Operations Research Forum (2022) 3:67
https://doi.org/10.1007/s43069-022-00177-1

RESEARCH

Credit Scoring with Drift Adaptation Using Local Regions of Competence

Dimitrios Nikolaidis[1,2] · Michalis Doumpos[1]

Received: 12 September 2022 / Accepted: 9 November 2022 / Published online: 25 November 2022

Abstract

Despite the advances in machine learning (ML) methods which have been extensively applied in credit scoring with positive results, there are still very important unresolved issues, pertaining not only to academia but to practitioners and the industry as well, such as model drift as an inevitable consequence of population drift and the strict regulatory obligations for transparency and interpretability of the automated profiling methods. We present a novel adaptive behavioral credit scoring scheme which uses online training for each incoming inquiry (a borrower) by identifying a specific region of competence to train a local model. We compare different classification algorithms, i.e., logistic regression with state-of-the-art ML methods (random forests and gradient boosting trees) that have shown promising results in the literature. Our data sample has been derived from a proprietary credit bureau database and spans a period of 11 years with a quarterly sampling frequency, consisting of 3,520,000 record-months observations. Rigorous performance measures used in credit scoring literature and practice (such as AUROC and the H-Measure) indicate that our approach deals effectively with population drift and that local models outperform their corresponding global ones in all cases. Furthermore, when using simple local classifiers such as logistic regression, we can achieve comparable results with the global ML ones which are considered "black box" methods.

Keywords Concept/population drift · Adaptive models · Local classification · Behavioral credit scoring · Lazy learning · Region of competence

✉ Michalis Doumpos
mdoumpos@tuc.gr

Dimitrios Nikolaidis
dnikolaid@gmail.com

[1] School of Production Engineering and Management, Technical University of Crete, Kounoupidiana, Greece

[2] Department of Research and Development, Tiresias S.A, Attica, Greece

1 Introduction

Information asymmetry has far reaching and well-studied consequences in the operation of financial markets, such as the impact on financial inclusion, financial intermediation and financial risk; see [1–5]. Thus, credit bureaus have emerged as the means to diminish information asymmetry and support the efficiency of credit institutions in their decision-making processes, and in tasks such as credit limit management, debt collection, cross-selling, risk-based pricing, prevention of fraud, etc. [6–8]. Credit scoring, as a principal tool of credit bureaus to identify good prospective borrowers, began as early as 1941 [9]. However, the automated and widespread application of credit scoring did not take place until the 1980s, when computing power to perform sophisticated statistical calculations became affordable. One definition of credit scoring is “the use of statistical models to transform relevant data into numerical measures that guide credit decisions” [10]. According to Thomas et al. [11], credit scoring has been vital in the “…phenomenal growth in the consumer credit over the last five decades. Without (credit scoring techniques, as) an accurate and automatically operated risk assessment tool, lenders of consumer credit could not have expanded their loan (effectively).”

However, credit scoring modeling and methodologies face theoretical issues as well as practical ones (as operated in practice by all credit bureaus):

- As with all predictive models, credit scoring suffers from population (or concept) drift, i.e., changes in the socio-economic environment cause the underlying distribution of the modeled population to change over time. [12–16]. To tackle this problem in practical terms, credit bureaus implement continuous monitoring cycles and periodic re-calibration or re-development of their models [10, 17, 18]. The calibration of credit scoring models or the lack thereof, has been mentioned in the literature as one reason (among others) for the subprime mortgage crisis of 2008 [19]. Specifically, FICO scores have been shown to having become a worse predictor of default between 2003 to 2006 [20, 21]. During that period, despite the rapid and severe deterioration of subprime portfolio quality, corresponding scores remained fairly stable [22].
- Development of credit scoring models require historical data of at least 1–2 years. Without counting the monetary cost incurred by such operations, adding the time to implement and put into production a new generation of models, sometimes results in a difference of three or more years between actual data that reflect the current population dynamics and the data used to build the models. This lag between data at model development time and actual time to be put into production, has become more obvious as data are generated in an ever-increasing pace and this acceleration puts an equally pressing pace in operations.
- Moreover, as credit scoring models depend on pre-defined sets of predictor (input) variables, when their weights are updated from time to time, they may lose their relevance and end up with a weight zero or close to zero. These pre-

dictors are called omitted variables and it has been shown that the omission of variables related to local economic conditions seriously bias and weaken scoring models [23].

- Credit bureaus do not use a single scoring model (sometimes referred to as "scorecard") for a specific purpose (such as estimation of the probability of default), but rather split the population into various segments using either demographic criteria, or risk-based ones. This happens for various reasons such as data availability (e.g., new accounts versus existing customers), policy issues (e.g., different credit policies for mortgages), inherently different risk-groups, etc., in order to (a) capture significant interactions between variables among the sub-population that are not statistically important within the entire population or cause the relevance of predictors to change between groups [24], (b) capture non-linear relationships (especially on untransformed data) and increase the performance of generalized linear models [24], which are even today the "golden standard" in the credit scoring industry (although to a far lesser extent than in past decades). Despite the fact that there is not enough academic consensus about the effects of segmentation in scorecards' performance [25], segmentation is a de facto approach throughout the credit scoring industry for another reason: robustness.

In this work, we investigate the use of local classification models for dynamic adaptation in consumer credit risk assessment aiming to handle the population drift and avoid the time-consuming endeavor of continuous monitoring and re-calibration/re-development procedures. The proposed adaptive scheme, searches the feature space for each candidate borrower ("query instance") to construct a "micro-segment" or local region of competence, using the K nearest neighbors algorithm (kNN). Thus, a region of competence is exploited as a localized training set to feed a classification model for the specified individual. Such a specialized local model serves as an instrument to achieve the desired adaptation for the classification process. We compare various classifiers (logistic regression as well as ML methods such as random forests and gradient boosting trees). All the explored algorithms are fed to training features extracted from a credit bureau proprietary database and evaluated in an out-of-sample/out-of-time validation setting in terms of performance measures including AUC and H-Measure [26]. Specifically, we explore three hypotheses:

H1: Do local methods outperform their corresponding global ones?
H2: Do results using ML methods differ significantly from logistic regression in the global as well as in the local setup?
H3: Does the choice of kNN-based local neighborhoods affect model performance over choosing randomly selected regions?

The results demonstrate the competitiveness of the proposed approach as opposed to the established methods. Thus, our contributions can be summarized as follows:

- Our analysis is using a real-world, pooled cross-sectional data set spanning a period of 11 years, including an economic recession, and containing 3,520,000 record-months observations and 125 variables. Availability of adequate, real-world credit related data is extremely scarce in the literature. In a very extensive benchmark study by [27] 28 papers were surveyed in terms of data sets used; the mean number of records/variables of all datasets was 6167/24, whereas the biggest dataset used in the study had 150,000 observation and 12 independent variables. Also, small datasets have been noted in the literature that may introduce unwanted artifacts and models built upon them do not scale up when put into practice [28, 29].
- Using local classification methods there is no need for continuous calibration of the models; adaptation to concept drift is part of the dynamic and automated model building process.
- Predictive models are always trained on the latest available data. The predictors used in the models are not fixed but they are always picked up to fit the changing conditions, thus bypassing the problem of omitted variables.
- For each query, a specialized micro-segment or region of competence is created dynamically, thus reaping the benefits of segmentation.
- Last by not least, the proliferation of ML/artificial intelligence methods for predictive modelling created a paradigm shift for the credit scoring as well [30–38]. The issue of performance improvement is but one side of the discussion, the other one being related to issues such as transparency, bias and fairness [39–44], which in the context of credit scoring have received special attention [45–47] due to the statutory and regulatory constraints (cf. GDPR, EU AI Act: COM/2021/206 final). In our work, we focus on the performance aspect and we compare statistical classification models versus well-advertised ML methods.

The rest of this paper is organized as follows. In Sect. 2, we present the theoretical background; Sect. 3 provides a formulation of the problem; Sect. 4 describes the experimental setup and all its parameters; Sect. 5 provides the empirical results; and Sect. 6 concludes with discussion of these results and possible directions of future work.

2 Background and Related Theoretical Work

2.1 Local Classification

Usually, the classification process is a two-phase approach that is separated between processing training and test instances:

- Training phase: a model is constructed from the training instances.
- Testing phase: the model is used to assign a label to an unlabeled test instance.

In global or eager learning, the first phase creates pre-compiled abstractions or models for learning tasks, which describe the relationship between the input variables and the output over the whole input domain [48]. In instance-based learning (also called lazy or local learning), the specific test instance (also called query), which needs to be classified, is used to create a model that is local to that instance. Thus, the classifier does not fit the whole dataset but performs the prediction of the output for a specific query [49–52].

The most obvious local model is a k-nearest neighbor classifier (kNN). However, there are other possible methods of lazy learning, such as locally-weighted regression, decision trees, rule-based methods, and SVM classifiers [53–55]. Instance-based learning is related to but not quite the same as case-based reasoning [56–59], in which previous examples may be used in order to make predictions about specific test instances. Such systems can modify cases or use parts of cases in order to make predictions. Instance-based methods can be viewed as a particular kind of case-based approach, which uses specific kinds of algorithms for instance-based classification.

Inherent to the local learning methods is the problem of prototype or instance selection where it can be defined as the search for the minimal set S in the same vector space as the original set of instances T, subject to accuracy(S) $\geq$ accuracy(T), where the constraint means that the accuracy of any classifier trained with S must be at least as good as that of the same classifier trained with T [60–62]. Instance selection methods can be distinguished based on their properties such as the direction of search for defining S (e.g., incremental search, where search begins with an empty S) and wrapper versus filter methods, where the selection criterion is based on the accuracy obtained by a classifier such as kNN, versus not relying on a classifier to determine the instances to be classified [60].

However, we shall *distinguish instance selection* from *instance sampling* de Haro-Garcia et al. [63], where the purpose is to formulate a suitable sampling methodology for constructing the training and test datasets from the entire available population. In particular, instance sampling deals with issues such as sample size and sample distribution (balancing; [64–66] and has been shown to be of major importance for credit scoring due to the inherent imbalance in the credit scoring data [67].

There are three primary components in all local classifiers [48, 49]:

1. Similarity or distance function: This computes the similarities between the training instances, or between the test instance and the training instances. This is used to identify a locality around the test instance.
2. Classification function: This yields a classification for a particular test instance with the use of the locality identified with the use of the distance function. In the earliest descriptions of instance-based learning, a nearest neighbor classifier was assumed, though this was later expanded to the use of any kind of locally optimized model.
3. Concept description updater: This typically tracks the classification performance and makes decisions on the choice of instances to include in the concept description.

A specific mention shall also be made to the concept of local weighted regression [53, 68–70] where the core idea lies on local fitting by smoothing: the dependent variable is smoothed as a function of the independent variables in a moving fashion analogous to a moving average. In similar manner kernel regression uses a kernel as a weighting function to estimate the parameters of the regression, i.e., the Nadaraya-Watson estimator [71, 72].

Local classification methods have not been studied extensively specifically in the context of credit scoring. Simple models such as basic kNNs expectedly do not yield satisfying results [27] and thus have not drawn much of the interest of the academic community nor of the practitioners for that matter. Some effort using advanced and/or hybrid methodologies such as self-organizing maps for clustering [73], combining kNN with LDA and decision trees [74], clustered support vector machines [75], fuzzy-rough instance selection [76], instance-based credit assessment using kernel weights [77], have shown somewhat promising results, albeit bearing into consideration the issues airing from the datasets used (size, relevance, real-world applicability).

2.2 Local Regions of Competence

Ensemble methods also known as Multiple Classifier Systems (MCS) combine several base classifiers through a conceptual three-phase process [78–81]:

1. Pool generation, where a diverse pool of classifiers is generated,
2. Selection, where one or a subset of these classifiers is selected, and
3. Integration, where a final prediction is made based on fusing the results of the selected classifiers.

The selection phase can be static or dynamic. Static selection consists of selecting base models once and using the resulting ensemble to predict all test samples, whereas in dynamic selection specific classifiers are selected for each test instance through evaluation of their competence in the neighborhood or otherwise on a local region of the feature space where the test instance is located. Thus, the neighbors of the test instance define a local region which is used to evaluate the competence of each base classifier of the ensemble.

The definition of the local region has been shown to be of importance to the final performance of dynamic selection methods [82, 83, 103] and there are papers pointing out that this performance can be improved by better defining these regions and selecting relevant instances [83–86]. One of the most common methodologies for defining local regions is kNNs (including its variations such as extended kNNs, especially for imbalanced data, which are of particular importance to credit scoring). Methods such as clustering [87, 88] can also be found in the literature.

Dynamic selection techniques in the context of credit scoring have received some attention in the literature [89–94]. In a recent paper, Melo Junior et al. [95] proposed a modification of the kNN algorithm, called reduced minority kNNs (RMkNN), which aims to balance the set of neighbors used to measure the competence of the base classifiers. The main idea is to reduce the distance of the minority samples from the predicted instance. As mentioned, imbalancing of the distribution of the classes is an important factor when considering sampling for credit scoring [64, 67, 85, 86, 93, 96, 97]. This issue becomes even more important when dynamic selection techniques are applied.

A related approach is the Mixture of Experts, which is composed of many separate neural networks, each of which learns to handle a subset of the complete set of training cases [98–101]. This method is established based on a divide-and-conquer principle [102], where the feature space is partitioned stochastically into several subspaces through a special employed error function and "experts" become specialized on each subspace. However, only multilayer perceptron neural networks are used as the base classifier [78, 103]. Mixture of Experts has not been extensively applied in the context of credit scoring and there are only a few studies on the subject [104, 105].

3 Problem Formulation and Parameters

Assuming a classification training set$\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$,$\mathbf{x} \in \mathbb{R}^n$,$y \in \{0, 1\}$, M is a global model trained on all $\left\{\left(\mathbf{x}_i, y_i\right)\right\}_{i=1}^{n}$, the local region of competence for a given test instance $\mathbf{x}$ (assuming its k-nearest neighbors) is denoted by $N_x = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\}$ and the learning set for the local classifier M_x is$\left\{\left(\mathbf{x}_i, y_i\right)\right\}_{\mathbf{x}_i \in N_x}$.

Specifically, for the credit scoring binary classification problem $\{\mathbf{x}_i\}$, $i = 1, \dots, n$, is considered the feature or variable space, denoting the characteristics of each borrower i and y_i is the corresponding objective or target variable denoting the class label (non-default or default sometimes referred also as "Good" or "Bad"). Each feature vector $\mathbf{x}_i$ is observed at a point in time T_0, called observation point, whereas the corresponding response y_i is recorded at a subsequent performance point $T_1 = T_0 + \tau$, where $\tau \geq 1$ is usually defined in months. The collected input data span an observation time window (or observation window) covering the period $[T_0 - \tau', T_0]$ ($\tau' \geq 1$ denoting months), whereas the outcome window refers to the period $(T_0, T_1]$ where the class label of y_i is defined. For the context of behavioral credit scoring, the feature space contains variables related the financial performance and behavior of borrowers such as credit amounts, delinquency status, etc.

The credit scoring literature has not provided definitive answers to defining optimally these parameters (default definition, observation window, outcome window). The recommendations in the literature vary the length of observation and outcome windows from 6 to 24 months [8, 11, 106].

Regarding the definition of default, Anderson [10] designated that financial institutions choose between: (a) a current status definition that classifies an account as good or bad based on its status at the end of the outcome window, and (b) a worst status approach that uses a time-period during the outcome window. Regulatory requirements are also of paramount importance and must be taken into consideration, such as a 90 days past due worst status approach that is commonly used in practice in behavioral scorecards and complies regulatory requirements, such as the Basel Capital Accords and the new definition of default by the European Banking Authority (EBA). Kennedy et al. [107] presented a comparative study of various values for these parameters. Their results indicated that behavioral credit scoring models using:

- default definitions based on a worst status approach outperformed those with current status.
- a 12-month observation window outperformed the ones with 6- and 18-month windows in combination with shorter (12 months or less) outcome windows.
- 6-months outcome window and a current status definition of default outperformed longer outcome windows; for the worst status approach the degradation occurs when outcome window extends beyond 12 months.

Finally, it should also be noted that credit scoring data sets are highly imbalanced, since the objective of all financial institutions is a low-default portfolio. There are quite a few studies and approaches in the literature analyzing the impact of imbalancing in classification, in general [108–114], as well as in the context of credit scoring [64, 67, 84, 93, 96, 115].

4 Experimental Setup and Methodology

4.1 Data and Variables

Our data set (pooled cross-sectional data) has been derived from a proprietary credit bureau database in Greece and spans a period of 11 years (2009q1 to 2019q4), resulting in total 44 snapshots (11 years by 4 quarters). At each snapshot, a random sample of 80,000 borrowers was retrieved with all their credit lines, including paid off and defaulted, resulting in 3,520,000 record-months observations.

In total, 125 proprietary credit bureau behavioral variables were calculated at the borrower level which fall within the following dimensions:

- Type of credit (consumer loans, mortgages, revolving credit such as overdrafts, credit cards, restructuring loans, etc.).
- Delinquencies (months in arrears, delinquent amount, etc.).
- Amounts (Outstanding balance, disbursement amount, credit limit, etc.).
- Time (months since approval, time from delinquencies, etc.).
- Inquiries made to the credit bureau database.
- Derogatory events, such as write-offs or events from public sources such as courts.

Besides "elementary" variables such as the ones described above, other derivative/combinatory variables along various dimensions were calculated, such as various ratios (ratio of delinquent balance over current balance for the last *X* months for a specific type of credit line), utilizations and their rate of their increase or decrease over a specific time-window (e.g., consecutive increase over last X months), giving the total of 125 variables.

4.2 Scoring Parameters

Our scoring parameters are defined as follows:

- *Observation window:* Time windows of 12 months prior to each observation point T_0. Our initial observation point has been at 2009q1 and every subsequent quarter thereafter up to 2018q4.
- *Scorable population:* At each observation point T_0, the following cases are excluded from the analysis: a) borrowers already having delinquency of 90 days past due (dpd) or more at T_0, b) cases lacking sufficient historical data i.e., less than 6 months of credit history, credit cards which are inactive balance within the observation window. The remaining observations constitute *the scorable population* for the specific T_0. The last T_0 is taken at 2018q4.
- *Outcome window:* a 12-month window after the observation point. For each observation point T_0, the period $T_1 = T_0 + 12$ is used as the outcome window. Thus, the last T_1 is taken at 2019q4.
- *Default definition*: The labeling of the scorable population at T_0 either as GOOD = 0 (majority class), BAD = 1 (minority or “default” class), depending on the information available during T_1, takes place using a worst status approach for each outcome window, i.e., the maximum (worst) delinquency over all accounts or a new derogatory event, is measured for the specific outcome window. Thus the corresponding classes are defined as: (a) $y = 1$ for cases with worst delinquency $\geq$ 90 dpd or a derogatory event occurs during the outcome period, otherwise (b) $y = 0$ is assigned to all other cases.

4.3 Methodology

Our approach is based on training local and global classifiers on the same sample and comparing their performance. Local classifiers are trained *for each instance* $\mathbf{x}$ of the test data set of each snapshot using the feature space defined by its neighborhood or region of competence within the training data set. A local model M_x is then used to predict the probability and the class label of the specific instance for which it was trained. Correspondingly, global classification models are trained on the entire training set and then used to predict the class probabilities of each instance on the test data set. For better simulating a real-world scenario, we retrain global classifiers every 2 years. The classifiers used both in the global as well as in the local scheme are logistic regression, random forests (RF), and extreme gradient boosting machines (XGB). The choice of the specific ML models was made based on recent credit scoring literature findings where they seem to be on par or outperform other machine learning and deep learning methods [32]. Specifically, Gunnarsson et al. [33] found that XGBoost and RF outperformed deep belief networks (DBN), Hamori et al. [34] found XGB to be superior to deep neural networks (DNN) and RF. Marceau et al. [35] found that XGB performed better than DNN, and Addo et al. [30] concluded that both XGB and RF outperform DNN.

For implementation we used Microsoft R Open v3.5.1 and the corresponding R libraries: speedglm 0.3–2, randomForest 4.6–14 and xgboost 0.71.2. In all cases, default parameter values were used and no hyper-parameter optimization was performed other than internally used by the methods.

During the training phase, the input data have been *pre-processed* using an expert-based process flow to:

- handle missing values, by excluding variables with greater than 70% missing values and filling the remaining blanks with a constant (since the variables are missing at random (MAR), in this work we use – 1 as constant value),
- retain only the useful variables, by removing those with zero variance or near zero variance,
- isolating non-correlated variables using an exclusion threshold of 0.7, and
- select the most discriminative among the remaining variables using the Information Value (IV) criterion. The exclusion thresholds were selected to match a practitioner's rule mentioned in the literature [18], where a variable is removed in case of having an IV lower than 0.3 and greater than 2.5.

Finally, as it has been noted in Sect. 3, credit scoring data are inherently imbalanced. In our case, the imbalancing is also observed in the regions of competence, which are used to build the local classification models. Such a fact, inevitably yields in some cases to non-convergence errors, when local logistic regression is used as a classification algorithm and the local region of competence contains very few minority class (default) cases for the algorithm to converge. In our experiments we found this non-convergence error to be on average 1.9% over all executions.[1] To address the non-convergence issue, in this work, we use a simple heuristic rule: anytime logistic regression algorithm fails to predict a class label for a test instance, the algorithm assigns the majority class from test instance's *region of competence*.

4.4 Local Classification

As detailed below, for each snapshot, the k nearest neighbors (k-NN) algorithm is used to define the *local region of competence* N_x for each test instance $\mathbf{x}$. A local model M_x is trained on this specific region N_x, which serves as an instrument to achieve the desired adaptation for the classification process. Figure 1 shows the overall flow for the proposed scheme:

The setup procedure is as follows: for each *snapshot*, the *scorable population* is defined as a random set (of 80,000 instances), sampled without replacement from the total population and the resulting data set is separated through a 50–50

[1] In total we executed 120 runs for local LR models (one run over all 40 snapshots for each k, where $k = \{2000,4000,6000\}$ the size of kNNs.

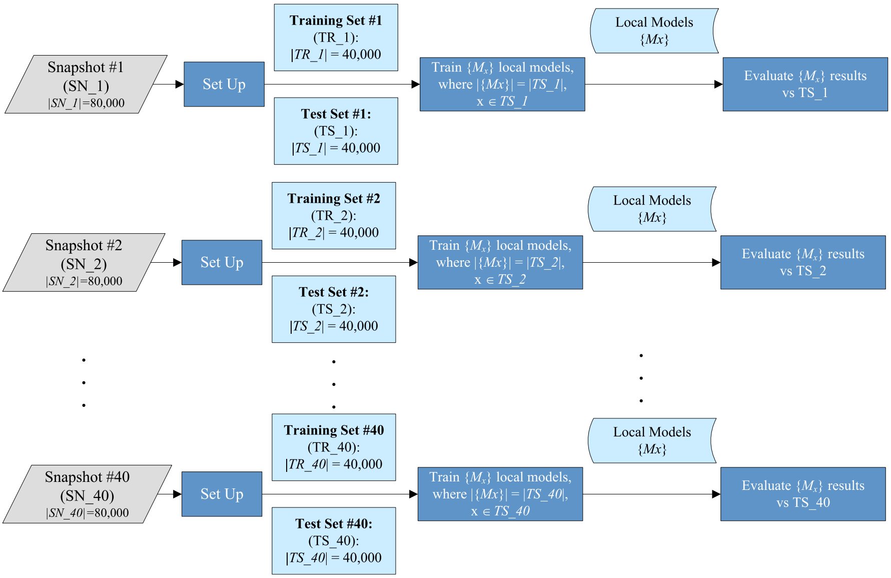


Fig. 1 High-level flow for the proposed local classification scheme (|S| denotes the cardinality of a set S)

split into training and test sets, to form the training and test sub-spaces of the original *feature space*. The distance metric used to define the local region of competence for each test instance, is determined using the *Euclidean* distance. Such a *region of competence* serves as a borrower-specific localized training set that will be used to build a *local classification model* for that borrower.

Regarding the size of the *k* parameter required by the nearest neighbors algorithm, it is worth to note a common rule of thumb that defines the selection of 1500 to 2000 examples per class, dating from the very beginning of credit scoring model development [116] and mentioned in many works thereafter [18, 24, 117]. Although the subject is not extensively researched, recent academic studies pointed to the direction that larger samples can improve the performance of linear models [67, 117] but there seems to be a plateau after 6000 goods/bads and almost no further benefit above 10,000. As a result, aiming to evaluate both claims, in this work we selected a *k* parameter that ranges from 2000 to 6000 examples ($k \in \{2000, 4000, 6000\}$). The resulting *region of competence* is used to train a *local classification model*, M_x, which is specialized for the corresponding test instance/borrower. In this study, local classification models are built using the classification algorithms considered in the analysis (i.e., logistic regression, random forests, gradient boosting trees). Figure 2 depicts the training phase for the proposed scheme (*pre-processing* refers to the flow described in Sect. 4.3).

To assess the performance of each *local classification model* M_{x_i}, which had been built for each test instance $\mathbf{x_i}$ on its specific *region of competence* N_{x_i}, i = {1,…|TS_L|} (where *i* is the number of the data points in the test set #L) is used to predict the probability of default (PD) for the considered test instance/

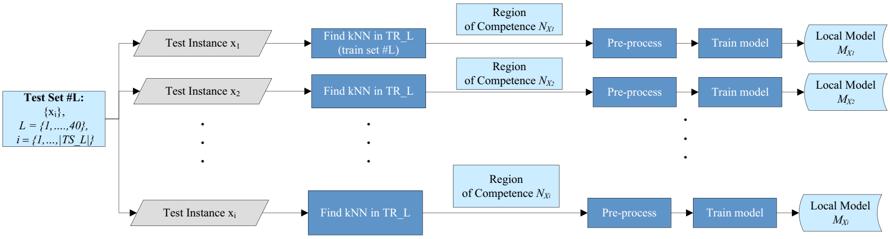


Fig. 2 Training phase for the proposed local classification scheme (|S| denotes the cardinality of a set S)

candidate borrower and assign a GOOD or BAD class label. This is compared to the actual labels available for the test instances.

4.5 Global Classification

As a baseline to benchmark our proposed local classifiers, we implement and evaluate a standard credit scoring classification scheme commonly used both by the scientific community and practitioners alike. In the global classification approach, the adaptation to population drift is achieved by retraining the models using new data from the contextual snapshot. Figure 3 shows the overall flow for the global scheme.

It should be noted that in order to have a real-word and realistic comparison of model performance we re-train our global models every two years (as retraining is applied in

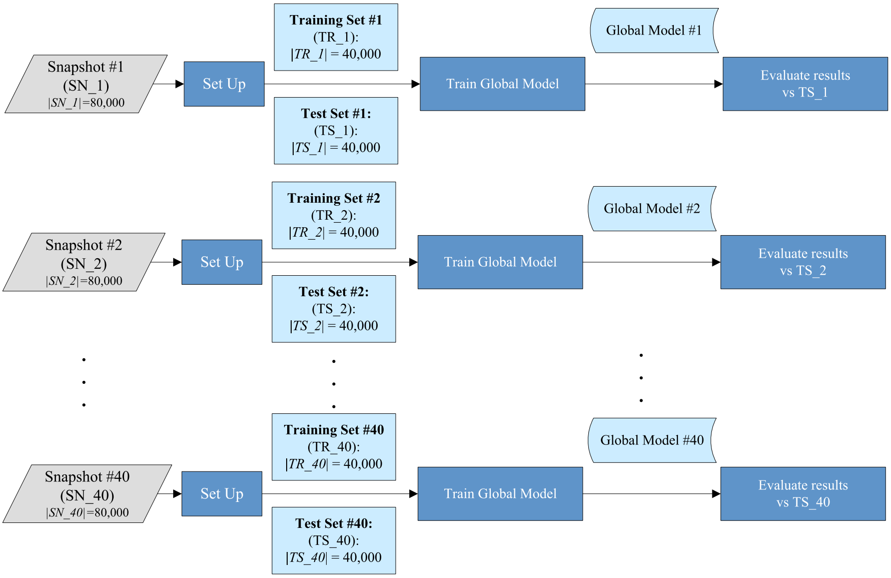


Fig. 3 Global classification scheme (|S| denotes the cardinality of a set S)

practice to all commercial credit scoring models). The performance of global models over all snapshots would degrade significantly in case of training only once for the initial snapshot data (indicatively: mean AUC = 0.8213 with standard deviation = 0.04 for global LR models when the training took place only at the first snapshot 2009q1 versus mean AUC = 0.8746 with standard deviation = 0.014 when the re-training of global LR occurs every 2 years).

4.6 Performance Measures and Comparison of Classifiers

There is a keen interest of the scientific research community regarding the appropriateness of the established performance measures used to evaluate classification models and especially those which are used in credit scoring applications, also considering the inherent imbalance of the credit scoring datasets [118–120]. Specifically, the credit scoring setup gives rise to methodological problems such as the accuracy paradox [121] and the different misclassification cost between type I and type II errors [26]. As a result, the most used approach avoids accuracy as a scorecard performance metric, adopting instead measures such as the area under the ROC (AUC), the GINI index, and the Kolmogorov–Smirnov distance or the F-measure. However, in the literature there has been a skepticism over their appropriateness and especially of the widely used AUC measure [122]. A coherent alternative namely the H-measure [26, 122, 123] has been proposed in the literature, which handles different misclassification costs and is indicated to be a better suited performance metric for the credit scoring context [120]. Thus, in this work, we use both AUC and the H-measure (using default values for the parameters for the calculation of H-measure as defined in the corresponding R package).

Comparisons among several classification algorithms on several datasets arise in machine learning when a new proposed algorithm is compared with the existing state of the art. From a statistical point of view, the correct way to deal with multiple hypothesis testing is by, first, comparing all the classification algorithms together by means of an omnibus test to decide whether all the algorithms have the same performance. Then, if the null hypothesis is rejected, we can compare the classification algorithms by pairs using post-hoc tests. In these kinds of comparisons, common parametric statistical tests such as ANOVA are generally not adequate as the omnibus test. The arguments are similar to those against the use of the t-test: The scores are not commensurable among different application domains and the assumptions of the parametric tests (normality and homoscedasticity in the case of ANOVA) are hardly fulfilled [124–126]. In this paper we use the non-parametric tests of Nemenyi post hoc and Friedman's aligned ranks. The selection of non-parametric tests is made because the underlying data distribution is not known. Since multiple classifiers should be compared, the Nemenyi test is selected for the pairwise comparisons among scheme and algorithm combinations, as proposed by Demsar [124]. Furthermore, Friedman's aligned rank test is utilized to correct the p values for multiple testing.

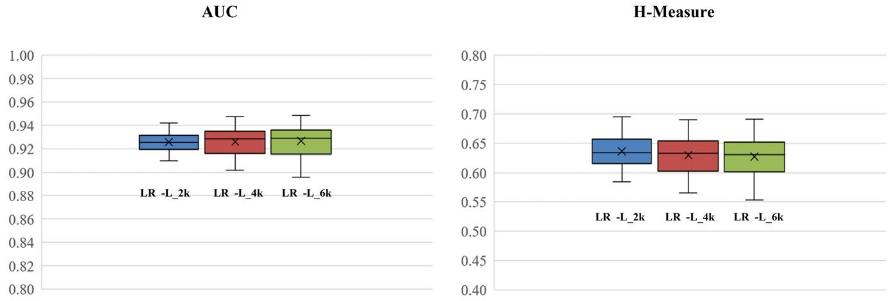


Fig. 4 Average performance for local LR on different $k = \{2000, 4000, 6000\}$

5 Empirical Results

For tackling the hypothesis regarding the superiority of local models over their global counterparts, we started by examining whether the size of the local region impacts the classification performance. Figure 4 summarizes the performance of local LR models for various k's whereas Table 2 in the Appendix provides the detailed results over all snapshots.

As evidenced the choice of k does not have a significant impact performance of logistic regression. Specifically, we observe that when using the H-measure, the performance results are slightly and non-significantly decreasing as k increases (mean = 0.6360, 0.6298, 0.6270 for $k = 2000, 4000, 6000$, correspondingly), whereas the opposite holds when using AUC as performance measure (mean = 0.9256, 0.9259, 0.9265 for corresponding k's). Thus, for the rest of our process we choose to use $k = 2000$ for local models since model performance is not significantly affected, whereas computational performance and memory requirements are considerably improved with lower k's.

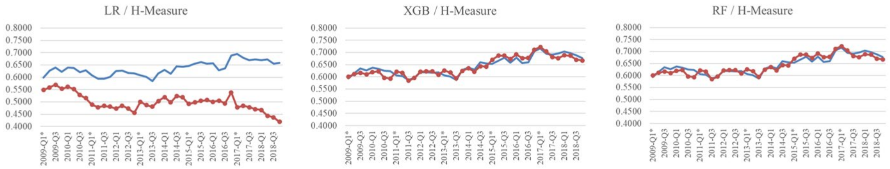
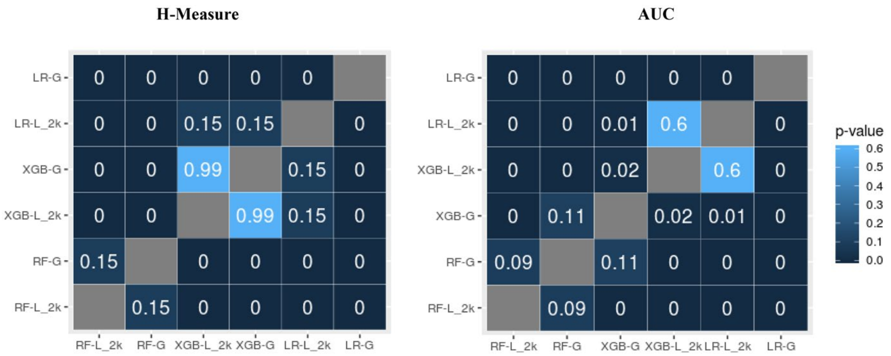


Fig. 5 Pairwise visual comparison between local/global classifiers (different y-axis scales, * = training snapshot for global classifiers) (LR = logistic regression, RF = random forrest, XGB = gradient boosting; solid blue line denotes local classifier, red line with markers global classifier)



H-Measure

	RF-L_2k	RF-G	XGB-L_2k	XGB-G	LR-L_2k	LR-G
LR-G	0	0	0	0	0	
LR-L_2k	0	0	0.15	0.15		0
XGB-G	0	0	0.99		0.15	0
XGB-L_2k	0	0		0.99	0.15	0
RF-G	0.15		0	0	0	0
RF-L_2k		0.15	0	0	0	0

AUC

	RF-L_2k	RF-G	XGB-G	XGB-L_2k	LR-L_2k	LR-G
LR-G	0	0	0	0	0	
LR-L_2k	0	0	0.01	0.6		0
XGB-L_2k	0	0	0.02		0.6	0
XGB-G	0	0.11		0.02	0.01	0
RF-G	0.09		0.11	0	0	0
RF-L_2k		0.09	0	0	0	0

Fig. 6 *p* Values of the pairwise differences from Friedman's aligned rank test (LR = logistic regression, RF = random forest, XGB = gradient boosting, L = local classifier, G = global classifier, 2 k = 2000 for kNN)

Comparing visually the results of the local classifiers with their corresponding global ones, we get a mixed picture (see Tables 3 and 4 in the Appendix for detailed results): whereas local LR models outperform their global counterparts, for XGB and RF the differences between global and local classifiers do not appear to be significant (Fig. 5).

To test for statistical differences between all classifiers (i.e., the case of multiple methods on multiple data sets as noted in [124], we use Friedman's aligned rank test [125] to assess all the pairwise differences between algorithms and then correct the *p* values for multiple testing (Fig. 6 visualizes the results in matrix format). We observe that in both measures (AUC and H-Measure) LR-G differs significantly from all other classifiers. Going in more details, in the AUC-based matrix two "clusters" of classifiers emerge for which the null hypothesis of not been equal cannot be rejected: a) XGB-G, RF-G, RF-L_2k and b) LR-L_2k and XGB-L_2k. For the H-measure-based *p* value matrix, the analogous "clusters" observed are as follows: (a) RF-L_2k, RF-G and (b) XGB-G, XGB-L_2k, LR-L_2k. Thus, there seems to be an "interlacing" between the performance of all ML models (both local and global) and LR-L_2k which cannot be statistically

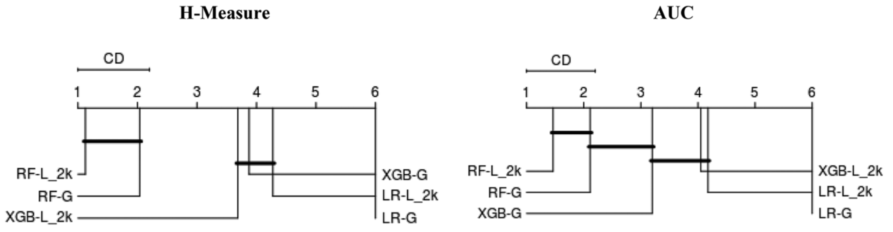


Fig. 7 Critical distances between local and global classifiers (LR = logistic regression, RF = random forest, XGB = gradient boosting, L = local classifier, G = global classifier, 2 k = 2000 for kNN)

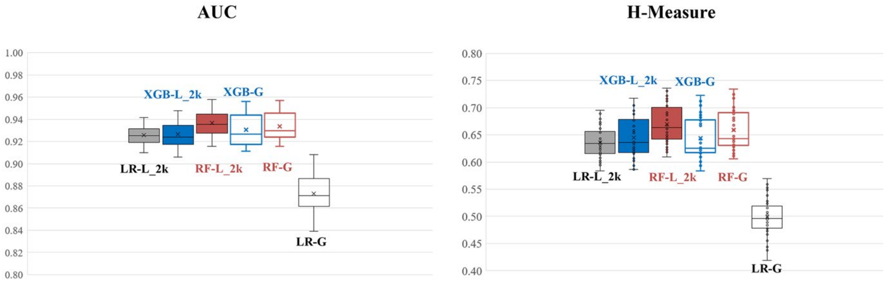


Fig. 8 Average performance over 44 snapshots (different *y*-axis scales) (LR = logistic regression, RF = random forest, XGB = gradient boosting, L = local classifier, G = global classifier, 2k = 2000 for kNN)

rejected and strengthens the evidence that local models are at least on par with their global counterparts. Especially for LR-L it is clearly evidenced that it outperforms LR-G with statistical significance.

As a next step, we use the Nemenyi post hoc test that is designed to check the statistical significance between the differences in the average rank of a set of predictive models. In the resulting *critical distance (CD)* graph (Fig. 7), the horizontal axis represents the average rank position of the respective model. The null hypothesis is that the average ranks of each pair of predictive models do not differ with statistical significance of 0.05. Horizontal lines connect the lines of the models for which we cannot exclude the hypothesis that their average ranks are equal. Any pair of models whose lines are not connected with a horizontal line can be seen as having an average rank that is different with statistical significance. On top of the graph a horizontal line is shown with the required difference between the average ranks (known as the critical distance or difference) for two pair of models to be considered significantly different.

Thus, it is further evidenced that the case of local LR consistently and statistically significantly outperforms global LR although the same conclusion does not seem to hold for RF and XGB, despite the minor difference in favor of the local methods when comparing average performance. This becomes more apparent upon examining the average AUC and the H-Measure over all snapshots (Fig. 8).

It is also noteworthy that although RF outranks XGB (in all cases; differences not statistically significant), the performance of Local LR does not differ statistically from the ML algorithms, contrasting the case of global LR which is vastly outranked and outperformed. The gain, when comparing these classifiers to the "baseline" global LR, is within the range of 6–8% (Table 1), which is well within the empirical

Table 1 Gain in AUC/H-Measure with respect to LR-G (LR = logistic regression, RF = random forest, XGB = gradient boosting, L = local classifier, G = global classifier, 2 k = 2000 for kNN)

	H-Measure	AUC
XGB-GL	29.02%	6.61%
RF-GL	32.09%	6.94%
XGB-L	29.22%	6.16%
RF-L	**34.24%**	**7.31%**
LR-L_2k	27.53%	6.04%

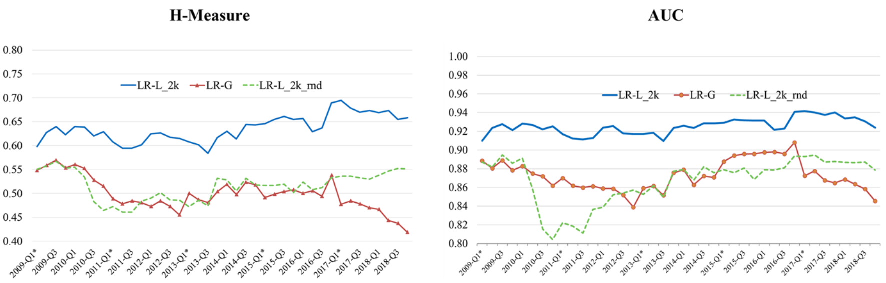


Fig. 9 kNNs vs random regions (different *y*-axis scales) (LR = logistic regression, G = global classifier, 2 k = 2000 for kNN, * = training snapshot for global LR)

range observed in other studies [32] when comparing ML algorithms to the basic logistic regression in credit scoring.

Finally, to examine whether the choice of a specific local region based on kNNs vs random sub-sampling plays a role in the performance, we trained a series of models LR-L_2k_rnd where for each test instance **x** its local region N_x is a set of randomly selected training cases, instead of employing the kNN scheme. Detailed results are provided in.

Table 5 (Appendix) whereas the following Fig. 9 highlights the fact that selecting local regions through kNNs does makes a difference and a performance gain with respect when to a random choice of regions. It should be noted here that the performance of LR-L_2k_rnd appears somewhat similar to the global one LR-G. This is of no surprise, since the attributes of a random sample are, by selection, more similar to the overall population from which the sample is drawn than from a sub-region with specific characteristics.

6 Conclusions and Future Work

The development of reliable models for credit scoring remains a challenge for researchers and practitioners. Technological advances in ML/AI provide new capabilities in this field, enabling the exploitation of large amounts of data. However, as conditions in the economic and business environment are in constant change, credit scoring models require regular updating. Motivated by this finding, this paper presented an adaptive behavioral credit scoring scheme which uses online training to provide estimates for the probability of default through an instance-specific basis.

Going back to our research hypotheses we can draw our conclusions:

> H1: With respect to the potential gain of local methods vis-a-vis their global counterparts our results indicate clearly that local logistic regression outperforms and outranks the baseline global logistic regression. This does not seem to hold for the ML methods we used (RF and XGB) where the differences between local and global models are not statistically significant.

H2: Concerning the superiority of ML methods over baseline LR-G our results fall within a range of performance improvement of 2–8% observed in various credit scoring applications of ML/AI found in literature [30–34, 127]. However, it is quite important to observe that the performance of Local LR is on par with RF and XGB.

H3: Finally, our analysis clearly indicates that the performance of a local model is affected by the selection of a region of competence based on similar characteristics with the queried test instance. A random selection of points from the feature space provides inferior results compared to the kNN approach adopted in this study.

Bearing into consideration the volume of the real-world data used and the extensive out-of-sample validation performed, thus safeguarding for overfitting, our work clearly indicates that using local LR methods can provide real-time adaptation therefore providing a solution to the problem of population drift and the need for continuous re-calibration (which holds for LR and ML models alike), yielding comparable results with complex state-of-the-art ML algorithms. Additionally, LR per se is not a "black box" model which is extremely beneficial for regulatory purposes. However, dealing with the complexities of model risk management and governance [128–130] in the case of using real-time, adaptive local models may pose equal or even greater challenges for their practical application.

Another issue that yields further examination is the reason that the tested ML methods do not get the benefit of applying the same local regions as in LR. One possible answer tends towards the direction of the intrinsic way that RF and XGB are working by exploiting combinations of predictors within the feature space, thus better capturing the specific dynamics of a sub-region. This needs to be further examined.

Further work can also be performed towards the direction of:

- exploring advanced balancing techniques such as SMOTE [131] or RUSBoost [132] for local sampling considering the highly imbalancing nature of credit datasets [64, 93] where balancing may affect not only performance in terms of misclassification errors but also non-convergence errors when using local LR,
- usage of penalized methods such as LASSO or Ridge [113, 133],
- usage of different distance metrics (e.g., Manhattan or Mahalanobis) or even different algorithms for choosing local regions instead of the basic kNNs, such as Reduced Minority kNNs [95].

Appendix

Table 2 Comparison of different local region sizes (kNNs)

	AUC			H-Measure		
	LR-L_2k	**LR-L_4k**	**LR-L_6k**	**LR-L_2k**	**LR-L_4k**	**LR-L_6k**
2009-Q1	0.9100	0.9112	0.9134	0.5983	0.6003	0.6000
2009-Q2	0.9236	0.9265	0.9255	0.6276	0.6306	0.6267

Table 2 (continued)

	AUC			H-Measure		
	LR-L_2k	**LR-L_4k**	**LR-L_6k**	**LR-L_2k**	**LR-L_4k**	**LR-L_6k**
2009-Q3	0.9278	0.9302	0.9298	0.6395	0.6392	0.6329
2009-Q4	0.9212	0.9225	0.9224	0.6228	0.6240	0.6204
2010-Q1	0.9282	0.9284	0.9283	0.6400	0.6350	0.6328
2010-Q2	0.9269	0.9279	0.9294	0.6385	0.6382	0.6374
2010-Q3	0.9222	0.9272	0.9260	0.6206	0.6248	0.6193
2010-Q4	0.9254	0.9242	0.9227	0.6289	0.6194	0.6148
2011-Q1	0.9169	0.9146	0.9137	0.6075	0.6030	0.5953
2011-Q2	0.9123	0.9084	0.9096	0.5944	0.5833	0.5799
2011-Q3	0.9113	0.9031	0.9116	0.5942	0.5759	0.5894
2011-Q4	0.9129	0.9115	0.9166	0.6019	0.5991	0.6004
2012-Q1	0.9240	0.9223	0.9233	0.6246	0.6195	0.6183
2012-Q2	0.9256	0.9200	0.9231	0.6264	0.6129	0.6169
2012-Q3	0.9178	0.9110	0.9149	0.6176	0.6004	0.6037
2012-Q4	0.9171	0.9092	0.9138	0.6151	0.6024	0.6090
2013-Q1	0.9171	0.9161	0.9108	0.6083	0.6013	0.5916
2013-Q2	0.9185	0.9117	0.9071	0.6015	0.5852	0.5774
2013-Q3	0.9098	0.9018	0.8957	0.5840	0.5653	0.5538
2013-Q4	0.9235	0.9230	0.9212	0.6166	0.6081	0.5995
2014-Q1	0.9259	0.9252	0.9228	0.6304	0.6231	0.6144
2014-Q2	0.9235	0.9157	0.9146	0.6140	0.5913	0.5854
2014-Q3	0.9285	0.9301	0.9301	0.6440	0.6363	0.6313
2014-Q4	0.9286	0.9322	0.9350	0.6433	0.6426	0.6400
2015-Q1	0.9293	0.9315	0.9298	0.6462	0.6434	0.6317
2015-Q2	0.9327	0.9355	0.9364	0.6552	0.6480	0.6466
2015-Q3	0.9317	0.9310	0.9359	0.6614	0.6516	0.6503
2015-Q4	0.9314	0.9352	0.9364	0.6548	0.6550	0.6553
2016-Q1	0.9314	0.9353	0.9352	0.6570	0.6583	0.6573
2016-Q2	0.9216	0.9290	0.9324	0.6289	0.6299	0.6294
2016-Q3	0.9232	0.9321	0.9300	0.6370	0.6421	0.6410
2016-Q4	0.9407	0.9472	0.9484	0.6891	0.6896	0.6910
2017-Q1	0.9417	0.9449	0.9460	0.6949	0.6882	0.6822
2017-Q2	0.9402	0.9434	0.9446	0.6791	0.6757	0.6790
2017-Q3	0.9377	0.9380	0.9374	0.6699	0.6642	0.6565
2017-Q4	0.9402	0.9393	0.9397	0.6731	0.6606	0.6586
2018-Q1	0.9337	0.9367	0.9366	0.6693	0.6613	0.6558
2018-Q2	0.9351	0.9359	0.9397	0.6730	0.6624	0.6649
2018-Q3	0.9306	0.9347	0.9359	0.6549	0.6439	0.6393
2018-Q4	0.9239	0.9309	0.9333	0.6581	0.6548	0.6522
Mean	**0.9256**	**0.9259**	**0.9265**	**0.6360**	**0.6298**	**0.6270**
StdDev	**0.0086**	**0.0115**	**0.0118**	**0.0278**	**0.0303**	**0.0308**

LR logistic regression, *L* local classifier, *2 k* 2000, *4 k* 4000, *6 k* 6000 for kNN

Table 3 Local vs global classifiers (AUC metric)

	AUC metric					
	LR-L_2k	XGB-L_2k	RF-L_2k	LR-G	XGB-G	RF-G
2009-Q1*	0.91	0.9059	**0.9202**	0.8885	0.9158	0.9193
2009-Q2	0.9236	0.9174	**0.9267**	0.8806	0.9113	0.9213
2009-Q3	0.9278	0.9219	**0.9336**	0.8889	0.9159	0.9246
2009-Q4	0.9212	0.9218	**0.9305**	0.8784	0.917	0.9214
2010-Q1	0.9282	0.9235	**0.9335**	0.8829	0.9183	0.9251
2010-Q2	0.9269	0.9249	**0.9368**	0.8749	0.9208	0.9276
2010-Q3	0.9222	0.9238	**0.9322**	0.872	0.9118	0.9198
2010-Q4	0.9254	0.9201	**0.9298**	0.8619	0.9111	0.9169
2011-Q1*	0.9169	0.9141	0.9257	0.8701	0.9246	**0.9265**
2011-Q2	0.9123	0.9154	**0.9238**	0.8618	0.9222	0.9236
2011-Q3	0.9113	0.9115	**0.9232**	0.8599	0.9114	0.9168
2011-Q4	0.9129	0.9101	**0.9216**	0.8613	0.9166	0.9196
2012-Q1	0.924	0.9218	**0.9299**	0.8589	0.9219	0.9264
2012-Q2	0.9256	0.9214	**0.9312**	0.8587	0.9243	0.9275
2012-Q3	0.9178	0.9173	**0.9297**	0.8519	0.9209	0.9251
2012-Q4	0.9171	0.9176	**0.9267**	0.8389	0.9169	0.9197
2013-Q1*	0.9171	0.9128	0.9239	0.8591	0.9253	**0.9256**
2013-Q2	0.9185	0.9118	0.9233	0.8617	0.9237	**0.9248**
2013-Q3	0.9098	0.9059	**0.9154**	0.8516	0.9142	**0.9154**
2013-Q4	0.9235	0.9218	**0.9321**	0.8757	0.9271	0.9288
2014-Q1	0.9259	0.9236	**0.9366**	0.879	0.9299	0.9318
2014-Q2	0.9235	0.924	**0.9338**	0.8628	0.9265	0.9302
2014-Q3	0.9285	0.9337	**0.9416**	0.8722	0.9333	0.9392
2014-Q4	0.9286	0.9318	**0.9413**	0.8708	0.933	0.9369
2015-Q1*	0.9293	0.9286	0.9392	0.8878	0.9397	**0.9404**
2015-Q2	0.9327	0.9348	0.9448	0.8941	0.9452	**0.947**
2015-Q3	0.9317	0.9332	0.9419	0.8958	**0.9457**	0.9455
2015-Q4	0.9314	0.9307	0.9434	0.896	0.9426	**0.9442**
2016-Q1	0.9314	0.9338	0.9462	0.8975	0.9489	**0.9515**
2016-Q2	0.9216	0.9305	0.9418	0.898	0.9428	**0.9454**
2016-Q3	0.9232	0.9301	0.9439	0.8959	0.9437	**0.9473**
2016-Q4	0.9407	0.9453	0.9561	0.9082	0.954	**0.9566**
2017-Q1*	0.9417	0.9477	**0.958**	0.8725	0.9559	0.9571
2017-Q2	0.9402	0.9467	**0.9556**	0.8776	0.9518	0.9516
2017-Q3	0.9377	0.9416	**0.9506**	0.8676	0.9432	0.9426
2017-Q4	0.9402	0.9437	**0.9524**	0.8649	0.9439	0.9451
2018-Q1	0.9337	0.944	**0.9517**	0.8686	0.9462	0.946
2018-Q2	0.9351	0.9446	**0.9526**	0.8635	0.947	0.9491
2018-Q3	0.9306	0.9412	**0.9446**	0.8583	0.9402	0.9389
2018-Q4	0.9239	0.9362	**0.9389**	0.8455	0.9375	0.9352
Mean	**0.9256**	**0.9267**	**0.9366**	**0.8729**	**0.9306**	**0.9334**
StdDev	**0.0086**	**0.0118**	**0.0111**	**0.0161**	**0.0138**	**0.0123**

LR logistic regression, *RF* random forest, *XGB* gradient boosting, *L* local classifier, *G* global classifier, *2 k* 2000 for kNN

*training snapshot for global classifiers, bold indicate the best classifier for the specific snapshot

Table 4 Local vs global classifiers (H-Measure metric)

	H-Measure metric					
	LR-L_2k	**XGB-L_2k**	**RF-L_2k**	**LR-G**	**XGB-G**	**RF-G**
2009-Q1*	0.5983	0.5936	**0.6224**	0.5485	0.6005	0.6151
2009-Q2	0.6276	0.6156	**0.6412**	0.5590	0.6109	0.6337
2009-Q3	0.6395	0.6347	**0.6607**	0.5695	0.6168	0.6418
2009-Q4	0.6228	0.6266	**0.6475**	0.5534	0.6100	0.6297
2010-Q1	0.6400	0.6369	**0.6620**	0.5607	0.6188	0.6396
2010-Q2	0.6385	0.6332	**0.6639**	0.5525	0.6231	0.6438
2010-Q3	0.6206	0.6257	**0.6474**	0.5281	0.5965	0.6230
2010-Q4	0.6289	0.6237	**0.6543**	0.5156	0.5931	0.6217
2011-Q1*	0.6075	0.6064	**0.6330**	0.4887	0.6215	0.6316
2011-Q2	0.5944	0.6023	0.6243	0.4779	0.6169	**0.6244**
2011-Q3	0.5942	0.5866	**0.6182**	0.4839	0.5840	0.6113
2011-Q4	0.6019	0.5964	**0.6230**	0.4809	0.5953	0.6155
2012-Q1	0.6246	0.6191	**0.6448**	0.4726	0.6203	0.6387
2012-Q2	0.6264	0.6180	**0.6463**	0.4842	0.6230	0.6405
2012-Q3	0.6176	0.6168	**0.6464**	0.4726	0.6225	0.6417
2012-Q4	0.6151	0.6193	**0.6416**	0.4555	0.6089	0.6275
2013-Q1*	0.6083	0.6066	0.6374	0.5005	0.6265	**0.6378**
2013-Q2	0.6015	0.6019	0.6267	0.4867	0.6174	**0.6285**
2013-Q3	0.5840	0.5865	**0.6090**	0.4806	0.5934	0.6055
2013-Q4	0.6166	0.6274	**0.6494**	0.5034	0.6244	0.6377
2014-Q1	0.6304	0.6368	**0.6675**	0.5188	0.6347	0.6476
2014-Q2	0.6140	0.6299	**0.6553**	0.4977	0.6216	0.6418
2014-Q3	0.6440	0.6597	**0.6801**	0.5236	0.6433	0.6677
2014-Q4	0.6433	0.6544	**0.6813**	0.5181	0.6423	0.6587
2015-Q1*	0.6462	0.6539	**0.6778**	0.4916	0.6703	0.6766
2015-Q2	0.6552	0.6661	0.6911	0.4982	0.6865	**0.6940**
2015-Q3	0.6614	0.6784	**0.6944**	0.5042	0.6870	0.6909
2015-Q4	0.6548	0.6576	**0.6899**	0.5072	0.6732	0.6809
2016-Q1	0.6570	0.6787	**0.7051**	0.5005	0.6918	0.7022
2016-Q2	0.6289	0.6562	0.6865	0.5053	0.6769	**0.6886**
2016-Q3	0.6370	0.6600	**0.6897**	0.4939	0.6776	0.6886
2016-Q4	0.6891	0.7052	**0.7307**	0.5385	0.7117	0.7241
2017-Q1*	0.6949	0.7172	**0.7361**	0.4776	0.7225	0.7341
2017-Q2	0.6791	0.6947	**0.7184**	0.4846	0.7041	0.7176
2017-Q3	0.6699	0.6906	**0.7122**	0.4779	0.6816	0.6916
2017-Q4	0.6731	0.6952	**0.7165**	0.4702	0.6769	0.6952
2018-Q1	0.6693	0.7039	**0.7220**	0.4669	0.6883	0.7026
2018-Q2	0.6730	0.6981	**0.7195**	0.4435	0.6872	0.7002
2018-Q3	0.6549	0.6882	**0.7028**	0.4373	0.6707	0.6806
2018-Q4	0.6581	0.6767	**0.7031**	0.4192	0.6670	0.6778
Mean	**0.6360**	**0.6445**	**0.6695**	**0.4987**	**0.6435**	**0.6588**
StdDev	**0.0278**	**0.0368**	**0.0351**	**0.0344**	**0.0382**	**0.0348**

LR logistic regression, *RF* random forest, *XGB* gradient boosting, *L* local classifier, *G* global classifier, *2 k* 2000 for kNN

*training snapshot for global classifiers, bold indicate the best classifier for the specific snapshot

Table 5 kNNs vs random sub-sampling

	AUC			**H-Measure**		
	LR-L_2k	**LR-G***	**LR-L-rnd**	**LR-L_2k**	**LR-G***	**LR-L-rnd**
2009-Q1*	0.9100	0.8885	0.8872	0.5983	0.5485	0.5499
2009-Q2	0.9236	0.8806	0.8818	0.6276	0.5590	0.5576
2009-Q3	0.9278	0.8889	0.8948	0.6395	0.5695	0.567
2009-Q4	0.9212	0.8784	0.8859	0.6228	0.5534	0.553
2010-Q1	0.9282	0.8829	0.8913	0.6400	0.5607	0.5543
2010-Q2	0.9269	0.8749	0.858	0.6385	0.5525	0.5342
2010-Q3	0.9222	0.8720	0.8156	0.6206	0.5281	0.4827
2010-Q4	0.9254	0.8619	0.8043	0.6289	0.5156	0.4644
2011-Q1*	0.9169	0.8701	0.8223	0.6075	0.4887	0.4725
2011-Q2	0.9123	0.8618	0.8186	0.5944	0.4779	0.4607
2011-Q3	0.9113	0.8599	0.8114	0.5942	0.4839	0.4607
2011-Q4	0.9129	0.8613	0.8366	0.6019	0.4809	0.4839
2012-Q1	0.9240	0.8589	0.8389	0.6246	0.4726	0.4904
2012-Q2	0.9256	0.8587	0.8523	0.6264	0.4842	0.5015
2012-Q3	0.9178	0.8519	0.8539	0.6176	0.4726	0.4866
2012-Q4	0.9171	0.8389	0.8571	0.6151	0.4555	0.4852
2013-Q1*	0.9171	0.8591	0.8525	0.6083	0.5005	0.4721
2013-Q2	0.9185	0.8617	0.8618	0.6015	0.4867	0.4854
2013-Q3	0.9098	0.8516	0.8494	0.5840	0.4806	0.4743
2013-Q4	0.9235	0.8757	0.8772	0.6166	0.5034	0.5317
2014-Q1	0.9259	0.8790	0.8792	0.6304	0.5188	0.5284
2014-Q2	0.9235	0.8628	0.8681	0.6140	0.4977	0.5055
2014-Q3	0.9285	0.8722	0.8823	0.6440	0.5236	0.5316
2014-Q4	0.9286	0.8708	0.8758	0.6433	0.5181	0.5177
2015-Q1*	0.9293	0.8878	0.8789	0.6462	0.4916	0.5162
2015-Q2	0.9327	0.8941	0.8758	0.6552	0.4982	0.5169
2015-Q3	0.9317	0.8958	0.8809	0.6614	0.5042	0.5191
2015-Q4	0.9314	0.8960	0.8686	0.6548	0.5072	0.502
2016-Q1	0.9314	0.8975	0.879	0.6570	0.5005	0.524
2016-Q2	0.9216	0.8980	0.8787	0.6289	0.5053	0.5079
2016-Q3	0.9232	0.8959	0.8811	0.6370	0.4939	0.5126
2016-Q4	0.9407	0.9082	0.8935	0.6891	0.5385	0.5326
2017-Q1*	0.9417	0.8725	0.8929	0.6949	0.4776	0.536
2017-Q2	0.9402	0.8776	0.8948	0.6791	0.4846	0.5359
2017-Q3	0.9377	0.8676	0.8872	0.6699	0.4779	0.5329
2017-Q4	0.9402	0.8649	0.8877	0.6731	0.4702	0.5299
2018-Q1	0.9337	0.8686	0.8868	0.6693	0.4669	0.5377
2018-Q2	0.9351	0.8635	0.8865	0.6730	0.4435	0.547
2018-Q3	0.9306	0.8583	0.8872	0.6549	0.4373	0.5524
2018-Q4	0.9239	0.8455	0.8787	0.6581	0.4192	0.5513
Mean	**0.9256**	**0.8729**	**0.8674**	**0.6360**	**0.4987**	**0.5151**
StdDev	**0.0086**	**0.0161**	**0.0253**	**0.0278**	**0.0344**	**0.0301**

LR logistic regression, *L* local classifier, *G* global classifier, *2 k* 2000 for kNN, *rnd* random

*training snapshot for global classifiers

Author Contribution DN implemented the models and the computational framework, analyzed the results, and prepared the manuscript. MD contributed to the design of the experimental analysis and the writing of the manuscript. All authors provided critical feedback and helped shape the research, analysis, and the manuscript.

Funding Open access funding provided by HEAL-Link Greece.

Data Availability Data subject to third party restrictions.

Code Availability Not applicable.

Declarations

Ethics Approval Not applicable

Consent to Participate Not applicable.

Consent for Publication Not applicable.

Conflicts of Interest The authors declare no competing interests.

References

1. Barci G, Andreeva G, Bouyon S (2019) "Data sharing in credit markets: does comprehensiveness matter?", European Credit Research Institute, Research Report no. 23, available at: https://bit.ly/3xfiW3v
2. Besanko D, Thakor AV (1987) Competitive equilibrium in the credit market under asymmetric information. Journal of Economic Theory 42(1):167–182
3. Jappelli T, Pagano M (1993) Information sharing in credit markets. J Financ 48(5):1693–1718
4. Morscher C, Horsch A, Stephan J (2017) Credit information sharing and its link to financial inclusion and financial intermediation. Financial Markets, Institutions and Risks 1(3):22–33
5. Stiglitz JE, Weiss A (1981) Credit rationing in markets with imperfect information. Am Econ Rev 71(3):393–410
6. Breeden J, Thomas L, McDonald J III (2007) Stress testing retail load portfolios with dual-time dynamics. Journal of Risk Model Validation 2(2):1–19
7. Hand DJ, Henley WE (1997) Statistical classification methods in consumer credit scoring: a review. J R Stat Soc A Stat Soc 160(3):523–541
8. Thomas LC, Malik M (2010) Comparison of credit risk models for portfolios of retail loans based on behavioral scores. In: Rausch D, Scheule H (eds) Model Risk in Financial Crises. Risk Books, pp 209–232
9. Durand D (1941) Credit-rating formulae. In Risk Elements in Consumer Installment Financing 83–91. NBER
10. Anderson R (2007) The credit scoring toolkit: theory and practice for retail credit risk management and decision automation. Oxford University Press

11. Thomas LC, Edelman DB, Crook JN (2002) Credit scoring & its applications (monographs on mathematical modeling and computation) (1st edition). Soc Ind Appl Math
12. Adams, NM, Tasoulis DK, Anagnostopoulos C, Hand DJ (2010) Temporally-adaptive linear classification for handling population drift in credit scoring. Lechevallier, Y. αnd Saporta.(Eds), COMPSTAT2010, Proceedings of the 19th International Conference on Computational Statistics 167–176
13. Gama J, Medas P, Castillo G, Rodrigues P (2004) Learning with drift detection. In Advances in Artificial Intelligence–SBIA 2004 286–295. Springer
14. Gama J, Žliobaite Ie, Bifet A, Pechenizkiy M, Bouchachia A (2014) A survey on concept drift adaptation. ACM Comput. Surv 46(4):44:1–44:37
15. Klinkenberg R (2004) Learning drifting concepts: example selection vs. example weighting. Intell Data Anal 8(3):281–300
16. Žliobaitė I, Pechenizkiy M, Gama J (2016) An overview of concept drift applications. In N. Japkowicz & J. Stefanowski (Eds.), *Big Data Analysis: New Algorithms for a New Society* (Vol. 16, pp. 91–114). Springer International Publishing
17. Jung KM, Thomas LC, So MC (2015) When to rebuild or when to adjust scorecards. Journal of the Operational Research Society 66(10):1656–1668
18. Siddiqi N (2005) Credit risk scorecards: developing and implementing intelligent credit scoring. Wiley, New York
19. Rona-Tas A, Hiss S (2008) Consumer and corporate credit ratings and the subprime crisis in the US with some lessons for Germany. SCHUFA, Wiesbaden
20. Ashcraft AB, Schuermann T (2008) Understanding the securitization of subprime mortgage credit. Foundations and Trends® in Finance 2(3):191–309
21. Demyanyk Y, Van Hemert O (2011) Understanding the subprime mortgage crisis. Review of Financial Studies 24(6):1848–1880
22. Breeden J (2014) Reinventing retail lending analytics—2nd impression. Risk Books
23. Avery RB, Bostic RW, Calem PS, Canner GB (2000) Credit scoring: statistical issues and evidence from credit bureau files. Real Estate Economics 28(3):523–547
24. Anderson R (2022) Credit intelligence and modelling: many paths through the forest. Oxfrod University Press
25. Bijak K, Thomas LC (2012) Does segmentation always improve model performance in credit scoring? Expert Syst Appl 39(3):2433–2442
26. Hand DJ (2009) Measuring classifier performance: a coherent alternative to the area under the ROC curve. Mach Learn 77(1):103–123
27. Lessmann S, Lyn C, Thomas Hsin-Vonn Seow, Baesens B (2013) Benchmarking state-of-the-art classification algorithms for credit scoring: A ten-year update. Credit Scoring and Credit Control XIII
28. Jamain A, Hand DJ (2009) Where are the large and difficult datasets? Adv Data Anal Classif 3(1):25–38
29. Perlich C, Provost F, Simonoff JS (2003) Tree induction vs. logistic regression: a learning-curve analysis. J Mach Learn Res 4:211–255
30. Addo P, Guegan D, Hassani B (2018) Credit risk analysis using machine and deep learning models. Risks 6(2):38
31. Albanesi S, Vamossy DF (2019) Predicting consumer default: a deep learning approach (Working Paper No. 26165; Working Paper Series). Nat Bur Econom Res
32. Alonso A, Carbó JM (2020) Machine learning in credit risk: measuring the dilemma between prediction and supervisory cost. Banco de España Working Paper No. 2032, available at: https://ssrn.com/abstract=3724374
33. Gunnarsson BR, Broucke S, Baesens B, Óskarsdóttir M, Lemahieu W (2021) Deep learning for credit scoring: do or don't? Eur J Oper Res 295(1):292–305
34. Hamori S, Kawai M, Kume T, Murakami Y, Watanabe C (2018) Ensemble learning or deep learning? Application to default risk analysis. Journal of Risk and Financial Management 11(1):12
35. Marceau L, Qiu L, Vandewiele N, Charton E (2019) A comparison of deep learning performances with others machine learning algorithms on credit scoring unbalanced data. ArXiv:1907.12363
36. Petropoulos A, Siakoulis V, Stavroulakis E, Klamargias A (2019) A robust machine learning approach for credit risk analysis of large loan level datasets using deep learning and extreme gradient boosting. IFC Bulletins chapters, in: Bank for International Settlements (ed.), The use

of big data analytics and artificial intelligence in central banking, volume 50, Bank for International Settlements

37. Sirignano J, Cont R (2018) Universal features of price formation in financial markets: perspectives from deep learning. Quantitative Finance 19(9):1449–1459
38. Sirignano J, Sadhwani A, Giesecke K (2016) Deep learning for mortgage risk. Available at SSRN 2799443. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2799443
39. Bussmann N, Giudici P, Marinelli D, Papenbrock J (2020) Explainable AI in fintech risk management. Frontiers in Artificial Intelligence 3:26
40. Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D (2018) A survey of methods for explaining black box models. ACM Computing Surveys (CSUR) 51(5):1–42
41. Hardt M, Price E, Srebro N (2016) Equality of opportunity in supervised learning. Adv Neutral Inf Proces Syst 29
42. Suresh H, Guttag JV (2019) A framework for understanding unintended consequences of machine learning. ArXiv Preprint https://arxiv.org/abs/1901.10002
43. Gilpin LH, Bau D, Yuan BZ, Bajwa A, Specter M, Kagal L (2018) Explaining explanations: an overview of interpretability of machine learning. In 2018 IEEE 5th International Conference on data science and advanced analytics (DSAA) 80–89. IEEE
44. Zafar MB, Valera I, Rodriguez MG, Gummadi KP (2019) Fairness constraints: mechanisms for fair classification. J Mach Learn Res 20(75):1–42
45. Aggarwal N (2021) The norms of algorithmic credit scoring. The Cambridge Law Journal 80(1):42–73
46. Hurlin C, Pérignon C, Saurin S (2021) The fairness of credit scoring models (SSRN Scholarly Paper ID 3785882). Soc Sci Res Net
47. Kozodoi N, Jacob J, Lessmann S (2022) Fairness in credit scoring: assessment, implementation and profit implications. Eur J Oper Res 297(3):1083–1094
48. Aggarwal, C (2014) Instance-based learning: a survey. In Charu Aggarwal (Ed), Data classification: Algoth Appl CRC Press
49. Aha DW, Kibler D, Albert MK (1991) Instance-based learning algorithms. Mach Learn 6(1):37–66
50. Bontempi G, Bersini H, Birattari M (2001) The local paradigm for modeling and control: From neuro-fuzzy to lazy learning. Fuzzy Sets Syst 121(1):59–72
51. Bontempi G, Birattari M, Bersini H (2002) Lazy learning: a logical method for supervised learning. In: Jain LC, Kacprzyk J (eds) New learning Paradigms in Soft Computing. Springer, Heidelberg, pp 97–136
52. Bottou L, Vapnik V (1992) Local learning algorithms. Neural Comput 4(6):888–900
53. Atkeson CG, Moore AW, Schaal S (1997) Locally weighted learning. Artif Intell Rev 11(1–5):11–73
54. Domeniconi C, Peng J, Gunopulos D (2002) Locally adaptive metric nearest-neighbor classification. IEEE Trans Pattern Anal Mach Intell 24(9):1281–1285
55. Zhang H, Berg AC, Maire M, Malik J (2006) SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). 2:126–2136
56. Aamodt A, Plaza E (1994) Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Commun 7(1):39–59
57. Jo H, Han I, Lee H (1997) Bankruptcy prediction using case-based reasoning, neural networks, and discriminant analysis. Expert Syst Appl 13(2):97–108
58. Vukovic S, Delibasic B, Uzelac A, Suknovic M (2012) A case-based reasoning model that uses preference theory functions for credit scoring. Expert Syst Appl 39(9):8389–8395
59. Xu R, Nettleton D, Nordman DJ (2016) Case-specific random forests. J Comput Graph Stat 25(1):49–65
60. Garcia S, Derrac J, Cano JR, Herrera F (2012) Prototype selection for nearest neighbor classification: taxonomy and empirical study. IEEE Trans Pattern Anal Mach Intell 34(3):417–435
61. Leyva E, González A, Pérez R (2015) Three new instance selection methods based on local sets: a comparative study with several approaches from a bi-objective perspective. Pattern Recogn 48(4):1523–1537
62. Olvera-López JA, Carrasco-Ochoa JA, Martínez-Trinidad JF, Kittler J (2010) A review of instance selection methods. Artif Intell Rev 34(2):133–143
63. de Haro-García A, Cerruela-García G, García-Pedrajas N (2019) Instance selection based on boosting for instance-based learners. Pattern Recogn 96:106959

64. Bischl B, Kühn T, Szepannek G (2016) On class imbalance correction for classification algorithms in credit scoring. In: Lübbecke M, Koster A, Letmathe P, Madlener R, Peis B, Walther G (eds) Operations Research Proceedings 2014. Springer, Cham, pp 37–43
65. Kuncheva LI, Arnaiz-González Á, Díez-Pastor J-F, Gunn IAD (2019) Instance selection improves geometric mean accuracy: a study on imbalanced data classification. Progress in Artificial Intelligence 8(2):215–228
66. More A (2016) Survey of resampling techniques for improving classification performance in unbalanced datasets. https://arxiv.org/abs/1608.06048
67. Crone SF, Finlay S (2012) Instance sampling in credit scoring: an empirical study of sample size and balancing. Int J Forecast 28(1):224–238
68. Cleveland WS, Devlin SJ, Grosse E (1988) Regression by local fitting: methods, properties, and computational algorithms. J Econom 37(1):87–114
69. Loader C (1999) Local regression and likelihood. Springer Science & Business Media
70. Schaal S, Atkeson CG (1998) Constructive incremental learning from only local information. Neural Comput 10(8):2047–2084
71. Nadaraya EA (1964) On estimating regression. Theory of Probability & Its Applications 9(1):141–142
72. Watson GS (1964) Smooth regression analysis. Sankhyā: Ind J Stat Ser A 359–372
73. Schwarz A, Arminger G (2005) Credit scoring using global and local statistical models. In: Weihs C, Gaul W (eds) Classification—The Ubiquitous Challenge. Springer, Berlin Heidelberg, pp 442–449
74. Li F-C (2009) The hybrid credit scoring strategies based on KNN classifier. Sixth International Conference on Fuzzy Systems and Knowledge Discovery 2009:330–334
75. Harris T (2015) Credit scoring using the clustered support vector machine. Expert Syst Appl 42(2):741–750
76. Liu Z, Pan S (2018) Fuzzy-rough instance selection combined with effective classifiers in credit scoring. Neural Process Lett 47(1):193–202
77. Guo Y, Zhou W, Luo C, Liu C, Xiong H (2016) Instance-based credit risk assessment for investment decisions in P2P lending. Eur J Oper Res 249(2):417–426
78. Britto AS, Sabourin R, Oliveira LES (2014) Dynamic selection of classifiers—a comprehensive review. Pattern Recogn 47(11):3665–3680
79. Dietterich TG (2000) Ensemble methods in machine learning. In: Multiple Classifier Systems. MCS 2000. Lect Notes Comput Sci 1857:1–15. Springer, Berlin, Heidelberg
80. Kuncheva LI (2004) Classifier ensembles for changing environments. In F. Roli J, Kittler, T Windeatt (eds) Multiple Classifier Systems (Vol. 3077, pp. 1–15). Springer Berlin Heidelberg
81. Kuncheva LI (2008) Classifier ensembles for detecting concept change in streaming data: Overview and perspectives. Proceedings of the 2nd Workshop SUEMA, 2008 5–10
82. Cruz RM. O, Cavalcanti GDC, Ren TI (2011) A method for dynamic ensemble selection based on a filter and an adaptive distance to improve the quality of the regions of competence. The 2011 International Joint Conference on Neural Networks 1126–1133
83. Cruz RM O, Zakane HH, Sabourin R, Cavalcanti GDC (2017) Dynamic ensemble selection VS K-NN: why and when dynamic selection obtains higher classification performance? 2017Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA) 1–6
84. García V, Marqués AI, Sánchez JS (2012) Improving risk predictions by preprocessing imbalanced credit data. In T. Huang, Z. Zeng, C. Li, & C. S. Leung (eds) Neural Information Processing 7664:68–75. Springer Berlin Heidelberg
85. García V, Marqués AI, Sánchez JS (2019) Exploring the synergetic effects of sample types on the performance of ensembles for credit risk and corporate bankruptcy prediction. Information Fusion 47:88–101
86. García V, Sánchez JS, Ochoa-Ortiz A, López-Najera A (2019) Instance selection for the nearest neighbor classifier: connecting the performance to the underlying data structure. In: Morales A, Fierrez J, Sánchez JS, Ribeiro B (eds) Pattern Recognition and Image Analysis. Springer International Publishing, pp 249–256
87. Kuncheva LI (2000) Clustering-and-selection model for classifier combination. KES'2000. Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies. Proceedings (Cat. No.00TH8516), 1:185–188
88. Soares RGF, Santana A, Canuto AMP, de Souto MCP (2006) Using accuracy and diversity to select classifiers to build ensembles. Proc Int Jt Conf Neural Netw 1310–1316

89. Abellán J, Castellano JG (2017) A comparative study on base classifiers in ensemble methods for credit scoring. Expert Syst Appl 73:1–10
90. Ala'raj M, Abbod MF (2016) Classifiers consensus system approach for credit scoring. Knowl-Based Syst 104:89–105
91. Ala'raj M, Abbod MF (2016) A new hybrid ensemble credit scoring model based on classifiers consensus system approach. Expert Syst Appl 64:36–55
92. Feng X, Xiao Z, Zhong B, Qiu J, Dong Y (2018) Dynamic ensemble classification for credit scoring using soft probability. Appl Soft Comput 65:139–151
93. He H, Zhang W, Zhang S (2018) A novel ensemble method for credit scoring: adaption of different imbalance ratios. Expert Syst Appl 98:105–117
94. Lessmann S, Baesens B, Seow H-V, Thomas LC (2015) Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. Eur J Oper Res 247(1):124–136
95. Melo Junior L, Nardini FM, Renso C, Trani R, Macedo JA (2020) A novel approach to define the local region of dynamic selection techniques in imbalanced credit scoring problems. Expert Syst Appl 152:113351
96. Marqués AI, García V, Sánchez JS (2012) On the suitability of resampling techniques for the class imbalance problem in credit scoring. J Oper Res Soc 64(7):1060–1070
97. Zhang H, Liu Q (2019) Online learning method for drift and imbalance problem in client credit assessment. Symmetry 11(7):890
98. Lasota T, Londzin B, Telec Z, Trawiński B (2014) Comparison of ensemble approaches: mixture of experts and AdaBoost for a regression problem. In N. T. Nguyen B, Attachoo B, Trawiński K, Somboonviwat (eds), *Intelligent Information and Database Systems* (Vol. 8398, pp. 100–109). Springer International Publishing
99. Masoudnia S, Ebrahimpour R (2014) Mixture of experts: a literature survey. Artif Intell Rev 42(2):275–293
100. Xu L, Amari S (2009) Combining classifiers and learning mixture-of-experts. In: Dopico JRD, Dorado J, Pazos A (eds) Encyclopedia of artificial intelligence. IGI Global, Hershey, PA, pp 318–326
101. Titsias MK, Likas A (2002) Mixture of experts classification using a hierarchical mixture model. Neural Comput 14(9):2221–2244
102. Jacobs RA, Jordan MI, Nowlan SJ, Hinton GE (1991) Adaptive mixtures of local experts. Neural Comput 3(1):79–87
103. Cruz RMO, Sabourin R, Cavalcanti GDC (2018) Dynamic classifier selection: Recent advances and perspectives. Information Fusion 41:195–216
104. Liang T, Zeng G, Zhong Q, Chi J, Feng J, Ao X, Tang J (2021) Credit risk and limits forecasting in e-commerce consumer lending service via multi-view-aware mixture-of-experts Nets. Proceedings of the 14th ACM International Conference on Web Search and Data Mining, 229–237
105. West D (2000) Neural network credit scoring models. Comput Oper Res 27(11–12):1131–1152
106. Mays E (2005) Handbook of credit scoring. Publishers Group Uk
107. Kennedy K, Mac Namee B, Delany SJ, O'Sullivan M, Watson N (2013) A window of opportunity: assessing behavioural scoring. Expert Syst Appl 40(4):1372–1380
108. Branco P, Torgo L, Ribeiro RP (2016) A survey of predictive modeling on imbalanced domains. ACM Computing Surveys (CSUR) 49(2):1–50
109. Ganganwar V (2012) An overview of classification algorithms for imbalanced datasets. International Journal of Emerging Technology and Advanced Engineering 2(4):42–47
110. Kaur H, Pannu HS, Malhi AK (2019) A systematic review on imbalanced data challenges in machine learning: applications and solutions. ACM Comput Surv 52(4):1–36
111. Rahman MM, Davis DN (2013) Addressing the class imbalance problem in medical datasets. Int J Mach Learn Comput 224–228
112. Sun Y, Wong AKC, Kamel MS (2009) Classification of imbalanced data: a review. Int J Pattern Recognit Artif Intell 23(04):687–719
113. Wang Q, Luo Z, Huang J, Feng Y, Liu Z (2017) A novel ensemble method for imbalanced data learning: bagging of extrapolation-SMOTE SVM. Comput Intell Neurosci 2017:1–11
114. Wang S, Minku LL, Yao X (2018) A systematic study of online class imbalance learning with concept drift. IEEE Transactions on Neural Networks and Learning Systems 29(10):4802–4821
115. Brown I, Mues C (2012) An experimental comparison of classification algorithms for imbalanced credit scoring data sets. Expert Syst Appl 39(3):3446–3453
116. Lewis EM (1992) An Introduction to Credit Scoring (2nd ed edition). Fair, Isaac and Co

117. Finlay S (2010) Credit scoring, response modelling and insurance rating. Palgrave Macmillan UK
118. Japkowicz N, Shah M (2011) Evaluating learning algorithms: a classification perspective. Cambridge University Press
119. Luque A, Carrasco A, Martín A, de las Heras, A. (2019) The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recogn 91:216–231
120. Parker C (2011) An analysis of performance measures for binary classifiers. 2011 IEEE 11th International Conference on Data Mining, 517–526
121. Valverde-Albacete FJ, Peláez-Moreno C (2014) 100% classification accuracy considered harmful: the normalized information transfer factor explains the accuracy paradox. PLoS ONE 9(1):e84217
122. Hand DJ, Anagnostopoulos C (2013) When is the area under the receiver operating characteristic curve an appropriate measure of classifier performance? Pattern Recogn Lett 34(5):492–495
123. Hand DJ, Anagnostopoulos C (2021) Notes on the H-measure of classifier performance. Adv Data Anal Classif 1–16
124. Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
125. García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Inf Sci 180(10):2044–2064
126. Garcıa S, Herrera F (2008) An extension on “statistical comparisons of classifiers over multiple data sets” for all Pairwise Comparisons. J Mach Learn Res 9:18
127. Kvamme H, Sellereite N, Aas K, Sjursen S (2018) Predicting mortgage default using convolutional neural networks. Expert Syst Appl 102:207–217
128. Guégan D, Hassani B (2018) Regulatory learning: how to supervise machine learning models? An application to credit scoring. The Journal of Finance and Data Science 4(3):157–171
129. Kiritz N, Sarfati P (2018) Supervisory guidance on model risk management (SR 11–7) versus enterprise-wide model risk management for deposit-taking institutions (E-23): a detailed comparative analysis. Available at SSRN 3332484
130. Morini M (2011) Understanding and managing model risk: a practical guide for quants, traders and validators. John Wiley & Sons
131. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16:321–357
132. Seiffert C, Khoshgoftaar TM, Van Hulse J, Napolitano A (2010) RUSBoost: a hybrid approach to alleviating class imbalance. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 40(1):185–197
133. Wang H, Xu Q, Zhou L (2015) Large unbalanced credit scoring using lasso-logistic regression ensemble. PLoS ONE 10(2):e0117844