instances pay back regularly (green dots) but BBM considers them remarkably risky so that they receive a large number in rankings. By contrast DBM gives them moderate rankings, so that these instances obtain smaller numbers in ranking than that from BBM. Consequently by taking a ranking difference, a large positive value screens out instances that may be misclassified.

It is notable that when taking the difference of the outputs from multiple different models, some problems could arise. Not all predictors are perfectly calibrated; they may be over- or underconfident when predicting a low or high probability. Or it is possible that the predictions clutter around a certain range. To deal with this trouble, in this paper, the output of BBM and DBM are log transformed before they are being compared.

Afterwards, the selected data is forwarded to DBM_Sub, which is normally a simple LR. By doing so, the subsamples could be fitted by more appropriate rules so that the errors made by BBM are potentially able to be fixed. Concerning the reasons for choosing LR as DBM_Sub, firstly, in accordance with top solutions of machine learning challenge on Kaggle[7], stacking model is argued to be highly inclined to overfitting. In this sense, it is in favor of doing a simple model that mitigates the overfitting risk. More importantly, with respect to experiment results, linear models provide more constant evaluations in the entire data set. Taking these factors into account, LR is the preferred model in this scene.

3.3.3 Blend and Test

Thanks to the Level Two Model Fitting, each instance in the sub data set obtains a renewed ranking. In the final stage, the top L instances, which are samples that have the least possibilities to default, are changed by zero on their predicted results. Afterwards their results are blended with results of BBM as the final combined prediction. In this way, the misclassifications are likely to be corrected.

[7] https://mlwave.com/kaggle-ensembling-guide/

In order to measure the performance of IPSBM, the AUC on a hold-out test of the three models in interest are compared. Specifically, the models consist of the original BBM, IPSBM and a normal stacking model that takes the prediction outputs of all BMs in Level One as input. In addition to the comparison of algorithm performance, another issue to be evaluated in the empirical study is the tuning approach of parameter L.

It is noticeable that the value of L is critical to the accuracy of IPSBM. This is because AUC is sensible to the misclassification of the minority class, which is 'bad' but is estimated as 'good' in this case. Taking into account the setting of misclassification costs, one mistake of this type requires to be compensated by a large number of misclassified 'bad' being correctly modified by 'good', which is the main task of IPSBM. However, according to Figure 3.3, it is possible that the top L instances contain a few 'bad', which are the red dots above the yellow line. In this sense, as long as these red dots are selected into L, a drop in AUC is highly likely to occur.

Consequently, the value of L is so determinant to the algorithm performance that it is critical to confirm that the tuning strategy of L is accurate and robust. In the following paragraphs, the state-of-the-art tuning strategy given in the original paper is discussed. Additionally, a novel approach to tuning this parameter is proposed.

In general, the tuning strategy of L of the original paper is to apply the pattern learnt from the observed data to the unseen data, based on the assumption that the distribution of the ranking distance varies little from observed to unseen. In particular, with respect to Figure 3.3, the ranking difference between XGB and LR follows a linear pattern that the slope and intercept remain constant across the entire data set. The original paper argues that this phenomenon is due to the fact that the classification rules of the two types of algorithms are consistent so that the ranking difference remains the same.

In this way, a critical hypothesis is made in that paper. That is, the knowledge that IPSBM learns from the distribution of the observed data is consistent with the unseen data. More specifically, the optimal value of parameter L in the observed data is applicable to the unseen data, as long as it is scaled down according to the change of data shape. Under this

setting, the value of L is substantively subjected to the distribution of the observed data and a minor difference between observed and unseen data may have great impact on the algorithm performance. Considering the rigorous hypothesis, an empirical study (See Chapter 5.2) is conducted in this paper so as to evaluate the given tuning strategy.

Furthermore, this study proposes another strategy to tune parameter L, which is much more straightforward and supposed to be robust against the change of data distribution. Specifically, this tactic is to draw lines in different intervals and select all samples above the lines to set as 'good', on the hypothesis that the distribution of dots inside the chosen intervals remain consistent. Indeed, these lines rely upon the consistent distribution of the observed ranking difference as well. However, it is not sensitive to the change of ranking inside the dot groups in the chosen intervals.

To obtain the optimal line, a Grid Search is suggested to be undertaken: tune the value of slope and intercept in Y = ax + b by looping through the given parameter sets of a in A = { $a_1, a_2, \ldots a_k$ } and b in B = { $b_1, b_2, \ldots b_j$ }. Nonetheless, considering the complexity of Grid Search, it is in favor of selecting values of interval, intercept and slope by drawing rough lines based on the observed data instead of tuning them. It makes sense in practice because it is less time-consuming and the payoff is supposed to be high.

In conclusion, the state-of-the-art strategy of tuning L is discussed and another tuning approach is proposed. Additionally, both of them are examined in the empirical study (See Chapter 5.2).

4 Experimental Design

This chapter firstly elaborates the data sets used for the experiment. Then it discusses the employed Individual Classifiers and corresponding libraries. Lastly data preparation is introduced.

4.1 Data Sets Characteristics

The characteristics of the data sets used in evaluating the performance of the improved model are given below in Table 4.1. Four real-world credit data sets from Internet finance industry are collected. They cover mainstream loan products from micro-credit to medium-term unsecured credit. As the data sets were published in recent two years via noted machine learning challenge platforms, the quality of the data is officially entrusted.

Table 4.1: Data Sets Characteristics

Original Resource	Resource	Loan Type	Region	Year	Shape	Target Variable	P/N
Cash Bus	DataCastle	micro-credit	China	2015	15,000*1,138	default	11:89
PaiPaiDai	Kesci	online credit	China	2016	30,000*231	delinquency	7:93
Rong360	DataCastle	short-term credit	China	2017	55,000*29	delinquency	7:93
QianHai	Kesci	unsecured loan	China	2017	40,000*491	default	13:87

The first dataset comes from Cash Bus, a micro-credit company that offers instant online loaning service to individuals.[8] Among all the Internet finance products, the micro-credit is seen as one of the most risky niche markets due to the profile of the target group (Anil K. Khandelwal 2007). This view is particularly justified by the relatively high bad loan ratio (P/N) in the data set. Following Cash Bus the second data set is from one of the leading Chinese P2P Lending companies PaiPaiDai and is published on Kesci, the major Big Data competition platform in China. However, the loan type is not clearly stated in the official document.[9] The third data set comes from Rong360[10], a credit product search engine that

[8] https://www.cashbus.com
[9] http://www.ppdai.com
[10] https://www.rong360.com

recommends appropriate credit products in accordance with the customers' profiles and preferences. As an intermediary it collects information such as customers' on-site behaviors as well as historical transaction records of different financial products from credit card, bank debit to micro-credit etc. Such third-party information is contained in the data set and hence distinguishes Rong360 from other data sets. Finally, the fourth one is provided by a big player in the rapidly rising individual credit reference industry in China, QianHai Credit Reference[11]. It belongs to Ping-An Insurance Group of China, one of the most powerful financial agencies there that covers business such as insurance, banking, asset management and Internet finance etc[12]. As one of the three most competitive non-official individual credit reference agencies in China, QianHai provides especially valuable information because its parent company gives it access to the customers' other financing records such as vehicle insurance, health insurance, bank deposit, credit card etc. Consequently the direct personal financial records are seen as a unique feature of the data from QianHai.[13]

Table 4.2: Benchmarking Credit Data Sets

Original Resource	Resource	Loan Type	Region	Year	Shape	Target Variable	P/N
Lending Club	Kaggle	P2P Lending	U.S.A	2016	887,383*75	delinquency	15:85 *
Give me some credit	Kaggle	bank loan	NA	2011	120,269*10	delinquency	6:94*
German Credit (Statlog)	UC Irvine*	bank loan	GER	1994	1,000*20	"bad loan"	30:70

Resource: Archange Giscard DESTINE, Forecasting P2P Credit Risk based on Lending Club data

Resource: Adam Pah, Kaggle "Give me some credit" challenge overview

Resource: UCI Machine Learning Repository

In addition to the general introduction, the attributes of the data sets are elaborated in the following section. Especially, the content below is organized by comparing the four data sets with the conventional popular credit scoring data sets that have already been extensively studied (West 2000; Eggermont et al. 2004; Emekter et al. 2015). Specifically, as demonstrated in Table 4.2, German Credit Data from the UCI Repository of Machine

[11] https://qhzx.pingan.com/intro_aboutus.html
[12] http://www.pingan.cn/en/ir/summary.shtml
[13] http://www.zhengxinbao.com/3534.html

Learning Databases (Blake 1998) and another two from Kaggle are used as benchmarks. In overall, the novel Internet loan data sets are featured with high dimensions, more diverse information resources, more missing values and severe data bias.

Firstly, it is not difficult to note that the data sets from these Chinese Internet companies normally have considerably more dimensions than the classical data sets. The primary reason is that China has not yet had a fully developed personal credit reporting system like FICO or Schufa (Huang et al. 2016). Hence, companies there have to collect much more information to increase the predictive power. Secondly, in terms of feature diversity, data sets from Chinese Internet financial agencies cover information such as personal information, online behaviors, online and offline transactions, financial products transactions, social network, locations etc. As shown below in Table 4.3, the overview of PaiPaiDai data provides an intuitive impression on how rich the dimensions are.

Table 4.3: PaiPaiDai Features

	Column Series Name	Column Series Description
Master Table	idx	unique key for each loan
	ListingInfo	loaning time
	UserInfo_*	user attributes
	WeblogInfo_*	online behavior information
	EducationInfo_*	user's education information
	ThirdParty_Info_PeriodN_*	third party information by different periods
	SocialNetwork_*	SNS information
	Target	1 = default, 0 = regular
Login Table	idx	unique key for each loan
	ListingInfo	loaning time
	LogInfo1	login index
	LogInfo2	login type
	LogInfo3	login time
Profile Table	idx	unique key for each loan
	ListingInfo	loaning time
	UserupdateInfo1	content of profile update
	UserupdateInfo2	time of profile update

Thirdly, the employed data sets are also suffering from high rate of absent records. Figure 4.1 plots the missing records per feature of QianHai data. Concerning the size of the data (40,000), except for User_Id and Target Variable, 330 of 500 features have around missing 15,000 of 40,000 records while the rest 168 columns have even higher missing value rates. This phenomenon is to some extent similar to all novel data sets while this condition for conventional data sets is much more moderate. Lastly, as shown in the last column of Table 4.1 and Table 4.2, the data imbalance is also much more serious for novel data sets. These four unfavourable situations are calling for more advanced models, such as IPSBM proposed in this paper.

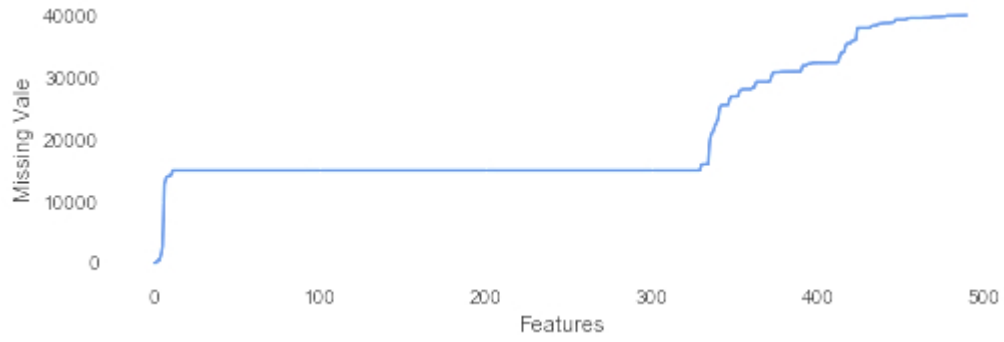

Figure 4.1: Frequency of Missing Values per Feature

4.2 Instantiation of Individual Classifier

In this chapter, the instantiation of individual classifiers is elaborated. All in all, it contains the following content: firstly, a table demonstrates the overview of employed classifiers. In the second part, the different versions of classifiers are disclosed. The last section explains how each classifier is instantiated.

The table below summarizes the employed BMs in this research. From LHS to RHS, it is structured by the type of classifier, name of classifier and the employed Python library. As can be seen in the first column, this research implements two types of BMs: LR and ensemble methods. Each type consists of several BMs: LR family includes Liblinear, SAG, Newton-cg and L-BFGS and the ensemble methods have five variations. Moreover, The last column demonstrates which specific package is used to instantiate these BMs.

Table 4.4: Individual Classifiers

Classifier Type	Classifier Name	Library	Details
Logistic Regression	Liblinear	sklearn.linear_model. LogisticRegression	(solver='liblinear')
	SAG		(solver='sag')
	Newton-cg		(solver='newton-cg')
	lbfgs		(solver='lbfgs')
Ensemble Methods	Bagging	sklearn.ensemble	BaggingClassifier
	RF		RandomForestClassifier
	GBDT		GradientBoostingClassifier
	AdaBoost		AdaBoostClassifier
	XGB	xgboost	xgboost

Apparently, there are still plenty of alternative classifiers available for credit scoring and they are so different from each other that they have great potential to coordinate with other classifiers to make good model ensemble. For instance, in accordance with the research of Lessmann et al. (2015), there is room for estimators such as Extreme Learning Machine, SVM and Artificial Neural Network to function as BMs in this research. Nevertheless, this paper does not really put these or other classifiers that are outside the table into consideration. This is mainly because of their computational complexities and time complexities. Fortunately, this could be compensated in further studies.

Concerning the implementation, by and large, the entire experiment is conducted in Python 3.6. Except XGB that uses the original package “xgboost”, all classifiers are implemented by the library sklearn, which is a widely used machine learning library for the Python programming language. As an integrated environment, it provides plenty of features to facilitate model training.

Before introducing the instantiations, it is important to notice that, each single BM in the table above actually has more than one employed son models. These son models all belong to the same BM but they differ from each other in terms of hyper parameter values or samples they are trained on. For instance, XGB has more than 6 derived models: xgb_1000 and xgb_2000 are different from rounds of iterations while xgb_2500 and xgb_2500_2

execute the same rounds of iterations but the latter is more extreme in penalizing the misclassified minority class. In order to keep the table more readable, the details of these son models are eliminated.

Concerning the implementation of LR with different regularizers and optimizers, sklearn library provides five types of solvers: “liblinear”, “sag”, “saga”, “newton-cg” and “lbfgs” (Pedregosa et al. 2011). Firstly, Liblinear, a popular linear model that once won KDD Cup 2010 (Fan et al. 2008), is a library for Large Linear Classification that supports both L1 and L2 penalty. Secondly, SAG stands for Stochastic Average Gradient, and this model minimizes finite sums with the Stochastic Average Gradient (Schmidt et al. 2017). Following SAG, Newton-cg is a modified Newton's method and uses a conjugate gradient algorithm to approximately invert the local Hessian (Shewchuk & Others 1994). Newton's method is based on fitting the function locally to a quadratic form. Lastly, L-BFGS, whose full name is Limited Memory BFGS, is an optimization algorithm in the family of quasi-Newton methods that approximates the Broyden Fletcher Goldfarb Shanno (BFGS) algorithm (Andrew & Gao 2007) using a limited amount of computer memory. It is a popular algorithm for parameter estimation in machine learning (Malouf 2002).

Regarding the instantiation of ensemble methods, Bagging has two editions in the research: both of them are instantiated by BaggingClassifier but one takes RandomForestClassifier as base_estimator while the other takes DecisionTreeClassifier. Similarly, RF, GBDT and AdaBoost are instantiated by class RandomForestClassifier, GradientBoostingClassifier and AdaBoostClassifier respectively.

4.3 Data Preparation

Data preparation of this research consists of four steps: data preprocessing, feature engineering, feature selection and data partitioning. By doing so, the original data is prepared for modeling. This paper follows a standard operation in processing all the data sets, the strategy is highlighted in the figure below. It is necessary to point out that, the idea and methodology of ranking are mainly employed in the entire data preparation procedure. Specifically, ranking in this context means sorting each observation (user) from a variety of perspectives so that the payback ability and payback willingness of each borrower is presented in a way that makes more sense.

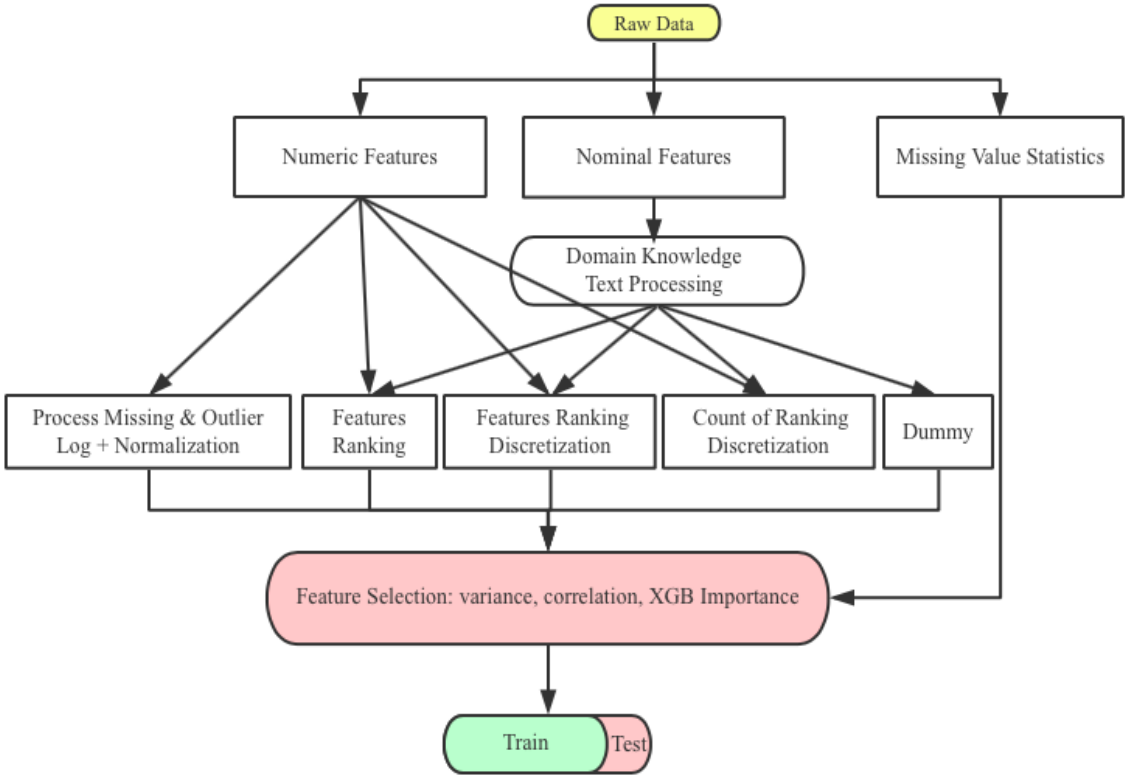

Figure 4.2: Data Preparation Strategy

4.3.1 Missing Value Statistics

Visualized by the rectangular on the RHS corner of the chart, missing value statistics is an indispensable part of Internet loan credit scoring. This is because missing value itself is critical information: in the case of information is intentionally hided by borrowers, missing value accentuates the group of people that are riskier, since some cheating borrowers or professional frauds want to cover their tracks. Meanwhile, the incomplete information may convey the development of employed credit risk controlling system which delivers useful information. In this way, recording and processing missing value contribute to modeling.

In practice, frequency of missing value per observation is firstly recorded. Presumably there are ten features in the data set and user No.1 has absent records (Na, blank, space, “Unknown” etc.) in five of the ten features, it will achieve value of 5 in the new feature “missing_count”. Furthermore, this value is sorted across all instances and discretized in an equal frequency manner. In this way three columns are generated: missing_count, missing_count_rank and missing_count_rank_dis. This is exactly the information group

“Missing Value Statistics” in the chart above. Lastly, missing value for numeric variables is replaced by median of the column, depending on if the customer paid back (0) or not (1). In the case of nominal variables, missing value is treated as an individual category.

4.3.2 Numeric Variables Processing

Numerical variable describes a borrower in a continuous manner, for instance, the borrower’s age, annual income and frequency of profile updating etc. Abstracted as the rectangular on the LHS corner, the raw numeric variables are firstly sorted ascendingly and then every individual’s position in this ranking column is recorded as its rank. It is notable that same values receive the same rank but the presence of duplicate values affects the ranks of subsequent values. Furthermore, equal frequency binning is undertaken to the ranking feature, then the ranking is discretized into 10 categories. Afterwards, the frequency of each observation's numerical feature discretization will also be recorded, and 10 columns called n1, n2 … and n10 are created to record the frequency of each observation's in each level.

On the whole, if there are n columns of numerical features in a data set, there would be additionally n ranking features, n discretization features and 10 count of discretization features generated. Thanks to the 2n+10 derived ranking features, the borrower’s profile is more vividly presented, this is because the ranking features tell the loan agency much more information than plain numeric information. For instance, the ranking of borrower’s annual income, ranking of month expense and ranking of car insurance cost reveal that whether the user is rich or poor while an absolute value could not ‘hit the point’. Additionally, ranking makes the model more robust against outliers since the variance of feature is significantly reduced; therefore, the risk of overfitting is alleviated as well. Eventually, the raw numerical features are log transformed and normalized (Yang et al. 2016).

Log Transformation :

$$LogAll(x_k) = \log(x_k - \min + 1)$$

Normalization:

$$Normalize(x_k) = \frac{x_k - mean}{std}$$

4.3.3 Nominal Variables Processing

In comparison to numeric features that can describe a borrower as rich or poor, nominal features are in general not that informative in telling the borrower's ability and willingness in paying back. However, one can learn if an individual belongs to majority or minority group from nominal features, for instance the ethnic groups, education level and employment situation etc. In order to fully extract this information, each categorical variable is sorted by the frequency of every single category of that categorical variable. In detail, for every nominal variable, the frequency of each category in the column is calculated and the variable is sorted by the frequency ascendingly. Afterwards, the equal-frequency binning and count of binning are recorded. By doing so, the popularity of each category is quantitatively presented so that the model explainability is improved.

After extracting the statistical information, the original nominal variables are converted into dummies. Furthermore, note that the nominal variables should be processed with domain knowledge first before conducting the statistical operations above. For example the PaiPaiDai data contains borrower's residence information, such as 'Hunan Road, Nanjing City, Jiangsu Province', for this reason the Chinese Word Segmentation is involved.

Last but not least, the combination of discretized ranking feature of numerical and categorical variables would also reveal important information. In detail, by taking the products of these two kinds of derived columns, the more specific information on the borrower's identity is disclosed: if this user is a heavily-leveraged minority or this user belongs to ordinary people in lower economic class. Nonetheless, owing to the complexity, this paper could not take these arithmetical operations into practice.

4.3.4 Feature Selection and Data Partitioning

Some final details are added to wrap up data preparation. Firstly, variables have variance lower than 0.001 are removed. Afterwards, features are added into a collection one by one, a new variable could join in only if it has correlation under 0.8 with any column in the set already. Subsequently, an XGB training on the collection of features is carried out, and only Top 80% of important features are left. Lastly, the data is partitioned to train (80%) and hold-out (20%).

5 Empirical Results

5.1 Comparison of Classifiers

This section presents comparison of the classifiers on the four data sets as evaluated via hold-out test. The results are shown in Table 5.1. On the whole, it is to observe that IPSBM is inferior to normal stacking model though it makes slight progress on the single best individual classifier. To note, the best result is highlighted by bold face.

Table 5.1: Model Performance Overview

Data Set	Model	Model Details	AUC
Cash Bus	BBM	XGB_2500	0.6803
	IPSBM	L=14, no error, 13 'good' saved	0.6815
	Stacking	stacking of 36 BMs, combined by LR_Liblinear	**0.6884**
PaiPaiDai	BBM	XGB_2500_6	0.7437
	IPSBM	L=32, no error, 29 'good' saved	0.7455
	Stacking	stacking of 36 BMs, combined by Adboost_20	**0.7458**
Rong360	BBM	XGB_2000_6	0.7801
	IPSBM	L=170, no error, 27 'good' saved	0.7805
	Stacking	stacking of 36 BMs, combined by GBDT_20	**0.7816**
QianHai	BBM	XGB_2000	0.7536
	IPSBM	L=58, no error, 58 'good' saved	0.7544
	Stacking	stacking of 36 BMs, combined by XGB_2000	**0.7607**

More specifically, based on BBM, IPSBM makes only moderate progress in four hold-out data sets. For example, in PaiPaiDai, AUC grows from 0.7437 to 0.7455 only. By contrast, Stacking, a simple stacking model that combines the prediction outputs of each base model, achieves much more remarkable increase in AUC than IPSBM. In the case of Rong360, AUC witnesses an increase by 0.0015 via Stacking while via IPSBM it grows by 0.0004 only. This is because, considering the average size (7,000) of the four hold-out data sets, in average only 32 ‘good’ instances could be modified when they were misclassified as ‘bad’. Such a small change is too low for a dramatic rising in AUC. In conclusion, comparing the workload with the payoff, IPSBM seems not to be an economical choice.

5.2 Comparison of Tuning Approaches

In Chapter 3.3.3, two strategies of tuning the critical parameter L for IPSBM are discussed. In this section, the performance of the two approaches are quantitatively evaluated. Note that, QianHai data is taken as example in this section because other data sets follow basically the same pattern.

In terms of experimental setting up, it is slightly different from the setting of model comparison in Chapter 5.1: similarly, the training data contains 4/5 of all observations while the hold-out takes 1/5. To note, there is additional 20% of the training used as local validation to tune the optimal value of L for Strategy One as well as the optimal intervals and ranking difference for Strategy Two. Put differently, the hold-out is to test the optimal parameters tuned in local validation. Take QianHai data as example, the original data that has 40,000 observations, is partitioned into training that has 32,000 instances and hold-out which is made of 8,000 instances. In addition, 6,400 of the 32,000 observations in training are used as local validation.

Firstly, the hypothesis of consistent distribution of ranking difference is evaluated. As can be seen from Figure 5.1, the shape of ranking difference remains almost the same in hold-out data, which is in line with the hypothesis of the original paper.

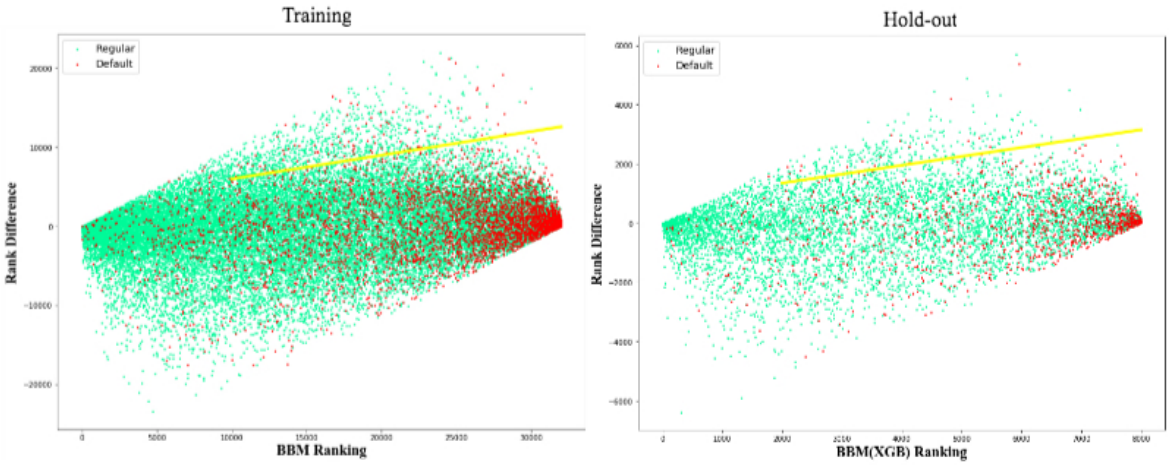

Figure 5.1: Comparison of Observed and Hold-out Data

Secondly, the robustness of Strategy One is tested. Table 5.2 compares the performance of the optimal L in the hold-out test. In the local validation, the optimal value of L is 455. In this scenario, there are 58 instances that are above the line Y = 0.3X + 3000/5. Important to note, the intercept of this line is scaled down according to data sizes ratio. The predictions of these 58 cases are modified as zero and none of them is misclassified, therefore it witnesses an increase in AUC by 7.58E-04 (See Table 5.2).

Table: 5.2: Robustness of Strategy One

	Rows	Numbers of Top L	Numbers of Instances Set to Zero	Correct	Wrong	Δ AUC
Local Validation	6400	200	22	22	0	2.66E-04
		455	58	58	0	**7.58E-04**
		460	60	59	1	4.35E-04
		1000	138	130	8	-8.28E-03
		5000	261	238	23	-1.25E-03
Hold-out	8000	100	10	10	0	1.15E-04
		170	19	19	0	**2.00E-04**
		175	20	19	1	-3.60E-05
		570	67	62	5	-6.16E-04
		5000	261	238	23	-8.28E-03

Considering the data set size ratio of local validation to hold-out, the optimal value of L in hold-out is scaled up to 570. As shown in the table above, this value causes a loss of AUC by -6.16E-04. The reason for this drop is that 67 of the chosen 570 instances are located above the line. However, 5 of them are ‘bad’, but are misclassified as ‘good’. By contrast, the actual optimal value in hold-out turns out to be 170, which is the number before the first ‘bad’ (L equals to 175) comes out.

In accordance with the experiment results, the given strategy from the original paper is over sensitive to the change of distribution, the phenomenon that the optimal value in local validation is inconsistent with the value in hold-out, is witnessed by other data sets as well. Specifically, the order that the first ‘bad’ occurs on the list of top L ‘good’ instances makes a great impact on the performance of IPSBM. As long as the the first ‘bad’ comes out

earlier than it is supposed to, AUC would suffer from a loss. In this sense, the given strategy is hardly practical.

Subsequently, the performance of the proposed strategy is examined. In contrast to Strategy One that determines L by re-fitting of specific samples, Strategy Two decides the value of L by drawing lines by different intervals (See Figure 5.2). This strategy takes the hypothesis that the relative positions of where the 'bad-free' region occurs are consistent from observed to unseen data. In this sense, the chosen intervals in observed data (LHS of Figure 5.2) would create a 'bad-free' space in the corresponding positions in unseen data as well.

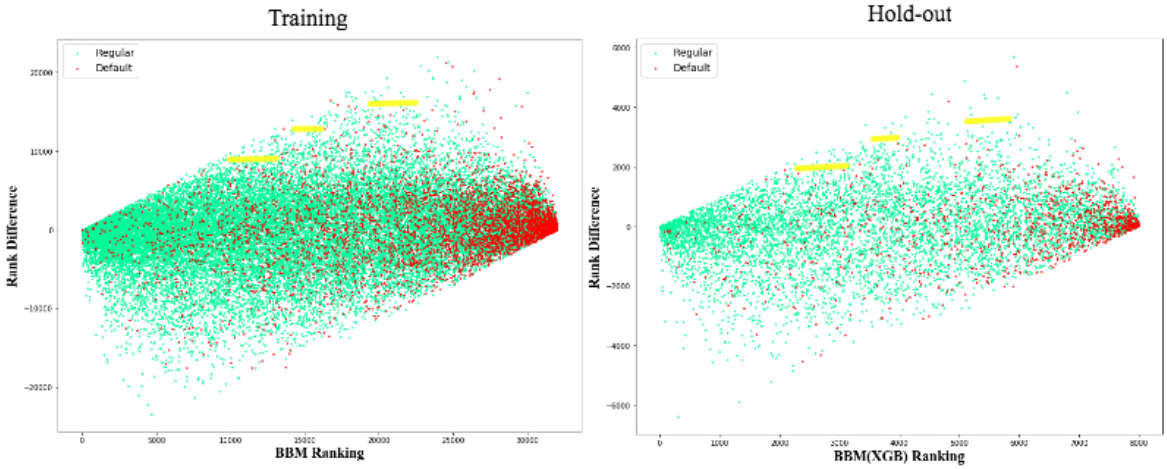

Figure 5.2: Comparison of Observed and Hold-out Data

The main purpose of this strategy is to avoid the mistakes of Strategy One. In particular, the red dots which come out in the top corner of the RHS in the chart are instances that are correctly classified by BBM as 'bad' but DBM makes an error. For this reason, these samples have high positive number in ranking difference. Therefore, these cases should be excluded from L though they own high ranking difference values. Ignoring these cases is exactly the mistakes made by Strategy One.

In order to test this strategy, firstly, a local validation is conducted so the values of two parameters are determined: the optimal intervals and the corresponding lowest ranking difference, which is the value of Y axis. Secondly, the corresponding optimal values are tested on the hold-out, as shown in Table 5.3, the tuned L makes an increase in AUC but it

is even smaller than the achievement of Strategy One. Even worse, this phenomenon could not repeatedly show up in other data sets, which means the verified 'bad-free' spaces contain several 'bad' in hold-out, this triggers a drop in AUC. In this context, Strategy Two is proved to be subjected to distribution as well. In overall it performs almost as badly as Strategy One, but it is much more straightforward to implement.

Table: 5.3: Robustness of Strategy 2

	Rows	Range	Minimum Difference	Numbers of Instances Set to Zero	Correct	Wrong	Δ AUC
Training	32000	9000-14000	9000	170	136	0	Na
		14000-16000	12000		12	0	
		20000-23000	15000		22	0	
Local Validation	6400	1800-2800	1800	27	22	0	**6.01E-04**
		2800-3200	2400		1	0	
		4000-4600	3000		4	0	
Test	8000	2250-3500	2250	36	27	0	8.20E-06
		3500-4000	3000		3	0	
		5000-5750	3750		6	0	

In conclusion, compared with Stacking, IPSBM is much more complex and is unable to provide more accurate predictions. In addition, the original strategy to tune the key parameter is fairly complicated and highly subjected to data set. Therefore, an easy-to-use approach is proposed but is proved to be unstable and inaccurate.

6 Conclusion

In the field of online banking, ensemble methods such as Random Forest and XGBoost have become the state-of-the-art approaches to identifying true creditworthiness of a potential loan borrower. These methods are proven to be powerful in predicting extremely trustworthy and considerably risky borrowers. Nonetheless, this comes at the cost of misclassifying some of the good borrowers as bad: these borrowers are not on the top or at the bottom when they are ranked based on scores of such classification rules. The industry has been exploring approaches to reducing this kind of errors in response to the negative impact from frustrated customers who are being rejected and the concern of losing market.

The 'Partially stacking blend based user credit assessment model' was proposed in this context. It claims to be able to provide promising improvement in saving the innocent good borrowers. In order to promote and prove this model, in this study, the details to implement this model are firstly provided. Additionally, a more practical approach to tuning the critical parameter is suggested. Lastly, a comparative assessment of three ensemble methods is conducted. In particular, the individual XGBoost, stacking, and the improved version of the model in interest are applied to four real world Internet credit data sets, i.e. Cash Bus, PaiPaiDai, Rong360 and QianHai Credit Reference. Experimental results reveal that the application of this model has brought rather small improvement for the best individual base learner, and the suggested tuning strategy is also ineffective. In conclusion, this model is barely able to make impressive progress in prediction accuracy.

Nonetheless, this algorithm provides an intuitive and efficient suggestion on which borrowers are likely to be inappropriately classified. With this information an improvement of prediction could be achieved by employing different algorithms or even experts' domain knowledge instead of a plain LR. Therefore, there is a considerable amount of room for further work. Firstly, further analyses are encouraged to explore different classifiers. Secondly, another interesting extension to the research would be to apply these techniques on even more biased data sets. Further, it would also be of interest to look into other methods to tune the value of L. Lastly, it is highly suggested to strive to correct the bad observations that are misclassified as good, though it is even more challenging.

References

Andrew, G. & Gao, J., 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th international conference on Machine learning - ICML '07*. Available at: http://dx.doi.org/10.1145/1273496.1273501.

Anil K. Khandelwal, 2007. Microfinance Development Strategy for India. *Economic and political weekly*, 42(13), pp.1127–1135.

Barrutia, J.M. & Echebarria, C., 2005. The Internet and consumer power: the case of Spanish retail banking. *Journal of Retailing and Consumer Services*, 12(4), pp.255–271.

Bequé, A. & Lessmann, S., 2017. Extreme learning machines for credit scoring: An empirical evaluation. *Expert systems with applications*, 86, pp.42–53.

Blake, C.L., 1998. UCI repository of machine learning databases. *http://www. ics. uci. edu/\ mlearn/MLRepository. html*. Available at: http://ci.nii.ac.jp/naid/10016466509/.

Bradley, A.P., 2013. ROC curve equivalence using the Kolmogorov–Smirnov test. *Pattern recognition letters*, 34(5), pp.470–475.

Breiman, L., 2001. Random Forests. *Machine learning*, 45(1), pp.5–32.

Bühlmann, P., 2011. Bagging, Boosting and Ensemble Methods. In *Handbook of Computational Statistics*. pp. 985–1022.

Cantor, R. & Packer, F., 1995. The Credit Rating Industry. *The Journal of Fixed Income*, 5(3), pp.10–34.

Chen, T. & Guestrin, C., 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. New York, NY, USA: ACM, pp. 785–794.

Chen, T. & He, T., 2015. Xgboost: extreme gradient boosting. *R package version 0. 4-2*. Available at: http://cran.fhcrc.org/web/packages/xgboost/vignettes/xgboost.pdf.

Dell'Ariccia, G., Friedman, E. & Marquez, R., 1999. Adverse Selection as a Barrier to Entry in the Banking Industry. *The Rand journal of economics*, 30(3), pp.515–534.

Dietterich, T.G., 2000. Ensemble Methods in Machine Learning. In *Lecture Notes in Computer Science*. pp. 1–15.

Duan, Q. et al., 2007. Multi-model ensemble hydrologic prediction using Bayesian model averaging. *Advances in water resources*, 30(5), pp.1371–1386.

Duong, N.T., Do Thi Thu, H. & Ngoc, N.B., 2017. The Application of Discriminant Model in Managing Credit Risk for Consumer Loans in Vietnamese Commercial Bank. *Asian Social Science*, 13(2), p.176.

Eggermont, J., Kok, J.N. & Kosters, W.A., 2004. Genetic Programming for Data Classification: Partitioning the Search Space. In *Proceedings of the 2004 ACM Symposium on Applied Computing*. SAC '04. New York, NY, USA: ACM, pp. 1001–1005.

Emekter, R. et al., 2015. Evaluating credit risk and loan performance in online Peer-to-Peer (P2P) lending. *Applied economics*, 47(1), pp.54–70.

Fan, R.-E. et al., 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of machine learning research: JMLR*, 9(Aug), pp.1871–1874.

Fawcett, T., 2006. An introduction to ROC analysis. *Pattern recognition letters*, 27(8), pp.861–874.

Francis, E., Blumenstock, J. & Robinson, J., 2017. Digital Credit: A Snapshot of the Current Landscape and Open Research Questions. Available at: http://escholarship.org/uc/item/88r1j7sz.pdf.

Friedman, J.H., 2001. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of statistics*, 29(5), pp.1189–1232.

Gilpin, S.A. & Dunlavy, D.M., 2008. Heterogeneous Ensemble Classification. *CSRI SUMMER PROCEEDINGS 2008*, p.90.

Hastie, T., Tibshirani, R. & Friedman, J.H., 2003. The elements of statistical learning, corrected ed. *Berlin: Springer. Haxby, JV, Gobbini, MI, Furey, ML, Ishai, A. , Schouten, JL, & Pietrini, P. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. Science*, 293(5539), p.24252430.

Huang, Z., Lei, Y. & Shen, S., 2016. China's personal credit reporting system in the internet finance era: challenges and opportunities. *China Economic Journal*, 9(3), pp.288–303.

Jahrer, M., Töscher, A. & Legenstein, R., 2010. Combining predictions for accurate recommender systems. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '10*. Available at: http://dx.doi.org/10.1145/1835804.1835893.

Jin, R. & Agrawal, G., 2003. Communication and Memory Efficient Parallel Decision Tree Construction. In *Proceedings of the 2003 SIAM International Conference on Data Mining*. Proceedings. Society for Industrial and Applied Mathematics, pp. 119–129.

Leong, C. et al., 2017. Nurturing a FinTech ecosystem: The case of a youth microloan startup in China. *International journal of information management*, 37(2), pp.92–97.

Lessmann, S. et al., 2015. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European journal of operational research*, 247(1), pp.124–136.

Li, P., 2012. Robust LogitBoost and Adaptive Base Class (ABC) LogitBoost. *arXiv [cs.LG]*. Available at: http://arxiv.org/abs/1203.3491.

Liu, Q., 2015. China's Internet financial ecosystem overview. In *2015 International Conference on Logistics, Informatics and Service Sciences (LISS)*. pp. 1–4.

Liu, S. et al., 2017. Visual Diagnosis of Tree Boosting Methods. *IEEE transactions on visualization and computer graphics*. Available at: http://dx.doi.org/10.1109/TVCG.2017.2744378.

Li, Y. & Wang, C., 2017. Risk identification, future value and credit capitalization: research on the theory and policy of poverty alleviation by Internet finance. *China Finance and Economic Review*, 5(1), p.1.

Li Zhou & Takeuchi, H., 2010. Informal Lenders and Rural Finance in China: A Report from the Field. *Modern China*, 36(3), pp.302–328.

Malekipirbazari, M. & Aksakalli, V., 2015. Risk assessment in social lending via random forests.

Expert systems with applications, 42(10), pp.4621–4631.

Malouf, R., 2002. A comparison of algorithms for maximum entropy parameter estimation. In *proceedings of the 6th conference on Natural language learning - Volume 20*. Association for Computational Linguistics, pp. 1–7.

Maurer, K., 2014. Where Is the Risk? Is Agricultural Banking Really More Difficult than Other Sectors? In *Finance for Food*. Springer, Berlin, Heidelberg, pp. 139–165.

Meng, Q. et al., 2016. A communication-efficient parallel algorithm for decision tree. *Advances in neural information processing systems*. Available at: http://papers.nips.cc/paper/6380-a-communication-efficient-parallel-algorithm-for-decision-tree.

Pappu, V. & Pardalos, P.M., 2014. High-Dimensional Data Classification. In *Clusters, Orders, and Trees: Methods and Applications*. Springer Optimization and Its Applications. Springer, New York, NY, pp. 119–150.

Pearson, K., 1909. DETERMINATION OF THE COEFFICIENT OF CORRELATION. *Science*, 30(757), pp.23–25.

Pedregosa, F. et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of machine learning research: JMLR*, 12(Oct), pp.2825–2830.

Ranka, S. & Singh, V., 1998. CLOUDS: A decision tree classifier for large datasets. In *Proceedings of the 4th Knowledge Discovery and Data Mining Conference*. pp. 2–8.

Ren, X. et al., 2017. A Novel Image Classification Method with CNN-XGBoost Model. In C. Kraetzer et al., eds. *Digital Forensics and Watermarking*. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 378–390.

Rooney, N. et al., 2004. Random Subspacing for Regression Ensembles. In *FLAIRS Conference*. Available at: https://ocs.aaai.org/Papers/FLAIRS/2004/Flairs04-092.pdf.

Schmidt, M., Le Roux, N. & Bach, F., 2017. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming. A Publication of the Mathematical Programming Society*, 162(1-2), pp.83–112.

Sesmero, M.P., Ledezma, A.I. & Sanchis, A., 2015. Generating ensembles of heterogeneous classifiers using Stacked Generalization. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(1), pp.21–34.

Shewchuk, J.R. & Others, 1994. An introduction to the conjugate gradient method without the agonizing pain. Available at: ftp://ftp.unicauca.edu.co/Facultades/.FIET_serepiteencuentasyocupaespacio/DEIC/docs/Materias/computacion%20inteligente/parte%20II/semana12/gradient/painless-conjugate-gradient.pdf.

Ting, K.M. & Witten, I.H., 1997. Stacked Generalization: when does it work? Available at: http://researchcommons.waikato.ac.nz/handle/10289/1066.

Wang, G. et al., 2011. A comparative assessment of ensemble learning for credit scoring. *Expert systems with applications*, 38(1), pp.223–230.

Wang, J.G., Xu, H. & Ma, J., 2015. The Business Model Analysis of Online Lending Platforms in China. In *Financing the Underfinanced*. Springer, Berlin, Heidelberg, pp. 87–107.

Weigel, A.P., Liniger, M.A. & Appenzeller, C., 2008. Can multi-model combination really enhance the prediction skill of probabilistic ensemble forecasts? *Quarterly Journal of the Royal Meteorological Society*, 134(630), pp.241–260.

Wei, S., 2015. Internet lending in China: Status quo, potential risks and regulatory options. *Computer Law & Security Review*, 31(6), pp.793–809.

West, D., 2000. Neural network credit scoring models. *Computers & operations research*, 27(11), pp.1131–1152.

Wolpert, D.H., 1992. Stacked generalization. *Neural networks: the official journal of the International Neural Network Society*, 5(2), pp.241–259.

Yang, D. et al., 2017. Internet Finance: Its Uncertain Legal Foundations and the Role of Big Data in Its Development. *Emerging Markets Finance and Trade*, p.null–null.

Yang, Y. et al., 2016. A Novel Hybrid Data Mining Framework for Credit Evaluation. In *Collaborate Computing: Networking, Applications and Worksharing*. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. International Conference on Collaborative Computing: Networking, Applications and Worksharing. Springer, Cham, pp. 16–26.

Yu, L., Wang, S. & Lai, K.K., 2008. Credit risk assessment with a multistage neural network ensemble learning approach. *Expert systems with applications*, 34(2), pp.1434–1444.

Zhang, Y.Z., 2015. *Internet Finance in China: An Event Study of Yu'e Bao*. Master Thesis. Stockholm School of Economics.

Zhao, Y., 2017. Research on the Consumer Finance System of Ant Financial Service Group. *American Journal of Industrial and Business Management*, 7(05), p.559.

Zheng zb et al., 2017. Partially stacking blend based user credit assessment model. *Patent*.

Zhou, J. et al., 2017. PSMART: Parameter Server Based Multiple Additive Regression Trees System. In *Proceedings of the 26th International Conference on World Wide Web Companion*. WWW '17 Companion. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, pp. 879–880.