Humboldt University of Berlin
Course: Numerical Introductory Seminar
Supervisor: Prof. Dr. Brenda López Cabrera

Stacking Algorithm for Ensemble Modelling

Submitted by

Frederik Schreck
Statistics M.Sc.
Humboldt University
580567

July 6, 2018

Contents

List of Figures

List of Tables

1 Motivation (with simple example)[1page]

"[W]hen our imperfect judgments are aggregated in the right way, our collective intelligence is often excellent."(Surowiecki, 2005, Foreword p.XIV)

In accordance with the title of his book, Surowiecki refers to what he calls the *wisdom of crowds*-phenomenon Surowiecki (2005). This social phenomen, that - under certain criteria - the aggregates of individual judgements can be superior to each individual judgement. While this effect can be found in the social world, it also applies to the world of statistics and machine learning. In the field of Stacking and Ensemble Modelling, research has shown different ways in which the aggregation of predictive models can deliver a more powerful model. Such Stacking and Ensemble Models currently belong to the most powerful machine learning tools and win many data science competitions (see Kaggle). Regarding the diversity of different approaches, this paper aims to discuss and apply the most established ones in a financial application of credit risk assessment. In depth, the concept of Bagging and the Random Forest model (Breiman, 1996, 2001), the concept of boosting and the Gradient Boosting model Freund and Schapire (1996); Friedman (2002) as well as the idea of Stacked Generalization (Wolpert, 1992) will be focused on. [!motivating examples: list some financial application, where ensembling brought a big benefit, e.g. saved many costs.]

The paper's structure is as follows: In the next section, the stacking and ensemble methods shall be introduced and reviewed with regards to the current state of research. Subsequently, the motivation and benefits of applying these models in the context of credit risk classification are discussed. In the following, the methodology for the practical evaluation of the models in such context is prepared. For that, firstly the credit risk data is presented, secondly the model building process is outlined and thirdly the metrics for model evaluation are introduced. In the next section, results of the empirical analysis are presented in detail. Finally, conclusions about comparative advantages and shortfalls of the models in the context of credit risk classification are drawn and needs of further research are identified.

2 Literature review[3-4pages]

Ensemble learning generally refers to the combination of multiple hypotheses in order to obtain a more powerful hypothesis. In the context of machine learning, the term *hypothesis* refers to the output of an algorithm, which aims to learn a target function $f(\mathbf{x})$ by using a set the features $\mathbf{x}$. Each algorithm that is used in the combination process of an ensemble learner is called a *base learner*.

Formally this means, that given training data $D^{train} := \{\mathbf{x}, y\} = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$, where $x_i = (x_{i1}, x_{i2}, ..., x_{iK})$ is the vector of feature values for each observational unit $i \in \{1, 2, ..., N\}$ and $y = f(\mathbf{x})$ is the target vector that can be modelled by an unknown function f of the features. Further let K denote the number of features and N denote the number of observations. A set of base learners of size M delivers hypotheses $h_1, h_2, ..., h_M \in H$ about the true function f, whereby H denotes the hypothesis space.

Different ways to combine the hypotheses of base learners exist. Generally, ensemble learning is most effective when combining diverse base learners. Hereby, diversity refers to error diversity, implying that the different base learners have different strengths in capturing structure in the data. Brown et al. (2005) showed that the combination methods of ensemble learning strategies enhance such diversity. Practical applications of ensembling techniques have shown, that stacking and ensembling models enhance diversity amongst their base learners by requiring different base learner algorithms, different hyperparameter settings, different feature subsets and different training sets (Güneş, 2017). [MORE RESEARCH]

In the following, different techniques to combine the hypotheses of base learners will be introduced. Furthermore, the current state of research is reviewed.

2.1 Bagging and the Random Forest model

The idea of Bagging was originally proposed by Breiman (1996). The term abbreviates bootstrap aggregating and refers to a manipulation of the training data: Each base learner m is fitted using a random sample D_m^{train} that is uniformly drawn from the training data D^{train}. Since sampling is done with replacement, D_m^{train} may contain duplicates of certain observational units. The hypotheses of the base learners is then aggregated by averaging in case of a regression problems or majority voting in case of a classification problem.

In so far, bagging is a meta-algorithm that can be used with every type of base learner models. However, especially unstable base learners that are sensitive to data manipulation should be combined (Breiman, 1996, p.124). Breiman therefore recommends to use Neural Networks, Decision Trees or subset selection in Linear Regression. By building the base learners on different subsets of the data, the bagging procedure enhances diversity amongst them and can lead to "substantial gains in accuracy" (Breiman, 1996, p.123).

The Random Forest model uses the bagging principle and supplements it by the random subspace approach (Ho, 1998; Breiman, 2001). This approach builds each base learner on a random sample with replacement of all available features. This implies a decorrelation of the hypotheses of the set of base learners used for ensembling. For Random Forests, Decision Trees are the preferred base learner algorithm (cp. Breiman, 2001).

A big strength of the Random Forest is the reduction in prediction variance compared to single Decision Trees, which stems from the diversification. Clearly, the computational costs of a Random Forest can be much higher than of a single Tree, since computational time increases linearly to the number of consulted Trees. Due to the independent building of individual base learners, Random Forest model building can be speeded up by parallelization on different cores of the computer. Breiman further notes that Random Forests give "useful internal estimates of error, strength, correlation and variable importance" (Breiman, 2001, p.10). Hence, even though decision rules of Random Forests are less transparent than those of single Decision Trees due to the sampling, relative variable importances can be calculated as a by-product by randomly permuting features and examining the influence on the prediction. As a single weakness, growing a large random forest can be computationally very expensive. Breiman proved that the generalization error of the ensemble converges almost surely to a limit with increasing number of Trees (Breiman, 2001, p.30). In practice, the number of Trees is however restricted by the amount of available computational resources.

2.2 Boosting and the Gradient Boosting model

Beneath Bagging, another powerful ensembling technique is Boosting. Boosting builds on the idea that the aggregation of simple base learners may lead to a strong learner. In the so-called Adaboost algorithm, Freund and Shapire (1996) start with an ensemble of one weak learner and iteratively add one more weak learner that aims to correct for the (pseudo) residuals of the current ensemble. Thereby, the calculation of these residuals is based on an iterative reweighting of the data. The weights of each datapoint $\mathbf{x_i}$ for model m depend on the prediction accuracy of the current ensemble hypothesis for that datapoint $h^{ensemble}_{\{1,2,\dots,m-1\}}(\mathbf{x_i})$.

With Boosting, it is possible to decrease the training error to zero (Freund and Schapire, 1996, p.11ff.). Furthermore, as long as base learners are better than random guessing, the Boosting technique is also able to reduce the generalization error independent of the base learning algorithm.

A development of the Boosting idea is the (Stochastic) Gradient Boosting model, that is currently the most commonly used Boosting model (Friedman, 2001, 2002). In contrast to the data weighting scheme in the Adaboost algorithm, Gradient Boosting minimizes the gradient of a loss function of the error by applying gradient descent. Typically, small Decision Trees are used as base learners of the Gradient Boosting model due to their propensity towards high prediction bias. Additionally, Gradient Boosting integrates the bagging idea. Friedman was able to show, that this integration could substantially improve accuracy and execution speed of the model (Friedman, 2002).

The possibility to be executed fastly is a reason for the heavy use of Gradient Boosting models for machine learning problems. Furthermore, they allow to gain insights into the dependence of output and features by enabling partial dependence plots (Friedman, 2001, p.1219ff.). However, due to its nature, Gradient Boosting models are highly prone to overfit and therefore must be accompanied with regularization methods (Friedman, 2002, p.1203).

2.3 Stacked Generalization model

Stacked Generalization models has been introduced by Wolpert 1992 and defines a way to combine multiple predictive algorithms by using a second-level algorithm. In contrast to Bagging or Boosting, Stacked Generalization is typically applied to a space of different base learning models. The idea is, that different kinds of models that are applied to the learning problem are able to capture only part of the problem. Combining models with different strengths in the right way would then lead to improved predictive accuracy. Stacked Generalization is therefore also referred to as a second-stage model.

The Stacking algorithm involves partitioning of training dataset $D^{train} = (\mathbf{x}, y)$ into $J = \{1, 2, ..., J\}$ disjoint parts $D_1^{train}, D_2^{train}, ..., D_J^{train}$. For each of these subsets D_j^{train}, called "level 0 learning set", a base learner $m \in \{1, 2, ...M\}$, also referred to as "level 0 generalizer", is built on behalf of the training dataset without this subset D_{-j}^{train} (Wolpert, 1992, cp.). In each of the iterations, the model built on D_{-j}^{train} is used to predict on subset D_j^{train}. The predictions of the J subsets are then combined again in order to obtain a prediction of the whole training dataset D^{train}. Further, each level 0 generalizer is used to predict on the test dataset D^{test}. Due to the level 0 generalizers being built J times on different disjoint subsets of the training data, Stacked Generalization can be seen as a sophisticated form of cross-validation. The next step is building a meta learner, referred to as "level 1 generalizer", that produces a prediction by using the training dataset predictions of the M level 0 generalizers as inputs. Wolpert could show that this stacking procedure is able to reduce the bias of the base learners and thus minimizes the generalization error rate. He even recommends to use a version of Stacked Generalization in any real-world problem (Wolpert, 1992, p.2).

Different meta learning algorithms can be used for the combination of base learners. An optimal meta algorithm finds the best way to use the strengths of the base learners. Overfitting problem is especially present in Stacked Generalization models. This is due to the base learners all predicting the same target (Güneş, 2017). As a consequence, cross-validation and regularization can be used. Further more, the chosen meta learning algorithm should not be sensible to collinearity. It is therefore especially recommended to use Random Forests, Regularized Regression, Gradient Boosting or hill climbing methods

(Güneş et al., 2017).

3 Ensemble modelling in Credit Risk Classification: Emphasize application for finance/statistics[1-2pages]

Techniques of Stacking and Ensemble Learning are applied to predictive problems of a broad range of topics. This paper will especially focus on the application of Ensemble Learning in credit risk classification problems. Credit risk assessment and especially the its modelling is an important part in the field of financial risk management since for most small- and medium-sized banks, interests on loans are still the primary financial source (Jacobson et al., 2006, p.2). The banking supervision accord Basel II, that was published in 2004 and applies to member states of the European Union since 2007, restricted the buffer capital on banks and therefore makes it especially important for them to estimate the riskiness of loan applicants (Basel Committee on Banking Supervision, 2004). For that, banks need to be able to distinguish between applicants into risky and non-risky applicants. Two opposing factors determine the banks' business rules regarding loans: On the one hand, more loans are better. On the other hand, a bank can not afford to make to many bad loans, since this would eventually lead to a collapse of the bank. A good strategy on loans will therefore be a compromise.

Applying Ensemble Learning techniques to credit risk modelling has already proven to be highly valuable. Zhu et al. 2017 investigate credit risk assessment for small- and medium-sized Chinese enterprises. For that, they carry out an experiment in which they compare the predictive performance of individual machine learning methods and ensembling methods of different complexity. They find especially the more complex ensembling methods to be of outstanding discriminative accuracy (Zhu et al., 2017, p.46f.). Yu et al. 2008 succesfully apply an ensemble learner comprising six levels of stacked Neural Networks in order to evaluate credit risk at the measurement level. Hereby, they further incorporate the Bagging approach. They conclude, that such technique "provides a promising solution to credit risk analysis" (Yu et al., 2008, p.1443). [MAYBE SOME MORE STUDIES]

4 Methodology[2pages]

In order to evaluate and compare the introduced stacking and ensemble models, an empirical evaluation study is conducted. The quantlets for replication of the study can be accessed in the corresponding github repository [LINK?]. In this section, the dataset used for the evaluation study is presented, the model building process is explained in detail

and the metrics for evaluation are introduced.

4.1 Data description

In order to evaluate the introduced Stacking and Ensemble models, the German Credit Dataset from the UCI machine learning repository is used (Dheeru and Karra Taniskidou, 2017). The dataset classifies people as either being good or bad customers for a bank with respect to credit risk. It comprises a total number of 1000 observations and 20 features. Table [REF!] presents the summary statistic on the dataset. [some notes on the summary statistic] The dataset is partitioned in a training dataset, comprising 750 observations, and a test dataset, comprising 250 observations.

4.2 Model building process

The model building process consists of feature selection, training , tuning and prediction. For the purpose of this study, an extensive set of models goes through this process: A Random Forest model and a Gradient Boosting model represent the Ensemble Learners. For the Stacked Generalization models, the Random Forest model and the Gradient Boosting model are rebuilt as level 0 generalizers. Additionally, a Decision Tree, a Neural Network as well as a Logistic Regression model are built in order to provide a diverse set of level 0 generalizers for the Stacked Generalization models. Four different such Stacking models are built by using different subsets of base learners' predictions, namely all versus a set of least correlated predictions, and by consulting different combiner algorithms, namely averaging and consulting a Random Forest model. All models deliver probabilistic predictions.

Before training the model, feature selection is a critical step. The aim of feature selection is dimension reduction. Building the models on this subset may reduce their training time, reduce the variance and make the model more easily interpretable (Guyon and Elisseeff, 2003). Since, the optimal subset of features depends on each model, a wrapper approach for feature selection is applied to each model specifically in form of wrapper approaches. Each model-specific wrapper approach starts with building the model on an intercept model and sequentially adds the next best feature (sequential foreward selection). The wrapper approach stops when no further feature can be added that increases the AUC measure (see section 4.3) by at least 0.00001 units. All wrapper approaches are run on 3-fold cross-validation in order to avoid overfitting problems (A larger number of folds leads to instabilities due to the small sample size). The subset of features identified by the model-specific wrapper approaches is subsequently used for training of the corresponding models. Since the Random Forest and the Gradient Boosting models

are built as Ensemble Models and as level 0 generalizer for the Stacked Generalization models, independent wrapper approaches are applied for both versions.

The training process for each model generally consists of establishing a broad-grid tuning of all relevant hyperparameters on the training dataset in order to find the (locally) optimal parameter choices. In order to avoid overfitting on the training dataset, each combination of hyperparameters is tested by a 3-fold cross-validation process. For the Random Forest and the Gradient Boosting model, their tuned versions can directly be used for prediction on the test dataset.

For the Stacked Generalization models, the training dataset is partitioned into five disjoint subsets. Each of the five level 0 generalizers is then build in five iterations as described in section 2.3. Again, each iteration involves a parameter tuning on 3-fold cross-validation. All level 0 generalizers are then used to predict the observations in the test dataset. Stacking model 1 is then built by averaging over the probabilistic predictions of all five level 0 generalizers. Stacking model 2 is constructed by averaging over the probabilistic predictions of the three least correlated predictions of level 0 generalizers. For that, the correlations of the level 0 generalizers on the training data are investigated. Stacking model 3 is built by using again all level 0 generalizers and a Random Forest as a combining model. The parameters of the combining model are again tuned under 3-fold cross-validation. Finally, Stacking model 4 is constructed by combining the three least correlated predictions of level 0 generalizers by using a Random Forest combiner.

4.3 Evaluation metrics

In credit risk modelling, the misclassification costs are often type-specific. This means that a false negative prediction may have different costs for the bank than a false positive prediction. When misclassification costs are known, a cost-sensitive model building strategy should be consulted. Since in the context of this paper misclassification costs are however unknown, equal misclassification costs for false negative and false positive predictions are assumed. Furthermore, the broad field of cost-sensitive learning may serve as a topic for other studies.

AUC: In the model building process, tuning of model parameters for each model m is evaluated on the Area Under Curve metric (AUC). The AUC is a ranking indicator that measures the area under the receiver operating characteristic curve (ROC curve) (Hanley and McNeil, 1982). For a probabilistic prediction, like in the context of this study, a visualization of the ROC curve can be obtained by plotting the sensitivity against $1-$ specificity for all cut-off thresholds between zero and one. The AUC value can therefore take values between zero and one as well. A random model would obtain an AUC value

of 0.5, which can thus function as a benchmark value in model evaluation on AUC. In a statistical sense, the AUC estimates the probability that a randomly chosen correct prediction is correctly ranked higher than a randomly chosen false prediction. For model m can be calculated as

$$AUC^m = \frac{1}{P \times N} \sum_{j=1}^{P} \sum_{k=1}^{N} (\hat{y}_j^m - \hat{y}_k^m) \tag{1}$$

, whereby P and N denote the positive (in our case "good") and negative ("bad") instances amongst the outcome values in the credit data. Further, $\hat{y}_j^m$ and $\hat{y}_k^m$ are the predictions for the positive instance y_j and the prediction for the negative instance y_k, respectively.

Accuracy: A further important metric in evaluation of classification models is the Accuracy metric, which can be interpreted as the percentage of correctly classified points.

$$Accuracy^m = \frac{TP^m + TN^m}{FP^m + FN^m + TP^m + TN^m} \tag{2}$$

, whereby TP^m is the number of true positive predictions for model m and TN^m, FP^m and FN^m are the corresponding number of true negatives, the number of false positives and the number of false negatives for model m, respectively.

Brier Score: The models will also be assessed on the Brier Score, which is identical to the Mean Squared Error metric in statistics.

$$Brier^m = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i^m)^2 \tag{3}$$

, whereby $\hat{y}_i^m$ denotes the predicted probability of model m for observation y_i.

Logarithmic Loss: The Logarithmic Loss is a metric for evaluating class predictions that penalizes for a high confidence about incorrect classifications. For the case of a binary outcome, the Logarithmic Loss is given by

$$LogLoss^m = -\frac{1}{N} \sum_{i=1}^{N} (y_i^m \log(p_i^m) + (1 - y_i^m) \log(1 - p_i^m)) \tag{4}$$

, whereby p_i^m is model m's prediction for observation y_i.

5 Results of empirical study[2pages]

[maybe include: BiasVarianceTradeoff-graph, AUCplot with all models in] In the following, the results of the empirical application of Stacking and Ensemble models on credit risk classification will be presented. Table 1 shows the evaluations of all models on the test dataset with respect to the metrics Accuracy, AUC, Logarithmic Loss and Brier Score.

It can be seen that the two Ensemble models, namely the Random Forest and the Gradient Boosting model, are the best performing models with respect to the AUC metric. Since they were tuned on behalf of the AUC measure this is ...

When comparing the five level 0 generalizers, the XXX model performs best on the AUC measure. Notably, the level 0 Random Forest and Gradient Boosting models perform a bit worse than their counterparts mentioned above. This seems to be a result of the partitioning of the dataset during their building process...

When evaluating the four Stacking models, different tendencies can be found. Stacking Model 1 was generated by unweighted average of the predictions of all five level 0 generalizers. When compared to the level 0 generalizers themselves, the Stacking Model 1 could not gain an increase in AUC performance...

	AUC	Accuracy	Logarithmic Loss	Brier Score
Random Forest	0.78	0.73	0.52	0.18
Gradient Boosting	0.78	0.72	0.52	0.18
Decision Tree (level 0)	0.69	0.65	0.60	0.21
Logit Regression (level 0)	0.71	0.74	0.57	0.19
Neural Network (level 0)	0.76	0.74	0.53	0.18
Random Forest (level 0)	0.73	0.69	0.56	0.19
Gradient Boosting (level 0)	0.80	0.74	0.52	0.17
Stacking Model 1	0.77	0.65	0.24	0.18
Stacking Model 2	0.78	0.63	0.24	0.18
Stacking Model 3	0.80	0.63	0.30	0.16
Stacking Model 4	0.81	0.65	0.26	0.17

Table 1: Model performance on the test dataset.

6 Conclusion[1page]

- advantages of bagging models and potential problems (e.g. higher computational costs)
- conclusions about chosen problem of analysis
- Further research

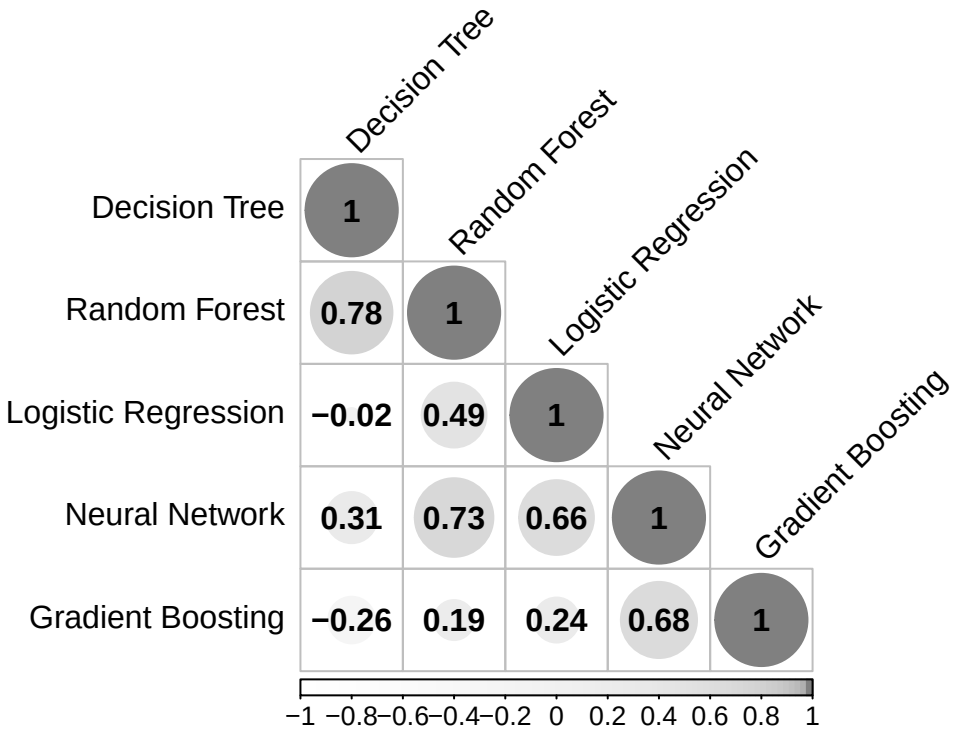

Figure 1: Correlation plot of training dataset predictions.

A Appendix-part1

B Appendix-part2

References

Basel Committee on Banking Supervision (2004). International Convergence of Capital Measurement and Capital Standards. URL: https://www.bis.org/publ/bcbs107.pdf. (Accessed: 03.07.2018).

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

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

Brown, G., Wyatt, J., Harris, R., and Yao, X. (2005). Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5–20.

Dheeru, D. and Karra Taniskidou, E. (2017). UCI machine learning repository. URL: http://archive.ics.uci.edu/ml. (Accessed: 03.07.2018).

Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Icml*, volume 96, pages 148–156. Bari, Italy.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378.

Güneş, F. (2017). Why do stacked ensemble methods win data science competitions? URL: https://blogs.sas.com/content/subconsciousmusings/2017/05/18/stacked-ensemble-models-win-data-science-competitions. (Accessed: 03.07.2018).

Güneş, F., Wolfinger, R., and Tan, P.-Y. (2017). Stacked Ensemble Models for Improved Prediction Accuracy (*SAS paper*).

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

Hanley, J. A. and McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844.

Jacobson, T., Lindé, J., and Roszbach, K. (2006). Internal ratings systems, implied credit risk and the consistency of banks' risk classification policies. *Journal of Banking & Finance*, 30(7):1899–1926.

Kaggle. URL: https://www.kaggle.com/. (Accessed: 05.07.2018).

Surowiecki, J. (2005). *The wisdom of crowds*. Anchor.

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2):241–259.

Yu, L., Wang, S., and Lai, K. K. (2008). Credit risk assessment with a multi-stage neural network ensemble learning approach. *Expert systems with applications*, 34(2):1434–1444.

Zhu, Y., Xie, C., Wang, G., and Yan, X. (2017). Comparison of individual, ensemble and integrated ensemble machine learning methods to predict china's sme credit risk in supply chain finance. *Neural Computing and Applications*, 28(1):41–50.