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

Stacking Algorithm and Ensemble Modelling

Submitted by

Frederik Schreck
Statistics M.Sc.
Humboldt University
580567

July 11, 2018

Contents

List of Figures

List of Tables

1 Abstract

2 Motivation

"[W]hen our imperfect judgements 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). By that, he refers to the social phenomenon, that the aggregates of individual judgements - under certain criteria - can be superior to each individual judgement alone. 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 e.g. the data science competition website Kaggle). In the context of credit risk assessment, where banks need to estimate credit worthiness of potential customers, accurate predictions are particularly valuable. In comparison to alternative predictive models, Stacking and Ensemble methods have shown to be highly effective (Yu et al., 2008; Zhu et al., 2017). In depth, this paper will introduce 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.

The paper's structure is as follows: In the coming section, the Stacking and Ensemble models shall be introduced and reviewed with regards to their state of research. Subsequently, the value of applying these models in the context of credit risk classification is discussed briefly. In the following, the empirical evaluation study of applying the introduced Stacking and Ensemble 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 evaluation study 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.

3 Stacking and Ensembling Modelling

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 $D^{train} := \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$, 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. Then, 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) show that the combination methods of ensemble learning strategies enhance such diversity. Furthermore, practical applications of ensembling techniques show, that Stacking and Ensemble models enhance diversity amongst their base learners by requiring different algorithms, hyperparameter settings, feature subsets and training sets for their base learners (Güneş, 2017).

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

3.1 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^{train}_m that is uniformly drawn from the training data D^{train}. Notably, D^{train}_m may contain duplicates of certain observational units, since sampling is done with replacement. The hypotheses of the base learners is then aggregated by averaging in case of a regression problem 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 algorithm. 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 accelerated 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 their influence on the prediction. As a single weakness, growing a large random forest can be computationally expensive. Breiman proves 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.

3.2 Boosting and the Gradient Boosting model

Beneath Bagging, another powerful ensembling technique is Boosting. Boosting builds on the idea that the aggregation of weak base learners may lead to a strong learner. In their 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 x_i for model m depend on the prediction accuracy of the current ensemble hypothesis for that datapoint $h^{ensemble}_{\{1,2,\ldots,m-1\}}(x_i)$.

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, which 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 shows, that this integration could substantially improve accuracy and execution speed of the model (Friedman, 2002).

The possibility to be executed fast 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 target and features by enabling partial dependence plots (Friedman, 2001, p.1219ff.). However, due to its nature, Gradient Boosting models are highly prone to overfit the training data and therefore must be accompanied with regularization methods (Friedman, 2002, p.1203).

3.3 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 learner algorithms. 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 diverging strengths in the right way then leads to improved predictive accuracy. Stacked Generalization is therefore also referred to as a second-stage model.

The Stacking algorithm involves partitioning of training dataset $D^{train} = (\mathbf{x}^{train}, y^{train})$ 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 iteration, the model built on D_{-j}^{train} is used to predict the target feature in subset D_j^{train}. The predictions of the J subsets are then combined again in order to obtain a prediction of the target over the whole training dataset D^{train}. Besides that, 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 shows 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 Regularized Regression, Gradient Boosting or hill climbing methods (Güneş et al., 2017).

4 Ensemble Modelling in Credit Risk Classification

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 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 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 decisions, since this would eventually lead to a collapse of the bank. A good strategy on loans will therefore be a compromise and minimizing the fraction of bad decisions increases revenue.

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 successfully 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).

5 Methodology

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 found in the corresponding github repository. 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.

5.1 Data description

In order to evaluate the introduced Stacking and Ensemble models, the German Credit Dataset from the UCI machine learning repository is consulted (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. Tables 2 and 3 in the Appendix present the summary statistics for the numerical and the categorical features in the original dataset, respectively. To ensure better model performance, the numerical data is standardized before model building. The dataset is partitioned into a training dataset, comprising 750 observations, and into a test dataset, comprising 250 observations.

5.2 Model building process

The model building process consists of feature selection, model training and tuning as well as prediction. For the purpose of this study, an extensive set of models went through this process: A Random Forest model and a Gradient Boosting model represent the Ensemble Learners. Furthermore, both models are built a second time as level 0 generalizers for the Stacked Generalization models. 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 Stacking. Four different such Stacking models are built by using different subsets of base learners' predictions. Stacking model 1 and 2 use all level 0 predictions versus the three best level 0 predictions, respectively. A simple averaging is used as a combiner method here. Stacking model 3 and 4 both use all level 0 predictions and combine them via a Gradient Boosting model and a Logistic Regression model, respectively. All models deliver probabilistic predictions. In order to speed up the model building, all models are constructed by use of parallelization across the cores of the machine.

Before training the model, feature selection is a critical step. The aim of feature selection is dimension reduction. Building the models on an optimal subset of features 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. Each model-specific wrapper approach starts with building an intercept model and sequentially adding the next best feature by using a sequential foreward selection approach. The wrapper approach stops when adding another feature cannot increase the

AUC measure (see section 5.3) by at least 0.00001 units. All wrapper approaches are run on 3-fold cross-validation in order to avoid overfitting problems. Notably, a larger number of folds led to instabilities due to the small sample size. Subsequently, the subset of features identified by the model-specific wrapper approaches is 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 3.3. 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 best predictions of level 0 generalizers in terms of AUC measure. For that, the correlations of the level 0 generalizers on the training data must be investigated in order to avoid multicollinearity problems. Stacking model 3 is built by using again all level 0 predictions and a Gradient Boosting model as combining algorithm. Finally, Stacking model 4 is constructed by combining all level 0 predictions by using a Logistic Regression combiner. The parameters of the combining algorithms are again tuned under 3-fold cross-validation.

5.3 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. The following metrics are used to evaluate the models.

AUC: In the model building process, tuning of model parameters for each model m is evaluated on the Area Under Curve metric (AUC). Tuning on the AUC is especially recommended when facing probabilistic predictions, since it generalizes over all possible cut-off thresholds. 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 the AUC can be calculated as

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

, 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. In contrast to the AUC, probabilistic predictions must be transformed into binary predictions for the Accuracy metric, which implies selecting a cut-off threshold. For the purpose of this study, a natural cut-off threshold of 0.5 is chosen.

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

, 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.

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

$$\mathrm{LogLoss}^m = -\frac{1}{N} \sum_{i=1}^{N} (y_i^m \log(p_i^m) + (1 - y_i^m) \log(1 - p_i^m)) \tag{3}$$

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

Brier Score: The models will further be assessed on the Brier Score, which is identical to the Mean Squared Error metric in statistics. For model m, the Brier score is defined as

$$\text{Brier}^m = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i^m)^2 \tag{4}$$

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

6 Results

In the following, the results of the empirical application of Stacking and Ensemble models on a credit risk classification problem will be presented. Table 1 shows the evaluations of all models on the test dataset with respect to the metrics AUC, Accuracy, Logarithmic Loss and Brier Score.

	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 (average, all)	0.77	0.64	0.24	0.18
Stacking Model 2 (average, best)	0.77	0.63	0.25	0.18
Stacking Model 3 (GB, all)	0.78	0.61	0.29	0.17
Stacking Model 4 (LR, all)	0.81	0.64	0.30	0.16

Table 1: Model performances on the test dataset. Values rounded on two digits after comma.

Since all models were tuned on the AUC, the comparison on behalf of this metric is most informative. The two Ensemble models, namely the Random Forest and the Gradient Boosting model, score highly on the AUC metric with a value of 0.78. In terms of AUC, they even outperform most level 0 generalizers, especially the Decision Tree and the Logistic Regression. It can be concluded that they rank a randomly chosen correct prediction comparatively comparatively higher than a randomly chosen false prediction in comparison with the level 0 generalizers. Notably, the Gradient Boosting level 0 generalizer and the Random Forest level 0 generalizer perform a bit different when compared to their counterparts. This may origin from the decreased training dataset size due to the partitioning that was applied for the level 0 generalizers.

The four Stacking models were built on top of the level 0 generalizers. Figure 1 shows the correlations of the predictions of the level 0 generalizers on the training data. It can be seen that the predictions of the level 0 generalizers are (mostly) correlated positively, which seems intuitive since they all predict the target in the similar context. However, correlations are not perfect (all $< |0.9|$). This reveals diversity of predictions, which already indicates that combining them may increase predictive performance.

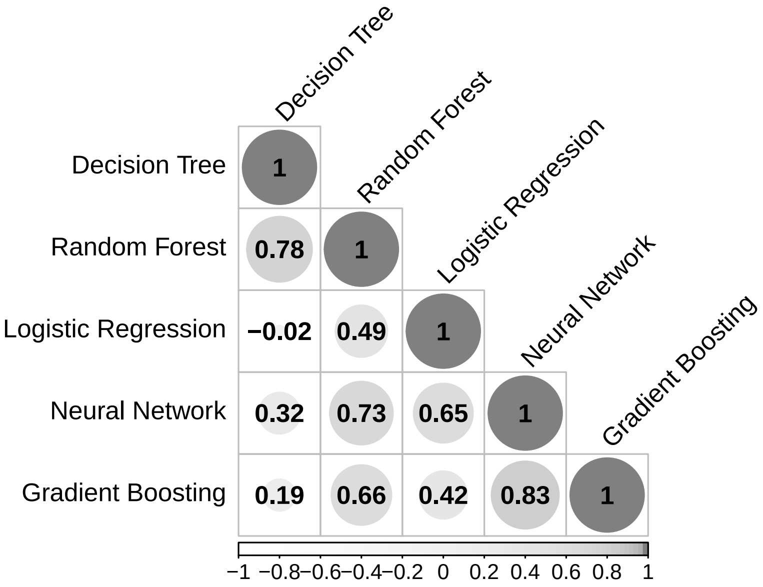

Figure 1: Correlation plot of training dataset predictions of level 0 generalizers.

Indeed, Table 1 reveals that the four Stacking models clearly outperform their level 0 generalizers in terms of AUC. Stacking models 1, 2 and 3 that are based on averaging predictions as well as a Gradient Boosting combiner have similar high AUC performance than the Random Forest and the Gradient Boosting model. Stacking model 4 that is based on combining level 0 predictions by Logistic Regression, respectively, even show a better AUC performance than the Random Forest and the Gradient Boosting model. For visual comparison, Figure 2 shows the ROC curves related to the AUC values for all models.

With regards to the Accuracy, the Logistic Regression and the Neural Network perform about equally well as the two Ensemble models. The Decision Tree is however outperformed by the Random Forest and the Gradient Boosting model. Interestingly, all four Stacking models show relatively bad Accuracy values. Regarding the relation of Accuracy and AUC, it can be concluded that the Stacking and Ensemble models do not necessarily perform better on the cut-off threshold of 0.5, which is implied by the Accuracy metric. However, generalizing over all possible cut-off threshold they clearly outperform most level 0 generalizers as can be seen by the corresponding AUC values. Furthermore,

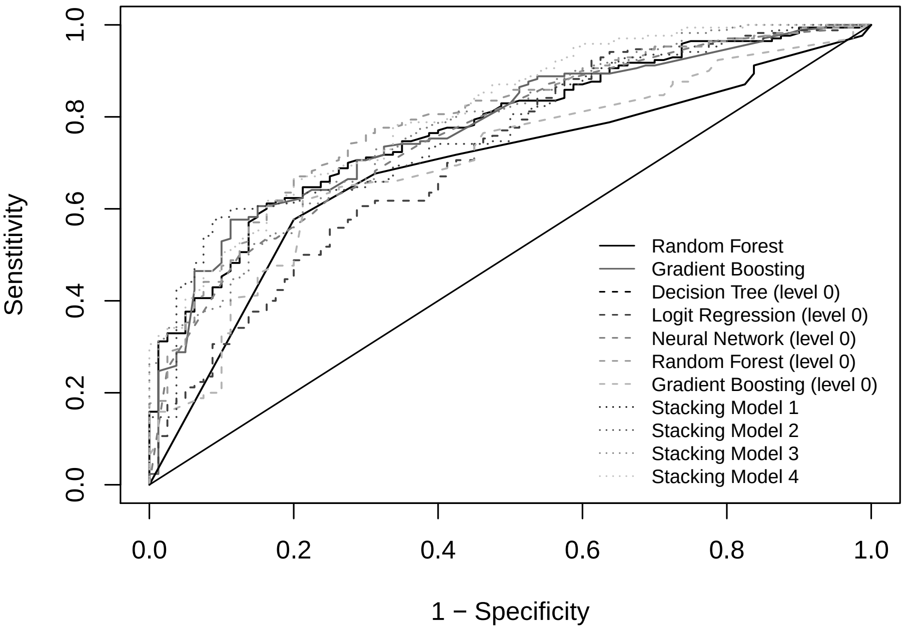

Figure 2: Receiver Operating Characteristic (ROC) curves for all models.

this indicates that the standard cut-off threshold for probabilistic predictions of 0.5 may not be optimal for those models.

The Logarithmic Loss metric gives even more information on the particular strengths and weaknesses of the models. Contrasting the level 0 generalizers with the Random Forest and the Gradient Boosting model, the latter show a slightly smaller Logarithmic Loss. The four Stacking models however, reveal a much better performance on behalf of the Logarithmic Loss metric. It can be concluded, that the Ensemble models and even more the Stacking models do not place as high confidences on their incorrect predictions.

Regarding the Brier Score, differences are not that present. In tendency, the Stacking and Ensemble Models perform slightly better on that metric than the level 0 generalizers. One time more, this shows that in binary classification a good model is not only defined by its errors but rather by which observations it is able to classify correctly.

With regards to all calculated metrics, Stacking models are be able to capture the structure in the data most effective. Already a simple averaging approach, as applied for Stacking models 1 and 2, leads to increased performance when compared to the level 0 models. Interestingly, reducing the set of level 0 generalizers, like for Stacking model 2, does not add more value. Nevertheless, it shall be noted that such restriction of input predictions could still decrease computational costs. Applying a more sophisticated level 1 combiner, like Gradient Boosting or Logistic Regression, furthermore improves prediction.

In particular, the Logistic Regression combiner is able to increase AUC performance.

7 Conclusion

This paper aimed to discuss and evaluate Stacking and Ensemble models in a financial application of credit risk assessment. Focus was set on introducing the concepts of Bagging, Boosting and Stacked Generalization as well as the corresponding Random Forest and Gradient Boosting models. In depth, it was explained how Bagging and Boosting aim at increasing performance by decreasing generalization variance of base learners and by decreasing generalization bias of base learners, respectively. Moreover, the Stacked Generalization model was outlined, that seeks for the optimal way to combine the specific strengths of level 0 generalizers in order to increase predictive performance even more.

A broad set of different Stacking and Ensemble models as well as standard machine learning models was applied and evaluated to a classification problem of credit risk assessment. Thereby, the Random Forest model and Gradient Boosting model outperformed standard machine learning models on behalf of the calculated metrics. Furthermore, all four Stacked Generalization approaches were able to establish a better performance on AUC, Logarithmic Loss and Brier Score than their level 0 generalizers. With regards to Accuracy metric, they performed worse, suggesting that the metric-implied cut-off threshold of 0.5 being suboptimal. Even a simple combiner algorithm like averaging could increase performance of the Stacking model, while the Logistic Regression combiner performed best. Restricting the subset of input predictions seems to be ineffective with regards to performance issues. To conclude, Stacking and Ensemble models could show their comparative strengths in this study. The results strongly confirm their value for prediction issues in credit risk assessment.

While this study focused on predicting binary outcomes, further research could evaluate the performance of Stacking and Ensemble models for regression problems in a similar context. A shortfall of Stacking and Ensemble models is their need of much computational resources caused by their complexity. In the context of *scalability*, current research tests ideas that restrict computational costs, e.g. by adaptive parallelization (Li et al., 2014) or by improving sub-model algorithms like stochastic gradient descent (Bottou, 2012). A further problem of such models is the absence of proven statistical properties like unbiasedness, consistency or asymptotic theory for construction of confidence intervals. In order to use machine learning models for research purposes, such properties are however necessary. Only recently, the field of *machine learning in economics* emerged, where the development of such properties is aimed at (see Athey (2017) ,Wager and Athey (in press)).

A Appendix - Summary Tables

	Mean	Std. Dev.	Median	Minimum	Maximum
Duration	20.90	12.06	18.00	4.00	72.00
Amount	3271.26	2822.74	2319.50	250.00	18424.00
Installment Rate	2.97	1.12	3.00	1.00	4.00
Residence Duration	2.85	1.10	3.00	1.00	4.00
Age	35.55	11.38	33.00	19.00	75.00
Number of Credits	1.41	0.58	1.00	1.00	4.00
Number of Liable People	1.16	0.36	1.00	1.00	2.00

Table 2: Summary statistics for numerical features in the German Credit Dataset.

Feature	Category	Count	Fraction
Customer Classification	good	700	70%
(Outcome Feature)	bad	300	30%
Account Status	x < 0 DM	274	27.4%
	0 DM < x < 200 DM	269	26.9%
	x >= 200 DM	63	6.3%
	no account	394	39.4%
Credit History	no credits taken/all paid back duly	40	4%
	all credits at this bank paid back duly	49	4.9%
	existing credits paid back duly till now	530	53%
	delay in paying off in the past	88	8.8%
	critical account	293	29.3%
Purpose	car (new)	234	23.4%
	car(used	103	10.3%
	furniture/equipment	12	1.2%
	radio/television	181	18.1%
	domestic appliances	280	28%
	repairs	12	1.2%
	education	22	2.2%
	vacation	50	5%
	retraining	9	0.9%
	business	97	9.7%
Savings	x < 100 DM	603	60.3%
	100 <= x < 500 DM	103	10.3%
	500 <= x < 1000 DM	63	6.3%
	x >= 1000 DM	48	4.8%
	unknown/no savings	183	18.3%

(continued on next page)

(continued)

Feature	Category	Count	Fraction
Employment Duration	unemployed	62	6.2%
	x < 1 year	172	17.2%
	1 <= x < 4 years	339	33.9%
	4 <= x < 7 years	174	17.4%
	x >= 7 years	253	25.3%
Status and Sex	male: divorced/separated	50	5%
	female: divorced/separated/married	310	31%
	male: single	548	54.8%
	male: married/widowed	92	9.2%
Other Debtors	none	907	90.7%
	co-applicant	41	4.1%
	guarantor	52	5.2%
Property	real estate	282	28.2%
	savings agreement/life insurance	232	23.2%
	car or other	332	33.2%
	unknown/no property	154	15.4%
Other Installment Plans	bank	139	13.9%
	stores	47	4.7%
	none	814	81.4%
Housing	rent	179	17.9%
	own	713	71.3%
	for free	108	10.8%
Job	unemployed/ unskilled - non-resident	22	2.2%
	unskilled - resident	200	20%
	skilled employee / official	630	63%
	management/self-employed/officer	148	14.8%
Telephone	none	596	59.6%
	yes	404	40.4%
Foreign Worker	yes	963	96.3%
	no	37	3.7%

Table 3: Summary statistics for categorical features in the German Credit Dataset.

B Appendix - Quantlets

References

Athey, S. (2017). The impact of machine learning on economics. In *Economics of Artificial Intelligence*. University of Chicago Press.

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).

Bottou, L. (2012). Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer.

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).

Li, M., Andersen, D. G., Smola, A. J., and Yu, K. (2014). Communication efficient distributed machine learning with the parameter server. In *Advances in Neural Information Processing Systems*, pages 19–27. NIPS.

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

Wager, S. and Athey, S. (in press). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*.

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 multistage 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.